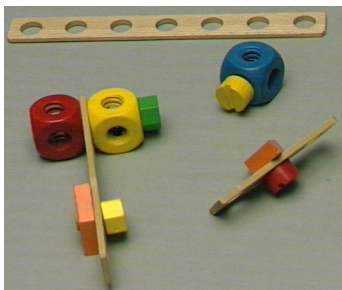
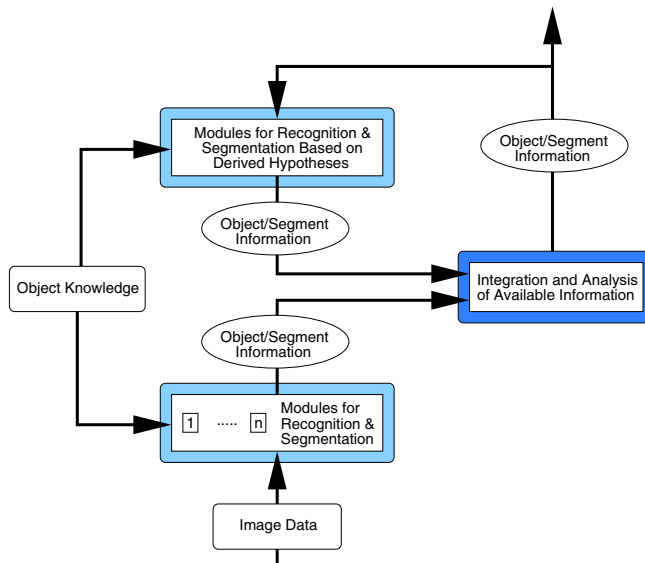


A Framework for Integrating Object Recognition Strategies

Elke Braun



Dipl.-Phys. Elke Braun
AG Angewandte Informatik
Technische Fakultät
Universität Bielefeld
email: ebraun@techfak.uni-bielefeld.de

Genehmigte Dissertation zur Erlangung des akademischen Grades
Doktorin der Ingenieurwissenschaften (Dr.-Ing.).
Der Technischen Fakultät der Universität Bielefeld
am 8. Dezember 2005 vorgelegt von Elke Braun,
am 27. April 2006 verteidigt und genehmigt.

Gutachter:

Prof. Dr. Gerhard Sagerer, Universität Bielefeld
Prof. Dr. Sven Dickinson, University of Toronto

Prüfungsausschuss:

Prof. Dr. Franz Kummert, Universität Bielefeld
Prof. Dr. Gerhard Sagerer, Universität Bielefeld
Prof. Dr. Sven Dickinson, University of Toronto
Dr. Sven Wachsmuth, Universität Bielefeld

Many Thanks

While preparing this thesis more people than I can mention here accompanied and supported me whom I want to thank a lot.

Special thanks go to my supervisor Gerhard Sagerer who gave me the chance of working in the area of scientific computer science and manages the difficult tightrope walk between guiding the process and giving the freedom for generating own ideas. I especially appreciate his way of integrating scientific work with the circumstances of real life, as in my case, the birth of two children.

The spontaneous positive reaction of Sven Dickinson from University of Toronto concerning the contents of the thesis greatly facilitated the process of completing it. Thanks a lot for the enthusiastic and detailed review that encourages me very much.

This work was embedded within the SFB 360 'Situating Artificial Communicators' at Bielefeld University. The SFB provides the conditions of many people working at related problems which results in productive discussions and a great pool of ideas as well as implementations. Thank you, Gunther Heidemann and Helge Ritter, for the cooperation within 'my' SFB projects. Thank you, Gunther Heidemann, Christian Bauckhage, and Jannik Fritsch for providing my integration work with valuable suggestions and real modules implementing segmentation and object recognition approaches.

Many thanks to all members of the working group 'Applied Computer Science' for generating an excellent climate for working. This includes the willingness to scientific discussions as well as having an open ear for personal affairs. I want to mention, especially, Lisabeth van Iersel, Franz Kummert and my office mates Daniel Schlüter and Jannik Fritsch. I am not sure whether this thesis would have been finished without their advice and support.

My parents and my brother always support and help us concerning all things that are beyond the contents of this thesis without discussing or demanding something in return. I do not take this for granted, thank you very much.

Finally, special thanks go to 'my men', my husband, Jens, and my children, Erik and Till. The basic requirement for finishing this thesis was Jens' support and especially his willingness to divide the family work. All the three proved a lot of patience, especially during the last phase of completing and defending this thesis.

A Framework for Integrating Object Recognition Strategies

Dissertation zur Erlangung des Grades
Doktorin der Ingenieurwissenschaften (Dr.-Ing.)

der Technischen Fakultät der Universität Bielefeld

vorgelegt von

Elke Braun

Dezember 2005

Contents

1	Introduction	1
1.1	Combining Simple Methods	1
1.2	Recognition Strategies	2
1.3	Object Context Knowledge	3
1.4	Proposed Integrating Framework	4
1.5	Outline	6
2	Image Segmentation	7
2.1	Basic Concepts	7
2.1.1	Image Data Driven Features and their Distances	8
2.1.2	Approaches for Segment Generation	10
2.1.3	Integrating Task Specific Knowledge to Segmentation	12
2.1.4	Model Based Top Down Segmentation	13
2.2	Choice of Color Image Segmentation Algorithms	14
2.2.1	Hierarchical Region Growing: Color Structure Code	15
2.2.2	Graph Based Segmentation Using the Local Variation Criterion	18
2.2.3	Feature Clustering Using Mean-Shift Algorithm	21
2.2.4	Image Segmentation by Pixel Color Classification	25
2.2.5	Perceptual Grouping of Contour Information	26
2.3	Summary	29
3	Object Recognition	31
3.1	Basic Concepts	31
3.1.1	Recognition Task: Detection, Segmentation, and Labeling	31
3.1.2	General Components of Object Recognition Systems	32
3.1.3	Object Knowledge Representation	34
3.1.4	Classifier Combination	38
3.2	Object Recognition Systems	43
3.2.1	Hybrid System Integrating Neural and Semantic Networks	44
3.2.2	Combining Region Based Classifiers for Recognition	47
3.2.3	Appearance Based Recognition System	51
3.2.4	Shape Based Recognition	57
3.3	Additional Information: Context Based Systems	60
3.3.1	Semantic Region Growing	62
3.3.2	Recognition based on Assemblage Rules	62
3.3.3	Monitoring the Assembly Construction Process	65
3.4	Summary	67

4	The Integrating Framework	71
4.1	Integrated System Architecture and Component Interaction	72
4.2	Common Representation of Segment and Object Information	73
4.2.1	Exemplary Effects in Data Driven Segmentation	74
4.2.2	Generating a Hierarchical Representation of Segmentation Results	77
4.2.3	Image Data Based Object Information	85
4.2.4	Summarized Characteristics of the Common Representation	89
4.3	Generating Hypotheses by Analyzing the Hierarchical Representation . . .	91
4.3.1	Object Labels from Probabilistic Integration	91
4.3.2	Selecting Hypotheses for Object Regions	94
4.3.3	Information Content of the Competing Hypotheses	102
4.4	Additional Information Dependent on Preliminary Hypotheses	102
4.4.1	Localizing Object Information from Expectation	104
4.4.2	Integrating Additional Information	107
4.5	Evaluation of Competing Hypotheses	108
4.5.1	Independent Evaluation Criteria	108
4.5.2	Combination of Individual Criteria	110
4.6	Summary	111
5	Evaluation of Realized Integrated Systems	113
5.1	Realized Systems and Evaluation Conditions	114
5.1.1	Components of the Realized Integrated Recognition Systems . . .	114
5.1.2	Test Sets and Evaluation Guidelines	115
5.2	The Integration of Segment Information	116
5.2.1	Evaluation Strategy	116
5.2.2	baufix [®] Task	118
5.2.3	Office Environment	122
5.2.4	Conclusions from Integrating Segment Information	125
5.3	Independent Image Data Based Object Information	125
5.3.1	Individual Evaluation of Object Information for the baufix [®] Task .	125
5.3.2	Evaluating the Object Label Integration for the baufix [®] Task . . .	130
5.4	Rule Based Analysis of the Common Representation	132
5.4.1	General Rules	133
5.4.2	Improvements by Integrating Data Based Modules	136
5.4.3	Competing Object Hypotheses	137
5.4.4	Discussing Domain Specific Restrictions for the baufix [®] Task . . .	137
5.5	Additional Knowledge Integration	139
5.5.1	Semantic Region Growing	139
5.5.2	Assembly recognition process	140
5.5.3	Expectations from Monitoring the Construction Process	141
5.5.4	Shape Based Office Object Recognition	143
5.6	Evaluation Scheme for Competing Hypotheses	143
5.7	Summarizing the Evaluation	146

6	Summary and Conclusion	149
6.1	The General Integrating Module	150
6.2	Realizing Integrated Object Recognition Systems	151
6.3	Conclusion	152
A	The Inspection Tool for Integrated Systems	153
B	The <i>baufix</i>[®] Domain	159
B.1	Motivation	159
B.2	Object Label Alphabets	160
B.3	Classification Error Matrix for Probabilistic Integration	161
B.4	Test Set Images	163
C	The Office Domain	165
C.1	Recognition Task and Strategy	165
C.2	Test Set Images	166
D	Statistical Significance of Recognition Results	169
	Bibliography	171

1 Introduction

We experience our environment through different sensors that deliver large amounts of visual, acoustic, and other sensory data. But for organizing our knowledge and communicating about the experience we abstract from the sensory data and assign symbols. Computer systems that aim at extracting knowledge from sensory data and/or communicating with a human user or a second computer system need some kind of symbolic representation, too. The visual sensory data that is available for a computer system is constituted by digital images and the task of assigning symbolic knowledge to them is called object recognition.

But what is an object? The answer to this rather philosophical question is given pragmatically here: An object is something that carries an object label. Either the nose within a face, the face itself, the whole person or even the person walking in the forest or standing at the street border are thinkable object units. It depends on the recognition task, which object unit is suited and whether it is necessary to represent several levels of object information. The recognition task also determines, whether instances of one object class, like John's and Helen's cup, are differentiated or summarized by the unique class label.

Before an object label can be assigned an assumption about the affected image data has to be made. It depends on the characteristics of the image material, whether one centered and image filling object is assumable or the detection of perhaps more than one object is part of the recognition process. Furthermore, many approaches to object recognition rely on segment information for determining the object label, others exploit the image data without information about the detailed object boundaries. Concerning the subsequent use of the object recognition results, the label and, in case of more than one object is visible, the position constitute the minimal information for communicating about the object. Additional tasks, like estimating the object position in three dimensions or grasping it, require the complete information about the object boundaries. The very general term of object recognition denotes all systems providing any kind of object labels, independent of whether object detection and segmentation are part of the system, required from preprocessing steps or not addressed at all.

Within the decades of object recognition research many approaches are developed and applied to more or less specific tasks. But the enormous human abilities of recognizing objects in arbitrary environments remain unrivaled.

1.1 Combining Simple Methods

G. Wong and H.P. Frei published a paper in 1992 that they called 'Object Recognition: the Utopian Method is Dead; the Time for Combining Simple Methods Has Come' [Wong 92]. This paper is representative for a significant part of the object recogni-

tion community that proposes the explicit postprocessing integration of results delivered from several independent modules for solving recognition problems.

Combining several unreliable modules in order to get more reliable results is done since the first days of computing [Neum 56]. The usage of vacuum tubes as primary logic components of computers at that time, with a lifetime in the orders of days to years, caused the necessity to make explicit consideration for reliable computing in the presence of failure.

In the context of object labeling and segmentation the diverse modules are also more or less reliable in the sense that they generally produce both, correct and false results with regard to the recognition task. If the individual modules differ internally by exploiting diverse features and/or by following different recognition strategies, it is justified to assume the failures to be mainly uncorrelated. A failure of one module can then be probably compensated by modules providing correct results within a combination scheme.

Modules used for integrating their resulting object labels by a general purpose combination scheme are required to share a common object label alphabet. The combination result is the element of the set of labels that is mostly supported by the individual modules. It is either right or wrong.

The situation is not comparably clear for two dimensional segmentation results. A boundary, that is supported by more than one module can be assumed to be more reliable than others. But it is very unlikely that two segmentation approaches produce results that cover each other exactly. Segments will at least differ by several pixels, as also segments generated by different humans do. Coping with this effect requires an unprecise combination scheme. Additionally, different segmentation approaches deliver coarser and finer segments, dependent on the segmentation strategy and parameter settings. The subsumed finer segments of one module may support the outer boundary of a coarser segment delivered from another module. But this constellation does not finally clarify, how much objects are involved and which of the segments represent correctly an object region. A combination scheme for two dimensional segment information is desirable but its realization is not as presaged as the integration of discrete object label information.

The main advantages of combining explicitly rather small systems instead of generating more and more complex holistic ones are the simplicity of the individual modules, the higher transparency of the decision process, and the flexibility in exchanging modules or adding new ones to the system. Easy exchangeability becomes important, if the recognition task changes and modules are involved that base on task specific assumptions. Adding new modules to the system is desirable, if either the system quality has to be further improved, the image context changes, or the set of objects to be recognized is extended by instances providing additional object characteristics.

1.2 Recognition Strategies

The recognition strategy roughly separates the existing systems into two categories [Tous 78]: Systems that start processing with the raw image data and go further to higher levels of abstraction are called data driven or bottom up systems. Those that

start from expectations or domain specific knowledge and search for the fulfillment of this abstraction within the image data are called conceptually driven, model based, or top down systems.

High level knowledge and expectations respectively are thought to be essential for successful object recognition approaches. Already in 1982 Dana Ballard and Christopher Brown wrote in the introduction of their book 'Computer Vision' [Ball 82]:

We perceive a world of coherent three-dimensional objects with many invariant properties. Objectively, the incoming visual data do not exhibit corresponding coherence or invariance; they contain much irrelevant or even misleading variation. Somehow our visual system, from the retinal to cognitive levels, understands or imposes order on chaotic visual input. It does so by using intrinsic information that may reliably be extracted from the input, and also through assumptions and knowledge that are applied at various levels in visual processing.

Conceptually driven systems exploiting high level knowledge have to be integrated with data driven generated information. A problem with solely conceptually driven systems is the effect of hallucination. If the expectation for an object is very dominant, the image structure that fits best will be identified, even, if there is nothing relevant in the scene. This problem can be addressed by defining constraints for the image material concerning the occurring objects, their positions, and their number. For more generally applicable systems prior high level knowledge is applied to intermediate data driven results for supporting or disputing them, for clarifying ambiguities, or for extending incomplete results.

1.3 Object Context Knowledge

An important special kind of high level knowledge is context. The term summarizes knowledge that is based on relations between several instances rather than on the instances themselves. For example, if a hypothesis for a nose arises, the object below is likely to be the mouth. Besides those explicitly defined context constraints between high level symbols, also context information incorporating interdependencies between lower level features, like the colors of neighbored regions, may be exploited. 'The use of context in pattern recognition' [Tous 78] is on the one hand an old idea of the 70's, but on the other hand identified by Jain et al. [Jain 00] as one of the frontiers of pattern recognition in the year 2000:

It is well-known that the human recognition process relies heavily on context, knowledge, and experience. The effectiveness of using contextual information in resolving ambiguity and recognizing difficult patterns is the major differentiator between recognition abilities of human beings and machines.

The usage of object context knowledge also for improving the image segmentation step was already proposed by Feldman and Yakimovsky in 1974 [Feld 74]. Their

'semantics-based region analyzer' does segmentation in the form of region growing alternated with object label assignment to the intermediate segment hypotheses. For this assignment a rule based system exploits image features, like the mean color, and contextual object knowledge constraints, like 'the heaven is above the street and their boundary is rather horizontal'. Candidates for further region growing are neighbored segments that carry the same object label and provide a 'weak boundary' concerning image feature differences. Main parts of this early system are realized by explicitly formulated task specific rules and thresholds that have to be reformulated in the case of a changing task. The strict coupling does not allow the integration of additional modules. Nonetheless, the authors show the capabilities of the strategy to integrate data and conceptually driven aspects exploiting knowledge about objects and their context for simultaneously improving both, object segmentation and labeling.

1.4 Proposed Integrating Framework

Following the presented considerations about object recognition and segmentation systems leads to the conclusion that the integration of several modules that independently rely on different recognition and segmentation strategies is a promising approach for realizing successful recognition systems. These systems unify the exploitation of manifold general and task specific knowledge with the simplicity of individually implemented recognition strategies. Integrating independent image data based modules with those that deliver additional information dependent on preliminary generated hypotheses, like modules exploiting object context knowledge, results in an architecture for such a system as shown in Fig. 1.1.

The heart of this architecture constitutes the integrating component that has to cope with several kinds of available segment and object information. Besides the visual data based results the integration of additional information that is derived from intermediate results has to be provided. I propose an implementation for the integrating component that relies on generally applicable mechanisms and is, thereby, independent of the recognition task and the participating modules. The implemented generally applicable integrating module together with the proposed system architecture constitutes a framework for realizing an integrated recognition system for a given task that has to be filled with suitable task specific modules.

Many modules that implement different recognition and segmentation strategies are trainable or adaptable to different tasks and, thereby, reusable. The generally applicable modules may be complemented by modules exploiting special characteristics of the task. The modules are, generally, different in their value for a definite task, concerning the individual evaluation of their results, but also concerning their combination with other modules. The choice of modules is leaded by the idea of integrating as many aspects as necessary, while keeping the whole system as small as possible. The finding of the optimal choice of modules for fulfilling the given recognition task is supported by the implemented integrating modules. It handles segments and object labels delivered by several modules without making general constraints concerning the number of elementary modules which allows flexible changes within the set of participating modules. This flexibility is not only indispensable during the setup of a new system for a given task,

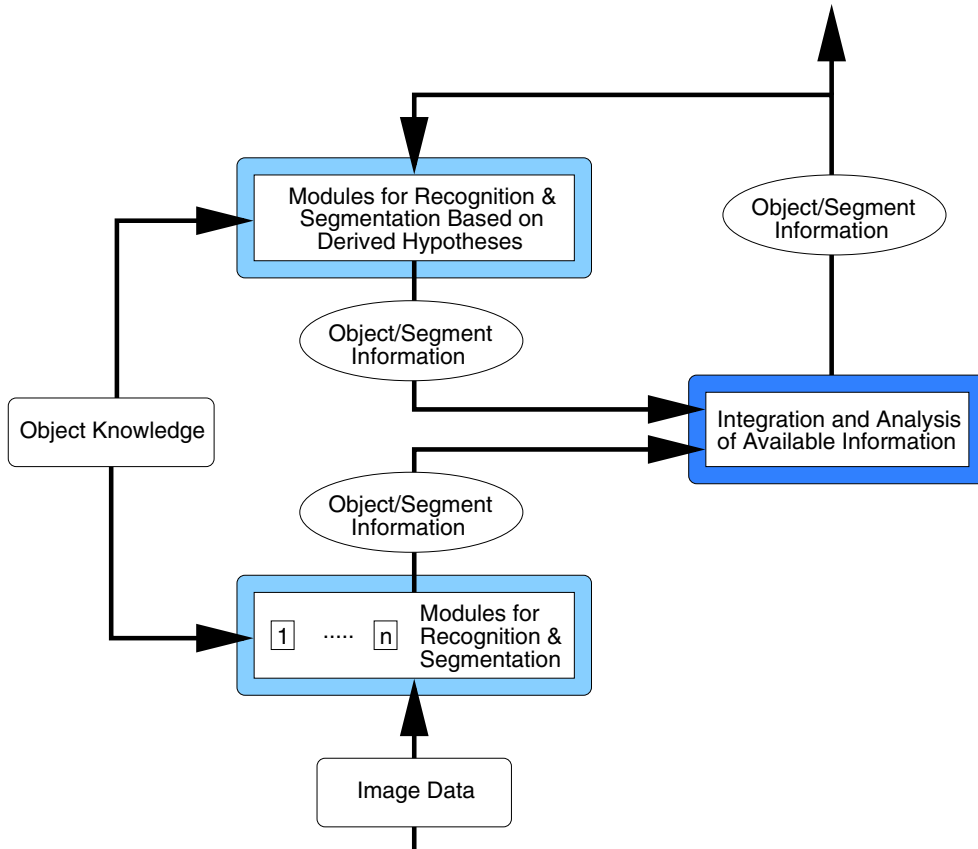


Figure 1.1: Architecture of an integrated object recognition system.

but supports later refinements by taking into account newly available modules.

The requirement of integrating object label and segment information of existing modules necessitates the integrating module to handle several kinds of information. There, generally, may occur labels concerning different object units that are either accompanied by segment information or not. Additionally, segment information that is not assigned to object information may be available. The integrating module accounts for the different kinds of valuable object and segment information within its internal data representation. In representing the fully detailed available information, the system remains open for equitably integrating additional information, as it is delivered from either very time-consuming processes or the modules exploiting higher level knowledge, like object context.

Integrated object hypotheses are generated by analyzing the available represented information. However, all the occurring object and segment information has to be assumed to contain more or less failures leading to disagreements within the represented information. These disagreements are considered by the interpretation process for generating object hypotheses that is part of the integrating module.

The proposed integrating module enables the realization of object recognition systems that integrate manifold available information by the flexible reuse of existing modules.

1.5 Outline

The outline of this thesis is as follows: Before starting the integration topic, I introduce some principles of image segmentation, on the one hand, and object recognition, on the other hand. The following Chapter 2 starts with an overview of image segmentation approaches. Segments group pixel information to meaningful units and, therefore, constitute the basis for many object recognition systems that aim at determining object regions in addition to the object labels. I present some techniques in more detail that are used within the realized integrated recognition systems of this thesis.

Chapter 3 does so for the recognition part. First some basic concepts of object recognition systems, object descriptions, and general purpose object label combination schemes are described shortly. Then, I present the exemplary individual modules that provide the basic object information for the realized integrated systems. Modules that rely on intermediate object hypotheses instead of basing on raw image data or segments are the topic of the final section of this chapter. Those modules deliver additional information about object labels and the evaluation of object hypotheses.

Chapter 4 describes the implementation of the proposed general integrating module. Segmentation results and different kinds of object information are stored within a unified common representation. The common representation builds the backbone for generating intermediate hypotheses and for integrating additional object information that leads to the final hypotheses.

The applicability of the general framework for realizing integrated systems that address different recognition tasks and rely on different recognition modules is shown by two examples. The first system deals with *baufix*[®] elements, which is a wooden construction system for children. The set of elementary pieces consists of colored bolts, nuts, and further connection pieces that can be used for constructing manifold assemblies. The *baufix*[®] domain is the common topic of the research projects within the Collaborative Research Center (SFB) 360 at Bielefeld University, see Appendix B for details. The second application of the integrating framework is the interpretation of scenes containing objects of an office domain. Here the determination of the object region is challenging due to the generally rich surface structure, see Appendix C for details. The realized systems and their quantitative results are described in Chapter 5.

Chapter 6 gives a summary and conclusions drawn from the realized systems.

2 Image Segmentation

Image segmentation is an essential part of many image processing systems. The representation in the form of segments decreases the amount of data to be handled by the following processing steps, while preserving as much image information as possible. The segmentation process is demanded to divide the image into semantically meaningful units, which are objects or parts of them and background information. These units build the input data for subsequent processing steps, like recognition or tracking approaches.

Due to its importance to image processing many segmentation approaches have been suggested during the last decades (see, e.g., [Fu 81], [Pal 93], [Skar 94], [Chen 01], [Frei 02], [Cufi 02] for reviews), but, nonetheless, the problem of image segmentation remains an active field of research.

In the following a short survey of the basic concepts and ideas concerning segmentation approaches is given, before I present some chosen algorithms in more detail. Those are the ones that are used within the systems presented in Chapter 5.

2.1 Basic Concepts

A set of segments S constitutes a compact description of the image I . It is characterized by a set of disjunct segments and a homogeneity prediction P that holds for each segment, but not for the unity of two neighbored ones.

Given the set of segments $S = (S_1, S_2, \dots, S_n)$, it is

$$\begin{aligned} S_1 \cup S_2 \cup \dots \cup S_n &= I && \wedge \\ S_i \cap S_j &= \emptyset, \quad \forall i, j \in \{1, \dots, n\} \wedge i \neq j \end{aligned}$$

The homogeneity prediction P is defined as a bivalued function on the segments. For a segmentation S it is:

$$\begin{aligned} P(S_i) &= \text{true}, \quad \forall i \in \{1, \dots, n\} && \wedge \\ P(S_i \cup S_j) &= \text{false}, \quad \text{if } i \neq j \wedge S_i \text{ neighbor of } S_j, \end{aligned}$$

The process of segmentation is generating a set of segments that fulfills the homogeneity requirements. Generally, there are several possibilities of defining the homogeneity criterion. The heart of such a definition is a distance measurement that has to be chosen adequately given the set of features to be used. Applying a threshold to the distance measured between, for example, the mean feature values of two segments is the simplest and commonly used method for generating a homogeneity criterion based on the distance function. The detailed formulation of the homogeneity predicate depends on the segmentation task, the chosen segmentation strategy and, finally, on the selected features and the appropriate distance measurement function. Some approaches are in common use and will be presented in the following.

2.1.1 Image Data Driven Features and their Distances

The quality of the segmentation result rises and falls with the calculation of suitable features from image data and the definition of the distance measurement and homogeneity criterion, respectively. Some basic ideas for features in common use and their distance measurements are given in the following.

Pixel Values The information basically given within image data is the grey level or color value of each pixel. Exploiting the distance between pixel values for measuring homogeneity therefore is the most conventional method. Defining City Block or Euclidean distance on grey level values is straightforward. Dependent on the chosen segmentation approach the criterion is defined for finding either a homogeneous area becoming one segment or the discontinuity in homogeneity dividing two segments.

From the psychological research concerning color perception, see, e.g., [Wand 95], [Kais 96] and [Webs 96], one knows how important color impressions are for human perception. Fig. 2.1 shows an example from the *baufix*[®] scenario that shows the discriminative power of color information in comparison to grey level values for segmenting the upper bolt and the cube.

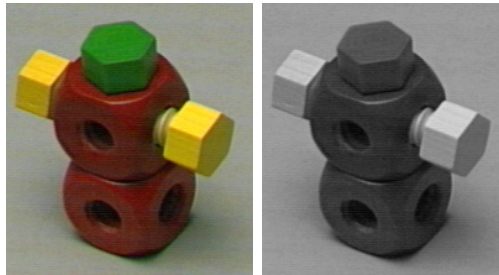


Figure 2.1: Example from the *baufix*[®] scenario for the discriminant power of color in comparison to grey level values for segmentation purposes.

Color comes into the focus of automatic segmentation research within the last years due to the fact that that increasing computational power makes an increased amount of data treatable. Instead of one grey level value, three coordinates related to a color space definition are commonly used for representing one color value. Thereby a variety of definitions for color spaces exist. A detailed presentation and discussion of common technical color spaces can be found in [Plat 00], [Fole 93], [Wysz 82].

From the technical point of view the basic color space is the *RGB*. Cameras possess sensors that are mostly sensitive for three different wavelengths within the light spectrum, corresponding to red, green, and blue. Also monitors realize the appearance of colors over a combination of red, green, and blue components. The *RGB* color space therefore is widely used for color image representation. Other color spaces that result from linear transformations of the *RGB* are used for special purposes, like the *YUV* for PAL and *YIQ* for NTSC video standard. The linear color spaces have in common the high mutual dependence between the components. This necessitates to take into account three or, at least, two of the color components for color comparisons.

Besides the linear transformations, there are several non linear color spaces, like the *HSI* or *HSV* that represent the hue, saturation, and intensity value, respectively, within separate components. The non linear color spaces need more computational effort for conversions and provide essential and non removable singularities. However, the components are rather independent from each other. That makes an analysis with a reduced number of color components more promising.

The *CIELAB* and *CIELUV* spaces are defined according to human color perception. The Euclidean distance calculated for the three dimensional color vectors of two pixels corresponds to human color distance perceptions. The meaningful definition of a color distance measurement distinguishes the *CIE* spaces from all the others.

Due to the complexity of the non linear transformation of color information to the *CIE* spaces, often the linear spaces or the *HSV* or *HSI* are used for segmentation. Many applications use standard distance measurements applied to the color components in spite of the lack of comparability to human color perception due to their generality and simplicity. Rehrmann [Rehr 97] uses the *HSV* together with manually defined distance tables, instead of a closely defined distance function.

Visual Texture Based Features Dependent on image characteristics also spatially extended patterns, called textures, may serve for the segmentation process. Fig. 2.2 gives some examples.

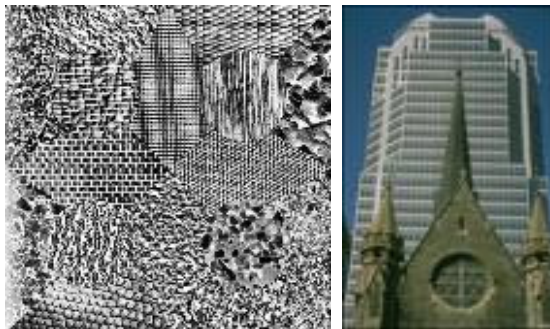


Figure 2.2: Example images with textures segments. (Left image from the texture lab database [Rand 99], right one origins from the database presented in [Mart 01])

An accurate definition of visual texture can not be found in literature, but it is common to call a spatial extended pattern generated by repetition to be texture. Wilson and Spann [Wils 88] formulate this as: 'They are spatially extended patterns based on the more or less accurate repetition of some unit cell.' This formulation shows that neither the kind of unit cell nor the manner how it is repeated is clearly defined. Especially for natural images the repetition process often shows irregularities and fluctuations.

Due to its occurrence in natural scenes as well as in scenes containing artificial objects texture analysis is a longterm and active field of research. This covers research concerning texture perception (e.g. [Jule 65], [Jule 81], [Gibs 87], [Saun 03]) on the one hand and technical approaches for automatic texture analysis and synthesis (e.g.

[Zhan 02b],[Chan 05]) on the other hand.

Automatic texture analysis searches for valid features that are characteristic for the special texture and are suited for discriminating different textures. Gonzalez [Gonz 87] classifies the approaches to texture analysis into three classes, the statistical, structural, and spectral approaches. Statistical approaches rely on the moments of the grey level distribution, while structural approaches rely on finding the texture elements and the rules of their repetition [Hara 79]. Spectral analysis exploits the periodic characteristic of texture by the projection onto a suitable set of basis functions, like Gabor functions or wavelets. Each of the basis functions is concentrated in a different area of the frequency domain. For a comparative study of various types of functions see [Pich 96], [Rand 99]. Also approaches using combinations of statistical and spectral features are proposed (e.g. [Clau 04]).

Besides standard distance measurement functions applied to the components of texture feature vectors, special multi dimensional distance functions are designed and optimized for proceeding segmentation based on texture features, see e.g. [Do 00], [Abba 03].

In spite of the longterm and intensive research on texture, the progress in texture analysis has been very slow, because natural textures turned out to be highly variable. Nonetheless texture is an important feature for image segmentation and one way of its successful exploitation is combining it with other features, like contour [Mali 01] or brightness and color [Mart 04] for natural image segmentation.

2.1.2 Approaches for Segment Generation

The great amount of different approaches to determine segments are divided roughly into three classes [Fu 81]: edge detection, region based approaches, and clustering, where region and edge based approaches work within the image domain, while clustering is accomplished within the feature domain.

Region Based Approaches to image segmentation generate segments by merging regions according to the defined homogeneity criterion. Starting from a set of primitive segments merging is done until the homogeneity criterion is violated. Within the classical approaches to region growing just local information, like the mean color values of segments, are compared to each other. Those approaches are simple and fast in computation (e.g. [Zuck 76],[Adam 94]). Problems arise in choosing suitable starting points for the region growing process, the seeds. This choice often is decisive for the segmentation result, as well as the merging sequence. Additionally, adaptations of the mean feature value of a segment that grows along the direction of a slight, but monotone, change in feature values lead finally to segments that cover very different pixels, called the chaining effect. For coping with those aspects, split-and-merge-approaches take the whole image as their starting point. The algorithms consist of a sequence of alternating splitting steps, following a predefined strategy, and merging steps according to the homogeneity criterion (e.g. [Fuka 80], [Rehr 98]).

Edge Detection The complementary edge detection approaches base on the assumption that abrupt changes of image features occur at the boundary between two segments (e.g. [Huec 73], [Marr 80]). Classically just very local information of a few pixels is exploited in searching for these discontinuities. After identifying edge pixels they have to be concatenated to boundary pieces by chaining [Rosi 89] or model approximation [Kass 88, Leon 93]. For getting closed boundaries that correspond to object surfaces, edge linking, or grouping techniques have to be applied to the mostly disconnected boundary pieces, which is in general a challenging problem [Schl 01].

Due to the local information determining the boundary location the results are susceptible to noise and difficult to group to closed segments on the one hand, but they preserve many details of the image, on the other hand. At image locations with low contrast, for example, edge detection successfully extends region based approaches. Integration of both strategies is done either by edge detection results taking influence on the parameters of the region based algorithm or vice versa or by combining the independently computed results within a postprocessing step. [Cufi 02] gives a survey and a discussion of integrated approaches.

Using global information of the image for edge detection is the main idea of graph based approaches to segmentation that deliver very promising results [Wu 93], [Cox 96], [Shi 00], [Felz 98]. Here, a weighted graph is generated with nodes representing the segments and edges representing the relations between the segments. Primitive segments are the individual pixels. The edge attributes are calculated based on spatial neighborhood and feature based relations between the segments represented by the participating nodes. Edge detection then becomes a partitioning problem. Global image information is taken into account for defining the location of cuts within the graph that correspond to boundaries of segments. Best results deliver the cut criterion proposed by Shi and Malik in [Shi 00] that integrates the within-group similarity and the between-group dissimilarity within one measure of dissociation. A disadvantage of this approach is the complexity of the problem associated with high computational effort necessary for getting results. But there are possibilities for simplifications conserving the main ideas that accelerate the process noticeably for practical use [Shar 00], [Felz 98].

Clustering Features Data clustering techniques are in common use for feature space analysis (see [Jain 99] for a review). Clustering techniques work primarily within the feature space. They identify areas within the feature space that are dense according to the defined distance measurement. The elements of each area are assigned to their representative or a class label. The reprojection of the quantized feature space to the image domain delivers an intermediate label image, whose segmentation is done by applying a region based approach, as described above.

Most clustering techniques are either hierarchical or they do iterative square-error clustering [Jain 00]. Hierarchical methods aggregate or divide the data based on a proximity measure comparable to the region based segmentation approach in the image domain. Iterative square-error clustering methods aim at finding a partition that minimizes the within-cluster scatter or maximizes the between-cluster scatter. The overall optimal solution to this problem is not computationally feasible due to the fact that all possible partitions has to be tested. Therefore, several simplifications and heuristics

have to be introduced involving the loss of the global optimality. Parametric techniques rely on prior knowledge about the number of clusters or make assumptions about the shape of the clusters. Non parametric methods regard the feature space as an empirical probability density function, whose maxima correspond to dense areas in the feature space. A cluster is associated firstly to each mode of this probability density function. The local structure of the feature space determines the final localization and form of the cluster [Robe 97], [Coma 02]. Practical problems arise for the reprojection of the clustered data to the image domain in the form of small and fragmented regions that have to be handled either by smoothing and/or by integrating image domain knowledge into the clustering process [Coma 02], [Makr 05].

2.1.3 Integrating Task Specific Knowledge to Segmentation

The concepts of feature distance measurements and segmentation approaches, as described above, are principally applicable to many segmentation tasks. But task specific knowledge is necessary for choosing the appropriate features and algorithmic strategies in order to achieve good segmentation results. Parameter settings like, for example, the threshold for the distance function or the number of feature values to be calculated, are mostly optimized based on a set of images that is characteristic for the task.

Using a classifier for representing a suitable definition of the homogeneity criterion is a further possibility of integrating task specific knowledge. The classifier is trained based on a labeled testset in order to distinguish several classes based on the given features. As described already for the unsupervised clustering an intermediate label image is generated, that has to be postprocessed. Using the classifier concentrates the task specific knowledge mainly within the classification step. Further on, the classifier is trained to implicitly represent an optimal homogeneity criterion, whose explicit formulation is problematic due to complex features or different kinds of features [Mart 04].

Texture features often have high dimensionality and the definition of similarity measures is not straight forward. Therefore texture classification as the first step of segmentation is in common use. [Turt 03] addresses the problem of defining classes of texture within natural outdoor images by applying self organizing maps for the training as well as for classification. A survey of proposed texture classifiers gives [Rand 99] and comparisons of classification power are done in, e.g., [McGu 02], [Hori 04].

Domain specific knowledge like restrictions in the appearance of colors are exploited in systems where pixel color values are classified and the resulting label image is segmented in a following general step. [Heid 96a] uses a polynomial classifier for the limited number of colors within the *baufix*[®] scenario (see Fig. 2.1 and Appendix B). Classifying pixel values to color classes is also done e.g. using histograms of a labeled testset [Band 00] or applying constant thresholding in appropriate color spaces [Bruc 00] for the RoboCup scenario. A special color tone that is important for many applications, like face and gesture recognition, is skin color. Pixel color based methods are applied for distinguishing skin or non-skin colors based on histograms or mixtures of Gaussians [Raja 98b], [Sori 00], [Jone 02]. Those pixel color based approaches deliver very fast and rather good results, as long as the environment remains constant. In the presence of changing illumination the appearance of colored surfaces changes, too, if no corrections are done. Applying processes for conserving the appearance of color sur-

faces in the presence of changing illumination is called color constancy. The human visual system possesses several adaptation steps for implementing its color constancy system [DZmu 86], [Webs 96]. For technical systems there are several approaches to color constancy (see [Funt 98], [Barn 02a], [Barn 02b] for a survey and comparison). All the approaches are computational time consuming. A possible way to cope with slightly changing illuminations is the application of adaptive systems, where classification criteria are adapted based on classification results of preceding steps (e.g. [Raja 98a], [McKe 98], [Weig 04]).

Besides domain specific information about color appearances, also object knowledge is usable for segmentation, if object classes can be distinguished based directly on pixel information of rather small windows [Heid 00]. The size of these windows determine then the final precision of the segmentation approach.

2.1.4 Model Based Top Down Segmentation

If prior knowledge about the desired segmentation result is available, expectations may lead the segmentation process top down from a model to the data. Segmentation then becomes a matching problem, where given templates are compared to the image, with the aim of detecting the corresponding structure in the image and segmenting it from the background. Templates may be either rigid or deformable. Examples for rigid models are image patches or shape models. In matching those templates directly to the image data, the best position for the template in the image is determined and the stored segmentation information is transferred to the image data. This requires the image data to correspond rather exactly to the template. Xu et al. [Xu 03] use rigid image patches or silhouettes and match those queries to the results of a multi scale data driven segmentation approach. The process generally determines a set of data driven segments to correspond to the query and to constitute the figure that is segmented from the background. The quality of the final shape is given by the data driven segments.

Deformable or active models are characterized by the adaptation of model parameters according to the image material during the matching process. This leads to good segmentation results even in case of variations between the model and the image data. However, an external process is needed to perform the initial localization of a model within the image data, which is crucial for the success of the adaptation process.

Kass et al. [Kass 88] introduce active contours, which are parameterized basic contour models that are deformed according to image material. [Yuil 92] uses deformable prototypes based on several contour elements for segmenting the facial elements eyes and mouth. The approach of active contours is extended in [Coot 95] to account for object specific flexibility of a shape. The shape is represented by a set of points and their geometric distribution. Knowledge about object specific flexibility is acquired during the training phase of the system. [Mard 97] applies shape based deformable templates on images showing multiple and partly occluded objects.

Another attempt to a flexible representation of shapes is to represent characteristic parts and flexible combinations of these parts. In using exemplar based models for those parts, model knowledge is automatically acquirable. [Bore 02] proposes an object class specific segmentation approach based on fragments. Prototypical shape fragments of an object class are determined in advance and stored together with a segmentation

template. The image is assumed to contain one object of the given class that is almost centered. A finer localization of the object is part of the segmentation approach that determines an optimal cover of the image with fragments. The final figure ground segmentation results from summarizing the segmentation information accompanying the fragments. A comparable approach to segmentation in combination with preceding object part detection is proposed in [Leib 04a]. Both approaches apply models for the characteristic parts that they acquired automatically in advance. Consequently, no further adaptation to the image data is possible. The segmentation is given within the model and the quality of the segmentation result is determined by the degree of correspondence between the fragment and the image. Model based systems generally deliver good segmentation results, as long as their assumptions are sustainable. Even for cluttered images, where data driven systems tend to lose themselves in details, model based approaches focus on their expectation and organize the chaotic input in direction of their model. However, this leads to the tendency of hallucinations, which are false positive results in the absence of the expected image content. Data driven verifications avoid those effects.

Additionally data driven results refine model based segments in cases, where the reality of image data goes beyond the model representation. [Bore 04] extends the fragment based top down approach by integrating it with a multi scale bottom up process. The final figure ground segmentation constitutes a compromise between model requirements and data constraints. A cost function determines the best compromise in evaluating the discrepancy between the segmentation hypothesis based on object knowledge on the one hand and the homogeneity concerning the image data on the other hand. Compared to the solely model based approach the results for unexpected part constellations are improved by the influence of the data driven generated segments.

A combination of data driven generated segments and expectations given as deformable shape model is the topic of [Liu 01]. Starting point of the algorithm are segments determined by a general purpose color segmentation process. A deformable shape model decides about merging neighbored regions and splitting others. The initialization of the deformable model is given by the data driven segments, whose deficiencies concerning under and over segmentation as well as smoothness of shape are corrected by the model knowledge.

The preceding descriptions show that data driven generated segments are crucial basics for image segmentation systems, where sparse prior knowledge about the scene makes the application of model knowledge difficult or where models are incomplete.

From the variety of existing data driven approaches those that are suited for a given task has to be selected and parameterized. The concrete segmentation modules used for the realized integrated object recognition systems of Chapter 5 are presented in the following.

2.2 Choice of Color Image Segmentation Algorithms

When building systems, one needs to choose from the great amount of segmentation approaches in the literature. But rather independent of the given task the selection of the one method that works successfully is mostly impossible. However, due to the variety

of approaches the integration of their results for improving the final segmentation is promising. This idea is realized with the general integrating framework as described in Chapter 4.

In the following, I will present those data driven color image segmentation approaches that are used within the realized integrated systems presented in Chapter 5. The segmentation modules are chosen to cover a wide range of the algorithmic spectrum in order to provide results that complement each other.

2.2.1 Hierarchical Region Growing: Color Structure Code

The segmentation algorithm proposed in [Rehr 98] uses a hierarchical region growing method that combines the advantages of local region growing with global split and merge techniques in order to find homogeneous color regions. It is called Color Structure Code (CSC). The algorithm is very efficient, running in linear time in the number of image pixels. It is mainly based on a local region growing technique. Global information makes it independent of the choice of the starting point and the order of processing. This is achieved by processing the region growing in the frame of a hierarchical hexagonal topology, formed by so-called islands. Islands of level 0 are formed from 7 pixels and overlap each other so that each pixel is covered by two islands (see Fig. 2.3). Islands of level l consists of 7 islands of level $l - 1$. They also overlap, means one island in level $l - 1$ is covered by two islands of level l . The island structure ends up in 1 island covering the whole image.

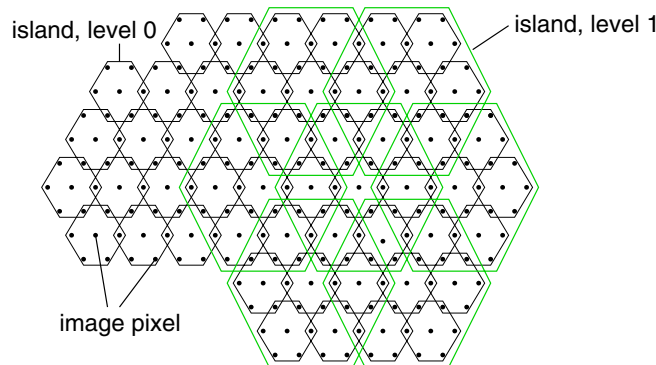


Figure 2.3: Hierarchical hexagonal island structure used for color structure code (re-draw with variation from [Rehr 98]).

The hexagonal topology ends up in an intuitive topology, but brings problems in application because of the orthogonal grid of image pixels. Therefore, the hexagonal scheme is mapped on the orthogonal grid, as shown in Fig. 2.4. For simplicity the hexagonal structure will be assumed for further explanations.

The segmentation algorithm consists of three phases. In the initialization phase the values of the pixels within each island of level 0 are compared to each other according to the defined color similarity measurement. Those pixels that are neighbored and have similar colors are summarized to one code element, where a code element represents a

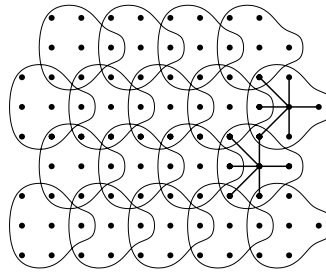


Figure 2.4: Mapping of hexagonal island structure to cubic pixel grid [Rehr 98].

connected region in the image. After the initialization phase one or more code elements are built up from the 7 pixels within each island of level 0, as shown in Fig. 2.5.

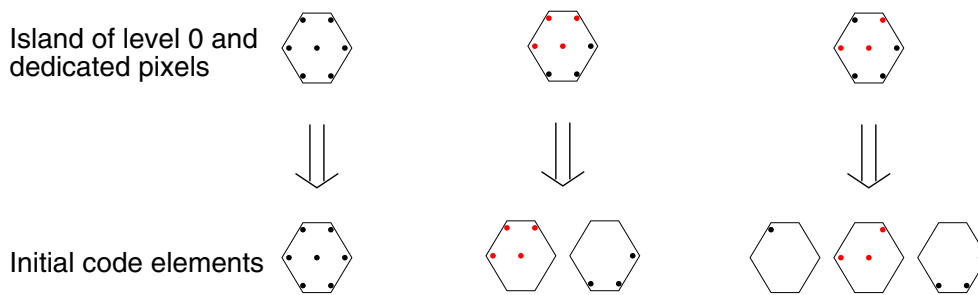


Figure 2.5: Examples for initial code elements according to color similarity of pixels assigned to one island of level 0 (redraw with variation from [Rehr 98]).

In the linking phase code elements of level $l + 1$ are generated separately for each island of level $l + 1$ by taking into account the associated code elements and islands of level l , as depicted in Fig. 2.6.

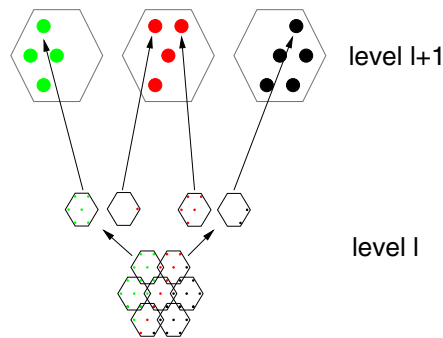


Figure 2.6: Linking of code elements (redraw with variation from [Rehr 98]).

Code elements of level l are linked, if the regions represented by them are neighbored and of similar color, analogous to the criterion applied in the initialization phase. Within

the hexagonal overlapping structure code elements are neighbored, if they share a common sub region in their common sub island. See Fig. 2.7 for an illustration. On level 1 this implies that they share a common pixel. For measuring color similarity of two code elements, the mean color values of the regions represented by the code elements are calculated and compared applying the color similarity measurement.

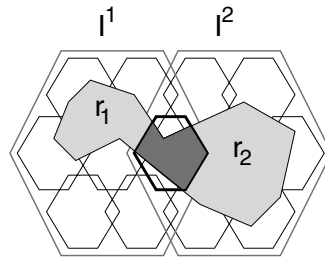


Figure 2.7: Two neighbored islands I^1, I^2 with their common sub island I (printed bold) and the common sub region of r_1 and r_2 located within I . [Rehr 98].

The new code element of level $l + 1$ is stored together with pointers to the code elements of the lower level from which it is generated. A homogeneous region within the image then is represented by a tree of code elements where the root of the tree is the highest level code element that is not linked to another element.

For region growing approaches that are based solely on local color similarity a chain of neighbored pixels with smoothly changing colors causes the linkage of differently colored regions. The CSC solves this problem in its splitting phase by exploiting the global information stored within the island hierarchy. In the chaining situation two neighbored code elements of level $l + 1$ like r_1 and r_2 in Fig. 2.7 have mean colors that are not similar enough to link them. Because of the smoothly changing colors, there exists a common subregion within the common sub island in level l that is similar to both code elements of level $l + 1$. If this situation is detected, one must partition the common subregion between the two differently colored regions, which means that a low contrast contour has to be detected. This is done by recursively assigning common submode elements to the one code element that provides a smaller color difference than the other. By doing this along the code element tree the region boundary is found rather accurate and the connectivity of the two regions is ensured. The described splitting allows to detect contours with low contrast using the global information stored within the island structure.

Until now the definition of color similarity is still open. The original implementation uses a scheme based on 48 thresholds within the HSV color space. The thresholds were determined by analyzing examples. Heidemann [Heid 98] uses in his implementation that is also used here the standard Euclidean distance calculated for RGB color values and applies one threshold for parameterizing the segmentation procedure.

Fig. 2.8 shows an exemplary image from the office domain together with two segmentation results delivered from differently parameterized CSC runs. The thresholds for the Euclidean distance between the RGB color values are 15.0 (middle) and 25.0 (right), respectively. For the results shown here no threshold for a minimal region size is applied resulting in many very small regions. It is obvious that the algorithm is able to

identify the main structures of the image in locating segment boundaries at the object boundaries. However, many small structures, like the image on the cup, result in rather small segments that have to be summarized for analyzing the image content. Adapting the threshold to a less strict homogeneity criterion results for the example in melting the black part of the mouse pad with the blue background. This shows the difficulty in defining those constant thresholds for region based segmentation approaches. Note the segmentation boundaries within the color ramp at the lower part of the pad. Locating boundaries within the area of slight changes is generally problematic, but the algorithm solved this problem well.



Figure 2.8: Example image with CSC segmentation results for two color distance thresholds. Euclidean distance between RGB color values is used with threshold 15.0 (middle) and 25.0 (right), respectively.

2.2.2 Graph Based Segmentation Using the Local Variation Criterion

A graph-based segmentation algorithm exploiting a criterion measuring the local variation of an image region was introduced by [Felz 98]. The key aspect of the segmentation algorithm is defining a predicate for measuring the evidence of a boundary between each two regions. This evidence depends on the intensity differences across the boundary, on the one hand, and the intensity differences between neighboring pixels within each region, on the other hand. Integrating these two measures allows, in principle, to preserve details in low-variability image regions, while ignoring details in high-variability regions. Fig. 2.9 shows an example with three perceptually distinct but non homogeneous regions that are separated well by the algorithm given a suitable parameter setting.

For starting the segmentation procedure, the image data is represented within a graph $G = (V, E)$. G is an undirected graph with vertices $v \in V$, the set of elements to be segmented, and edges $(v_i, v_j) \in E$ corresponding to pairs of vertices. Each edge has a corresponding weight $w((v_i, v_j))$. In this graph based formulation, a segmentation S is a partition of V into components such that each component $C \in S$ corresponds to a connected component in a graph $G' = (V, E')$, where $E' \subseteq E$. With other words, any segmentation S of $G = (V, E)$ is induced by a subset of the edges in E .

The pairwise region comparison predicate $D(C_1, C_2)$ determines, whether or not there is evidence for a boundary between two components of a segmentation. For taking into account local characteristics of the data, the predicate compares an indicating measurement for the occurring differences between the nodes within each component with an

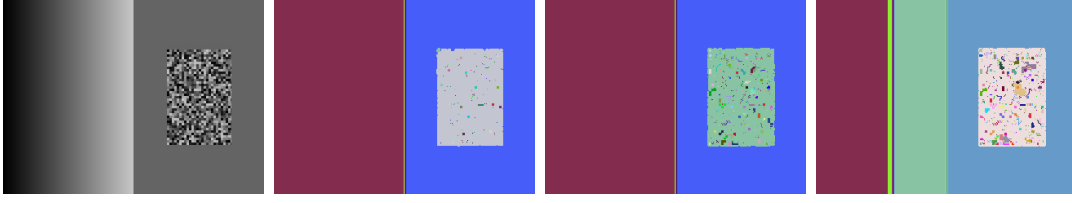


Figure 2.9: Example image with three perceptually distinct but non homogeneous regions and result of graph-based color segmentation. Results are calculated using the parameters $s=0.8$ and $k = 1000$, $k = 500$, and $k = 300$ respectively. For parameter explanations, see text.

appropriate one concerned with the differences between the nodes of two components, instead of applying any global threshold. Differences between two nodes thereby are coded based on appropriate features and their distance function by the weight of the connecting edge $w((v_i, v_j))$. Per definition a high value for w denotes a high difference.

For determining the measurement for the internal differences of a component C , the minimum spanning tree MST of the component is determined. $MST(C, E)$ denotes the cheapest subset of edges that keeps the graph connected. The internal difference values for the component C then is defined to be the largest weight occurring in the $MST(C, E)$:

$$Int(C) = \max_{e \in MST(C, E)} w(e)$$

The given component C only remains connected, if edges providing a weight of at least $Int(C)$ are considered.

The difference measurement for two components C_1, C_2 is defined to be the minimum weight edge connecting the two components:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j)$$

If there is no edge connecting C_1 and C_2 , $Dif(C_1, C_2)$ is set to ∞ . Taking into account only the smallest edge weight between two components instead of a more intuitive measure, like the median weight, simplifies the problem significantly: The problem of finding the median weight would be NP-hard.

Evidence of a boundary between two components is given, if the difference between the components, $Dif(C_1, C_2)$, is large relative to at least one of the internal difference values, $Int(C_1)$ and $Int(C_2)$:

$$D(C_1, C_2) = \begin{cases} \text{true,} & \text{if } Dif(C_1, C_2) > \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)) \\ \text{false,} & \text{otherwise} \end{cases}$$

The threshold function $\tau(C)$ is introduced to allow some external influence on the region comparison predicate. Any non-negative function of a single component can be

used. It causes the segmentation method preferring components with special characteristics concerning, e.g., size or shape. Per default a normalization of the component size is used:

$$\tau_{ns}(C) = k/|C|$$

This threshold function ensures that for small components there must be a high difference between the components in order to have an evidence for a boundary. The constant k then sets a scale of observation, with increasing k the preferred size of a component is increased, too. This does not imply that smaller components are not allowed, but there must be an even stronger difference between them to get evidence for a boundary.

An algorithm for calculating a segmentation S with components $C_1 \dots C_r$ from an input Graph $G = (V, E)$ containing n vertices and m edges using the criterion D can be formulated as follows:

1. Sort E into $\pi = (e_1, \dots, e_m)$ by non-decreasing edge weight.
2. Start with a segmentation S^0 , where each vertex v_i is located at its own component C_i .
3. Repeat step 4 for $q = 1, \dots, m$.
4. Construct S^q from S^{q-1} :
 Let v_i and v_j denote the vertices connected by the q -th edge in the ordering, i.e. $e_q = (v_i, v_j)$. If v_i and v_j are located at disjoint components of S^{q-1} and $w(e_q)$ is small compared to the internal difference of both those components, then merge the two components, otherwise do nothing. More formally, let C_i^{q-1} be the component of S^{q-1} containing v_i and C_j^{q-1} the component containing v_j . If $C_i^{q-1} \neq C_j^{q-1}$ and $w(e_q) \leq \text{MInt}(C_i^{q-1}, C_j^{q-1})$, S^q is obtained from S^{q-1} by merging C_i^{q-1} and C_j^{q-1} . Otherwise $S^q = S^{q-1}$.
5. Return $S = S^m$.

For doing image segmentation with this algorithm it must be defined, which data should be represented in vertices, edges and weights.

Felzenszwalb and Huttenlocher [Felz 98] define in their original implementation that is used here the undirected graph $G = (V, E)$ by generating for each pixel a vertex. Vertices are connected by edges corresponding to the pixels of an eight neighborhood in the image. The weight of an edge is calculated from the difference in intensity provided by the connected pixels. The authors call this 'grid-graph'. Color images are handled by independently segmenting each color plane and fusing the results by calculating the intersection between the detected components. This procedure avoids the definition of a color value distance function and turns out to be less susceptible to low contrast edges, because each color channel has to provide low contrast to fail the edge. Two further aspects concerning the practical use of the segmentation modules are the intended preprocessing application of a Gaussian filter for smoothing that is parameterized by its

standard deviation s and the implemented postprocessing step for avoiding small segments that is parameterized by a threshold m for the minimal segment size. Segments that are smaller than the threshold are pixel wise merged with the most similar neighbored segment.

The general graph cut algorithm is implemented to run in $O(m \log m)$ time, where m is the number of edges in the graph. In this point the algorithm differs from most graph-based approaches. For the application to image segmentation the number of edges, m , is of the same magnitude than the number of vertices or image pixels, n , resulting in the algorithm running in $O(n \log n)$.



Figure 2.10: Example image from the office domain (left) and segmentation results delivered from the segmentation method proposed in [Felz 98]. Parameter settings are $k=150$, $s=0.8$. Results without accounting for segment sizes (middle) and with additional postprocessing using minimal segment size threshold $m=64$ (right).

