

Wavelet Frame Accelerated Reduced Vector Machine for Efficient Image Analysis

Inauguraldissertation

zur

Erlangung der Würde eines Doktors der Philosophie

vorgelegt der

Philosophisch-Naturwissenschaftlichen Fakultät

der Universität Basel

von

Matthias Rätsch

aus Potsdam, Deutschland

Basel, 2008

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Prof. Dr. Thomas Vetter, Universität Basel, Dissertationsleiter

Prof. Dr. Gerd Teschke, University of Applied Sciences Neubrandenburg,
Korreferent

Basel, den 24.06.2008

Prof. Dr. Hans-Peter Hauri, Dekan

Abstract

We propose a new approach for face and facial feature detection combined with the advantages of the Morphable Model.

The presented method reduces the runtime complexity of a Support Vector Machine classifier and the new training algorithm is fast and simple. This is achieved by an Over-Complete Wavelet Transform that finds optimally sparse approximations of the Support Set Vectors. The wavelet-based approach provides an upper bound on the distance between the decision function of the Support Vector Machine and our classifier. The obtained classifier is fast since the used Haar wavelet approximations of the Support Set Vectors allow efficient Integral Image-based kernel evaluations. This provides a set of double-cascaded classifiers of increasing accuracy for an early rejection. The algorithm yields an excellent runtime performance that is achieved by hierarchically discriminating with respect to the number and approximation accuracy of incorporated Reduced Set Vectors.

The proposed algorithm is applied to the problem of face and facial feature detection, but it can also be used for other image-based classifications. The algorithm presented, provides a 530-fold speed-up over the Support Vector Machine, enabling face detection at more than 25 fps on a standard PC.

Summarizing, we propose very fast and efficient to train classifiers that improve the detection performance by involving the advantages of the Morphable Model. On one hand to improve the fitting algorithm of the Morphable Model by automatic anchor point detection and on the other hand to use the Morphable Model for improving the training by synthetic data sets and to reduced the False Acceptance Rate.



Keywords

Over Complete Wavelet Transform, Haar-like Feature, Integral Images, Reduced Support Vector Machine, Coarse-to-Fine Classifier, Cascaded Evaluation, Real-Time Face Detection, Facial Feature Localisation, Tracking, 3D Morphable Model, Machine Learning, Computer Vision, HCI, CHIL

Contents

Contents	5
Notations and Abbreviations	9
Chapter 1 Introduction	13
Chapter 2 Wavelet Approximated Reduced Vector Machine	21
2.1. Support Vector Machine and Reduced Set Expansion	21
2.2. Outline of the W-RVM approach	23
2.3. Approximation of the W-RSV's	24
2.3.1. Integral Images for Efficient Kernel Evaluations	25
2.3.2. Haar-like Approximations of Reduced Set Vectors.....	27
2.3.3. Soft-Shrinkage to Build Rectangular Structured W-RSV's	32
2.3.4. Over-Complete Wavelet Transform.....	33
2.4. Hyper-plane Approximation	34
2.5. Hierarchical Evaluation via Resolution Levels	36
2.6. Algorithm to Generate Hierarchically Refined W-RSV's	37
2.7. Detection Process	41
2.8. Adjustment of Resolution Levels and Number of W-RSV's per Level	43
Chapter 3 Face and Facial Feature Detection based on the W-RVM	45
3.1. Data Sets for Training and Validation	45
3.1.1. Generation of Training Sets using the 3D MM.....	46
3.1.2. Generation of Synthetic Training Sets	48
3.1.3. Generation of Negative Training Sets.....	51
3.2. Applying the W-RVM for Face Detection	52
3.3. Applying W-RVM for Facial Feature Classifiers	56
3.3.1. Multi Criteria Evaluation of Optimal Facial Features.....	56

3.3.2.	Training of the W-RVM's for Facial Features	61
3.4.	Probabilistic W-RVM Classifier	67
3.4.1.	Variants of Non-parametric Techniques for PDF Estimation.....	67
3.4.2.	Probabilistic W-RVM using Sigmoid Fitting	70
3.5.	Cascaded Framework for Facial Feature Detection	72
3.5.1.	Single W-RVM Detector.....	72
3.5.2.	W-RVM Facial Feature Set Detector	73
3.6.	Evaluation of the Final Feature Assortment by PSM	76
Chapter 4	Applications	85
4.1.	Applications Demonstrating the W-RVM	86
4.1.1.	Application for W-RVM's – FD_FFpDetectApp.....	86
4.1.2.	Fast Face Detection – FaFaDe	86
4.1.3.	Fast Facial Feature Detection – FaFaFeDe	87
4.1.4.	Pose Estimation integrated to FaFaFeDe	89
4.1.5.	Fd_camFFDViewer.....	90
4.2.	I-Search project	90
4.3.	HCI, CHIL Applications using W-RVM	92
4.3.1.	Face and Facial Feature Point Tracking.....	92
4.3.2.	Avatar Following with Eye and Head Motion	95
4.3.3.	Switching Faces – a Perception Psychological Installation	96
Chapter 5	Perspective and Conclusion	99
5.1.	Further Unification of W-RVM and 3D MM	99
5.1.1.	Morphable Model.....	99
5.1.2.	Automatic Anchor Point Detection for the MM-Fitting	103
5.1.3.	Further Unification W-RVM and 3D MM.....	108
5.2.	Relevance of the W-RVM Hyper-plane Approximation	109
5.2.1.	Single-Stage W-RVM	109
5.2.2.	Multi-feature and Multi-invariant W-RVM	111
5.2.3.	Wavelet Approximated Vector Regression – W-RVR	112
5.2.4.	Tracking of Higher Feature Parameters	115
5.2.5.	W-RVM Real-time Learning	116

5.2.6. Adaptive and Invariant Kernel	118
5.2.7. Further Optimisation of the W-RVM.....	120
5.3. Conclusion	121
Appendix A UML Documentation	125
Component Diagrams	125
 Component Diagram <i>FFDTraining</i>	125
 Component Diagram <i>FFDWorking</i>	131
Appendix B Used Data Sets	143
Appendix C Trained W-RVM Classifiers	147
List of Figures	153
List of Tables	157
Curriculum Vitae	159
Bibliography	163

Notations and Abbreviations

a	Vectors are denoted by lowercase bold letters
a_i	i -element of the vector a
A	Matrices are denoted by uppercase bold letters
$\mathbf{A}_{M \times N}$	Matrix with M rows and N columns
\mathbf{A}'	Transpose of the matrix A
\mathbf{A}^+	Pseudo-inverse of matrix A
$\mathbf{A}_{i,j}$	(i, j) -element of the matrix A ; this is a scalar
$\mathbf{A}_{\cdot,j}$	Column vector formed by the j -th column of the matrix A
I	Identity matrix whose dimension depends on the context
$\mathbf{1}_{M \times N}$	$M \times N$ matrix for which all elements are equal to one
$\text{vec}(\mathbf{A})$	Vectorisation of the matrix A . If A is a $M \times N$ matrix, $\text{vec}(\mathbf{A})$ is a $MN \times 1$ column vector
$\text{sgn}(x)$	Signum function ($\text{sgn}(x) = -1$ if $x < 0$, 0 if $x = 0$, 1 if $x > 0$)
$\mathbf{a}^{(M)}$	If a is a $R \times 1$ column vector, $\mathbf{a}^{(M)}$ is a $M \times R/M$ matrix. It is assumed that R/M is an integer value
$\langle \mathbf{a}, \mathbf{b} \rangle$	Scalar product of vectors a and b
$\mathbf{a} \times \mathbf{b}$	Cross vector product of vectors a and b
Ψ_{SVM}	Support Vector Machine (SVM)
\mathbf{x}_i	Support Set Vector (SSV) with $i = 1, \dots, N_x$ from a SVM
Ψ_{RVM}	Reduced Support Vector Machine (RVM)
\mathbf{z}_i	Reduced Set Vector (RSV) with $i = 1, \dots, N_z$ from a RVM

Ψ_{W-RVM}	Wavelet Approximated Reduced Vector Machine (W-RVM).
\mathbf{u}_i^l	Wavelet Approximated Reduced Set Vector (W-RSV) with approximation level $l = 1, \dots, L$ and number of incorporated vectors $i = 1, \dots, N_z(l)$
Φ	Mapping function into the higher dimensional feature space, $\Phi: \mathcal{X} \rightarrow F$, $\mathbf{x} \mapsto \Phi(\mathbf{x})$
FRR	False Rejection Rate
FAR	False Acceptance Rate
DR	Detection Rate (DR=1-FRR)
R.O.C.	Receiver Operating Characteristic (ratio of DR and FAR)
S_μ	Soft-shrinkage operator with threshold parameter μ
W, W^{-1}	Wavelet transform and inverse wavelet transform operator
z_λ, u_λ	Corresponding wavelet coefficients $z_\lambda = \langle \mathbf{z}, \psi_\lambda \rangle$, $u_\lambda = \langle \mathbf{u}, \psi_\lambda \rangle$ of the RSV's and W-RSV's to the wavelet basis $\{\psi_\lambda\}_{\lambda \in \Lambda}$, where Λ is the index set over all possible locations, scalings and wavelet species
PDF	Probability density function
$p(x y)$	Likelihood of x with respect to y , e.g. $p(t_{fd} \mathbf{x}_{fd})$ probability density of the W-RVM output t_{fd} w.r.t. \mathbf{x}_{fd} (likelihood that the W-RVM classifier for faces produces the output t , given \mathbf{x} is an image position of a face).
RBF	Radial Basis Function kernel, $k(\mathbf{x}_j, \mathbf{x}_i) = \exp(-\ \mathbf{x}_j - \mathbf{x}_i\ ^2 / (2\sigma^2))$
PSM	Prior Shape Model
HCI	Human Computer Interaction
CHIL	Computers in the Human Interaction Loop
ROI	Region of interest
FOI	Field of interest
ML	Maximum likelihood
3D MM	Three-dimensional Morphable Model

Chapter 1

Introduction

Das wird nächstens schon besser gehen, Wenn Ihr lernt alles reduzieren. Und gehörig klassifizieren.

[Mephistopheles in Faust, First Part of the Tragedy, Johann Wolfgang Goethe]

General Background and Contribution of the Thesis

One of the main fields in image analysis is the extraction of meaningful information. This purpose can be achieved by different means, e.g. pattern recognition, morphological filtering, or deconvolution. In this thesis, the focus is on pattern recognition. In particular, we aim to locate and analyse faces or facial features in given images. In general, there exist two philosophies: One principle is to apply a complex face model to a given image and extract the desired information within the model space (top-down principle). Another way is to act directly in image space and to extract the information by correlation principles (bottom-up principle). Both philosophies clearly have advantages and disadvantages. For the purpose of face analysis, it turned out that the application of neither the first nor the second principle alone is optimally suited. Within this thesis, we suggest an improvement by combining the two approaches. In particular, we propose to overcome the disadvantage (model initialisation) of the top-down principle by a correlation/classification principle. The development of a very efficient classification principle is the main contribution of the thesis.

The 3D Morphable Model for Complex Face Modelling

Our purpose is to improve the 3D Morphable Model (MM, [101], [5], [6], [7]) by combining it with a image-based classifier. The Morphable Model is an example-based three-dimensional face model, derived by transforming the shape and texture of the examples into a vector space representation. New faces and expressions can be modelled by forming linear combinations of the prototypes. Shape and texture constraints derived from the statistics of our example faces are used to guide manual modelling or automated matching algorithms. In this framework, it is easy to control complex facial attributes, such as gender, attractiveness, body weight, or facial expressions. Attributes are automatically learned from a set of faces

rated by the user, and can then be applied to classify and manipulate new faces. Given a single photograph of a face, we can estimate its 3D shape, its orientation in space and the illumination conditions in the scene. Starting with manually labelled landmark points, our algorithm roughly estimates the size, orientation and illumination, and optimises the model parameters along with the face's internal shape and surface colour to find the best match to the input image. The face model extracted from the image can be rotated and manipulated in 3D. The MM can be used for an interactive modeller tool where a wide range of relevant attributes of faces can be controlled and for the synthesis of many representations of variations within the class of human faces. One of the main disadvantages of the MM is the manual initialisation.

Combining the 3D Morphable Model with an Efficient Classifier

The aim of this thesis is to invent a fast discriminating and efficient to train 2D image-based classifier. The classifier (bottom-up approach) shall be unified with the 3D MM (top-down approach) to improve the image-analysis performance of both approaches. On one hand, the 2D classifier will improve the fitting algorithm of the Morphable Model by automatic anchor-point detection and on the other hand, the Morphable Model will be used for improving the training and the detection rate of the image-based approach.

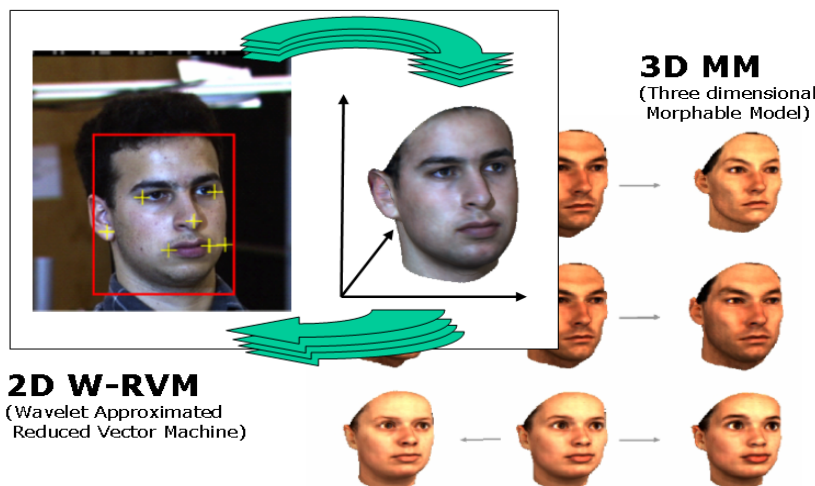


Figure 1-1: Images analysis by unifying a 2D image-based classifier and a 3D face model

The loops of unification of the 2D image-based classifier (W-RVM, left) and the 3D face model (Morphable Model, right) form the general background of this thesis. The main contribution is the invention of an adequate classifier for face and facial feature detection.

This unification can be realised by the iterative loop schematised in Figure 1-1: Firstly, the 3D MM is used to generate synthetic 2D data sets with ground truth in arbitrary size and diversity. Then, with classifiers trained on these synthetic data sets, the 2D locations for the anchor points can be detected. The local image representation of facial features – e.g. corner of the mouth or bridge of the nose – is ambiguous within the image. Hence, lists of possible

candidates are detected. To find the most likely final feature assortment among all combinatory possible candidate sets, the correlation between the feature points must be exploited (e.g. the true detection of the nose tip should be located between the centre of the eyes; all other candidates can be rejected). The 3D model is used for this correlation classifier on the second unification loop. Taking advantage of the 3D MM, the False Acceptance Rate (FAR) of the classifier can be reduced by rejecting false candidates. The remaining 2D feature assortment can now be used as anchor points to initialise the first fitting stage of the Morphable Model. At the next loop of unification, the first rough 3D shape estimation of the Morphable Model can be used to enrich the 2D feature point extraction. The estimated pose can help to eliminate those feature points that are not visible within the 2D view of the face to be analysed. For the so obtained set of feature points, the corresponding search area is typically of smaller dimensions. Therefore, more-complex 2D classifiers can be applied without any extra computational cost.

We will show some applications taking advantage of the further unification of the 3D Morphable Model and the 2D classifier, e.g. pose estimation, tracking of facial feature points, or HCI applications like head- and eye-tracking by an avatar or exchanging faces of subjects in video streams. As future work, we plan to take advantage of further unification loops. For instance, automatic labelling of the landmarks for more-refined fitting stages of the Morphable Model, reducing the number of the training stages of the W-RVM classifier, and adjustment of the whole approach for Support Vector Regression issues.

Summary of the Motivation

The loops of unification of the 2D image-based classifier and the 3D Morphable Model form the general background of this thesis. The essential ingredient, before exposing the unification loops, is the invention of an adequate classifier for face and facial feature detection. This will be the main contribution of this thesis. Additionally, we present the first attempt to integrate the developed classifier within further loops of unification. This is illustrated by a number of possible applications in the field of face analysis and synthesis.

Efficient 2D Image-based Classifier and Hyperspace Approximation

Now we want to introduce how to develop the fast-discriminating and efficient to train 2D image-based classifier for the unification loops with the 3D MM.

Image-based classification tasks are time consuming. For instance, detecting a specific object in an image, such as a face or a facial feature point like the nose tip, is computationally expensive, as all the pixels of the image are potential object centres. Hence, all the pixels must be classified. Therefore, numerical accelerations are required. We propose to use a

wavelet frame accelerated Reduced Support Vector Machine for a sparse hyper-plane representation.

Relation to Recent Research

Recently, methods that are more efficient have emerged based on a cascaded evaluation of hierarchical filters: image patches that are easy to discriminate are classified by a simple and fast filter, while patches that resemble the object of interest are classified by more-involved and slower filters. In the area of face detection [79], cascaded-based classification algorithms were introduced by Keren et al. [49], by Romdhani et al. [73], and by Viola and Jones [102]. To apply the detector, proposed by Keren et al. [49], the negative examples need to be Boltzmann distributed and smooth. This strong assumption often leads in the presence of a cluttered background to an increased FAR. Romdhani et al. [73] use a Cascaded Reduced Set Vectors (RSV) expansion of a Support Vector Machine [100]. The bottleneck of [73] is that at least one convolution of a 20×20 filter has to be carried out on the full image, resulting in a computationally expensive evaluation of the kernel with an image patch. Kienzle et al. [50] present an improvement of this method, where the first (and only the first) RSV is approximated by a separable filter. Viola & Jones [102] use Haar-like oriented edge filters with a block-like structure, enabling a very fast evaluation by the use of an Integral Image. These filters are weak, in the sense that their discrimination capability is poor. Among this finite set of filters, an AdaBoost algorithm is applied to choose the one with optimal discrimination power. A drawback of their approach is that it is not clear that the cascade achieves optimal generalisation performances. Practically, the training proceeds by trial and error, and often, the number of filters per stage must be manually selected, so that the FAR decreases smoothly. Another drawback of the method is that the set of available filters is limited and has to be selected manually. The training for the classifier is of "the order of weeks" ([102], Section 5.2), as every filter (about 10^5) is evaluated on the whole set of training examples and this is done every time a filter is added to a stage of the cascade.

Key Ideas of Wavelet Frame Accelerated Reduced Support Vector Machines

Considering the above-mentioned problems, we developed a novel classification algorithm. The optimal approximation of the hyper-plane for an efficient classifier is the central point of interest of this thesis.

The following features make the algorithm accurate and efficient:

- 1) Support Vector Machine: Use of an SVM classifier [100] that is known to have optimal generalisation capabilities in a wide range of tasks [86], [64], including object detection, recognition and face detection [41], [58], [59].

- 2) Reduced Support Vector Machine: The RVM uses a reduced set of Support Vectors [87], [73].
- 3) Double Cascade: For non-symmetric data (i.e. only a few positives to many negatives) we achieve an early rejection of easy to discriminate vectors. It is obtained by the two following cascaded evaluations over coarse-to-fine Wavelet Approximated Reduced Set Vectors (W-RSV's):
 - a) Cascade over the number of used W-RSV's
 - b) Cascade over the resolution levels of each W-RSV
- 4) The Double Cascade constitutes one of the major novelties of our approach. The trade-off between accuracy and speed is essentially reduced.
- 5) Integral Images: As the RSV's are approximated by a Haar wavelet transform, the Integral Image method (Section 2.3.1) is used for the evaluation of the decision function, similarly to [102].
- 6) Wavelet Frame: We use an over-complete wavelet system to find the best representation of the RSV's.

The learning stage of our proposed Wavelet Approximated Reduced Vector Machine (W-RVM) is fast, straightforward, automatic, and does not require the manual selection of ad-hoc parameters. For example, the training time (Section 2.6) is two hours, which is a vast improvement over former detectors.

The paradigm of our classification method is that, instead starting by a poor classifier and getting more complex by heuristic knowledge, we first build a classifier that is proven to have optimal generalisation capabilities. The focus then becomes runtime efficiency while maintaining the classifier's optimal accuracy. To avoid complex search over the parameter space, we do not start with the full parameter space, but with the proved optimal performance of an SVM. Then we reduce the complexity by a Reduced Vector Set and the Over-complete Wavelet Approximation. Hence, our approach is straightforward.

In order to obtain a sparse block-like structure of the image patch we apply in our approach an Over-Complete Wavelet Transform (OCWT) to the Reduced Support Vector Machine, and do not transform the input space as a pre-processing like [48], [35].

This thesis presents the coherent and complete framework of our approach (still published in [70], [68], [71]). The improvement of [68] compared to [70] are the features 3. a) and 5. (see above): The Simulated Annealing optimisation using morphological filters was replaced by a sparse wavelet frame representation of the RSV's. Simulated Annealing does not provide the

global optimum of the RVM approximation in all cases and it is difficult to adjust the resolution level.

In this thesis, we take advantage of recent progress in wavelet analysis. In particular, we apply a soft-shrinkage threshold operation in order to obtain an optimal sparse signal approximation (rectangular structure) in wavelet space. Applying the proposed recursive refinement of the wavelet frame representation of the RSV's, we obtain the double-cascade structure of the learning and detection process.

Relevance to Research and Praxis

The developed classifier is applied to the problem of face and facial feature detection. We compare the number of operations needed to approximate the decision hyper-plane or needed to reject image areas without objects of interest. Therewith we show the improvement by a drastic decrease of needed operations from a theoretical and practical point of view. In our experiments we validate our face detector on the well-known FERET database [61], so that our results can be compared by other researchers. In addition, we show the improvement in detection and training compared to state-of-the-art detectors, like [102], [79], and [85].

We published the new approach [70], [68], [71] and worked on this thesis to verify the relevance of the methods in theory and experiments. Alongside this we proved the efficiency, accuracy and robustness of the classifier already in real-life environments. For example we developed webcam applications for live presentations; plug-ins, e.g. for Adobe [33]; and common API interfaces for several HCI and CHIL projects. Ideas of our approach are adapted and the API's integrated for practical use by firms and institutes, like the Cognitec GmbH, the partners of the joint-project I-Search [54], the Konrad-Zuse-Institute Berlin (ZIB), the University of Applied Sciences Neubrandenburg [99], the University of Applied Sciences Northwestern Switzerland [13], the Academy of Art and Design Basel [57], and others.

The applicableness of the invented approach is also verified within bachelor, diploma, and master theses using the API's of the detectors or the classifiers. Additionally we show in this thesis that our invented approach is not only usable for detection, but also adaptable to other fields of research and applications, like condensation tracking or other function approximations, like the approximation of regression functions.

Dissertation Structure

The thesis is organised as follows: After motivating in this chapter the general background and the main contribution of this thesis we introduce in Chapter 2 the developed 2D classifier. Most of the theoretical work of the proposed thesis is concentrated in this chapter. We show

in Section 2.4 that the wavelet frame approach provides an upper bound of the hyper-plane approximation error. Exploring this characteristic, the training of the W-RVM works without heuristics and is fast. We also detail in Section 2.4 the relation between the hyper-plane approximation error of the decision functions and a training parameter to control the trade-off between sparsity and approximation. As demonstrated in Section 2.8 the parameter for setting the approximation accuracy does not play a decisive role, as opposed to former methods, using only one resolution level. Finally, we summarise the training (Section 2.6) and detection algorithm (Section 2.7) of our novel approach. In Chapter 3, we apply the novel classifier for face and facial feature detection and take first advantage of the unification of the 3D MM and the W-RVM. It is shown in Chapter 3 that the new expansion yields a comparable accuracy to the SVM while providing a significant speed-up. In addition to the first publications of the approach, we carried out experiments on well-known databases, like FERET [61] in order to provide comparability to other approaches.

In Chapter 4, we demonstrate the practical relevance of the W-RVM approach within several applications using the new face and facial feature detection method. In Chapter 5, after introducing the Morphable Model (Section 5.1.1), we show the integration of the W-RVM into the unification with the 3D MM and we give a guideline on how to continue the unification of the Morphable Model and the W-RVM by further iterative loops. Also in Chapter 5, we show how to take advantage of the W-RVM hyper-plane approximation for other fields of function approximations like Support Vector Regression and show opportunities to improve the approach, e.g. a further simplification of the training by a single-stage approximation. This proves the theoretical relevance and opportunities of the research results propose in this thesis.

Chapter 2

Wavelet Approximated Reduced Vector Machine

2.1. Support Vector Machine and Reduced Set Expansion

Before we introduce the new approach we briefly recall Support Vector Machines (SVM) [100] used as classifier, and outline the usage of an approximation of SVM's called Reduced Support Vector Machines (RVM) [87]. RVM provide a hierarchy of classifiers of increasing complexity. Their use for fast face detection is demonstrated in [73] and [75].

Suppose that we have a labelled training set consisting of a series of 20×20 image patches $\mathbf{x}_i \in \mathcal{X}$ (arranged in a 400 dimensional vector) along with their class label $y_i \in \{\pm 1\}$. Support Vector classifier implicitly map the data \mathbf{x}_i into a dot product space F via a (usually nonlinear) map $\Phi: \mathcal{X} \rightarrow F$, $\mathbf{x} \mapsto \Phi(\mathbf{x})$. Often, F is referred to as the feature space. Although F can be high dimensional, it is usually not necessary to explicitly work in that space [10]. By Mercer's theorem, it is shown that it exists a class of kernels $k(\mathbf{x}, \mathbf{x}')$ to compute the dot products in associated feature spaces, i.e. $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$. The training of an SVM provides a classifier with the largest margin [100], i.e. with the best generalisation performances for given training data and a given kernel. Thus, the classification of an image patch \mathbf{x} by an SVM classification function, with N_x support vectors \mathbf{x}_i with non-vanishing coefficients α_i and with a threshold b , is expressed as follows:

$$y(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^{N_x} \alpha_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b \right) \quad (2.1)$$

For our purpose, the common Gaussian Radial Basis Function Kernel (RBF) is used:

$$k(\mathbf{x}, \mathbf{x}_i) = \exp \left(- \frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{(2\sigma^2)} \right) \quad (2.2)$$

We performed experiments with linear, polynomial and RBF kernels and it turned out that RBF performed best for our specific classification problem. We also focus in this paper on Gaussian kernel, because we can show in Section 2.4 in an analytically way the necessary approximation bounds. The advantage of polynomial kernels is that the Reduced Set Vectors can be derived explicitly, even for non homogenous kernels [9], [96]. However, for good performance with polynomial kernels a feature-space normalisation is necessary. The focus of the thesis is the space representation of the decision hyper-plane not a feature-space transformation of the data.

