Skin Segmentation for Robust Face Image Analysis

Inauguraldissertation

zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel
von

Jean-Sébastien Pierrard

aus Freiburg i. Br., Deutschland

Basel, 2008

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Prof. Dr. Thomas Vetter, Universität Basel, Dissertationsleiter
Prof. Dr.-Ing. Hans Burkhardt, Universität Freiburg i. Br., Korreferent

Basel, den 20.05.2008

Prof. Dr. Hans-Peter Hauri, Dekan

Abstract

This thesis presents novel techniques to address the challenge of outlier detection and removal in the context of face analysis from photographs. Given a face image, under arbitrary scene conditions, our goal is to automatically compute a binary map that indicates the locations of facial occlusions, such as hairstyle, beard, clothing or glasses, and other atypical elements. The motivation is that this information can help other face processing methods, which do not tackle this problem on their own, to improve their results with minimal algortihmic adjustments. The 3D Morphable Model is a good example for such a method, and serves as testbed for our findings.

Usually outliers are difficult to capture. By definition they represent unpredictable deviations from facial appearance, which elude a systematical analysis. The problem is, that outliers impair a face description by perturbing extracted features. This can lead to wrong classifications or otherwise defective outputs. Therefore, in the face recognition literature, several methods have been devised to deal with this phenomenon. However, these solutions are neither comparable to our approach, nor applicable to our target applications, as they are often suited to a specific feature representation and not comprehensive.

We address the outlier problem, for the first time, as a classical segmenation task. The main contribution of our work is an algortihm, which determines the location of outliers on a pixel scale, by partitioning a face image into skin and “non-skin” regions. The algortihm is designed to work completely automatic and, unlike conventional skin detection techinques, it does not depend on color input. The latter is accomplished by means of a novel low-level texture analysis procedure, which comprises an illumination compensation step and a subsequent matching of image regions with respect to a given sample of skin texture. The resulting texture features are segmented with a customized version of the supervised *GrabCut* method. In order to facilitate automation, we incorporate structural knowledge on faces from the 3D Morphable Model. It allows us to mark specific facial areas, which are utilized as skin samples as well as to initialize the actual segmentation routine.

We demonstrate the significance of the skin segmentation on three applications. First, it serves as main component to create an outlier map, that works in combination with a slightly modified fitting algorithm, to greatly improve the visual quality of 3D Morphable Model reconstructions. The second application extends this capability and reuses the image content, associated with the outliers, to realize a high level photo manipulation, called *Face Exchange*. The aim here is to substitute faces between different images, without affecting the rest of the scene. The last contribution represents a novel approach to face recognition. We localize prominent irregularities in facial skin, particularly moles, in order to use their characteristic configuration within a face for identification. For this task the skin segments are of utmost importance, to ensure high detection accuracy, and expressiveness of the extracted features.

Contents

1 Introduction **9**

1.1 Motivation & Overview 10

1.2 Related Work 12

2 3DMM Face Representation **15**

2.1 Model Construction 15

2.1.1 Correspondence using Optical Flow 15

2.1.2 Face Vectors 16

2.1.3 Statistical Analysis 18

2.2 3D Morphable Model Fitting 22

2.2.1 Rendering Parameters 22

2.2.2 Formulation as Minimization Problem 23

2.2.3 Landmark Assisted Fitting 24

2.2.4 Dense Mapping to/from Image 25

2.2.5 Texture Extraction 25

3 Review on Segmentation Techniques **27**

3.1 Segmentation with Graph Theoretic Methods 27

3.1.1 Unsupervised Clustering Techniques 28

3.1.2 Supervised Segmentation Techniques 32

3.1.3 Advanced Affinity Measures 35

3.2 Clustering in Feature Space 38

3.3 Deformable Contours 40

4 Skin Segmentation for Faces **43**
4.1 3DMM Deficiencies . . . 44
4.1.1 Causes of Bad Reconstructions . . . 44
4.1.2 Segmentation Hints . . . 47
4.2 Skin Features for Gray Level Images . . . 48
4.2.1 Distinguishing Texture by Analogy . . . 49
4.2.1.1 Application to Faces . . . 52
4.2.2 Illumination Compensation . . . 53
4.2.2.1 Combined Application with Texture Similarity . . . 60
4.2.2.2 Application to Skin Segmentation . . . 60
4.3 *GrabCut* . . . 61
4.3.1 Problem Formulation . . . 62
4.3.2 Algorithm . . . 63
4.3.3 Implementation Specifics . . . 65
4.3.4 Results . . . 66

5 Soft Segmentation by Alpha Matting **71**
5.1 Background . . . 71
5.2 Closed-Form Solution . . . 74
5.3 Spectral Matting . . . 75
5.4 A Practical Assessment . . . 78
5.5 Reconstruction of Foreground & Background . . . 79

6 Applications **81**
6.1 Face Recognition from Skin Details . . . 81
6.1.1 Overview . . . 82
6.1.2 Mole Candidate Detection . . . 84
6.1.3 Skin Segmentation . . . 85
6.1.4 Local Saliency . . . 86
6.1.5 Identification Experiments . . . 88
6.2 Outlier Masking for 3DMM Fitting . . . 92
6.2.1 Selective Fitting . . . 92
6.2.2 Automatic Outlier Mask Generation . . . 94

6.2.3 Alpha Matting for Outliers 95
6.3 Face Exchange 97
6.3.1 Overview 97
6.3.2 Counteracting Artefacts 101
6.3.2.1 Seamless Blending 101
6.3.2.2 Shape Scale Adjustment 101
6.3.2.3 Facial Occlusions 102
6.3.2.4 Background Restoration 103
6.3.3 Workflow & Results 106

7 Conclusion **111**

List of figures and tables **115**

Bibliography **118**

Chapter 1

Introduction

The task of an automated face analysis system is to infer from acquired sensory data, like digital photographs or videos, some higher level knowledge, based on which the computer can make application specific decisions and perhaps control real-world processes. Human faces communicate a multitude of informations that can be targeted by such a system. They range from mere measurements like pose (position, scale, orientation) and shape, over descriptive attributes including gender, race and age to concepts like identity and expression/emotion. Within this field the problem of face recognition has always been of major interest. Compared to other biometric characteristics, which can be used for identification and which are more accurate, such as fingerprints or iris and retina scans, only faces can be recorded in a non-intrusive manner and from greater distance without the subject's cooperation. Besides the practical importance, face recognition is also the humans' primary method of person identification. Since the advent of computer vision it has inspired innumerable contributions and popularized many of today's established pattern recognition and learning techniques. Still, after over thirty years of research, the problem is not yet solved generally.

The generic face recognition task can be formulated as simple as follows: *Given two face images, decide whether the depicted persons are the same*. In practice one has to differentiate between gallery and probe sets. The gallery provides a list of known people, *i.e*. faces with associated identity labels, from which the face recognition system is supposed to derive individual and discriminative features as an abstract mathematical representation of the identity. From a probe (query) face, the same features are extracted and compared to those in the gallery. The result is a similarity score, based on which the identity can be determined (identification) respectively confirmed (verification). Unfortunately the task is complicated by the fact that, in real-world scenarios, where the gallery and probe image have been acquired under different conditions, faces may exhibit dramatic intra-subject appearance variations. The main reason is the relatively complex 3D structure which, combined with changed pose and illumination, affects a face's shading, partial visibility and self-shadowing. Other internal factors of change are expressions (non-rigid motion), aging and presence or absence of facial hair. Besides illumination, occlusions and background clutter are common external sources of image variation. In particular

occlusions can be a significant handicap, for robust recognition and face analysis in general, because the location, extent and appearance of the affected image areas is usually impossible to predict. Current systems have advanced to be fairly accurate only under constrained scenarios. That means they are at best able to cope with a few less pronounced variations simultaneously without major degradation in matching quality. The ultimate challenge of face recognition systems is to find features that are invariant with respect to all the extrinsic imaging parameters and truly capture a person's identity.

A very promising way to master the problem are generative face models, such as the 3D Morphable Model (3DMM). This approach represents a face as a linear combination of facial prototypes between which a dense correspondence is defined and which span the whole object class. In association with additional parameters for pose and illumination, realistic artificial views of a modelled face can be created. Given an image, the depicted face is reconstructed as a 3D model by means of an iterative analysis-by-synthesis procedure which adjusts the model's parameters such that the generated 2D projection matches the input image. In the result the estimated linear model coefficients capture specific facial properties of shape and texture. These can be used as features for identification, while the rendering parameters independently describe the extrinsic conditions. A 3DMM reconstruction also establishes a dense mapping between the image pixels and the model's vertices. Along with the possibility to synthesize realistic novel views of a face, this capability renders the 3DMM a powerful tool for several applications, beyond mere face recognition.

However, the method also has its drawbacks. One of them is a sensitivity to outliers. In terms of the 3DMM, outliers are observations which lie outside the range of modeled and thereby expected appearance. This notion comprises in particular facial occlusions such as glasses, clothing and hairstyle. But also an open mouth or raised eyebrows may be included in this definition, since the original 3DMM approach does not support facial expressions. In the presence of outliers in the input face, a reconstruction is corrupted because of poorly estimated model parameters. The problem is reflected in degraded visual quality and under severe conditions in bad correspondence. The impairment grows with the affected areas and increasing divergence from normal facial appearance. Without going into depth (for now), there are two main causes for this behaviour, both of which are intrinsic to the model's design and not easily eliminated. Hence, the question arises, whether there are alternative and ideally non-intrusive methods to compensate for the lack of robust occlusion/outlier handling. That is the starting point for this thesis.

1.1 Motivation & Overview

Our work was motivated by two applications which utilize the aforementioned 3DMM capabilities. The first application represents a novel approach to face recognition [PV07]. Our aim is to localize irregularities like moles or birthmarks in facial skin and to use their individually characteristic configuration across a face for identification. Here the 3DMM provides the means to compare feature point locations between faces in different images. One central

problem of the idea is that the desired features only have a simple blob-like appearance. Without constraining the search to regions that actually display skin, the corresponding feature detector would report false positives all over the face. Therefore a binary segmentation of the face into skin and non-skin areas is required. In this case the non-skin part comprises hair (beard, hairstyle, eyebrows), eyes, nostrils, lips and potential occlusions, since all of these elements may exhibit blob-like structures at various scales. As we detail later, the 3DMM cannot directly and reliably deliver such a segmentation, partly due to the outlier problematic, so that a customized solution must be found.

The second application represents a type of high-level photo manipulation, called "*Face Exchange*" (*e.g.* [BSVS04]). Given an image, the goal is to replace the depicted face with that of another person or with modified features, while retaining certain aspects of the original image like the hairstyle, clothing and the scene background. For this task the 3DMM is used to represent faces and the extrinsic parameters independently, such that they can be re-synthesized under different scene conditions. In order to obtain convincing outputs, an additional segmentation into foreground and background layers is needed to correctly handle changing object occlusions which appear as result of the manipulation. As opposed to the hard skin segmentation in the first application, this scenario demands for a soft image decomposition. That means each layer and pixel is associated with a "coverage" value that defines its opacity. Only with that, one can seamlessly blend multiple layers, in particular hair, to create photo-realistic manipulated face images. Also here the 3DMM is not suited to directly determine the occluded areas.

Despite completely different objectives, it turns out that both applications share a common problem. The first one depends on an explicit labeling of all pixels not belonging to facial skin, the second involves special treatment of facial occlusions, which also mostly affect skin regions. Apparently both tasks demand for a procedure that separates outliers respectively occlusions from skin. Therefore, this work addresses the outlier problematic as a classical binary image segmentation problem. To our knowledge such an approach is unprecedented in the domain of face analysis applications. In order to be useful for the two described applications, our solution has to meet certain requirements. Most importantly, the segmentation should be performed automatically. This is essential for the mole detection and recognition scenario, where hundreds of images from a large face database have to be processed. A second request is support for gray scale images, which is a novelty among techniques dealing with skin detection/segmentation. While the conventional methods rely entirely on discrimination of color information per pixel, the gray scale setting demands for more elaborate texture based algorithms. Naturally the segmentation has also to be accurate. These design goals contribute considerably to the complexity of the segmentation task. We will show that, although established and well understood off-the-shelf methods are employed, in order to obtain high quality solutions, additional effort has to be made to adequately pre-process the face images and to develop the "right" interplay between the specialized algorithms.

The expertise concerning face representation is provided by the 3DMM. It is a key component in this work, since all applications utilize some of the model's capabilities. Chapter 2 introduces the basics of the face model and of the ded-

icated fitting algorithm. This work touches two distinct topics in computer vision, namely face analysis and segmentation. Since there is only little intersection between the methodologies adopted in these fields, Chapter 3 provides a brief review of segmentation techniques, with strong emphasis on graph-based methods, which is addressed to readers who are less familiar with this topic. Chapter 5 details the *Spectral Matting* technique for soft segmentation, which is required to compute opacity values of layers in the *Face Exchange* application. The main contribution of this thesis is presented in Chapter 4. We develop a robust skin segmentation procedure by extending the *GrabCut* approach with automatic initialization, specifically designed texture features and a novel method to reduce the influence of illumination on these features. Another novelty, presented in Chapter 6, is the aforementioned identification method that exploits small mole-like details in facial skin. Furthermore, this chapter explains how the skin segments can be used to realize 3DMM fittings without corruption by outliers and how they facilitate an automatic and artefact-free *Face Exchange*. Finally, Chapter 7 concludes our work.

1.2 Related Work

In the face recognition community it is well known that in holistic representation schemes, the changes of facial appearance, as caused by illumination, non-rigid motion and occlusion, affect the entire set of feature descriptors, even if the actual image variations are local. The classical Eigenfaces [TP91] approach, like most PCA-based methods, is a perfect example for this lack of robustness. One way to deal with the issue, is to build more complex models that incorporate the sources of possible appearance variations. This has been done for illumination and facial expression. In practically relevant scenarios, however, occlusions are merely a spatially coherent form of outliers and elude such a systematical analysis.

A widespread paradigm for robust recognition relies on sparse representation. The underlying argument is, that local features, computed only from a fraction of the image pixels, are less likely to be corrupted by occlusions than holistic features. An early attempt at deriving local features [BP93] was purely based on measured geometric configurations as the size of facial organs and their relative positions. More recently, EBGM [WFKvdM97] successfully combined geometric properties with image based features. This method represents faces as planar labeled graphs. Their nodes are placed consistently on certain landmark points and associated with bundles of Gabor Wavelet responses, called *jets*, which are extracted from the underlying image at the node locations. Graphs from multiple (training) faces are stacked into a *bunch graph*. This structure can be matched to novel faces by constrained geometrical transformations of the nodes and simultaneous combinatorial selection of the best fitting jet for each node respectively. Within the appearance based domain various approaches, such as SPCA, ICA and LNMF, adopt sparseness by projecting the face image into subspaces with locally concentrated bases. The idea of SPCA [CJ01] is to create a sparse basis only by transforming a conventional PCA basis with a suitable rotation matrix. *I.e.* the orthonormality property is retained. Driven

by a simple cost function, the algorithm iteratively rotates hyperplanes in the principal subspace such that directions with little correlation in the data set become maximally sparse. This comes at the cost of introducing correlations in the output coefficients. The ICA technique [HO00] is a generalization of PCA that decorrelates the higher order statistics in addition to second-order moments, and treats the input face images as linear combination of statistically independent basis images [BLS98, DBBB03]. The sparseness of the ICA basis images results as side-effect of the employed non-Gaussianity maximization. The rationale behind LNMF [LHZC01] is that for certain processes a non-negative representation is "natural". For example, gray scale images or firing rates of neural cells have non-negative intensities. In contrast to ICA, LNMF explicitly seeks a decomposition of such data into non-negative factors with additional constraints for locality and orthogonality.

Another form of locality is realized through component-based approaches, where the standard holistic techniques are only applied to certain parts of the face. Usually the motivation for such representations is to accommodate pose variations, quite similar to EBGM, by allowing a flexible geometrical relation between the individually modelled components, *e.g.* [PMS94, HSP07]. A popular probabilistic approach to the part-based concept is proposed in [Mar02]. The authors divide a face into six fixed elliptic shaped local areas. From the training faces all patches within one region are grouped into a corresponding eigenspace which is modelled by means of a Gaussian Mixture Model. In order to compensate for localization errors (the local areas are static) additional virtual training samples are generated using an image perturbation method. In the identification stage, the test images are also divided into the same six areas and are projected onto the above computed eigenspaces respectively. A global probability of a test face is computed by adding all local probabilities as defined by the Gaussian distribution.

The last category of algorithms, we want to consider here, is based on classical robust regression techniques [Ste99], such as random sampling (*e.g.* RANSAC) or M-estimators. Although it has not been demonstrated on faces, Leonardis and Bischof [LB00] proposed an interesting general object recognition approach within the eigenimage framework. Instead of computing coefficients through direct projection of the data, they use a random subset of pixels to robustly generate a representation hypothesis. This is done by iteratively examining the error distribution and discarding a certain fraction of pixels with the highest errors, thus many outliers are rejected. Moreover, several hypotheses are created and then selected according to the MDL principle. In connection with the 3DMM an alternative to the original Stochastic Newton Descent fitting algorithm was developed [RV03, Rom05]. While the prime focus was to improve efficiency, this work also addressed the outlier problematic by introducing an iteratively reweighted least squares scheme into the cost function with the aim of limiting the influence of large residuals. Last but not least, most methods, including those mentioned above, are concerned with robustness in the recognition stage, *i.e.* they assume that the images in the training set are "ideal" and that the visual model is essentially correct. De la Torre and Black [lTB01] presented a method for robust PCA learning, also by incorporating the concept of M-estimation into the definition of the reconstruction error, which leads to an iterative minimization algorithm.

Chapter 2

Face Representation with the 3D Morphable Model

2.1 Model Construction

The 3D Morphable Model (3DMM) is constructed from a set of 200 example faces, provided in form of 3D laser scans. These samples capture (and also limit) the variation of facial attributes, which the model is able to represent. A semi-automatic procedure first removes scanning artefacts and unwanted data like the back of the head and aligns the faces in 3D. After this preprocessing the central step of model creation is to establish a dense point-to-point correspondence between each face and a single arbitrary reference face. The laser scanner records facial data as radius r (*i.e.* depth) and RGB color in a cylindrical representation $\boldsymbol{I}(h,\phi) = (r(h,\phi), R(h,\phi), G(h,\phi), B(h,\phi))^T$ using respectively 512 vertical and angular sampling steps. Correspondence between two scans is defined through a vector field $\boldsymbol{v}(h,\phi) = (\Delta h(h,\phi), \Delta\phi(h,\phi))^T$ such that each point in the first scan $\boldsymbol{I}_1(h,\phi)$ corresponds to the point $\boldsymbol{I}_2(h+\Delta h, \phi+\Delta\phi)$ in a second scan. A modified optical flow algorithm [BV03a] is used to estimate $\boldsymbol{v}$.

2.1.1 Correspondence using Optical Flow

The majority of optical flow methods adopt the notion of brightness constancy, *i.e.* in a gray scale image sequence $I(x,y,t)$ pixels are assumed to conserve their intensity between frames: $I(x(t),y(t),t) = I(x(t_0),y(t_0),t_0)$. In differential form this yields the following condition on the velocity components $v_x = \frac{dx}{dt}$ and $v_y = \frac{dy}{dt}$:

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0. \tag{2.1}$$

Equation (2.1) is under-determined, therefore an additional constraint [LK81] is introduced which assumes that the flow is constant within a small neighborhood R (generally a 5×5 window) of each pixel. A unique least-squares solution

is then obtained by minimizing at each point (x_0, y_0) the expression:

$$E(x_0, y_0) = \sum_{x,y \in R(x_0,y_0)} \left(\frac{\partial I(x,y)}{\partial x} v_x + \frac{\partial I(x,y)}{\partial y} v_y + \frac{\partial I(x,y)}{\partial t} \right)^2 \tag{2.2}$$

For pairs of images $\boldsymbol{I}_1$, $\boldsymbol{I}_2$ the partial derivatives are approximated by finite differences, in particular $\frac{\partial I}{\partial t} = \Delta \boldsymbol{I} = \boldsymbol{I}_2 - \boldsymbol{I}_1$. In order to also capture larger displacements a coarse-to-fine strategy can be implemented by applying the described method on a Gaussian pyramid, starting from the lowest resolution and refining the estimated flow on each subsequent level.

For the 3D laser scans this procedure is generalized to multi-channel data by replacing the squared bracket in (2.2) with a weighted norm on vector-valued pixels, $\|\boldsymbol{I}(h,\phi)\|^2 = w_r r^2 + w_R R^2 + w_G G^2 + w_B B^2$:

$$E = \sum_{h,\phi \in R} \left\| \frac{\partial \boldsymbol{I}(h,\phi)}{\partial h} v_h + \frac{\partial \boldsymbol{I}(h,\phi)}{\partial \phi} v_\phi + \Delta \boldsymbol{I}(h,\phi) \right\|^2 . \tag{2.3}$$

The heuristically chosen weights compensate for the different value ranges between the channels and control the influence of shape versus texture. The optical flow algorithm [BV03a] includes two more enhancements. A Laplacian pyramid is used to improve correspondence between scans which differ in overall size or brightness. To obtain reliable results even in regions of the face with no salient structures, a specifically designed smoothing and interpolation algorithm is added to the matching procedure on each level of resolution.

2.1.2 Face Vectors

After all scans have been registered to a common reference frame, they contain the same number $n = 75972$ of vertices and each vertex (ideally) represents the same "semantic" location in every sample face. The 3D Cartesian coordinates and associated colors (RGB-tuples) of a face's vertices are stored as shape vector $\boldsymbol{S} \in \mathbb{R}^{3n}$ respectively texture vector $\boldsymbol{T} \in \mathbb{R}^{3n}$, (for the remainder of the chapter we refer by the term sample face/scan to this form of representation):

$$\boldsymbol{S} = vec \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{pmatrix}, \quad \boldsymbol{T} = vec \begin{pmatrix} r_1 & g_1 & b_1 \\ r_2 & g_2 & b_2 \\ \vdots & \vdots & \vdots \\ r_n & g_n & b_n \end{pmatrix}. \tag{2.4}$$

The reference frame defines a 2D parametrization for those vectors. It thereby also provides a natural way to derive a triangulation of the vertices required to render the faces. By means of the dense correspondence it is possible to combine properly registered scans to produce previously unseen faces and further to "learn" how to generalize from a few samples to the whole object class of human faces.

An important observation is that a linear combination of two registered face scans again represents a human face, *i.e.* given two sample shapes $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, it

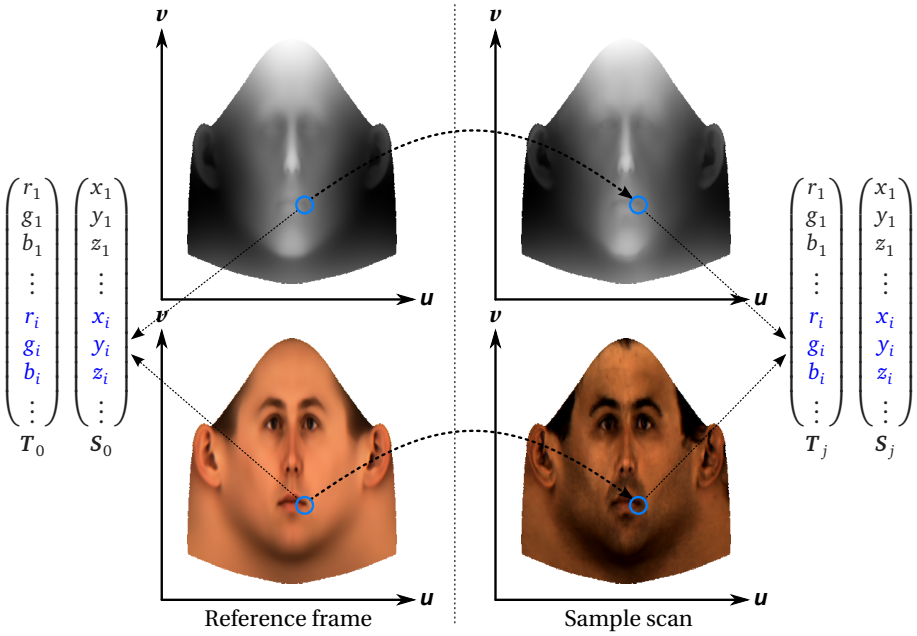


Figure 2.1: By means of the reference frame a dense correspondence between 3D scans is established that relates a vertex $(x_i, y_i, z_i; r_i, g_i, b_i)$ from different scans to the same semantic location within the face. The reference also defines a common 2D parametrization for the sample shape/texture vectors.

is possible to create a *morph*:

$$\boldsymbol{S} = (1-a)\cdot \boldsymbol{S}_1 + a\cdot \boldsymbol{S}_2 \quad \text{with} \quad 0 \le a \le 1. \tag{2.5}$$

By repeatedly morphing additional shapes, Equation (2.5) can be generalized to create new faces from any convex combination of m samples:

$$\boldsymbol{S} = \sum_{i=1}^{m} a_i \cdot \boldsymbol{S}_i \quad \text{with} \quad 0 \le a_i \le 1, \sum_{i=1}^{m} a_i = 1. \tag{2.6}$$

If one considers $\mathcal{V} = \mathbb{R}^{3n}$ as the space of all possible 3D objects composed of n vertices, then apparently facial shapes populate only an affine subspace, spanned by the vectors $\boldsymbol{S}_i$, with very low dimensionality ($\le m-1$) compared to $\mathcal{V}$. This property analogously holds for the texture vectors which can be morphed in the same manner, either in combination with or independent of the shape, as shown in Figure 2.2. The linearity assumption imposed in the construction in Equation (2.6) does not necessarily reflect the true nature of a "face space", which is unknown. However, due to the high dimensionality of $\mathcal{V}$ compared to the small number of available face samples it would be very difficult and unreasonable to conjecture on a more complex structure of this space.

2.1.3 Statistical Analysis

In Equation (2.5) and (2.6) the composition of faces with hard constrained coefficients is arbitrary. For one the crossover between plausible faces and exaggerations is not clearly defined and also varies individually. Secondly those limits are mathematically circumstantial to handle. It would be better to model faces in a probabilistic framework. The simplest way to do that would be to discard the constraints and assume that shape and texture are distributed uniformly. This, however, does not lead to a realistic model, since it allows the generation of very unlikely faces. Instead the 3DMM is based on the presumption that the underlying data follows a Gaussian distribution. In the following we briefly recapitulate the construction of the model, exemplary for shapes, using principal-component analysis (PCA) as a statistical analysis tool and we discuss some of it's properties.

Principal-component analysis (also referred to as *Karhunen-Loève Transform*, KLT) is commonly motivated as the search for a transformation on a group of random variables, such that the transformed variables are decorrelated. Mathematically this is achieved by using the eigenvectors of the sample covariance matrix as new basis for representation of the data set. In order to apply PCA to our given ensemble of training shapes (with $3n$ random variables and m samples) we calculate the mean

$$\bar{\boldsymbol{s}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{S}_i, \tag{2.7}$$

subtract $\bar{\boldsymbol{s}}$ from the samples and stack the resulting vectors into a data matrix

$$\boldsymbol{x}_i = \boldsymbol{S}_i - \bar{\boldsymbol{s}}, \quad \boldsymbol{X} = \begin{pmatrix} | & | & & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_m \\ | & | & & | \end{pmatrix} \in \mathbb{R}^{3n\times m}. \tag{2.8}$$

Then the covariance matrix can be written as

$$\boldsymbol{C} = \frac{1}{m}\boldsymbol{X}\boldsymbol{X}^T. \tag{2.9}$$

The high dimensionality of $\boldsymbol{C}$ forbids direct computation [1] of its eigenvectors. Instead, first an "economic" form of singular-value decomposition (SVD) of the data matrix is computed. It accounts for the fact that $rank(\boldsymbol{C}) < m \ll 3n$ and results in orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{3n\times m}$, $\boldsymbol{V} \in \mathbb{R}^{m\times m}$ and diagonal matrix $\boldsymbol{\Lambda} = diag(\lambda_1,\ldots,\lambda_m) \in \mathbb{R}^{m\times m}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T. \tag{2.10}$$

With that the covariance matrix can be expressed as:

$$\boldsymbol{C} = \frac{1}{m}\boldsymbol{X}\boldsymbol{X}^T = \frac{1}{m}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Lambda}^T\boldsymbol{U}^T = \frac{1}{m}\boldsymbol{U}\boldsymbol{\Lambda}^2\boldsymbol{U}^T. \tag{2.11}$$

The column vectors of $\boldsymbol{U}$, called *principal components* are mutually orthonormal (by definition of SVD) and Equation (2.11) shows that they are the eigenvectors

[1] Already storage of $\boldsymbol{C}$ would require $(3n)^2 \cdot sizeof(\text{float}) \approx 200\text{Gb}$ of memory.

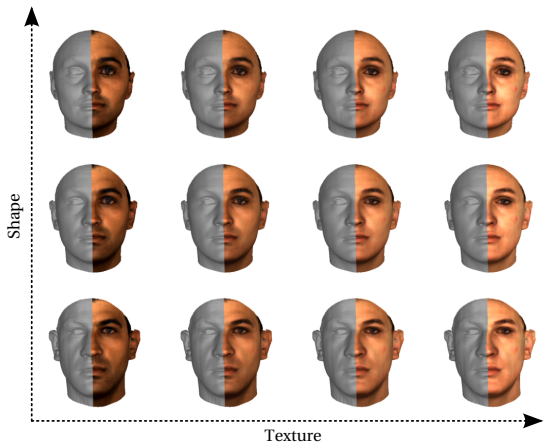


Figure 2.2: Illustrates the ability to morph between properly registered 3D scans independently for shape and texture.

of $\boldsymbol{C}$, with corresponding eigenvalues λ_i^2/m. If we define by $\boldsymbol{B} = \boldsymbol{U}^T\boldsymbol{X}$ the projection of all mean-free shape vectors into the space spanned by the principal components, it is easily verified that the resulting coordinates are decorrelated:

$$cov(\boldsymbol{B}) = \frac{1}{m}\boldsymbol{B}\boldsymbol{B}^T = \frac{1}{m}\boldsymbol{U}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{U} = \frac{1}{m}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Lambda}^2\boldsymbol{U}^T\boldsymbol{U} = \frac{1}{m}\boldsymbol{\Lambda}^2. \tag{2.12}$$

We denote the principal component vectors by $\boldsymbol{s}_i$ and the variances of the projected data $\boldsymbol{B}$ by $\sigma_i^2 = \lambda_i^2/m$. Without loss of generality we can assume that the λ_i and thereby also the σ_i are sorted in descending order,

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M,$$

with the corresponding $\boldsymbol{s}_i$ arranged accordingly within $\boldsymbol{U}$. It can be shown that of all subspaces, the one spanned by the principal components minimizes the mean square error between a vector $\boldsymbol{x}$, sampled from the same population as the $\boldsymbol{x}_i$ in Equation (2.8), and its approximation $\tilde{\boldsymbol{x}}$ in this subspace. In particular, if $\boldsymbol{x} = \sum_{i=1}^{m} b_i\boldsymbol{s}_i$ and $\tilde{\boldsymbol{x}}$ is the projection of $\boldsymbol{x}$ onto the first k principal components, then:

$$\begin{aligned} E\left[\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2\right] &= E\left[\left\|\sum_{i=k+1}^{m} b_i\boldsymbol{s}_i\right\|^2\right] = E\left[\sum_i\sum_j (b_i\underbrace{\boldsymbol{s}_i)^T(\boldsymbol{s}_j}_{\delta_{ij}}b_j)\right] \\ &= \sum_{i=k+1}^{m} E\left[b_i^2\right] \overset{(2.12)}{=} \sum_{i=k+1}^{m} \sigma_i^2. \end{aligned} \tag{2.13}$$

This property shows that the described subspace is optimal (for the given training data) for dimensionality reduction, since the last $k+1$ components, which are left out in an approximation, are the ones with smallest variance and therefore carry the least information.

As mentioned above, the 3DMM approach assumes that the facial shape comes from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Its maximum-likelihood parameter estimates for the available samples are $\boldsymbol{\mu} = \bar{\boldsymbol{s}}$ and $\boldsymbol{\Sigma} = \boldsymbol{C}$. With that the probability distribution for shape vectors $\boldsymbol{x} \in \mathbb{R}^{3n}$ can be expressed:

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^{3n}\,|\boldsymbol{C}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{s}})^T \boldsymbol{C}^{-1}(\boldsymbol{x}-\bar{\boldsymbol{s}})}. \tag{2.14}$$

Inversion of the covariance matrix is realized via Equation (2.11) which implies that directions outside the principal component subspace $span\{\boldsymbol{s}_i\}$ attain a probability of zero. For mean-free shape vectors $\boldsymbol{x} \in span\{\boldsymbol{s}_i\}, \boldsymbol{x} = \boldsymbol{U}\boldsymbol{b}$ the exponent in (2.14) can be simplified,

$$\left\langle \boldsymbol{x}, \boldsymbol{C}^{-1}\boldsymbol{x} \right\rangle = \left\langle \boldsymbol{U}\boldsymbol{b}, \left(\boldsymbol{U}\tilde{\boldsymbol{\Lambda}}^{-1}\boldsymbol{U}^T\right)\boldsymbol{U}\boldsymbol{b} \right\rangle = \left\langle \boldsymbol{b}, \tilde{\boldsymbol{\Lambda}}^{-1}\boldsymbol{b} \right\rangle = \sum_i \frac{b_i^2}{\sigma_i^2}, \tag{2.15}$$

and so the pdf basically reduces to a product of one-dimensional normal distributions directly parameterized by the vector's coefficients:

$$p(\boldsymbol{x}) = p(\boldsymbol{b}) \sim e^{-\frac{1}{2}\sum_i \frac{b_i^2}{\sigma_i^2}} = \prod_i e^{-\frac{1}{2}\frac{b_i^2}{\sigma_i^2}}. \tag{2.16}$$

The previous analysis applies analogously to the scanned textures and is performed independently of the shapes. Now, instead of interpreting a (novel) face $(\boldsymbol{S}, \boldsymbol{T})$ as a morph between examples, the face is encoded as linear combination of m_S principal components $\boldsymbol{s}_i$ for shape and m_T principal components $\boldsymbol{t}_i$ for texture $(m_S, m_T \leq m-1)$:

$$\boldsymbol{S} = \bar{\boldsymbol{s}} + \sum_{i=1}^{m_S} \alpha_i \boldsymbol{s}_i, \quad \bar{\boldsymbol{s}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{S}_j, \quad p(\boldsymbol{S}) \sim e^{-\frac{1}{2}\sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2}} \tag{2.17}$$

$$\boldsymbol{T} = \bar{\boldsymbol{t}} + \sum_{i=1}^{m_T} \beta_i \boldsymbol{t}_i, \quad \bar{\boldsymbol{t}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{T}_j, \quad p(\boldsymbol{T}) \sim e^{-\frac{1}{2}\sum_i \frac{\beta_i^2}{\sigma_{T,i}^2}} \tag{2.18}$$

Figure 2.3 shows the first three principal components of the shape and texture model as well as their influence on facial appearance. The latter is visualized by adding or subtracting a multiple of the respective component to the average face while leaving all other modes unchanged. Based on these images we make two observations. First, some principal components correspond to meaningful facial attributes, *e.g.* $\boldsymbol{s}_1$ and $\boldsymbol{t}_1$ appear to model gender, $\boldsymbol{s}_2$ represents fullness of a face and $\boldsymbol{t}_3$ seems to affect the hairline and overall skin tone. This no longer holds for components with smaller variance. Secondly, the components have global support. That means, changing any of the coefficients α_i or β_i will have an effect on every vertex of the face. Conversely, encoding only a local change in a face, still requires the adaptation of all coefficients of the respective model.

