

Single-Image Inverse Lighting of Faces with a Virtual Light Stage

DISSERTATION

zur Erlangung des Grades eines Doktors
der Ingenieurwissenschaften

vorgelegt von

M.Sc. Davoud Shahlaei

bei der Naturwissenschaftlich-Technischen Fakultät

der Universität Siegen

Siegen 2017

gedruckt auf alterungsbeständigem holz- und säurefreiem Papier

Betreuer und erster Gutachter

Prof. Dr. Volker Blanz

Universität Siegen

Zweiter Gutachter

Prof. Dr. Martin Fuchs

Hochschule der Medien Stuttgart

Tag der mündlichen Prüfung

12. März 2018

Abstract

This dissertation addresses the problem of inverse lighting from a single image of a face. No information is given about the face or the imaging conditions, yet, the goal is to estimate a physically plausible lighting that reproduces plain and harsh illumination effects with respect to the appearance of the face in the given image. First, a 3D Morphable Model is fit to the 2D input face. Then, a generating set of images is rendered under all the same conditions as the input image, but different lights. Each image is rendered under a single light source with unit intensity. The light sources build a fixed set that is called a Virtual Light Stage in this dissertation. We assume that the input image belongs to the *synthetic* illumination cone that this generating set spans. We estimate the coefficients, so that the linear combination of the generating set is as similar as possible to the input image. To aim for more realistic illumination effects, this thesis uses a non-Lambertian reflectance that considers Fresnel specular highlights at grazing angles. Analysis and synthesis of cast shadows under complex lighting conditions is another important subject of the thesis. For the parameter estimation, two probabilistic modeling approaches are proposed. A hierarchical Bayesian model automatically suppresses inconsistencies between the generative model and the input. The nonnegative optimization algorithm finds the optimal spectral intensities of the Virtual Light Stage light sources for the input face. To enhance the performance of the algorithm on complex illumination effects, such as cast shadows, the hyperparameters of the hierarchical approach are controlled by constraints. This dissertation is a contribution to single image face and environment modeling and analysis with applications in realistic scene reconstruction, intrinsic face model decomposition, relighting and lighting design.

Zusammenfassung

Im Mittelpunkt der vorliegenden Arbeit steht das Thema der inversen Beleuchtung eines Einzelbildes von einem Gesicht. Darüber hinaus sind keine Informationen über das Gesicht oder die Abbildungsbedingungen vorhanden. Somit muss eine physikalisch plausible Beleuchtung geschätzt werden, die sowohl einfache als auch harte Lichteffekte in Bezug auf das Aussehen des Gesichtes in dem gegebenen Bild reproduziert. Zuerst wird ein 3D Morphable Model an das 2D Eingabegesicht angepasst. Dann wird eine generative Menge von Bildern unter genau den gleichen Bedingungen wie das Eingabebild, aber mit unterschiedlichen Beleuchtungen gerendert, indem jedes Bild unter einer einzigen Lichtquelle mit Einheitsintensität synthetisiert wird. Die Menge der Lichtquellen ist vorgegeben und wird als *Virtual Light Stage* bezeichnet. Es wird davon ausgegangen, dass das Eingabebild zu dem *synthetischen Beleuchtungskonus* gehört, den dieser generierende Satz aufspannt. Dann werden die Koeffizienten geschätzt, so dass die Linearkombination des generierenden Satzes möglichst ähnlich dem Eingangsbild ist. Um realistischere Lichteffekte zu ermöglichen, wird eine nicht-Lambert'sche Reflektanzverteilungsfunktion verwendet, die Fresnel-Reflexion bei flachen Winkeln berücksichtigt. Es wird viel Aufwand in die Analyse und Synthese von Schlagschatten unter komplexen Lichtverhältnissen investiert. Für die Parameterschätzung werden zwei probabilistische Modellierungsansätze vorgeschlagen. Ein hierarchisches Bayes'sches Modell unterdrückt automatisch Inkonsistenzen zwischen dem generativen Modell und dem Input. Der Optimierungsalgorithmus mit Nicht-Negativität als Nebenbedingung findet die optimalen spektralen Intensitäten der Virtual Light Stage für das Eingangsbild. Diese Dissertation ist ein Beitrag zur Gesichtsbild- und Umgebungsmodellierung. Einige Anwendungen des vorgeschlagenen Algorithmus sind zum Beispiel die realistische Rekonstruktion des Eingabebildes, die intrinsische Gesichtsmodellzerlegung, die Wiederbeleuchtung und der Beleuchtungsentwurf.

Acknowledgments

This work has only been possible with all the support and help I have received. Here, I would like to express my gratitudes to:

Prof. Dr. Volker Blanz, my first advisor, for giving me the chance to do my dream job for almost 5 years, for his continuous support of my work, his lifesaving immense knowledge of the field and his contagious enthusiasm and noble approach to research. It has been a great experience for me to work in his team. Also, thanks for all the wonderful conversations about culture, arts, life, the world and everything.

Prof. Dr. Martin Fuchs, who accepted to be the co-advisor of this dissertation, for his helpful feedback and challenging questions which for sure contributed to the quality of this work and the joy of defending it.

The DFG funded Research Training Group GRK 1564 'Imaging New Modalities' and the University of Siegen who financially supported the work. Also, my second supervisor at the GRK, Prof. Dr. Marcin Grzegorzec, and the Spokesman of the GRK, Prof. Dr. Andreas Kolb, for their motivating support and helpful advice during my Ph.D studies.

My colleagues at the Chair for Media Systems: Joanna, Michael, Netti, Matthias, Marcel, Tim, Björn, Mai Lan, and Thomas, for the stimulating discussions, their expert input and for all the fun coffee breaks we had during these years. Also, special thanks to the fellow researchers at the GRK from whom I have learned about subjects that are out of the scope of my research, especially Miguel, with whom we worked on a Compressive Sensing approach for the Inverse Rendering problem. I also would like to thank the coordinators of the GRK, Nadine and Christian, who went beyond their official duties in supporting the GRK and me. Thanks for the amazing colleagues and friends that you have been ever since.

My family and friends for believing in me and for their moral and emotional support throughout my studies and life in general. Especially, I would like to thank Sougand, Shirin and Kiumars for their essential and heart warming support. I also thank Mashall, for our inspiring discussions about the theories of probabilistic modeling, and Arya, for helping me with the proofreading of this thesis.

And at last, to my lovely mother, Azar, who plays a key role in shaping my dreams ever since she started reading me biographies of the greatest minds as bedtime stories. I know no parent is perfect, but what she did for me seems like a fine tuned plan that brought me all the way so far. It is because of her noble vision if I do anything good in my life.

Finally, I would like to pay respects to my eternal best friend, role model and life coach, my father, Mohsen. He has always been the first to know about my success and my troubles. I miss our book reading sessions and musical holiday mornings very much. Without his guidance, life would have been a much more confusing experience. I owe it all to Azar and Mohsen.

Dedication

I humbly wish to dedicate this work to the children of wars
and to peace, my childhood dream.

‘Look at light and admire its beauty. Close your eyes and then look again;
what you saw is no longer there and what you will see later is not yet.’

Leonardo da Vinci

Contents

Abstract	i
Zusammenfassung	iii
Acknowledgements	v
1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Input and Output of the Proposed Inverse Lighting	6
1.2.1 Input Image	6
1.2.2 Approaching the Problem and Overview of the Algorithm	7
1.3 Contributions	9
1.4 Publications	10
2 Background and Related Work	11
2.1 Previous Work	11
2.1.1 Data-Driven Approaches	11
2.1.2 Single-Image Approaches	14
2.1.3 Lighting Design	16
2.2 The Appearance of Human Faces	18
2.3 Face Model Extraction with 3DMM from a 2D Image	21
2.3.1 The 3D Morphable Model (3DMM) of Faces	21
2.3.2 3DMM Fitting Algorithm and Joint Lighting Estimation	22
2.3.3 Model-Based Texture Extraction from 2D Image	23
2.4 Illumination Cone	25
2.4.1 Lighting Models	27
2.4.2 Light Sources	27
2.4.3 Light Stage	28
2.4.4 Formalizing the Superposition Principle	29
2.4.5 Estimation of RGB Parameters of Light Sources	30
2.5 Image Formation	30
2.5.1 Rendering	30
2.5.2 Phong Model	31
2.5.3 Torrance-Sparrow and Dipole Functions	32
2.5.4 Soft Cast Shadow Mapping	34
2.5.5 Color Correction	36

2.6	Cast Shadow Detection and Segmentation	36
2.7	Image Compression	37
2.7.1	Downsampling Filters	37
2.7.2	Compressive Sensing Matrix	38
2.7.3	Superpixel Segmentation	38
2.7.4	Texture Patches	39
3	Inverse Lighting and Relighting	41
3.1	The Inverse Lighting Framework	41
3.1.1	The Use Case Diagrams for the Proposed Inverse Lighting	41
3.1.2	An Activity Diagram for the Inverse Lighting Algorithm	44
3.2	To Span a Synthetic Illumination Cone	45
3.2.1	Virtual Light Stage (VLS)	46
3.2.2	Adopting a Measurement-Based Reflectance	49
3.2.3	Color Correction	50
3.2.4	3DMM Masks	51
3.3	Cast Shadow Detection and Segmentation	51
3.4	Image Compression	54
3.4.1	Masked Gaussian Downscaling	55
3.4.2	Geometric Superpixels	55
3.5	Inverse Lighting Algorithm with Superpixels	58
3.6	Relighting	59
3.6.1	Intrinsic Texture Decomposition	59
3.6.2	Lighting Design	62
3.7	An Effort to Improve the Estimated Shape after Inverse Lighting	67
4	Estimation by Optimization	69
4.1	Linear Model	69
4.2	Cost Function from the Generative Model	70
4.2.1	Least Squares	72
4.3	Maximum A-Posteriori (MAP)	73
4.4	Joint Maximum A-Posteriori (JMAP) for Hyperparameter Optimization	75
4.4.1	Expected Values of the Priors	76
4.5	Optimization with Newton-Raphson	77
4.5.1	Enforcing Nonnegativity and Light-Weight Sparsity	79
4.5.2	The Hyperparameter Optimization Mechanism	80
4.5.3	Handling Occlusions and Areas of Misalignment	81
4.5.4	The Challenge of Cast Shadows	82
4.6	Implementation Tricks and Magic Numbers	83
4.7	Alternative Methods	84
4.7.1	Nonlinear Model $\vec{x} = e^{\vec{\alpha}}$	84
4.7.2	Richardson-Lucy (RL)	86

4.7.3	Full Hessian Inverse	87
5	Evaluation and Results	89
5.1	Mean Illumination SIMilarity (MISIM)	89
5.2	Results	91
5.2.1	Results of MAP	92
5.2.2	Ambiguity of the Estimated Lighting	96
5.2.3	The JMAP Results	96
5.2.4	Experiments with Different Number of Light Sources n	98
5.2.5	Experiments with Different Number of Superpixels m_S	98
5.2.6	Cast Shadows	102
5.3	Relighting Results	103
5.3.1	Lighting Design	104
6	Conclusion	107
6.1	Summary of Attempts and Achievements	108
6.2	Unanswered Questions or Future Work	109
	Index	111
	Bibliography	114

List of Tables

1.1	Table of Requirements (Assumption vs. Estimation)	8
5.1	MISIM and MSIER Results	100

List of Figures

1.1	Portraits by Johannes Vermeer	2
1.2	Sixth Sense	3
1.3	Built-in Illumination Estimation of the 3DMM	4
1.4	Inverse Lighting Goal	5
1.5	UML Acitivity Diagram of Inverse Lighting	8
2.1	Light Stage Results, Courtesy of Debevec et al.	12
2.2	Single Image Results, Courtesiy of Wang et al.	15
2.3	Taxonomy of Skin from [INN07]	20
2.4	Overview of the 3DMM Fitting	23
2.5	Light Stage LS6 of USC ICT	28
2.6	Shadow Buffer	35
2.7	Motivation for Color Correction	36
3.1	Use Case Diagram for the Human User	42
3.2	Use Case Diagram for the 3DMM Framework	43
3.3	Use Case Diagram for the Inverse Lighting Framework	43
3.4	UML Activity Diagram of Inverse Lighting (extended version)	44
3.5	Rendering the Generating Set	45
3.6	Activity Diagram for the Preparation of the Generating Set	46
3.7	Virtual Light Stage (VLS)	47
3.8	Cast Shadow Segmentation	53
3.9	Geometric Superpixel	56
3.10	Superpixel Generating Set	57
3.11	Activity Diagram for Intrinsic Texture Decomposition	60
3.12	Remove the Illumination Effect from the Face Model	61
3.13	Activity Diagram for Lighting Design	62
3.14	The Process of Lighting Design in Images	64
3.15	Overview of the Lighting Design Algorithm	65
3.16	Error in Average Texture	66
4.1	Illumination Estimation on Synthetic Input	75
4.2	PGM Diagram for JMAP for Color Images	76
4.3	JMAP Overview Diagram	78

4.4	Hyperparameter Optimization	79
4.5	Three Candidate Distributions	85
4.6	Exponential Coefficients $\vec{x} = e^{\vec{\alpha}}$ Results	86
4.7	Richardson-Lucy Results	88
5.1	MISIM Results	91
5.2	Inverse Lighting for an Image with the MAP Approach from [SB15b]	93
5.3	Inverse Lighting for an Image with the MAP Approach from [SB15b]	94
5.4	Inverse Lighting for an Image with the MAP Approach from [SB15b]	95
5.5	Rendered Spheres under Lighting from Similar Illumination	96
5.6	Different Number of Superpixels	99
5.7	Results for 9 Various Examples with $n = 100$ JMAP	101
5.8	Improvement of Cast Shadow Estimation	102
5.9	Lighting Transfer	103
5.10	Lighting Design with Automatic Landmark Localizer	105
5.11	Lighting Design Results On A Single Face	105
5.12	Lighting Design Results on Five Images	106
6.1	Activity Diagram for the Preparation of the Generating Set (Extended)	110

Chapter 1

Introduction

1.1 Motivation and Objectives

Faces are among the most often photographed and painted subjects. They are also interesting in the fields of computer vision and graphics because they are nonrigid, their surfaces have concave areas and their reflectance behaviors are complex. Such properties let the facial appearance in a 2D image change radically, and partly even nonlinearly, with respect to changes in lighting in the 3D environment. The changes are so significant that a harsh lighting makes face model analysis and face recognition [SB15b, MAU94, RHVK06, CRA⁺13] even by humans [BKTT98] a difficult task. Analyzing illumination of a facial image is of great interest for the phenomenology of imaging, lighting, facial appearance modeling, and face recognition from real images. Hence, even a coarse estimation of the lighting contributes to the solutions in different fields, including face recognition, image forgery detection and visual arts. Furthermore, illumination can reveal valuable information about the off-screen environment of the face, which might be useful for use cases beyond image forgery detection. Let us motivate the idea of using computers for lighting estimation by discussing how observant we, humans, are when it comes to analyzing the illumination.

Suitably called the ‘Master of Light’ [Kra01], Johannes Vermeer paints his face on canvas (Figure 1.1a) with his well-known magnificent design of illumination. Some facial features of the face in Figure 1.1a are covered under a pale shadow or kept invisible due to his pose and under occlusions by his hair and hat, yet, we clearly see his celebrating smile. With a guided look, we notice the brighter lower jaw and the highlights on the tip of his nose. A similar reflection is visible in his master piece in Figure 1.1b on the lower jaw toward the ear. We might argue about where and of what spectral properties the main light sources in each picture are and how the subtle highlights are caused because the answers to such questions do not pop into the mind in a first glance. When only a part of the scene is visible in the given image and when the properties of the objects’ appearances



Figure 1.1: Portraits by Johannes Vermeer. (a) is part of the painting "The Procuress," 1656, and (b) "Girl with a Pearl Earring," 1665. Notice the color bleeding from the white collars over the lower jaws of both subjects.

are unknown, it is almost impossible to come to a conclusion about the constellation of the light sources. Moreover, the emitters form only a fraction of the environment that illuminates the face. Usually, brighter surrounding objects have their impacts on the global illumination; objects such as walls and ceilings, green grass that the subject lies on, a nearby red car, white collar of a shirt, the surface under the visor of a blue cap, a bright pavement in a hot summer noon etc. Such color permeations between nearby objects are caused by indirect lighting, and called color bleeding in the literature [Bir00]. There is no need to be a master of light to see such subtle reflections. They are always in front of our eyes, however, the fact that it takes a meticulous artist to intentionally draw them, shows the challenge of recognizing these effects without extra hints. All in all, unless you pause in front of the frame and look for them, these subtle illumination effects usually do not pop into the attention. Even after spotting a reflection on the face, it takes some spatial imagination to say from which source or even roughly which direction the face is being illuminated.

Well aware of our struggle, artists produce physically inconsistent scenes to pursue other artistic goals. Take the example of the famous 'I see dead people' scene from the 'Sixth Sense' [Shy99]. Four screen shots of this scene are provided in Figure 1.2. Look at the



Figure 1.2: Screenshots from the ‘I see dead people’ scene, ‘Sixth Sense’ [Shy99].

environment in both frames on the left. The light sources that illuminate the actors’ faces in the close-ups on the right are absent where we expect them to be. The light source that illuminates the face of the grown-up actor in the bottom right frame should have been visible on the wall in the top left frame. This wall is not bright enough to be the main light source on the actor’s face, compared to the window. Also, the bright window in the background of the bottom left image should have illuminated the child actor’s face in his close-up, top right frame. Can we say it is occluded by the actor by looking at the pictures on the bottom? One might argue about the detail, nonetheless, the illumination is not consistent in this scene and we cannot conclusively give an analysis of what is physically consistent about the design of the lighting. Film makers know very well that their audience barely minds or even spots such inconsistencies when they are following the story and the acts.

The analysis of lighting in an image is disrupted by 3D phenomena, such as occlusion, cast shadows of off-screen objects, missing data about the facial geometry and unknown light sources. Moreover, the absence of a detailed skin reflectance for the face at the moment of photography leaves us with a significant lack of data. Such limitations are common between analysis by human and computer. Computationally seen, when the scene is unknown and the object of study is not a mirrorlike reflector, inferring the position and type of the surrounding light sources from a noncalibrated 2D image –irradiance from radiance– is a difficult inverse problem with many unknowns. A common way to analyze such sophisticated realistic scenes has been analysis by synthesis [YK06]. It is modeling a scene through reconstructing it with the given measurements and using priors to cor-

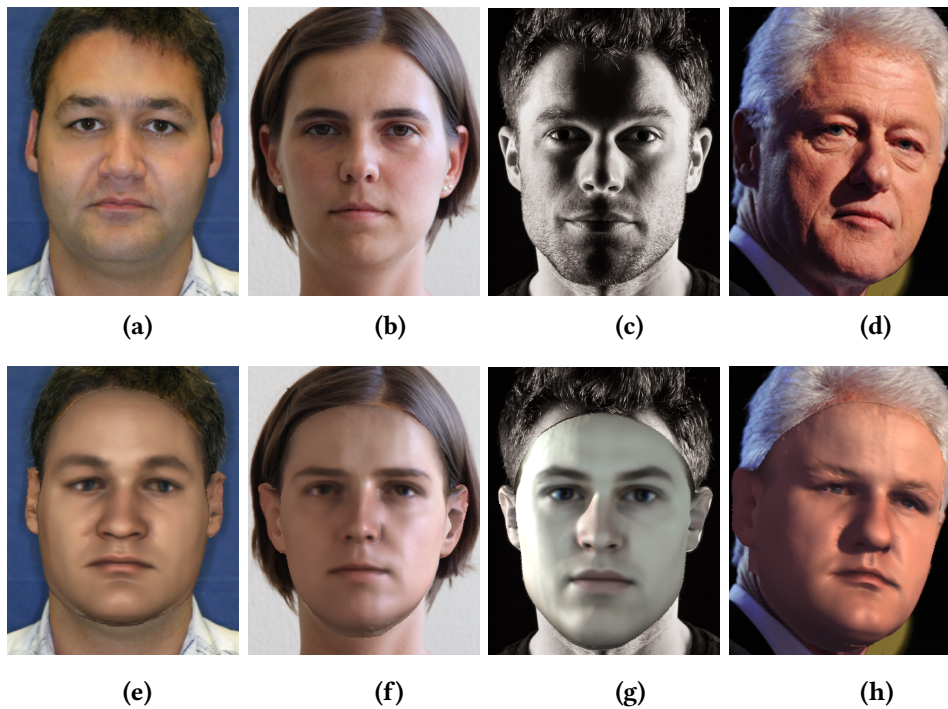


Figure 1.3: Built-in illumination estimation of the 3DMM framework, rendered on the estimated shape and average texture, to suppress possible errors in texture estimation. The estimated illumination in Figure 1.3e looks promising. In Figure 1.3f the lighting is a bit off but still usable. In Figure 1.3g and Figure 1.3h the estimated illumination exhibit critical errors. The former misses the darker middle and brighter right side of Figure 1.3c completely. Figure 1.3h misses the blueish highlight close to the ear and the cast shadow under the nose in Figure 1.3d is only minimally replicated. Images Figure 1.3a is from [SBB02] database and Figure 1.3c is courtesy of Barrie Spence©.

rect the reconstruction and the model simultaneously [BV99] or subsequently [SB15b]. Inspired by the success of the measurement-based methods (see Section 2.1), this thesis builds upon the single-image inverse rendering of 2D face images by Blanz and Vetter [BV99].

The built-in inverse lighting of [BV99] estimates the lighting parameters together with the shape and texture of the face in a joint model fitting process. It works well on images with simple lighting conditions, such as Figure 1.3a and 1.3b. The former is a very simple frontal illumination, which is reconstructed quite well in Figure 1.3e. The latter looks good in the first glance, however, the rim-light effect on the left side of the image is completely missing in the reconstruction Figure 1.3f. The estimation of lighting from Figure 1.3b proved to be very challenging in the experiments because the highlight ribbon close to the right ear (left side of the image) is in high frequency domain. These results are usable for modeling and recognition purposes, however, this algorithm fails to achieve a roughly correct replication of the lighting for an image with only three differently illuminated areas such as Figure 1.3c or one with more complexities such as Figure 1.3d (see image caption). By separating the lighting estimation from the rest of the inverse rendering, this

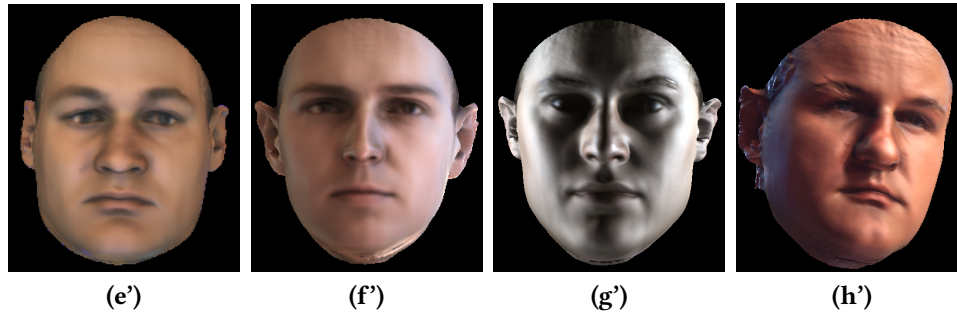


Figure 1.4: This is the goal that the thesis achieves, compared with Figure 1.3. In case of Figure 1.4e' and Figure 1.4f' the improvements are limited to subtle effects. In Figure 1.4f', the transition of the lighting from one side of the image to the other side and the high frequency highlight close to the right ear (left of the image) are added qualities compared to Figure 1.3f. In Figure 1.4g' three differently illuminated areas correctly appear on the face, two bright sides and the dark area in the middle. In Figure 1.4h', the blueish highlight close to the ear is reconstructed and the cast shadow of the nose is improved. Compare with Figure 1.3.

thesis aims for a physically plausible inverse lighting, which tackles harsh illumination conditions, causing colorful reflections, cast shadows, and highlights in mirroring and grazing angles.

In portrait photography, the costs of lighting equipment, their maintenance and the lack of training for using them makes lighting inaccessible to the growing number of consumers of point and shoot cameras and Smartphone photographers. The captured portraits are usually monocular pictures of faces under uncontrolled and unknown lighting. Even post-production, now the inseparable part of digital photography, can become hopeless in presence of harsh illumination effects. Furthermore, pre-existing illumination effects interfere with important tasks which need face model extraction, e.g. face recognition.

Although, the proposed algorithms presume no lab or special equipment, the goal is to address the analysis and synthesis of images with realistic and complicated lighting conditions. Relighting of a given face image (Section 3.6) is one of the most interesting motivations for a software solution which can estimate and remove the lighting, attain the illumination invariant features and render the face under a novel lighting condition. Estimating a physically plausible lighting from an image of a face is a challenging task, given the fact that there are almost no constraints on the input image and it hardly provides reliable or even enough input data to solve the inverse problem in a numerically stable manner. To understand the challenge, we start by discussing the initial requirements and the direct outputs of the proposed inverse lighting. Then, an overview of the proposed algorithm follows.

1.2 Input and Output of the Proposed Inverse Lighting

Looking at the proposed algorithm as a black box, it gets a face image I and some landmarks on it to return a lighting configuration with respect to the visible illumination on the face. For further applications of the lighting estimation, other inputs might be necessary. Relighting (Section 3.6) needs an image to estimate the target face model from and a source for the target lighting. We explain both when the relighting algorithm is proposed.

1.2.1 Input Image

For the input image, the following properties are expected. It has to be a digital or digitized image in RGB color space. A human face of an unknown identity, age, gender or ethnicity must be visible in it. The face might have any pose in any direction and might be occluded as far as some facial features and skin are still visible. Portrait paintings are also accepted as far as the facial morphology is preserved (no cubism). In case the input is synthesized or painted without a physically plausible lighting, the proposed algorithm estimates an optimal physically plausible lighting for it anyway. The input image is without any extra metadata and the image size, resolution, saturation, contrast and brightness are not under any constraints.

Input Facial Landmarks

A landmark assigns a pixel from the image I to a vertex of a generic face geometry. The landmarks initialize the face model estimation by providing a coarse correspondence between the 2D input image and the 3D shape. The algorithm can be initialized by as few as 7 landmarks. Each landmark is clicked on the mesh and then on the image, manually. Theoretically, the position of each vertex of the mesh can be mapped on the image, however, finding a corresponding pair of image pixel and model vertex on smooth areas, like the cheek, is too ambiguous. Usually, if they are visible in the image, the corners of the eyes, the mouth and the tip of the nose are safe to be clicked. One can put more landmarks on the silhouettes of the face on the image to avoid misalignments in the silhouettes area. In Chapter 5, we see that the illumination effects around the silhouettes are of great interest and are not allowed to be undermined due to wrong alignment of the silhouettes region on the background. In Chapter 4, two estimation approaches are proposed: MAP and JMAP. The former relies more strictly on the inputs, thus, we need to be very careful that the selected landmarks do not lead to disadvantageous alignment of the silhouettes on the background. The latter accounts for unreliabilities in the input and ignores local inconsistencies, such as misalignment of the silhouettes, automatically.

Input Occlusion Mask

Although the algorithm can be initiated only with the input image and the landmarks, it also considers an occlusion mask if provided. Whenever the face is occluded by unknown objects, e.g. glasses and hair, it is suggested to provide an occlusion mask to avoid fitting the model to the unknown occlusion. Without an occlusion mask, the face model estimation step (Section 2.3) fails to estimate a promising texture from an occluded image. Nevertheless, the subsequent lighting estimation, which is equipped with the probabilistic modeling JMAP in Chapter 4, deals with such inconsistencies automatically. See the examples G and H in Figure 5.7.

Output

The inverse lighting algorithm provides a physically plausible lighting, corresponding to the illumination effects on the face in the input image. The estimated lighting is supposed to be theoretically implementable in ideal lab conditions, therefore, we aim for a nonnegative solution with distinct light sources. Accordingly, the direct output of the algorithm is a vector of light source intensities in three color channels (RGB), denoted with \vec{x} , for predefined directions. Estimating the real light source positions, directions and shapes is out of the scope of this thesis. The reason is the limitations of the input, which is an unknown face with its diffuse reflectance and unknown surface normals, which is used as a light probe. Instead, a physically plausible lighting is estimated that can reconstruct the original lighting in an analysis by synthesis sense. Furthermore, use cases such as intrinsic image decomposition, relighting and lighting design are discussed in Chapter 3 and their results are provided in Chapter 5. These use cases rely on the output of the inverse lighting algorithm, and therefore show the limitations of the proposed algorithm quite fairly.

This dissertation shows the intermediate and final results to show the performance and limitations of the proposed algorithms. Moreover, in Section 5.1, a novel objective measure of illumination similarity is proposed which can inspire research on quantitative measures for assessment of illumination estimation.

1.2.2 Approaching the Problem and Overview of the Algorithm

The under-constrained input raises fundamental requirements for the inverse lighting problem. These requirements are bound to be met by estimation or assumption. In Table 1.1, a list of the main requirements is given. For each entry of the list it is also indicated how it is handled in this dissertation. They are mostly rendering and imaging parameters, which play significant roles in the proposed analysis by synthesis approach. Assumptions might be empirical, measured, or even results of generalization on previous estimations.

By estimation, we mean estimation during the course of the proposed algorithm as part of the 3DMM fitting (see Figure 1.5). To approach the problem, first the inverse rendering procedure (model fitting algorithm) of the 3D Morphable Model (3DMM) framework [BV99] is employed to estimate the 3D face model and its alignment from the input. Then, a *lighting from radiance, shape and texture* algorithm estimates the lighting. In doing so, the rendering parameters are fed to the Estimate Lighting module in Figure 1.5 to build a generative set and parametrize the illumination. Finally, a cost function is minimized to find the coefficients of the generative model so that the generated image is as similar as possible to the input image. The estimated coefficients give the lighting configuration.

Table 1.1: Requirements of the proposed inverse lighting algorithm. The required rendering and imaging parameter are supplied by assumption or estimation in this thesis.

Problem	Approach	Ref.
Face 3D geometry, pose and correspondence	estimation	Section 2.3
Face reflectance and texture	assumption	Section 3.2.2
The color and position of light sources	estimation	Chapter 3 and 4
Color saturation, contrast and brightness . . .	estimation	[BV99] and Section 3.2.3

The thesis is structured to be thorough with respect to the main topic, yet, concise for readers from different backgrounds. The basic concepts and previous work are gathered in Chapter 2, the algorithms for image processing, and computer vision and graphics approaches are presented in Chapter 3, and the mathematical modeling of the lighting, optimization algorithms together with their alternatives and numeric considerations are collected in Chapter 4. The results and the evaluation methods are provided in Chapter 5, while conclusion and future work are grouped in Chapter 6. Especially for the readers who hold a hard copy, an index is placed before the bibliography. Furthermore, a measurement-

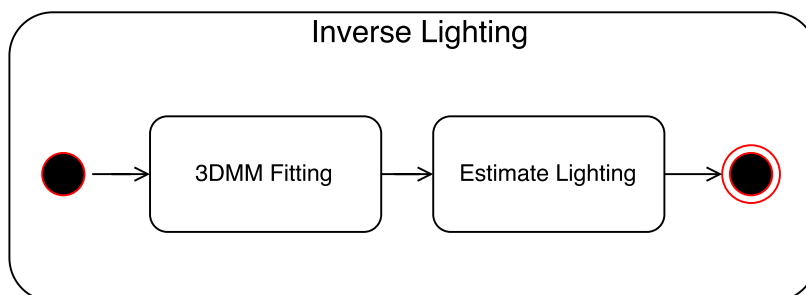


Figure 1.5: UML Activity Diagram of the proposed inverse lighting. Activity Diagrams are explained in Section 3.1 where more detailed diagrams are shown. The proposed inverse lighting is designed in two subsequent steps, i.e. 3DMM Fitting and Estimate Lighting. The algorithm is initialized with the input image and landmarks by the user (see Figure 3.1), the 3DMM fitting algorithm estimates the 3D face model and other rendering parameters (See Section 2.3). Then, the light sources are estimated according to an analysis by synthesis approach. See Figure 3.4 for another version of this diagram with lower abstraction of the “Estimate Lighting” activity.

based face reflectance is adapted to produce more realistic reflectance effects for faces. The Virtual Light Stage (VLS) is proposed to model the global lighting in absence of a real environment map. The nature of the problem promotes the use of image filters to prepare the data or remove the noise (Chapter 3). Different approaches to nonnegative optimization are explained and successfully implemented, upon the idea of Bayesian decision theory, to estimate the coefficients of the VLS (Chapter 4). For a qualitative assessment, results on both simple and harsh lighting inputs are given and as an objective approach to evaluation of illumination estimation, a quantitative illumination similarity measure is proposed in Chapter 4. Some applications of the inverse lighting are intrinsic face model decomposition, relighting and lighting design, which are included. The creative lighting design algorithm (Section 3.6.2) provides a useful tool for facial image editing. The creative lighting design is a fashionable product of this project, yet, together with intrinsic texture decomposition and relighting, it is provided as an ultimate test for the promises of proposed inverse lighting.

1.3 Contributions

Building upon previous work in facial appearance modeling, generative models, constrained optimization and image processing, this thesis tackles unsolved problems and introduces the state of the art holistic approach for inverse lighting of single 2D images of faces. Thereby, a new class of lighting complexity is experimented with which has been absent in previous work. Moreover, a physically plausible global lighting representation is introduced, which helps with the analysis and synthesis of realistic illumination effects. Qualitative evaluations are facilitated with the help of experiments with different types of real input images and demonstration of further applications of the estimated lighting to uncover the strength and limitations of the proposed methods. In addition, a novel quantitative assessment for illumination estimation is proposed, which can be used for evaluation of future work against the state of the art. The implementation of the algorithms can work stand alone, however, they are embedded in the 3D Morphable Model framework. Hence this thesis can be seen as a contribution to the inverse rendering with 3DMM. In that sense, it also adds the Virtual Light Stage as a physically plausible lighting model to the 3DMM framework and removes the existing Phong model of reflectance in the 3DMM with an empirical BRDF of human face. From a different point of view, the inverse lighting includes the fitting of 3D Morphable Model together with other methods that are proposed, adapted from other domains or directly employed in this dissertation. Also, the amount of data and memory consumption, the complexity of the inverse lighting optimization and the noise are handled with the help of a novel illumination friendly superpixel segmentation. The proposed model-based superpixel approach forms a considerable contribution to the performance and efficiency of the algorithm. This dissertation

proposes the first inverse lighting algorithm that estimates a physically plausible lighting model, using a reflectance function that goes beyond Lambert law and Phong model to consider and reconstruct specularities in grazing angles and cast shadows. It includes a framework to experiment with even more comprehensive reflectance functions. Besides, the proposed inverse lighting algorithm is used in further sophisticated algorithms for applications, such as intrinsic texture extraction from the image, relighting and a software solution for creative lighting design.

1.4 Publications

Primarily in [SB15b], the core idea of the proposed inverse lighting is established. In a collaboration [HCSBL15], a Compressive Sensing approach is investigated and proposed for the parameter estimation step. Later in [SPB16], a software solution for post hoc lighting design is introduced. More advanced topics, such as the JMAP optimization and the geometric superpixel method, are proposed in the article [SB17]. These four publications are cited in green through the dissertation to differentiate between self citations and other resources, which are in blue.

Chapter 2

Background and Related Work

2.1 Previous Work

One of the main goals of lighting estimation is the neutralization of illumination effects in intrinsic appearance decomposition as a step in face modeling and model-based face recognition. These are separately surveyed by Huq et al. [HAKA07] and Zhao et al. [ZCPR03]. The pioneer methods that consider the lighting usually assume Lambertian reflectance and convex geometry for faces. While these assumptions underestimate the physical problem, under circumstances they lead to relatively promising results. Hallinan employs the superposition principle of light and builds a basis for lighting by performing Principal Component Analysis (PCA) on *boundary images* of different point light sources, already in 1994 [Hal94]. In his work, the faces are modeled with 5 eigenfaces inspired by Turk and Pentland [TP91]. Belhumeur and Kriegman calculate a basis to span the illumination cone of a face image, given frontal images of that face taken under different lighting conditions [BK96]. Inverse rendering can be implemented as an approach to illumination invariant face modeling and recognition [KHSB98, PC05, LLS05, ZKM07, PKA⁺09a, CKZX13, AVM⁺14, NCG15] or employed to satisfy a major requirement of model-based facial image relighting. Choudhury et al. [CCH07] provide a comparison of image-based relighting methods. Next, the most related efforts on inverse lighting and relighting are reviewed. These methods can be loosely divided in data-driven and single-image approaches. Although this thesis is a contribution to single-image inverse lighting, it proposes a physically plausible lighting model that can also inspire future data-driven methods.

2.1.1 Data-Driven Approaches

The work of Belhumeur and Kriegman [BK96, BK98] on the idea of illumination cone (see Section 2.4) is followed by Georghiades et al. on face image synthesis [GBK99] and illumination invariant face recognition [GBK01]. These efforts have inspired the use of a light stage to capture images of the face under controlled lighting conditions [USC08,



Figure 2.1: Light stage results, courtesy of Debevec et al.: “. . . (a) an original light stage image taken by the left camera. (b) recovered surface normals . . . derived from the fitted diffuse reflectance lobe for each pixel; (b) the RGB value for each pixel encodes the X, Y, and Z direction of each normal. (c) Estimated diffuse albedo . Although not used by our rendering algorithm, such data could be used in a traditional rendering system. (d) Estimated specular energy , also of potential use in a traditional rendering system. (e) Face geometry recovered using structured lighting. (f) Face rendered from a novel viewpoint under synthetic directional illumination. (g,h) Face rendered from a novel viewpoint under two sampled lighting environments . . .” [DHT⁺00].

[Deb12]. Debevec et. al. show the acquisition of the reflectance field of a face using many captured images under a light stage. The light stage is equipped with optical filters, high quality cameras and a structured light scanner [DHT⁺00]. They estimate reflectance from the collected data to synthesize new images from original view point under novel lighting conditions. The target lighting is sampled or modeled and prepared in the form of an environment map. To change the view point, they use a skin reflectance model and a sampled 3D model of the face. Their results are realistic and include all subtle illumination effects on the face, as far as they are captured under the light stage, and modeled by the reflectance for novel view points. This includes a great amount of the illumination effects, e.g. intensity and color of the lighting, cast shadows and color bleeding from nearby objects due to inter-reflections (see Figure 2.1).

Graham et. al. measure the micro-geometry of skin by analyzing the light stage images [GTB⁺13]. They calculate surface normals and produce a bump map for the skin, using classic photometric stereo. They use a 12-light dome and polarized lighting to estimate

the BRDF. Weyrich et. al. measure the reflectance of face skin using light stage images and measuring devices, such as translucency sensor [WMP⁺06]. Their results are generalized and used as an assumption for skin reflectance in this dissertation (see Sections 2.5.3 and 3.2.2).

Fuchs et. al. present a method for relighting of real objects [FBS05]. They use photos of a probe object -a black snooker ball- near the target object to calculate the effects of the environment illumination. To generate a target lighting condition on the target objects, the coefficients of a linear combination of photos of the probe under the desired lighting are estimated with a Maximum A-Posteriori (MAP) approach. They consider a prior which is estimated by applying PCA (Principal Component Analysis) on a set of probe samples. Finally, adding up the photos of the target object, with the estimated coefficients, gives the image of the target object under the target lighting. The idea of using a probe object in the scene to estimate the lighting is also used in [FBL05] to measure the reflectance of human faces from images, taken in a lighting lab and using a 3DMM fitting for estimation of the 3D face model. In a later work, Fuchs et al. [FBL07] extend the idea of the reflectance sampling from still scenes with a more sophisticated sampling pattern, adapted to the estimation of the reflectance field dynamically. Nishino and Nayar use the eye of the subject as a light probe to estimate the illumination of the scene [NN04]. They calibrate their imaging system by computing the 3D coordinates and orientation of the cornea from a single image. Then they use the law of reflection to calculate more than half of the environment map of the scene from the corneal reflections. They apply their method to dark colored eyes to avoid the reflections of the iris. The appearance of the face is then sampled from several images (one second video) of the face under different lightings. From these images the shape and albedo of the face are recovered.

In another image-based approach, Ren et al. [RDL⁺15] estimate a light transport matrix of a scene from a few images under different lighting conditions, limiting the lighting conditions to light sources that are on the same plane. To estimate the lighting condition, they solve a regression problem with a neural network of two hidden layers. Their literature review of light transport methods is recommendable.

Compared to single-image approaches, the data-driven methods promise more realistic results, however, they demand at least a few ideally captured input images or lab equipment and presence of the subjects. These works show the challenges of appearance acquisition in labs under controlled environment. Well aware of the fact that an image, taken from one view point, misses a great amount of information about the appearance of the photographed face, data-driven approaches decide to capture the facial appearance of the subject with designated hardware. This consequently invests a great deal of time and effort that returns the benefit of high quality detailed results, which appeal to the lofty standards of the film industry [USC08].

2.1.2 Single-Image Approaches

Data-driven approaches are limited to specific setups and use cases. In many real life scenarios, e.g. in civil security and arts, nothing more than a single 2D image is available. In this work, the input is limited to a single image of a face, taken from an unknown source, captured under unknown conditions. Active light single-image approaches [LCLZ07], or those which use a phantom object as light probe [FBS05], are out of the scope of this thesis. Comparable methods are those that work with a single image, taken under uncontrolled conditions and might have an unknown, harsh and complex lighting.