Fig. 2.10 shows an exemplary image from the office domain and the appropriate segmentation results. The careful application of the threshold for the minimal segment size reduces the amount of noise segments significantly. The algorithm was able to segment the main object structures by locating segment boundaries correctly at the object boundaries. The letters on the pad demonstrate an interesting effect. It would be desirable for this task to handle them as one structure, comparable to the random area in Fig. 2.9, and not to separate each letter out. For the example with the applied parameterization some letters are summarized to one structure (upper left corner), while others (upper right and lower right) constitute separate segments.

2.2.3 Feature Clustering Using Mean-Shift Algorithm

The mean-shift algorithm is a general non parametric technique for feature space analysis. Clustering following this approach works without prior knowledge about the number of occurring clusters and without making assumption about the feature distribution. Clusters are located at dense areas within the feature space, where many instances of feature vectors occur. They are identified from the feature probability density function by determining areas of high feature probability.

The procedure is applied to color image segmentation by Comaniciu and Meer, proposed in [Coma 97]. The significant colors of the image are identified as the centers of dense areas within the color feature space. The available color information is appropriately reduced and reprojected to the image plane for generating segments, see

Sec. 2.1.2. Fig. 2.11 shows an example grey level image together with its histogram approximating the feature probability density and the estimated cluster locations at the modes of the function. The results of assigning the image pixels to one of the five dominant colors is shown by pseudo color coding at the right side.

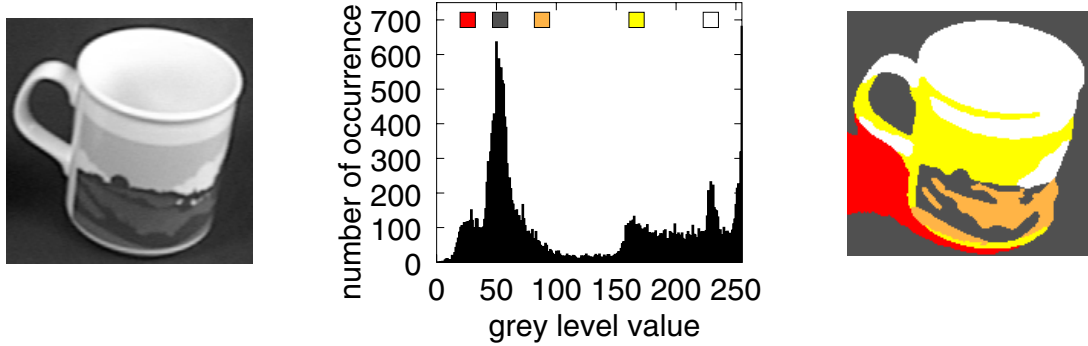


Figure 2.11: Example for mean-shift segmentation. Left: Grey scale example image. Middle: Histogram of grey values corresponds to probability density function. The five stars indicate the cluster localization done by mean-shift. Right: Image pixel assignment to the significant colors determined by the mean-shift approach.

The mean-shift idea was proposed in 1975 by Fukunage and Hostetler [Fuku 75], generalized by Cheng [Chen 95] and further analyzed by Fashing and Tomasi [Fash 05]. The mean-shift procedure generally locates dense areas within the feature space by determining the modes of the appropriate probability density function. There, the feature probability is high and the gradient of the probability density function is low or zero. For determining these areas a search window is located firstly randomly at feature vector x . The algorithm aims at determining a vector, that is added to the central vector x and thereby shifts the search window in the direction of the next mode of the probability density function. The process works iteratively and ends, if the window is centered at the location of the mode.

For starting the calculation, the search window has to be defined in its form and size, before it is randomly located at the feature vector x . The search window S_x is demanded to contain all feature vectors y near the central vector x up to a given maximal distance threshold r .

$$|y - x| \leq r \quad \forall y \in S_x^r$$

By using, for example, the Euclidean distance applied to the d dimensional feature vectors, the distance does not depend on the location of the vectors within the space and the search window becomes a sphere of radius r .

The mean-shift vector $\mu(S_x^r)$ is determined based on the given probability density function p as the expected value of the vector $z = y - x$

$$\mu(S_x^r) = E[z|S_x^r] = \int_{S_x^r} (y - x)p(y|S_x^r)dy = \int_{S_x^r} (y - x) \frac{p(y)}{p(y \in S_x^r)}$$

This value denotes the region within the search window, where the probabilities for features vectors to be located is highest. The formula given above can be rewritten, as shown in [Coma 97].

$$\mu(S_x^r) = \frac{r^2}{d+2} \frac{\nabla p(x)}{p(x)}$$

The mean-shift vector is proportional to the gradient of the probability density function and reciprocal to the value of the probability density function. If the center of the search window is located near a mode of the density function, this results in already large $p(x)$ and small $\nabla p(x)$ leading to small shifts. Larger shifts occur, if the dense region is located at the border of the search window.

The iterative algorithm for localizing a mode of the probability density function can be summarized, as follows.

1. Choose the radius r of the search window.
2. Choose randomly the initial location of the window.
3. Compute the mean-shift vector and shift the search window by that amount.
4. Repeat step 3 till convergence.

For the image segmentation task, the feature space to be analyzed is the color space. In order to get an isotropic feature space, calculations are done in the *CIELUV* color space, where the Euclidean distance between two color values is perceptually motivated. The steps necessary for image segmentation are:

1. Definition of segmentation parameters

The user has to set the parameter for the radius of the search window. Instead of setting a constant value, it is set dependent on the visual activity in the image. Because for an image with large homogeneous regions, the radius is chosen smaller than for an image with many textured areas. The radius parameter p_r is set by the user and determines the final color radius of the search window by multiplying it with the normalized square root of the trace of the image covariance matrix that is a measurement for the visual activity. The user has the possibility to set some additional parameters. For the determination of significant colors, a threshold for the minimum number of pixels generally supporting this color, N_{min} and a threshold for the minimum size of at least one connected component providing this color, N_{con} are accounted for. Additionally, a threshold for the minimal general connected component size, N_{reg} , avoids the generation of too small segments.

2. Definition of the location search window

The initial location of the search window in the feature space should be close to a high density region. To support this, about 25 locations within the image domain are randomly chosen and examined. The mean color values calculated within small neighborhoods of these locations are mapped into feature space. Chosen for starting the algorithm is the feature point, where the sum of densities within the search window is maximal.

3. Mean-shift algorithm

The mean-shift algorithm, as described above, is applied to the selected search window until the magnitude of the shift becomes less than a given threshold.

4. Removal of the detected feature

Feature values lying inside the final search window are removed from the feature space and the pixels of the image domain carrying these features are discarded for further processing. The pixels located within an eight neighborhood of concerned pixels are additionally discarded, in order to account for artifacts probably occurring at the borderline between two colors in the image.

5. Iterations

Repeat steps 2 to 4, until the number of feature vectors no longer exceeds the given minimum number required for a significant color, N_{min} , which implies that no significant color remains.

6. Determining the initial feature palette.

For initializing the feature palette, the significant colors from those so far extracted are determined. At least N_{min} pixels must support the color and a significant region containing at least N_{con} connected pixels providing the color has to exist. Extracted colors that are not significant are not accepted for the feature palette.

7. Determining the final feature palette.

All pixels are reallocated based on the feature palette. In succession pixels yielding feature vectors inside the final search windows of a significant color are allocated to that color, i.e., one value of the initial feature palette. After that, the search windows are inflated to double volume. Pixels providing features of the extended window and being neighbored in image domain to a pixel that is already assigned to the appropriate class are allocated now.

For determining the final feature palette the mean value of all feature vectors assigned to one class is taken. There may remain a few pixels unclassified. They are mapped to the closest color in the final feature palette.

8. Postprocessing

Within the image domain all connected components with less than N_{reg} pixels are removed. They are allocated to the majority color of their 3x3 neighborhood. A smoothing step follows optionally .

The implementation of the segmentation algorithm used here, is mainly based on the available original one of [Coma 97], but the interfaces are adapted to the locally used inter process communication system and the local implementation allows the external adjustment of the occurring parameters.

Fig. 2.12 shows an example from the office and the baufix[®] domain and the segmentation results delivered from the mean-shift approach, applying different color radius parameters. In the result images each color class is coded by one pseudo color tone. The lower color radius and thereby smaller search windows leads to the results shown in the

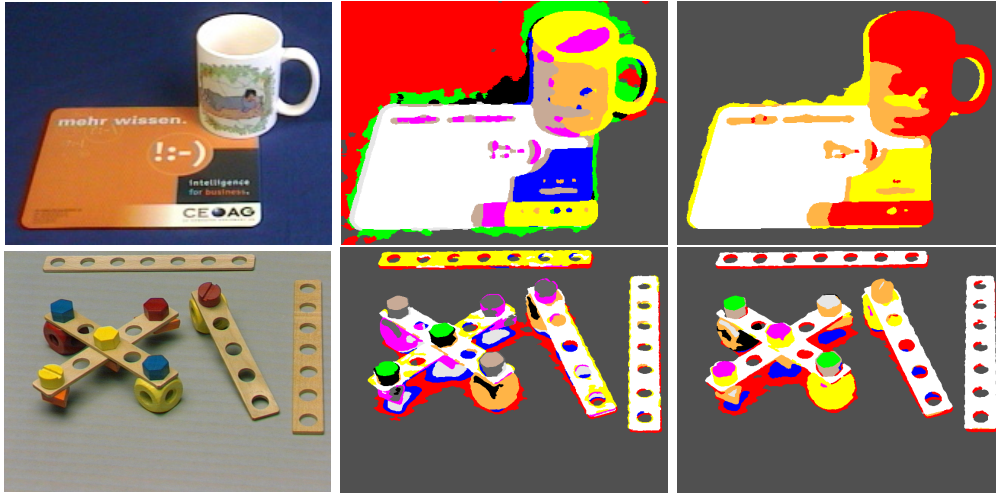


Figure 2.12: Example image from the office and the **baufix**[®] domain and results of the mean-shift color segmentation applying color radius parameter $p_r=3.0$ and $p_r=7.0$, respectively. Other parameters are constant, $N_{min}=N_{con}=648$, $N_{reg}=64$. Each detected significant color is coded by one pseudo color tone.

middle of Fig. 2.12. The main object boundaries are mostly detected, except from the rhombs within the **baufix**[®] example, but there are always several segments representing inner structures of the object regions that have to be taken into account for object segmentation. The number of color classes and the number of segments decrease with increasing the color radius, as shown in the right of Fig. 2.12. Although, here also the main object boundaries are detected and even less segments representing inner structures occur, single contours are not located as well as with the former threshold. For example, the lower contour of the cup is mixed with the edge of the pad and the yellow cube is merged with parts of the upper bar.

Generally, due to the image feature clustering approach, segments are assigned to few globally comparable color classes. Subsequent color classification or other assignment processes may benefit from this.

2.2.4 Image Segmentation by Pixel Color Classification

If the significant colors of an image domain are known a priori, this domain specific knowledge can be exploited for segmentation, see Sec. 2.1.3. Under the additional precondition that the illumination remains constant and thereby no significant changes in the color appearances arise, an appropriate static classifier for color values can be generated. The resulting label image is segmented by the comparably simple search for connected components.

Segmentation by pixel wise classification was successfully applied to the **baufix**[®] domain [Heid 96a], where the significant **baufix**[®] colors are known a priori. A static polynomial color classifier is trained based on a test set of appropriately labeled color val-

ues and applied for pixel wise classification. After classification the resulting image is smoothed within a 11x11 window by setting the central pixel to the color class that holds the majority of pixels in the window. The smoothing reduces noise caused by individual classification failures. Connected components are determined for the smooth label image.

For the practical use of this approach to image segmentation the classification is not explicitly calculated for each pixel. Instead a lookup table that contains an entry for each occurring color value and the assigned color label is determined once in advance and used for each segmentation procedure. Doing so the segmentation for an image is reduced to generating the label image from the original via the lookup table, smoothing it and searching connected components.

Under the assumption of constant conditions concerning occurring color tones and illumination, this procedure delivers very fast results of good quality, as shown in Fig. 2.13.

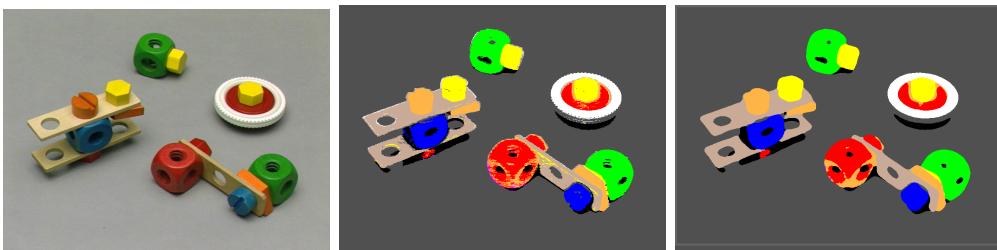


Figure 2.13: *baufix*[®] example image and results from pixel wise color classification and and subsequent smoothing.

2.2.5 Perceptual Grouping of Contour Information

In contrast to segmentation systems that search for homogeneous areas providing similar features, edge based segmentation identifies discontinuities in the feature distribution, see Sec. 2.1.2. Edges and corners are thought to play a role in human attention and recognition mechanisms [Sano 98] and are used as basic features in technical systems for segmentation [Cufi 02] and also for recognition systems, e.g., [Brau 98, Mali 87].

Candidates for edge pixels are identified from the image from analyzing the feature distribution. Neighbored edge pixels are summarized to elementary contours that represent more compactly, for example, parts of an object boundary. Larger image structures, like the surfaces of an object, will generally be constituted from several firstly disconnected elementary contours. For reconstructing them, appropriate contour elements have to be grouped together.

The grouping process of elementary contours is motivated by the Gestalt laws that are known for long terms in psychology [Wert 23], [Lowe 85]. Several approaches attempt to make use of the Gestalt laws for designing the perceptual organization of technical systems [Sark 94], [Kubo 00]. In [Mass 95], [Schl 01] a hierarchical grouping process for elementary contours following the ideas of the Gestalt laws was developed, as shown in Fig. 2.14.

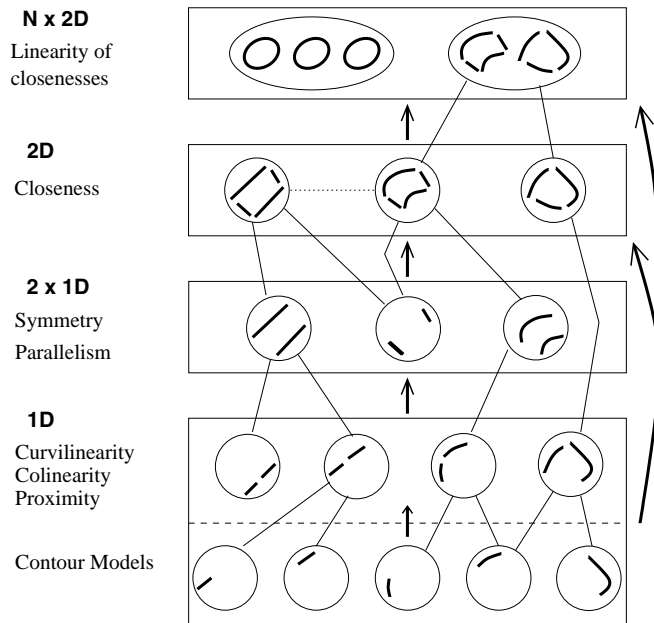


Figure 2.14: Hierarchy of contour based grouping mechanism (reprinted from [Schl 01]).

The procedure of generating contour based hypotheses starts by detecting edge pixels within an image. This is done by a standard Sobel filter after smoothing the images by a median filter. Edges are thinned by applying a non-maximum suppression. After binarizing using a hysteresis threshold edge pixels are identified. In order to reduce the amount of data and to get to a more robust description the resulting edge pixels are approximated by contour primitives, namely line and ellipse segments. Within a first step, edge pixels are described by several, differently parameterized contour models. The best fitting model is chosen within a separate second step according to an error function based on the distances between parameterized contour model and edge pixels. The contour models are in principle allowed to overlap each other, which implies that one edge pixel may be approximated by one or more contour models.

These contour models are the basis for the following hierarchical process that groups together disjunct contour models that are neighbored to each other. In order to concretize the term of neighborhood or proximity, experiments with humans were done, resulting in areas for the perceptual focus, that depends on the grouping task to be done (for details s. [Mass 95]).

In the first step linear structures are searched for, i.e. pairs of line segments that are neighbored and collinear and pairs of ellipse segments, that are neighbored and curvilinear, respectively. Additional pairs of neighbored contour models constitute proximity groups. The basic contour models and the linear groups generated within the 1D level are referred to as linear groups and treated equally in the following. On the 2x1D level of the grouping hierarchy all linear groups are taken into account in order to identify parallel structures. Parallel structures often occur at the boundaries of rectangular shaped

objects and therefore build the basis for finding these object surfaces in the following step. The groups of level 2D are the closenesses. For identifying these all grouping hypotheses of the lower levels are taken into account and a closed path via different groups is searched for. A closed group represents an image segment, that is thought to be handled as a whole, comparable to the segments resulting from region based segmentation approaches. The elements of the upper level of the hierarchy which are linearities of closenesses will not further be regarded here. For results of the hierarchy based grouping process see Fig. 2.15.

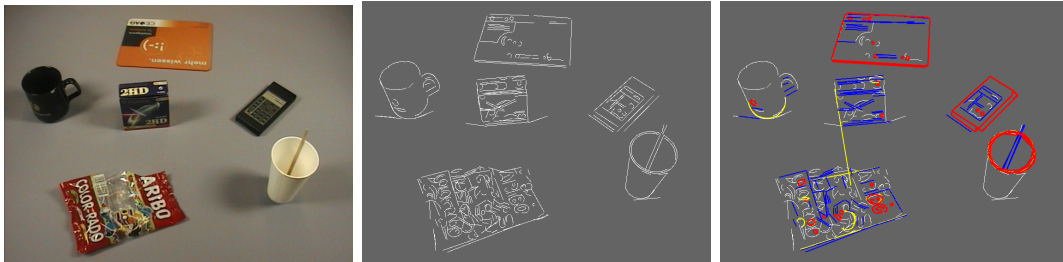


Figure 2.15: Example image (left), contour approximation of edges as lines and curves (middle) and results after perceptual grouping of contours (right).

The groups of level 1D are drawn in yellow, those of level 2x1D in blue and those of level 2D in red, respectively. Notice, that higher level groups overwrite lower level ones in this depiction. The curvilinearity of the bottom of the cup is described well, but it was not possible to find the closed boundary of the cup, because of the great distances between the contour models occurring due to bad contrasts within the original image. On the other hand, the collinear group between the disc box and the bag describes a non existing structure. Parallel structures are found well for the surface structures of the remote control and the stick within the mug, but were not established describing the boundaries of the disc box. The closeness groups found within the example image describe several small and non interesting surface structures like the letters on the bag. On the other hand, for example, the boundary of the mouse pad was found as a closeness. The mouse pad was thereby segmented as a whole from the ground, although the surface is not homogeneous in color or texture.

In a postprocessing step the generated grouping hypotheses are evaluated by applying a Markov random field. Based on the great set of redundant grouping hypotheses the globally most suitable hypotheses are rated high (for details see [Schl 01]) and may be chosen by thresholding the evaluation value.

Fig. 2.16 shows another example from the office domain, where objects have common borders. The perceptual grouping here is able to identify the main object structures as closed segments (red). But there are ambiguous solutions as can be seen on the right, where the closeness hypotheses are extracted. Besides the segments belonging to the pad and the cup respectively, there is also one including both objects into one segment. In addition, the handle for the cup is described by several overlapping hypotheses, that also overlap the hypothesis for the whole cup. The given evaluation of the closeness hypotheses makes a first selection. Applying a threshold of 0.6 to the evaluation value

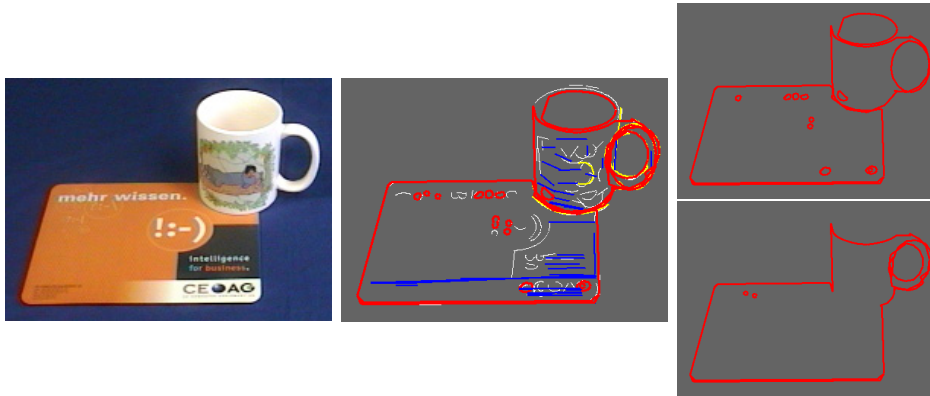


Figure 2.16: Example image from the office domain and results of the perceptual grouping segmentation process (middle). Closed structures (on the right) divided into good (up) and less good (below) by applying a threshold of 0.6 to the evaluation value given.

(parameter setting derived from personal communication to D. Schlüter) leads to the separation shown in the right part of Fig. 2.16. The closeness hypotheses shown in the upper drawing are evaluated good, the ones in the lower image less good with respect to the threshold. Nonetheless, ambiguous structures remain in principle independent from the choice of the value for the threshold. Within the example the handle segment and the cup segment are both evaluated with a value of 1.0. Processes, that incorporate the closeness hypotheses of the perceptual grouping approach must be able to handle non-disjunct results. The groups thereby differ from the strict definition of a segmentation constituted by disjunct segments, given in the beginning of this Chapter.

For integrating grouping hypotheses with region based segmentation results, a process for matching groups and region based generated segments is available, as described in [Schl 98, Schl 00]. The matching results are used there for supporting the grouping process of contours by region based generated segments, on the one hand, and for complementing region based generated segments with more detailed contour information. The matching process is mainly based on distance values, calculated between the support points of the group and the boundary of the segment. A polynom classifier is used for distinguishing, whether the group matches the boundary or a structure of the individually generated segment or does not match at all. The matching process will be applied for integrating groups with individually generated segments that is topic of Chapter 4.

2.3 Summary

Any kind of image segmentation process aims at summarizing image pixel data to segments that are assumed to constitute a semantically meaningful unit. The representation of an image in the form of homogeneous segments decreases the amount of data to be handled by the following processing steps, while preserving as much image information

as possible. As presented in the preceding sections, there exist many approaches to image segmentation. They differ in the definitions of the homogeneity criteria that take into account features from simple grey value to anyway determined object affiliation. Further, the different processing strategies reach from local region growing based on spatial nearest neighbor tests over global graph cut optimization to object based model applications. Each approach exploits different image characteristics and requires more or less specific knowledge about the task and the occurring objects.

From the amount of algorithms those that are suited for a given task has to be selected and parameterized appropriately. Generally, due to the variety of approaches, individual methods will fail, where others are successful and vice versa. Therefore, the integration of several individually generated segmentation results is a promising strategy for achieving an improved final result. This idea is realized by the the proposed integrating framework, as it is described in Chapter 4.

The recognition systems of Chapter 5 that are realized based on the integrating framework uses the presented data driven color segmentation modules. They have been chosen to cover a variety of different segmentation approaches in order to support a successful integration.

As with segments also object information is derived from many sources that may complement one another within an integrated object recognition system. Approaches for object recognition are the topic of the following chapter.

3 Object Recognition

Object recognition is the part within the field of computer vision, that assigns symbolic object knowledge to the image data. This implies that an object recognition system interconnects low level sensory image data with high level symbolic object knowledge represented within the system.

In the following section some basic concepts of object recognition are introduced. This includes the problem definition and a discussion of the general components of an object recognition system. Further, some paradigms for object knowledge representation and acquisition are presented. As the last general aspect, some strategies for integrating object labels from several modules are presented. This leads to improved recognition results, if the assumption is justified that the failures of the individual modules are not correlated to each other.

After the general aspects, some available recognition systems are presented. They follow different recognition strategies and are used within the realized integrated systems presented in Chapter 5.

3.1 Basic Concepts

Before addressing some general aspects in the field of object recognition, first, the definition of a recognition problem containing the demands for the information content of the resulting object hypotheses, are discussed.

3.1.1 Recognition Task: Detection, Segmentation, and Labeling

The main information about an object that is expected to be delivered from a recognition system is the object label. For characterizing and comparing systems their abilities of distinguishing the elements of different object label alphabets is therefore commonly used.

But besides object labeling also object detection and object segmentation are subsumed by the general term object recognition. In analyzing and comparing recognition systems one must take into account, whether detection and segmentation are done within the system, or such information is assumed as a prerequisite, or this aspect is not addressed at all. Of course, the requirements for the information content of an object hypothesis depend on the predefined recognition task. Assume, for example, the recognition task is to distinguish the different objects of an image database, like the commonly used Columbia Object Image Library (COIL) [Nene 96]. Each image of this database contains one centered object and provides a homogeneous background. For this task, object detection and segmentation can be avoided, due to the constraints

of the image material. The same situation occurs generally for all tasks, where an image as a whole has to be distinguished from others, e.g. [Swai 91],[Mura 95],[Schi 96],[Schm 97].

However, think of quality inspections for industrial constructional elements, where an exact segmentation of the element shape is necessary to detect the differences from the demands. It depends on the given constraints, whether the object must be detected within the image first or whether knowledge about the object position is available from the system design, like it is in e.g., [Khaw 96], [Stos 04]. Without exploiting prior knowledge and for images containing more than one object, their detection is an essential part of the recognition system.

An object region segmentation is not necessarily part of the recognition system, for example, for systems that aim at verbal communication with the user concerning objects, e.g., [Heid 05], [Haas 04]. Also for detecting given objects within other images which is one possible application of content based image retrieval systems, detailed object segmentation is mostly not necessary, e.g., [Smeu 00].

But for analyzing less restricted scenes, where, in principle, many objects may occlude each other, object segmentations are generally needed. Final conclusions about the number of objects within those scenes are solely possible based on the segmentation of the object borders within the image. This situation is given, for example, for the analysis of assemblies that is based on the recognition of their parts, e.g., [Khaw 96], [Bauc 02].

These considerations show that the term object recognition system is used for a system that classifies a formerly detected and segmented region to represent a specific object, as well as, for a system that generates object segmentation and classification results from less restricted image material. For characterizing object recognition systems besides the object labeling also their assumptions and approaches concerning the aspects of object detection and segmentation have to be taken into account.

3.1.2 General Components of Object Recognition Systems

Recognition systems are roughly distinguishable into two categories concerning their recognition strategy [Tous 78]. Systems that start processing with the raw image data and go further to higher levels of abstraction are called data driven or bottom up systems. Those that start from expectations or domain specific knowledge and search for the fulfillment of those abstraction within the image data are called conceptually driven or top down systems. As introduced before, both reliable data driven features and knowledge based approaches are desirable and necessary for building successful recognition systems.

The general components of an object recognition system, as identified by [Dick 99] and shown in Fig.3.1 remain constant, independent of the applied recognition strategy.

Based on the image data, suitable features are extracted and groups of features are generated. The chosen recognition strategy determines the details of the processes. Data driven processing generates groups based on feature characteristics, like a pre-defined feature homogeneity criterion, as described already in Chapter 2. The presented perceptual grouping of contour information follows more complex but still general strategies for generating groups without exploiting details of the object database.

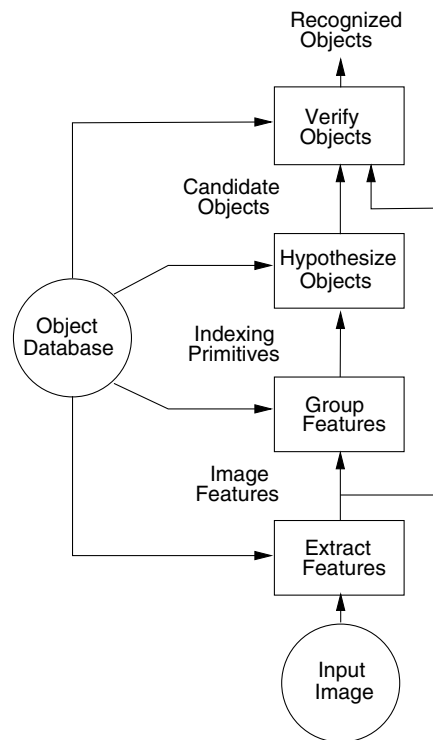


Figure 3.1: Components of an object recognition system (redraw from [Dick 99])

The data driven generated groups are used for indexing the object database by applying a suitable matching approach.

Conceptually based processing determines groups that matches the object context by projecting model knowledge either down to the image features or to an intermediate level like data driven generated segments.

The central aspect of each object recognition system is the representation of object knowledge and the interrelated matching process which will be discussed in more detail in the following section. Object hypotheses are the result of the matching process. Dependent on the recognition task and the chosen methods, each hypothesis consists of an object label, the object position and/or segment information. A subsequent verification step supports or disputes object hypotheses and clarifies disambiguities. Verification can be done by just ranking the object candidates based on a measurement derived from the matching process. If additional object knowledge either defined for each individual object or based on the context of several objects is available, this knowledge may be exploited to extend incomplete results, as will be discussed below.

The following subsection deals with some general aspects of object knowledge representation used for recognition systems.

3.1.3 Object Knowledge Representation

Within the last decades of object recognition research many approaches for object knowledge representation are proposed.

The Development of Object Modeling Techniques

In [Kese 01a] the development from the very general geometrical models of the 1970's to the exemplar appearance based models of the 1990's is drafted. The early systems represent general prototypical objects by their decomposition into volumetric basic parts, like generalized cylinders or the more general set of distinct volumetric shapes, called geons. Matching those general high level components to reliably detectable low level image features turned out to be generally impossible. For realizing recognition systems only idealized images are suited, where feature extraction and grouping are feasible. Image data has to be adapted to these object models, for solving the recognition task.

In order to enable systems to cope with more realistic image data, models become more sophisticated in the following. Detailed geometrical characteristics are modeled, for example, in three dimensional CAD models. This requires, firstly, the reliable detection of the corresponding feature groups, like contours, within the image data. Furthermore, these feature groups have to be assigned to characteristic object model features which depends, in general, on the viewpoint. Either hierarchical or graph based structures containing several view dependent model features are used, in order to disburden the matching process from the view dependence, e.g., [Bowy 90],[Gigu 91]. Establishing those geometric models for new objects requires more or less manual work.

The approach that becomes popular and computationally feasible in the 1990's is storing and matching appearance based object models. Models are acquired directly from image data either by storing characteristic image patches or by calculating and storing a prototypical set of suitable features from the image patches. In order to cope with slight variations within the appearance of an object, the mean values of several image patches are taken into account for the model generation. This implies that for acquiring new models appropriate image material from the new object has to be generated and integrated in the existing framework. But no or less manual modeling work is needed. Real image data is used for the model generation process and the models are applicable to real image data also for the recognition step. Due to the fact that the appearance of an object may vary significantly dependent on the view points, most appearance based models represent the appearance of one view of an object. For recognizing objects without predefining one view either some characteristic views and the appropriate view based object models have to be taken into account [Ullm 91] or the appearance based model must become more complex and flexible to represent several viewpoints, as well as, their characteristic appearance [Mura 95]. Additionally, the appearance based model generally represents one exemplar object, while another exemplar of a geometrically or functionally equivalent object class is content of a second independent model.

Going along with the drafted development within object recognition research from geometric to appearance based models the problem definition altered from generic to exemplar object recognition [Kese 01a]. Due to their different tasks, systems applying geometric models are hardly comparable to those applying appearance based models.

Part Based versus Holistic Modeling

The main object units that occur within the resulting recognition hypotheses are determined by the recognition task. But the object models may be generally structured by defining parts and their relations. Parts thereby may be characteristics that are easy to detect within the image features, like the corners within a shape based model [Lour 98], or parts may be semantically meaningful or functional sub units, like the eyes and the nose within a face [Wisk 97] or the handle of a hammer [Rivl 95]. Several levels of semantic object information are valuable for enhancing the resulting object hypothesis with part information. In addition, the part definition allows the representation of a great object variability with one object model, because many constellations of sub objects can be modeled by appropriate definitions of part-to-part-relations. This flexible knowledge representation is the prerequisite for recognizing the flexible part constellations.

Graph based approaches are commonly used for representing topological or geometrical constraints between parts, where the relational information is stored within the edges, e.g., [Lade 93], [Wisk 97], [Lour 98], [Bauc 04a]. Matching image features to object model data then becomes a graph matching problem. General frameworks for representing functional or semantic knowledge about parts and their relations are graphical models. Semantic networks, see, e.g., [Ball 82], [Brac 85], [Niem 90], and Bayesian networks, see, e.g., [Pear 88], [Jens 96], [Jord 99] provide formalisms for structured object knowledge representation and knowledge propagation.

In [Hans 78] a system for understanding natural scenes is proposed, where a set of coupled semantic nets is used for representing object knowledge at different levels of abstraction. They reach from vertices and edges over regions and surfaces to objects and scenes. A Bayesian network is used, for example, in [Luo 05] for integrating low level features and semantic labels for image understanding tasks. The system is applied to the automatic detection of the main subject of photographs. The fixed structure of the Bayesian network thereby represent the specific aspects of the task, like the influence of face detector results or the general dependence between an object location and its probability to be main subject. The conditional probabilities connecting the different influences represented within the nodes and steering the knowledge propagation through the net are trained based on training data.

Within those structured approaches of knowledge representation the designer decides about the features and the object part relations that he considers to be discriminative and models this task specific and subjective knowledge explicitly. A commonly used procedure to avoid some of this manual work, but to conserve the advantages of structured knowledge representation, is implementing a hierarchy or cascade of classifiers. They are called multiple or learning classifier systems, see, e.g., [Lanz 03], [Oza 05] for an overview and some state of the art systems. (General classifier combination strategies that are independent from the knowledge representation aspect will be discussed below.) Classifier systems acquire knowledge about the relations between part during the training phase of the higher level classifier, instead of explicitly modeling them. Doing so, the two aspects of recognizing parts and joining parts to a whole are separated from each other. For example, [Moha 01] proposes a classifier system for the recognition of persons. Individual classifiers are trained and applied for the parts of a body, like arms, head, etc, before a second level classifier recognizes the whole person, based on the

part classification results. The decision about relevant parts thereby is taken by selecting the training data for the component classifier manually from the image material.

Favoring machine learning approaches instead of manual adaptations leads to complex holistic appearance based object models. Those approaches require large amounts of preprocessed training material for estimating many model parameters and sophisticated statistical matching processes, but they avoid further manual object modeling effort. Good discriminative power is achieved for textured objects like the COIL objects [Mura 95] or human faces [Turk 91], [Viol 04]. Heidemann and Ritter present a holistic appearance based model for assemblies of *baufix*[®] elements [Heid 96b] and show its limitations in discriminating exemplars, that differ in details. For analyzing those differences a decomposition of the complex object into its elements is proposed. Another drawback of holistic object models arises in case of occlusions. In this case, a significant part of the object is probably not visible and does not contribute to the feature calculations, but these just partly appropriate features are matched to the model features concerning the whole object. With part based recognition approaches, the situation of occluded parts may be taken into account in the realization of the matching strategy and then, it depends on the non occluded object part, whether this is discriminative enough for recognition.

Combining the advantages of part based approaches concerning their performance in case of occlusions and their model compactness with the machine learning abilities of holistic appearance based models leads to the approach of generating holistic models for more or less small object parts. For detecting the characteristic parts in the image, so called interesting or key points are taken as candidates. They are located by searching for special characteristics within the pixel values. The arrangement of the points, on the one hand, and/or characteristic features calculated at those points, on the other hand, are exploited for generating object models. Desirable characteristics of those points and their features are stability against changes in view point, lighting and scale. Several methods for key point detection are proposed, see, e.g., [Schm 98], [Heid 04], [Lowe 04]. The cloud of interesting points of a query image as a whole is used for image retrieval within a database by [Schm 97], [Mohr 97]. Thereby, local invariants are calculated at different scales and matched individually without taking into account relations between the points and without assigning any semantics. The matching results for the individual points of the cloud are evaluated as a whole in order to retrieve the best fitting image from the database. Heidemann [Heid 98] uses interesting points for detecting parts of *baufix*[®] elements. The individual points are classified appearance based, without taking into account other points and without any further object model based on the parts. Lowe [Lowe 04] detects key points and calculates characteristic features at their positions, the so called SIFT features. For object detection and recognition, the key points and features are calculated for the image and matched to model data acquired from training images. For verifying the matching result, feature points that agree on an object and its pose are clustered and those clusters are checked by a detailed geometric fit to the model. Query objects can be detected within cluttered scenes and in the presence of occlusions by exploiting characteristic image features located at the key points and information about the location and arrangement of the points.

Appearance Based Object Categorization

If appearance based features are invariant between exemplars of one class, those models are also applicable to object categorization. Weber et al. [Webe 00] present a system for object categorization, where object classes are represented as flexible constellations of parts which are constituted by image patches. The generated model consists of a number of parts and the shape that describes the mutual geometric relations of the parts. The initial set of candidates for object parts is given by image patches at key points of training images. Prototypical parts are calculated from them by clustering the patches applying a vector quantization approach. For determining the object class model possible constellations of a given number of the prototypical parts are generated and tested on a validation data set until no further model improvement occurs. The approach is applied to faces and the backview of cars. This approach of detecting object parts based on image patches at interesting points and their clustering is also used by [Agar 04] for generating a part vocabulary for the sideviews of cars. For the recognition step, parts are identified within an image and their mutual relations are determined by distance and direction. The indices of the occurring parts and their mutual geometric relations are used as features for a classifier. The classifier avoids the explicit probabilistic model is for the part constellation used in [Webe 00].

Leibe et al. [Leib 04a] extend the approach of [Agar 04] by an implicit shape model used for the recognition step and an additional model based segmentation step. For each characteristic part of an object class its position relative to the object center within the training images is stored. Additionally, a segmentation mask for the part at that position is extracted from manually segmented training images and stored. Doing so, with the detection of a part during the recognition step hypotheses for the object center position and the object boundary are available. A probabilistic framework integrates the contributions from all parts to the final object segmentation result. Leibe and Schiele [Leib 04b] extend the approach further to multiple scales in using scale invariant interesting points and consequently different characteristic parts for several scale levels.

Ullman et al. [Ullm 01] also propose an object categorization system based on automatically detected parts. In contrast to the systems presented above, the characteristic image patches, the so called fragments, are not determined by evaluating general purpose interesting point detectors. The fragments are identified specifically for an object class by comparing training images containing an object of the class with each other and with images not containing an object of the class. Several scales are involved in those comparisons. For object recognition the results of the individual fragment identification for a test image are integrated using two different definitions for the combination scheme. A simple scheme that bases primarily on the presence or absence of fragments and a more complex scheme that models the probability distribution of the fragments are shown to deliver comparable results.

The latter approaches to object categorization take a step from the conventional appearance based recognition of exemplars to the more general recognition of object classes using appearance based models. Keselman and Dickinson [Kese 01b], [Kese 05] go further in the direction of acquiring generic object models from the appearance of exemplars. They aim at generating the invariant shape properties of an object class based on the segmentation results for each object. The segments concerning one object are

summarized within a region adjacency graph representing an exemplar segment based appearance of the object. The model generation process finds common representation for the class based on the exemplar graphs by iteratively merging neighbored regions and comparing the occurring graphs to each other. The method is suited to generate an object model from exemplar sideviews of cups that consists of the characteristic regions also occurring in line drawings of cups. The approach thereby decreases further the gap between appearance based and generic object models.

The preceding sections should give an idea of the great variety of existing approaches to object knowledge representation and the resulting different methods of determining symbolic object labels. Independent of the internal representation of object knowledge, the used features and features groups, and the applied recognition strategy, the heart of each object recognition system constitute one or more modules that finally assigns symbolic labels to more or less preprocessed image data, the classification modules. Mostly, different features, different levels of symbolic data and different classification methods contribute their part of valuable information for solving a recognition task. Besides integrating these aspects implicitly into one classification module, systems realizing combinations of multiple classifiers are in common use. In the following, basic general combination schemes for results from different classifiers are discussed.

3.1.4 Classifier Combination

Combining several classifiers for solving a given task has mainly two reasons: efficiency and accuracy [Kitt 98]. A large number of general combination schemes exist, see [Xu 92], [Jain 00] for an overview. In [Jain 00] various combination schemes are grouped into three main categories according to their architecture: 1) cascading (or serial combination), 2) hierarchical (tree-like), and 3) parallel. In the cascading architecture, individual classifiers are invoked in a linear sequence. The number of possible classes for a given pattern is gradually reduced as more classifiers in the sequence are invoked. For the sake of efficiency, inaccurate but cheap classifiers (low computational and measurement demands) are considered first, followed by more accurate and expensive classifiers.

In the hierarchical structure, individual classifiers are combined into a structure that simulates a decision tree. The tree nodes, however, may now be associated with complex classifiers. This structure is flexible in exploiting the discriminant power of different types of features.

In the parallel architecture, all the individual classifiers are invoked independently and their results are integrated afterwards by a combination step. Most implemented systems rely on this parallel architecture [Jain 00], because the systems are flexible in exchanging individual classification modules.

Within the parallel combination architecture there are several approaches for designing the individual classifiers. Either the classifiers follow the same strategies and work on the same feature based representation of the input pattern or not. In the first case the individual classifiers are of equal type, like neural nets with given net topology or k-nearest-neighbor classifiers using the same measurement vector, but vary in their parameter setting. This variation occurs due to different training settings, namely the choice of training samples or the choice of training parameters like distance functions

and thresholds [Opit 99]. The ensemble is usually better than an isolated classifier in dealing with outliers and non optimal training sample constellations. The opposed strategy is to set up classifiers that rely on explicitly different internal representations just solving the same classification task [Kitt 98], [Haym 02].

For the combination step, Jain et al. [Jain 00] distinguish between static and trainable combiners, on the one hand, and adaptive and non adaptive schemes, on the other hand. Combination schemes that take into account the confidence value delivered with the classification result at least for the most probable class are called adaptive, while non adaptive algorithms treat all their inputs the same. Static combiners treat the results of the individual classifiers along a fixed decision rule for deducing their result, e.g., [Kitt 98], [Haym 02]. The decision rule remains constant and independent from the involved modules or image data. In contrast to that, trainable combiners generally lead to further improvements [Chou 03], but are specialized to the given ensemble of classifiers results. Exchanging the classification task and/or individual models necessitates an additional training set that bases on additional training material.

The combination schemes that allow the highest flexibility concerning the usage of available classifier modules and the exchange of individual modules are those that are parallel and static. In order to exploit the information given by the individual modules completely, the combination scheme should be adaptive to the delivered confidence values. Combination schemes providing these characteristics are discussed in more detail in the following.

Parallel, Static, and Adaptive Schemes for Classifier Combination

The simplest form for realizing a parallel, static, and adaptive combination scheme is voting. Its definition and some aspects of voting schemes will be discussed in the following section. The voting strategy and some other static combination rules in common use are covered by a more general theoretical framework for probabilistic integration that is presented afterwards.

Integration by Voting Voting deals with the integration of several input data objects and the deduction of an output data object that is supported most by the input data. Generally each input data object and the output data is accompanied by a confidence value, also called the vote for the data object. The voting scheme specifies, how the result is determined from the input data (for an overview, see [Parh 94]).

Voting has a long tradition in its application for integrating multiple unreliable implementations in order to get reliable data [Neum 56]. At that time it was mainly the unreliable hardware that causes failures. The earliest software voters were used in the design of modular multiprocessors with replicated software. Nowadays, voting schemes are used for integrating the results obtained from several programs that solve the same task applying different methods.

Based on the occurring variations in input and output data and the accompanying confidence values, voting algorithms are classified using a binary 4-cube scheme, as shown in Fig. 3.2.

Input data is defined to be exact, if the values of the input objects are discrete, like

	Input	Output
Data	Exact/ Inexact	Consensus/ Compromise
Vote	Preset/ Adaptive	Threshold/ Plurality

Figure 3.2: Binary 4 cube classification scheme for voting algorithms (redraw from [Parh 94]).

integers or the members of a predefined set of symbols. Inexact input data is continuous, like float values, and has to be compared using a distance measurement function.

Output data of type consensus denote output data elements to be one of the input data elements opposed to output data generated, for example, by the mean of input data objects. It depends on the input data and the task, whether a consensus or a compromise is suitable. If the input data is ordered, as, for example, integer values are, principally both kinds of output data can be calculated. However, if the input consists of a set of unordered data, as, for example, symbolic class labels, a consensus has to be found. The integration of the competing class labels 'mouse' and 'elephant' has to deliver one of both and not a compromise between them.

Preset input votes are defined during the design time of a voting scheme, while adaptive votes accompany each individual input data object.

The output data vote serves for distinguishing two voting strategies. Threshold voting requires the output vote to exceed a given threshold in order to decide on this output. Plurality voting always delivers the output data object that is supported most by the input and thereby guarantees an output, even if it might provide low confidence.

For the integration of symbolic class labels generally delivered together with confidence measurements, the scheme of adaptive or weighted consensus voting of exact input data is suitable, which is defined in Tab. 3.1 [Parh 94].

The selection of one of the weighted consensus voting strategies depends on the participating input data and the integration task. For realized systems, see among many others [Brau 98],[Faym 99],[Camp 00], [Haym 02].

Common to all voting strategies is the definition of the input to consist of N scalar data objects. This results for the task of integrating object label information from several classification modules in each module contributing one label, possibly accompanied by a confidence value. If the individual classifiers deliver their output with confidence information for all classes, this information for less probable classes can not be exploited by the voting approach. In this case more general probabilistic integration methods are suitable that are described in the following.

Definition: Weighted Consensus Voting

Given N input data objects x_n with N associated non-negative real votes/weights v_n , with $\sum_{n=1}^N v_n = V$. Compute the output data object y and its vote w such that y is supported by several input data objects, where the exact input data x_n supports y , if $x_n = y$. w has to satisfy a condition associated with the desired threshold or plurality voting sub scheme.

A. Threshold voting sub schemes:

Unanimity voting: $w = V$

Byzantine voting: $w > \frac{2}{3}V$

Majority voting: $w > \frac{1}{2}V$

t -out-of- V voting: $w \geq t$; if $t \leq \frac{1}{2}V$, then y can be non-unique

m -out-of- n voting ($v_i = 1$): $w \geq m$; if $m \leq \frac{1}{2}n$, then y can be non-unique

B. Plurality voting sub scheme:

No other y' is supported by inputs having more votes

Table 3.1: Definition for weighted consensus voting according to [Parh 94].

Probabilistic Methods for Integration Probabilistic integration methods fully exploit the confidence information given from several classification modules. In contrast to the simpler voting strategy, where only the most confident class delivered by each module contributes with its weight, the probabilistic method generally takes into account also the confidence information given for the less probable other classes. This improves the integration quality, as shown, for example, in [Mohr 97] for the integration of many individual classifications results for image retrieval.

Kittler et al. [Kitt 98] develop a common theoretical framework for probabilistic combination methods and derives from that the voting approaches, as sketched in the following.

The task is to assign a pattern Z to one of m possible classes $(\omega_1, \dots, \omega_m)$ based on R classifiers. Assume that each classifier i represents the given pattern Z by a distinct measurement vector x_i . In the space of these measurements each class ω_k is modeled by the probability density function $p(x_i|\omega_k)$ and its a priori probability of occurrence $P(\omega_k)$. Classes are considered as mutually exclusive that results in one class is associated with each pattern. The pattern should be assigned to class ω_j , if the a posteriori probability of that interpretation is maximal, i.e.,

$$\begin{aligned} \text{assign } Z &\rightarrow \omega_j \text{ if} \\ P(\omega_j|x_1, \dots, x_R) &= \max_k P(\omega_k|x_1, \dots, x_R) \end{aligned}$$

Using Bayes' theorem the a posteriori probability can be written as:

$$P(\omega_k|x_1, \dots, x_R) = \frac{p(x_1, \dots, x_R|\omega_k)P(\omega_k)}{p(x_1, \dots, x_R)}$$

The joint probability distribution of the representations extracted by the classifiers

3 Object Recognition

$p(x_1, \dots, x_R)$ are expressed by the sum of the conditional distributions given the pattern belongs to class ω_k :

$$p(x_1, \dots, x_R) = \sum_{j=1}^m p(x_1, \dots, x_R | \omega_j) P(\omega_j)$$

Assuming the occurring representations to be statistically independent results in:

$$p(x_1, \dots, x_R | \omega_k) = \prod_{i=1}^R p(x_i | \omega_k)$$

where $p(x_i | \omega_k)$ describes the measurement model of the i th representation or classifier. Applying again Bayes' rule to the distribution $p(x_i | \omega_k)$ results in:

$$p(x_i | \omega_k) = \frac{P(\omega_k | x_i) p(x_i)}{P(\omega_k)}$$

Substituting the formulas leads for the a posteriori probability to:

$$P(\omega_k | x_1, \dots, x_R) = \frac{P(\omega_k) \prod_{i=1}^R p(x_i | \omega_k)}{\sum_j^m P(\omega_j) \prod_{i=1}^R p(x_i | \omega_j)} = \frac{P(\omega_k)^{-(R-1)} \prod_{i=1}^R P(\omega_k | x_i) p(x_i)}{\sum_j^m P(\omega_j)^{-(R-1)} \prod_{i=1}^R P(\omega_j | x_i) p(x_i)}$$

Simplifying this expression further by assuming equally distributed classes ω_k ($P(\omega_1) = \dots = P(\omega_m)$) and patterns x_i ($p(x_1) = \dots = p(x_R)$) and substituting the a posteriori probability within the decision rule leads to the, so called, product rule:

$$\begin{aligned} \text{assign } Z &\rightarrow \omega_j \text{ if} \\ \prod_{i=1}^R P(\omega_j | x_i) &= \max_k \prod_{i=1}^R P(\omega_k | x_i) \end{aligned}$$

The product rule is used for classifier combination, but the fact that one classifier can inhibit all the others by delivering a probability or confidence value close to zero is often not tolerable.

Kittler et al. [Kitt 98] derive further the a posteriori probability for cases, where the available discriminatory information is highly ambiguous. This results in the assumption of:

$$P(\omega_k | x_i) = P(\omega_k)(1 + \delta_{ik})$$

with $\delta_{ik} \ll 1$. By substituting the term within the a posteriori probability and neglecting higher powers of the δ_{ik} 's results in:

$$\begin{aligned} P^{-(R-1)}(\omega_k) \prod_{i=1}^R P(\omega_k | x_i) &= P(\omega_k) \prod_{i=1}^R (1 + \delta_{ik}) \\ &= P(\omega_k) + P(\omega_k) \sum_{i=1}^R \delta_{ik} \end{aligned}$$

Using this expression for the decision rule leads to the, so called, sum rule:

$$\begin{aligned} \text{assign } Z &\rightarrow \omega_j \text{ if} \\ \sum_{i=1}^R P(\omega_j | x_i) &= \max_k \sum_{i=1}^R P(\omega_k | x_i) \end{aligned}$$

The sum rule accounts for the sums of all a posteriori probabilities for one class k for deciding for the resulting class. Even if there are many approximations and assumptions to be made on the way to this expression, it shows in experiments to be better or equal to the product rule [Kitt 98], [Alex 01].

Demanding each individual classifier to make a decision for one class before integrating them transfers the probabilistic decision rule into a voting scheme. In the present nomenclature the decision of one classifier for the most probably class can be written as:

$$P_{ki} = \begin{cases} P(\omega_k|x_i) & \text{if } P(\omega_k|x_i) = \max_l P(\omega_l|x_i) \\ 0 & \text{otherwise} \end{cases}$$

Substituting this term to the sum decision rule leads to the plurality voting sub scheme:

$$\text{assign } Z \rightarrow \omega_j \text{ if } \sum_{i=1}^R P_{ji} = \max_k \sum_{i=1}^R P_{ki}$$

Restricting the information delivered from the individual classifiers even more by neglecting the confidence value can be written as:

$$\Delta_{ki} = \begin{cases} 1 & \text{if } P(\omega_k|x_i) = \max_l P(\omega_l|x_i) \\ 0 & \text{otherwise} \end{cases}$$

Substituting this to the sum decision rule leads to the majority voting sub scheme:

$$\text{assign } Z \rightarrow \omega_j \text{ if } \sum_{i=1}^R \Delta_{ji} = \max_k \sum_{i=1}^R \Delta_{ki}$$

The presented derivation identifies many existing decision schemes as special cases of the general compound classification, where all representations are used jointly to make a decision. In experimental comparisons that are presented in [Kitt 98] the sum rule turns out to deliver the integrated results. based mainly on its ability to compensate for estimation errors of individual classifiers.

With discussing these general combination strategies for integrating individually generated symbolic class labels, the part of introducing basic concepts of object recognition ends. In the following realized and available systems are presented that are used for realizing the integrated object recognition systems of Chapter 5.

3.2 Object Recognition Systems

In the following, four available recognition systems are presented, with each of them realizing different object knowledge representation and recognition strategies. Three of them are implemented for recognizing objects of the *baufix*[®] domain, see Appendix B, and one for classifying office domain objects, see Appendix C.

First, a recognition system is presented that is based on a hybrid object knowledge representation scheme integrating holistic and semantic part based modeling for *baufix*[®] elements. Thereby an artificial neural network classifier constitutes the the holistic part

of the system and generates initial hypotheses. The semantic part based object knowledge is used for verifying and evaluating the initial element hypotheses.

The second module does not contain any predefined geometrical or semantic model of the *baufix*[®] objects, but acquires knowledge based on color segmentation results during its training phase. The object units are defined by the strong dependence on segmentation results to be the uniformly colored parts of *baufix*[®] elements, which are identical to the elements, except for the bolts.

The third module provides object detection and classification based on appearance based object models. For classifying the characteristic appearance the color pixel information within an image window of predefined size is used. It does not rely on segmentation results and makes no attempt to object segmentation. In order to address occlusions of the *baufix*[®] elements that are assembled, the object units are chosen to be sub elementary. Relations between sub elements and elements are not modeled within this module but has to be generated by a subsequent module.

Besides these *baufix*[®] recognizers, a shape based recognition for the office domain objects is realized. Based on segments the object silhouette is determined and shape features are classified using a support vector machine.

This short presentation of the modules shows that they follow different recognition strategies which makes them promising candidates for further improvement by integration. However, they provide different kinds of object information concerning the object units and the presence of segment information that has to be accounted for in the integration process. The individual modules are presented in more detail in the following.

3.2.1 Hybrid System Integrating Neural and Semantic Networks

The hybrid recognition system described in the following, consists of a holistic appearance based classification step and a semantic verification step exploiting part based object knowledge [Heid 96a], [Kumm 98]. The system is designed for recognizing isolated or slightly occluded elements of the *baufix*[®] scenario. The original implementation is available and used here. For exemplary results, see Fig. 3.3.

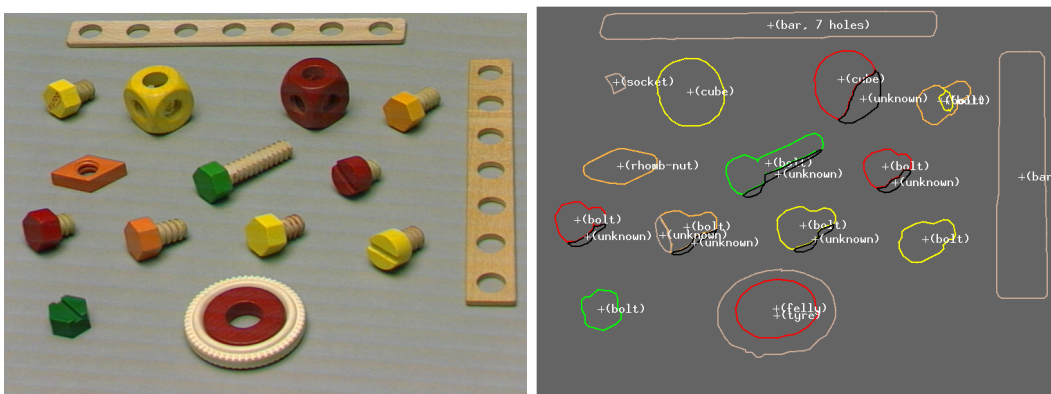


Figure 3.3: Exemplary results of the hybrid recognition system for *baufix*[®] objects.

For detecting objects or, in other words, for focusing the classification system to interesting regions segment information is used. Segmentation results are delivered from the pixel classification approach that is already described in Sec. 2.2.4. These color regions serve as regions of interest for the holistic recognition system that is realized in the form of an artificial neural network of the type local linear map (LLM).

Each centroid of a segment determines the center of an image window with a size in the order of magnitude of a *baufix*[®] rhomb. The edge enhanced intensity image data within the window is taken into account for calculating features based on Gabor filter masks. The features resulting from a training set of images and their assigned class labels are used for the combined unsupervised and supervised training of the local linear map.

The verification step for the initial neural hypotheses is based on part based object knowledge that is modeled in the form of a semantic network realized in the network language ERNEST [Sage 97]. The knowledge base represents object information like the bars containing three, five or seven holes or the decomposition of a screw into its parts head and thread and the correspondence between objects and parts of them to image color regions. Fig. 3.4 shows the knowledge base together with the data base and the connection to the holistic recognition for the example 'screw'.