This last-mentioned property is desirable from a theoretical point of view, as it ensures that only uncorrelated sets of features can be altered individually.

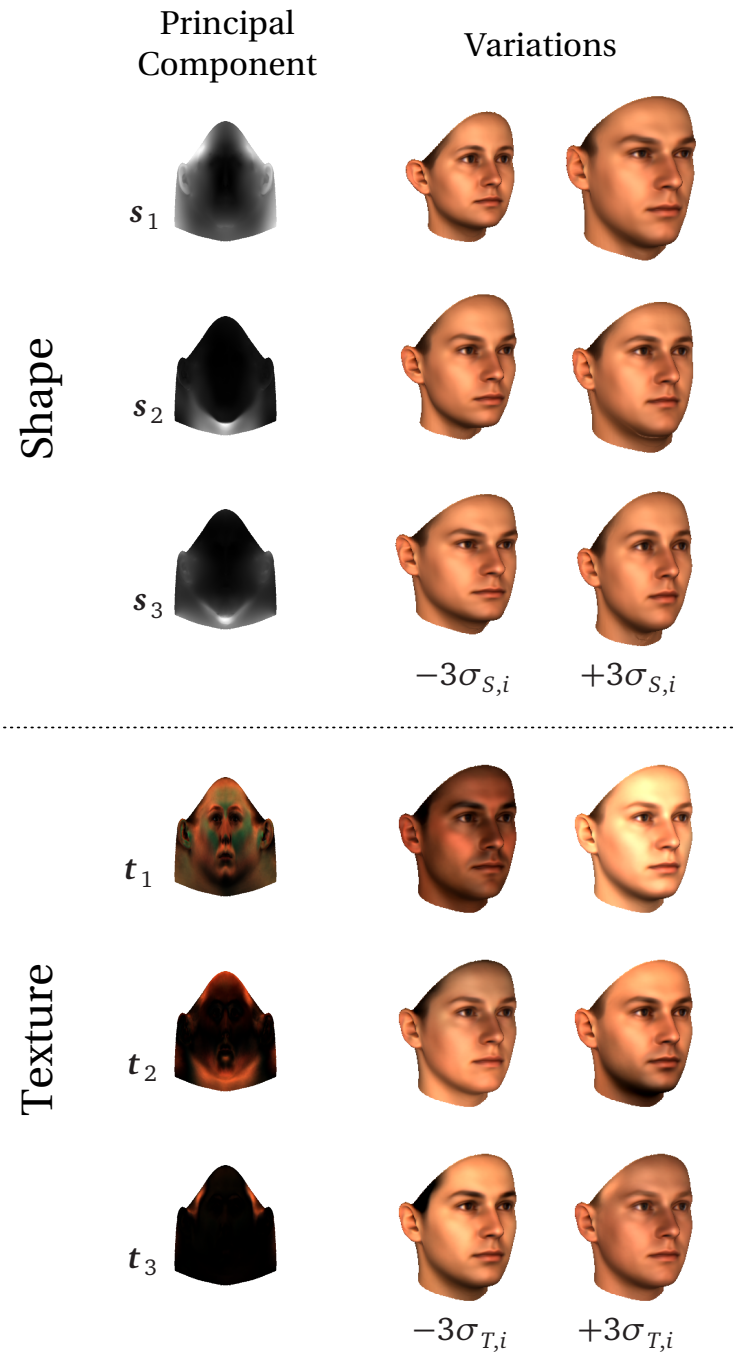


Figure 2.3: First three principal components of the shape and texture model. The component vectors are visualized (shape as range, texture as RGB) in the reference frame, normalized to $[0,1]$. The right columns show their effect on facial appearance by subtraction/addition of the respective component, scaled by a multiple of its corresponding standard deviation, on the average face $(\bar{\boldsymbol{s}}, \bar{\boldsymbol{t}})$.

Thereby unnatural constellations, *e.g.* enlarging only one side of a face, are prevented. However, in practice the number of training faces is far too small to represent the full spectrum of facial diversity and thus to model the true dependencies between local facial regions. To overcome this limitation the Morphable Model is segmented (in the reference domain) into four regions: eyes, nose, mouth and surrounding area. Each of these regions can be encoded by a different set of model coefficients. The results are assembled using a multi-scale blending procedure [BA85].

2.2 3D Morphable Model Fitting

The Morphable Model can be used to estimate the 3D structure of a novel face from a single photograph. To accomplish this goal, an iterative analysis-by-synthesis scheme is used to adapt the model's parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ such that the assembled shape and texture, projected into the image frame, match the depicted face. The reconstruction obtained by this procedure also provides appearance estimates for facial regions which are occluded in the photograph.

2.2.1 Rendering Parameters

In order to obtain photo realistic renderings of face models additional parameters are required. We distinguish between two sets of image formation (rendering) parameters:

- **Pose Parameters**
  The object-centered vertex coordinates $\boldsymbol{v}_k = (x_k, y_k, z_k)^T$ are mapped to a position relative to the virtual camera (located at the origin) subject to the rigid transformation:

  $$(w_{x,k}, w_{y,k}, w_{z,k})^T = \boldsymbol{R}_\gamma \boldsymbol{R}_\theta \boldsymbol{R}_\phi \boldsymbol{v}_k + \boldsymbol{t}_w. \tag{2.19}$$

  The angles ϕ and θ define in-depth rotations about the vertical (*yaw*) and horizontal (*pitch*) axis, γ represents a rotation in the image plane (*roll*). $\boldsymbol{t}_w$ acts as 3D translation. The world coordinates are then perspectively projected to the image plane, parameterized by the cameras focal length f and the principal point (P_x, P_y) (position of optical axis in the image plane):

  $$p_{x,k} = P_x + f\frac{w_{x,k}}{w_{z,k}}, \quad p_{y,k} = P_y + f\frac{w_{y,k}}{w_{z,k}}. \tag{2.20}$$

- **Illumination Parameters**
  Illumination of the 3D model requires surface normals $\boldsymbol{n}_k$ to be defined per-vertex. The triangulation (obtained in the reference frame) allows straight-forward computation of one normal per triangle. Vertex normals are then simply averaged from the normals of adjacent triangles.
  The fitting procedure employs the standard Phong illumination model, see for example [FvDFH97] and [Wat93], assuming only one directed light source with ambient ($\boldsymbol{L}_{amb}$), diffuse and specular ($\boldsymbol{L}_{dir}$) intensities and

hard cast shadows. The lights' direction vector $\boldsymbol{l}$ is specified [2] through azimuth and elevation angles θ_l and ϕ_l:

$$\boldsymbol{l} = \left(\sin(\phi_l)\cos(\theta_l), \sin(\theta_l), \cos(\phi_l)\cos(\theta_l)\right)^T. \tag{2.21}$$

For each vertex with color $\boldsymbol{c}_k = (r_k, g_k, b_k)^T$ the amount of reflected light is computed as:

$$\boldsymbol{L}_k = \boldsymbol{c}_k \cdot \left(\boldsymbol{L}_{amb} + \boldsymbol{L}_{dir}\langle \boldsymbol{n}_k, l\rangle\right) + k_s \cdot \boldsymbol{L}_{dir}\left\langle 2\langle \boldsymbol{n}_k, l\rangle \boldsymbol{n}_k - l, -\boldsymbol{v}_k\right\rangle^{\alpha} \tag{2.22}$$

The material parameter α influences the "hardness" of the specular reflection. Finally the lit mesh is rasterized, using a z-buffer for hidden surface removal and Gouraud shading to interpolate vertex colors inside triangles. After rendering a global color transformation is applied to the image pixels $\boldsymbol{I}_{r,g,b}(x,y)$ to compensate for scene specific conditions like tint or contrast. An important application of this measure is to facilitate matching of the face model to gray scale images which otherwise would require modification of the underlying sample textures and retraining of the model. The color adjustment includes offset (o_r, o_g, o_b), gain (g_r, g_g, g_b) and contrast c. With the luminance of a pixel denoted by $Y = 0.3\boldsymbol{I}_r + 0.59\boldsymbol{I}_g + 0.11\boldsymbol{I}_b$, the transformation is defined individually for each color channel as:

$$\tilde{\boldsymbol{I}}_{r,g,b}(x,y) = g_{r,g,b}\left(c\boldsymbol{I}_{r,g,b}(x,y) + (1-c)Y(x,y)\right) + o_{r,g,b} \tag{2.23}$$

In all there are 22 parameters, concatenated in a vector $\boldsymbol{\rho}$, which control the rendering output.

2.2.2 Formulation as Minimization Problem

Given an input image $\boldsymbol{I}_{input}(x,y)$, a 3D reconstruction is obtained by searching for the most likely model and rendering parameters that can explain the observed scene. This approach is formally expressed as maximization of the conditional probability

$$p\left(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho} | \boldsymbol{I}_{input}(x,y)\right). \tag{2.24}$$

According to Bayes rule, and under the presumption that the parameter sets are independent, it is equivalent to maximize

$$p\left(\boldsymbol{I}_{input}(x,y) | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}\right) \cdot P(\boldsymbol{\alpha}) \cdot P(\boldsymbol{\beta}) \cdot P(\boldsymbol{\rho}). \tag{2.25}$$

The prior probabilities $P(\boldsymbol{\alpha})$ and $P(\boldsymbol{\beta})$ were already estimated by PCA (2.17), (2.18). For the ρ_i the fitting process assumes individual normal distributions with ad-hoc values for mean and variance. The leftmost term in Equation (2.25) models the deviation per pixel between the input image and the image $\boldsymbol{I}_{model}(x,y)$ synthesized from the parameters, again assuming independent Gaussian noise with standard deviation σ_I. Altogether, maximization of the posterior probability (2.24) can be reformulated as minimization problem with the cost function:

$$E(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}) = E_I + \eta \cdot E_P \tag{2.26}$$

[2] We use a right-hand oriented coordinate system with y-axis pointing up.

where

$$\begin{aligned} E_P &= -2\log\left(P(\boldsymbol{\alpha})P(\boldsymbol{\beta})P(\boldsymbol{\rho})\right) \\ &= \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{(\rho_i - \bar{\rho}_i)^2}{\sigma_{R,i}^2} \end{aligned} \tag{2.27}$$

and

$$\begin{aligned} E_I &= -2\log p\left(\boldsymbol{I}_{input}(x,y)|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\rho}\right) \\ &= \frac{1}{\sigma_I^2}\sum_{x,y}\sum_{\lambda=r,g,b}\left(I_{\lambda,input}(x,y) - I_{\lambda,model}(x,y)\right)^2 \end{aligned} \tag{2.28}$$

E_I describes the Euclidean distance between two color images while E_P is an expression for the plausibility of the parameter estimates. The factor η is used to bias the influence of the priors, either towards more likely reconstructions, closer to the average face, or towards more accurate fittings which might exhibit artefacts when viewed under different pose or illumination.

The cost function is minimized with a stochastic version of Newton's method [JP98] which evaluates E and it's analytical derivatives in each iteration only for a small random subset of pixels resp. triangles. Additionally, to avoid local minima, a coarse-to-fine strategy is employed. In the beginning only a few coefficients are optimized and η is set to put high weight on the prior probabilities (2.27). Later, the number of fitted principal components is increased, the bias is changed in favor of matching quality and in the final iterations the eye, nose, mouth and surround segments are optimized individually, while the rendering parameters remain unchanged.

2.2.3 Landmark Assisted Fitting

An extension [BV03a] to this algorithm also incorporates externally defined feature point locations, like the tip of the nose or the corners of the eyes, to improve fitting performance. The cost function is augmented with a term, that encodes the discrepancy between the locations of user/software provided feature coordinates $(q_{x,k}, q_{y,k})$ and the projected locations $(p_{x,k}, p_{y,k})$ of the corresponding vertices, based on the current pose and shape parameters:

$$E(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\rho}) = E_I + \eta \cdot E_P + \sum_k \left\| \begin{pmatrix} q_{x,k} \\ q_{y,k} \end{pmatrix} - \begin{pmatrix} p_{x,k} \\ p_{y,k} \end{pmatrix} \right\|^2 . \tag{2.29}$$

Provided that the landmark positions are determined manually, they represent the only available ground-truth information in the fitting process. In such cases the average distance between the $\boldsymbol{q}_k$ and $\boldsymbol{p}_k$ can be utilized as an indicator for the quality of the 3D reconstruction in terms of correspondence, *e.g.* to obtain a suitable search range when matching local point features between faces. In the following we shall refer to this measure as *alignment error*.

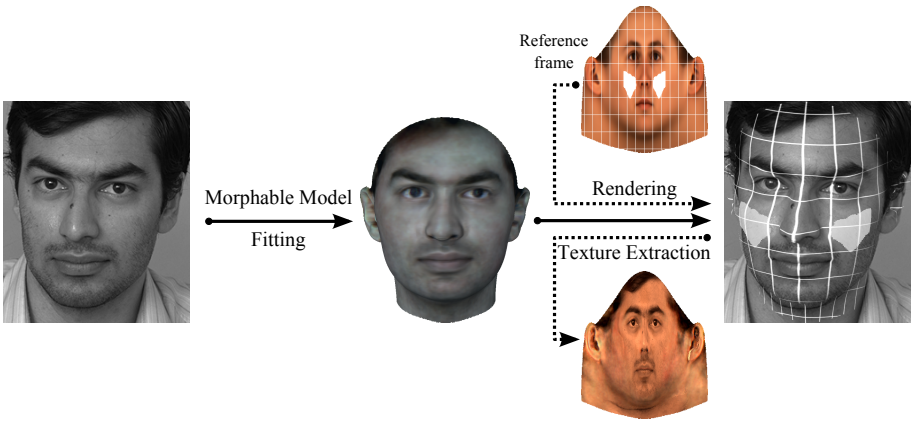


Figure 2.4: Illustration of the point/region mapping technique between the 3DMM reference coordinates (sketched as white grid lines) and image coordinates. In this example a binary mask, marking the cheeks, was selected in the reference frame. Via a fitting of the 3DMM this mask can be mapped to the photograph and vice versa the true facial texture can be mapped to the model.

2.2.4 Dense Mapping to/from Image

Amongst other outcomes, the fitting algorithm realizes a dense correspondence between the fixed reference frame of the model and the input image frame, whereby the 3DMM reconstruction serves as intermediary between the two distinct coordinate systems. A mapping from model to image is realized by projecting vertices, either as points or as triangular mesh, via the estimated shape and pose parameters to their corresponding locations in the image. Standard rendering techniques then provide the means to determine their visibility. With this mapping it is possible to transfer pre-selected sets of vertices to the image in order to mark specific facial areas, for example the facial organs or the cheeks, which is illustrated in Figure 2.4. The reverse mapping can be performed indirectly, by first rendering the reconstruction and then using a look-up table to determine which visible triangle (and associated vertices) projects onto a query location in the image.

2.2.5 Texture Extraction

The linear combination of facial prototypes is not nearly flexible enough to capture many of the skin's local characteristics like varying pigmentation (moles, freckles), wrinkles or scars. While modeling of such details remains an unsolved problem, it is possible to transfer this information from the image to the 3DMM reconstruction, in a post-processing, for later reproduction. This capability adds considerably to the photo-realistic appearance of re-synthesized faces.

Given a fitting result, the position and visibility (in the image) of each vertex can be computed, as stated above. For visible vertices the underlying color value from $\boldsymbol{I}_{input}$ can be retrieved and mapped into a texture in the reference frame.

In order to render the extracted texture independent of the input's specific pose and illumination, the "true" albedo has to be separated from shading and shadowing effects. Using the estimated parameters from the fitting result, this is accomplished by processing the extracted colors through the inverted equations for lighting (2.22) and color adjustment (2.23). The albedo in hidden parts of the face is filled in from the modeled texture. Note, that the extracted albedo is always a color value, *i.e*. illumination inversion also maps gray scale inputs to the canonical color space of the model's original texture samples. Hence, the extracted textures are colored and usually visualized in the reference frame as shown in Figure 2.4.

Chapter 3

Review on Segmentation Techniques

3.1 Segmentation with Graph Theoretic Methods

In recent years many researchers in the field of image segmentation have focused on methods utilizing various graph-theoretical results that aim at partitioning a graph into disjoint branches which then represent separate regions in the image. While there exists a number of partitioning techniques, they all share the same underlying representation. An image is interpreted as undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The nodes/vertices $\mathcal{V}$ represent the image's basic elements, usually pixels or feature descriptors. Sometimes pixels with similar properties are combined in a pre-processing step to form small coherent patches (known as super-pixels [RM03]). Using these as building blocks of the image, reduces the associated graph complexity. $\mathcal{E}$ is the set of edges connecting the nodes to a graph. Each edge is associated with a weight w_{ij} which encodes the affinity, essentially a notion of similarity, between the linked nodes v_i and v_j. A precise definition of which pixels are connected by edges depends on the particular method. The graph topology given by $\mathcal{E}$ and the link weights can be stored as sparse and symmetric adjacency (affinity) matrix:

$$\boldsymbol{A}_{ij} = \begin{cases} w_{ij} & \text{if } v_i \text{ is linked with } v_j \\ 0 & \text{otherwise} \end{cases} . \tag{3.1}$$

Node affinities can be derived from various visual cues, like intensity, color, texture and so on. Commonly the edge weights also include a distance term to attenuate or annul links between nodes that do not lie close to each other. This ensures sparsity in the adjacency matrix, with considerable impact on the computational costs. Once the image has been translated into a graph the segmentation is performed by grouping the graph nodes according to their affinities. The rationale is that pixels with strong edges are similar and belong to homogeneous image regions while pixels connected via weak links probably originate from structurally different regions.

3.1.1 Unsupervised Clustering Techniques

For many applications segmentation is an intermediary step, serving merely as a method to form meaningful clusters of pixels for further bottom-up analysis. Often no problem specific knowledge is available at this level and the segmentation algorithms are expected to work unsupervised, driven only by generic cues.

With the introduction of the minimum cut criterion Wu and Leahy [WL93] founded a whole class of segmentation methods suitable for this purpose. Their common approach is to minimize the similarity between pixels that are assigned to different regions, based on a global criterion of the according node links. Formally, a graph $\mathcal{G}$ can be partitioned into two disjoint sets by removing the links connecting both parts. The set of capped edges $\mathcal{C} \subset \mathcal{E}$ defines a cut and induces a new graph $\mathcal{G}(\mathcal{C}) = (\mathcal{V}, \mathcal{E} \setminus \mathcal{C})$. Each cut is associated with a cost value, measuring the degree of dissimilarity between the severed sets:

$$cut(A,B) = \sum_{v_i \in A, v_j \in B} w_{ij} = \sum_{e \in C} w_e. \tag{3.2}$$

Wu and Leahy define an optimal bi-partitioning as the one minimizing this cost term. For K-way partitions the same criterion is applied recursively to the previously obtained subgraphs.

In order to find the minimum cut efficiently, they translate the problem into one of computing a network's maximum flow [Die05]. This formulation uses a graph $\mathcal{G}'$ augmented by two special *terminal* nodes, called source S and sink T, which are each connected to one of the original nodes. In such a graph edges between pixels are called n-links (n stands for "neighbor") and edges connecting pixels to terminal nodes are called t-links. The task is then to find the cheapest cut that disconnects source from sink. High weights on the t-links ensure that the cut comprises only edges found in the original graph $\mathcal{G}$. The *Ford-Fulkerson Theorem* [FF62] states that this problem is equivalent to finding the maximum flow from source to sink through $\mathcal{G}'$. A vivid analogy of this process is a network of pipes carrying water. Each pipe has a transport capacity given by the corresponding edge strength. Pumping sufficient quantities of water from S to T will eventually saturate some of the pipes. Once the amount of pumped water cannot be further increased, the flow through the saturated pipes corresponds to the maximum flow from source to sink in this network and the saturated lines compose the minimum cut. The maximum flow can be computed efficiently in low-order polynomial time [FF62], [GT88].

In their work, Wu and Leahy determine the optimal cut by testing the ST-minimum cut for all pixel pairs. Even with an efficient algorithm available also for this task, the number of nodes required in real-world segmentation problems is still to large. Therefore the authors further reduce the graph complexity by condensing branches that are likely not to share any edges with the min-cut.

Despite the innovative methodology in this work –many aspects can still be found in todays state-of-the-art segmentation techniques– there remains one particular problem. As the authors mention themselves, the minimum cut criterion is strongly biased towards cutting small sets of isolated nodes in the graph,

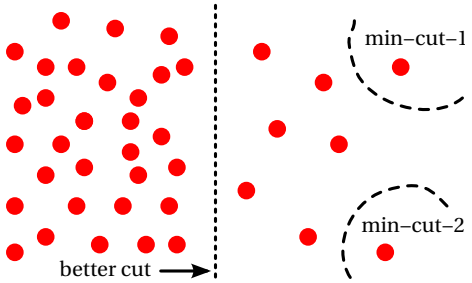


Figure 3.1: Example when the minimum cut criterion creates unwanted partitions. Assuming the edge weights are inversely proportional to the nodes' distances, severing the individual nodes on the right side results in a smaller cost value than the cut that separates the right and left sides.

resulting in either meaningless segments or in a massive over-segmentation. Apparently, when optimizing equation (3.2), it is often cheaper to cut a few strong links than many weak ones. Figure 3.1 illustrates this behaviour.

In [SM00] Shi and Malik address this problem and propose a different cost function. Instead of looking at the value of total edge weight connecting two partitions, their measure computes the cut cost as a fraction of the connectivity wrt. to the whole graph. They call this the *normalized cut*:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}, \tag{3.3}$$

where

$$assoc(A,V) = \sum_{v_p \in A, v_q \in V} w_{pq} \tag{3.4}$$

is the total affinity from nodes in A to all graph nodes V, and $assoc(B,V)$ is similarly defined. With this definition a cut receives small cost if it separates two components that have few edges of low weight between them, and many internal edges of high weight. For the example cuts in Figure 3.1, the *Ncut* value will be high, since the unnormalized cut is 100% of the total connection of the capped nodes. Shi and Malik show that computing an optimal *Ncut* exactly is equivalent to the NP-complete problem:

$$\begin{aligned} &\min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\boldsymbol{D}-\boldsymbol{A})\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y}} \\ &\text{s.t.} \quad y_i \in \{1,-b\}, 0 < b \leq 1, \boldsymbol{y}^T\boldsymbol{D}\boldsymbol{1} = 0 \end{aligned} \tag{3.5}$$

where $\boldsymbol{A}$ is the graph's affinity matrix and the diagonal *degree matrix* $\boldsymbol{D}_{ii} = \sum_j w_{ij}$. Further, if these conditions are relaxed and $\boldsymbol{y}$ is allowed to take on real values, then approximate minimization can be achieved by solving for the second smallest eigenvalue of the generalized eigenvalue problem:

$$(\boldsymbol{D}-\boldsymbol{A})\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y}. \tag{3.6}$$

In order to obtain a partition of the graph the corresponding eigenvector is thresholded, using several test values and the one yielding the minimum *Ncut* value is chosen.

Compared to [WL93] the normalized cut criterion represents a substantial progress in terms of segmentation quality. Yet, the computational complexity for an exact solution is prohibitively high and the errors introduced by the approximate solution are not well understood. In practice even the approximations are costly to compute. The number of non-zero entries in $\boldsymbol{A}$ is equal to the number of pixels times the size of the affinity neighborhood (distance term), which has to be fairly large for this algorithm to work. This limits normalized cuts to relatively small images.

Fowlkes *et al*. [FBCM04] propose an extension that makes the framework applicable to large images. It relies on the *Niström method* to find an approximation to the eigenvalue problem, based on a small number of randomly sampled pixels. These pixels are used to build a reduced, non-square affinity matrix. The leading eigenvectors of this matrix, which can be computed at significantly lower costs, are then linearly combined to derive the approximative solution to a normalized cut. Their experiments show that about one percent of the total image pixels is sufficient to obtain stable segmentations comparable to the ones obtained with the original method.

Another approach to reduce the complexity of normalized cut problem is presented by Cour *et al*. [CBS05]. Based on a statistic analysis of link properties on random images, the authors propose to decompose the graph links into disjoint scales, according to their underlying spatial separation. The result is a graph with multiple layers, similar to an image pyramid, where the links on higher levels represent larger scale connections and the nodes are defined by subsampling of the image pixels at the corresponding distances. Compared to other multi-scale simplification schemes (which handle each scale sequentially), the key idea is here to process the layers in parallel, *i.e*. to seek a consistent segmentation simultaneously across all scales. That is achieved by specifying cross-scale constraints, which enforce the propagation of information through all levels. This approach leads to a more constrained formulation of the optimization problem (3.5) and to a more complex numerical approximation scheme. However, the authors show that it can be solved much faster than the direct method.

Several alternative minimum-cut objectives have been proposed ([CRZ96], [JI99], [WS01], [WS03]) which, in contrast to *Ncut*, are exactly solvable. For example, Wang and Siskind [WS01] consider a minimum mean cut to alleviate the bias of minimum-cut towards small boundaries:

$$\bar{c}(A,B) = \frac{cut(A,B)}{L} \tag{3.7}$$

where L is the length of the boundary dividing A and B. They present a polynomial time algorithm for 2-way cuts and like related methods apply the same procedure recursively to produce finer segmentations. Since their method may result in cuts that do not correspond to any image edges, a region merging step based on equation (3.7) is applied: neighboring regions s_i and s_j with maximum cut $\bar{c}(s_i,s_j)$ are successively joined, until the cut falls below a (manual) threshold.

Other approaches utilize the original minimum-cut framework (including maximum flow solver) and work around the associated shortcomings by different *ST*-graph composition ([IG98], [Vek00]).

Ishikawa and Geiger [IG98] seek a classification of pixels into a small set of gray level labels (posterization). First their algorithm detects image junctions. Then it finds the smallest number of gray level thresholds, such that the junctions in the segmented image are preserved. The authors reason that, due to image noise, simply classifying pixels to the closest gray level would be useless. Instead they re-interpret the problem as one of energy minimization with an assignment error term and a smoothness constraint that encourages nearby pixels to share the same label. This formulation is translated onto a directed graph structure, with a cut representing an assignment function, and using the maximum-flow algorithm to compute the global minimum.

In the algorithm introduced by Veksler [Vek00], the idea is to place the sink node T "outside" the image and link it with appropriately small weights to all image boundary pixels. For every pixel p inside the image a minimum cost contour separating p from the image can be found by means of the minimum-cut framework that disconnects p from T. Veksler argues that for two different pixels p and q the resulting cuts are either nested or disjoint and therefore represent a natural partition of the image. The proposed segmentation algorithm would compute a pT-cut for every pixel p (selecting it as the current source node). However, several optimizations to reduce the number of processed pixels and to reduce the graph size are discussed.

Another strategy for unsupervised segmentation is adopted by agglomerative algorithms ([DHS00]) which start with a trivial partition of n clusters with size one and then subsequently merge pairs of clusters according to some similarity measure.

In Felzenszwalb and Huttenlocher [FH98] the similarity criterion for image regions is based on two measures of image variation. Internal variation of a region is defined as

$$Int(A) = \max_{e \in MST(A,\mathcal{E})} w_e \tag{3.8}$$

with $MST(A,\mathcal{E})$ denoting the minimum spanning tree of A wrt. the edges in E. And external variation is defined as the lowest edge weight connecting two segments:

$$Ext(A,B) = \min_{v_i \in A, v_j \in B} w_{ij}. \tag{3.9}$$

The submitted algorithm works by merging together regions, if the external variation between them is small compared to their respective internal variations, *i.e.* if:

$$Ext(A,B) \leq \min\big(Int(A) + \tau(A), Int(B) + \tau(B)\big) \tag{3.10}$$

with a threshold function $\tau(A) = k/|A|$ that controls to which extent the external variation can actually be larger than the internal ones and still be considered equal. The authors claim that, although this procedure uses a greedy decision criterion, the resulting segmentation satisfies certain global criteria for not being an over- or under-segmentation. Their runtime analysis indicates near linear complexity.

Gdalyahu *et al.* [GWW01] present an interesting randomized agglomerative clustering variant. Based on an algorithm by Karger and Stein [KS96] which approximates the minimum cut in a probabilistic fashion, they generate

slightly different candidate segmentations. The set of regions provides information about how often pairs of pixels share the same cluster. From this they can estimate the probability that a given edge is a bridging link in a "typical" cut. Edges with more than 50% probability of linking regions are finally removed to obtain the segmentation.

3.1.2 Supervised Segmentation Techniques

Often the ambiguities emerging in automatic segmentation techniques can be alleviated or even resolved by a small amount of user input. In recent years the potential of such expertise has received more attention and the focus within the segmentation community has shifted towards semi-automatic methods, not least thanks to new developments on efficient optimization procedures. In particular applications for medical segmentation and photo editing seem to readily accept supervision as a minor drawback, given the remarkable improvements in segmentation speed and quality. Supervised segmentation usually implements one of two paradigms for guidance: **1)** Specification of either boundary elements on the object of interest or a roughly localized boundary template that evolves towards the desired object contours. **2)** Specification of pixels belonging to the desired object and/or pixels that are part of the background.

The *Intelligent Scissors* algorithm by Mortensen and Barrett [MB95] is an example for a boundary-driven interactive image cutout tool. They define the cost between two connected pixels as a weighted sum of three edge sensitive image features, Laplacian zero-crossing, gradient magnitude and gradient direction, such that graph links along an image edge have low weights and links crossing an edge receive high weights. To start a segmentation the user has to select a starting point on an object contour. The system then employs Dijkstra's algorithm ([Dij59], [CLR90]) to compute the shortest (=cheapest) path from every other node to the seed pixel. As soon as the user moves the mouse away, the optimal path from the current position to the starting point is known and instantly displayed. If the computed path deviates from the desired boundary, further seed points can be placed, each time holding the current path fixed and initiating a new search from the last provided seed.

In 2001 Boykov and Jolly [BJ01] (detailed journal version [BFL06]) presented a new segmentation principle that explores the use of minimum cuts in the augmented ST-graph representation for binary image labeling and under the condition that some knowledge about the location and extent of the two image regions is available. Although based on the same techniques that were used in [WL93], their work represented a major breakthrough. They restated the segmentation objective in a more general framework of energy minimization, capable of exploiting a wide range of model-specific boundary- and region-related cues as well as topological constraints. One particular novelty was that their method generalizes easily to n-dimensional application domains (*e.g.* segmenting volume data from computer tomography).

Given a set $\mathcal{P}$ of (n-dimensional) data elements and a neighborhood system $\mathcal{N}$, a segmentation is represented as a binary vector $\boldsymbol{A} = (\boldsymbol{A}_1, \ldots, \boldsymbol{A}_p, \ldots, \boldsymbol{A}_{|P|})$ whose components specify the assignment of each pixel in $\mathcal{P}$ to either one of two

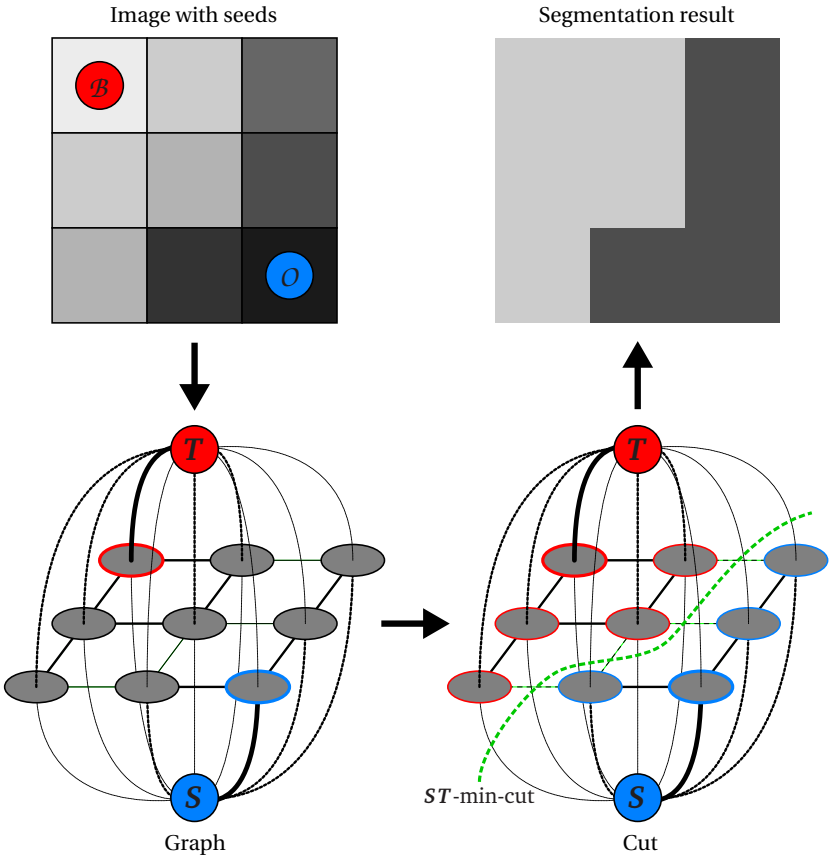


Figure 3.2: Principle of binary segmentation via ST-graph cuts, illustrated for an example 3×3 pixel image. The cost of cutting n-links is defined by the boundary term, encoding the similarity of neighboring pixels, and reflected in the respective edge thickness. The regional term and hard constraints for both labels ([O]bject, [$\mathcal{B}$]ackground) are encoded in the t-links. Inexpensive edges are attractive choices for the minimum cost cut.

labels "object" or "background". The segmentation is driven by soft constraints, which impose certain boundary and regional properties of $\boldsymbol{A}$ and are expressed in a combined cost function:

$$E(\boldsymbol{A}) = \lambda \cdot R(\boldsymbol{A}) + B(\boldsymbol{A}) \tag{3.11}$$

with

$$\text{(regional term)} \qquad R(\boldsymbol{A}) = \sum_{p \in \mathcal{P}} R_p(\boldsymbol{A}_p) \tag{3.12}$$

$$\text{(boundary term)} \qquad B(\boldsymbol{A}) = \sum_{\{p,q\} \in \mathcal{N}} B_{\{p,q\}} \cdot \delta_{\boldsymbol{A}_p \neq \boldsymbol{A}_q} \tag{3.13}$$

and

$$\delta_{\boldsymbol{A}_p \neq \boldsymbol{A}_q} = \begin{cases} 1 & \text{if } \boldsymbol{A}_p \neq \boldsymbol{A}_q \\ 0 & \text{if } \boldsymbol{A}_p = \boldsymbol{A}_q. \end{cases} \tag{3.14}$$

The term $B(\boldsymbol{A})$ influences the boundary attributes of the segmentation. Each $B_{\{p,q\}}$ is non-negative and should be interpreted as a penalty for discontinuity between p and q, *i.e.* $B_{\{p,q\}}$ is large if pixels p and q are similar and close to zero otherwise. If the elements in $\mathcal{P}$ are not spatially evenly distributed, it might be useful to let the boundary term decrease as a function of distance between p and q. A simple and frequently used ad-hoc boundary cost measure is given by:

$$B_{\{p,q\}} \sim \exp\left(-\frac{(\boldsymbol{I}_p - \boldsymbol{I}_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}. \tag{3.15}$$

This expression strongly penalizes unequal labeling between pixels with intensity differences smaller than σ and intuitively models the amount of noise among neighboring pixels, wherein σ can be estimated as camera noise. The regional term $R(\boldsymbol{A})$ introduces individual penalties for the assignment of either label to pixel p. Typically this measure incorporates some knowledge about the element distribution within each region and correspondingly constitutes two parts, $R_p(\text{“obj”})$ and $R_p(\text{“bkg”})$. For example, given two models of the pixel intensities in the object and the background, (3.12) could be defined by:

$$R_p(\text{“obj”}) = -\ln P(\boldsymbol{I}_p|\text{“obj”}) \tag{3.16}$$

$$R_p(\text{“bkg”}) = -\ln P(\boldsymbol{I}_p|\text{“bkg”}) \tag{3.17}$$

The coefficient $\lambda \geq 0$ in (3.11) specifies the relative importance of the regional properties versus the boundary term. In addition to these cost terms which act as segmentation guidelines, the formulation also adopts hard constraints in form of a pre-labeling of pixels. Given two subsets $\mathcal{O} \subset \mathcal{P}$, $\mathcal{B} \subset \mathcal{P}$ with $\mathcal{O} \cap \mathcal{B} = \emptyset$ for which the assignment of labels is known a priori, the goal is then to find the global minimum of (3.11) among all segmentations that satisfy:

$$\begin{aligned} \forall p \in \mathcal{O}: &\quad \boldsymbol{A}_p = \text{“obj”} \\ \forall p \in \mathcal{B}: &\quad \boldsymbol{A}_p = \text{“bkg”}. \end{aligned} \tag{3.18}$$

Boykov and Jolly prove that, by imposing appropriate edge weights, this objective can be projected into an ST-graph, such that a standard minimum-cut/maximum-flow algorithm will efficiently solve (in polynomial time) the global minimization problem with respect to the hard constraints. Figure 3.2 shows the principle of this approach. The authors also suggest that, besides enforcing a fixed labeling on a subset of pixels, the seed regions $\mathcal{O}$ and $\mathcal{B}$ may serve as training samples for models used to derive the regional term (3.12). For example in [BJ01] the respective pixel intensities are used to build histograms for the “object” and the “background” regions.