The Support Set Vectors (SSV) form a subset of the training vectors. The classification of one patch by an SVM is slow because there are many support vectors. The SVM can be approximated by a Reduced Set Vector (RVM) expansion [87], [88]. We denote by $\Psi_{SVM} \in F$, the vector normal to the separating hyper-plane of the SVM, and by $\Psi_{RVM} \in F$ the vector normal to the RVM with N_z vectors: $\Psi_{SVM} = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$, $\Psi_{RVM} = \sum_{i=1}^{N_z} \beta_i \Phi(\mathbf{z}_i)$, with $N_z \ll N_x$. The \mathbf{z}_i are the Reduced Set Vectors and are found by minimising $\|\Psi_{SVM} - \Psi_{RVM}\|^2$ with respect to \mathbf{z}_i and to β_i [87]. They have the particularity that they can take any values; they are not limited to be one of the training vectors, like for the support vectors. Hence, much less Reduced Set Vectors might be enough to approximate the SVM. For instance, an SVM with more than 8000 Support Vectors can be accurately approximated by an RVM with 100 Reduced Set Vectors.

The second advantage of RVM is that they provide a hierarchy of classifiers. It was shown in [73] that the first Reduced Set Vector is the one that discriminates the data at most; and the second Reduced Set Vector is the one that discriminates most of the data that were misclassified by the first Reduced Set Vector, etc. Figure 2-1 demonstrates the effects on the classification boundary of sequential reduced set vector evaluation.

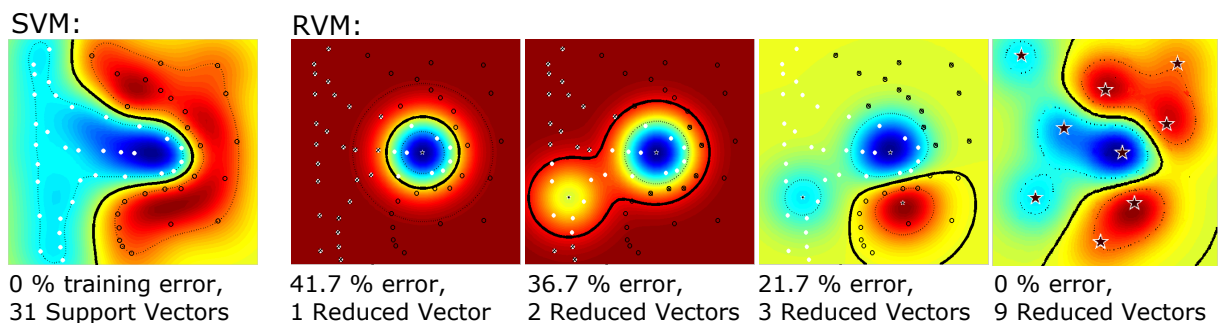


Figure 2-1: Toy example demonstrating the sequential RVM

The result of the sequential application of RSV's (stars) to a 2D classification problem, showing (left to right) the original SVM and the result of using 1, 2, 3 and 9 RSV's. Darker regions indicate strong support for the classification. With only 9 RSV's instead of 31 SSV's the RVM gains the same error rate as the SVM. Using only the first RSV's yields high error rates, but data points (with a large negative distance to the classification boundary) can be early rejected as negative points, without further evaluation cost.

This hierarchy of classifiers is obtained by first found β_1 and \mathbf{z}_1 that minimises $\|\Psi_{SVM} - \beta_1 \Phi(\mathbf{z}_1)\|^2$. Then the Reduced Set Vector \mathbf{z}_k is obtained by minimising the norm $\|\Psi_k - \beta_k \Phi(\mathbf{z}_k)\|^2$, where $\Psi_k = \Psi_{SVM} - \sum_{i=1}^{k-1} \beta_i \Phi(\mathbf{z}_i)$. The optimal $\beta_{k,i}, i = 1, \dots, k$ are jointly computed [87].

Romdhani et al. used in [73] a cascaded evaluation based on an early-rejection principle, to that the number of Reduced Set Vectors necessary to classify a patch is, on average, much less than the number of Reduced Set Vectors, N_z . Therefore, the classification of a patch \mathbf{x} by an RVM with j Reduced Set Vector is:

$$y_j(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^j \beta_{j,i} k(\mathbf{x}, \mathbf{z}_i) + b_j \right) \quad (2.3)$$

2.2. Outline of the W-RVM approach

Before we introduce in the following sections the novelties of the developed efficient classification algorithm, we give an outline of the core ideas of the new approach:

Support Vector Machine: Use of an SVM classifier that is known to have optimal generalisation capabilities.

SVM: $\Psi_{SVM} = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$, \mathbf{x}_i are the support vectors

Decision function: $y(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^{N_x} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b \right)$ with kernel function $k(\cdot, \cdot)$, e.g.

Gaussian kernel $k(\mathbf{x}, \mathbf{z}_i) = \exp(-\|\mathbf{x} - \mathbf{z}_i\|^2 / (2\sigma^2))$

Reduced Support Vector Machine: The RVM uses a reduced set of Support Vectors ($N_z \ll N_x$).

RVM: $\Psi_{RVM} = \sum_{i=1}^{N_z} \beta_i \Phi(\mathbf{z}_i)$, \mathbf{z}_i are the Reduced Set Vectors (RSV's)

Decision function: $y(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^{N_z} \beta_i k(\mathbf{x}, \mathbf{z}_i) + b_i \right)$

Double Cascade: For non-symmetric data (i.e. only few positives to many negatives) we achieve an early rejection of easy to discriminate vectors. It is obtained by the two following cascaded evaluations over coarse-to-fine Wavelet Approximated Reduced Set Vectors (W-RSV's):

Cascade over the number of used W-RSV's and

Cascade over the resolution levels of each W-RSV.

The Double Cascade constitutes one of the major novelties of the W-RVM approach. The trade-off between accuracy and speed is very continuous.

Integral Images: As the RSV's are approximated by a Haar wavelet transform, the Integral Image method is used for their evaluation.

Wavelet Frame: We use an over-complete wavelet system to find the best representation of the RSV's. The learning stage of our proposed Wavelet Approximated Reduced Vector Machine (**W-RVM**) is fast, straightforward, automatic and does not require the manual selection of ad-hoc parameters. For example, the training time is two hours, which is a vast improvement over former detectors. The Over-Complete Wavelet Transform (OCWT) is applied at the W-RVM training. That is opposite to several other approaches using a wavelet input space transformation as a pre-processing at detection time.

2.3. Approximation of the W-RSV's

We first build an SVM classifier that is proven to have optimal generalisation capabilities. However, using an SVM for detecting a specific object in an image, such as a facial feature point like the nose tip or a face, is computationally expensive. Using a brute-force approach as seen in Figure 2-2 all pixels of the image are potential object centres. Therefore, numerical accelerations are required, while maintaining the classifier's high accuracy.

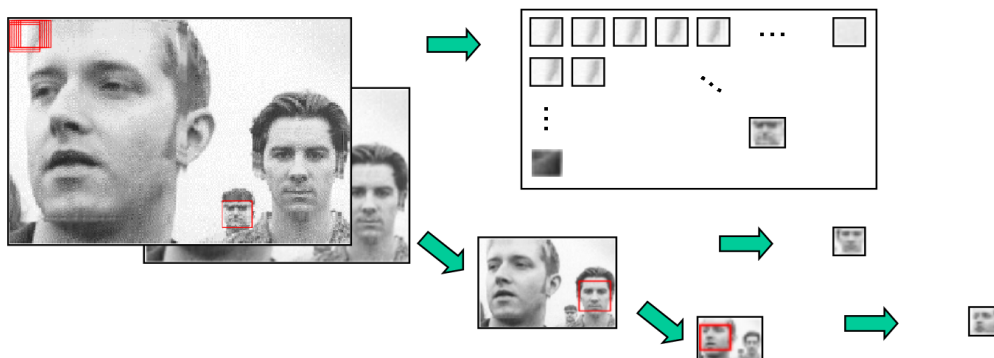


Figure 2-2: Object detection as an example for complex classification tasks

Detection using a brute-force approach is computationally expensive. Utilising a sliding observation window a patch is cut out and classified at each column and each row of the image. To detect larger objects an image pyramid is used by down-sampling the image several times. For a VGA image, about 10^6 patches must be classified, if the classification for one patch takes only 1ms the detection takes several minutes for the full image.

In order to improve the runtime performance, we approximate the SVM by a Reduced SVM (RVM) in combination with a cascaded evaluation as proposed in [73], [102]. The RVM aims to approximate the SVM by a smaller set of new Reduced Set Vectors (RSV's), instead of the Support Vectors (see Section 2.1). The RVM approach provides a significant speed-up over the SVM, but is still not fast enough, as the image has to be convolved in steps of full

convolutions, e.g. by 20×20 RSV's. The algorithm presented in this thesis improves this method since it does not require this convolution to be performed explicitly. Instead, it approximates the RSV's by Haar-like vectors and computes the evaluation of a patch using an Integral Image of the input image. They can be used to compute very efficiently the dot (or inner) product of an image patch with an image that has a block-like structure, i.e. rectangles of constant values.

2.3.1. Integral Images for Efficient Kernel Evaluations

Definition of Integral Images

We use Summed Area Tables [19], a.k.a. Integral Images [102] to reduce the computational cost of the convolution of a patch with one Reduced Set Vector. The value of the Integral Image, ii , at point (x, y) (Figure 2-3) is the sum of all the pixels, in the input image i , above and to the left of (x, y) :

$$ii(x, y) = \sum_{a \leq x, b \leq y} i(a, b) \quad (2.4)$$

The advantage of Integral Images is that they can compute the sum of the pixel's values of the input image in a rectangle, in constant time, by only four additions (see Figure 2-3):

$$\sum_{\substack{x_1 < a \leq x_4, \\ y_1 < b \leq y_4}} i(a, b) = ii(x_4, y_4) - ii(x_2, y_2) - ii(x_3, y_3) + ii(x_1, y_1) \quad (2.5)$$

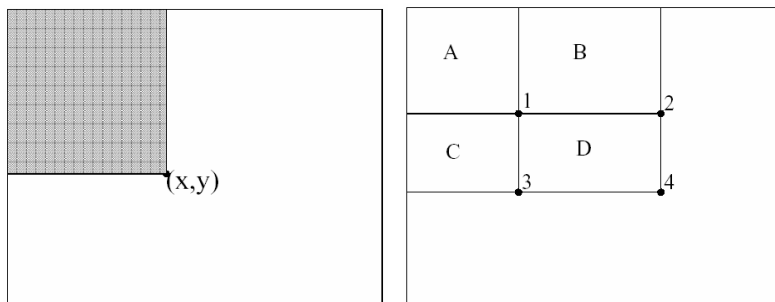


Figure 2-3: Definition and advantage of Integral Images

The value of the Integral Image (*left*), $ii(x, y)$, at a point is the sum of all the pixels above and to the left. The sum of the pixel's values of the input image in a rectangle, is computed in constant time, by only four additions (*right*, $D = ii(4) - ii(2) - ii(3) + ii(1)$).

Additionally, an integral image can be computed in one pass over the original image, using the following recursive formulae:

$$\begin{aligned} s(x, y) &= s(x, y-1) + i(x, y) \\ ii(x, y) &= ii(x-1, y) + s(x, y), \end{aligned} \quad (2.6)$$

where $s(x, y)$ is the cumulative sum.

Using Integral Images for Efficient Kernel Evaluations

During an RVM evaluation, most of the time is spent for kernel evaluations. In the case of the Gaussian kernel, $k(\mathbf{x}, \mathbf{z}_i) = \exp(-\|\mathbf{x} - \mathbf{z}_i\|^2 / (2\sigma^2))$ (chosen here) the computational cost is spent in evaluating the norm of the difference between a patch and a RSV. This norm can be expanded as follows:

$$\|\mathbf{x} - \mathbf{z}_k\|^2 = \mathbf{x}'\mathbf{x} - 2\mathbf{x}'\mathbf{z}_k + \mathbf{z}_k'\mathbf{z}_k \quad (2.7)$$

As \mathbf{z}_i is independent of the input image, it can be pre-computed. The sum of squares of the pixels of a patch of the input image, $\mathbf{x}'\mathbf{x}$ is efficiently computed using the Integral Image ([19], [102]) of the squared pixel values of the input image. As a result, the computational load of this expression is determined by the term $2\mathbf{x}'\mathbf{z}_i$.

The novelty of our approach is to approximate \mathbf{z}_i in $2\mathbf{x}'\mathbf{z}_i$ (which is the classical scalar product) so that it can be computed very fast with the integral image approach as well. We approximate the RSV's, \mathbf{z}_i , by optimally wavelet frame approximated Reduced Set Vectors (W-RSV's), \mathbf{u}_i which have a block-like structure. Optimally approximated means here, the usage of an optimally shifted wavelet basis, which represents the image as sparse as possible. Then the term $2\mathbf{x}'\mathbf{u}_i$ can be evaluated very efficiently using the Integral Image. The term can be re-sorted by

$$2\mathbf{x}'\mathbf{u}_i = 2\sum_{k=1}^D x_k u_{i,k} = 2\sum_{r=1}^{R_i} v_{i,r} \sum_{j=1}^{D_r} x_j \quad (2.8)$$

where D is the dimension of the vectors (e.g. 400 pixels by a patch size 20×20 as in Figure 2-4), R_i is the number of rectangles of \mathbf{u}_i , $v_{i,r}$ the grey values of the rectangle r and $x_j, j = 1, \dots, D_r$ all pixel-values of \mathbf{x} within the r -th rectangle. Because $\sum_{j=1}^{D_r} x_j$ can be computed by adding three pixels of the Integral Image of the input image [19], the scalar product is evaluated in constant time by four additions per rectangle and one multiplication per grey value.

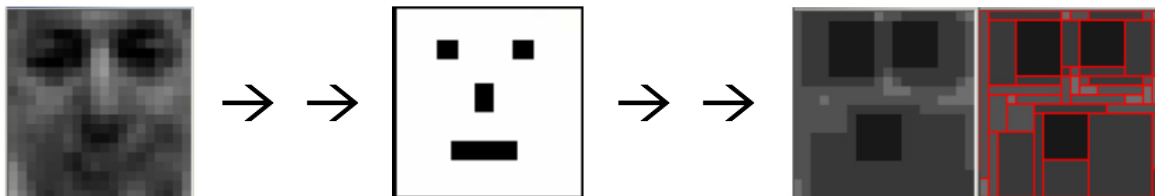


Figure 2-4: Use of Integral Images for fast kernel evaluation

If we approximated a RSV (*left*) by rectangles of constant values (*middle*: value 1 for black and 0 for white areas) only the sums over the black rectangles have to be considered. All other terms (grey value 0) in (2.8) become to zero. Naturally more values between 0 and 1 are necessary (as seen *right*) for a suitable approximation. By re-sorting the terms by their grey values, only one multiplication per grey value and four additions per rectangle have to be evaluated.

2.3.2. Haar-like Approximations of Reduced Set Vectors

To obtain a Haar-like structure of the Reduced Set Vectors we introduce and compare three methods. The morphological filter and wavelet frame approximation turned out to achieve the best results, concerning sparsity. The advantage of the morphological filter is that this method is simple to implement, but the wavelet frame approximation is more straightforward and faster than the Simulated Annealing optimisation. Thus, we use mostly this approximation technique at the further work.

Polynomial Approximation

We approximate \mathbf{z} by $\mathbf{p} = \mathbf{M}\boldsymbol{\rho}$, with $\boldsymbol{\rho} = (\rho_1, \dots, \rho_h)'$, $\mathbf{M} = (\mathbf{m}_1, \dots, \mathbf{m}_h)$, $\mathbf{m}_1=1$, $\mathbf{m}_2=x$, $\mathbf{m}_3=y$, $\mathbf{m}_4=x^2$, $\mathbf{m}_5=xy$, $\mathbf{m}_6=y^2$, $h=6$ by order 2.

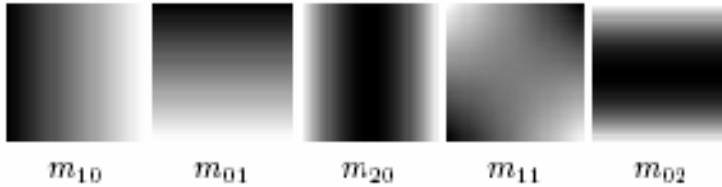


Figure 2-5: Templates corresponding to algebraic moments

It is shown in [89] Equation (8) that $\boldsymbol{\rho} = \mathbf{B}^{-1}\mathbf{U}_z$, with $\mathbf{B} = \mathbf{M}'\mathbf{M}$ and $\mathbf{U}_z = \mathbf{M}'\mathbf{z}$ to minimise $\|\mathbf{p} - \mathbf{z}\|^2$ and so \mathbf{z} is approximated by $\mathbf{p} = \mathbf{M}\mathbf{B}^{-1}\mathbf{U}_z$.

For the term $\mathbf{x}'\mathbf{z}_k$ in (2.7) we obtain $\mathbf{x}'\mathbf{p}_k = \mathbf{U}_x\mathbf{B}^{-1}\mathbf{U}_z$, because $\mathbf{U}_x = \mathbf{x}'\mathbf{M}$. \mathbf{U} denotes a vector of centralised moments μ_{pq} which can be represented by algebraic moments m_{pq} ([89] eq. (5) and Figure 2-5). The algebraic moments, and hence the centralised moments, can be computed by Integral Images. We thus arrive that we can compute the term $\mathbf{x}'\mathbf{p}_k$ with Integral Images by the polynomial approximation of \mathbf{z} with \mathbf{p} .

In Figure 2-6 the polynomial approximations (middle column) for two Reduced Set Vectors (left column) are shown. As it can be seen on the difference vectors (right column) the low-frequency information of the vectors are well represented, but the high-frequency parts, like the eyes or the mouth cannot be represented by the moments.

For the optimisation, we used Monte Carlo methods and the Simulated Annealing Method (ASA), to handle the high number of extremes (see Figure 2-7 top row). The distance of the approximated decision hyper-plane to the original SVM describes by incorporating more approximated vectors.

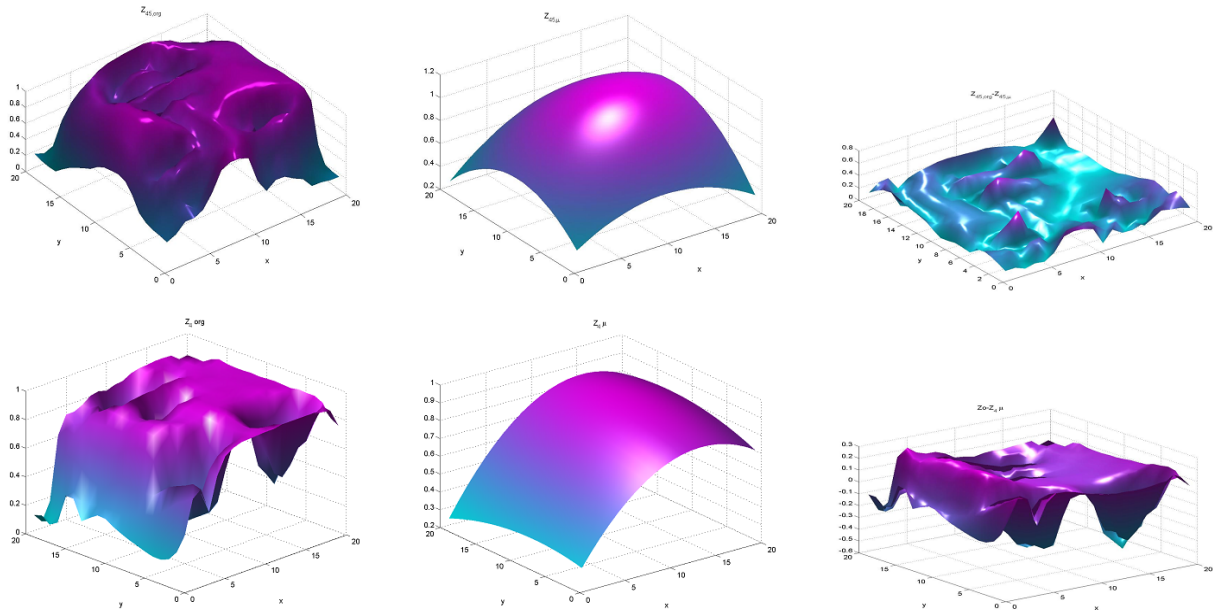


Figure 2-6: Examples for polynomial approximation of Reduced Set Vectors
Left: Reduced Set Vectors, *middle:* polynomial approximations; *right:* differences.

However, the discrepancy becomes early asymptotical (greater than 90% of the initial distance) compared with the RVM (see Figure 2-7 bottom right). It would be interesting to experiment with a piecewise polynomial approximation and to compare the efficiency with the other Haar-like approximations.

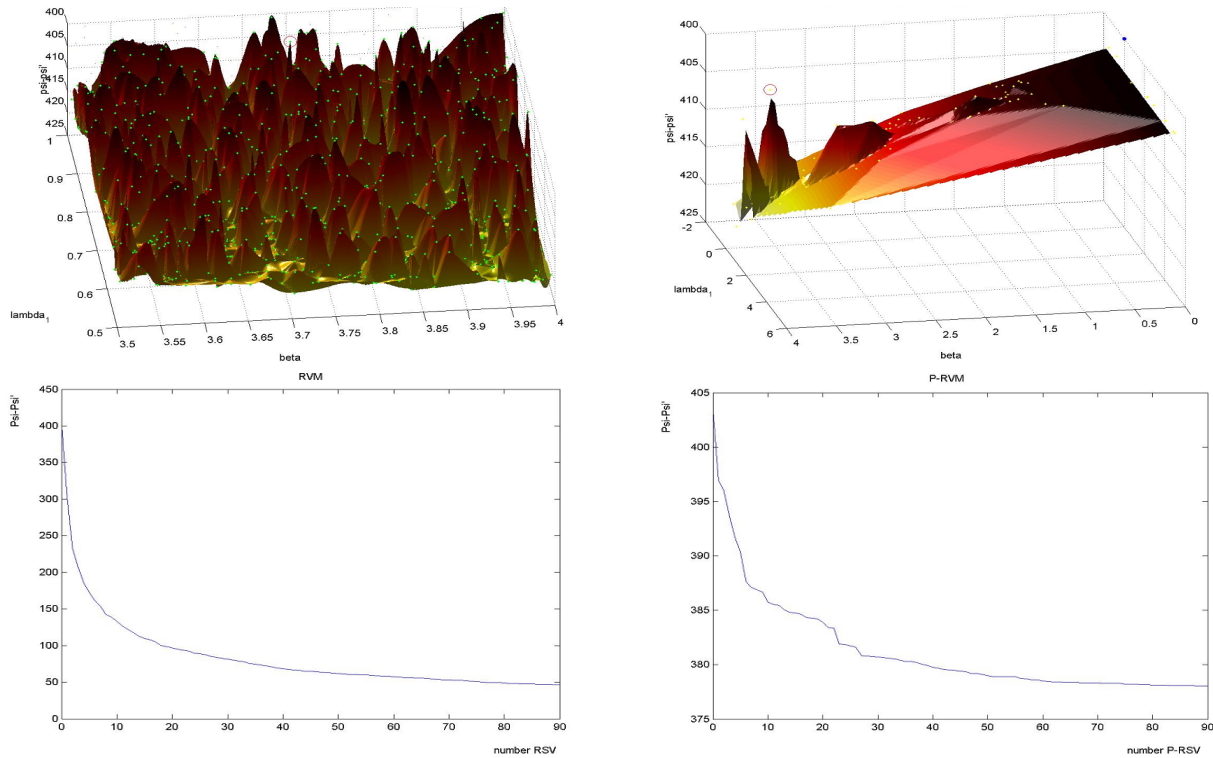


Figure 2-7: Polynomial approximation of the RVM
Top: Optimisation of the hyper-plane distance with many local minima (blue circle is the global extreme); *bottom:* Hyper-plane distance of the RVM (*left*) and the polynomial approximation (*right*).

Approximation using Morphological Filters

We want to introduce a second approach to obtain the Haar-like structure. A posterisation is used as quantisation and morphological filter to achieve the block-like structure.

The block-like Reduced Set Vectors must (i) be a good approximation of the SVM by minimising $\|\Psi_{RVM} - \Psi_{H-RVM}\|$, and (ii) have a few rectangles with constant value to provide a fast evaluation. Hence, to obtain the k -th Reduced Set Vector instead of minimising $\|\Psi_k - \beta_k \Phi(\mathbf{z}_k)\|^2$, where $\Psi_k = \Psi_{SVM} - \sum_{i=1}^{k-1} \beta_i \Phi(\mathbf{z}_i)$ as detailed in Section 2.1 and [73], we minimise the following energy with respect to β_k and to \mathbf{z}_k :

$$E_k = \|\Psi_k - \beta_k \Phi(\mathbf{z}_k)\|^2 + w(4\#[\mathbf{z}_k] + \nu), \quad (2.9)$$

where $\#[\mathbf{z}_k]$ is the number of piecewise constant rectangles, ν the number of grey values of \mathbf{z}_k and w is a weight that trades off the accuracy of the approximation with the runtime efficiency of the evaluation of \mathbf{z}_k with an input patch. To minimise the energy E_k , we use Simulated Annealing, which is a global optimisation method. The starting value of this optimisation is the result of the minimisation of $\|\Psi_k - \beta_k \Phi(\mathbf{z}_k)\|^2$, i.e. the Reduced Vector as computed in [73]. To obtain a block-like structure the following two operations are performed, as shown in Figure 2-9:

1st Quantisation: The grey values of \mathbf{z}_k are quantised into ν bins, applying a posterisation. The threshold values of this quantisation are the $1/\nu$ percentiles of the grey values of \mathbf{z}_k . For instance if $\nu = 2$, then \mathbf{z}_k will be approximated by 2 grey levels, and the 50% percentile is used as a threshold: the pixels of \mathbf{z}_k for which the grey values are lower than the threshold are set to the mean of these pixels. The result of this quantisation on two Reduced Set Vectors is shown in the second column of Figure 2-9.

2nd Block structure generation: The quantisation reduces the number of grey level values used to approximate a Reduced Set Vector \mathbf{z}_k , but it does not produce a block structure. To obtain a block structure two types of morphological operations [2] seen in Figure 2-8 are used (third column of Figure 2-9): opening (a dilatation followed by an erosion) or closing (an erosion followed by a dilatation) similar to [67], [66]. The parameter which must be opti-

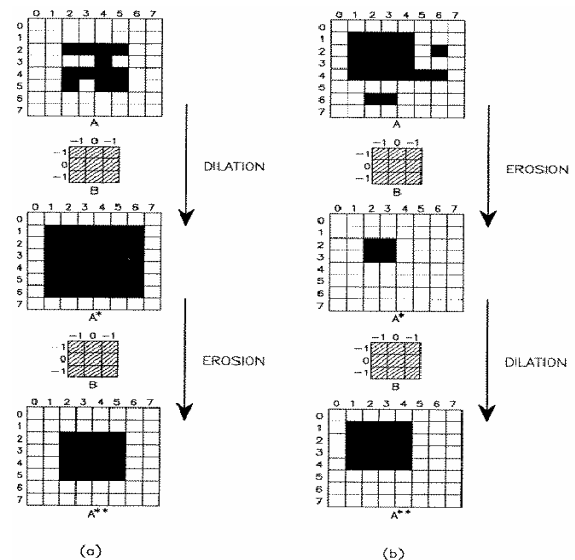


Figure 2-8: Morphological operations
a) Closing, b) Opening (B is the structuring elements).

mised are the type of morphological operations applied is denoted by $A = \{\text{opening, closing}\}$, and the size of the structuring elements is denoted by B . The coordinates of the rectangles are obtained by looking for the maximum width and height of disjoint rectangular areas at the same grey level (fourth column).