Single-image inverse lighting techniques in [BV99, WLH⁺07, BKD⁺08, KSB11, AS13, LZL14] and the proposed method are forced to use estimations and assumptions for the whole involving parameters. Their results differ based on the extent of the assumptions and the accuracy of the intermediate estimations. For ideal illumination conditions, even a simple lighting model, such as the one used in [BV99], proves to be sufficient for most use cases. They use one directional light together with ambient parameters for an *ad hoc* Phong model to estimate the lighting of the face.

A few other previous works use 3DMMs for estimation of geometry and texture from a single image as part of their single image approach [WLH⁺07, AS13, LZL14]. The first two use spherical harmonics to represent reflectance and lighting. Aldrian and Smith separate the illumination estimation in two parts, i.e. diffuse and specular. Each of these components is modeled with a different set of spherical harmonics. A physically plausible lighting is not their goal, nonetheless, the image reconstructions show impressive results, especially in the absence of harsh illuminations under colored multidirectional lights and cast shadows [AS13]. Wang et al. specifically address the problem of harsh illumination. They add a Markov Random Field (MRF) term to the energy function to promote the statistical distribution and spatial coherence of the texture of 3DMM, which leads to robustness against harsh illumination. They minimize a cost function jointly for lighting, albedo and the shape together with the MRF term [WLH⁺07, WZL⁺09]. Their results are presented on frontal or close to frontal poses and gray-scale image under one main light source. It is impossible to judge the estimated lighting because they do not reuse it to render the face or other objects. However, their intrinsic texture extraction results are impressive, in terms of avoiding adverse illumination effects (see Figure 2.2). Bitouk et al. follow a similar scenario considering the lighting differences also in colored images, for the goal of face swapping [BKD⁺08]. They assume Lambertian reflectance and use spherical harmonics to model lighting. Li et. al. [LZL14] use results of a 3DMM fitting as prior for geometry and albedo, the reflectance from [WMP⁺06] as prior for the skin reflectance and a set of environment maps modeled with Gaussian Mixture Model as prior for lighting. Then, they combine these priors with a holistic cost function of independent parameters for: geometry, texture, reflectance and lighting, and optimize jointly for all

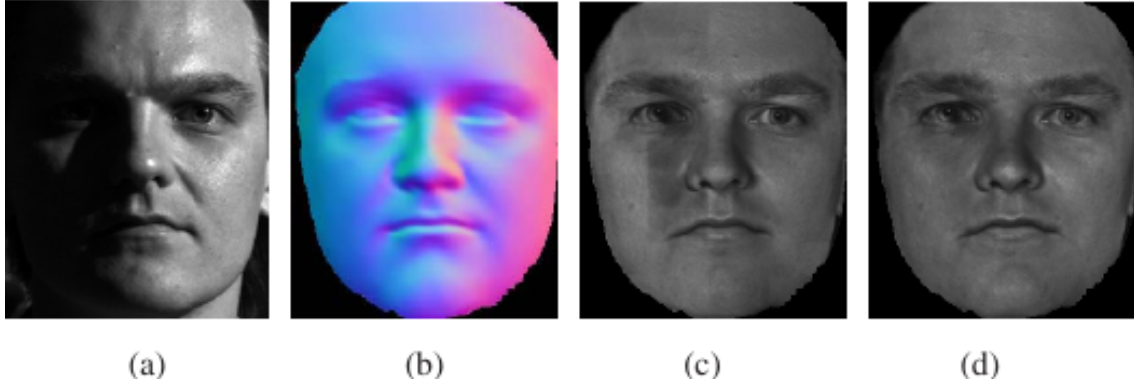


Figure 2.2: Single image results, courtesy of Wang et al.: “(a) is the original image taken under a harsh lighting condition, (b) shows the recovered surface normal from our method (where R,G,B color values represent the x,y,z components of the normal), and the recovered albedo from our method is shown in (c) without the spatial coherence term and (d) with the spatial coherence term. As is evident, the region inconsistency artifacts in (c) are significantly reduced in (d).” [WLH⁺07].

these rendering parameters.

From the point of view of addressing harsh illumination, this thesis could be compared to Wang et al. [WLH⁺07], yet, they work on gray-scale images under relatively simpler lighting conditions and do not claim to estimate the harsh *lighting*, rather the facial texture from a face image with harsh illumination. Also there, a 3DMM is used to estimate the facial model and alignment from the input image (see Section 2.3). Their most significant contribution is rather to the intrinsic texture decomposition (see Section 2.3.3). In single-image inverse lighting of faces, e.g. [WLH⁺07], [AS13] and [LZL14], the use of spherical harmonics keeps the algorithm from estimation of realistic effects such as specular highlights and cast shadows, by lighting model. This thesis aims at changing this paradigm and motivates the future use of a physically plausible model for lighting.

While monocular image inverse rendering has delivered good results for images with uniform or simple lighting situations, it is still an open problem in Computer Vision when it comes to complex lighting. Therefore, some research projects propose to circumvent harsh lighting or in a more progressive manner use the information that the harsh illumination effects reveal about the face. Zhao et. al. show that even without 3D geometry, it is possible to perform illumination invariant face recognition [ZSK13], whereas Romdhani and Vetter use the illumination effects (specular highlights) for a better multi-feature face model fitting [RV05]. Similarly, Kemelmacher-Shlizerman and Basri take advantage of the nonambient illumination of input images to infer the surface normals. Their model of lighting is based on spherical harmonics. They show that even 4 harmonics model most of the data. Their results on face model estimation from real images are promising [KSB11].

Because results of the related work are on simpler illuminations, it is difficult to compare them with our results on more challenging input images. Previous single-image methods

rarely make any claims towards the replication of cast shadows or inference of a physically plausible model for lighting. Although [WLH⁺07] consider cast shadows as an unwanted artifact, they remove it with image processing tricks, i.e. the ratio image method and MRF penalty term, and not through lighting estimation (see Section 2.3.3). As a result, their estimated lighting might fail to replicate the original lighting effects in case of harsh illumination.

In contrast to previous work, the proposed method considers different pose, complicated high and low frequency illumination effects, colorful, multi-directional, hard and soft lightings with elaborate cast shadows and highlights. The estimated colorful lighting can be used to render images of the same object or other objects with classic rendering procedures from Computer Graphics. A hierarchical Bayesian approach with hyperparameters is applied to promote illumination complexities, such as cast shadows, and to automatically handle adverse phenomena that the model and its formulation cannot capture, e.g. occlusions. An illumination-friendly and geometry-based data reduction is proposed that cancels the image noise and reduces the complexity of the optimization (see Section 3.4.2). The proposed inverse lighting provides a discrete environment map that can be used to reconstruct lighting effects or creatively design novel lighting conditions (see Section 3.6.2).

2.1.3 Lighting Design

Professional lighting in studios need expensive equipment, expertise and time. Moreover, most of the photos of digital era are taken under uncontrolled lighting conditions by end users. The goal of lighting design, as far as this thesis concerns, is to enable the user to make decisions about the lighting of the scene. Birn [Bir00] and Akers et al. [ALK⁺03] discuss different functional reasons of lighting design, both as a concept and as a tool in the hands of artists or scientist. A survey of lighting design methods is provided by Kerr and Pellacini [KP09]. A challenge of working on real images is to acquire the surface normals. Okabe et al. propose to rely on a coarse normal map, drawn by the user in a single view of a scene [OZM⁺06]. Henz and Oliviera propose a method to apply artistic lighting on paintings. They get the coarse shading of paintings from the user and refine them by shading-color correlation [HO15]. Some image-based lighting design methods [ADW04, PTMD07] promise higher quality, however, are restricted to applications where high quality light stage data of the scene are available. Mohan et al. use an area light to capture many (less than 200) images of a static relatively small object under different lighting [MTB⁺05, MBW⁺07]. They estimate a physically plausible lighting from sketches on an image of that object. Thereby, they use an out of the box optimizer for a constrained least squares cost function. For a survey on image-based relighting see [CCH07].

The Ambiguity of Paint-Based Lighting Design

Kerr and Pellacini [KP09] explain that users perform poorly with paint-based methods because they tend to sketch rather than accurately paint goal images. In contrast with the paint-based methods that they refer to, the proposed method focuses on the face. Hence, the user works more goal-oriented with a more familiar geometry. The algorithm performs well on both sketches and goal-based drawings, by incorporating the 3DMM framework, and well-designed priors on the lighting estimation (see Chapter 4 and 5).

The proposed method is a single-image sketch-based approach which works on real images of faces, taken under unknown condition. No hardware or extra data is required. It can be operated by inexperienced users, also as an integrated part of an image editing tool. In contrast with other lighting design methods [ADW04, PF92, PRJ97, CSF99, PTG02, SL01, HO15, AP08], the proposed algorithm focuses on human faces, which are nonconvex real objects with complex and regionally varying reflectance behavior. Hence, this method is specifically designed to replicate realistic illumination effects that apply to human skin and facial shape, e.g. the skin’s glossy behavior in grazing angles, colorful illumination and cast shadows.

Unlike other single-image and even some data-driven inverse lighting algorithms [GBK01, WLH03, FBS05, ZS06, WLH⁺07, AS13, LZL14], realistic specular reflection in grazing angles, cast shadows and color correction of the rendered result with respect to real images are addressed both in the lighting model (introduced in Section 3.2.1) and in the reflectance model (see Section 2.5), and thus in the resulting generating set, \mathbf{C} . The generating set is used to build a cost function and a nonnegative gradient-based optimization is designed to estimate the lighting model parameters by minimizing that cost function. The estimated lighting is then used to calculate the illumination on the face model, refine the intrinsic face model with deilluminated facial pixels from the high quality input image and render the face in new poses and under novel illuminations.

So far, no reliable quantitative evaluation method has been available for realistic single-image inverse lighting, therefore, the focus is on providing qualitative experiments by discussing challenging tasks, such as relighting and intrinsic texture decomposition, on challenging inputs, in the Chapter 5. Moreover, a quantitative measure for the performance of the illumination estimation is proposed, which is inspired by the Structural Similarity measure of Wang et al. [WBSS04] (see Section 5.1). Because the rest of the thesis focuses on the implemented algorithms and their results, all the basics are covered through the rest of this chapter, starting with a brief review of the facial appearance.

2.2 The Appearance of Human Faces

Faces make very interesting topics of study in different disciplines. Human faces are non-convex geometric objects with generic and specific features. On the one hand, their generic features, such as constellation of facial components (eyes, nose, ears and mouth), inspire modeling and generative approaches. On the other, the specific features of each face, such as subtle differences in skin tone, properties of the wrinkles and moles make the study of each individual face a great challenge for experts of different disciplines. Igarashi et al. name four domains that are interested in the appearance of skin: Computer Graphics, Computer Vision, Medicine and Cosmetology [INN07]. In this survey, they distinguish between three levels of skin components with respect to size. Refining the human skin taxonomy, in Figure 2.3, these three levels are divided to six sublevels in total. In the rightmost column, a corresponding physical phenomena or model for each sublevel is drawn.

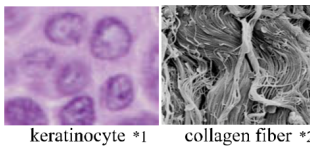
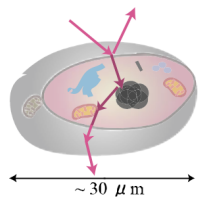
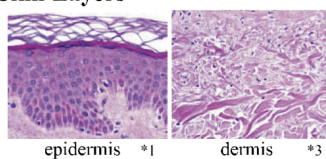
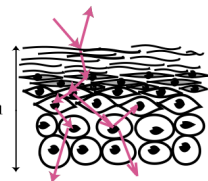
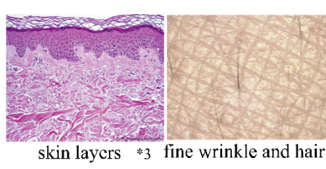
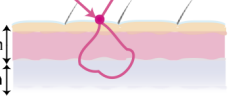
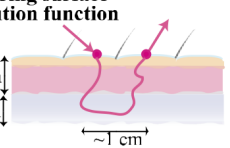
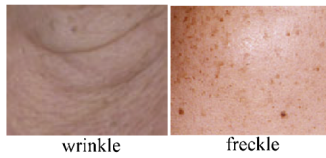
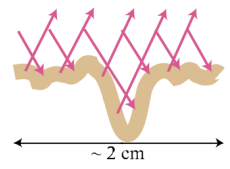

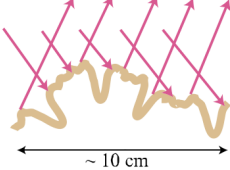
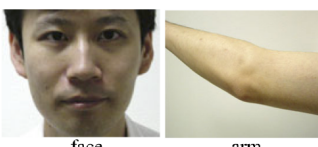
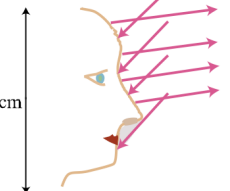
Microscale components are in cellular level and exist through the layers of skin. Although these components are hardly visible, they directly influence the reflectance of the skin. Even their chemical properties derive the appearance of larger components, hence, they need to be considered in skin appearance analysis. The microscale components, which are indistinguishable to the naked eye, remain obscured in a normal RGB portrait. Moreover, such components are under the influence of bodies biochemical and neurological control systems, therefore, they undergo constant changes, leading to unaccountability of measurements in such a fine scale.

Mesoscale components are visible and to some extent even distinguishable from each other, fine wrinkles and hair, skin surface lipid and other skin features, such as pores, moles, freckles, belong to this category. The nonrigid behavior of faces influences the mesoscale components. Wrinkles might appear in one gesture or disappear in another. In the course of time, many mesoscale features show up and vanish inexplicably. Cula et al. explain the complexity of skin appearance under different imaging conditions [CDMR05]. For instance, they show that a frontal lighting leads to a better capture of skin color (albedo), while fine surface geometry is more visible under inclined lighting angles.

Larger components, e.g. nose, eyebrow and lips, are categorized under macroscale components. On the one hand, it is necessary to model all the levels of detail if a realistic appearance of a face is desired. On the other, the task of reconstructing the skin appearance has proven to be a challenge in all three levels. The macroscale components might be invisible in some pose or under occluding geometry, e.g. glasses, hands and beard, or under darker shadows. Also, the unknown optic of the camera distorts the appearance of the macroscale components. Faces change their geometry to form gestures or speak. They also change their form depending on the biochemical conditions of an individual between

two different pictures of the same person. The use of CGI in films shows that the observers might be satisfied with a much simpler estimation of the whole system. Moreover, the lack of depth and reflectance data in a 2D image leads to ill-posedness in the estimation of an exact model of the face. Thus, measurements and estimation of face appearance from single images might stay an open problem.

A single image shows the appearance of the face from one direction, under one lighting situation, both of which unknown. Under such circumstances, the perspective projection, the 3D geometry, surface normals, albedo and reflectance are all missing. Under inconvenient imaging conditions, in the areas of saturated highlights or underflowed shadows, no detail information about the objects appearance are stored. To propose a solution to such problems, we use a software framework that tackles the human face appearance modeling from a single image.

Scale	Level	Physiological / Anatomical Components	Physical Phenomena / Models
Micro	1	<p>Cellular Level Elements</p> <ul style="list-style-type: none"> • keratinocyte • melanocyte • erythrocyte • collagen fiber . . .  <p>keratinocyte *1 collagen fiber *2</p>	<p>cellular optics</p>  <p>~ 30 μm</p>
	2	<p>Skin Layers</p> <ul style="list-style-type: none"> • epidermis • dermis • subcutis  <p>epidermis *1 dermis *3</p>	<p>cutaneous optics</p>  <p>0.04 ~ 1.6mm</p>
	3	<p>Skin</p> <ul style="list-style-type: none"> • skin surface lipid • hair • skin layers • fine wrinkle . . .  <p>skin layers *3 fine wrinkle and hair</p>	<p>bidirectional reflectance distribution function (BRDF)</p>  <p>0.5 ~ 4.0 mm 4.0 ~ 9.0 mm</p> <p>bidirectional scattering surface reflectance distribution function (BSSRDF)</p>  <p>0.5 ~ 4.0 mm 4.0 ~ 9.0 mm</p> <p>~ 1 cm</p>
Meso	4	<p>Skin Features</p> <ul style="list-style-type: none"> • wrinkle • pore • mole • freckle...  <p>wrinkle freckle</p>	<p>bidirectional texture function (BTF)</p>  <p>~ 2 cm</p>
	5	<p>Body Regions</p> <ul style="list-style-type: none"> • nose • finger • elbow • knee ...  <p>nose finger elbow</p>	<p>region appearance</p>  <p>~ 10 cm</p>
	6	<p>Body Parts</p> <ul style="list-style-type: none"> • face • arm • leg • torso ...  <p>face arm</p>	<p>part appearance</p>  <p>30 cm</p>
Macro			

*1 Photo courtesy of Christopher Shea, MD, Duke University Medical Center.

*2 Photo from Nanoworld Image Gallery, Centre for Microscopy and Micronanoanalysis, The University of Queensland.

*3 Photo courtesy of T.L. Ray, MD, University of Iowa College of Medicine.

Figure 2.3: The taxonomy of skin from [INN07].

2.3 Face Model Extraction with 3DMM from a 2D Image

When adequate measurements are available, each one of shape, reflectance and lighting can be calculated from the other two. This triangle has made a field of continuous work and constant improvements [HB89, Mar98, ZTCS99, DHT⁺00, RH01c, BF01, PP03, PF06, WMP⁺06, DFS08], to name a some of them. In the absence of 3D scans, Suwajanakorn et al. show that an illumination-invariant ‘persona’ of a face can be estimated from images of that face, taken from Internet [SSKS15]. Many adverse visual effects of lighting disappear in the result texture, after a multi-scale blending of image data. Liang et al. estimate the head from hundreds of photos [LSKS16], where the illumination effects are averaged out, similar to [SSKS15]. Piotraschke and Blanz achieve promising results with much fewer images by a regionally supervised averaging on the result of the 3DMM fittings [PB16], which estimates and removes the lighting from each single image.

In a single image, however, the only available data is the gray scale or RGB radiance, modulated by unknown camera and post-processing. Nonetheless, the problem of estimating shape, illumination and reflectance from the image can be viewed as one of a statistical inference nature [BM15]. Specifically for faces, generative models lead the model estimation and image synthesis for a long time. Turk and Pentland propose the idea of 2D eigenfaces in [TP91]. They build a basis of eigenfaces from frontal face images. The basis supposedly produces any given frontal face image. The eigenfaces span a pose-dependent face space in 2D.

The 3D Morphable Model (3DMM) is introduced by Blanz and Vetter for analysis and synthesis of human faces [BV99]. With the estimation of the high quality 3D model and its dense correspondence with the input image, the idea of 3DMM sets a new milestone in human face modeling. Subsequent works on 3DMM have established their usability and significance in the field for a longer time, e.g. [BV03, BMVS04, RV05, PKA⁺09b, AS13, SB15b, PB16, SEMFV16]. Paysan et al. have published the Basel Face Model (BFM), which is distributed for noncommercial use by University of Basel [PKA⁺09b]. Relying on an already great body of literature, this study avoids a thorough survey of 3DMM and 3D face reconstruction from 2D image. However, a minimum explanation of the method [BV99, BV03] seems necessary.

2.3.1 The 3D Morphable Model (3DMM) of Faces

The 3DMM is based on the linear combination of 200 colorful laser scans of young adults (with CyberwareTMScanner). After pre-processing the faces are represented in approximately 70,000 points. The face model is separated in a 3D geometry and a 2D colorful texture components, which are in correspondence with each other, for each single face and across the 200 face models. This data set is segmented in independent regions (subspaces) on the

face, e.g. eyes, nose, mouth and rest. The subspace of faces is then transformed to an orthogonal coordinate system with Principal Component Analysis (PCA) [Jol02] on the 200 scanned models. The PCA also aids the dimensionality and noise reduction. This spans a face space of shape vectors $\vec{s} \in \mathbb{R}^{3m}$ and texture vectors $\vec{t} \in \mathbb{R}^{3m}$, where $m \approx 70,000$. The linear combinations of the \vec{s}_i and \vec{t}_i ($i \in [1..n]$) construct a face:

$$\vec{s}_{mod} = \sum_{i=1}^n a_i \vec{s}_i \quad , \quad \vec{t}_{mod} = \sum_{i=1}^n b_i \vec{t}_i \quad (2.1)$$

where $n \leq 200$ and each index i signifies a principal component (the regional separation is suppressed in the formulation). The morphable model is a span over $(\vec{s}_{mod}, \vec{t}_{mod})$ pairs with parameters $\vec{a} \in \mathbb{R}^n$ and $\vec{b} \in \mathbb{R}^n$. Each given face can be modeled by finding the appropriate \vec{a} and \vec{b} . In practice, n is less than 100 [BV99].

2.3.2 3DMM Fitting Algorithm and Joint Lighting Estimation

The 3DMM fitting is initialized by the input image and a few landmarks that are set manually by the user on the face features (eyes, nose, lips, etc.). It estimates a high quality face model and imaging parameters for the input image (see Figure 2.4). With the help of a pose-invariant landmark localization algorithm [ZR12, BAPD13, JCK15], it is possible to make a fully automatic fitting algorithm [SPB16]. A different concept, proposed by Schonborn et al. [SEMFV16], implements a MCMC (Markov Chain Monte Carlo) approach to fully automatically fit the 3DMM to a single image of a face. They compare their method to state of the art [BV03, RV05, AS13]. In [SPB16], the idea of a fully automatic fitting is addressed, thereby the user interaction is removed with the landmark localization method from Zhu and Ramanan [ZR12]. A fully automatic fitting is out of the scope of this thesis, however, for a fully automatic fitting to satisfy the demands of the proposed inverse lighting, not only is the 3D model needed to be estimated properly, but also its dense correspondence with the input image. A precise correspondence is crucial around the silhouettes and contours of the face, because the inverse lighting algorithm solves a commutative cost per pixels and background pixels that are wrongly matched with face model will cause wrong illumination effects.

The 3DMM fitting algorithm [BV99, BV03], initialized with the input image and the landmarks, delivers an estimated face model (3D shape and texture), its alignment (pose, size and perspective projection), scene parameters (a simplified lighting) and imaging parameters (color management and camera properties). A matching between the 3DMM and the face from an input image is provided with the following minimization:

$$\underset{\vec{a}, \vec{b}, \vec{p}}{\operatorname{argmin}} \|\mathbf{I}_{input} - \mathbf{I}_{model}\|_2^2 \quad (2.2)$$

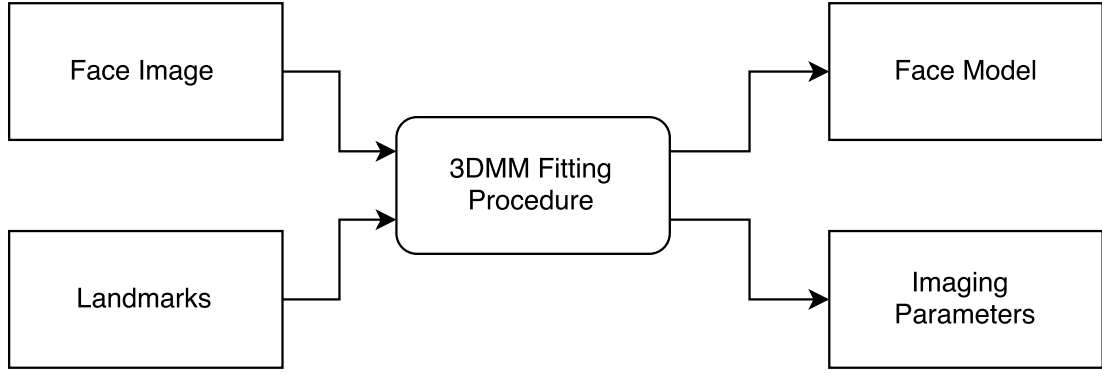


Figure 2.4: An overview of the 3DMM fitting algorithm. Input: Face Image and Facial Landmark positions on it. Output: Face Model (Shape, Texture, Pose), Imaging Parameters (Color Management, Alignment of the face in the image)

where I_{input} the colorful input image, I_{model} the reconstruction with the model, $\vec{\alpha}$ and $\vec{\beta}$ the shape and texture coefficients corresponding to \vec{a} and \vec{b} from (2.1), and $\vec{\rho}$ represents the other imaging (rendering) parameters. The vector $\vec{\rho}$ stands for parameters such as camera position, object scale, image plane transformations, lighting parameters, and color correction. The last two items are directly connected with this project and are explained in depth later. For now, it is sufficient to know that all the above parameters are being optimized together and the rest of the rendering parameters, such as reflectance functions or camera distance are set to constant in the algorithm. The rendering is done with a Phong illumination and Phong shading. The reconstruction image is formed with perspective projection into the image plane.

This Least Squares optimization (2.2) is an ill-posed inverse problem that is regularized with priors on $\vec{\alpha}$ and $\vec{\beta}$, calculated from the data set and an *ad hoc* prior on $\vec{\rho}$. The cost function complies with a Maximum A Posteriori (MAP) estimation, according to the Bayesian statistics. The optimization is a stochastic gradient descent. As far as we are concerned, this algorithm fits the 3DMM and the imaging parameters to the single input image, providing directly the \vec{a} and \vec{b} coefficients to build the face model $(\vec{s}_{mod}, \vec{t}_{mod})$, corresponding to the face in I_{input} . Thus, we have the estimated 3D shape and texture of the face, stored as a 3D mesh and a 2D texture. As mentioned, the fitting algorithm also provides the pose and alignment of the face in the image and the color correction parameters for the input image. These are necessary parameters for the further steps of the proposed inverse lighting.

2.3.3 Model-Based Texture Extraction from 2D Image

The estimated 3D model does not include microgeometry –mesoscale skin features–, such as fine wrinkles, hair and moles. Instead, these are represented in a low spatial frequency, mostly influencing the coarse color of the skin in the respective region. However, to achieve better reconstruction quality, the 3DMM framework allows to extract these high

frequency features of the skin in a subsequent procedure. Incremental work on 3D morphable models [BV99, BV03, BMVS04, RV05, PKA⁺09b, AS13, SEMFV16] still lack the necessary detail for generative modeling of the mesoscale skin properties and hair. This is a systematic problem of PCA-based methods, because PCA, due to the averaging and generalization of features in the most significant eigenvectors, works as a low-pass filter. Thus, the micro-geometry features are neglected as part of the high spatial frequency signal.

Nonetheless, fitting a 3DMM delivers the most promising result for 3D modeling from a single 2D input image. To add detail to the estimated 3D model, the 3DMM framework uses the pixel values from the original input image and rewrites the estimated texture with the de-illuminated pixel values. As an extension to this idea, the MRF term of Wang et al. covers the inconsistencies in the estimated 3DMM facial texture, be they caused by specularity, cast shadow or occlusion [WLH⁺07]. Also, they calculate the ratio between the input image and a blurred version of the input image. As far as their explanation implies, a ratio image is calculated according to (2.3).

$$\text{ratioimage} = \frac{\text{image}}{\text{gauss}(\text{image})} \quad (2.3)$$

where, $\text{gauss}(\cdot)$ is a Gaussian blur. The ratio image does not contain the intensities of shadow or highlights; it only contains the high frequency range. When the ratio is multiplied to the estimated texture, it applies the high frequency to the texture and makes it richer in detail without changing the illumination (see Figure 2.2). Theoretically, the sharp edges of differently illuminated areas, such as hard shadow edges, are included in the ratio image and transferred to the texture. This refinement works only on visible areas of the face, which are not in absolute highlight (pixel values ≥ 255) or absolute shadow (pixel values = 0).

The 3DMM framework [BV99] applies the texture from a visible symmetric area of the face to the invisible area whenever possible. There are some recent independent attempts, by Schumacher et al. [SPB15] and Dessein et al. [DSWH15], to add detail to the facial texture of 3DMM even in the absence of a high quality input image. The idea is roughly to adopt the missing details from other faces.

The 3DMM fitting provides a dense correspondence between the pixels of the image I_{input} , vertices of the model \vec{s}_{mod} and the texels of the texture \vec{t}_{mod} . This correspondence, together with an elaborate inverse lighting, allows us to de-illuminate the image pixels and use them as texture values in \vec{t}_{mod} . For each pixel of the face from I_{input} , its value is decomposed into an intrinsic texture by inverting the color correction (explained in 2.5.5) and removing the illumination. The intrinsic value is accepted for the respective texel directly as long as the respective vertex is visible in the input image. Assuming faces are symmetric, the extracted values from visible areas of the face in the input image can be used for invisible

areas on the opposite side of the face model. The algorithm also interpolates between the intrinsic values that are read from the input image and the values estimated by the 3DMM fitting that are already stored in \vec{t}_{mod} , depending on the angle between the surface normals and the view direction. This is to keep a smooth transition between the intrinsic texture that is decomposed from the image for visible areas and the 3DMM texture which is the best possible choice for pixels, for which no intrinsic value is readable.

To reconstruct the scene, the estimated face model is rendered with the improved texture values under the estimated illumination and color correction [BV99]. With a more precise illumination estimation, this thesis contributes to the results of the intrinsic texture decomposition. In contrast with ratio method [WLH⁺07], the quality of extracted texture values from the image directly depends on the estimated illumination. Therefore, the improved texture is proposed as a qualitative measure for the performance of the inverse lighting algorithm. Whenever the illumination effects do not show up in the intrinsic texture, they are captured by the estimated lighting model. The ultimate inverse lighting algorithm is supposed to lead to a perfect intrinsic texture decomposition. In Chapter 5, we see that the proposed method achieves distinct improvements towards this ultimate goal.

2.4 Illumination Cone

A linear cone is a subset of a vector space, such as $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots, \vec{x}_n\}$, which is closed against multiplication with positive scalar values. If the subset is closed against addition too, then it is a convex cone [Ber09]. This definition is given in (2.4).

$$\{\vec{x} \mid \vec{x} = \sum_{i=1}^n a_i \vec{x}_i, \quad a_i \in \mathbb{R}_0^+\} \quad (2.4)$$

where $n \in \mathbb{N}$.

Belhumeur and Kriegman explain the *illumination cone* of a face as a set of images of the face under all possible lighting conditions [BK98]. They prove that this set of images is a convex cone [BK98, BKY99] and [GBK01]. They explain that a basis for the illumination cone can be built through singular value decomposition of a few captured images of the same face under all the same conditions but different lighting. Moreover, it is possible to estimate the face model from a single image [AS13] or with stereo algorithms from multiple images [GBK01] to analytically build or render a generating set. In addition to the face model and reflectance, a lighting model is necessary for the synthesis of the generating set. Each captured or rendered image of the face is an element of the illumination cone with all the same imaging properties, except for lighting, subjected to Lambert reflectance and convex geometry [BK98]. These two assumptions are necessary to avoid nonlinear

terms, such as cast shadow or specular highlights, which interfere with the convexity of the cone and the formulation of the generative model. Nonetheless, it is necessary to mention two problems that come up while dealing with captured images.

Offset in Captured Images

Moreover, for real images that are captured in uncontrolled conditions, the response function of the camera sensor or post-processing on the captured image might add nonlinearities to the image. Even if these effects are limited to a constant offset, this offset is not negligible because it violates the definition of convex cone. Let us assume \mathcal{V} is a convex cone of images, so that the addition of two members of this set is also a member of the set

$$\forall I_i, I_j \in \mathcal{V} : I_i + I_j \in \mathcal{V}. \quad (2.5)$$

Let us define another set \mathcal{V}^c by adding a constant $c \in \mathbb{R}$ to the images of \mathcal{V} :

$$\mathcal{V}^c = \{I^c \mid \exists I \in \mathcal{V}, I^c = I + c\} \quad (2.6)$$

To show that \mathcal{V}^c is not a convex cone, it is enough to prove that it is not closed against addition. By adding two distinct members of \mathcal{V}^c , such as I_i^c and I_j^c , we see that the result does not belong to the set \mathcal{V}^c because

$$I_i^c + I_j^c = I_i + c + I_j + c = I_i + I_j + 2c, \quad (2.7)$$

which is not a member of \mathcal{V}^c according to the definition of \mathcal{V}^c in (2.6) and the fact that $\forall c \neq 0, c \neq 2c$.

In this dissertation the generating set is rendered without an additive constant. However, an input image might have an offset. Spherical harmonic solutions (see Section 2.4.1) include an additive constant harmonic, i.e. the first term, which models this offset. Unlike an spherical harmonic basis, the basis in this thesis is a set of rendered images which does not include any constant term or offset. In Chapter 3, the offset is modeled separately in a color correction term (see Section 2.5.5). Thereby, the nonlinearity of the sensitivity curve of the (unknown) camera is ignored to simplify the color correction model. The color correction also models the image tint and facilitates the analysis and synthesis of gray-scale images as well as images with a very extreme color saturation, e.g. greenish, blueish and yellowish tints. Only with such a color correction, the proposed analysis by synthesis approach is properly applied to images from unknown sources, gray-scale images and painted portraits (see also Section 3.2.3 and Chapter 5).

Digital Images Store Clipped Signals

Another issue rises by the fact that RGB values of pixels are bounded by two numbers, e.g. $[0, 1]$ or $[0, 255]$, in digital images. As a result of harsh illumination or suboptimal camera settings, pixels of an area of the image might be saturated to the upper bound, $(r, g, b) = (1, 1, 1)$, and another area might be blank, $(r, g, b) = (0, 0, 0)$. At these positions, the information is lost. On the one hand, HDR (High Dynamic Range) imaging techniques can be used to register the information that is lost due to the limitations of the dynamic range. On the other hand, the proposed algorithms are intended to be not restricted to an imaging technique. Because we want to estimate lighting in harshly illuminated images, completely ignoring such areas is also not an option, because they provide cues about the areas where harsh lighting effects happen. Instead, the proposed algorithms in Chapter 4 introduce a regularization term that prevents overfitting. Moreover, with the hyperparameter optimization approach in Section 4.4, these areas are automatically suppressed whenever they have adverse effects on the optimization process. Experimenting with HDR input and a thorough comparison of the performance of the algorithms on HDR and non-HDR images are out of the scope of this dissertation.

2.4.1 Lighting Models

A computational representation of lighting might be inspired by physical lighting methods, with light sources, or by numeric methods such as spherical harmonics. The lighting representation is different from, but in direct relation to the reflectance function. This thesis avoids the use of nonrealistic lighting models, including an ambient term. An **ambient light** has no position or direction. It reaches equally all the surfaces from all directions. Ambient light can be defined with three degrees of freedom, e.g. (R, G, B) . The Phong model (Section 2.5.2) usually contains an ambient term. We show that for a physically plausible lighting, a limited number of light sources can produce not only harsh illumination conditions but also close to ambient lighting.

2.4.2 Light Sources

A **point light** is a light with a location in space. We need to know its position and color (x, y, z, R, G, B) , where (x, y, z) is not normalized or there is another value for distance. A **spot light** is similar to a point light, however, it illuminates only a known solid angle in an specific direction. The color, position and solid angle are known properties of an spot light. An **area light** is a surface of emitting area with known size, form, and surface normal in a known distance from the object. A **directional light** is a light source with known color and direction. It can be represented with 6 parameters (x, y, z, R, G, B) , where the direction vector (x, y, z) is normalized. In common practice, each light source in far distance (> 10 meters) is considered to be in infinity.

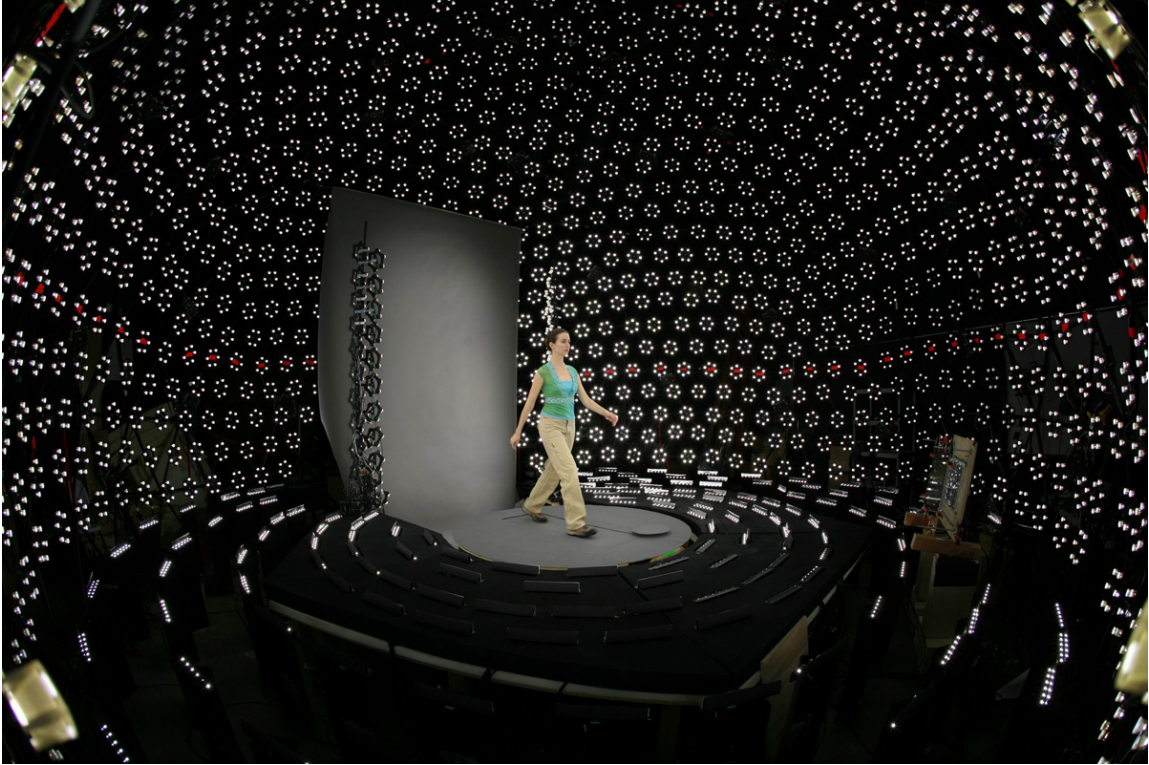


Figure 2.5: This is the light stage LS6 of USC ICT in LA. Picture is taken from ‘www.fxguide.com’

2.4.3 Light Stage

Although other configurations exist, a light stage is usually a dome or a full sphere of concentric light sources [MDA02, DWT⁺02, HED05, MMS⁺05, USC08, Deb12, WK15]. The object of interest is positioned in the center and its appearance is captured under different (all possible) lighting conditions. Because of superposition principle, it is enough to capture the appearance under all possible single light sources from the light stage. A real light stage might be equipped with multiple high quality video cameras, 3D scanners, optical filters and more. The cost of making and maintaining such labs can be very high.

According to Debevec, an environment map can be approximated with a discrete representation [Deb06], hence, a Light Stage represents the environment lighting of the scene. In Section 2.2, we saw that human skin has a complex and rough surface which in general does not act as a mirror, therefore, only a discrete approximation of the environment map is enough to model the global lighting for faces.

Spherical Harmonics

Spherical harmonics are not used in this thesis, however, they are reviewed here as an alternative approach to represent lighting in a linear system. MacRobert explains the historical background and formalism of spherical harmonics [Mac48]. Cabral et al. [CMS87] and Sillion et al. [SAWG91] use them to represent BRDFs. Later, Ramamoorthi and Hanra-

han [RH01b, RH01a] and Basri and Jacobs [BJ03] independently explain a linear modeling of lighting with spherical harmonics. The real spherical harmonics are the angular part of a solution set to the Laplace differential equation and build orthogonal coordinates in spherical space. These basis functions are similar to a Fourier basis. Most of the generative methods, e.g. [BK98, BKY99, GBK01, AS12, AS13], and many other approaches that have a Lambertian reflectance and convex geometry assumptions, e.g. [WLH⁺07, KSB11], use spherical harmonics representation of lighting. Some of them are reviewed at the beginning of this chapter.

Spherical harmonics are not physically plausible; they contain an ambient-like term, i.e. the first harmonic, yet, negative values appear in the basis from the second harmonics. Moreover, they do not cause a global illumination effect, such as cast shadows or even local phenomena, like specularities. Other decomposition methods, e.g. SVD, to build the illumination cone's generating set lead to similar limitations. Ramamoorthi and Hanrahan show that although illumination estimation from a Lambertian surface is ill-posed, it is still possible to estimate radiance from irradiance when spherical harmonics are used [RH01b]. They argue that the 'perturbation' of lighting –the components of the estimated spherical harmonics coefficients– might have both negative and positive components, with the only physical requirement that the net lighting is nowhere negative. Having said that, they admit that the lighting *model* must be positive to be physically possible, and recognize the importance of this condition as a further constraint on allowable perturbations.

Here, instead of the spherical harmonics or any other linear decomposition for lighting representation, the elements of the generating set are rendered faces under single directional light sources of a virtual light stage (see Section 3.2.1). A weighted summation of these generating elements spans the subspace of all possible lighting conditions for the given input image. Based on the superposition principle of light, the weights are equal to RGB values of the respective light sources for the reconstruction of the input image. Thus, these weights are a solution to the problem of inverse lighting from an analysis by synthesis point of view.