The powerful concept of object segmentation via binary graph cuts entailed a large number of recent publications building on the outlined principles. We here only mention some of the proposed extensions, which are relevant or related to our work. For a comprehensive overview of related publications, refer to [BFL06]. One particularly intriguing enhancement, called *GrabCut*, aims at including regional cues based on Gaussian Mixture Models and at reducing the placement of seeds to only one label ([RKB04], see also Section 4.3). Another approach incorporates a priori knowledge of the segment shapes to improve

boundary accuracy in gray level images where "object" and "background" regions display similar intensity profiles ([FZ05]). The *LazySnapping* [LSTS04] method employs pre-computed over-segmentation (superpixels based on watersheds) for improved speed and combines object marking and boundary editing in the same user interface. It should be noted that for more than two labels minimization of an energy function of the type in (3.11) is generally NP-hard. However, Boykov *et al*. [BVZ01] present an extension, applicable for multi-way cuts, which efficiently finds a local minimum (and thus only an approximative solution) within a known bound of the global minimum.

Grady and Funka-Lea [GFL04] (journal version [Gra06]) pursuit an interesting premise for a seed-based and multi-label segmentation using random walks. The underlying principle assumes a hypothetical disoriented person starting to walk from a given location and moving randomly, one step at a time and independently of previous decisions, into one of a given number of possible directions. The actual "choice" is governed by probabilities assigned to the possible paths. A typical question that arises from such behaviour is: will the walker ever reach a designated target point ? This concept can be used to describe various physical phenomena, such as Brownian motion. Random walks are applied in [GFL04] to image graphs. Each edge is assigned a weight corresponding to the likelihood of the walker crossing that edge (a value of zero would render the link impassable). Similar to (3.15) the weight is derived from the intensity difference of the adjoining pixels. The question then is: starting a random walk from any of the unlabeled pixels, what is the probability that it first reaches one of the seed pixels ? The authors show that a solution to this problem can be found analytically, without actually having to conduct the simulated walk, by solving a sparse linear equation system for each label. Random walk segmentations have several appealing properties: 1) Each unlabeled pixel is assigned a k-tuple of probabilities that a random walk starting from this pixel first reaches one of the k label seeds. In addition these values are a weighted average of the k-tuples of neighboring pixels. 2) Each segment is guaranteed to be connected to seed points with the same label. 3) Weak boundaries (*e.g*. small gaps) can be found if they are part of a consistent boundary.

3.1.3 Advanced Affinity Measures

The popularity of graph based segmentation methods can be accredited to their high capability of customization. For one, they readily adopt arbitrary topologies which extends graph formulations to many important application domains, *e.g*. space variant imaging, volumetric data and 3D meshes. Secondly, they only rely on the evaluation of an affinity function between each pair of nodes. That means in particular that an embedding of the segmentation cues into a common vector space with "meaningful" distance measure is not required. Despite this flexibility, only a few algorithms use more than one cue in their similarity measure and most applications employ affinity functions of the form:

$$aff_{f,\sigma_f}(\boldsymbol{I},p,q) \sim \exp\left(-\frac{\|f(\boldsymbol{I},p)-f(\boldsymbol{I},q)\|^2}{2\sigma_f^2}\right) \tag{3.19}$$

where f is responsible for extracting features like color or local texture properties. This is in contrast to common agreement that integration of multiple cues and image specific measures increases the robustness of segmentations. In this section we review three advanced techniques, two of which are themselves based on graph algorithms, that demonstrate how the affinity measure can be improved by adapting to global image characteristics or by combining region, contour and texture based cues.

In [GSAW05] Grady *et al*. use a random walks approach to tackle the alpha matting problem for color images, which is closely related to segmentation. Instead of using the Euclidean norm in (3.19), they propose to apply *Locality Preserving Projections* (LPP), developed by He and Niyogi ([HN03]), in order to distinguish object boundaries as good as possible. The goal of LPP is to find a linear projection for dimensionality reduction similar to PCA or *Linear Discriminant Analysis* (LDA). While the two latter aim at maximizing the remaining variance respectively the between-class scatter over intra-class scatter, LPP has the objective to preserve local structure. *I.e*. it tries to keep elements in proximity to each other if they are nearby in the original space. This is achieved by mapping the spatial relationships of the data points (in the input feature space) to an adjacency graph. The projection is found as solution to a generalized eigenvalue problem, based on this graph's Laplacian. With Q denoting the LPP projection and c_i the color at pixel i, Grady *et al*. replace the Euclidean norm inside the affinity function (3.19) by: $(c_i - c_j)^T Q^T Q (c_i - c_j)$.

Omer and Werman [OW06] present an affinity function based on distance and densities in feature space. Their motivation is that two points lying in one dense region are more likely to originate from the same source than two points with the same Euclidean distance, but which are separated by a sparsely populated region. Formally this notion is expressed as a trade-off between finding a geodesic that connects two feature points (by definition it has minimal length) and on the other hand avoids low density regions (bottlenecks). The authors translate this problem to a graph construction by connecting each data point to a fixed number of its nearest neighbors with an edge weight proportional to their respective distance. A local density estimate for each node is obtained by averaging the adjoining edge weights. Then, in order to (locally) couple distance and density information, each edge weight is divided by the minimum density of the two linked nodes. This creates expensive bottlenecks for edges passing sparse regions. Dijkstra's algorithm is utilized to determine a shortest path between any two points and the cost of this path defines their affinity. Judging from a comparison of normalized cut segmentations on color images, this new affinity measure results is significantly better object separation compared to simple Euclidean metric based affinities.

Probably the most established techniques for unsupervised image clustering originate from the contributions on normalized cuts ([SM00]), providing the underlying segmentation framework, and on contour ([LM98]) and texture cue integration [MBSL99],[MBLS01] by Malik *et al*., providing a robust and versatile affinity measure. The combination of these two techniques has become a reference for segmentation evaluation and ranges among the most cited approaches in relevant literature.

Malik *et al*. derive their affinity from a scale and orientation selective analysis of the image. For this purpose it is first convolved with a filter bank composed of elongated even- and odd-symmetric filters, based on Gaussian derivatives, at three different scales and six orientations, as well as center-surround (DoG) filters at four scales. The vector of filter outputs on each pixel is a multi-scale characterization of its local neighborhood and serves as input to the contour and texture analysis.

Contour The even and odd filters $f^e_{\theta,\sigma}$ resp. $f^o_{\theta,\sigma}$ are devised to form quadrature pairs. Their responses are sensitive to edge like intensity profiles at a specific scale and orientation, so that an oriented contour "energy" can be defined as

$$OE_{\theta,\sigma} = \left(\boldsymbol{I} * f^e_{\theta,\sigma}\right)^2 + \left(\boldsymbol{I} * f^o_{\theta,\sigma}\right)^2. \tag{3.20}$$

Comparing this measure over all scales yields the dominant orientation $\theta^* = \arg\max_\theta OE_{\theta,\sigma}$ and a corresponding energy OE^* for every pixel. Potential contours are precisely localized, using non-maximal suppression, and then assigned an ad-hoc probability: $p_{con} = 1 - \exp(-OE^*/\sigma_{IC})$. Based on this definition the authors propose one component of the affinity measure to be:

$$aff_{IC}(i,j) = 1 - \max_{x \in \overline{ij}} p_{con}(x). \tag{3.21}$$

The intuition behind (3.21) is that two pixels should receive a high link weight only if the line connecting them crosses no significant intervening contours.

Texture In terms of intensity variation due to texture, the filter response vectors constitute an overly redundant encoding. Malik *et al*. therefore suggest to extract a representative set of these feature by clustering the responses using k-means. The resulting cluster centers, called *textons*, can be interpreted as prototypes of local textural appearance. After that each pixel is assigned the index (texton channel) of its nearest cluster center. Using this integer-valued image representation, the similarity between two textured regions centered over pixel i and j is compared by means of the χ^2 statistic on the respective texton histograms h_i and h_j. The texture related affinity component becomes:

$$aff_{TX}(i,j) = \exp\left(-\chi^2(h_i,h_j)/\sigma_{TX}\right). \tag{3.22}$$

Combined Cues Both affinity measures are modulated by a weighting component p_{tex} that depends on the image's texturedness. *I.e*. on a measure that indicates whether a pixel is part of one uniformly textured region or if it is located on a boundary between two differently textured regions. It is obtained in two steps. 1) First the local scale is estimated, by measuring the median of distances between a pixel and its Delaunay neighbors that belong to the same texton channel and still lie within a larger scale (relative to image size) radius of this pixel. 2) Then a disk-shaped window is placed over every pixel and partitioned in two halves, such that the border between them is aligned with the contour orientation OE^*. Its size is given by the local scale. From a χ^2 comparison of the texton distribution in both halves the authors define the

probability-like texturedness value p_{tex}. For the final combined affinity the contour cue p_{con} in (3.21) is multiplied by $1 - p_{tex}$ and the texture cue (3.22) is computed on modified histograms which use p_{tex} to suppress pixels near region boundaries. Then aff_{IC} and aff_{TX} are multiplied.

Several related approaches have been proposed. For example Martin *et al.* [MFM04] adopt the methodologies to measure contour energy and texture discontinuities and extend them to color and patch-based features. Their goal is to derive an image P_b that estimates the posterior probability of a pixel belonging to a boundary. Instead of combining the available cues heuristically, they attempt to learn the necessary parameters for an optimal fusion from human hand labeled segmentations.

3.2 Clustering in Feature Space

The methods presented in Section 3.1.1 use a representation of pairwise relationships between pixels, *i.e.* affinities, for grouping. A second class of unsupervised segmentation algorithms is derived from well established general purpose clustering techniques and operates directly on vectorial representations of the image features (color, filter responses, spatial location, *etc.*). These methods expect that points/pixels being similar with regard to the image cues will lie close to each other in feature space. Segmentation then is equivalent with identifying clusters of feature vectors.

The image retrieval system, presented by Carson *et al.* in [CBGM02], is based on grouping pixels in an 8-dimensional feature space. It is composed of three color coordinates (in $L^*a^*b^*$ space), three texture descriptors (contrast, anisotropy, polarity) and the actual (x, y) pixel coordinates as positional features. The distribution of pixels in this space is modelled as a mixture of Gaussians, using the EM-algorithm ([DLR77], [Bil97]) to estimate their parameters. The final segmentation is obtained by assigning to each pixel the label corresponding to the Gaussian mixture component responsible for the measured vector.

In [CM97] and [CM99] Comaniciu and Meer propose the application of the mean shift algorithm to detect clusters a of 5-dimensional joint spatial-range domain (three colors and two normalized pixel coordinates). The mean shift procedure ([FH75], [Che95]) is a non-parametric technique designed to locate the modes, *i.e.* the regions of highest density in feature space, of an unknown multivariate distribution, without having to estimate the density itself. Its formulation is based on estimating density gradients. Given n sample points $\{\boldsymbol{x}_i\}_{i=1\ldots n}, \boldsymbol{x}_i \in \mathbb{R}^d$, a kernel density estimate with kernel $K(\cdot)$ and window radius h, is expressed as:

$$\hat{f}(\boldsymbol{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right) \tag{3.23}$$

Comaniciu and Meer show that, when using the *Epanechnikov* kernel, the den-

sity gradient can be estimated as:

$$\hat{\nabla} f(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) \frac{d+2}{h^2} \underbrace{\left(\frac{1}{n_{\boldsymbol{x}}} \sum_{\boldsymbol{x}_i \in S_h(\boldsymbol{x})} \boldsymbol{x}_i - \boldsymbol{x} \right)}_{M_h(\boldsymbol{x})}. \tag{3.24}$$

Here $S_h(\boldsymbol{x})$ is a hypersphere of radius h centered on $\boldsymbol{x}$ and $n_{\boldsymbol{x}}$ is the number of feature vectors within the sphere. Equation (3.24) shows that the *mean shift vector* $M_h(\boldsymbol{x})$ has the same direction as the estimated gradient density, thus it points towards the direction of maximum increase of the density. This property leads to an iterative search scheme. Starting from an initial guess of a cluster center at position $\boldsymbol{x}$, successively:

1. compute the mean shift vector $M_h(\boldsymbol{x})$
2. translate the window $S_h(\boldsymbol{x})$ by this vector, *i.e.* $\boldsymbol{x} \mapsto \boldsymbol{x} + M_h(\boldsymbol{x})$

The authors prove that the density along this path increases monotonically and that the procedure converges. For segmentation purposes the mean shift procedure is applied to each pixel, resulting in a set of convergence points. If these cluster center candidates lie in close proximity (in feature space), they are fused. The pixels are relabeled according to which cluster the iterative procedure converged to. Compared to clustering techniques like k-means or Gaussian mixture models, this approach has the advantage that the number of clusters is obtained automatically. On the downside, a good choice of the only free parameter, the bandwidth h, is not trivial.

The algorithm presented by Vezhnevets and Konouchine in [VV05] also aims at a labeling of d-dimensional image data. It belongs to the group of semi-automatic cutout methods, since it is driven by user-provided seeds and allows (but does not require) interaction. The authors employ a cellular automaton ([Neu66]) to solve the labeling task, which is unique compared to previously published methods. The automaton represents each pixel p as a cell with an associated state tuple $(l_p, \theta_p, \boldsymbol{c}_p)$, holding the cell's label l_p, its current "strength" θ_p and the pixel's feature vector $\boldsymbol{c}_p$ (here simply the RGB values). Initially the cell labels are distributed corresponding to the given seed labels. Unseeded pixels receive a "void" label and zero strength. The strength of seeded pixels can be set (user-defined) to values in the range $(0, 1]$, where a value of one reflects a hard constraint, *i.e.* unchangeable labeling, and values smaller than one reflect soft constraints. The segmentation is obtained iteratively by letting each cell p evolve, influenced by its direct neighbors $N(p)$, according to the rule:

$$\begin{aligned} \forall q \in N(p) : \text{if } & g(\|\boldsymbol{c}_p - \boldsymbol{c}_q\|_2) \cdot \theta_q^t > \theta_p^t \text{ then} \\ & l_p^{t+1} = l_q^t, \quad \theta_p^{t+1} = g(\|\boldsymbol{c}_p - \boldsymbol{c}_q\|_2) \cdot \theta_q^t \\ & \text{else} \\ & l_p^{t+1} = l_p^t, \quad \theta_p^{t+1} = \theta_p^t \end{aligned} \tag{3.25}$$

with a monotonous decreasing function $g(x)$, bounded by $[0, 1]$. An intuitive interpretation of this labeling process is the struggle for domination of different bacteria. At each time step a cell "attacks" its neighbors with a force given

by the cell's strength and the attack distance between the offender's and the defender's feature vectors. If the attacking force is greater than the defender's strength the cell is invaded and its label and strength are changed. The procedure continues until a stable condition is reached, which is guaranteed due to the monotonically increasing and bounded strength values. Remarkable properties of this approach are its simplicity (straightforward implementation) as well as independence of computation time from the number of processed labels.

3.3 Deformable Contours

The last group of segmentation algorithms considered in this chapter falls under the category of deformable contour models (also called active contours). In essence one can distinguish two main approaches based on their mathematical construction: *snakes*, using explicit and *level set* methods using implicit boundary representations. We here only outline the fundamentals of both concepts. For a broad overview and related extensions, further references can be found in [BFL06], [NA02] and [OP03].

The explicit form of active contours was originally introduced by Kass *et al*. [KWT88]. A *snake* is a closed parametric curve $\boldsymbol{v}(s) = \{(x(s), y(s)) \mid 0 \leq s \leq L\}$ that changes its shape and location, driven by internal and external forces, in order to reach a minimum-energy state. The energy of the curve incorporates two components:

$$E_{snake}(\boldsymbol{v}) = \int_0^L E_{int}(\boldsymbol{v}(s)) + E_{image}(\boldsymbol{v}(s)) + E_{con}(\boldsymbol{v}(s))\, ds. \tag{3.26}$$

The term E_{int} is responsible for the contour's intrinsic properties, such as elasticity, smoothness and curvature. A common choice is given by:

$$E_{int}(\boldsymbol{v}) = \alpha|\boldsymbol{v}'(s)|^2 + \beta|\boldsymbol{v}''(s)|^2 \tag{3.27}$$

where α controls the curve's tension and β its rigidity. The force that actually lets the contour evolve, by attracting it to certain image features, is expressed through the external term E_{image}. For example, an energy changing inversely with respect to the image's gradient magnitude will "pull" the contour towards edges:

$$E_{image}(x, y) = -|\nabla I(x, y)|^2. \tag{3.28}$$

E_{con} denotes a constraint energy, used to incorporate higher level information to control the snake. The curve energy is minimized using an iterative procedure based on gradient descent. Since the functional (3.26) typically exhibits many local minima, this method depends on a reasonable initialization of the snake's position in order to converge to a desired object boundary. On the other hand, once a snake has adapted to a boundary of interest, if this boundary moves slightly, the energy minimization procedure will draw the snake towards the new boundary location and thereby offers a way to track moving objects. Another interesting feature of this technique is the ability to reconstruct subjective contours, *i.e*. edges that are not actually present in an image, but are perceived

nevertheless (*e.g.* the well known *Kanizsa triangle*). Besides the sensitivity to proper initialization, an important shortcoming of the classic snakes formulation is the inability to change topology (split or merge) during evolution.

Active contours based on level set formulations ([OS88],[MSV95]) avoid this problem. The idea is to represent the curve implicitly through a higher dimensional function $\phi : \mathbb{R}^2 \mapsto \mathbb{R}$ such that a particular level (usually the zero level) defines the contour: $\boldsymbol{v} \equiv \{\boldsymbol{x} \in \mathbb{R}^2 | \phi(\boldsymbol{x}) = 0\}$. A frequently used choice for such a function is the signed distance to the initial curve:

$$\phi(\boldsymbol{x}) = \begin{cases} \text{dist}(\boldsymbol{x}, \boldsymbol{v}), & \text{if } \boldsymbol{x} \text{ is inside } \boldsymbol{v} \\ 0, & \text{if } \boldsymbol{x} \text{ is on } \boldsymbol{v} \\ -\text{dist}(\boldsymbol{x}, \boldsymbol{v}), & \text{if } \boldsymbol{x} \text{ is outisde } \boldsymbol{v}. \end{cases} \tag{3.29}$$

As in the explicit case, contour evolution is caused by certain forces, denoted by a speed function F that specifies how each point on the curve moves along its normal direction. The movement of the level set function ϕ that matches the evolving contour is then described by a partial differential equation:

$$\frac{\partial \phi}{\partial t} + F \cdot |\nabla \phi| = 0. \tag{3.30}$$

Within the speed function many criteria like physically motivated or image based influences on the curve's behaviour can be implemented. For example in [MSV95] the authors consider the function $F = g(\boldsymbol{I})(c + \kappa)$ for boundary detection. This term comprises three contributions: **1)**The constant velocity c is similar to a balloon force, pushing the curve inwards or outwards. **2)**The term $\kappa = \text{div}\left(\frac{\nabla \phi}{\|\nabla \phi\|}\right)$ introduces a regularizing component by generating a flow that decreases the total curvature and at the same time shortens and smoothes the contour. Together $(c+\kappa)$ act as an intrinsic force comparable to E_{int} in the energy based snake model. **3)**The stopping function $g(\boldsymbol{I}) = \left(1 + |\nabla G_\sigma * \boldsymbol{I}(x, y)|\right)^{-1}$ (G_σ is a Gaussian filter) represents an external image dependent force. Its goal is to stop the evolving curve when it reaches prominent object boundaries.

An outstanding characteristic of the level set approach is that any topological configurations are handled naturally. That means merging, splitting, initial placement and detection of any number of contours is possible without taking extra care, since actually all points are always connected (by the same topology) through the level set function ϕ. Also, in contrast to snakes, this method can be generalized for hyper-surfaces (*e.g.* in 3D). This flexibility of course comes at the price of extra complexity, introduced by the PDE formulation in a higher dimensional space. Early level set implementations suffered from low performance, as they required computations on the whole image plane (ϕ updated everywhere). With the invention of more efficient narrow band techniques (computing only within a confined thin region around the evolving contour) and fast marching methods (applicable if the contour is guaranteed to move only in one direction) this is no longer a handicap.

Chapter 4

Skin Segmentation for Faces

In this chapter we develop a novel framework for automatic binary segmentation of facial images into skin and other facial components. The resulting maps serve as indicators for outliers respectively occlusions and can be used to perform further in-depth analysis of meaningful regions like facial organs or hair. Although our methods are partly based on established segmentation algorithms, we enforce two design goals that make this work non-trivial.

- The framework should be applicable to gray scale images. This stands in contrast to many existing techniques dealing *e.g.* with skin detection or segmentation (not necessarily faces), which usually only classify individual pixels based on their color, *e.g.* [JR02, VSA03]. The requirement for gray scale skin segmentation originated from the demand to work with a certain subset of the FERET face database on which the Morphable Model had already been extensively tested. However, it also represents an idealistic point of view. For human observers the luminance channel contains sufficient features to deliver a detailed labeling of all components in a face image. It is therefore desirable to develop segmentation procedures which attempt to make best use of the same information before depending on additional color input. Such algorithms then have a wider field of application and hopefully can perform even better if color is available.

- A strong emphasis is put on the ability to obtain a decomposition of a face automatically, *i.e.* without the need for human guidance. Prior to this work, several face manipulation tasks based on the Morphable Model required tedious manual image masking in order to give satisfactory results. This meant a strong limitation. For one, because manual input is always subjective and not exactly reproducible, and more important, because of the substantial amount of work involved in large scale experiments. By employing automated masking procedures these manipulations suddenly become interesting for laymen users and for off-line applications like face recognition.

Our segmentation results have several potential fields of application. Yet, to motivate and develop the necessary processing steps in detail, we concentrate

in the following on the two exemplary challenges that were mentioned in the introduction:

- Face recognition from mole-like irregularities in the skin.
- Face exchange for automated high-level photo manipulation.

The Morphable Model plays a key role throughout this work since each of the proposed applications accesses at least one of its special capabilities (face description, rendering, dense mapping). It therefore suggests itself to also employ the model for segmentation. This, however, is not straight-forward. In the next section we study some of the weaknesses of the Morphable Model with respect to our objectives and demonstrate why a fitting result obtained by this technique alone is not sufficient to directly derive a reliable segmentation of a face image. The main contribution of our work, consisting of the three subtopics texture features, illumination compensation and segmentation, is presented in Section 4.2.

4.1 3D Morphable Model Deficiencies

One of the applications discussed before, depends on a binary segmentation of a face into skin and non-skin components. The non-skin region can be seen as composition of two contributions. Part one comprises the characteristic facial organs which do not appear as "normal" skin: eyebrows, eyes, nostrils and mouth. Part two comprises outliers. By that term we refer to all kind of unexpected objects in the sense, that they do not appear in every face. This definition includes beard, hairstyle, glasses, *etc*.

Ideally, we should be able to derive the first contribution from a Morphable Model reconstruction. It would allow us to mark the corresponding vertices in the reference frame and then via the estimated shape and pose to project this selection to the original image domain. Unfortunately, as we show in section 4.1.1, this approach is often not perfectly reliable.

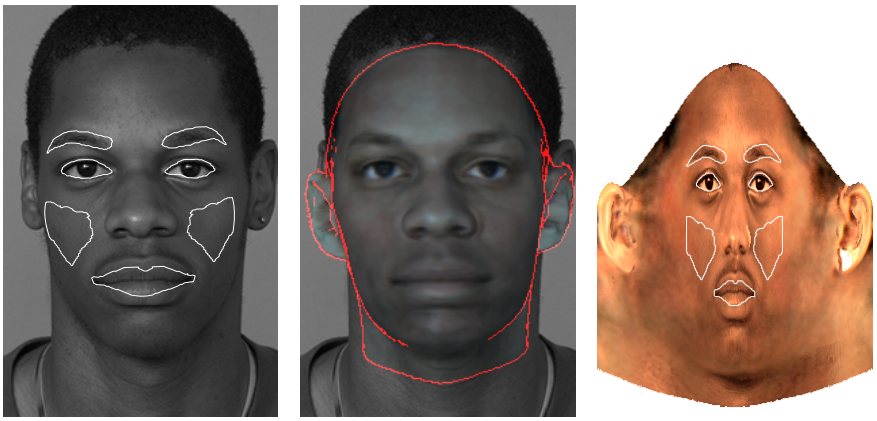
The second non-skin region is even more difficult to handle, since the Morphable Model offers no clear strategy how to deal with outliers. On one hand for example beard and the hairline can be reproduced in the texture since they are part of several training samples. Therefore they usually do not provoke high matching errors. Yet, if such a feature is encoded in the model parameters, but it is not consistently outlined in the reference frame, then the segmentation problem for this feature is simply deferred to the reference domain and we gain no advantage. On the other hand hairstyle or appearance changes due to facial expressions are not represented by the model. Even worse, if larger areas of the face contain such outliers, they can seriously perturb the model parameters and corrupt a reconstruction in several ways.

4.1.1 Causes of Bad Reconstructions

Usually a bad fitting occurs when the matching algorithm encounters unknown factors, *i.e*. when it attempts to fit the parameters to structures or conditions

(a) Input & proj. features **(b)** Reconst. & contours **(c)** Extracted texture

Figure 4.1: Example of a bad fitting result due to facial expression. The left image shows the input marked with the estimated outlines of eyebrows, eyes, lips and cheeks, given by the reconstruction. In the right image the same regions are marked in the reference frame on top of the extracted texture. The center image shows the actual rendered model with its contours highlighted.

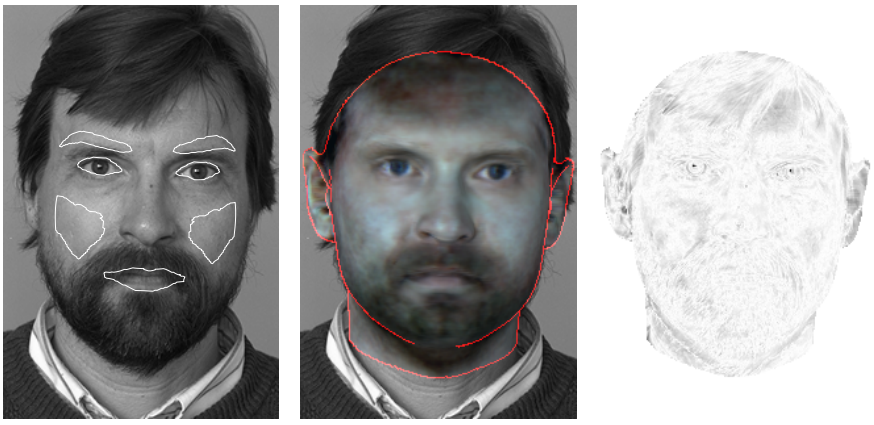


(a) Input & proj. features **(b)** Reconst. & contours **(c)** Extracted texture

Figure 4.2: Example for a facial feature, namely thick lips, that is not part of the training set and therefore cannot be represented adequately by a Morphable Model fitting.

that were not anticipated in its design and setup.

The Morphable Model was trained entirely from scans of Caucasian faces with neutral expression and of middle range age. Consequently it is not suited to represent deformations that appear due to aging, under varying expressions (*e.g.* open mouth, closed eyes, lifting eyebrows) or features which are specific for other races (*e.g.* thick lips). Figures 4.1 and 4.2 display the extent of correspondence errors we have to deal with in such situations. For the portrait



(a) Input & proj. features **(b)** Reconst. & contours **(c)** Matching error

Figure 4.3: A fitting result impaired by the outlier hair patch on the forehead, noticeable from the blotchy appearance of the modeled skin. The locations of higher (darker) matching errors do not correspond to the real locations of the outliers. The well reconstructed beard in this example shows how differently the Morphable Model matching handles facial hair from hairstyle.

in 4.1(a) it is obvious that a segmentation of skin and facial organs cannot be derived from the locations predicted by this model. The fitting result in Figure 4.2(b) appears to match the original face (apart from the ears and neck). However, the visualized outline of the mouth, taken only from the estimated shape, here also reveals that the lips are still misaligned.

Another form of disturbance are occlusions from hairstyle and glasses. Such outliers can cover significant areas of the face and thereby cause the following problems: **1)** Due to the holistic representation, adapting the modeled texture to outliers comes at the cost of higher reconstruction errors in other regions. As result differences between the real and the rendered image "even out". We observe this phenomenon in Figure 4.3. Note, that the reconstruction error on the forehead is not consistently (*e.g.* high for outliers, low for normal facial area) distributed. **2)** Considerably lighter or darker areas might be mis-interpreted as illumination effect instead of an unexpected change of the face's albedo. Thus the estimated light parameters are diverted, most likely resulting in a less realistic approximation, and with the risk of introducing wrong cast shadows into the synthesized image. **3)** The reconstructed shape deteriorates and leads to bad correspondence and thereby misaligned features. We emphasize again, that this perception of deviations from the model is not equivalent to our notion of outliers in the search for skin segments, because the hairline and beard are part of the texture model, see Figure 4.3.

Besides these foreseeable cases, the matching algorithm sometimes produces misaligned results in particular at the eyebrows, even in the absence of such deviations. This behaviour is illustrated by two examples in Figure 4.4. Direct comparison of the input image and the rendered model shows no discrepancies. The eyebrows' appearance seems to be adequately modeled. Yet, a closer look

Figure 4.4: The displaced eyebrows derived from visually correct reconstructions indicate a problem with the correspondence of the Morphable Model and necessitate alternative methods to robustly segment this region.

on their back-projected shape reveals that they are displaced. This indicates a problem in the registration of the 3D scans prior to the model building phase. Apparently for some of the face scans the correspondence is not correct. The problem may be fortified by the fact, that the fitting procedure optimizes texture and shape parameters independently and also separately for the four model segments.

4.1.2 Segmentation Hints

Our conclusion from the demonstrated shortcomings is that we cannot trust on the Morphable Model approach to deliver estimates for the locations of all the prominent facial features with the required accuracy. We can also not rely on the matching error per pixel to indicate the existence or location of outliers. In fact there exists not even a viable criterion to determine if a fitting result is actually good or bad. Furthermore the Morphable Model has by design a limited domain which does not extend beyond ears and forehead. Consequently it is not suited to directly draw any conclusions on such areas like the top or the back of the head which generally contain the hairstyle.

The situation is still not entirely hopeless. By experience we know that the estimated pose (position, scale and orientation) and parts of the shape are very reliable contributions. Figures 4.1 to 4.4 show that in particular the cheek area can be robustly localized, even under adverse conditions. That is not surprising, since this region exhibits no distinctive features which could disclose a displacement between model and image (the dense correspondence is here somewhat arbitrary). While the same is valid for the forehead, the key advantage of the cheeks is that we can expect them to be “unharmed” by the typical outliers like hairstyle, beard and so on.

To summarize: from a 3DMM fitting we cannot infer the location of all non-skin pixels but we can robustly and automatically estimate the location of two regions which very likely belong to the skin segment. These are now usable as image specific hints for skin appearance and as so-called *seeds* to initialize a general purpose supervised segmentation algorithm.

4.2 Skin Features for Gray Level Images

In gray scale, skin and non-skin are only distinguishable by luminosity and/or texture. Skin typically appears as smooth area with more or less pronounced intensity gradients. Its albedo can be assumed constant which means the varying pixel intensities can be accredited to shading of the curved facial surface. In this simplified view, details like folds and wrinkles are treated as deviations in the skin's texture and not as attributes of the face's 3D shape. The most frequent non-skin segments feature a multitude of visual patterns:

- **Nostrils**, most predictable non-skin component with simple blob-like appearance.
- **Lips**, albedo varies according to skin type (*i.e*. lightness due to amount of contained melanin). Lips appear usually darker than "normal" skin, but depending on lighting conditions they can have a significantly brighter glossy reflection.
- **Eyes**, complex appearance due to composition of eyelids, eyelashes and eyeball with pupil and iris. In simple terms it is made of small patches with nearly constant brightness separated by high intensity edges and intermittent sharp specular highlights (often the brightest and the darkest pixels in a face are located on the eyes).
- **Hair (eyebrows, beard, hairstyle)**, for most people "facial hair" is darker than surrounding skin while the color (and geometry) of hairstyle is not directly correlated with any facial feature. The appearance depends on hair length and density. Due to the complex interaction of several lighting phenomena (*e.g*. diffuse/specular reflection, translucency and self-shadowing on multiple layers) it ranges from patches of constant brightness over structured texture with curvilinear aligned wisps to stochastic texture in areas with stubble. Because of the small scale details of hair, compared to the image resolution, matting plays an important role. This effect manifests itself among other things as a blending between the actual hair and skin regions involving pixels with intermediary gray levels.

Other non-skin contributions (glasses, teeth, *etc*.) have been omitted from this list, as they share similar issues and occur only infrequently.