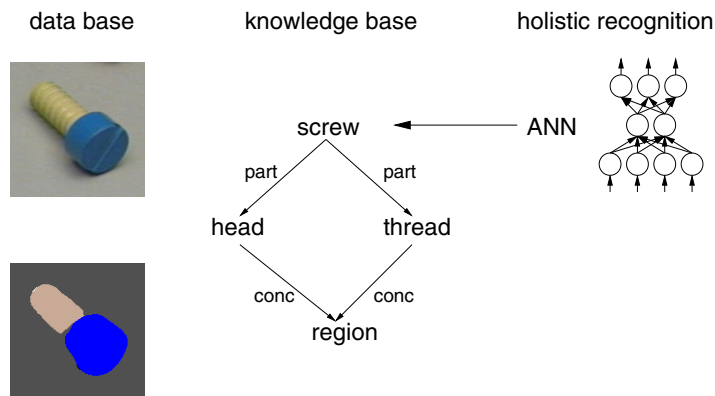


Figure 3.4: Example for hybrid representation (figure redrawn from [Kumm 98])

The knowledge base of the semantic network generally consists of concepts and links between them. For the links three different types are defined namely the decomposition 'part', the specialization 'spec' and the concretization 'conc'. Decomposition links 'part' connect, for example, the *baufix*[®] screw with its head and thread. Further on concepts representing objects at different concretization levels, like the thread or the head and the corresponding color region, are connected by a link of type 'conc'. Finally, a concept and a specialized form of it are connected by a 'spec' link. For example, a concept describing a three holed bar is specialized form of the concept representing the bar.

Fig. 3.5 gives an overview of the hybrid system realized for the *baufix*[®] scenario, including the semantic knowledge base and its connections to the holistic recognition system, on the one hand, and to the segment information, on the other hand. The holistic hypothesis activates the instantiation process for the semantic network at the concept OBJECT and results in a holistic instance for one of the specializations of the concept OBJECT, like SCREW or CUBE. Based on the holistic instance a structured instance

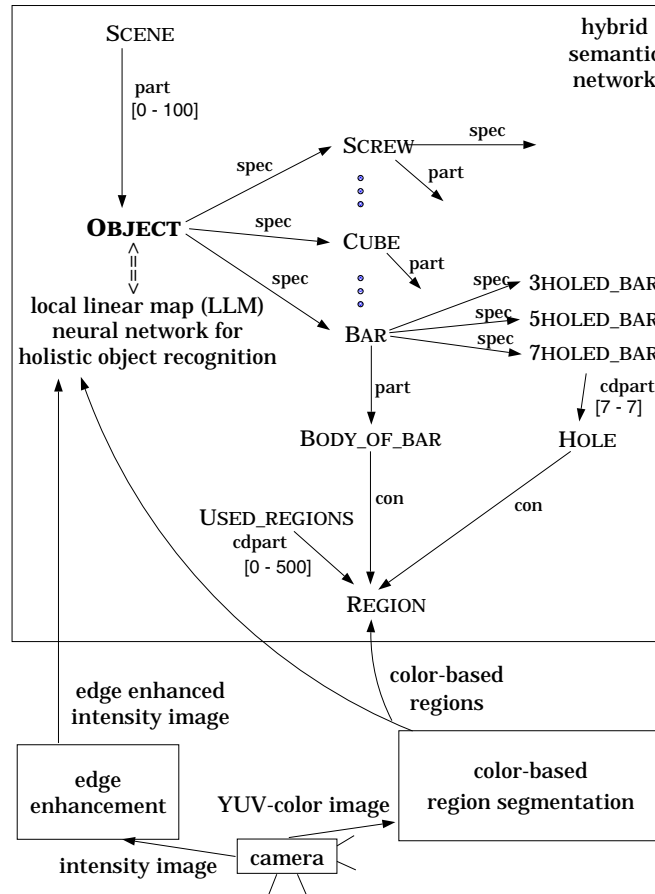


Figure 3.5: Hybrid recognition system constituted from neural and semantic networks (figure from [Kumm 98]).

is generated by exploiting the part based object knowledge stored within the knowledge base. Object parts are instantiated based on their concretizations, the color regions, by exploiting the stored procedural knowledge about suitable color regions for specific object parts. The final structured instance contains the complete information about the holistic object, its parts and the assigned image regions. The holistic recognition system delivers up to three competing results accompanied by an evaluation that are analyzed semantically. The instantiation process starts with three holistic instances in parallel and either verifies or rejects them.

Measurements of the recognition rate of the system are done in [Heid 96a] and [Kumm 98]. The recognition rate for the holistic system on a testset is about 80% that is improved by the semantic verification achieving finally 90%. The hybrid recognition system relies on segments delivered from the pixel based color classification approach and, therefore, requires constant lightning conditions. Additionally, due the predefined knowledge base has to be reimplemented, when transferring the recognition system to other domains. However, the hybrid approach turns out to be is highly reliable for the task it is designed for, namely the recognition of isolated and slightly occluded objects.

Feature Evaluation

The different nature of features determine different kinds of processing for creating object label hypotheses from the feature values. There are either features with discrete values and providing no defined order or features that provide such an order, because the feature values are numerical values.

For evaluating the discrete and non ordered feature values for estimating object labels, one estimation value per feature value and per object label is necessary. It is convenient to arrange this data within a matrix. For the *baufix*[®] system implementation the matrix representation is used for the feature '*baufix*[®] color class'. It is composed by multiplying one matrix that contains the fixed object color code based on prior knowledge with another, that takes into account color classification errors based on a test set. The color code matrix is defined by the *baufix*[®] element set as, for example, the color 'blue' occurs for the heads of the bolts and the cubes but not for bars or rhombs. The color code matrix, therefore, gets for the feature value 'blue' entries of 0.33 for the object labels 'cube', 'head of a round bolt', and 'head of a hexagonal bolt', respectively, and zero otherwise. For taking into account color classification errors, a color error matrix is generated from a test set. For the *baufix*[®] task the color classification process is very reliable and therefore most entries of the color error matrix are zero. Multiplying the color code matrix and the color error matrix results in the evaluation matrix that contains estimations for object label confidences based on one value of the feature '*baufix*[®] color class'. Parts of this matrix are shown in Tab. 3.2. Note that different values of this feature are very different in their discriminant power for the *baufix*[®] recognition task.

	<i>3-hole-bar</i>	<i>5-hole-bar</i>	<i>7-hole-bar</i>	<i>cube</i>	<i>rhomb</i>	<i>felly</i>	<i>tyre</i>	<i>socket</i>	<i>fling</i>	<i>very short thread</i>	<i>very long thread</i>	<i>round bolt head</i>	<i>hexagon bolt head</i>	<i>no part</i>
red	0	0	0	0.25	0	0.25	0	0	0	0	0	0.25	0.25	0
orange	0	0	0	0	0.33	0	0	0	0	0	...	0	0.33	0.33
white	0	0	0	0	0	0	0.83	0.17	0	0	0	0	0	0
wood	0.065	0.065	0.065	0	0	0	0	0.013	0	0.065	0.065	0	0	0.0065
...														

Table 3.2: Parts of the estimation matrix encoding the relation between the color feature value and object labels for the *baufix*[®] task.

Besides those features that deliver non ordered and discrete values, most features result in numeric values, whose order is obvious. For estimating the confidence of an object label derived from a given feature value the probability for an object label occurring together with a special feature value is determined from exemplars within a training set of images. A histogram for each feature and each label that plots the frequency of label occurrence for a given interval of feature values, the histogram bin, is desirable. Thereby, the silhouette of the histogram function strongly depends on the mostly fixed

definition of the width of the histogram bins. If the bins are too small, many of them gets no entry and most of them will contain very few data points each. If they are too large, they subsume probably inner structures of the data. In general, bins should be small in areas of many data points in order to account for the data structures and may be wider in areas of few data points.

This motivates a different definition of the histogram, where the value of the histogram function $h(x_n)$ is set for each individual data point x_n dependent on the neighbored points x_{n-1} and x_{n+1} . Each of the N data points x_n is assigned to its own histogram bin with a normalized area A_N of size $1/N$. The width of the bin belonging to data point x_n is half the distance from x_{n-1} to x_n plus half the distance from x_n to x_{n+1} . The value for the histogram function $h(x_n)$, then, results from the area normalization, see Fig. 3.7.

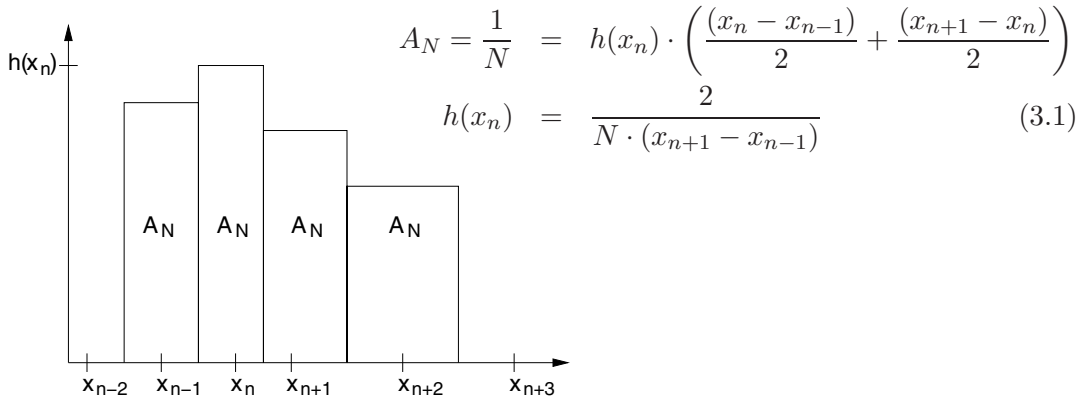


Figure 3.7: Definition of the feature value histogram function for a given object label based on N data points x_n (redraw from [Menz 99]).

In order to achieve a more compact representation and, particularly, to cover the whole range of possible feature values without holes, a functional representation of the confidence value for an object label dependent on the feature value is desirable. This functional representation is generated by approximating the histogram data by a mixed normal distribution function, using the vector quantization approach LBG [Lind 80]. The function is determined by Γ Gaussian functions with individual weights w_γ and each of them parameterized by its mean value μ_γ and its standard deviation σ_γ :

$$f(x) = \sum_{\gamma=1}^{\Gamma} \frac{1}{w_\gamma} \frac{1}{\sqrt{2\pi}\sigma_\gamma} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu_\gamma}{\sigma_\gamma}\right)^2} \quad (3.2)$$

For evaluating the approximation of the histogram data by the mixed Gaussians the difference area between the two functions of Eqn. 3.2 and Eqn. 3.1 is estimated by:

$$F_{err} = \sum_{n=1}^N F_n = \sum_{n=1}^N |h(x_n) - f(x_n)| \cdot \frac{(x_{n+1} - x_{n-1})}{2}$$

According to the error function the approximation is optimized by testing several

values for the number of Gaussians Γ to be used by the LBG vector quantization and determining the parameters of the function $f(x)$ that provides minimal error.

Fig. 3.8, left, shows the histogram data together with the approximation function for the feature 'size' and the object label 'hexagon bolt head'. The right side shows the approximations for some additional object labels and the feature 'size'.

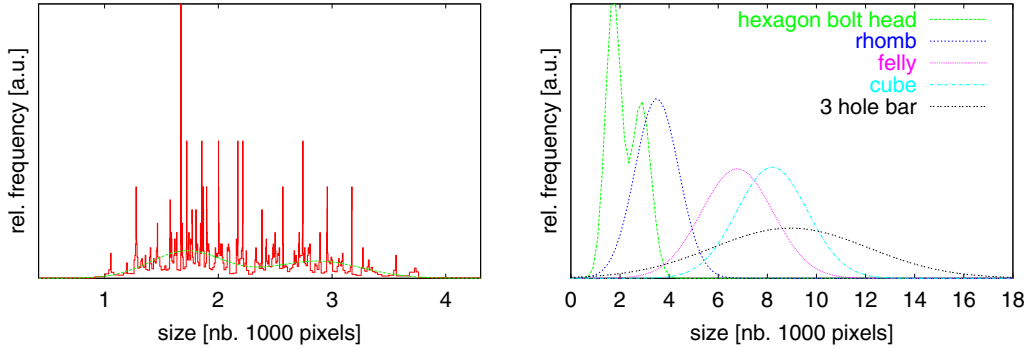


Figure 3.8: Histogram data and approximated function for feature 'size' for object label 'hexagon bolt head' (left) and approximated functions for some additional objects (right).

Calculating the approximation function according to Eqn. 3.2 for each feature k and each object i results in a set of approximation functions $f_i^k(x^k)$. Based on this set and one value x_0^k of feature k object label estimations for all objects i are available and subsumed to a normalized object estimation vector with the components $v_i^k(x_0^k)$:

$$v_i^k(x_0^k) = \frac{f_i^k(x_0^k)}{\sum_{i=1}^I f_i^k(x_0^k)}$$

The estimation vectors resulting from classifying different features are combined in the subsequent, independent step as described in the following.

Combining Feature Evaluations by Extended Voting

In Sec. 3.1.4 general approaches for combining classifier results are described. For parallel, static and adaptive integration there are, on the one hand, the voting strategies that are characterized by accounting for one object label and its evaluation per classification module. The probabilistic integration schemes, on the other hand, take into account the complete object label confidence vector. Menzel [Menz 99] implements and tests several voting schemes and the probabilistic sum rule for integrating the region based feature classifier for the *baufix*[®] scenario. The best results delivers a combination of voting and probabilistic integration that Menzel called 'extended voting'. The integration is basically done following the probabilistic sum rule but some basics of the voting ideas are implemented for rejecting probably bad results. This is firstly the m-out-of-n strategy implying that at least m modules have to support the final objects label. After-

wards a threshold for the minimal evaluation of the final object label is defined also for rejecting probably bad hypotheses.

For the region based *baufix*[®] recognition system in [Menz 99] seven features are taken into account and classified individually, which are '*baufix*[®] color class', 'compactness', 'eccentricity', 'size', 'lines', 'circles', and 'ellipses'. The results are probabilistically integrated where the individual confidence values are additionally weighted with the confidence of the source of information. These weights are set according to the individual recognition rates to 0.228095, 0.106667, 0.156667, 0.188571, 0.112857, 0.085714, and 0.121429 respectively. At least five of the seven modules have to support the integrated label and a minimal final evaluation of 0.005 has to be achieved. Otherwise the object label is rejected. This system delivers for the testset used in [Menz 99] and based on a detailed system containing 17 classes a recognition rate of about 76%, where 90% of all regions are classified, while 10% are rejected due to the threshold settings. The recognition results of this system are dependent on the segments that serve for feature generation. Occlusions that change significantly the size and shape of object regions, therefore, generally cause problems. The test set evaluated above contains non or just slightly occluded elements. Because the selected segmentation approach, the pixel based classification approach, is susceptible to changes in illumination that changes the color appearance, also the recognition system needs a constant environment.

3.2.3 Appearance Based Recognition System

A trainable appearance based object recognition system is presented in [Heid 98] and [Heid 99]. The original implementation of the system by G. Heidemann is used here and presented in the following.

The system does object detection as a processing step independent from the object classification by applying an interesting points detector. At these points the classification step of the system assigns one object class label or the rejection class based on appearance based object models. The system is trained for the detection and classification of parts of *baufix*[®] elements that remains visible within assemblies, where elements are often occluded. The system provides its object labels without further information about object segmentation and about the combination of parts belonging to one element. Exemplary results of the system are shown in Fig. 3.9.

Data Driven Focusing

For recognizing objects within an image that contains generally several objects at arbitrary locations systems that do not rely on segment information can either apply a classifier at any location and analyze the results for simultaneous object detection and classification, like it is done in, e.g., [Viol 04] or they apply the classifier only at locations that are assumed to be probable object locations. Selecting promising locations for applying a classifier is done here and called focusing.

For the data driven identification of characteristic image locations several kinds of key point detectors are proposed, as described in Sec. 3.1.3. As also presented there, key points are often used for representing the characteristics of an image as a whole or for coding characteristics of an object in combination with neighbored points. The

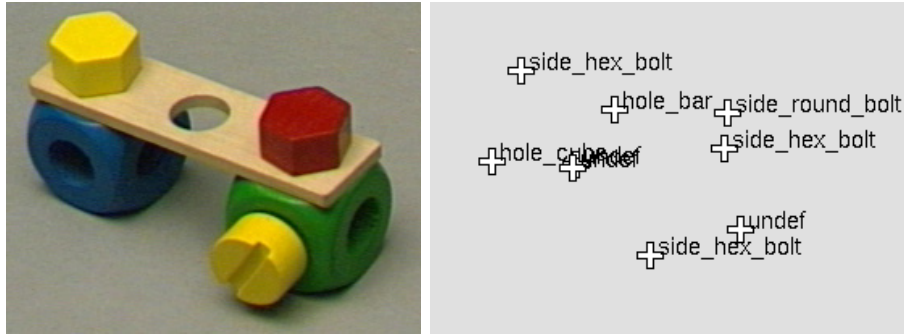


Figure 3.9: Example image and recognition results of the appearance based recognition system applied to parts of **baufix**[®] elements [Heid 98].

key points here are used individually for focusing and therefore called focus points in the following. The focus points determine the location where the classifier is applied and therefore the focusing and the classification step has to be adjusted in order to take care of focusing to the object units that the subsequent classifier supports. Similarly, the classifier has to support object units that can be focussed by the data driven process. For the **baufix**[®] domain these object units are characteristic parts of elements, as, for example, the holes.

The focusing step analyzes three image data based features, namely entropy, symmetry, and color homogeneity and is adapted to the required object units by parameterization.

Local Entropy The local entropy evaluates the local information content of an image area and image areas that carry a high amount of information are assumed to be interesting, e.g., [Kali 96].

The term information content originates from information theory, where the information content of a message is defined over the probability for this message in comparison to all possible messages. The information content is high, if the probability of this message is low and vice versa. A way of calculating the information content S from the probability of a message X_i is:

$$S(X_i) = -\log p(X_i)$$

The expected value for the information content of messages X_i then is given as:

$$E\{S(X_i)\} = -\sum_i p(X_i) \log p(X_i)$$

This term matches the thermodynamic definition of the entropy and was introduced by Shannon ([Shan 48]). It is adapted to image processing in taking the pixel grey level value $I(x, y)$ at position (x, y) the role of the message and approximating the probability of the message by the normalized histogram $\tilde{H}(x, y, q)$. The histogram $H(x, y, q)$ is calculated for a window containing $n \times n$ pixels ($n \geq 3$) that is centered at position (x, y) by counting the occurrences of each pixel value q :

$$H(x, y, q) = \sum_{y'=y-\tilde{n}}^{y+\tilde{n}} \sum_{x'=x-\tilde{n}}^{x+\tilde{n}} \delta_{I(x',y'),q} \quad , \text{ with } \tilde{n} = \frac{n-1}{2}$$

$$\tilde{H}(x, y, q) = \frac{H(x, y, q)}{\sum_{q'} H(x, y, q')}$$

Based on the normalized histogram the local entropy or information content can be calculated for every pixel position (x, y) :

$$B_E(x, y) = - \sum_q \tilde{H}(x, y, q) \cdot \log \tilde{H}(x, y, q)$$

The local entropy is used in [Heid 98] for doing rough focusing, i.e. identifying interesting areas with high information content for accelerating subsequent detailed focusing steps. Therefore, the entropy calculation is done based on down sampled grey level images. The attention map is binarized applying a given threshold resulting in the image to be divided into interesting and non interesting areas, as shown in Fig. 3.10 for the example image of Fig. 3.9.

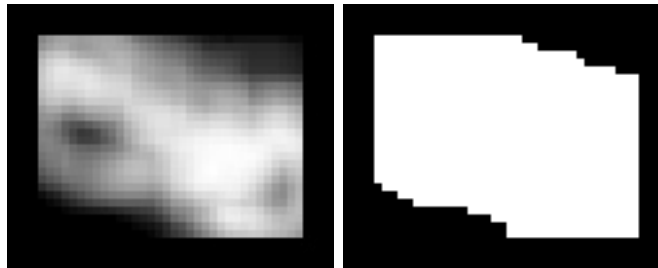


Figure 3.10: Entropy map and result of binarization for distinguishing roughly interesting areas (white) from non interesting (black) for the example image of Fig. 3.9.

Local Symmetry Symmetry plays an important role in human attention. This turns out of several experiments, where the trajectories of humans looking at images were analyzed. For example, in [Loch 87] it was found that people first attend to symmetry axes and afterwards look at the modern art images that are mirrored along these axes. In experiments described in [Bruc 75] people had to find the discrepancies between two almost identical images. They found the differences better when comparing a pattern and its mirrored image than in comparing a pattern and its repetition. For applying symmetry in computer vision Reifeld et al. [Reis 95] introduce a continuous evaluation scheme based on local grey level gradients.

Heidemann [Heid 98] exploits these ideas together with an extension to color information for determining focus points. The process starts with identifying edge pixels p in the input image by applying a Laplace filter and binarizing the resulting image. From

applying Sobel filter masks in two directions at these positions two maps, one containing the gradient amplitude $G(x, y)$ and the other containing the gradient direction to the x-axis $\theta(x, y)$, are calculated.

For calculating the symmetry value for a pixel p pairwise opposing pixels (p_i, p_j) within its environment of radius r are taken into account. The pairs are analyzed independent from each other before the results are summarized leading to the final value for pixel p . For each pair of opposed pixels (p_i, p_j) the relation between the gradient directions of the two pixels, their individual gradient amplitudes and a color similarity estimation are taken into account:

$$B_S(p_i, p_j) = \underbrace{PGF(p_i, p_j)}_{\text{gradient relations}} \cdot \underbrace{GGF(p_i) \cdot GGF(p_j)}_{\text{gradient amplitudes}} \cdot \underbrace{FGF(p_i, p_j)}_{\text{color similarity}}$$

The gradients of the opposed pixel pair (p_i, p_j) are in case of perfect symmetry both parallel to the connection line between p_i and p_j with opposed orientation. These aspects are accounted for the estimation of symmetry by combining the evaluation of each gradient direction relative to the connection line with the evaluation of the relative orientation:

$$PGF(p_i, p_j) = (1 - \cos((\theta_i - \alpha_{ij}) + (\theta_j - \alpha_{ij}))) \cdot (1 - \cos(\theta_i - \theta_j))$$

with α_{ij} denoting the angle between the connection line of p_i and p_j and the x-axis.

The gradient amplitude refers to the contrast of an image structure. Symmetry structures with high contrast are assumed to be more significant and, therefore, the individual gradient amplitudes of the opposing pixels p_i and p_j are taken into account as one part of the symmetry measurement:

$$GGF(p_k) = \log(1 + G(x_k, y_k)) \quad , \text{ with } k = i, j$$

Finally, color similarity between the opposed pixels is evaluated. In taking into account only grey level values, symmetries are found also between two neighbored similarly formed but differently colored objects and between objects and their shadows. These effects are reduced in weighting the symmetry between similarly colored pixels higher than the symmetry between differently colored pixels. For evaluating the color similarity of the opposed pixels p_i and p_j , the color angle or hue $h(p_k)$, a component of the color coordinates of the HSV color space (see Sec. 2.1.1), is calculated for each pixel. The difference between the two color angles is compared to a threshold t_{colSim} , for evaluating the color based symmetry:

$$FGF(p_i, p_j) = \begin{cases} 1 & , \text{ if } |h(p_i) - h(p_j)| < t_{colSim} \vee 2\pi - |h(p_i) - h(p_j)| < t_{colSim} \\ g_f & \text{ otherwise} \end{cases}$$

The choice of the parameter g_f influences the strength of the color similarity evaluation within the overall symmetry value.

The symmetry value for each pixel p is calculated by summarizing the symmetry values for all pairs of opposed pixels (p_i, p_j) within an environment of radius r of pixel p . Focus points are generated from the map containing all the symmetry values by smoothing

the map using a Gaussian filter, binarizing the result and finding centers of blobs in the binarized map. Fig. 3.11 shows the symmetry map and the resulting focus points for the example image shown in Fig. 3.9. The most significant parameter for this focusing step is the radius r determining the environment of pixel p . It determines the scale at which symmetry is accounted for. For the *baufix*[®] task the environment covers roughly the size of a hole. The influence of the color symmetry is selected to be high by setting the parameter g_f to zero, according to the saturated colors of the *baufix*[®] scenario.

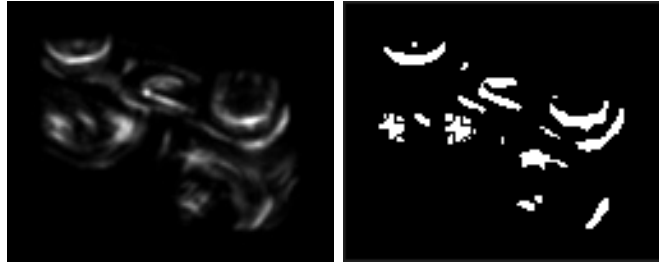


Figure 3.11: Symmetry map and derived focus points for the example image of Fig. 3.9.

Color Homogeneity and Contrast Factor The second independent way of calculating focus points in [Heid 98] is based on homogeneous color regions that provide high contrasts to their surrounding. Candidates for focus points are the centroids of segments that are delivered from the region based color segmentation approach CSC, described in detail in Sec. 2.2.1. For estimating the contrast of a segment to its surrounding the mean local contrast for all border pixels is determined and compared to a threshold. The local contrast is given by evaluating the results of a Laplace filter located at each border pixel.

The initial segmentation results and the selected segments with high contrast determining the focus points are shown in Fig. 3.12 for the example image of Fig. 3.9.



Figure 3.12: CSC color segments and selected segments that provide high color contrast and determine focus points for the example image of Fig. 3.9.

An evaluation of the focusing step, including the entropy, symmetry and color based approaches, is presented in [Heid 04]. The focus points show good stability under changing lighting conditions and object rotations.

Classification of Focus Points

Focus points determine the locations in the image, where a classification step is applied. This implies, on the one hand, that non focussed objects get lost for the recognition system which has to be avoided by parameterizing the focusing step accordingly. On the other hand, the classification step has to deliver an object label for the focussed locations. Therefore, the object label alphabet has to be determined by clustering the focus points of an image training set into semantically meaningful and for the classifier distinguishable object classes.

For identifying parts of *baufix*[®] elements, this results in the following object label alphabet [Heid 98], see also Appendix B, Tab. B.1:

```
part_cube, hole_cube, edge_cube  
part_rhomb, hole_rhomb, side_rhomb  
head_round_bolt, head_side_round_bolt  
head_hexagon_bolt, head_side_hexagon_bolt  
hole_bar, edge_bar  
felly, hole_felly  
tyre  
undefined
```

The classification step constitutes of a combination of vector quantization, local principal component analysis and neural networks. The image data information within a window centered at a focus point is taken into account. The window size generally depends on the recognition task. For the *baufix*[®] parts, it is chosen to cover a bit more than the area of a hole. The color pixel values within the window are summarized to one feature vector used for classification.

For the classifier training a set of feature vectors is generated by localizing focus points for representative sample images and cutting out the appropriate image data. The first part of the training is an unsupervised clustering of the feature vectors. The vector quantization approach structures the input features by generating a given number of clusters and their centroids or representatives. The clusters are analyzed separately in the following by calculating their local principle components and using them for the supervised training of an artificial neural net classifier, of type local linear map (LLM). For details about the classification step, see [Heid 98]

For classifying the image at focus points the feature vector is generated from the window data, it is assigned to the appropriate cluster and the locally valid principal components are calculated. These features are the input of local linear map.

Fig. 3.9, on page 52, already presented an exemplary result showing classified focus points for the *baufix*[®] domain. A quantitative evaluation of the classification step that determines the part of correctly classified points from all the focus points for a test set is reported in [Heid 98]. For the detailed object label alphabet given above the classifications rate is 80.3 %. Ignoring exchanges between those classes that belong to one *baufix*[®] element, for example, *hole_bar* and *edge_bar* are counted both to be correct, if located at a bar, increases the classification rate to 86.7 %.

The detailed evaluation of the correct and false classifications on the test set delivers information about the exchange of classes. This information is summarized in an error

matrix and estimates, after normalization, the probability for the existence of a class n in the image, if the classifier gives class m . Thereby, the error matrix delivers confidence values for the classification results. The error matrix for the presented `baufix`[®] sub element classifier is given in the Appendix Sec. B.3.

3.2.4 Shape Based Recognition

Shape is an important visual feature for object recognition and categorization within the human perception system [Imai 94], [Lako 87]. Consequently, shape is used in many technical systems for object categorization and recognition [Supe 04], [Seba 04], [Belo 02], where the silhouette is characteristic for an object or object category. Many different shape features are developed and used for object recognition and image retrieval tasks. An overview can be found in [Zhan 04]. Shape features are either based solely on the outer boundary of the shape or they are calculated based on the shape region which describes the interior of the boundary. Fig. 3.13 shows two example shapes together with a possible boundary and region based description, respectively.

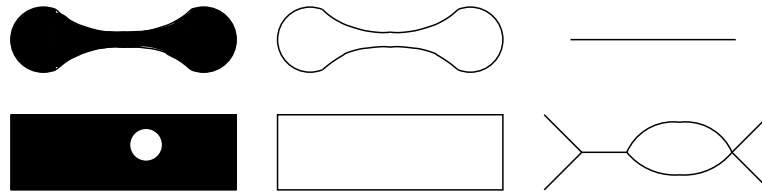


Figure 3.13: Two example shapes (left), described either by their boundary (middle) or by their medial axis based on the shape region (right). (A point on the medial axis is the midpoint of the greatest possible circle that fits into the shape.)

The boundary is represented in the simplest form as an unsorted cloud of points ([Gold 96], [Kend 99]), extended in [Belo 02] by defining the shape context as the relationship of a boundary point to all the other boundary points. Matching is done based on structural and local features of the shape boundary. Global information about the shape is taken into account, for example, in matching the whole set of boundary points to a query set or in calculating general features like perimeter, eccentricity or compactness. Global shape information is also represented in calculating 1-D functions based on the sorted sequence of all the boundary points $(x(t), y(t)), t = 0, 1, \dots, L - 1$. The functions are called shape signatures ([Zhan 02a]). Among others a shape signature may describe the distance of the boundary point to the centroid or the curvature of the boundary at a boundary point. Besides matching shape signatures in the spatial domain, often a Fourier transform is done and several Fourier coefficients constitute the feature vector for matching.

Region information generally affects global features like the area of the shape or different kinds of moments, e.g., geometric, Zernike, and Legendre moments [Teag 80]. The region information is also taken into account by representing the topology of the shape in the form of a skeleton-graph. Zhu and Yuille [Zhu 96] find the descriptor by

using two deformable primitives (worm and circle). Blum [Blum 67] defines the medial axis by the centers of the maximal disc that fits into the shape (Fig. 3.13, right).

Sebastian et al. [Seba 04] propose a graph representation based on a set of shocks that are derived from the propagation of boundaries by a grass fire algorithm [Sidd 99]. Matching of shapes then becomes a graph matching problem. In general, information about the shape region is especially important for recognition tasks, where the internal structure is more discriminant than the boundary information or in cases, where the boundary information is not reliably available. Local or structural information allows partial mapping especially needed in cases of strong occlusions or clutter. However, the descriptors are more complex in calculation and the relationships between different portions of the shape are often not captured in the match. Spatial domain shape matching methods are in general more sensitive to noise and boundary variations than methods that describe shapes in transform domains, like scale space or spectral space.

In [Zhan 01a], [Zhan 01b], and [Zhan 02a] comparative studies are reported applying different descriptors to databases with shapes resulting from affine distortion, rotation and scaling of basic shapes. The shapes are not occluded and have no inner structure. [Zhan 01a] and [Zhan 02a] show that within the descriptors based on Fourier transform of several shape signatures the central distance function leads to best results. In [Zhan 01b] the descriptor based on the Fourier transform of the centroid distance function is compared to an approach representing the boundary in scale space and two region based approaches. The Fourier descriptor was shown to be better or at least equivalent to the others. In [Zhan 02a] the centroid distance function was shown to be one of the most discriminative descriptors for shape. This results in just few of the Fourier descriptors, namely about ten, are needed to describe the shape well. The chosen Fourier coefficients are used as features for the classification step.

In the task here I attempt to categorize objects from an office domain based on their characteristic boundary. Objects with different inner structure arising from different coloring should be assigned to one class. Additionally, the algorithm should be robust against effects caused by scaling, rotation, distortions due to image noise, view point changes and slight occlusions. Following the considerations above, a student worker, C. Lange, and I decided to implement a boundary based system that provides features in the form of Fourier coefficients determined based on centroid distances.

In a digital image the boundary of a shape is described by the sequence of boundary coordinates. In order to compensate for scaling effects the sequence of L boundary points is sampled to a fixed number of N points. This sampling is done by taking equidistantly every L/N point from the chain of original points. Based on the N sampled boundary points $(x(t), y(t)), t = 0, 1, \dots, N - 1$ the centroid distance function $r(t)$ is calculated based on the centroid of the shape (x_c, y_c) :

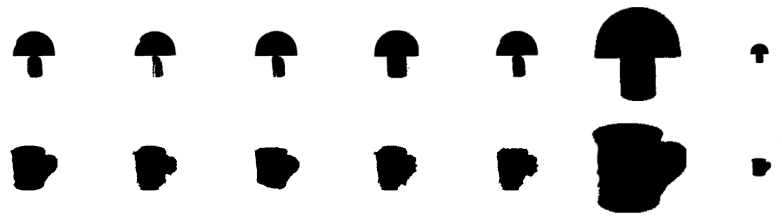
$$r(t) = \sqrt{(x(t) - x_c)^2 + (y(t) - y_c)^2} \quad \text{with} \quad x_c = \frac{1}{N} \sum_{t=0}^{N-1} x(t), \quad y_c = \frac{1}{N} \sum_{t=0}^{N-1} y(t)$$

The centroid distance is invariant to translation. The discrete Fourier transform of $r(t)$ is given by:

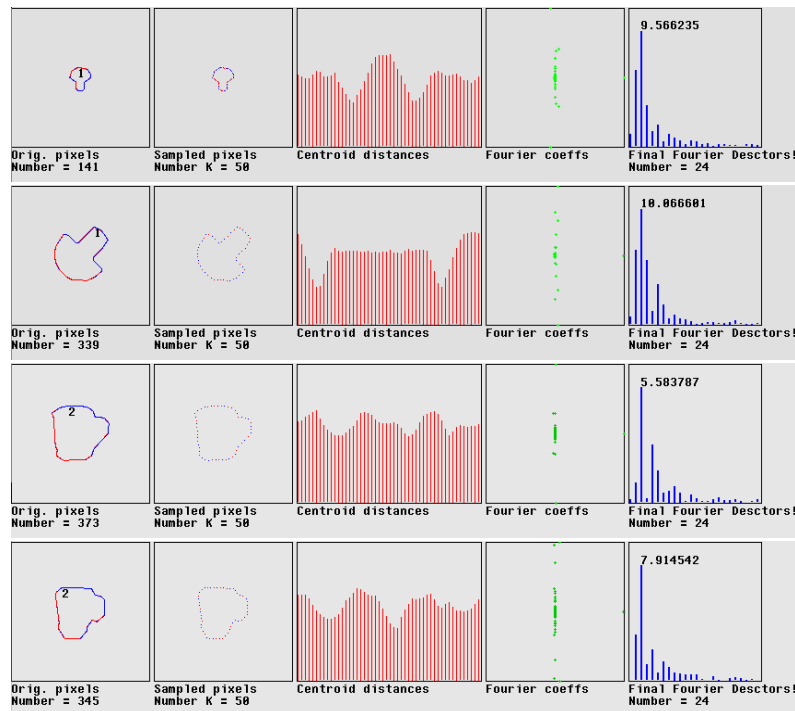
$$u_n = \frac{1}{N} \sum_{t=0}^{N-1} r(t) \exp \frac{-i2\pi nt}{N}, \quad n = 0, 1, \dots, N - 1$$

The coefficients $u_n, n = 0, 1, \dots, N - 1$ are called the Fourier descriptors of the shape. They serve as features for shape based recognition by classifying them using a support vector machine (SVM) ¹. For details of the classification approach, see [Bose 92].

The functionality of the shape based recognition module is tested by applying it to manually generated silhouettes of two different objects. The resulting silhouette is varied by rotation, changing its size and adding artificial noise, see Fig 3.14(a). Exemplary calculated Fourier descriptors are shown in Fig. 3.14(b).



(a) Exemplary artificial shape variations for two objects.



(b) Exemplary resulting Fourier descriptors.

Figure 3.14: Images and results for functional test of Fourier based shape recognition.

¹The software library libSVM used here, is distributed for free by Chih-Jen Lin, working at the Department of Computer Science at the National Taiwan University, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html>

The parameter setting used here is a fixed number of 50 equidistant sample points on the shape. The first 24 descriptors are used for classification. As expected, the representations for the two rotated mushrooms are very similar. The occurring differences are caused by the sampling of the shape. The two shapes representing the cup differ principally caused by the added noise. These differences has to be tolerated by the classification step. It turns out that the SVM classifier trained with just 24 images for each of the both classes is able to solve this two class problem reliably.

For the office domain addressed by the integrated system of Chapter 5 The system is exemplary applied for distinguishing six object classes of an office domain: cup, computer mouse, mouse pad, pen, hole punch, stapler and, additionally, the rejection class. Segments representing one object each are generated for training by applying the mean-shift segmentation algorithm, presented in Sec. 2.2.3, to images that contain the isolated objects and manually correcting over segmentations by merging. For the proof of concept just 93 objects are used for training which are too few for realizing a working shape classifier for seven classes (evaluations are given in Chapter 5). Nonetheless the module shows its principal applicability to this recognition task.

3.3 Additional Information: Context Based Systems

Besides the individual, visual data based recognition of objects applying either data or conceptually driven approaches the integration of additional high level knowledge delivers valuable information for the recognition step. Besides knowledge derived from other modalities like results of speech and gesture understanding processes, knowledge about the object context constitutes an important kind of additional information. Therefore, in the following firstly some basic concepts and approaches of generating context based information are presented, before afterwards available modules for exploiting context knowledge that are used in the integrated systems of Chapter 5 are discussed in more detail.

Generally, different kinds of context of an object can be distinguished. First, there is the scene as a whole, where an object is located in. Analyzing and classifying the scene delivers information about typical occurring object types and positions. Bose and Grimson [Bose 04] use scene specific context for identifying objects like cars and pedestrians in far field surveillance video sequences. Torralba [Torr 03] exploits statistical background features for scene classification. Typical object types and localizations are learned from scene features. Scene based predictions for object locations are used in [Murp 04] for controlling the application of data driven object recognition modules. The final evaluation for an object location is determined from probabilistically combining the evaluation of the context based expectation with the belief of the data driven recognition result.

Instead of taking into account global features of the scene Campbell et al. [Camp 97] use local neighborhoods for determining context based features. The results of pixel wise classifying the neighbored pixels of the current region constitute the base for determining local context based features. They are used together with data driven ones within a unique classification step for image regions. Fink and Perona [Fink 04b] attempt of parallel learning the detection of multiple objects within an image window in a

boosting framework. They train several classifiers exploiting local dependencies, where the relevant relations occur during the training phase.

Besides characteristics of the surrounding scene the context of an object may be represented by spatial relations between formerly determined semantic object units. Knowledge about object constellations thereby can be represented explicitly or implicitly by system parameters learned from training material. Hanson and Riseman propose in 1978 their system VISIONS [Hans 78], where related objects are explicitly described in the form of typical arrangements of objects in scenes: cars are driving on the street, sky is blue and is located above other objects like houses or grass. Explicitly rule based formulations of object neighborhood relations are used in [Feld 74] to control a simultaneous segmentation and recognition procedure.

Strat and Fischler [Stra 91] also define explicitly their context sets that contain rules for hypothesis generation and verification. Thereby the generation rules control specialized operators that run successfully if special assumptions are fulfilled which is ensured by the context set. A similar idea is realized in [Bobi 95] for static recognition and tracking. For example, a tracking algorithm is started, if an appropriate special object, like the head of a person, is detected in the image.

Probabilistic spatial context models are realized in [Sing 03] together with a set of individual material detection algorithms for realizing a 'context-aware material detection system'. The recognition of key semantic material types in images such as sky, grass, foliage, water, and snow is improved by constraining the beliefs to conform to the probabilistic spatial context models.

Instead of reducing the context of an object to its direct neighbors, Torralba et al. [Torr 05] model context between object classes by a dense and long range Conditional Random Field. Object classes are firstly detected by template based data driven classifiers. In a training step context relations between objects like monitor, mouse and keyboard that are represented as connections within the random field are learned. Carbonetto et al. [Carb 04] generate a statistical model based on a Markov Random Field that learns spatial context between blobs, which are feature sets calculated based on data driven generated segments or grid segments. The random field calculates a global optimization of isolated blob assignment and spatial context.

Besides exploiting spatial context occurring within one image that represents the scene at one point of time temporal context may support the analysis of the current scene. In [Kolo 04] the rules of a tennis match are used for generating a contextual model for event based sequence annotation. The event sequence that is represented by a Hidden Markov Model constitute the temporal context for an isolated event and is exploited to compensate detection errors for the isolated event.

Also symbolic information different from objects like detected actions and/or gestures and their typical coexistence with an object may constitute the context for object recognition. Moore et al. [Moor 99] realize a general class model for each object, where besides image templates and image data based statistical features also associated manipulative gestures like flipping for the book or picking up for the telephone are stored. Gestures are detected based on Hidden Markov Models and hand trajectories. Experiments are described, where action recognition results and the stored relation to objects are used for object recognition. Hanheide et al. [Hanh 04] also model the relations between objects and typical manipulative gestures. They use a common Bayesian net-

work for both kinds of symbolic information, whose structure mirrors the dependencies between object and action recognition. For example, a typing action occurs probably together with objects like monitor, mouse, and keyboard.

Object context delivers valuable information for supporting the object recognition process. However, this information should be integrated with data based information in order to avoid hallucinations. Mechanisms for the integration step are part of the description of the proposed general integration framework in Chapter 4. But before addressing this, the available recognition modules for exploiting different kinds of context knowledge that are applied for realizing the integrated systems presented in Chapter 5 are described in the following.

3.3.1 Semantic Region Growing

Object regions often are over segmented by data driven general purpose segmentation approaches, due to significant inner structures within the object surfaces. This is caused intentionally by, e.g., paintings or accidentally by shadows or highlights. For generating the object region as a whole appropriate neighbored segments have to be merged. The criterion for this region growing step is the semantic information that the segments get in the form of symbolic object labels from the object recognition modules. Consequently, candidates for merging are neighbored areas that carry the same object label. This strategy of semantic region growing delivers new hypotheses for object regions based on the object context.

3.3.2 Recognition based on Assemblage Rules

Within the *baufix*[®] scenario assemblies can be built from the set of elementary parts, like cubes, bars or bolts. For some examples, see Fig. 3.15. Generally, a huge amount of assemblies can be built that makes it impossible to model each individual assembly in advance, whether in the form of training images nor as symbolic construction plans. Therefore, the recognition of those assemblies is done in [Bauc 02] based on recognition results for the elementary parts as input and modeling the construction rules for assemblies within this domain.

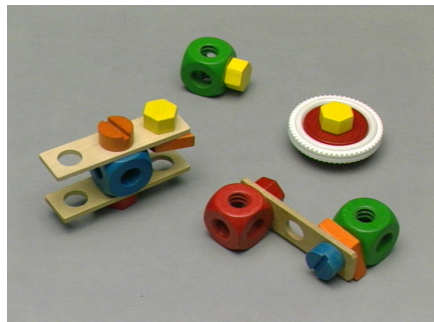


Figure 3.15: Example image for different *baufix*[®] assemblies.

Assemblies are defined to be a set of interconnected parts representing a stable unit, where attachments eliminate all degrees of freedom for relative motion. In the *baufix*[®] construction kit the elements are bolts of different length, some miscellaneous parts that can be put on the bolt threads but not fastened and finally nuts that are fastened to the bolts. The given definition of assemblies leads to the demand of having at least a bolt and a nut in every assembly and optionally one or more miscellaneous parts. The functional parts may be fulfilled by a sub assembly that contains itself at least a bolt and a cube. The construction rules of the domain are formalized as a context free grammar $G = (N, T, P, S)$. The variables of this grammar are assemblies and their functional parts:

$$N = \{ASSEMBLY, BOLT_PART, MISC_PART, NUT_PART\}.$$

The set of terminal symbols contains the *baufix*[®] elements:

$$T = \{\text{bolt, bar, ring, felly, socket, cube, rhomb_nut}\}.$$

The start symbol is $S = ASSEMBLY$ and the productions P are:

```

ASSEMBLY  →  BOLT_PART NUT_PART | BOLT_PART MISC_PART NUT_PART
BOLT_PART →  ASSEMBLY | bolt
NUT_PART  →  ASSEMBLY | cube | rhomb_nut
MISC_PART →  ASSEMBLY | felly | socket | ring | bar |
              ASSEMBLY MISC_PART |
              bar MISC_PART | ring MISC_PART |
              felly MISC_PART | socket MISC_PART
    
```

The rules of the context free grammar are transferred directly to the knowledge base of a semantic net realized within the semantic network formalism ERNEST. The variables of the grammar become concepts of the knowledge base and the production rules occur as part-of-links between the concepts. Furthermore, attributes are associated to the concepts that model context sensitive and domain specific aspects.

The analysis starts from the recognition results for the *baufix*[®] elements that are expected in the form of labeled image regions. A necessary but not sufficient precondition for an assembly is the existence of a cluster of neighbored labeled image regions. Within the cluster a region that is labeled as a bolt is instantiated for the BOLT_PART of an assembly. After marking the region as considered, the adjacent elementary objects are considered. They are tested for being miscellaneous or nut parts of the assembly. The process ends, if no longer valid constellations for assemblies are found.

Fig. 3.16, left, shows the recognition results for the elements for an example assembly. On the right, the resulting description for this assembly within the above described formalism is given. The assembly as a whole is built from the red bolt, an assembly that fulfills the MISC_PART role, and the red cube. The miscellaneous sub assembly contains another sub assembly that fulfills the role of the BOLT_PART and the green cube constitutes the NUT_PART. Finally, the BOLT_PART sub assembly is built from the blue bolt, the bar, and the rhomb_nut. Note, that the given description is not unique. There are other possibilities to describe the assembly within this formalism like there are different ways to build this assembly from the given set of elements. For the purpose

3 Object Recognition

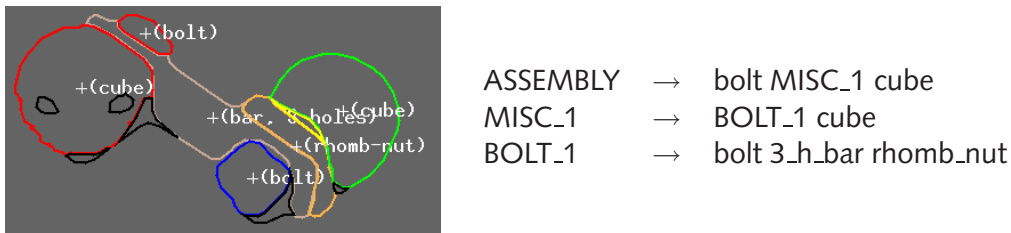


Figure 3.16: Exemplary recognition results for **baufix**[®] elements within an assembly and resulting assembly structure description.

of recognition dealt with here, one possible description is sufficient. Problems occurring from the ambiguity are discussed in [Bauc 02].

So far, I described how assemblies can be recognized from visual input data applying syntactical analysis. But, the other way around, information gathered during the syntactical analysis allows to make predictions for elementary objects that are not recognized by the data driven modules. Based on the construction rules and the contextual information about a partial assembly predictions about further essential components, can be made within a rather small image area. If there exists a suitable region that was not labeled yet a prediction for the label of this region is possible. Fig 3.17 shows an assembly where the element recognizer fails to label the upper blue cube due to its partly occlusion. During the syntactical analysis of the assembly the missing object label is predicted.

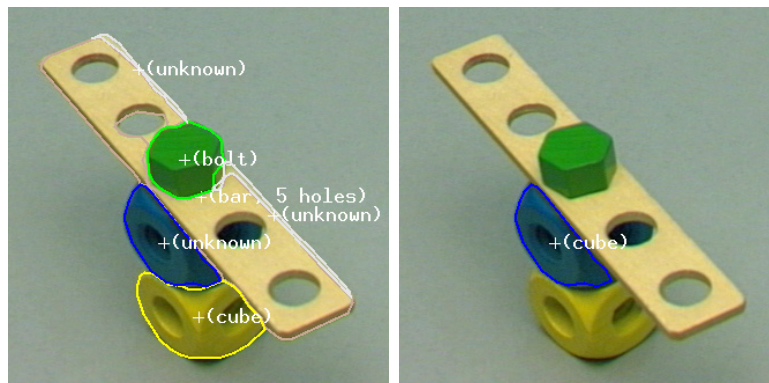


Figure 3.17: Example for the generation of a hypothesis about a **baufix**[®] element based on assemblage knowledge: The blue cube was not recognized data driven due to occlusions, but the hypothesis can be made from assemblage knowledge.

Besides missing elementary object labels ambiguous object labels may occur. If within a cluster of labeled object regions a region gets two labels, for example, a 'cube' and a 'felly' label, the syntactical assembly recognition process delivers the information that the cluster with the cube hypothesis leads to a valid assembly description but the cluster with the felly does not, as it is shown in Fig. 3.18. Assuming that there is an assembly

in the image, the recognition result is favored that leads to more elements participating in assemblies than others.

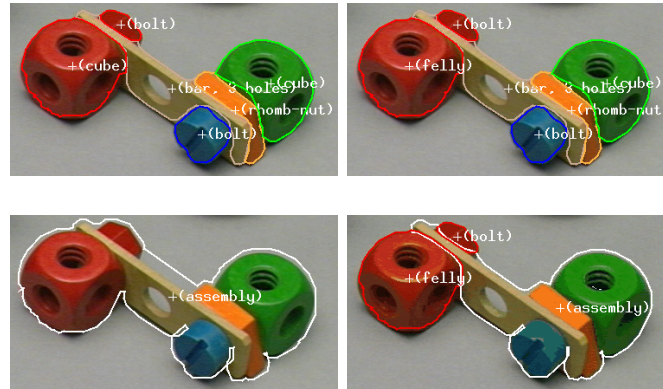


Figure 3.18: Two competing results of elementary recognition (top) lead to different results for assembly recognition (bottom). The competing results are evaluated based on the number of assembled elements.

The recognition module for **baufix**[®] assemblies relies on the one hand on valid recognition results for the elements but on the other hand supports the elementary recognition modules with contextual knowledge by delivering predictions for object labels and evaluations.

3.3.3 Monitoring the Assembly Construction Process

Within the approaches presented so far, visual information from static scenes is used for recognizing **baufix**[®] elements or assemblies. Recognition of the objects of a given static scene and monitoring the process leading to this scene, in this case the assembly construction process, complement one another [Sage 02]. Monitoring the process of constructing assemblies happens over a period of time and can support the recognition process with expectations based on the temporal context of the current scene. The process of building an assembly from a given set of elements can be described as a sequence of the actions picking a part from the table/scene, connecting two parts that are in the hands and placing the connected parts onto the table again.

The approach for detecting these actions, proposed in [Bauc 99], is based on information about changes within the set of objects in the scene and a model for the content of the two hands of the constructor. Each hand is either empty or contains an element, a partial assembly, for example, a bolt and a bar put on it, or a complete assembly with at least a bolt and a nut that are connected rigidly. Two hand models represent the content of the two hands of the constructor.

Possible states of a hand model:

Empty | BOLT | MISC | NUT | (BOLT MISC⁺ 2) | ASSEMBLY

²The plus operator indicates the possibility to put at least one MISC on a BOLT limited by its length.

3 Object Recognition

Information about symbolic changes, namely new and disappeared parts, is obtained by comparing object recognition results in every time step with the previous state of the scene. Actions are inferred following a set of rules based on the symbolic scene changes in connection to the states of the two hand models.

Pick X

Preconditions: $X \text{ disappeared} \wedge \text{hand: Empty}$

Effects: $\neg (X \text{ on_table}) \wedge \text{hand: X}$

Connect X Y

Preconditions: $\text{hand}_1^3: X \text{ (BOLT_PART)} \wedge$

$\text{hand}_2^3: Y \text{ (MISC_PART | NUT_PART)}$

Effects: $\text{hand}_1: XY \wedge \text{hand}_2: \text{Empty}$

Place X

Preconditions: $X \text{ new} \wedge \text{hand: X}$

Effects: $X \text{ on_table} \wedge \text{hand: Empty}$

When a scene change is detected, the preconditions for the different actions are checked and, if met, the action is inferred and the state of the two hand models is changed appropriately. The **Connect** action does not change the content of the scene and, therefore, its detection can not be triggered directly by a symbolic change. Instead, when the next scene change occurs the **Connect** is assumed to have happened to either construct an assembly before placing it or to provide a free hand to pick an object. If a part disappears while both hands contain parts so that the preconditions for a connection are met, a **Connect** action is inferred, resulting in one hand becoming empty. This empty hand and the disappeared part are valid preconditions for a **Pick** action. If a **Connect** action is not possible and both hands contain parts an error message is generated because the disappeared part can not be taken by the hands. If, in contrast, a new object appears in the scene and none of the hands contain a rigidly connected assembly, but the precondition for a **Connect** action are met, it is assumed that the connection was done before placing the complete assembly.

As long as the action detection relies solely on the information about new and disappeared parts the constructor must obey some restrictions in order to avoid erroneous action detection:

- No parts are put down outside the visible scene.
- New elements may only be put into the scene if no similar elements (same type and color) are in the hands.
- Another constraint for a human constructor which is implicitly true for a robotic manipulator is that each hand can hold only one element or (partial) assembly.

Additionally, the action detection module relies on valid recognition results for the elementary objects. If an object label is false, the element can not fulfill the functional

³The notation hand_1 and hand_2 is only used to indicate different hand states, each of the two hand states can be hand_1 or hand_2 .

role within an assembly and therefore actions may not be detected correctly because the predictions for applying the rules are not given. If element detection is not stable in time, namely an object label disappears from one image to another caused by a recognition failure, this scene change may misleadingly be detected as a **Pick** action. These restrictions can be relaxed, if another camera is available that observes the hands of a human constructor [Frit 03]. In this case actions are detected based on visual information about the hand trajectory in connection to symbolic information about the relevant objects.

Based on reliable recognition results for the *baufix*[®] elements the action detection module collects information about elements within assemblies from the temporal context or construction process. At the end of a construction process, when the assembly is replaced into the scene the action detection module has collected information about the contained elements that serve as predictions for the element recognition [Brau 01].

Fig. 3.19 shows some steps of building an assembly and replacing it into the scene. Parts of the scene are shown together with detected scene changes, derived actions, and the states of the hand models. For the first three steps objects disappear from the scene and are consequently assumed to be content of one of the hands. In the last step new regions appear in the scene and the hand models are not empty. Then, a *Place* action is assumed and expectations for the appearing elements are generated based on the state of the hand models. These expectations are not located within the image but may support the data driven element recognition process. The *Place* is accomplished by updating the content of the hand models appropriately not before the element recognition process delivers its results and thereby confirm the assumption formerly based on region information.

3.4 Summary

The procedure of assigning symbolic object knowledge to visual sensory data is generally called object recognition. Object symbols are used as a basis for higher level processing like for example scene analysis and human computer communication.

The variety of object recognition tasks is enormous and a manifold of different approaches are developed. Systems can be distinguished by their (main) processing direction from the data to the model (bottom up) or vice versa (top down). Further, the kind of features and models determine, whether a system is designed for recognizing object exemplars or the members of object classes. And finally, systems differ in their strategy of proceeding a holistic recognition of objects or dividing objects into characteristic parts. For the selection of promising approaches one has to take into account the detailed requirements for the recognition task, available a priori knowledge about objects and their locations, lighting conditions, occurring occlusions, and many other aspects.