2.4.4 Formalizing the Superposition Principle

The implication of the superposition principle for a rendering machine can be explained by an equation between a rendered image I_L under a set of light sources L and the summation of some other images I_i , each rendered under a single light source l_i of the set $L = \{l_1, l_2, \dots, l_n\}$, according to (2.8)

$$I_L = \sum_{i=1}^n I_i \quad (2.8)$$

In Chapter 3, we assume that the input image is achieved similar to I_L and the set L can be represented by light sources of a virtual light stage (VLS). Then, we estimate the intensities and color of the light sources of the VLS so that the equation (2.8) applies with minimum error for all the three color channels.

2.4.5 Estimation of RGB Parameters of Light Sources

In Chapter 3, a generative model is built to represent the lighting. Moreover, the task of inverse lighting is reduced to estimation of coefficients for this generative model. This kind of problem is usually solved with direct search [HJ61], iterative optimization algorithms [Luc74, Ric72] or with a gradient based approach for a linear Least Squares cost function. Schmidt gives a practical survey on optimization algorithms for Least Squares (LS) [Sch05], comparing L2 regularization (Tikhonov regularization) with benefits of L1 regularization (LASSO) algorithms from [Tib94] and [CDS01]. Summarizing a great body of literature, Chen and Plemmons present a survey [CP09] on nonnegative optimization. Because a great deal of effort has been put in finding a proper cost function and manipulating the optimization procedures, the whole Chapter 4 is dedicated to this topic.

To construct the generating set, which is a necessary part of the cost function, and to produce the results, we need an image formation function that accepts light source directions and RGB values as parameters and renders I_i and I_L images. The same rendering function must be used to render the generating set, the reconstruction, relighting and other results; see Chapter 5.

2.5 Image Formation

The proposed algorithm is an analysis by synthesis method, therefore, we need to understand the imaging procedures. To synthesize an image, the physics of the 3D scene must be described computationally and then the projection of the scene into a 2D image of given resolution must be calculated. The former is called (forward) rendering and the latter is rasterization. The imaging procedures of the 3DMM framework also take care of the color correction for reconstruction of real images with different colorful saturation, contrast and brightness. This section explains these procedures. The applied reflectance function in Chapter 3 is explained here in Section 2.5.3. Moreover, we take a look into the shadow mapping method in Section 2.5.4 before explaining the color correction model of 3DMM in Section 2.5.5.

2.5.1 Rendering

In a captured digital image, the pixel values are quantized representations of the radiance that leaves a small surface of the object in the camera's direction. The reflecting surfaces

location and size corresponds to the pixel location and size on the sensor, with respect to the camera optics. To synthesize an image, this recorded radiance is calculated (modeled) by rendering. In 3DMM framework [BV99], the rendering function requires 3D shape and surface normals, texture (albedo) at each vertex or triangle, the view direction, the RGB values of the ambient light and the RGB values and directions of light sources. Upon these parameters, the renderer calculates the specular and diffuse terms and puts them in the imaging function (2.9).

$$\mathbf{I}_L(p) = \mathbf{D}_L(p) \cdot \mathbf{M}(p) + \mathbf{S}_L(p) \quad (2.9)$$

where p is an integer number which is mapped to the row and column number of a pixel, $\mathbf{I}_L(p)$ is the pixel number p (p th pixel) of the image \mathbf{I}_L . This image is rendered under the light sources of the set L . The diffuse term at the p th pixel is shown by $\mathbf{D}_L(p)$, the texture with $\mathbf{M}(p)$ and the specular term with $\mathbf{S}_L(p)$. The rasterization detail and color correction are suppressed in (2.9).

Boyce defines reflectance for a diffuse surface as ‘the ratio of the luminous flux reflected from the surface, to the ratio of the luminous flux incident on it’ [Boy14]. In other words, it describes the light that the surface reflects toward the camera, with respect to the light that is arrived at the surface from the environment.

Reflectance is often modeled with a Bidirectional Reflectance Distribution Function (BRDF). Montes and Ureña give an overview of BRDFs, explain their categories and summarize their properties [MUn12]. The 3DMM uses the Phong model which this thesis replaces with a measurement-based Torrance-Sparrow for human face. Hence, these two BRDFs are explained next.

2.5.2 Phong Model

The Phong model of reflection [Pho75] is an empirical BRDF that accounts for three separate terms for non-Lambertian surfaces. The **ambient** term models the light that is equally scattered around the object. The ambient reflectance depends only on the intensity of the ambient light x_a . The **diffuse** term follows the Lambert reflectance according to (2.10) and describes the surface brightness, result of the diffusely reflected light. The **specular** term of Phong is given in (2.12).

$$D_i(p) = \langle \vec{l}_i, \vec{n}_p \rangle x_i \quad (2.10)$$

where $D_i(p)$ is the diffuse reflectance of Phong at pixel p for the light source i , \vec{l}_i a normalized vector pointing at the light source number i , \vec{n}_p the surface normal of the geometry at the 3D position on the mesh that corresponds to pixel p , and x_i is the intensity of the

light source number i . The diffuse term (2.10) is calculated for R, G and B color channels separately. Moreover, the diffuse term (2.10) must be calculated for all the n light sources in the set L and added to the ambient term, according to (2.11).

$$\mathbf{D}_L(p) = k_a x_a + k_d \sum_{i=1}^n D_i(p) \quad (2.11)$$

where the k_a and k_d are two empirical constants that are multiplied to the ambient and diffuse term respectively, x_a represents the ambient light intensity. The \mathbf{D}_L is the diffuse reflectance map for the whole image. It shows the diffuse shading all over the image in the color of the lighting. When applied to a face geometry the diffuse map should look like a face made of plaster. This diffuse map is multiplied by the modulation texture and added to the specular term according to (2.9). The total specular at location p –the $\mathbf{S}_L(p)$ from (2.9)– is calculated according to (2.13).

$$S_i(p) = \langle \vec{r}_{i,p}, \vec{v} \rangle^\alpha x_i \quad (2.12)$$

where $S_i(p)$ is the specular reflection toward the camera position, \vec{v} the view vector that points at the camera, $\vec{r}_{i,p}$ the perfect (mirror) reflection vector at the 3D position which corresponds to pixel p and for the light source number i . The perfect reflection vector is calculated as $\vec{r}_{i,p} = 2\langle \vec{l}_i, \vec{n}_p \rangle \vec{n}_p - \vec{l}_i$. Moreover, the constant α influences the shininess (or roughness) of the surface. Also, the specular term must be calculated separately for the three color channels and for all the light sources for each visible pixel according to 2.13 to give the specular map \mathbf{S}_L for the given face geometry and the set of light sources L .

$$\mathbf{S}_L(p) = k_s \sum_{i=1}^n S_i(p) \quad (2.13)$$

where k_s is an empirical constant which controls the overall proportion of the specular compared to the other two terms.

The fitting algorithm of the 3DMM estimates x_a , a single light source direction \vec{l}_1 and intensities x_1^R, x_1^G, x_1^B , however, the k_s and α are set empirically, and the k_a and k_d are set to 1. In Phong model only the proportion of the specular term to the rest is important and energy conservation and physical plausibility are sacrificed for the sake of simplicity [MUn12]. The Phong model ignores Fresnel term which dominates the appearance of the human face in grazing angles. Therefore, a more complicated BRDF is used in this thesis.

2.5.3 Torrance-Sparrow and Dipole Functions

Faces, with their complexities (see Section 2.2), show a wide range of effects under different lighting conditions. Their concavities cause cast shadows and their elaborate reflectance

leads to intense highlights in grazing angles. The reflectance of the skin changes across different regions of the face [WMP⁺06, JB02]. The recognizable illumination effects start from changes in diffuse color under different illumination colors, to hard and soft cast shadows depending on the lighting direction and softness, to different types of highlights appearing in mirroring angles and even more in grazing angles. Therefore, instead of the Phong model a realistic facial reflectance is used, which is more suitable for human skin according to Jensen et al. [JB02]. This BRDF has been used for the measurements of Weyrich et al. [WMP⁺06] at Mitsubishi Electric Research Laboratories (MERL) and ETH Zurich. Based on [JB02], the diffuse term is calculated according to the dipole function and then added to the Torrance-Sparrow term.

In an approximation of dipole function by Jensen et al. [JB02], the diffuse subsurface reflectance of light $R_d(x_i, x_o)$, entering at point x_i and exiting at point x_o , is modulated by the incident transmittance Fresnel factor F_{ti} and view angle transmittance Fresnel factor F_{to} , according to (2.14).

$$f_d \approx \frac{1}{\pi} F_{ti} R_d(x_i, x_o) F_{to} \quad (2.14)$$

The Fresnel effect and contributions are explained later in this section. Although, the skin surface is assumed to be smooth in this BRDF, the transmittance Fresnel terms, used in the dipole function (2.14), consider the diffuse subsurface light transfer [WMP⁺06].

Specular reflections are the reflections near the mirroring angle. They appear as highlights in the light source color, independent of surface albedo. In Torrance-Sparrow model [TS67], which is a theoretical BRDF with a validated [MWM⁺98] formulation, the specular term depends on the microfacet distribution D , a geometry attenuation term G , and the reflective Fresnel term F_r . While the dipole term (2.14) substitutes the ambient and diffuse term in the Phong model, the Torrance-Sparrow term (2.15) substitutes the specular term. Thus, this BRDF can be seen as an extension to the Phong model.

$$f_{TS} = \rho_s \frac{D G F_r}{\pi \langle \hat{n}, \hat{l} \rangle \langle \hat{n}, \hat{v} \rangle} \quad (2.15)$$

where G is geometry, and D is the Beckmann's microfacet distribution, F_r is the reflective Fresnel term and, \hat{n} the surface normal, \hat{l} and \hat{v} the vectors which point at the light source and the camera, respectively, and \hat{h} is the halfway vector between the camera and the light source vectors.

$$G = \min\left\{1, \frac{2\langle \hat{n}, \hat{h} \rangle \langle \hat{n}, \hat{v} \rangle}{\langle \hat{v}, \hat{h} \rangle}, \frac{2\langle \hat{n}, \hat{h} \rangle \langle \hat{n}, \hat{l} \rangle}{\langle \hat{v}, \hat{h} \rangle}\right\} \quad (2.16)$$

$$D = \frac{1}{4m^2 \cos^4 \delta} e^{-\left(\frac{\tan \delta}{m}\right)^2} \quad (2.17)$$

where δ is the angle between the surface normal and halfway vector. The constants m (for the roughness) and ρ_s and the diffuse albedo are provided by [WMP⁺06]. The 4 and π in the denominators of (2.15) and (2.17) might be absent or swapped with each other in some implementations; compare [Wey06, WMP⁺06] with [Kur11], p.245. In Section 3.2.2, we deal with this inconsistencies to use the ETH/MERL [WMP⁺06] reflectance function with the texture of the 3DMM [BV99].

Fresnel Effect

Reflective Fresnel contribution explains the proportion of the incident light that reflects due to the Fresnel effect. It is the contribution of the Fresnel factor to the specular term, with respect to the Fresnel effect. The Fresnel effect depends on the differences between the refraction indices of the two materials at the reflecting surface, i.e. air and skin. This contribution is significant at the grazing angles for human skin. Schlick approximates the reflective Fresnel contribution with respect to the angle θ between the halfway vector and the camera direction [Sch94]. In (2.18), n_1 and n_2 are the refraction indices of the adjacent materials.

$$F_r = R_0 + (1 - R_0)(1 - \cos(\theta))^5, \quad R_0 = \frac{(n_1 - n_2)^2}{(n_1 + n_2)^2} \quad (2.18)$$

where $n_1 = 1$ refractive index of air and $n_2 = 1.38$ is the average refractive index of skin [JB02, WMP⁺06]. Moreover, R_0 is the Fresnel reflection at 0 deg, also called specular color. The value of R_0 changes if we consider conductor material.

Transmittance Fresnel contribution explains the amount of light that enters the medium when the incident light meets the surface at a grazing angle. For an incident light, the transmittance Fresnel contribution added to the reflective Fresnel contribution must be equal to one, $F_{ti} + F_r = 1$, ignoring the parts that are wasted in form of heat [JB02].

2.5.4 Soft Cast Shadow Mapping

Attached shadow appears on the surfaces which are faced away from the light. For examples, in presence of one light source from the left side of a sphere, the right side of the sphere is under attached shadow. In contrast with attached shadows, cast shadows are global phenomena and not addressed by local BRDFs. Cast shadows appear on surfaces which are towards the light source, yet, their path to the light source is blocked by an opaque object. The area in complete darkness is *umbra* and the area close to the edges of the shadow which receives a part of the light is called *penumbra* [CPC84]. To calculate the cast shadows, the program produces a shadow buffer for each light source, in which the



Figure 2.6: Shadow buffer (right) for a light source that is used to render the face (left), similar to rendering of a z-buffer. The angle of the light source is $\approx (52^\circ, 20^\circ)$. Note that the shadowed areas are not visible to the light source, and thus hidden in the visualization of the shadow buffer.

brightness of the buffer at a given location is related to the distance that a surface or vertex of the mesh has from the light source; similar to a z-buffer, see Figure 2.6. Since we use directional light sources –light sources in infinity– the real distance from the light source is meaning less; only a relative distance is needed to decide which vertices or triangles are *seen* by the light source and which are occluded.

To visualize the shadow buffer for each directional light source, a coordinate system is built. Thereby, the horizontal and vertical axes are calculated and the scene is transferred to this coordinate system through parallel projection (or perspective projection in case of a point light). Then, the scene is moved to the center of the vertical and horizontal plane and scaled to be inside the frame, depending on the given size. After that, from the light source direction the shadow buffer is set for each triangle, depending on how far it is from an imaginary camera. In other words, the shadow buffer stores a depth buffer (z-buffer) from the light source point of view (see Figure 2.6).

For the rendering of shadows in the image, after the diffuse and specular terms are calculated, a shadow factor is produced depending on the condition that the vertex is visible in the respective shadow buffer (shadow factor = 1) or not (shadow factor = 0). If soft shadowing is active (to simulate shadows with a minimal penumbra), the shadow factor is calculated as a value between 0 and 1, which undergoes a smoothing by adding a fraction to the shadow factor for each pixel of the shadow buffer that is in shadow and close to one which is not in shadow, like blurring with a manually set window size. Soft shadowing smoothens the edges of the cast shadow area and avoids zig-zag shadow edges, that are caused by shadowing on triangles of the mesh. The window size is 9×9 in Figure 2.6 (left).



Figure 2.7: Compared to its gray-scale version (right), the yellowish tint of the colored image (left) is even more salient to the observer. The color correction term of the 3DMM framework models a tint that can be considered constant all over the image.

2.5.5 Color Correction

Due to different camera settings and post-processing, images differ in color contrast (see Figure 2.7) and brightness. The 3DMM framework models this with (2.19).

$$\begin{aligned}
 L &= 0.3R_{in} + 0.6G_{in} + 0.1B_{in} \\
 R_{corrected} &= (\xi(R_{in} - L) + L) G_r + O_r \\
 G_{corrected} &= (\xi(G_{in} - L) + L) G_g + O_g \\
 B_{corrected} &= (\xi(B_{in} - L) + L) G_b + O_b
 \end{aligned} \tag{2.19}$$

where R , G and B the pixel values in their respective color channels, L the color intensity, O_r , O_g and O_b the offset values, G_r , G_g and G_b the estimated gain for each channel and ξ is the color contrast [BV99].

The color correction is the final step of rendering. It is necessary for reproducing the color saturation and color contrast of the input image. It is mandatory for proper estimation of illumination from real images using a model-based approach. Applying this color contrast model enables the proposed algorithms in Chapter 3 to process colorful and close to gray scale images, such as examples C and H that are provided in the Chapter 5 in Figure 5.7. Generally, a color correction must be considered when the generating set and the input images are from different sources. Moreover, this is where a nonlinear sensor characteristic would have to be considered, which we do not.

2.6 Cast Shadow Detection and Segmentation

Kersten et al. [KMK97a] and Mammassian et al. [MKK98] discuss the human perception of cast shadows. With many experiments, they provide evidence for the winning position of cast shadows in perception of movement and depth against other cues in the scene. Knill

et al. explore the amount of information that shadows and especially cast shadows carry about the geometry of the scene and lighting [KMK97b]. Barje et al. findings show that humans do not make use of this information to infer lighting conditions from cast shadow. Moreover, cast shadows lead to lower performance in face recognition [BKTT98]. Therefore, cast shadow detection and segmentation is of a great interest in scene understanding community. Finding the lighting condition that causes a given cast shadow is accepted as a goal for this thesis; a goal that introduces a great challenge to the inverse lighting problem. The forward rendering of cast shadows is already explained in Section 2.5.4.

In Chapter 3 and 4, we see that to promote the appearance of the cast shadows in the results, it is useful to segment them in a preliminary step. Although shadows can be removed from the image similar to specular removal algorithms [ABC11], the segmentation of cast shadows can be helpful in finding the light source positions [PF92, PRJ97]. To detect occurrence of shadows, methods as simple as thresholding at a constant low pixel value [WLH⁺07] are used, however, previous work on shadow detection and removal [RKB05, SL08, PF92, PWSP11] demonstrate that shadow analysis is a challenging task. Thereby, the work of Ramamoorthi et al. takes a pioneering step in the formal analysis of cast shadows with convolution and Fourier decomposition [RKB05].

The proposed inverse lighting algorithm performs well on attached shadows because they usually appear on large areas and contribute significantly to the error per pixel in the proposed generative approach. Cast shadows, on the other hand, contribute a very small value to the cost function (see Chapter 4). In Section 3.3 a cast shadow segmentation algorithm is proposed that is used to mark the shadow area. The marked area is used in the hyperparameter optimization (Chapter 4) to promote the estimation of a lighting model that reconstructs the cast shadows of the input image.

2.7 Image Compression

While noise and facial micro and meso structures appear in high frequency domains, major lighting effects are usually in lower spatial frequencies on facial images. Therefore, a method of Image compression can be used to both reduce the noise and the redundancy of data and representation of the model. In Chapter 4, we see that the optimization algorithm can be very time and resource consuming. A well-defined image compression for the purpose of illumination effect analysis is a key feature of an efficient image-based inverse lighting. Without extreme reduction of the data, the proposed algorithms (two optimization algorithms are proposed in Chapter 4) fail to return a result in tolerable time.

2.7.1 Downsampling Filters

Image filters, such as the bilateral, Gaussian or binomial, are used to make images smoother. A downsampling with a blurring filter (to remove the aliasing) satisfies both needs of noise

and data reduction. Due to their ease of implementation and frequency of use in literature, it is safe to believe that they can be used also for the intended application of inverse lighting. However, they do not consider the semantics of the image; especially close to the boundaries of the face and the background, the colors and intensities might mix adversely in Gaussian or binomial filter, or a wrong background value might be selected for the silhouette of the face when the median filter is used (the median filter also interferes with the linearity of the generative model. See Chapter 4). Nevertheless, in Chapter 3, a version of the Gaussian downsampling is provided that preserves the face pixels in the region of silhouettes by avoiding the background.

2.7.2 Compressive Sensing Matrix

Section 2.7.2 is from the joint work [HCSBL15]. The compressive sensing method is the contribution of our co-authors, especially Miguel Heredia Conde, and the details of this approach are out of the scope of this thesis.

Another approach is the use of a sensing matrix. Usually, a Hadamard, a random binary or a random nonbinary matrix is used to scan the image into a small number of values, which represent the whole image. The theory of Compressive Sensing explains the benefits of using a sensing matrix for data reduction, promotion of sparsity and more, when applied on a generating set or a vector basis. A semantic-aware version of this approach is used in [HCSBL15] to solve the raised inverse lighting problem, with the benefits of time and resource efficiency and robustness to noise.

2.7.3 Superpixel Segmentation

Another method of down sampling is the superpixel segmentation. Superpixel approaches cluster the neighboring pixels, based on the pixel values, to deduce an indirect low level semantic. Each cluster (or segment) is represented by a *center*, which could be a mean value or a median of the colors and positions of all the pixels in the cluster. There are two superpixel segmentation groups, graph-based and gradient-based algorithms, which are thoroughly compared in [ASS⁺10, ASS⁺12]. There is no guarantee, for example in the widely used SLIC (Simple Linear Iterative Clustering) approach by Achanta et al. [ASS⁺10, ASS⁺12], that the foreground and the background are separated or superpixels are selected in a way which is suitable for illumination analysis on a 3D nonconvex object, such as a human face. Therefore, a major paradigm shift is necessary to adapt the idea of superpixels to the requirements of the proposed illumination estimation method (see Section 3.4.2). Here, we explain the currently state of the art superpixel algorithm of SLIC.

The clustering of SLIC starts with an initial step of grid-sampling to select k centers. The centers are then moved to the lowest gradient position in a limited pixel-neighborhood.

This step reduces the chance of selecting a noisy pixel as center. Next, each pixel is assigned to the nearest center to include the pixel in the center's search region. Limiting the search region makes the algorithm faster, compared to a classical k-means clustering. A limited search region is inspired by the distance value between pixels. The pixel distance can be calculated depending on the color space and a distance measure, such as Euclidean, with respect to pixel color and intensities. After all pixels are visited and associated with a center, the position and color of each center is corrected to be the mean value of all pixel positions and all pixel colors in the respective cluster. Then, the Euclidean distance between the old and new center is calculated. These assignment and update steps are repeated until the error value for all clusters converges, however, a few number of iterations are enough in practice. At the end, disjoint pixels are clustered with their nearest clusters to enforce connectivity of superpixels. Achanta et al. [ASS⁺10, ASS⁺12] suggest the LABxy distance for CIELAB color space. The color or position distances can get different weights to promote accuracy of superpixels in terms of pixel values or the compactness of segmentation, respectively.

Since it has been published [ASS⁺10, ASS⁺12], the SLIC superpixel has become a clear choice among superpixel segmentation algorithms, due to its simplicity, accuracy and efficiency. There is not much to improve about it for the purpose of this thesis, however, it is extended in Chapter 3 to depict superpixels from geometry cues. From the point of view of geometry processing, the proposed superpixel approach is a combination of the ideas from image processing i.e. SLIC superpixel algorithm, and from geometry processing, e.g. surface patches.

2.7.4 Texture Patches

An alternative compression method is the texture patches. Dessein et al. [DSWH15] builds them by sampling with the greedy farthest point strategy and geodesic re-meshing, similar to [PC06]. Related patching methods are applied on different objects for other applications, i.e. texture stitching and mesh segmentation [DSWH14, DSWH16]. For the face recognition application by Li et al. [LSH06], the patching algorithm is a contribution to spin image representation, first proposed in [JH98], where Li et al. involve the surface normals to define surface patches, with respect to the object [LSH06]. A proper use of patches leads to a uniform sampling in 3DMM's texture domain that, due to solved correspondence, indirectly promotes a 3D semantic-aware way to represent a facial image with a lower number of values than the pixels of the face in the original image. Texture patches are in correspondence with the face regions in 3D geometry; they provide a model-based approach of downsampling of a *mesh*. In contrast to texture patches that use surface normal cues to segment a mesh, the proposed superpixel in Chapter 3 is a geometry-based approach (also based on surface normals) for superpixel segmentation of a 2D *image* of an

object with a given geometry. In Section [3.4.2](#), this algorithm is proposed with minimal changes in the implementation of SLIC superpixel segmentation. Hence, texture patches are not used in this thesis.

Chapter 3

Inverse Lighting and Relighting

3.1 The Inverse Lighting Framework

As proposed in Chapter 1, the main goal of this thesis is to provide an analysis by synthesis approach for inverse lighting from a face image. The inverse lighting is performed in two essential separate steps: 3DMM fitting and lighting estimation. This is shown in the Activity Diagram Figure 1.5. The first step provides some of the requirements (see Section 2.3) for the subsequent lighting estimation. These requirements include a colorful high quality 3D face model for the given 2D face image, pose of the face, dense correspondence and color correction parameters for the given image. The second step, called Estimate Lighting in the diagram, is divided in sub-steps in Activity Diagram of Figure 3.4. It relies on the superposition principle for light and the illumination cone (see Section 2.4) to propose a linear generative model for the lighting. The parameters for the generative model are estimated by optimization in Chapter 4. With some novel algorithms, the preparation of the generating set is explained next. Moreover, the relighting and lighting design algorithms are proposed by the end of the chapter.

3.1.1 The Use Case Diagrams for the Proposed Inverse Lighting

A UML Use Case Diagram is usually used to analyze the interactions of the extern actors with the system and the behavior of the system. The actors (shown with stick figure) are responsible for tasks or actions (use cases). The use cases (shown with ellipses) are major tasks that the system needs to accomplish. These use cases might be done by performing their *included* use cases. They also might be *extended* with optional use cases [Agi]. A Use Case Diagram only shows actors, use cases and their relationships, nonetheless, the captions provide sufficient descriptions. We address three actors to analyze the requirements of the inverse lighting algorithm: a human “User” in Figure 3.1, the “Inverse Lighting Framework” in Figure 3.2 and the “3DMM Framework” in Figure 3.3. In Figure 3.1, we see the responsibilities of the user. The user has to provide the input image and

the facial landmarks. It may also provide the occlusion mask. In Figure 3.2, the 3DMM framework is seen as an actor. It estimates the rendering parameters and provides the algorithm with other necessary resources and procedures. In Figure 3.3, the set of the proposed solutions are viewed as the Inverse lighting Framework. The Inverse Lighting Framework is seen as an actor itself to address its responsibilities. Use Case Diagrams are usually kept abstract for the sake of readability. In Figure 3.3, a third level use case (Call 3DMM Fitting) is drawn because it is an essential task in the proposed inverse lighting.

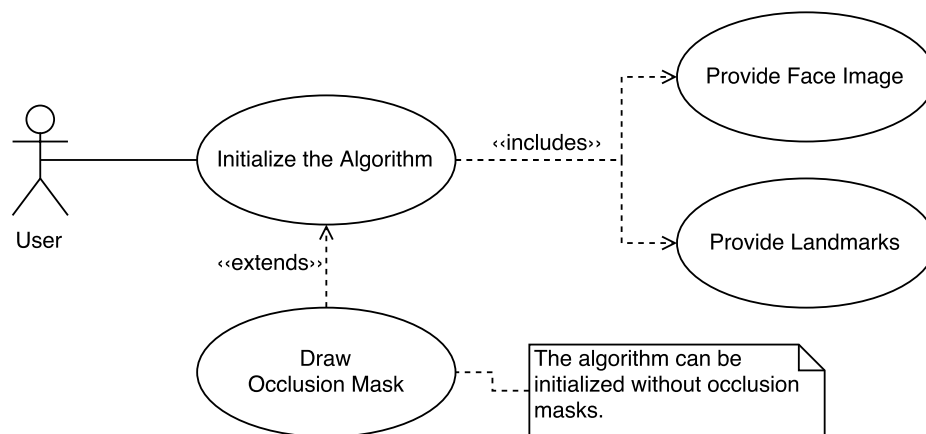


Figure 3.1: This is the use case diagram for the actor “User.” A human user initializes the algorithm by providing a 2D image of a face and clicking some landmarks on it. The user might draw an occlusion mask on the image to help the algorithm ignore occluded areas. The occlusion mask is discarded in the state of the art algorithm in Section 4.4.

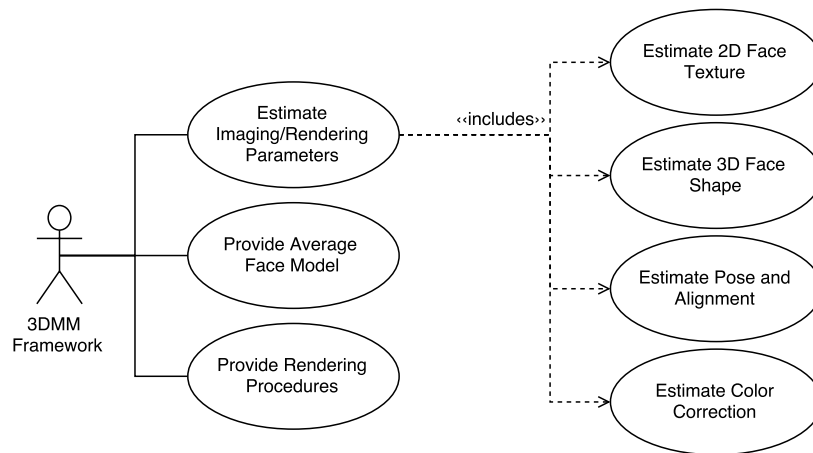


Figure 3.2: This is the Use Case Diagram for the actor “3DMM Framework”. From the point of view of the Estimate Lighting step, we look at the 3DMM Framework as an external agent. It performs inverse rendering by estimating the rendering parameters, and provides the average human face model (3D shape and texture) and resources, such as rendering procedures. Estimation of the rendering parameters that are important for the inverse lighting algorithm includes texture and shape estimation (although the estimated texture is replaced with the average texture in the next step), estimation of the dense correspondence, i.e. pose, projection and dense alignment, between the 3D model and the 2D input, and the estimation of the color correction parameters. See Section 2.3.

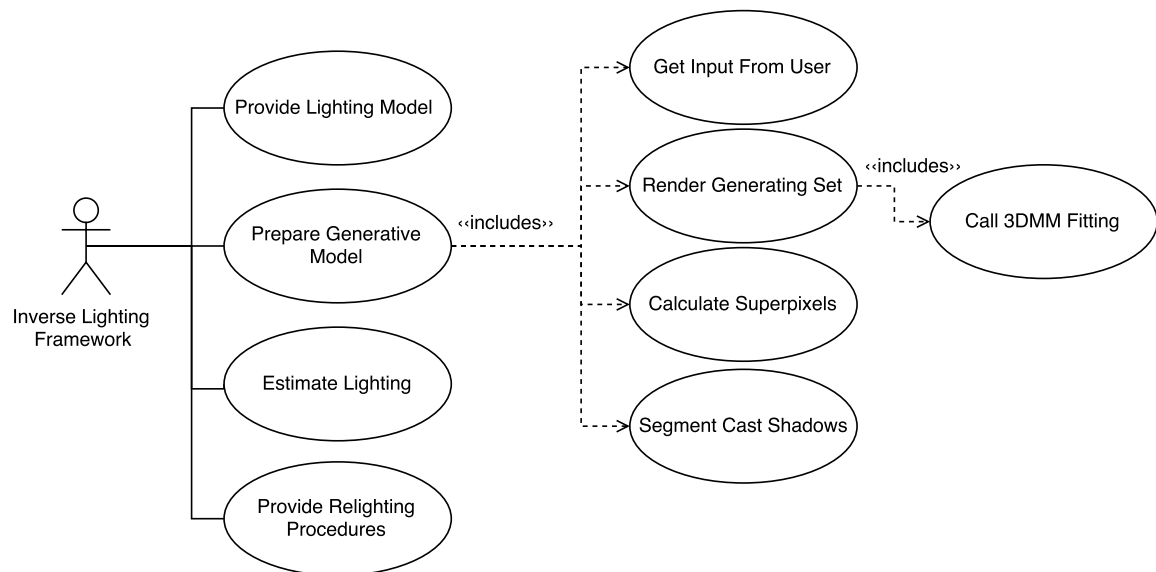


Figure 3.3: This is the use case diagram to show the duties of the major business actor in the proposed system. Especially for further applications, the Inverse Lighting Framework can be seen as an external actor. It provides the geometry of the lighting model (the light source directions of the VLS (Section 3.2.1)). It also prepares the generative model, estimates the lighting (the RGB values of light sources) by optimization and provides procedures for relighting (Section 3.6). To prepare the generative model, it needs to get the input image and the landmarks from the user, feed them to the fitting algorithm of the 3DMM to estimate the face model and other rendering parameters (Section 2.3), render n images of the estimated face under n light sources from the VLS (Section 3.2), calculate superpixel representations of the n images in the gallery and of the input image (Section 3.4 and 3.4.2), and mark the cast shadow segments (Section 3.3). The preparation of the generating set is explained completely in this chapter.

3.1.2 An Activity Diagram for the Inverse Lighting Algorithm

A UML Activity Diagram is usually used for business process modeling of the algorithm that resembles a use case or a usage. It is the object oriented equivalent of the flow charts and data flow diagrams. If the algorithm is so complicated that its Activity Diagram gets overwhelmingly huge, usually it would be better to write down the algorithm instead of the Activity Diagram. Nevertheless, these diagrams are useful to show the overview of the major steps. They have an initial (filled in circle) and usually a final node (filled in circle with a border). The activities are represented with rounded rectangles and the flow is shown with arrows. For parallel activities a fork is used, which is a black bar with one arrow going in and more than one arrows leaving it. Parallel flows can join in another black bar which has only one outgoing arrow [Agi]. However, Activity Diagrams can be very simple, such as the one in Figure 1.5. It is expanded in Figure 3.4 to show the flow of the Estimate Lighting activity which is proposed through this chapter, especially in Section 3.5. Later, we see a few more Activity Diagrams in this chapter. The algorithm for the preparation of the generating set is shown in Figure 3.6, the diagram for the intrinsic texture extraction algorithm in 3.11 and the proposed lighting design algorithm in 3.13.

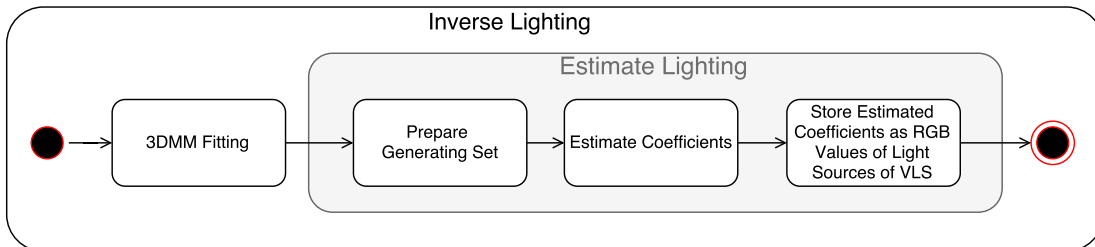


Figure 3.4: UML Activity Diagram of Inverse Lighting extended by three sub activities of “Estimate Lighting” Activity from Figure 1.5. It shows the algorithm flow for the estimation of lighting from one face image. The algorithm is initialized with the input image and landmarks by the user, the 3DMM fitting estimates the 3D face model and other rendering parameters (see Section 2.3). Then, the face model and other rendering parameters are used to render n images which are prepared as the generating set in Section 3.2. Finally, the coefficients of the linear combination of the generating set are estimated so that the result would be as close as possible to the input image with respect to a distance function between images in Chapter 4.

3.2 To Span a Synthetic Illumination Cone

In an illumination cone approach (Section 2.4) a generating set is needed whose linear combination spans the illumination cone of the given facial image. Theoretically, the linear combination with the correct coefficients gives the reconstruction of the input, which is a member of the illumination cone. This thesis proposes to synthesize the generating set and to infer the physically plausible lighting according to superposition principle of light. Accordingly, the inverse lighting problem is reduced to a coefficient estimation (Chapter 4) for a generative model which generates the input image as a linear combination of the images of the generating set.

To synthesize the generating set, rendering procedures and computer graphics models of the scene and the face are needed. These are provided by the 3DMM framework and discussed in Section 2.5 and 2.3. However, the 3DMM framework uses an ambient term and one directional light source to render a textured 3D face model with a Phong reflectance. The same lighting and reflectance models are assumed during the 3DMM fitting.

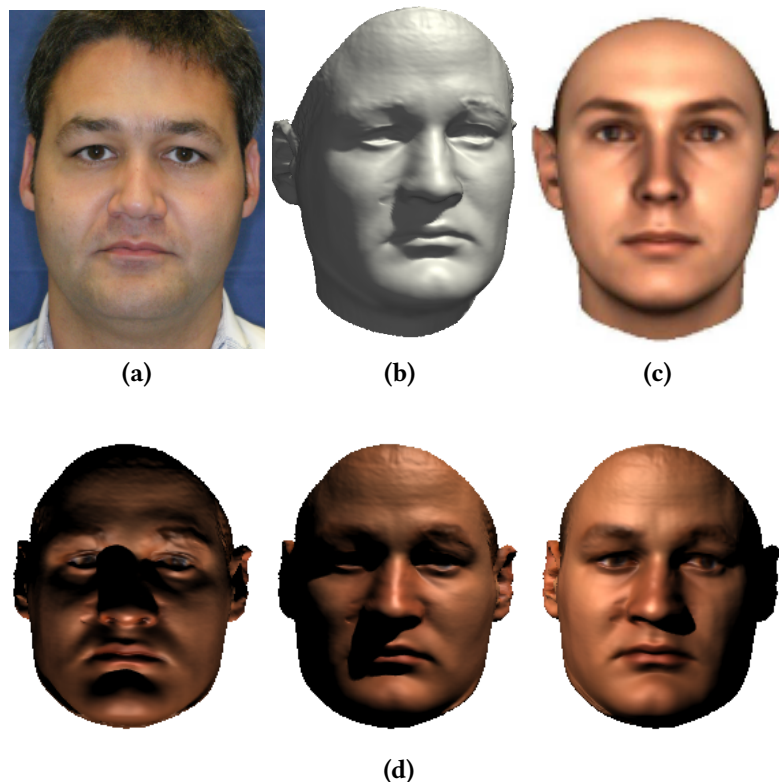


Figure 3.5: An input image is shown in Figure3.5a. The 3DMM fitting algorithm estimates the 3D shape Figure3.5b and texture (discarded). The 3DMM framework also provides an average human face model Figure3.5c, of which we only use the average texture for inverse lighting. The estimated 3D shape, pose and average texture are combined to render images under the VLS Figure3.5d. In this thesis, n images are rendered and usually $n = 100$ unless specified otherwise.

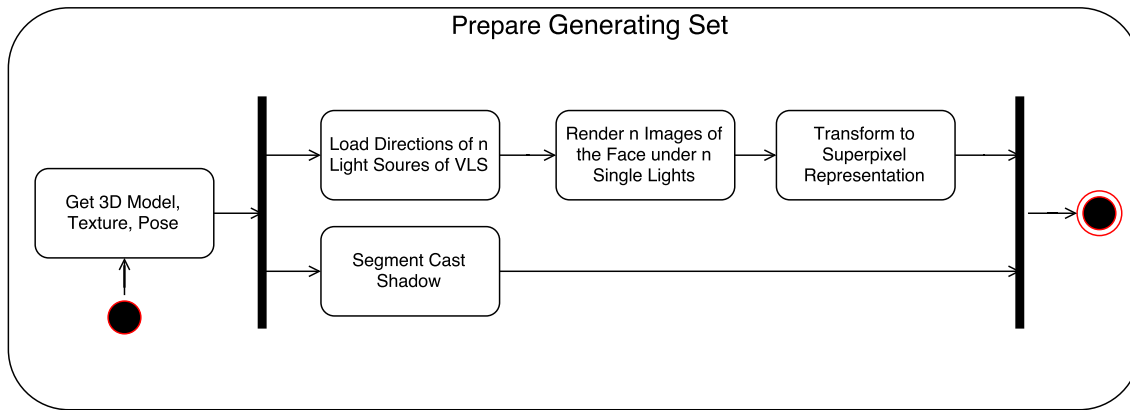


Figure 3.6: This is the UML Activity Diagram for generating set preparation task. It shows the algorithm flow and tasks that are carried out in the order that they are done for preparation of a generating set for a synthetic illumination cone from one face image. The algorithm is initialized with the estimated 3D face model, pose, alignment, average human face texture and color correction parameters. From the proposed lighting model (a VLS that is explained in Section 3.2.1), the light source directions are loaded in memory. Then, the face is rendered n times under all the same rendering conditions as the input image except n different lightings. Each time the face is rendered under a single light source i , from the VLS, with unit intensity and color values (see also Figure 3.5). After that, the input image and n rendered images are compressed in an illumination friendly superpixel representation (Section 3.4.2) and stored as the generating set in the output. Parallel to the above steps, the cast shadow areas are automatically marked, if available, in the original image (Section 3.3).

Consequently, the estimated 3D model is suboptimal for images with harsh illumination. Usually, the 3DMM's joint fitting algorithm propagates the nonreproducible illumination effects into the estimated texture, which is supposed to contain the intrinsic albedo of the face. Thus, both the estimated lighting and texture by 3DMM are assumed useless and discarded in the proposed inverse lighting. To render the generating set, instead of the estimated texture an average human texture (Figure 3.5c) is used with the estimated 3D shape of the face (Figure 3.5b). The average human face texture is provided by the 3DMM framework. The n rendered faces share all the same rendering parameters and differ only in the lighting, as seen in Figure 3.5d. They represent the appearance of the face under the predefined lighting conditions. In Figure 3.6, rendering the generating set is followed by image compression. The model-aware cast shadow segmentation is used for cast shadow promotion in Chapter 4).