While the 3DMM provides a model based prediction for the facial organs (including eyebrows and beard), subject to the aforementioned restrictions, there exists no equivalent technique to deal with the complex appearance of hairstyle. Considering the above listing we conclude that the best alternative approach for gray scale skin segmentation is a rather simple strategy. Skin regions should be identified by asserting a certain relatively narrow brightness range and a minimum smoothness respectively typical skin texture. Areas which do not meet either of these conditions should be rejected as the "anti-case" to skin, without actually knowing which element caused the deviation. The parameters of this process should be obtained from image samples.

Due to shading and blending effects on the two segments their respective histograms overlap. That means, they cannot be expected to be separable on

basis of per-pixel decisions in the input image domain. Instead, the measurement we use to characterize skin, both relate to texture properties and therefore require consideration of the pixels' local neighborhoods. In the next section we propose a simple procedure to find skin regions by example using the cheeks as texture template. Then, in Section 4.2.2, we introduce a novel technique for illumination compensation to level out slowly changing intensity gradients and thereby render the texture comparison results more robust against lighting effects.

4.2.1 Distinguishing Texture by Analogy

In Section 4.1.2 we argued that the 3DMM fitting of a face can be used to reliably indicate where in the image the cheeks' skin patches are located. This knowledge can now be used to evaluate the remainder of the face in terms of texture similarity. The simple approach we follow here was inspired by a technique for texture synthesis, developed by Efros and Leung [EL99]. In the particular setting they address, the problem is to generalize from a relatively small sample of a texture a larger image while avoiding visible seams and blunt patch copies which easily lead to a noticeable tiling effect. Efros and Leung synthesize a texture one pixel at a time by repeatedly matching the neighborhoods around unprocessed pixels in the synthesis image against all possible source patches extracted from a sample texture. The center pixels of the minimum error patches then build up the synthesized texture.

With modifications this idea can be used as analysis tool to compute a feature of texture similarity for an image (*target*) with respect to a given sample of the texture (*source*). Let I_{tgt} be the target image for which the similarity should be computed. Further we denote with I_{src} a source image and with Ω_{seed} an associated binary mask, both defining a texture sample region. The similarity is then computed for each pixel $p \in I_{tgt}$ independently by taking its local neighborhood N^p_{tgt} and searching within the seed region of I_{src} for the best matching patch N^q_{src}. We use the sum of squared distances (SSD) as perceptual distance measure between two patches, unlike [EL99], without imposing different weights on the neighborhood's pixels. The texture similarity error per target pixel p is:

$$E_{ts}(p) := \min_{q | N^q_{src} \subset (I_{src} \cap \Omega_{seed})} d_{SSD}\left(N^p_{tgt}, N^q_{src}\right). \tag{4.1}$$

This measurement does not yet take the statistics of the sample texture into account. In order to determine how likely a target pixel may originate from this texture we actually compute the k-nearest-neighbors to N^p_{tgt}. The error E^k_{ts} is then defined, analogous to equation (4.1), as the average of the corresponding closest-patch distances:

$$E^k_{ts}(p) := \frac{1}{k} \sum_{j=1}^{k} \left(\min_{j \atop q | N^q_{src} \subset (I_{src} \cap \Omega_{seed})} d_{SSD}\left(N^p_{tgt}, N^q_{src}\right) \right). \tag{4.2}$$

Figure 4.5 depicts a schematic view of the process, described by equation (4.2). It is controlled by two parameters: the number k of closest matches to

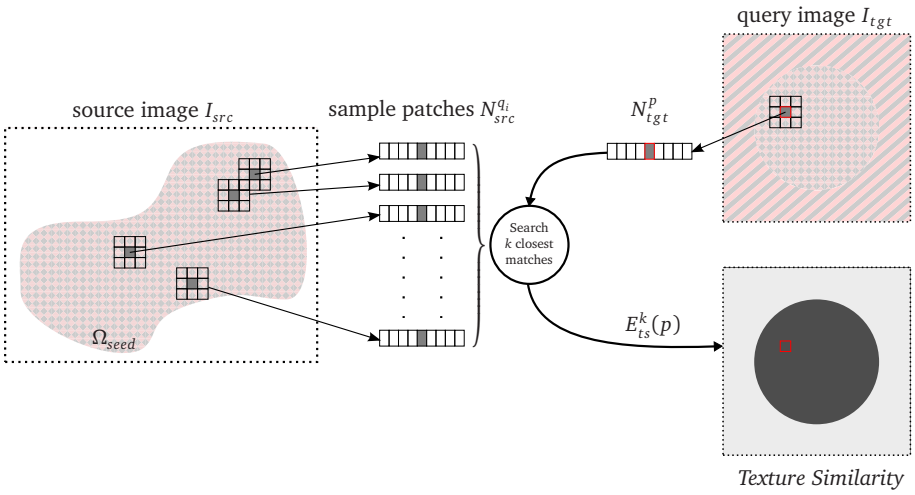
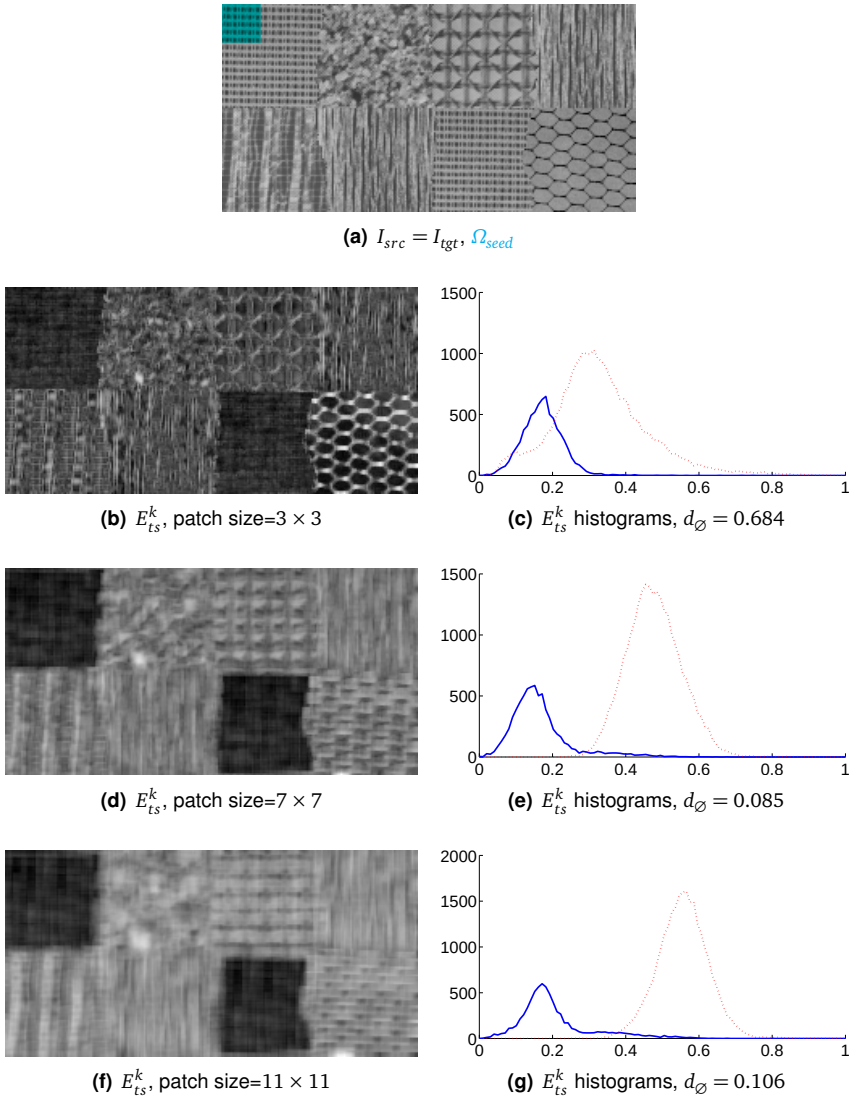


Figure 4.5: Illustration of the procedure that is used to compute a measure of texture similarity, as defined in equation (4.2), using a dictionary of small sample image patches.

consider during averaging and the size and shape of the local neighborhoods. In practice we implement the latter as square image patches. Their size should be chosen such that the patches capture the building blocks of the sample texture. The influence of the size parameter on E^k_{ts} is demonstrated on a toy example in Figure 4.6. It reproduces matching errors obtained by using neighborhoods of varying size on a multi-textured target image and given texture sample. The associated histograms on the right side display the distribution of E^k_{ts} separated in two segments, the one actually containing the sample texture (blue) and the one containing all other textures (red), based on a ground-truth segmentation. The amount of overlap between the red and blue histograms is a negative indicator for the discriminative power of the respective texture feature. It can be computed by means of the *histogram intersection* [SB91] similarity measure for two histograms g and h:

$$d_{\varnothing}(g,h) := \frac{\sum_i \min\left(h[i], g[i]\right)}{\min\left(|h|, |g|\right)}, \tag{4.3}$$

where $|h|$ and $|g|$ denote the magnitudes of each histogram, *i.e.* the total number of binned samples. For this particular case the numerous low values in the texture similarity image 4.6(b) and the large histogram overlap 4.6(c) clearly point out that 3×3 pixel patches do not yet adequately represent the unique structure of the sample texture. This is especially noticeable on the hexagonal pattern in the lower right section. By increasing the neighborhood size in 4.6(d) to 4.6(g) we achieve a much better distinction between regions containing the seeded texture and those which are dissimilar. This is expressed in the larger contrast between the respective segments in the E^k_{ts} images as well as in the better separated histograms. However, using too large neighborhoods is also not advisable. Apart from the quadratically growing computational effort, this comes at the cost of smoothing out the desired texture segment boundaries (also indicated by the larger histogram intersection of $d_{\varnothing} = 0.106$ using a 11^2 sized



(a) $I_{src} = I_{tgt}$, Ω_{seed}

(b) E_{ts}^k, patch size=3×3

(c) E_{ts}^k histograms, $d_{\varnothing} = 0.684$

(d) E_{ts}^k, patch size=7×7

(e) E_{ts}^k histograms, $d_{\varnothing} = 0.085$

(f) E_{ts}^k, patch size=11×11

(g) E_{ts}^k histograms, $d_{\varnothing} = 0.106$

Figure 4.6: Texture similarity procedure (with $k = 5$) applied to multi-textured image of 256×128 pixels (a), which serves simultaneously as target and source. The seed region Ω_{seed} (24×24 pixels) is highlighted in color. Images (b), (d) and (f) show the resulting matching error E_{ts}^k (darker pixels correspond to smaller error, *i.e.* higher similarity) obtained by using square neighborhood patches of varying sizes. The plots on the right show the respective distribution of E_{ts}^k separated according to the ground-truth segmentation. In (b) and (c) the distribution of errors shows that patches of 3×3 pixels are insufficient to capture the intrinsic structure of this particular sample texture. For larger neighborhoods the measure distinguishes much better between the sample and other textures, but increasing the patch size comes at the cost of smoothing out the boundaries between different textures.

patches versus $d_{\varnothing} = 0.085$ for 7^2 patches).

As for the second parameter k, the impact on the results is less obvious. Let us assume the case $k = 1$ (respectively no averaging is performed) and $I_{src} = I_{tgt}$. That means the sample texture holds exact patch copies of parts of the target image. In such a situation the corresponding similarity error will equal to zero for all seed pixels, while for natural (not completely regular) textures perfectly matching patches are very unlikely. If the seed area is moved to another location belonging to the same texture, then suddenly the previously seeded pixels receive higher E^k_{ts} values and the new seed pixels drop to zero. This behaviour is not desirable because the feature in the seed regions does not anymore reflect the stochastic properties of the texture. Usually small values of $k > 2$ (we use $k = 5$ in all experiments) are already sufficient to ensure that E^k_{ts} becomes robust against this effect. Setting k to much higher values (*e.g.* ~ 20) does not yield any significant improvement but renders the procedure computationally a lot more expensive.

4.2.1.1 Application to Faces

The texture similarity method is easily adopted to our binary skin segmentation problem. As explained earlier, the cheeks constitute a facial area which is unlikely to contain outliers and which holds samples of typical skin texture. We are able to robustly determine the corresponding region in a novel face from its 3DMM reconstruction which provides the skin seed mask Ω_{seed}. Recall that the exemplary application requiring a hard skin segmentation also depends on the dense mapping of the 3DMM. Therefore, at this stage, we do not yet care for results outside the estimated facial area which is defined by the support of the Morphable Model fitted to the given input image. Let Ω_{supp} denote this domain. The algorithm is applied to the face image (again $I_{src} = I_{tgt}$), but constrained to target pixels within the model's support.

Under the assumption that the selected seeds contain only skin, the output E^k_{ts} inside these areas defines the range of matching errors one can expect for similarly textured regions. A basic segmentation can then be obtained by using the maximum of this range as threshold to the whole E^k_{ts} image:

$$I_{skin}(p) := \begin{cases} 1 & \text{if } E^k_{ts}(p) \leq \max\limits_{q \in \Omega_{seed}} E^k_{ts}(q) \ \wedge \ p \in \Omega_{supp} \\ 0 & \text{otherwise} \end{cases} . \tag{4.4}$$

It should be noted that without averaging over the k-nearest-neighbors in (4.2) all errors inside Ω_{seed} would be zero. This in turn means we could not derive a suitable threshold for the segmentation approach of (4.4).

Figures 4.7 and 4.8 display segmentation results obtained by this technique. The images in 4.7 demonstrate that under ideal conditions the texture similarity feature is indeed capable of separating major non-skin components from the rest of the face. By using a patch size of only 3×3 pixels, thus minimizing the associated smoothing effect, we further attain a fairly precise masking of several small scale outliers, *e.g.* the hair strand (top row) and certain moles. On the

other hand the skin segment gaps near each face's right side and chin as well as on the necks suggest that the method can be negatively affected by shading. That is to be expected because, strictly speaking, the E_{ts}^k feature does not purely measure texture resemblance but also the overall gray level difference between the compared patches. Consequently, if the intensity of shaded respectively illuminated skin areas in the face differs to much from the "learned" range within the seed region, E_{ts}^k will be high, despite them actually exhibiting the same texture. This behaviour is confirmed by the negative examples in Figure 4.8. In the next section we propose a complementary method to circumvent this problem.

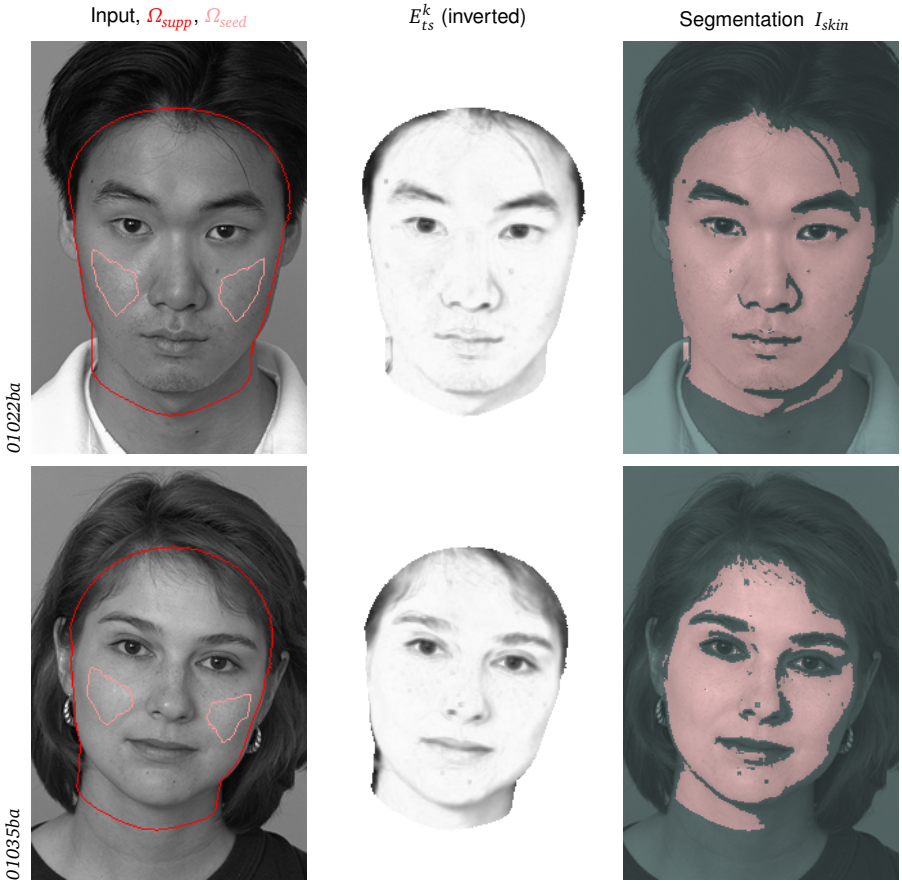


Figure 4.7: Examples of successful binary segmentation, obtained by thresholding the output of our texture similarity algorithm which has been applied directly on the unprocessed face images. Despite the simplicity of this approach, prominent non-skin components as well as small scale details like a hair strand, several moles and a specular highlight could be excluded from the skin.

4.2.2 Illumination Compensation

In the previous section we pointed out that significant changes in the skin's luminosity have a negative impact on the performance of the texture similarity

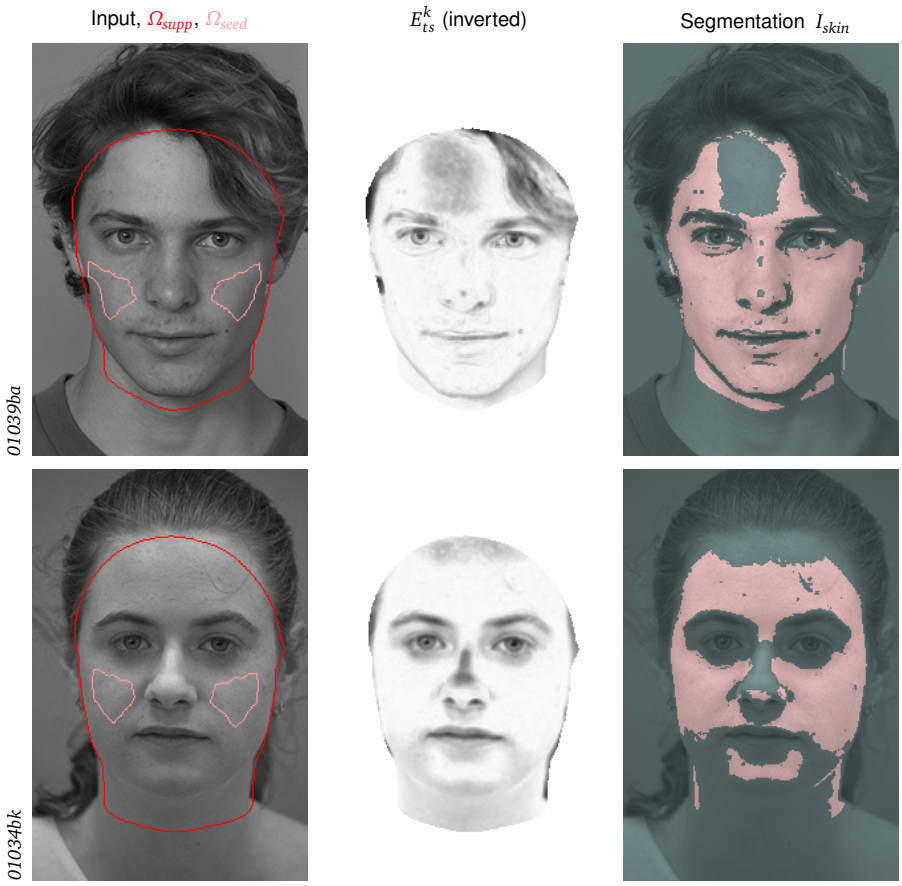


Figure 4.8: Examples for the negative impact of illumination effects on the performance of our segmentation approach (4.4). The results were obtained analogous to Figure 4.7. In both cases the overall brightness in Ω_{seed} does not reflect the gray level range in skin areas that are affected by illumination / shading. This leads to higher matching errors in the respective regions and to an underestimation of the threshold value.

algorithm. Later (in Chapter 6) we will see that the mole detector employed in our novel face recognition scheme is also susceptible to skin shading. Therefore we introduce a method to counteract this effect by performing illumination compensation, based on a variant of homomorphic filtering [GW01].

The underlying simplified reflectance model assumes that for each pixel location (x, y) the image can be described by the product of reflectance and illumination: $I(x, y) = R(x, y) \cdot L(x, y)$. Thus, to recover R one would simply need to divide the image by the illumination. Unfortunately L is unknown. However, the model further suggests that lighting changes slowly and smoothly across an image while reflectance manifests itself in high frequency components. The idea is now to approximate L by a low-pass filtered version of the image, here denoted by $\mathcal{F}_{lp}(I)$. Since the frequencies of function products are not directly

separable [GW01], this is done in the log-domain. The reflectance becomes:

$$\begin{aligned}\log\left(R(x,y)\right) &= \log\left(\frac{I(x,y)}{L(x,y)}\right)\\ &= \log\left(I(x,y)\right) - \log\left(L(x,y)\right)\\ &\approx \log\left(I(x,y)\right) - \left[\mathcal{F}_{lp}\left(\log(I)\right)\right](x,y).\end{aligned} \tag{4.5}$$

The exact type and application (spectral or spacial domain) of filter vary among different homomorphic filtering methods. Here we pursue a novel technique in which an approximation to the illumination contribution is computed by locally fitting bivariate quadratic functions to the logarithm of the image's brightness surface. This variant is related to solutions for curvature estimation in polygonal meshes (*e.g.* [GI04]). The goal is to locally approximate a surface by smooth (usually polynomial) functions, in order to facilitate the computation of the surface's differential characteristics such as the principal directions. Our approach differs from this view in the sense, that we are actually only interested in those contributions of the surface respectively image which can not be explained by the approximation.

Given an image I, the fitting procedure works as follows. For each pixel p we interpret pixels in it's neighborhood N_p as points on a 3D surface. N_p is translated into local coordinates $(x_i, y_i, I_i)_{i=1\ldots|N_p|}$ such that the center pixel p becomes $(0,0,0)$. Then we compute a least-squares fit of the quadratic function

$$z = f(x,y) = ax^2 + bxy + cy^2 + dx + ey + f \tag{4.6}$$

to these points. Let $z_p(q)$ denote the least-squares solution for patch N_p evaluated at pixel $q \in N_p$. The approximation induces an error on each pixel of the fitted patch. As this procedure is repeated for the whole image, every pixel $p \in I$ receives errors from several patches, namely those neighborhoods which somewhere overlap with p. We accumulate these error contributions, separated into positive and negative components:

$$E^+_{ic}(p) := \sqrt{\frac{1}{|N_p|}\sum_{\{q|p\in N_q\}}\Big(\max\big(0, I(p) - z_q(p)\big)\Big)^2} \tag{4.7}$$

respectively

$$E^-_{ic}(p) := \sqrt{\frac{1}{|N_p|}\sum_{\{q|p\in N_q\}}\Big(\min\big(0, I(p) - z_q(p)\big)\Big)^2}. \tag{4.8}$$

If this procedure is applied to the logarithm of an image, the errors can be interpreted as the right side in equation (4.5), where the low-pass filter has been implemented as average of smooth function approximations of the neighborhood. Taking the exponential, brings us back to the image domain and results in two reflectance images. For further reference we denote

$$R^-(I)(x,y) = \exp\left(E^-_{ic}(\log(I(x,y)))\right) \tag{4.9}$$

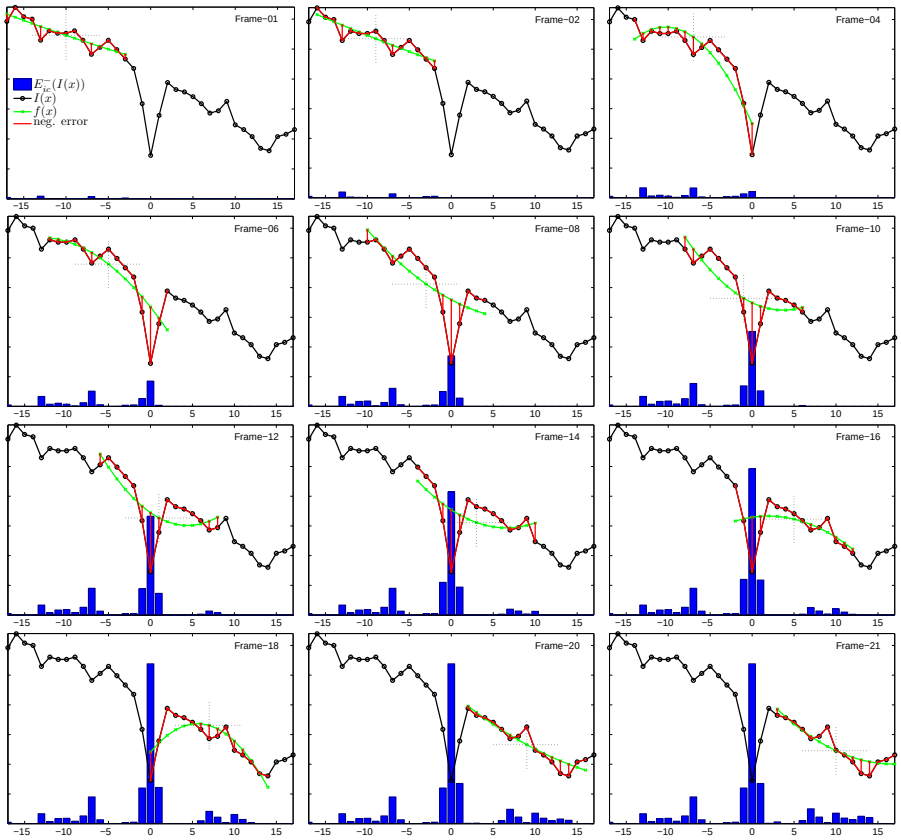


Figure 4.9: Illumination compensation, illustrated on a 1D example. The black line represents an image (here an intensity profile passing through shaded skin with a mole in the center) with some high frequency details and a superimposed shading gradient. A quadratic function is successively fitted (green line) to the local neighborhoods surrounding each pixel. For a given fitting window this results in approximation errors on each of the enclosed pixels. As this scheme progresses through the entire image, the errors are accumulated, separately, according to their sign. Here red lines visualize the negative error component and blue bars the associated accumulated error. The latter is the desired shading free image.

and $R^{+}(I)$ analogous. The reason for separating positive and negative errors in Equation (4.7) and (4.8) is that we can isolate different reflectance contributions. For example $R^{-}(I)$ represents the details with darker appearance like creases, moles or pupils, whereas $R^{+}(I)$ captures brightness peaks like sharp specular highlights.

Figure 4.9 illustrates the described algorithm in a one dimensional sample setting. The objective is to eliminate the global intensity gradient while retaining the distinctive "valleys". For instance these could represent important skin pigmentation features. The consecutive frames show how the quadratic function template locally adapts to the image intensity profile. They mark the location and magnitude of associated approximation errors and show how the

accumulated errors evolve as the pixel of interest p (marked as dotted cross) respectively its local neighborhood N_p traverse the image. Notice in particular the behaviour between *Frame-06* and *Frame-16* where the fitting window passes through the most prominent dent. From a fitting point of view such a detail represents an outlier. Since we only perform standard least squares approximation, the fit $f(x)$ is perturbed by such outliers. However, due to its quite large support the function still has a strong tendency to match the overall shape of the image profile rather than the details spanning only a few pixels. Therefore in average, *i.e.* considering the overlapping neighborhoods, all small scale deviations from the ideal linear or quadratic image gradient will be registered in the cumulative $E_{ic}^{-}(x)$.

An important aspect of this way of collecting errors is the precise localization of outliers in the resulting image. After the completed procedure the error peaks (blue bars) clearly coincide in position and scale with the corresponding valleys in the input image. Compared to other methods, *e.g.* such employing the euclidean distance per patch, the extracted errors here do not blur out. This preservation of sharpness is to a large extent independent of the size of the sliding neighborhoods – the algorithm's only parameter. Figure 4.10 documents the influence of $|N_p|$ by plotting the cumulative errors obtained on the same representative image intensity profile versus increasing neighborhood sizes. Two tendencies can be observed. For one, using too small $|N_p|$ values delivers very prominent and sharp errors but comes at the cost of introducing noise in form of erratic peaks. Secondly, for a quite large range of $|N_p|$ values the ridges representing pronounced errors (and thereby interesting features) remain stable and well delimited, despite an overall decrease of contrast. Since the processing of larger neighborhoods also entails much higher computational cost, the parameter choice is a trade-off between feature stability and runtime.

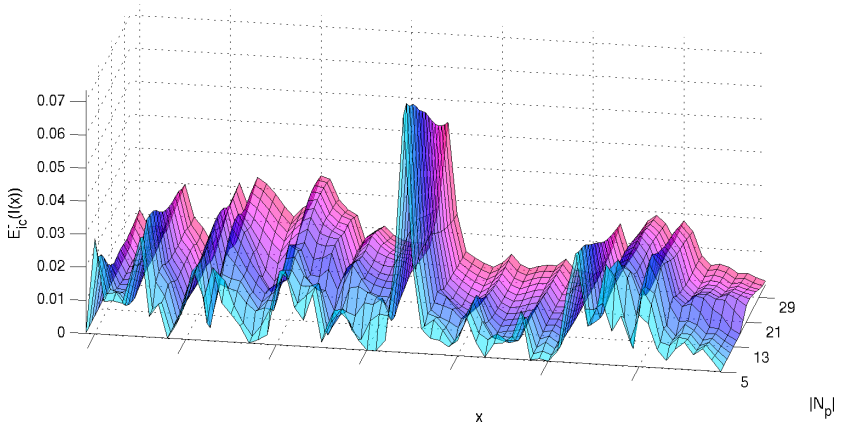
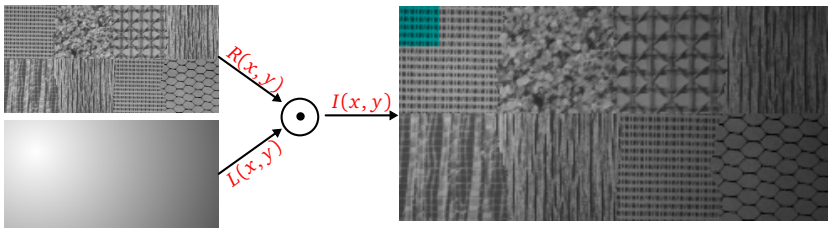
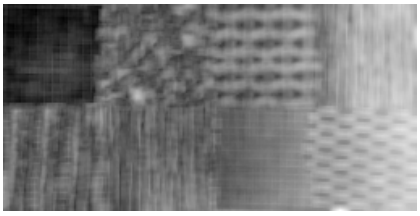


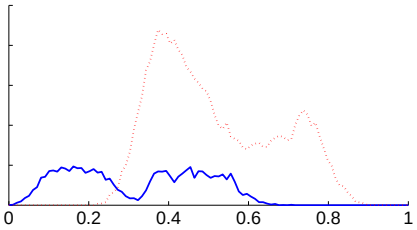
Figure 4.10: Plot of the influence of increasing the neighborhood size parameter in illumination compensation with respect to the resulting cumulative errors, all computed on the same image intensity profile.



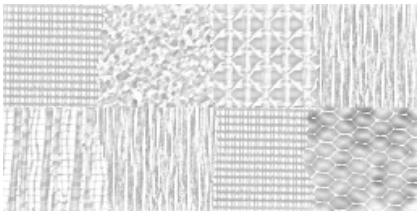
(a) Artificially shaded image provided as input to texture similarity algorithm, $I_{src} = I_{tgt}$, Ω_{seed}



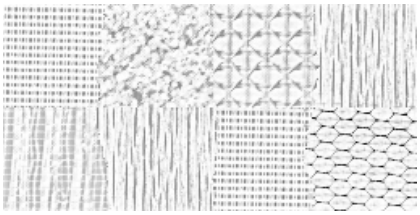
(b) Tex. sim. on shaded image $E^k_{ts}(I)$



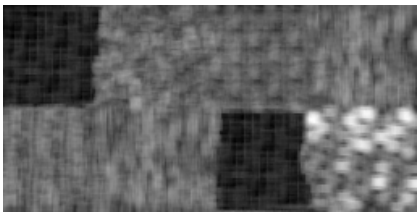
(c) $E^k_{ts}(I)$ histograms, $d_{\varnothing} = 0.563$



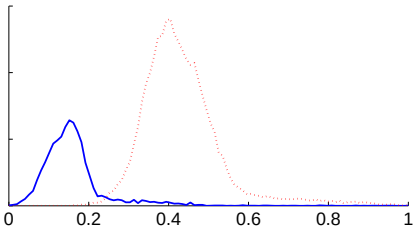
(d) Illumination compensation $R^+(I)$



(e) Illumination compensation $R^-(I)$



(f) Tex. sim. after compensation $E^k_{ts}(R^+(I))$



(g) $E^k_{ts}(R^+(I))$ histograms, $d_{\varnothing} = 0.112$

Figure 4.11: Demonstrates the potential of combined application of illumination compensation and texture similarity in situations where the latter alone fails. A multi-texture image (see also Figure 4.6) is subjected to shading (a), to simulate the conditions encountered in faces. If this modified image is presented as input to texture similarity, the algorithm fails to deliver the desired discriminative texture features ((b), (c)). After illumination compensation the resulting reflectance images ((d), (e)) are shading free. Texture similarity can now be computed on these images, which yields outputs comparable to those obtained on the original unshaded image. In other words: illumination compensation makes the texture similarity algorithm robust against certain lighting artefacts.

Figure 4.12: Examples of binary skin segmentation by thresholding the output of the texture similarity algorithm which has been computed on illumination compensated images $R^{-}(I)$. Compared to previous results in Figure 4.7 and 4.8, without illumination compensation, the combined approach leads to more robust and accurate segmentations.

4.2.2.1 Combined Application with Texture Similarity

The practical benefit of the illumination compensation algorithm can be easily demonstrated on the same multi-texture image that previously served as test case for texture similarity (see page 51). This time the image is first multiplied with a circular intensity gradient (see Figure 4.11(a)) to simulate the shading caused by diffuse reflection on a curved surface. For the same reasons as discussed in conjunction with the application on faces, texture similarity computed on this image (4.11(b), 4.11(c)) does not produce the anticipated discriminative texture features. The corresponding histogram points out, that the range of E_{ts}^{k} values in the darker shaded segment of the sample texture overlaps with those of "foreign" texture. To prevent this, the image is preprocessed with the illumination compensation procedure which outputs the two reflectance images $R^{+}(I)$ (4.11(d)) and $R^{-}(I)$ (4.11(e)). As intended the results are free of shading and they both retain the distinctive texture patterns found in the original image before the illumination component has been imposed. Actually the results are virtually identical to the outcome we get if illumination compensation operates on the original image. Hence, the algorithm appears to be invariant to this kind of blending with smooth gradients. After preprocessing, texture similarity can be employed as usual. It depends on the field of application, whether the positive or negative reflectance component should be used for further processing. In this example both alternatives lead to similar performance. In Figure 4.11(f) we show the output of $E_{ts}^{k}(R^{+}(I))$ since it yields slightly better texture separation, *i.e.* smaller histogram intersection. When dealing with faces, the negative reflectance part clearly carries the more useful information, because all important facial and skin features appear darker than the surrounding skin and thus turn up in $R^{-}(I)$. As for the results of the example: illumination compensation manages to effectively cancel out the artificially introduced shading and enables the texture similarity algorithm to deliver nearly as discriminative features for segmentation as in the unmodified case.