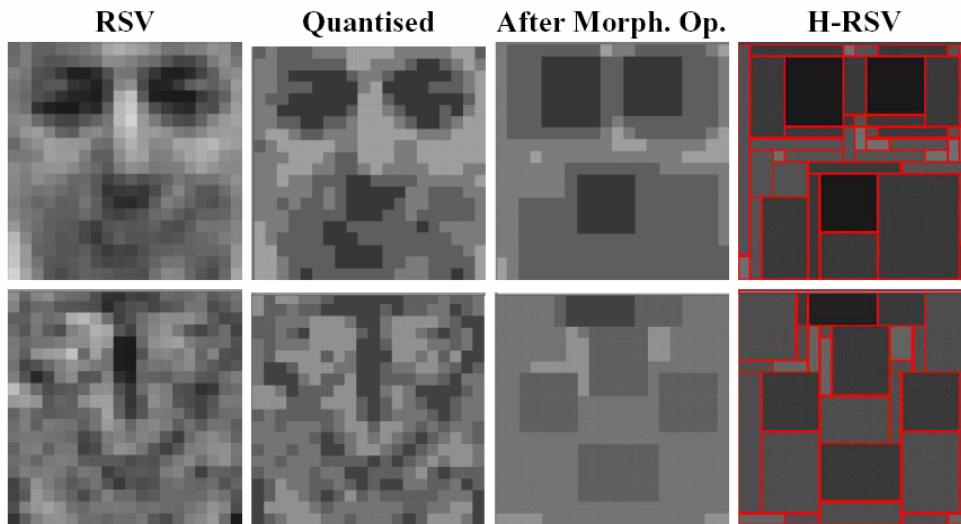


Figure 2-9: Stages of the Haar-like approximation using morphological filter. Approximation of a face (*top row*) and an anti-face (*bottom row*) Reduced Set Vector (*1st column*), quantisation using posterisation (*2nd column*), morphological filter to obtain the Haar-like structure (*3rd column*) and H-RSV's with red marked rectangle coordinates (*4th column*).

Simulated Annealing is used to obtain a minimum of the energy E_k by selecting the parameters v , A and B that minimises E_k . As these new Reduced Set Vectors have a Haar-like structure, we call them Haar-Reduced Set Vectors (H-RVM).

Wavelet Frame Approximation

The Haar-like RVM approximation at the previous section using Simulated Annealing optimisation does not provide the global optimum of the approximation in all cases and it is difficult to adjust the optimal resolution level. Hence, we introduce a third method using a sparse wavelet frame representation of the RSV's to obtain a Haar-like structure of the RSV's.

We apply the wavelet transform to find the best representation of the RSV's to the Reduced Support Vector Machine itself, and not of the input space as a pre-processing at working time, like [48] or [35]. For more details about wavelets, we would refer the reader to Mallat [56], Teolis [94], or Burrus et al. [8] and for a compact introduction Stollnitz et al. [91], [92]. Using the wavelet transformation instead of a heuristic optimisation, like used in [102] enables the fast and automatic learning stage of our proposed Wavelet Approximated Reduced SVM. The training is straightforward, and does not require the manual selection of ad-hoc parameters. For example, the training time is two hours, which is a vast improvement over previous detectors.

The wavelet approximation is the superior method to obtain a Haar-like structure of the RSV's and is used in the training of our W-RVM classifier applied to face, facial feature detection and further applications (see Chapter 3-5). How to generate the Wavelet Approximated RSV's is detailed in the next sections and summarised in Section 2.6.

In this thesis, we take advantage of recent progress in wavelet analysis. In the next two sections we detail how we apply a soft-shrinkage operation and an over-complete wavelet system in order to obtain an optimal sparse signal approximation in wavelet space.

Let us first briefly recall smoothness characterisation properties of wavelets and Besov norms (see [98], [83], [15]) we want to use it in the next section. One can determine the membership of a function in many different smoothness functional spaces by examining the decay properties of its wavelets coefficients. For a comprehensive introduction and overview on this topic we would refer the reader to the abundant literature, see e.g. Daubechies [21], [22], Cohen et al. [15], Dahmen [20], DeVore et al. [27], [26], Frazier et al. [32], Triebel [98]. For readers interested more in the gist of the theory than in a more elaborate, mathematically precise description, it suffices to know that:

- Wavelet expansions provide successive approximations at increasingly finer scales. If a function f is given, and $f_{(j)}$ is its approximation at scale 2^{-j} , then the next finer approximation $f^{(j+1)}$ can be written as

$$f_{(j+1)} = f_{(j)} + \sum_{i,k} \langle f, \tilde{\psi}_{j,k}^i \rangle \psi_{j,k}^i, \quad (2.10)$$

where $\psi_{j,k}^i(\mathbf{x}) = 2^j \psi^i(2^j x_1 - k_1, 2^j x_2 - k_2)$ are the wavelets used in the expansion, and $\tilde{\psi}_{j,k}^i$ a corresponding dual family. The index i indicates that in dimensions larger than 1, one typically uses several wavelet templates. In 2 dimensions, there are usually 3 different wavelets, and i takes the values 1,2,3 (Note that the details of the approximation scheme that computes $f_{(j)}$ from f depend on the wavelet family under consideration.). If $\psi \in C^s$ (i.e. ψ has ‘differentiability’ of order s , where s need not to be integer), then f has differentiability of order $r < s$ if and only if

$$\left| \langle f, \tilde{\psi}_{j,k}^i \rangle \right| \leq C 2^{-j(r+s)}. \quad (2.11)$$

For the sake of convenience, we shall often ‘bundle’ i, j, k into one index λ , and write, $\langle f, \tilde{\psi}_\lambda \rangle$ simply as f_λ . In this case $|\lambda|$ stands for j . In this notation, the requirement (2.11) becomes $|f_\lambda| \leq C 2^{-|\lambda|(r+s)}$.

- One can characterise the smoothness of f in detail by using several parameters to describe it, such as e.g. in Besov spaces. For smoothness $r < 1$, for instance, we define

$$\omega_r(f; t)_p = \sup_{|\mathbf{h}| \leq t} \left[\int |f(\mathbf{x} + \mathbf{h}) - f(\mathbf{x})|^p d\mathbf{x} \right]^{1/p} \quad (2.12)$$

(this is an ‘ L_p -measured modulus of continuity’ for f), and

$$|f|_{B_q^r(L_p(\Omega))} = \left(\int_0^\infty (t^{-r} \omega(f;t)_p)^q dt/t \right)^{1/p} \quad (2.13)$$

(Basically, this measures, in a fine ‘ q -gained scale’, whether $\omega(f;t)_p$ decays at least as fast as t^r when $t \rightarrow 0$.) For instance, if we consider, on $\Omega = (0, 1]^2$ the function $f(\mathbf{x}) = x_1 + x_2 - \lfloor x_1 + x_2 \rfloor$, where $\lfloor x \rfloor = \max \{n \in \mathbb{Z}; n < x\}$, which has a discontinuity along the diagonal $x_1 + x_2 = 1$ in the square, then we find

$$\omega(f;t)_1 \sim C |t| \text{ as } |t| \rightarrow 0$$

and one easily checks, $|f|_{B_1^{1-\varepsilon}(L_1(\Omega))} < \infty$ for all $\varepsilon > 0$. In fact, one has $B_1^1(L_1(\Omega))$ (i.e. with $r = 1$) as well, but to verify this we need a fancier ‘ L_1 -measured modulus of continuity’. One important link of wavelets to these detailed smoothness spaces is that they provide a good estimate of Besov norms. In particular, in 2 dimensions,

$$f \in B_1^{s+1}(L_1(\Omega)) \Leftrightarrow \sum_\lambda 2^{|\lambda|s} |f_\lambda| < \infty ; \quad (2.14)$$

for $s = 0$ this shows that $f \in B_1^1(L_1(\Omega))$ if and only if its coefficients are in ℓ_1 .

2.3.3. Soft-Shrinkage to Build Rectangular Structured W-RSV's

In order to exploit the Integral Image method a block-like approximation of the Reduced Set Vectors must be used, i.e. they must have a rectangular (Haar--like) structure with piecewise constant grey values. Therefore, we use Haar wavelets and not wavelets with more vanishing moments (e.g. Daubechies wavelets of higher order), even if they would in general result in a more sparse approximation.

We are searching for an approximation of a given image \mathbf{z} by a piecewise block structured image \mathbf{u} , which is as sparse as possible. This optimisation problem can be casted in the following variational form

$$\min_{\hat{\mathbf{u}}} \left\{ \|\mathbf{z} - \hat{\mathbf{u}}\|_{L_2}^2 + 2\mu |\hat{\mathbf{u}}|_{B_1^1(L_1)} \right\}, \quad (2.15)$$

where $B_1^1(L_1)$ denotes a particular Besov semi-norm. It is known that the Besov (semi) norm of a given function can be expressed by means of its wavelet coefficients. In two spatial dimensions the Besov penalty is nothing else than a ℓ_1 constraint on the wavelet coefficients (promoting sparsity as required). See Section 2.3.2 for an introduction and for more details we refer the reader to [98], [83] and for a comprehensive discussion of the problem to [15].

The minimisation of (2.15) is easily obtained: Let $\{\psi_\lambda\}_{\lambda \in \Lambda}$ be the underlying wavelet basis, where Λ is the index set over all possible locations, scalings and wavelet species. Then we may express \mathbf{z} and $\hat{\mathbf{u}}$ as follows: $\mathbf{z} = \sum_{\lambda \in \Lambda} z_\lambda \psi_\lambda$, $\hat{\mathbf{u}} = \sum_{\lambda \in \Lambda} \hat{u}_\lambda \psi_\lambda$, where $z_\lambda = \langle \mathbf{z}, \psi_\lambda \rangle$ and $\hat{u}_\lambda = \langle \hat{\mathbf{u}}, \psi_\lambda \rangle$. Thus, we may completely rewrite (2.15) in sequence space

$$\mathbf{u} = \arg \min_{\hat{\mathbf{u}}} \sum_{\lambda \in \Lambda} \{(z_\lambda - \hat{u}_\lambda)^2 + 2\mu |\hat{u}_\lambda|\}. \quad (2.16)$$

Minimising summand-wise, we obtain the following explicit expression for the optimum u_λ , see, e.g. [23],

$$u_\lambda = S_\mu(z_\lambda) = \text{sgn}(z_\lambda) \max\{|z_\lambda| - \mu, 0\}, \quad (2.17)$$

where S_μ is the soft-shrinkage operation with threshold μ . Consequently, the optimum \mathbf{u} is simply obtained by soft-shrinking the wavelet coefficients of \mathbf{z} , i.e.

$$\mathbf{u} = \sum_{\lambda \in \Lambda} S_\mu(z_\lambda) \psi_\lambda = W^{-1} S_\mu(W\mathbf{z}), \quad (2.18)$$

where W stands for the wavelet transform operator.

2.3.4. Over-Complete Wavelet Transform

We propose an optimal match by Over-Complete Wavelet Transform [94], [1], [36] using translated wavelet bases optimisation to overcome the windowing effect. Typically, orthogonal or so-called non-redundant representations and filtering very often creates artefacts in terms of undesirable oscillations or non-optimally represented details, which manifest themselves as ringing and edge blurring (also called Gibbs or windowing effect). For our purpose, it is essential to pick a representation that optimally meets the local image structure (see Figure 2-10). The most promising method for adequately solving the windowing problem has its origin in translation invariance (the method of cycle spinning, see e.g. [16]), i.e. representing the image by all possible shifted versions of the underlying (Haar) wavelet basis. But contrary to the idea of introducing redundancy by averaging over all possible representations of \mathbf{z} , we aim to pick only that one that optimally meets the given image structure.

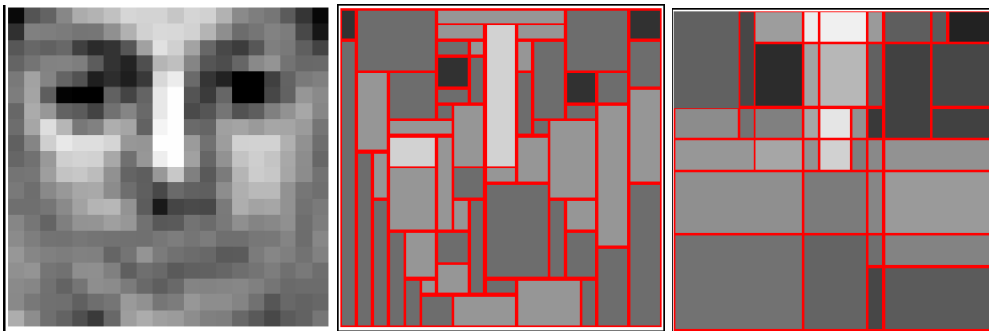


Figure 2-10: Examples for Haar-like approximations

RSV (*left*) approximated using morphological filter (H-RSV, *middle*) and using an OCWT (W-RSV, *right*). The OCWT representation meets optimally¹ the local image structure. Hence the ratio of the decreasing of the hyper-plane distance to the used operations is more efficient for the W-RSV (here e.g. 0.73) than for the H-RSV (0.51).

In order to give a rough sketch of this technique, assume that we are given an RSV \mathbf{z} with $2^M \times 2^M$ pixel. Following the cycle-spinning approach, see again [16], we have to compute $2^{2(M+1-j_0)}$ different representations of \mathbf{z} with respect to the $2^{2(M+1-j_0)}$ translates s of the underlying wavelet basis. The scale j_0 denotes the coarsest resolution level of \mathbf{z} . The family $\{\mathbf{z}^s\}_s$ generated this way serves now as our reservoir of possible wavelet representations of one single \mathbf{z} . The best shift s^* is that one for which we have a minimal discrepancy to the SVM hyper-plane per operations for the kernel-evaluation. We evaluate all possible local shifts (in our case 64), hence the global optimum shift is guaranteed.

2.4. Hyper-plane Approximation

We use a two-stage hyper-plane approximation from the original SVM to the Reduced SVM (RVM) and from the RVM to the Wavelet Approximated Reduced Vector Machine (W-RVM). The first reduction step was computing the RVM by minimising the hyper-plane distance $\|\Psi_{SVM} - \Psi_{RVM}\|_F$ in the feature space F [70] and [73] (see Section 2.1).

This yields $\Psi_{RVM} = \sum_{i=1}^{N_z} \beta_i \Phi(\mathbf{z}_i)$, with the mapping function $\Phi: \mathcal{X} \rightarrow F$, $\mathbf{x} \mapsto \Phi(\mathbf{x})$ as used for the SVM.

As outlined above, an essential improvement can be achieved by accelerating the numerical integration. To this end, we have suggested the use of Haar-like sparse approximations \mathbf{u}_i of \mathbf{z}_i that generates rectangular representations of the images and fits thus well with the concept of Integral Images. Replacing \mathbf{z}_i by \mathbf{u}_i amounts to $\sum_{i=1}^{N_z} \beta_i \Phi(\mathbf{u}_i)$. The change of the sup-

¹ Optimally approximated means here, the usage of an optimally shifted wavelet basis that represents the image as sparse as possible.

porting vectors might likely require a slight adjustment of the β_i 's which is done iteratively (see below), i.e. the second hyper-plane approximation we are proposing finally reads as

$$\Psi_{W-RVM} = \sum_{i=1}^{N_z} \gamma_i \Phi(\mathbf{u}_i). \quad (2.19)$$

The natural question that arises is how well approximates the reduced and Haar-like designed Ψ_{W-RVM} (2.19) the original SVM Ψ_{SVM} , i.e. we have to consider the quantity

$$\|\Psi_{SVM} - \Psi_{W-RVM}\|_F \leq \|\Psi_{SVM} - \Psi_{RVM}\|_F + \|\Psi_{RVM} - \Psi_{W-RVM}\|_F, \quad (2.20)$$

where the first misfit term on the right-hand side is minimised through the iterative method in [70] and [73]. It remains to analyse the second discrepancy between Ψ_{RVM} and Ψ_{W-RVM} .

By making use of kernel-based evaluations of the inner products (and using $k(\mathbf{z}_i, \mathbf{z}_i) = 1$) and Cauchy-Schwarz we obtain

$$\begin{aligned} \|\Psi_{RVM} - \Psi_{W-RVM}\|_F^2 &\leq \left(\sum_{i=1}^{N_z} \|\beta_i \Phi(\mathbf{z}_i) - \gamma_i \Phi(\mathbf{u}_i)\|_F \right)^2 \\ &= \langle I_{N_z \times N_z} \mathbf{1}, (\|\beta_1 \Phi(\mathbf{z}_1) - \gamma_1 \Phi(\mathbf{u}_1)\|_F, \dots, \|\beta_{N_z} \Phi(\mathbf{z}_{N_z}) - \gamma_{N_z} \Phi(\mathbf{u}_{N_z})\|_F) \rangle^2 \\ &\leq N_z \sum_{i=1}^{N_z} \|\beta_i \Phi(\mathbf{z}_i) - \gamma_i \Phi(\mathbf{u}_i)\|_F^2 \\ &= N_z \sum_{i=1}^{N_z} \{\beta_i^2 + \gamma_i^2 - 2\gamma_i \beta_i k(\mathbf{z}_i, \mathbf{u}_i)\} \\ &= N_z \left\{ \sum_{i=1}^{N_z} (\beta_i - \gamma_i)^2 + 2 \sum_{i=1}^{N_z} \gamma_i \beta_i (1 - k(\mathbf{z}_i, \mathbf{u}_i)) \right\} \\ &= N_z \left\{ \|\beta - \gamma\|^2 + 2 \sum_{i=1}^{N_z} \gamma_i \beta_i (1 - k(\mathbf{z}_i, \mathbf{u}_i)) \right\}. \end{aligned} \quad (2.21)$$

Now, when choosing the Gaussian kernel with kernel parameter, σ (optimised by the SVM training [100]) we may approximate $(1 - k(\mathbf{z}_i, \mathbf{u}_i))$ in (2.21) as follows

$$1 - k(\mathbf{z}_i, \mathbf{u}_i) = 1 - \exp\left(\frac{-\|\mathbf{z}_i - \mathbf{u}_i\|^2}{2\sigma^2}\right) = \frac{\|\mathbf{z}_i - \mathbf{u}_i\|^2}{2\sigma^2} + O(\|\cdot\|^4). \quad (2.22)$$

Thus the data misfit discrepancy is directly controlled by the ℓ_2 distance of the sparse approximation \mathbf{u}_i of \mathbf{z}_i (which is minimised under sparsity constraints) and the distance $\|\beta - \gamma\|$. Thus, up to higher-order terms, we achieve

$$\|\Psi_{RVM} - \Psi_{W-RVM}\|_F^2 \approx N_z \{ \|\beta - \gamma\|^2 + \sigma^{-2} \sum_{i=1}^{N_z} \gamma_i \beta_i \|\mathbf{z}_i - \mathbf{u}_i\|^2 \}, \quad (2.23)$$

where the relation between the error of the Wavelet Approximated Reduced Set Vectors and the threshold parameter μ needs to be made. This is important to control the trade-off between sparsity (i.e. computational cost) and the approximation (classification) preciseness per approximated vector.

At first, we consider the difference of the Reduced and Wavelet Approximated Reduced Set Vectors and express them by means of the corresponding wavelet coefficients, i.e.

$$\|\mathbf{z}_i - \mathbf{u}_i\|^2 = \sum_{\lambda \in \Lambda} (z_{i,\lambda} - S_\mu(z_{i,\lambda}))^2. \quad (2.24)$$

Assuming further that \mathbf{z} consists of $2^M \times 2^M$ pixel and $(z_{i,\lambda} - S_\mu(z_{i,\lambda}))^2 \leq \mu$ using (2.17), we have

$$1 - k(\mathbf{z}_i, \mathbf{u}_i) \leq 1 - \exp\left(\frac{-2^{2M} \mu^2}{2\sigma^2}\right). \quad (2.25)$$

Applying this to (2.21) an upper bound E for the worst-case error is then given by

$$\begin{aligned} \|\Psi_{RVM} - \Psi_{W-RVM}\|_F^2 &\leq N_z \{ \|\beta - \gamma\|^2 + 2 \left(1 - \exp\left(\frac{-2^{2M} \mu^2}{2\sigma^2}\right) \right) \sum_{i=1}^{N_z} \beta_i \gamma_i \} \\ &=: E(\mu). \end{aligned} \quad (2.26)$$

Neglecting higher-order terms of the exp series, we may write

$$E(\mu) \approx N_z \left(\sigma^{-2} 2^{2M} \mu^2 \sum_{i=1}^{N_z} \beta_i \gamma_i + \|\beta - \gamma\|^2 \right). \quad (2.27)$$

From the last formula we see that the influence of μ is of quadratic nature which assures a rapid error decay of the left-hand summand. The quantity $\|\beta - \gamma\|^2$ will be studied below when we have exploited a rule for deriving the vector γ . In the limit case, $\mu \rightarrow 0$, we then achieve $\lim_{\mu \rightarrow 0} E(\mu) = 0$, which shows that the proposed scheme acts in the limit case as the RVM. For the case in which we really achieve complexity reduction by sparsity and thus a significant gain in computational time and cost, we refer to Section 3.2.

2.5. Hierarchical Evaluation via Resolution Levels

The early rejection of easy to discriminate vectors is achieved by a Double Cascade. The inner cascade is a hierarchy over the number $i = 1, \dots, N_z$ of incorporated W-RSV's, \mathbf{u}_i^l . After incorporating a certain number of W-RSV's with a constant resolution level l it is more efficient to improve the approximation accuracy of the first (already incorporated) vectors. Hence, we train $l = 0, \dots, L$ sets of W-RSV's for the outer cascade of coarse-to-fine resolution levels. To use the cascade over the resolution levels as inner loop and over the W-RSV's as

outer loop should result in similar performance. To keep the method simple, we only propose one realisation of the Double Cascade. The trade-off between the two cascades is determined in Section 2.8. To exploit these cascades is the superior way to reject most image points by only a few operations. Moreover, this novel method is robust since the adjustment of only one optimal resolution level is sensitive. The proposed evaluation selects the most efficient approximation accuracy automatically at detection time based on the image patch to be classified. In contrast to former methods, the trade-off between accuracy and speed is smooth, so that image points are rejected earlier. Therefore, the approach is robust, not sensitive to the parameter choice at training time, simple to use and fast.

2.6. Algorithm to Generate Hierarchically Refined W-RSV's

The algorithm is based on residual Haar wavelet approximations of the RSV's \mathbf{z}_i , which are pre-computed by minimising $\|\Psi_{SVM} - \Psi_{RVM}\|_F^2$ via the algorithm suggested in [73].

Before presenting the algorithm, we introduce the basic quantities. To find the optimal match (see OCWT in Section 2.3.4) we use a translated wavelet bases with an offset up to $2^J \times 2^J$. To avoid the ringing effect $J = \log_2(2^M/4)$ (i.e. about a quarter of the dimensions of \mathbf{z}) is sufficient. Starting with computing 2^{2J} different initial Haar-like approximations $\mathbf{r}_i^{0,s}$ by (2.18), where $s \in \{1, \dots, 2^J\}^2$ is the shift of the underlying Haar wavelet basis, we recursively define for $l = 0, \dots, L$ and $i = 1, \dots, N_z$

$$\begin{aligned} \mathbf{u}_i^l &= \sum_{j=0}^l \mathbf{r}_i^{j,s^*}, \\ \mathbf{r}_i^{l+1,s} &= (W^s)^{-1} S_\mu(W^s(\mathbf{z}_i - \mathbf{u}_i^l)), \end{aligned} \quad (2.28)$$

where the shift s^* denotes the best shift (selected by an optimally criterion introduced below) of the residual at resolution level l , see Figure 2-11. Note that s^* may differ for each $\mathbf{r}_i^{l,s}$. Within this setting each Reduced Set Vector \mathbf{z}_i is then approximated at level l by \mathbf{u}_i^l . The benefit of the residual structure is that (i) \mathbf{u}_i^l converge to \mathbf{z}_i , if $l \rightarrow \infty$, (ii) we can store all the residuals and thus they do not need to be recomputed in the cascade step when tuning the resolution (i.e. the accuracy of the W-RSV representation) from coarse to fine, and (iii) the evaluation of the kernel at runtime is more efficient (detailed later at (2.38) in Section 2.7).