For complex recognition tasks the combination of different characteristics of an object will be generally necessary for distinguishing it from others. This can be realized by either using appropriate high dimensional features and complex processing strategies within one recognition module or by externally comparing the object symbols delivered from specialized recognizers. Different integration strategies are proposed in the literature where the probabilistic sum rule that takes into account all the available object

3 Object Recognition

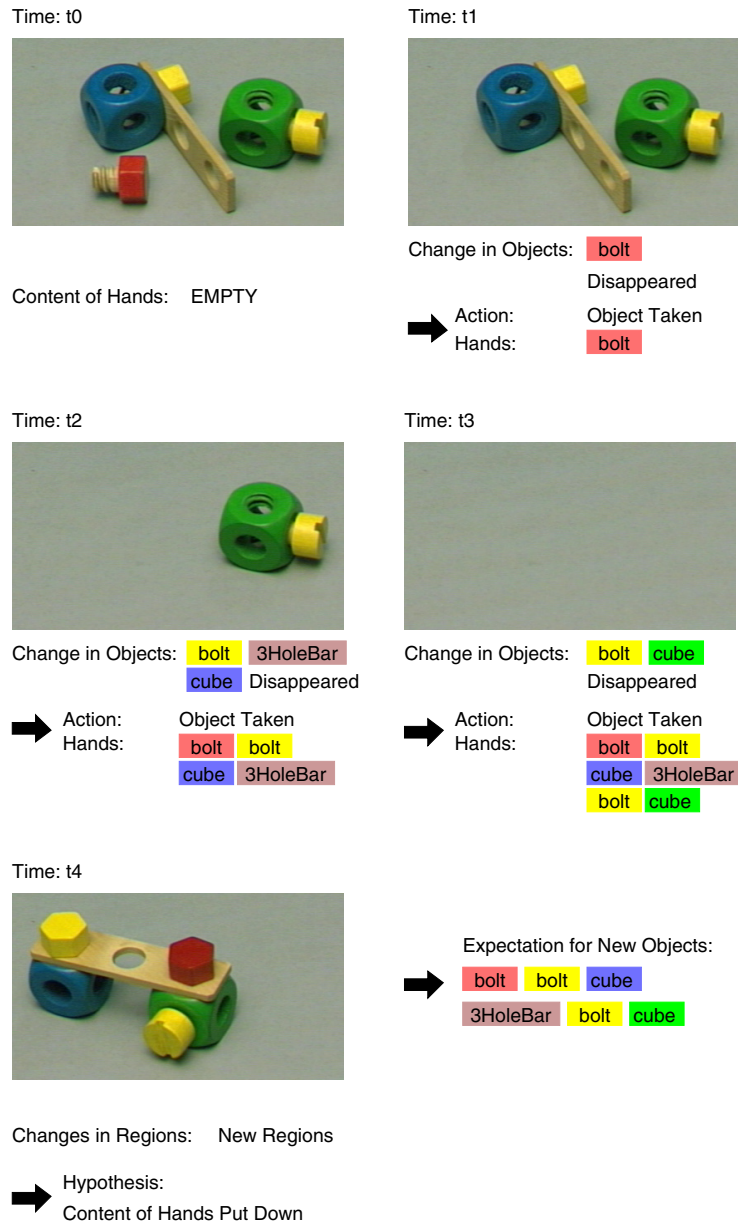


Figure 3.19: Image sequence showing the scene and detected changes as well as contents of the hand model, while building an assembly. After the assembly is placed into the scene again the expectation for the contained elements is generated from hand contents based on process knowledge.

symbols and their beliefs turns out to be the most general and promising. The external combination of several recognition modules leads to systems that are constituted from several comparably simple modules. The system is flexible according to module exchanges.

The presented recognition modules are available for the two exemplary tasks of recognizing *baufix*[®] elements and office objects that are the fields of application of the integrated systems evaluated in Chapter 5. The modules are differently specialized for the task and there is an approach relying directly on the image data for detecting object parts and some relying on preprocessed segment information for recognizing the whole objects. The differences between the approaches makes a successful external integration of their results probable.

For segmenting and recognizing an object additional information delivered based on preliminary hypotheses constitutes valuable information for complementing data driven results, clarifying ambiguities and evaluating competing one. A prominent kind of this additional information is knowledge about the spatial or temporal context of individual objects. A simple and task independent applicable module of that kind is the procedure of assuming two neighbored regions with same object label to descend from one object instead of two. For the *baufix*[®] task there are two further context based modules. The first one compares the spatial neighborhood of objects with the domain specific assemblage rules and the second one generates expectations for objects to be recognized from the history or temporal context of the present scene. Context based information has to be integrated with the sources of individual object information in order to find the recognition result that is mostly supported from the participating modules.

The following chapter introduces a general integrating framework that realizes a strategy for integrating individually generated segment and object information with additional high level knowledge.

3 *Object Recognition*

4 The Integrating Framework

The preceding two chapters show the great variety of existing modules for image segmentation and object recognition. They reach from data driven image segmentation and segment labeling approaches to conceptually driven strategies for segment and label determination based on the fulfillment of prior expectations. Different aspects and strategies have to be integrated in order to generate successful systems. Following the idea of explicitly combining simple modules, instead of generating monolithic complex systems, I propose a general framework, that allows the integration of several kinds of object related information resulting from independent modules. The framework is open for adding new modules and removing others and it is independent from the given recognition task. Instead of either relying on segments for object recognition or on object models for segmentation all sources of information are integrated equivalently, in order to solve the joint problem.

Already Feldman and Yakimovsky [Feld 74] propose the 'semantics-based region analyzer', which iteratively applies a segment based object recognition and an object based image segmentation process. Object labels are assigned based on segments by exploiting image features, like color, and context knowledge about the objects, like favored neighborhoods. Candidate segments for region growing are identified in detecting neighbored segments of same object labels and weak boundaries in the sense of image features. Object labels are assigned again to the new segments. Many decisions within the system are taken based on rules that require detailed knowledge about the task and much manual adaptation work when changing the task.

The joint problem of segmentation and part based object recognition is also addressed in [Yu 02], [Yu 03]. An affinity graph is built up with one node for each pixel and the edges representing the affinity between pixels determined based on image data driven edge features. An object recognition algorithm independently detects object patches, their partial segmentation and information about the spatial configuration of patches constituting the whole object. A patch graph represents the object patches as nodes and their relation within its edges. These two graph based representations are coupled and the global partition is calculated based on a cost function that integrates pixel level saliency and patch level consistency. The approach successfully segments and labels persons [Yu 02] and other objects [Yu 03] from cluttered background. Graph matching and cost functions are specialized for the concrete modules applied here. The system is not open for integrating further components.

Addressing the joint segmentation and recognition problem and being open for additional and related information leads to the strategy of independently generating a common representation for the available information and analyzing it. The formalism defines as few requirements as possible for the individual modules, in order to make the integration of existing modules possible. The representation is open for integrating additional information without starting all calculations from the beginning. At any point

of time, where the representation is internally consistent, after completely integrating one piece of information, an analysis process can generate object segmentation and recognition hypotheses by identifying those that are mostly supported by the integrated information delivered from the individual modules.

Before discussing the integration module in more detail, first its interaction with the other components of the whole integrated system are presented in the following.

4.1 Integrated System Architecture and Component Interaction

The idea of integrating information from several independent data based and high level knowledge based modules requires a flexible architecture for the integrated recognition system. Fig. 4.1 shows an overview of the system components.

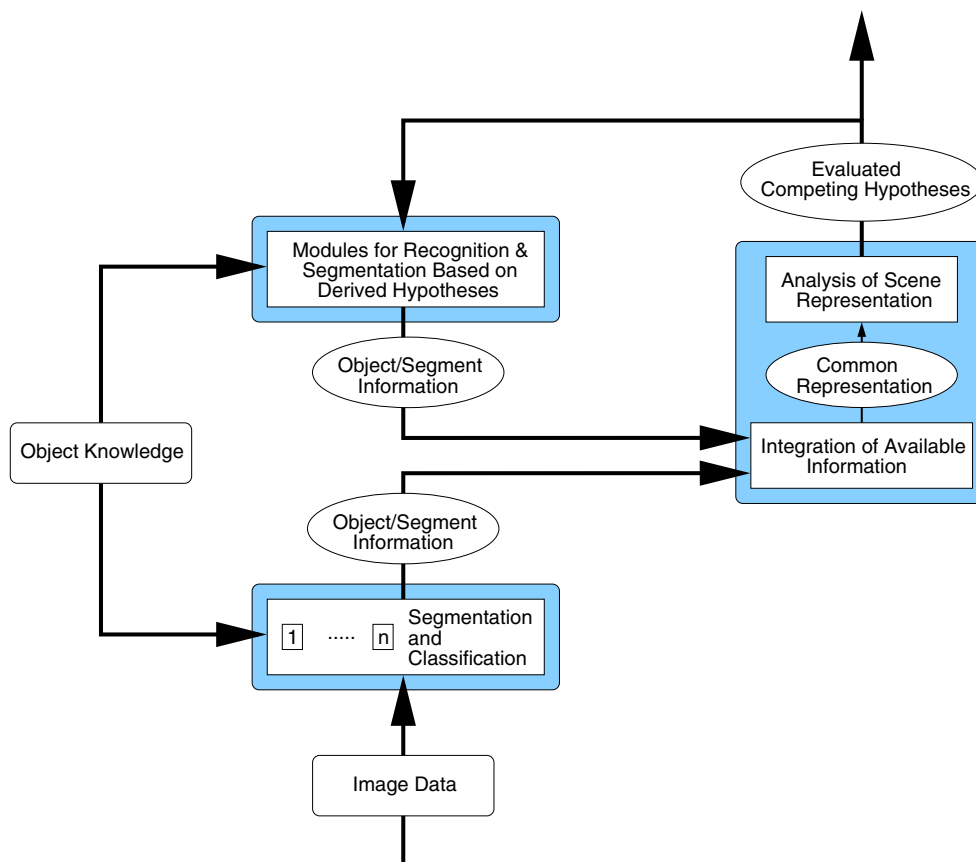


Figure 4.1: General system architecture for integration of several sources of information for object segmentation and recognition.

Image data is processed in several segmentation and classification modules that exploit more or less specialized object knowledge. This data based information is integrated within a common representation that is analyzed subsequently to obtain object hypotheses. They serve as input for modules that deliver additional information based

on the input hypotheses by exploiting higher level knowledge, like object context models. The additional information that is constituted from object hypotheses and/or confidence measurements is integrated with the already represented information and the analysis is restarted for identifying consistent hypotheses for object segmentation and labeling.

The system consists of several individual segmentation and recognition modules and the integration unit. Existing modules should be reusable and exchanging single modules within the integrated system should be possible without great efforts. This leads to the implementation of individual processes for the system components and the necessity for a flexible communication network that synchronizes and transfers data between the processes. In order to accelerate the overall calculation time the communication framework should be able to connect processes running on one machine as well as those distributed over several machines.

These requirements are fulfilled by the Distributed Applications Communication System (DACS) [Fink 95], [Jung 98] that was developed within the SFB360 at Bielefeld University and, therefore, is available here. DACS provides inter process communication facilities by establishing data streams between processes and by supporting remote procedure calls.

The data streams allow user defined data packages, like segmentation results for an image or the image itself, to be transferred from one process to another. The offering process creates the stream by denominating it with a system wide unified name and describing the data format. Other processes that know about the data format and the name order the stream and wait until a data package is available, before starting their work on this data. The demand streams, as realized in DACS, are a suitable mechanism for building up cascades of processes, where one process works on the results of the previous ones. The second communication mechanism that DACS provides are remote procedure calls. This mechanism simplifies the definition of a function, including input and output parameters, to be called from external processes. Within the integration scheme described here, both methods of communication are used. The independent segmentation and classification processes offer their results on demand streams that are read from the integration module. The higher level processes that exploit preliminary hypotheses are called using the remote procedure call mechanism. Intermediate and final results of the system are available via a demand stream for further use.

The realization of the integration modules is the topic of the remaining part of this chapter. The description starts with the generation of the common representation for segmentation and object information and its analysis. The incorporation of additional information that is based on intermediate results is described after that, before finally the evaluation scheme for determining confidence values for the recognition results is presented.

4.2 Common Representation of Segment and Object Information

As the backing of the integration system serves a common representation for the results of different segmentation and classification processes. In order to do object segmentation and recognition the recognition system has to identify the region within the image,

that is covered by an object and the associated object class label.

The common ground for both the segment information and the object information is the image. Therefore, the main idea for generating a common representation is representing the image in the form of related segments and assigning the object information to the segments, where the segments influence the assignment of object labels and the object labels influence the choice of segments to be candidates for object regions. For motivating the generation strategies for the segment representation, in the following first, some general effects occurring in data driven segmentation results are described.

4.2.1 Exemplary Effects in Data Driven Segmentation

The ideal result of an image segmentation process from the point of view of object recognition is a set of segments, where each segment represents one object region. But in reality, object regions are split into several segments, on the one hand, or adjacent objects are covered by just one segment, on the other hand. The splitting of the object region, over segmentation, often is caused by shading or highlights, but also by the inner structure of the objects surface. Merging more than one object region, under segmentation, is mostly caused by low contrasts at the object boundaries. In Fig. 4.2 some objects from an office environment and in Fig. 4.3 an example for a *baufix*[®] assembly are shown together with some segmentation results that illustrate the effects mentioned above.

Fig. 4.2 illustrates the effects of shading and inner structure of the object surface and the problems with low contrast at the borders. The segments are delivered by three region based segmentation algorithms, namely the approaches based on local color variation (Fig. 4.2(b), see Sec. 2.2.2), the mean-shift approach applied to color features (Fig. 4.2(c), see Sec. 2.2.3), and the hierarchical region growing approach color structure code (Fig. 4.2(d), see Sec. 2.2.1), and the forth contour based perceptual grouping approach (Fig. 4.2(e), see Sec. 2.2.5). All the segmentation approaches work image data driven based on different general purpose homogeneity criteria and do not exploit object knowledge.

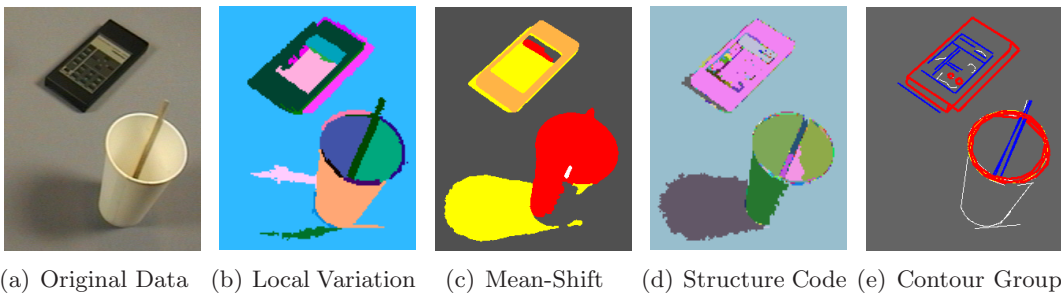


Figure 4.2: Two objects from an office domain with different segmentation results.

The surface of the black remote control has an inner structure caused by the array of buttons. This structure is handled different by the individual approaches. Either it is represented as a whole in form of a separately extended area within the object

surface (local variation, Fig. 4.2(b) and mean-shift approach, Fig. 4.2(c)), or many small segments arise within the object region, as for the color structure code, Fig. 4.2(d). The contour based perceptual grouping, Fig. 4.2(e), delivers some parallel but no closeness groups for the button structure. All algorithms identify the silhouette or object region of the remote control. Different from this, the silhouette of the mug is available only from one approach, namely the local variation criterion, Fig. 4.2(b). The others have problems with the low contrast and merge the cup with the background. The inhomogeneity given by the shading of the front of the mug results in separate segments for three approaches, while the mean-shift based segmenter, Fig. 4.2(c), subsumes the parts of the mug and even the stick to one segment.

Fig. 4.3 shows results for a *baufix*[®] assembly. This example illustrates mainly effects of shading and highlighting. The results are delivered by three region based approaches, the pixel based classification (Fig. 4.3(b), see Sec. 2.2.4), the mean-shift algorithm applied to color segmentation (Fig. 4.3(c), see Sec. 2.2.3), and the color structure code (Fig. 4.3(d), see Sec. 2.2.1), and the forth contour based perceptual grouping approach (Fig. 4.3(e), see Sec. 2.2.5).

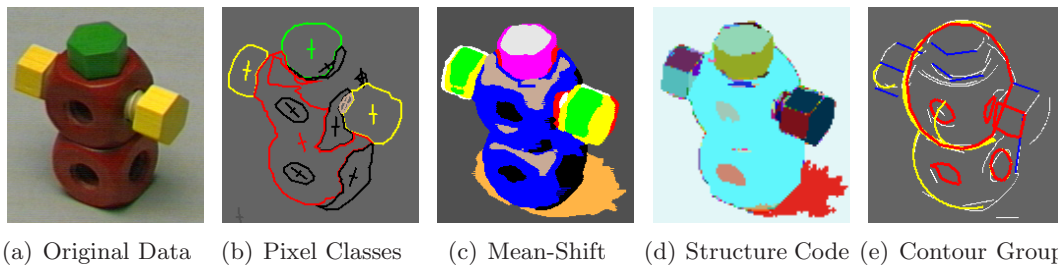


Figure 4.3: Example for a *baufix*[®] assembly and different segmentation results.

The approach based on pixel wise classification, Fig. 4.3(b), uses knowledge about the objects or, to be more precisely, about the object colors occurring within the domain. It is able to deliver good results for the three bolts, apart from the one small shadow region of the green bolt. Due to the classification strategy of the algorithm, it is not able to distinguish the two red cubes, but the correct result of this algorithm would cover the two cubes within one segment. Instead, at the right side two individual black shadow regions arise. The upper right part of the cubes is merged to the background. Shading and highlighting reduce the color saturation and lead the algorithm to classify the colors to the grey background. The other region based and edge based segmentation processes do not use domain specific knowledge. The mean-shift approach, Fig. 4.3(c), typically results in two distinguished tones for each *baufix*[®] color and the background, where one represents the bright and the other the shaded tone. For the green bolt at the top, for example, the algorithm identifies the bright upper side (pseudo colored in bright grey) and the darker surrounding (pseudo colored in pink). This results in at least two, mostly more regions per object. The two red cubes are mainly covered by one segment, because of low contrast and just small shadow at the boundary. There are many more segments than objects, but apart from the borderline between the two equally colored

cubes the object boundaries correspond to segment boundaries. This implies that the object regions are principally reconstructible from the segments by generating the union of several suitable ones. The color structure code algorithm is based mainly on local information following region growing ideas, Fig. 4.3(d). Although the color distance threshold is selected to be so small that the bolts get two or more segments caused by the contrast between the highlighted and the shaded area, it is not possible to distinguish the two cubes. Even the highlights and shadows on the two cubes are covered by the resulting one segment for the two cubes. Finally, the contour based grouping, Fig. 4.3(e), is able to detect the border between the two cubes, but is not able to find appropriate closeness groups for the two cubes. The lower cube is left out, the upper cube is merged with a bolt.

Fig. 4.4 shows another example of a *baufix*[®] assembly together with segmentation results that is especially interesting due to the result for the bar delivered by the contour based perceptual grouping process (Fig. 4.4(d), see Sec. 2.2.5). For completion the results of the segmentation based on the pixel based classification (Fig. 4.4(b), see Sec. 2.2.4) and the mean-shift approach (Fig. 4.4(c), see Sec. 2.2.3) are depicted, too.

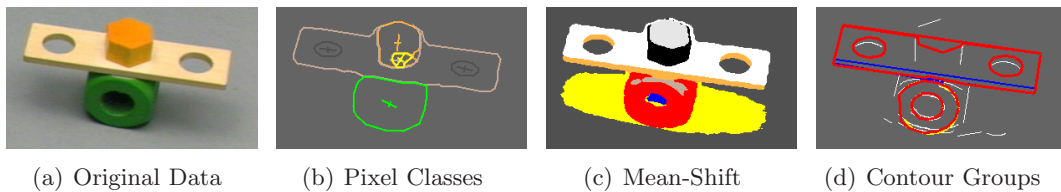


Figure 4.4: Example for a *baufix*[®] assembly with results from three different segmentation algorithms, especially interesting due to the contour based grouping hypothesis for the bar.

The two region based segmentation processes both are able to detect the object boundaries. The pixel wise classification, Fig. 4.4(b), delivers well formed segments for the cube and bar, just a yellow colored highlight is separated from the orange bolt. The mean-shift approach, Fig. 4.4(c), delivers several regions for each object due to highlights and shadows, as already discussed for the previous example. The most remarkable result here delivers the grouping process, Fig. 4.4(d), with the original rectangular segment for the bar. In contrast to the other segmentation approaches, the perceptual grouping is able to bridge over the gap within the upper contour of the bar resulting from the occlusion by the bolt.

The effects of highlights, shadows, low contrast etc., illustrated by the examples, are well known in the area of image segmentation. Integration of several, independent approaches often allows to detect and even to solve those problematic cases. This is done based on the main idea that disagreements for the borders of the resulting segments indicate uncertainties. In order to integrate the segments information a common hierarchical representation is built up that establishes special relations between segments that cover the same image area.

4.2.2 Generating a Hierarchical Representation of Segmentation Results

Segments from independent segmentation approaches have different characteristics that complement one another. For exploiting the different sources of segment information a common representation is generated that differently relates those segments that cover the same image area and those that do not.

Boundary Representation for Segments Resulting in Areas

The segmentation approaches originally assign each pixel to one segment and, thereby, deliver disjoint segments that cover the whole image plane. For embedded segments the area for the inner segment is explicitly excluded from the outer one. Nevertheless, both segments are related because they describe the same image area, as, for example the bar and its holes in Fig. 4.4.

For determining the image area that a segment covers the outer boundary of the segment is decisive. To get an even more compact representation, the boundary is approximated by support points that are considered to be connected by lines. The number and position of support points is controlled by an error threshold, that defines the maximum difference between the approximation and the original chain of boundary pixels. The outer segment border finally is described by a polygon consisting of linear connected support points. Considering areas derived from segments by just taking into account their outer boundary, the relation between areas is characterized by their intersection.

Considering the disjoint set of segments delivered from one segmentation module, a pair of derived areas either provide no intersection, called they are *independent* in the following, or one area constitutes the intersection area, if it is completely *contained* within the other. Fig. 4.5 shows the set of segments resulting from the mean-shift approach, as shown in Fig. 4.4(c), ordered by descending pixel counts and the boundary based representation of differently related areas. Two independent areas are connected by a white edge and one area that contains another is connected to it by a black edge.

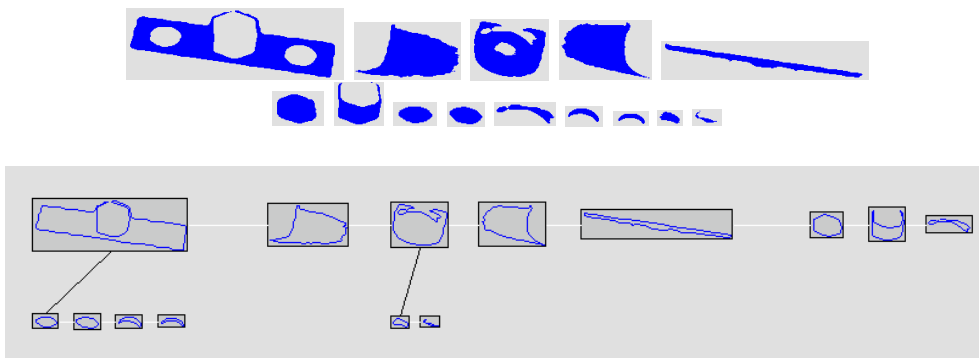


Figure 4.5: Upper part: Segments resulting from the mean-shift approach, see Fig. 4.4(c), ordered by decreasing size. Lower part: Relational representation of the areas descended from the mean-shift segments. White: areas are independent, Black: one area contains another.

Determining the relations between all pairs of areas results in a fully connected graph. For avoiding redundant information just necessary relations are explicitly represented, while others are avoided.

Classifying Relations between Areas from Different Processes

The situation concerning the occurring area intersection changes significantly, if the segments delivered from more than one segmentation approach are taken into account. Fig. 4.6 shows the three different segmentation results, already presented in Fig. 4.4, depicted by the outer boundaries of the segments in red, blue, and green, respectively.

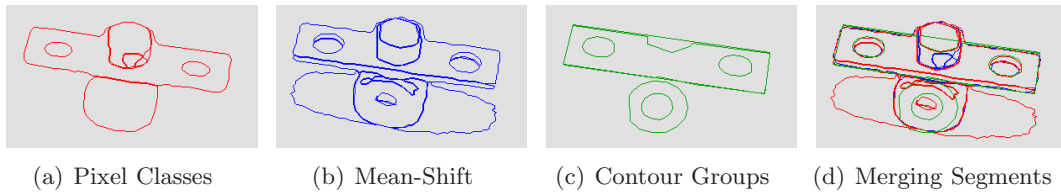


Figure 4.6: Segments of Fig. 4.4 shown by their boundaries.

By merging all available segment information, as shown in Fig. 4.6(d), it becomes obvious that for the bolt only the summary of the four segments delivered from the two region based approaches deliver a good object region, while the contour based grouping provides no hypothesis at all.

The grouping results do, generally, not constitute a disjoint set of segments, as shown in Fig. 4.6(c), that provides two rather similar closeness groups representing the bar, see also 2.2.5. The groups are delivered as boundary descriptions. Relating groups by evaluating the overlapping area works, although more general cases of intersection has to be expected.

Areas derived from segments that originate from different segmentation approaches also provides more variety in their intersections. However, all segmentation processes work on the same image data and, therefore, the resulting segments are not arbitrarily distributed and not arbitrarily overlapping. There are common boundaries, if these boundaries are meaningful and there will be disagreements for boundaries due to the different characteristics of the segmentation processes including their different handling of shadows, highlights, bad contrast etc. Based on this fact, the relation between a pair of overlapping areas is not just represented by the continuous value for the intersection area, but the relation is classified into semantically meaningful classes based on the degree of intersection, as already done above for the two classes of *independent* and *contained* areas.

Besides them, there are additionally areas that are *similar* to each other, if independent segmentation processes almost agree on the boundary of an area or similar closeness groups are identified. The general case of *partialoverlap* occurs, where there is a remarkable intersection, but this intersection does not cover at least one of both areas. This case suggests disagreements and, therefore, uncertainty for a (part of a) boundary.

For classifying the relation between two areas, the ratio between the intersection area and the original areas is calculated, respectively. A certain degree of deviation from the ideal case for this ratio has to be tolerated that is caused by slight differences in locating boundaries between the different segmentation approaches in the presence of real and, therefore, noisy data. The tolerated deviation is steered by a threshold, leading to the definitions for area relation given in Tab. 4.1.

$similar(a_i, a_j) \iff \frac{a_i \cap a_j}{a_i} > t \wedge \frac{a_i \cap a_j}{a_j} > t$
$contains(a_i, a_j) \iff \overline{similar}(a_i, a_j) \wedge a_i > a_j \wedge \frac{a_i \cap a_j}{a_j} > t$
$partialOverlap(a_i, a_j) \iff \overline{similar}(a_i, a_j) \wedge \overline{contains}(a_i, a_j) \wedge \overline{contains}(a_j, a_i) \wedge ((a_i > a_j \wedge \frac{a_i \cap a_j}{a_j} > 1 - t) \vee (a_i \leq a_2 \wedge \frac{a_i \cap a_j}{a_i} > 1 - t))$
$independent(a_i, a_j) \iff \overline{similar}(a_i, a_j) \wedge \overline{contains}(a_i, a_j) \wedge \overline{contains}(a_j, a_i) \wedge \overline{partialOverlap}(a_i, a_j)$

Table 4.1: Definition of relation classes between two areas a_i and a_j according to their intersection with fixed threshold $t \in (0.5, 1)$

Given one pair of areas the definition for the type of relation is unique. As mentioned above, calculating relations between all possible pairs of areas would lead to a fully connected graph of related areas. For avoiding redundant relations and further structuring this representation, I exploit the different characteristics of the relations.

As already shown in Fig. 4.5, for a group of independent areas not all relations are explicitly established, but instead each area is connected to not more than two areas, resulting in a list of independent areas. This is sufficient under the precondition that other relations would be explicitly represented and, therefore, the avoided relations between the list members are defined to be of type independent. Furthermore, several areas that are contained within a superior one are not individually related to this. Instead, the group of contained areas is related to each other, for example, as being independent, and just one element of this group is related explicitly to the superior area. Establishing additionally relations to all the other contained areas constitutes redundant information and is, therefore, avoided.

Additionally, the relation between contained areas is transitive, because it is:

$$\text{contains}(a_i, a_j) \wedge \text{contains}(a_j, a_k) \rightarrow \text{contains}(a_i, a_k)$$

This allows in the case of a region a_i containing a_j and a_k and a_j containing a_k to avoid the explicit representation of the relation between a_i and a_k . Doing so, the representation becomes hierarchical and expresses in general several layers of contained areas. The relations are represented explicitly just for neighbored layers.

If two areas are similar, their relations to other areas are comparable. Therefore, a set of similar regions is handled as a union concerning the representation of relations to other areas. Relations are represented explicitly just for one representative of the set of similar areas.

For partially overlapping areas, the part of the boundary that is controversial is located at the intersection of both areas. The controversy can not be solved here, therefore, the two areas are handled as a strongly related union, by generating the union of the two areas as a new artificial area, as shown for the bolt in Fig. 4.7

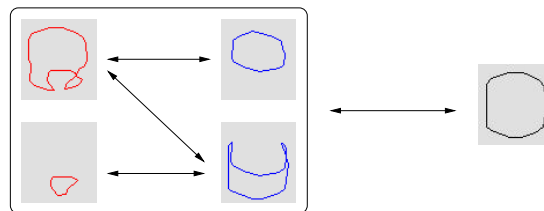


Figure 4.7: Relations between segments of two segmentation processes that cover the same image area. All original segments area related to the unified area built from the originals.

Further calculations for relations to other regions are carried out based on this union area. As an exception from this, partially overlapping areas resulting from perceptual grouping are handled. The perceptual grouping mechanism delivers boundary information based on the Gestalt laws, where possibly no edges are present in the image, see Sec. 2.2.5 for details. Therefore, partial overlaps with those areas are not necessarily caused by uncertainty but occur also due to occlusions, as for the bolt occluding the upper boundary of the bar in Fig. 4.4 and Fig. 4.6. If grouping hypotheses partially overlap other areas, no artificial union area is generated, in order to avoid merging several object regions into one area. Nonetheless, such a closeness group constitutes an alternative to the representation given by one or more region based segments that is taken into account later on for hypothesizing object regions. Due to their special characteristic the partially overlapping groups remain firstly independent until the construction of the hierarchical representation is finished. Then, a group is related to an appropriate represented region based generated area, if the matching procedure is successful that has been already proposed in [Schl 00] for supporting object recognition by grouping results, see Sec. 2.2.5. Partially overlapping grouping hypotheses that does not match to any region based generated area are not incorporated within the common representation and remain within a separate list for further use.

Fig. 4.8 shows the hierarchical representation of the segmentation results shown in Fig. 4.6.

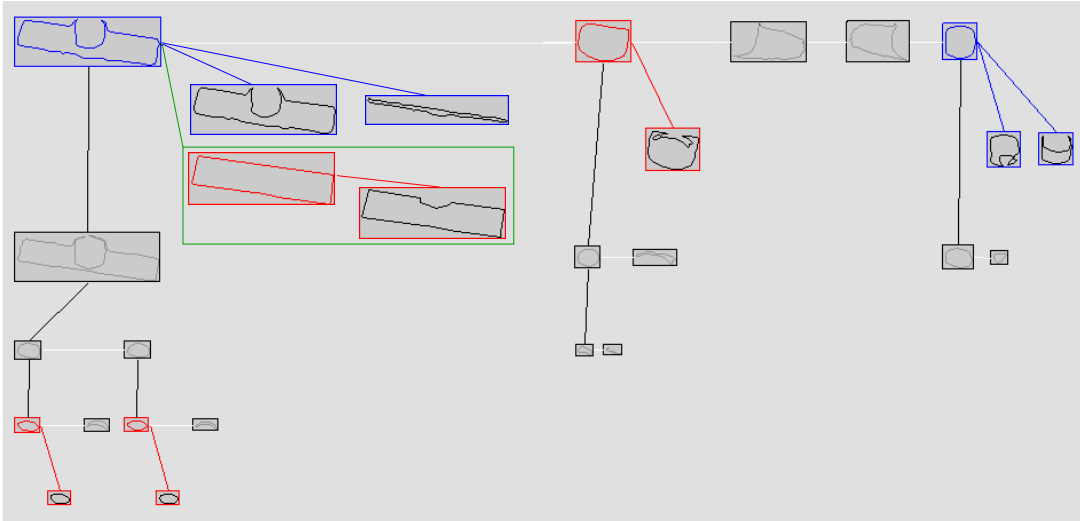


Figure 4.8: Hierarchy of segmentation results including the segments from the three segmentation processes shown in Fig. 4.6, related according to the definitions given in Tab. 4.1 with threshold t set to 0.85. Red: Similar areas. Blue: Partially overlapping areas and artificially generated union representative. Green: Partially overlapping grouping areas. Black: Area containing another area. White: Areas are independent.

The representation is mainly characterized by the independent areas of the upper level (white edges) most of them constituting the root of a tree of contained areas (black edges). The relation between similar areas (red) simplifies the representation by finally reducing the number of areas to be considered. The two occurring union areas of partially overlapping ones (blue) lead to better object region representations for the bolt and just slightly changes for the bar in comparison to the original segments. Partially overlapping grouping hypotheses are explicitly related (green) but no artificial union area is generated. This avoid the merging of the area representing the bar with those representing the bolt. The closeness group representing the front of the cube is left out for this representation because it is not matched to one of the region based generated segments.

The hierarchical representation is based on the classification of the pairwise relations between areas which is controlled by the threshold parameter t . The hierarchy shown in Fig. 4.8 is calculated by setting t to 0.85. In comparison, Fig. 4.9 shows the representation resulting from changing the threshold from 0.85 to 0.83 for this example. A difference occurs for the bar, where no longer partial overlaps are detected and instead, the area derived from the greatest original segment occurs in the upper layer. The less strict formulation of similarity leads to the classification of the two main bar areas to be similar instead of one is contained in the other. A comparable situation occurs for the small long bar area that is classified to be contained now, while before it was partially overlapping. The two parameter settings take influence on the representation, but it

is hardly to decide, which representation is better. The detailed discussion of this parameter concerning its numerical setting and its influence on the results is part of the evaluation of the realized integrating systems given in Chapter 5.

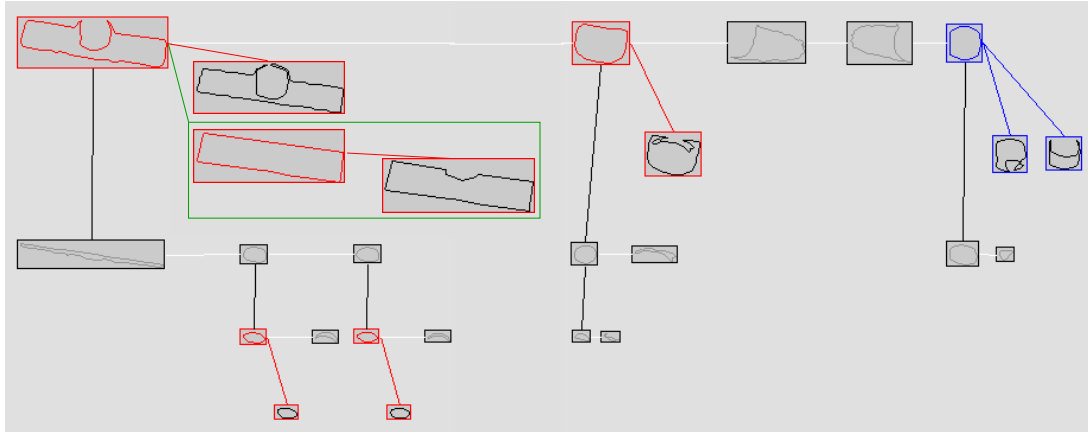


Figure 4.9: Hierarchy of segmentation results similar to the one presented in Fig. 4.8, but with threshold parameter t set to 0.83.

Algorithmic Aspects of Generating the Hierarchical Representation

The simplest way for building up the hierarchical representation is considering all possible pairs of areas, calculating their pairwise intersection, and determining their relation by classifying the intersection. This results for N areas in calculating $\frac{N \cdot (N-1)}{2}$ intersections which means a quadratic increase with the number of areas. For example, twenty original areas have to be taken into account from two segmentation processes, shown in Fig. 4.6(a) and Fig. 4.6(b), resulting in 190 possible pairs of areas to be compared. Due to the fact that one new artificial area is introduced by two partially overlapping areas, see Fig. 4.9, the overall number of comparisons increases even by 18 resulting in 208. These numbers are just to give an impression of the number of necessary calculations for the small example assembly. The number of necessary comparisons and the time necessary for each comparison are the two rather independent parameters that determine the processing time for the whole process of generating the area based representation. Both parameters are topic of the following considerations about accelerating the process.

The general intersection calculation for a pair of areas based on their boundary data is time consuming and, therefore, is avoided whenever possible. If the two areas do not intersect and are even in a certain distance from each other the simple comparison of the rectangular bounding boxes of the two areas is sufficient for detecting this case of no intersection. The rectangular bounding box is a rough approximation of the areas boundary and, therefore, the support points of the polygons are compared next, if the bounding boxes overlap. The number of support points of the smaller area of the pair that are inside and outside the bigger area are counted, respectively. In case of all points of the smaller area are excluded from the bigger one, there is no intersection. If all

points are contained the whole area is contained resulting in the intersection area given directly by the pixel count of the smaller area. Finally, the situation that some points are contained and others are excluded leads to the conclusion that there is an intersection whose area has to be calculated for classifying the relation between the pair of areas.

I do the general intersection calculation using a rectangular pixel mask that corresponds to the rectangular bounding box of the bigger area. The entries of the pixel mask represent, whether the corresponding pixel is included in none, one or both areas. Those pixels marked as included in both areas are counted for the intersection area. For classifying the relations between the 208 pairs of areas for the example, just 13 general intersection calculations are necessary. In 145 cases the simple comparison of the rectangular bounding box is sufficient for concluding that no intersection is present and 50 decisions are made based on the support point comparisons.

Besides accelerating the individual comparison the number of pairs of areas that have to be compared to each other is reduced by exploiting the characteristics of the relations. In the beginning, the list of all areas is sorted according to descending pixel counts. In a first pass through this list, those areas that are similar are marked. Just one of them has to be taken into account for finding candidates for partial overlaps and for finding contained areas. In a second pass, partial overlapping areas are identified. The artificial union area is generated and further compared instead of the original ones. For these two passes, pairs of areas originating both from one segmentation process, that delivers an unambiguous partition the image plane, can be skipped for the comparison. Those areas can not be similar or partially overlapping. However, the grouping process does not deliver disjoint segments, as described above, leading to the necessity for carrying out the calculations in this case. In a final third pass areas that are contained within others are determined. An area that is contained within another one is removed from the list and related appropriately to the bigger area. The algorithm for searching contained areas is recursively started for the contained area and the list of remaining, smaller, areas. For data sets providing more than one layer of contained areas this procedure avoids the explicit generation of relations between areas that are implicitly given by the transitivity of the contain relation. In a postprocessing step the partially overlapping groups are matched to areas of the hierarchy.

To summarize, the following steps for building up the hierarchical representation of segmentation results are performed:

1. Sort list of areas:
Sort the list of areas according to descending pixel counts.
2. Relate similar areas:
Search for each area of the list similar areas within the rest list starting with the next element. Search stops, when elements of the rest list become too small for being similar. Similar areas are taken from the rest list and related to the area they are similar to.
3. Relate partially overlapping areas:
Search for each area of the list partially overlapping areas within the rest list starting with the next element. If list elements are linked to similar ones, the partially overlap criterion must be fulfilled for all the similar areas. If at least one of the

areas is descending from a grouping process, mark the relation and continue the search. Else exchange the partially overlapping elements in the list by the artificial union area and link the originals to this. Search further partially overlapping areas for the artificial union area, until nothing more found. Search similar areas for the final new artificial one. Finally, reorder the list again in descending order pixel counts.

4. Relate contained areas:
Search for each element in the list areas that are contained. If one is contained, delete it from the list and relate it to the surrounding area. Start recursive search of the rest list with the contained area, possibly building up several layers of contained areas.
5. Match partially overlapping groups to areas of the hierarchy.

This process results in a list of independent areas that themselves are differently related to other areas.

For the example discussed at the beginning of the considerations about algorithmic aspects 20 original areas have to be taken into account, resulting, firstly, in 190 calculations for intersections. Due to a newly generated artificial union the number of comparisons increases for the example to 208.

Applying the stepwise procedure, 10 candidate pairs for similarities descending from different segment sources and providing similar pixel counts are identified, where 5 simple bounding box comparisons are negative. 5 general intersections have to be calculated, where 4 pairs of similar areas are found. 1 calculation does not result in a similarity classification, but the information about the intersection of the pair is stored for later use.

Instead of the original 20 areas, the following pass is started with a list length of 16 areas. The 4 similar ones are just considered, if possible partial overlaps have to be confirmed. This case does not occur within the example. Therefore, the complete set of area pairs reduces from 190 to 120 elements after the first pass.

Within the second pass of the example 54 pairs of areas are extracted, as descending from from different segment sources They are candidates for partial overlaps. 30 pairs are rejected by the simple bounding box criterion, another 17 by the support point comparison. For the remaining 7 pairs, general calculation of the intersection was necessary, where 1 result was already calculated and stored during the first run. 1 pair is identified as partially overlapping and the new artificial union area is generated during the pass.

Finally, for the third and last pass the segment list includes 15 areas, resulting in theoretically 105 comparisons. For the calculation of parts that exploits the transitivity of the relation 63 comparisons have to be done, where 25 of them have already been done during the first and second pass. For the remaining 38 pairs, 29 relations are decided by the simple bounding box criterion and another 8 relations by support point comparisons. Just 1 general intersection has to be calculated.

Summarizing all these numbers, for the example, the given stepwise procedure reduces the number of necessary pairwise comparisons from 208 to 101. Considering the individual intersection calculations, 12 instead of 13 general calculations have to be processed. The number of calculations based on the support points of the boundaries

is halved by decreasing from 50 to 25, the number of bounding box comparisons drops from 145 to 64.

In general, the computational time depends on the composition of the considered N areas. If there are many independent sources of areas, more comparisons for finding similar areas have to be made. However, there will occur more similar areas that are identified at the first stage of calculation reducing the number of effective areas for the following steps. Applying stricter segmentation parameters or analyzing objects that exhibit more inner structure, increases the amount of areas, but the number of contained layers will also increase. Then the acceleration caused by the recursive processing for contained areas attains good results. If the increase of areas is due to enlarged objects more and more classifications of pairs of areas can be decided based on the simple bounding box comparison. For systems with critical processing time constraints, these aspects should be considered for selecting the number of different segmentation processes to be integrated and the parameterization of the processes themselves.

The generated representation contains besides area representing objects also those representing the background. Background areas generally enlarge the representation without supporting the recognition task. Their identification within the uppermost layer of the hierarchy is rather simple, if there exists one area or several similar ones that contain all the other areas. If the representation contains more than one independent area at the uppermost level, additional candidates can be optionally identified by their size, where the threshold applied for the realized system is set to about half of the image size. The identification of all background areas is not a requirement for applying the following interpretation, but it simplifies and accelerates it.

This is also the motivation for further generating sub units within the representation concerning a whole image by constituting clusters of neighbored areas/objects and separating the clusters from each other. Therefore, the independent areas at the uppermost layer of the hierarchy that do not exceed a given distance to each other are summarized into a cluster. The smaller sub units are treated independent from each other and thereby simplifies the interpretation and the evaluation.

4.2.3 Image Data Based Object Information

Object recognition processes, such as described in Sec. 3.2, deliver object information that is based on image data. Like discussed for the segmentation problem before, a single recognition process is generally not able to do a recognition task without failures. In order to get more reliable results, different recognition processes based on different approaches and assumptions should complement one another.

Thereby, the question arises, what object units are suitable to identify. For example, within the *baufix*[®] scenario elements as well as parts of elements, like a hole or the head of a bolt, and also assemblies of elements are meaningful object units. For the objects from an office environment, either the cup standing on a table or its handle, but also the fish sketch printed on the cup are candidates for the object units. The main object units are defined by the recognition task. Other object labels generally occur by applying different recognition strategies to the task. Their integration becomes important for the interpretation step, which is topic of the next section.

Additionally, the results of different recognition processes are distinguished by either containing segment information accompanying the object label or not. This difference is firstly independent from the given task. Generally applicable data driven recognition processes either base on segment information for object classification and, therefore, deliver segments together with an object label or they directly use the image pixel information and, consequently, deliver no segment information. The appropriate pixel data is collected from a predefined image area, like the circular or rectangular surrounding of an object location point. The two strategies lead to segment and point based object information, respectively.

Fig. 4.10 shows object recognition results for the *baufix*[®] domain generated from the hybrid approach (Fig. 4.10(b), Sec. 3.2.1) and the neural appearance based approach (Fig. 4.10(c), Sec. 3.2.3)

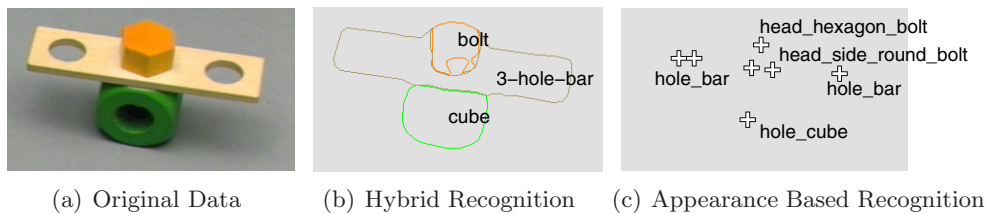


Figure 4.10: Exemplary object recognition results generated from two different recognition approaches for the *baufix*[®] domain.

The hybrid approach consists of a neural hypotheses generation and a semantic object model based verification step realized for *baufix*[®] element recognition. Color region information is used for detecting objects for the neural classification and influences strongly the model based verification step. Occlusions of elements within assemblies that result in significant changes of the object region shape, therefore, principally lead to uncertain recognition results and failures. The bar in Fig. 4.10(b) is an example for that even if the recognition process tolerates the slight occlusion here. Generally, the approach delivers a recognition result consisting of object label and object region information.

In contrast to this segment information, interesting points and the pixel information within a fixed circular surrounding are used for detecting and classifying objects by the appearance based neural classification approach. The module is parameterized to the *baufix*[®] task by taking into account image areas of sizes in the order of magnitude of the size of a typical hole. These small areas correspond to characteristic inner structures of the elements that mostly remain visible and, thereby, detectable even in the presence of occlusions of elements within assemblies. However, the classification itself is rather uncertain (about 30% misclassification on average), caused by the smallness of the windows and the accompanying small amount of underlying image information. Additionally, the results in the form of classified focus points do not contain any information about the number of objects and their segmentation from the ground, see Fig. 4.10(c).

The different segment information accompanying the object labels lead to different

strategies for their integration with the hierarchical representation of segmentation results, as described in the following.

Region Based Object Label Integration

For object labels delivered with region information the corresponding area is already included within the hierarchical common representation. Consequently the object label is assigned firstly to this one area, reflecting the complete information given by the recognition task. If this area is member of a set of similar ones, the object information is transferred to all members of the set. The object information assigned to an area that partially overlaps another one and, therefore, is part of an artificially generated union area is assigned also to the union area. The information about the special situation remains within the system for probable later use.

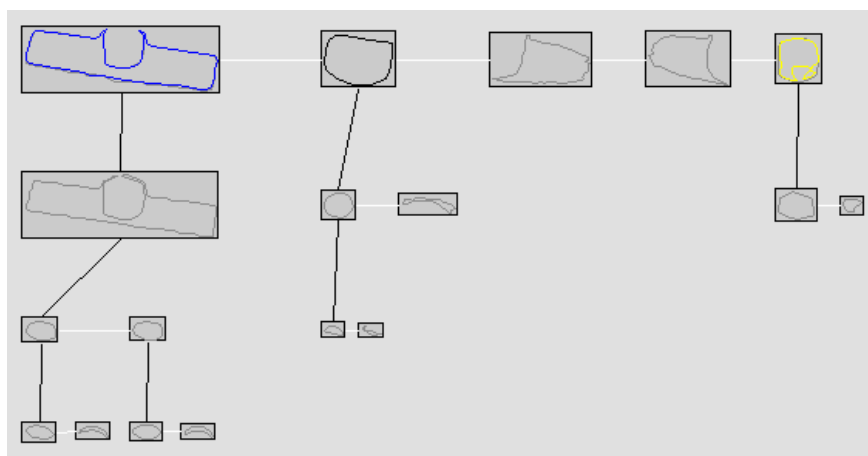


Figure 4.11: Simplified hierarchical representation of Fig. 4.8, on page 81, extended with region based object information of Fig. 4.10(b), on page 86. Object labels are color coded with blue for 'bar', black for 'cube', and yellow for 'bolt'.

Fig. 4.11 shows the hierarchical segment representation, already presented in Fig. 4.8, on page 81, extended by the region based object information depicted in Fig. 4.10(b), on page 86. The hierarchical representation is simplified by just showing independent and contained areas and leaving out similar and partially overlapping original ones.

The region based object labels are attached to the superior areas representing the bar, cube, and bolt, respectively. Thereby the superior bar and bolt areas are artificial union areas that each get the object information from one of its members.

Point Based Object Label Integration

Point based object hypotheses identify the surrounding of the focus point to depict some object class. The object region information has to be taken from the independently generated segmentation results by assuming image areas to be probable object regions

that cover the focus point carrying object information. Special attention must be paid to points that are located near the border of an area and with this near the neighbored or superior area. The surrounding of those points whose pixel information leads to the object label classification then is shared by the two areas and an unique assignment to one of the two areas may be not justified. For taking care of this situation, I introduce a threshold for the minimal distance between a focus point and the boundary of the related area. By choosing an adequate value for this parameter it is possible to require a significant part of the surrounding being covered by the area in order to assign the point based object information directly to it.

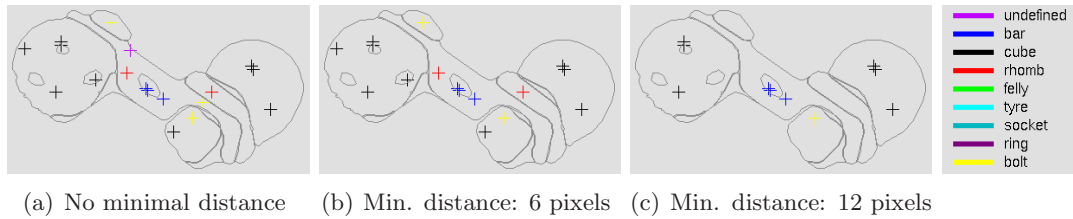


Figure 4.12: Effects of three different distance parameter settings on the point based object label assignment to segment information. Objects labels are color coded for visualization purposes.

Fig. 4.12 shows exemplarily the effect of three different distance parameter settings for the assignment of point based object labels delivered by the neural appearance based recognition approach, see Sec. 3.2.3, and the pixel color classification based segmentation, see Sec. 2.2.4, for the *baufix*[®] domain. The classification module uses image data within a radius of 25 pixels for determining the object label. Fig. 4.12(a) shows all object labels and their location relative to the segment information. Some of the object labels are located near object borders which makes the assignment to the appropriate object area doubtful. The most evident example is the 'bolt' label located on the bar between the lower bolt and the rhomb. Requiring a minimal distance between the object label location and the boundary of the area as a prerequisite for the assignment leads to less object labels being assigned. A distance threshold of 6 pixels results in 2 missing object labels, see Fig. 4.12(b) in comparison to Fig. 4.12(a). Because both former assignments were incorrect, the effect of the threshold is solely positive. A distance threshold of 12 pixels, which corresponds to the half of the radius that has been used for classification, results in already 8 missing labels of a whole of 18, with four of them were correct, see Fig. 4.12(c) in comparison to Fig. 4.12(a).

For integrating the point based object information with the hierarchical common representation of segmentation results, all segments that cover the object location by taking into account the potentially set distance threshold parameter are related to the object information. This relation is realized by directly assigning the object information to the innermost of these areas, i.e., the one not containing another area covering the point. The relation between the object label and the superior areas is given indirectly by the relation between the areas.

Fig. 4.13 shows the simplified hierarchical segment representation already used in

Fig. 4.11 with integrating the point based object knowledge that is depicted in Fig. 4.10(c), on page 86. For the area assignment a minimal distance between object location and area border of 6 pixels is required.

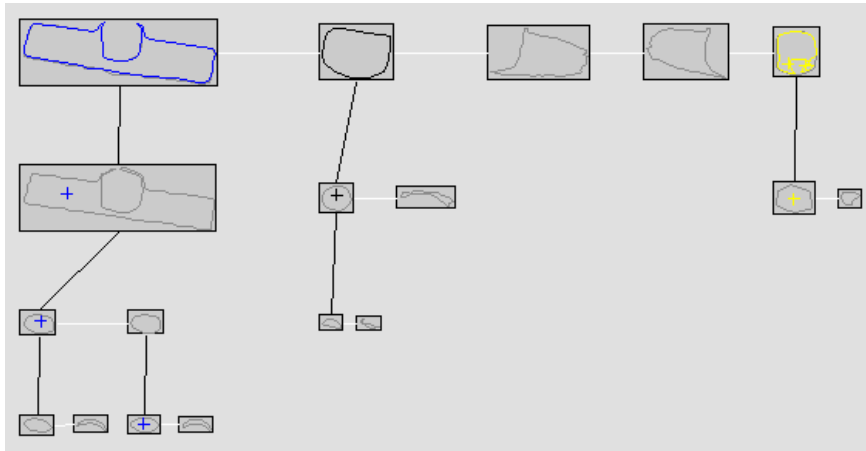


Figure 4.13: Hierarchical representation of Fig. 4.11 extended with point based object information of Fig. 4.10(c), on page 86. Minimal distance between object location and area border is 6 pixels. Objects labels are color coded with blue for 'bar', black for 'cube', and yellow for 'bolt' parts.

The point based object labels from the appearance based neural approach identify characteristic structures of the *baufix*[®] elements, like the holes of the bar and cube and the characteristic surfaces of the head of the bolt. These structures are mainly also represented by the segment information. Consequently, most object labels are directly assigned to subordinated areas that represent these structures. The minimal distance threshold thereby prohibits well the assignment to the very small inner areas located at the hole of the cube or the head of the bolt. However, for the object label located at the border of the left hole of the bar the application of the threshold results in the assignment of the point to the superior area representing the whole bar and inhibits a relation to an area representing the hole. The detailed discussion of the threshold parameter concerning its numerical setting and its influence on the results of the realized integration system is part of Chapter 5.

4.2.4 Summarized Characteristics of the Common Representation

The common hierarchical representation of segment and object information presented in the preceding section constitutes a general tool for structuring independently generated information of different types with regard to their integrated use for solving an object segmentation and recognition task. The representation scheme collects and stores the available information and structures it by establishing relations between different parts of information.

Independently generated segmentation results are related based on their common ground, the image plane. For the comparison each segment is represented by its outer

boundary and the degree of overlap between each two of the resulting areas determines their relation. A threshold based classification of the intersection area and the original areas delivers one of four different types of relations for pair of areas. The two areas are identified to be either similar to each other, one is contained within the other, they overlap partially, or they are independent, i.e., they have no or just a small intersection area. The common representation is generated by exploiting the semantics of the relations. A set of similar areas is represented by one member of the set and partially overlapping areas are merged resulting in an artificial union area that represent them within the hierarchy. For areas that contain others and those that are independent from each other not all of the pair relations are represented explicitly, but instead each area does not get more than two explicit relations of each of the two types. With this simplification the resulting representation is characterized by lists of independent areas that each generally constitutes the superior of a list of independent contained areas and so on. Individual ones of the independent areas are further related to similar or partially overlapping ones. The structure thereby provides less explicit and many implicit relations and is recursive without preferring one layer to another.

Exceptions have to be handled, if grouping hypotheses are integrated into the system due to the special capabilities of bridging gaps within the sensory data which complicates the comparison to other segment information principally. Partially overlapping groups get a special type of relation to areas within the hierarchy, which does not disturb the general structure.

The resulting representation of all the occurring segment information is simplified by identifying and deleting areas that represent the background and by generating independently treatable smaller sub units of the data constituted by clusters of neighbored areas.

Available object information is integrated with the hierarchical representation of the segmentation results by attaching object labels to appropriate areas within the representation. For object information accompanied by segment information the label is attached to the corresponding area. Point based object information is directly attached to the smallest area that covers the point and indirectly related to all superior areas via the area relations.

Due to the general formulation of the data integration the common hierarchical representation can be generated based on available segment and object information without the need of defining the number and kind of processes in advance. The set of processes may change from one run to another. This allows the flexible exchange of modules for firstly finding a good composition of the available processes and for replacing modules by improved ones with their availability.

Additionally during one run a representation generated based on the available information at one point of time can be extended by belatedly arriving parts of information. Belated information in this sense is not only delivered from time consuming processes for which the integrating system does not have to wait with its first results. The situation also principally occurs, if modules are included that are based on preliminary results like those exploiting object context. For generating the preliminary results available information is collected, represented, and interpreted. The new pieces of information generated based on that have to be integrated with the existing representation in order to provide an integrated reinterpretation. New segment information, potentially accompanied by

object information, introduce new relations into the system that generally change, at least locally, the former calculated structure. However, most of the earlier calculated relations remain valid. Especially the relations between the areas that are contained in a new one are not affected and constitute together with the attached object information the base of the extended representation. Belated object information either accompanied by already known segment information or not based on segments at all can be added without any changes in the structure of the former representation.

The common representation of available segment and object information constitutes the base for the following interpretation step that identifies object regions and labels based on the stored information.

4.3 Generating Hypotheses by Analyzing the Hierarchical Representation

The hierarchical common representation of segmentation and object information contains many pieces of information that have to be integrated to come to a consistent interpretation of the image containing of hypotheses for object regions and appropriate object labels. In the presence of uncertainties and failures the success of the approach to come up with indisputable object recognition results seems doubtful. A more promising proceeding is to identify generally several competing solutions in order not to lose the correct result by a precipitate decision.

By the integrating additional high level knowledge, like knowledge from object context (see Sec. 3.3), former less probable hypotheses might become favorites and vice versa. An early decision for one hypothesis prevents the integration of additional knowledge but allows its application just for verifying or discarding the candidate.

Due to the amount of information stored within the representation it is not desirable to generate hypotheses for all possible combinations of areas and occurring object labels. Instead a manageable reduced set of competing probable hypotheses has to be identified.