4.2.2.2 Application to Skin Segmentation

We now have the means to refine the binary skin segmentation results simply by replacing the input of the texture similarity algorithm with the illumination compensated reflectance image $R^{-}(I)$. Figure 4.12 presents the intermediary outputs and novel segmentations for the same faces used earlier in Figure 4.7 and 4.8. A direct comparison reveals significant changes in segmentation quality. For the faces in the two top rows (*01022ba*, *01035ba*) the skin segments now extend into the darker area on the right side of the face and on the neck. For the other two faces (*01039ba*, *01034bk*) the improvement is more dramatic. Because of illumination compensation the large gaps, caused by major discrepancies between skin shading in the seed region and the remaining face, have been removed. It is further noteworthy, that in two cases several small holes in the skin segments, introduced by specular highlights, are gone as well. Since we only pass on the negative reflectance component such phenomena can be effectively suppressed. Unfortunately the new combined approach also has a downside. The same principle that urged us to introduce illumination compensation has its benefit in other situations. As explained earlier, E_{ts}^{k} is not purely

a measure of texture resemblance. It is sensitive to any kind of gray level mismatch (clearly, two patches of constant but different gray level can lead to the same error value as two differently textured patches). By eliminating the shading component from an image, we essentially level out the gray scale differences between all patches in favor of pure texture comparisons. This can pose a problem, if an object that should be segmented out, differs from skin primarily by its gray value. One such example is the fuzzy hair on the forehead in face *01035ba* where segmentation results are actually better without performing illumination compensation.

While the thresholding method served us to motivate useful texture features, it is not the final answer to the segmentation task. Instead of attempting to further tune performance on the feature level, we investigate in the next section a more sophisticated alternative segmentation technique, called *GrabCut*. Among all methods, mentioned in the review in Chapter 3, only this algorithm is suited to deal with supervision (in the sense of guiding constraints), based on the currently available skin seeds.

4.3 *GrabCut*

GrabCut [RKB04] is a segmentation method designed primarily as interactive image cutout tool. Its purpose is to serve as powerful alternative to established selection procedures like *Magic Wand* or *Intelligent Scissors* [MB95] which are commonly distributed with professional image manipulation programs. The *GrabCut* framework is based on the efficient *Graph Cuts* formulation of Boykov and Jolly [BJ01] for optimal binary image labeling, but extends their approach through several enhancements. First, the monochrome image model, implemented via histograms, is substituted with a Gaussian Mixture Model (GMM) to facilitate multi-feature based (*e.g.* color channels) segmentations. Secondly, the "single-shot" minimum cut solver is embedded into an iterative energy minimization scheme which alternates between estimation and parameter learning. This contribution is very useful because it enables the algorithm to revise a previously computed segmentation according to changed evidence in the model or the externally defined constraints. A particularly striking consequence of this ability is that it simplifies the interface for (human) guidance. To be precise, that means for *GrabCut*, contrary to other supervised graph cut methods, it suffices to provide seeds only for either the object or the background segment. Thanks to this "relaxed" prerequisite the algorithm is predestined for our problem setting. The third extension targets the issue of matting. In order to produce more realistic results (*e.g.* when pasting cutout objects into other images) for objects with intricate boundaries a novel scheme for border matting is applied, once the object's outline has been determined. Since this approach only operates on narrow strips around the hard segment boundaries, it fails to deal with translucency effects within the objects.

The remainder of this section details the principles of the *GrabCut* algorithm, with regard to our implementation, and explains problem specific extensions and adjustments. We do not perceive the border matting extension as an intrinsic part of the framework. It is merely a post processing of the actual segmen-

tation result that can be replaced by several alternative solutions to the general matting problem, with more convincing results. Therefore this contribution is omitted here. However, the issue of image matting will be revived in the next chapter and in Section 6.2.3.

4.3.1 Problem Formulation

A multi-feature image is an array $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n, \ldots, \boldsymbol{z}_N)$ of d-dimensional vector valued pixels (*e.g.* RGB-tuples), addressed by a single index n. The spacial relationship, *i.e.* connectivity, between pixels is represented in a set $\mathcal{C}$ of unordered pairs $\{p, q\}$ denoting the neighboring elements' indices. The segmentation of an image is expressed as an array of opacity values $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ at each pixel. For hard segmentations these values are limited to $\alpha_n \in \{0, 1\}$ and interpreted as labels for background respectively foreground (object). The color/feature distribution in the image is described by two Gaussian Mixture Models (one for each region) with K components. Their parameters, in dependency of the selected segment, are

$$\boldsymbol{\Theta}_{k,\alpha} = \left\{\pi_{k,\alpha}, \boldsymbol{\mu}_{k,\alpha}, \boldsymbol{\Sigma}_{k,\alpha}\right\} \quad \text{and} \quad \boldsymbol{\Theta} = \bigcup_{\substack{k=1\ldots K \\ \alpha=0,1}} \boldsymbol{\Theta}_{k,\alpha} \tag{4.10}$$

where $\pi_{k,\alpha}$ are mixture weighting coefficients, subject to the constraints: $0 \leq \pi_{k,\alpha} \leq 1$ and $\sum_{k=1}^{K} \pi_{k,\alpha} = 1$. The weights act as a priori probabilities that a pixel $\boldsymbol{z}_n$ was generated by component k so that the mixture density is

$$p(\boldsymbol{z}_n|\boldsymbol{\Theta}, \alpha_n) = \sum_{k=1}^{K} \pi_{k,\alpha_n}\, p\left(\boldsymbol{z}_n|\boldsymbol{\Theta}_{k,\alpha_n}\right) \tag{4.11}$$

with

$$p\left(\boldsymbol{z}_n|\boldsymbol{\Theta}_{k,\alpha}\right) = \mathcal{N}\left(\boldsymbol{z}_n; \boldsymbol{\mu}_{k,\alpha_n}, \boldsymbol{\Sigma}_{k,\alpha_n}\right). \tag{4.12}$$

Following the exemplar approach of Boykov and Jolly, an energy function is defined in such a way that its minimum corresponds to a good segmentation. The function expresses conditions which characterize the desired segment properties in terms of regional coherence and of conformity with a given model, in the form:

$$E(\boldsymbol{\alpha}, \boldsymbol{\Theta}, \boldsymbol{Z}) = \lambda \cdot U(\boldsymbol{\alpha}, \boldsymbol{\Theta}, \boldsymbol{Z}) + V(\boldsymbol{\alpha}, \boldsymbol{Z}). \tag{4.13}$$

The data term $U(\cdot)$ evaluates how well a particular choice of opacity values $\boldsymbol{\alpha}$ reflects the observed pixels $\boldsymbol{Z}$, taking into account the prediction made by the current models:

$$\begin{aligned} U(\boldsymbol{\alpha}, \boldsymbol{\Theta}, \boldsymbol{Z}) &= -\log p(\boldsymbol{Z}|\boldsymbol{\Theta}, \boldsymbol{\alpha}) \\ &= \sum_{n=1}^{N} -\log p(\boldsymbol{z}_n|\boldsymbol{\Theta}, \alpha_n). \end{aligned} \tag{4.14}$$

The second term $V(\cdot)$ encodes boundary properties of the segmentation. It is composed of individual penalties for neighboring pixels which have been assigned to different segments although they are similar (measured in Euclidean

distance):

$$V(\boldsymbol{\alpha}, \boldsymbol{Z}) = \sum_{\{p,q\} \in \mathcal{C}} \delta_{\alpha_p \neq \alpha_q} \cdot \exp\left(-\frac{\left\| \boldsymbol{z}_p - \boldsymbol{z}_q \right\|^2}{2\sigma^2} \right) \cdot \frac{1}{\operatorname{dist}(p,q)} \tag{4.15}$$

In this definition, $\operatorname{dist}(\cdot,\cdot)$ denotes the Euclidean distance between two pixel coordinates, and δ is an indicator function (*true* $\mapsto 1$, *false* $\mapsto 0$) for the property $\alpha_p \neq \alpha_q$. The variance σ^2 controls the extent to which discontinuities are penalized and can be interpreted as the image's noise floor. The coefficient $\lambda \geq 0$ in (4.13) specifies the relative importance of the data term versus the boundary term.

4.3.2 Algorithm

The task is now to compute a segmentation by minimizing the energy function $E(\cdot)$. In contrast to the original *Graph Cuts* formulation, not only the opacity values are unknown but also the true image model parameters. Moreover $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$ are non-trivially coupled. The key to solve this dilemma is to temporarily assume a fixed segmentation. Then the data term (4.14) becomes simply a (negative) log-likelihood function of the parameters, given the data $\boldsymbol{Z}$. The classical "stand-alone" approach to find Maximum Likelihood estimates of a GMM (resp. minimize $U(\cdot)$) is the *Expectation Maximization* (EM) algorithm [DLR77, Bil97]. In order to make the optimization problem analytically tractable this method posits the existence of unobserved data, namely the values which specify which mixture component is responsible for each data item. The algorithm then breaks the likelihood maximization into two steps. In the E-step an expectation value of the hidden data is computed, using a current estimate of the model parameters. In the second M-step the likelihood function is maximized under the assumption that the unobserved data is known. This scheme guarantees a monotonically increasing likelihood and thereby convergence, at least to a local maximum.

The idea of *GrabCut* is to combine this two-step iterative procedure with an additional graph cut optimization step such that each subtask only minimizes $E(\cdot)$ with respect to either set of unknowns. In detail the sequence of operations is:

0. **Initialization**
   The algorithm expects an initial labeling of the image, a so-called trimap, which assigns each pixel to one of three regions: T_F (foreground), T_B (background) or T_U (unknown). T_F and T_B constitute the seed regions, which serve a double purpose. First, they represent hard constraints, *i.e.* they specify a fixed assignment on the respective subsets of opacity values ($\alpha_n = 0$ if $n \in T_B$, $\alpha_n = 1$ if $n \in T_F$) which remains untouched in the following iterations. Secondly, the corresponding pixel values are used to obtain an initial estimate of the GMM parameters. An important feature of *GrabCut* is that incomplete labeling is supported. That means, instead of a full trimap only one of the regions T_F or T_B needs to be provided. Let's assume only pixels in the foreground were marked, *i.e.* $T_B = \emptyset$ and $T_U = \overline{T_F}$. Then a provisional initialization is performed, by setting $\alpha_n = 0$ for $n \in T_U$ and estimating

$\boldsymbol{\Theta}$ accordingly. In this case, however, no hard constraints are used for the background. The iterative minimization will take care of adjusting the preliminary labels, based on constantly refined model parameters (see step 3).

1. **E-step**
   Based on the current model parameter estimates, an expression for the distribution of the unobserved data is evaluated to determine the responsibility $P(k|\boldsymbol{z}_n)$ of each mixture component for each data element

$$p_{k,n} \equiv P(k|\boldsymbol{z}_n) = \frac{p(\boldsymbol{z}_n|k)P(k)}{p(\boldsymbol{z}_n)}. \tag{4.16}$$

   This is done separately for the foreground and background model, using only the pixels in the respective segment as indicated by the current $\boldsymbol{\alpha}$ estimate. The results are two sets of component probabilities (indexed by α):

$$p_{k,n,\alpha} = \frac{\pi_{k,\alpha_n} \mathcal{N}\left(\boldsymbol{z}_n; \boldsymbol{\mu}_{k,\alpha_n}, \boldsymbol{\Sigma}_{k,\alpha_n}\right)}{\sum_{i=1}^{K} \pi_{i,\alpha_n} \mathcal{N}\left(\boldsymbol{z}_n; \boldsymbol{\mu}_{i,\alpha_n}, \boldsymbol{\Sigma}_{i,\alpha_n}\right)}. \tag{4.17}$$

2. **M-step**
   The component responsibilities are used as data weights to obtain new parameter updates (which maximize the likelihood of the joint density of data and hidden values):

$$\pi'_{k,\alpha} = \frac{1}{N} \sum_{n=1}^{N} p_{k,n,\alpha} \tag{4.18}$$

$$\boldsymbol{\mu}'_{k,\alpha} = \frac{\sum_{n=1}^{N} p_{k,n,\alpha} \boldsymbol{z}_n}{\sum_{n=1}^{N} p_{k,n,\alpha}} \tag{4.19}$$

$$\boldsymbol{\Sigma}'_{k,\alpha} = \frac{\sum_{n=1}^{N} p_{k,n,\alpha} \left(\boldsymbol{z}_n - \boldsymbol{\mu}'_{k,\alpha}\right)\left(\boldsymbol{z}_n - \boldsymbol{\mu}'_{k,\alpha}\right)^T}{\sum_{n=1}^{N} p_{k,n,\alpha}}. \tag{4.20}$$

   Steps 1 and 2 combined correspond to computing:

$$\boldsymbol{\Theta}' = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}}\, U(\boldsymbol{\alpha}, \boldsymbol{\Theta}, \boldsymbol{Z}). \tag{4.21}$$

3. **Segmentation**
   In this step the energy function is minimized with respect to the opacity values, given the preliminary $\boldsymbol{\Theta}'$:

$$\boldsymbol{\alpha}' = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, E(\boldsymbol{\alpha}, \boldsymbol{\Theta}', \boldsymbol{Z}). \tag{4.22}$$

   As shown by Boykov an Jolly this can be efficiently achieved by conveying the problem into an ST-graph (compare Section 3.1.2, page 32) and then using a standard min-cut algorithm on this graph to find the global minimum.
4. **Repeat from step 1**
   Each minimization step is designed to decrease the energy $E(\cdot)$ with respect to one set of variables $\boldsymbol{\Theta}$, $\boldsymbol{\alpha}$ in turn. Hence, $E(\cdot)$ must decrease monotonically and convergence to a local minimum (this limitation is inherited from the EM-algorithm) is guaranteed. In oder to define a good stop-condition

it would be unnecessarily costly to explicitly evaluate the energy term at each iteration. Instead one can simply measure the rate of change in either set of parameters. For example, our implementation uses the log likelihood $L = \log(\boldsymbol{Z})$ and stops, if:

$$\Delta L = \frac{L_{\text{curr}} - L_{\text{prev}}}{L_{\text{prev}}} < 1e^{-3}. \tag{4.23}$$

4.3.3 Implementation Specifics

This section summarizes some algorithmic details, which are not mentioned in the original *GrabCut* paper [RKB04] or which deviate from the proposed workflow.

- A major difference is the implementation of "true" soft assignments, *i.e.* of probabilities, of mixture components on given pixels, as a result of incorporating a full-fledged EM-scheme. The authors of *GrabCut* state that EM involves too high computational expenses for negligible practical benefit and therefore use hard assignments, *i.e.* each pixel is associated only to one unique GMM component. While their argument concerning speed may be true, our experiments with an exact re-implementation of the presented theory could not reproduce their claim of decreasing the energy in each step and thus reaching a stable segmentation. [1]
- In both versions the boundary energy depends on a constant which ensures that the exponential term switches appropriately between high and low contrast. The corresponding value β in *GrabCut* is fixed (chosen by optimizing segmentation performance on a small training set of image). In our implementation this noise parameter σ^2 varies in each image and every pixel and is estimated as the sample variance within a 5×5 window around the respective pixel. This approach is particularly useful when the algorithm is confronted with non-color features where value range, contrast ratios and therefore a good global noise threshold are not known in advance.
- The initial foreground and background model are each constructed by a self-contained pass of the EM-algorithm, which in turn uses randomized k-means to obtain a first clustering. The resulting (hard) cluster assignments are then used to compute component priors and sample covariances in both data segments (indicated by $\boldsymbol{\alpha}$). Note, that despite all efforts, the EM-algorithm is sensitive to the choice of starting parameters and may get stuck in a local maximum.
- Another vulnerable point of the EM-scheme is that it's possible to run into degenerative situations. For example, if too few data points are (softly) assigned to a certain component it may cause the associated covariance matrix to become singular or ill-conditioned. In such cases it is useful to impose constraints in form of a minimum diagonal covariance $\boldsymbol{\Lambda}$ (*e.g.* the smallest possible pixel value difference). This is done via eigen-decomposition of $\boldsymbol{\Sigma}_{k,\alpha}$,

[1] Of course it is impossible to rule out coding errors (sample code was not provided). However, the crossover to soft clustering resulted in the desired properties.

after completing the M-step:

$$1. \quad US^2U^T \leftarrow \text{SVD}(\boldsymbol{\Sigma}) \tag{4.24}$$

$$2. \quad S'_{ii} = \max(S_{ii}, \Lambda_{ii}),\ i = 1 \ldots d \tag{4.25}$$

$$3. \quad \boldsymbol{\Sigma}' = US'U^T. \tag{4.26}$$

Whenever covariance constraints are applied, the monotonic behaviour of the likelihood function may be disrupted. A possible alternative could have been to discard offending components in the first place, thereby reducing K.

- Extensions for interactive editing of the segmentation (adding additional constraints) were not implemented, as the purpose in this work is to apply the algorithm only in unsupervised scenarios.

4.3.4 Results

It is straight-forward to apply *GrabCut* to the skin segmentation problem. First, the image domain is constrained to the support Ω_{supp} of the 3D Morphable Model reconstruction. This is achieved simply by re-indexing the pixels within Ω_{supp} and by eliminating all connections from the neighborhood set $\mathcal{C}$ where at least one pixel lies outside this region. In essence, it means that the outer image parts are hidden from the algorithm. This is necessary in order to prevent pixels from the background (clothes, *etc.*) to "pollute" the statistics associated with the two segments. Secondly, the foreground region T_F is defined as the skin seed Ω_{seed} and the background region T_B implicitly encompasses all remaining pixels, *i.e.* $T_B = \Omega_{supp} \setminus T_F$. Then *GrabCut* is run, with $\lambda = 1$, until the stop-condition is reached.

Results on the same set of four faces, used already in earlier demonstrations, are shown in Figure 4.13. A few more faces with a greater variety (and difficulty) of hair appearances are depicted in Figure 4.14 and results for occlusions by glasses are shown in Figure 4.15. In Figure 4.13 *GrabCut* was applied once on the output $E^k_{ts}(R^-(I))$ and once on the original gray scale images. For the latter, the algorithm is unable to produce one good segmentation, although it can be considered state-of-the-art. This points out once more how much better the developed image features are suited to describe the skin region than the raw gray scale data. Besides that, a more interesting question is: How does the algorithm perform in direct competition, *i.e.* given the same input, with the simple thresholding method ?

We observe that *GrabCut* tends to generate less scattered segments with smoother segment boundaries. This is a direct consequence of the influence of the boundary energy term. In particular there are far less false positives (pixels wrongly assigned to skin) so that the outcome can be best characterized as a "safe" expansion of the seed regions over the entire face. On the downside, however, *GrabCut* cuts off too many pixels in highly shaded regions, especially around nose and chin, and thus produces larger gaps in the skin segment. In the end, it depends on the field of application whether thresholding or *GrabCut* should be favored. Concerning the two exemplary applications which motivated this work, the requirements are as different as the methods' results. For mole-based face recognition the segmentation is only one of three steps to extract the

Figure 4.13: Comparison of skin segments obtained from thresholding of texture similarity $E_{ts}^{k}(R^{-}(I))$ (top row) and from our implementation of *GrabCut* (2^{nd} and 3^{rd} row). The center row shows *GrabCut*'s results, applied to the original face image I. In the bottom row the algorithm operated on $E_{ts}^{k}(R^{-}(I))$. In all images the 3DMM support and seed regions are highlighted. Compared to thresholding, *GrabCut* has the advantage of producing fewer small and isolated segments with the downside that it is too conservative in shaded regions.

relevant features: detector, skin filter and saliency filter. Each step can be seen as a high sensitivity, *i.e.*

$$\frac{\#\{true\ positives\}}{\#\{true\ positives\} + \#\{false\ negatives\}},$$

pixel classifier that produces as few as possible false negatives at the cost of more false positives. The latter are only ruled out by combining several such classifiers. Obviously, the skin segments obtained with thresholding match this profile much better than the conservative *GrabCut* results. In the "*Face Exchange*" application a major subtask is to derive a soft segmentation between skin and hair. The class of algorithms that target this problem is reviewed in

Figure 4.14: More segmentation results comparing the thresholding and the *GrabCut* method, both applied on the texture similarity output $E^k_{ts}(R^-(I))$. The image samples were chosen with focus on the problem of segmenting out different kinds of hair.

Chapter 5. It will then become clear that the seed regions available from the 3DMM are insufficient to guide this process. Instead the knowledge of skin segments from a hard segmentation can be used as intermediary result to constrain the soft segmentation. This approach, however, only works if the skin segments exhibit as few as possible false positives, which is exactly what *GrabCut* appears to deliver.

The illumination compensation procedure is motivated by the desire to level out smoothly varying intensity gradients in order to match otherwise similarly textured image areas. The argumentation is, that the combined feature $E^k_{ts}(R^-(I))$ is then able to discriminate the presumed smooth skin from other, not necessarily repetitive, image structures. Figure 4.14 demonstrates that this idea works well for the common outliers from hair and facial organs, especially in connection with the *GrabCut* algorithm. In Figure 4.15 we show some results for faces wearing glasses. It is striking that our method manages to capture the narrow rims of normal glasses in the two left images quite accurately, while it fails to completely segment the far more prominent occlusions by sunglasses in the two right images. This behaviour is caused mainly by the illumination compensation. The $R^-(I)$ measure only registers negative intensity deviations with respect to the surrounding area and up to a certain width (related to the support size $|N_p|$ of the local quadratic function). The absolute gray level information is discarded. As consequence, lighter regions (from reflections or specular highlights) as well as near constant dark areas in the sunglasses' interior attain similarly high values in the $R^-(I)$ image; the first due to suppression of positive error contributions, the latter due to local smoothness. The same problem emerges for outliers from uniformly colored clothing.

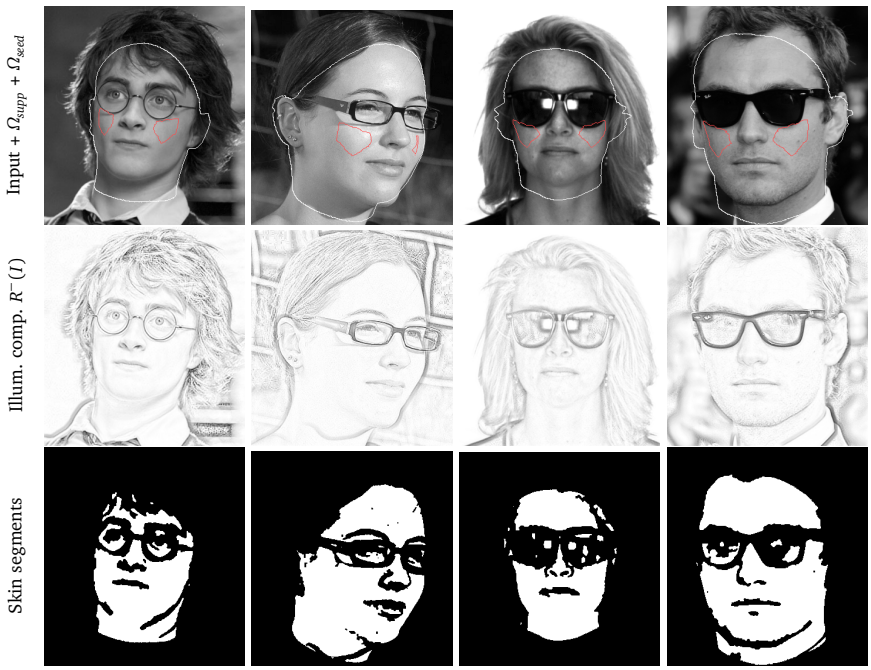


Figure 4.15: *GrabCut* skin segmentation results for faces with glasses.

Chapter 5

Soft Segmentation by Alpha Matting

In image composition tasks, such as our second reference application, "*Face Exchange*", an adequate representation of the object at hand requires a combined description of its color and opacity. While hard segmentations provide only an on/off switch for each pixel, a soft segmentation (usually referred to as *matte*) can assume any value between 1 (opaque) and 0 (fully transparent). This is especially important when dealing with fuzzy objects like smoke or hair, because such objects can only be convincingly mixed with other image material by smooth cross-fading operations. Essentially a matte has to capture the blending effects caused by transparency, aliasing, blur and motion blur during image formation. To demonstrate this necessity, Figure 5.1 compares two compositing results, one obtained using a hard segmentation and exhibiting artefacts, the second one based on a matte. This chapter first provides a brief overview on the natural image matting problem and on a selection of popular solution techniques. Sections 5.2 to 5.5 then focus on the derivation and qualities of the *Spectral Matting* approach.

5.1 Background

In general the process of image matting takes as input an image $\boldsymbol{I}$ which is assumed to be a composite of a foreground image $\boldsymbol{F}$ and a background image $\boldsymbol{B}$. The underlying compositional model specifies that the i^{th} pixel is a convex combination of the corresponding foreground and background colors (also known as the *over*-operation for image blending):

$$\boldsymbol{I}_i = \alpha_i \boldsymbol{F}_i + (1 - \alpha_i)\boldsymbol{B}_i. \tag{5.1}$$

The task is to reconstruct the $\boldsymbol{\alpha}$, $\boldsymbol{F}$ and sometimes $\boldsymbol{B}$ images, from the source image $\boldsymbol{I}$. For 3-channel color images this formulation thus involves determination of 7 unknowns from only 3 equations per pixel, which means the matting

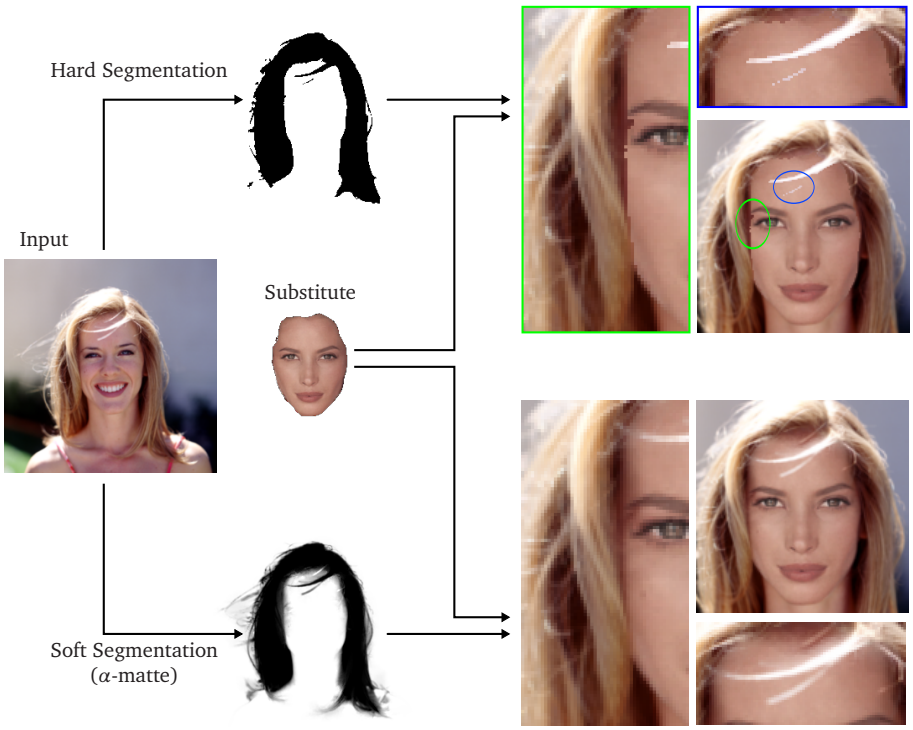


Figure 5.1: Image composition task: a part of the input image is to be replaced by an object (substitute) from another source. Using a hard segmentation to describe the original image occlusions produces artefacts, like visible seams and loss of fine details, while a soft segmentation defines proper cross-fading coefficients.

problem is heavily under-constrained. In order to make it tractable, constraints have to be imposed, usually involving user defined pre-segmentations or ad-hoc assumptions on the components' distribution and smoothness properties. For example, the well known *Blue Screen* technique for live action matting basically works by placing the object against a known constant-color background and by adding simple constraints based on thresholds of color ratios. In the more general scenario of natural image matting, the recording conditions are not controllable and so the problem becomes a lot more difficult. For most recent matting methods the starting point is always a user defined trimap which is supposed to provide a rough segmentation of the image into three regions: foreground, background and unknown (*i.e.* blending of colors).

In the early commercial package *Knockout* [BVD00] $\boldsymbol{F}$ and $\boldsymbol{B}$ from pixels in the unknown region are first extrapolated from colors along the border of proximate foreground/background regions (makes use of smoothness assumption). The α value is then calculated as weighted average of the alleged known colors. This algorithm is quick but is known to perform poorly when the true colors in the unknown region are not consistent with those along the corresponding region boundary, which is often the case.

Ruzon and Tomasi [RT00] introduced simple local color statistics into the

matting process. They partition the unknown segment into sub-regions each of which also encompasses some pixels from the known foreground and background. In these local areas the known colors are then clustered and modeled as mixtures of non-oriented Gaussians. The α value is calculated under the assumption that the observed color $\boldsymbol{C}$ stems from an intermediate distribution which is an interpolation between pairs of clusters from the foreground and background distributions. The algorithm maximizes the probability density of this distribution (in point $\boldsymbol{C}$). The related *Bayesian Matting* approach of Chuang *et al*. [CCSS01] also uses color statistics but employs per pixel estimation of color distributions. A circular sliding window is used to determine for each pixel a subset of neighboring known colors which are then modeled by mixtures of oriented Gaussians. Pixels are processed in a scanning order that marches from the known foreground and background region borders inward, so that previously computed values can be taken into account in the current estimates. The algorithm formulates the search for the optimal matte parameters as a *maximum a posteriori* problem,

$$\underset{\boldsymbol{F},\boldsymbol{B},\alpha}{\operatorname{argmax}} P(\boldsymbol{F},\boldsymbol{B},\boldsymbol{\alpha}|\boldsymbol{C}) = \underset{\boldsymbol{F},\boldsymbol{B},\alpha}{\operatorname{argmax}} \frac{P(\boldsymbol{C}|\boldsymbol{F},\boldsymbol{B},\alpha)P(\boldsymbol{F})P(\boldsymbol{B})P(\alpha)}{P(\boldsymbol{C})} \tag{5.2}$$

$$\hat{=} \underset{\boldsymbol{F},\boldsymbol{B},\alpha}{\operatorname{argmax}} L(\boldsymbol{C}|\boldsymbol{F},\boldsymbol{B},\alpha) + L(\boldsymbol{F}) + L(\boldsymbol{B}), \tag{5.3}$$

with log likelihood $L(\cdot)$. The conditional probability is defined through the difference between the observed color $\boldsymbol{C}$ and a prediction by the parameters, the terms $L(\boldsymbol{F})$ and $L(\boldsymbol{B})$ are obtained as described from labeled image data, and $P(\alpha)$ and $P(\boldsymbol{C})$ are assumed constant. Such methods, which assume relatively simple color distributions for either known region, are reported to work quite well if the distributions do not overlap and if the unknown region in the trimap is small enough.

In *Poisson Matting* [SJTS04] $\boldsymbol{F}$ and $\boldsymbol{B}$ are assumed to be smooth in the unknown region. Their values are initially guessed at each pixel by propagating colors from the foreground/background regions boundary and blurring the result. Then the matte gradient field is approximated as $\nabla \boldsymbol{I}/(\boldsymbol{F}-\boldsymbol{B})$, by taking the gradient of (5.1) and neglecting the gradient contributions in $\boldsymbol{F}$ and $\boldsymbol{B}$ (due to smoothness). A matte is then reconstructed by solving the Poisson equation (with Dirichlet boundary conditions given by the trimap labeling) for a function whose gradients are similar to the approximated matte gradient field. The result is used to contract the unknown region by reassigning pixels which are close (in terms of α) to either foreground or background and the procedure is repeated until convergence. In practice the kind of smoothness assumption, Sun *et al*. use here, is often not met. In these cases the global matte might be erroneous and expensive interactive local manipulations are required to obtain good solutions.

The next section gives a more detailed view on a very recent approach that has several advantages (theoretical as well as practical) over the outlined methods and which we use in Chapter 6 to obtain mattes for the face and the hairstyle.

5.2 Closed-Form Solution

Levin *et al.* [LLW06] suggest as smoothness constraints that the foreground and background values should be assumed approximately constant within a small neighborhood of each pixel, typically a 3×3 pixel window. By that proposition, the representation of discontinuities in the mixture image is implicitly deferred to the matte channel. The key is to realize that for the case of gray scale images this notion can be used to rewrite Equation (5.1) and directly express $\boldsymbol{\alpha}$ in each window w as linear function of the image:

$$\alpha_i \approx a I_i + b, \quad \forall i \in w \quad \text{with} \quad a = \frac{1}{F-B}, b = -\frac{B}{F-B}. \tag{5.4}$$

Using this relation, it is now possible to translate the matting problem into one of global minimization of the cost function:

$$J(\boldsymbol{\alpha}, \boldsymbol{a}, \boldsymbol{b}) = \sum_{j \in \boldsymbol{I}} \left(\sum_{i \in w_j} \underbrace{(\alpha_i - a_j I_i - b_j)^2}_{\circledast^1} + \epsilon a_j^2 \right). \tag{5.5}$$

The last part is a regularization term on $\boldsymbol{a}$ which improves numerical stability and biases the solutions towards smoother $\boldsymbol{\alpha}$. For the practically more relevant case of color images, Levin *et al.* replace the linear model of (5.4) with a 4D linear function:

$$\alpha_i \approx \sum_c a^c I_i^c + b, \quad \forall i \in w. \tag{5.6}$$

On closer examination this model turns out to be more than a transition to multiple color channels. It is shown [LLW06] that (5.6) also generalizes the assumption of constant $\boldsymbol{F}$ and $\boldsymbol{B}$ in each window to one where the foreground and background are each merely linear mixtures of two colors. That means, (5.6) holds as long as all $\boldsymbol{F}_i$ in w (the same for $\boldsymbol{B}_i$) lie on a single line in RGB space. Based on this *color line model*, a cost function similar to (5.5) is defined, only with $\circledast^1$ replaced by $\alpha_i - \sum_c a_j^c I_i^c - b_j$ and the regularization term $\epsilon \|a_j\|^2$.

The construction principle of the cost function (we now always refer to the color case) via overlapping windows couples the parameter values of each pixel to its neighbors and so allows information to propagate through the image. In its original form the function is quadratic in $\boldsymbol{\alpha}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ with $5N$ unknowns for an RGB-image of N pixels. Fortunately, the model coefficients $\boldsymbol{a}$, $\boldsymbol{b}$ can be eliminated, which yields a quadratic cost function of $\boldsymbol{\alpha}$ only:

$$J(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \boldsymbol{L} \boldsymbol{\alpha}. \tag{5.7}$$

Here $\boldsymbol{L}$ is a sparse and symmetric $N \times N$ matrix whose $(i,j)^{\text{th}}$ entry is given by:

$$\sum_{k|(i,j) \in w_k} \left(\delta_{ij} - \underbrace{\frac{1}{|w_k|} \left(1 + \overbrace{(\boldsymbol{I}_i - \boldsymbol{\mu}_k)^T \left(\boldsymbol{\Sigma}_k + \frac{\epsilon}{|w_k|} \boldsymbol{Id}_3\right)^{-1} (\boldsymbol{I}_j - \boldsymbol{\mu}_k)}^{\circledast^2} \right)}_{\circledast^3} \right), \tag{5.8}$$

with $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ denoting mean and covariance of colors in window w_k, $|w_k|$ counting the number of pixels in this window and $\boldsymbol{Id}_3$ being the 3×3 identity matrix. This formulation, remarkably, does not involve the unknown foreground or background colors, *i.e.* it depends only on the observed (mixed) colors.