To incorporate the next optimal W-RSV we have to evaluate the computational cost and the discrepancy of the cascaded W-RVM to the original SVM. Such a discrepancy depends on the resolution level l and the number i of incorporated W-RSV's. Only $\mathbf{r}_i^{l,s}$ changes for the

optimisation steps over all offsets s . Therefore using the expanded form (2.28) in (2.19) the discrepancy of the hyper-planes becomes,

$$\delta_i^l(s) = \left\| \Psi_{SVM} - \sum_{k=1}^{i-1} \gamma_k^{l,i} \Phi(\mathbf{u}_k^l) - \gamma_i^{l,i} \Phi(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}) - \sum_{k=i+1}^{N_z} \gamma_k^{l,i} \Phi(\mathbf{u}_k^{l-1}) \right\|_F^2, \quad (2.29)$$

where we set $\mathbf{u}_i^{-1} = 0$. The cascade structure is thus achieved when adding residuals $i \rightarrow i+1$ and then, after reaching $i = N_z$, passing to the next level $l \rightarrow l+1$, i.e. subsequently adding $\mathbf{r}_i^{l,s}$. Note that for each added residual $\mathbf{r}_i^{l,s}$ we have to compute a new vector $\gamma^{l,i} = (\gamma_1^{l,i}, \dots, \gamma_{N_z}^{l,i})'$. Since we are searching for the best shift s for $\mathbf{r}_i^{l,s}$ and the optimal $\gamma^{l,i}$, we have to minimise $\delta_i^l(s)$. The optimal vector $\gamma^{l,i}$ can be computed explicitly. Introducing the $N_x \times N_z$ matrix

$$\Phi_{\mathbf{x}, \mathbf{u}}^{l,i,s} = \begin{pmatrix} \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{u}_1^l) \rangle & \dots & \langle \Phi(\mathbf{x}_{N_x}), \Phi(\mathbf{u}_1^l) \rangle \\ \vdots & \ddots & \vdots \\ \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{u}_{i-1}^l) \rangle & \dots & \langle \Phi(\mathbf{x}_{N_x}), \Phi(\mathbf{u}_{i-1}^l) \rangle \\ & & \mathbf{v}^s \\ \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{u}_{i+1}^{l-1}) \rangle & \dots & \langle \Phi(\mathbf{x}_{N_x}), \Phi(\mathbf{u}_{i+1}^{l-1}) \rangle \\ \vdots & \ddots & \vdots \\ \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{u}_{N_z}^{l-1}) \rangle & \dots & \langle \Phi(\mathbf{x}_{N_x}), \Phi(\mathbf{u}_{N_z}^{l-1}) \rangle \end{pmatrix}$$

with the i -th row

$$\mathbf{v}^s = (\langle \Phi(\mathbf{x}_1), \Phi(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}) \rangle, \dots, \langle \Phi(\mathbf{x}_{N_x}), \Phi(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}) \rangle)$$

and the same way the $N_z \times N_z$ matrix $\Phi_{\mathbf{u}, \mathbf{u}}^{l,i,s}$ with entries $\langle \Phi(\mathbf{u}_i^l), \Phi(\mathbf{u}_i^l) \rangle$ but where the i th row is replaced with

$$\mathbf{w}^s = (\langle \Phi(\mathbf{u}_1^l), \Phi(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}) \rangle, \dots, \langle \Phi(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}), \Phi(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}) \rangle, \dots, \langle \Phi(\mathbf{u}_{N_z}^{l-1}), \Phi(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}) \rangle)$$

and i -th column with $(\mathbf{w}^s)'$, we recast the discrepancy $\delta_i^l(s)$ as follows,

$$\delta_i^l(s) = \left\| \Psi_{SVM} \right\|_F^2 - 2(\gamma^{l,i})' \Phi_{\mathbf{x}, \mathbf{u}}^{l,i,s} \alpha + (\gamma^{l,i})' \Phi_{\mathbf{u}, \mathbf{u}}^{l,i,s} \gamma^{l,i}, \quad (2.30)$$

where α is the vector of the non-vanishing coefficients of the SVM hyper-plane $\Psi_{SVM} = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$. Evaluating the derivative of the discrepancy (2.30) and setting it to 0, the optimal $\gamma^{l,i}$ is then obtained by

$$\gamma^{l,i}(s) = (\Phi_{\mathbf{u}, \mathbf{u}}^{l,i,s})^{-1} \Phi_{\mathbf{x}, \mathbf{u}}^{l,i,s} \alpha \quad (2.31)$$

and depends thus on s . With the explicit expression (2.31), the discrepancy (2.30) becomes

$$\delta_i^l(s) = \left\| \Psi_{SVM} \right\|_F^2 - \alpha' (\Phi_{\mathbf{x}, \mathbf{u}}^{l,i,s}) (\Phi_{\mathbf{u}, \mathbf{u}}^{l,i,s})^{-1} \Phi_{\mathbf{x}, \mathbf{u}}^{l,i,s} \alpha. \quad (2.32)$$

This of course requires the existence of $(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s})^{-1}$ what clearly means then linear independency of all involved $\Phi(\cdot)$'s. If this cannot be assured, we have to consider a regularised version of $\delta_i^l(s)$, namely

$$\delta_i^l(s) = \|\Psi_{SVM}\|_F^2 - 2(\gamma^{l,i})' \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} \alpha + (\gamma^{l,i})' (\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s} + \rho) \gamma^{l,i}.$$

This yields

$$\gamma^{l,i}(s) = (\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s} + \rho)^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} \alpha \quad (2.33)$$

and thus

$$\delta_i^l(s) = \|\Psi_{SVM}\|_F^2 - \alpha' (\Phi_{\mathbf{x},\mathbf{u}}^{l,i,s}) (\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s} + \rho)^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} \alpha. \quad (2.34)$$

With the matrix notation, the double-cascade structure now becomes more visible: beside the residual cascade with respect to l in the approximation of each \mathbf{z}_i by \mathbf{u}_i^l , there is for each l a matrix cascade structure with respect to i that allows to store the entries up to the i -th row in $\Phi_{\mathbf{x},\mathbf{u}}^{l,i,s}$ and up to i -th row and i -th column in $\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s}$. The remaining entries $(\Phi_{\mathbf{x},\mathbf{u}}^{l,i,s})_{n,m}$ for $m > i$ and $(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s})_{n,m}$ for $n, m > i$ can be taken from the previous level $l-1$.

We summarise our findings and design the algorithm for the learning stage of the W-RVM (see Table 2-1). An example of the approximation of an RSV with the proposed approach is shown in Figure 2-11.

Learning stage of the W-RVM classifier:

Input: SVM with $\alpha_i, \mathbf{x}_i, i = 1, \dots, N_x$,
RVM with $\beta_i, \mathbf{z}_i, i = 1, \dots, N_z$

Output: W-RVM with the rectangle structures of r_i^l and the coefficient vectors $\hat{\gamma}^{l,i}, i = 1, \dots, N_z(l), l = 0, \dots, L$

1. Set $\Psi_{SVM} = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$, $\mathbf{u}_i^{-1} = 0$ and set $l = 0$
2. Start with $i = 1$
3. Compute for $s \in \{1, \dots, 2^J\}^2$, $J = \log_2(2^M / 4)$

$$\mathbf{u}_i^{l-1} = \sum_{j=0}^{l-1} r_i^{j,s^*}$$

$$r_i^{l,s} = (W^s)^{-1} S_\mu(W^s(\mathbf{z}_i - \mathbf{u}_i^{l-1})),$$

where s^* denotes the best shift, S_μ is the shrinkage function (2.17) with the sparsity parameter μ discussed in Section 2.8 and W is the wavelet transform operator

4. Compute $\forall s \in \{1, \dots, 2^J\}^2$ the decrement of the discrepancy (2.34)
 - if $i = 1, l = 0$: $\delta \Delta_i^l(s) = \|\Psi_{SVM}\|_F^2 - \delta_1^0(s)$
 - if $i = 1, l > 0$: $\delta \Delta_i^l(s) = \delta_{N_z}^{l-1}(s^*) - \delta_1^l(s)$
 - else: $\delta \Delta_i^l(s) = \delta_{i-1}^l(s^*) - \delta_i^l(s)$

and the number of operations

$$\omega \Delta_i^l(s) = 4 \#[r_i^{l,s}] + v(r_i^{l,s}),$$

where $\#[r_i^{l,s}]$ is the number of piecewise constant rectangles and $v(r_i^{l,s})$ the number of grey values of $r_i^{l,s}$

5. Select the best shift s^* out of $\{1, 2, \dots, 2^J\}^2$ by

$$s^* = \arg \max_s \frac{\delta \Delta_i^l(s)}{\omega \Delta_i^l(s)}$$

6. Save the rectangle structure of r_i^{l,s^*} and the coefficient vector

$$\hat{\gamma}^{l,i} = \gamma^{l,i}(s^*) = (\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s^*})^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s^*} \alpha$$

7. If $i < N_z(l)$, increment i and proceed to step 3.
If $i = N_z(l)$ and $l < L$, increment l and proceed to step 2 ($N_z(l)$ and L are obtained using (2.39),(2.40));
else, stop.

Table 2-1: Summary of the Training of the W-RVM classifier

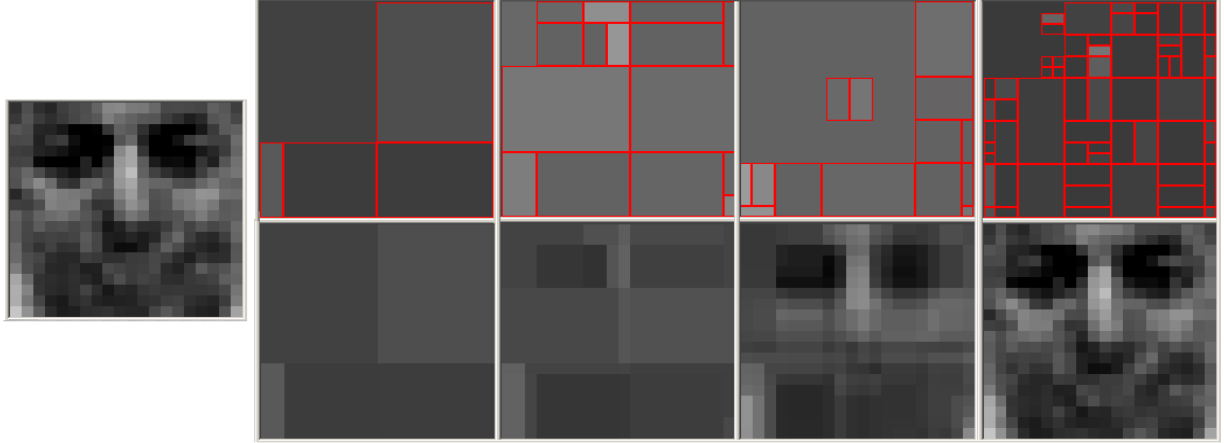


Figure 2-11: Example of cascaded approximating a RSV

Left: a RSV; right, bottom row: W-RSV's at different resolution levels (left to right: level: 0,1,9,18); top row: related wavelet approximated residuals (left to right: level: 0,1,9,18).

Finally, as a by-product of this section and as a contribution to Section 2.4, we are now able to quantify $\|\beta - \gamma\|$. Assume, the SVM is given by N_x Support Vectors \mathbf{x}_i and the RVM by N_z Reduced Set Vectors \mathbf{z}_i , then with $(\Phi_{z,z})_{i,j} = \langle \Phi(\mathbf{z}_i), \Phi(\mathbf{z}_j) \rangle$ and $(\Phi_{x,z})_{i,j} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{z}_j) \rangle$ it is common that $\beta = \Phi_{z,z}^{-1} \Phi_{z,x} \alpha$, see [73]. Consequently,

$$\|\beta - \hat{\gamma}^{l,i}\| \leq \|\Phi_{z,z}^{-1} \Phi_{x,z} - (\Phi_{u,u}^{l,i,s^*})^{-1} \Phi_{x,u}^{l,i,s^*}\| \|\alpha\| \quad (2.35)$$

and since we have $\|\mathbf{u}_i^l - \mathbf{z}_i\| \leq C_{\mu,l}$, by perturbation arguments we also have an entry-wise perturbation estimate for the full matrices which in turn yield an estimate for $\|\beta - \hat{\gamma}^{l,i}\|$ in dependence on μ and l (we omit a detailed examination here). Moreover, as the approximations \mathbf{u}_i^l at resolution level l tend to \mathbf{z}_i as μ tends to 0, we have an entry-wise convergence

$$\Phi_{u,u}^{l,i,s^*} \rightarrow \Phi_{z,z}, \quad \Phi_{x,u}^{l,i,s^*} \rightarrow \Phi_{x,z} \quad (2.36)$$

and hence

$$\|\Phi_{z,z}^{-1} \Phi_{x,z} - (\Phi_{u,u}^{l,i,s^*})^{-1} \Phi_{x,u}^{l,i,s^*}\| \xrightarrow{\mu \rightarrow 0} 0 \quad (2.37)$$

2.7. Detection Process

The classification function of the W-RVM, denoted by $y_i^l(\mathbf{x})$ of the input patch \mathbf{x} , using N_z W-RSV's at the levels $0, \dots, l-1$ and i W-RSV's at the level l is as follows:

$$\begin{aligned} y_i^l(\mathbf{x}) &= \text{sgn}\left(\sum_{k=1}^i \hat{\gamma}_k^{l,i} k(\mathbf{x}, \mathbf{u}_k^l) + \sum_{k=i+1}^{N_z} \hat{\gamma}_k^{l,i} k(\mathbf{x}, \mathbf{u}_k^{l-1}) + b_i^l\right) \\ k(\mathbf{x}, \mathbf{u}_i^l) &= \exp\left(-\frac{1}{2\sigma^2}(\mathbf{x}'\mathbf{x} - 2\mathbf{x}'\mathbf{u}_i^l + \mathbf{u}_i^l\mathbf{u}_i^l)\right), \end{aligned} \quad (2.38)$$

where the kernel k is efficiently evaluated using Integral Images (Section 2.3.1). For the term $2\mathbf{x}'\mathbf{u}_i^l = 2\mathbf{x}'\mathbf{u}_i^{l-1} + 2\mathbf{x}'\mathbf{r}_i^{l,s*}$ only $2\mathbf{x}'\mathbf{r}_i^{l,s*}$ has to be computed, since $2\mathbf{x}'\mathbf{u}_i^{l-1}$ can be stored at the previous level. The thresholds b_i^l are obtained automatically from an R.O.C. (Receiver Operating Characteristic) for a given accuracy. These thresholds are set to yield a given False Rejection Rate (FRR) so that the accuracy of the W-RVM can be same as the one of the full SVM (see [70] for details). The given False Rejection Rate also controls the trade-off between computational cost and detection performance and depends on the requirements of the application. If only few false rejections are acceptable (yields higher computational cost and more false alarms), a smaller FRR should be adjusted. This ratio between FRR and FAR is the only parameter of our algorithm to be set by the user. This ensures a simple to adjust detection approach.

Realising our Double Cascade algorithm (Section 2.5) the detection process goes as see in Table 2-2.

Working stage of the W-RVM classifier:

1. Start at the first resolution level $l = 0$.
2. Start with the first W-RSV, \mathbf{u}_1^l at the level l .
3. Evaluate $y_i^l(\mathbf{x})$ for the input patch \mathbf{x} using (2.38).
4. If $y_i^l < 0$ then the patch is classified as not being the object of interest, the evaluation stops.
5. If $i < N_z(l)$, i is incremented and the algorithm proceeds to step 3; else if $l < L$, l is incremented and the algorithm proceeds to step 2; otherwise the full SVM is used to classify the patch.

Table 2-2: Summary of the working stages of the W-RVM classifier

2.8. Adjustment of Resolution Levels and Number of W-RSV's per Level

When computing an approximation of an SVM, it is not clear how many approximation vectors N_z should be computed (see [73]). This number of vectors may vary depending on the level l of the approximation. To this end, it may be useful to let N_z depend on l . The reason is that at a certain point of the evaluation algorithm it is more efficient to increment l (and reset i), rather than to increment i . The best value of $N_z(l)$ is computed in an offline process using a validation dataset: $N_z(l)$ is set to the smallest i for which empirically

$$\frac{\text{Nops}(y_{i+1}^l)}{\text{Nrecs}(y_{i+1}^l)} > \frac{\text{Nops}(y_1^{l+1})}{\text{Nrecs}(y_1^{l+1})}, \quad (2.39)$$

where Nops stands for the number of operations and Nrecs stands for the number of rejections of the negative examples.

By a similar evaluation the last used resolution level L can be achieved. The optimal L is the smallest l that fulfils

$$\frac{\text{Nops}(y_i^{l+1})}{\text{Nrecs}(y_i^{l+1})} > \frac{\text{Nops}(y)}{\text{Nrecs}(y)}, \quad (2.40)$$

where y denotes the decision function of the full SVM (2.1). For this L it is more efficient to classify the last few remaining patches by the SVM, instead of incrementing l . L depends also on the threshold parameter μ . The smaller μ , the closer is \mathbf{u}_i^l to \mathbf{z}_i and the fewer resolution levels are required. However, the number of levels does not play a decisive role as the higher L , the sooner the evaluation process selects the next level, i.e. the less $N_z(l)$. Therefore our proposed approach is not very sensitive to the parameter for setting the approximation accuracy (e.g. for μ in (2.17) a constant $\mu = 0.8 \max(\text{abs}(z_\lambda))$ can be used). Opposite to former methods, using only one resolution level, the approach is simple and not sensitive to the parameter choice. The evaluation selects the most efficient approximation accuracy automatically at detection time.

Chapter 3

Face and Facial Feature Detection based on the W-RVM

In this chapter, we develop a real-time face detection system by applying the invented classifier from Chapter 2. We show that the novel Wavelet Approximated Reduced Vector Machine approach (W-RVM) yields a comparable accuracy to the SVM while providing a significant speed-up. We carried out experiments on well-known databases, like FERET [61] to provide the comparability to other approaches. In Section 3.3, we want to apply the W-RVM and the developed detector principles for a facial feature detection system. We will take advantage of a first unification of the 3D Morphable Model and the W-RVM. The 3D MM is used to generate synthetic training and validation data. Applying the W-RVM classifiers trained on this data we generate detection candidates for all feature points as input for the 3D model again. The 3D Prior Shape Model function uses the 3D MM to find the final feature assortment within all combinatory possible sets of the candidates for the feature points.

3.1. Data Sets for Training and Validation

Large and accurate labelled data sets are needed to train and validate classifier. In the case of faces there are databases available, like the MIT-CMU database [93] used for face and facial feature detection e.g. in [84], [79], [73], the Face Recognition Technology (FERET) program database [61] used e.g. in [14], [47], [80], the Biometric Experimentation Environment (BEE) for the Face Recognition Vendor Tests (FRVT) and Face Recognition Grand Challenge (FRGC) [62] used e.g. in [103], the CMU Pose, Illumination, and Expression (PIE) Database [90] used e.g. in [42], or the BioID Face Database [34] used for face detection e.g. in [39], [51].

We used mainly two databases for the training and validation of the face classifier: The first set was crawled from the WWW (WebCrawled face set) by the research group of Hans Burkhardt of the University of Freiburg and contains 18,213 faces and 93,630 non-faces. We chose this dataset because it has large variations in lighting, pose and expression. As second face database, we used the greyscale version of FERET [61]. We chose this well-known dataset to provide the comparability to other approaches. In the experiments, we used particularly face images sets from our previous projects [73] and from MIT-CMU and BEE.

If we want to train other objects like arbitrary facial feature points there are no datasets available or labels are missing. Moreover, it is important to cover all variations of lighting, pose or image noise conditions. Therefore, we use the 3D Morphable Model to generate images as training and validation data sets. With the Morphable Model fitting function, 3D face representations can be generated from 2D images. Faces can be rendered under varying pose and illumination conditions to build a large set of natural and synthetic images.

The advantage of synthetic data sets is that the illumination, the pose, the noise, etc. can be controlled. The size of the training set can be reduced, because specific characteristics can be trained and the known range of the parameters (e.g. the pose) can be used to optimise the training and detection process.

3.1.1. Generation of Training Sets using the 3D MM

On a random face generated by the Morphable Model Toolbox [77] as seen in Figure 3-1, we mark points of interest once, e.g. with a vertex picker the red dots as seen left in Figure 3-3. These points correspond to vertex numbers within the 3D face space.



Figure 3-1: Random face and corresponding vertex numbers

The fitting function of the 3D MM generates from a single 2D photograph a 3D face shape from a person. Each vertex of this shape is registered to the corresponded point at the 2D image (see Section 5.1.1 for an introduction to the MM and the registration). Therefore, taking advantage of the 3D MM registration, we know the 2D location for each of the once-

selected points within each fitted image, e.g. the corner of the mouth, the eyes or the nose tip. Now we define the facial feature areas around the defined points. For example as seen in Figure 3-2 the feature area of the left eye is defined as the area centred at the average point (white cross in image top left) of the left and right corners of the eye (white points, these are the corresponding vertex positions of the fitting). The height of the feature is defined equal to the distance between the eye corners, and the width is defined as the double of the eye corner distance. The once defined area (white corners) can now cut out (middle left image) and zoomed to the defined patch size (top left image) for all fitted images (bottom right image).

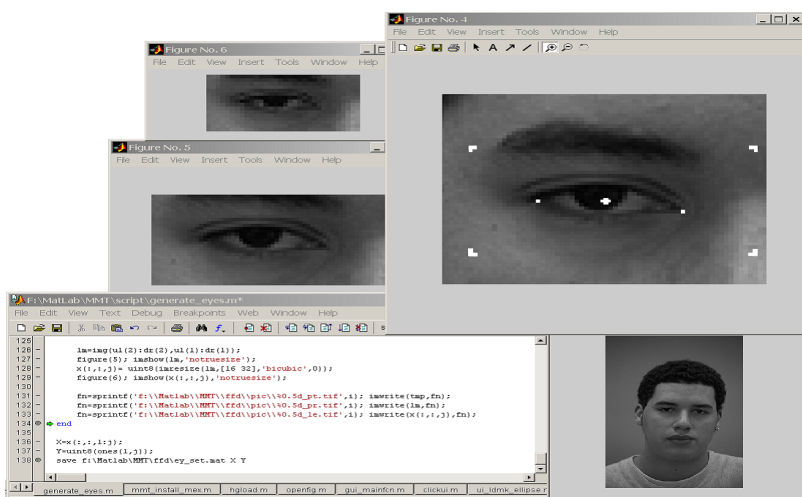


Figure 3-2: Definition of the facial feature areas
 Using the once pointed vertexes the facial feature areas can be defined in a script, i.e. the location, the size within the face, the aspect ratio and the resolution (*top right*). Then they can be automatically cut out (*middle left*) in all registered images (*bottom right*) and zoomed to the defined patch size (*top left*).

In Figure 3-3 the process is shown from picking the vertex positions (left), defining once the areas (middle) and cutting the patches (right) for example for the training set for the left and right eye, the nose tip, and the left and right mouth corner.



Figure 3-3: Generation of training sets from the FERET database
 Selecting once points at a random model face, we can select them on all 3D MM registered faces and cut out automatically training patches from large sets of images.

3.1.2. Generation of Synthetic Training Sets

In the previous section, we described how to generate training patches from images fitted by the 3D MM. The advantage using the MM is that we do not have to label manually the full database. In this thesis, we used 1920 fittings from FERET [61]. If we want to use larger training sets or specific variations, we can take advantage of the 3D Morphable Model, by altering the images within the face space and render new synthetic 2D images. We will generate synthetic images at different stages. The first class is taken from the original images without any modifications; in the second stage, we alter the environment parameter like the pose and the background; in the third, we alter also the texture and at the last stage we use additionally a synthetic shape. The last class is full synthetic; that means no fitting result of a real existing person is used. All classes have advantage and disadvantage which we will discuss in the following.

Generation of Training Sets with Natural Environment, Texture and Shape

If we automatically generate training examples as described in Section 3.1.1, we obtain parts of original images without any modifications of the subjects or the environment. The advantage of this image class is that the environment, like the lighting, background, image noise, the pose and the in-plane rotation and also the shape and texture of the persons are natural or natural distributed and so most comparable to image we want to classify with the trained W-RVM's. The disadvantage is that the size of the training sets is limited to the size of images fitted from the database, also that the variability of the subjects is restricted to the available fittings. We used here mostly the FERET database [61]. In that database, for example all images have almost no in-plane rotation, so we could not train rotated faces or facial features. In addition, we have to consider that the fitting results are flawed. We manually excluded not correctly fitted images, but all fittings have a small registration error within the range up to 10% of the inter-eye distance, so in the case of the FERET images up to five pixels. Therefore, our training and validation sets have in that image class a labelling error correlated to the fitting accuracy.

Generation of Training Sets with Synthetic Environment

To overcome the limitations using only the original images we want to take advantage of the Morphable Model by altering the environment parameter at the 3D space and render faces with for example different roll, yaw and pitch angle. The face with the new pose does not fit into the original background. In the case of FERET, the original background is anyhow unfavourable, because all images have the same homogeneous background with about the

same greyscale value. Therefore we generate a synthetic background with Gaussian distribute noise, contrast and brightness. The MM uses the phong lighting model. We alter the 3D position and intensity of the single light source and the intensity of the ambient light. Also we add Gaussian noise to the rendered face area to simulate the different quality of image sources. Variations of example images from one fitting of an FERET image are seen in Figure 3-4. Because the once selected vertex locations are moved simultaneous within the 3D space, we can cut faces or various facial feature areas the same way than for the first image class.

The advantage of this image class is that the subjects are natural, and we can cover all variations of the environment parameters as pose, illumination or background. The disadvantage is that we are limited to the number of subjects contained in the fitted set and have to handle the fitting inaccuracy.



Figure 3-4: Generation of training sets with synthetic environment
Changing the environment like the background, lighting, or pose.

Generation of Training Sets with Synthetic Texture

In that image class we do not use the original extracted texture, but the by the MM estimated model texture, as seen in Figure 3-5 for one subject. All the variations of the environment parameter are the same as by the previous class. The advantage is that the accuracy of the labels is higher and changing the illumination of the faces has fewer artefacts. The disadvantage is that beards, glasses and other details not represented in the face model are not present in these synthetic images. In this class there are also limitations concerning open-mouth examples and the gaze direction of the eyes.

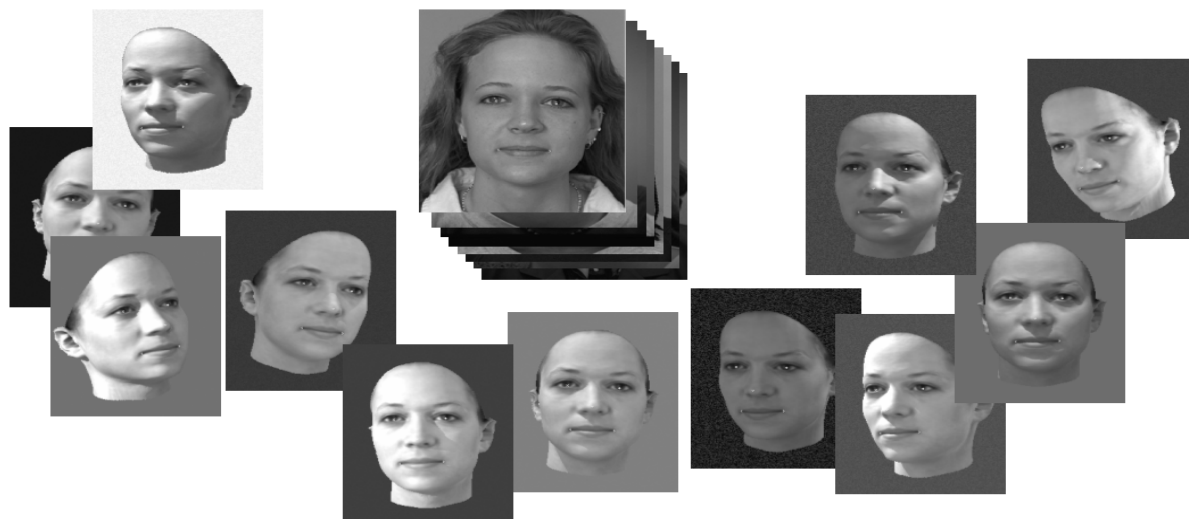


Figure 3-5: Generation of training sets with synthetic texture

Generation of Training Sets with full Synthetic Subjects

Examples where no fitting of a subject is used are shown in Figure 3-6. The texture and the shape parameter of the 3D MM are randomly chosen. This class can be used for the synthesis of many variations within the class of human faces, because each data point in the model space is again a face. The advantage is that an unlimited number of faces can be generated and the labels are absolutely correct, because no fitting is used. The disadvantage is that, like in the previous class, only characteristics represented in the model can be rendered.

Taking in account that all classes have advantages and disadvantages we use a mixture of examples from these four classes. We could show that the accuracy of the classifiers improved using different classes of synthetic images. We worked out specific experiments using the introduced synthetic images classes within the W-RVM Regression project (see Section 5.2.3).



Figure 3-6: Synthetic texture and shape of not existing individuals

3.1.3. Generation of Negative Training Sets

In addition, a wide range of negative examples is needed to train the classifier. In most approaches, images where no faces are present for sure are used to generate negative examples. However, in our approach we want to use the facial feature detectors only within the before detected face areas (see Section 3.5). Therefore, the negative examples should be mostly taken from these images areas to obtain best results, but we have to ensure that no positive examples are included. The 3D MM can be exploited again.