3.2.1 Virtual Light Stage (VLS)

In Section 2.4.3, we saw that a *real* light stage might be equipped with multiple high quality video cameras, 3D scanners, optical filters and more. Aside from the costs of the necessary hardware, a lab requires the presence of the object, in this case the human face, for appearance acquisition. Conversely, a *Virtual Light Stage* (VLS) is introduced to make an analysis by synthesis approach for single-image scenarios possible. The VLS is a set of n

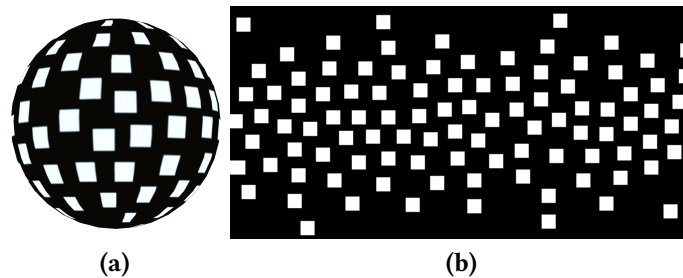


Figure 3.7: The sphere of 100 light source directions is shown in Figure 3.7a from the frontal view. A rectangular plot of it is shown in Figure 3.7b. Each white square represents a light source direction. Here, there are 100 light source directions distributed on the entire sphere with a random uniform distribution. The squares in the middle of the rectangle represent light sources from the frontal direction toward the face. Those close to the top and bottom are light sources from up and down, and those close to the middle of left and right edges are light sources from behind the head.

light sources, uniformly distributed in the whole space around the object and oriented in its direction. In Section 3.2.1, we spread the light source directions by uniformly distributing n points over an entire imaginary sphere around the face. Taking the latitude angle as y -axis and altitude as x -axis, this sphere is represented completely on a 2D rectangle in Figure 3.7. The directions of the light sources are fixed so that the inverse lighting from a given input image I is reduced to the estimation of only the intensity and colors for these directional light sources. Moreover, the VLS is a virtual lab, equipped with the 3DMM framework that provides the necessary rendering parameters and procedures (see Section 2.3). It is common sense to spread the light sources of the VLS uniformly on the sphere so that a fair diversity of lighting situations are representable.

Uniform Distribution on a Sphere

There are more than one suitable solution for uniform distribution of points on a sphere. Choosing a method over another one depends on the requirements. Taking the VLS as a discrete environment map, to increase its resolution means to increase the number of the light sources –dimension of the VLS. To facilitate experiments with different number of light sources we design a function that gets an arbitrary positive number $n \in \mathbb{N}$ and distributes n directions uniformly in the 3D space. A uniform distribution of directional light sources can be explained as the distribution of points on a sphere with infinite radius. Since only the directions are interesting for our purpose, it is equal to the distribution of n points on a unitary sphere –a sphere with radius equal to one. The 3D position vectors of these points are the wanted direction vectors of the light sources. To spread the points, a random number generator is used to produce (x, y, z) triplets, where $x, y, z \in [-1, 1]$. This spreads n points uniformly in a 2×2 cube. At this stage, we could have discarded the points in the corners of the cube, $\|(x, y, z)\| > 1$, to avoid an anisotropic distribution.

Although, the current implementation does not include the discarding step, this has no effect on the results due to the final step that optimizes the uniform distribution on the surface of the sphere. Then, the (x, y, z) vectors are normalized. The normalization brings the points on the surface of the sphere. Finally, the minimum distance between each pair of points is maximized with a relaxation-based optimization algorithm to achieve a close to uniform distribution.

This algorithm satisfies the requirement for a random and uniform distribution. Moreover, it only needs to be performed once for each desired n . In course of this project, we experiment with $n = 50$, $n = 100$, $n = 300$, $n = 1000$ and $n = 10000$.

Alternative Methods: One can use a regular polyhedron. The positions of the vertices for an icosahedron ($F = 20, E = 30, V = 12$) are given in 3.1.

$$\begin{pmatrix} 0 & , & \pm 1 & , & \pm\phi \\ \pm 1 & , & \pm\phi & , & 0 \\ \pm\phi & , & 0 & , & \pm 1 \end{pmatrix} \quad ; \quad \text{where } \phi = \left(1 + \frac{\sqrt{5}}{2}\right) \quad (3.1)$$

Starting from these 12 points, one can include the middles of the icosahedron faces to the set, normalize them, and produce 42 uniformly distributed points on a sphere. Similarly, a dodecahedron ($F = 12, E = 30, V = 20$) can be constructed. In this case, we have to include the middle of the faces of the dodecahedron. The number of the uniformly distributed points is limited to an integer factor of the number of vertices (V) added by integer factors of the number of edges (E) or faces (F), depending on the expansion strategy. For example, a uniform distribution of 17 (or 100) light sources is not achievable with these platonic solids. Another straightforward way of distributing the light sources is to start from one position on the sphere and make equal angular steps in azimuth and zenith directions to add new points and stop when there is not enough space to put another point on the sphere with the desired distance. This usually leads to spiral-like forms on the sphere with increasing obliqueness depending on the number of steps from the initial point. Such algorithm gets a distance as input, instead of an arbitrary n .

The VLS framework for inverse lighting and relighting involves more than only a set of virtual light sources. To estimate a 3D face model, the 3DMM fitting algorithm from Section 2.3 is employed, which also provides a dense correspondence between the estimated 3D model and the input image. The rendering procedures are a combination of those from 3DMM framework and a more realistic reflectance function compared to the original Phong model (Section 2.5).

3.2.2 Adopting a Measurement-Based Reflectance

The measurements of human face reflectance by Weyrich et al. at ETH/MERL [WMP⁺06] are publicly available. The reflectance parameters are provided as average values per region for a total of nine regions on the face. Compared to that, the average texture of the 3DMM framework has a much higher resolution. Also, the average texture of the 3DMM is used in the fitting algorithm of the 3DMM, which also estimates the color correction parameters, see Chapter 2 and Section 3.2.3. Using a modulation texture with a different brightness or color saturation leads to unwanted inconsistencies with the rest of the framework. Therefore, we propose to use the 3DMM’s texture, no matter what reflectance function is used to calculate the diffuse and specular terms. Consequently, the BRDF parameters from the database and this texture need to be adapted to each other so that the proportion of the diffuse and specular terms stays faithful to the measurements of [WMP⁺06], when they are used in the rendering equation (2.9).

First, a generic BRDF parameter set for the specular and the shininess coefficients, ρ_s and m (in the Beckmann’s distribution of Torrance-Sparrow), are created. We do so by calculating the average of these parameters, in the ETH/MERL database, for all the stored entries. The averages are calculated separately for each of the 9 regions of the face, according to [WMP⁺06]. These regional averages are stored in a texture structure to keep it in correspondence with the texture coordinates. They are manually blurred at the borders of the regions to avoid visibility of the region borders in rendered faces.

The modulation texture of 3DMM is generally brighter than what the measurements expect. To preserve the proportion of the specular to diffuse according to the measurements of [WMP⁺06], the specular term of [WMP⁺06], f_{TS} , is multiplied by the ratio between t_{mean} and α_{mean} , which are the average intensity of the 3DMM’s texture, given in (3.3), and the average intensity of the measured albedos of ETH/MERL, given in (3.4), respectively. As a result, the used specular term in this thesis, f_s , is corrected as below:

$$f_s = \frac{t_{mean}}{\alpha_{mean}} f_{TS}. \quad (3.2)$$

Thereby, t_{mean} is given as:

$$t_{mean} = 0.3t^{red} + 0.6t^{green} + 0.1t^{blue} \quad (3.3)$$

where t^{red} , t^{green} and t^{blue} are the averages of the 3DMM’s average texture in the red, green and blue channels, respectively. Also, α_{mean} is calculated from:

$$\alpha_{mean} = 0.3\alpha^{red} + 0.6\alpha^{green} + 0.1\alpha^{blue} \quad (3.4)$$

where α^{red} , α^{green} and α^{blue} are averages of the albedos in the respective color channels,

extracted from the database [WMP⁺06].

This ratio is calculated once and used consistently through out the thesis. With this ratio, the dipole diffuse reflectance from [JB02] and the Torrance-Sparrow specular reflectance function (see Section 2.5.3) are implemented in the 3DMM's rendering procedures.

Despite previous work on texture hallucination [DSWH15, SB15a] and on the synthesis of the micro-geometries (mesoscale facial features) in [GTB⁺13], which lead to more realistic looks of the rendered faces, a detailed model of the facial texture and geometry can not be extracted from a single input image yet. Whenever a more accurate model of the face appearance is available, for instance real light stage data, then more detailed facial reflectance functions, such as [JZJ⁺15, JSG09], might be beneficial. Until then, generic BRDF functions cannot deliver accurate results, no matter how many physical features of skin are probabilistically considered in the function. Whenever desired, future changes of the BRDF function require no further modifications to the proposed inverse lighting as long as the superposition principle of light is preserved.

3.2.3 Color Correction

In Section 2.5.5, color correction is modeled with seven parameters: color contrast, gains and an offset for each color channel. These seven color correction values are provided by the 3DMM fitting. Directly from the offsets for color channels, we make the RGB value \mathbf{o} , and T is calculated from the color contrast scalar ξ and three channel specific gains (g_r, g_g, g_b) :

$$T = \begin{pmatrix} (0.7\xi + 0.3)g_r & (0.6 - 0.6\xi)g_r & (0.1 - 0.1\xi)g_r \\ (0.3 - 0.3\xi)g_g & (0.4\xi + 0.6)g_g & (0.1 - 0.1\xi)g_g \\ (0.3 - 0.3\xi)g_b & (0.6 - 0.6\xi)g_b & (0.9\xi + 0.1)g_b \end{pmatrix} \quad (3.5)$$

The mapping for each pixel is given in (3.6) as a transformation from color contrast, saturation and brightness of 3DMM's rendered *neutral* image I' to those of the captured input image I .

$$I(p) = \mathbf{o} + T I'(p) \quad (3.6)$$

The color correction must be applied after the superposition, therefore, it is used in the rendering of the generation set under the VLS. Accordingly, $\vec{\sigma}$ is set to zero and T to identity matrix for the rendering of the generative set images (see Figure 3.5). In Chapter 4, we see that the color correction (3.6) with estimated values from 3DMM fitting appears in the cost function for the optimization process that estimates the lighting.

3.2.4 3DMM Masks

The 3DMM framework provides infrastructure to draw masks on the face model in a 2D user interface. These masks mark an area on the 3D face model. This area might be ignored in the algorithm (blind mask) or extra considered as a Region Of Interest (ROI). After 3DMM fitting, the mask can be applied to the corresponding area in the input image. This is possible because the 3DMM fitting algorithm returns a dense correspondence between the 3D model and the face in the 2D image. In this thesis, a blind mask for the neck is always used to put the focus on the inverse lighting of the face. In the earlier experiments, also a blind mask on the eyes and ears are used because the estimated model is often unreliable in those areas, considering the sensitivity of the subsequent parameter estimation (Section 4.3). The eye and ear masks are not used in the JMAP approach (Section 4.4) because the hyperparameter optimization automatically handles the inconsistencies between the model and the input image. Furthermore, a few masks as ROIs are prepared to incorporate the regional BRDF measurements from ETH/MERL in the 3DMM's renderer (Section 3.2.2). And a ROI mask is designed for the surround area of the nose for the detection and segmentation of cast shadows in Section 4.5.4.

3.3 Cast Shadow Detection and Segmentation

Cast shadows usually cover a relatively small number of face pixels on the 2D image and vary nonlinearly with respect to changes in the lighting direction. They depend highly on the global accuracy of the geometries that cast the shadow and surfaces that the shadow is casted on. As a result, they hardly contribute to the linear cost function. The previous work [BV99] can model the cast shadow of one directional light, however, it performs generally better when cast shadows are ignored during the fitting. After the light direction is estimated the forward rendering procedures can draw a cast shadow. Other previous work that use nonphysical lighting models, such as spherical harmonics, ignore cast shadows completely. Among all the single-image approaches in Chapter 2, only Wang et al. [WLH⁺07] focus on face images in harsh illumination, however, they remove the cast shadows without modeling them. Modeling cast shadows needs accurate 3D information and a cost function that, beside the intensity of the cast shadow, considers the form and location of the edges of a cast shadow. The goal is to model the form and the intensity of the cast shadow by estimating the light sources. Finding such an edge-based cost function for cast shadows is left to the future work. The proposed algorithm is limited in what can be achieved with a pixel-based cost function, however, it is the first attempt to address this problem by estimating a physically plausible lighting model.

In a pixel-based cost function, cast shadows need special treatment. In Chapter 4, a probabilistic approach (JMAP) introduces hyperparameters that influence the importance of

different pixels or regions of the face on the estimation of its lighting. We observed that the cast shadows are replicated more accurately when the hyperparameters corresponding to the cast shadow regions are manually set to a large value. To automate this step, a cast shadow segmentation method is introduced in this section that segments the cast shadow area without any user interaction.

For the proposed inverse lighting, estimating the cast shadow of the nose is more significant and most of the times sufficient. Unlike the cast shadows around the eyes and lips, the nose shadow is in most cases a reliable indication of the main light source direction. Thus, we mark the area around the nose in the 3D face model by designing a generic model-based mask. This mask is applied to the estimated 3D face model and mapped into the image plane on the corresponding area Figure 3.8b. The occurrence of cast shadows is only investigated in this region (see Section 3.2.4). In Section 2.6, we see that the previous efforts usually use thresholding to detect shadowed pixels. Considering the differences in brightness of different images and the varying face albedo, selecting a scalar threshold for all images and faces in every pose is an underestimation of the problem. Instead, we need to consider the general brightness of the lighting and of the image, and the facial albedo to predict if a pixel (or a region) of the input image is under cast shadow. To consider these factors, the face model is rendered under an optimal diffuse ambient illumination according to (3.7). We calculate the appropriate threshold for each pixel of the face from the image I_{amb} (see Figure 3.8d).

$$I_{amb} = \mathbf{o} + T(\mathbf{a} I_{alb}) \quad (3.7)$$

where \mathbf{a} denotes the optimal ambient coefficients in RGB channels that are calculated according to (3.9), \mathbf{o} and T are the color correction parameters from 3DMM fitting (see Section 2.5.1), and I_{alb} is the average texture (albedo) rendered in correspondence with the face in the input image.

To render the I_{alb} , the estimated 3D face model is rendered with only an ambient term ($a_R = a_G = a_B = 1$) and the average facial texture. The RGB coefficient \mathbf{a} is calculated according to (3.9) supposed to (3.8).

$$\underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{o} + T(\mathbf{a} I_{alb}) - \mathbf{I}\|_2^2 \quad (3.8)$$

$$\mathbf{a} = \frac{\langle I_{alb}, \mathbf{I}' \rangle}{\|I_{alb}\|_2^2} \quad (3.9)$$

where the numerator is the scalar product of two images (summation over their pixel-wise multiplication). The image \mathbf{I}' is a neutralized version of the input image \mathbf{I} that is calculated by inverting the color correction according to (3.10).

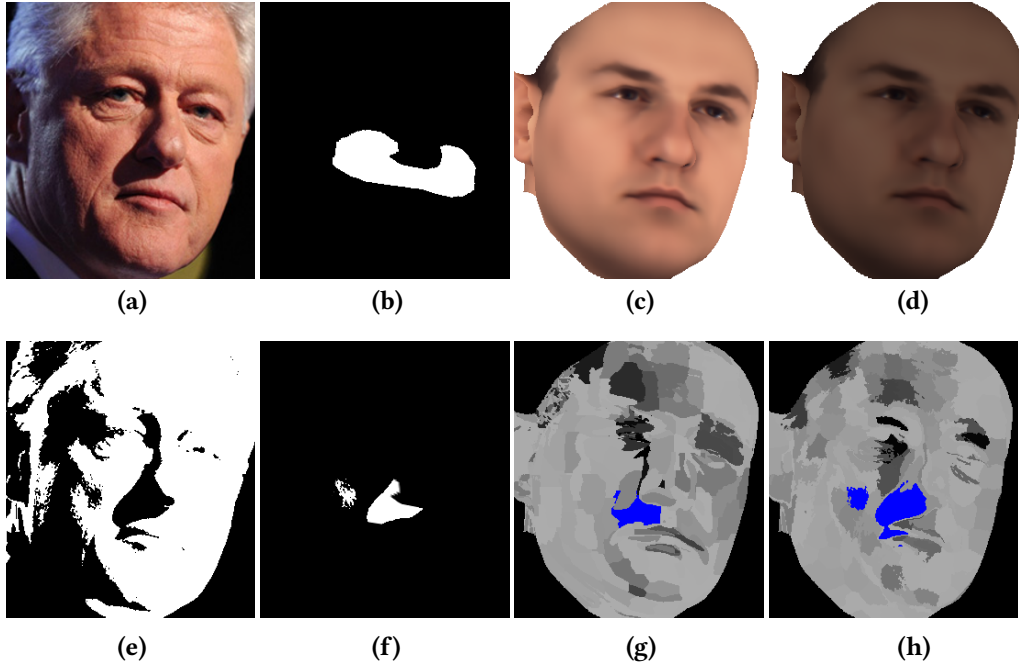


Figure 3.8: An input image with a hard cast shadow below the nose toward the right cheek of the subject is shown in Figure 3.8a. Figure 3.8b is an automatically marked area using a model-based mask for the area around the nose and the estimated 3D model for the input image. Figure 3.8c is the albedo image I_{alb} and Figure 3.8d is $(a \cdot I_{alb})$ according to (3.7), which represents the optimal ambient illumination reconstruction for the input image. Figure 3.8d is calculated with the equation: $1 - (Figure 3.8d - Figure 3.8a)$. Note that the darker areas of the input image will lead to negative values, visualized as black pixels, in this image. Finally, mark the negative pixels in Figure 3.8d only if they are also marked in Figure 3.8b to generate the cast shadow mask Figure 3.8f. Here, the small spot on the left is marked as cast shadow of the nose, which is wrong. However, such errors are negligible for the purpose of this paper. The partially masked $\vec{\beta}$ is mapped on the image plane, for geometric superpixels Figure 3.8g and photometric superpixels Figure 3.8h. The blue areas are the superpixels for which the β_p is fixed to a large value.

$$\mathbf{I} = \mathbf{o} + T\mathbf{I}' \quad \Rightarrow \quad \mathbf{I}' = T^{-1}(\mathbf{I} - \mathbf{o}) \quad (3.10)$$

Then we calculate the shadow map \mathbf{I}^{s_map} (see Figure 3.8e) according to (3.11). This shadow map shows everything in white except for the darker cast and attached shadow areas which are black or gray. In this image, we threshold the darker areas inside the region of interest –around the nose– as cast shadow segments, see Figure 3.8f. Thus, after 3DMM fitting and before lighting estimation, we can mark the cast shadow areas on the x, y image coordinates and their corresponding superpixels.

$$\mathbf{I}^{s_map} = 1 - (\mathbf{I}_{amb} - \mathbf{I}) \quad (3.11)$$

This algorithm has one limitation: if there is a darker area on the skin surrounding the nose, e.g. birth mark, it will be marked as cast shadow incorrectly. Also, if an attached

shadow appears inside the region of interest (around the nose), it will be marked as cast shadow. This has happened in Figure 3.8. In the experiments, we did not encounter any case in which this issue leads to a failure of the inverse lighting. In Figure 5.7, we see that the attached shadow is rarely mistaken for cast shadow of the nose. Compared to this approach, even a precisely drawn cast shadow mask can only negligibly improve the general lighting estimation. For our purpose, the coarse cast shadow mask from the above algorithm is sufficient (see Figure 3.8f and the blue segments on the estimated β in Figure 5.7).

Optional Improvement: The cast shadows are already estimated in the first run of the [SB15b] algorithm, but look brighter on the rendered image compared to the cast shadows in the input image. Hence, we have a much better cue about the occurrence of cast shadows after performing inverse lighting on a face image. With this cue, a more accurate cast shadow segmentation can be achieved. A cast shadow map is calculated after one pass of inverse lighting by subtraction of two images; one rendered with cast shadows (set the cast shadow flag in the renderer) and the other one without the cast shadows (reset the cast shadow flag in the render). The difference between these two images marks the cast shadow areas. The improved cast shadow mask is calculated by an intersection between the shadow map Figure 3.8e and this cast shadow map. This intersection resembles the areas that are relatively darker on the input image in Figure 3.8a (with respect to the optimal ambient image in Figure 3.8d) and are rendered under cast shadow with the estimated VLS lighting from the first pass. Using this approach, the white spot on Figure 3.8f, which marks an attached shadow as cast shadow, is removed and the improved cast shadow border is more accurate with respect to the input image. According to experiments, this approach delivers improved cast shadow segmentation at the cost of an extra “Estimate Lighting” pass. Furthermore, we observed that this improved approach has the potential to deliver a very exact segmentation of cast shadows on a harshly illuminated face image. In doing so, the Estimate Lighting and Calculate Cast Shadow Map step mutually correct each other after each iteration and converge to a very accurate cast shadow segmentation. This might inspire future work on cast shadow segmentation, however, for the purpose of this thesis after image compression (Section 3.4) the increased accuracy is obsolete, and therefore not used in Chapter 5. An accurate cast shadow segmentation is, accordingly, out of the scope of this thesis.

3.4 Image Compression

The input image and the generating set \mathbf{C} (see Figure 3.10) are together $n + 1$ images. These images are stored in the memory and the per pixel operations are applied on all the face pixels. The number of pixels m can be very large and the given illumination

information can be redundant, because Illumination effects appear mostly in the lower spatial frequency domain. Consequently, each extra pixel adds an unnecessary equation to the equation system in Chapter 4. A direct solution for this redundancy in data is to downscale the images (see Section 2.7.1). For the proposed inverse lighting, three different approaches are experimented with. A Compressive Sensing approach is proposed in [HCSBL15] (see Section 2.7.2) and the other two methods are explained here. When we propose the cost function in Section 4.2, a formal proof is provided that these compressions preserve the linearity of the system.

3.4.1 Masked Gaussian Downscaling

The Gaussian blur is a useful antialiasing for which is used for downsampling. Using this linear filter, each pixel in the smaller image represents a number of pixels from the larger image, however, the blurring mixes the face pixels with the background pixels in areas that are close to the contours of the face. Therefore, the implementation needs to consider only the face pixels. After the 3DMM fitting, the face model is estimated and aligned with the face pixels on the image. From this alignment, the face pixels can be marked with a mask. With the help of this mask, the implemented smoothing filter ignores the background pixels. The downsampling window is moved consistently over all images and masks of the generative system; hence, the correspondence is preserved between them, and the formulation of the cost function remains intact [SB15b].

3.4.2 Geometric Superpixels

The above downscaling methods consider the low frequency nature of illumination effects in the 2D image plane. However, a glance at the reflectance functions in Section 2.5 is enough to remind us that the reflected radiance strongly depends on the surface normals. The estimated lighting for one pixel on an area of pixels with a common surface normal is sufficient to get the lighting for that area. This inspires an image segmentation method that considers the surface normals, instead of pixel values. In this thesis, the SLIC superpixel segmentation [ASS⁺12] is re-purposed to segment the face pixels in the image based on their corresponding surface normals on the face geometry. The terms “geometric superpixels” and “geometric superpixel segmentation” are used for this novel superpixels and segmentation approach. For the sake of clarity, let us call the classic superpixels “photometric superpixels.”

The SLIC superpixel segmentation algorithm by Achanta et al., which is reviewed in Section 2.7.3, divides the face pixels of the input image in a few clusters and assigns the average of the spectral values to the *centers* of the respective clusters [ASS⁺12]. Hence, it has two separate steps: superpixel segmentation and center value assignment. The segmentation step clusters neighboring pixels based on their local distance on the image plane and their

spectral similarity. The center value assignment step calculates a triple (an scalar per color channel) that represents the spectral values of each segment. To extract the geometric superpixels, we propose surface normals instead of the spectral values in the segmentation step:

1. After 3DMM fitting on the input image I , generate a surface normal map I_N –an image with surface normal XYZ values instead of RGB values for each pixel.
2. Perform an adapted version of SLIC segmentation (explained below) on the surface normal map I_N (see Figure 3.9a).
3. For each superpixel segment from the previous step, calculate the average of the pixel values, separately in R, G and B channels from I .

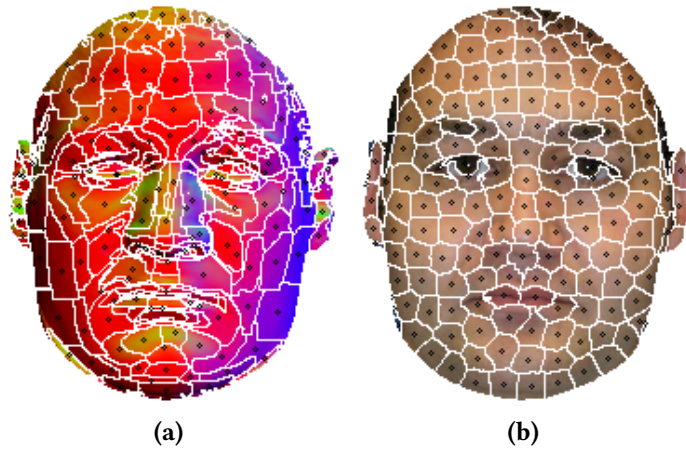


Figure 3.9: On the left, you see the result of the proposed geometric superpixel segmentation Figure 3.9a where the surface normal directions are color coded, and on the right, the result of the standard photometric superpixel segmentation is shown Figure 3.9b. Both are appended together and calculated for the whole gallery.

The corresponding normals for each face pixel is mapped from the 3D model that is estimated by 3DMM fitting. Note that the I_N is in dense correspondence with the input image. The adaptation of the SLIC is limited to changing its distance function and the numerical gradient descent so that they to work based on the angular distance between the normal vectors –their scalar product– instead of spectral distance functions or gradients of the classic SLIC. Thus, the algorithm delivers a list of clusters of the neighboring pixels with similar surface normals on the corresponding 3D shape. Compare the segmentation of the (classic) photometric superpixels Figure 3.9b with geometric superpixels Figure 3.9a. Only the segmentation is shown and not the center values. In this novel illumination friendly superpixel segmentation, the number of the segments is independent from the resolution of the input image. In addition, the segments are identical for the input image I and all the n images in the gallery \mathbf{C} to preserve the correspondence.

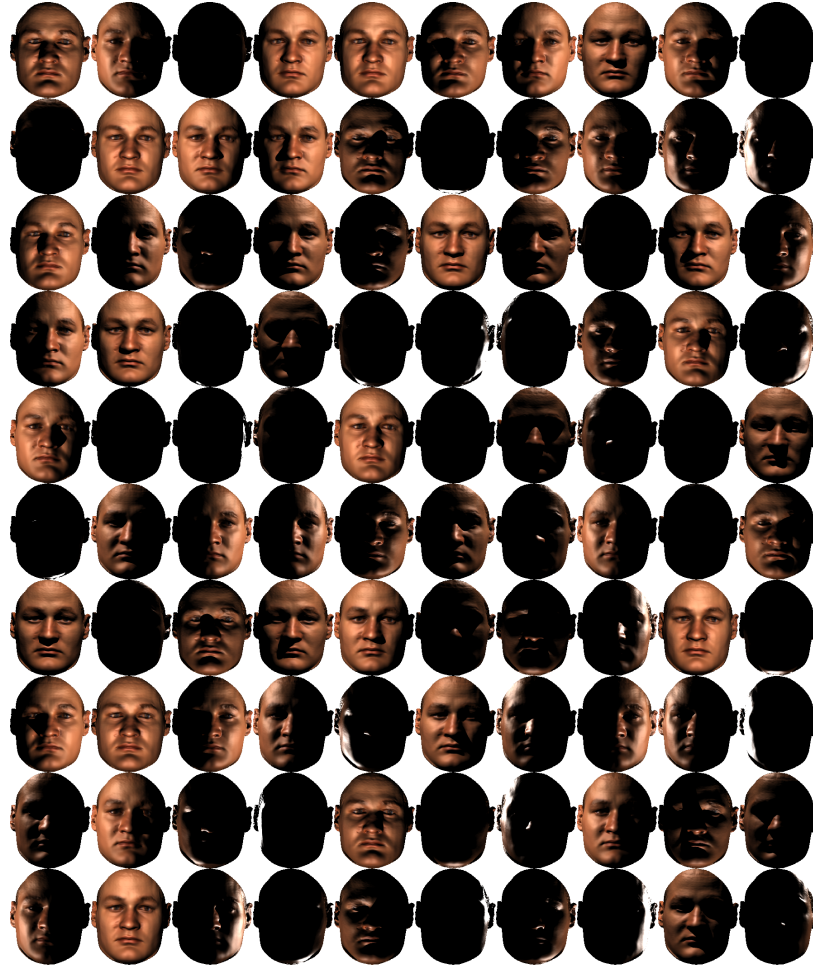


Figure 3.10: These are the 100 light stage images for the same input image that is used for Figure 3.9. The gallery of light stage images, no matter which input image is used, is shown with \mathcal{C} through this thesis. This image representation is clearly very large compared to a superpixel representation, shown with \mathcal{A} , that only stores ≈ 300 pixels for each image. The necessary storage for the \mathcal{A} that is calculated for this example stores $303 \times 100 \times 3 = 90900$ real values in total, which is much smaller than the necessary storage for one of the 2 mega pixel images in \mathcal{C} . Note that the light stage images are rendered in correspondence with and as large as the input image.

The geometric superpixels are enough to deliver a promising result for the proposed inverse lighting. However, using only the geometric superpixels might oversimplify the mathematics in presence of illumination and texture effects which do not depend on the local distribution of the surface normals, such as cast shadows and texture artifacts,. Therefore, we append the photometric superpixels (see Figure 3.9b) to the geometric superpixels according to (3.12) to increase the stability of the algorithm against such cases. In Section 4.2, we show how the JMAP algorithm automatically handles the importance of each superpixel with jointly optimized hyperparameters. Based on our experiments, using only geometric superpixels does not lead to the same robustness as when both photometric and geometric superpixels are used. The photometric clustering is done on the input image according to an exact implementation of [ASS⁺12], explained in Section 2.7.3. The

background pixels are masked out during the segmentation and center assignment. The center values of superpixels for the image I and for each C_i are calculated from the pixel values of the respective image and appended to the geometric superpixels representation of the respective image (3.12).

$$\begin{aligned} Y &= (\text{suppix}(G, I) \mid \text{suppix}(Ph, I)) \\ \forall i \in [1..n] : A_i &= (\text{suppix}(G, C_i) \mid \text{suppix}(Ph, C_i)) \end{aligned} \quad (3.12)$$

where G and Ph are two separate lists of segments that are clustered with the geometric and photometric SLIC approach, respectively. The function $\text{suppix}(\cdot, \cdot)$ delivers a vector of centers for the segments from the segment list (either G or Ph), where the centers are calculated from the spectral values of the image that is passed to this function. The sign \mid denotes a simple attachment between two vectors, which leads to a vector of the combined length. The output vector can be seen as an image with a single row of m_S pixels. Beside the input image and the gallery, all the image masks that are used in Section 4.2 are also transferred to superpixel masks with identical clustering as the input image and the gallery, preserving the correspondence. This approach leads to a major reduction in the data and the number of the equations of the mathematical system (Section 4.2) and at the same time works as a more illumination-friendly low pass filter that reduces noise. The rate of reduction is not constant and depends on the size of the input image. However, no matter how large the input image is, the face is represented with ≈ 300 superpixels in 3 color channels. These 900 real values per image take only a fraction of the memory that is needed for the storage of a given image of a face.

Alternatively, the intersection of the geometric and photometric segments can be used before the center value assignment. According to our experiments, the hyperparameters help a simple collective use of both lists of superpixels to address the complexities properly. The presentation of the segmentation is more meaningful when they are not intersected. Moreover, the averaging for the calculation of the center values is similar to using a box blur filter over a nonregular and dynamically changing window, hence, it is as compatible with the linear system as any other linear image filter (see Section 4.2).

3.5 Inverse Lighting Algorithm with Superpixels

The algorithm is initialized with an image of a face and the position of some facial landmarks, the same as the 3DMM fitting (see Use Case Diagram in Figure 3.1). The following lists the steps of the proposed algorithm for inverse lighting with superpixels. Also, see the extended Activity Diagram in Figure 3.4). The step 2 and 3 are shown in Figure 3.6. The direct result of this algorithm is the RGB values for fixedly positioned light sources

of the VLS. They are used to reconstruct the input image, or for applications such as relighting, lighting transfer and lighting design. Next, these applications are explained. Then, some miscellaneous observations and experiments are reported.

1. The 3DMM fitting algorithm estimates the rendering parameters, including shape, texture, pose, perspective projection and correspondence according to [BV99].
2. The generating set \mathbf{C} is rendered for the estimated face model, its pose and alignment.
3. An image compression (Section 3.4.2) is applied on the images \mathbf{I} and $C_i \in \mathbf{C}$, so that $\mathbf{I} \rightarrow \mathbf{Y}$, $C_i \rightarrow A_i$, accordingly $\mathbf{C} \rightarrow \mathcal{A}$. Then, only \mathbf{Y} and \mathcal{A} are stored.
4. The RGB of the light sources of the VLS are estimated by minimizing the difference between \mathbf{Y} and a generator function built upon \mathcal{A} (see Chapter 4).

3.6 Relighting

To relight a face, given in an uncalibrated 2D image, first we need an intrinsic face model, which is neutral with respect to effects of lighting. This intrinsic face model is then re-illuminated with a novel lighting. Because different features of the face might be hidden under cast shadows or saturated highlights, single image relighting (and realistic lighting design [SPB16]) is impaired by the originally harsh illumination of the input image. In this thesis, we employ the proposed inverse lighting to estimate the harsh illumination before applying the intrinsic texture decomposition in Section 3.6.1. To show results for illumination transfer, we use the estimated light sources from one input image on the intrinsic face model of another image, both achieved with the proposed inverse lighting. Some examples are presented in Figure 5.9. The pixel-based cost function inspires another relighting application where the target lighting is estimated from an image with user inputs that indicate lighting effects. We also explore this idea and propose a novel paint-based lighting design for single face images. Before that the intrinsic texture decomposition is explained.

3.6.1 Intrinsic Texture Decomposition

With the estimated illumination for the given face, the face pixels of the input image are de-illuminated and used as a more realistic and high quality intrinsic texture (albedo) $M_{alb}(p)$ for the estimated face model. See the overview of the algorithm in the UML Activity Diagram in Figure 3.11. We explain the algorithm as implemented in the 3DMM framework [BV99], however, with the proposed inverse lighting. First, in each pixel number p

of the input image I , the effect of color correction is inverted according to (3.13) to produce image I' . Then, to calculate the value of the pixel p of the de-illuminated image I'' , in each color channel of $I'(p)$, the effects of the estimated illumination are removed according to (3.14).

$$I'(p) = T^{-1}(I(p) - \mathbf{o}). \quad (3.13)$$

$$I''(p) = \frac{I'(p) - S(p)}{D(p)}, \quad (3.14)$$

where $D(p)$ and $S(p)$ are calculated with the estimated intensities and colors of the 100 VLS virtual light sources and the rendering function from Section 2.5.1. Close to the grazing angles, the image contains almost no information about the texture, therefore, the estimated values from 3DMM fitting are used. The transition between estimated 3DMM texture and the one calculated with (3.14) is interpolated depending on the angle between the surface normals of the face geometry and the camera direction. The whole face texture M_{alb} for the visible and invisible parts of the face is calculated as a result. In Figure 3.12, you see examples with variously complex illumination and the estimated textures before (mid row) and after (bottom row) the proposed inverse lighting. More examples are compared in Figure 5.7.

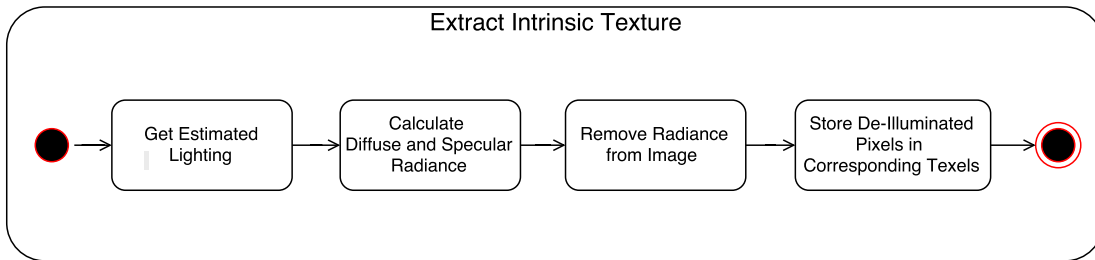


Figure 3.11: This is the UML Activity Diagram for the algorithm that extracts intrinsic texture from the input image, through de-illumination. This algorithm gets the estimated lighting, calculates the diffuse and reflectance on the already estimated 3D face model, removes the illumination from the corresponding face pixels in the input image and stores these de-illuminated pixels as texels in the respective position in the face model.

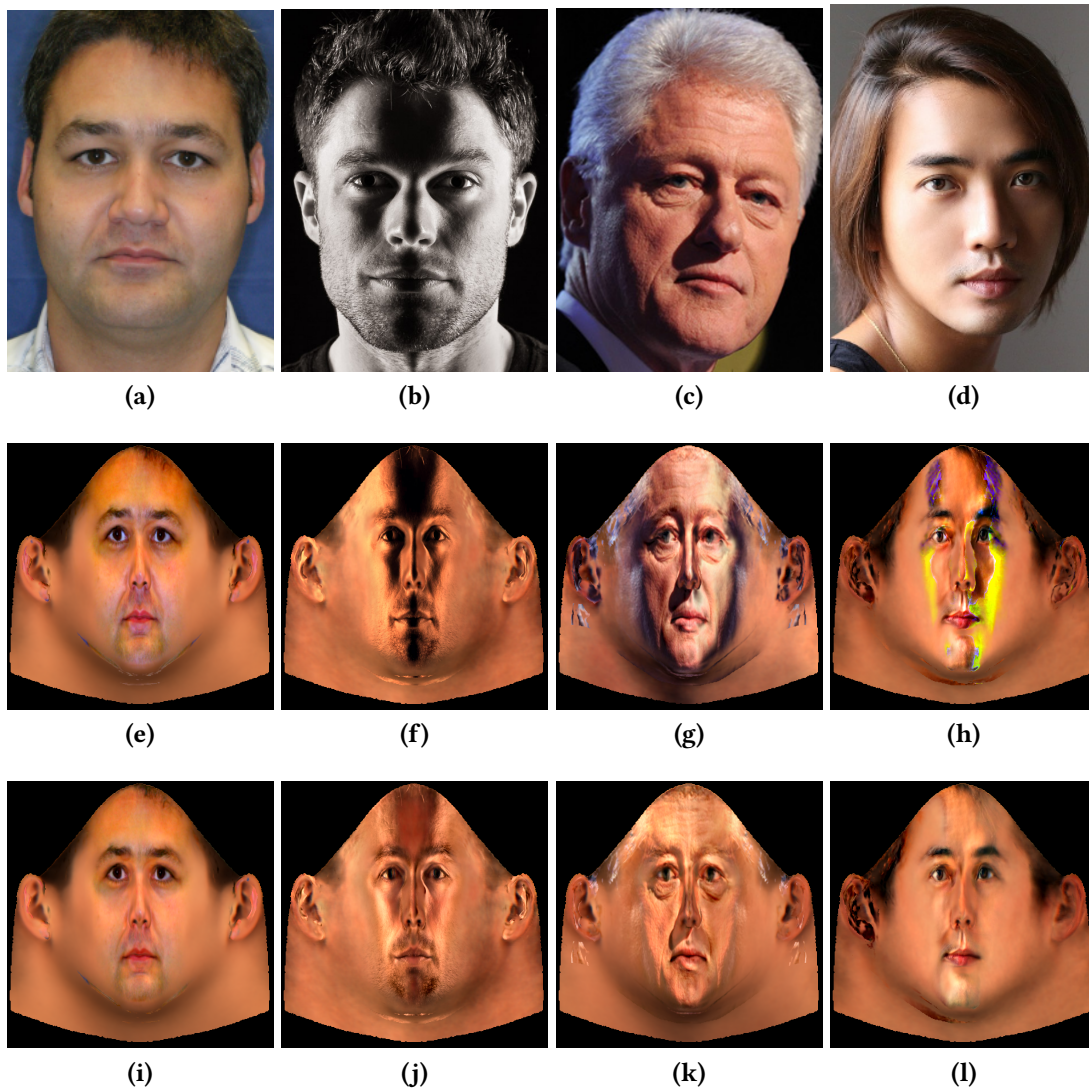


Figure 3.12: Remove the illumination effect from the face model. The mid row is the result of the 3DMM. The bottom row is with proposed inverse lighting.

3.6.2 Lighting Design

Section 3.6.2 explains the joint work from [SPB16]. The adaptation of an automatic landmark localization algorithm to the 3DMM fitting, which enables a fully automatic 3DMM fitting, is the contribution of our co-author, Marcel Pietraschke. The automatic landmark localization is briefly explained as an alternative to the manual initialization of the 3DMM fitting.

The goal of a post hoc lighting design is to provide a method for applying new lighting to an already captured image or even a painting of a face. The user draws a coarse sketch of the desired lighting on the face image with a few strokes, and our algorithm renders the realistically relighted face into the original image. Lighting design methods are briefly reviewed in Section 2.1.3. We need to deal with the following challenges:

- The 3D information must be estimated from the input image. This estimation should include surface normals and concavities of the object, necessary for calculation of reflected light and cast shadows.
- The model must be aligned on the face in the image.
- The albedo (intrinsic texture) of the face must be extracted from the 2D image, taken under uncontrolled condition.
- The illumination setup must be inferred from the user’s coarse sketch.