In order to better understand the properties of $\boldsymbol{L}$, we look at the matrix entries (5.8) while holding a fixed neighborhood w_k. From this point of view the pixel indices i and j each vary independently so that expression $\circledast^2$ describes all pairwise normalized correlations between the colors in this window. These form a symmetric $|w_k| \times |w_k|$ matrix whose rows (and columns) sum to zero. Adding $1/|w_k|$ to each entry then yields row-sums of one. The contents of this matrix are distributed throughout $\boldsymbol{L}$ by adding them to the respective entries belonging to pixel (i, j). Hence, for each time a window comprises a certain pixel $\boldsymbol{I}_{i'}$ the corresponding row $\boldsymbol{L}_{i',*}$, representing all neighborhood relations with this pixel, receives values which increment the total row-sum by one. The term $\sum_{k|(i,j)\in w_k} \delta_{ij}$ counts just how often this happens and therefore is equal. Consequently each row of $\boldsymbol{L}$ sums to zero.

Due to this property, every constant vector is part of the null space of $\boldsymbol{L}$ and thus trivially minimizes $J(\boldsymbol{\alpha})$. In order to obtain meaningful solutions the matte has to be constrained, usually by a user, by pre-determination of some $\boldsymbol{\alpha}_i$ values. The constraints are here supplied as black ($\boldsymbol{\alpha}_i = 0$) and white ($\boldsymbol{\alpha}_i = 1$) brush strokes, so-called "scribbles". In essence this is equivalent to the trimap interface used in other matting algorithms, only that the scribbles can be much more sparse and therefore their definition requires less effort. Given an image $\boldsymbol{S}$ with scribbled pixels, the constrained matte is extracted as:

$$\boldsymbol{\alpha} = \text{argmin}\ \boldsymbol{\alpha}^T \boldsymbol{L} \boldsymbol{\alpha} \quad \text{s.t. } \boldsymbol{\alpha}_i = s_i, \forall i \in \boldsymbol{S}. \tag{5.9}$$

Let $\boldsymbol{b}_S$ be the vector containing the specified alpha values for the constrained pixels and zeros otherwise and $\boldsymbol{D}_S$ be a diagonal matrix whose diagonal entries indicate by 1 or 0 whether the corresponding pixel is constrained. Then, for a large number λ, the matte solution to (5.9) can be computed by solving the sparse linear system:

$$(\boldsymbol{L} + \lambda \boldsymbol{D}_S)\boldsymbol{\alpha} = \lambda \boldsymbol{b}_S. \tag{5.10}$$

5.3 Spectral Matting

The structure of $\boldsymbol{L}$ corresponds to that of a graph's Laplacian. The right part of (5.8), *i.e.* the sum over only expression $\circledast^3$, can be interpreted as affinity function $\boldsymbol{W}_{i,j}$ between two pixels. If $\boldsymbol{D}$ denotes the diagonal matrix with $\boldsymbol{D}_{i,i} = \sum_j \boldsymbol{W}_{i,j}$, which measures the degree of each node/pixel in the graph, then the matrix can be written as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$. Suitably, $\boldsymbol{L}$ is also referred to as *matting Laplacian*. In analogy to hard segmentation methods like *Normalized Cuts* [SM00] (see also 3.1.1) an extension to the described algorithm, called *Spectral Matting* [LRAL07], performs a spectral analysis of the Laplacian to reveal the image's connectivity structure and to facilitate the extraction of better image mattes.

The aim of *Spectral Matting* is to find a decomposition of the image (similar to an over-segmentation) into matting components $\boldsymbol{\alpha}^k$, which represent elementary building blocks and can be used to express the image as composite of multiple layers instead of only foreground and background: $\boldsymbol{I}_i = \sum_{k=1}^{K} \alpha_i^k \boldsymbol{F}_i^k$. It is shown that $\boldsymbol{L}\boldsymbol{\alpha}^k = \mathbf{0}$ also holds for the individual matting components, if one of the following (ideal) conditions is met in every local window:

1. Only a single component is active.
2. Two components are active and the colors in the corresponding layers obey the color-line model.
3. Three components are active and the colors in each layer are constant and linearly independent.

In practice images hardly ever fulfill these assumptions exactly, which means that the matting Laplacian might not have multiple eigenvectors with eigenvalue 0. However, the authors observe that the smallest eigenvectors of $\boldsymbol{L}$ often suffice to extract an approximation of the desired decomposition.

The eigenvectors $\boldsymbol{E} = [\boldsymbol{e}^1 \dots \boldsymbol{e}^K]$ corresponding to the K smallest eigenvalues of $\boldsymbol{L}$ form an orthonormal basis which is unique only up to rotations. Thus the matting components are in the span of $\boldsymbol{E}$ but unlikely to coincide with these vectors. In order to recover the components, Levin *et al*. propose to search for a linear transformation of the eigenvectors subject to the constraints that the resulting vectors sum to one and that the individual components should be sparse (*i.e*. as close as possible to binary vectors). The associated cost function is non-convex and optimized iteratively via Newton's method. Unfortunately this procedure is computationally very expensive and therefore not used in our implementation. A simpler alternative is to apply only a k-means algorithm on $\boldsymbol{E}$. The resulting clusters can be expressed as binary indicator vectors $\boldsymbol{m}^k$ and projected into the span of the eigenvectors:

$$\boldsymbol{\alpha}^k = \boldsymbol{E}\boldsymbol{E}^T \boldsymbol{m}^k. \tag{5.11}$$

All eigenvectors (except of course the constant vector) exhibit the typical fuzzy structure of a matte. The $\boldsymbol{\alpha}^k$ are the closest possible (in the sense of minimum Euclidean distance) points to the associated cluster indicators, which are still contained in the space of eigenvectors. Correspondingly, they inherit the fuzzy boundary property and their "activity" concentrates around the pixels comprised in a given cluster. However, without explicitly enforcing sparseness, the matting components may yet have a global support. Since $\sum_k \boldsymbol{m}^k = \mathbf{1}$, and by exploiting the fact that the constant vector (with unit length) is part of $\boldsymbol{E}$ and orthogonal to all other $\boldsymbol{e}^k$, it is easy to verify that the $\boldsymbol{\alpha}^k$ in (5.11) still satisfy the prerequisite that their sum on each pixel is one:

$$\sum_{k=1}^{K} \boldsymbol{\alpha}^k = \boldsymbol{E}\boldsymbol{E}^T \left(\sum_{k=1}^{K} \boldsymbol{m}^k \right) = \boldsymbol{E}\boldsymbol{E}^T \mathbf{1} = \mathbf{1}. \tag{5.12}$$

The matting components themselves are only an intermediate result. The final task is to group them in such a way that the combined matte exposes some desired object or image region. A grouping is defined through a K-dimensional

vector $\boldsymbol{b}$, with $b_k \in \{0,1\}$, that indicates whether a component contributes to the final matte, *i.e.* $b_k = 1 \iff \boldsymbol{\alpha}^k$ belongs to the foreground component:

$$\boldsymbol{\alpha} = \sum_{k=1}^{K} b_k \boldsymbol{\alpha}^k. \tag{5.13}$$

For any given $\boldsymbol{b}$, the associated matting cost can now be written as:

$$J(\boldsymbol{\alpha}) = (\sum_k b_k {\boldsymbol{\alpha}^k}^T)\, \boldsymbol{L}\, (\sum_k b_k \boldsymbol{\alpha}^k) = \boldsymbol{b}^T \boldsymbol{\phi}\, \boldsymbol{b} \tag{5.14}$$

with

$$\boldsymbol{\phi}_{k,l} = \boldsymbol{\alpha}^k \boldsymbol{L} \boldsymbol{\alpha}^l. \tag{5.15}$$

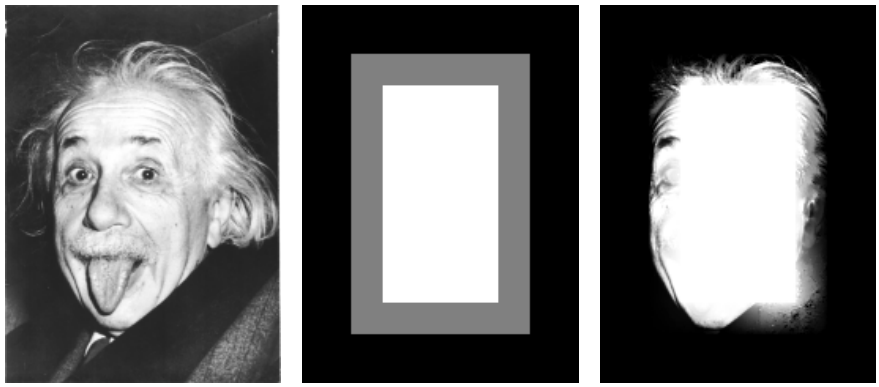
Since the $K \times K$ matrix $\boldsymbol{\phi}$ can be pre-computed, this concept provides a very efficient way to evaluate the cost function for a great number of "hypothesis" $\boldsymbol{b}$. The total number of possible constellations is 2^K. For small K these could be tested systematically for the optimal cost, and by including some heuristics on the expected number of pixels in foreground and background, a matte could be drawn without supervision. In most scenarios this is not an option. First, K may be too high (*e.g.* we use ~ 40 components). Secondly, a global minimum formulation via Equation (5.14) does not incorporate higher level knowledge on the image. Therefore the unsupervised matting task is ambiguous if multiple visually complex objects are to be separated, as it is the case for faces and hairstyle. Like in the *Closed-Form Solution* setting (5.9) a small amount of (user defined) guidance, provided as scribbles, is required to direct and constrain the grouping. Instead of probing all valid combinations of $\boldsymbol{b}$, the idea is to convert the cost (5.14) to an energy function over an *ST*-graph, which can be minimized in polynomial time. In this approach the b_k (and thereby the corresponding matting components) are interpreted as nodes for which an optimal binary segmentation is sought. As explained in Section 3.1.2 (page 32), the associated cost involves sums over regional and boundary properties of the nodes,

$$E(\boldsymbol{b}) = \sum_k E_k(b_k) + \sum_{k,l} E_{k,j}(b_k, b_l), \tag{5.16}$$

where the regional penalties are $E_k(0) = \infty$ if the k^{th} component is constrained to belong to the foreground, $E_k(1) = \infty$ if it is constrained to be background and 0 otherwise. Note, that the nodes b_k are not spatially arranged, so the only sensible topology in the second term includes all pairwise combinations. By defining

$$E_{k,l}(b_k, b_l) = -\max(0, \boldsymbol{\phi}_{k,l})(b_k - b_l)^2 \tag{5.17}$$

one can show [LRAL07] that the boundary cost term approximates, under certain conditions even equals, the original cost function $J(\boldsymbol{\alpha})$. The maximum in the last equation ensures non-negative pairwise energies which is a requirement of the min-cut algorithm that is now used to solve (5.16) and yields the final grouping $\boldsymbol{b}$ and by (5.13) also the $\boldsymbol{\alpha}$ matte.



(a) Input **(b)** Constraints (trimap) **(c)** Closed-form α-matte

(d) Matting comp. $\boldsymbol{\alpha}^1 \cdots \boldsymbol{\alpha}^4$ **(e)** Matting comp. $\boldsymbol{\alpha}^5 \cdots \boldsymbol{\alpha}^8$ **(f)** Spectral α-matte

Figure 5.2: Compares a matte obtained by *Closed-Form Solution* with one obtained by *Spectral Matting* on the same input data. Due to conceptually different handling of constraints the two results are very different. In particular these images demonstrate the superiority of the spectral approach when the problem is "mis-constrained".

5.4 A Practical Assessment

A legitimate question is, how or even if *Spectral Matting* is superior to the closed-form approach. After all, the additional, and for larger images very expensive, eigenvalue decomposition must be justified. One obvious advantage is that the main computational work is separated from the actual quick matte solver which processes the constraints. It allows many different constraints to be tested in a short time. This is ideal for interactive applications where the user edits scribbles and then immediately gets an updated matting result. Since our aim is to develop an unsupervised system, this property is of little value to us. There is, however, another issue concerning the conceptually different way of how constraints affect the matting which makes this algorithm our first choice.

Consider the input image in Figure 5.2(a) with matting constraints specified as trimap 5.2(b) (white indicates foreground and black background). Provided

with this data, the closed-form matting solution outputs the result in 5.2(c). The spectral method, on the other hand, generates a completely different matte 5.2(f), which is already very close to a true segmentation from background to face. Why are the two results so different ?

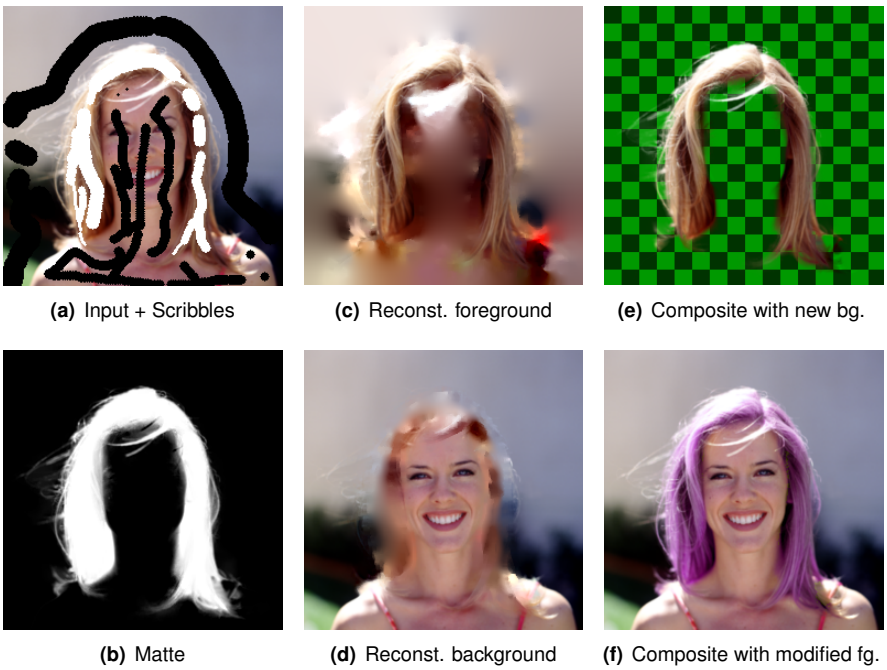
In the closed-form approach the constraints are hard-coded directly, as $\boldsymbol{D}_S$ and $\boldsymbol{b}_S$, into the solver (5.10). The factor λ is chosen so large, as to insure that the least squares approximation adopts the corresponding values virtually untouched into $\boldsymbol{\alpha}$. In contrast to this, in the *Spectral Matting* method constraints merely act as guides. To understand this, recall the regional penalties from Equation (5.16). Each summand $E_k(b_k)$ depends on whether the respective matting component is constrained and if so to which region. That is a binary predicate which is computed by correlating $\boldsymbol{\alpha}^k$ with indicator vectors for foreground and background constraints respectively and afterwards comparing which is higher. If we neglect other factors (like the boundary term), it means that the final result basically originates from majority votes on the number of overlapping pixels of each matting component with either constraint region. For example in Figure 5.2 there are two components ($\boldsymbol{\alpha}^4$ and $\boldsymbol{\alpha}^5$) which have a high correlation with the object marker, and are consequently selected for the matte, while the main support of the remaining components is close to the edge/background.

The fact that *Spectral Matting* does not treat constraints as hard evidence is very convenient for us. It means that we have the option to solve misconstrained matting problems, *i.e.* cases where the region markers are not guaranteed to be located entirely within their respective segment (an example is the background in 5.2(b) which overlaps with a hair patch of the anticipated foreground). In particular, the extreme condition where the constraints leave no pixel undecided (see Section 6.2.3) remains manageable.

5.5 Reconstruction of Foreground & Background

Typically, an application that depends on a matte also requires the foreground colors to depict an extracted fuzzy object in another context or the background colors to reveal occluded image regions after object removal, and sometimes both. Neither the *Closed-Form Solution* nor the *Spectral Matting* approach compute the colors along with the matte. Instead, they formulate a subsequent minimization problem over $\boldsymbol{F}$ and $\boldsymbol{B}$, given a fixed $\boldsymbol{\alpha}$. The conditions which guide the reconstruction are defined by the compositing equation (5.1) and by a smoothness prior which is proportional to the strength of edges in the matte. This corresponds to the notion of the color-line model, where colors are assumed locally smooth, and sudden color changes are attributed to a change in opacity. A cost function based on these terms is:

$$\sum_i \sum_c \left(\alpha_i F_i^c - (1-\alpha_i) B_i^c - I_i^c\right)^2 + |\alpha_{i_x}| \left((F_{i_x}^c)^2 + (B_{i_x}^c)^2\right) + |\alpha_{i_y}| \left((F_{i_y}^c)^2 + (B_{i_y}^c)^2\right) \tag{5.18}$$



(a) Input + Scribbles **(c)** Reconst. foreground **(e)** Composite with new bg.

(b) Matte **(d)** Reconst. background **(f)** Composite with modified fg.

Figure 5.3: Results obtained with the *Closed-Form Solution* α-matting (b) and subsequent color reconstruction (c),(d) procedures. The manually defined constraints are shown along with the input image (a) as white (object) and black (background) brush strokes. Two novel composite images demonstrate the "correctness" of the recovered $\boldsymbol{F}$ and $\boldsymbol{B}$. In (e) the object is placed over a new background, yet structure and color of the transparent hair strands appear as in the original. In (f) the hair color is changed, simply by shifting the hue of the entire $\boldsymbol{F}$ and blending it again with $\boldsymbol{B}$.

where c indexes the color channel and $\cdot_{i_x}$ and $\cdot_{i_y}$ denote the derivatives in x and y direction at the respective i^{th} pixel. The cost is quadratic in its variables and can be minimized by solving a sparse linear system. Figure 5.3 displays a matte and the $\boldsymbol{F}$ and $\boldsymbol{B}$ images recovered by this procedure.

Chapter 6

Applications

In this chapter we explore a series of higher-level applications which can greatly benefit from the custom gray scale image features and the segmentation procedures derived in the preceding chapters. First, a new face recognition scheme is presented, that relies entirely on local skin features for identification [PV07]. Since masking of non-skin areas is crucial to this approach and eventually a large number of images has to be processed, its practicability adheres directly to the ability to automatically compute such segmentations. For the other applications, presented below, automation is not a prerequisite but rather a convenience. Basically the results demonstrated in this chapter could as well be obtained with the aid of human input. This, however, is usually tedious and time consuming work. Section 6.2 presents a method for the 3DMM to deal with outliers. A simple, yet very effective, modification to the original 3DMM fitting algorithm is introduced, that allows us to prevent certain image regions from affecting a reconstruction. Combined with the ability to automatically segment outliers, we manage to render the model robust to such influences. This capability is further exploited in Section 6.3 for the purpose of photo realistic editing of face images, specifically the exchange of faces. Besides handling of occlusions, the focus here lies on methods to counteract artefacts that would give away the manipulation.

6.1 Face Recognition from Skin Details

Facial skin exhibits various small scale structures in the surface (wrinkles, scars) and the texture (nevi – a general term for pigment lesions like birthmarks and moles) that stand out from normal skin appearance and represent potentially valuable references for individual distinction. Among such skin irregularities moles are especially suited for identification. Their predictable appearance, also under changing illumination, facilitates detection. And their numerous appearance in conjunction with unique distribution patterns scales well with extensive galleries. Furthermore moles require no abstract encoding, in contrast to most other facial features. For one, this property allows for a straightforward

handling in terms of comparisons, storage, *etc*., since the encoding is basically transparent and independent of the generating algorithm. Secondly, mole-based features could contribute to an automated facial annotation in human-readable form. This, in turn, could be exploited to search a face database without having to provide a sample face, for example, by formulating a query such as: "*search all faces with a birthmark near the upper right lip*".

6.1.1 Overview

By combining one of the presented segmentation algorithms with additional techniques for detection and validation of moles, we will expose skin features spanning only a few pixels, that are still prominent enough to be used for identification. Relying on such small scale variations is an unusual approach in face recognition. Conventional recognition algorithms are designed to work on low resolution images. For example the well known Eigenfaces approach [TP91], representative for linear appearance based subspace methods, performs dimensionality reduction using PCA on the raw image data and thereby implicitly treats local variations as noise. Also model based algorithms like the Active Appearance Model in 2D [CET01] or the Morphable Model in 3D [BV99] use PCA to model intra class variations. These methods cannot capture small unexpected details in their reconstruction without severe overfitting, which would render the whole method useless. There exist many techniques based on local descriptors using *e.g*. textons, DCT coefficients or Gabor wavelet features. However, none of these methods involve an explicit representation of one of the aforementioned skin features.

Currently the only other known attempt to exploit mole-like features for identification comes from Lin and Tang [LT06]. Their work comprises a multi-layer representation of a face in global appearance, facial features (organs), skin texture and irregularities, which all contribute to the identity. The SIFT framework [Low03] is used for detection and description of irregular skin details which are then combined in an elastic graph for recognition. Their approach tackles stability and distinctiveness issues by validating interest regions using multiple gallery samples per person and by ensuring dissimilarity to normal skin regions. Therein lie the main differences and also drawbacks, compared to the method we develop in this section. For one, the requirement of more than one gallery sample constrains the applicability in many recognition scenarios. Secondly, Lin and Tang neglect to mention how they obtain the partitioning and correspondence for the local regions (organs and skin). It is not clear whether human guided or automated methods are used for this step. Furthermore, all experiments are conducted on frontal views which suggests that their method cannot deal with significant pose variations.

In order to avoid such limitations, our recognition system is designed to take advantage of the 3DMM for face representation, which provides unsupervised (except for the landmark points required to initialize a fitting), pose independent and, to a large degree, illumination independent processing of faces. While the 3DMM itself delivers features that can be used for recognition [BV03a], it is here primarily utilized as a preprocessing to establish the dense correspondence between image pixels and the model's vertices. The reference frame then

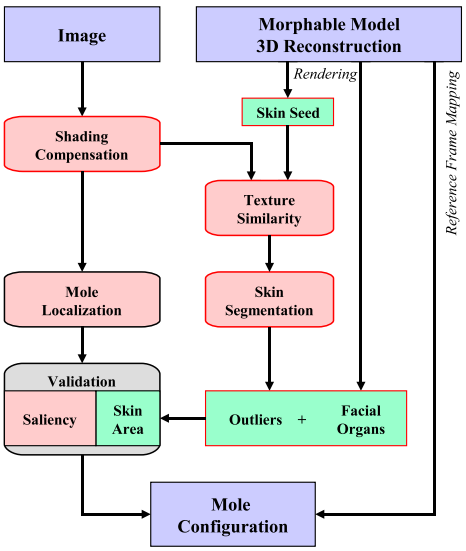


Figure 6.1: Diagram of processing steps and dependencies in the mole detection framework. The left lane shows the main actions to obtain locations and saliency measures for moles. Starting from a Morphable Model reconstruction, the right lanes illustrate how the prior knowledge of the 3D face model is incorporated into the system. Subtasks involved in the skin segmentation chain are highlighted by red borders.

acts as intermediary "universal" coordinate system, so that locations of feature points in different images can be encoded and compared in a pose independent manner. On the downside this dependency on the 3DMM forces us to work around some of its flaws, namely the lack of consistent handling of hair and the occasionally faulty correspondence.

The system to extract mole-like features is divided into three main steps corresponding to the three properties that characterize a local region as birthmark, see also Figure 6.1:

- **Appearance** From distance a mole appears simply as small dark region of circular shape surrounded by a brighter area, *i.e.* a so called "blob". This description also holds under varying viewing conditions (pose/illumination). A very sensitive multi scale detection scheme is employed, see Section 6.1.2, to identify even the most subtle mole candidates.

- **Location** Due to its sensitivity, the detector also responds to typical facial features such as nostrils, corners of eyes, eyebrows and mouth as well as to unexpected deviations like hair strands. These points are not discriminative across individuals, and it is crucial for this recognition scheme that they are rejected. In order to rule out points in such non-skin areas a binary segmentation of the face is computed, using the algorithm delineated in Section 4.2. As opposed to other popular skin detection/segmentation schemes [VSA03], this approach is entirely texture based and therefor requires no color input. Thus we avoid to impose an additional constraint on the input imagery merely for

this step (all other components are already perfectly suited to handle gray scale data).

- **Context** Finally the notion of saliency is introduced in Section 6.1.4 which allows the system to assess the importance of each birthmark candidate for recognition. This procedure takes the relationship between a point's size and contrast and the texture of it's neighborhood into account. In essence it represents a combined measure of uniqueness and confidence. Points below a certain saliency threshold are immediately discarded.

6.1.2 Mole Candidate Detection

Moles are detected by means of normalized cross correlation (NCC) matching. A Laplacian-of-Gaussian filter mask serves as template, because of its very close resemblance to the blob-like appearance of moles. NCC is not scale invariant and the object size is not known a priori. Consequently the matching has to be computed for several resolutions, using templates of varying scale. With a growing number of resolutions a straight forward implementation becomes very inefficient. A theoretically more appealing alternative would be to apply the Lindeberg blob detector [Lin98] which is also used in the SIFT framework and directly searches in scale-space for extrema of differences of Gaussians. In practice the problem with that approach is that there is no minimal spacing of samples (in scale) that will detect all extrema as they can be arbitrarily close together. Given the relatively low image resolution of the database used in the experiments, this means: 1) Using a small number of scale samples, the Lindeberg detector hits only very obvious blobs. 2) In order to detect small but still prominent moles (sometimes consisting only of a few pixels) scale would have to be sampled more densely, up to 30 samples per octave in our experiments. This turned out to be too expensive. Therefore, inspired by Mikolajczyk and Schmid [MS01], the matching here is carried out in separated steps for candidate point localization in space and scale respectively.

At first NCC is computed for a small subset of scales, distributed evenly across the desired search range. Then all local maxima $(x_i, y_i; s_i)$ in the output image of each scale s_k are determined in order to pinpoint candidate positions in 2D. Only these points are further considered. In the second step correlation coefficients for the remaining points are computed, using templates that correspond to mole sizes in the range $[0.5 \cdot s_i, 2 \cdot s_i]$. If the maximum response across these scales is below a fixed threshold the point is discarded. Otherwise the template with maximal correlation defines the points scale for subsequent processing. Handling scale and space independently, has the drawback of causing duplicate point detections, meaning candidates located at different scales and/or coordinates but actually responding to the same feature in the image. Such cases are easy to identify so that all duplicates except for the one with largest scale can be removed.

Another problem arises in areas of changing brightness as cause of shading (changing shape or illumination). The intensity gradients surrounding a mole conflict with the uniform area assumption coded in the mole templates. An example for which the described method fails, can be seen in Figure 6.2. The two

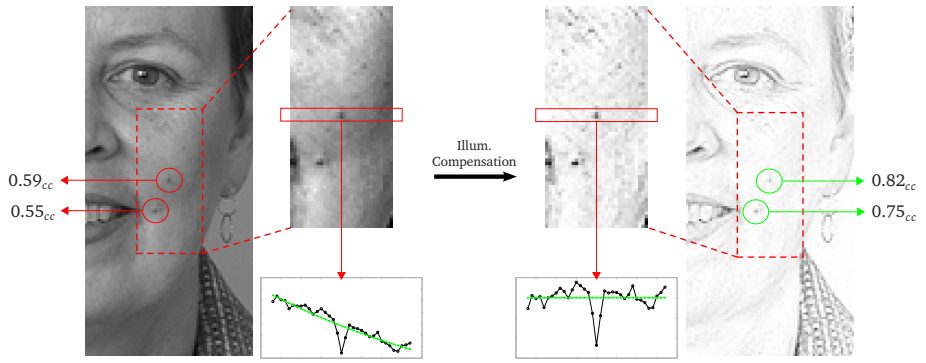


Figure 6.2: Example of two prominent moles where detection in the original image (left) fails. The corresponding magnified section shows multiple gradients in vicinity of both moles, especially noticeable in the depicted horizontal pixel profile passing through the top mole. After applying illumination compensation to this region the detector succeeds on both moles. Intensities in the magnified sections have been normalized for visual clarity.

obvious solutions to handle such situations are not applicable in this scenario. Lowering the correlation threshold would produce too many false positives in less problematic facial regions. Matching against additional templates on multiple scales that also incorporate skin shading, would dramatically increase the computational effort. Instead, the input image is compensated for shading by the previously introduced illumination compensation technique. Since this image transformation removes only gradients and not details, the mole detector can then simply be applied to the output of this procedure, with significantly better results.

Before conducting the experiments, a few gallery images were manually labeled for the locations of subjectively salient moles. The number of scales (range & sample steps) and the NCC threshold were then chosen such that all marked points could be located. Template detection typically reduced the number of candidates for further processing to 1-2‰ of the pixels representing a face.

6.1.3 Skin Segmentation

The template detector does not incorporate any specific knowledge as to where moles can appear. As consequence it may nominate any facial feature with similar appearance, *e.g.* pupils, nostrils or corners of the mouth. Moreover one must expect sporadic hits in areas with hair (beard, hairstyle). Since none of these findings are characteristic for a person, they have a negative impact on the recognition performance and must therefore be eliminated.

This problem is counteracted by incorporating a hard segmentation of the face into skin and non-skin. Mole candidates lying outside the skin segment can then be rejected. The non-skin region is composed of two parts. One part is directly derived from the 3D reconstruction with the Morphable Model, by

defining the subset of vertices which belong to eyes, nostrils and lips, and then projecting this selection to the image domain. Due to imperfect reconstructions the resulting mask may not be very precise. This is taken into account by dilating the mask according to the face's specific measured alignment error. The mask is supplemented by a narrow margin (width also taken from the alignment error) along each side of the model's predicted contours. Detected points in these areas would be problematic because the respective image edges interfere with the saliency computation, and moreover because the correspondence near the contours is less accurate as a consequence of perspective distortion. The second component marks outliers. Chapter 4 proposed two possible solutions for this task: segmentation by thresholding and *GrabCut*. An important point to notice is that both methods also treat larger moles as outliers. Therefore a simple heuristic is employed to prevent such areas from being excluded from further processing: If a mole candidate is located inside a hole of the skin segment, it is still accepted if the gap's size is less than two times larger than the candidate's scale. Note that because of this rule it is not possible to reverse the execution order of detection and then rejection. Of the two available algorithms the simpler thresholding approach was chosen for this job. Although the *GrabCut*-based segmentations generally possess the more favorable properties (see 4.3.4) this method still cuts off too many pixels in highly shaded regions (especially around the nose) and thus produces larger gaps in the skin segment. This is unacceptable for the current application, since it may cause loss of important moles, also due to the aforementioned heuristic, implemented on segment holes.

6.1.4 Local Saliency

Saliency is commonly used as synonym for discriminative power. The more salient a feature is, the better it should be distinguishable from others. The exact definition, however, depends on the actual application. Walker *et al*. [WCT98] formulate this notion over the probability density in feature space and reason that salient features should lie in low density areas. Hence, intuitively saliency corresponds to rarity. In their paper the probability density function (PDF) is approximated by mixtures of Gaussian kernels. Hall *et al*. [HLS02] take on the same definition but use a more accurate Parzen windows technique for density estimation, which is also adopted here.

Having constrained the detected mole candidates to skin regions, the goal is now to define a measure that makes it possible to differentiate between prominent and more or less coincidental hits. The latter may occur in "noisy" regions, *e.g*. in the presence of freckles or stubble, where a single dark spot has no significance. A point's scale and correlation coefficient (from detection) contribute to this assessment but are not sufficient. We therefore combine two more properties, the contrast and the uniqueness of a point with respect to it's neighborhood, into a saliency value.

Consider a mole candidate composed of a group of d pixels, *e.g*. defined by a circular or a square mask centered on the candidate's position (x_i, y_i) with the radius given by its intrinsic scale s_i. The pixel group is stored as a d-dimensional

vector $\boldsymbol{q}$. Further assume a square neighborhood $N_{\boldsymbol{q}}$ centered around the same location. This neighborhood defines the domain over which the saliency of a given candidate should be determined. In the conducted experiments its width remains constant for all images and corresponds roughly to the pixel distance between both nostrils in a frontal view. This minor limitation is tolerable since all images used in the recognition experiments have similar resolutions. From $N_{\boldsymbol{q}}$ all possible shifted and mirrored regions $\boldsymbol{r}_j = T(\boldsymbol{q})$ are extracted (where $T(\cdot)$ denotes the combined transformation of translation and rotation by 90°, 180° or 270° around the midpoint), under the condition that $\boldsymbol{r}_j \subset N_{\boldsymbol{q}}$, $\boldsymbol{r}_j \cap \boldsymbol{q} = \emptyset$. That means the transformed regions have the same shape as $\boldsymbol{q}$ but should not share any pixels with it. Let's assume there are M such regions, which populate a d-dimensional feature space. We then consider a hypersphere with radius ϵ and volume V_ϵ around $\boldsymbol{q}$ and determine the number k of feature points lying within the sphere, as shown in Figure 6.3. The ratio $\frac{k/M}{V_\epsilon}$ is then an estimate for the probability density at $\boldsymbol{q}$. Based on the measurement of k, saliency is defined here as:

$$\mathrm{sal}_\epsilon(\boldsymbol{q}) := \begin{cases} \min\limits_{\boldsymbol{r}_j \subset N_{\boldsymbol{q}}, \boldsymbol{r}_j \cap \boldsymbol{q} = \emptyset} \frac{\|\boldsymbol{q} - \boldsymbol{r}_j\|}{\epsilon} & \text{for } k = 1 \\ \frac{M-k}{M} & \text{for } k > 1. \end{cases} \tag{6.1}$$

The radius is chosen as $\epsilon = d \cdot \sigma^2_{N_{\boldsymbol{q}}}$, with $\sigma_{N_{\boldsymbol{q}}}$ denoting the standard deviation of all pixels in $N_{\boldsymbol{q}}$ but not in $\boldsymbol{q}$. Let us take a closer look at the two cases in Equation (6.1):

- $\mathbf{k > 1}$: As more points fall within the ϵ-sphere, the estimated density around $\boldsymbol{q}$ increases by the ratio $\frac{k}{M}$. The saliency simply decreases by the same rate, taking values in the range $[0, 1)$.

- $\mathbf{k = 1}$: No other feature is closer than ϵ to $\boldsymbol{q}$. The distance to it's nearest neighbor in multiples of the sphere radius is computed. Since ϵ is related to the sample variance in $N_{\boldsymbol{q}}$ the saliency becomes a normalized measure of how much the pixel ensemble in $\boldsymbol{q}$ stands out from the noise in it's neighborhood, ranging from $[1, \infty)$.