For generating this set, the interpretation step needs criteria for identifying probable object regions within the hierarchical representation and a mechanism for integrating the occurring different object labels.

The latter is the topic of the following subsection, before in the successive subsection the criteria for the identification of probable object regions are discussed.

4.3.1 Object Labels from Probabilistic Integration

Accounting for different recognition modules leads generally to several different object hypotheses that are related to an area within the hierarchical representation and have to be integrated in order to generate a resulting label for the area. There are different point or region based object labels directly attached to the area. Additionally, point based object information that is attached to contained areas has to be taken into account, if the contained areas are assumed to represent inner structures of the object. The choice of candidate areas and the treatment of contained areas is the topic of the following subsection and is assumed to be given here. Among the related object labels, there are

besides the correct labels generally also false ones, due to recognition failures within the individual recognition modules or due to establishing false relations between point based hypotheses and the available segment information.

Independent of their characteristic to be point or region based different object label alphabets may occur within the object information, like holistic labels for the whole object or recognition results for object parts delivered from part based modules. Collecting the object information related to an area of the hierarchical representation, therefore, generally results in a set of firstly incomparable object labels from that an integrated resulting label according to the recognition task and an appropriate evaluation of its (un-)certainty have to be determined.

For making the occurring different labels comparable the occurring original label alphabets, $OL_1 \dots OL_n$, the alphabet of the resulting object labels, RL , and their coherence have to be defined in once advance. The alphabet of the resulting labels may be equal to one of the incoming sets or may be additionally defined. A standardized coherence definition containing the mapping from each original label into the alphabet of resulting labels make these labels addressable for a general integration mechanism. The table containing the occurring object labels for the *baufix*[®] task is shown in the Appendix B, Sec. B.2.

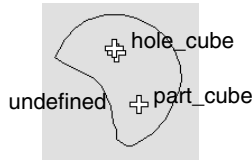
An object recognition result consists in its general form of a vector of confidence values, where each component represents an object label of the corresponding alphabet. A special result that is constituted by a label without any confidence information is equivalent to a vector with just one entries being non zero. It depends on the recognition module that generates the object hypothesis which amount of information about confidence is available.

Tab. 4.2 illustrates the different incoming pieces of object information for an area representing a *baufix*[®] cube. The region based 'undefined' hypothesis is based on the element label alphabet, while the point based hypotheses are defined within the set of sub elementary parts. The element label is given without any confidence information while for the sub elementary hypotheses evaluations are determined from the appropriate classification error matrix, see Appendix B, Sec. B.3 for details.

Mapping an incoming evaluation vector corresponding to an original object label set, OL_i , to another one corresponding to the resulting label alphabet, RL , is defined by the label to label mapping. If different original labels are summarized by one resulting label, their evaluations are summarized. The other way around, if one original label supports different resulting ones the corresponding evaluation is split. The resulting confidence vectors based on the common resulting object label alphabet RL are comparable and thereby integrable.

Integrating several object labels is the focus of classifier combination schemes, as presented and discussed in Sec. 3.1.4. The scheme that turns out to be generally most promising is the probabilistic integration by applying the sum rule. This rule demands to summarize the participating evaluation vectors and to determine the most probable label and its confidence value from this sum vector by identifying the component that provides the maximal value. This rule is applied here for integrating the set of object labels attached to a chosen area of the hierarchical representation. If several labels provide the maximal confidence, they are treated further as competing hypotheses.

If information about the relative reliability of the different sources of object hypothe-



OL_1	'undefined'	OL_2	'part_cube'	'hole_cube' (2 times)
3_hole_bar	0.0	undefined	0.1199	0.0510
5_hole_bar	0.0	part_cube	0.5341	0.0848
7_hole_bar	0.0	hole_cube	0.2125	0.8277
cube	0.0	edge_cube	0.0899	0.0000
rhomb	0.0	part_rhomb	0.0000	0.0000
felly	0.0	hole_rhomb	0.0000	0.0000
tyre	0.0	side_rhomb	0.0109	0.0064
socket	0.0	head_round_bolt	0.0109	0.0036
ring	0.0	head_side_round_bolt	0.0109	0.0000
bolt	0.0	head_hexagon_bolt	0.0109	0.0064
undefined	1.0	head_side_hexagon_bolt	0.0000	0.0137
		hole_bar	0.0000	0.0000
		edge_bar	0.0000	0.0064
		felly	0.0000	0.0000
		hole_felly	0.0000	0.0000
		tyre	0.0000	0.0000

Table 4.2: Object hypotheses related to an exemplary **baufix**[®] cube presented as evaluation vectors based on different object label alphabets.

ses is available this can be generally accounted for by weighting the confidence values of a hypothesis dependent on its origin before integrating it, as, for example, done in the system presented in Sec. 3.2.2. In the application here, where the recognition modules delivering object labels are flexible exchangeable, relative reliabilities between the sources of information are not applied. Instead, each object label contributes confidence values that are normalized to sum up to one, independent of its origin.

Tab. 4.3 shows the unified evaluation vectors of Tab. 4.2 and the integrated vector after applying the probabilistic sum rule and normalizing the resulting confidence values. In spite of the one occurring false label 'undefined', the integration step results in the correct object label 'cube' evaluated by a value of 0.67.

RL	'undefined'	'part_cube'	'hole_cube' (2 times)	integrated result
bar	0.0	0.0000	0.0064	0.0032
cube	0.0	0.8365	0.9125	0.6654
rhomb	0.0	0.0109	0.0064	0.0059
felly	0.0	0.0000	0.0000	0.0000
tyre	0.0	0.0000	0.0000	0.0000
socket	0.0	0.0000	0.0000	0.0000
ring	0.0	0.0000	0.0000	0.0000
bolt	0.0	0.0327	0.0237	0.0200
undefined	1.0	0.1199	0.0510	0.3055

Table 4.3: Object hypotheses of Tab. 4.2 unified to the common label set of general **baufix**[®] elements with their appropriate evaluation vectors and the integrated evaluation vector resulting from applying the probabilistic sum rule and normalizing.

For generally analyzing the common hierarchical representation of segment and object information the described probabilistic sum rule is implemented and applied, whenever an object label is assigned to an area of the hierarchy based on the different pieces of related object information. The integration step accesses the encapsulated task specific implementation concerning the occurring object label alphabets and their mappings via a generalized functional interface.

Equipped with the mechanism of integrating a given set of different object labels the analysis of the common hierarchical representation concerning the identification of probable object regions is addressed in the following.

4.3.2 Selecting Hypotheses for Object Regions

Probable object regions has to be identified from the great amount of segment information that is represented in the hierarchical structure. The represented relations support this interpretation process by clustering the the information that has to be taken into account for interpreting an image area. Further on, the one representative that is chosen for a set of similar areas and the one artificially generated union area covering several partially overlapping areas reduce the number of areas to be considered. Related grouping hypotheses are used as alternative object region representation within a postprocessing step after selecting probable object regions from the hierarchy.

Nonetheless, even for rather small examples the hierarchy contains several layers of contained areas that has to be taken into account for identifying object regions, as shown for two examples in Fig. 4.14 and Fig. 4.15. Areas that are contained within others occur due to inner structures of the object surfaces resulting from either surface paintings or highlights and shadows. Additionally different object sizes resulting in embedded object area projections, as for the bolt, felly, and tyre in Fig. 4.14, and under segmentation, as for the two bars in Fig. 4.15, are responsible for contained areas.

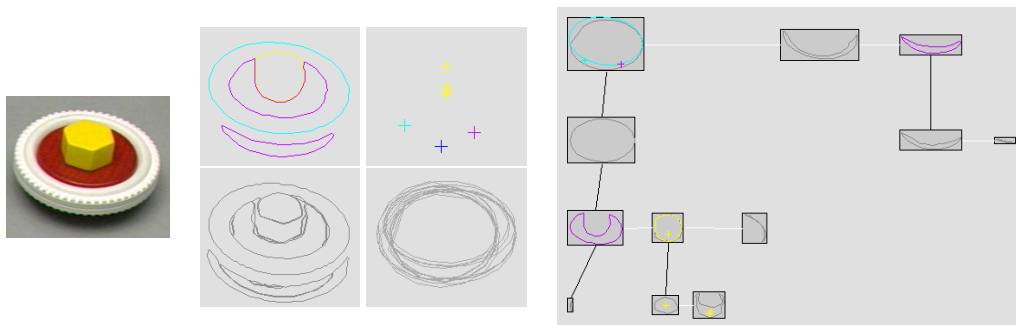


Figure 4.14: Exemplary hierarchical representation for a **baufix**[®] assembly. Projection of differently sized object regions determine the location of object regions. Left: Original image. Middle: Individual segment and color coded object recognition results (yellow: 'bolt', cyan: 'tyre', magenta: 'undefined', blue: 'bar'). Right: Common hierarchical representation.

These considerations also show that it is not sufficient just to assume each area of the uppermost level to represent one object and interpret the contained ones as less impor-

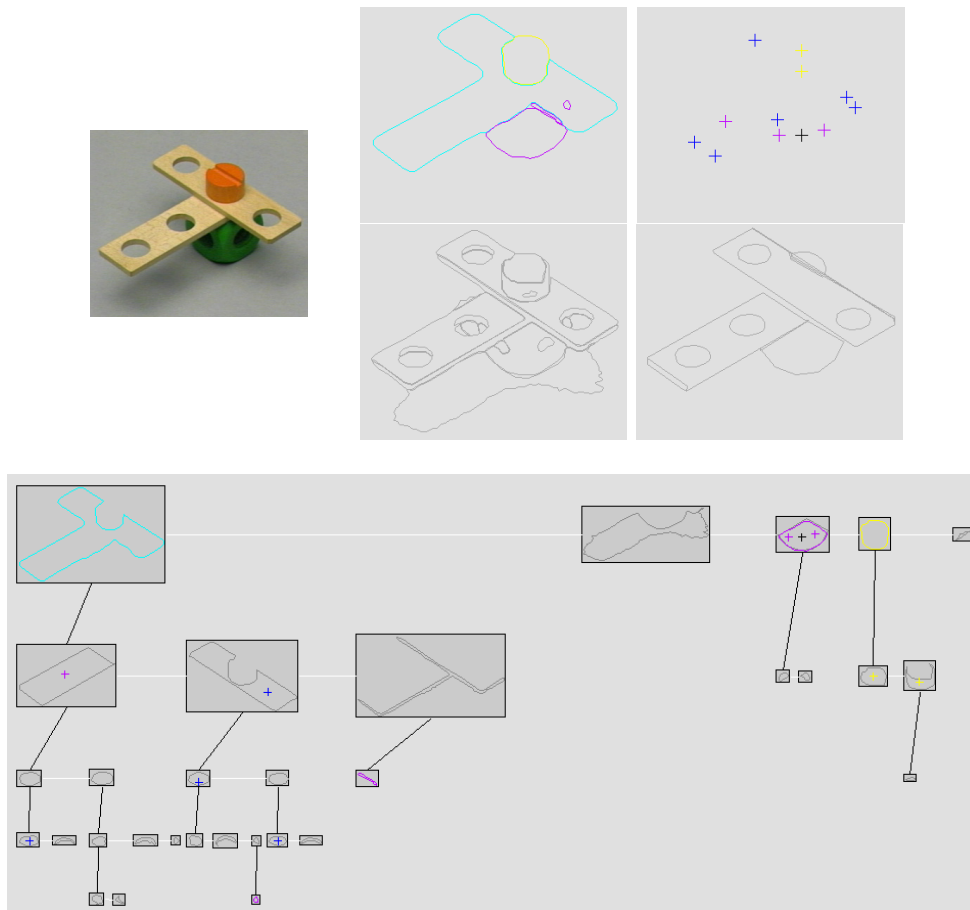


Figure 4.15: Exemplary hierarchical representation for a **baufix**[®] assembly. Under segmentation of the bar causes the location of object regions at the subordinated level. Upper left: Original image. Upper right: Individual segment and color coded object recognition results (yellow: 'bolt', cyan: 'tyre', magenta: 'undefined', blue: 'bar'). Lower: Common hierarchical representation.

tant inner structure. Instead, object regions possibly occur at each layer of the hierarchical representation. Criteria taking into account the contained areas and the attached object information are necessary for identifying them. This results in the general strategy for the analyzing step to interpret each area independent from its location within the hierarchy based on its directly attached information and the formerly generated interpretation for the contained areas. The analysis for each area proceeds recursively by first calling the analysis procedure for the contained areas and taking into account these results for the interpretation of the current area. Independent recursive interpretation procedures are started for the areas of the uppermost layer.

Uncertainties in the interpretation arises generally anywhere within the recursive interpretation process. Therefore, the interpretation step is generally allowed to deliver

several competing hypotheses, that constitute the basis for the integration of additional information like results from the context based modules, presented in Sec. 3.3.

After determining the general analyzing strategy the details of generating object hypotheses for a given area are as follows.

Rule Based Generation of Competing Hypotheses

The basic consideration for the interpretation of a given area within the hierarchical common representation is that it is firstly assumed to represent a significant structure of the image. This leads to generally hypothesizing the current area representing one object region and contained areas corresponding to inner structure of the one object. In the presence of structured surfaces and lighting effects causing highlights and shadows this hypothesis is justified. In addition to this unified hypothesis, competing interpretations arise by assuming the contained areas to represent individual objects. In order to keep the number of competing hypotheses small, the alternative interpretation is just build up if there is a hint for its justification.

This is given, if the current area is similar to the union of the directly contained areas. Then, it is not decidable without additional information, whether the current area is the result of under segmenting the individual object regions or the contained areas are the result of over segmenting the object region represented by the current area. The definition of two areas being similar to each other has already been used for generating the common representation, as described in Sec. 4.2 and is used here appropriately. The hypothesis providing individual objects for contained areas competes the unique interpretation of the current area.

Furthermore, an additional interpretation is justified by disagreements within the object information related to the current and the contained areas. A hypothesis containing several objects instead of one unique reduces the necessity of integrating disagreeing object label and, therefore, better corresponds to the given object information. One or more directly or indirectly contained areas are assumed to represent individual objects besides the current area embedding them and carrying the remaining object information.

These considerations lead to the rules for generating competing object hypotheses for a given area within the hierarchy, as summarized in Tab. 4.4.

Recursive Interpretation Procedure

The given set of rules is applied for generating competing hypotheses for a selected area of the hierarchy. For interpreting the current area by applying rule 2 and 3 object hypotheses for the contained areas have to be available. They are calculated by recursively starting the analysis for the contained areas, before the competing hypotheses for the current area are generated based on these results. The recursive analysis function proceeds a list of independent areas by interpreting each area dependent on the results of analyzing its list of contained areas. The resulting competing hypotheses for each area of the list are combined leading to a set of competing hypotheses each concerning the whole list of independent areas. The set of competing hypotheses interpreting the independent areas of the list constitutes the return value for the analysis function at each

<ol style="list-style-type: none"> 1. Generate the unified hypothesis for the current area by assuming all the contained areas and related object labels constituting one object. 2. Are the current area and the union of the directly contained areas similar to each other? If so, generate a competing hypothesis for the current area by assuming the equivalent contained areas to represent individual objects. 3. Are there disagreements between the object information directly attached to the current area and that delivered from the analysis of the contained, not equivalent, areas ? If so, generate a competing hypothesis for the current area including the interpretations of those subordinated areas carrying the disagreeing labels and the current area, together with the remaining segment and object information, as separate objects, respectively.

Table 4.4: General rules for generating competing object hypotheses for an arbitrary area within the hierarchical representation based on the results for the contained areas and the directly attached information.

step of the recursion. The algorithm for the recursive analysis of a list of independent areas is summarized in the following:

Analyze(independent_areas):

```

FOREACH area FROM independent_areas
  competing_independent_area_hypotheses = Apply_rule_1(area)
  competing_contained_hypotheses = Analyze(contained_areas)
  FOREACH contained_hypothesis FROM competing_contained_hypotheses
    competing_independent_area_hypotheses +=
      Apply_rule_2(area, contained_hypothesis)
  competing_independent_area_hypotheses +=
    Apply_rule_3(area, contained_hypothesis)
  END.
  Append (hypotheses, competing_independent_area_hypotheses)
END.
competing_list_hypotheses =
  Combine_independent_area_hypotheses(hypotheses)
RETURN (competing_list_hypotheses)

```

Starting the analysis for the independent areas located at the uppermost layer of the hierarchy, results in a set of competing hypotheses for these areas by considering the stored segment and object information.

The example, already used in Fig. 4.14, serves for illustrating the interpretation procedure. Fig. 4.16 shows the known hierarchical representation at its left and competing

hypotheses generated rule based during the recursive interpretation procedure at its right side. For simplicity the areas carrying no object information at the leaves of the tree that do not influence the interpretation procedure are left out.

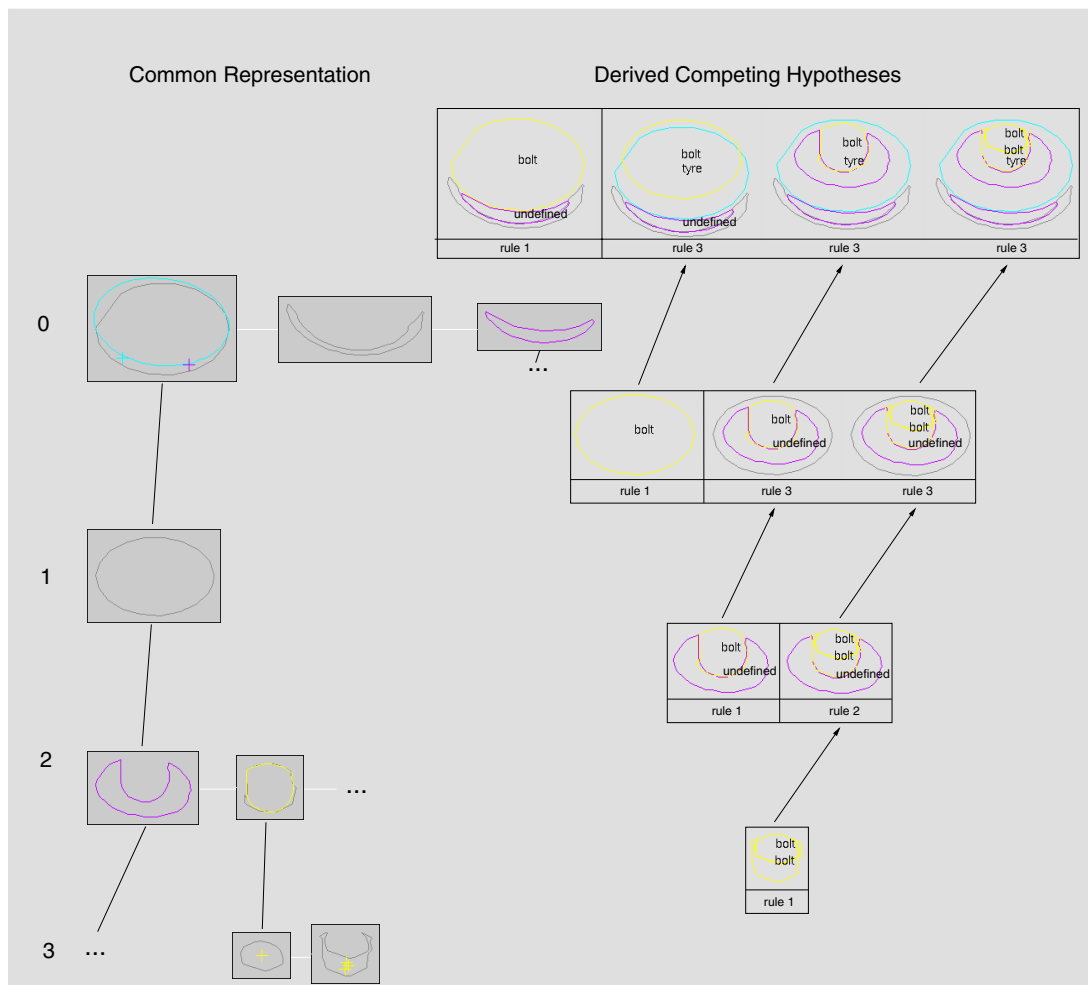


Figure 4.16: Rule based generation of competing hypotheses during the recursive analysis of the exemplary common hierarchical representation, already shown in Fig. 4.14. The result of the analysis are the competing hypotheses for the areas located at the uppermost layer that take into account the subordinated areas and related object information.

The hierarchy contains three areas at the uppermost layer due to the over segmentation of the area representing the tyre by all the integrated segmentation approaches. Without additional information this over segmentation can not be detected and, consequently, the two separated areas representing the contact surface of the tyre are assumed to represent individual objects and analyzed separately. Their analysis is rather simple and will not be shown in detail here. More interesting is the interpretation of the big area that represents the main part of the tyre and that includes the object regions

for the felly and the bolt. The analysis of this area first generates a hypothesis based on rule 1 by assuming the whole area representing one object and integrating all directly and indirectly attached object information. The object label integration, as described in Sec. 4.3.1, results in assigning the label 'bolt' due to the amount of indirectly attached point based 'bolt' labels. Further competing hypotheses for the interpretation of the area are generated based on the hypotheses for the contained areas. For the interpretation of the area at layer 1 again the unique hypothesis is generated following rule 1 and further interpretations depend on the contained areas. The recursion stops at layer 3, where no more areas are contained. The two innermost areas at level 3 are interpreted following rule 1 resulting in a hypothesis containing two independent bolts. This result determines the interpretation of the area representing the bolt at layer 2. Rule 2 is applicable for this area. The current area is similar to the union of contained areas and consequently the hypothesis containing the two independent bolts competes the unique interpretation of the bolt. For the area representing the felly at this layer just the unique interpretation with the object label 'undefined' is generated, because the requirements for further competing hypotheses are not fulfilled. The hypotheses for the interpretation of the areas located at layer 2 leads to the competition in the interpretation of the area at layer 1. Both results for layer 2 provide disagreements within the object information and thereby fulfill independent from each other the prerequisite for applying rule 3 for competing interpretations of the area at layer 1. This situation occurs again at the next level resulting in competing hypotheses for the areas at layer 0, which constitute the result of the interpretation step.

Fig. 4.17 shows some of the competing hypotheses generated for the example of Fig.4.15, on page 95. The special feature of this example is the simultaneous existence of over and under segmented areas for the bars and the bolt, respectively. It demonstrates the justification of rule 2 that leads to competition, if the layers at different areas are equivalent. A decision for one of the competing hypotheses is not possible without additional knowledge. The two competing hypotheses for each of the two areas result after generating all possible combinations in four competing hypotheses at all. Additional competition for the bars that is not shown here is caused by the different object labels related to this area and the application of rule 3.

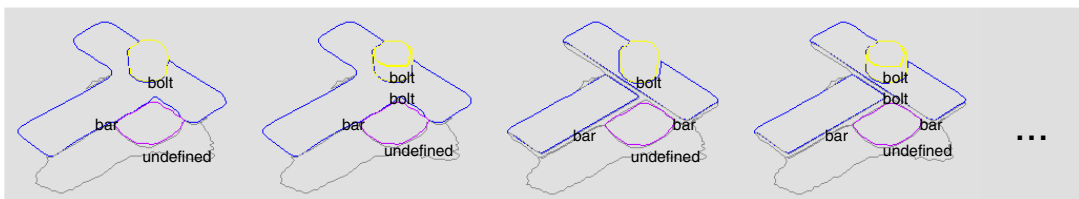


Figure 4.17: Chosen competing hypotheses for the examples of Fig. 4.15, on page 95.

The recursive procedure for analyzing the hierarchical representation exploits its recursive structure resulting in a short and comprehensive algorithm. The stored information about related segment and object information is exploited for generating object hypotheses for the image area that is covered by the areas at the uppermost level of the hierarchical representation. If there are uncertainties for the interpretation due to equiv-

alent, but different segmentation results, or disagreements within the object information competing hypotheses are generated in order to allow the integration of additional, for example context based, knowledge for clarifying and/or evaluating the results. The recursive procedure selects areas from the hierarchical presentation that are assumed to represent objects. Possible alternatives for these object regions are given by related grouping hypotheses that are analyzed within a postprocessing step.

Postprocessing Grouping Hypotheses

Grouping hypotheses that partially overlap other areas are exceptionally handled during the generation process of the hierarchical representation, as described in Sec. 4.2. This refers to their special characteristic concerning bridging gaps within the sensory data. Those groups are either matched to the areas originating from other sources by applying a matching procedure that is applied already in [Schl 00] or they remain in a list separate from the common representation.

A matched group is related to the appropriate area within the hierarchy. The matching process determines, whether the group is assumed either to represent an alternative for the boundary of the area, or to represent an inner structure of the area. Each matched group opens an alternative for the related area that does not influence the generation of competing hypotheses during the recursive interpretation step. Therefore, the groups are not taken into account during this process, but just for the finally selected areas that are assumed to represent object regions. Within a postprocessing step, additional competing hypotheses are generated, by assuming the closeness group to represent an object area. In case of a boundary match, the appropriate area does not occur within the hypothesis, in case of a structure match, the hypothesis gets an additional object assumed to be represented by the group. The object information formerly related to the area is integrated appropriately.

The closeness groups that does not match to one area and, therefore, remain within a separate list are, finally, taken into account. A matching area can principally not be identified, if the group concerns more than one area. Such a group is used, also, for generating a competing interpretation by hypothesizing the group to represent an object region and leaving out the objects whose areas are contained within the group.

The postprocessing of partially overlapping grouping hypotheses generate competing hypotheses in addition to the rules, given in Tab. 4.4, as formulated in Tab. 4.5.

The rule is applied to all the competing hypotheses resulting from the recursive analysis. Fig. 4.18 shows examples for additional competing hypotheses based on matched grouping hypotheses. The first example, shown in Fig. 4.18(a), demonstrates the effect of a boundary match, where the group substitutes the appropriate area. Fig. 4.18(b) shows the competing interpretation based on a structure match. The group representing the upper bar is matched as a structure to the cross area at the upper level, while the expectable boundary match to the corresponding area at the subordinated layer can not be established. In this case, the structure is assumed to represent an individual object, while the superior area remains within the set of hypotheses and is assumed to be hidden by the object represented by the group. Object information is assigned correspondingly, resulting, here, in the two hypotheses for a bar and a tyre, respectively.

<p>G Do one or more object regions of the current set of object hypotheses overlap one or more grouping hypotheses ?</p> <p>If a region hypothesis matches a group, generate a new competing hypothesis by adding the group as newly assumed object region and substitute the matched area, if its boundary matches the group.</p> <p>If a group is not matched, generate a new competing hypothesis by adding the group as newly assumed object region and substitute the object region hypotheses that are contained within the group.</p> <p>Newly assign and integrate the involved object information.</p>

Table 4.5: Rule for additionally generating competing hypotheses based on partially overlapping grouping hypotheses.

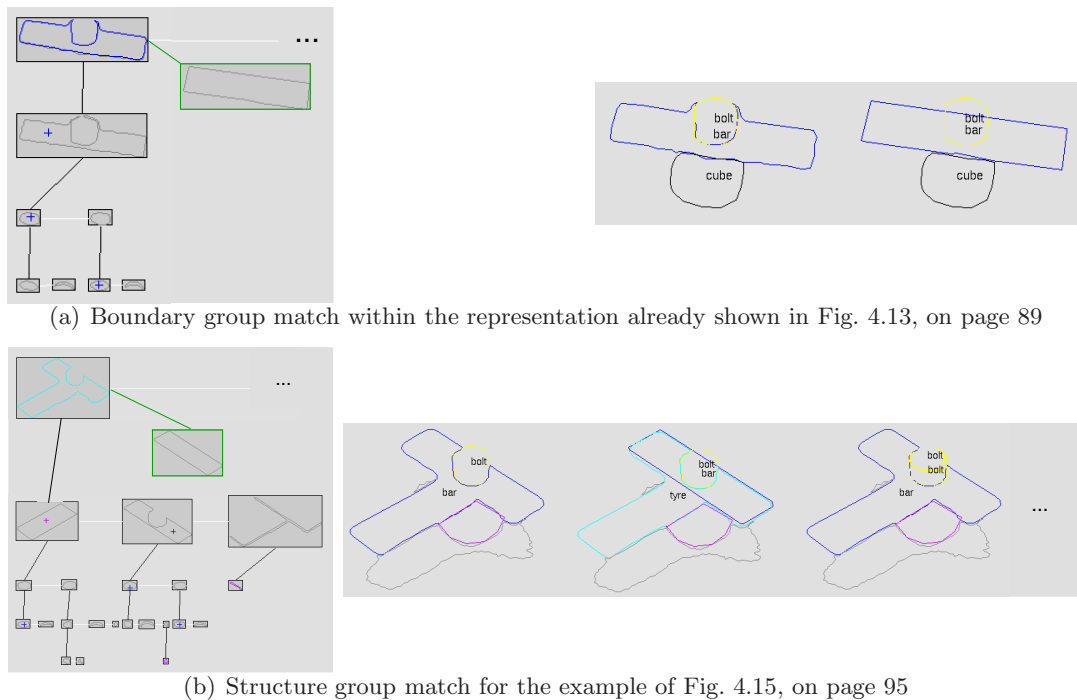


Figure 4.18: Exemplary competing object hypotheses generated based on matched grouping hypotheses.

Fig. 4.19 shows an example, where a group is not included within the hierarchical representation, because it provides partial overlaps with some of the shadow areas at the border of the mousepad and is not matched, because it concerns more than one area. The structured mouse pad is over segmented by the integrated segment information. The contour based grouping is able to identify the object region as a whole and summarizes the three former independently hypothesized object regions.

By the postprocessing analysis of the group matches the additional independent seg-

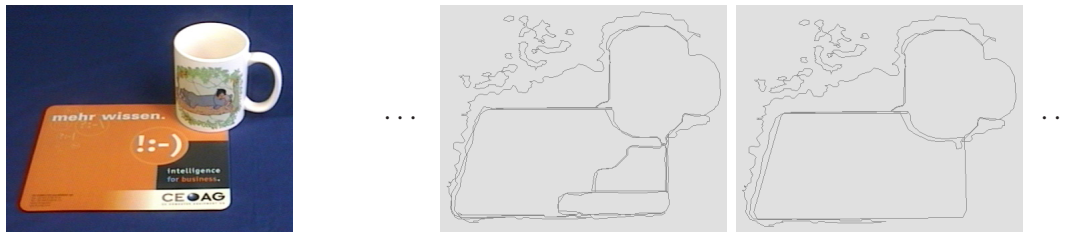


Figure 4.19: Competing hypotheses arising from an assigned grouping hypothesis that summarize several hypothesized object regions.

mentation information that is provided by the groups is exploited, in spite of their exceptional status that results from their special characteristics.

The competing object hypotheses generated by integrating independent image data based modules constitute the basis for integrating additional sources of knowledge. Information that depends on intermediate hypotheses is, for example, context knowledge that exploits relations between neighbored objects. The competing hypotheses are prepared to enable the equitable integration of the additional information by preserving the information that caused each hypothesis.

4.3.3 Information Content of the Competing Hypotheses

Competing hypotheses constitute, generally, not the final result, but serve as a basis for integrating further information, for example from context knowledge or expectations. For the belated integration of information, the hierarchical common representation is suited, as discussed in Sec. 4.2. Therefore, the information that is used for deriving a hypothesis is stored in the form of the hierarchical representation.

Fig. 4.20 shows the configurations for the competing hypotheses of Fig. 4.16. The area at the uppermost level represents the assumed object region, while contained areas are assumed to represent inner structures. Object information remains related to the appropriate area.

The given competing hypotheses together with their knowledge representation are open for additional information by reusing the described integration and interpretation mechanisms.

4.4 Additional Information Dependent on Preliminary Hypotheses

Independently generated image data based segment and object knowledge are integrated for generating intermediate object hypotheses. These hypotheses are verified, extended and evaluated by appropriately exploiting and integrating additional modules. They are characterized here by their dependence on preliminary results, in contrast to the already discussed independent modules that work directly on the image data, see Fig. 4.1, on page 72. The so defined additional modules can be divided into two classes dependent on their output.

4.4 Additional Information Dependent on Preliminary Hypotheses

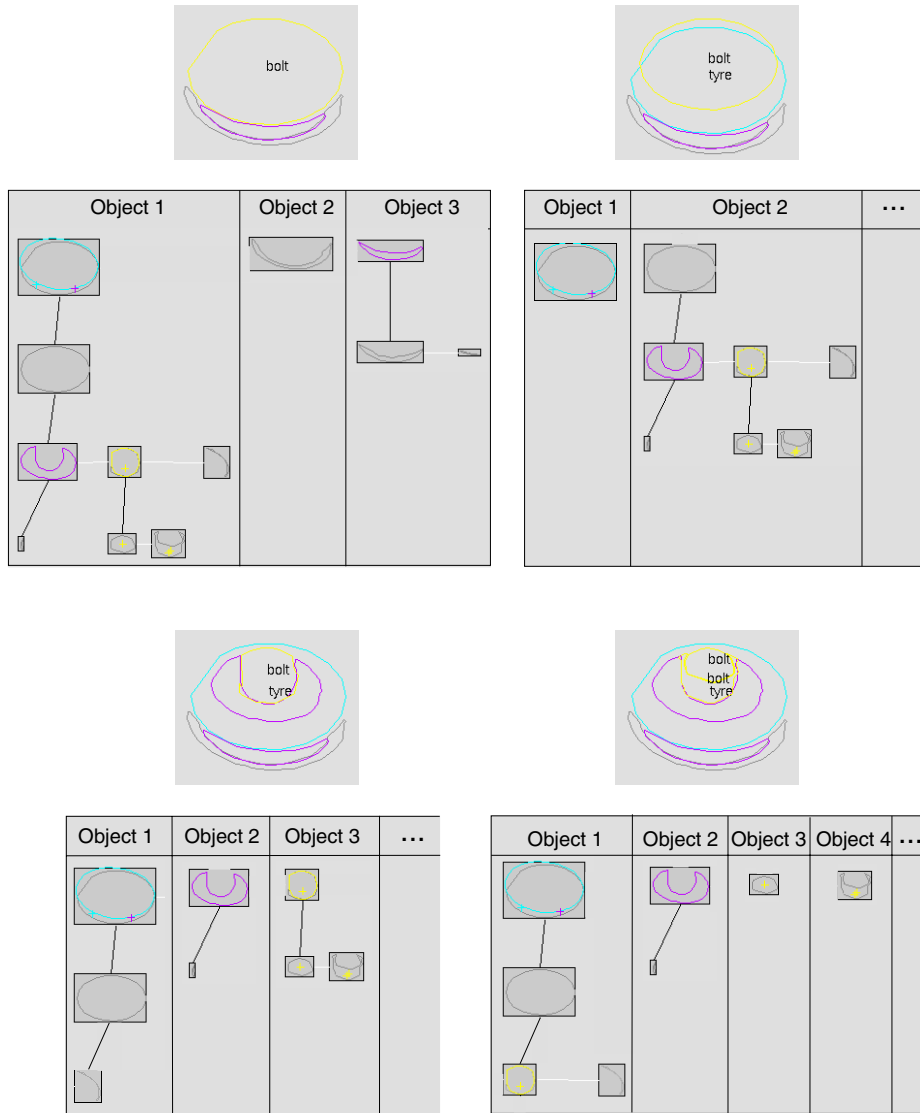


Figure 4.20: Information content of the competing hypotheses of Fig. 4.16 preserved in the form of the hierarchical representation.

On the one hand, there are the modules that take preliminary object hypotheses as their input and generate their additional information related to this input data. The sources of knowledge they exploit internally can be very different reaching from high level context knowledge, as for the *baufix*[®] assembly recognition module, to a comparably low level classifier, like the shape based recognition module for office objects.

On the other hand, sources of information that are beyond the visual data, like the temporal context or history of the scene and other modalities, like speech recognition and understanding, provide valuable additional data. Their output consists of object information that constitutes expectations for the recognition results without being firstly related to the image data or the preliminary object hypotheses. For integrating this in-

formation the relation has to be established explicitly, as described in the following. The subsequent section, finally, treats the integration of the localized additional information originating either from related expectations or from applying higher level recognition modules directly to the preliminary object hypotheses.

4.4.1 Localizing Object Information from Expectation

Expectations for the objects to be recognized in an image are characterized by consisting of object labels and probably additional features, like the object color, but generally not containing any information about the object localization in the image. Furthermore, often no information about the visibility of the object in the image is given. For example, monitoring the construction process of an *baufix*[®] assembly, as described in Sec. 3.3.3, provides expectations for the participating elements, when the assembly is finished and placed onto the table again.

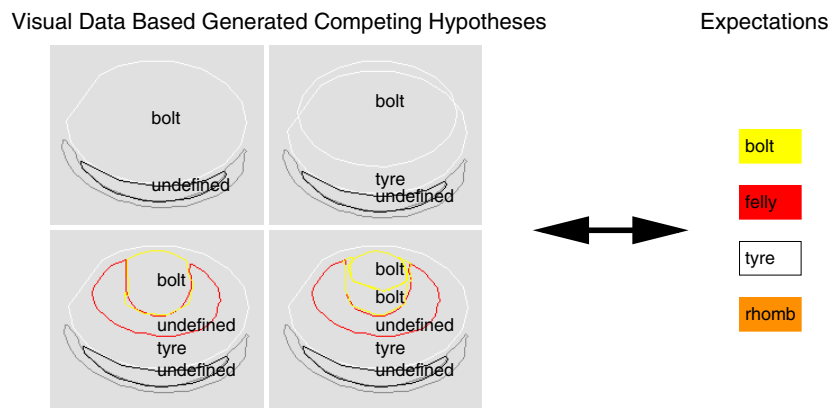


Figure 4.21: Left: Competing hypotheses generated from integrating image data based color segment and object information. Right: Expectations derived from additional sources of information, here construction process monitoring, containing color and label information.

Fig. 4.21 shows on the left for an example assembly already discussed before four competing results containing object region, color and label hypotheses and on the right side an appropriate set of expectations. The example motivates, on the one hand, that the localization and integration of the expectations offer the possibility of recognizing elements, here the *felly*, which the image data based modules are not able to identify. On the other hand, the integration step has to avoid to hallucinate the *rhomb* anywhere in the image, although it is also expected. For generating the desired hypothesis containing the object label '*felly*' together with the red region, the expectations has to be related to the existent hypotheses.

The process of establishing relations is complicated by occlusions and recognition failures, like occurring false color classes and false, missing, or superfluous object labels. Therefore, determining locally correct assignments is not promising, but a set of globally consistent and probable matches has to be identified. The procedure is implemented

based on definitions of costs for possible pairwise matches. For a set of object hypotheses and appropriate expectations all possible pairwise matches are generated and evaluated. Additionally, a match of each expectation to no object that may occur due to an occlusion and a match of each hypothesis to no expectation that may occur due to a falsely identified object are considered. From the created area of costs the globally most consistent matches are identified by minimizing the summarized matching costs.

The definitions of the cost for a match, thereby, depend on the kind of expectations and the amount of object features and therefore, generally, have to be adapted to the recognition task. For the *baufix*[®] task, where expectations from monitoring the construction process are available, information about object labels and colors are given and considered for the cost definitions, given in Tab. 4.6.

Cost	Assumed for a Match between
0.0	equally colored and labeled hypothesis and expectation
0.5	hypothesis and equally labeled but differently colored expectation or color region with no object label and equally colored expectation or expectation and no hypothesis
1.0	hypothesis and equally colored but differently labeled expectation or color region with no or undefined object label and no expectation
100.0	differently colored and labeled hypothesis and expectation or hypothesis and no expectation

Table 4.6: Definition of costs for matches between an object expectation generated from monitoring the *baufix*[®] assembly construction process and a image data based generated preliminary object hypothesis.

These costs are determined manually following the assumptions that matches between partners that agree on color and label cause no costs. Isolated disagreements in either color or type and due to missing information has to be tolerated, because they are assumed to originate from recognition failures and occlusions. Thereby color misclassifications are assumed to be more probable than label misclassification. High costs are defined for matches, where both color and type disagree and in case of a valid hypothesis can not be matched to an expectation.

After generating an area of costs containing all possible pairwise combinations between expectations, hypotheses and the special matching partners representing no hypothesis and no expectation the globally most consistent solution has to be identified by minimizing the sum of costs caused by the assumed matches. Thereby, each expectation has to be matched definitely once and each hypothesis has to be matched definitely once, while matches containing the special matching partners representing no hypothesis or no expectation are allowed to occur more than once. The search for the optimal match is implemented by searching an optimal path through the area of costs taking into account the prerequisites of unique matches. First, the match for the first hypothesis that causes minimal costs is identified. Possible successors are determined by identifying the remaining possible matches for the second hypotheses. The summarized costs for all intermediate path possibilities decide about, whether one path is followed for the next hypothesis or an alternative path has to be chosen that provides different

matches for the last and actual hypothesis. After assuming a match for each hypothesis, possibly remaining expectations are handled separately. By taking into account all possible matches the optimal solution is found by this procedure. In case of more than one path with equally minimal summarized costs exist, the resulting matches finally depend on the sequence of the occurring hypotheses and expectations.

Fig. 4.22 shows the areas of matching costs and the resulting optimal path for matching the four sets of competing hypotheses to the appropriate expectations for the **baufix**[®] assembly depicted in Fig. 4.21 by applying the matching cost definitions given in Tab. 4.6.

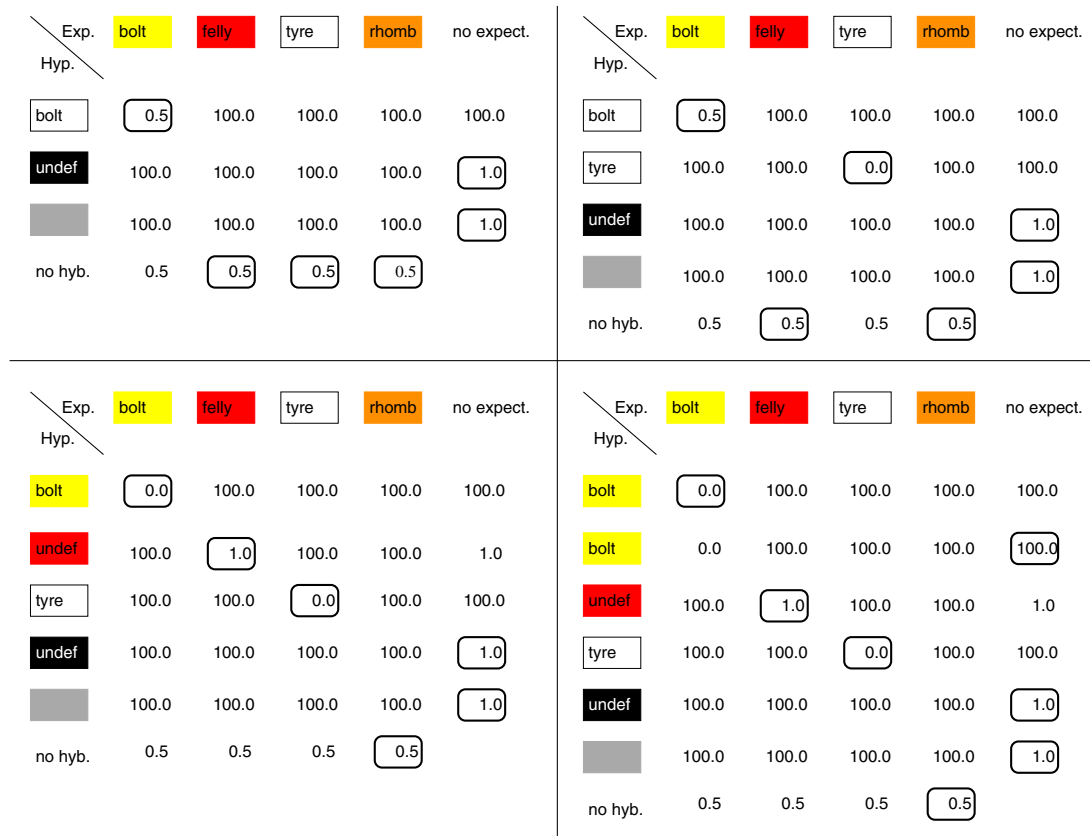


Figure 4.22: Matching results for expectations and the four sets of competing hypotheses for the **baufix**[®] assembly shown in Fig. 4.21 using the matching cost definitions given in Tab. 4.6.

For the third and fourth set of hypotheses, the occurring red object region labeled 'undefined' is correctly matched to the 'felly' expectation. The expected rhomb that is not visible is also matched correctly for all sets of hypotheses leading not to a false knowledge based object label. The marked matches of the fourth set of hypotheses (lower right) are not unique, because the both yellow bolt hypotheses are equivalent and it is not decidable based on the costs, whether one or the other matches the expectation and, thereby, is supported by integrating the information from expectations. Besides

additional information about objects the matching of expectations delivers a valuable aspect for evaluating one set of competing hypotheses in comparison to the others. The formulation of an evaluation criterion based on the degree of fulfilled expectations is described in Sec. 4.5. In the following firstly the mechanism of integrating the additional information with the data driven information represented within the competing hypotheses is described.

4.4.2 Integrating Additional Information

Additional information is characterized by its dependence on preliminary recognition hypotheses, as stated above. Therefore, its integration has to be done independently for each of the competing results. Thereby, a equivalent integration of the additional information is aspired, instead of a general dominance of one source of information. Image data based objections to a knowledge based information are possible. Similarly, agreements between data based and additional high level information increase the belief in this information or clarify formerly uncertain situations.

Fig. 4.23 shows exemplary integrated visual data based object information for a *baufix*[®] assembly, where the disagreement within the object information for the upper cube leads to a false recognition result. As shown, the difference between the weights for the 'cube' and the 'undef' label is very small, the decision is uncertain resulting in a confidence value of just 0.52 (after normalization) for the winning 'undef' label. An additional 'cube' label from higher level knowledge, like for example the context based assembly recognition module or an expectation localized by matching, as described above, clarifies the situation. Then, two 'cube' labels from different sources are opposed to one 'undef' label, leading to a rather sustainable decision without preferring one source of information to the other.

The belated integration of additional knowledge with the image data based information is supported by the internal knowledge representation of each preliminary hypothesis. It is of the form of the common hierarchical representation for segment and object information used for formerly generating the hypothesis, see Sec.4.2. belated integration of information. Additional segment information probably changes parts of the hierarchy. Object information is related to the appropriate area, where an individual weight for the given label is accounted for, if it is delivered from the recognition module, as it is done with the data based information. In order to preserve the flexibility of exchanging modules the different sources of information do not get individual weights, as also argued for the data driven integration. The integration of additional information is completed by reinterpreting the hierarchical representation using the mechanism described in Sec. 4.3 resulting in competing hypotheses taking into account all available information. For further processing the object recognition results either a decision for the best believed hypothesis or at least accompanying confidence values are necessary. The following section deals with this topic.

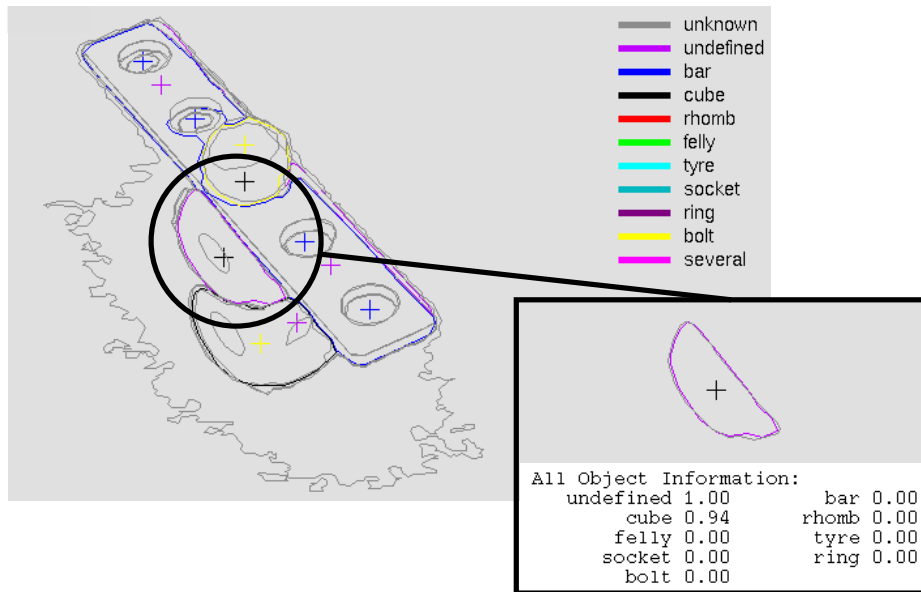


Figure 4.23: Integrating data driven information containing disagreeing object information leads to a false and uncertain interpretation for the upper *baufix*[®] cube. The situation can be clarified by additional object knowledge.

4.5 Evaluation of Competing Hypotheses

For generating a unique recognition result the competing object hypotheses generated by analyzing the common representation has to be evaluated. Even if a unique result is not required, a belief value that accompanies each of the competing hypotheses constitutes valuable additional information for subsequent processing, like, for example, the integration with results from other modalities, as proposed in Wachsmuth [Wach 01].

For reliably evaluating competing hypotheses, on the one hand, different kinds objects characteristics should be exploited, and, on the other hand, a resulting system should be comprehensible and flexible for integrating new aspects. This results in the need for a flexible integration scheme for evaluation information, as it is argued for the recognition step. Independent evaluation criteria that are either generally applicable or task specific constitute the basis of the evaluation step.

In the following first individual criteria are shortly discussed, before the general combination scheme is addressed in the subsequent subsection.

4.5.1 Independent Evaluation Criteria

For the evaluation of object hypotheses, besides task specific aspects, a generally applicable criterion is applicable that measures the degree of accordance for the integrated object information, as described in the following.

Mean Degree of Object Label Accordance

The competing hypotheses that have to be evaluated and compared to each other consist of object hypotheses whose object labels are integrated probabilistically from different sources of object information, as described in Sec. 4.2.3. Each integrated object label, thereby, gets a confidence value that accounts for agreements and disagreements within the related object information.

For all systems realized based on the general integrating framework, where object information is available from at least either one individually evaluated source or two not necessarily individually evaluated independent sources the criterion can be calculated and applied for evaluating competing results.

Because a competing result consists, generally, of a set of object hypotheses, the set is evaluated by the mean object label confidence:

$$\text{eval}_{\text{meanObj}} = \frac{1}{\#\text{all objects}} \cdot \sum \text{eval}_{\text{individualObj}}$$

Fig. 4.24 shows the competing hypotheses of Fig. 4.20 with appropriate confidence values for the individual objects and the mean value calculated for the competing sets of hypotheses.

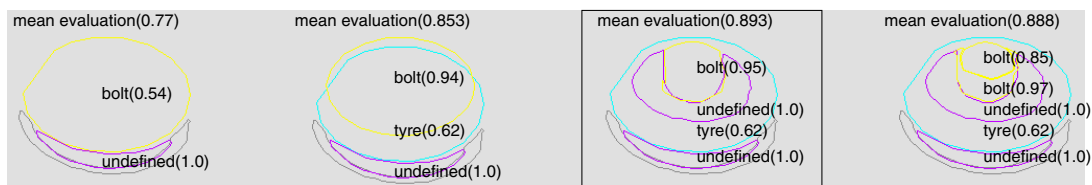


Figure 4.24: Image data based generated competing hypotheses of Fig. 4.20 with object label confidence values resulting from the probabilistic integration of different sources of object information and the mean values evaluating the competing sets of hypotheses.

The belief value is maximal for the third set of hypotheses, which is really the best. The mean object label evaluation differs significantly for the first set of hypotheses due to the low individual value for the 'bolt' label that is caused by the disagreement between 'bolt' and 'tyre' labels. In identifying the tyre as an individual object this disagreement is avoided and the individual as well as the mean object label evaluations increase appropriately. The two remaining sets provide with the felly area labeled 'undefined' an additional, well believed object that increases the mean evaluation further. The difference between them is caused by the object information concerning the bolt. A region based object hypothesis, see Fig. 4.20, supports the correct interpretation and is not taken into account for the variation of splitting the area representing the bolt, as it is part of the last set of hypotheses.

However, the criterion based on the mean object label evaluations tends towards favoring over segmented hypotheses, because they provide, generally, fewer disagreements within their related object information.

Domain Specific Criteria

Besides the generally applicable criterion for evaluating competing hypotheses a realized recognition system addressing a definite task often provides specialized strategies for evaluation. For example, for the *baufix*[®] task, a set of neighbored object hypotheses is assumed to represent the elements of a valid *baufix*[®] assembly. A set of hypotheses, therefore, is evaluated by taking into account the part of assembled elements from all. Besides calculating such continuous confidence values, the bivalued evaluation, whether individual object hypotheses or a constellation of objects fit into appropriate models or not is applicable to do a rough division of the competing hypotheses. The pre divided hypotheses have to be finally processed by applying additional criteria concerning the object labels and regions.

4.5.2 Combination of Individual Criteria

The general integrating module realizes the combination of individual evaluation criteria by implementing a frame for realizing a cascade of evaluation steps.

Thereby, the system designer defines a sequence for the application of available criteria dependent on the task. The integrating module calls the evaluation function with highest priority first and determines the ranking of competing hypotheses following its results. In case of ambiguity, the criterion located at the next step of the cascade is called and its results are applied for ranking within the sub set of formerly ambiguous hypotheses, etc.

The strategy of independently calculating and applying criteria at different levels of the cascade provides the great advantage of avoiding the direct comparison of the different confidence values. This comparison is carried out, for example, by calculating a sum, product, or other functional value based on the participating confidence values. Calculations based on the independent values become easily comparisons of apples and oranges that do not provide a common basis. For example, what is the common basis for comparing a confidence value based on a object region criterion with one evaluating the object context, in spite of their probable common data range, the interval $[0, 1]$? Independent of the semantic of the confidence value, the direct comparison is, generally, problematic and leads to undesired effects in case of very differently distributed confidence values. Examples are the continuous evaluation of the mean object label accordance in contrast to a bivalued pre dividing function.

The cascade for integrating evaluation information preserves the independence of the individual criteria that are made available from the system designer. Thereby, applicable criteria reach from simple plausibility tests to complex evaluation functions that might internally consider several aspects of the hypotheses for determining their result.

The ranking of the competing hypotheses and, thereby, the selection of the best believed one is done based on the defined cascade of evaluation criteria. The individual resulting object hypotheses are accompanied by the confidence value determined from evaluating the integrated object label information.

In summary, the cascade for integrating evaluation information implemented within the general integrating module provides a basis for flexibly realizing an evaluation scheme for a given task based on generally applicable or task specific criteria.

4.6 Summary

This chapter describes the implemented integrating module that constitutes the central part of a general framework for realizing recognition systems by integrating different segment and object recognition modules. The framework proposes the explicit integration of independently generated image data based information with additional information that is characterized by being dependent on preliminary object assumptions. The framework addresses the joint problem of object segmentation and object recognition.

The implemented integration module is characterized by three major parts, the common representation of available information, its analysis resulting, generally, in competing object hypotheses, and the comparison and evaluation of these competing hypotheses.

The common representation of the available information concerning segment and object knowledge has to store the amount of different pieces of information and simplify the interpretation by establishing suitable relations between them. Segment and object information is related, if it concerns common areas of the image plane. The integration step is flexible concerning the number of integrated modules. Further on, the data structure is open for the belated integration of additional information descending from modules with longterm processing times or modules relying on previously generated hypotheses.

The analysis of the generated representation constitutes the second major part of the integrating module. It exploits the information stored within the relations of the common representation and generates hypotheses containing object region and label assumptions. The process follows general rules and results due to uncertainties and failures mostly in several competing hypotheses. The rules fulfill the task of concurrently avoiding premature and, thereby, probably false decisions and constraining the amount of occurring hypotheses in order to keep the system capable for further processing.

Competing hypotheses, firstly, generated from integrated image data based information constitute the basis for further considering additional information. It is delivered from sources, whose applicability depends on preliminary results, like knowledge about spatial object context, or non localized expectations. The integration of additional knowledge with the data based generated hypotheses is equitable without one source of information dominating the other, which is supported by the strategy of the constructing the common detailed representation.

The evaluation strategy for competing hypotheses constitute the third part of the integrating module. The evaluation serves for generating a unique, best believed result and provides additional information about the confidence of the object hypotheses for subsequent use. The evaluation scheme is implemented as a cascade of individually applied evaluation criteria, where the first step determines the main ranking of the competing hypotheses and the following ones are applied in case of ambiguity.

The proposed integrating module implements general and, thereby, domain independent strategies for explicitly and flexibly considering different kinds of segment and object information. For realizing an integrated recognition system for a given task, individual modules following more or less domain specific strategies are reused, adapted or newly implemented. The integration of the available information does the general implemented module that is supplemented with task dependent information. This concerns

4 The Integrating Framework

object labels and their coherence, as well as suitable criteria for evaluating competing hypotheses and localizing expectations, if they are available.

The topic of the next chapter is the qualitative and quantitative evaluation of recognition systems that are realized based on the proposed integrating framework providing different system configurations and addressing two different recognition tasks.

5 Evaluation of Realized Integrated Systems

The general framework proposed in the previous chapter is able to integrate different kinds of information for object segmentation and recognition. This chapter deals with the realization of recognition systems addressing two different tasks based on the integrating framework. Fig. 5.1 shows exemplary images and recognition results.