The lighting design algorithm consists of the following steps (see also Figure 3.13 and 3.15):

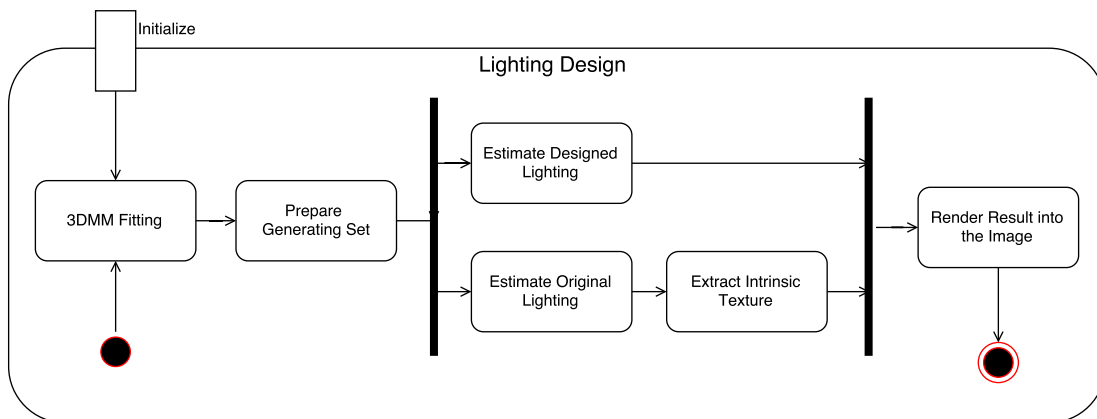


Figure 3.13: This is the UML Activity Diagram for the proposed lighting design algorithm. It is initialized by the user with the painted-on image which indicates the desired lighting. The 3DMM fitting also needs the original image and the landmarks (Figure 3.14b) to estimate the 3D face model, pose, etc. (Figure 3.14c). A generating set is prepared for the original image. Then, the algorithm uses this generating set to, in parallel but separately, calculate the lighting from the original and the painted-on image. The lighting from the original image is only necessary for intrinsic texture extraction (Figure 3.14d). The face model with the extracted intrinsic texture from the original image is then rendered under the estimated lighting from the painted-on image (Figure 3.14e) to produce the result (Figure 3.14f). See also a more detailed pipeline which shows the processes, inputs, intermediates products and final result in Figure 3.15.

1. The user paints coarse shading into the image (Figure 3.14a).
2. Manually (or automatically), the facial landmarks are set (Figure 3.14b).
3. The face model is estimated with 3DMM fitting (Figure 3.14c).
4. The intrinsic texture is recalculated (see 3.6.1 and Figure 3.14d).
5. The designed lighting is estimated from the painted-on image (Figure 3.14e).
6. The 3D face is rendered under new lighting (see 2.5.1 and Figure 3.14f).
7. The rendered face is composited into the original image (Figure 3.14f).

The advanced algorithm is shown in three figures. Figure 3.13 provides an abstract overview of the steps. In Figure 3.14, we see the inputs, intermediate products and final result of the algorithm. These two are combined in the extended diagram in Figure 3.15. All the information that is given about the face is provided in a single 2D image. The proposed inverse lighting delivers promising results for harsh illumination conditions and estimates realistic illumination effects, such as soft cast shadows, colorful lighting, multiple light sources, specular highlights and Fresnel in grazing angles. These are necessary to produce a variety of desirable results. In Figure 3.13, parallel to the intrinsic texture extraction from the original input image, a realistic lighting is estimated from the painted-on image (see Figure 3.14). This is a software-only solution which can be integrated in a photo editing tool. In [SPB16], we see that this tool can work fully automatically by adapting a facial landmark localizer [ZR12] to initiate the 3DMM fitting algorithm.

Alternative Initialization with an Automatic Landmark Localization

In [SPB16], the landmark localizer of Zhu and Ramanan [ZR12] is used to initialize the 3DMM fitting algorithm. This makes the inverse rendering algorithm fully automatic. As a result, the user interaction in the lighting design process is limited to providing the input image and the paint strokes. The algorithm from [ZR12] detects more than 50 landmarks for a frontal face image, however, not all of them are useful in 3DMM fitting. Also, the useful landmarks need to be matched to positions on the 3DMM's 3D face model. Therefore, the output of the landmark localizer needs to be adapted to what 3DMM fitting expects as input. Although, the automatic landmark localization and its adaptation to the 3DMM fitting are not a contribution of this dissertation, as explained at the beginning of this section, a short explanation of this method follows.

First the algorithm [ZR12] finds the locations of the facial landmarks on the input image. These locations are used to crop the input image down to a bounding box that includes the face. This cropped image is used in the rest of the lighting design algorithm. Not all detected landmarks are used or are even usable for the initialization of the 3DMM fitting.

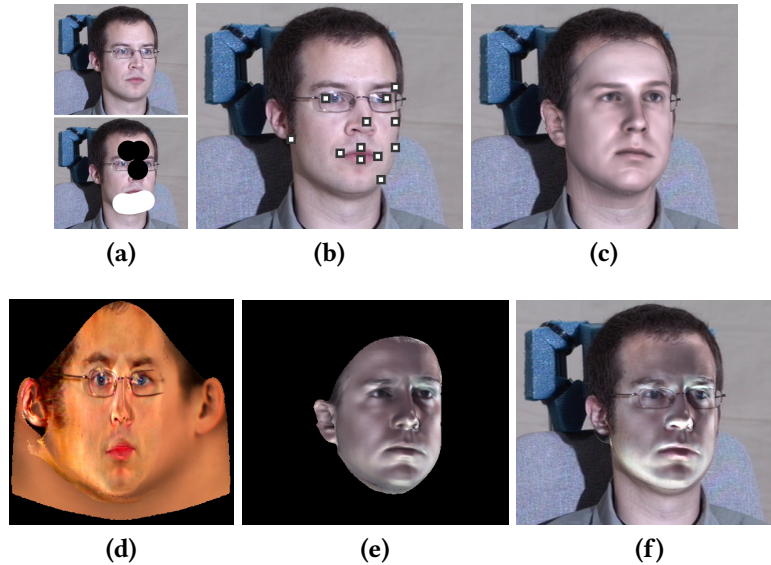


Figure 3.14: The process of lighting design: Based on the input image 3.14a (top) from PIE [GMC⁺10], the algorithm finds landmarks 3.14b and reconstructs a 3D model 3.14c using 3DMM fitting with a simplified lighting model. The 3DMM texture is replaced by a better estimate intrinsic texture 3.14d. Lighting estimation on the painted-on image 3.14a (bottom) and relighting of the 3D face using an empirical BRDF model simulates the new appearance 3.14e, which is composited into the image 3.14f. This demonstration is from [SPB16]. See also Figure 3.13 3.15.

Depending on the pose of the face, an optimal set of landmarks are selected automatically. These landmarks are in the 2D image coordinates and need to be matched with their corresponding locations on the 3D face model of 3DMM. Thereby, two different types of landmarks are considered; fixed landmarks and contour landmarks. The fixed landmarks indicate the positions of the facial features, e.g. the eyes, the nose and the mouth. The contour landmarks mark the silhouettes of the face. For each contour landmark, the nearest vertex on the silhouette of the 3D face model is searched for in an iterative process that re-estimates the pose based on the current set of contour markers. Finally, among all the visible landmarks, only those that are most suitable for the 3DMM fitting are used (see Figure 3.14b). This also prevents unnecessary noise, as a result of the inaccuracies of the automatically detected landmarks, to be fed to the 3DMM fitting. For more detail and illustrations of this process see [SPB16].

Figure 3.14 shows typical input data, intermediate results of automatic landmark localization, 3DMM fitting, extracted texture, estimation of designed lighting and the final result. Using a physically plausible lighting model –the VLS– and putting the focus on faces lead to a more user-friendly interaction that overcomes weaknesses of paint-based lighting design methods, addressed mentioned by [KP09].

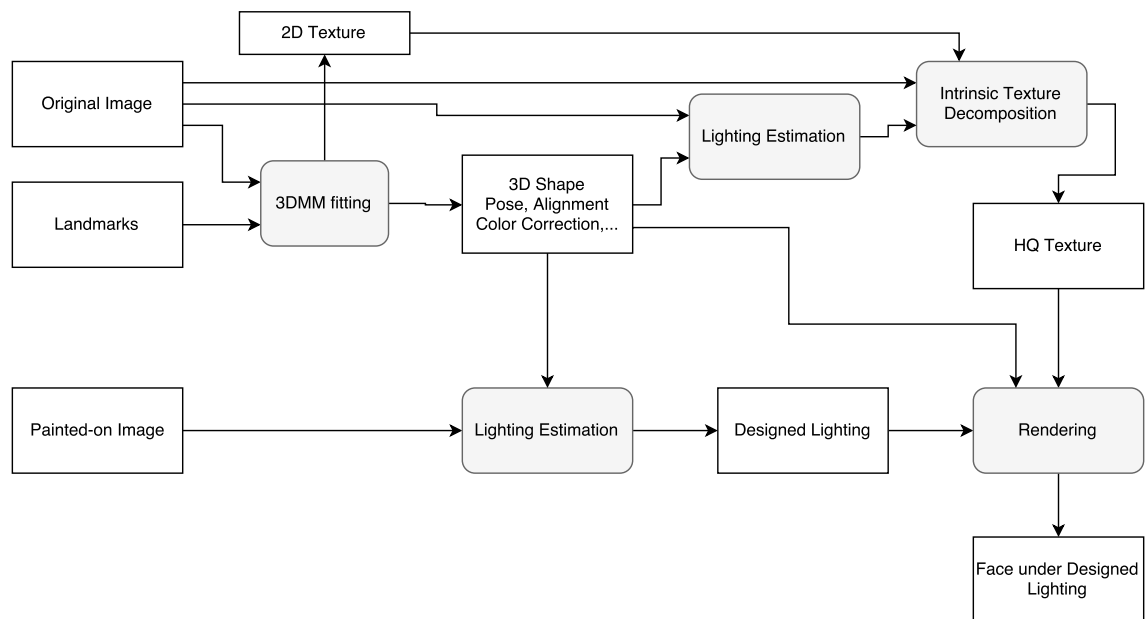


Figure 3.15: This diagram is a nonstandard diagram similar to the UML Activity Diagram in Figure 3.13, to show the pipeline and inputs, intermediate products and final outputs of the lighting design algorithm. The rectangles with sharp corners are input/output data. The only inputs of the lighting design algorithm are the three inputs without an incoming arrow on the left side of the diagram. The rectangles with rounded corners are processes. The final output of the diagram is the one on the bottom right with no arrow leaving from its rectangle. See also Figure 3.13 and 3.14.

Wrong Skin Albedo under the Chin

This error shows on some images when experimenting with the proposed *lighting design* software. The average texture which is used to render the gallery images includes shading effects that are apparently persistent in the original 3D scans that are captured for the construction of the 3DMM. The texture for the area under the chin is darker than the rest of the skin, due to the lower illumination during the 3D scans which are averaged to build this average texture Figure 3.16a. In our experiments, usually this is compensated for a few low intensity light sources which illuminate this area but do not represent any light sources in the environment of the face in the input image. However, some unwanted artifacts appear on the face after lighting design (the extra highlights which appear under the chin in Figure 3.16c), which disappear (see Figure 3.16) when the average texture is manually corrected (Figure 3.16b) before it is used in the preparation of the generating set. Such systematic errors exist in any scanning system because the diffuse reflectance of a perfectly illuminated face is an open problem in appearance acquisition. Previous works which intend to estimate the diffuse albedo address this challenge with multi-modal approaches, e.g. [WMP⁺06]. However, in absence of a capable scanner setup, post processing manipulations are necessary to remove unwanted errors. In our experiments the use of this manually corrected average texture is only necessary for the lighting design application where the inverse lighting algorithm is challenged with the nonrealistic lighting effects (the coarse sketches by the user) on the face.

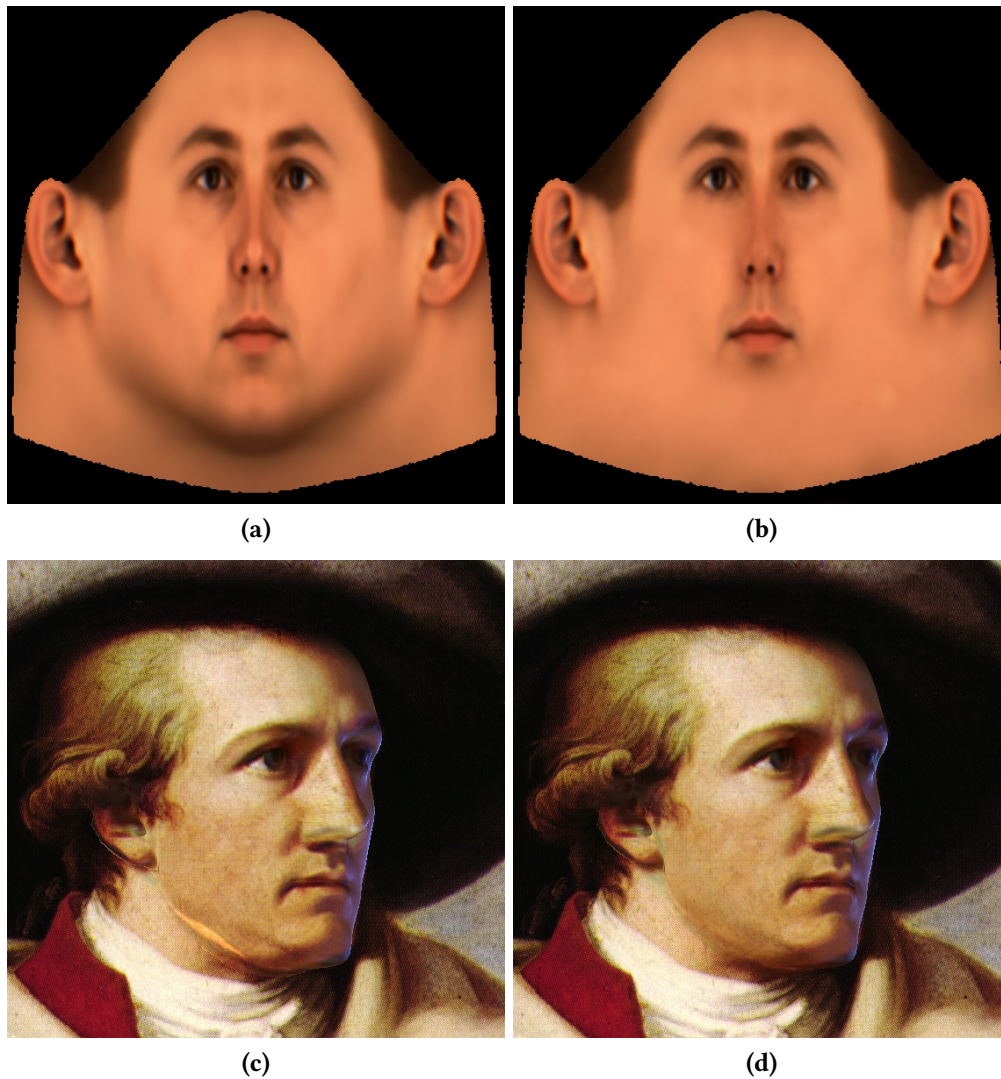


Figure 3.16: This is a comparison of the proposed lighting design using inverse lighting from [SB15b] with two different average textures. Figure 3.16a is the average texture from 3DMM which is used in [SB15b]. The rendered result shows unintentional lighting effects which appear as highlights under the chin in Figure 3.16c. Figure 3.16b is the proposed corrected texture which leads to the relighting result 3.16d.

3.7 An Effort to Improve the Estimated Shape after Inverse Lighting

Although 3DMM fitting algorithm delivers promising result, the accuracy of the estimated shape of the face from the 2D image is still limited. In presence of harsh illumination, the 3DMM fitting performs worse than on neutrally illuminated images. The idea is to use the better estimation of lighting, as proposed in this dissertation, to increase the performance of the 3DMM fitting in a second round. Unfortunately, a visible improvement of the 3DMM's fitting algorithm is not achieved in our efforts, nonetheless, the extents of the modifications in the fitting algorithm are explained to open the topic for a possible future work. The 3DMM fitting operates in two separated spaces of eigenvectors: S space which spans a facial shape and T space which spans the facial texture. The fitting is an stochastic gradient descent to find the coefficients for the linear combination of these eigenvectors in their respective space, together with lighting and other rendering parameters. The lighting parameters are limited to RGB color for an ambient term and RGB color and XYZ direction vector for one directional light. One straightforward idea is to add more light sources which expands the number of the optimization parameters and consequently the dimensionality of the linear system. Also, the reflectance function needs to be replaced with the Torrance-Sparrow and dipole functions (see Section 3.2.2), which affects the calculation of the gradients. To limit the scope of this research, a simpler approach is discussed here.

Let the vector \vec{q} contain the lighting parameters in the optimization algorithm of 3DMM fitting and s_{mean} and t_{mean} be the average shape and average texture, which are the centers of coordinates in the S and T spaces respectively. The lighting is estimated from the input image, according to the proposed algorithm in Section 3.5. The estimated lighting is applied to the face model to calculate the illumination maps for diffuse and specular. Then, these illumination maps are applied to the average texture t_{mean} to be considered in the morphable model as a per se fixed illumination. Moreover, the lighting parameters vector \vec{q} is fixed to $(0, 0, 0, 0, 0, 0, 0, 0, 0)^T$ to make sure the cost function of the fitting algorithm is being minimized without changing the lighting. The rest of the 3DMM fitting remain intact, except for the tuning that needs to be touched. The final results show a very negligible lower error per pixel compared to the result from the previous step, however, the images look identical. Maybe a shape from shading or shape from cast shadow algorithm leads to more visible improvements, yet, this remains out of the scope of this thesis.

Chapter 4

Estimation by Optimization

4.1 Linear Model

The superposition principle of light and the idea of illumination cone (Section 2.4) lead to an analysis by synthesis hypothesis in Section 3.2, upon which the formal solution to the inverse problem is proposed in this chapter. A primary cost function is introduced based on commutative pixel error between the generated image and the input image. This distance function is regularized and minimized in this thesis. Two different probabilistic modelings are proposed and some alternatives are discussed. We start by explaining the cost function from a simple abstract state to the complicated modifications that are forced by the nature of the problem, such as the nonnegativity constraint, and those that are forced by the probabilistic modeling, such as hyperparameter optimization.

Assuming the input image I belongs to the synthetic illumination cone that the generating set \mathbf{C} spans, the unknown intensities of n light sources from the VLS (Section 3.2.1) are equal to the coefficients $\mathbf{x}_i, i \in [1..n]$, for the linear combination (4.1).

$$\mathbf{I} = \mathbf{C}\vec{\mathbf{x}} \tag{4.1}$$

The summation of images $\mathbf{C}\vec{\mathbf{x}} = \sum_{i=1}^n \mathbf{x}_i \mathbf{C}_i$ is inherently an image from the synthetic illumination cone, and with the right coefficients \mathbf{x}_i it is equal to the input image. The model (4.1) is inaccurate because \mathbf{I} is real (not rendered) and images in the gallery \mathbf{C} are synthetic. Finding a solution $\vec{\mathbf{x}}$ involves more than just the (pseudo-)inverse of \mathbf{C} , for the following reasons [SB15b, SB17]:

1. For the purpose of this thesis, only nonnegative $\vec{\mathbf{x}}_i$ are allowed.
2. The surface normals are unavailable for calculating \mathbf{C} . Instead an estimated 3D face model is used which is suboptimal. Note that the focus is on complex lighting.

3. The reflectance function of the subject's skin is not available. Instead a generic skin reflectance according to Section 3.2.2 with the average texture from 3DMM are used.
4. The unknown sensitivity curve of the camera sensor is slightly nonlinear. The input images are not calibrated.
5. Estimation of quality of an image, its white balance, and optical features of the camera and lens are open problems in computer vision. These unknowns might even introduce nonlinearities to the mathematical model.
6. Digitizing and storing the input image also involves compression, the use of format specific color spaces and other lossy manipulations which change the data in an unrecoverable way.
7. We need to deal with unreliable input. Parts of the face might be covered under hair or other objects with completely unknown reflectance and geometry. These areas add harmful noise to the linear combination.
8. The dense correspondence problem is solved also by estimation. Especially, when the background pixels might be mistaken as face, the summation is adversely affected.

For these reasons, solving (4.1) with SVD (Singular Value Decomposition) hardly leads to an acceptable result. This kind of problem is usually solved with stochastic optimization algorithms [Ric72, Luc74], direct search [HJ61] or with a gradient based approach such as Newton-Raphson for a linear Least Squares representation, as proposed primarily in [SB15b]. The regularized least squares from [SB15b] gives the first solution, which is extended with hyperparameter optimization in [SB17]. This dissertation proposes both, however, the latter handles occlusions automatically and it is a more stable solution for unreliable input data or estimated models.

4.2 Cost Function from the Generative Model

Considering (3.6) and (3.12), we construct (4.2) which is the color corrected generative model on the superpixel representation of the linear combination (4.1). Accordingly, \mathbf{Y} is a vector of RGB values of m_S superpixels that represents the input image (measurements or observation), and \mathcal{A} is a $(m_S \times n)$ matrix of superpixels that represents the generating set \mathcal{C} . The offset \mathbf{o} and a color saturation and contrast transformation matrix $T_{3 \times 3}$ are from Section 3.2.3.

$$\mathbf{Y} = \mathbf{o} + T \mathcal{A} \vec{x} \quad (4.2)$$

Note that all operations are performed separately in each color channel. The only operation that mixes color channels is the matrix-vector multiplication of the matrix $T_{3 \times 3}$ in each 3×1 pixel of $\mathcal{A} \vec{x}$. The color correction term is the only operation in the whole dissertation that mixes color channels and it is applied only after the superposition, according to (4.3).

$$\mathbf{Y}(p) = \mathbf{o} + T \begin{pmatrix} \sum_{i=1}^n a_i^r(p) x_i^r \\ \sum_{i=1}^n a_i^g(p) x_i^g \\ \sum_{i=1}^n a_i^b(p) x_i^b \end{pmatrix}_{3 \times 1} \quad (4.3)$$

where $a_i^r(p)$ is the value of the red channel of p th superpixel from \mathbf{A}_i of the gallery \mathcal{A} . Similarly, x_i^r is the value of the i th coefficient in the red channel. The formulation is similar for green and blue channels.

By solving (4.2), each triple \mathbf{x}_i gives the RGB values of the corresponding light source number i of the VLS, which has been used to render \mathbf{C}_i during the generative set preparation step in Chapter 3.

Using Image Compression Must Preserve the Additivity of Light

Not every image compression preserves the linear system (4.1), e.g. median filter. The offset \mathbf{o} rises the main concern regarding this problem. One option is to remove the offset or the color correction completely from the equation by applying its inverse to the input image on the left side of the equation 4.1, $\mathbf{I} \rightarrow T^{-1}(\mathbf{I} - \mathbf{o})$ before the compression and suppress it on the right side of the equation accordingly, such as in the compressive sensing method in [HCSBL15] and in the approach from Section 4.7.2.

The proposed methods in this thesis (i.e. the Gaussian blur in Section 3.4.1, used in [SB15b, SPB16], and the averaging step of the superpixel method from Section 3.4.2) preserve the linearity and the validness of the equation system. They both calculate weighted summations of the pixels of a given area or filter window, with the weights such as a_p that build a affine hull:

$$\sum_{p \in \text{face}} a_p = 1. \quad (4.4)$$

In the following paragraphs, we show the condition (4.4) for two pixels (p and q) of the face and a generating set built of two images (\mathbf{C}_i and \mathbf{C}_j). The generalization to more pixels and larger generating sets is then trivial.

The equation (4.6) says two pixels are color corrected superpositions of two corresponding pixels over the generating set.

$$\begin{aligned} \mathbf{I}(p) &= \mathbf{o} + T(\mathbf{x}_i \mathbf{C}_i(p) + \mathbf{x}_j \mathbf{C}_j(p)) \\ \mathbf{I}(q) &= \mathbf{o} + T(\mathbf{x}_i \mathbf{C}_i(q) + \mathbf{x}_j \mathbf{C}_j(q)) \end{aligned} \quad (4.5)$$

Their weighted summation is equal to the color corrected superpositions of the weighted summation of the corresponding pixels in the generation set:

$$a_p \mathbf{I}(p) + a_q \mathbf{I}(q) = \mathbf{o} + T \left(\mathbf{x}_i (a_p \mathbf{C}_i(p) + a_q \mathbf{C}_i(q)) + \mathbf{x}_j (a_p \mathbf{C}_j(p) + a_q \mathbf{C}_j(q)) \right) \quad (4.6)$$

where a_p and a_q are two constant scalar values. These are the filter coefficients for averaging in superpixels or for Gaussian blur. We know that the color correction term is raised by the difference between the overall color contrast and brightness of the input image and the rendered images, thus, there is an image \mathbf{I}' , where $\mathbf{I}(p) = \mathbf{o} + T\mathbf{I}'(p)$. The $\mathbf{I}'(p)$ is a pixel which is in the *neutral* color contrast and brightness of the rendering machine. As a result, \mathbf{I}' is equal to the linear combination of the rendered images at pixel p without color correction (4.7).

$$\mathbf{I}' = \mathbf{C}\vec{\mathbf{x}} = \sum_{i=1}^2 \mathbf{x}_i \mathbf{C}_i \quad (4.7)$$

Now, when we consider the predicate of (4.6) for those two pixels of \mathbf{I}' , see (4.8), a problem shows up if the weights of the compressing filter do not build an affine hull ($a_p + a_q \neq 1$).

$$\begin{aligned} a_p \mathbf{I}(p) + a_q \mathbf{I}(q) &= a_p (\mathbf{o} + T\mathbf{I}'(p)) + a_q (\mathbf{o} + T\mathbf{I}'(q)) = (a_p + a_q) \mathbf{o} + T (a_p \mathbf{I}'(p) + a_q \mathbf{I}'(q)) \\ &\neq \\ &\mathbf{o} + T \left(\mathbf{x}_i (a_p \mathbf{C}_i(p) + a_q \mathbf{C}_i(q)) + \mathbf{x}_j (a_p \mathbf{C}_j(p) + a_q \mathbf{C}_j(q)) \right) \end{aligned} \quad (4.8)$$

The Gaussian blur and the averaging in center assignment step of superpixels both use only normalized weights a for each window (or cluster) and preserve the linear system with no further considerations. This explanation is to provide the precondition for the future work which might involve novel ideas for compression, be it an image-based filter or a numeric approach.

4.2.1 Least Squares

The unknown coefficient vector $\vec{\mathbf{x}}$ can be estimated by minimizing the distance between the left and right side of (4.2). A classic distance function is the Euclidean distance that leads to a least squares cost function (4.9).

$$\vec{\mathbf{x}}_{solution} = \operatorname{argmin}_{\vec{\mathbf{x}}} \|\mathbf{o} + T(\mathcal{A}\vec{\mathbf{x}}) - \mathbf{Y}\|_2^2 \quad (4.9)$$

The minimization problem (4.9) with a nonnegativity constraint on $\vec{\mathbf{x}}$ is the main objective of this chapter. However, we never minimize the least squares without regularization in this thesis. The probabilistic modeling backgrounds of the regularized least squares are

explained with MAP approach to prepare the theories for a more elaborate probabilistic modeling –the Joint Maximum A-Posteriori (JMAP)– that currently conforms the state of the art [SB17].

4.3 Maximum A-Posteriori (MAP)

The cost function (4.10) is a regularized version of the least squares (4.9). Although the denominators are usually set to one (ignored) in the literature [MD96], they are mentioned here to be a prelude for the topic of hyperparameters in the following sections.

$$\mathbf{e}_{MAP}(\vec{\mathbf{x}}) = \sum_{p=1}^{m_S} \frac{(\mathbf{o} + T \sum_{i=1}^n \mathbf{A}_i(p) \mathbf{x}_i - \mathbf{Y}(p))^2}{\sigma_p^2} + \eta \sum_{i=1}^n \frac{(\mathbf{x}_i - \vec{\mathbf{x}})^2}{\sigma_i^2}, \quad (4.10)$$

where the triple \mathbf{e}_{MAP} denotes the accumulative cost (called also error, energy or distance) for all pixels in separate R, G and B channels and depends on the independent vector $\vec{\mathbf{x}}$. The \mathcal{A} is the matrix representation of the generating set in (4.2), $\vec{\mathbf{x}}$ the vector of unknown coefficients and \mathbf{Y} the superpixels of the input image (observation). The constant vector of RGB values $\vec{\mathbf{x}}$ is the expected value for the unknown parameters $\vec{\mathbf{x}}$, and η is the regularization factor that is tuned by hill climbing. We will now derive the MAP formulation of (4.10) from the posterior probability (4.11) with respect to Bayesian law.

$$\prod_{p=1}^{m_S} P(\vec{\mathbf{x}} | \mathbf{Y}(p)) = \prod_{p=1}^{m_S} P(\mathbf{Y}(p) | \vec{\mathbf{x}}) \prod_{i=1}^n P(\mathbf{x}_i), \quad (4.11)$$

where $P(\cdot)$ is the Gaussian probability density function:

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.12)$$

The minus logarithm of the right side of the equation (4.11) leads to the right side in (4.10).

$$\begin{aligned} -\log \left(\prod_{p=1}^{m_S} P(\mathbf{Y}(p) | \vec{\mathbf{x}}) \cdot \prod_{i=1}^n P(\mathbf{x}_i) \right) = \\ \sum_{p=1}^{m_S} \frac{(\mathbf{Y}(p) - (\mathbf{o} + T \sum_{i=1}^n \mathbf{A}_i(p) \mathbf{x}_i))^2}{\sigma_p^2} + \eta \sum_{i=1}^n \frac{(\mathbf{x}_i - \vec{\mathbf{x}})^2}{\sigma_i^2} \end{aligned} \quad (4.13)$$

Thereby, the normalization factors, the $\frac{1}{\sqrt{2\pi}\sigma^2}$ parts, are ignored, as they do not change with the independent variables. Also, the position of the expected value and the random

variable are swapped in the exponent of the likelihood term in the cost function (4.10) to avoid the minus sign in further calculations of the derivatives.

Minimizing the cost function (4.10) is the same as minimizing the minus logarithm of the right side of the Bayes law (4.11) which gives the posterior. This is equivalent to maximization of the posterior probability ($\prod_{p=1}^{m_S} P(\vec{x}|Y(p))$), and is therefore called Maximum A-Posteriori estimation or in short “MAP”. Without the regularization, the probabilistic model is reduced to the maximization of the likelihood, called Maximum Likelihood (ML) approach and is not used in this thesis.

The (σ_p^2) and (σ_i^2) are nothing but the standard deviations of Gaussian probability distribution functions for likelihood and prior in (4.11) respectively. The standard deviations –the denominators in log of the likelihood and prior– are heuristically set. For the likelihood, it is $(\sigma_p^2 = 1 \forall p \in [1..m_S])$. However, a geometric blind mask (Section 3.2.4) is applied to remove the eyes and ears from the equations. Moreover, a manually drawn occlusion mask omits the occluded areas from the equations in [SB15b]. Hence, one can argue that the use of such masks is equal to giving the standard deviation of the likelihood a very large value (infinity) for occluded and masked pixels of the face. We expand on this argument later when the JMAP approach is introduced. The standard deviation σ_i^2 of the prior is set based on the heuristic (4.14).

$$\frac{1}{\sigma_i^2} = \frac{m_{i,nonzero}}{m}, \quad (4.14)$$

where $m_{i,nonzero}$ the number of nonzero (nonblack) pixels in image C_i (before segmentation), and m number of nonoccluded face pixels. Based on this heuristic, the regularization term penalizes the deviations of \mathbf{x}_i from the $\bar{\mathbf{x}}_i (= \mathbf{0})$ even more strongly when the light source i illuminates fewer face pixels in the corresponding rendered image C_i . This makes sense because at least at the beginning steps of each optimization, illuminating the scene with a few number of light sources that cover larger areas are more efficient than illuminating it with a large number of light sources that each cover only a small area. In extreme cases, when a light source does not illuminate any visible facial pixel, $\frac{1}{\sigma_i^2} \rightarrow 0$ and consequently $\sigma_i \rightarrow \infty$, the \mathbf{x}_i is set to zero and the dimension i is discarded during the optimization. In less extreme cases, for instance when a backlight illuminates only a small ribbon on the silhouette of the face, the corresponding coefficient \mathbf{x}_i is strongly penalized due to the small denominator σ_i of the regularization term, however, if the illuminated ribbon corresponds to an effect which exists in the input image, the least squares term finally wins the battle and optimizes the \mathbf{x}_i to a nonzero value that leads to the replication of the highlights. This win is promoted by a *gradual decrease* of the regularization factor η in the implementation of the iterative optimization (see Section 4.5) and by selecting a zero prior $\mathbf{x}_i = (0, 0, 0)^\top \forall i \in [1..n]$, explained in Section 4.4.1.



Figure 4.1: Result on synthetic input is almost perfect.

The nonnegativity of the \vec{x} is forced through projection of the negative values to zero in every several iteration (see Section 4.5.1). Although the results of MAP [SB15b] are promising and even almost perfect on synthetic input image (see Figure 4.1), this approach is very sensitive to noise, e.g. occlusion and error in the estimated geometry or its correspondence. Next, the joint parameter and hyperparameter optimization is proposed to address such problems automatically.

4.4 Joint Maximum A-Posteriori (JMAP) for Hyperparameter Optimization

Arguably, some hyperparameters or metaparameters already exist in the MAP approach [SB15b]. The manually drawn occlusion masks or the blind masks are binary weights that are applied to the likelihoods of the pixels. With an ideal probabilistic modeling, a proper standard deviation (σ_p) of the likelihood distribution for each pixel should have taken care of these. Even the σ_i s, which are heuristically set for the regularization term to discriminate between different light sources, can be seen as metaparameters. Unfortunately, these metaparameters are not given. Therefore, in [SB17], we propose an analytic approach to estimate equivalent hyperparameters as probabilistic entities (hidden random variables). More specifically, the hyperparameters $\vec{\beta}$ and $\vec{\theta}$ are introduced and optimized together with the coefficient vector \vec{x} by minimizing the cost function (4.15). We show this hierarchical Bayesian model of parameters and hyperparameters with a Probabilistic Graphical Model (PGM) diagram [KF09] in Figure 4.2.

$$\begin{aligned}
 e(\vec{x}, \vec{\theta}, \vec{\beta}) = & \sum_{p=1}^{m_S} \beta_p \left(\mathbf{o} + T \sum_{i=1}^n \mathbf{A}_i(p) \mathbf{x}_i - \mathbf{Y}(p) \right)^2 + \eta \sum_{i=1}^n \theta_i (\mathbf{x}_i - \bar{\mathbf{x}}_i)^2 \\
 & + \eta_\beta \sum_{p=1}^{m_S} (\beta_p - \bar{\beta}_p)^2 + \eta_\theta \sum_{i=1}^n (\theta_i - \bar{\theta}_i)^2,
 \end{aligned} \tag{4.15}$$

where, η , η_β and η_θ are tuning parameters which are set manually for the algorithm. They control the weight of the regularization that is applied to parameters \vec{x} and the hyperpara-

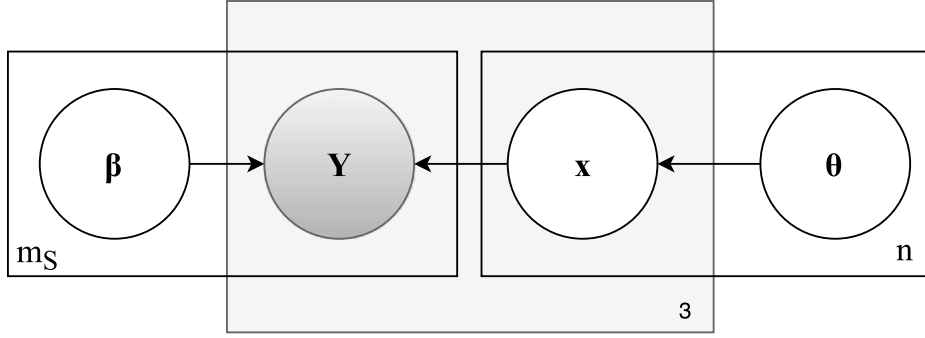


Figure 4.2: A PGM (Probabilistic Graphical Model) diagram [KF09] for the proposed JMAP approach shows the dependencies between the observation Y , parameters \vec{x} and hyperparameters $\vec{\beta}$ and $\vec{\theta}$. For example, Y depends on \vec{x} and $\vec{\beta}$. The constant in the corner of each plate shows the dimension of the enclosed random variables. Note that the hyperparameters do not discriminate between the three color channels, however, the parameters (\mathbf{x}) do. Furthermore, the dependencies are described with probability density functions which are not shown in the diagram explicitly. Compare (4.16) with (4.15).

eters $\vec{\beta}$ and $\vec{\theta}$. The \bar{x}_i -s, $\bar{\beta}_p$ -s and $\bar{\theta}_i$ -s are expected values for the respective parameters and hyperparameters. Minimizing the cost $e(\vec{x}, \vec{\theta}, \vec{\beta})$ resembles the maximization of the joint probability law (4.16), based on a hierarchical Bayesian approach.

$$\prod_{p=1}^{m_s} P(\vec{x}, \vec{\theta}, \beta_p | Y(p)) = \prod_{p=1}^{m_s} P(Y(p) | \vec{x}, \beta_p) \cdot \prod_{i=1}^n P(x_i | \theta_i) \cdot \prod_{p=1}^{m_s} P(\beta_p) \cdot \prod_{i=1}^n P(\theta_i), \quad (4.16)$$

where $\vec{\theta}$ is a vector of n , and $\vec{\beta}$ a vector of m_s scalar values. Similar to [SB15b], we use a Gaussian distribution over the nonnegative space for \vec{x} (see Section 4.5.1). The logarithm of the probabilistic model (4.16) leads to the cost function (4.15). In probabilistic terms, these hyperparameters are sometimes called “precision,” as they are inverses of the standard deviations for the likelihood $\beta_p = \frac{1}{\sigma_p} \forall p \in [1..m_s]$ and for the prior $\theta_i = \frac{1}{\sigma_i} \forall i \in [1..n]$. Therefore, they are also of a nonnegative nature (See Section 4.5.1).

4.4.1 Expected Values of the Priors

Before the first iteration, the initial parameters are initialized with a small positive value,

$$\mathbf{x}_i = (0.01, 0.01, 0.01)^\top \forall i \in [1..n]. \quad (4.17)$$

This is for a VLS with 100 light sources. Many other small positive values work as well. The hyperparameters θ_i s and β_p s are initialized with 1, which means all directions and

all superpixels contribute to the cost function equally at first (see Section 4.5.4 for further considerations in the values of β_p s). Next we see toward which expected values these parameters are optimized.

The expected values \bar{x}_i ($\forall i \in [1..n]$) are set to zero for its numerical benefits and to promote low intensity light sources and sparsity. Based on our experiments, an L1 prior gets rid of the low intensity light sources too aggressively for the proposed inverse lighting approach. This is observed in the relatively rapid drop of the coefficients to zero and the suboptimal development of the linear combination image. The expected values $\bar{\theta}_i$ are zero too. This leads to weaker regularization on an x_i toward the end of the optimization, so that the prior term gradually becomes obsolete and the log-likelihood term takes over for dimension i , unless θ_i remains large (see Section 4.5.2). The expected values $\bar{\beta}_p$ are set to one to keep the superpixels in the equation system as long as they are helpful in the minimization of the least squares. With a look into the optimization steps and how the requirements, i.e. converging to the desired optimum, the nonnegativity and unreliable data are taken care of in Section 4.5.

4.5 Optimization with Newton-Raphson

Equation (4.15) is the actual cost function that, after experimenting with many modifications of different approaches, turned out the most suitable for the purpose of this thesis. Here an iterative optimization approach is proposed that works in three fronts (see Figure 4.3) iteratively to find the desired optimum \vec{x} . Thus, the error $\vec{e}(\vec{x}, \vec{\theta}, \vec{\beta})$ is minimized with Newton update functions for the color channels of \vec{x} according to the update function (4.18).

$$\vec{x}^{k+1} = \vec{x}^k - \lambda(H^k)^{-1}\nabla^k\vec{e}, \quad (4.18)$$

where the next \vec{x}^{k+1} is calculated based on the previous \vec{x}^k , updated with a product of the pseudo-inverse of Hessian $(H^k)^{-1}$ and the gradient $\nabla^k\vec{e}$ with small steps. This update function can be used for each channel of the coefficient vector \vec{x} (i.e. \vec{x}_r , \vec{x}_g or \vec{x}_b), the $\vec{\beta}$ or $\vec{\theta}$ individually. For the \vec{x} only the diagonal of the pseudo-inverse of the Hessian is calculated according to (4.19).

$$\begin{aligned} h_{ij}^{-1} &= \frac{1}{h_{ij}} & i = j \quad \text{AND} \quad h_{ij} > \varepsilon \\ h_{ij}^{-1} &= 0 & \text{otherwise} \end{aligned} \quad (4.19)$$

where $i, j \in [1..n]$, h_{ij} an entry of the Hessian matrix (second partial derivative with respect to x_i and x_j) and h_{ij}^{-1} is the corresponding entry from the pseudo-inverse of the Hessian matrix. The threshold ε can be a very small value (almost zero) and only avoids the division by zero. A larger threshold leads to smoother illumination.