The described procedure is applied on every mole candidate location using the illumination compensated image $R^-(I)$, however, **constrained to skin segments**. It is important to mask other potentially blob-like structures from $N_{\boldsymbol{q}}$ in order to prevent them from interfering with the noise and density estimates. An example of evaluated points is shown in Figure 6.4. In the left image all points delivered by the mole detection process have been highlighted. The red squares mark candidates which lie either in non-skin regions or which have a computed saliency $sal_\epsilon < 1$. The remaining points are deemed salient and will be used for identification. Of course not all accepted points are equally "interesting". Figure 6.4 also depicts the processed patches (right column) of the three most salient moles and one of the rejected points. The non-skin parts are masked out (green) and the remaining pixels are normalized. Clearly the saliency correlates with mole size and is higher for points with less variation respectively noise in the surrounding.

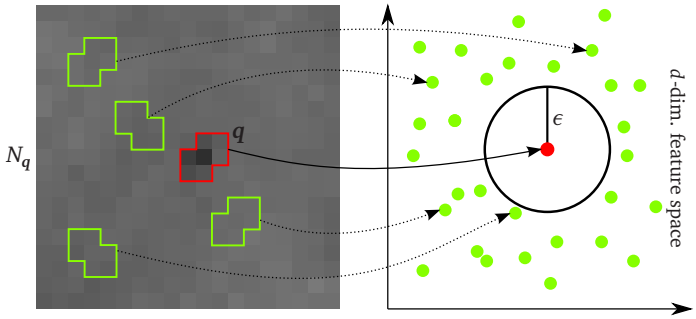


Figure 6.3: Illustration of density estimation for saliency. A pixel group $\boldsymbol{q}$ (red) and all other similarly shaped constellations (green) within it's neighborhood $N_{\boldsymbol{q}}$ populate a multidimensional feature space. The density is estimated by the number of samples lying within a spherical Parzen window around the feature point $\boldsymbol{q}$.

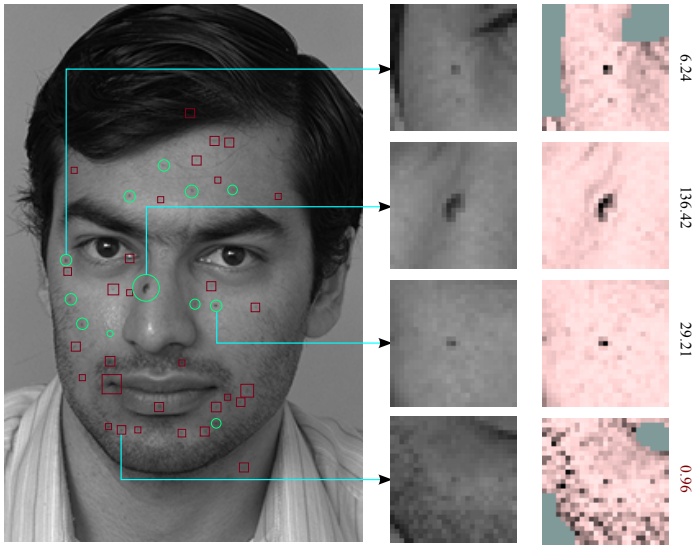


Figure 6.4: Filtering of mole candidates according to saliency. Circles mark points with saliency $sal_\epsilon \geq 1$ which are later used to identify this face. Zoomed neighborhoods of four candidates with corresponding patches from $R^-(I)$ show that this saliency measure indeed relates a point's size and contrast to the surrounding noise and delivers an intuitive measure of importance.

6.1.5 Identification Experiments

This section presents experiments that utilize the described framework for face recognition. For demonstration purposes identification is performed purely based on the previously detected moles on a subset (reported in [BV03a]) of the FERET [PWHR98] face database. This subset consists of gray level images with resolutions in the range of 50-80 pixels eye distance. It contains images of 194 individuals in 11 poses from which the set *ba* (frontal view) serves as the gallery and the sets *bc*-*bh* (head rotated by $\pm 40°, \pm 25°, \pm 15°$) and *bk* (frontal

view with different illumination) provide the probe faces. Most recognition experiments are limited to persons for which the respective gallery image contains at least one mole with a saliency greater than some threshold. The similarity measure between two faces F and G is an ad hoc definition, based on the mole locations in 3DMM reference coordinates and their associated saliency values. It is computed as follows:

1. A proximity threshold σ_{thr} is defined as the average of the alignment errors of both faces (recall Section 2.2, page 24).

2. The saliency values of all moles of a face are transformed to relative weights
$$w_i = \frac{\text{sal}_i}{\sum_{j=1}^{n} \text{sal}_j}. \tag{6.2}$$
This maps the theoretically unbounded sal_i values to a common range, equal for all faces, and relates a mole's importance to the other points.

3. For each mole location i in F the closest point j from G is determined. If the distance between both positions is smaller than $3\sigma_{\text{thr}}$ the point i is considered matched and a matching value
$$v_i = \frac{\min(w_i^F, w_j^G)}{\max(w_i^F, w_j^G)} \tag{6.3}$$
is defined. In this case the point j is removed from G so that it cannot match any other locations in face F. Otherwise (distance greater than proximity threshold) i remains unmatched, the corresponding v_i is set to zero and the evaluation continues with the next mole. The rationale behind Equation (6.3) is to ensure that matched feature points contribute more to the final score if they have similar prominence. If, on the other hand, the values w_i^F and w_j^G differ significantly, this could indicate that two different moles were matched. That is likely to occur if one face exhibits many proximate moles, or if a second face happens to have a mole on/near the same location as the first one. In any case such erroneous allocations must be penalized by a lower score.

4. After a v_i has been assigned to every mole in F, the similarity score is computed as
$$\text{sim}(F,G) = \frac{\sum_{i=1}^{n_F} v_i}{\max(n_F, n_G)}, \tag{6.4}$$
where n_F and n_G denote the number of salient moles in the respective face. Normalization by their maximum takes care of situations, where face G contains much more feature points than F, which clearly cannot all be matched. The larger the number of unmatched points is, the less likely the two faces are assumed to be identical. The other case ($n_F \gg n_G$) is already implicitly covered by point 3, since unmatched moles cause $v_i = 0$.

For a given probe face the gallery face with the highest similarity score is attributed with the inquired identity. Tables 6.1 to 6.3 report various identifica-

	Saliency threshold *(Gallery subset size of 194)*					
	5 *(156)*		**10** *(107)*		**15** *(83)*	
Probe	Fail	**Performance**	Fail	**Performance**	Fail	**Performance**
bc	69	55.77	39	63.55	26	68.67
bd	34	78.20	13	87.85	8	90.36
be	17	89.10	7	**93.45**	4	**95.18**
bf	20	87.18	5	**95.32**	5	**93.97**
bg	47	69.87	24	77.57	17	79.51
bh	68	56.41	30	71.96	21	74.70
bk	42	73.07	22	79.44	13	84.33

Table 6.1: Performance of identification purely based on detected moles. The gallery (frontal view, *ba*) and probes are limited to faces, which contain at least one mole in the gallery with a saliency greater than the denoted threshold. Performance is listed as number of unidentified faces from the respective gallery subset (*Fail*) and in percent.

	Saliency threshold *(Gallery subset size of 194)*					
	5 *(156)*		**10** *(107)*		**15** *(83)*	
Probe	Fail	**Performance**	Fail	**Performance**	Fail	**Performance**
bc	86	44.87	49	54.21	34	59.04
bd	75	51.92	42	60.75	26	68.67
be	52	66.67	26	75.70	20	75.90
bf	64	58.97	34	68.22	25	69.88
bg	91	41.67	51	52.34	37	55.42
bh	98	37.18	58	45.79	40	51.81
bk	87	44.23	51	52.34	34	59.04

Table 6.2: Displays identification results based on detected moles, analogous to Table 6.1, however **without** employing **skin segmentation**. Thereby the overall quality (reliability) of the alleged salient moles is reduced, which leads to an immense decline in recognition performance.

Gallery / Probe	Saliency threshold	Gallery subset size (194=complete)	Fail	Performance
ba / *be*	1	194	42 (65)	78.35 (66.49)
ba / *bf*			46 (64)	76.28 (67.01)
bb / *bc*	1	194	23 (40)	88.14 (79.83)
	5	180	17 (32)	90.55 (82.22)
bi / *bh*	1	194	24 (39)	87.63 (79.90)
	5	184	21 (32)	88.58 (82.60)

Table 6.3: Identification results using only moles as features. The performances were determined for the complete gallery set (*ba*) and for two non-frontal galleries versus probe sets with the respective rotationally closest pose ($\pm 15^\circ$ and $\pm 20^\circ$). The red numbers in brackets denote the results obtained **without** skin segmentation.

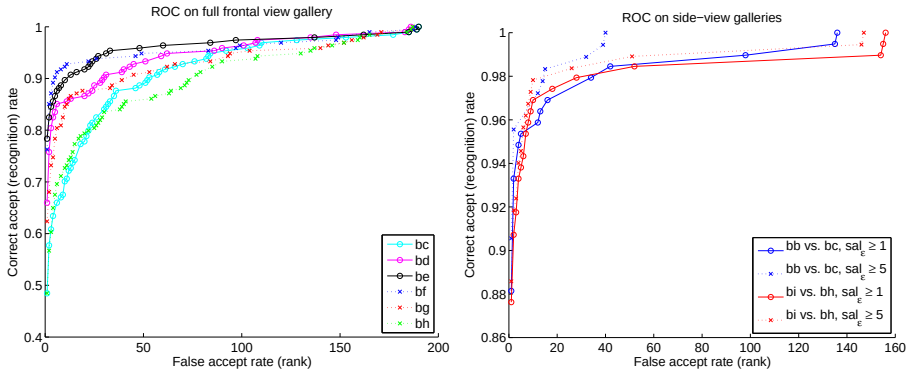


Figure 6.5: ROCs for identification tasks on the FERET face database. In the left plot the gallery is from a frontal view with standard illumination and the probes vary with respect to the pose angle. The recognition rate was measured, using all available salient moles. That means the full gallery could be processed. For the right plot two opposing side-views serve as gallery and the respective probe faces are rotated by 20° towards the camera. These two constellation were tested without and with a low saliency threshold.

tion results and Figure 6.5 plots the according ROCs. A few observations can be made on behalf of these findings:

- The recognition rate drops with increasing rotation angle between gallery and probe, independent of the gallery pose. This is to be expected, since the overlapping area in which moles from both faces can be matched shrinks. Table 6.3 also compares two side-view galleries (*bb* and *bi*, azimuth = ±60°) with their respective probes selected to have the rotationally closest pose (off by 20°). The results are even better than for the frontal gallery, both, in terms of the number of available faces for a given saliency threshold and in terms of recognition rate. A likely explanation is that such side views simply offer a larger surface so that more salient moles can be detected, as this table shows:

Number of detected salient (sal$_\epsilon \geq 1$) moles per database subset									
Pose	*bi*	*bh*	*bg*	*bf*	*ba*	*be*	*bd*	*bc*	*bb*
Angle	−60	−40	−25	−15	0	+15	+25	+40	+60
Moles	3174	3010	2312	2203	2179	2277	2380	3104	3225

- Recognition under different illumination (*bk* set) suffers from the lower contrast between moles and skin which results in lower saliency values and thus more rejections. The total number of detected moles is ∼1600, whereas in all other sets one can account for more than 2100 moles.

- At least 80% of the faces have some prominent moles (saliency ≥ 5) for which we obtain recognition performances above 87%. This is quite remarkable, considering that in average about 5-10 locations, representing less than 0.3% of the pixels in a face, determine it's identity. Enforcing more prominent moles leads to better performance but greatly reduces the number of usable faces. Somewhere between saliency thresholds of 10-15

is the limit beyond which the number of misclassified faces decreases less than the number of usable faces.

A very important conclusion from the experiments is that skin segmentation indeed has a strong (positive) impact on recognition performances. The actual improvement was evaluated by repeating each experiment without using the outlier components in the non-skin mask. That means in this setting candidate points were only rejected as indicated by the Morphable Model prediction of eyes, eyebrows, nostrils and lips. Indirectly the omitted outlier segments also influenced the saliency computations (unmasked outliers affect the neighborhood sample noise and thereby ϵ, see Equation 6.1). Wherever a gallery/probe constellation included a constraint by a minimum saliency, the corresponding subset of faces was determined under standard conditions (*i.e.*, using the moles obtained with skin segmentation) and reused in the experiments without skin segmentation. For the application on a frontal gallery and three different saliency thresholds a comparison of Table 6.1 versus Table 6.2 shows that skin segmentation can lead to performance boosts of over 25%. For unconstrained sets and non-frontals galleries (Table 6.3) the improvement is less pronounced but nonetheless significant.

6.2 Outlier Masking for 3DMM Fitting

As explained earlier in Section 4.1.1, a 3DMM reconstruction can be corrupted by the presence of unexpected respectively unrepresented features in the input face, such as large hair patches or other occluding objects. Since the fitting procedure attempts to adjust the globally acting model parameters in order to also capture such outliers, the overall quality of the fit suffers. This affects shape (visible as misalignment of facial features and contours) as well as texture (leading to noisy/blotchy skin areas and strong visible seams between a rendered reconstruction and the input image). We believe that the lack of proper outlier handling is the greatest weakness of the original 3DMM approach. As a workaround, this section presents a simple and robust method to "hide" problematic areas from the fitting algorithm, based on our previous skin segmentation results.

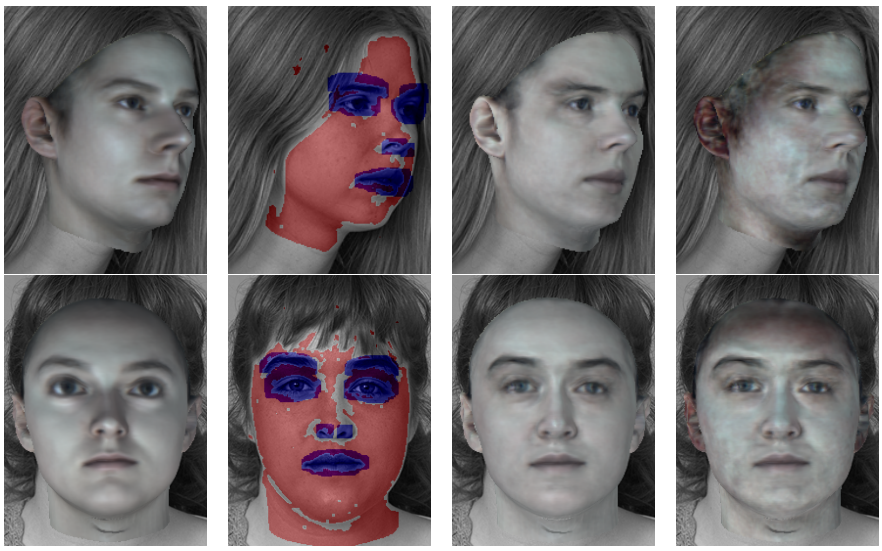
6.2.1 Selective Fitting

Let us assume for the moment, that the locations of facial outliers in the image are already known and provided as binary image $\Omega_{outl}(x,y)$, which indicates by the values 1 or 0 for each pixel, whether it should be considered in the 3DMM reconstruction or not. In order to confine the reconstruction to certain image areas, it is necessary to alter the fitting algorithm. A minimally invasive way to do this, is to replace the image dependent part E_I of the cost function (2.26) with a new term

$$E_{I,\Omega_{outl}} = \frac{1}{\sigma_I^2} \sum_{x,y,\lambda} (1 - \Omega_{outl}(x,y)) \left(I_{\lambda,input}(x,y) - I_{\lambda,model}(x,y) \right)^2 . \qquad (6.5)$$

Figure 6.6: Example for quality improvement by selective fitting. The left column shows the input face and a manually defined outlier mask. The center and right column show the 3DMM reconstruction obtained conventionally (without masking) respectively through selective fitting with the depicted Ω_{outl}.

That means we simply suppress the contribution of matching errors in masked regions. This approach is consistent with the optimization scheme. Recall, that minimization is performed by a stochastic version of Newton's method which evaluates the derivatives of the cost function only for a very small number of randomly selected points in each iteration. If any of these points is marked in the outlier mask Ω_{outl} our modified implementation sets the corresponding gradients to zero, thus effectively eliminating their influence on the parameter update. With a well defined outlier mask this method can lead to dramatic improvements in fitting quality. Figure 6.6 compares a fitting result from the original algorithm with one obtained by selective fitting with manually generated Ω_{outl}. Note, that the overall texture appearance is much smoother and more realistic. Moreover the average reconstruction error $(I_{input}(x,y) - I_{model}(x,y))^2$ within the unmasked region is considerably smaller (drop from 0.0239 to 0.0058 gray scale units, *i.e.* difference factor of 4).



(a) Coarse fit **(b)** Skin + organs **(c)** Selective fit **(d)** Conventional fit

Figure 6.7: Three stages are involved in fitting faces without influence of outliers. A coarse fit (a) is used to estimate locations of facial organs and seeds for skin segmentation (b). All remaining pixels are assumed outliers. A second complete run of the selective fitting algorithm yields the improved result (c). For comparison (d) also shows results obtained without outlier masking.

6.2.2 Automatic Outlier Mask Generation

The question remains: how do we generate outlier masks automatically? The idea is of course to derive them from our skin segmentation solution based on *GrabCut*. So far we are only able to provide a "negative" mask, specifying which pixels are definitely not outliers. By design it does not include eyes, eyebrows, nostrils and lips. For the current application this is too restrictive. These features are crucial for a realistic 3DMM reconstruction, especially when is it supposed to represent the identity of the input face [1], and therefore have to be visible to the fitting algorithm. The easiest way to accomplish that is to render the corresponding (preselected and fixed) set of vertices from the 3DMM reference frame into the image domain and then simply merge the resulting mask with the skin segments.

The attentive reader should have noticed by now that our masking approach seems to conflict with the aim of this section: We want to fit a face using an outlier mask which is to be derived from a skin segmentation and projected facial features which in turn both rely on an existing Morphable Model reconstruction. Currently the only way to cope with this dilemma is to fit the model twice. In a first run we compute only a coarse reconstruction. That means the fitting process is stopped before entering the last phase of iterations in which the individual segments are optimized. This takes about 50-60% less time than a full re-

[1] That is exactly why the 3DMM fitting concentrates in its last phase on these areas by adapting shape and texture individually for eyes, nose and mouth.

construction. A coarsely fitted model already captures many of the face's global attributes. In particular it provides quite good estimates for the head's pose and overall shape so that it is appropriate to derive the seed (Ω_{seed}) and support (Ω_{supp}) regions, as required by *GrabCut*, from such a result. On a pixel/vertex level, a coarse reconstruction is usually still inaccurate. Hence, projected feature locations can be seen only as rough estimates. We compensate for the uncertainty by enlarging (morphological dilatation) the facial organs mask by a small empirically determined amount. After the skin segments and outlier mask have been computed, a second complete run of the fitting procedure then yields the final and improved 3D reconstruction. These three stages are displayed in Figure 6.7.

6.2.3 Alpha Matting for Outliers

An outlier mask (more precisely its inverse) combined from skin and organ segments cannot be expected to be very accurate, due to the following sources of errors:

- The conservative segmentation policy of our *GrabCut* implementation prioritizes correct foreground over false background assignments (for example strongly shaded skin regions are often not classified as such).
- The dense correspondence established by a coarse fitting is still incorrect. As result the projected feature locations may be off by several pixels.
- The dilatation heuristic used to overcome bad correspondence may cover too few or too many pixels. In the latter case it would even override the outlier labeling from the skin segments, because both "opinions" are simply merged by an *or* operation.

As far as the 3DMM fitting is concerned these are minor flaws. For a good reconstruction it only counts that we ruled out the majority of outliers. The fitting algorithm respectively the holistic model representation are then robust against a small number of potentially remaining misclassified pixels.

Basically this means our initial task is solved. However, the now achievable high quality results implicate a novel problem. Let's assume a reconstructed face model is to be used in a graphical application that needs to replace the input face with a synthesized version while keeping the original image context like background, hair and clothing intact. There are several such tasks, *e.g.* facial manipulation for psychological experiments or pose normalization (see also 6.3), which demand for a realistic composite of the two sources. Realistic here means that a human observer should not be able to notice the artificial nature of the manipulation or even of the synthesis. The last image in Figure 6.6 demonstrates that in the presence of facial occlusions (which are the main cause for outliers) a rendered 3DMM reconstruction will no longer "blend in", simply by pasting it over the image. Instead we ask for a seamless fusion by means of the compositing equation and an appropriate matte.

The aim to find a matte which blends between the face (object/foreground) and the outliers (background) is very similar to the initial masking problem. Actually a soft segmentation has richer information and it can be always turned into a binary outlier mask by thresholding. The converse way is also possible but requires more effort as we explain below. Note that the previously introduced outlier mask cannot be used directly as the α-channel. Because of its binary state and the inherent inaccuracies, it produces visible seams to which human observers are very sensitive.

We employ the *Spectral Matting* method to generate an outlier matte. For the current purpose it suffices to compute a solution only within the support Ω_{supp} of the 3DMM reconstruction. Beyond that occlusions are undefined. The results are governed by the following inputs.

- **Parameters** In our implementation the only free parameter is the number K of matting components to use. It should amount at least to the number of independent (in terms of appearance) object units. A higher value produces an over segmentation which then gives the algorithm more freedom in the grouping phase. Above a certain number the results are no longer sensitive to this parameter. To be on the safe side we set $K = 40$.

- **Image data** The matte has to discriminate between skin and outliers just like the skin segmentation. It is therefore reasonable to assume that the algorithm should yield better results if provided with the same input as *GrabCut*, instead of the original image. This was confirmed by many tests. We still introduce two small enhancements specifically for this task. First, the error which represents texture similarity (4.2) is split into accumulated positive and negative components analogous to (4.7) and (4.8). The positive component is discarded while its complementary part constitutes a sharpened version of texture similarity. By replacing $E^k_{ts}(R^-(I))$ with this input we achieve better reproduction of fine details in the resulting matte. Furthermore, we add a second feature channel that holds texture similarity values (again the sharpened version), computed directly on the original image. Thereby we integrate information on regions which differ from the skin seeds primarily by gray level (*e.g.* this would emphasize patches of dark hair with no apparent texture, see also Section 4.2.2.2). A combined input of these two image features has proven to deliver very good matting results.

- **Constraints** The key ingredients for a usable α-matte are the right constraints. In this application the same problem occurs as during skin segmentation: Somewhat reliable seeds are only available for the foreground region. These are the skin mask and with less confidence the facial organs mask. We have no solid indicators for any outlier region whatsoever. Our solution is also comparable to the approach *GrabCut* takes. We use the automatically computed binary outlier mask as marker for foreground and simply assign its complement to the background constraint. Hence, the constraints cover the entire domain. Still, as detailed in Section 5.4, *Spectral Matting* will be able to derive a valid and meaningful matte thanks to the employed grouping technique.

Figures 6.8 and 6.9 display matting results on gray scale and on color images. The first and second column show several faces with various overlapping hair styles and their respective outlier matte. The third column presents the final composite of original image and 3DMM reconstruction which was computed via selective fitting with an outlier mask derived directly from the matte (by thresholding at a fixed value of 0.75). In the fourth column a conventional fitting result is shown. By means of the features we supply, the matting process handles monochrome and color images identically. Therefore, the results exhibit no significant differences in quality. The only place in our implementation where color has an influence is the skin segmentation with *GrabCut*, to which the color channels and the $E_{ts}^{k}(R^{-}(I))$ image are simply presented as combined input. This sometimes helps to better discriminate skin from occluding areas without distinctive texture.

6.3 Face Exchange

The term *Face Exchange* [BSVS04] designates a group of image manipulation tasks which involve the transfer of faces or some of their attributes between two arbitrary images. Besides the objective of the exchange itself the main attention lies usually on obtaining photo realistic results. With traditional tools for image editing such a process is limited to sources with nearly identical viewpoints and illumination conditions. But even then the retouch of details like tonal balance and changing occlusions remains time consuming work. With aid of the 3D Morphable Model it is possible to acquire higher-level knowledge on a scene's content, specifically on illumination, pose and unseen facial regions, which facilitates the automatized transfer of faces. Despite this potential, until now all applications that used the 3DMM capabilities for face exchange had to revert to manual input to fix problems with hair and other outliers. Thanks to the skin segmentation and the derived outlier matte we are, for the first time, able to implement the process completely off-line.

6.3.1 Overview

Three applications have been reported, that implement variants of face exchange by means of the 3DMM.

In-place Morphs Shape and texture features of a 3DMM reconstruction can be used to control the appearance of a face in terms of specific descriptive attributes. This is done by interpreting the PCA coefficients as point and the attribute as direction in the face space spanned by the model's 200 training samples. A linear morph then moves the point along the given direction to obtain the desired change. For example, a morph of a face along the line that connects it with the model's mean face in the direction pointing away from the mean, emphasizes individually characteristic features without changing the face's identity, thereby creating a caricature. Other possible directions encode race, age or gender transformations. The latter has been used to create subtle more masculine

Figure 6.8: Results for the outlier matting problem. The columns from left to right display: 1) input image, 2) outlier matte derived from a coarse fit of the input, 3) composite of input image and a 3DMM reconstruction generated by selective fitting (with outlier mask converted from the matte), 4) conventionally reconstructed face model overlaid on input image.

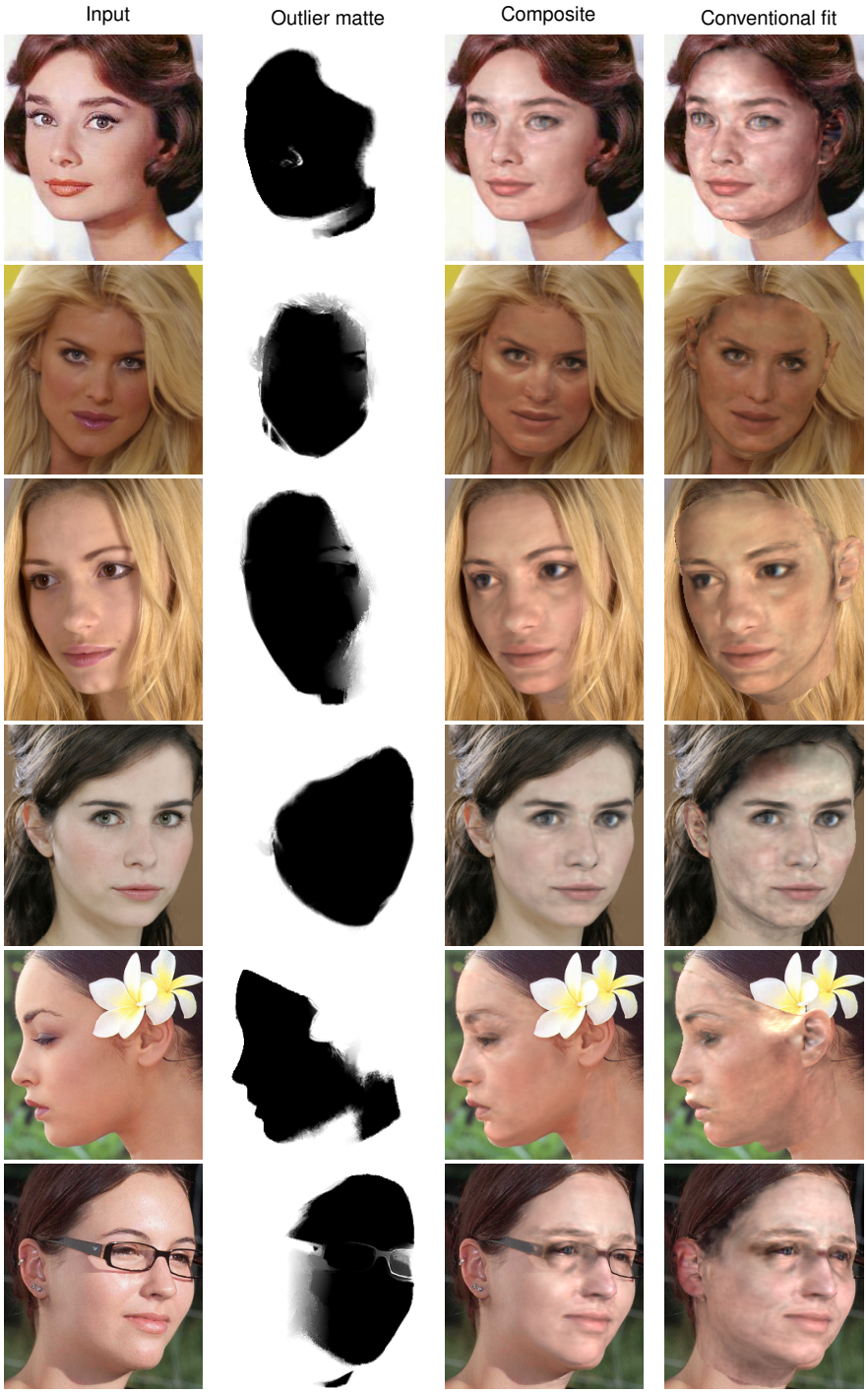


Figure 6.9: Results for outlier matting problem on color images. The denotation of the columns is equal to Figure 6.8.

and feminine variations of a face to serve as stimuli in psychological experiments. Their aim was to determine the influence of gender-specific features on the assessment of applicants. Generating the modified images for the experiments involved considerable human supervision and a specialized software to hide artefacts introduced by the manipulation.

Pose Normalization Most face recognition systems are designed to process frontal views of faces and their performance drops significantly if the faces are rotated. In the *Face Recognition Vendor Test* (FRTV2002) this issue was addressed in an unconventional way by investigating whether a pre-processing step which normalizes the pose [BV03b] could improve the results. Normalization is performed by fitting the non-frontal face with the Morphable Model, so that the 3D structure is recovered and hidden areas can be completed by the model's estimate. Then an image with a frontal view of the reconstruction is synthesized. In the FRTV2002 experiments a standard frontal face image of one person was selected as target. The 3D faces were rendered to this image using its pose, scale and illumination parameters (these were determined by fitting the target face). Finally the target's hair layer was superimposed. The normalization technique proved to be very effective. Provided with the pre-processed faces of 87 individuals, nine out of ten of tested systems showed increased performances compared to unprocessed inputs.

Virtual Hairstyle A generalized notion of pose normalization also supports target images with arbitrarily oriented faces. The transfer procedure is identical: the target image must be fitted to obtain the scene's rendering parameters, the source face to be moved is fitted to recover shape and texture and then rendered into the target scene. However, additional measures have to be adopted to deal with occlusions on both faces and with "uncovered" regions in the target. Blanz *et al*. [BSVS04] proposed such a system to implement virtual try-on for hairstyles. While they used the 3DMM to automatically conduct the face transfer, the actual novelty of creating convincing composites of hair and face relied on manual segmentation.

No matter what the purpose of a manipulation is, all applications are confronted with artefacts that are introduced by pasting a face (*i.e*. its rendered reconstruction) into a foreign image (or in the originating image). The following list gives an overview of the related problems.

- Facial occlusions are the prime cause for corruptions in face exchange results. We must distinguish between those occurring in the source and in the target face. An occlusion (*e.g*. a patch of clothing) of the source face is unwanted information and needs only to be concealed so that it does not impair the 3DMM reconstruction. The same holds, if the extracted texture of the source is to be used. Ideally, the source should always provide a "clean" face model. In contrast, an occlusion in the target image usually represents important contextual information. While it might be necessary to hide such areas from the fitting which determined the target's scene parameters, the main concern is to preserve the respective areas during the transfer. This is more difficult than the masking problem, as it requires higher precision and might involve a soft segmentation step with little or

no prior knowledge on the nature of the occlusion.
- If the novel face is smaller or has a considerably different shape than the target face, the synthesized replacement layer might leave some parts of the original face uncovered. This results in double contours.
- Visible seams may appear between the target image and the mesh boundaries of the pasted source face, since the domain of the Morphable Model only reaches until the forehead and about over half of the neck. Usually the seams are caused by mismatching textures (*e.g.* smooth skin combined with skin that has freckles or wrinkles) or by discrepancies in the illumination. The latter are a consequence of employing in the 3DMM the relatively simple Phong model which can only coarsely approximate many real world lighting conditions and material properties, in particular that of skin. In addition, differences of lower level image characteristics such as the noise print can be perceived.

6.3.2 Counteracting Artefacts

This section is concerned with fully automatized measures that prevent or at least extenuate the above mentioned artefacts. Essentially we have to reproduce the actions a human user would undertake to manipulate critical areas with image editing software. For that task we can only resort to the "knowledge" of the face models and the computed segmentations.

6.3.2.1 Seamless Blending

Visible seams on mesh boundaries can be effectively suppressed by blending the synthesized source face and the target image with a so-called *feather mask*. Similar to a matte, this is an α-channel to be used for compositing (5.1). It represents a linear transition over a fixed distance (*feather radius*) ranging from transparent on the edge to fully opaque on the interior of the foreground object (here the source face). The feather mask for neck and forehead boundaries was defined once for all in the 3DMM reference frame. For a specific face exchange it is then projected with the source face's shape into the target image. In that way the actual width of the transition area can vary and is always proportional to the face's size. The used mask and the attained blending are displayed in Figure 6.10.

6.3.2.2 Shape Scale Adjustment

To compensate for differences in overall shape and size, we slightly adjust the source shape by a global translation and scaling operation in the 3DMM canonical coordinate system. Let $S_\lambda(i)$ and $T_\lambda(i)$ ($\lambda = x, y, z$) denote the i^{th} vertex' coordinate of source respectively target shape. The approach is to find, for each axis independently, scalar parameters a and b such that the transformed source shape S'_λ meets the conditions:

$$S'_\lambda(i) = a \cdot S_\lambda(i) + b \quad \text{s.t.} \quad \overline{\boldsymbol{S}'_\lambda} = \overline{\boldsymbol{T}_\lambda} \quad \text{and} \quad \left\|\boldsymbol{S}'_\lambda\right\| = \left\|\boldsymbol{T}_\lambda\right\|. \tag{6.6}$$

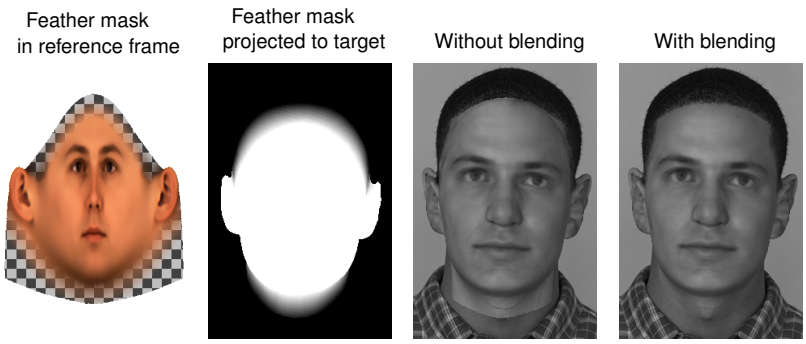


Figure 6.10: Illustration of the feather mask used to seamlessly blending the forehead and neck region of the source face into the target image.

I.e. we want the source shape to have the same mean and magnitude as the target face before it is processed in the rendering pipeline with the target's scene parameters. The solution to (6.6) is determined by a simple quadratic equation. Since the transformation is not necessarily isotropic, it can, for example, adjust a round face to better match an oval shaped face in order to reduce the risk of double contours in the chin region. This is shown in Figure 6.11.