Comparable to the definition of facial feature areas in Section 3.1.1, we select points of interest in a random model face (e.g. red crosses in Figure 3-7, top left). Now we can use the selected vertex positions in all registered images, fitted by the MM. We define once a face area we want to exclude for this feature point. In Figure 3-7, top left, this area is visualised for the left eye as bright area. The excluded area is in that case defined by the distance between the left and right corner of the eye around the eye corners (so that the overlap of a negative patch with the nearest positive examples is less than a quarter).

Now we can randomly select patches from the image and add them to the negative training or validation set if their centre lay not within the excluded area. For example in Figure 3-7, top right the bright patch is not cut out, zoomed to the defined patch size and added. We also compute the variance within the image and threshold areas with a small value as seen in Figure 3-7, top middle as white areas. Only few negative examples should be located at the background or homogeneous parts of the images. If this limited number per image is reached (visualised as black patches in Figure 3-7, top right), then only patches with a higher variance, i.e. not located in the white parts of the top middle image, are added (visualised as white patches in Figure 3-7, top right).

At the working process of the detectors, different zooming levels of the images are used to detect facial feature points with different size. Therefore, also the negative examples should cover images areas with different resolution. This is realised using the same procedure above on different zooming levels of the example images as demonstrated in Figure 3-7, bottom row.

The four classes of synthetic images are used the same way as defined for the positive examples. Images are synthesised with many variations within the class of human faces and variations of the appearance, such as pose or illumination. To improve the accuracy of the classifier bootstrapping can also be used. That means after a first training of a classifier the negative training set is enlarged by false positive patches obtained by this classifier. Then, a new improved classifier can be trained on the larger training set.



Figure 3-7: Generation of Negative Training Sets

3.2. Applying the W-RVM for Face Detection

First, we applied our novel Wavelet Approximated Reduced Vector Machine to the task of face detection. In Section 3.1, we introduced the FERET and WebCrawled face database we used for training and validation.

The training set includes 3500, 20×20 , face patches and 20000 non-face patches from the WebCrawled dataset. The SVM computed on the training set yielded about 8000 Support Vectors that we approximated by $N_z = 90$ W-RSV's at $L = 5$ resolution levels by the method detailed in the previous chapter (e.g. L using (2.40)). For the OCWT (Section 2.3.4) we used the classical mirroring for adequately continuing the image beyond the boundaries.

As first validation set (set I) we used 1000 face patches, and 100,000 non-face patches randomly chosen also from the WebCrawled dataset images, but disjoint from the training examples. The first graph on Figure 3-8 plots the residual distance of the RVM (dashed line) and of the W-RVM (plain line) to the SVM (in terms of the distance $\Psi_{SVM} - \Psi_{RVM}$ and $\Psi_{SVM} - \Psi_{W-RVM}$) as a function of the number of vectors used. It can be seen that for a given accuracy more Wavelet Approximated Set Vectors are needed to approximate the SVM than for the RVM. However, as shown on the second plot, for a given computational load, the

W-RVM rejects much more non-face patches from the validation set I than the RVM. This explains the improved runtime performances of the W-RVM. Additionally, it can be seen that the curve is smoother for the W-RVM, hence a better trade-off between accuracy and speed can be obtained by the W-RVM.

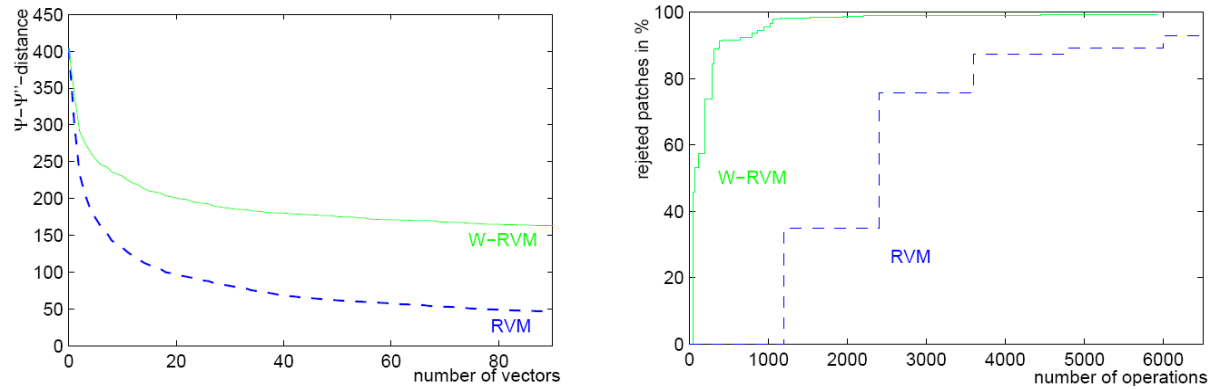


Figure 3-8: Distance of the hyper-planes and rejections over number of operations
Left: Hyper-plane discrepancy as function of the number of vectors for the RVM (*dashed line*), and the W-RVM (*solid line*). *Right:* Percentage of rejected non-face patches as a function of the number of operations required.

The improved runtime performances of the W-RVM compared to the RVM is convincingly evidenced in Figure 3-9. Here we compare the percentage of rejected non-faces of the W-RVM (plain lines) and RVM (dashed lines) over number of operations required only for the patches left at each set vector.

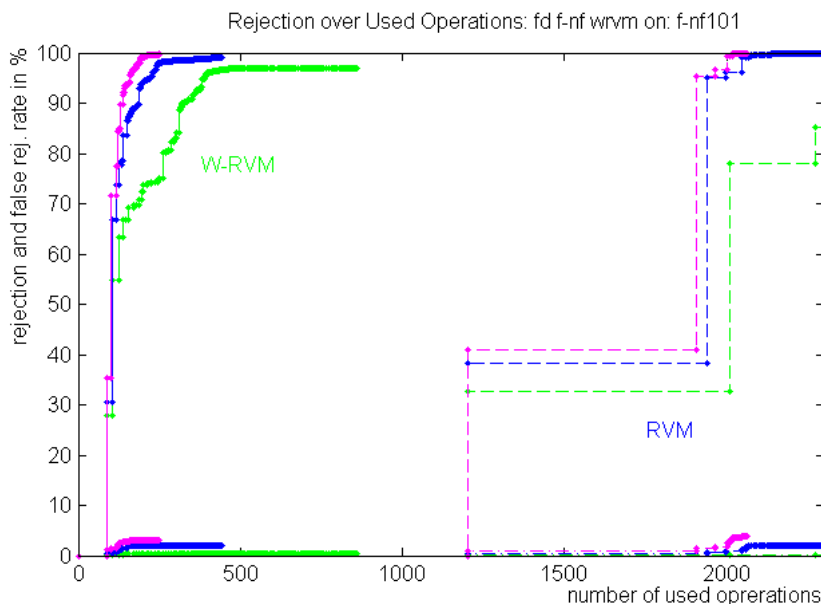


Figure 3-9: Percentage of rejection over number of operations
 Percentage of rejected non-face patches as a function of the number of operations required only for the patches left at each set vector.

Using higher thresholds for the RVM, b_j in (2.3) and the W-RVM, b_i' in (2.7) we obtain a higher rejection rate but also more false rejections (see Section 2.7 for the adjustment of the

trade-off between the False Acceptance and False Rejection Rate). Hence, we have to compare the rejection rate by the same FRR for the W-RVM and RVM (see dash-dotted lines). We plotted the rejection rate for three different threshold sets (green, blue and pink lines) for the RVM and W-RVM. For a comparable FFR the W-RVM needs for the same percentage of rejections 10 to 20 times fewer operations compared to the RVM approach. This drastic decrease of operations yields a very efficient detection method by an early rejection of large parts of the images as seen by the detections experiments or on the example in Figure 3-11.

Figure 3-10 shows the R.O.C.'s, computed on the validation set I for the SVM (dotted line), the RVM (dashed line) and the W-RVM (plan line). It can be seen that the accuracies of the three classifiers are similar without (left plot) and almost equal with the final SVM classification stage (see step 5. of the evaluation algorithm in Table 2-2) for the remaining patches (right plot).

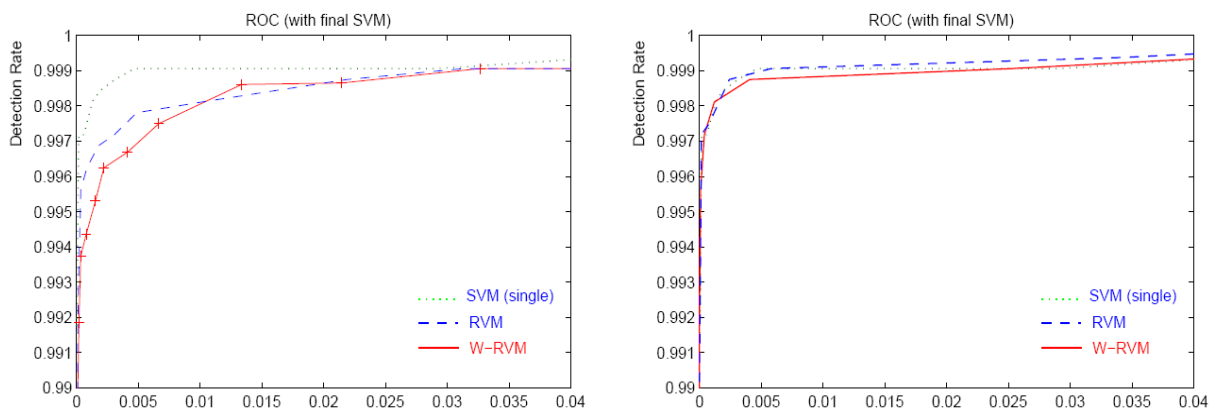


Figure 3-10: R.O.C.'s for the SVM, the RVM and the W-RVM

R.O.C.'s for the SVM, the RVM and the W-RVM (*left*) without and (*right*) with the final SVM classification for the remaining patches. The FAR is related to the number of non-face patches.

Table 3-1 compares the accuracy and the average time required to evaluate the patches of the validation set I. The speed-up over the former approach [70] is about a factor 2.5 ($3.85\mu s$). The novel W-RVM algorithms provides a significant speed-up (530-fold over the SVM and more than 15-fold over the RVM), for no substantial loss of accuracy.

method	FRR	FAR	time per patch
SVM	1.4%	0.002%	$787.34\mu s$
RVM	1.5%	0.001%	$22.51\mu s$
W-RVM	1.4%	0.002%	$1.48\mu s$

Table 3-1: Comparison of the efficiency of the approaches

Comparison of accuracy and speed improvement of the W-RVM to the RVM and SVM.

Figure 3-11 shows an example for applying the trained W-RVM classifier for face detection. The algorithm summarised in Table 2-2 is applied on each pixel location of the image using a sliding-window method on the image pyramid (Figure 2-2). The example demonstrates the fast rejection of large image areas. By increasing the stages of the cascade fewer and fewer patches have to be evaluated. After the last W-RSV, only five image locations have to be classified using the full SVM. For large areas of the image, the evaluation already stops after incorporation of only a few W-RSV's.

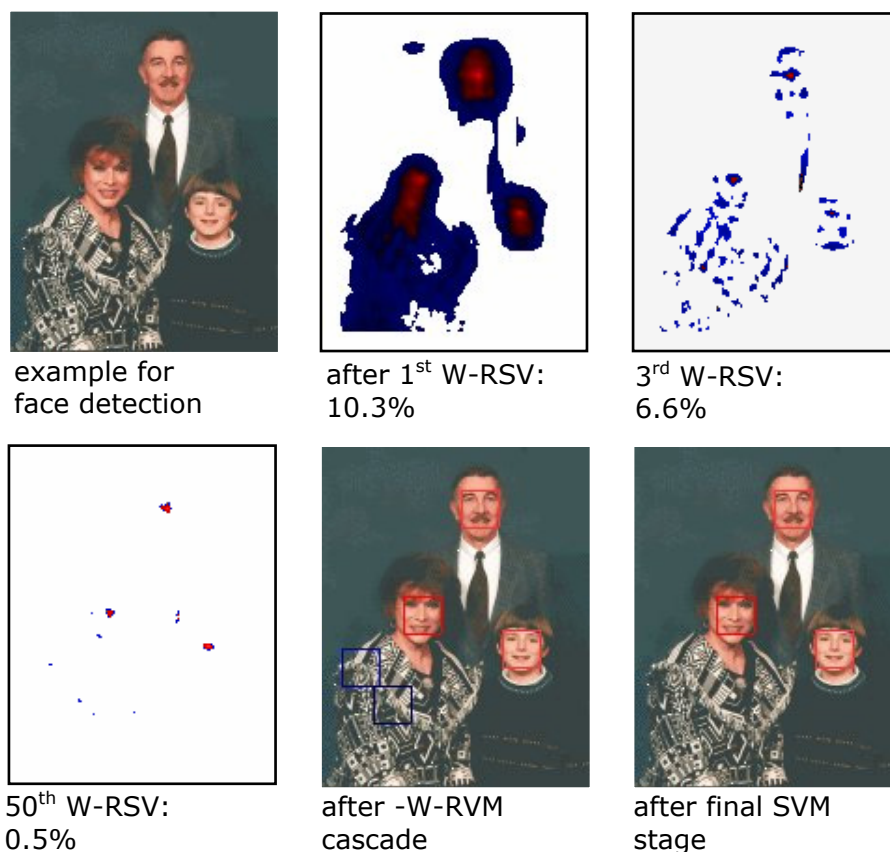


Figure 3-11: Example demonstrating fast rejection of large image areas

From top-left to bottom-right: Input image from MIT-CMU database [93] followed by images showing the amount of rejected locations at the 1st, 3rd and 50th stages of the cascade. The white pixels are rejected patch centre and the blue-to-red colour gradient of a pixel is proportional to the output of the W-RVM evaluation. The penultimate image shows a box around the pixels alive at the end of the W-RVM cascade and the last image, after the full SVM is applied.

The validation set II contains 500 frontal and half profile images from the FERET database [61]. We compared our approach with the Viola & Jones method [102] implemented in OpenCV (version b5a). The Viola & Jones detector yields on set II a detection rate of 90.9% by 0.32 false acceptances (FA) and 0.29 sec per image (on a Pentium M Centrino 1600 CPU). Compared to the results given in [102] the processing time is slower since the image size of the FERET images is larger. The results on FERET are more accurate because of the higher quality of the images. With the W-RVM we obtained on the same PC and set II a detection rate of 90.1% by 0.25 FA and 0.15 sec processing time per image.

Our proposed classifier is more efficient at detection, but mainly at training time than the AdaBoost method [102] and classifies about 25 times faster than the Rowley-Baluja-Kanade detector [79] and about 1000 times faster than the Schneiderman-Kanade detector [85].

We also proved the performance and detection accuracy under real-life conditions, e.g. during the I-Search project [54] and in several other applications (see Chapter 4). With the webcam application FaFaDe (see Section 4.1.2) we obtained accurate face detection at real time by 25 fps (on a Intel Pentium M Centrino 1600 CPU, at a resolution of 320×240 , step size 1 pixel, 5 scales).

3.3. Applying W-RVM for Facial Feature Classifiers

Now we want to apply the W-RVM and the developed detector principles for a facial feature detection system. First, we want discuss which facial points we want to choose on a catalogue of criteria. Then we will train the W-RVM classifiers on the variety of synthetic training and validation sets.

3.3.1. Multi Criteria Evaluation of Optimal Facial Features

We want to choose facial feature points, which can be detected with a high detection rate. The results should be comparable with other approaches, common in computer vision and medicine so that they can be used by many applications. The feature points should also allow an automatic fitting of the 3D MM. These are only some of the criteria we want to consider, but already they are partly contradictory.

To find a multi criteria optimal set of feature points we start by two for computer vision and computer science common sets (Figure 3-12). The first set is the MPEG-4 standard [72], defined in the norms "ISO/IEC 19794-5:2005 for Face Image Data" and "ISO/IEC 14496-2:2001(E) for Coding of Audio-visual Objects in Annex C" from the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). The MPEG-4 standard is also used by the US face image standard ANSI/INCITS 386-2004. The MPEG-4 standard is important because it is used by many applications to store or transfer facial data, or to animate facial expressions of characters. The second set are the Anthropology Landmarks, also know as Farkas points [28] obtained from anthropologists and forensic medical experts.

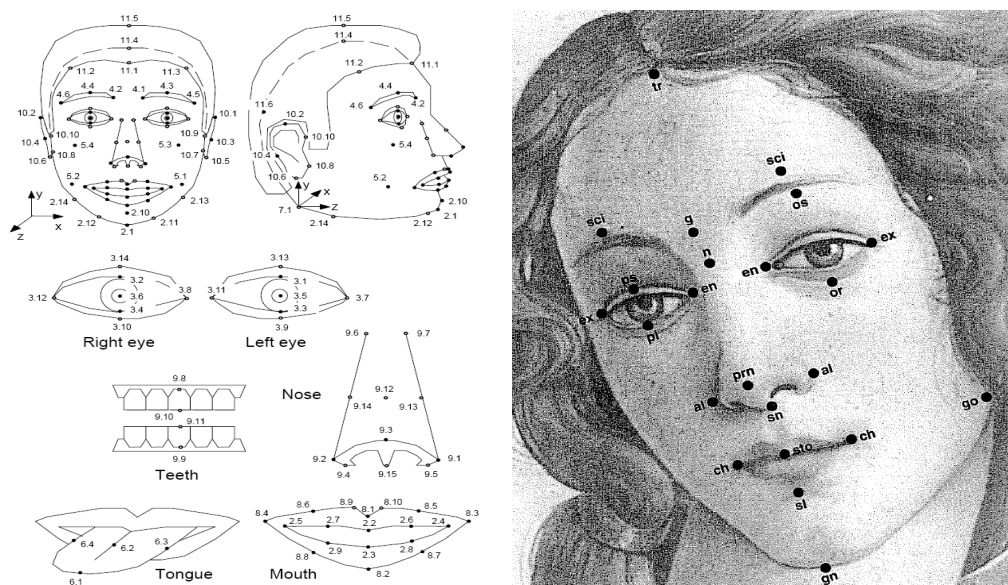


Figure 3-12: Standard facial features in computer vision and medicine
 The MPEG-4 ISO norm (*left*) [72], and Anthropology landmarks (*right*) from Farkas [28].

In medicine, some points are used as references, because they are invariant and can reference facial points by the Canonical Anthropometric Coordinate System (“Frankfurter Horizontal” (FH)). In the Canonical Anthropometric coordinates the point of origin O coincides with the pronasale (prn) anthropometric landmark, which is merely the nose endpoint. Z-axis is formed by the intersection of the Frankfurt Horizon (FH) and the vertical symmetry plane and is oriented in the direction of the face sight. Y-axis is the normal of FH going through prn oriented upwards. X-is oriented in a way to form a standard orthogonal Cartesian coordinate system OXYZ with the other two axes [28]. In the comparative anatomy often used to reference facial points or to measure the "facial angle" are the Camper plane² and the Occlusion plane (Figure 3-13).

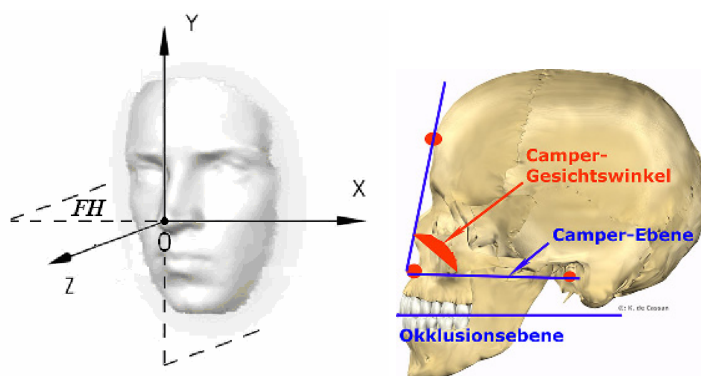


Figure 3-13: Medical criteria for the evaluation of optimal facial features

² Peter Camper is known for his theory of the "facial angle" in connection with intelligence. This "facial angle" was abused by scientific racism theories, by claiming that antique Greco-Roman statues presented an angle of 90°, Europeans of 80°, Black people of 70° and the orangutan of 58° [97].

Our goal is to use the facial feature points for automatic anchor-point detection used to initialise the 3D MM fitting (see Section 5.1.2). In Figure 3-14 we show anchor-points which turned out for best-fitting results. For contour points the W-RVM detector will not be the appropriate method, because template-based approaches i) cannot detect feature, where large parts of the area are unknown background and ii) the position on the contour is not fix defined, hence contour-based methods are better suitable. We did experiments to rank the Anthropology landmarks concerning the fitting criteria. The feature points should be useful for other applications, like for pose estimation and accurate with respect to manual labelling.



Figure 3-14: Anchor-points optimal as initialisation for the 3D MM fitting

Criteria for the facial feature points are also a high reliability of finding the points, they feature should have high saliency in a certain neighbourhood, so that they are not confound with other features, and have a high estimability, by using locations of other feature points. Feature points taking in account above criteria should yield a high detection rate. Heisele et al. proposed an automatic strategy to find optimal components by learning relevant features from sets generated by the 3D MM. It starts with a set of small seed regions. These regions are translated and grown by minimising a bound on the expected error probability of an SVM (see e.g. in Figure 3-15). The advantage is that no manual interaction is required for choosing and extracting the components [41], [42], [43].

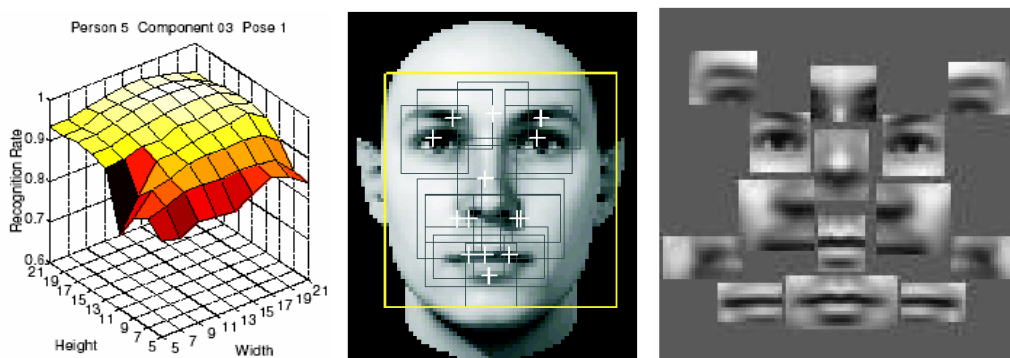


Figure 3-15: Optimisation of the position and size of facial features by Heisele et al. For instance, as seen in the *left image*, the dimensions of the feature "centre of mouth" are adjusted to a maximal recognition rate. This yields a set of optimal features, used for component-based face detection (*right*).

Important for the selection of the facial features is also invariance of the feature positions w.r.t. expression and aging and of the areas used to detect the feature point, w.r.t. scale, pose, and illumination, e.g. if the features are visible on a large range of pose variations. Here we take advantage again from Heisele et al. (see Table 3-2); he shows a ranking of the features in assumption with Farkas landmarks according to the standard deviation of the recognition rate across pose and according to the standard deviation across identity.

Comp	CVRate	Comp	σ_{Pose}	Comp	σ_{ID}
1, 2 (p)	0.925	6 (prn)	0.009	14 (pg)	0.018
14 (pg)	0.915	11, 12 (sci)	0.025	4, 5 (ch)	0.021
11, 12 (sci) ³	0.905	7, 8 (al)	0.028	11, 12 (sci)	0.021
4, 5 (ch)	0.892	3 (sto)	0.031	13 (g)	0.023
3 (sto)	0.888	14 (pg)	0.032	9, 10 (?)	0.023
9, 10 (?) ⁴	0.877	1, 2 (p)	0.033	1, 2 (p)	0.027
13 (g)	0.864	13 (g)	0.035	7, 8 (al)	0.030
7, 8 (al)	0.846	4, 5 (ch, ch)	0.047	3 (sto)	0.042
6 (prn)	0.796	9, 10 (?)	0.061	6 (prn)	0.075

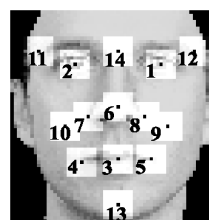


Table 3-2: Optimal features in the scenes of recognition rate and invariance

The table *left* shows the recognition rate via cross validation (*CVRate*) of the optimal components (*right*) in assumption with Farkas landmarks and invariance w.r.t. pose and subjects [44].

From all above discussed criteria we build a multi-criteria catalogue (see Table 3-3) and estimated a ranking (see e.g. Table 3-4 and complete ranking in Table C-4) of the set of features defined by Farkas landmarks and the MPEG-4 standard (see Figure 3-16).

Criteria for evaluation of optimal facial features:

- a) Invariance of position w.r.t. expression and aging
- b) Invariance of the area used to detect the feature point, w.r.t. scale, pose, and illumination
- c) Visibility
- d) Reliability of finding the point
- e) Saliency (in a certain neighbourhood)
- f) Estimability (using locations of other feature points)
- g) ISO compatibility
- h) Accuracy of manual labelling
- i) Usefulness for pose estimation
- j) Usefulness for the MM fitting

Table 3-3: Criteria for evaluation of optimal facial features

³ no Farkas anagoges, for MPEG4 4.5, 4.6, sci nearest Farkas landmark

⁴ no Farkas anagoges, for MPEG4 5.3, 5.4

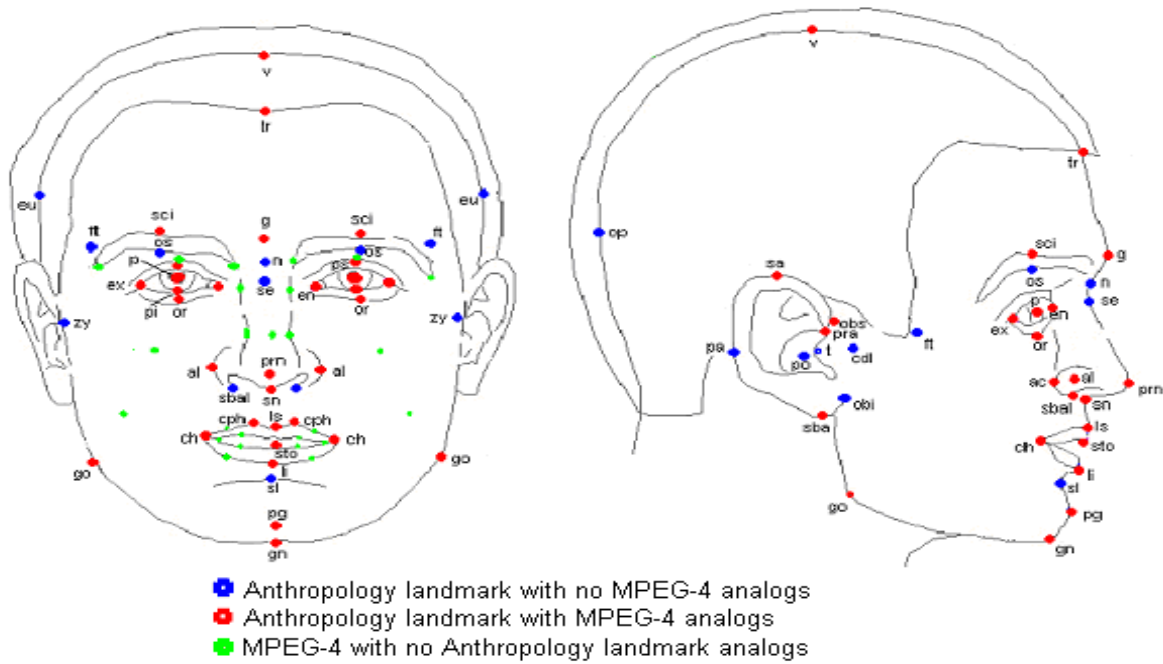


Figure 3-16: Definition of MPEG-4 and Anthropology landmarks
 Definition of the MPEG-4 ISO norm [72] and Anthropology landmarks from Farkas [28] as used in the first and second column in Table 3-4, 3-5, and C-4.