For $\vec{\theta}$ and $\vec{\beta}$, the second derivative –the Hessian– has no effect, and therefore is ignored.

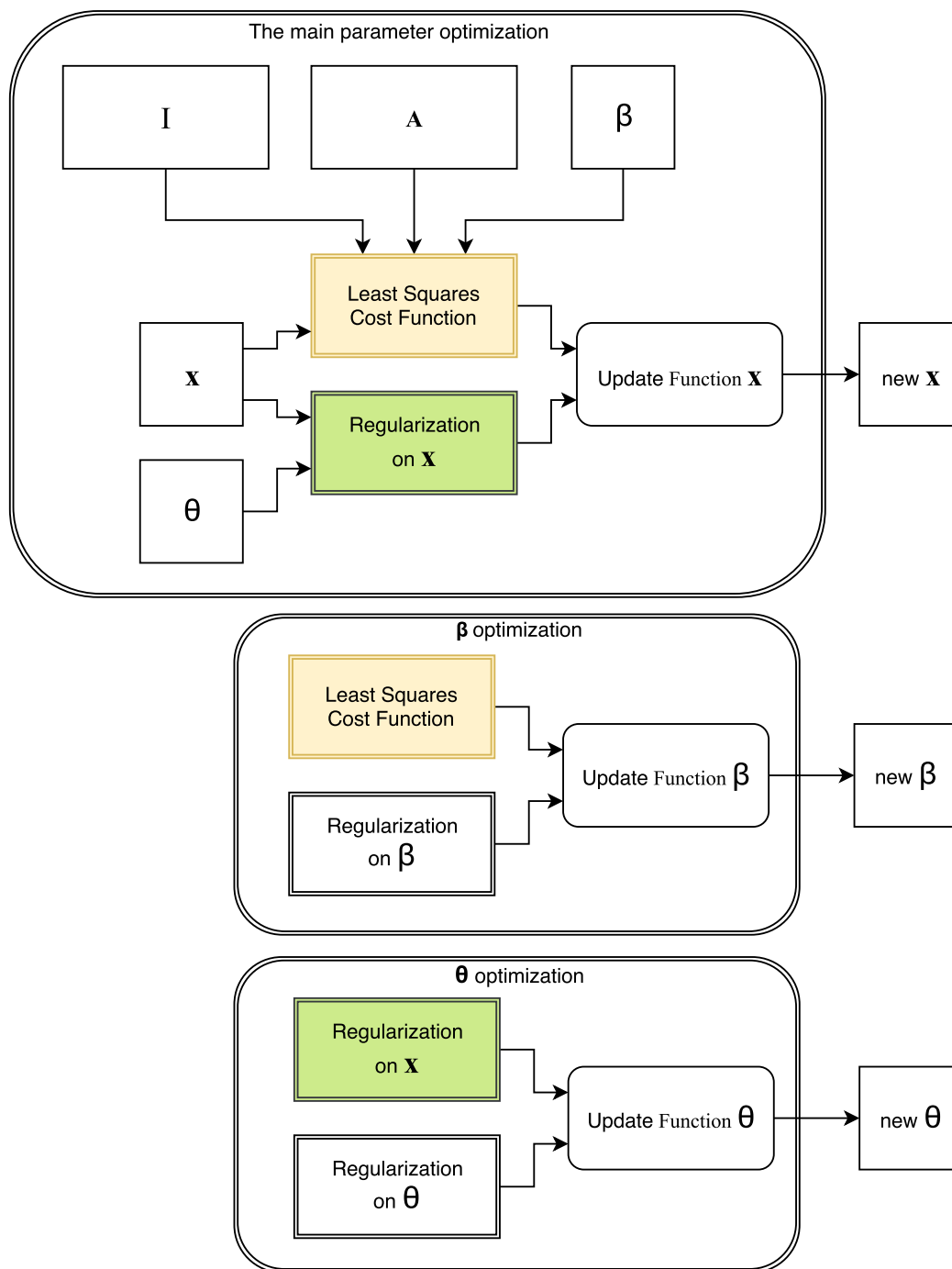


Figure 4.3: These diagrams summarize the proposed optimization for JMAP. The box I is the input image and the box A is a generative set (denoted with \mathcal{A} through the dissertation) which generates an image similar to I if the parameters \mathbf{x} are set optimally. The main goal is to estimate optimal \mathbf{x} , however, two hyperparameters β and θ are jointly optimized and discarded at the end in a JMAP approach. Each $\beta_p \in \beta$ discriminates the error per pixel p and each $\theta_i \in \theta$ discriminates the prior for each parameter $x_i \in \mathbf{x}$. Ignore both hyperparameter optimization modules and you have the overview of the MAP approach with $\beta = 1$ and each $\theta_i = \frac{1}{\sigma_i^2}$ that are (in this case) set heuristically. This chapter explains both approaches in detail.

Consequently, the update function of the hyperparameters becomes a gradient descent with tunable regularization factors η_θ and η_β . The hyperparameters $\vec{\theta}$ and $\vec{\beta}$ are optimized

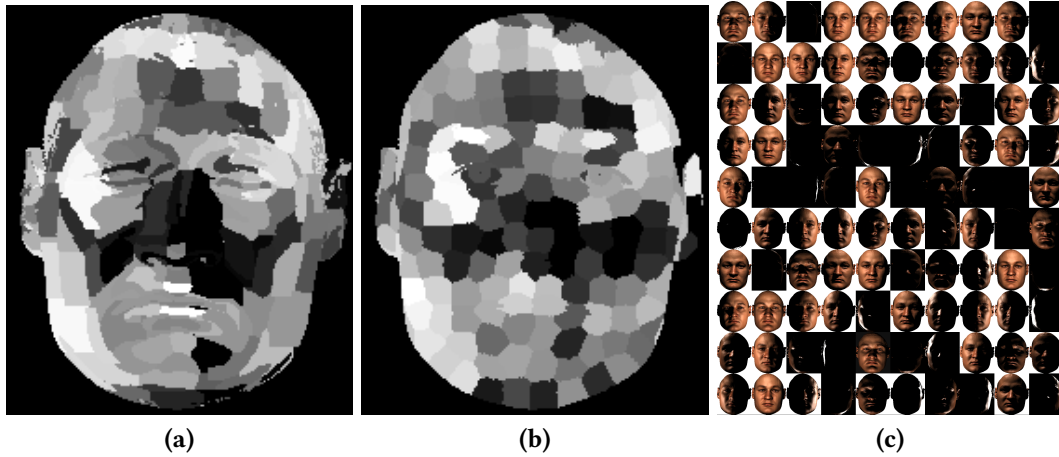


Figure 4.4: This image shows the optimized values for parameters β on geometric superpixels Figure 4.4a, photometric superpixels Figure 4.4b, and θ on the gallery Figure 4.4c calculated during optimization with JMAP approach. The values are normalized to fit in the $[0, 1]$ range. The value of β_p s are optimized to a smaller value or zero in darker areas on the face. The brighter β_p s are those with larger values. The values of θ_i s in Figure 4.4c are converted to binary values and inverted to put the less regularized C_i s in white background. All images C_i in black background are those with $\theta_i \neq 0$ in the middle of the optimization, thus, the corresponding x_i -s are being regularized. Zoom in PDF or compare with Figure 3.5d if necessary. We tune the algorithm so that by the end of the optimization $\theta_i = 0 \forall i$. Therefore, here we show an intermediate step of θ .

less frequently compared to \vec{x} , and not at all towards the final steps.

In Figure 4.4, you see the optimized $\vec{\beta}$ for the input image in Figure 3.5a and an intermediate result for optimization of $\vec{\theta}$. The joint optimization of $\vec{\beta}$ leads to lower significance of the occluded areas, e.g. the forehead's hair area in Figure 4.4a and 4.4b in the cost function (4.15). Similarly, the optimized $\vec{\theta}$, which is applied to the prior on \vec{x} , leads to a heavier regularization of coefficients x_i , whenever C_i contain unwanted highlights or irrelevant illumination to the (in this case) diffuse input image. In Figure 4.4c, you see the generating set \mathcal{C}_{100} . There, a C_i image with a black background indicates that the respective x_i is regularized to zero, meaning the θ_i is optimized to a rather large scalar. The white background informs that the respective x_i is not regularized strongly, meaning the θ_i is optimized towards zero.

4.5.1 Enforcing Nonnegativity and Light-Weight Sparsity

Every few steps, all negative values of \vec{x} are projected to zero (thresholding), with a threshold ε , according to (4.20). This is a bit different for the hyperparameters. A $\theta_i = 0$ would mean that the optimization of x is left completely to the least squares without any regularization, therefore, we choose a nonzero lower bound $\varepsilon_\theta > 0$. Based on a similar logic, the $\vec{\beta}$ has also a ε_β lower bound to keep the pixels in the game. Moreover, our experiments show that selecting a positive lower bound for the hyperparameters leads to a

more regularized, and therefore more stable algorithm. In the final steps, only the parameters \vec{x} are updated, and the thresholding is applied after *every* iteration to secure the nonnegativity of light sources with the condition (4.20). Moreover, during these final steps the thresholding is applied to the luminescence of the light source to avoid accumulation of light sources with very small nonzero value in a single channel and all zeros in other channels. Notice the difference between the nonnegativity constraint on the parameters and on the hyperparameters in (4.20).

$$\begin{aligned} x_{i,\gamma} &= 0 & \forall x_{i,\gamma} < \varepsilon & & i \in [1..n] \ \& \ \gamma \in \{r, g, b\} \\ \theta_i &= \varepsilon_\theta & \forall \theta_i < \varepsilon_\theta & & i \in [1..n] \\ \beta_p &= \varepsilon_\beta & \forall \beta_p < \varepsilon_\beta & & p \in [1..m_S] \end{aligned} \quad (4.20)$$

where ε , ε_θ and ε_β are small positive values, such as 0.001. It is not necessary that these three thresholds are equal. In case of ε it is possible to theoretically find a meaningful threshold where smaller values would have no effect in the final image, depending on the bit rate of the image format, the number of light sources, etc. The $\varepsilon = 0.001$ is not too far from that value for $n = 100$ and integer images ($\in [0..255]$), and is a proper choice according to the experiments. The smaller the ε , the more likely are very small light sources that have no effect in practice and only increase the rendering time. Conversely, a large ε leads to nonsmooth illuminations and omission of the desired optimum from the system. This configuration promotes sparsity only when it is necessary. In other words, a sparse solution is not the primary goal of the optimization but a welcome feature if it doesn't omit the desired low intensity light sources. Hence, we refer to this as light-weight sparsity. An alternative sparse solution is given in [HCSBL15] for applications with low rendering time requirements.

4.5.2 The Hyperparameter Optimization Mechanism

The Newton-Raphson update function (4.18) subtracts a fraction of the first partial derivative of the cost function (4.15) with respect to the updating parameter from the its current value. As a result, whenever the gradient is positive and large, the parameter is decreased towards zero. Now, let us see the first partial derivative of the cost function (4.15) with respect to β_p in (4.21).

$$\frac{\partial e}{\partial \beta_p} = \left(\mathbf{o} + T \sum_{i=1}^n \mathbf{A}_i(p) \mathbf{x}_i - \mathbf{Y}(p) \right)^2 + \eta_\beta 2(\beta_p - \bar{\beta}_p) \quad (4.21)$$

This derivative is positive when it is the addition of two nonnegative terms. The first term of (4.21) only depends on the Euclidean distance at the superpixel p between the \mathbf{Y} and the generative model. If the generative model (the color-corrected linear combination) continuously fails to reproduce a value equal to \mathbf{Y}_p at the respective index p , this leads to a

large first derivative of the cost function (4.15) with respect to β_p , causing a larger decrease of the value of β_p in each iteration. As a result of a nonnegative optimization, β_p converges towards zero for superpixels that are hardly possible to be modeled with the generative model. Consequently, the least squares term for superpixel p is barely considered in the update of the parameters \vec{x} , making the inconsistency areas obsolete or very insignificant in the optimization of \vec{x} . However, the second term of the first derivative (4.21), which comes from the prior on β , keeps the value of the β_p close to one. In other words, the prior $\bar{\beta}_p = 1$ suggests the generative model and the observation Y are consistent with each other unless an occlusion, blind mask or cast shadow ROI mask previously change the value of $\bar{\beta}_p$ to zero or a larger value. A similar strategy leads to a proper optimization of the hyperparameter θ_i , depending on how numerically beneficial the prior assumption over x_i is for minimization of the cost function. Hence, θ_i approaches zero (or ε_θ) during the optimization, when the x_i remains a large value. As a result, x_i is regularized more and more weakly the longer it remains a large value.

This mechanism is essential in the hierarchical Bayesian modeling. Based on our experiments, through watching the gradual development of the values of these hyperparameters and the results, this system works as explained to automatically handle the relative regional inconsistencies between the image and the generative model.

4.5.3 Handling Occlusions and Areas of Misalignment

Hyperparameter optimization automatically handles the inconsistencies between the input image and the model, e.g. misalignments and relative wrong reflectance, suboptimal texture, inaccuracies of the estimated face model and occlusions. Occlusions of the face by hair, hands, glasses, etc. add noise to the estimation of the face model and lighting. In the previous work [SB15b], the user has to draw an occlusion mask on the image to let the algorithm ignore the occluded area. Here, the JMAP automatically handles the occlusions and any other condition that is not estimated with the model (4.2), by optimizing the hyperparameter $\vec{\beta}$ to a smaller value for the respective area. It is possible to incorporate the drawn masks as constraints or expected values for the hyperparameter $\vec{\beta}$, however, it can lead to only marginal improvements. This confirms a clear benefit of hyperparameter optimization [SB17] against the MAP approach in [SB15b].

Inconsistencies Due to Overflown Saturated Highlights

Beside the inconsistencies that are introduced by occlusion, misalignment, inaccuracies in face model estimation and missing reflectance, another source of inconsistency lays in the limited dynamic range. This shows up only after the linear combination on the right side of the equation (4.2) is computed. In saturated areas of the input image (left side of (4.2)), the signal is clipped to a flat constant line, no matter what shape the original signal might

have been. The real value is somewhere equal to or higher than the higher bound of the range (255 for 8-bit images), however, it is unknown and the area provides no information about the original geometry, reflectance, albedo and lighting. On the right side of the equation (4.2), there is no such saturated area in the data because no clipping is involved in any stage of the rendering of each image or after the addition of images in the linear combination. Consequently, the equation (4.2) suggests that a clipped signal is equal to a nonclipped signal. This is a systematic mistake that we can address by simply clipping the right side at 255 and consequently introducing more nonlinearities to the system. Since this does not solve the real problem, we propose to leave it to the probabilistic modeling. The MAP approach has a regularization factor η which is empirically set to a larger value to prevent the overfitting of the model to such clipped signals. Unlike the MAP approach, the hyperparameter optimization of JMAP gradually reduces the β_p for such areas and learn the limitations of the generative model in representation of those areas. We observe this meaningful optimization of the hyperparameters by watching the values. It is also evident in the results in Chapter 5. Notice the lower values the hyperparameter for areas where the model cannot represent the input image in Figure 5.7, the ' β_{opt} ' row. This shows the potential of the hyperparameter optimization approach in dealing with unreliable data. Although the hyperparameters seem ad hoc, since they are discarded after the optimization, their joint optimization with the parameters introduces clear benefits to the proposed algorithm. However, if a lighting effect belongs to a rare class that is not modeled with the generative model, it will be ignored by hyperparameters. Cast shadows are one of these classes of effects.

4.5.4 The Challenge of Cast Shadows

Previous works, which test their algorithms on images with cast shadows, prefer to ignore the cast shadows during the optimization (see Chapter 2). Wang et al. remove the residual of the cast shadow on the intrinsic texture without modeling the light correctly [WLH⁺07]. As a result the estimated lighting does not cause the desired cast shadows anymore and that information is lost. In [BV99], although the cast shadows are accounted for in the lighting model, the algorithm performs generally better when the cast shadows are ignored during the 3DMM fitting. Moreover, the lighting model uses only one directional light which has its limitations.

In contrast with previous work, the proposed approach intends to go even beyond considering cast shadows and instead of seeing them as a disturbing factor, model them through lighting estimation. We propose to do so by discriminately enforcing the cost function to lay more weights on the cast shadow areas. However, the so far explained JMAP optimization, similar to the way that occlusions are handled, tends to suppress cast shadows in the optimization process with rapid decrease of the β_p for the cast shadow areas. To avoid

this, we need to fix β_p s to a constant large value for these areas. In the visualization of the optimized $\vec{\beta}$, e.g. in Figure 3.8g, Figure 3.8h and in the Chapter 5, the superpixels for which the β_p is automatically fixed to a large value are in blue color. The cast shadow areas are marked by the proposed model-based cast shadow detection and segmentation from Section 3.3. This algorithm automatically marks the superpixels of cast shadow areas from the input image and produces the cast shadow ROI mask. This leads to a better estimation of the lighting with respect to cast shadows. However, overdoing the constraint by setting the β_p s to an insanely large value in cast shadow areas might lead to an undesired global illumination that only primarily casts the cast shadows, causing wrong highlights on other areas.

4.6 Implementation Tricks and Magic Numbers

This section covers the empirical part of the implementation that is usually overlooked in the publications. It is without any intention to give positive knowledge facilitate the future implementation of the algorithm and show the numerical challenges that it includes.

Compared to a classic MAP estimation, the proposed method with hyperparameter optimization has more tunable parameters. The λ , η , λ_θ , η_θ , λ_β and η_β are defined in the implementation. The extra parameters that are set manually are lower and upper bounds for these and other hidden variables of the optimization pipeline, intervals that special events happen during the iterating loop, etc. In most cases, the tunable parameters are not really fatal. One might set a group to an arbitrary constant and tune only one or two parameters. This way the degree of freedom is artificially reduced.

Dynamic Step Size and Termination Condition

In gradient based optimizations, usually the precision of the result or the absolute value of the gradient is used as termination condition. In this thesis, a more comprehensive approach is used to control the step size, and simultaneously another termination condition is proposed. To make sure a very small second derivative or a large gradient do not lead to adverse large steps, the initial λ is selected between 0 and 1. The average values of the first derivatives are watched over every few steps to make sure they are decreasing. This indicates the algorithm is converging to an optimum. Whenever it is diverging in direction i , the λ_i is slightly decreased to make the step size in dimension i smaller than before. Whenever it is converging the λ_i is slightly increased. There is an upper bound to this ($\text{MAX}(\lambda_i) = 0.5$). When $\lambda_i < \varepsilon$ then $\lambda_i = 0$ and the optimization is terminated for i -th dimension. The optimization is terminated prematurely at iteration 1000 to avoid infinite loop. In most cases the model is optimized in lower than 600 iterations. Therefore, we do not allow termination before 600, only to make sure during a number of the final iterations the zero-projection from Section 4.5.1 is applied every time.

Moreover, the Hessian must be always positive for a convex function otherwise there is no local minimum or the optimization is not converging to a minimum. Since a negative Hessian is not possible for the cost functions in this thesis, it can only happen due to numerical inconsistencies. Additionally, in the update function (4.18) the value of $(H^k)^{-1}\nabla^k\vec{e}$ is pre-calculated and clipped in an interval $[-r, r]$. In the current implementation $r = 1$. This works as a double check that the Hessian-inverse does not include implausibly large values.

4.7 Alternative Methods

For the proposed optimization problem, a number of other approaches are experimented with, e.g. adding prior terms for promotion of gray light and barrier functions to force nonnegativity. Here, two alternative optimization approaches are explained and their performance is compared to the proposed one.

4.7.1 Nonlinear Model $\vec{x} = e^{\vec{\alpha}}$

The equation (4.1) fails to address the nonnegativity of light sources implicitly. An implicit way to handle this is to remove the coefficient vector with a nonnegative function $f : \mathfrak{R}^n \rightarrow (\mathfrak{R}^+)^n$, such as probability distribution functions, logistic, softmax, exponential or a simple squared function. Here, the case with $\vec{x} = e^{\vec{\alpha}}$ is discussed. With this substitution we have (4.22) instead of (4.1).

$$\mathbf{I} = \mathbf{C}e^{\vec{\alpha}} \quad (4.22)$$

where $e^{\vec{\alpha}}$ is a vector of n RGB values $\forall i \in [1..n]$; $\mathbf{x}_i = e^{\alpha_i} = (e^{\alpha_i^r}, e^{\alpha_i^g}, e^{\alpha_i^b})$.

The exponential function is monotonically increasing and bijective from its domain –real numbers– to its range –positive real numbers. Moreover, it is differentiable and invertible. These features allow us to perform gradient based optimization and return each final value α to a single x and vice versa ($x = e^\alpha \rightarrow \alpha = \ln(x)$). This is necessary for further regularization functions which are interpretable in both x and α domains, such as in (4.23).

$$e(\vec{\alpha}) = \|\mathbf{C}e^{\vec{\alpha}} - \mathbf{I}\|_1 + \|\vec{\alpha} - \vec{\bar{\alpha}}\|_2^2 \quad (4.23)$$

Here, the likelihood term is from the logarithm of a Laplacian distribution. The coefficients \mathbf{x}_i are naturally nonnegative and the regularization term is based on a Gaussian prior on the α_i -s, where $\alpha_i = \log(x_i)$. In other words, minimizing the cost function (4.23) is the same as maximizing a posterior distribution that is the result of multiplying a Laplacian likelihood by a normal prior on α_i -s. This is basically an alternative implementation for a normal likelihood times a lognormal prior on \mathbf{x}_i s. The lognormal distribution (4.24)

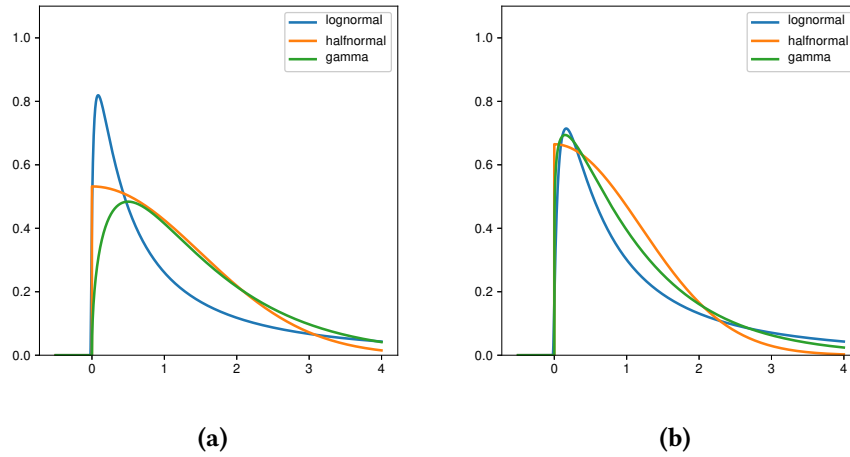


Figure 4.5: The lognormal, halfnormal and gamma distributions with two different shape parameter settings to make them look more different Figure 4.5a or similar Figure 4.5b.

implies that the logarithm of light source intensities in each color channel follows normal distribution.

$$P_{\text{lognormal}}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-0.5\frac{(\log(x)-\log(\bar{x}))^2}{\sigma^2}} \quad (4.24)$$

This cost function (4.23) is minimized with gradient descent (without Hessian). Also here, the limitation on the update term (see Section 4.6) in an interval $[-r, r]$ proves to be necessary for the convergence.

The iterations get sluggish when the coefficients approach zero, therefore, it is hard to reproduce cast shadows with this approach. Because of this limitation and the time consumption of the exponential functions, this approach cannot compete with the proposed optimization. In Figure 4.5, halfnormal (proposed), lognormal and gamma distributions are shown together in two different configurations. It is clear that only the halfnormal distribution allows all the lower intensities (close to zero) and promotes zero values as well, whereas gamma and lognormal distributions suppress zero and low intensities. Therefore, the halfnormal prior is proposed for the proposed inverse lighting in this thesis in both MAP and JMAP approaches.

The results in Figure 4.6 (bottom row) show that this alternative approach does not perform well on harsh illumination, therefore, it is not proposed for such images. However, it naturally converges to a nonnegative solution and an approximation of the lighting which might still be better than other approaches of previous work. Moreover, this method is explained to point out the limitation of the proposed generative model (4.2) in modeling nonnegativity and to show the need for a more suitable generative model for the superposition principle by avoiding subtractions, which are inevitable when negative coefficients



Figure 4.6: Results for optimization with $\vec{x} = e^{\vec{\alpha}}$ coefficients compared with the proposed JMAP results. The EXP results look incorrect when compared with the JMAP row, however, they still provide an approximation of the input lighting.

are allowed.

4.7.2 Richardson-Lucy (RL)

Another optimization algorithm of the Bayesian family is the Richardson-Lucy deconvolution by Richardson [Ric72] and Lucy [Luc74]. It is used for denoising and deblurring images when the Point Spread Function (PSF) is known. Recently, Ingaramo et al. apply RL to find the best merging of differently blurred images in multiview fluorescence microscopy [IYH⁺14], while simple averaging of the views produces an even worse image. The RL iterations converge toward the maximum likelihood of the unknown parameters \vec{x} , which are assumed to have a Poisson distribution. Fish et al. explore the performance of RL for blind deconvolution on noisy input [FWBP95]. White explains the RL update function (4.25), yet, it provides a modified version of it that reduces noise amplification in the final result [Whi94].

Similar to [IYH⁺14], we want to estimate a signal I as a weighted summation of other signals C . We assume that each light source follows Poisson distribution in each color

channel separately. We use the Richardson-Lucy update function (4.25) to reconstruct the image \mathbf{I} .

$$x_i^{t+1} = \frac{x_i^t}{\sum_{p \in \text{face}} C_i(p)} \sum_{p \in \text{face}} \left(\frac{\mathbf{I}(p)}{\left[\sum_{j=1}^n C_j x_j^t \right] (p)} C_i(p) \right) \quad (4.25)$$

The notations are the same as previous in this thesis. The $\left[\sum_{j=1}^n C_j x_j^t \right]$ is an image of the same dimensions as \mathbf{I} , result of a pixel-wise addition of the images C_j with coefficients x_j for $j \in [1..n]$ from iteration t . Then the notation $[...] (p)$ in the denominator addresses the pixel p of that image. All the multiplication and divisions are per pixel. Moreover, the color correction is removed from \mathbf{I} by subtracting the offset from the image and then multiplying the result for each pixel by inverse of the T matrix (see Section 3.2.3) before the optimization starts. The x_j -s are set to zero if they are below a threshold. Whenever $x_i^t = 0$, the iteration can be discarded for the i -th dimension, which increases the performance. Furthermore, the denominator must be watched for division by zero error.

The algorithm has no tuning parameters other than the initial parameters x_j^0 ; when initialized by a nonnegative value, the result is also guaranteed to be nonnegative. This is a major motivation of using RL for the inverse lighting problem. Otherwise the algorithm is relatively time consuming and its results on complex lighting condition (especially with dominant cast shadows) ripen only after too many iterations. Any attempt to make the RL algorithm work stably with superpixels (see Section 3.4.2) was rather unsuccessful. The MISIM measure is generally slightly lower for the results of RL compared to the proposed (JMAP) method. In conclusion, JMAP still delivers better results much earlier than RL although the RL results might look aesthetically better due to their smoothness. However, aesthetically better does not equal to closer to reality. Notice the better estimation of the attached and cast shadows in JMAP result. A comparison between the JMAP and RL is provided in Figure 4.7.

Because of the smooth convergence of RL from a uniform illumination to a fair estimation of the complex lighting (the algorithm does not jump from a lighting configuration to another), it can be used to make animations of gradual formation of a lighting by producing a frame per iteration.

4.7.3 Full Hessian Inverse

For both the MAP and JMAP approaches, the update function of Newton with the full Hessian with respect to parameter \vec{x} has been tested. The SVD pseudo-inverse from Numerical Recipes (NR) [PTVF96] and OpenCV [B⁺00] are compared. Even after numeric improvements of the input and sanity checks on the output, which proved to be necessary at least for NR implementation of SVD, it was impossible to get any promising results



Figure 4.7: Results for optimization with Richardson-Lucy (RL) update formula. Initialized with $x_i^0 = 0.1$, terminated at 500th iteration. In the middle row the results with JMAP approach are provided for comparison. The RL results look smoother than the JMAP results. Also, notice the smoother cast shadow of the nose in RL result in the leftmost column.

with the full Hessian. The NR implementation pseudo-inverse works properly on values that are in the $[-1, 1]$ neighborhood, while the OpenCV implementation is robust against numerical issues as such. The full Hessian is an overkill for the inverse lighting algorithm and makes the program more time and memory consuming, and as a result impossible to tune to get an acceptable outcome. In this thesis, the simple pseudo-Hessian leads to the provided results in Chapter 5.

Chapter 5

Evaluation and Results

5.1 Mean Illumination SIMilarity (MISIM)

There is a lack of quantitative measures for the objective assessment of the inverse lighting results. It is evident that a pixel-wise difference between two images does not give any meaningful information about the illumination. A standard qualitative method needs quality measures and a scale to compare illuminations and put the results in an order of best to worst in a meaningful way. Furthermore, it has to decide about the importance of the illumination features, e.g. intensity and color of highlights on different areas of the face, softness and hardness of the shadow edges, cast shadow intensity and shape of these illumination effects on the given geometry. In addition, it has to decide between the improvements (or errors) that are caused by the estimated illumination and other estimated parameters, e.g. shape, reflectance or color correction. Obviously, we can't hold the inverse lighting accountable for the achievements of the 3DMM fitting. Designing such a standard qualitative assessment method is out of the scope of this thesis. Nonetheless, this section proposes a novel method that uses the wide spread MSSIM (Mean Structural SIMilarity) by Wang et al. [WBSS04] to measure the similarity between the input image and the reconstruction with respect to the estimated illumination. The MSSIM is a symmetric measure for comparison of two images of the same scene based on their structural similarity. It intends to provide a quantitative evaluation that is closer to human perception, compared to MSE (Mean Squared Error) and SNR (Signal to Noise Ratio). However, the MSSIM measure between the result image I_{VLS} and the input image I is a very high value (usually $> 90\%$, and rarely below 95%). We cannot take this as a measure for the performance of the proposed lighting estimation because most of the similarity between the input and the rendered image is achieved by the 3DMM fitting that estimates the 3D face model, color correction and dense correspondence. Even rendering the estimated face geometry with the average face texture and a simple ambient lighting leads to an already high MSSIM score. Therefore, the MISIM (Mean Illumination SIMilarity) is proposed that

measures the similarity that is achieved exclusively through illumination estimation.

To calculate MISIM, the input image \mathbf{I} , the optimal color corrected diffuse ambient image \mathbf{I}_{amb} from (3.7), the result image \mathbf{I}_{VLS} and a face mask image \mathbf{I}_{face} are provided. All pixel values are between $[0, 1]$ in RGB channels. The face mask is a model-based binary mask that marks the face pixels. It is automatically generated for each input image after 3DMM fitting (see Section 3.2.4). Using the face mask \mathbf{I}_{face} , we calculate the MISIM only for the face pixels as below:

1. Calculate MSSIM between the image \mathbf{I}_{amb} and the input image \mathbf{I} .
2. Calculate MSSIM between the result image \mathbf{I}_{VLS} and the input image \mathbf{I} .
3. Calculate MISIM from (5.1).

$$\text{MISIM}(\mathbf{I}, \mathbf{I}_{VLS}) = \frac{\text{MSSIM}(\mathbf{I}, \mathbf{I}_{VLS}) - \text{MSSIM}(\mathbf{I}, \mathbf{I}_{amb})}{1 - \text{MSSIM}(\mathbf{I}, \mathbf{I}_{amb})} \quad (5.1)$$

$$\text{MSIER}(\mathbf{I}, \mathbf{I}_{VLS}) = \frac{\text{MSE}(\mathbf{I}, \mathbf{I}_{amb}) - \text{MSE}(\mathbf{I}, \mathbf{I}_{VLS})}{1 - \text{MSE}(\mathbf{I}, \mathbf{I}_{amb})} \quad (5.2)$$

The proposed MISIM calculates the contribution of the estimated illumination to the achieved similarity between the reconstruction and the input in terms of MSSIM and with respect to the diffuse ambient reconstruction. With the same strategy the MSIER (Mean Squared Illumination Error Reduction) is calculated according to (5.2) to measure how much the Mean Squared Error (MSE) is reduced under the estimated illumination. We calculate the mean value through all the pixels and channels. The Figure 5.1 shows the MISIM and MSIER for the examples in Figure 5.7. Especially when the MSIER and MISIM measures are not similar for one image (e.g. image A), we see that the MISIM gives an evaluation which is closer to human perception. In image A, the improvement that is achieved through the proposed algorithm is rather negligible because the original lighting is a simple frontal lighting, however, the MSIER measure conveys that the result is 68.72% better than the diffuse ambient image. Looking at the results, it is easier to agree with the more humble value of 32.51% that the MISIM measure suggests for the image A.

Alternatively, it is possible to provide a more illumination-oriented structural similarity measure by using the proposed geometric superpixels (Section 3.4.2) instead of SSIM's original Gaussian filter. Then, the expected value μ in [WBSS04] can be provided individually for each superpixel. However, such improvements depend on the implementation of the proposed superpixel, which itself depends on the estimated normals by 3DMM fitting. Instead, we use the well established MSSIM as a black box to make it easily reproducible for other researchers. The blind mask is needed only at the end when the mean of the SSIM map is being calculated otherwise the implementation is exactly according to [WBSS04].



















	A	B	C	D	E	F	G	H	I
Input									
VLSavg									
MISIM	32.51%	84.79%	71.77%	65.01%	60.07%	78.57%	82.24%	69.32%	65.44%
MSIER	68.72%	83.28%	68.41%	65.53%	58.71%	75.04%	78.97%	67.37%	58.14%

Figure 5.1: MISIM and MSIER for the input images (first row), shown with I in the equations, and the estimated result (second row) with the VLS approach (JMAP), shown with I_{VLS} in the equations, for the examples of Figure 5.7. These numbers show the average improvement of 67.75% in terms of MISIM and an average improvement of 69.35% in MISIER terms. Larger values are better. Although the averages are very similar for both measures, where the lighting is not that harsh, e.g. column A, we see that the value of MISIM is more meaningful. For simple lightings, a diffuse ambient lighting is already close enough to the input image, thus, the proposed algorithm, (or any other algorithm) cannot make a large difference in the estimation of the global lighting. Moreover, the MSSIM measure is for all the samples larger than 90% and rarely below 95%, meaning that the images are very similar. Therefore, MSSIM conveys no valuable information for the assesment of the performance of the inverse lighting approach and is not used in this thesis.

5.2 Results

The best way to evaluate the performance of an illumination estimation algorithm is still through educated subjective judgment. In absence of a formal standard for evaluation of the algorithm, we show how the algorithm performs on a variety of illumination complexities with evidential experiments. This chapter shows the pros and cons of the proposed methods by showing the intermediate and final results. Firstly, the results of the MAP and then those of the JMAP approach are shown. Then, the results of the proposed relighting and lighting design methods are shown. The results of alternative approaches are given in the respective section in previous chapters, therefore, we only show the results of the proposed methods and compare with the results from 3DMM [BV99, BV03]. In the case of synthetic input, the proposed method delivers almost zero error with no visible difference between synthetic input and the reconstructed image. Even the light direction and colors are estimated accurately which lead to accurate intensity and contours of the cast shadow in the reconstruction image (Figure 4.1).

For a critical judgment, pay special attention to the following features in the input and rendered images although we can not say which features are more important in this list:

- Occurrence of colorful lighting and highlights.
- Highlights in grazing angles.
- Color bleeding from brighter surrounding objects, e.g. white collars that reflect under the jaw.
- Intensity and shininess of the specular highlights.
- Complex illumination effects by multiple light sources, such as the somewhat brighter triangle beside the nose in Rembrandt lighting in the last four examples of Figure 5.7.
- Occurrence of cast shadows, especially around the nose.
- Intensity and form of the cast shadows.
- Artifacts on the estimated intrinsic texture.

To appreciate the performance of the proposed algorithm, pay attention to the extent of the improvements compared to previous work –the 3DMM results. As mentioned in Chapter 2, there is no other previous work that can fairly be compared to the proposed method because none of them provides results of their lighting estimation on harsh illumination conditions. This thesis may inspire future work on physically plausible and harsh illumination estimation in lowly constrained scenarios.

5.2.1 Results of MAP

The MAP approach relies more on the input data and demands more user interaction –a manually drawn occlusion mask– when the data is somehow unreliable. Whenever user interaction is desired or unavoidable in the application, such as for lighting design, the MAP approach can be considered. Especially, as seen in this section and in Section 5.3.1, the results of MAP are promising and the difference between the MAP and JMAP results is not very large. The Figure 5.2, Figure 5.3 and Figure 5.4 are from [SB15b], where the MAP approach with $n = 100$ and Gaussian blur is applied. The caption of Figure 5.2 explains all the images in these three figures. With the exception of this section and the lighting design results in Section 5.3.1, the rest of the results in this chapter are generated with the JMAP approach on superpixel representation according to [SB17].

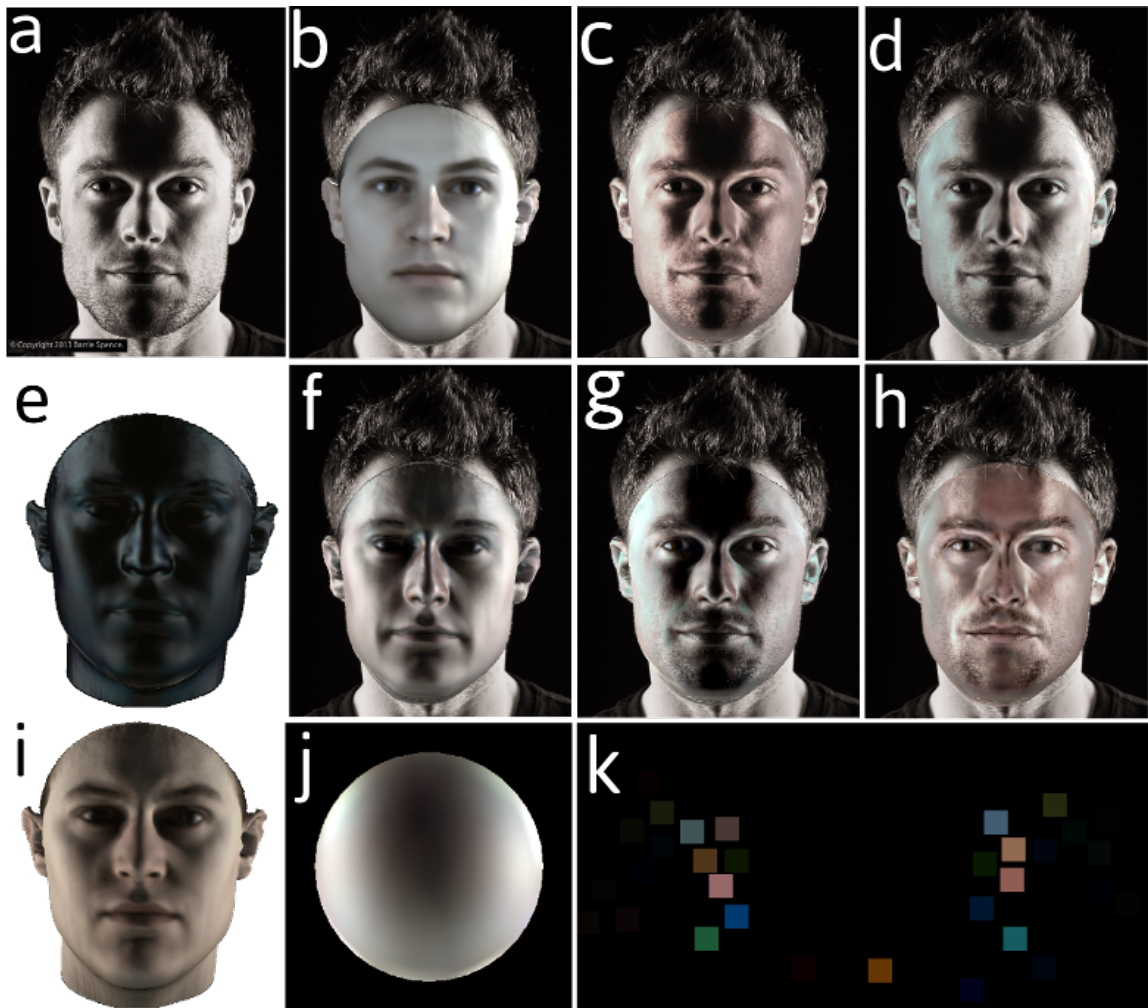


Figure 5.2: Results are generated with the MAP approach from [SB15b]. a) Original image (This picture is Copyrighted by Barrie Spence 2013). Images b, c and d are results generated with the 3DMM framework. Images e-k are results of the proposed algorithm. b) 3DMM full reconstruction rendered with *average texture* to show the lighting estimation. c) 3DMM result with image-based texture values using ambient and a directional light with Phong model. d) 3DMM result for image-based textures with uniform diffuse lighting to show the intrinsic texture. e) Specular map result of proposed algorithm to show the specular shading on the geometry. f) Result of estimated illumination rendered with average texture compared to b. g) Result of proposed rendered with image-based texture compared to c. h) Result of proposed image-based texture rendered with uniform diffuse lighting, compared to d. i) The diffuse map, result of proposed method to show the diffuse shading when applied on average texture. j) A sphere rendered with average skin BRDF and estimated light sources from proposed method. k) Light source distribution which shows the direction and color of estimated light sources around the face. The orientation in spherical coordinates is explained in Chapter 3 (see the caption of Figure 3.7). In this example, the improvements are vast. The colorfulness of the light sources in k plays almost no role in the cost function because the low color contrast of the input image provides little amount information about the used light colors. In the results, the saturation is correctly reduced by the color balance term.