A qualitative and quantitative evaluation of the system facilities is discussed. Thereby, the quantitative evaluation is not restricted to the final result, the rate of correctly segmented and labeled objects, but addresses the key aspects of the integration method. These aspects are the construction of the common representation, the process for extracting object hypotheses from the representation, the integration of additional knowledge and the estimation of confidence values for the results. Before getting into the detailed evaluation, the recognition tasks and the components of the realized systems as well as the conditions, their evaluation is based on, are presented in the following.

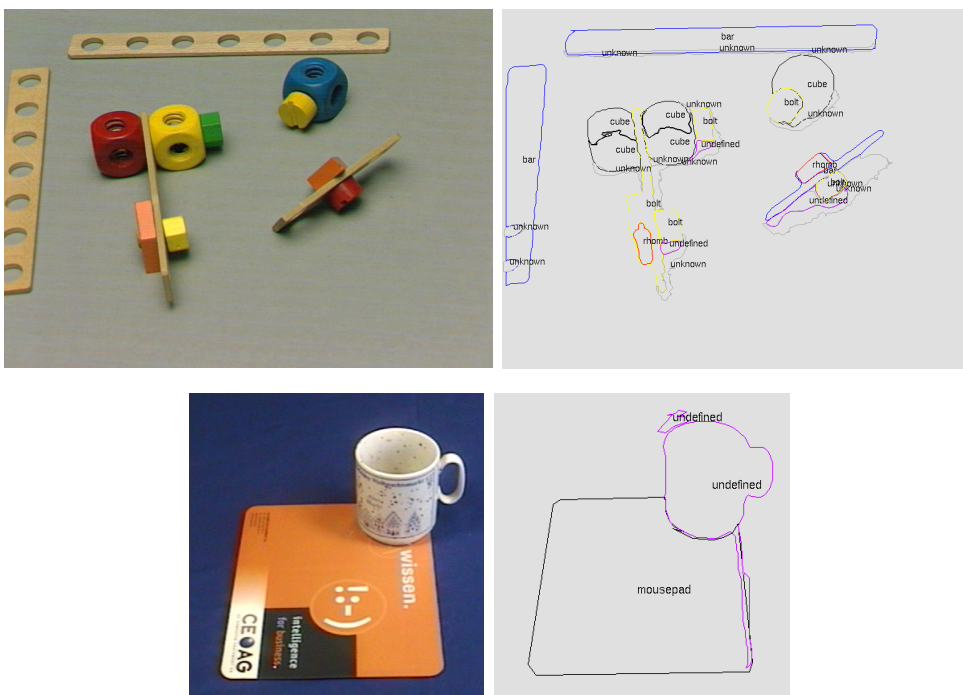


Figure 5.1: Example images for the two tasks of recognizing elements within baufix[®] assemblies and office environment objects.

5.1 Realized Systems and Evaluation Conditions

The integrating framework is applied for realizing systems addressing two different tasks. *baufix*[®] elements that are isolated or built into assemblies are the topic of the collaborative research center 360 at Bielefeld University, for details, see Appendix B. This implies that within the different projects of the center different aspects concerning these objects are addressed that can be exploited for recognition.

The second task concerns objects within an office environment that are commonly available and aware. Therefore, I chose this domain for demonstrating the facilities of the integrating framework concerning the adaption to different recognition tasks and conditions without significant changes. The task is defined to be categorization of the objects into object classes in contrast to the also justified task of individual exemplar based recognition. For details about this task, see also and C.

5.1.1 Components of the Realized Integrated Recognition Systems

For the task of recognizing *baufix*[®] parts several modules dealing with segmentation and recognition are available, as described in Chapter 2 and Chapter 3. They cover a wide range of processing strategies and, therefore, the integration of their results in order to improve them is promising. For addressing the office object categorization task the generally applicable segmentation approaches, presented in Chapter 2 and the shape based recognition module, described in Chapter 3, Sec. 3.2.4 are available. The office objects provide, in contrast to the *baufix*[®] objects, rich surface structure which complicates the identification of the object region.

Fig. 5.2 shows the available modules for addressing the tasks and their location within the general integrating architecture that is presented in the previous Chapter 4, Sec. 4.1.

For recognizing the *baufix*[®] elements segmentation processes based on the mean-shift algorithm, the pixel classification approach and the edge based contour grouping, see Sec. 2.2, are applied. For visual data based object labels the appearance based focus point classifier, the hybrid *baufix*[®] element detector, and the region based classifier combination, see Sec. 3.2, are used. Thereby, the two region based recognition modules are not used simultaneously, but two system constellations are realized, one combining the hybrid module with the appearance based approach and the segmentation modules and one integrating equivalently the region based classifier combination approach. The two different module constellations lead to different image data based information, whose integration and interpretation is compared to each other within the later evaluation. Besides the visual data based processes, additional information is integrated for extending and evaluating the preliminary recognition results for the *baufix*[®] task. This is delivered from the context based semantic region growing approach, the assembly recognition module and the construction process monitoring, see Sec. 3.3.

For recognizing the objects of the office environment, three data driven segmentation processes are integrated. These are the modules based on the mean-shift algorithm, the color structure code and the local variation approach, see Sec. 2.2. The shape based recognition process, see Sec. 3.2.4, delivers object labels based on the previously integrated segment information. Note this system does not contain an independent visual data based recognition module.

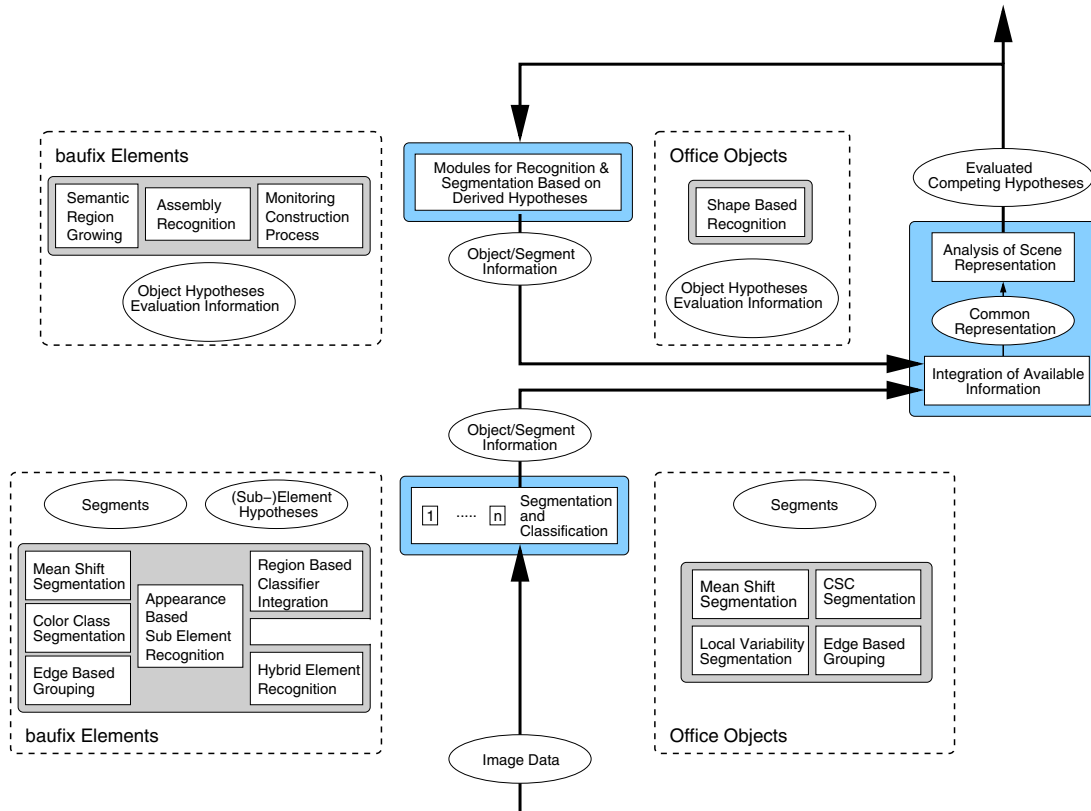


Figure 5.2: System components for recognizing **baufix**[®] elements and office objects and their location within the general integrating architecture presented in Sec. 4.1.

5.1.2 Test Sets and Evaluation Guidelines

The following quantitative evaluation is done based on image testsets as given in Appendix B and Appendix C. For the **baufix**[®] domain a set of 10 images containing 167 elements at all is used. 130 of the elements are built within 34 assemblies, resulting in 37 being isolated. The testset is designed from a naive user with the aim of being challenging for recognition. This results in complex assemblies causing occlusions and many examples of neighbored elements of same or similar color which complicates the segmentation task.

For the office domain a test set of 12 images containing 40 objects is used. This testset is also designed to show some aspects of the integrating system and account for the demands of the available modules, rather than to generate a representative sample. The objects are arranged to be close to each other, which mirrors the realistic situation and complicates the generation of suitable object regions. But the individual objects remain visible with regard to the applied global shape recognizer that is not very robust against shape deformations, as they occur due to occlusions. The system addressing the office domain shows what is possible based on the integration of few general purpose modules

without claiming to solve the object categorization task for general office objects.

Both testsets are rather small, because the following evaluation addresses various aspects of the integrating framework, instead of aiming at maximizing the resulting absolute recognition rate for both tasks. This requires much more effort but, therefore, allows insight into the integration step by identifying the reasons for success and also for failure of the approach. Identifying the reasons for failure is the first step of refining the system.

The small test sets generally cast doubt on the relevance of quantitative evaluation results. However, for evaluating the different aspects of the integration method not absolute rates but relative improvements constitute the core of the quantitative evaluation. The occurring differences are evaluated applying the method of determining their statistical significance, see Appendix D.

5.2 The Integration of Segment Information

Results from different segmentation processes are integrated within the hierarchical common representation, as presented in Sec. 4.2.2. By evaluating this integration step first, I conclude for the realized systems, whether the number of represented object regions is improved in comparison to the individual modules and what modules contribute to that improvement.

5.2.1 Evaluation Strategy

For evaluating the integration procedure for segmentation results the number of represented areas that correctly segments an object region is counted and compared with the appropriate results determined based on the individual modules. The evaluation is done manually without relying on a ground truth for the segmentation result. Instead of classifying the deviation from the correct result, the decision, whether an area represents the significant parts of an object region or not, is made by individual inspection of the areas. This definition is somewhat subjective, but it allows to differentiate. For example, if an area is in its whole some pixels too small, it will be counted to represent the object region correctly, while another is not, whose numerical deviation from the correct area may be similar, but that misses a part of the object region, for instance, due to a highlight.

A special situation occurs, if the object region is divided into several parts that are not neighbored to each other. This occurs in the *baufix*[®] domain, if the end of a thread is visible after a rhomb nut or cube is fastened. Such divided object regions are probably detectable by postprocessing matching of a detailed object model for the bolt in this example. Due to the fact that an appropriate module is not available for the chosen task, I take into account just the head of the bolt for the segment evaluation, while the segmentation and labeling of the thread is ignored. For the long green bolt, shown in Fig. 5.3, whose thread is visible behind the red cube, the segment for the head is sufficient for evaluating the object to be correctly segmented.

Based in this strategy for determining areas that correctly represent object regions, firstly the individual modules are evaluated. Fig. 5.3, left, shows exemplarily the region

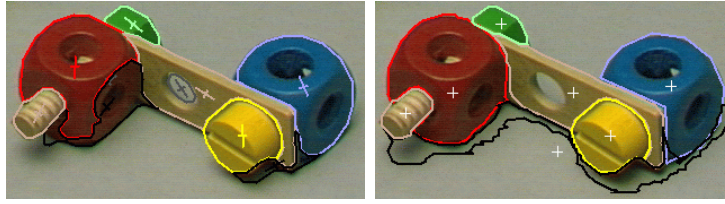


Figure 5.3: Segments originating from pixel color classification based segmentation (left) and the best fitting areas from the common hierarchical representation of several independently calculated segmentation results (right).

hypotheses originating from the segmentation approach based on the pixel color classification. Four of the five object regions are determined to be correctly segmented by this module, while there is no representation for the object region of the red cube. Minor differences, like for the yellow bolt in front are tolerated.

These individual quotes are compared to that determined for the common representation of several sources of segments. Fig. 5.3, right, shows those areas from the common representation that fit best to the object regions. It turns out that all object regions are represented well by at least one area of the common representation, which constitutes an improvement to the individual source concerned above.

Besides the total number of area correctly segmenting object regions that occurs within the hierarchy their origin is worth analyzing. For the example, shown in Fig. 5.3, both bolts and the bar are best segmented by the pixel color based segmentation process, as depicted at the left side. The red cube is correctly represented by an area originating from the mean-shift approach. For the blue cube, there exist several good candidates that are similar to each other. Although the area delivered by the pixel color classification approach is also denoted to be correct for the individual analysis, the best region representation originates from the contour based grouping. It is slightly better, because it covers also the lower corner of the cube. A special source of areas is the generation of union areas from partially overlapping ones during the construction of the common representation. It occurs within the evaluation of the integration scheme for segmentation results equivalent to the other sources of segment information.

Besides the origin of areas representing correctly segmented objects their location within the hierarchical representation is interesting with regard to the subsequent interpretation step. And, last but not least, besides analyzing some aspects of successful representation, also the reasons for failures are interesting. Those results probably show up possible strategies for system refinements.

The integration process resulting in the common hierarchical representation of segmentation results is influenced by the numerical setting of the threshold parameter t . This parameter controls the classification of the relation between each two areas based on their intersection. For determining the influence of the parameter value on the resulting representation and identifying an appropriate numerical setting, the integration process and the evaluation of its result is repeated with varying parameter values.

The strategy for evaluating the integration of independent segmentation results for the realized systems applied to the presented test sets is summarized in Tab. 5.1.

- Evaluate the independently generated segmentation results:
For each object within the testset, determine, which module delivers an appropriate area within its individual segmentation result.
- Evaluate the common integrated representation dependent on the construction parameter value t :
For each object within the testset, proof, whether an area is represented that correctly describes the object region.
If so,
 - denominate one or more sources that deliver areas representing best the true object region.
 - determine the layer of the hierarchical representation, where the best area is located at.
- If not,
 - determine, whether is the object region is over or under segmented.

Table 5.1: Strategy for evaluating the integration of independently generated segmentation results within the common hierarchical representation.

5.2.2 **baufix[®]** Task

For the **baufix[®]** task, I evaluate a test set of 167 elements. The segmentation results of the three segmentation approaches pixel color classification, mean-shift and contour based grouping are integrated with segment information given by the hybrid recognition process.

Tab. 5.2 gives the results for the individual modules delivering areas that correctly represent object regions. For each object several correct sources are possible.

	areas correctly representing the regions for 167 objects
pixel color	103 (61.7%)
hyb. rec.	99 (59.3%)
meanShift	44 (26.3%)
grouping	38 (22.8%)

Table 5.2: Evaluation of the individual modules delivering correctly segmented object regions for the **baufix[®]** task. For each object several correct sources are possible.

The results show that the pixel based classification approach delivers with 103 correct areas, which corresponds to 61.7% of the 167 objects, the best results. This does not surprise, if accounting for the degree of specification and adaption of this process to the task. The hybrid recognition process uses these segments as input for verifying and assigning segments to the neural holistic object hypotheses. The hybrid recognizer

generates additional segments by merging them based on object knowledge. The number of correct areas for this module is slightly lower than that for the raw segmentation approach. This is due to the newly generated segments are by their majority false, because the represented object knowledge is designed for isolated elements and falsely applied to mainly assembled elements here. The two segmentation approaches mean-shift and contour based grouping contribute just 26.3% and 22.8% correct areas. Both approaches are not specialized for the task, but broadly applicable and just adapted to the task by suitable parameter selections. As discussed above, they are, generally, not able to deliver one segments per object region due to shadows and highlights that introduce variance.

The individual segmentation results are the basic material for the integration process resulting in the hierarchical common representation. The following evaluations concerning the common representation are done for 5 different values for the construction parameter t varying it within a wide range from 0.7 to 0.9. The numerical setting for the parameter is discussed at the end of this subsection after evaluating its influence on the various aspects of the segment integration step.

Tab. 5.3 shows the number of represented areas that correctly describe object regions and their origins dependent on the integration parameter t .

	t=0.90	t=0.85	t=0.80	t=0.75	t=0.70
correct obj.	116 (69.5%)	120 (71.9%)	126 (75.4%)	127 (76.0%)	126 (75.4%)
best areas:					
pixel color	38 (32.8%)	44 (36.7%)	49 (38.9%)	50 (39.4%)	52 (41.3%)
hyb. rec.	42 (36.2%)	44 (36.7%)	48 (38.1%)	51 (40.2%)	57 (45.2%)
meanShift	20 (17.2%)	17 (14.2%)	19 (15.1%)	17 (13.4%)	18 (14.3%)
grouping	23 (19.8%)	26 (21.7%)	25 (19.8%)	29 (22.8%)	30 (23.8%)
part.Union	26 (22.4%)	25 (20.8%)	24 (19.0%)	21 (16.5%)	14 (11.1%)
part.Group	3 (2.6%)	3 (2.5%)	3 (2.4%)	3 (2.4%)	2 (1.6%)

Table 5.3: Evaluation of the integration of segmentation results for the **baufix**[®] task according to the origins of the areas best representing object regions and dependent on the integration threshold parameter t . Several good areas for one object originating from different sources are possible and appropriately counted for each source independently.

The main result from this evaluation is that the number of represented correct areas clearly exceeds the number of correct areas delivered from the individual modules shown in Tab. 5.2. All differences are statistically significant. Further, the evaluation of the sources for each of the best areas shows that all integrated individual modules contribute. The part of correct areas delivered from the pixel classification segmentation approach and the hybrid recognition process is highest, as to be expected from the individual evaluation above. But also the general purpose segmenters contribute to the improvement and so does also the special segment source, the process of generating union areas from partially overlapping ones that is executed during the constitution phase of the common representation. The grouping approaches occurs two times for

differentiating the hypotheses that are completely integrated to the representation and those that are exceptionally handled due to their partial overlap to other segments and the special characteristics of the grouping hypotheses, see Sec. 4.2.2.

As to be expected, the setting of the threshold t influences the result, but the changes are not statistically significant in spite of the wide range of tested parameter values. Nonetheless, a parameter setting of 0.8, 0.75, or 0.7 seems most promising for maximizing the number of represented correct areas.

Generally, lower values for the threshold parameter lead to less strict definitions of the area relations, i.e., the range of tolerated deviation from the ideal relation increases. Practically this leads to more independent, contained and similar and less partially overlapping areas. Does this have any effect on the quality of the areas represented within the hierarchy? The evaluation above allows a qualitative answer to this question by analyzing the part of object regions that are best represented by the union of originally partially overlapping areas. This part decreases with decreasing parameter, t , because union areas that are formerly denoted as representing the object region better than original ones are no longer generated. The sufficient, but suboptimal, original areas take their place and, thereby, decrease the quality of the area representation with decreasing parameter. From this point of view the value for the parameter should generally be set as high as possible, where the quantitative evaluation shows a remarkable effect just for the lowest values 0.75 and 0.7.

With regard to the subsequent rule based analysis of the hierarchical representation, as it is described in Sec. 4.3.2 and will be evaluated for the realized systems below, the location of the areas representing correctly segmented object regions with respect to the layers of the hierarchy is interesting.

	t=0.90	t=0.85	t=0.80	t=0.75	t=0.70
overall correct	116	120	126	127	126
layer 0	77	89	96	99	100
spec. groups related to layer 0	3	3	3	3	2
layer 1	36	28	26	23	21
layer 2	-	-	1	2	3

Table 5.4: Distribution of correctly represented object areas over the layers of the hierarchy for the *baufix*[®] task.

Tab. 5.4 shows that most object areas are located at the uppermost layer 0. This is due to the area representing the constant background and containing all the other areas is eliminated successful, as described in Sec. 4.2.2. The table shows further that also subordinated areas represent object regions. These results underline the motivation for applying an interpretation strategy that principally includes all layers for object area assumptions. However, the distribution of the areas representing objects justify the preference for the upper layers that is implemented within the analysis scheme. An area at the uppermost layer always is assumed to represent an object, while competing assumptions based on subordinated layers are connected to special situations. The partially overlapping groups that are handled exceptionally are all concerned with areas at the uppermost layer.

Concerning the threshold parameter setting, it is advantageous to decrease the parameter value, which leads to an increasing part of correct areas represented at layer 0, as the evaluation results show. Thereby, the effect is comparably small for the parameter range 0.8 to 0.7.

Finally, for discussing probable refinements of the integrated system the reasons for not correctly representing areas are interesting. Tab. 5.5 shows, whether these object regions are over or under segmented.

	t=0.90	t=0.85	t=0.80	t=0.75	t=0.70
overall	51	47	41	40	41
over segmented	2	4	5	5	5
under segmented	49	43	36	35	36

Table 5.5: Evaluation of reasons for not correctly represented areas for the *baufix*[®].

The main part of the not represented object regions are under segmented, i.e., the automatically determined area covering this regions additionally covers at least parts of neighbored objects. Under segmentation is caused, generally, by low contrast at the object region boundaries. This occurs for the *baufix*[®] task especially in case of neighbored, equally colored objects, where also very sensitive color segmentation processes get into trouble. The problem would be addressable by integrating additional knowledge about object model based segmentation. The over segmentation of object regions can be addressed by additionally integrating context based methods, like the semantic merging of neighbored areas. This aspect will be discussed later with view to the whole system evaluation results.

Concerning the dependence on the threshold parameter value, the number of under segmented object regions decrease, while the number of over segmented ones increase with decreasing parameter value t . This is caused by the decreasing number of union areas generated from partially overlapping originals with decreasing the parameter value. On the one hand, fewer unions lead to fewer under segmented object regions, if the missing union falsely merges several areas that actually represent independent objects. On the other hand, fewer unions lead also to more over segmented object regions if the overlapping areas are false assumed to be represent independent objects and the union area is missing. The increase of over segmented object regions is slightly, while the decrease of the more problematic under segmentations is remarkable for the lower parameter values 0.8, 0.75, and 0.7.

For finally defining a suitable numerical setting for the threshold parameter t for the realized system, the different aspects discussed above should be taken into account. Even, if the differences occurring in dependence on the parameter setting are not statistically significant due to the small testset, they are remarkable and considered for the choice. Maximizing the total number of represented correctly segmented object regions and minimizing the part of under segmentations for the not correctly represented areas leads to the parameter range 0.8 to 0.7. This range is also promising for enhancing the representation of correct areas at layer 0 of the hierarchy. However, the lower values probably decreases the quality of the represented correct areas by avoiding the construction of artificial union areas, as discussed above. Consequently I take the mediate value

of $t = 0.8$ for the further evaluation of the integrated system realized for the *baufix*[®] task.

The evaluations until now show that the general integrating framework for segmentation results is successfully applicable to the *baufix*[®] domain. In the following the integration of segmentation results for the office domain is evaluated in the same manner.

5.2.3 Office Environment

The integration of segmentation results for the office domain is evaluated following the same evaluation strategy as for the *baufix*[®] task. Although the smallness of the testset with containing just 40 objects does not allow to expect statistically significant results, it should either affirm or disprove the results for the *baufix*[®] domain.

Segmentation results for the office domain are delivered from four general purpose segmentation approaches namely the mean-shift, color structure code (csc), local variation and contour based grouping approach. Their individual results concerning object region segmentation are the topic of Tab. 5.6. For each object several sources of correct areas are principally possible.

	areas correctly representing the regions for 40 objects
meanShift	12 (30%)
csc	4 (10%)
localVar	2 (5%)
grouping	7 (18%)

Table 5.6: Evaluation of the individual modules delivering segmentation results for the office domain. For each object several sources of correct areas are principally possible.

Due to the generally structured surfaces of the office domain objects the individual segmentation approaches are mostly not able to deliver correctly segmented object regions within one segment. Best of them performs the mean-shift approach with 12 areas representing an object region, while the local variation approach delivers just two. Nonetheless, all modules are integrated and the results are evaluated dependent on the integration threshold parameter t , as shown in Tab. 5.7.

The main result of this evaluation is that by integrating the individual segmentation results into the common hierarchical representation the part of correctly segmented object regions increases dramatically. In comparison to the individual modules, where maximal 12 out of 40 object regions are correctly segmented within one segment, the integrated representation provides up to 30 correct areas.

The great increase of the number of areas representing uniquely an object region origins mainly from generating these areas by merging partially overlapping original ones during the construction of the common representation. As a consequence of the important role of merging partially overlapping areas the dependence of the results from the parameter t is higher than it is for the *baufix*[®] system, evaluated above. However, the confidence interval for the value 30 is with [24, 34] very wide due to the smallness

	t=0.90	t=0.85	t=0.80	t=0.75
correct objects	27 (68%)	30 (75%)	25 (63%)	22 (55%)
best areas from:				
meanShift	2	3	4	3
csc	2	3	4	3
localVar	1	2	2	2
grouping	2	2	2	2
part.Union	22	23	17	15
part.Group	3	4	4	4

Table 5.7: Evaluation of the integration of segmentation results for the office domain according to the origin of the object area and dependent on the integration parameter t .

of the testset. Thereby, it includes the results for the upper three of the overall four parameter settings. This affirms the conclusion taken for the *baufix*[®] domain that the parameter setting is not critical over a wide range of values.

With respect to the small number of uniquely segmented object regions delivered from the individual modules, the question arises, whether the results of all the four modules contribute to the integration success or at least one module could be left out without consequences. For evaluating this, I leave out one module after the other and repeat the integration step for the three remaining modules using the parameter value $t = 0.85$. Tab. 5.8 shows how many of the correctly represented areas remain and how many disappear, if one module is left out.

correctly represented object regions		
integrating 4 modules	30	
integrating 3 modules and leaving out		
meanShift	9	-21
csc	23	-7
localVar	21	-9
contour	26	-4

Table 5.8: Evaluation of the contribution of the individual modules to the success of the segment integration scheme for the office domain by leaving out each one module.

As to be expected from the previous results the mean-shift algorithm plays the dominant role within this integrated system. But also the other modules contribute to the good integration result to a higher degree than the evaluations concerning the individual segment sources, see Tab. 5.6, and the sources of best integrated areas, see Tab. 5.7, allow to expect.

Concerning the distribution of the correctly represented areas over the layers of the hierarchy, Tab. 5.9 shows a strong concentration within the uppermost layer 0 that re-

sults from the background elimination, see Sec. 4.2.2. Nonetheless, also other layers are involved for storing areas that represent object regions, which confirms the appropriate result stated for the *baufix*[®] domain.

	t=0.90	t=0.85	t=0.80	t=0.75
overall correct	27	30	25	22
layer 0	21	24	17	15
spec. groups related to layer 0	3	4	4	4
layer 1	3	2	3	2
layer 2	-	-	1	1

Table 5.9: Distribution of correctly represented object areas over the layers of the hierarchy for the office domain.

As the final step in evaluating the common representation of segmentation results, I address the reasons causing not correctly represented object regions, as shown in Tab. 5.10.

	t=0.90	t=0.85	t=0.80	t=0.75
overall	13	10	15	18
over segmented	4	4	9	13
under segmented	3	0	0	0
false at all	6	6	6	5

Table 5.10: Evaluation of reasons causing not correctly represented object regions for the office domain.

Besides the two already known categories of over and under segmented object regions within this domain also completely false segments occur. Bad contrasts at the object borders coexistent with high contrast within the object surface lead to areas covering parts of several objects that are denominated to be false at all. Besides these, the evaluation shows few under segmented, but many over segmented areas that occur due to inner structures of the object surfaces. The number of over segmented areas increases with decreasing the integration threshold t . With a lower value for t , greater overlaps between independent areas are tolerated, while for higher values the areas would be classified to be partially overlapping causing the generation of an union area. A too strict definition of the allowed overlap leads to under segmentations for $t = 0.9$.

Having in mind these evaluations for selecting the numerical setting for the parameter t for this system results in the value of 0.85. This value leads to the maximum of correctly represented object regions, even if the maximum is not statistically significant. The other evaluations concerning the reasons for not correctly represented areas and concerning the distribution of the correct areas over the hierarchy layers supports this decision.

5.2.4 Conclusions from Integrating Segment Information

The general integrating framework for segmentation results that results in a common hierarchical representation is applied to two different domains. The appropriate evaluations show comparable results with the key note that the integrated common representation improves clearly the participating individual results. For both domains four available segmentation approaches are integrated and it turns out that they all contribute to the successful integration. The integration process itself generates additional segment hypotheses by merging areas that originally overlap each other partially due to uncertainties for the boundary. The generated areas constitute a remarkable fraction of those that represent correctly segmented object regions.

The integration result is influenced by the threshold parameter t that determines the classification of the relation between each two areas constituting the basic structure of the common representation. The evaluation of both systems concerning the two different tasks show the comfortable result that the numerical setting of this parameter is not critical over a wide range of possible values. The two selected values that are fixed for the following evaluations are with $t = 0.8$ for the *baufix*[®] system and $t = 0.85$ for the office object system similar to each other.

With its structured representation of segment information that is delivered from several independent sources the common hierarchical representation provides best prerequisites for a successful subsequent object knowledge integration and interpretation, which is the topic of the following section.

5.3 Independent Image Data Based Object Information

The hierarchical representation of segmentation results is extended by independently generated image data based object information. This kind of object information is treated here, while the later integration of additional higher level object knowledge is discussed in subsequent sections.

Looking at the two realized systems as depicted in Fig. 5.2, on page 115, shows that for the office domain no independent object recognition module is integrated. Therefore, the rest of this section deals with the *baufix*[®] system that integrates totally three data based recognition modules within two different configurations.

The independent recognition modules are evaluated individually, before the object label integration is addressed.

5.3.1 Individual Evaluation of Object Information for the *baufix*[®] Task

The available image data based object recognition modules are evaluated independent from each other based on the common testset. These results define the baseline for the integrated system. The individual object recognition modules deliver their results within different object label alphabets. All the occurring object labels for the *baufix*[®] task are defined in the Appendix B, Tab. B.1. In order to ensure the comparability between the individual results and the integrated result the evaluations are done based on the common general object label alphabet, as defined in Tab. B.1, column 1. Because the

focus here lies on the evaluation of the integration method, the recognition rates of the individual modules are not optimized, for example, by parameter tuning or by ensuring optimal lighting conditions.

Hybrid baufix[®] Element Recognizer The hybrid recognition module for baufix[®] elements is described in detail in Sec. 3.2.1. It achieves very high recognition rates with about 90% correct classification and segmentation for isolated, non occluded elements, as shown in [Kumm 98]. Because the results of this module contain object label as well as object segment information, both parts are considered for the evaluation here. The object classes delivered from the hybrid module are baufix[®] elements, as given in the Appendix B, Tab. B.1, column 2. As discussed above, for ensuring comparability, an object label is counted to be correct, if it is correct within the general label alphabet. This implies, for instance, that an exchange between a three hole bar and a five hole bar that are both summarized by the general element 'bar' does not have an effect for the evaluation result.

The definition of a correctly segmented object region corresponds to that applied for the evaluation of the segmentation results, given in Sec. 5.2.1. An area is manually classified to represent an object region correctly if it covers the significant parts of the true object region. Partially correct results occur, if the object label is correct, but the accompanying segment does not correctly represent the object region.

The hybrid recognition module is applied to the current testset containing 167 objects with 37 isolated and 130 assembled elements. Results are shown in Tab. 5.11.

# elements to recognize	167
correct	96 (57.5%)
from that	
isolated elements	31 (84% of 37)
assembled elements	65 (50% of 130)
partially correct	30 (18.0%)
false	41 (24.6%)

Table 5.11: Individual evaluation of the hybrid recognition module for baufix[®] elements on the common testset, considering object classification and segmentation. Results containing a correct label but a non completely matching segment are denominated to be partially correct.

The recognition rate for isolated objects is with 84% for the general element set in the same order of magnitude as the rates for the element set given in the original paper [Kumm 98] with 89.4% up to 93.5% dependent on the image down sampling rate that is not applied here.

The recognition rate for assembled elements decreases to 50%, resulting in totally 57,5% correct results for the testset. Further 30 object hypotheses, which corresponds to 18%, provide at least a correct object label, even if the segmentation fails. Problems in segmentation of the assembled elements cause the majority of failures. From the 71 just partially correct and false results, there are 68 false due to false segment hypotheses,

while just in three cases the classification of a correct segment fails. However, this does not imply that a correct segmentation guarantees a high recognition rate. The algorithm relies strongly on global element features that change in case of element occlusions leading to principal problems in recognizing assembled elements with this approach.

Region Based Classifier Combination A recognition module for *baufix*[®] parts following the approach of combining classifiers based on isolated region parameters is described in Sec.3.2.2. The recognition module used here combines classifiers based on the four region based features: 'size', 'compactness', 'eccentricity', and '*baufix*[®] color class'. For purposes of simplicity a smaller set of features than within the original work is used. For a given region the label is determined that carries the maximal confidence value after combining the individual classifier. The region is appropriately classified, if at least three of the four individual classifiers support this label with a confidence value of at least 0.005. Otherwise the region is not classified for avoiding false labeling.

The object label alphabet delivered from the module is given in Appendix B, Tab. B.1, column 3. The labels differ from the *baufix*[®] elements mainly by the fact that bolts are decomposed into their head and their thread. Thereby, the 167 elements of the current testset become 186 object regions to be labeled by the classifier combination scheme.

The evaluation is, firstly, based on the 186 object regions. An object label is again counted to be correct, if it is correct within the general label alphabet. The definition of a correctly segmented object region again corresponds to that applied for the evaluation of the segmentation results, given in Sec. 5.2.1. Partially correct results occur, if the object label is correct, but the accompanying segment does not correctly represent the corresponding object region.

Tab. 5.12 shows the results for the combination of region based classifiers evaluated individually for the 186 segments of the testset, which are not simply comparable to the results generated by the hybrid module above.

# segments to classify	186	
correct	112	(60.2%)
from that		
isolated elements	38	(84% of 45)
assembled elements	74	(52% of 141)
partially correct	23	(12.4%)
false	43	(23.1%)
not classified	8	(4.3%)

Table 5.12: Results of the region based classifier combination module for the *baufix*[®] task evaluated individually for each segment.

The recognition rate for the isolated elements is with 84% for the general element set comparable to the rate of 76% for the detailed class system given in the original work [Menz 99]. This result is achieved in spite of the the reduced feature set used here.

Taking into account the assembled elements, the recognition rate decreases to 60%, which does not surprise due to the transfer of the system from mostly isolated to mostly

assembled elements without any adaptations of the classifiers. At this point, I remind to the issue of this evaluation, where results for the individual modules serve as baseline for comparisons but the absolute values are not within the focus.

For getting a result that is better comparable with the hybrid module and the final integrated system the results of the region based classifier combination are rearranged with respect to the *baufix*[®] element object regions. Therefore, I distinguish between bolts, where heads and threads are represented by neighbored but separated segments and the other *baufix*[®] elements. The latter also include the heads of bolts, where the thread is not visible, as it occurs mostly in assemblies, or a part of the thread is occluded, while the separated rest remains visible. Tab. 5.13 presents the *baufix*[®] element based evaluation of the region based classifier combination module. The categories for evaluating the separated bolts are adapted by denominating a result correct, if both parts are correct, false, if both parts are false and partially correct otherwise.

	#bolts with neighbored head and thread	# others
# elements	11	156
correct	8	85
partially correct	3	21
false	-	42
not classified	-	8

Table 5.13: Results of the region based classifier combination for the *baufix*[®] task evaluated with respect to *baufix*[®] elements.

The process delivers for the existent 11 bolts with visible threads in 8 cases the best principally achievable result, namely the correct segmentation and labeling for the two independent parts. For a completely correct object segmentation result this information has to be extended with additional information, for example, the context based approach of semantic region growing, see Sec. 3.3.1.

Completely correct results are delivered for 85 elements which results related to the 167 presented elements in a recognition rate of 50.9%. With respect to the principal problems concerning the divided object regions for bolts it is justified to leave them out and relate the correct results to the 156 principally possible correct results which leads to a rate of 54.5%. The 85 correct results are generated based on 103 correctly segmented object regions delivered from the pixel classification based segmentation process that has been analyzed before within the evaluation of the integration scheme for segmentation results, see Sec. 5.2. Based on the correctly segmented object regions the multiple classifier combination achieves with its 85 correct results a rate of 83%. Together with the fraction of partially correct results this shows that failures within the segmentation results causes the majority of failures for the recognition module. This result is equivalently stated for the hybrid recognition module. It is also equivalent that it is not sufficient to deliver correct segments for achieving high recognition rates. The classification step depends on global segment parameters and is, even after proceeding an appropriate training step, likely to produce more failures due the higher variability of segment parameters occurring for elements within assemblies. The classification power

of the multiple classifier combination module is generally lower than that of the hybrid recognition module due to its more general approach.

Appearance Based *baufix*[®] Sub Element Recognizer The appearance based recognition module is described in detail in Sec. 3.2.3. The process is adapted to the *baufix*[®] task by parameterizing it in order to detect characteristic parts of elements, like the holes. Interesting points, detected based on general mechanisms exploiting characteristics in symmetry and color, focus an artificial neural network classifier system to interesting image areas. The classification bases on image information provided by the surrounding of the focus point and delivers sub elementary class labels, as given in Appendix B, Tab. B.1, column 4. The classified focus points do not deliver information about object regions and generally more than one are located at one *baufix*[®] element. They are firstly evaluated point by point, before a manual summary of those points concerned with one elements is done below.

For the individual point evaluation those points that are located at objects and those located besides objects are distinguished. This aspect allows to evaluate the quality of focusing. A focus point is denominated to be correctly classified, if it is located at an object and carries the correct object label within the general *baufix*[®] element class, given in Tab. B.1, column 1, for reasons of comparability with other evaluations. Its also correct, if it is located beside an object and carries the label 'undefined'. The results generated for the current *baufix*[®] testset are summarized in Tab. 5.14.

	total	correct	false
located at objects	459	332	127
located besides objects	14	10	4
	473	342 (72.3%)	131 (27.7%)

Table 5.14: Evaluation of 473 appearance based classified focus points occurring within the *baufix*[®] test set.

The initial focusing step works well by locating just very few points besides objects. Concerning the focus point classification the rate of correct class labels is with 72.3% statistically significant lower than the rate given with about 85% within the original work [Heid 98], although the evaluation here is done based on the smaller general element class set. This is a result of porting the original system to a different experimental setup that provides especially changed lighting conditions. Better results are expectable after a suitable training phase that is not in the focus here.

The number of generated focus points clearly exceeds the number of elements for the test set caused by several focus points located at one element in mean. The appropriate focus points have to be summarized and their classification to be integrated in order to get a recognition result for an element, as desired for the defined *baufix*[®] task. For evaluating the element recognition abilities of the classified focus points, I summarize the focus points concerned with one element manually and integrate the object information using the probabilistic integration that is presented in Sec. 4.3.1 and applied within the integrated recognition system. The probabilistic integration accounts

for the classification error matrix of this classifier, as given in Appendix B, Sec. B.3, for determining confidence values for the class labels. Tab. 5.15 presents the results for the 167 objects of the testset.

	total	all labels agree	labels are different
correct	126 (75.4%)	88	38
false	35 (21.0%)	20	15
elements not focussed	6 (3.6%)		

Table 5.15: Evaluation of probabilistically integrating the object information given by classified focus points that are manually assigned to each one **baufix**[®] element.

The manual summary of focus points concerning one **baufix**[®] element and the probabilistic integration of their object information leads for 126 of the 167 elements to the object label that is correct within the general element label set, which corresponds to a recognition rate of 75.4%. Thereby, for 88 objects the classified points all agree for the appropriate correct label, while for 38 objects, there exist false labels that are correctly suppressed by the integration process. For further 15 elements, there exist correct object information that is outvoted from false labels, but that supports the correct label if it is integrated with object information from other sources. Concerning the focusing step, it turns out that 6 elements are not focussed at all, which is a noticeable but not crucial part.

The evaluation shows that the probabilistic integration of object labels carried by suitably summarized focus points is an appropriate method for **baufix**[®] element recognition. For additionally determining object regions, the object information has to be related to segment information within the integrating module. The integration of object information is the topic of the following subsection.

5.3.2 Evaluating the Object Label Integration for the **baufix**[®] Task

The object information delivered by the independent object recognition modules has to be integrated with the hierarchical representation of segmentation results, as described in Sec. 4.2.3.

The integration process distinguishes region based labels that are already associated with segments and point based labels that have to be assigned to available external segment information. Region based object information, as delivered for this task from the hybrid recognition approach and the region based classifier combination, is integrated by attaching it to the appropriate area within the hierarchical representation. An appropriate area, thereby, constitutes the represented segment itself, a similar one that represents the set, or the union area that represents a set of partially overlapping areas. The assignment is unique and therefore the process of region based object knowledge needs no further evaluation.

Point based object information has to be related to the areas that are concerned with the image at the key point. Uncertainties arise for points that are located near area

borders. This aspect is addressed by introducing a distance parameter whose effect has to be evaluated here.

Integration of Point Based Object Information The object information given by the classified focus points for the *baufix*[®] task has to be related to segment information in order to exploit it for object segmentation and recognition. An object label carried by a focus point is, firstly, related to all areas that cover the key point. Thereby, it is not guaranteed that all points can be related to segment information, because not each image point is covered by an represented area due to the approximation of the segment boundaries by linearly connected support points and the elimination of background areas. Additionally, for a point that is located insight, but near the border of an area, a significant part of the image information that was used for the classification is not contained within the area. The question arises, whether the object label accompanying this point interprets reliably the area the point is located in.

This problem is addressed by introducing a minimal distance parameter d in the process of integrating point based object information, as discussed in Sec. 4.2.3. A focus point and its object label are related just to those areas within the hierarchy that provide a minimal distance of d pixels between their boundary and the focus point. The process of integrating the classified focus point information into the common hierarchy of segmentation results is repeatedly evaluated for several distance parameter settings based on the current *baufix*[®] testset. The dependence of the number of related point based object labels on the distance parameter is shown in Tab. 5.16.

d	left out	localized on objects	localized on shadows
0	6 (1.3%)	455 (96.2%)	12 (2.5%)
4	40 (8.5%)	424 (89.7%)	9 (1.9%)
8	72 (15.2%)	392 (82.9%)	9 (1.9%)

Table 5.16: Evaluation of the integration of the 473 classified focus points for the *baufix*[®] testset with segment information dependent on the required minimal distance between point and boundary of the attached area.

As expected, the number of focus points that can not be attached to an area increases with increasing the required distance between point and area boundary. The differences are statistically significant. Without applying a distance parameter, $d = 0$, 6 object labels are not related due to the approximation of the segment boundary and the elimination of background areas. With increasing minimal distance, nearly all of the suppressed points are formerly attached to areas representing object regions. Generally, it is desirable to leave out false information that would disturb the interpretation process, but it is not desirable to leave out correct information that would support the recognition process. Tab. 5.17 shows the fractions of remaining and left out points that are correctly and falsely classified.

The variation of the minimal distance parameter does lead to just small differences that are not statistically significant for the fractions of correctly classified from the totally attached point based object labels. This implies that not significantly more false labels are

d	left out			localized on objects			localized on shadows		
	all	correct	false	all	correct	false	all	undef	false
0	6	3	3	455	331 (72.7%)	124 (27.3%)	12	8	4
4	40	31	9	424	305 (71.9%)	119 (28.1%)	9	6	3
8	72	45	27	392	291 (74.2%)	101 (25.8%)	9	6	3

Table 5.17: Evaluating the effect of the minimal distance parameter for integrating correctly and falsely classified focus points.

left out at the area borders than correct ones. Nonetheless, there is a slight improvement for $d = 8$ that seems to justify the appliance of the parameter. For confirming this suggestion, the evaluation is complemented by considering the manual summary of object labels concerning each one *baufix*[®] element. Tab. 5.18 shows the dependence of the *baufix*[®] element based integration on the distance parameter.

d	correct	false	no information for element
0	126 (75.4%)	35 (21.0%)	6 (3.6%)
4	124 (74.3%)	32 (19.2%)	11 (6.6%)
8	116 (69.5%)	33 (19.8%)	18 (10.8%)

Table 5.18: Evaluating the effect of the minimal distance parameter on the *baufix*[®] element based recognition rate.

The increasing parameter value does not improve the element based recognition results statistically significant. However, the results show a development. With increasing distance parameter the number of correctly classified objects decreases and the number of objects that get no object information increases, while the number of false classification remains nearly stable. Demanding a minimal distance is, therefore, counterproductive for the integration of the point based object information provided by the appearance based *baufix*[®] sub element recognition module. The numerical setting for the distance parameter is $d = 0$ for the following evaluations concerning the interpretation of the common hierarchical representation.

5.4 Rule Based Analysis of the Common Representation

The common hierarchical representation contains the available segmentation and object information. The amount of information is structured by establishing relations between those segments and object labels that should probably be taken into account collectively for locally solving the object segmentation and recognition task. Hypotheses for individual objects are generated by analyzing the representation applying a recursive rule based analysis procedure as presented in Sec. 4.3.2. The interpretation procedure is evaluated in the following by considering the realized integrated system addressing the *baufix*[®] and the office task.

The interpretation step has to identify those areas within the hierarchy that represent object regions and assign an object label to it which is complicated and lead to uncertainties due to individual failures within the segment and object information. In the presence of those uncertainties the analysis is allowed to generate several competing object segment and label hypotheses. The competing results are further open for the integration of additional information from higher level knowledge, like provided from context based modules.

The strategy for evaluating the interpretation step is to determine, whether it is able to identify an area that correctly represents an object region under the precondition that an appropriate area is represented within the hierarchy. For each correctly selected area, the procedure of integrating the available object information in order to generate the final label for the area is evaluated. In order to show, whether all rules are necessary and contribute, it is analyzed, which of the rules has been applied for correctly selecting areas.

5.4.1 General Rules

The interpretation step works recursively starting with the uppermost layer of the hierarchy. For each area, the generation of object region and label hypotheses is done based on the set of general rules that are described in Sec. 4.3.2, Tab. 4.4, Tab. 4.5. The applicance of the rules to each area is influenced by hypotheses concerning contained areas that are recursively calculated and by the related object information that is integrated, as described in Sec. 4.3.1.

The interpretation step is evaluated, firstly for the *baufix*[®] task. The participating modules for this task are depicted in Fig. 5.2, on page 115. Based on the available modules two system constellations for the *baufix*[®] task are constructed, that differ by involving either the hybrid object recognition module or the region based classifier combination, while the other modules remain without changes. The two module constellations results in differences concerning the available data based object information.

The results of generating hypotheses for object regions and labels based on the common representation of data based segment and object information shows Tab. 5.19 for both module constellations applied to the current *baufix*[®] test set.

	using hybrid element recognizer	using region based classifier combination
correct object regions resulting in	126 (75.4%)	115 (68.9%)
correct recognition results	108 (85.7%)	97 (84.3%)
false recognition results	18 (14.3%)	18 (15.7%)
false object regions	41 (24.6%)	52 (31.1%)

Table 5.19: Evaluation of the abilities of the rule based interpretation step in identifying areas from the hierarchy that correctly represent object regions and assigning correct integrated object labels for the two *baufix*[®] system constellations.

The system using the hybrid recognizer delivers the better result with 126 of 167 elements represented, see also Tab. 5.3, and 108 of them also selected and classified correctly. Exchanging the hybrid recognition with the region based classifier combination module leads to changes within the available information and consequently within the common representation. The integration of the available segment information leads to just 115 correctly represented object regions. The decrease is due to the missing additional segment information from the hybrid recognition approach. Using the region based classifier combination module instead of the hybrid recognizer implies that objects consisting of two differently colored regions, which are the bolts with their head and thread, cause a principal problem. They can not be recognized correctly without integrating additional object knowledge, as for example the semantic region merging does (see Sec. 3.3.1).

The main result of Tab. 5.19 is that the rule based interpretation step is able to generate correct object segmentation and recognition results for about 85% of the correctly represented areas for both systems. Tab. 5.20 shows a more detailed evaluation of the reasons leading to either the correct or the false results regarding the two aspects of the analysis, namely the integration of available object information for a given area and the selection of appropriate areas that probably represent object regions.

	using hybrid element recognizer	using region based classifier combination
correct recognition results	108	97
object label integration:		
overall correct labels	72	54
partially false object information	36	43
false recognition results	18	18
object label integration:		
partially false object information	7	8
overall false object information	11	10
no object information	-	-
area selection:		
correctly selected area	18	18
correct area missed	-	-

Table 5.20: Detailed analysis of the reasons leading to correct or false recognition for the two **baufix**[®] system constellations.

Concerning the object label integration the results show that even if the great majority of the correct results provide non contradictory object labels a remarkable fraction of them is cause by correctly integrating contradictory object information. For the objects where the integration step delivers false results this is mostly due to completely false object information.

The interpretation step is able to identify the area within the hierarchy that correctly represents the object region for both system constellations. The area selection process is determined by the applied set of general rules, that reduce the complexity of the system

by selecting promising candidates and suppressing others. The set of rules must be, on the one hand, as comprehensive as necessary for selecting the best represented result, but, on the other hand, it should be as small as possible in order to keep the system simple and avoid the generation of too many superfluous competing hypotheses. Tab. 5.20 shows that all the areas correctly representing an object region within the hierarchy are identified. This result shows that the set of just 4 rules seems to be comprehensive enough to reach all desired areas within the hierarchy. For addressing the question, if, the other way around, there may be rules that do not contribute, Tab. 5.21 contains the number of appliance of the individual rules for selecting the correct areas.

rule	#applications	
	using hybrid element recognizer	using region based classifier combination
1	93 (73.8%)	93 (80.9%)
2	20 (15.9%)	10 (8.7%)
3	10 (7.9%)	9 (7.8%)
G	3 (2.4%)	3 (2.6%)
	126	115

Table 5.21: Evaluation of the appliance of rules, see Tab. 4.4, on page 97, Tab. 4.5, on page 101, for generating correct object region hypotheses from the common representation for the two **baufix**[®] system constellations.

Rule 1 is most important, as expected from the fact that most of the areas correctly representing object regions are located within the uppermost layer of the hierarchy, as already stated in Tab. 5.4, on page 120, for the **baufix**[®] system using the hybrid recognition module. The difference between the number of correct areas located at the uppermost layer (96) and the number of successful applications of rule 1 for this system constellation (93) is caused by the areas at the uppermost layer that embed further individual objects. This constellation is addressed by rule 3. Those correct areas that are located at the subordinated layers are all identified following the rules 2 and 3, respectively. Although their contribution is much smaller than that of rule 1 the evaluation shows their justification.

Rule G is applied for improving already identified areas by grouping hypotheses. For the evaluated **baufix**[®] testset this situation occurs rather seldomly, as already stated for the **baufix**[®] system using the hybrid recognition module in Tab. 5.3, on page 119, and, consequently, this rule contributes less to the final result here.

The recognition system realized for the office domain works without integrating an independent visual data based object recognition module. Consequently, for the analysis of the common representation rule 3 that accounts for disagreements within the available object information is not applied. Tab. 5.22 shows, that rule 1 is the most important, while rule 2 does also not contribute to the selection of areas that correctly represent object regions for the office domain system applied to the testset of 40 objects.

Comparing this results with the distribution of areas correctly representing object regions over the layers of the hierarchical representation, depicted in Tab 5.9, on page 124, shows that the 24 correct areas at the uppermost layers are selected by rule 1,

rule	#applications
1	24
2	-
3	-
G	4
	28

Table 5.22: Appliance of rules, see Tab. 4.4, on page 97, Tab. 4.5, on page 101, for generating hypotheses from the hierarchical representation of segmentation results for the office domain.

while 2 correct areas located at the subordinated layer 1 can not be identified due to the missing object information. The 4 grouping hypotheses are correctly selected by the appropriate rule.

5.4.2 Improvements by Integrating Data Based Modules

Several individual image data based segmentation and recognition modules are integrated within the realized system, as presented above, finally, for improving the recognition rate of the individual modules by the integrated system.

For the office domain no independent recognition approach is available that serves for the comparison, but for the *baufix*[®] task, there are those modules.

The hybrid *baufix*[®] element recognition approach achieves applied to the current test set containing 167 elements, a rate of 57.5% correctly segmented and labeled objects (96 of 167, see Tab. 5.11, on page 126). Integrating this module with two further color segmentation modules and the appearance based sub element recognition approach, improves the part of correct results that occur within the competing hypotheses to 64.7% (108 of 167 objects, see Tab 5.19, on page 133). This improvement is remarkable, especially taking into account that just one additional source of object information causes it. The difference is just at the border to be statistically significant due to the small test set.

The second system addressing the *baufix*[®] task exchanges the specialized hybrid recognition module with the more general region based classifier combination approach. The individual module delivers a recognition rate of 50.9% (85 of 167 elements, see Tab. 5.13, on page 128). The integrated system achieves 58.1% (97 of 167 elements, see Tab. 5.19, on page 133) of correctly recognized objects. Also this improvement is remarkable and at the border to be statistically significant.

A similar comparison to the appearance based recognition approach is not possible, due to the missing segment information accompanying the object labels delivered from this module.

The comparisons show that the integration procedure is, in fact, able to benefit from the information delivered from several independently processing modules for improving their recognition results.

5.4.3 Competing Object Hypotheses

The preceding subsections deal with the process for extracting object hypotheses from the common representation of segment and object information within the realized systems for recognizing *baufix*[®] elements and office objects. The recursive, rule based interpretation process applied to the common representation delivers, generally, several competing hypotheses due to uncertainties caused by missing and false individual recognition and segmentation results. The competing hypotheses are probably extended with additional information for refinement and clarification. The number of preliminary competing results generated by integrating the individual data based modules depends on many aspects including the number of elements per cluster, their topology and the amount of occurring disagreements.

Within the *baufix*[®] test set the number of competing hypotheses generated for one cluster of objects reaches from one to several hundreds, even if the number of competing hypotheses is restricted by the rule based interpretation step and is far away from the number of principally possible competing hypotheses. Nonetheless, the effort for the subsequent integration of additional information increases with increasing numbers of competing hypotheses, because each competing hypothesis has to be handled independently from the others. From this point of view the question arises, whether their number can be further restricted by applying domain specific knowledge that is more restrictive than the general rules. This aspect is discussed exemplarily in the next subsection.

5.4.4 Discussing Domain Specific Restrictions for the *baufix*[®] Task

A domain specific restriction has to ensure that not the correct or the nearly correct hypothesis is disregarded. Thereby, neglecting hypotheses is a non returnable decision, in contrast to the evaluation of competing hypotheses discussed in the following section.

The fact that the correct recognition result is, of course, not known, leads to the strategy of comparing competing hypotheses and disregarding those that seem significantly less probable than others. Promising criteria for the *baufix*[®] task are the color rules of the elements, excluding, for example, blue rhombs, or rules about the visibility of one object located behind another based on the relative object sizes. For example, a cube is a large element that can not be embedded by other *baufix*[®] elements. Fig. 5.4 shows an example, where the number of competing hypotheses is reduced to be half of the original number, if the domain specific restriction is applied.

For the example assembly, consisting of eleven elements, six elements provide two competing hypotheses each. The combinatorial combination results in finally 64 competing hypotheses generated for the assembly. For the left cube a hypothesis containing a cube embedded by a felly is generated, a constellation that is not possible to create with *baufix*[®] parts, in contrast to the competing hypothesis containing just the cube. Disregarding the less probable competing hypothesis, results in 5 instead of 6 elements providing two competing hypotheses, which leads to 32 hypotheses at all.

The reduced number of competing hypotheses generally accelerates the further processes applying additional knowledge based on the preliminary results. The restriction, as applied above, does not disregard the the correct recognition result, if it is already

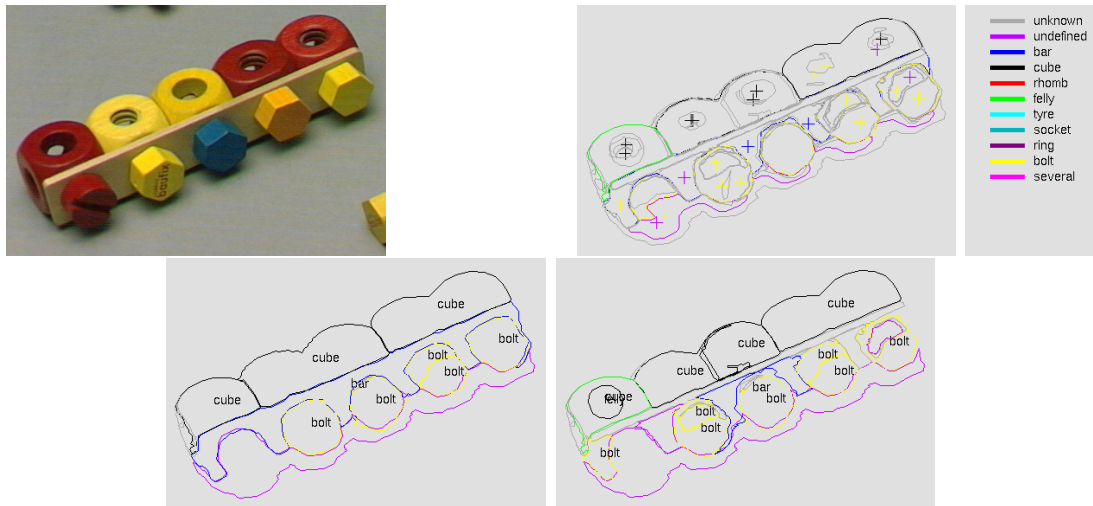


Figure 5.4: Integrated information for an example assembly results after applying the general rules in 2 competing hypotheses for 6 elements. Combinatorics delivers 64 competing hypotheses at all.

included within the preliminary result. But problems arise, if this requirement is not fulfilled, as shown exemplary in Fig. 5.5.