Point ID	MPEG4	Anthropometric point name	1) Invariance of position wrt. to expression, subj. and aging	2) Invariance of the area wrt. scale, pose and illumination	3) Visibility (as percentage of images) 5 all, 4 3/4, 3 1/2, 2 1/4, 1 0	4) Reliability of finding the point (depends on the classifier)	5) Saliency (in a certain neighbourhood)	6) Estimability (using locations of other feature points)	7) ISO compatibility (point listed in MPEG4 ISO standard)	8) Accuracy of manual labelling	9) Usefulness for pose estimation	10) Usefulness for the MM fitting.	Average	Resume-Rank for FFD ⁶	Comment	How to point
p	3.5.3.6	Center point of pupil	5	4	4	5	5	5	4	5	4	4	4,5	1	label available!	Is determined when the head is in the rest position and the eye is looking straight forward
prn	9.3	Pronasale	2	5	5	2	3	5	4	2	5	5	3,8	1	label available!	The most protruded point of the apex nasi
ch	8.3.8.4	Cheilion	2	2	3	4	5	4	4	5	4	5	3,8	1	label available!	The point located at each labial commissure
ex	3.7.3.12	Exocanthion (or ectocanthion)	4	4	4	4	4	4	4	5	5	5	4,3	2	label available!	The point at the outer commissure of the eye fissure
ls	8.1	Labiale (or labrale) superius	3	4	4	4	4	5	4	5	4	5	4,2	2		The midpoint of the upper vermillion line
sbal		Subalare	4	4	3	4	3	4	2	5	3	5	3,7	2		The point at the lower limit of each alar base, where the alar base disappears into the skin of the upper lip

Table 3-4: Part of the evaluation table for evaluation of optimal facial features
 The complete evaluation is seen in Table C-4. Even if *prn* and *ch* have smaller average values, they have got the highest rank, because for these feature points are ground truth as labels available, moreover *prn* is essential for pose estimation and *ch* for the initialisation of the MM-fitting.

As resume, we evaluated an average of all criteria and made a ranking over the catalogue, evaluating which facial features we want to train first, see Table 3-5. The first five features are *le*, *re*, *nt*, *lm*, and *rm*. It turned out by the training of the facial feature detection system

⁵ see: Heisele et al. [44], [41], [42], [43]

⁶ 1: first, 2: second,... 5: last

(Section 3.5) that more facial features are needed. We chose the features lx, rx, ls, lb, and rb using the ranking of our multi-criteria catalogue. Taking advantage from a further unification of the W-RVM detectors and 3D MM we want to apply detectors also e.g. for sci, g, li, en, and sba. However, the detection can be applied in areas with reduced size using rather small uncertainty areas obtained by the PSM (see Section 3.6 and uncertainty areas in Figure 3-35).

Anthropometric (Farkas) point	MPEG-4	Average over all criteria	Ranking	English description	Abbreviation in the thesis	Used colour in the thesis	Used maker in the thesis	How to point
p (centre point of pupil) ⁷	3.5	4,5	1	left eye	le	red	'x'	Is determined when the head is in the rest position and the eye is looking straight forward
ditto	3.6	4,5	1	right eye	re	blue	'+'	ditto
prn (pronasale)	9.3	4.0	1	nose tip	nt	green	'v'	The most protruded point of the apex nose
ch (cheilion)	8.3	3.8	1	left mouth corner	lm	yellow	'<'	The point located at each labial commissure
ditto	8.4	3.8	1	right mouth corner	rm	magenta	'>'	ditto
ex (exocanthion)	3.12	4.3	2	left eye corner	lx	orange	'└'	The point at the outer commissure of the eye fissure
ditto	3.7	4.3	2	right eye corner	rx	cyan	'┘'	ditto
ls (labiale superius)	8.1	4.2	2	upper lip point	ls	beige	'┐'	The midpoint of the upper vermilion line
sbal (subalare)	-	3.7	2	left nose corner	lb	brown	'^'	The point at the lower limit of each alar base, where the alar base disappears into the skin of the upper lip
ditto	-	3.7	2	right nose corner	rb	lemon	'┘'	ditto

Table 3-5: Facial feature points chosen for the thesis

As result of the multi-criteria evaluation, we chose first the *top five* and later additional the *bottom five* points as facial features.

3.3.2. Training of the W-RVM's for Facial Features

After choosing the optimal facial features in the previous section we trained the first three W-RVM classifiers (one for the left and right eye (le, re), one for the nose tip (nt), and one for the left and right mouth corner (lm, rm)). Later we added W-RVM classifiers for the eye corners (lx, rx), the upper lip point (ls), and nose corners (lb, rb).

⁷ We choose the average point of the left and right eye corners, what is similar to p if the eye is looking straight forward, as described for "how to point" p.

Because of the symmetry of some features, we can train only one of the classifiers and generate the second classifier by mirroring the W-RSV's (and SSV's for the final SVM stage). For instance, it is sufficient to train the left eye classifier and to use the mirrored version for the right eye.

We implemented an application for a full automatic training of the classifiers. For the generated training and validation sets, the training of the SVM, the RVM, and the W-RVM is processed. To verify the accuracy and efficiency of the W-RVM training the decrease of the distance to the SVM hyper-plane over number of operations is plotted in comparison to the RVM approach. Also the adjustment of the trade-off between FAR and FRR, controlled by the thresholds for the W-RSV's is automatically generated. For the validation of the accuracy of the trained classifiers, the R.O.C.'s are generated automatically for the SVM, RVM, and W-RVM stages. Additionally the evaluation of the percentage of rejections over the number of used operations to validate the efficiency of W-RVM's are generated. The application is detailed in Appendix B, C, and in the online help of the software packed).

For the training of the W-RVM's for the facial features, we used the identical approach as for the face classifier. The training algorithm in Table 2-1 and detailed in Section 2.6 is used. Therefore, we will concentrate here on describing the results of the training and validation.

Training of the SVM's for all Facial Features

We generated for all classifiers the training and validation set as described in Section 3.1 by using a mixture of original images and different classes of synthetic images. For pairwise existing features we can use the mirrored training and validation sets, e.g. for the right eye generated patches to train the left eye classifier. For a detailed description of the sets see Table B-1 in Appendix B.

For the training of the Support Vector Machines, we used a recursive grid search to optimise the kernel parameter and bound C . We used the open library libSVM as implementation.

We trained for all features two classifiers, one small classifier for first tests and one large classifier for real applications. A detailed description of the resulting SVM's, e.g. the number pos./neg. number of SSV's, the optimal kernel parameter and bound C , etc. can be seen in Table C-3.

The trained R.O.C.'s for all facial features are compared in Figure 3-17. The facial features are more ambiguous as the full face (black line), therefore the accuracy of the face classifier is the highest. Within the working range of $1e-2$ to $1e-1$ (1% to 10%) FAR the feature "nose corners" (lb) performs best. The most difficult to train feature is the nose tip (nt), because this feature depends much on pose and lighting conditions.

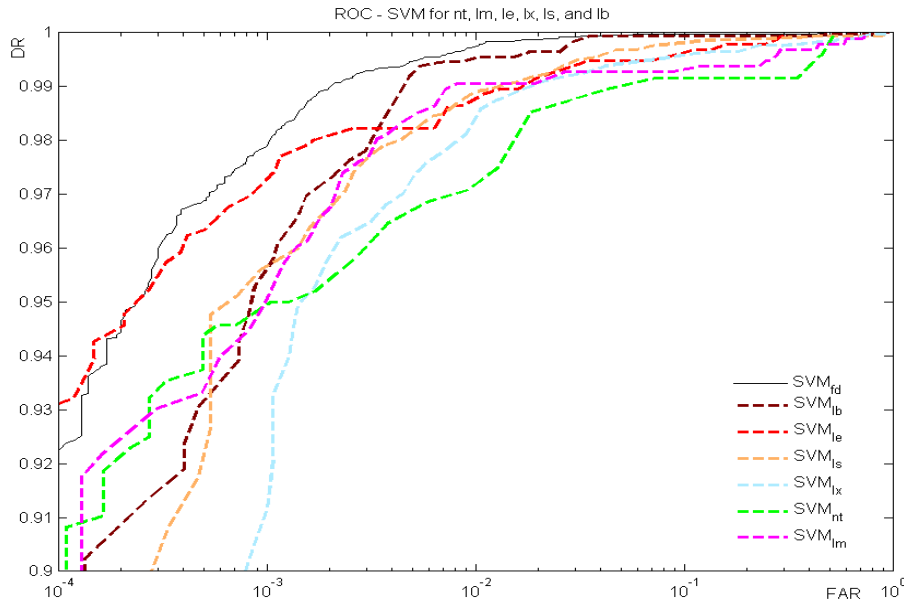


Figure 3-17: Trained Support Vector Machines for all features.

The trained R.O.C.'s for all facial features are compared. The facial features are more ambiguous than the full face (*black line*), therefore the accuracy of the face classifier is the highest. Within the working range of 10^{-2} to 10^{-1} (1 to 10%) FAR nose corners (*lb*) perform best. The most difficult to train feature is the nose tip (*nt*), because this feature depends much on pose and lighting conditions.

Training of the W-RVM's for all Facial Features

After the training of the SVM's we generate the RVM's and the application is preceding the training of the W-RVM's as detailed in Table 2-1 and described in Section 2.6

To verify the accuracy and efficiency of the W-RVM training the decrease of the distance to the SVM hyper-plane ($\|\Psi_{SVM} - \Psi_{W-RVM}\|^2 / \|\Psi_{SVM}\|^2$) over number of operations is plotted in comparison to the RVM approach ($\|\Psi_{SVM} - \Psi_{RVM}\|^2 / \|\Psi_{SVM}\|^2$).

As example, we show in Figure 3-18 for the facial feature "upper lip point" (*ls*) the decrease of the distance to the SVM decision hyper-plane over number of used operations during the training process. The W-RVM (green) uses significant fewer operations as the RVM (blue) for the same decrease of the distance to the SVM hyper-plane. Most rejections are done by the first approximation level (label L1). Up until this approximation level, the W-RVM needs 10-fold fewer operations as the RVM. That means we gain a speed-up factor about one magnitude for efficiency by the approximation the SVM decision hyper-plane. This theoretical improvement must be verified for the classification on validation sets.

W-RVM's Stages for all Facial Features

As result of the W-RVM training, we obtain the W-RSV's and weights of the vectors (see Output of the W-RVM learning stage Table 2-1). Because of the Double Cascade, non-feature patches are rejected after incorporating each W-RSV's. The therefore used set of thresholds is

set for a given FRR using the R.O.C.'s to guaranty not to lose more features patches as adjusted. An array of threshold sets is automatically adjusted for given FRR's (see Section 2.7 and 2.8). So the trade-off between FAR and FRR can be calibrated by the user of front-end applications.

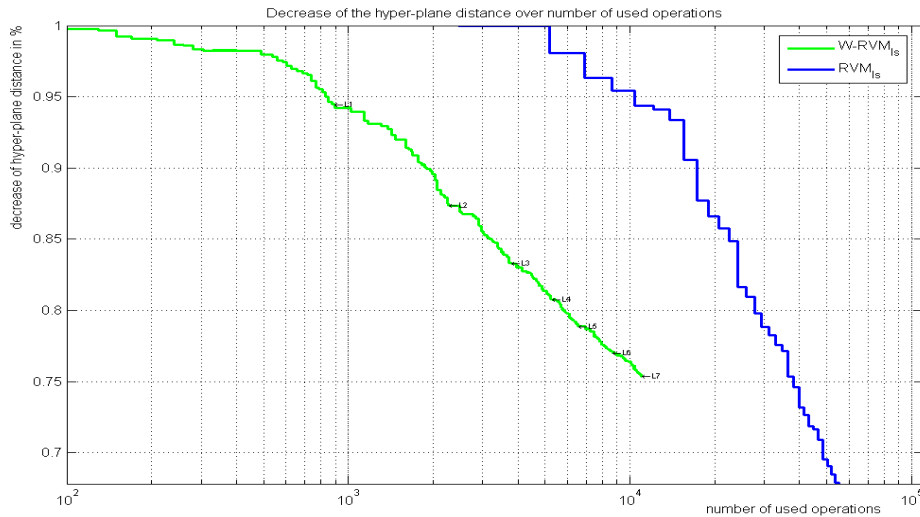


Figure 3-18: Decrease of the hyper-planes distance over number of used operations

These curves demonstrate on the example of the facial feature "upper lip point" (l_s) that the W-RVM (*green*) uses significant fewer operations as the RVM (*blue*) for the same decrease of the distance to the SVM hyper-plane. Most rejections are done by the first approximation level (label $L1$). Up until this approximation level, the W-RVM needs 10-fold fewer operations as the RVM.

All trained W-RVM's at this thesis are detailed in Appendix C, Table C-1, e.g. which training and validation set was used, which SVM and RVM, and all training parameters.

Each W-RVM classifier contains the fast W-RVM stage and a final complex SVM for the remaining patches after applying the Double Cascade (see Table 2-2). In Figure 3-19, we show on the example of the mouth corner (l_m , magenta lines) the R.O.C.'s of the trained W-RVM stages. The R.O.C.'s of the SVM's of the facial features (e.g. the magenta plan line for the SVM_{l_m}) are not as good as from the SVM of the face classifier (black plan line). One reason is that the facial features are more ambiguous regarding the local image representation and the other reason is that we used for the training and validation of the features more complex data by taking advantage of generating synthetic sets with a large verity concerning pose, lighting, noise, etc. conditions. Therefore, the hypothesis space to be learned by the SVM is more complex.

For the same detection rate (respectively same FRR) the FAR of the first W-RVM stage for the features (e.g. dotted line in Figure 3-19) is higher than the FAR of the single SVM (plan magenta line). However, this stage rejects many non-feature points by few operations. With the final full SVM stage, the W-RVM (dashed line) gains the same classification accuracy as the single SVM within the working range of $1e-2$ to $1e-1$ (1% to 10%) FAR.

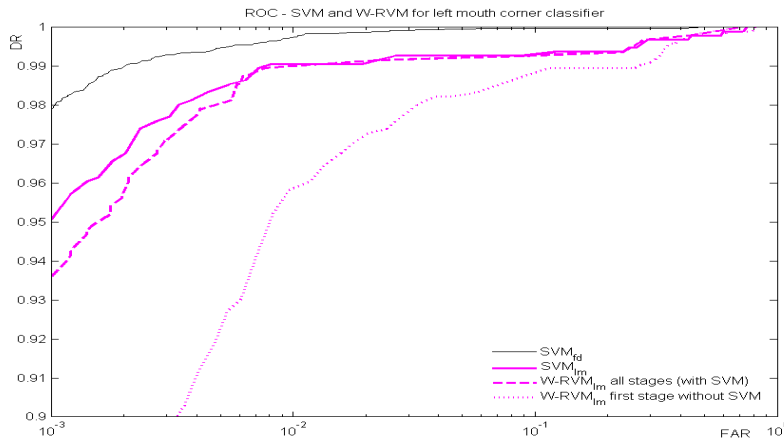


Figure 3-19: Training stages of the W-RVM for the mouth corner
 The R.O.C.'s for the trained W-RVM stages for the mouth corner (*lm*, *magenta*) are shown. For the same detection rate (respectively same FRR) the FAR of the first W-RVM stage (*dotted magenta line*) is higher than the FAR of the single SVM (*plan magenta*). However, this stage rejects many non-feature points by few operations and with the final full SVM stage the W-RVM (*dashed line*) gains the same classification accuracy as the single SVM within the working range of 10^{-2} to 10^{-1} (1 to 10%) FAR.

Verifying the Accuracy and Efficiency of the W-RVM's

The R.O.C.'s of the SVM's of the facial features (e.g. the *magenta plan line* for the SVM_{lm} in Figure 3-19) are not as good as of the SVM of the face classifier (black *plan line*) as discussed above. For the same reason also the R.O.C.'s of the final W-RVM's in Figure 3-20, dashed lines cannot achieve the performance of the face classifier (*plan line*).

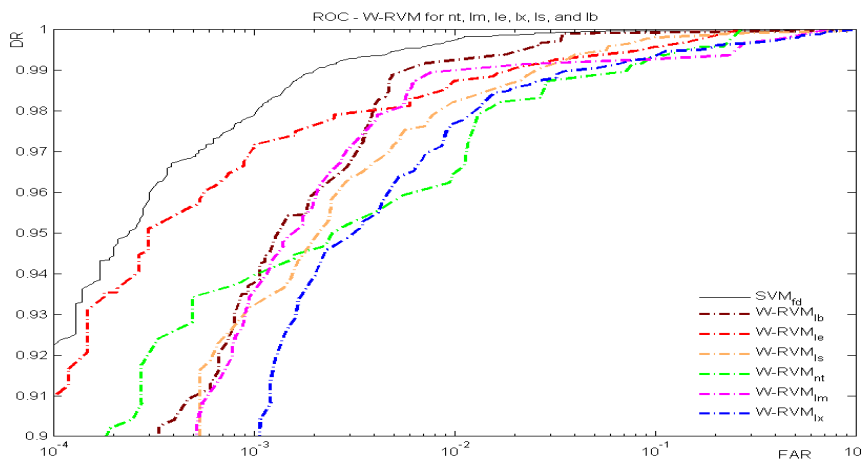


Figure 3-20: Trained W-RVM's for all facial features
 The trained R.O.C.'s of the W-RVM's for all facial features are compared. The facial features are more ambiguous than the full face (*black line*), therefore the accuracy of the face W-RVM classifier is the highest. Within the working range of 10^{-2} to 10^{-1} (1 to 10%) FAR the nose corners (*lb*) perform best. The most difficult to train feature is the nose tip (*nt*), because this feature depends much on pose and lighting conditions.

In Section 3.5, we introduce a cascaded detection framework to overcome the higher FAR of the facial feature classifiers. The FAR is reduced by a correlation classifier applied in Section 3.6.

The R.O.C.'s of the final trained W-RVM's for all facial features are compared in Figure 3-20. Within the working range of $1e-2$ to $1e-1$ (1% to 10%) FAR the nose corners (lb) perform best. The most difficult to train feature is the nose tip (nt), because this feature depends much on pose and lighting conditions. Hence, the to-be-learned hypothesis space of the nose tip is more complex.

The improved runtime performances of the W-RVM's compared to the RVM's is convincingly evidenced for the upper lip feature in Figure 3-21. Here we compare the percentage of rejected non-features of the W-RVM (plain lines) and the RVM (dashed lines) over the number of operations required for the patches left at each step of the cascade. We compare the rejection rate (RR, lines with marker ' \bullet ') for two threshold sets (red plots have a lower FRR as the green plots). To compare the RR a similar FRR (lines with marker '+') for the W-RVM (dotted lines) and RVM (dash-dotted) have to be used. The W-RVM_{Is} needs for the same rejection rate (plan lines) factor 10 to 20 fewer operations as the RVM_{Is} (dashed lines). The speed-up over the original Support Vector Machine approach is about the factor 500.

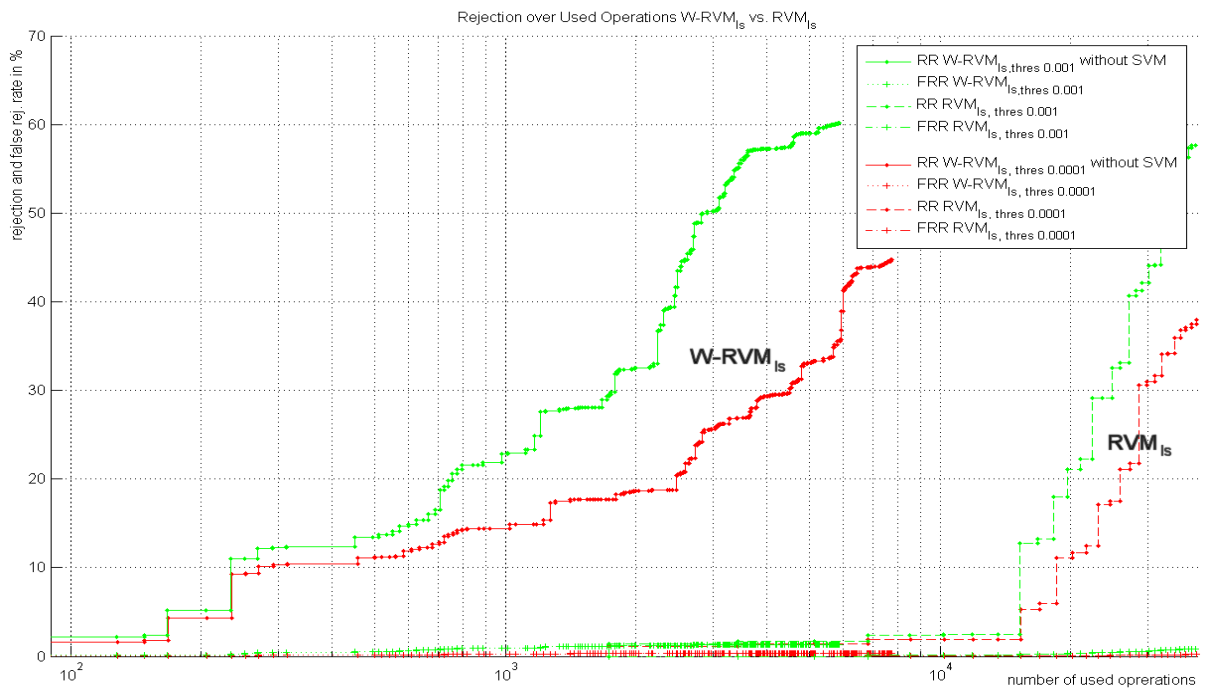


Figure 3-21: Rejection rate of the W-RVM_{Is} and RVM_{Is} over number of used operations. The Percentage of rejected non-features as a function of the number of operations is shown for two threshold sets (red with lower FRR as green lines). Only the operations required for the patches left at each set vector are considered. The W-RVM_{Is} needs for the same rejection rate (RR, plain lines, marker ' \bullet ') factor 10 to 20 fewer operations as the RVM_{Is} (dashed lines, marker ' \bullet ') by a comparable FAR's (dotted and dash-dotted lines, marker '+'). This yields a speed-up compared to the original SVM approach by a factor about 500.

3.4. Probabilistic W-RVM Classifier

The evaluation function of an SVM and a W-RVM computes for a data point the distance to the decision hyper-plane. The classification is obtained by applying the signum function on the distance (e.g. (2.1) for the SVM). The distance to the hyper-plane can be used as certainty for the classification (a high absolute value indicates a certain decision). However, for post-processing and several applications it is more suitable to use a probability instead of a not calibrated distance measure. For instance, probabilistic output is required to compare the output of different classifiers, by the PSM in Section 3.6 to find the final facial feature assortment, or for the Condensation tracking in Section 4.3.1 and 5.2.4. Standard SVM's and our W-RVM from Chapter 2 do not provide such probabilities. We introduce some techniques to obtain a posterior probability output and obtain best results for our Probabilistic W-RVM by a Sigmoid Fitting function.

We compute the a-posteriori probability $p_{\text{W-RVM}_{ffp}}(\mathbf{x}_{ffp} | t_{\text{W-RVM}_{ffp}})$ (or shorter $p_{ffp}(\mathbf{x}_{ffp} | t_{ffp})$) of the W-RVM outputs of the feature detectors, where $p_{ffp}(\mathbf{x}_{ffp} | t_{ffp})$ is the probability that the image position \mathbf{x}_{ffp} is a valid feature point ffp given the output t_{ffp} of detector W-RVM _{ffp} of feature ffp . We compute $p_{ffp}(\mathbf{x}_{ffp} | t_{ffp})$ by

$$p_{ffp}(\mathbf{x}_{ffp} | t_{ffp}) = \frac{p(t_{ffp} | \mathbf{x}_{ffp})p(\mathbf{x}_{ffp})}{p(t_{ffp} | \mathbf{x}_{ffp})p(\mathbf{x}_{ffp}) + p(t_{ffp} | \sim \mathbf{x}_{ffp})p(\sim \mathbf{x}_{ffp})} \quad (3.1)$$

where $p(\mathbf{x}_{ffp})$ is the prior that the position is a correct ($p(\sim \mathbf{x}_{ffp})$ not correct) detection for the feature ffp (i.e. the distance to a label is smaller (larger) as a given threshold, or the ratio of the number of positive (negative) to the number of all examples in a validation set).

3.4.1. Variants of Non-parametric Techniques for PDF Estimation

For the likelihood $p_{ffp}(t_{ffp} | \mathbf{x}_{ffp})$ of the W-RVM output t_{ffp} w.r.t. \mathbf{x}_{ffp} (likelihood that the classifier W-RVM for the facial feature point ffp produces the output t , given \mathbf{x} is an image position of the feature point ffp), we estimate the density function (PDF) by different non-parametric techniques and discuss the results. First, we used histogram methods as seen in Figure 3-22.

The improvement of the selection of the final feature assortments using histogram methods for the PDF of the W-RVM outputs was not significant. The PDF estimation by non-parametric histogram methods was not sufficiently accurate.

To improve the PDF estimation for the likelihood $p_{ffp}(t_{ffp} | \mathbf{x}_{ffp})$ of the W-RVM output t_{ffp} w.r.t. \mathbf{x}_{ffp} we tested then a parzen windows method.

$$p_{ffp}(t_{ffp} | x_{ffp}) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{t_{ffp} - t_{ffp_i}}{h}\right), \quad (3.2)$$

where N is the number of examples in a validation set. As kernel function we tried a smooth kernel $K(z) = 1/\sqrt{2\pi} \exp(-1/2 z^2)$.