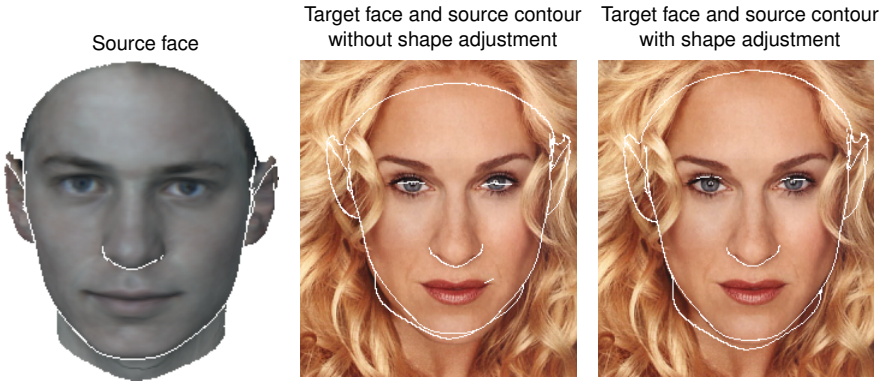


Figure 6.11: A simple shape adjustment can help to prevent double contours during face exchange. Without this procedure the chin contours of the pasted source (left) and the rather longish target face don't match (center). With the adjustment the pasted face will cover the target's original contours (right).

6.3.2.3 Facial Occlusions

In all documented face exchange applications occlusions of the target face need only to be preserved within the support of 3DMM reconstructions, *i.e.* in the area which is actually overwritten by the source face. The solution to this problem is already given by the outlier matte (Section 6.2.3), which we compute anyway in order to derive outlier masks and thus avoid corrupted fitting results. The face

exchange is then realized as a composite of the source face as background and the foreground, reconstructed from the matte and the target image as explained in Section 5.5.

For the source image the respective outlier matte is equally helpful in hiding occlusions, not only from the fitting algorithm but also in the extracted texture. Since texture extraction simply maps to each visible triangle of the reconstructed mesh the underlying portion of the input image, outliers are always extracted, even if they were masked from the fitting algorithm. We did not modify the responsible code in the course of implementing selective fitting because this behaviour is sometimes desired. Instead, we fix the texture retroactively. For this purpose the extraction procedure is also applied to the matte. In the 3DMM reference frame the result is then converted to a binary mask that indicates which parts of the extracted texture are invalid and should be replaced by the estimated texture from the 3DMM reconstruction. A smooth blending between the valid and the replaced region prevents artificial seams. This approach is ultimately only an extension of the default handling for invisible facial areas which are also replenished from the Morphable Model. An example for the technique, and the impact it has on the final exchange result, is shown in Figure 6.12.

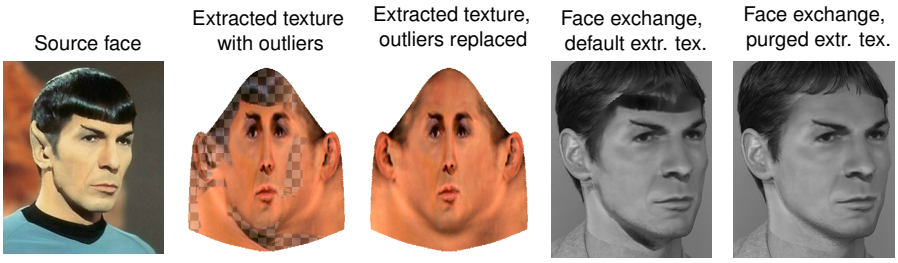


Figure 6.12: Example for the use of an outlier matte to conceal facial occlusions in the extracted texture of a 3DMM reconstruction. The conventionally extracted texture of the depicted source face takes over all outliers. The matte, mapped to the reference frame, indicates which areas should be replaced (displayed transparent) by the model's estimate. While the unmodified extracted texture used in a face exchange will most likely yield unrealistic results, the purged texture can be combined with any target face and context.

6.3.2.4 Background Restoration

Another occlusion related issue emerges if and where the pasted source face is smaller than the target face, which causes the corresponding areas in the original image to "peek through". This leads to distracting double contours. Using the 3DMM reconstructions of both faces, the extent of the uncovered region can be measured as the set difference between pixels in the model's support of target face and replacement face [2]. Actually not all image parts in

[2] Of course this depends very much on a precise 3DMM estimate of the target's shape. If the contours of the reconstruction do not match the true facial contours, the determined region will be

(a) Restoration task **(b)** Attempt with blending **(c)** Mirroring method **(d)** Elastic method

Figure 6.13: Comparison of methods to deal with uncovered original facial areas in a face exchange. The problematic area is located between the outlines of source (green) and target (red), (a). Conventional blending (b) can only conceal the inner contour which impairs the characteristic appearance of the source face. The mirroring method (c) sometimes produces very disturbing artefacts at contour endpoints or mis-aligned 3DMM model contours. Our approach overcomes these flaws at the cost of modifying the background image in a wider area (d).

this set are critical. The shape boundaries of forehead and neck (which do not constitute contours) are artificial and it can be safely assumed that the real face/head of the source extends beyond these limits. It is therefore not harmful if these boundaries lie somewhere within the target's support, since this simply means that the target image delivers the natural continuation of the pasted face/head. Here only a blending (see above) is required to compensate for seams due to surface discrepancies (skin texture, surface normals, *etc.*). For the truly problematic region, located within the target's support but outside the source face's contour (Figure 6.13(a)), blending can only help to reduce or eliminate the visible discontinuity if it hides the source's contour, see Figure 6.13(b). The consequence would be that the resulting image does not correctly reflect the characteristic shape of the novel face. Clearly this contradicts the aim of face exchange. Instead the problem is dealt with by extending the target's background in a mostly texture preserving fashion into the relevant areas.

Given the target face contours, a straight-forward technique for background restoration reflects pixels from the region outside the face along its boundary to the inside by means of a smooth warp field [BBVP03]. Let $d(x,y)$ denote the distance of all pixels to the closest point on the contour. Then the normalized gradient $\boldsymbol{g}$ of this distance map is orthogonal to the contour and defines the reflection direction. For each pixel in the uncovered region the warp is computed as $\boldsymbol{\Delta}(x,y) = -2d(x,y) \cdot \boldsymbol{g}(x,y)$. In many situations, in particular if the background is relatively unstructured, this method works very well. However, there are some notorious conditions where the method introduces further artefacts. Figure 6.13(c) shows such an example. Two problems are noticeable: 1) Although the crossover from background to mirrored content is continuous, a seam appears if the texture on either side has a predominant orientation which does not coincide with the reflection direction (*i.e.* not continuous in gradient domain). 2) The critical area is defined through contours. At intersection points to shape boundaries the warp field therefore abruptly ends which can lead to

misaligned and our corrective measures operate in the wrong place.

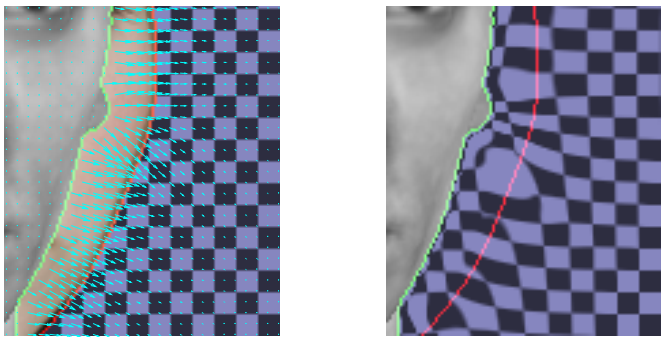


Figure 6.14: Visualization of our "elastic" background restoration method. The area between the inner contour of the source face (green line) and an outer contour of the target face (red line) must be covered by content from the target image background (checker pattern). The left image displays the downscaled warp field (backwards warp) and the right image the resulting effect on the background.

sharp cuts between prior and replaced content. A continuation along the entire target shape boundary is no option. Amongst other reasons, the location of the hair line is unpredictable which could cause verbatim inwardly shaded copies to be used.

We invented an alternative technique which solves both issues by renouncing the properties that texture should be replicated exactly and that the image outside the problematic area is not modified. Our method also employs a smooth image warp. The idea is to treat the background image as an elastic cloth (with the image/texture overlaid) which is pulled into the uncovered region. In order to adapt to the new shape the material has then to stretch within a certain radius of the pulling force. This results in a continuous and smooth displacement of the image content. The whole process is simulated heuristically without the need for complex physical models.

Initially the (backwards) warp is defined only for pixels within the fill region. The direction is computed as gradient in the distance map from the inner (*i.e.* the source's) contour. The length is computed as the distance between the two closest points of either contour (inner and outer) to the current pixel. In that way we adapt the warp magnitude and accordingly the subsequently introduced image distortion to be proportional to the width of the uncovered region. In this state the warp would fill the uncovered region but also produce hard seams at the outer contour like the mirror method. A second stage of the algorithm simulates the elastic stretch by iteratively propagating the field outward. The propagation front is determined as the outer perimeter of the currently defined warp field. For each pixel in the front we compute a weighted (Gaussian kernel) average of the available vectors within a 5×5 window and scale the resulting warp vector with a constant smaller than one (here 0.9). This factor represents the material's elasticity: the lower it is, the faster the field will dissipate. Eventually the magnitude of the warp vectors becomes too small to have an effect, and the iteration stops. The method is visualized in Figure 6.14.

By design our technique prevents any hard edges in the warped image. Another advantage is that it is robust against mis-located contours. As one can see in Figure 6.13(a) the true facial contour on the lower right cheek is slightly outside the model's prediction. With mirroring this part remains in its original position and is even emphasized. Our method, instead, pulls the entire contour towards the face (*i.e.* under the pasted source face) and thereby out of sight, Figure 6.13(d). The drawback is that this manipulation is invasive in the sense that it also changes the target image outside of the necessary region. The visible effects are blurring and distortion of linear structures.

6.3.3 Workflow & Results

With the presented counter measures it suffices to establish one fixed sequence of operations to implement all of the mentioned face exchange applications. The inputs are the fitting results and original images of source and target face. We assume here, that the reconstructions are computed through the three-stage process (see Section 6.2.2): coarse fit, skin segmentation and outlier matting, selective fit. Starting from this point a face exchange is generated in the following order.

1. Occlusions (outliers) in the source's extracted texture are removed by means of the respective outlier matte. Since selective fitting usually provides good quality reconstructions, also in the texture, the use of the extracted texture is optional.
2. If the application involves a specific face manipulation, *e.g.* gender transform, it is performed.
3. The source face's scale is adjusted globally to improve the overall match of support and contours between both faces.
4. Exposed areas of the original face are filled in by our elastic background restoration method. We reduce the impact of the manipulation on non-critical parts of the target's image by limiting it to areas in the proximity of a visible source contour. The visibility in turn is defined by the matte. That means, even if the facial contours predict that parts of the overwritten target face will be uncovered, this "opinion" can be overridden if the matte indicates that the respective area is actually occluded. The result replaces the prior target image in all subsequent steps.
5. The source face is rendered with the target's pose and illumination parameters into the target domain as separate layer $\boldsymbol{B}$.
6. Based on the matte, a compositing foreground $\boldsymbol{F}$ is reconstructed from the target image. Then the original occlusions are restored by creating the face exchange composite with the matte, $\boldsymbol{F}$ and $\boldsymbol{B}$.
7. Using a feather mask for forehead and neck the composite is smoothly blended into the target image. In occluded regions (*i.e.* high α values) the composite is nearly identical to $\boldsymbol{F}$ so that the effect of the blending is only visible in non-occluded skin parts of the source face.

In Figure 6.15 we show the individual image components which participate in the final composition steps. Figures 6.16 and 6.17 display several face exchange results for different poses (also across source and target), lighting and color

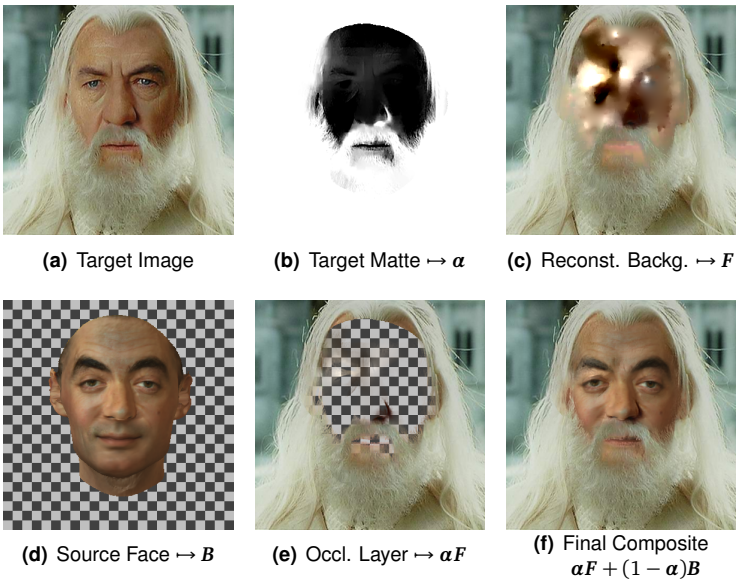


Figure 6.15: Display of the image components involved the three final steps of the face exchange workflow.

conditions and a variety of facial occlusions by hair and clothing. In all images we used the source's extracted texture (with substituted outliers).

Besides certain technical details of how artefacts are resolved, there is a conceptual difference in the way we address face exchange, compared to the "virtual hairstyle" approach of Blanz *et al.* [BSVS04].

In their method three layers are distinguished: the background (with applied restoration), the intermediate novel face and the top occlusion layer. This notion of scene composition is physically plausible but can also be tricky to handle correctly. There are situations when parts of the novel face model must not overwrite the background image in order to give satisfactory results. Examples can be seen in Figure 6.9 (page 99) on the first and second last row in the 4$^{\text{th}}$ column (even though the images do not show actual face exchanges the effects are the same). In both cases the synthesized neck significantly protrudes the target's skin region. Blending does usually not hide this problem entirely. Instead, the affected region in the target image should be included in the occlusion layer, although it belongs in fact to the scene background. Apparently the method of [BSVS04] delegates this responsibility to the human user and manual editing.

In our approach the outlier matte provides exactly the missing information, as it realizes a soft segmentation from skin to any other region, including the image background. Formally, matte and blending mask are defined with respect to the same foreground channel and therefore can be multiplied to combine both effects in one compositing step. All together that means, we only have to deal with two layers: the novel face below and the target image on top with a combined α-channel that defines which areas of the novel face show through.

Figure 6.16: Automatically computed face exchange results for male subjects.

Figure 6.17: Automatically computed face exchange results for female subjects.

Chapter 7

Conclusion

This thesis presented novel techniques to automatically compute segmentations of face images into areas composed of either skin or of facial organs and extrinsic parts. As a link between pixel-level detail information and high-level object description, these segmentations were devised to provide valuable guidance to enhance existing face processing algorithms as well as to help establish completely new approaches for face analysis. We demonstrated the practical impact of our results on the basis of three applications. In conjunction with the 3D Morphable Model, the non-skin segments served as main component to create outlier maps. Combined with a modification of the 3DMM fitting algorithm, that we introduced, these maps were used to significantly improve the visual quality of 3DMM reconstructions in the presence of outliers. This capability was reapplied and extended for the *Face Exchange* scenario. In order to mix hairstyle or other occlusions from the image context of one person with the face of another person, a proper blending, based on the occluding segments, was computed with the aid of a general purpose matting algorithm. The key benefit was here, that our segmentation could act as substitute for user supervision to this algorithm. Together with the improved 3DMM reconstructions, this method represents a substantial contribution to *Face Exchange* applications, because it allows, for the first time, to perform this type of manipulation automatically, for a large variety of face images, and without qualitative losses. Our third accomplishment is a novel approach to face recognition. The idea was, to exploit the distribution of local skin irregularities across a face as personal features. For this purpose, we developed a framework for mole detection and validation, in which the segmentation of skin regions constituted an essential step, to suppress false positives. We evaluated the discriminative power of the mole-based features by conducting identification experiments across pose and illumination on a subset of the FERET face database, containing 194 individuals. The results confirmed that it is possible to determine a person's identity, based on only a few well-chosen pixels, provided that the face exhibits sufficiently prominent moles.

As stated earlier, the skin segmentation procedure and subsequent applications rely on prior knowledge of facial structure, that is incorporated through the 3DMM. In connection with these hints we have to consider certain technical

and methodical restrictions. An important piece of structural information is the model's support region Ω_{supp}. It is used in *GrabCut*, to ensure that the statistics for skin and non-skin models are not influenced by unrelated and unpredictable image areas, like the background. The downside of this dependency is, that all results are also confined to the model's domain. Currently we cannot offer a generally reliable strategy to extend the segmentation beyond these borders (at least not without resorting to user input), which means that the field of further application is rather limited.

Another element of prior knowledge, that we employ, is Ω_{skin}. The skin seeds play a central role in the derivation of our texture features and in the initialization of the hard segmentation techniques. Strictly speaking, they are the key to automation. Their size and location on the cheeks was manually selected, with regard to our target applications, such that, in the majority of faces, they would not be corrupted by the anticipated outliers. In other problem settings, alternative selection schemes may be more appropriate. As an example, let us consider a subset of the AR face database [MB98] which is often used as benchmark for face recognition under occlusion. It contains disguised face images, wearing either dark sunglasses (*e.g.* page 69, Figure 4.15, 3rd image) or scarfs that cover the lower half of the face. In order to prevent overlap with these outliers, in addition to hair *etc.*, the seeds would have to be chosen adaptively. One way to tackle this problem could be to define a larger number of small candidate cells, similar to an over-segmentation, in typical skin regions (including chin and forehead). A certain number or percentage of individual cells could then be selected according to a best-match criterion, based on skin related image properties, such as smoothness, within the cells' support. Also cell selection itself could be approached as an unsupervised binary segmentation task, with a graph based formulation.

The third 3DMM hint is the facial organs mask. It is utilized during outlier masking, to reinstate the represented areas as accepted facial "inliers", before the image and the segmentation are further processed, either by fitting or by matting. Since we know that the organs mask is inaccurate, this can be problematic. For example, compare the segmentation of the second face in Figure 4.15 with its corresponding matting result in Figure 6.9 (page 99): *GrabCut* managed to extract the detailed structure of the eyebrows and glasses, but the mask then canceled out some of this information, so that the matting algorithm perceived the whole area as belonging to the face. Conversely, if the prediction of the organs covers too few (or even none) of the actual associated pixels, the resulting matte in these areas will be incomplete. This in turn can impair a face exchange, because the affected part in the target face is not overwritten by the source face. In practice, these effects only occur infrequently. Still, the concept, that inaccurate model predictions are used to override potentially pixel-precise segmentations, remains unsatisfactory.

The same problem can be interpreted from a different perspective. Within the set of non-skin components we would like to further identify which pixels originate from facial organs and label them accordingly. This knowledge would bring us closer to a complete semantic decomposition of a face and in particular circumvent the aforementioned drawbacks. In order to discriminate between true outliers and organs, we believe, that a model-based segmentation

step should be adopted, which is more flexible than the 3DMM and takes shape constraints into account. Under such premises, a popular choice would be an *Active Shape Model* (ASM) [CCTG95]. The implementation could be accommodated to our setting as follows:

- Sample shapes of the facial organs are extracted from the 3DMM training data, projected (via 3DMM reconstruction) into the current image, and used to learn the Point Distribution Model (PDM). This avoids manual labeling and definition of correspondence for the training points.
- For the ASM fitting, the initial shape points can be taken from the 3DMM reconstruction of the current face. Here, wrong locations are not critical, since the fitting adapts scale and translation independently of the shape. This facilitates automatic initialization.
- The statistical model of image appearance around each model point is replaced with a constant target profile in normal direction, which matches image edges with specific gradient sign (*e.g.* dark inside the shape and bright on the outside). Accordingly, instead of the original face photograph, the illumination compensated image $R^{-}(I)$ or even the segmentation (*i.e.* its binary image) are used to fit the model. This forces the shapes to adapt to the prominent edges in either image, but only within the limits of the PDM constraints.

We successfully tested this idea for exact segmentation of eyebrows in frontal images. However, further research and experimentation would be required to make it applicable under non-frontal poses and to obtain reliable results for the other facial parts.

Our mole-based recognition framework leaves room for improvements, as well. In the current state it should only be taken as proof of concept because it cannot perform in the same way as conventional methods. The critical issue is the lack of a backup solution to support cases where no moles are present. In practice it is clearly not an option to constrain a face database only to people which have such prominent features. Hence, the true potential of our framework lies in the fusion with a complementary method. In the simplest form this could be setup as a cascade: The mole features would be used to narrow down the number of candidate faces in the gallery, which is then processed by a default face recognition routine. Basically, two criteria for pre-selection are conceivable.

1. The probe face has salient moles. In this case all gallery faces are matched against these features, and a certain percentage of best matches (*i.e.* lowest ranks) is passed on to the second method. In this context it could be beneficial to investigate other robust means of measuring the similarity between two mole distributions, for instance by employing some form of graph matching.
2. The probe face has no salient moles. Assuming that the detection accuracy is high enough, it makes then sense to drop all gallery faces which do have moles in the area corresponding to the visible parts of the probe face.

Faces with too many moles would have to be handled differently. [1] With growing numbers (and denser arrangement), the assignment between two point sets

[1] For example, in the *bb* set from our experiments there are nine faces which exhibit more than

with inherent uncertainty of correct localization becomes ambiguous. Therefore, such faces in the gallery should always be presented to the default method, independent of their status in the list of best matches. For probe faces with this problem one should only resort to the backup solution. The outlined approach would have the advantage of supporting collaboration with any conventional face recognition technique. However, a real fusion, on the feature level or by using a theoretically sound probabilistic formulation, is still out of reach.

The mole recognition framework is not the only contribution which could qualify as an advancement in the field of face recognition. Compared to related methods which address the outlier problematic, our outlier masking approach has a conceptual advantage. The outcome is neither encoded on a feature level nor does it adhere to a particular local partitioning. Instead, we simply provide a comprehensive binary indicator image. This independence of a specialized representation provides the opportunity to extend its application to face recognition methods, other than the 3DMM. Given that an algorithm can be modified to incorporate the outlier map, our hope would be that the additional information helps to improve its recognition performance. It might even be possible to establish outlier masking on a grand scale, as a general preprocessing step for face recognition. Inspired by pose normalization, the idea would be, to synthesize a novel image of a face, in which all occluded areas have been replenished with reasonable "face-like" content. This would have the benefit of being non-intrusive.

10 moles of saliency $sal_e \geq 5$, one face even counts 27 such moles. For lower thresholds it rises up to 53. These are not false detections. The affected faces actually have conspicuous pigmentations, *e.g.* strong freckles, which are partly responsible for the high false alarm rates in the corresponding ROCs (see Figure 6.5, page 91).

List of Figures

2.1 Dense correspondence and 2D parametrization of faces via reference frame. 17

2.2 Independent morph of texture and shape in registered 3D scans. 19

2.3 Visualization of principal components in the 3DMM. 21

2.4 Illustration of the point/region mapping between an image and its 3DMM reconstruction. 25

3.1 Drawback of the min-cut segmentation criterion. 29

3.2 Illustration of segmentation principle via ST-graph cuts. 33

4.1 Bad 3DMM reconstruction due to facial expression. 45

4.2 Misaligned lips in 3DMM reconstruction. 45

4.3 3DMM reconstruction corrupted by outliers / occlusions. 46

4.4 Example of misaligned features in visually correct 3DMM reconstructions. 47

4.5 Illustration of texture similarity procedure. 50

4.6 Influence of patch size parameter in texture similarity. 51

4.7 Skin segmentation by thresholding of texture similarity measure. 53

4.8 Negative impact of illumination on the thresholding approach. . 54

4.9 Illustration of illumination compensation procedure in 1D. 56

4.10 Influence of neighborhood size parameter in illumination compensation. 57

4.11 Toy example with combined application of illumination compensation and texture similarity. 58

4.12 Binary skin segmentation results on faces. 59

4.13 Comparison of skin segments obtained with thresholding and with *GrabCut*. 67

4.14 More results comparing thresholding with *GrabCut*. 68
4.15 *GrabCut* skin segmentation results for faces with glasses. 69

5.1 A typical image composition task. 72
5.2 *Closed-Form Solution* versus *Spectral Matting*. 78
5.3 Foreground and background reconstruction from input image and computed matte. 80

6.1 Diagram of mole detection framework. 83
6.2 Mole detection before and after illumination compensation. . . . 85
6.3 Illustration of density estimation for saliency. 88
6.4 Mole candidate selection by saliency. 88
6.5 ROCs for identification based on moles on FERET database. . . . 91
6.6 Example for quality improvement by selective fitting. 93
6.7 Automated selective fitting versus conventional fits. 94
6.8 Outlier matting results on gray scale images. 98
6.9 Outlier matting results on color images. 99
6.10 Illustration of feather mask for seamless blending. 102
6.11 Example for shape adjustment. 102
6.12 Example for the application of an outlier matte to conceal facial occlusions from the source face. 103
6.13 Comparison of methods that deal with uncovered facial areas. . . 104
6.14 Visualization of elastic background restoration method. 105
6.15 Image components used in final steps of face exchange. 107
6.16 Face exchange results for male subjects. 108
6.17 Face exchange results for female subjects. 109

List of Tables

6.1 Performance of identification from moles under pose variation. . 90

6.2 Performance of identification from moles, without employing skin segmentation in the detection framework. 90

6.3 Identification results from moles with low saliency thresholds. . . 90

Bibliography

[BA85] P.J. Burt and E.H. Adelson. Merging images through pattern decomposition. In *Applications of Digital Image Processing VIII*, pages 173–181, 1985.

[BBVP03] Volker Blanz, Curzio Basso, Thomas Vetter, and Tomaso Poggio. Reanimating faces in images and video. In *EUROGRAPHICS 2003*, volume 22, pages 641–650, Granada, Spain, 2003. Blackwell.

[BFL06] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, November 2006.

[Bil97] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, 1997.

[BJ01] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *International Conference on Computer Vision, (ICCV)*, pages 105–112, 2001.

[BLS98] M. Bartlett, H. Lades, and T. Sejnowski. Independent component representations for face recognition. *Proceedings of the SPIE Symposium on Electonic Imaging: Science and Technology; Human Vision and Electronic Imaging III, volume 3299, San Jose, CA, January 1998. SPIE Press.*, 1998.

[BP93] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[BSVS04] V. Blanz, K. Scherbaum, T. Vetter, and H.P. Seidel. Exchanging faces in images. In *EUROGRAPHICS 2004*, volume 23 of *Computer Graphics Forum*, pages 669–676, Grenoble, France, 2004. Blackwell.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 187–194. Addison Wesley, 1999.

[BV03a] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

[BV03b] Volker Blanz and Thomas Vetter. Generating frontal views from single, non-frontal images. Face Recognition Vendor Test 2002, Technical Appendices NISTIR 6965, Nat. Inst. of Standards and Technology (NIST), 100 Bureau Drive, Stop 8940, Gaithersburg, MD 20899, March 2003.

[BVD00] A. Berman, P. Vlahos, and A. Dadourian. Comprehensive method for removing from an image the background surrounding a selected object. U.S. Patent 6,134,345, 2000., 2000.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[CBGM02] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1026–1038, 2002.

[CBS05] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 1124–1131, Washington, DC, USA, 2005. IEEE Computer Society.

[CCSS01] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proc. of IEEE CVPR 2001*, volume 2, pages 264–271, December 2001.

[CCTG95] D.H. Cooper, T.F. Cootes, C.J. Taylor, and J. Graham. Active Shape Models - Their training and application. *Computer Vision and Image Understanding*, (61):38–59, 1995.

[CET01] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:681–685, January 2001.

[Che95] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):790–799, 1995.

[CJ01] Chakra Chennubhotla and Allan D. Jepson. Sparse pca: Extracting multiscale structure from data. In *ICCV*, pages 641–647, 2001.

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[CM97] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 750, Washington, DC, USA, 1997. IEEE Computer Society.

[CM99] Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1197, Washington, DC, USA, 1999. IEEE Computer Society.

[CRZ96] I.J. Cox, S.B. Rao, and Y. Zhong. "ratio regions": A technique for image segmentation. *icpr*, 02:557, 1996.

[DBBB03] B. Draper, K. Baek, M. Bartlett, and J.R. Beveridge. Recognizing faces with pca and ica. *Comput. Vis. Image Underst.*, 91(1-2):115–137, 2003.

[DHS00] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley Interscience, 2nd edition, 2000.

[Die05] Reinhard Diestel. *Graph Theory*, volume 173. Springer-Verlag, Heidelberg, 3rd edition, July 2005.

[Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[DLR77] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[EL99] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, pages 1033–1038, Corfu, Greece, September 1999.

[Est05] F. J. Estrada. *Advances in Computational Image Segmentation and Perceptual Grouping*. PhD thesis, Department of Computer Science,University of Toronto, June 2005.

[FBCM04] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

[FF62] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.

[FH75] K. Fukunaga and L.D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, 21:32–40, 1975.

[FH98] P. F. Felzenszwalb and D. P. Huttenlocher. Image segmentation using local variation. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 98, Washington, DC, USA, 1998. IEEE Computer Society.

[FvDFH97] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics. Principles and Practice*. Addison-Wesley, 1997.

[FZ05] Daniel Freedman and Tao Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 755–762, Washington, DC, USA, 2005. IEEE Computer Society.

[GFL04] Leo Grady and Gareth Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *ECCV Workshops CVAMIA and MMBIA*, pages 230–245, 2004.

[GI04] Jack Goldfeather and Victoria Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, 23(1):45–63, 2004.

[Gra06] Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[GSAW05] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive alpha-matting. In J. J. Villanueva, editor, *Proceedings of the Fifth IASTED International Conference on Visualization, Imaging and Image Processing*, pages 423–429, Benidorm, Spain, Sept. 2005.

[GT88] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988.

[GW01] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[GWW01] Yoram Gdalyahu, Daphna Weinshall, and Michael Werman. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1053–1074, 2001.

[HLS02] Daniela Hall, Bastian Leibe, and Bernt Schiele. Saliency of interest points under scale changes. In *British Machine Vision Conference (BMVC'02)*, Cardiff, UK, September 2002.

[HN03] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16 (NIPS 2003), Vancouver, Canada*, 2003.

[HO00] Aapo Hyvärinen and Erkki Oja. Independent component analysis: A tutorial. *Neural Networks*, 13(4-5):411–430, 2000.

[HSP07] Bernd Heisele, Thomas Serre, and T. Poggio. A component-based framework for face detection and identification. *Int. J. Comput. Vision*, 74(2):167–181, 2007.

[IG98] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 125, Washington, DC, USA, 1998. IEEE Computer Society.

[JI99] Ian H. Jermyn and Hiroshi Ishikawa. Globally optimal regions and boundaries. *iccv*, 02:904, 1999.

[JP98] Michael J. Jones and Tomaso Poggio. Multidimensional morphable models. In *ICCV*, pages 683–688, 1998.

[JR02] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[KS96] David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, 1996.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[LB00] Aleš Leonardis and Horst Bischof. Robust recognition using eigenimages. *Comput. Vis. Image Underst.*, 78(1):99–118, 2000.

[LHZC01] S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *Proceedings of Computer Vision and Pattern Recognition (CVPR'01)*, volume 1, December 2001.

[Lin98] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.

[LK81] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981.

[LLW06] Anat Levin, Dani Lischinski, and Yair Weiss. A closed form solution to natural image matting. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 61–68, Washington, DC, USA, 2006. IEEE Computer Society.

[LM98] Thomas K. Leung and Jitendra Malik. Contour continuity in region based image segmentation. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume I*, pages 544–559, London, UK, 1998. Springer-Verlag.

[Low03] David G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[LRAL07] Anat Levin, Alex Rav-Acha, and Dani Lischinski. Spectral matting. In *CVPR '07: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[LSTS04] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 303–308, New York, NY, USA, 2004. ACM Press.

[LT06] Dahua Lin and Xiaoou Tang. Recognize high resolution faces: From macrocosm to microcosm. In *CVPR '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition*, pages 1355–1362, 2006.

[lTB01] Fernando De la Torre and Michael J. Black. Robust principal component analysis for computer vision. In *ICCV*, pages 362–369, 2001.

[Mar02] Aleix M. Martínez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):748–763, 2002.

[MB95] Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198, New York, NY, USA, 1995. ACM Press.

[MB98] A.M Martinez and R. Banavente. The AR face database. CVC Tech. Report 24, june 1998.

[MBLS01] Jitendra Malik, Serge Belongie, Thomas K. Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[MBSL99] Jitendra Malik, Serge Belongie, Jianbo Shi, and Thomas Leung. Textons, contours and regions: Cue integration in image segmentation. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 918, Washington, DC, USA, 1999. IEEE Computer Society.

[MFM04] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004.

[MS01] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pages 525–531, 2001.

[MSV95] Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[NA02] Mark S. Nixon and Alberto S. Aguado. *Feature extraction and image processing*. Newnes, Oxford, 2002.

[Neu66] John Von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA, 1966.

[OP03] Stanley Osher and Nikos Paragios. *Geometric Level Set Methods in Imaging,Vision,and Graphics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[OS88] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[OW06] Ido Omer and Michael Werman. The bottleneck geodesic: Computing pixel affinity. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1901–1907, Washington, DC, USA, 2006. IEEE Computer Society.

[PMS94] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, June 1994.

[PV07] Jean-Sébastien Pierrard and Thomas Vetter. Skin Detail Analysis for Face Recognition. In *CVPR '07: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[PWHR98] P. Jonathon Phillips, Harry Wechsler, Jeffrey S. Huang, and Patrick J. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[RKB04] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, August 2004.

[RM03] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proc. 9th Int'l. Conf. Computer Vision*, volume 1, pages 10–17, 2003.

[Rom05] Sami Romdhani. *Face image analysis using a multiple features fitting strategy*. PhD thesis, Department of Computer Science,University of Basel, 2005.

[RT00] Mark A. Ruzon and Carlo Tomasi. Alpha estimation in natural images. In *Proc. of IEEE CVPR 2000*, volume 01, page 1018, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[RV03] Sami Romdhani and Thomas Vetter. Efficient, robust and accurate fitting of a 3d morphable model. *iccv*, 01:59, 2003.

[SB91] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[SJTS04] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. *ACM Transactions on Graphics*, 23(3), 2004.

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Ste99] Charles V. Stewart. Robust parameter estimation in computer vision. *SIAM Rev.*, 41(3):513–537, 1999.

[TP91] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[Vek00] Olga Veksler. Image segmentation by nested cuts. In *CVPR '00: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 01, 2000.

[VSA03] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. *in Proc. Graphicon*, 2003.

[VV05] Vezhnevets V. and Konouchine V. "GrowCut"- Interactive Multi-Label N-D Image Segmentation By Cellular Automata. *Graphicon*, 2005.

[Wat93] Alan Watt. *3D Computer Graphics*. Addison-Wesley, 1993.

[WCT98] K.N. Walker, T.F. Cootes, and C.J. Taylor. Locating salient object features. In *British Machine Vision Conference (BMVC)*, volume 2, pages 557–566. BMVA Press, 1998.

[WFKvdM97] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face Recognition by Elastic Bunch Graph Matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[WL93] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.

[WS01] Song Wang and Jeffrey Mark Siskind. Image segmentation with minimum mean cut. *iccv*, 01:517, 2001.

[WS03] Song Wang and Jeffrey Mark Siskind. Image segmentation with ratio cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):675–690, 2003.