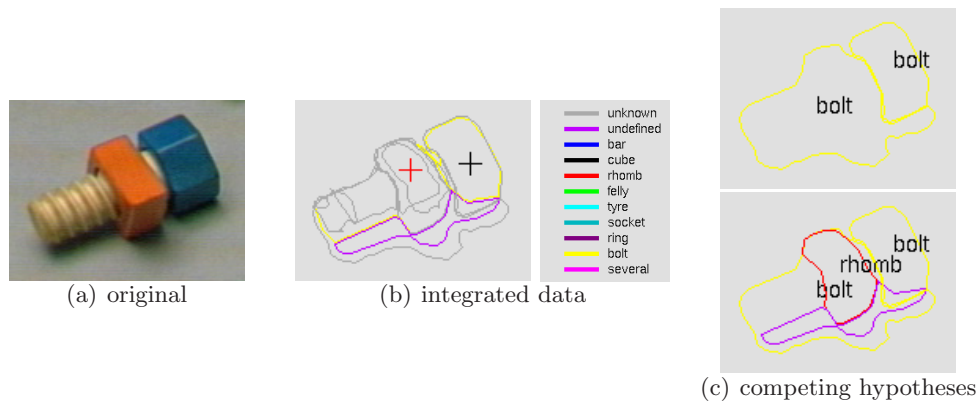


Figure 5.5: Assembly with integrated segmentation and object information and the two competing hypotheses resulting from applying the general rules. The second hypothesis contains the better result but contradicts the **baufix**[®] task object constellation constraints.

The two competing hypotheses differ in the interpretation of the rhomb and the neighbored thread. The area summarizing the rhomb and the thread is hypothesized to represent a bolt, see Fig. 5.5(c), upper part. A competing hypothesis is generated based on the disagreement within the object information and assumes the contained area to represent an individual rhomb that is embedded by a bolt, see Fig. 5.5(c), lower

part. The latter hypothesis contains the better result, but contradicts the *baufix*[®] element constellation rule in contrast to the first one.

In summary, domain specific knowledge may be applied for significantly restricting the number of preliminary competing object hypotheses and, thereby, accelerating further processing.

However, making the hard decision of neglecting competing hypotheses may be prematurely, if the correct hypothesis that, of course, fulfills all domain specific expectations, is not represented. Therefore, if it is compatible with probable computational time constraints for the system, definite decisions at this early point of processing, based on probably incomplete information are avoided. Instead, appropriate domain specific evaluation criteria for the whole set of competing areas may be applied and accounted for by selecting the sequence of hypotheses for further processing.

The competing object hypotheses constitute the basis for integrating additional information. This aspect is the topic of the following section.

5.5 Additional Knowledge Integration

The proposed integrating framework claims the ability of taking into account different kinds of object information delivered from different kinds of recognition modules for generating its recognition result, see Sec. 4.4. There are, on the one hand, the object recognition modules generating their hypotheses based on the image data, like presented in the previous section. On the other hand, there are modules that exploit knowledge about detailed individual object models, object context models, or relations to other modalities like speech and gesture recognition. Examples are the semantic region growing approach and the *baufix*[®] assembly recognition module that base on object segmentation and label hypotheses and extend these hypotheses by additional information. The integrated system architecture, proposes the equitable integration of the additional information with the former collected pieces of information in order to achieve a recognition result that is supported by all available kinds of information. In the following the integration of additional information is evaluated for the *baufix*[®] and the office object recognition systems, respectively.

5.5.1 Semantic Region Growing

For each set of object hypotheses neighbored regions carrying the same object labels are assumed to represent an over segmented form of one object, as described in Sec. 3.3.1. By applying the merging approach an additional area is generated from two or more ones represented within the hierarchical representation of the current hypothesis, see Sec. 4.3.3. Adding the area at the uppermost layer of the hierarchy and re-interpreting it, results in an additional competing hypothesis assuming the new area to represent the object region.

For the *baufix*[®] scenario, Tab. 5.5, on page 121, shows that for the system constellation using the hybrid recognizer applied to the current test set containing 167 elements, there are just 5 object regions that are over segmented. The main part of not correctly represented object regions are under segmented and, therefore, are not addressed here.

For the 5 objects, just one correctly represented object region is constructed and gets the correct label by the merging approach. Semantic region growing does not play a role within this integrated system.

The situation differs for the system constellation using the less specialized region based classifier combination module. This module principally delivers the individual parts of an isolated bolt, instead of an area representing the whole object region. The test set contains 11 bolts with neighbored head and thread, as already presented in Sec. 5.3.1, Tab. 5.13, on page 128, with the individual evaluation of the recognition module. For 2 bolts, no areas that correctly segment both parts of the object region are represented. For 1 further bolt, merging is not carried out, due to different object labels delivered with the parts. The semantic region growing approach leads to correct segmentation and object label results for the remaining 8 isolated bolts. Additionally, the head of an assembled bolt, a bar and two rhombs are merged from their over segmented parts and labeled correctly. In summary, integrating object context knowledge in the form of the simple semantic region growing approach improves the rate of occurring correct recognition results within the set of competing hypotheses for the *baufix*[®] system using the region based classifier combination module from 58.1% to 65.3% (97 and 109 of 167 objects, respectively, see Tab. 5.19, on page 133). Comparing this result with the rate of 50.9% (85 of 167 objects, see Tab. 5.13, on page 128) achieved for the individual module shows a remarkable and, in spite of the small test set, statistically significant improvement.

5.5.2 Assembly recognition process

The assembly recognition process described in Sec. 3.3.2 is based on recognition results for the *baufix*[®] elements. Based on the preliminary, competing results generated from integrating visual data based modules, a semantic model of the current assembly is constructed that allows hypothesizing the labels for missing elements. Principally, the approach of delivering additional object labels from assemblage knowledge is reduced to objects carrying no valid data based object label. Object labels are generated for colored areas in a way that label and area information fit into the partially constructed semantic model of the current assembly.

The additional labels are integrated with the existing visual data based information represented in the common hierarchical form, see Sec. 4.3.3. The object information is related to the corresponding area that occurs within the upper layer of the hierarchy by applying the region based object integration approach, see Sec. 4.2.3. Re-analyzing this representation delivers again a hypothesis for the corresponding area, while the related object information is probabilistically integrated. Thereby, data based labels that are formerly outvoted but correspond with the new label come into play again. Similarly, strong data based hypotheses against a valid object label may be stronger than the new label. However, if the assemblage recognition module delivers a label for an area that carries no data based object information this label is accepted and leads to knowledge based hallucinations.

For the current data set, the integrated system using the hybrid recognition module delivers just for 6 elements hypotheses for object regions without valid labels that are candidates for improvements by the assemblage knowledge. For just 2 of them, addi-

tional object labels are generated based on assemblage knowledge and lead to improved recognition results. The correct object region hypotheses for the remaining 4 objects get no additional label, due to the topology of the surrounding parts or due to recognition errors for the the surrounding parts that inhibit the construction of an appropriate model for the assembly.

The improvement of the recognition rate by the assemblage knowledge, thereby, is not significant. This is last but not least caused by the principal restriction that just those objects hypotheses that carry no valid label are addressed. Many dirt regions carrying no data based object information get object labels from the assemblage knowledge and lead to the above mentioned hallucination of elements.

Besides delivering additional object labels the assemblage knowledge may be exploited for evaluating the competing hypotheses, which is discussed in Sec. 5.6.

5.5.3 Expectations from Monitoring the Construction Process

Knowledge from monitoring the construction actions leading to an assembly constitutes the temporal context of the given scene and is exploited as an additional source of information for the recognition process, see Sec. 3.3.3. Therefore, knowledge about assembled elements is gathered during monitoring the process of taking elements and, finally, placing the assembly on the table.

The set of probably assembled elements serves as expectations for elements to be recognized at that point of time, when the assembly is supposed to be placed on the table. Detecting the point of time of placing an object may be done based on hints from speech or noise recognition or, as it is done actually here, by monitoring changes within the scene, in the form of appearing color regions. Color regions delivered from the pixel color classification approach are used here. If new regions appear and the module monitoring the construction process contains information about disappeared elements that can be connected to an assembly, the appropriate assembly is supposed to be placed on the table. The elements of this assembly, then, are expected to be recognized within the current scene and an appropriate stream of information is generated. Fig. 5.6 shows the static recognition system and its connections to the components, necessary for monitoring the construction process.

Information about expected elements generated from process knowledge are not related to an image location, i.e., just element classes and colors are given, but no image coordinates. Integration into the common representation, therefore, is done based on matching the expectations to intermediate object hypotheses generated from sources of data based information, as described, generally, for non located additional object information in Sec. 4.4. Competing hypotheses are processed independently one after another.

The matching process is controlled by defining costs for pairwise matches between each one hypothesis and one expectation dependent on their agreement, here, in label and color, as described in Sec. 4.4.1. The globally optimal match for a set of objects is determined by the set of pairwise matches that provides each object hypothesis and each expectation definitely once and minimizes the sum of costs.

As a result of this matching process an additional object label from expectation is available for each object without any segment information. This object label is integrated

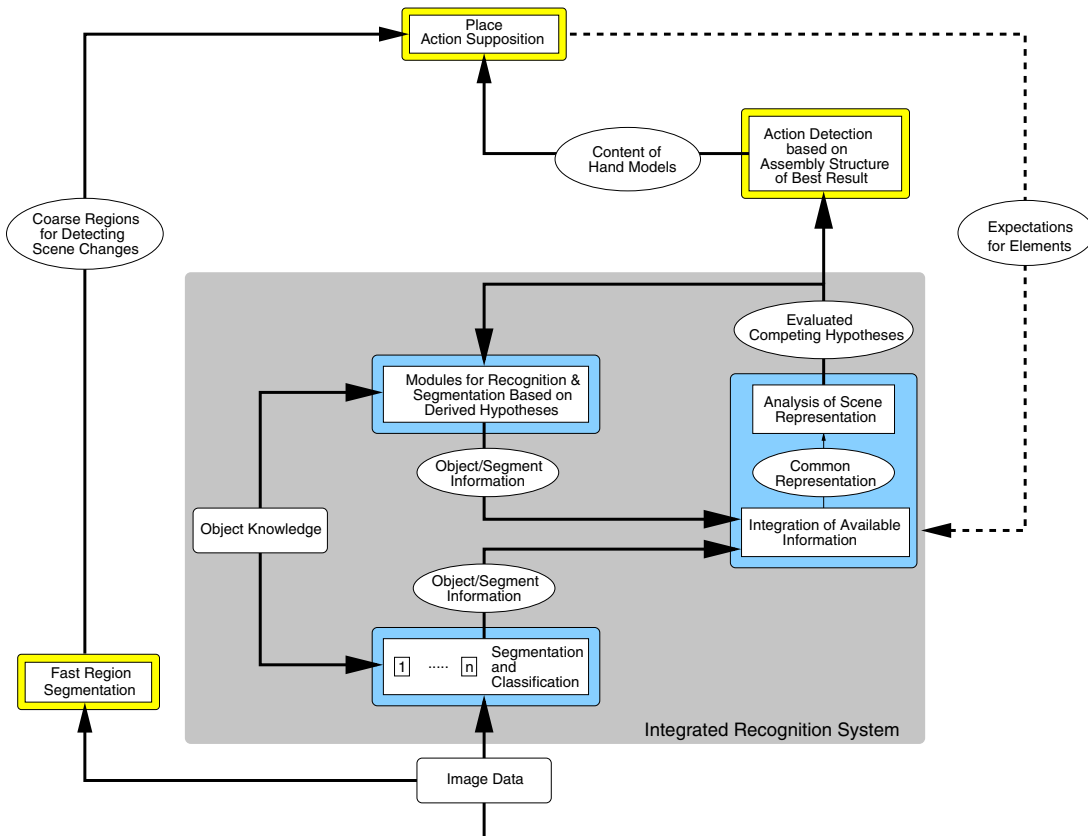


Figure 5.6: System configuration for generating expectations for the integrated baufix® element recognition from monitoring the construction process.

with the former available information for this object by relating it to the corresponding area of the hierarchical representation using the region based object integration step, described in Sec. 4.2.3. Re-interpreting the hierarchical structure leads to hypotheses that take into account all the available information for this object.

Correct expectations are generated for the 167 baufix® elements of the current image test set and integrated with the system constellation using the hybrid recognition module. The integrated data based processes of this system deliver 6 elements with no or undefined labels, and 12 with false object labels that are 18 candidates for improvements, see Tab. 5.20, on page 134. The matching process is always successful, leading to correctly related additional object labels. For 2 of the undefined objects and 10 of the false objects the expectations correct the failures of the data based processes. For the remaining 6 objects the false object labels from the data based processes outvote the correct one within the probabilistic integration procedure. The integration of correct expectations improves the number of occurring correct object recognition results within the competing hypotheses of the system statistically significantly from 108 to 120, corresponding to 64.7% and 71.9%, respectively.

5.5.4 Shape Based Office Object Recognition

A somewhat different kind of additional knowledge is integrated for the office domain in the form of a shape based recognition system, as is it described in Sec. 3.2.4. The recognition module is generally applicable to segment information like it is delivered from any individual data based segmentation process and it is a candidate for constituting a data based recognition module within other systems addressing different tasks. But the recognition process depends crucially on global shape properties and, thereby, on segment information that correctly represents an object region. The evaluation of the integration scheme for segmentation results for the office objects, given in Sec. 5.2.3, shows that nearly no object region is correctly segmented by an individual module, while the common representation contains 30 of 40 areas at all that correspond to object regions. 28 of them are identified from the rules based interpretation step as probably representing object regions and contained within the preliminary, integrated, competing hypotheses. Therefore, for this task the shape based recognition is much more promising to apply to the integrated results than to the segments delivered from the individual approaches. The shape based classification module was trained with a very small data set for this task, as described in Sec. 3.2.4, which results in rather bad recognition results. From the 28 candidate areas just 10 are classified correctly into one of the 6 valid object classes. The undefined class was trained with shadow and dirt regions. Within the test, it covers indeed most of the shadow regions but also many of the valid objects. Better recognition results are surely expectable after proceeding a more extensive training phase. Concerning the integration aspect that is the focus here, the integrated system realized for the office domain shows the flexible applicability of the general integrating framework and the improvements of integration in comparison to the individual modules.

5.6 Evaluation Scheme for Competing Hypotheses

The interpretation step of the common representation and the further integration of additional knowledge generally results in competing hypotheses for an image area. These results are evaluated based on different criteria that are determined individually and can be combined by applying the evaluation cascade described in Sec. 4.5.

For the *baufix*[®] domain, the generally applicable criterion concerning the object label integration and, additionally, an evaluation criterion based on the degree of assemblage and one based on the degree of fulfilled expectations is applied. The criteria are calculated for the competing hypotheses generated by integrating the image data based processes of the *baufix*[®] system using the hybrid recognition module. The office domain system relies currently on the individual confidence value of the shape based recognition. Its results are not in the current implementation not sustainable and, therefore, not investigated further concerning their confidence values.

Competing hypotheses are generated and, thereby, have to be evaluated based on clusters, the image sub units constituted by neighbored areas, as introduced in Sec. 4.2.2. For the current test set the 167 elements are processed within 64 automatically generated clusters that mainly but not always correspond to *baufix*[®] assemblies and isolated

elements, respectively.

Mean Object Label Confidence For evaluating competing results without taking into account domain specific knowledge, a confidence value is calculated based on the individually evaluated participating object labels and their integration, as described in Sec. 4.5. Based on the confidence value for each object generated from the probabilistic integration process, a set of hypotheses is evaluated by the mean object label confidence.

Within the current test set 64 clusters of *baufix*[®] elements are evaluated. 43 evaluations are successful, i.e., the best competing hypothesis gets the highest score. They consider 24 isolated elements and 19 assemblies or agglomerates of elements. However, for 21 clusters, containing 6 isolated elements and 15 assemblies or agglomerates a non optimal result is chosen. The best evaluated hypotheses contain 85 of 108 correctly segmented and recognized elements and 24 superfluous elements.

The evaluation method of scoring the mean object label evaluation provides the problem of tending to score more detailed hypotheses higher than others, because there is a lower probability of disagreements within the available object information. This fact is underlined by the number of 24 superfluous elements, which are generated by hypothesizing, for example, two bolts instead of one, if areas representing an over segment object region exist besides the correctly segmenting one.

Assemblage knowledge The evaluation scheme based on assemblage knowledge is based on the assumption, that there are one or more assemblies in the image. Then those elementary results are expected to be reliable that lead to as few as possible assemblies containing as many as possible elements. This motivates to the following evaluation function:

$$\text{eval}_{\text{assemblage}} = \frac{\#\text{assembled elements}}{\#\text{all elements}} \cdot \frac{1}{\#\text{assemblies}}$$

This evaluation function is again applied to the competing hypotheses, generated from the integrated visual data based results, providing 64 clusters of *baufix*[®] elements for the current test set. 48 evaluations based on assemblage knowledge are successful, scoring the best competing hypothesis highest. These hypotheses contain 106 of 126 correctly represented and 95 of 108 correctly recognized elements as well as 13 superfluous elements.

The domain specific criterion based on assemblage knowledge delivers more sustainable evaluations but the results are far from guaranteeing the choice of the best competing result. What are the reasons for the failure of the evaluation function? The problem occurs, because a recognition result containing all elements correctly segmented and labeled is often not available. This is due to recognition failures, but also due to occlusions. Missing and false object hypotheses, especially concerning the functional parts of an assembly, the bolts and nuts, often avoid the construction of a suitable semantic assembly model that constitutes the basis of this evaluation criterion.

Expectations Expectations are non localized and have to be related to the competing hypotheses by a matching process, as described in Sec. `refsecmatchingExpectations`. This matching process is applied in the previous section for generating additional object labels from expectation. But the accordance between hypotheses and expectations can also be exploited for determining a confidence value for the set of hypotheses.

The confidence for a set of objects is calculated by, firstly, evaluating each determined pairwise match between an object hypothesis and an expectation based on the manually chosen confidence values, given in Tab. 5.23.

Eval	Given for a Match between
1.0	equally colored and labeled hypothesis and expectation
0.9	color region with no valid object label and no expectation
0.0	differently colored and labeled hypothesis and expectation or hypothesis and no expectation
0.5	other constellations, for example, equally colored and differently labeled hypothesis and expectation or expectation and no hypothesis, etc.

Table 5.23: Evaluation of matches between an object hypothesis and an evaluation based on the differences concerning object label and color.

The evaluation of each match is defined to cover the commonly used range between zero and one. Agreements for color and label are best evaluated, while complete disagreements worst. Good evaluated is also the match between a non valid hypothesis, carrying an undefined or no label and probably representing a shadow and no expectation. The remaining cases of partially agreements get the mean value of 0.5.

The resulting confidence value for a set of hypotheses and expectations is calculated by summarizing the appropriate values for each hypothesis and, additionally, for the non matched expectations. The sum is normalized by dividing by the number of entities accounted for.

This evaluation function is applied to the competing hypotheses generated for the `baufix`[®] testset from integrating the image data based results and the formerly generated set of correct expectations.

Within the test set the 167 elements are arranged in 37 isolated ones and 34 assemblies. The 71 units are taken into account separately, resulting in 60 successful evaluations. The selected hypotheses contain 116 of 126 correctly represented and 102 of 108 correctly recognized elements as well as 9 superfluous elements.

Because this analysis is done based on completely correct expectations, it delivers the best result. Nonetheless, also this criterion does not guarantee the selection of the correct hypothesis, again due to the fact that the correct recognition result for the cluster often is not available. Too many recognition failures disturb the expectation matching process resulting in non valid evaluations.

Cascade The different evaluation criteria are organized applying a cascade, i.e., they, firstly, get a priority. Starting with the highest priority the corresponding function is

decisive and only for competing hypotheses that get an equal evaluation following this function the evaluation function of the next priority level is done. Within this scheme it is no principal problem if single functions are not applicable due to the current system state, for example, in case of missing data. Evaluating the degree of accordance with expectations is only possible, if expectations are available. For the *baufix*[®] domain expectations are just generated at the end of a construction process.

The results for the individual evaluation criteria for the *baufix*[®] domain, which are described above, show that, if available, the expectations give the most reliable results. Therefore, the corresponding evaluation criterion gets the highest priority. If they deliver ambiguous results or are not available, the evaluation based on assemblage knowledge is calculated. The mean object label evaluation provides principal problems with over segmentations, as discussed above. Therefore, this evaluation criterion gets the lowest priority. But it can be calculated as far as one individually evaluated or two not necessarily individually evaluated sources of object information are available and, thereby, constitutes the fall back of the cascade.

Generally, independent sources of information should be taken into account for hypotheses generation and evaluation, if possible. For example, if expectations are used for generating additional object labels the resulting competing hypotheses should not be evaluated by the degree of fulfilled expectations in order to avoid hallucinations due to the strong influence of this knowledge. This also concerns the exploitation of assemblage knowledge for the *baufix*[®] domain. The decision, which source of information is where to use

For improving the evaluation step, besides the presented criteria, further approaches implementing domain specific restrictions for element colors, constellations, or other characteristics, as motivated in Sec. 5.4.4, are principally applicable. They can do, for example, a pre selection of promising candidates by distinguishing valid from non valid hypotheses at the uppermost level of the cascade.

5.7 Summarizing the Evaluation

The preceding sections contain the evaluation of object segmentation and recognition systems that are realized based on the general integrating framework proposed by this thesis. The evaluated systems are concerned with two different tasks, namely the recognition of *baufix*[®] elements and the recognition of objects occurring in an office environment.

For realizing a system based on the general integrating framework the participating individual modules have to be available and adapted to the communication framework DACS. Concerning the integrating module the domain specific parts have to be supplemented, which are the definition of the occurring object labels and their coherence as well as sustainable evaluation criteria. The heart of the system, the hierarchical representation of segmentation and classification results and its interpretation procedure remains constant and can be adapted to a new domain by setting few parameters. The two central parameters, controlling the classification of areas relations for the common hierarchical representation and the integration of point based object information with external segments, turn out to be uncritical in their numerical setting for the realized

systems.

For quantitatively evaluating an object recognition system the rate of correct results from all tested ones is central. Nonetheless, in the scope of the presented evaluations are relative improvements achieved by applying the integrating framework and intermediate results for the parts of the integrating module.

For all realized systems the common representation of segment and object information is shown to improve the rate of represented correctly segmented object regions in comparison to the participating modules. The rule based interpretation step is shown to be suitable for selecting the correct areas from the common representation and include it into at least one of generally several competing object hypotheses.

For generating an object label for a given area the available object information is integrated probabilistically. The evaluation of the available *baufix*[®] object data shows that for many areas contradicting labels from the different sources occur and, therefore, underlines the need of a suitable integration scheme. Due to the aim of flexible combination of available modules, no confidence for the different sources of object are considered. In spite of these simplifications, the integration scheme turns out to be successful for the *baufix*[®] labels.

The competing object segmentation and recognition hypotheses that result from interpreting the common hierarchical representation constitute the basis for integrating additional knowledge that depends on preliminary hypotheses. For the office domain system the one and only available object recognition module needs the preliminary integrated segment hypotheses. Even, if the recognition rates for this module are not satisfying, the realized system shows the principal necessity of integrating object labels based on preliminary integrated results. For the *baufix*[®] domain, information from assemblage knowledge, expectations from the temporal history of the scene, and assumptions from semantic region growing are available for the integration with data based results. The modules are applied one after the other to the preliminary hypotheses and they show different degrees of refinements.

Uncertainties in the interpretation of disagreeing different kinds of information lead to competing results. In order to evaluate the competing hypotheses, generally, the individually evaluated object labels that are integrated and their accordance, are determined. For the office system just the individual evaluations delivered from the one source of object information are taken into account. For the *baufix*[®] task, additionally domain specific criteria are supplemented calculating confidence values based on the degree of fulfilled expectations and the degree of assemblage. It turns out that the implemented evaluation criteria are not sufficient for guaranteeing the selection of the best hypothesis out of the competing ones. For optimizing the results additional domain specific evaluation criteria has to be defined and integrated to the realized systems.

The quantitative evaluation of three realized integrated system constellations for two different tasks demonstrates the broad applicability of the proposed general integrating framework. The detailed quantitative evaluations show that the integrated system exceeds its parts significantly. The results validate the promising results achieved by a former, smaller system configuration for *baufix*[®] recognition, described in detail in [Brau 99] and [Sage 01].

6 Summary and Conclusion

Object segmentation and recognition is one of the most active research fields in computer science. Symbolic information assigned to visual sensory data plays an important role for computational applications reaching from industrial quality assurance to advanced human machine communication. However, compared to the human object recognition abilities the automatic systems have to be developed much further to become an adequate communication partner.

Many approaches to object segmentation and recognition exist that follow diverse processing strategies, rely on different assumptions, and exploit different characteristics of tasks and objects. Results from data driven processes searching for models that correspond to the data, are opposed to results achieved by conceptually driven processes that determine sensory data that corresponds to models. For a given recognition task, generally, various individual approaches deliver valuable pieces of information that are promising to complement one another. The question arises, how to integrate the different aspects within one object recognition system.

One possibility is integrating diverse features and strategies for exploiting different sources of information implicitly within one monolithic system. Such a system depends on the set of completely calculated features and, therefore, has to be appropriately redesigned, if feature sources are exchanged, like a classifier operating on a high dimensional feature space. Additionally, the influence of the isolated features is obscured within the whole system. The contrary approach integrates explicitly individually generated results, each based on comparably simple recognition mechanisms. Appropriate existent generally applicable recognition and segmentation modules, therefore, may be reused by adapting them to a given task. Generally applicable external combination schemes are flexible with respect to exchanges of individual modules. Several approaches are proposed for combining object labels from several sources based on a common label alphabet. However, these approaches are not easily extendable to the simultaneous consideration of segmentation information.

Besides modules that individually assign image data to object symbols following the one or the other processing direction, additional object knowledge serves for verifying object hypotheses, clarifying ambiguities, and extending missing hypotheses. This kind of knowledge is concerned with individual object models, object context models or relations to other modalities, like gestures or speech. It delivers additional valuable information for recognition, if it is integrated with the image data based generated information.

This thesis proposes a framework that provides a standardized external integration scheme for diverse visual data based object labels and segment information with additional information based, for example, on higher level object model, like context knowledge. The integrating framework supports the fast and flexible realization of integrated object recognition systems based on a strategy that is independent of the recognition

task and the set of participating modules. In contrast, systems applying advanced learning approaches for acquiring knowledge about joint probabilities concerning the available information depend strongly on the set of participating modules. Realizing a system necessitates a training phase and exchanging individual parts is equivalent to re-generating the whole system.

The central part of the proposed framework constitutes the integrating module that takes into account the available information for generating object hypotheses.

6.1 The General Integrating Module

The implemented general integrating module consists of three major parts, the common knowledge representation, its analysis resulting in competing object hypotheses, and the evaluation of the competing results.

The common hierarchical representation of segment and object information collects the available information from different sources of knowledge. The amount of information is structured by relating those pieces of information that cover a common image area and, therefore, has to be taken into account collectively for the appropriate local interpretation. The integration of belatedly arriving information concerning an image area is supported because it leads to just local adaptations of the data structure.

Generating object hypotheses based on the available information is often ambiguous due to disagreements within the available information caused by individual recognition failures. Uncertainties in analyzing the available information results in, generally, more than one competing hypotheses containing different object region and label assumptions. Probable candidates are identified from the available information by applying a set of general interpretation rules. The rules restrict the number of competing results in order to enable the subsequent integration of additional high level information for clarifying uncertainties. The premature restriction to one hypothesis is especially not promising as long as not all information is integrated. However, additional information is characterized by its dependence on preliminary object hypotheses, like knowledge from object based context. Consequently, the additional information has to be applied and its results has to be integrated separately to the competing preliminary hypotheses, requiring a restricted number of them. The proposed unified knowledge representation and interpretation strategy applied for integrating individually generated pieces of information is reusable for the belated consideration of additional information. Its integration with data based information is equitable and avoids the dominance of one source of information over the other. The effect of hallucinating objects due to strong expectations based on object knowledge is avoided, then, by considering contradiction based on sensory data.

The third part of the general integrating modules constitutes a scheme for comparing and evaluating competing hypotheses in order to identify the most believed unique result and to provide confidence values concerning the hypotheses to be used by subsequent processes. The general evaluation scheme is a cascade that is prepared for analyzing several independent evaluation criteria. The most reliable and adequately prioritized criterion is applied first, while following ones serve for clarifying ambiguities within the first one. A generally applicable criterion evaluates the disagreements within the inte-

grated object information related to each object. For achieving discriminative evaluation criteria, the scheme can be easily extended by meaningful task specific evaluations.

6.2 Realizing Integrated Object Recognition Systems

For realizing a recognition system for a given task based on the general integrating framework it is necessary to provide suitable participating recognition and segmentation modules. In order to enable their communication with the implemented integrating module their interfaces have to be adapted to the standards supported by the inter process communication system DACS. Concerning the integration process, itself, the different occurring task specific object label alphabets, accompanied probably by individual confidence information, and their coherence have to be supplemented. The main parameters of the integration module concerned with the determination of relations between pairs of areas and between areas and object information, are adapted to the task and the participating modules. The implemented integrating module provides a standardized data representation and interpretation step that both, generally, do not depend on the recognition task and remain constant for all systems.

The presented realized systems address two different domains, the recognition of *baufix*[®] elements and of objects occurring in an office environment. The detailed quantitative analysis of the realized application systems underlines the comprehensive representation of the integrated information that improves each of the individual sources of information. Further on, the interpretation mechanism is able to extract the correct results from the representation together with other competing ones. However, the implemented general and domain specific evaluation criteria should be improved in order to reliably extract the best unique result from the set of competing ones. The problems arise mainly due to the rather high rate of recognition failures that disturb especially the evaluation criteria based on higher level knowledge for the *baufix*[®] domain. Thereby, neither the selection of participating modules, nor the individual modules, nor the applied evaluation criteria have been optimized for achieving absolutely optimal results for the recognition systems. In contrast, the integrated systems are realized for proofing the standardized and simple integration and interpretation strategy to be able to improve the results of the participating isolated modules. The systems demonstrate the general applicability of the general framework to different tasks.

The major part of failures for the realized systems originate from segmentation failures and from that the under segmentations. There is currently no mechanism implemented for actively splitting under segmented areas. Besides failures within the individual modules, also the construction of artificial union areas by the integrating module from partially overlapping original ones constitutes a source of under segmented areas. The problem can be addressed within the integrating framework by introducing a further module into the system that implements an object model based segmentation approach, like proposed in [Bore 02], [Bore 04], and delivers additional information based on preliminary object hypotheses.

6.3 Conclusion

The proposed integrating framework constitutes a valuable tool for addressing a given object segmentation and recognition task based on the main idea of considering equitably different sources of information. The different sources and kinds of information complement one another for achieving sustainable object segmentation and labeling results. Segment information determines the appropriate clustering of available object information, while object information determines the selection of probable candidates for object regions from the amount of available segment information.

Individual segmentation and recognition modules deliver the different kinds of information by following different processing strategies. The equitable integration of available information supports the extension of image data based generated object hypotheses with valuable additional information, as delivered, for example, from object context knowledge. Concurrently, this approach avoids conceptually driven hallucinations by appropriately considering sensory data.

The applied integration mechanism is external and static in order to preserve the independence of participating modules. This guarantees flexibility with respect to exchanges within the set of participating modules or changes of the application domain. The proposed integrating framework constitutes the basis for realizing successful, as well as, smart and flexible recognition systems.

A The Inspection Tool for Integrated Systems

An object recognition system has to deliver symbolic object information assigned to a digital image. The object module is defined by its interfaces and a parameter setting that has to be fixed during the adaption of the system to the given task. For a system based on the proposed integrating framework during the system adaptation phase besides parameter settings also decisions about the choice of available modules to be integrated has to be made. Making these decisions just based on the final result is very difficult due to the manifold effects that happen in the interaction of different input data and parameter settings. In the phase of the system constitution for a new task the inspection of the content of internal data structures is indispensable. Additionally, for false recognition results, an inspection tool helps to identify the reason of the error, which is the first step of solving the problem.

Given the requirements of providing generally the possibility of system inspection, but working mostly without visualization overhead leads to an implementation that is managed by a command line containing input and output data descriptions and parameter settings. A command line parameter also specifies, whether a visualization of system internals is wanted or not. If so, the necessary information about the system internals are submitted to a visualization sub system. Besides showing details of the inner data structures, it provides a graphical user interface to the recognition module. The user controls the system by selecting the pieces of information to be integrated, changing parameter values, and introducing break points.

Internal system information that is valuable for visualization occurs at several points within the recognition module and data is submitted as soon as it constitutes consistent units. Otherwise, the visualization would cause the storing of many intermediate data within the recognition module. The decentral submission of data units is realized by the interface functions provided by the open source window management library 'Fast Light Tool Kit' (FLTK) that is distributed from www.fltk.org. It offers basic windowing management functionality combined with an easy to use user interface in the form of C++ classes and methods. It is implemented for several platforms.

Fig. A.1 shows a screen shot of the main window of the visualization sub system implemented for systems realized based on the general integrating framework.

The main window shows within its central part, firstly, the data that is integrated within the common hierarchical representation. For addressing the different aspects of the representation, like the layer structure, sets of similar or partially overlapping areas, or the relation of object information, the local menu bar of the central window offers different visualizations. Here, the layers of the hierarchical representation of segment information are color coded.

Besides the central data part, in the upper left corner of the main window the input streams are listed. A click at the central buttons opens a window presenting the appropriate input information, as exemplary shown in Fig. A.2.

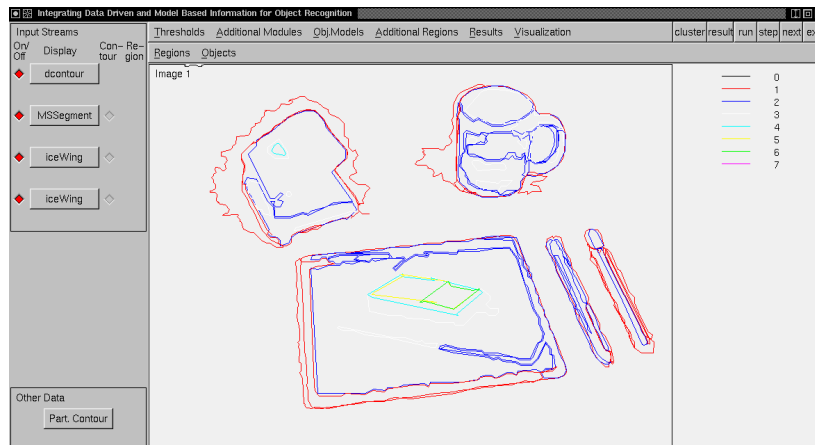


Figure A.1: Main window of the visualization tool for inspecting an integrated system showing, here, the color coded hierarchical common representation of the individual segment inputs.

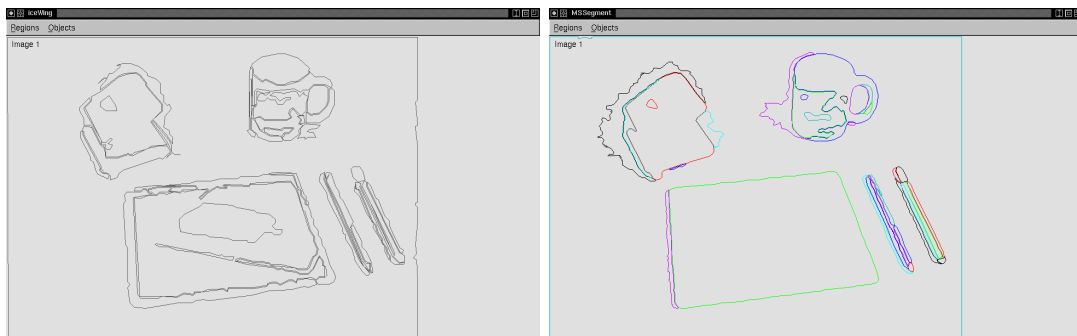


Figure A.2: Different input data stream visualizations for segments from the local variation approach presented uniquely in grey and the mean-shift approach drawn in random colors.

The available input data streams can individually switched on and off by the radio buttons of the left column. The two columns of radio buttons right of the input data buttons within the main window provide the possibility of selecting or neglecting special data from the input stream, like assigned contours, if available, from a stream delivering segment information or segment information given with a stream of object information. This allows to flexibly generate different combinations of input data and test it for its effects on the integrated representation and, finally, on the recognition result.

The button at the lower left of the main window hides information about the exceptionally handled partially overlapping grouping hypotheses.

The menu bar of the main window contains entries for influencing the integration process by, for example, parameter settings, switching on and off the usage of additional modules for integration, and controlling the application of different criteria for evaluating the results.

The buttons located at the upper right corner of the main window serve for switching the content of the central area (two buttons) and for controlling the progress of the program (four buttons). The latter determine, whether the program stops after the next recognition unit or runs without break just visualizing the data, and, whether the calculations are repeated on the same data or new data is loaded from the input streams.

For the central area of the main window further contents are selectable, as shown in Fig. A.3.

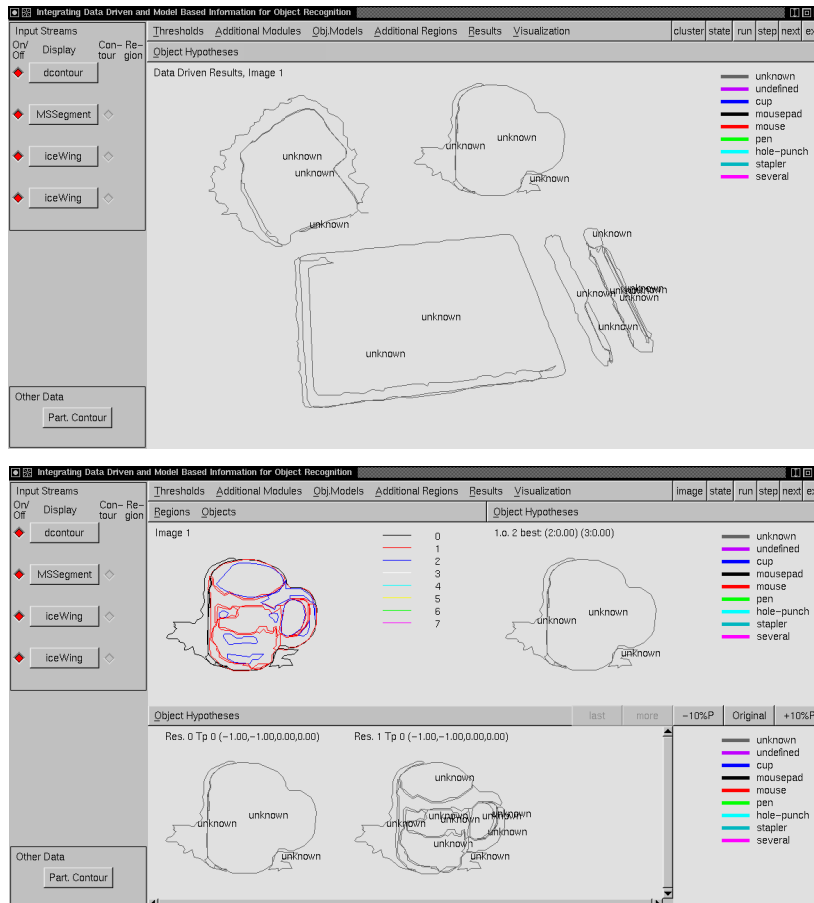


Figure A.3: Alternative selectable contents for the main window of the inspection tool showing either the best believed object hypotheses for the image or detailed information about one segment cluster.

The upper part shows the main window with its central part visualizing the best believed object hypotheses for the current image. The lower part presents a central window divided into three parts providing detailed information concerning one cluster of areas. The upper left quarter of the central area shows the data integrated within the common hierarchical representation for the current cluster, comparable to mode for visualizing these data for the whole image, shown in Fig. A.1. The lower half of the window shows competing results together with their evaluations that are generated while interpreting the hierarchy. Finally, the upper right quarter shows the best believed

result, comparable to mode for visualizing these results for the whole image, shown in the upper part of the figure.

For a better insight into the hierarchical representation the areas and their relations to each other may be visualized as a graph within an individual window. Thereby, the graph consists of nodes representing the occurring areas and edges representing the explicitly represented relations between them. Areas that are independent are related via a white edge, while the connection between one area containing another is drawn in black. The menu bar of the windows allows further variations in the presentation, like showing just the main structure constituted by independent (white) and contained (black) areas or extend this by similar and partially overlapping original areas, as depicted in Fig.A.4.

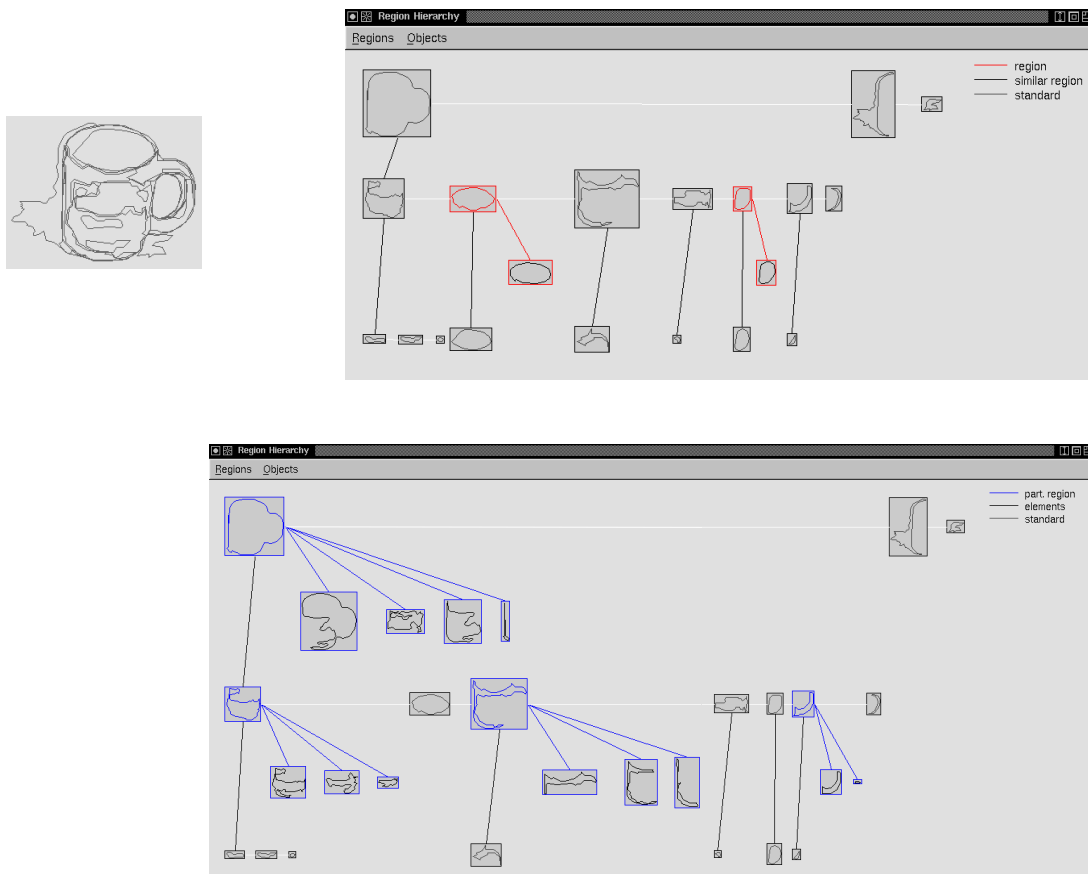


Figure A.4: Exemplary graph presentation of the common hierarchical representation showing the main structure of independent (white) and contained (black) areas extended by similar areas (upper) and the partially overlapping original areas (lower).

If object information is available from image data based modules, this can also be depicted within the graph visualization of the common representation, as it is shown in Fig. A.5.

Besides the overview presentation of the occurring areas and their relations, detailed

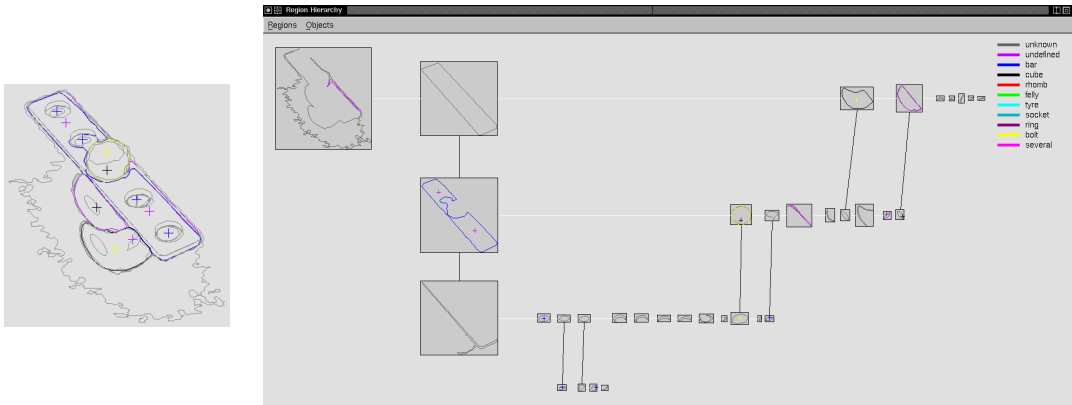


Figure A.5: Graph visualization of the common hierarchical representation containing segment and object information.

information about each of the areas is also possible to visualize within separate windows. The information is demanded by a double click to the area within one of the presentations containing the data of the common representation. Examples for the detailed area information shows Fig. A.6.



Figure A.6: Examples for individual area information containing spatial informations and related object information.

The lower part of Fig. A.3 already shows the cluster based version of the central part of the main window. Within the lower half all the present competing results of the system are depicted. These are, firstly, the preliminary competing hypotheses that are generated by interpreting the common hierarchical representation of image data based segment and object information. For each object within the competing hypotheses, information about the object label, its evaluation and the rule that cause its existence are available and printed in user defined combinations for better readability. Fig. A.7 shows the cluster

based main window containing these preliminary competing hypotheses for a *baufix*[®] example, where the objects are color coded and the labels are printed without further information. The preliminary competing hypotheses constitutes, generally, the basis for applying additional higher level modules, like the *baufix*[®] assembly recognition module. The additional object information delivered for each competing result is visualized by a separate windows that is opened, if required, by a mouse click on the appropriate hypothesis. Fig. A.7 shows the windows opened for two hypotheses. One of them gets no additional object information from the assemblage knowledge, while the second gets two additional hypotheses for cubes. The next program step integrates this information with the hierarchical representation of information underlying this hypothesis starts the re-interpretation taking into account the extended information.

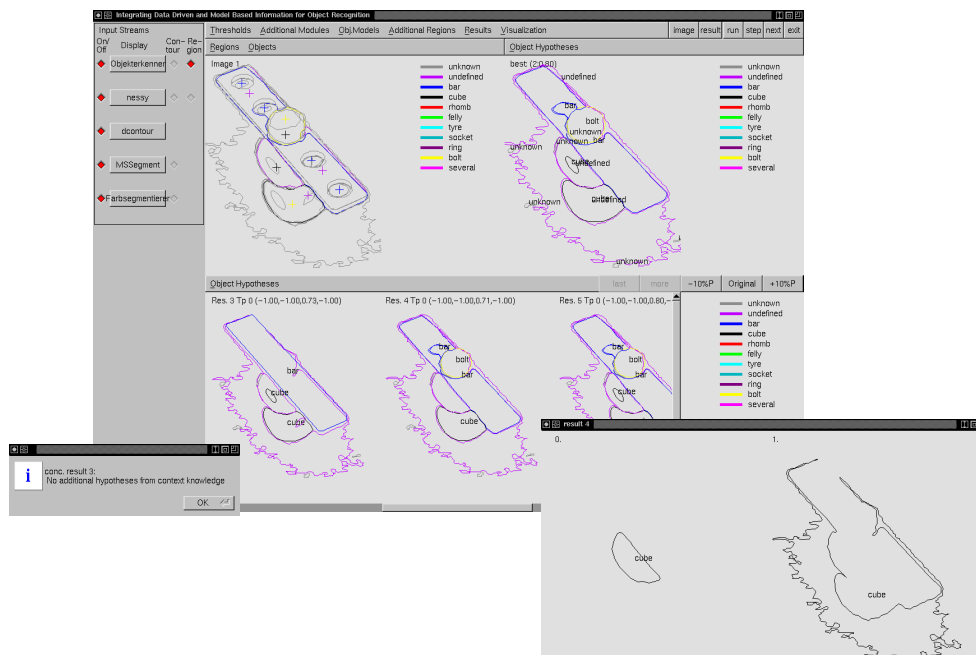


Figure A.7: Image data based generated competing hypotheses with assigned higher level object hypotheses, here delivered from the *baufix*[®] assemblage knowledge.

The implemented visualization tool for recognition systems that are realized based on the integrating framework allows an insight into the content of the internal data structures of the recognition system during its runtime. Detailed information about the participating original data sets, the common representation, and the resulting hypotheses allows the user to comprehend the analysis step by step. These features are indispensable in the phase of composing new systems. Additionally, for false recognition results the visualization tools allows the detection of the reason for the failure which may lead finally to an improvement of the system. Last but not least, the proposed visualization can be hidden completely and just some requests for the value of the visualization flag remains within the recognition module. This is the runtime mode, where the recognition module just has to do its work, as successful as possible.

B The baufix[®] Domain

As one application for the integrating recognition framework the domain of recognizing elements and assemblies of the wooden toy kit baufix[®] is chosen. baufix[®] allows the construction of assemblies from a set of elementary parts, as shown in Fig. B.1.

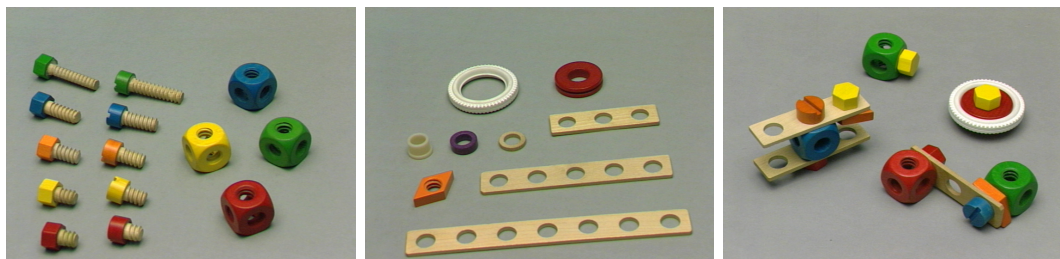


Figure B.1: Elementary baufix[®] parts and some exemplary assemblies

The following sections motivate this domain by its central position in several research projects and gives details that are necessary for the realization of the integrated system.

B.1 Motivation

The baufix[®] construction process is selected to be the application domain of the research projects summarized by the collaborative research center (SFB 360) at Bielefeld university. Topic of the SFB 360 is the development of 'Artificial Situated Communicators' [Rick 96]. Those systems are characterized by their ability to get into a dialog with a human user, where the dialog is embedded within a predefined situational context. This context is chosen to be the cooperative construction of baufix[®] assemblies. For realizing a dialog with the human instructor the computer system must be able to deal with speech and vision perceptions in order to appropriately process the user commands. Besides speech and object recognition higher level processes like speech understanding and multi modal scene understanding are necessary. Fig. B.2 shows the SFB scenario with a human instructor and robot constructor.

Within the vision perception centered projects some modules dealing with segmentation and recognition of elements and assemblies are developed and applied. The possibilities of constructing baufix[®] assemblies from the set of elements are manifold, but the construction generally follows some basic rules. Therefore the task of recognizing assemblies is addressed by modeling the assembly based on the construction rules and recognition results for the elements, see Sec. 3.3.2 or [Bauc 02]. This attempt requires reliable element segmentation and labeling that is addressed by the realized integrated recognition system.

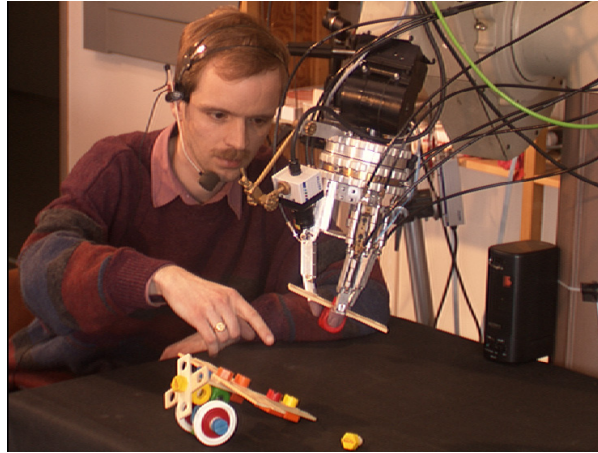


Figure B.2: The SFB 360 scenario with a human instructor and robotic constructor.

Fig. B.1 shows the elementary *baufix*[®] parts. Bolts and cubes are present in five and four colors, respectively. Several lengths occur with bolts and the bars. Additionally bolts may provide a round or hexagonal head. The remaining parts are unique in color and form. Each part, except the bolts, exhibit just one color. The set of occurring colors is reduced to nine rather saturated ones. No texture is provided by the flat surfaces, which supports the application of color segmentation approaches, but which also complicates the definition of discriminative features for the recognition.

In spite of the limited set of elements and colors the recognition of elements built in assemblies is difficult, due to the high spatial density of elements causing occlusions. Problems arise in detecting the element boundaries, due to shadows, highlights or neighbored elements of exactly the same color.

B.2 Object Label Alphabets

Different recognition strategies for *baufix*[®] elements that are isolated or connected with others in assemblies are implemented. Thereby, the modules rely more or less on preprocessing results and on more or less domain specific knowledge. There is the module that assigns an object label to each segment of unique color, the one that does not rely on segments at all, but instead identifies characteristic non occluded parts of an object, and the other one that has an explicit model of the bolt to be constituted from a head and a thread, see Sec. 3.2 for details. Additionally elements can be hypothesized from assemblage knowledge or information about an assembly gathered during the construction process, see Sec. 3.3. The different recognition strategies cause the individual modules to deliver different object label alphabets. Tab. B.1 shows the occurring object labels and their coherence. For applying a general integration scheme, a common label alphabet is necessary, which is given in the left column of Tab. B.1. The general element alphabet is designed to ensure that each member of each alphabet can be uniquely assigned to one of its members.

B.3 Classification Error Matrix for Probabilistic Integration

General Elements	Elements	Region Based Parts	Sub Elements
bar	3_hole_bar 5_hole_bar 7_hole_bar	3_hole_bar 5_hole_bar 7_hole_bar	edge_bar hole_bar
cube	cube	cube	part_cube edge_cube hole_cube
rhomb	rhomb	rhomb	part_rhomb side_rhomb hole_rhomb
felly	felly	felly	hole_felly felly
tyre	tyre	tyre	tyre
socket	socket	socket	
ring	ring	ring	
bolt	bolt	head_bolt thread_bolt	head_round_bolt head_side_round_bolt head_hexagon_bolt head_side_hexagon_bolt
undefined	undefined	undefined	undefined
unknown			

Table B.1: Object class labels and their coherence defined for the `baufix`[®] element recognition scheme.

B.3 Classification Error Matrix for Probabilistic Integration

For probabilistic object label integration, see Sec. 3.1.4 and Sec. 4.3.1, an estimation of the probability for all labels of the actual alphabet is desirable. This kind of data is available for the appearance based `baufix`[®] sub element recognition system, see Sec. 3.2.3 in the form of a classification error matrix.

Estimations for the probabilities for objects given a classified label m are derived from a comparison of classification results to a manually labeled ground truth data set. In counting, how many ground truth class labels n are automatically classified to class label m , it results a $N \times N$ matrix of estimations for probabilities with N the total number of classes. The true object label n is defined to denote the row index the classified label m the column index. The diagonal elements of the matrix contain the correct classifications, which are summarized to determine the classification rate. Row n represents the distribution of classification results with true object class n , column m the distribution of true objects that are classified to be m . The latter serve for the probabilistic integration,

after normalizing each column sum to one. Tab. B.2 shows the resulting error matrix for the `baufix`[®] sub element recognition system that is made available from the author Gunther Heidemann in personal communication.