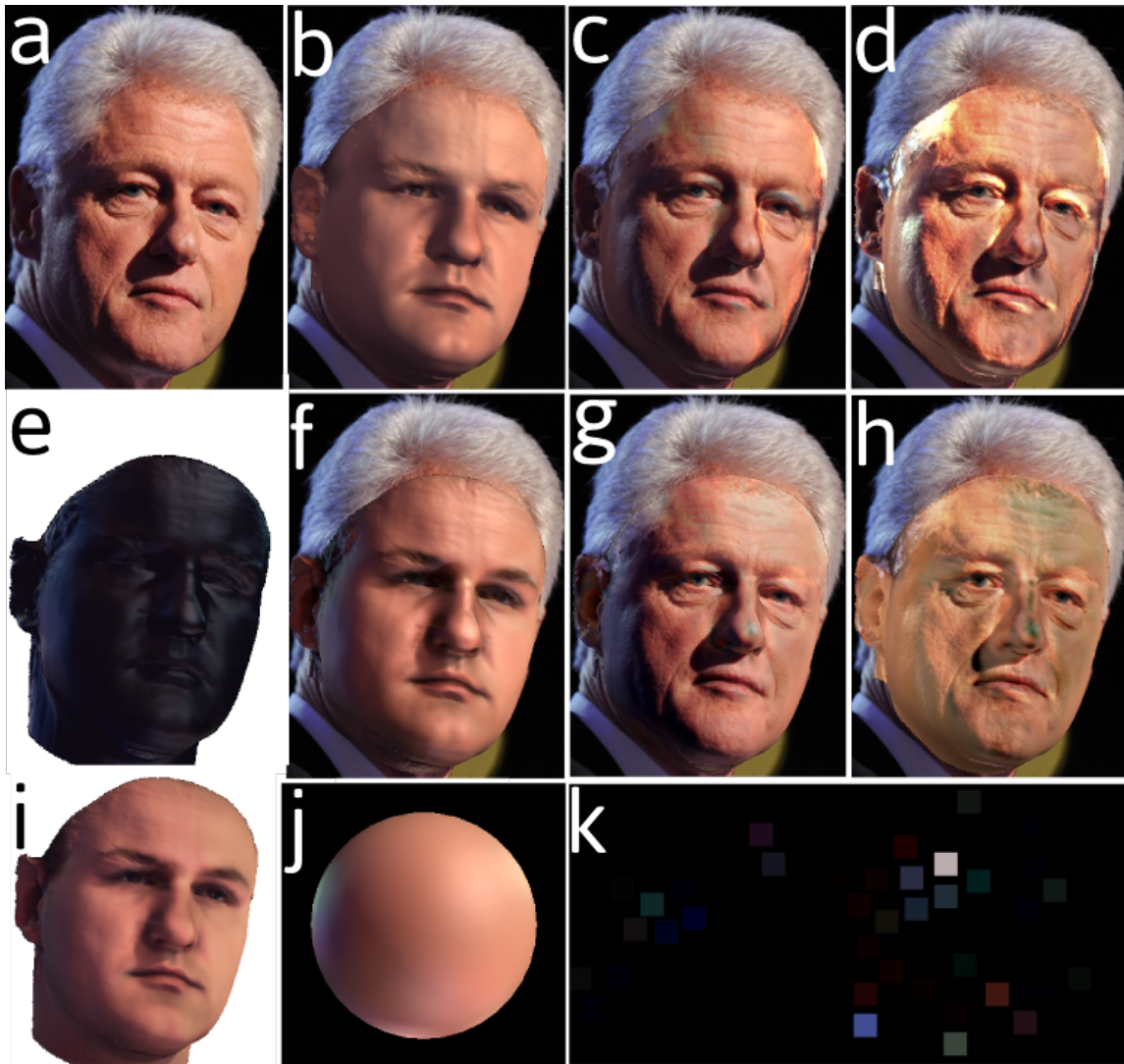


Figure 5.3: Results are generated with the MAP approach from [SB15b]. For description of labels see caption of Figure 5.2. Estimation of cast shadows of micro structures, e.g. deeper wrinkles, are not achievable because the respective geometry is not estimated. The cast shadow of the nose is estimated, as visible in f and i. However, it is too weak to remove the shadow from intrinsic texture in image h. Images f, g and h show visible improvements compared to b, c and d. For instance note the appearance of the bluish highlights on the left side of the face in f. This highlight is visible in the rendered sphere under the estimated lighting, shown in image j.

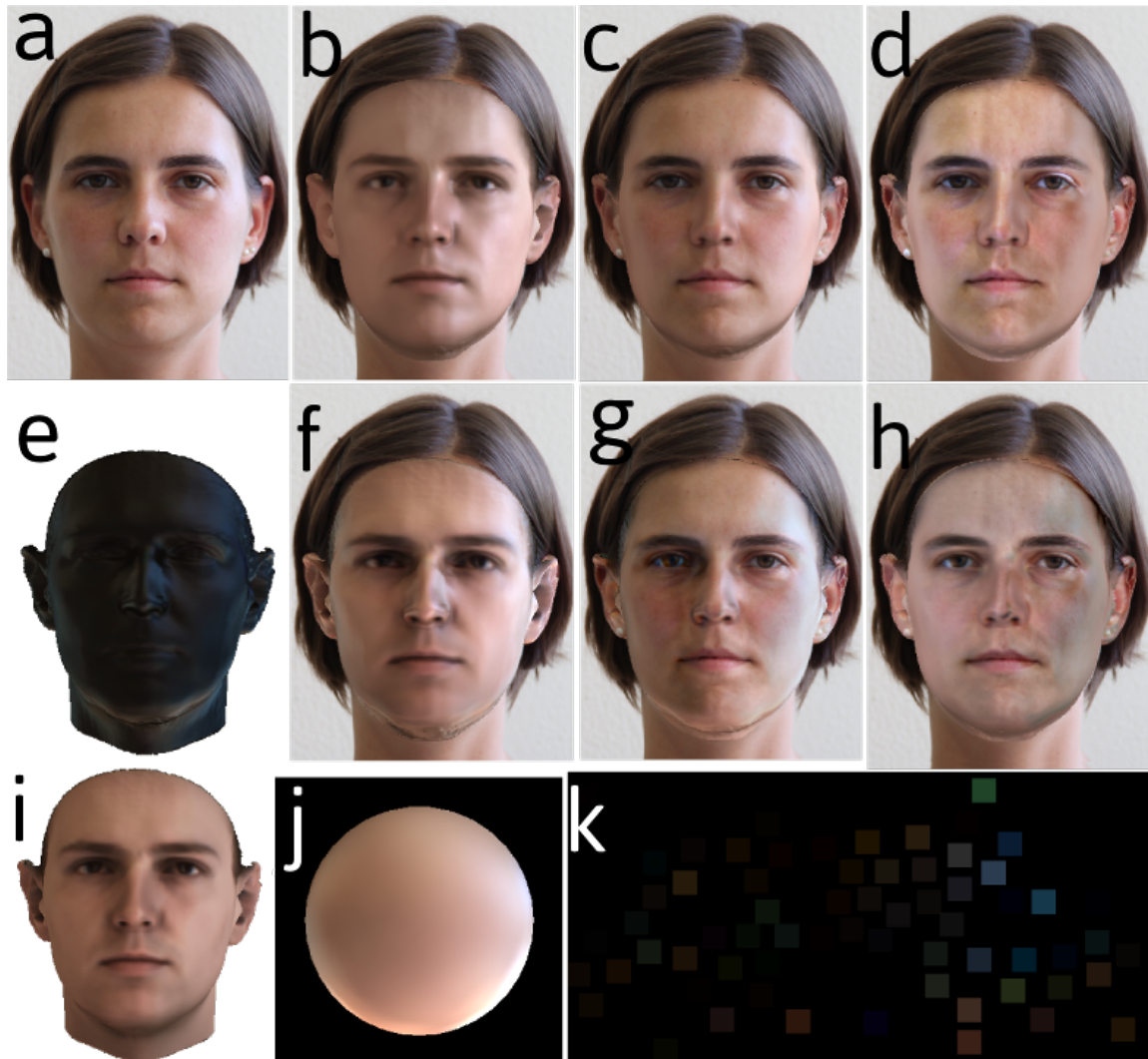


Figure 5.4: Results are generated with the MAP approach from [SB15b]. For description of labels see caption of Figure 5.2. This input image has a high frequency low intensity illuminated area on the left side of the face, which the 3DMM results in the top row do not reproduce; not even as part of the texture in image c. Yet, the proposed method estimates even the subtle lighting effects such as this one and leads to general improvement of the appearance of the rendered image. Image h shows a more "illumination-free" intrinsic skin texture than image d. Images e and i show that the highlights on the left side of the face are more of specular nature than diffuse.

5.2.2 Ambiguity of the Estimated Lighting

Here, we estimate the lighting for a number of images from CMU PIE database [SBB02]. The selected images are taken under all the same conditions, including similar lighting, from different subjects. The proposed inverse lighting is performed on each image separately. To show all the estimated illuminations on a single reference geometry, for each image a sphere with average BRDF of skin is rendered under the estimated VLS Figure 5.5. This is a representative set of the results for direct comparison. In these images, you can see that the color of lighting is affected by the difference between the color of the average texture and the skin type. One reason for this is the focus of 3DMM on a limited variety of skin types, therefore, the average texture is biased. This figure shows that the coarse directions and intensities of lights are consistent among images which are taken under the same lighting condition. Moreover, the color correction term takes care of the missing lighting color and replicates the color contrast and saturation of the input image in the respective rendered spheres (third row).

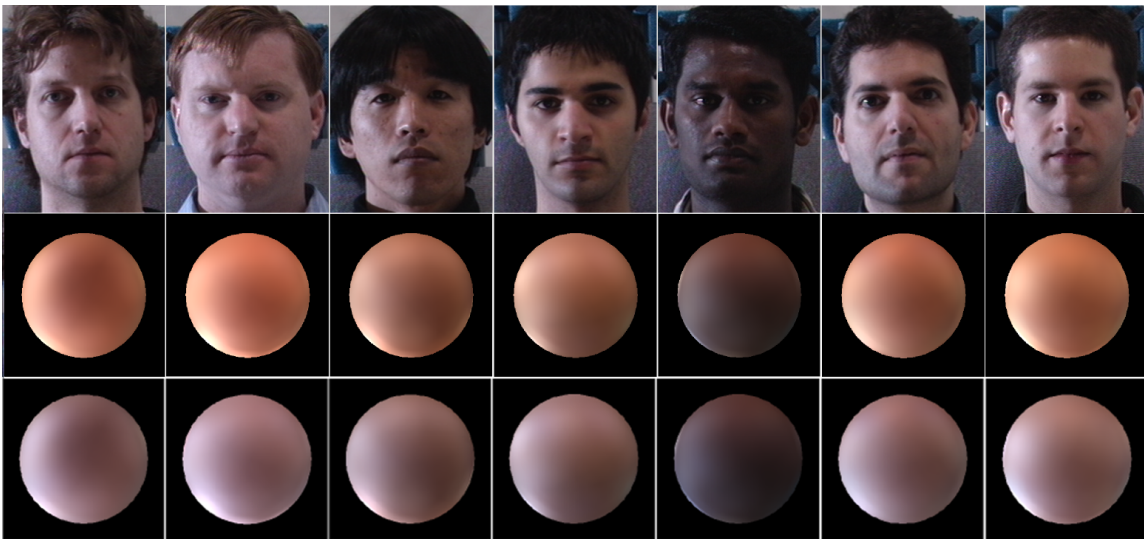


Figure 5.5: First row shows the input images of different subjects captured under a common illumination from PIE database [SBB02]. The second row shows the spheres with the average skin BRDF and a generic color correction, rendered under the estimated VLS from the respective input images above them. The estimated color correction for each input image is used in rendering of the spheres in the third row. Note that some spectral properties of the skin are wrongly attributed to the lighting by the proposed algorithm.

5.2.3 The JMAP Results

The results of JMAP approach are presented on one page in Figure 5.7. For each “Input,” the following results are shown: “3DMMavg” is the result of the 3DMM fitting [BV99] rendered with average texture and the built-in lighting estimation of the 3DMM to be

compared with the “VLSavg” result which is the same model rendered with the estimated lighting from the proposed method. In contrast with the 3DMM results in Section 5.2.1, in this section we set the cast shadow flag so that the 3DMM considers the cast shadows in its inverse rendering –fitting– and forward rendering steps. Compared with Figure 5.2, 5.3 and 5.4, images (b), here not the estimated texture but the average texture is used to make the comparison easier. The row “ $n = 300$ ” shows the result when the VLS consists of 300 light sources instead of 100. See also Section 5.2.4). “VLS” is the label of the estimated Virtual light Stage in rectangular representation (see Figure 3.7). “3DMMalb” is the intrinsic texture (albedo) that is extracted from the input image with the built-in lighting estimation of the 3DMM, which is to be compared with the result from the proposed “VLSalb.” This intrinsic texture is rendered on the 3D face geometry in “De-Illum.” The row “ β_{opt} ” shows the optimized hyperparameter $\vec{\beta}$ on geometric superpixels which are mapped to the image plane. We also show the illumination decomposition D and S , labeled with “Diffuse” and “Specular” respectively. The last row shows a high quality full reconstruction “Full Recons.” with the estimated VLS and extracted intrinsic texture.

The results of the proposed algorithm show considerable improvements compared to the previous work [BV99], especially whenever lighting conditions are too complex to be estimated by Phong model and a directional light. The improvement is even more obvious when the “VLSalb” is compared to the “3DMMalb”. Table 5.1 provides the MISIM and MSIER measures, proposed in Section 5.1, for the images of the “VLSavg” row in Figure 5.7. The estimated illumination introduces up to $\approx 85\%$ more similarity to the input image when the “VLSavg” is compared with an image of the same model with an optimal ambient illumination (no directional lights).

In the proposed approach, the light color and intensities for the directional light sources – the VLS– are estimated, however, the directions of these light sources are fixed. The exact real light sources are out of access and cannot be inferred from the given data because of two reasons. Firstly, the manipulated saturation and brightness of the input image scramble the intensity and spectral properties of the reflected light. Secondly, the specular term of the skin reflectance is not shiny enough, compared to a mirrorlike surface or a black snooker ball. For shiny surfaces, small displacements in the position of the main light cause visible changes in their appearance. Therefore, instead of trying to estimate the exact light sources, we estimate a physically plausible model for the lighting of the given face image. In Section 5.2.2, we investigate the ambiguity of the lighting estimation in depth. In Figure 5.7, you see the result of estimated VLS for each input image (row labeled with “VLS”). The black rectangles are flattened spheres, representing the VLS. Each light source is represented with a spot in the estimated color and intensity of the light source and in its corresponding position. Frontal light sources are in the middle of the rectangle, light sources from up and down are represented close to the upper and lower edges of the

rectangle and light sources from the back of the face are close to the middle of the left and right edges of the rectangle (see Figure 3.7). In the case of smooth illuminations, because a higher number of light sources is active, their individual intensities are very low, e.g. Figure 5.7 A and B. In case of multi-directional lighting situations, the light sources group together in spatially different positions on the rectangle, e.g. Figure 5.7 C – I. When distinct light sources exist in the scene, e.g. sun in Figure 5.7 E, the VLS map indicates a bright light in an approximately expected direction.

5.2.4 Experiments with Different Number of Light Sources n

We achieve slightly better results by increasing n , the dimension of the VLS, from 100 to 300. Some specular areas or cast shadows are more precise in the case of $n = 300$ compared to $n = 100$. In Figure 5.7 compare the “VLSavg” row for $n = 100$ to the “ $n = 300$ ” row for D and E columns. Also, a diffuse lighting, such as column A, is easier to estimate with $n = 300$. In spite of the time and storage costs that are imposed by a 300-dimensional VLS, the improvements are unstable and negligible in most cases. A VLS with $n = 100$ proves to be a good compromise. This is a common conclusion in different experiments [SB15b, HCSBL15, SB17].

The maximum number of light sources that might still make sense to use in inverse rendering depends on the number of distinct surface normals of the object of interest, for a highly specular surface and point lights. This can lead to a very large number. In contrast to specular surfaces that reflect a point light as a small spot, Lambertian surfaces diffuse the incident light. Therefore, a large number of the light sources hardly leads to significant improvements in the performance of the algorithm for such surfaces. Debevec shows that in practice the necessary number of light sources is rather limited [Deb06].

5.2.5 Experiments with Different Number of Superpixels m_S

The proposed segmentation method reduces the amount of data from $m \times (n + 1)$ to $m_S \times (n + 1)$, where n number of images (columns) in the gallery \mathcal{C} , m the number of pixels in each image, and m_S is the number of combined superpixels (number of rows in matrix \mathcal{A}). The value of m_S does not depend on the number m and $m_S \ll m$. In other words, the image size has no effect on the memory consumption, computational complexity and runtime of the optimization (Section 4.2). Figure 5.6 shows the quality of the lighting estimation for approximately 300, 20 and 10 geometric superpixels. You see that for easier illumination conditions a few superpixels are enough, however, for more complicated lighting conditions the quality of the result increases with the number of superpixels. In this thesis $m_S = 300$ is proposed.

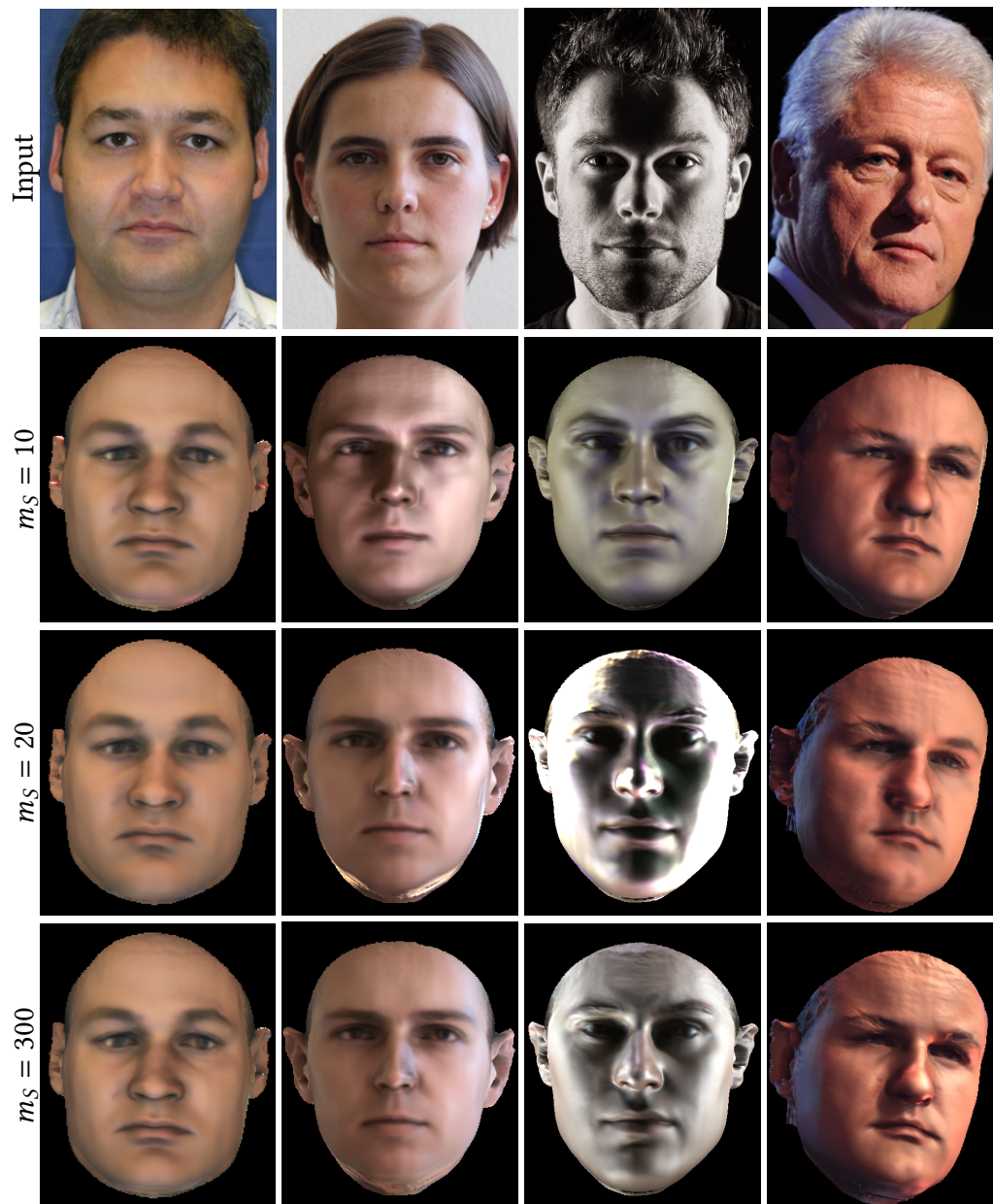


Figure 5.6: Results for different number of superpixels. First row shows the input images, second row is the corresponding results with 10 geometric superpixels, third row shows the results for 20 superpixels and last row are corresponding results, when the faces are segmented in only 300 superpixels. Spatial complexity of the lighting of input images rise from left to right.

Table 5.1: MISIM and MSIER for examples from Figure 5.7. Larger values are better. This table repeats the values of Figure 5.1 to show them close to the Figure 5.7 for convenience reasons.

Ex.	MISIM	MSIER
A	32.51%	68.72%
B	84.79%	83.28%
C	71.77%	68.41%
D	65.01%	65.53%
E	60.07%	58.71%
F	78.57%	75.04%
G	82.24%	78.97%
H	69.32%	67.37%
I	65.44%	58.14%
avg.	67.75%	69.35%

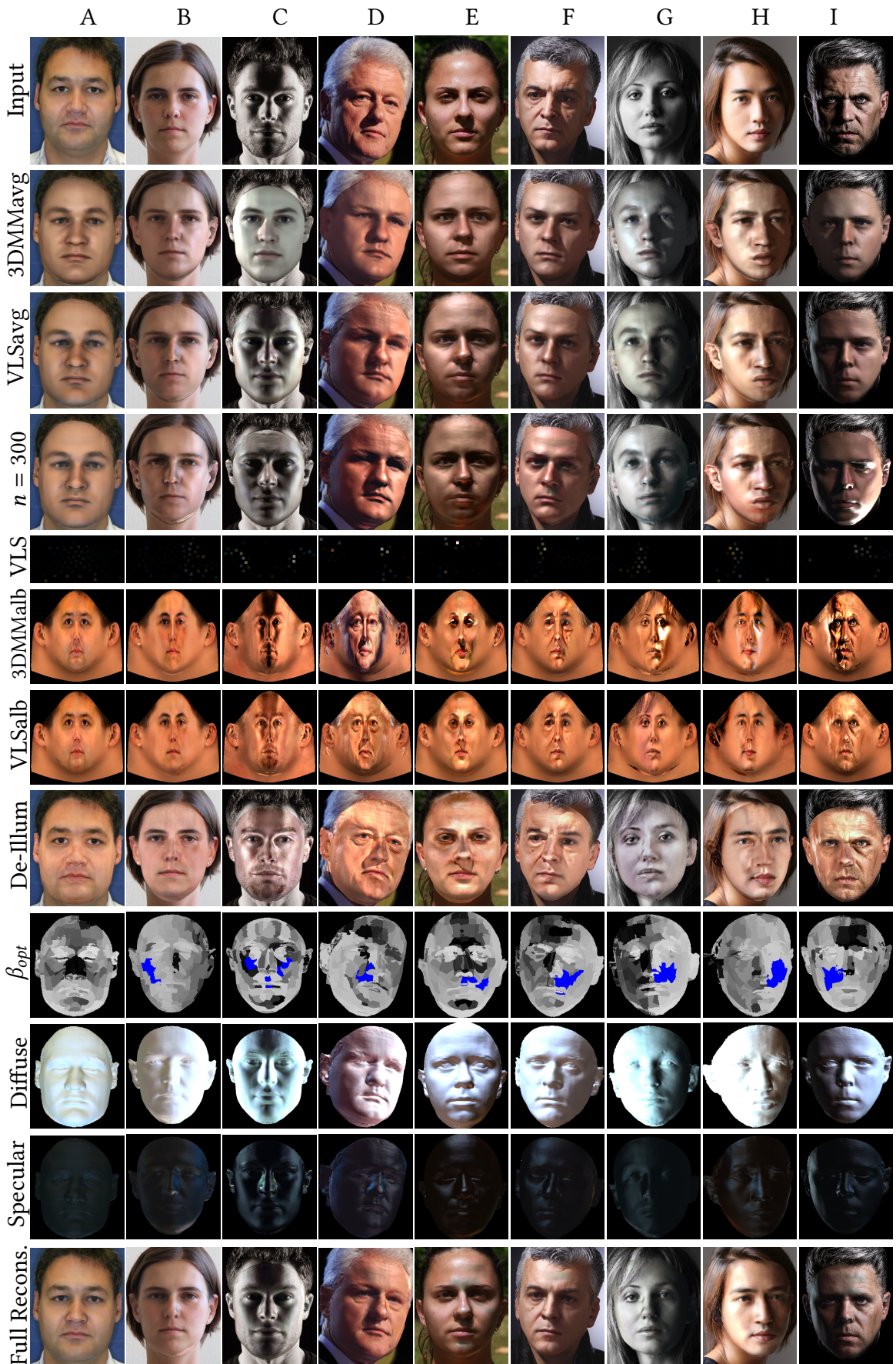


Figure 5.7: Results with the JMAP approach $n = 100$, in comparison with the inverse lighting of 3DMM. One row with $n = 300$ shows that larger n might deliver better cast shadow estimation but the stability suffers. See Section 5.2.3.

5.2.6 Cast Shadows

Figure 5.8 shows the improvement of cast shadow estimation from almost no cast shadows in the “3DMM” results to the state of the art with the “JMAP” approach. Shadow intensities are more accurate and the shape of the shadow is improved in JMAP results. We achieve negligibly better results with more light sources (experimented with 100, 300 and 1000 light sources), yet, the improvement is not very significant with respect to the increased cost. The error in the shape of the cast shadow is a problem that has three causes. Firstly, the errors in the estimated face geometry and correspondence do not allow a realistic cast shadow to be formed. This is a chicken and egg problem. If the reconstructed shadow is not long enough, is the shadow casting geometry (e.g. nose) estimated too short or the geometry that the shadow is casted upon (the cheek) is too high? Secondly, the VLS is only a discrete representation of an environment map and might miss light sources exactly in the direction that is necessary for the reconstruction of a given shadow. For an example, see the slightly improved cast shadows in Figure 5.7 in the $n = 300$ row. Thirdly, the noncontinuous relationship of the cost function, which is defined on the pixel intensities in RGB space (Chapter 4) with the cast shadows, which are nonlocal geometric effects, prevents the algorithm from considering the error in the shape of the cast shadow. This might open another topic for the future work. Pushing the algorithm too much towards estimating a cast shadow that can not be rendered due to the mentioned limitations leads to overfitting.

Cast shadow of the nose is an important part of the well-known Rembrandt lighting in portrait art. With JMAP approach, it is possible to estimate the lighting and illumination of such portraits more accurately. In Figure 5.7. F – I, you see successful estimation of the Rembrandt lighting, including the bright area side, the attached shadow side and the

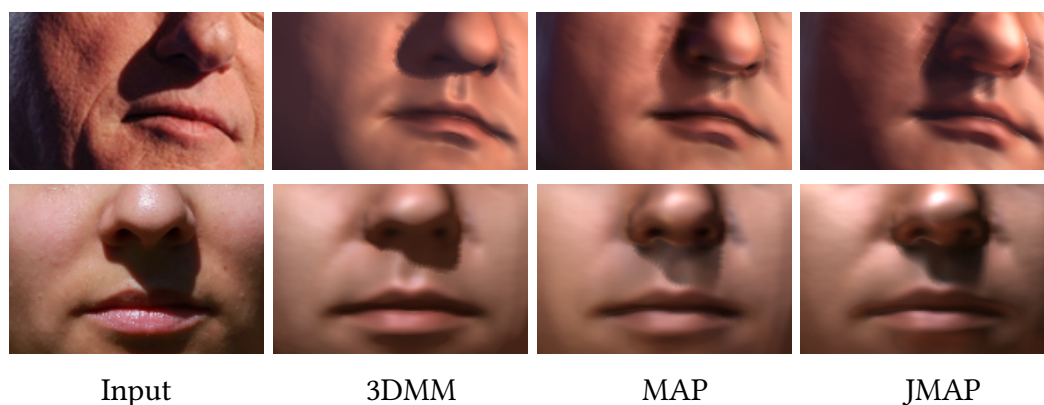


Figure 5.8: This figure shows the improvement of the cast shadow reconstruction on two input images in separate rows (See also Figure 5.7 D and E). The column “Input” is the input images which contain obvious cast shadows of the nose. In the “3DMM” column, the lighting is estimated by 3DMM fitting. The columns “MAP” and “JMAP” show the cast shadows estimated by MAP and JMAP approaches respectively.

cast shadow of the nose. Moreover, in some examples (H and I) a rim-light appears on the attached shadow side, which has been estimated accordingly.

5.3 Relighting Results

Single image relighting is impaired by the lack of significant information about the object and its environment. A single shot hardly provides all the necessary information about the reflectance of the skin [INN07]. Moreover, in an uncalibrated image from an unknown source, Lighting effects, e.g shadows, hid the reflectance features of the face. Considering that, the proposed algorithm delivers promising relighting results. We estimate the illumination from one face and apply it to the intrinsic face model estimated from another input image. For each transfer, the inverse lighting algorithm is called twice, once for the estimation of the target lighting and once for the estimation of the target intrinsic texture. Some examples for lighting transfer are presented in Figure 5.9. The images on the left column provide the target lighting. Their lightings are swapped on the right column. In the middle columns, two different target faces are rendered. We see that under the same pose and lighting similar effects appear on the face. This tool can be used to add faces to images which is an interesting application for arts and image forgery detection studies. More results of relighting are shown in the next section where the inverse lighting algorithm is used to make a software tool for post hoc creative lighting design.

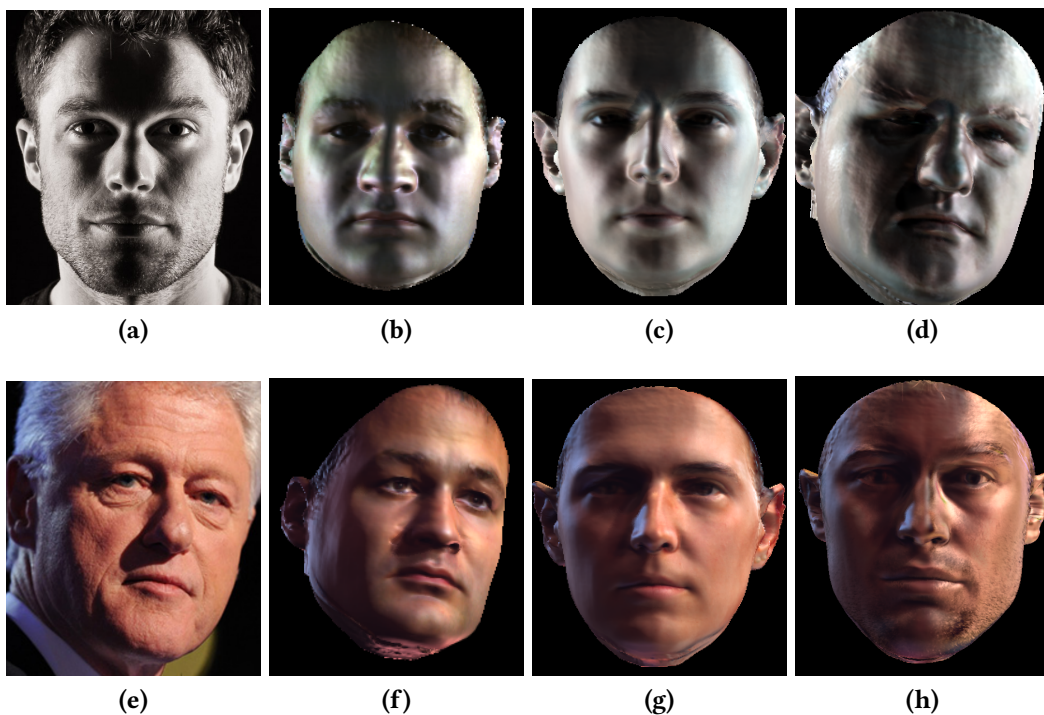


Figure 5.9: This figure shows the transfer of lighting from a face image to another. Two target lightings are estimated from the Figure 5.9a and 5.9e images. The original images from which the target faces are estimated can be seen in Figure 5.7.

5.3.1 Lighting Design

We apply the method from Section 3.6.2 to photos from the Labeled Faces in the Wild database [HRBLM07], and to some paintings and a portrait from public domain. The examples in this section are from [SPB16]. With lighting design it is possible to emphasize silhouettes or structures, or to add depth to the image with coarse sketches on the image. The algorithm estimates a realistic lighting with respect to the painted-on image. Both mild, e.g. Figure 5.12c, and intense modifications, e.g. Figure 5.10d, are possible. Adding rim-light Figure 5.10d, 5.11 (right most example), and removing rim-light Figure 5.11 (left and middle examples) are done successfully. The position of main and fill light can be swapped Figure 5.12b. It is possible to give a new cinematic look to the portrait by introducing different lighting color and directions Figure 5.10b, 5.10c, 5.10d, 5.12a, 5.12c and 5.11. The algorithm is flexible and delivers promising results, however, it is possible to sketch a lighting which is not possible or does not lead exactly to what the user has in mind, especially whenever the 3D shape of the face and physical plausibility are overlooked by the user. In Figure 5.11, the original rim-light is removed with different methods to demonstrate the flexibility of the algorithm.

As you see in some results, e.g. Figure 5.12e (right), the lack of detail in the face geometry and absence of a measured reflectance lead to visible errors, i.e. wrong highlights on the left side and cast shadows under the eye look nonrealistic. These errors are more visible in some lighting conditions and less in other for the same input image, see Figure 5.11 (left). This experiment also shows that the facial appearance cannot be captured under a single lighting condition. Nevertheless, the proposed lighting design is a pioneering method that addresses both simple and harsh illumination conditions, works on a given single noncalibrated image and delivers promising result with elaborate realistic lighting.

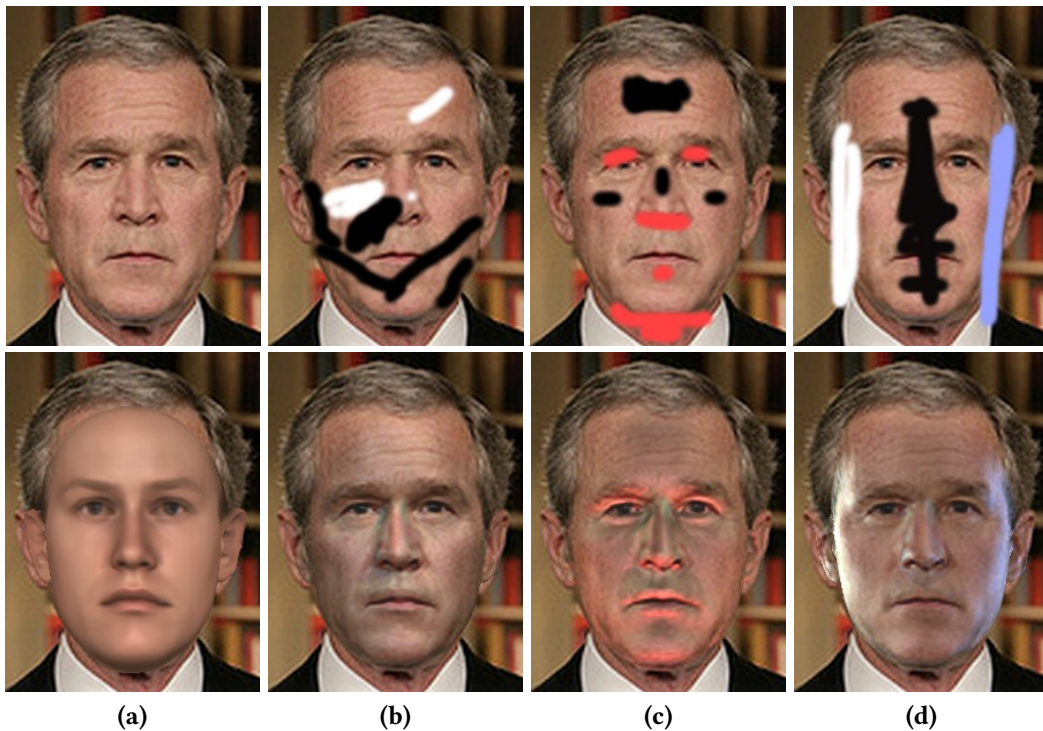


Figure 5.10: Lighting design with automatic landmark localizer from [SPB16]. Figure 5.10a the original image (top) from LFW database [HRBLM07] and the estimated 3D face rendered with average human face texture (bottom). Figure 5.10b, 5.10c and 5.10d are automatically generated results (bottom) with the proposed algorithm for the user-defined lightings.

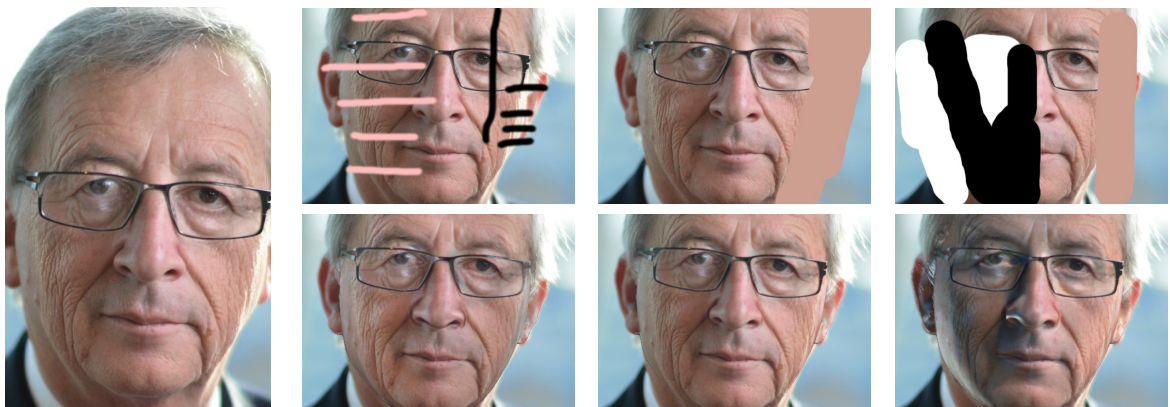


Figure 5.11: This figure shows three different designs on one original face image. It is possible to remove the rim-light with completely different sketches which indicate the same intention (the left most and the middle examples) or apply the Rembrandt lighting (right) on the same original image from LFW [HRBLM07]. The landmark localization has been done manually by the user in [SPB16].