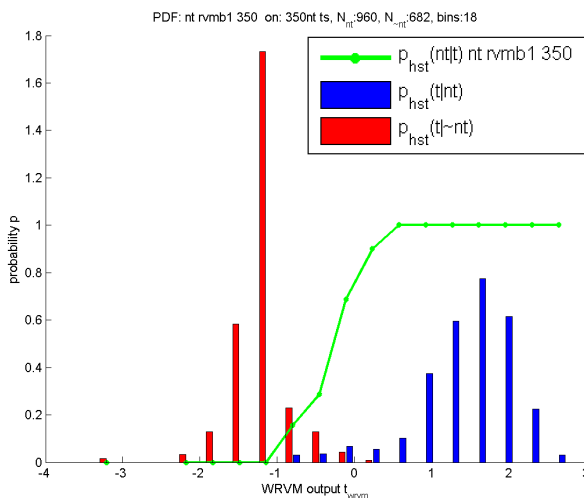


Figure 3-22: PDF estimation by non-parametric histogram method
On the example of the nose tip feature, *nt* the histogram of the classifier output for the negative examples (red), the positive (blue), and the estimated PDF (green) is shown.

However, similar to the size of the bins (used by the histogram technique), the parzen-window method is sensitive to the size of h (see Figure 3-23).

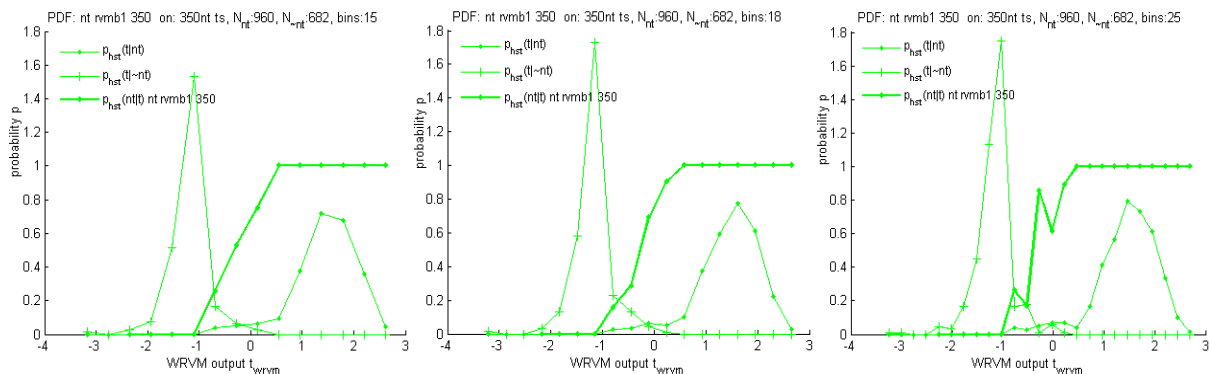


Figure 3-23: Histogram method is sensitive to the size of the bins
On the example of the nose-tip feature, it is seen that the size of the bins is sensitive. Using 18 bins (middle) is suitable, but e.g. 15 bins are too few (left) and 25 too many (right).

The choice of the bandwidth h is still critical, since we tested also k-NN estimation, where the PDF is estimated from N training samples by centring an interval $d(t_{ffp})$ around t_{ffp} and let

it grow until it captures k examples. So we evaluate the prior by $p(t_{ffp}) = k / (Nd(t_{ffp}))$, the density with $p(t_{ffp} | \mathbf{x}_{ffp}) = k_{\mathbf{x}_{ffp}} / (n_{\mathbf{x}_{ffp}} d(t_{ffp}))$ and the evidence with $p(\mathbf{x}_{ffp}) = n_{\mathbf{x}_{ffp}} / N$, where $k_{\mathbf{x}_{ffp}}$ is the number of samples in k with correct position \mathbf{x}_{ffp} for the feature ffp (and respective $k_{\sim \mathbf{x}_{ffp}}$ the number of not correct position, so that $k = k_{\mathbf{x}_{ffp}} + k_{\sim \mathbf{x}_{ffp}}$) and $n_{\mathbf{x}_{ffp}}$ is the number of positive and $n_{\sim \mathbf{x}_{ffp}}$ the number of negative samples in N for ffp ($N = n_{\mathbf{x}_{ffp}} + n_{\sim \mathbf{x}_{ffp}}$). Now we can evaluate the a-posteriori probability (3.1) with $p_{ffp}(\mathbf{x}_{ffp} | t_{ffp}) = k_{\mathbf{x}_{ffp}} / k$, since

$$p_{ffp}(\mathbf{x}_{ffp} | t_{ffp}) = \frac{k_{\mathbf{x}_{ffp}} / (n_{\mathbf{x}_{ffp}} d) n_{\mathbf{x}_{ffp}} / N}{k_{\mathbf{x}_{ffp}} / (n_{\mathbf{x}_{ffp}} d) n_{\mathbf{x}_{ffp}} / N + k_{\sim \mathbf{x}_{ffp}} / (n_{\sim \mathbf{x}_{ffp}} d) n_{\sim \mathbf{x}_{ffp}} / N} = \frac{k_{\mathbf{x}_{ffp}} n_{\mathbf{x}_{ffp}} / (n_{\mathbf{x}_{ffp}} d N)}{k_{\mathbf{x}_{ffp}} n_{\mathbf{x}_{ffp}} / (n_{\mathbf{x}_{ffp}} d N) + k_{\sim \mathbf{x}_{ffp}} n_{\sim \mathbf{x}_{ffp}} / (n_{\sim \mathbf{x}_{ffp}} d N)}$$

$$= \frac{k_{\mathbf{x}_{ffp}} / (d N)}{k_{\mathbf{x}_{ffp}} / (d N) + k_{\sim \mathbf{x}_{ffp}} / (d N)} = \frac{k_{\mathbf{x}_{ffp}}}{k_{\mathbf{x}_{ffp}} + k_{\sim \mathbf{x}_{ffp}}} = \frac{k_{\mathbf{x}_{ffp}}}{k}$$

or shorter using $p_{ffp}(\mathbf{x}_{ffp} | t_{ffp}) = (p(t_{ffp} | \mathbf{x}_{ffp}) p(\mathbf{x}_{ffp})) / p(t_{ffp})$

$$p_{ffp}(\mathbf{x}_{ffp} | t_{ffp}) = \frac{k_{\mathbf{x}_{ffp}} / (n_{\mathbf{x}_{ffp}} d(t_{ffp})) n_{\mathbf{x}_{ffp}} / N}{k / (Nd(t_{ffp}))} = \frac{k_{\mathbf{x}_{ffp}} n_{\mathbf{x}_{ffp}} Nd(t_{ffp})}{n_{\mathbf{x}_{ffp}} d(t_{ffp}) N k} = \frac{k_{\mathbf{x}_{ffp}}}{k} \tag{3.3}$$

The choice of k is also critical and comparable to the bandwidth h by the parzen-window method, but simpler to compute and the results improved as seen in Figure 3-24.

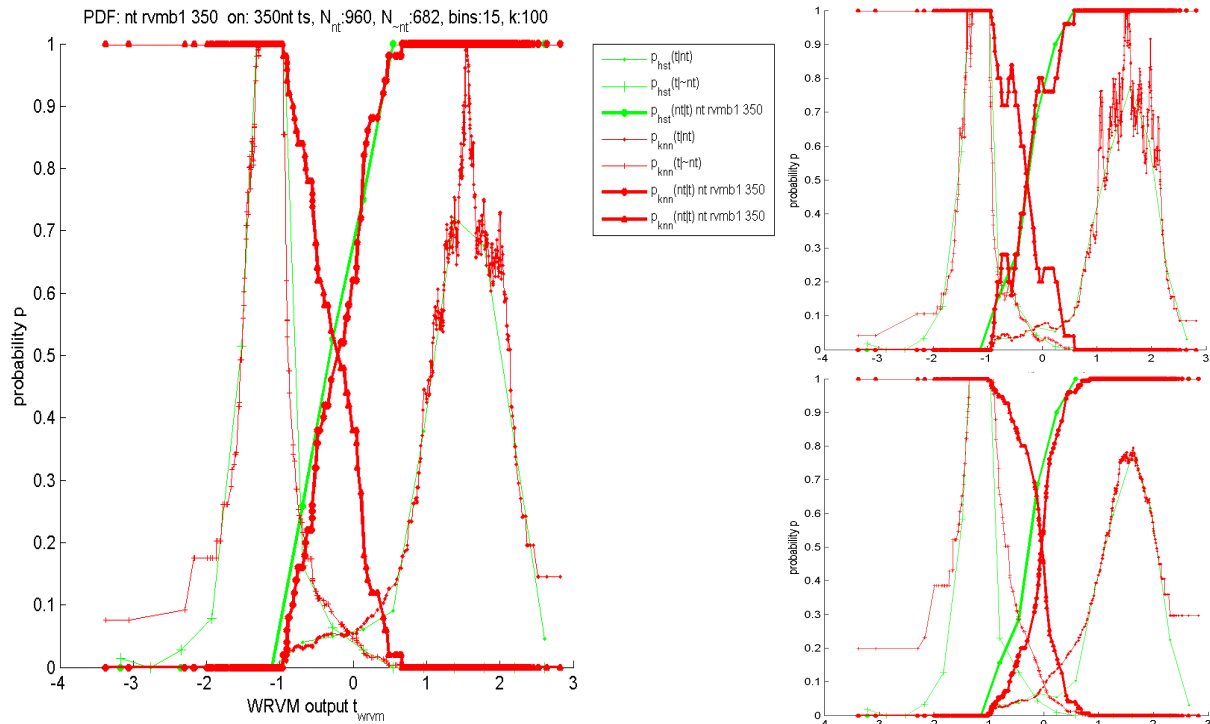


Figure 3-24: PDF estimation by non-parametric k-NN estimation
 The k-NN method is also sensitive to the size of k in (3.3). Using $k=100$ (left) is suitable, but e.g. 50 is too small (right top) and 200 too large (right bottom).

3.4.2. Probabilistic W-RVM using Sigmoid Fitting

The estimation of the PDF (class-conditional probability) using histogram, parzen-window, or k-NN methods is not stable enough. The best results we obtain using fitting a sigmoid function for the posterior probability.

The sigmoid function fitting [63] is a model-trust algorithm, based on the Levenberg-Marquardt algorithm [65]. The method for extracts probabilities from SVM outputs, which is useful for classification post-processing. The method adds a trainable post-processing step which is trained with regularised binomial maximum likelihood. A two-parameter sigmoid is chosen as the post-processing, since it matches the posterior that is empirically observed.

$$p_{ffp}(\mathbf{x}_{ffp} | t_{ffp}) = \frac{1}{1 + \exp(A t_{ffp} + B)} \tag{3.4}$$

The sigmoid fitting trains iterative the parameters A and B of the sigmoid function to map the W-RVM output into probabilities. In Figure 3-25, the iterative Fitting of the sigmoid function (blue lines) are visualised and can be compared with the histogram (green) and k-NN (red) techniques.

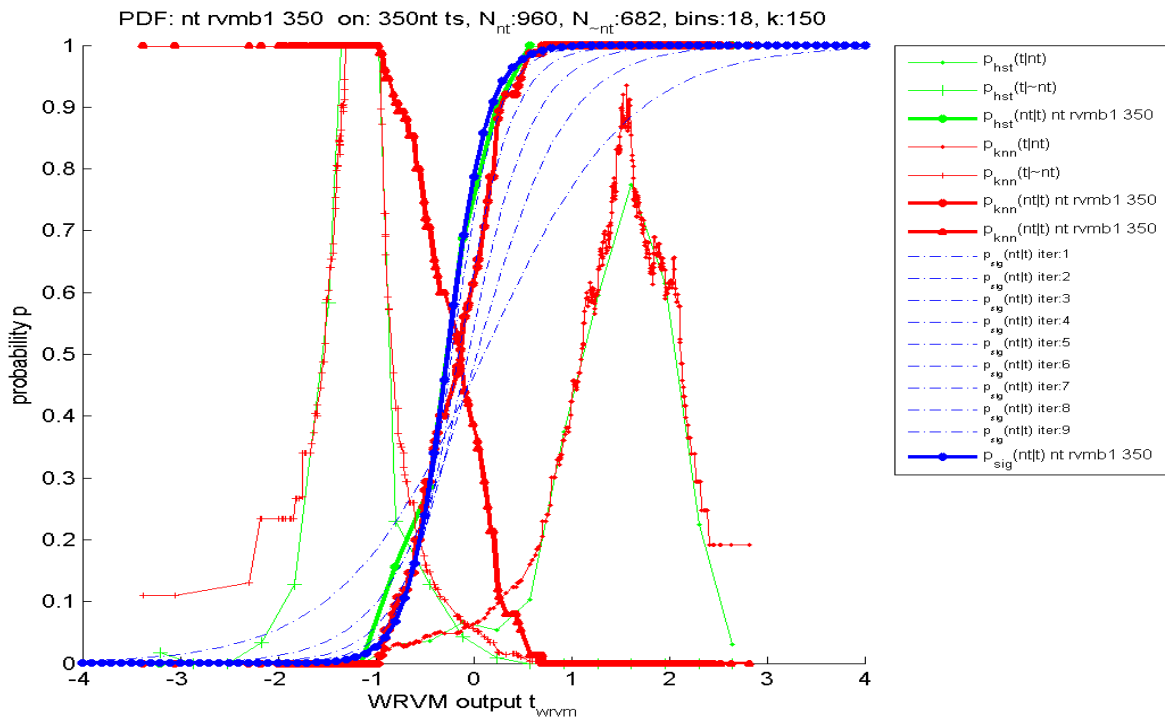


Figure 3-25: Probabilistic W-RVM using sigmoid function fitting
 On the example of the nose-tip feature, nt the estimation of the PDF is shown, using the histogram method (green thin lines $p_{hst}(t|nt)$ ⁸ and $p_{hst}(t|\sim nt)$), and using the k-NN technique (red thin lines $p_{knn}(t|nt)$ and $p_{knn}(t|\sim nt)$). The thick green curve, $p_{hst}(nt|t)$ shows the obtained probabilistic W-RVM outputs for the histogram method and the thick red curve, $p_{knn}(nt|t)$ for k-NN. For the sigmoid function the fitting iterations are shown (dot-dashed blue lines $p_{sig}(nt|t)$) and the final probabilistic W-RVM output, obtained by the sigmoid fitting (thick blue line).

⁸ nt is used as abbreviation for \mathbf{x}_{nt}

In [63] it is shown that the SVM+sigmoid combination is comparably to a raw SVM and a kernel method entirely trained with regularised maximum likelihood. The SVM+sigmoid combination preserves the sparseness of the SVM while producing probabilities that are of comparable quality to the regularised likelihood kernel methods.

Using the sigmoid fitting, we find a method to obtain a calibrated posterior probability output of our classifier to enable post-processing. Standard SVM's and our W-RVM do not provide such probabilities without the sigmoid fitting.

Compared to non-parametric methods introduced above the training of probabilistic W-RVM output using sigmoid fitting is stable and sensitive to the parameters. We trained and compared the probabilistic W-RVM output estimation using the iterative fitting of the sigmoid function (blue lines in Figure 3-26) with the histogram (green) and k-NN (red) techniques for all trained features. As examples, the methods are visualised in Figure 3-26 for the left eye (*le*) and left mouth corner (*lm*). The Probabilistic W-RVM can be used to find the final feature assortment and for the advanced PSM in the following sections.

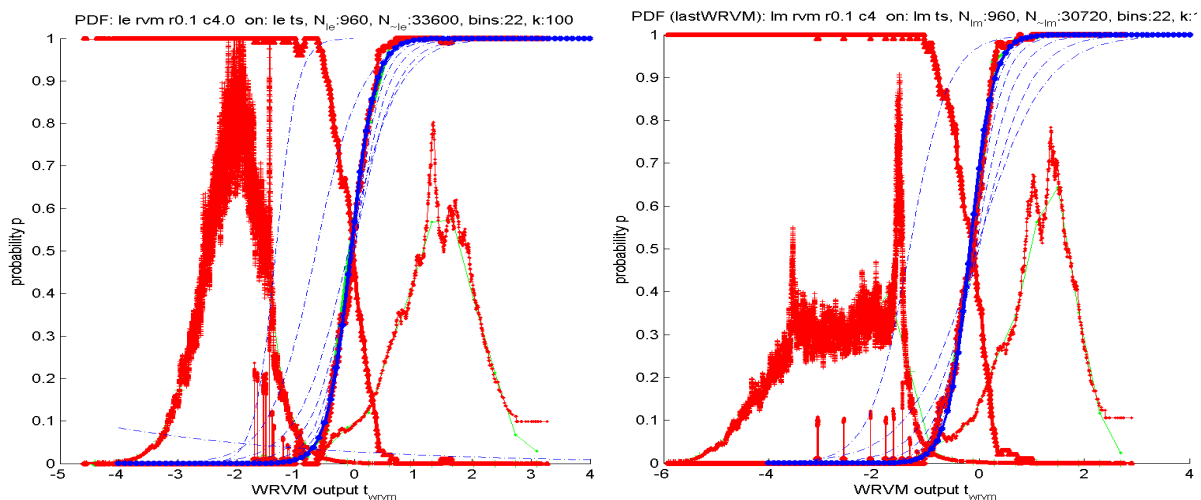


Figure 3-26: Stable Probabilistic W-RVM for all facial features

We trained and compared the estimation of Probabilistic W-RVM's, using an iterative fitting of the sigmoid function (*blue lines*), with histogram (*green*), and k-NN (*red*) techniques for all trained features. As examples, the results for the left eye (*le*) and left mouth corner (*lm*) are visualised (see Figure 3-25 for the legend).

3.5. Cascaded Framework for Facial Feature Detection

3.5.1. Single W-RVM Detector

The in the previous sections trained W-RVM classifiers can now be applied for detection. Identical to the face detector we use a sliding-window method. That means an observation window of the size of the feature is slid over each column and row of the pyramid image. The image pyramid is used to detect objects with different sizes by not changing the dimensions of the observation window (Figure 2-2). At each pixel location, the classifier (Table 2-2) is executed on the patch under the observation window. A cluster of detections is obtained for each object, because the classifiers are slightly translation invariant. Hence, we use an overlap-elimination after incorporating the last W-RSV and after the SVM as final classifier. The overlap-elimination reduces the clusters to the locations with the best detection certainties. The outputs of the four stages of the W-RVM detector are seen in Figure 3-27 for the left eye feature on an example image. The stages of the single W-RVM detector are summarised in Table 3-6.

Single W-RVM detector:

1. Fast W-RVM classifier stage (see Chapter 2) at each location of the observation window sliding over each pixel of the pyramid image. This stage rejects efficient large image areas as non-objects,
2. Overlap-elimination for the clusters of the detections, this extracts a set of best detections per cluster,
3. Full SVM using all Support Vectors for the remaining locations,
4. The final overlap-elimination extracts only the detection of each cluster with the highest certainty.

Table 3-6: Stages of a single W-RVM detector

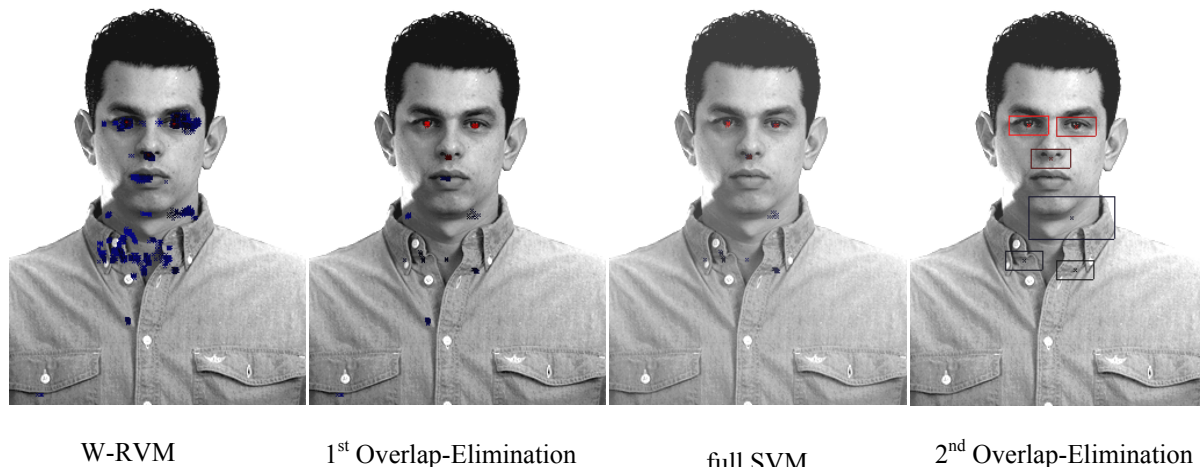


Figure 3-27: Example for detection process of the left eye for one image
 We use four stages (from left to right) for each detector. Red coloured Pixels indicate locations of the centres of patches with a high and blue with a low detection certainty. The boxes within the right image show the detection candidates on the full image. The left eye is best detected. The certainty at the right eye is also high (because the eyes are similar). The certainty of the four False Acceptances is low.

In Figure 3-28, we visualise some examples of the output of the W-RVM detector for the left mouth corner. Facial features are ambiguous within a face at a local view. For example, false candidates are also detected near the eye corners on the right image.

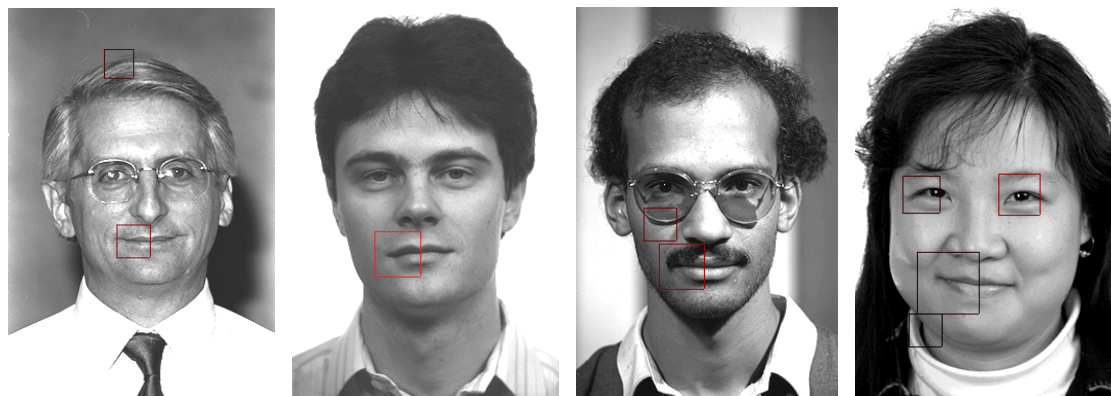


Figure 3-28: Examples for detection results of the left mouth corner
 Red coloured boxes show detection candidates with high and blue with low certainties. All mouth corners are detected, but facial features are ambiguous within a face at a local view. For example in the right image, we get false candidates also near the eye corners.

3.5.2. W-RVM Facial Feature Set Detector

The goal is to build a multi feature detector for the in Section 3.3.1 (see Table 3-5) chosen facial feature points. To reduce the problem that the features are ambiguous and to speed up the detection process over all features we first run the face detector on an image. Relative to the detected face we define a field of interest (FOI) containing the regions where to apply the facial feature detectors. The FOI for the detectors is defined empirically. As discussed in Section 5.1.3 the Prior Shape Model using the Morphable Model can be used for an automatic

and more precise adjustment of the ROI's. If the face cannot be detected (e.g. if the face is partly occluded) the ROI's are defined over the full image.

Comparing the left and middle images in Figure 3-29 it can be seen that the FAR is reduced using this cascaded algorithm. Although we get after running all single detectors for each feature a list of candidates and have to find the final feature assortment.

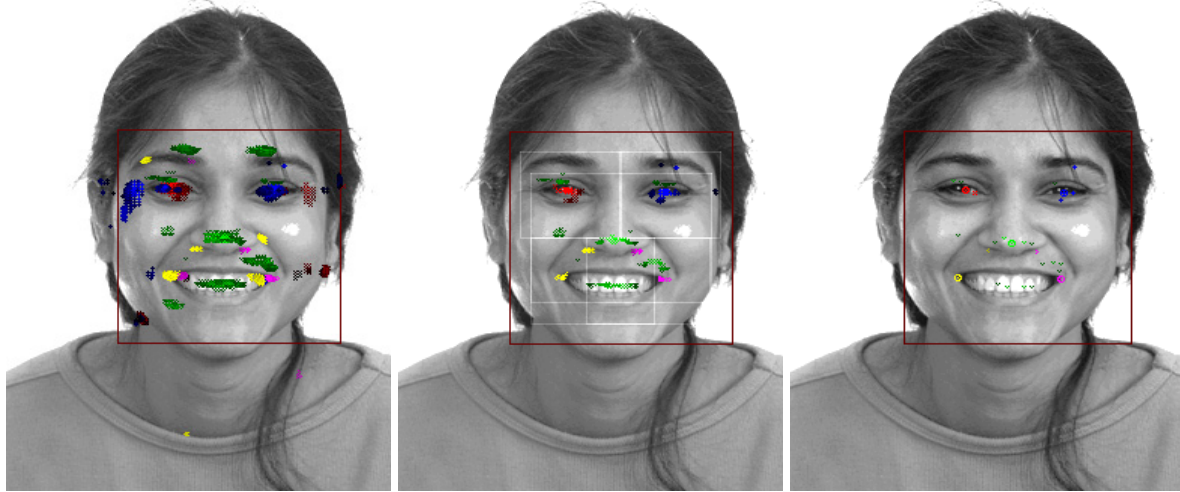


Figure 3-29: Reducing FRR by a cascaded framework and use of a FOI

The *left* image shows the detection result of the W-RVM face detector (*red box*) and the results of the first W-RVM stage for five features (left (*red*) and right (*blue*) eye, nose tip (*green*), left (*yellow*) and right (*magenta*) corner of the mouth; *bright* colours indicate high, *dark* low certainties). Applying the detectors in the FOI (*brightened boxes*) reduces the FAR of the first W-RVM stage in the *middle* image. The *right* image shows the remaining detections after the last W-RVM detectors stage. The final feature assortment with the highest certainty per feature is *encircled*.

If $\mathbf{x}_{ffp, i_{ffp}} \in \mathbb{R}^2$ with $i_{ffp} = 1, \dots, N_{ffp}$ are the 2D locations of the N_{ffp} detection candidates, e.g. of one of the in Table 3-5 chosen feature $ffp \in \{le, re, nt, lm, rm, lx, rx, ls, lb, rb\}$, so the simplest idea is to choose for each feature the index of the candidate with the maxima detection certainty

$$\hat{i}_{ffp} = \arg \max_{i_{ffp}} cert_{ffp}(\mathbf{x}_{ffp, i_{ffp}}), \quad (3.5)$$

where $cert_{ffp}(\mathbf{x}_{ffp, i_{ffp}})$ is the certainty of the i_{ffp} -th detection by the W-RVM detector for the facial feature ffp . The final feature assortment for e.g. the first five facial features from Table 3-5 is defined as

$$\hat{\mathbf{h}} = (\hat{i}_{le}, \hat{i}_{re}, \hat{i}_{nt}, \hat{i}_{lm}, \hat{i}_{rm}). \quad (3.6)$$

All other detections can be rejected as non-feature points.

In Figure 3-30 the W-RVM stages and the selection of the final feature assortment based on (3.5) and (3.6) are demonstrated for all ten features chosen in Table 3-5.