B The *baufix*[®] Domain

undefined	0.7507	0.1199	0.0510	0.0705	0.0000	0.0170	0.0609	0.0217	0.0554	0.0662	0.1008	0.0540	0.1868	0.0000	0.0000	0.1442
part_cube	0.0184	0.5341	0.0848	0.2115	0.0000	0.0000	0.0058	0.0085	0.0085	0.0243	0.0045	0.0000	0.0098	0.0000	0.0000	0.0000
hole_cube	0.0167	0.2125	0.8277	0.0256	0.0000	0.0000	0.0058	0.0000	0.0000	0.0088	0.0045	0.0000	0.0056	0.0000	0.0597	0.0000
edge_cube	0.0083	0.0899	0.0000	0.5705	0.0000	0.0000	0.0000	0.0000	0.0000	0.0088	0.0125	0.0000	0.0000	0.0000	0.0000	0.0000
part_rhomb	0.0000	0.0000	0.0000	0.0000	0.8416	0.0170	0.0223	0.0058	0.0000	0.0088	0.0000	0.0000	0.0056	0.0000	0.0000	0.0000
hole_rhomb	0.0022	0.0000	0.0000	0.0000	0.0396	0.9149	0.0000	0.0058	0.0000	0.0088	0.0000	0.0000	0.0056	0.0000	0.0000	0.0000
side_rhomb	0.0206	0.0109	0.0064	0.0000	0.0396	0.0170	0.8722	0.0318	0.0000	0.0088	0.0079	0.0000	0.0056	0.0000	0.0000	0.0000
head_round_bolt	0.0039	0.0109	0.0036	0.0256	0.0000	0.0170	0.0000	0.8104	0.0640	0.0331	0.0045	0.0000	0.0000	0.0000	0.0000	0.0000
head_side_round_bolt	0.0000	0.0109	0.0000	0.0000	0.0000	0.0000	0.0000	0.6955	0.8060	0.0155	0.0170	0.0000	0.0000	0.0000	0.0000	0.0000
head_hexagon_bolt	0.0122	0.0109	0.0064	0.0000	0.0396	0.0000	0.0000	0.0000	0.0085	0.5717	0.1302	0.0000	0.0056	0.0000	0.0000	0.0000
head_side_hexagon_bolt	0.0206	0.0000	0.0137	0.0962	0.0000	0.0000	0.0142	0.0217	0.0405	0.2362	0.7010	0.0031	0.0000	0.0000	0.0000	0.0000
hole_bar	0.0679	0.0000	0.0000	0.0000	0.0396	0.0170	0.0000	0.0058	0.0085	0.0000	0.0079	0.9248	0.0309	0.0000	0.0000	0.0337
edge_bar	0.0640	0.0000	0.0064	0.0000	0.0000	0.0000	0.0304	0.0101	0.0085	0.0088	0.0045	0.0133	0.7388	0.0000	0.0000	0.0192
felly	0.0039	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.7895	0.0000	0.0000
hole_felly	0.0022	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0058	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9403	0.0192
tyre	0.0083	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0045	0.0049	0.0056	0.2105	0.0000	0.7837

Table B.2: Error matrix for appearance based *baufix*[®] sub part recognition system. Column m denotes the distribution of objects classified to class label m .

B.4 Test Set Images

An integrated recognition system for the **baufix**[®] domain is realized based on available individual modules and the general integration framework, see Chapter 5. The images used for the quantitative evaluation of this system are shown in Fig. B.3.

They are taken using a setup consisting of an operation table of constant background and a camera directed to the table at an angle of about 45° using constant camera parameters and indoor lighting. The test set consists of 10 images, containing 167 elements. 130 of the elements are used within 34 assemblies, resulting in 37 being isolated. The testset was designed from a naive user, the student worker Christian Lange, who did not know the recognition system. His task was to design a test set that is challenging for recognition. As a result, the images are characterized by containing complex assemblies providing many occluded elements and many examples of neighbored elements with same or similar color, which constitutes the main difficulty for the segmentation task within this domain.

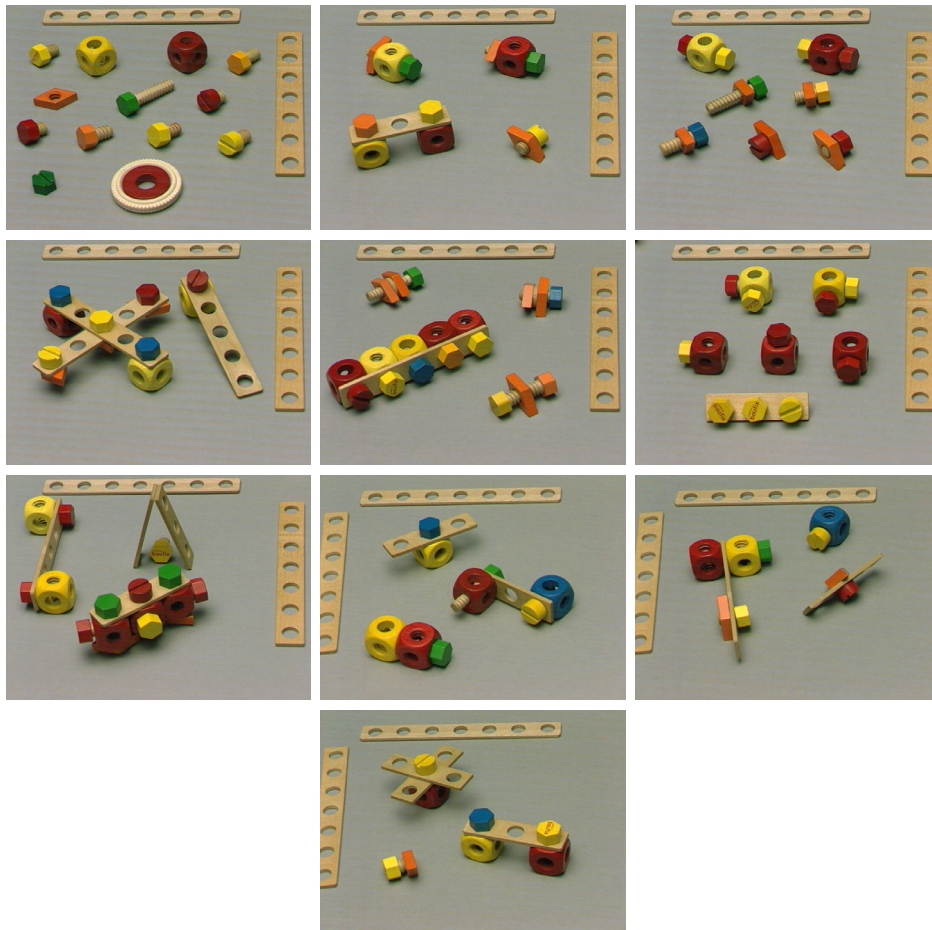


Figure B.3: Images containing **baufix**[®] elements and assemblies used for testing and evaluating the integrated recognition system for **baufix**[®] element recognition.

C The Office Domain

As an application for the general integrating framework a system for recognizing objects within an office environment is realized.

In general, the recognition of objects occurring within an office environment is addressed within many applications, due to the common availability and presence of those objects. Object recognition facilities for this domain are used, for example, for realizing advanced human computer interaction, like described in [Haas 04, Fink 04a, Frit 03, Bauc 04b]. Within the visual perception based modules of such systems both, object categorization as well as recognition of individual object exemplars, is needed dependent on the context. These challenges and the common availability of the objects are the reason for choosing the office environment as an application domain for the integrating recognition framework.

However, the system realized and evaluated for the office domain, here, is rather small and does not aim at solving the object recognition task within the office environment. But it shows the flexibility of the general integrating framework concerning its application to different domains.

In the following the recognition task is described, before the test data set used for evaluating the realized system is presented.

C.1 Recognition Task and Strategy

The realized recognition system deals with object categories in contrast to object exemplars. Several different exemplars exist for each category, as shown in Fig. C.1.

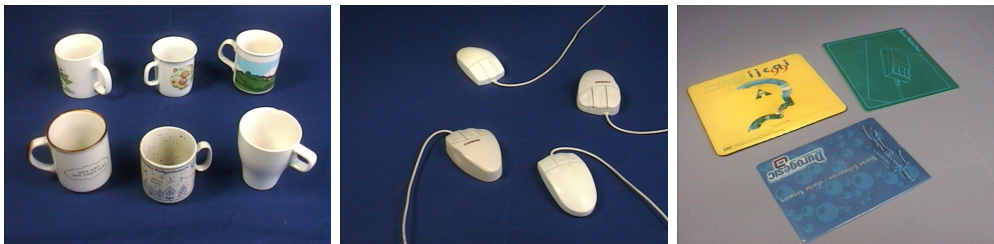


Figure C.1: Images containing exemplars of office objects belonging to three categories.

Object categorization implies the assignment of a known object category to an unknown object. Therefore, features are needed that abstract from the exemplar and that are discriminative for the object class. Shape features serve for this in the system realized here.

The exploitation of shape features requires a reliable hypothesis for the object region. Due to the structured object surfaces an individual data driven segmentation process will not be able to reliably deliver the object region represented as one segment. Therefore, the recognition strategy applied here, is to determine the object region by the integration of several data driven segmentation processes. The label class assignment does the shape based recognizer based on the integrated segment information. The choice of the best interpretation is done based on the confidence for the object label given by the recognition step.

Shape based features are rather susceptible for distortions originating from occlusions and shape is especially discriminative for characteristic views of an object. In the presence of arbitrary object views one shape based model for each category is not sufficient and has to be extended, for example, to an aspect graph or something equivalent. Nonetheless, the shape based classifier is applied to distinguish the object categories: cup, mousepad, mouse, pen, hole-punch, stapler, and undefined for rejection.

C.2 Test Set Images

An integrated system for office object categorization is realized as described in Chapter 5. The system was tested and evaluated based on a test set of images as given in Fig. C.2.

The test set consists of 12 images containing 40 objects lying on a table of constant background. The camera is at a fixed position above the table at an angle of about 45° and constant camera parameters and indoor lighting are used.

The content of the images is characterized by objects being close to each other, which occurs in realistic scenarios and complicates significantly the generation of a suitable object region hypothesis. But the objects are just slightly occluded in order to give the shape based recognizer a chance. Some objects within the test data were not included in the training data, like the orange mouse pad and some of the cups.

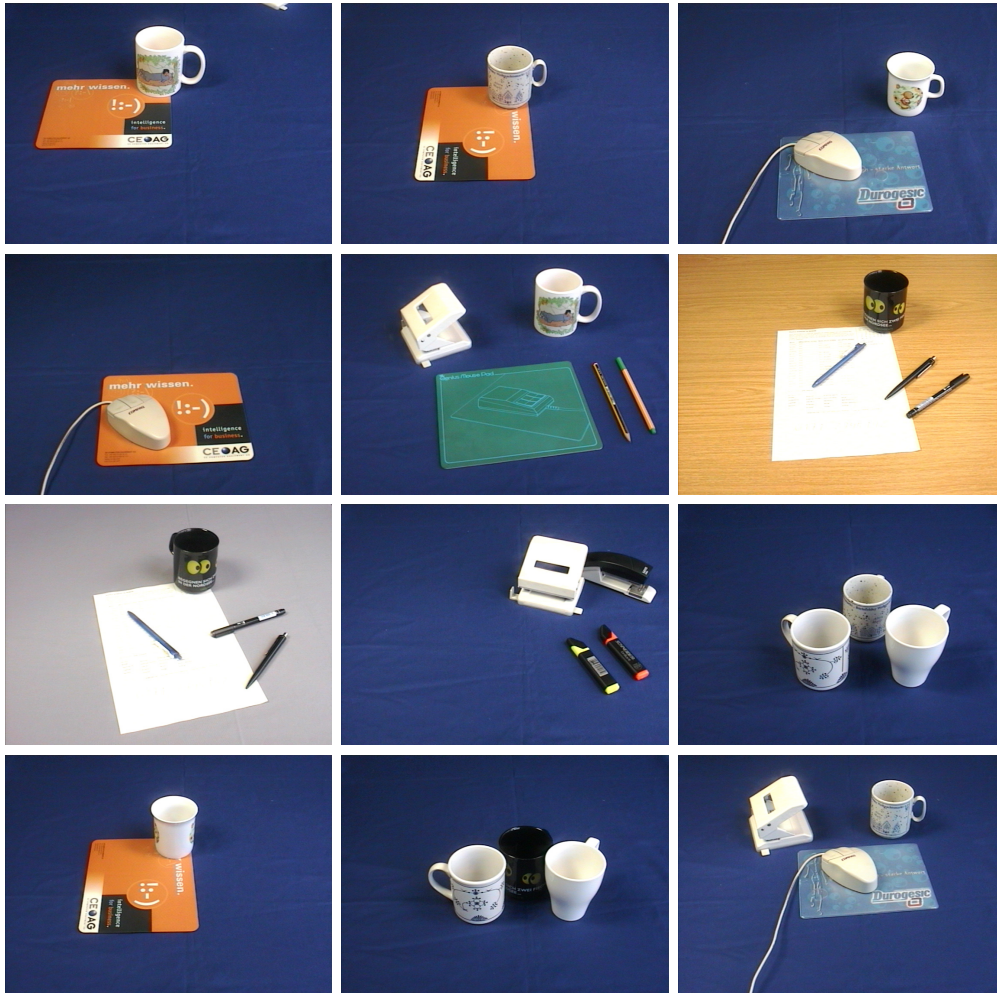


Figure C.2: Images containing office objects used for testing and evaluating the integrated system for office object categorization.

D Statistical Significance of Recognition Results

The significance a recognition rate that is determined based on a set of samples depends on the number of tested samples. Therefore, the isolated number for the recognition rate should be accomplished by a measurement of its significance.

Analyzing a recognition process by counting correct and false recognition results for a set of samples is from the statistical point of view like an experiment, where n independent trials are done, and for each trial the result is either 'success' or 'failure'.

Such experiments are described by the Binomial probability distribution [Krey 74] resulting in the probability of getting maximal i times the result 'success' given as:

$$P(X \leq i) = \sum_{j=0}^{j \leq i} P(X = j) = \sum_{j=0}^{j \leq i} \binom{n}{j} p^j \cdot (1-p)^{n-j}$$

where p is the probability for one isolated trial to be successful.

For the recognition process the probability p is estimated to be the recognition rate, $p = k/n$, where k is the number of correct recognition results on a sample of n cases at all.

The significance of the determined statistical value, here, the the recognition rate is usually given in the form of a confidence interval $[p_l, p_u]$. The borders of this interval are assigned to fulfill the specification that the real recognition rate lies with a given statistical certainty or confidence within this interval. It should be:

$$P(p_l \leq p \leq p_u) = \gamma$$

with γ is the confidence number, mostly chosen to be 0.95.

With $p_l = k_l/n$, k_l is the minimum number of successful trials and with $p_u = k_u/n$, k_u is the maximum number of successful trials that fulfill the requirement of the resulting rate lying within the confidence interval. Then, it is with using the general properties of discrete distributions:

$$P(X \geq k_l \wedge X \leq k_u) = P(X \leq k_u) - P(X \leq k_l) = \gamma$$

For calculating k_u and k_l , assuming symmetric characteristics results in:

$$P(X \leq k_l) = (1 - \gamma)/2 \quad P(X \leq k_u) = (1 + \gamma)/2$$

From these two equations, k_u and k_l can be calculated, if γ , p and n are given.

Instead of iteratively calculating the interval borders, [Krey 74] gives an analytic approximation, for the interval boundaries, valid for great n , with given c , k , and n :

D Statistical Significance of Recognition Results

$$p_l = \frac{2k + c^2}{2(n + c^2)} - \sqrt{\left(\frac{(2k + c^2)}{2(n + c^2)}\right)^2 - \frac{k^2}{n(n + c^2)}}$$

$$p_u = \frac{2k + c^2}{2(n + c^2)} + \sqrt{\left(\frac{(2k + c^2)}{2(n + c^2)}\right)^2 - \frac{k^2}{n(n + c^2)}}$$

The approximation for great n can be applied for $n > 100$ following [Krey 74]. Fig. D.1(a) shows results of the correct and approximated calculation for $n = 40$ in dependence of k using the commonly used confidence number of $\gamma = 0.95$. The differences are very small, so I take the approximated solutions for all calculations within this thesis.

Results for the approximated interval calculations for $n = 40$ and $n = 167$ in dependence of k , are given in Fig. D.1(b). As expected, if a testset of 167 samples is used, the confidence interval is much smaller than it is based on a test set of 40 samples. A recognition rate calculated based on the greater testset is more significant.

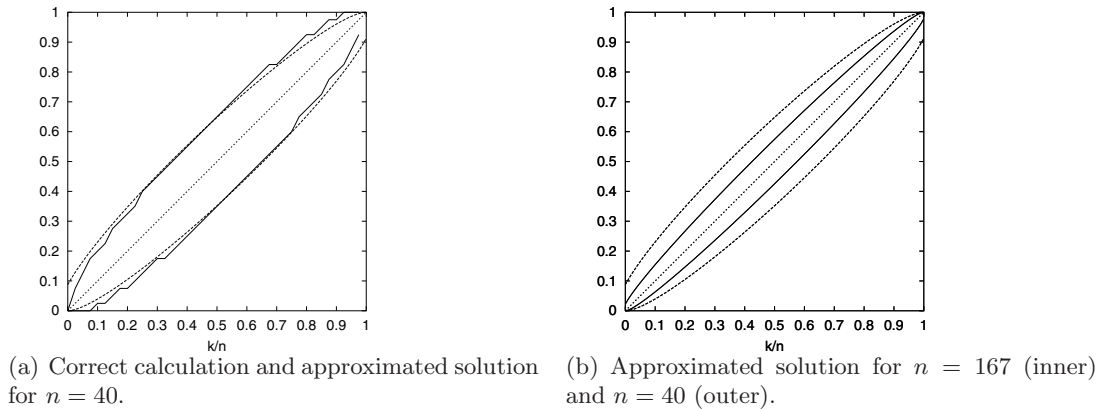


Figure D.1: Recognition rates and the corresponding band of significance calculated with confidence number $\gamma = 0.95$.

Bibliography

- [Abba 03] N. Abbadeni. "A New Similarity Matching Measure: Application to Texture-based Image Retrieval". In: *Proc. Int. Workshop on Texture Analysis and Synthesis*, pp. 1–6, Nice, 2003.
- [Adam 94] R. Adams and L. Bischof. "Seeded Region Growing". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 6, pp. 641–647, 1994.
- [Agar 04] S. Agarwal, A. Awan, and D. Roth. "Learning to detect objects in images via a sparse, part-based representation". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 11, pp. 1475–1490, 2004.
- [Alex 01] L. Alexandre, A. Campilho, and M. Kamel. "On combining classifiers using sum and product rules". *Pattern Recognition Letters*, Vol. 22, No. 12, pp. 1283–1289, 2001.
- [Ball 82] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, 1982.
- [Band 00] T. Bandlow, M. Klupsch, R. Hanek, and T. Schmitt. "Fast Image Segmentation, Object Recognition and Localization in a RoboCup Scenario". In: *RoboCup-99: Robot Soccer World Cup III*, pp. 174–185, Springer, 2000.
- [Barn 02a] K. Barnard, V. Cardei, and B. Funt. "A Comparison of Computational Colour Constancy Algorithms; Part One: Methodology and Experiments with Synthesized Data". *Trans. on Image Processing*, Vol. 11, No. 9, pp. 972–984, 2002.
- [Barn 02b] K. Barnard, L. Martin, A. Coath, and B. Funt. "A Comparison of Computational Colour Constancy Algorithms. Part Two. Experiments on Image Data". *Trans. on Image Processing*, Vol. 11, No. 9, pp. 985–996, 2002.
- [Bauc 02] C. Bauckhage. *A Structural Framework for Assembly Modeling and Recognition*. PhD thesis, Universität Bielefeld, Technische Fakultät, 2002.
- [Bauc 04a] C. Bauckhage, E. Braun, and G. Sagerer. "From Image Features to Symbols and Vice Versa – Using Graphs to Loop Data- and Model-Driven Processing in Visual Assembly Recognition". *Int. Journal of Pattern Recognition and Artificial Intelligence*, Vol. 18, No. 3, pp. 497–517, 2004.
- [Bauc 04b] C. Bauckhage, M. Hanheide, S. Wrede, and G. Sagerer. "A Cognitive Vision System for Action Recognition in Office Environments". In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 827–832, 2004.

Bibliography

- [Bauc 99] C. Bauckhage, J. Fritsch, F. Kummert, and G. Sagerer. "Towards a Vision System for Supervising Assembly Processes". In: *Proc. Symposium on Intelligent Robotic Systems (SIRS'99)*, pp. 89–98, Coimbra, 1999.
- [Belo 02] S. Belongie, J. Malik, and J. Puzicha. "Shape Matching and Object Recognition Using Shape Contexts". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 4, pp. 509–522, 2002.
- [Blum 67] H. Blum. "A transformation for extracting new descriptors of shape". In: W. Wathen-Dunn, Ed., *Models for the Perception of Speech and Visual Form*, pp. 362–380, MIT Press, Cambridge, MA, 1967.
- [Bobi 95] A. Bobick and C. Pinhanez. "Using approximate models as source of contextual information for vision processing". In: *Proc. ICCV Workshop on Context-Based Vision*, pp. 13–21, Cambridge, 1995.
- [Bore 02] E. Borenstein and S. Ullman. "Class-Specific, Top-Down Segmentation". In: *Proc. European Conf. on Computer Vision*, pp. 109–122, Copenhagen, 2002.
- [Bore 04] E. Borenstein, E. Sharon, and S. Ullman. "Combining Top-Down and Bottom-up Segmentation". In: *Proc. Workshop on Perceptual Organization in Computer Vision*, p. , Washington DC, 2004.
- [Bose 04] B. Bose and W. E. L. Grimson. "Improving Object Classification in Far-Field Video.". In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 181–188, 2004.
- [Bose 92] B. Boser, I. Guyon, and V. Vapnik. "A training algorithm for optimal margin classifiers". In: *Proc. Workshop on Computational Learning Theory*, pp. 144–152, Pittsburgh, 1992.
- [Bowyer 90] K. W. Bowyer and C. R. Dyer. "Aspect graphs: An introduction and survey of recent results". *Int. Journal of Imaging Systems and Technology*, Vol. 2, pp. 315–328, 1990.
- [Brac 85] R. Brachman and J. Schmolze. "An overview of the KL-ONE-knowledge representation language". *Cognitive Science*, Vol. 9, pp. 171–216, 1985.
- [Brau 01] E. Braun, J. Fritsch, and G. Sagerer. "Incorporating Process Knowledge into Object Recognition for Assemblies". In: *Proc. Int. Conf. on Computer Vision*, pp. 726–732, Vancouver, 2001.
- [Brau 98] C. G. Bräutigam. *A Model-Free Voting Approach to Cue Integration*. PhD thesis, Kungl Tekniska Högskolan, Stockholm, 1998.
- [Brau 99] E. Braun, G. Heidemann, H. Ritter, and G. Sagerer. "A Multi-directional Multiple Path Recognition Scheme for Complex Objects Applied to the Domain of a Wooden Toy Kit". *Pattern Recognition Letters*, Vol. 20, pp. 1085–1091, 1999.

- [Bruc 00] J. Bruce, T. Balch, and M. Veloso. "Fast and inexpensive color image segmentation for interactive robots". In: *Proc. Int. Conf. on Intelligent Robots and Systems*, pp. 2061–2066, 2000.
- [Bruc 75] V. Bruce and M. Morgan. "Violations of symmetry and repetition in visual patterns". *Perception*, Vol. 4, No. 3, pp. 239–249, 1975.
- [Camp 00] T. E. Campos, R. S. Feris, and R. M. Cesar Jr. "A Framework for Face Recognition from Video Sequences Using GWN and Eigenfeature Selection". In: *Workshop on Artificial Intelligence and Computer Vision*, 2000. electronic proceedings, <http://www.ime.usp.br/cesar/events/waicv00>.
- [Camp 97] N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, and T. Troscianko. "Interpreting Image Databases by Region Classification". *Pattern Recognition (Special Edition on Image Databases)*, Vol. 30, No. 4, pp. 555–563, April 1997.
- [Carb 04] P. Carbonetto, N. de Freitas, and K. Barnard. "A Statistical Model for General Contextual Object Recognition.". In: *Proc. European Conf. on Computer Vision*, pp. 350–362, 2004.
- [Chan 05] M. Chantler and L. V. Gool (editors). "Special Issue on Texture Analysis and Synthesis". *Int. Journal of Computer Vision*, Vol. 62, No. 1-2, 2005.
- [Chen 01] H.-D. Cheng, X. Jiang, Y. Sun, and J. Wang. "Color image segmentation: advances and prospects.". *Pattern Recognition*, Vol. 34, No. 12, pp. 2259–2281, 2001.
- [Chen 95] Y. Cheng. "Mean shift, mode seeking, and clustering". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8, pp. 790–799, 1995.
- [Chou 03] Y.-Y. Chou and L. G. Shapiro. "A hierarchical multiple classifier learning algorithm". *Pattern Analysis and Applications*, Vol. 6, No. 2, pp. 150–168, 2003.
- [Clau 04] D. A. Clausi and H. Deng. "Feature Fusion for Image Texture Segmentation.". In: *Proc. Int. Conf. on Pattern Recognition*, pp. 580–583, 2004.
- [Coma 02] D. Comaniciu and P. Meer. "Mean Shift: A Robust Approach toward Feature Space Analysis". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp. 603–619, 2002.
- [Coma 97] D. Comaniciu and P. Meer. "Robust Analysis of Feature Space: Color Image Segmentation". In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 750–755, 1997.
- [Coot 95] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. "Active shape models-their training and application". *Computer Vision and Image Understanding*, Vol. 61, No. 1, pp. 38–59, 1995.

Bibliography

- [Cox 96] I. Cox, Y. Zhong, and S. Rao. "Ratio regions: A technique for image segmentation." In: *Proc. Int. Conf. on Pattern Recognition*, pp. 557–564, 1996.
- [Cufi 02] X. Cufi, X. Munoz, J. Freixenet, and J. Marti. "A Review on Image Segmentation Techniques Integrating Region and Boundary Information". In: P. W. Hawkes, Ed., *Advances in Imaging and Electron Physics*, pp. 1–39, Academic Press, 2002.
- [Dick 99] S. J. Dickinson. "Object Representation and Recognition". In: E. Lepore and Z. Pylyshyn, Eds., *Rudgers University Lectures on Cognitive Science*, pp. 172–207, Basil Blackwell publishers, 1999.
- [Do 00] M. N. Do and M. Vetterli. "Texture Similarity Measurement Using Kullback-Leibler Distance on Wavelet Subbands". In: *Proceedings International Conference on Image Processing*, pp. 730–733, 2000.
- [DZmu 86] M. D'Zmura and P. Lennie. "Mechanisms of color constancy". *Journal of the Optical Society of America*, Vol. 3, No. 10, pp. 1662–1672, 1986.
- [Fash 05] M. Fashing and C. Tomasi. "Mean Shift is a Bound Optimization". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 3, pp. 471–474, 2005.
- [Faym 99] J. A. Fayman, P. Pirjanian, H. Christensen, and E. Rivlin. "Exploiting Process Integration and Composition in the Context of Active Vision". *IEEE Transactions on Systems, Man, and Cybernetics. Part C: Applications and Reviews*, Vol. 29, No. 1, pp. 73–86, 1999.
- [Feld 74] J. A. Feldman and Y. Yakimovsky. "Decision Theory and Artificial Intelligence: I. A Semantics-Based Region Analyzer". *Artificial Intelligence*, Vol. 5, pp. 349–371, 1974.
- [Felz 98] P. F. Felzenszwalb and D. P. Huttenlocher. "Image Segmentation Using Local Variation". In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 98–104, 1998.
- [Fink 04a] G. A. Fink, J. Fritsch, S. Hohenner, M. Kleinhagenbrock, S. Lang, and G. Sagerer. "Towards Multi-Modal Interaction with a Mobile Robot". *Pattern Recognition and Image Analysis*, Vol. 14, No. 2, pp. 173–184, 2004.
- [Fink 04b] M. Fink and P. Perona. "Mutual Boosting for Contextual Inference." In: S. Thrun, L. Saul, and B. Schölkopf, Eds., *Advances in Neural Information Processing Systems 16*, MIT Press, Cambridge, MA, 2004.
- [Fink 95] G. A. Fink, N. Jungclaus, H. Ritter, and G. Sagerer. "A Communication Framework for Heterogeneous Distributed Pattern Analysis". In: *Int. Conf. on Algorithms And Architectures for Parallel Processing*, pp. 881–890, Brisbane, 1995.

- [Fole 93] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice*. Addison Wesley, 1993.
- [Frei 02] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cufi. "Yet Another Survey on Image Segmentation: Region and Boundary Information Integration". In: *Proc. European Conf. on Computer Vision*, pp. 408–422, Copenhagen, 2002.
- [Frit 03] J. Fritsch. *Vision-based Recognition of Gestures with Context*. PhD thesis, Universität Bielefeld, Technische Fakultät, 2003.
- [Fu 81] K. Fu and J. Mui. "A Survey on Image Segmentation". *Pattern Recognition*, Vol. 13, No. 1, pp. 3–16, 1981.
- [Fuka 80] Y. Fukada. "Spatial Clustering Procedures for Region Analysis". *Pattern Recognition*, Vol. 12, pp. 395–403, 1980.
- [Fuku 75] K. Fukunaga and L. Hostetler. "The Estimation of the Gradient of a Density Function". *Trans. on Information Theory*, Vol. 21, pp. 32–40, 1975.
- [Funt 98] B. Funt, K. Barnard, and L. Martin. "Is Colour Constancy Good Enough?". In: *Proc. European Conf. on Computer Vision*, pp. 445–459, 1998.
- [Gibs 87] J. J. Gibson and B. Bridgeman. "The visual perception of surface texture in photographs". *Psychological Research*, Vol. 49, No. 1, pp. 1–5, 1987.
- [Gigu 91] Z. Gigus, J. Canny, and R. Seidel. "Efficiently Computing and Representing Aspect Graphs of Polyhedral Objects". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, pp. 542–551, 1991.
- [Gold 96] S. Gold and A. Rangarajan. "A Graduated Assignment Algorithm for Graph Matching". *PAMI*, Vol. 18, No. 4, pp. 377–388, 1996.
- [Gonz 87] R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison-Wesley Publishing Company, Inc., 1987.
- [Haas 04] A. Haasch, S. Hohenner, S. Hüwel, M. Kleinhagenbrock, S. Lang, I. Toptsis, G. A. Fink, J. Fritsch, B. Wrede, and G. Sagerer. "BIRON – The Bielefeld Robot Companion". In: E. Prassler, G. Lawitzky, P. Fiorini, and M. Hägele, Eds., *Proc. Int. Workshop on Advances in Service Robotics*, pp. 27–32, Fraunhofer IRB Verlag, Stuttgart, Germany, May 2004.
- [Hanh 04] M. Hanheide, C. Bauckhage, and G. Sagerer. "Memory Consistency Validation in a Cognitive Vision System". In: *Proc. Int. Conf. on Pattern Recognition*, pp. 459–462, IEEE, 2004.
- [Hans 78] A. Hanson and E. Riseman. "Visions: a computer system for interpreting scenes". In: A. Hanson and E. Wiseman, Eds., *Computer Vision Systems*, pp. 303–333, Academic Press, New York, 1978.

Bibliography

- [Hara 79] R. Haralick. "Statistical and Structural Approaches to Texture". *Proceedings of IEEE*, Vol. 67, No. 5, pp. 786–804, May 1979.
- [Haym 02] E. Hayman and J.-O. Eklundh. "Probabilistic and Voting Approaches to Cue Integration for Figure-Ground Segmentation". In: *Proc. European Conf. on Computer Vision*, pp. 469–486, Copenhagen, 2002.
- [Heid 00] G. Heidemann, D. Lücke, and H. Ritter. "Segmentation of Partially Occluded Objects by Local Classification". In: *Proc. Int. Joint Conf. on Neural Networks*, pp. 152–157, Como, 2000.
- [Heid 04] G. Heidemann. "Focus-of-Attention from Local Color Symmetries". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 7, pp. 817–830, 2004.
- [Heid 05] G. Heidemann, H. Bekel, I. Bax, and H. Ritter. "Interactive Online Learning". *Pattern Recognition and Image Analysis*, Vol. 15, No. 1, pp. 55–58, 2005.
- [Heid 96a] G. Heidemann, F. Kummert, H. Ritter, and G. Sagerer. "A Hybrid Object Recognition Architecture". In: C. von der Malsburg, W. von Seelen, J. Vorbrüggen, and B. Sendhoff, Eds., *Artificial Neural Networks – ICANN 96*, 16.-19. July, pp. 305–310, Springer-Verlag, Berlin, 1996.
- [Heid 96b] G. Heidemann and H. Ritter. "A Neural Recognition Architecture for Composed Objects". In: B. Jähne, P. Geißler, H. Haußecker, and F. Hering, Eds., *Mustererkennung 1996*, 18. DAGM-Symposium, pp. 475–482, Springer Verlag, 1996.
- [Heid 98] G. Heidemann. *Ein flexibel einsetzbares Objekterkennungssystem auf der Basis neuronaler Netze*. PhD thesis, Universität Bielefeld, Technische Fakultät, 1998.
- [Heid 99] G. Heidemann and H. Ritter. "Combining Multiple Neural Nets for Visual Feature Selection and Classification". In: *Proc. Int. Conf. on Artificial Neural Networks*, pp. 365–370, 1999.
- [Hori 04] Y. Horikawa. "Comparison of Support Vector Machines with Autocorrelation Kernels for Invariant Texture Classification.". In: *Proc. Int. Conf. on Pattern Recognition*, pp. 660–663, 2004.
- [Huec 73] M. H. Hueckel. "A Local Visual Operator Which Recognizes Edges and Lines". *Journal of the ACM*, Vol. 20, No. 4, pp. 634–647, 1973.
- [Imai 94] M. Imai, D. Gentner, and N. Uchida. "Children's theories of word meaning: The role of shape similarity in early acquisition". *Cognitive Development*, Vol. 9, pp. 45–75, 1994.
- [Jain 00] A. K. Jain, R. P. W. Duin, and J. Mao. "Statistical Pattern Recognition: A Review". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, pp. 4–37, 2000.

- [Jain 99] A. K. Jain, M. N. Murty, and P. J. Flynn. "Data clustering: a review". *ACM Computing Surveys*, Vol. 31, No. 3, pp. 264–323, 1999.
- [Jens 96] F. Jensen. *An Introduction to Bayesian Networks*. Springer, London, 1996.
- [Jone 02] M. J. Jones and J. M. Rehg. "Statistical color models with application to skin detection". *Int. Journal of Computer Vision*, Vol. 46, No. 1, pp. 81–96, 2002.
- [Jord 99] M. I. Jordan, Ed. *Learning in Graphical Models*. MIT Press, Cambridge, MA, USA, 1999.
- [Jule 65] B. Julesz. "Texture and visual perception". *Scientific American*, Vol. 212, No. 2, pp. 38–49, 1965.
- [Jule 81] B. Julesz. "Textons, the elements of texture perception and their interactions". *Nature*, Vol. 290, pp. 89–97, 1981.
- [Jung 98] N. Jungclaus. *Integration verteilter Systeme zur Mensch-Maschine-Kommunikation*. PhD thesis, Universität Bielefeld, Technische Fakultät, 1998.
- [Kais 96] P. K. Kaiser and R. M. Boynton. *Human Color Vision*. Optical Society of America, 1996.
- [Kali 96] T. Kalinke and W. von Seelen. "Entropie als Maß des lokalen Informationsgehalts in Bildern zur Realisierung einer Aufmerksamkeitssteuerung". In: *Pattern Recognition, Proc. DAGM Symposium*, pp. 627–634, 1996.
- [Kass 88] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes: Active Contour Models". *Int. Journal of Computer Vision*, Vol. 1, No. 4, pp. 321–331, 1988.
- [Kend 99] D. Kendall, D. Barden, T. Carne, and H. Le. *Shape and Shape Theory*. John Wiley & Sons, Inc., 1999.
- [Kese 01a] Y. Keselman and S. Dickinson. "Bridging the representational gap between models and exemplars". In: *Workshop on Models versus Exemplars in Computer Vision*, Kauai, 2001.
- [Kese 01b] Y. Keselman and S. Dickinson. "Generic Model Abstraction from Examples". In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 856–863, Kauai, 2001.
- [Kese 05] Y. Keselman and S. Dickinson. "Generic Model Abstraction from Examples". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 7, pp. 1141–1156, 2005.
- [Khaw 96] K. W. Khawaja, A. A. Maciejewski, D. Tretter, and C. A. Bouman. "A multi-scale assembly inspection algorithm". *Robotics and Automation Magazine*, Vol. 3, No. 2, pp. 15–22, 1996.

Bibliography

- [Kitt 98] J. Kittler. "On combining classifiers". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp. 226–239, 1998.
- [Kolo 04] I. Kolonias, W. J. Christmas, and J. Kittler. "Use of Context in Automatic Annotation of Sports Videos." In: *Proc. Iberoamerican Congress on Pattern Recognition*, pp. 1–12, Springer, 2004.
- [Krey 74] E. Kreyszig. *Statistische Methoden und ihre Anwendung*. Vandenhoeck & Ruprecht, Göttingen, 1974.
- [Kubo 00] M. Kubovy and S. Gepshtein. "Gestalt: From phenomena to laws". In: K. L. Boyer and S. Sarkar, Eds., *Perceptual Organization for Artificial Vision Systems*, pp. 41–71, Kluwer Academic Publishers, Boston, March 2000.
- [Kumm 98] F. Kummert, G. A. Fink, G. Sagerer, and E. Braun. "Hybrid Object Recognition in Image Sequences". In: *Proc. Int. Conf. on Pattern Recognition*, pp. 1165–1170, Brisbane, 1998.
- [Lade 93] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen. "Distortion Invariant Object Recognition in the Dynamic Link Architecture". *Trans. on Computers*, Vol. 42, pp. 300–311, 1993.
- [Lako 87] G. Lakoff. *Women, fire, and other dangerous things: What categories reveal about the mind*. University of Chicago Press, Chicago, 1987.
- [Lanz 03] P. L. Lanzi and R. L. Riolo. "Recent trends in learning classifier systems research". In: *Advances in evolutionary computing: theory and applications*, pp. 955–988, Springer, New York, 2003.
- [Leib 04a] B. Leibe, A. Leonardis, and B. Schiele. "Combined Object Categorization and Segmentation with an Implicit Shape Model". In: *Proc. Workshop on Statistical Learning in Computer Vision*, p. , Prague, 2004.
- [Leib 04b] B. Leibe and B. Schiele. "Scale Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search". In: *Pattern Recognition, Proc. DAGM Symposium*, pp. 145–153, 2004.
- [Leon 93] A. Leonardis. *Image Analysis Using Parametric Models*. PhD thesis, University of Ljubljana, 1993.
- [Lind 80] Y. Linde, A. Buzo, and R. M. Gray. "An Algorithm for Vector Quantizer Design". *Trans. on Communications*, Vol. COM-28, No. 1, pp. 84–95, 1980.
- [Liu 01] L. Liu and S. Sclaroff. "Region Segmentation via Deformable Model-Guided Split and Merge." In: *Int. Conf. on Computer Vision*, pp. 98–104, 2001.
- [Loch 87] P. Locher and C. Nodine. "Symmetry catches the eye". In: A. Lévy-Schoen and J. O'Regan, Eds., *Eye Movements: From Physiology to Cognition*, pp. 353–361, North Holland Elsevier Science Publisher B.V., 1987.

- [Lour 98] T. Lourens and R. P. Würtz. "Object Recognition by matching symbolic edge graphs". In: *Proc. Asian Conf. on Computer Vision*, pp. 193–200, 1998.
- [Lowe 04] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". *Int. Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.
- [Lowe 85] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.
- [Luo 05] J. Luo, A. E. Savakis, and A. Singhal. "A Bayesian network-based framework for semantic image understanding.". *Pattern Recognition*, Vol. 38, No. 6, pp. 919–934, 2005.
- [Makr 05] S. Makrogiannis, G. Economou, S. Fotopoulos, and G. Bourbakis. "Segmentation of Color Images Using Multiscale Clustering and Graph Theoretic Region Synthesis". *Trans. on Systems, Man, and Cybernetics - Part A*, Vol. 35, No. 2, pp. 224–238, 2005.
- [Mali 01] J. Malik, S. Belongie, T. Leung, and J. Shi. "Contour and Texture Analysis for Image Segmentation". *Int. Journal of Computer Vision*, Vol. 43, No. 1, pp. 7–27, 2001.
- [Mali 87] J. Malik. "Interpreting Line Drawings of Curved Objects". *Int. Journal of Computer Vision*, Vol. 1, No. 1, pp. 73–104, 1987.
- [Mard 97] K. V. Mardia, W. Qian, and K. M. A. de Souza. "Deformable Template Recognition of Multiple Occluded Objects.". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 9, pp. 1035–1042, 1997.
- [Marr 80] D. Marr and E. C. Hildreth. "Theory of edge detection". In: *Proc. of Royal Society London*, pp. 187–217, 1980.
- [Mart 01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics". In: *Int. Conf. on Computer Vision*, pp. 416–423, July 2001.
- [Mart 04] D. Martin, C. Fowlkes, and J. Malik. "Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 5, pp. 530–549, 2004.
- [Mass 95] A. Maßmann and S. Posch. "Mask-Oriented Grouping Operations in a Contour-Based Approach". In: *Proc. Asian Conf. on Computer Vision*, pp. 58–61, Singapore, 1995.
- [McGu 02] G. McGunnigle and M. Chantler. "Comparison of three rough surface classifiers". *Vision, Image and Signal Processing*, Vol. 149, No. 5, pp. 263–271, 2002.

Bibliography

- [McKe 98] S. McKenna, Y. Raja, and S. Gong. "Object tracking using adaptive colour mixture models". In: *Proc. Asian Conf. on Computer Vision*, pp. 615–622, Hongkong, 1998.
- [Menz 99] M. Menzel. *Untersuchung verschiedener Abstimmverfahren bei der datengetriebenen Objekterkennung*. Master's thesis, Universität Bielefeld, Technische Fakultät, 1999.
- [Moha 01] A. Mohan, C. Papageorgiou, and T. Poggio. "Example-Based Object Detection in Images by Components". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 4, pp. 349–361, 2001.
- [Mohr 97] R. Mohr, S. Picard, and C. Schmid. "Bayesian Decision versus Voting for Image Retrieval". In: *Proc. Int. Conf. on Computer Analysis of Images and Patterns*, pp. 376–383, Kiel, 1997.
- [Moor 99] D. J. Moore, I. A. Essa, and M. H. Hayes. "Exploiting Human Actions and Object Context for Recognition Tasks.". In: *Int. Conf. on Computer Vision*, pp. 80–86, 1999.
- [Mura 95] H. Murase and S. K. Nayar. "Visual learning and recognition of 3-D objects from appearance". *Int. Journal of Computer Vision*, Vol. 14, No. 1, pp. 5–24, 1995.
- [Murp 04] K. Murphy, A. Torralba, and W. T. Freeman. "Using the Forest to See the Trees: A Graphical Model Relating Features, objects, and Scenes". In: S. Thrun, L. Saul, and B. Schölkopf, Eds., *Advances in Neural Information Processing Systems 16*, MIT Press, Cambridge, MA, 2004.
- [Nene 96] S. A. Nene, S. K. Nayar, and H. Murase. "Columbia Object Image Library (COIL-100)". Tech. Rep. CUCS-006-96, Department of Computer Science, Columbia University, 1996.
- [Neum 56] J. von Neumann. "Probabilistic logics and the synthesis of reliable organisms from unreliable components". *Automata Studies (Annals of Mathematics Studies)*, pp. 43–98, 1956.
- [Niem 90] H. Niemann, G. Sagerer, S. Schröder, and F. Kummert. "ERNEST: A Semantic Network System for Pattern Understanding". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 9, pp. 883–905, 1990.
- [Opit 99] D. Opitz and R. Maclin. "Popular Ensemble Methods: An Empirical Study". *Journal of Artificial Intelligence Research*, Vol. 11, pp. 169–198, 1999.
- [Oza 05] N. C. Oza, R. Polikar, J. Kittler, and F. Roli, Eds. *Multiple Classifier Systems, 6th International Workshop, MCS 2005, Seaside, CA, USA, June 13-15, 2005, Proceedings*, Springer, 2005.
- [Pal 93] N. R. Pal and S. K. Pal. "A Review On Image Segmentation Techniques". *Pattern Recognition*, Vol. 26, pp. 1277–1294, 1993.

- [Parh 94] B. Parhami. "Voting Algorithms". *Trans. on Reliability*, Vol. 43, No. 4, pp. 617–629, 1994.
- [Pear 88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.
- [Pich 96] O. Pichler, A. Teuner, and B. Hosticka. "A Comparison of Texture Feature-Extraction Using Adaptive Gabor Filtering, Pyramidal and Tree-Structured Wavelet Transforms". *Pattern Recognition*, Vol. 29, No. 5, pp. 733–742, 1996.
- [Plat 00] K. N. Plataniotis and A. N. Venetsanopoulos. *Color image processing and applications*. Springer, New York, 2000.
- [Raja 98a] Y. Raja, S. McKenna, and S. Gong. "Colour Model Selection and Adaptation in Dynamic Scenes". In: *Proc. European Conf. on Computer Vision*, pp. 460–474, 1998.
- [Raja 98b] Y. Raja, S. McKenna, and S. Gong. "Segmentation and tracking using colour mixture models". In: *Proc. Asian Conf. on Computer Vision*, pp. 607–614, Hongkong, 1998.
- [Rand 99] T. Randen and J. Husoy. "Filtering for texture classification". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 4, pp. 291–310, 1999.
- [Rehr 97] V. Rehrmann and L. Pries. "Fast and Robust Segmentation of Natural Color Scenes". Tech. Rep., Computer Science Department, University of Koblenz-Landau, 1997.
- [Rehr 98] V. Rehrmann and L. Pries. "Fast and Robust Segmentation of Natural Color Scenes". In: *Proc. Asian Conf. on Computer Vision*, pp. 598–606, Hongkong, 1998.
- [Reis 95] D. Reisfeld, H. Wolfson, and Y. Yeshurun. "Context-Free Attentional Operators: The Generalized Symmetry Transform". *Int. Journal of Computer Vision*, Vol. 14, pp. 119–130, 1995.
- [Rick 96] G. Rickheit and I. Wachsmuth. "Collaborative Research Centre "Situated Artificial Communicators" at the University of Bielefeld, Germany.". *Artificial Intelligence Review*, Vol. 10, No. 3-4, pp. 165–170, 1996.
- [Rivl 95] E. Rivlin, S. J. Dickinson, and A. Rosenfeld. "Recognition by functional parts". *Computer Vision Image Understanding*, Vol. 62, No. 2, pp. 164–176, 1995.
- [Robe 97] S. Roberts. "Parametric and Non-Parametric Unsupervised Cluster Analysis". *Pattern Recognition*, Vol. 30, No. 5, pp. 833–839, 1997.
- [Rosi 89] P. L. Rosin and G. A. West. "Segmentation of edges into lines and arcs". *Image and Vision Computing*, Vol. 7, No. 2, pp. 109–114, 1989.

Bibliography

- [Sage 01] G. Sagerer, C. Bauckhage, E. Braun, G. Heidemann, F. Kummert, H. Ritter, and D. Schlüter. "Integrating Recognition Paradigms in a Multiple-path Architecture". In: *Int. Conf. on Advances in Pattern Recognition*, pp. 202–211, Springer, Rio de Janeiro, 2001.
- [Sage 02] G. Sagerer, C. Bauckhage, E. Braun, J. Fritsch, F. Kummert, F. Lömker, and S. Wachsmuth. "Structure and Process: Learning of Visual Models and Construction Plans for Complex Objects". In: G. Hager, H. Christensen, H. Bunke, and R. Klein, Eds., *Sensor Based Intelligent Robots*, pp. 317–344, Dagstuhl Castle, Germany, 2002.
- [Sage 97] G. Sagerer and H. Niemann. *Semantic Networks for Understanding Scenes. Advances in Computer Vision and Machine Intelligence*, Plenum Publishing Corporation, New York, 1997.
- [Sano 98] T. Sanocki, K. W. Bowyer, M. D. Heath, and S. Sarkar. "Are Edges Sufficient for Object Recognition". *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 24, No. 1, pp. 1–10, 1998.
- [Sark 94] S. Sarkar and K. L. Boyer. *Computing Perceptual Organization in Computer Vision*. World Scientific Publishing, River Edge, 1994.
- [Saun 03] J. A. Saunders. "The effect of texture relief on perception of slant from texture". *Perception*, Vol. 32, No. 2, pp. 211–233, 2003.
- [Schi 96] B. Schiele and J. L. Crowley. "Object Recognition Using Multidimensional Receptive Field Histograms". In: *Proc. European Conf. on Computer Vision*, pp. 610–619, 1996.
- [Schl 00] D. Schlüter, F. Kummert, G. Sagerer, and S. Posch. "Integration of regions and contours for object recognition". In: *Proc. Int. Conf. on Pattern Recognition*, pp. 944–947, Barcelona, 2000.
- [Schl 01] D. Schlüter. *Hierarchisches Perzeptives Gruppieren mit Integration dualer Bildbeschreibungen*. PhD thesis, Universität Bielefeld, Technische Fakultät, 2001.
- [Schl 98] D. Schlüter and S. Posch. "Combining Contour and Region Information for Perceptual Grouping". In: *Pattern Recognition, Proc. DAGM Symposium*, pp. 393–401, 1998.
- [Schm 97] C. Schmid and R. Mohr. "Local Grayvalue Invariants for Image Retrieval". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 5, pp. 530–534, May 1997.
- [Schm 98] C. Schmid, R. Mohr, and C. Bauckhage. "Comparing and evaluating interest points". In: *Int. Conf. on Computer Vision*, pp. 230–235, Bombay, 1998.

- [Seba 04] T. B. Sebastian, P. N. Klein, and B. B. Kimia. "Recognition of Shapes by Editing Their Shock Graphs". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 5, pp. 550–571, 2004.
- [Shan 48] C. E. Shannon. "A mathematical theory of communication". *Bell Systems Technical Journal*, Vol. 27, pp. 379–423, 1948.
- [Shar 00] E. Sharon, A. Brandt, and R. Basri. "Fast Multiscale Image Segmentation". In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 70–77, 2000.
- [Shi 00] J. Shi and J. Malik. "Normalized Cuts and Image Segmentation". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 888–905, 2000.
- [Sidd 99] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. "Shock Graphs and shape matching". *Int. Journal of Computer Vision*, Vol. 35, No. 1, pp. 13–32, 1999.
- [Sing 03] A. Singhal, J. Luo, and W. Zhu. "Probabilistic spatial context models for scene content understanding". In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 235–241, 2003.
- [Skar 94] W. Skarbeck and A. Koschan. "Colour Image Segmentation - A Survey". Tech. Rep. 94-32, TU Berlin, 1994.
- [Smeu 00] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. "Content-Based Image Retrieval at the End of the Early Years". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 12, pp. 1349–1380, 2000.
- [Sori 00] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen. "Skin Detection in Video Under Changing Illumination Conditions". In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 839–842, 2000.
- [Stos 04] D. Stöbel, M. Hanheide, G. Sagerer, L. Krüger, and M. Ellenrieder. "Feature and Viewpoint Selection for Industrial Car Assembly". In: *Pattern Recognition, Proc. DAGM Symposium*, pp. 528–535, 2004.
- [Stra 91] T. Strat and M. Fischler. "Context-Based Vision: Recognizing Objects Using Information from Both 2-D and 3-D Imagery". *Pattern Analysis and Machine Vision*, Vol. 13, No. 10, pp. 1050–1065, 1991.
- [Supe 04] B. J. Super. "Fast Correspondence-based System for Shape Retrieval". *Pattern Recognition Letters*, Vol. 25, No. 2, pp. 217–225, 2004.
- [Swai 91] M. J. Swain and D. H. Ballard. "Color indexing". *Int. Journal of Computer Vision*, Vol. 7, No. 1, pp. 11–32, 1991.
- [Teag 80] M. Teague. "Image analysis via the general theory of moments". *Journal of the optical society of America*, Vol. 70, No. 8, pp. 920–930, 1980.

Bibliography

- [Torr 03] A. Torralba. "Contextual Priming for Object Detection". *Int. Journal of Computer Vision*, Vol. 53, No. 2, pp. 169–191, 2003.
- [Torr 05] A. Torralba, K. P. Murphy, and W. T. Freeman. "Contextual Models for Object Detection Using Boosted Random Fields". In: L. K. Saul, Y. Weiss, and L. Bottou, Eds., *Advances in Neural Information Processing Systems 17*, pp. 1401–1408, MIT Press, Cambridge, MA, 2005.
- [Tous 78] G. T. Toussaint. "The use of context in pattern recognition". *Pattern Recognition*, Vol. 10, No. 1, pp. 189–204, 1978.
- [Turk 91] M. Turk and A. Pentland. "Eigenfaces for recognition". *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71–86, 1991.
- [Turt 03] M. Turtinen and M. Pietikinen. "Visual Training and Classification of Textured Outdoor Scene Images". In: *Proc. Int. Workshop on Texture Analysis and Synthesis*, pp. 101–106, Nice, 2003.
- [Ullm 01] S. Ullman, E. Sali, and M. Vidal-Naquet. "A Fragment-Based Approach to Object Representation and Classification". In: *Proc. Int. Workshop on Visual Form*, pp. 85–102, 2001.
- [Ullm 91] S. Ullman and R. Basri. "Recognition by Linear Combinations of Models". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 10, pp. 992–1006, 1991.
- [Viol 04] P. Viola and M. J. Jones. "Robust Real-Time Face Detection". *Int. Journal of Computer Vision*, Vol. 57, No. 2, pp. 137–154, 2004.
- [Wach 01] S. Wachsmuth. *Multi-modal Scene Understanding Using Probabilistic Models*. PhD thesis, Universität Bielefeld, Technische Fakultät, 2001.
- [Wand 95] B. A. Wandell. *Foundations of Vision*. Sinauer Associates, Sunderland, Massachusetts, 1995.
- [Webe 00] M. Weber, M. Welling, and P. Perona. "Unsupervised Learning of Models for Recognition". In: *Proc. European Conf. on Computer Vision*, pp. 18–32, 2000.
- [Webs 96] M. A. Webster. "Human Colour Perception and its Adaptation". *Network: Computation in Neural Systems*, Vol. 7, No. 4, pp. 587–634, 1996.
- [Weig 04] T. Weigel, D. Zhang, K. Rechert, and B. Nebel. "Adaptive Vision for Playing Table Soccer". In: *Proc. German Conf. on Artificial Intelligence*, pp. 424–438, Ulm, 2004.
- [Wert 23] M. Wertheimer. "Untersuchungen zur Lehre von der Gestalt II". *Psychologische Forschung*, Vol. 4, pp. 301–350, 1923.
- [Wils 88] R. Wilson and M. Spann. *Image Segmentation and Uncertainty*. Research Studies Press Ltd., Letchworth, Hertfordshire, England, 1988.

- [Wisk 97] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. "Face Recognition by Elastic Bunch Graph Matching". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 775–779, 1997.
- [Wong 92] G. Wong and H. Frei. "Object Recognition: the Utopian Method is Dead; the Time for Combining Simple Methods Has Come". In: *Proc. Int. Conf. on Pattern Recognition*, pp. 185–188, 1992.
- [Wu 93] Z. Wu and R. Leahy. "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation". *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11, pp. 1101–1113, 1993.
- [Wysz 82] G. Wyszecki and W. S. Stiles. *Color science : concepts and methods, quantitative data and formulae*. Wiley, 1982.
- [Xu 03] Y. Xu, P. Duygulu, E. Saber, A. M. Tekalp, and F. T. Yarman-Vural. "Object-based image labeling through learning-by-example and multi-level segmentation". *Pattern Recognition*, Vol. 36, No. 6, pp. 1407–1423, 2003.
- [Xu 92] L. Xu, A. Krzyzak, and C. Suen. "Methods for combining multiple classifiers and their applications in handwritten character recognition". *Trans. on Systems, Man, and Cybernetics*, Vol. 22, pp. 418–435, 1992.
- [Yu 02] S. X. Yu, R. Gross, and J. Shi. "Concurrent Object Recognition and Segmentation by Graph Partitioning". In: S. T. S. Becker and K. Obermayer, Eds., *Advances in Neural Information Processing Systems 15*, pp. 1383–1390, MIT Press, Cambridge, MA, 2002.
- [Yu 03] S. X. Yu and J. Shi. "Object-Specific Figure-Ground Segregation". In: *Int. Conf. on Computer Vision and Pattern Recognition*, p. , Madison, 2003.
- [Yuil 92] A. Yuille, D. Cohen, and P. Hallinan. "Feature Extraction from Faces Using Deformable Templates". *Int. Journal of Computer Vision*, Vol. 8, No. 2, pp. 99–111, 1992.
- [Zhan 01a] D. S. Zhang and G. Lu. "A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures". In: *Proc. Int. Conf. on intelligent multimedia and distance education*, pp. 1–9, Fargo, 2001.
- [Zhan 01b] D. S. Zhang and G. Lu. "Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study". In: *Proc. IEEE International Conference on Multimedia & Expo*, Tokyo, 2001.
- [Zhan 02a] D. S. Zhang and G. Lu. "A Comparative Study of Fourier Descriptors for Shape Representation and Retrieval". In: *Proc. Asian Conf. on Computer Vision*, Melbourne, 2002.
- [Zhan 02b] J. Zhang and T. Tan. "Brief review of invariant texture analysis methods". *Pattern Recognition*, Vol. 35, No. 3, pp. 735–747, 2002.

Bibliography

- [Zhan 04] D. S. Zhang and G. Lu. "Review of Shape Representation and Description Techniques". *Pattern Recognition*, Vol. 37, No. 1, pp. 1–19, 2004.
- [Zhu 96] S. Zhu and A. Yuille. "FORMS: a flexible object recognition and modelling system". *Int. Journal of Computer Vision*, Vol. 20, No. 3, pp. 187–212, 1996.
- [Zuck 76] S. Zucker. "Region Growing : Childhood and Adolescence". *Computer Graphics and Image Processing*, Vol. 5, pp. 382–399, 1976.