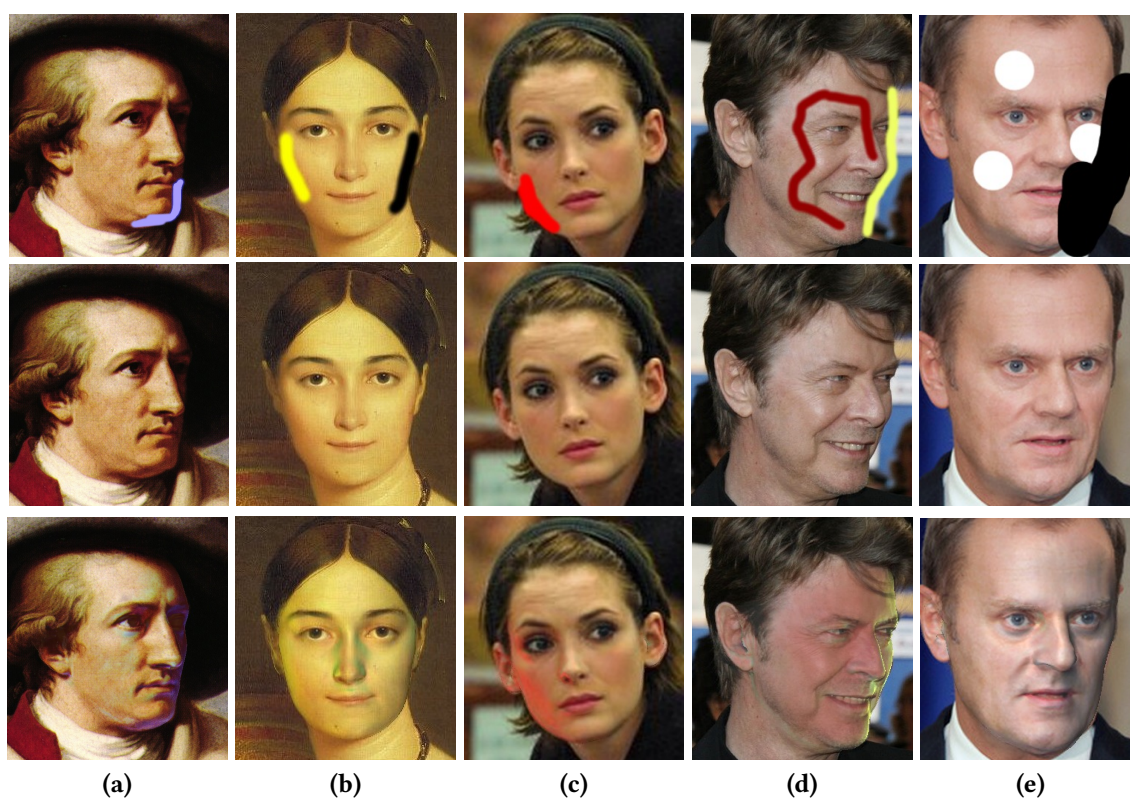


Figure 5.12: Five different examples 5.12a - 5.12e in five columns, each showing the painted-on image (top), original image (middle) and the result image (bottom). Image 5.12d is from German Wikipedia page for David Bowie, the rest are from LFW [HRBLM07]. The landmark localization has been done manually by the user in [SPB16].

Chapter 6

Conclusion

In this dissertation, a Virtual Light Stage framework is proposed for physically plausible and realistic inverse lighting of single face images. The illumination estimation step is performed separately after the 3D face model estimation. With this framework, unknown illumination conditions can be estimated, removed from the estimated face model and transferred to other face models. It is a contribution to realistic face and environment modeling. The proposed inverse lighting tackles the harsh problem of multi-directional and colorful lighting estimation from a single uncalibrated face image which includes cast shadows and specular highlights. It goes beyond any of the previous work on much simpler lighting conditions and Lambert assumptions. A novel geometric superpixel segmentation is proposed for an illumination-friendly data and noise reduction. Many different probabilistic modelings are explored and two of them, i.e. MAP and JMAP, are proposed. In the latter, the hyperparameter optimization automatically handles the inconsistencies between the generative model and the input image. The JMAP approach suits the limitations of missing and unreliable input data and an inaccurate generative model. Furthermore, an automatic model-based method for coarse cast shadow segmentation is introduced to enforce the estimation of a lighting model which supports the reconstruction of the cast shadows. The results show improvements in cast shadow estimation compared to previous work. By enforcing their estimation as significant lighting effects, this dissertation proposes a different paradigm in dealing with cast shadows, compared to related work. Moreover, a novel strategy of quantitative assessment of illumination estimation is proposed. Based on this objective measure, the proposed inverse lighting achieves an average of 70% improvement in terms of increased structural similarity and reduced mean squared error for the input images with a vast variety of illumination complexities. A simple Euclidian distance between the input image and the rendered result gives values that are inconsistent with the perceived visual quality (usually lower than 0.05%). Bringing two different worlds of real and synthetic images together, this dissertation shows that a constrained linear optimization can be numerically stabilized for a

synthesized generating set that is supposed to model real images with high expectations. The rather simple implementation of the nonnegativity constraint for a gradient-based optimization is very efficient for analysis by synthesis approaches on real input images. This is a viable option to the numerous problems in computer vision that demand a non-negative solution. The extent of experiments that are performed to ripen the proposed algorithm to the promising state is beyond the scope of this dissertation. Especially, the experiments with compressive sensing optimization are left out [HCSBL15]. The product of the attempts is a software framework which is designed to handle lowly constrained minimum input data. It can inspire holistic algorithms and other approaches which have more top-down information, for instance when cues about the skin type exist or multiple views, or depth information are provided [SB15b, SB17].

To show the strength of the proposed inverse lighting, a paint-based post hoc lighting design algorithm for portraits is proposed in this dissertation. It is a pose invariant algorithm that handles different imaging conditions and harsh illuminations. The algorithm maintains a good color contrast for novel colorful lighting that are introduced to the scene based on the user's paint strokes. It delivers aesthetically pleasing results corresponding to the user's expectations, and does so with minimum input and effort. An obvious benefit of the physically plausible VLS is the possibility to modify each single light source directly before rendering the result. Together with the paint-based interface, the VLS framework provides a direct interface for lighting design out of the box. Similar to the recent progress in the computational photography on post hoc manipulation of depth-of-field, the proposed algorithm is an editor tool to add apparent depth to images, emphasize silhouettes and structures, and make images more appealing by changing the illumination. It can be entertaining for users to be creative and play with different illuminations. Furthermore, the lighting design algorithm provides a training field for the exploration and study of the lighting effects in portraits [SPB16].

6.1 Summary of Attempts and Achievements

An abstract list of the efforts that are put in thesis and the achievements is provided below:

1. The novel physically plausible model for lighting: the virtual light stage (VLS).
2. Adopting a measurement-based BRDF function for use within the 3DMM framework.
3. The novel geometric superpixel approach for model based segmentation of face images.
4. Two Bayesian approaches to formulate the cost function of the inverse lighting: MAP and JMAP.
5. Interpreting and using hyperparameters for meaningful manipulation of inverse lighting with JMAP.

6. An ad hoc implementation of Newton-Raphson for nonnegative minimization of least squares.
7. Providing a new application field for Compressive Sensing solutions [HCSBL15].
8. A novel model-based cast shadow segmentation.
9. Cast shadow estimation through constrained optimization.
10. Automatic handling of unreliable input, e.g. occlusion, that are inconsistent with the model.
11. Relighting, lighting design, de-illumination and other products of realistic 3D illumination estimation.
12. A tool for creative lighting design for portraits.
13. An objective quantitative measure for assessment of illumination estimation in images of objects with estimated intrinsic model (MISIM and MSIER).

6.2 Unanswered Questions or Future Work

This thesis opens a new door for experiments in the areas of illumination invariant face modeling and recognition, environment modeling through environment lighting estimation, reflectance estimation and more. Let us discuss some of the more specific experiments that I wish I could do but are left out in this dissertation.

First of all, the experiments with the Richardson-Lucy algorithm triggers the idea of using Poisson probability distribution functions for the prior of the light source intensities in the regularized least squares cost function, instead of the proposed half-normal. Unlike Richardson-Lucy which is a maximum likelihood algorithm, this would be a MAP (or JMAP) approach. This might affect the sparsity of the result and indirectly lead to better estimation of the cast shadows and specular highlights due to the sparsity.

Second, I would like to see how the proposed algorithm could be adapted to different, multi-modal input data. Especially, I would test if a different 2D to 3D algorithm leads to better results. Another future work strategy is to adapt the algorithm in Light Field, High Dynamic Range or video data.

My third idea is to work on the intrinsic texture estimation, enhance it with the Markov Random Field penalty term to cover the extracted texture's inconsistencies with ratio method, encouraged by [WLH⁺07]. It might make sense to model the mesoscale facial features, i.e. micro-geometries and texture detail, to achieve higher qualities of rendering.

Forth, I would invest time on the representation of the reflectance and algorithms to estimate a better reflectance from the input image, subsequent to the inverse lighting, for instance the Polynomial Texture Mapping (PTM) representation for the reflectance.

Fifth, the topic of cast shadows raises a few open problems in inverse lighting. In presence of unreliable input data, an unreliably estimated face model and many missing parameters, a more accurate cast shadow estimation can be a goal for future work. Especially, a cost

function on features other than the pixel intensities or the use of generative approaches that model the cast shadow's shape might help.

Finally, the rendering of the VLS images can be parallelized in different levels, as it is currently a bottleneck which takes about the same time as the 3DMM fitting for $n = 100$. The rendering time rises linearly with n . A straightforward parallel design is given in Figure 6.1 where using multiple CPUs together with GPU rendering can reduce the overall rendering time.

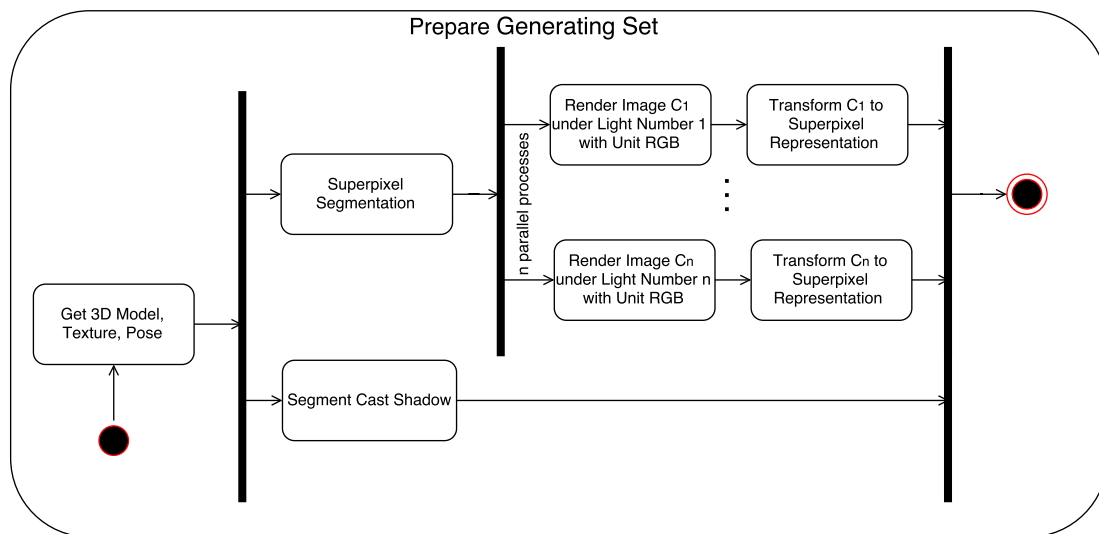


Figure 6.1: This is the UML Activity Diagram is an extension of Figure 3.6 for generating set preparation. This is one way of parallelization which reduces the run time from a few minutes to a few seconds.

Index

Symbols

3DMM, 14, 21, 21, 23, 24, 36, 39, 46–48, 50, 51, 53, 65, 66, 70, 93, 95, 97
3D Morphable Model framework, 8
3D morphable model, 9, 24
3DMM fitting, 8, 13, 21, 22, 22–25, 41, 44, 45, 50–52, 55, 56, 58–60, 62–64, 67, 89, 90, 96, 102
3DMM framework, 4, 17, 24, 30, 45, 108

A

albedo, 18, 31, 46, 52, 59, 62, 82, 97
ambient, 31, 45, 67, 89, 90, 93
ambient light, 27, 31
analysis by synthesis, 3
assessment, 109
attached shadow, 34, 34, 37, 54, 87, 102
average texture, 4, 65, 70

B

barrier function, 84
Bayesian, 9, 16, 73, 108
BFM, 21
blind mask, 51, 74, 75, 81
BRDF, 13, 28, 50, 51, 93

C

cast shadow, 3–5, 10, 12, 14–17, 24, 26, 29, 32, 33, 34, 34, 35, 36, 37, 43, 46, 51, 51–54, 57, 59, 62, 63, 67, 81, 82, 83, 85, 87, 89, 91, 94, 98, 102, 103, 109
cast shadow map, 54
cast shadow mask, 54
color bleeding, 2, 12

color contrast, 36, 36, 50
color correction, 17, 24, 25, 30, 31, 36, 41, 43, 50, 52, 60, 72, 87, 89, 96
Compressive Sensing, 55
convex, 11
convex geometry, 29
correspondence, 6, 8, 22, 24, 39, 41, 48, 51, 52, 55, 56, 58, 59, 70, 75, 89, 102

D

data-driven, 11, 11, 13, 14
depth, 19
diffuse, 7, 31, 33, 49, 50, 65, 67, 79, 90, 93
dipole, 33, 50, 67
directional light, 14, 27, 29
dynamic range, 81

E

eigenfaces, 11, 21
environment map, 9, 14, 47, 102
Estimate Lighting, 41

F

freckle, 18

G

gamma, 85
Gaussian, 71, 72, 76
generative model, 26, 41
geometric superpixel, 10
global lighting, 9
grazing angle, 33

H

hair, 18, 23, 24

- halfnormal, 85
- harsh illumination, 5, 15, 46, 92
- HDR, 27
- Hessian, 77, 84, 85, 87, 88
- hierarchical, 16
- hyperparameter, 51, 57, 58, 75, 76, 78, 79, 81, 108
- hyperparameter optimization, 51, 69, 70, 75, 81, 81–83
- hyperparameters, 16
- I**
- illumination cone, 11, 25, 25, 29, 41, 45, 69
- illumination estimation, 25
- illumination invariant, 11
- illumination map, 67
- illumination-invariant, 21
- image-based, 16
- indirect lighting, 2
- input image, 6
- inter-reflection, 12
- intrinsic, 7, 9, 46, 59, 63, 95, 109
- intrinsic texture, 9, 10, 24, 82, 94, 97
- inverse lighting, 4, 8, 14–16, 30, 37, 41, 45, 47, 48, 57, 88
- inverse rendering, 4, 8, 11, 15
- irradiance, 29
- J**
- JMAP, 6, 7, 10, 51, 73, 75, 76, 78, 81, 82, 85, 87, 88, 91, 92, 101, 102, 108
- L**
- Lambert, 25
- Lambertian, 11, 14, 29
- landmark, 6, 6
- least squares, 70, 72, 72–74, 77, 79, 81, 109
- light probe, 7
- light stage, 11, 12
- lighting, 18
- lighting design, 7, 16, 17, 65
- lighting estimation, 1, 11, 17, 41, 53, 82
- linear cone, 25
- lognormal, 85
- M**
- macroscale, 18, 18
- MAP, 6, 13, 73, 75, 81–83, 85, 87, 91, 92, 94, 108
- mask, 51
- mesoscale, 18, 18, 23, 24
- metaparameter, 75
- microgeometry, 23
- microscale, 18, 18
- misalignment, 6, 81
- MISIM, 87, 89, 90, 91, 97, 100
- mole, 18
- moles, 23
- MRF, 24
- MSE, 89, 90
- MSIER, 90, 91, 97, 100
- MSSIM, 89, 90
- N**
- Newton-Raphson, 109
- noise, 9
- nonnegative, 7, 9, 69, 76, 79, 81, 84, 85, 87, 109
- nonnegative optimization, 30
- nonnegativity, 72, 75, 77, 80, 84
- normals, 16
- O**
- occluding, 18
- occlusion, 3, 7, 16, 24, 70, 74, 75, 81, 81, 109
- occlusion mask, 7, 42, 92
- OpenCV, 87, 88
- P**
- PCA, 11, 13, 24
- penumbra, 34

- PGM, 75, 76
- Phong, 14, 27, 31, 31, 45, 48, 93, 97
- physically plausible, 6, 7, 7, 11, 14–16, 29, 92, 97, 108
- Poisson, 86
- pore, 18
- pose, 18
- pseudo-Hessian, 88
- pseudo-inverse, 77, 87, 88
- Q**
- quantitative, 9, 109
- R**
- radiance, 29, 30
- ratio image, 24
- reflectance, 3, 7–14, 17–19, 25, 29, 31, 48–50, 55, 65, 70, 81, 82, 97
- reflectance function, 67
- reflective Fresnel, 33, 34
- regularization, 30, 73–75, 77–79, 82, 84
- regularized, 70, 72, 73
- relighting, 7, 11, 13, 16, 48
- response function, 26
- Richardson-Lucy, 86, 87, 88
- RL, *see also* Richardson-Lucy, 86, 87
- S**
- saturation, 36
- segmentation, 74, 109
- shadow buffer, 34
- shadow map, 53
- shadows, 18
- silhouette, 6
- single-image, 4, 11, 14, 14, 15, 17
- sketch-based, 17
- skin, 18, 24
- SLIC, 38–40, 55, 56, 58
- SNR, 89
- specular, 15, 17, 26, 29, 31, 33, 33, 49, 50, 63, 67, 93, 95, 97, 98
- specularity, 24
- spherical harmonics, 14, 15, 26, 28, 28, 29
- SSIM, 90
- subsurface light transfer, 33
- superpixel, 9, 39, 46, 53, 55, 55–58, 70–73, 77, 80, 81, 83, 87, 90, 92, 98, 99, 108
- superpixel segmentation, 38, 39, 40, 55, 56
- superposition, 11, 28, 29, 29, 41, 45, 69
- surface normal, 7, 39, 56, 57, 60, 69, 98
- SVD, 29, 87
- synthetic illumination cone, 46, 69
- T**
- termination condition, 83
- texture, 59
- texture estimation, 4
- texture patches, 39, 40
- Torrance-Sparrow, 31, 33, 49, 50, 67
- transmittance Fresnel, 33, 34
- U**
- umbra, 34
- UML Activity Diagram, 44, 65
- UML Use Case Diagram, 41
- V**
- virtual light stage, 29, 30, 46
- VLS, *see also* virtual light stage, 9, 30, 43, 46, 46–48, 59, 60, 69, 71, 97, 98, 102, 108
- W**
- white balance, 70
- wrinkle, 18, 23

Bibliography

- [ABC11] Alessandro Artusi, Francesco Banterle, and Dmitry Chetverikov. A survey of specular removal methods. *30(8):2208–2230*, 2011.
- [ADW04] Frederik Anrys, Philip Dutré, and YD Willems. Image based lighting design. In *The 4th IASTED International Conference on Visualization, Imaging, and Image Processing*, volume 2. Citeseer, 2004.
- [Agi] Agile modeling home page, effective practices for modeling and documentation. <http://agilemodeling.com>. Accessed: 2017-03-30.
- [ALK⁺03] David Akers, Frank Losasso, Jeff Klingner, Maneesh Agrawala, John Rick, and Pat Hanrahan. Conveying shape and features with image-based relighting. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 46. IEEE Computer Society, 2003.
- [AP08] Xiaobo An and Fabio Pellacini. AppProp: all-pairs appearance-space edit propagation. *ACM Transactions on Graphics (TOG)*, 27(3):40, 2008.
- [AS12] Oswald Aldrian and William AP Smith. Inverse rendering of faces on a cloudy day. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012.
- [AS13] Oswald Aldrian and William AP Smith. Inverse rendering of faces with a 3D morphable model. *IEEE transactions on pattern analysis and machine intelligence*, 35(5):1080–1093, 2013.
- [ASS⁺10] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels. Technical report, 2010.
- [ASS⁺12] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

- [AVM⁺14] Amr Almaddah, Sadi Vural, Yasushi Mae, Kenichi Ohara, and Tatsuo Arai. Face relighting using discriminative 2D spherical spaces for face recognition. *Machine Vision and Applications*, 25(4):845–857, 2014.
- [B⁺00] Gary Bradski et al. The opencv library. *Doctor Dobbs Journal*, 25(11):120–126, 2000.
- [BAPD13] Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1513–1520, 2013.
- [Ber09] Dennis S Bernstein. *Matrix mathematics: theory, facts, and formulas*. Princeton University Press, 2009.
- [BF01] Matt Bell and ET Freeman. Learning local evidence for shading and reflectance. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 670–677. IEEE, 2001.
- [Bir00] Jeremy Birn. *Digital Lighting and Rendering*. New Riders Publishing, Thousand Oaks, CA, USA, 2000.
- [BJ03] Ronen Basri and David W Jacobs. Lambertian reflectance and linear subspaces. *IEEE transactions on pattern analysis and machine intelligence*, 25(2):218–233, 2003.
- [BK96] Peter N Belhumeur and David J Kriegman. What is the set of images of an object under all possible lighting conditions? In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 270–277. IEEE, 1996.
- [BK98] PeterN. Belhumeur and DavidJ. Kriegman. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):245–260, 1998.
- [BKD⁺08] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K Nayar. Face swapping: automatically replacing faces in photographs. *ACM Transactions on Graphics (TOG)*, 27(3):39, 2008.
- [BKTT98] Wendy L Braje, Daniel Kersten, Michael J Tarr, and Nikolaus F Troje. Illumination effects in face recognition. *Psychobiology*, 26(4):371–380, 1998.
- [BKY99] Peter N Belhumeur, David J Kriegman, and Alan L Yuille. The bas-relief ambiguity. *International journal of computer vision*, 35(1):33–44, 1999.

- [BM15] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2015.
- [BMVS04] Volker Blanz, Albert Mehl, Thomas Vetter, and Hans-Peter Seidel. A statistical method for robust 3D surface reconstruction from sparse data. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 293–300. IEEE, 2004.
- [Boy14] Peter Robert Boyce. *Human factors in lighting*. Crc Press, 2014.
- [BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [BV03] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3D morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, 2003.
- [CCH07] Biswarup Choudhury, Sharat Chandran, and Jens Herder. A survey of image-based relighting techniques. *Journal of Virtual Reality and Broadcasting*, 4(7), 2007.
- [CDMR05] Oana G Cula, Kristin J Dana, Frank P Murphy, and Babar K Rao. Skin texture modeling. *International Journal of Computer Vision*, 62(1-2):97–119, 2005.
- [CDS01] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, January 2001.
- [CKZX13] Guangyi Chen, Sridhar Krishnan, Yongjia Zhao, and Wenfang Xie. Illumination invariant face recognition. In *International Conference on Intelligent Computing*, pages 385–391. Springer, 2013.
- [CMS87] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflection functions from surface bump maps. In *ACM Siggraph Computer Graphics*, volume 21, pages 273–281. ACM, 1987.
- [CP09] Donghui Chen and Robert J Plemmons. Nonnegativity constraints in numerical analysis. *Symposium on the Birth of Numerical Analysis*, pages 109–140, 2009.
- [CPC84] Robert L Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *ACM SIGGRAPH Computer Graphics*, volume 18, pages 137–145. ACM, 1984.

- [CRA⁺13] Vincent Christlein, Christian Riess, Elli Angelopoulou, Georgios Evangelopoulos, and Ioannis Kakadiaris. The impact of specular highlights on 3D-2D face recognition. *Proc. SPIE*, 8712:87120T–87120T–13, 2013.
- [CSF99] António Cardoso Costa, António Augusto Sousa, and Fernando Nunes Ferreira. Lighting design: A goal based approach using optimisation. In *Rendering Techniques' 99*, pages 317–328. Springer, 1999.
- [Deb06] Paul Debevec. A median cut algorithm for light probe sampling. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.
- [Deb12] Paul Debevec. The light stages and their applications to photoreal digital actors. *SIGGRAPH Asia Technical Briefs*, 2, 2012.
- [DFS08] Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding*, 109(1):22–43, 2008.
- [DHT⁺00] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. *Proc. 27th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '00*, pages 145–156, 2000.
- [DSWH14] Arnaud Dessein, William AP Smith, Richard C Wilson, and Edwin R Hancock. Seamless texture stitching on a 3D mesh by Poisson blending in patches. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2031–2035. IEEE, 2014.
- [DSWH15] Arnaud Dessein, William AP Smith, Richard C Wilson, and Edwin R Hancock. Example-Based Modeling of Facial Texture from Deficient Data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3898–3906, 2015.
- [DSWH16] A Dessein, WAP Smith, RC Wilson, and ER Hancock. Symmetry-aware mesh segmentation into uniform overlapping patches. In *Computer Graphics Forum*. Wiley Online Library, 2016.
- [DWT⁺02] Paul Debevec, Andreas Wenger, Chris Tchou, Andrew Gardner, Jamie Waese, and Tim Hawkins. *A lighting reproduction approach to live-action compositing*, volume 21. ACM, 2002.
- [FBLS05] Martin Fuchs, Volker Blanz, Hendrik Lensch, and H-P Seidel. Reflectance from images: A model-based approach for human faces. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):296–305, 2005.

- [FBLS07] Martin Fuchs, Volker Blanz, Hendrik Lensch, and Hans-Peter Seidel. Adaptive sampling of reflectance fields. *ACM Transactions on Graphics (TOG)*, 26(2):10, 2007.
- [FBS05] Martin Fuchs, Volker Blanz, and Hans-Peter Seidel. Bayesian relighting. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques, EGSR'05*, pages 157–164, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.
- [FWBP95] DA Fish, JG Walker, AM Brinicombe, and ER Pike. Blind deconvolution by means of the Richardson–Lucy algorithm. *JOSA A*, 12(1):58–65, 1995.
- [GBK99] Athinodoros S Georghiadis, Peter N Belhumeur, and David J Kriegman. Illumination-based image synthesis: Creating novel images of human faces under differing pose and lighting. In *Multi-View Modeling and Analysis of Visual Scenes, 1999.(MVIEW'99) Proceedings. IEEE Workshop on*, pages 47–54. IEEE, 1999.
- [GBK01] Athinodoros S. Georghiadis, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001.
- [GMC⁺10] Ralph Gross, Iain Matthews, Jeffrey F. Cohn, Takeo Kanade, and Simon Baker. Multi-PIE. *Image and Vision Computing*, 28(5):807–813, 2010.
- [GTB⁺13] Paul Graham, Borom Tunwattanapong, Jay Busch, Xueming Yu, Andrew Jones, Paul Debevec, and Abhijeet Ghosh. Measurement-Based Synthesis of Facial Microgeometry. *Comput. Graph. Forum*, 32(2pt3):335–344, May 2013.
- [HAKA07] S Huq, B Abidi, SG Kong, and M Abidi. A survey on 3D modeling of human faces for face recognition. In *3D Imaging for Safety and Security*, pages 25–67. Springer, 2007.
- [Hal94] Peter W Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 995–999. IEEE, 1994.
- [HB89] Berthold KP Horn and Michael J Brooks. *Shape from shading*. MIT press, 1989.

- [HCSBL15] Miguel Heredia Conde, Davoud Shahlaei, Volker Blanz, and Otmar Loffeld. Efficient and robust inverse lighting of a single face image using compressive sensing. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.
- [HED05] Tim Hawkins, Per Einarsson, and Paul E Debevec. A dual light stage. *Rendering Techniques*, 5:91–98, 2005.
- [HJ61] Robert Hooke and T. A. Jeeves. “direct search” solution of numerical and statistical problems. *J. ACM*, 8(2):212–229, April 1961.
- [HO15] Bernardo Henz and Manuel M Oliveira. Artistic relighting of paintings and drawings. *The Visual Computer*, pages 1–14, 2015.
- [HRBLM07] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.
- [INN07] Takanori Igarashi, Ko Nishino, and Shree K. Nayar. The appearance of human skin: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(1):1–95, January 2007.
- [IYH⁺14] Maria Ingaramo, Andrew G York, Eelco Hoogendoorn, Marten Postma, Hari Shroff, and George H Patterson. Richardson-lucy deconvolution as a general tool for combining images with complementary strengths. *Chemphyschem: a European journal of chemical physics and physical chemistry*, 15(4):794, 2014.
- [JB02] Henrik Wann Jensen and Juan Buhler. A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.*, 21(3):576–581, July 2002.
- [JCK15] László A Jeni, Jeffrey F Cohn, and Takeo Kanade. Dense 3D face alignment from 2D videos in real-time. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–8. IEEE, 2015.
- [JH98] Andrew Edie Johnson and Martial Hebert. Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16(9):635–651, 1998.
- [Jol02] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.
- [JSG09] Jorge Jimenez, Veronica Sundstedt, and Diego Gutierrez. Screen-space perceptual rendering of human skin. *ACM Trans. Appl. Percept.*, 6(4):23:1–23:15, October 2009.

- [JZJ⁺15] Jorge Jimenez, Károly Zsolnai, Adrian Jarabo, Christian Freude, Thomas Auzinger, Xian-Chun Wu, Javier der Pahlen, Michael Wimmer, and Diego Gutierrez. Separable subsurface scattering. In *Computer Graphics Forum*. Wiley Online Library, 2015.
- [KF09] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [KHSB98] AZ Kouzani, Fangpo He, Karl Sammut, and A Bouzerdoum. Illumination invariant face recognition. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 5, pages 4240–4245. IEEE, 1998.
- [KMK97a] Daniel Kersten, Pascal Mamassian, and David C Knill. Moving cast shadows and the perception of relative depth. *Perception*, 26(2):171–192, 1997.
- [KMK97b] David C Knill, Pascal Mamassian, and Daniel Kersten. Geometry of shadows. *JOSA A*, 14(12):3216–3232, 1997.
- [KP09] William B. Kerr and Fabio Pellacini. Toward evaluating lighting design interface paradigms for novice users. *ACM Trans. Graph.*, 28(3):26:1–26:9, July 2009.
- [Kra01] Joe Krakora. *Vermeer: Master of Light*. Interface Media Group, 2001.
- [KSB11] Ira Kemelmacher-Shlizerman and Ronen Basri. 3D face reconstruction from a single image using a single reference face shape. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):394–405, February 2011.
- [Kur11] Noriko Kurachi. *The magic of computer graphics*. CRC Press, 2011.
- [LCLZ07] Stan Z Li, RuFeng Chu, ShengCai Liao, and Lun Zhang. Illumination invariant face recognition using near-infrared images. *IEEE Transactions on pattern analysis and machine intelligence*, 29(4):627–639, 2007.
- [LLS05] Dang-Hui Liu, Kin-Man Lam, and Lan-Sun Shen. Illumination invariant face recognition. *Pattern Recognition*, 38(10):1705–1716, 2005.
- [LSH06] Yang Li, William AP Smith, and Edwin R Hancock. Face recognition using patch-based spin images. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 408–411. IEEE, 2006.
- [LSKS16] Shu Liang, Linda G Shapiro, and Ira Kemelmacher-Shlizerman. Head reconstruction from internet photos. In *European Conference on Computer Vision*, pages 360–374. Springer, 2016.

- [Luc74] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astron. J.*, 79(6):745–754, 1974.
- [LZL14] Chen Li, Kun Zhou, and Stephen Lin. Intrinsic face image decomposition with human face priors. In *European Conference on Computer Vision*, pages 218–233. Springer, 2014.
- [Mac48] Thomas Murray MacRobert. *Spherical harmonics: An elementary treatise on harmonic functions, with applications*. Dover Publ., 1948.
- [Mar98] Stephen Robert Marschner. *Inverse rendering for computer graphics*. PhD thesis, Citeseer, 1998.
- [MAU94] Yael Moses, Yael Adini, and Shimon Ullman. Face recognition: The problem of compensating for changes in illumination direction. In Jan-Olof Eklundh, editor, *Computer Vision — ECCV ’94*, volume 800 of *Lecture Notes in Computer Science*, pages 286–296. Springer Berlin Heidelberg, 1994.
- [MBW⁺07] Ankit Mohan, Reynold Bailey, Jonathan Waite, Jack Tumblin, Cindy Grimm, and Bobby Bodenheimer. Tabletop computed lighting for practical digital photography. *IEEE transactions on visualization and computer graphics*, 13(4):652–662, 2007.
- [MD96] Ali Mohammad-Djafari. A full Bayesian approach for inverse problems. In *Maximum entropy and Bayesian methods*, pages 135–144. Springer, 1996.
- [MDA02] Vincent Masselus, Philip Dutré, and Frederik Anrys. The free-form light stage. In *ACM SIGGRAPH 2002 conference abstracts and applications*, pages 262–262. ACM, 2002.
- [MKK98] Pascal Mamassian, David C Knill, and Daniel Kersten. The perception of cast shadows. *Trends in cognitive sciences*, 2(8):288–295, 1998.
- [MMS⁺05] Gero Müller, Jan Meseth, Mirko Sattler, Ralf Sarlette, and Reinhard Klein. Acquisition, synthesis, and rendering of bidirectional texture functions. In *Computer Graphics Forum*, volume 24, pages 83–109. Wiley Online Library, 2005.
- [MTB⁺05] Ankit Mohan, Jack Tumblin, Bobby Bodenheimer, Reynold Bailey, and Cindy Grimm. Table-top computed lighting for practical digital photography. In *ACM SIGGRAPH 2005 Sketches*, page 76. ACM, 2005.
- [MUn12] Rosana Montes and Carlos Ureña. An overview of BRDF models. Technical report, University of Granada, 2012.

- [MWM⁺98] Gerhard Meister, Rafael Wiemker, Rene Monno, Hartwig Spitzer, and Alan Strahler. Investigation on the Torrance-Sparrow specular BRDF model. In *Geoscience and Remote Sensing Symposium Proceedings, 1998. IGARSS'98. 1998 IEEE International*, volume 4, pages 2095–2097. IEEE, 1998.
- [NCG15] Aditya Nigam, Gitesh Chhalotre, and Phalguni Gupta. Pose and illumination invariant face recognition using binocular stereo 3D reconstruction. In *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2015 Fifth National Conference on*, pages 1–4. IEEE, 2015.
- [NN04] Ko Nishino and Shree K Nayar. Eyes for relighting. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 704–711. ACM, 2004.
- [OZM⁺06] Makoto Okabe, Gang Zeng, Yasuyuki Matsushita, Takeo Igarashi, Long Quan, and Heung-Yeung Shum. Single-view relighting with normal map painting. In *Proc. Pacific Graphics*, pages 27–34, 2006.
- [PB16] Marcel Pietraschke and Volker Blanz. Automated 3D face reconstruction from multiple images using quality measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3418–3427, 2016.
- [PC05] Rohit Patnaik and David Casasent. Illumination invariant face recognition and impostor rejection using different minace filter algorithms. In *Defense and Security*, pages 94–104. International Society for Optics and Photonics, 2005.
- [PC06] Gabriel Peyré and Laurent D Cohen. Geodesic remeshing using front propagation. *International Journal of Computer Vision*, 69(1):145–156, 2006.
- [PF92] Pierre Poulin and Alain Fournier. Lights from highlights and shadows. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 31–38. ACM, 1992.
- [PF06] Emmanuel Prados and Olivier Faugeras. Shape from shading. In *Handbook of mathematical models in computer vision*, pages 375–388. Springer, 2006.
- [Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.
- [PKA⁺09a] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. IEEE, 2009.

- [PKA⁺09b] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. Ieee, 2009.
- [PP03] Gustavo Patow and Xavier Pueyo. A survey of inverse rendering problems. In *Computer graphics forum*, volume 22, pages 663–687. Wiley Online Library, 2003.
- [PRJ97] Pierre Poulin, Karim Ratib, and Marco Jacques. Sketching shadows and highlights to position lights. In *Computer Graphics International, 1997. Proceedings*, pages 56–63. IEEE, 1997.
- [PTG02] Fabio Pellacini, Parag Tole, and Donald P Greenberg. A user interface for interactive cinematic shadow design. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 563–566. ACM, 2002.
- [PTMD07] Pieter Peers, Naoki Tamura, Wojciech Matusik, and Paul Debevec. Post-production facial performance relighting using reflectance transfer. *ACM Trans. Graph.*, 26(3), July 2007.
- [PTVF96] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical recipes in C*, volume 2. Cambridge university press Cambridge, 1996.
- [PWSP11] Alexandros Panagopoulos, Chaohui Wang, Dimitris Samaras, and Nikos Paragios. Illumination estimation and cast shadow detection through a higher-order graphical model. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 673–680. IEEE, 2011.
- [RDL⁺15] Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. Image based relighting using neural networks. *ACM Trans. Graph.*, 34(4):111:1–111:12, July 2015.
- [RH01a] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. *Proc. 28th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '01*, pages 497–500, 2001.
- [RH01b] Ravi Ramamoorthi and Pat Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex Lambertian object. *J. Opt. Soc. Am. A*, 18(10):2448, 2001.

- [RH01c] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, pages 117–128, New York, NY, USA, 2001. ACM.
- [RHVK06] S. Romdhani, J. Ho, T. Vetter, and D.J. Kriegman. Face recognition using 3-D models: Pose and illumination. *Proc. IEEE*, 94(11):1977–1999, November 2006.
- [Ric72] William Hadley Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972.
- [RKB05] Ravi Ramamoorthi, Melissa Koudelka, and Peter Belhumeur. A fourier theory for cast shadows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(2):288–295, February 2005.
- [RV05] Sami Romdhani and Thomas Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02, CVPR '05*, pages 986–993, Washington, DC, USA, 2005. IEEE Computer Society.
- [SAWG91] Francis X Sillion, James R Arvo, Stephen H Westin, and Donald P Greenberg. A global illumination solution for general reflectance distributions. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 187–196. ACM, 1991.
- [SB15a] Matthaeus Schumacher and Volker Blanz. Exploration of the correlations of attributes and features in faces. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–8. IEEE, 2015.
- [SB15b] Davoud Shahlaei and Volker Blanz. Realistic inverse lighting from a single 2d image of a face, taken under unknown and complex lighting. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–8. IEEE, 2015.
- [SB17] Davoud Shahlaei and Volker Blanz. Hierarchical Bayesian inverse lighting of portraits with a virtual light stage. *submitted to: TPAMI*, 2017.
- [SBB02] Terence Sim, Simon Baker, and Maan Bsat. The CMU pose, illumination, and expression (PIE) database. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, FGR '02*, pages 53–, Washington, DC, USA, 2002. IEEE Computer Society.

- [Sch94] Christophe Schlick. An inexpensive BRDF model for physically-based rendering. In *Computer graphics forum*, volume 13, pages 233–246. Wiley Online Library, 1994.
- [Sch05] Mark Schmidt. Least squares optimization with L1-norm regularization. *Proj. Report, Univ. Br. Columbia*, (December), 2005.
- [SEMFV16] Sandro Schönborn, Bernhard Egger, Andreas Morel-Forster, and Thomas Vetter. Markov chain monte carlo for automated face image analysis. *International Journal of Computer Vision*, pages 1–24, 2016.
- [Shy99] Manoj Night Shyamalan. *Sixth Sense*. Hollywood Pictures ,Spyglass Entertainment (presents) Kennedy/Marshall Company, Barry Mendel Productions, 1999.
- [SL01] Ram Shacked and Dani Lischinski. Automatic lighting design using a perceptual quality metric. In *Computer graphics forum*, volume 20, pages 215–227. Wiley Online Library, 2001.
- [SL08] Yael Shor and Dani Lischinski. The shadow meets the mask: Pyramid-based shadow removal. In *Computer Graphics Forum*, volume 27, pages 577–586. Wiley Online Library, 2008.
- [SPB15] Matthaeus Schumacher, Marcel Piotraschke, and Volker Blanz. Hallucination of facial details from degraded images using 3d face models. *Image and Vision Computing*, 40:49–64, 2015.
- [SPB16] Davoud Shahlaei, Marcel Piotraschke, and Volker Blanz. Lighting design for portraits with a virtual light stage. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1579–1583. IEEE, 2016.
- [SSKS15] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. What makes tom hanks look like tom hanks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3952–3960, 2015.
- [Tib94] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.
- [TP91] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.
- [TS67] Kenneth E Torrance and Ephraim M Sparrow. Theory for off-specular reflection from roughened surfaces. *JOSA*, 57(9):1105–1112, 1967.

- [USC08] The Light Stages at UC Berkeley and USC ICT. <http://gl.ict.usc.edu/LightStages/>, 2008. Accessed: 2016-11-06.
- [WBSS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [Wey06] Tim Alexander Weyrich. *Acquisition of human faces using a measurement-based skin reflectance model*. PhD thesis, ETH, MERL, 2006.
- [Whi94] Richard L White. Image restoration using the damped richardson-lucy method. In *1994 Symposium on Astronomical Telescopes & Instrumentation for the 21st Century*, pages 1342–1348. International Society for Optics and Photonics, 1994.
- [WK15] Michael Weinmann and Reinhard Klein. Advances in geometry and reflectance acquisition. In *SIGGRAPH Asia 2015 Courses*, pages 1:1–1:71. ACM, 2015. Article No. 1.
- [WLH03] Zhen Wen, Zicheng Liu, and Thomas S Huang. Face relighting with radiance environment maps. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–158. IEEE, 2003.
- [WLH⁺07] Yang Wang, Zicheng Liu, Gang Hua, Zhen Wen, Zhengyou Zhang, and Dimitris Samaras. Face re-lighting from a single image under harsh lighting conditions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [WMP⁺06] Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, and Markus Gross. Analysis of human faces using a measurement-based skin reflectance model. *ACM Trans. Graph.*, 25(3):1013–1024, July 2006.
- [WZL⁺09] Yang Wang, Lei Zhang, Zicheng Liu, Gang Hua, Zhen Wen, Zhengyou Zhang, and Dimitris Samaras. Face relighting from a single image under arbitrary unknown lighting conditions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1968–1984, 2009.
- [YK06] Alan Yuille and Daniel Kersten. Vision as Bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006.

- [ZCPR03] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003.
- [ZKM07] Xuan Zou, Josef Kittler, and Kieron Messer. Illumination invariant face recognition: A survey. In *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on*, pages 1–8. IEEE, 2007.
- [ZR12] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.
- [ZS06] Lei Zhang and Dimitris Samaras. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):351–63, March 2006.
- [ZSK13] Xi Zhao, Shishir K Shah, and Ioannis A Kakadiaris. Illumination alignment using lighting ratio: Application to 3D-2D face recognition. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–6. IEEE, 2013.
- [ZTCS99] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999.