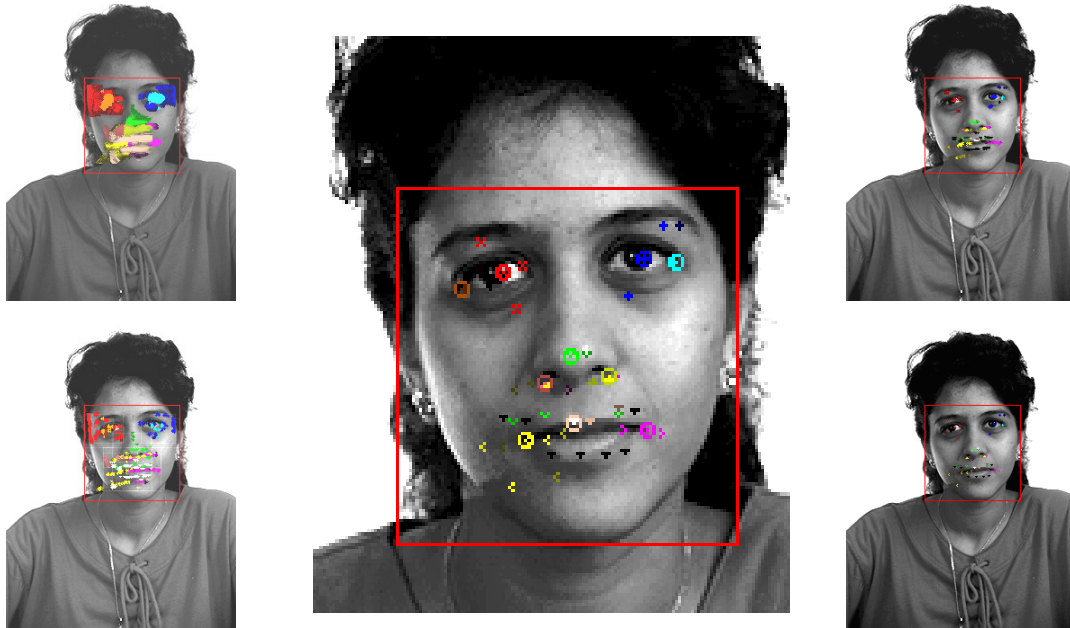


Figure 3-30: Stages of the W-RVM with ten facial features

The face detector result (*red box*) and the four stages are shown for ten facial features (the *colours* and the *markers* are defined in Table 3-5; *light colours* indicate high and *dark colours* low detection certainties). The *left top* image shows the result after the first fast filter stage of the W-RVM, *left bottom* the results after the 1st overlap-elimination, *right top* after applying the full SVM for the remaining patches and *right bottom* the final result after the 2nd overlap-elimination. The *middle image* shows the face detail where the detections with the highest certainty per feature are *circled*.

This simple maximum measure finds for several examples the best assortment as seen in Figure 3-31. However, the detection rate is not satisfying using only the 2D appearance model (Figure 3-32), because the 2D information is too ambiguous at this only local view. Hence, we want to use a 3D Prior Shape Model function taking advantage of the 3D MM to find the final feature assortment within all combinatory possible candidate sets for the feature points. This correlation classifier will be detailed in the next section and is used as last stage of the W-RVM facial feature set detector. We summarise the cascaded framework in Table 3-7.

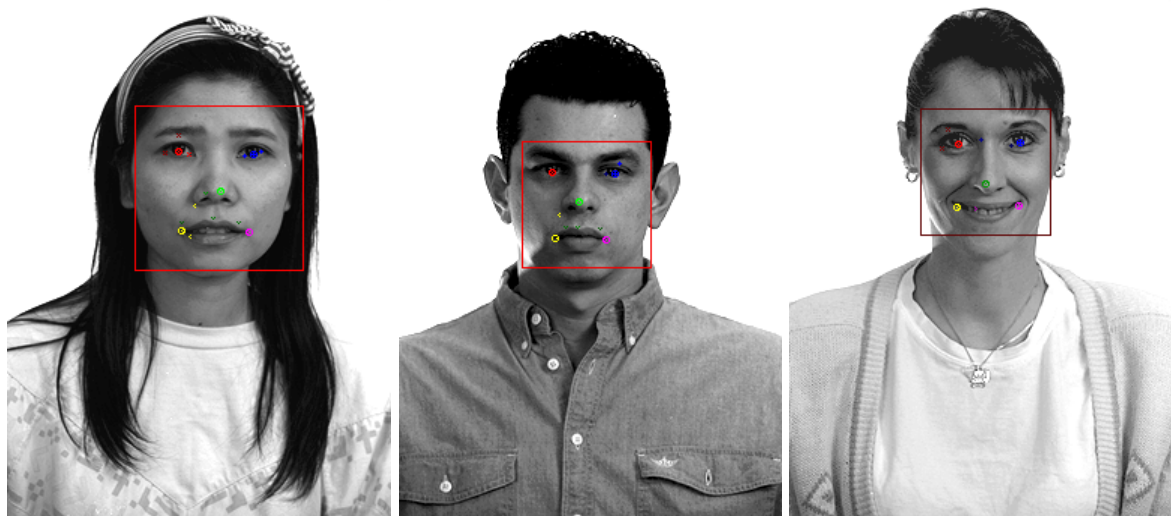


Figure 3-31: Good examples for final feature assortments by maximum rule

The final feature assortments (*encircled marks*) are shown for five features (Table 3-5) based on the maximum detection certainty per feature (see (3.6) and (3.5)).

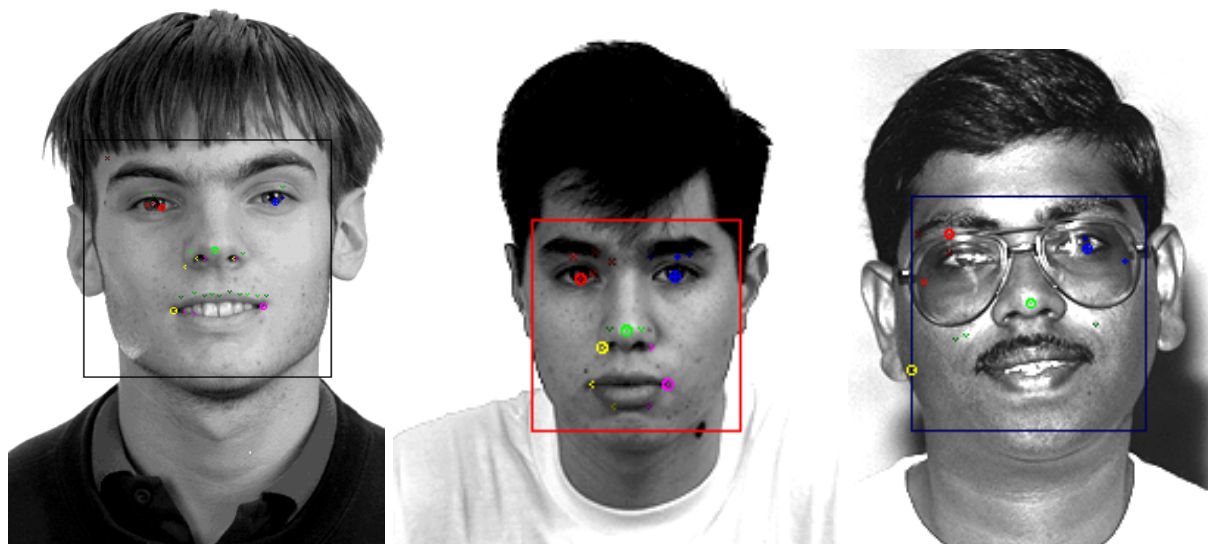


Figure 3-32: Not satisfying examples for the final feature assortments

The final feature assortments (*encircled marks*) are obtained as in the previous figure. In several examples the true assortment is found (*precious figure*) but the error rate is too high (*examples here*), especially if the features are difficult to detect, e.g. because of occlusions like at the *right* image.

W-RVM facial feature set detector:

1. Apply the single W-RVM detector (Table 3-6) for faces on the full image,
2. Define a FOI for the facial features relative to the detected faces by the first stage (Section 3.5.2). Apply the single W-RVM detectors (Table 3-6) for all features within their regions of interest,
3. Apply a correlation classifier, e.g. the PSM (Section 3.6) for the list of candidates of all features to find the final feature assortment.

Table 3-7: Stages of the W-RVM facial feature set detector

3.6. Evaluation of the Final Feature Assortment by PSM

For evaluating the final feature assortment from all combinatory possible hypotheses, we tried different approaches. The simplest was based on a maximum rule of the detection certainties (see Section and Figure 3-33).

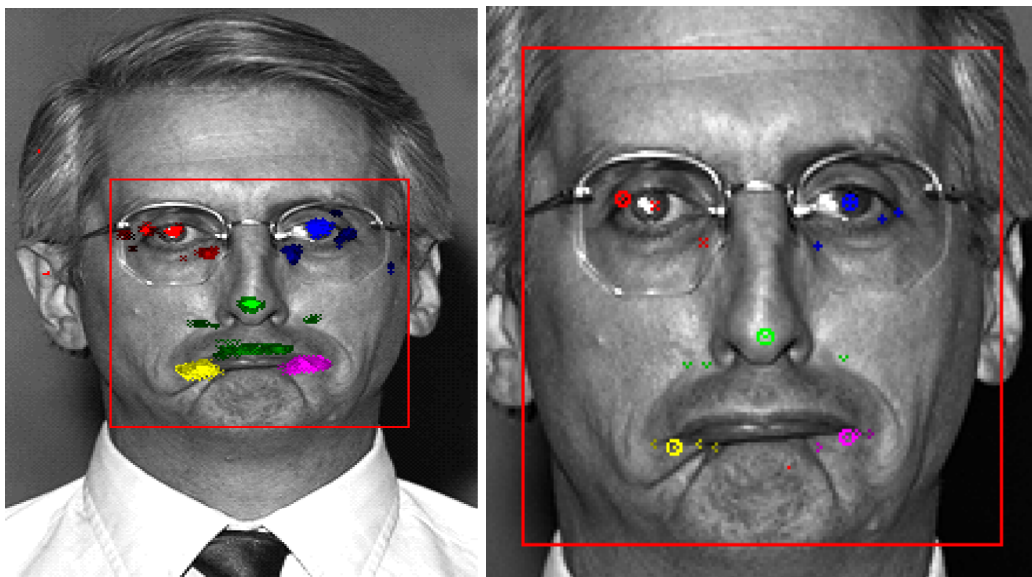


Figure 3-33: Final feature assortment using solely the 2D appearance model
 The W-RVM output of the first (*left*) and last (*right*) detector stage are visualised. The optimal final feature assortment (*encircled marks right*) is not found using the maximum detection certainty per feature (see (3.5) and (3.6)).

But because of poor image quality, like in Figure 3-33 or that the local image representation of facial features are ambiguous within the image, like in Figure 3-28 a model-based correlation classifier should be exploit. As face model, our 3D MM can be used to take advantage of the correlations between the detection candidates. That means to find the best feature assortment we unify the 2D appearance model using the W-RVM detectors and the 3D Prior Shape Model taking advantage from the Morphable Model as schematised in Figure 3-34.

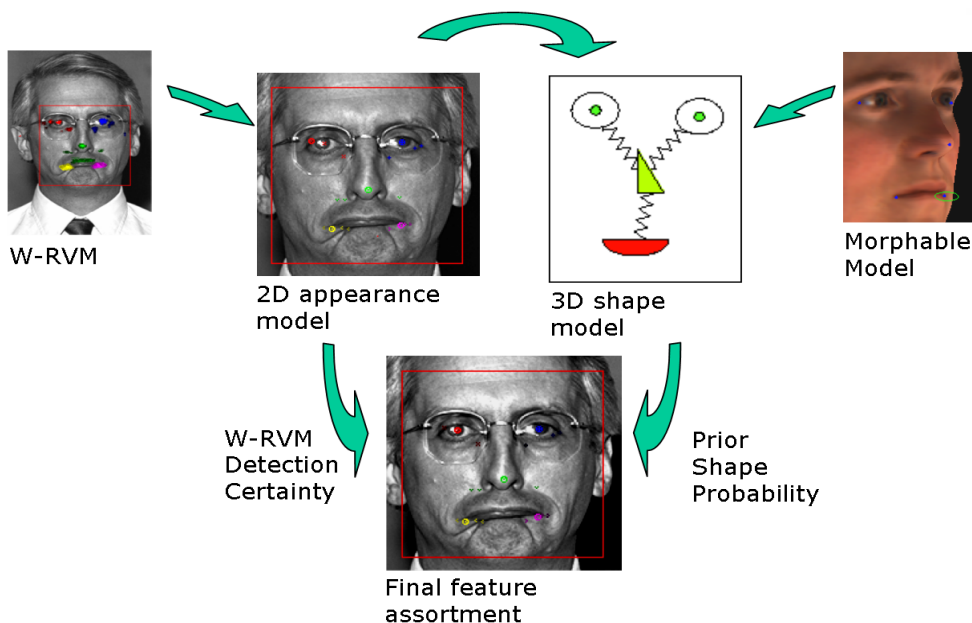


Figure 3-34: Unifying 3D Prior Shape Model and W-RVM as 2D appearance model
 The final feature assortment is found by unifying the certainty obtained from 2D appearance model (W-RVM detectors) and the probability of the Prior Shape Model, taking advantage from the 3D Morphable Model.

In our research, it turned out that a PSM restricted to only five facial features and using only the average face of the 3D MM is not sufficient for the correlation classifier. In addition, applications on the found feature points, like the estimation of the pose of the face, were not stable by only five facial points. For this reason, we chose more facial features in Section 3.3.1 based on the multi-criteria evaluation. In Section 3.3.2 and the training and validation of the W-RVM detectors is described and in Section 3.4, we fit a sigmoid function to obtain a probabilistic output of the classifier.

Now we want to introduce an advanced Prior Shape Model able to use more facial feature points and taking more advantage of the MM by using the first principle components of the model. Moreover, the method uses an occlusion probability. Here we want to summarise and adapt the by Romdhani et al. [74], [78] introduced approach for our intention. For more background on probabilistic feature point detection Fergus et al. [30] and Felzenszwalb et al. [29], [18] can be considered. The main difference to [78] is that we will use the W-RVM detectors instead of the SIFT key point detection. The SIFT algorithm delivers a general set of key points, they have to be rated to the different appearance models of facial feature points. In our case the candidates for the detection points are already related to specific feature points, hence we do not use a set of key points, but list of candidates per features. Romdhani et al. use a probabilistic model based on scale and orientation estimation of the key points. For the present, we have to adapt the approach not using this feature. However, we started a project for regression function estimation to be able to take advantage of the scale and orientation likelihood (see Section 5.2.3).

We want to estimate the probability of the position of a set of N_p feature points for a class of objects, independently of the viewpoint.

$$p(\mathbf{X} | Object) \approx p(\mathbf{X} | \hat{\theta}) = p(\mathbf{X}_r | \hat{\theta}) p(\mathbf{X}_{\bar{r}} | \hat{\theta}), \quad (3.7)$$

where \mathbf{X} (a $2 \times N_p$ matrix) are the 2D locations in the image of the feature points, \mathbf{X}_r the N_r reference feature points (a $2 \times N_r$ matrix), $\mathbf{X}_{\bar{r}}$ the non-reference feature points (a $2 \times N_p - N_r$ matrix), and $\hat{\theta}$ the maximum likelihood (ML) shape model parameters, estimated from the reference feature points.

The idea is i) to make a 3D model for the class of objects, then ii) to estimate the model parameters from a small set of N_r reference points by:

$$\hat{\theta} = \arg \max_{\theta} p(\mathbf{X}_r | \theta), \quad (3.8)$$

and iii) to derive the probability of the ML parameters using the full set of reference and non reference points, by assuming that their position is independent *given the maximum likelihood of the model parameters*:

$$p(\mathbf{X} | \hat{\theta}) = \prod_i^{N_p} p(\mathbf{x}_i | \hat{\theta}). \quad (3.9)$$

The 2D positions in the image of the feature points \mathbf{X} are obtained by multiplying the 2×4 weak-perspective matrix \mathbf{P} with the shape matrices as follows:

$$\mathbf{X} = \mathbf{P}\mathbf{S}^0 + \sum_{j=1}^L \alpha_j \mathbf{P}\mathbf{S}^j, \quad (3.10)$$

$$\text{where } \mathbf{P} = \left(\left(\begin{array}{ccc} f & 0 & 0 \\ 0 & f & 0 \end{array} \right) \cdot R_\gamma R_\zeta R_\phi t_{2d} \right). \quad (3.11)$$

$\mathbf{S}^0, \mathbf{S}^j$ are the mean and principal components, arranged as $4 \times N_p$ matrices (the 4th row of \mathbf{S}^0 is 1 and the ones of \mathbf{S}^j are 0), and α_j are the shape parameters. In Equation (3.11) is f the focal length, γ is the image plane rotation (roll angle), ζ is the elevation rotation (pitch angle), ϕ is the azimuth rotation (yaw angle) and t_{2d} is a 2D translation.

In this framework, the model parameter, θ , is composed of the shape coefficients, α_j , the projection parameters: f, γ, ζ, ϕ , and the translation t_{2d} . The mean and the covariance matrix of the projection parameters and of the shape parameters are obtained by solving the following problem:

$$\begin{aligned} \{\hat{\mathbf{P}}, \hat{\alpha}\} &= \arg \max_{\mathbf{P}, \alpha} p(\mathbf{X}_r | \mathbf{P}, \alpha) \\ &= \arg \min_{\mathbf{P}, \alpha} \sigma_{X_r^d}^2 \left\| \mathbf{P}\mathbf{S}_r^0 + \sum_j^M \alpha_j \mathbf{P}\mathbf{S}_r^j - \mathbf{X}_r \right\|^2, \end{aligned} \quad (3.12)$$

where M is the number of shape parameters that are estimated ($M \leq 2N_r$) and $\sigma_{X_r^d}^2$ is the variance of the detection noise of the reference points, assuming a Gaussian distribution.

The mean and covariance of the model parameters can be estimated from (3.12). For clarity of the presentation, we do not provide detail of the following functions $f(\cdot)$, but they are provided in [78] (see there Equation (7), (10)):

$$\mu_{\hat{\mathbf{P}}} = f(\mathbf{X}_r), \quad \Sigma_{\hat{\mathbf{P}}} = f(\sigma_{X_r^d}^2) \quad \text{and} \quad \mu_{\hat{\alpha}} = f(\mathbf{X}_r), \quad \Sigma_{\hat{\alpha}} = f(\sigma_{X_r^d}^2) \quad (3.13)$$

The projection matrix must agree to the following constraints. This leads to an early rejection rule and makes the algorithm efficient.

$$\left\| \hat{\mathbf{P}}_{1,1:3} \right\|^2 - \left\| \hat{\mathbf{P}}_{2,1:3} \right\|^2 = 0, \quad \text{and} \quad \hat{\mathbf{P}}_{1,1:3} \hat{\mathbf{P}}'_{2,1:3} = 0 \quad (3.14)$$

We can now proceed to the estimation of $p(\mathbf{x}_i | \hat{\theta}) = p(\mathbf{x}_i | \hat{\alpha}, \hat{\mathbf{P}})$, where \mathbf{x}_i denotes the i -th column vector of the matrix \mathbf{X} (the position of the i -th feature). According to our model, it is shown in [78] (12), (14) that we can evaluate the mean and the covariance matrix using (3.13) by:

$$\mu_{x_i} = f(\mu_{\hat{\mathbf{p}}}, \mu_{\hat{\alpha}}), \quad \Sigma_{x_i} = f(\Sigma_{\hat{\mathbf{p}}}, \Sigma_{\hat{\alpha}}, \text{non-estimated shape PC}), \quad (3.15)$$

and so the probability of $p(\mathbf{x}_i | \hat{\alpha}, \hat{\mathbf{P}})$ can be computed using a Gaussian model.

Using this ML we can show in Figure 3-35 the contour containing 99% of the probability of the estimation of the position of the mouth corner given four reference points (respectively 12 at the right-hand side face). A Gaussian noise with a STD of 3 pixels (respectively 10 pixels in fourth face from left-hand side) was added to all reference points. Note that the ellipses change as the reference points vary with the pose of the face, the number of reference points or the noise. As it can be seen, the uncertainty area is rather small.

The small uncertainty shows that the proposed shape model cannot only be used for detection applications, but also as a general view invariant probabilistic shape model, which could supersede other shape models such as the Active Shape Model [17], for instance.



Figure 3-35: Probability of a facial feature point given model parameters

The *first three* synthetic faces (from left to right) show the contour (ellipse) containing 99% of the probability of the estimation of the position of the mouth corner given four reference points and a STD of 3 pixels (pink crosses). The size of the uncertainty area depends on the (unknown) pose (size 18.7^2 , 18.6^2 and 17.9^2 pixel square), the noise (*4th face* $\sigma_{x_i}^2 = 10$) and number of reference points (*last face* $N_r = 12$).

Now the above introduced viewpoint invariant probabilistic shape model is applied as correlation classifier. The task of finding the final feature assortment is to find correspondence between the set of N_p model feature points and an assortment of points from the N_{ffp} detection candidates. The 2D image positions $\mathbf{x}_{ffp, i_{ffp}} \in \mathbb{R}^2$ with $i_{ffp} = 1, \dots, N_{ffp}$ are the N_{ffp} detection candidates, of one of the feature $ffp \in \{le, re, nt, lm, rm, lx, rx, ls, lb, rb\}$, e.g. chosen in Table 3-5. A set of correspondences is represented by an N_p dimensional hypothesis vector, \mathbf{h} , whose element \mathbf{h}_i is the index i_{ffp} of the candidate point in correspondence with feature point ffp . If no such corresponding point exists, then the value of \mathbf{h}_i is set to zero, hence, $0 \leq \mathbf{h}_i \leq N_{ffp}$. The size of the ensemble \mathbf{H} of values that \mathbf{h} can take, is $|\mathbf{H}| = \prod_{ffp=1}^{N_p} N_{ffp}$, e.g. by $\forall_{ffp} N_{ffp} = 10$ detection candidates for $N_p = 10$ facial features, then we have to consider $1e10$ combinations as hypotheses.

The final feature assortment of all hypotheses at the configuration space \mathbf{H} is found by:

$$\hat{\mathbf{h}} = \arg \max_{\mathbf{h} \in \mathbf{H}} \underbrace{p(\mathbf{a} | \mathbf{h}, \hat{\theta})}_{\text{appearance}} \underbrace{p(\mathbf{x} | \mathbf{h}, \hat{\theta})}_{\text{shape}} \underbrace{p(\mathbf{h} | \hat{\theta})}_{\text{occlusions}} / p_b, \quad (3.16)$$

where \mathbf{a} is the appearance of the detection candidates, \mathbf{x} are the 2D positions of the candidate points, and p_b is the background likelihood. For the appearance probability, we use the output of the Probabilistic W-RVM classifiers at the detection locations as introduced in Section 3.4. All other detection candidates not included within the index vector $\hat{\mathbf{h}}$ are rejected as non-feature points.

Using a brute force approach is very slow, because of the dimensionality of the set of all correspondences \mathbf{H} . For efficiency, we take advantage of the Bellman principle (also known as a dynamic programming equation, [4]). Only all combinations of reference point positions have to be considered. The non-reference points are optimised independently to each other, so (3.9) can be re-written as

$$p(\mathbf{X} | \hat{\theta}) = \prod_i^{N_p} p(\mathbf{x}_i | \hat{\theta}) = p(\mathbf{X}_r | \hat{\theta}) \prod_{i \in \bar{r}} p(\mathbf{x}_i | \hat{\theta}). \quad (3.17)$$

This reduces the complexity from $N_a^{N_p}$ to $N_a^{N_r+1}$, if we use e.g. a constant number of $\forall_{\text{fpp}} N_{\text{fpp}} = N_a$ candidates per feature, so we have 1e5 instead of 1e10 combinations to evaluate.

The second opportunity for efficiency is to use the projection constrains (3.14) for the projection matrix \mathbf{P} . As a pre-stage all combinations of reference points leading to invalid projection matrices, i.e. one term on the left-hand side larger than a defined threshold, are rejected.

In Figure 3-36, we show the improvement on an example image. The left image shows the final feature assortment encircled using the maximum rule from Section 3.5.2. Using a correlation classifier based on only five facial features does not find the true assortment (middle). But unifying the 2D appearance model based on the Probabilistic W-RVM detectors and the advanced PSM for ten features and incorporating more the 3D MM we gain the true facial feature point assortment (right).



Figure 3-36: Estimation of the final feature assortment using the PSM

The *left* image shows the final feature assortment *encircled* using the maximum rule from Section 3.5.2. Using only five facial features for the PSM does not find the true assortment (*middle*). But unifying the 2D appearance model based on the Probabilistic W-RVM detectors and the advanced PSM for ten features we gain the true facial feature point assortment (*right*).

To validate the correlation classifier we tested on a set of FERET images disjoint to the training sets. We used for all correlation classifier experiments the same detection results. Therefore, the face detection results and the detected candidates per facial feature and so the configuration space \mathbf{H} of all hypotheses and the appearance certainties of the candidates were constant. The best (nearest to by hand-labelled points) final feature assortments $\mathbf{h}^* \in \mathbf{H}$ was labelled and used as ground truth. We tested the introduced methods in Section 3.4 to evaluate the appearance likelihood, \mathbf{a} in (3.16). We obtained best results using the Probabilistic W-RVM, gained by fitting a sigmoid function (Section 3.4.2). The Relative Maximal Feature Error d_{ffp} is used as measure. It is defined for the pairwise used facial feature classifiers (like the eye corners, eye centre, mouth corners, and nose corners) by:

$$d_{ffp} = \frac{\max(\|\mathbf{e}_{ffp_l} - \mathbf{a}_{ffp_l}\|, \|\mathbf{e}_{ffp_r} - \mathbf{a}_{ffp_r}\|)}{\|\mathbf{a}_{le} - \mathbf{a}_{re}\|} \quad (3.18)$$

where \mathbf{e}_{ffp_i} , \mathbf{a}_{ffp_i} are the estimated and annotated facial feature points for i in $\{l, r\}$ (left, right facial point) and $\|\mathbf{a}_{le} - \mathbf{a}_{re}\|$ is the inter-eye distance. For a single feature, like the nose tip or the upper lip point, the measure is defined as $d_{ffp} = \|\mathbf{e}_{ffp} - \mathbf{a}_{ffp}\| / \|\mathbf{a}_{le} - \mathbf{a}_{re}\|$.

In Figure 3-37, we show the cumulative histogram of d_{ffp} in percentage for all feature sets, where at least one candidate per features was detected. It turned out at the experiments that using ten instead only five facial features improved the localisation accuracy. Further experiments with larger data sets are planned. For our goals, it is interesting to see if the found facial features can be used as initialisation for the fitting of the 3D MM (automatic

landmark point detection). This evaluation will be done by working on the further unification of the 2D appearance model and 3D MM; see Section 5.1.2.

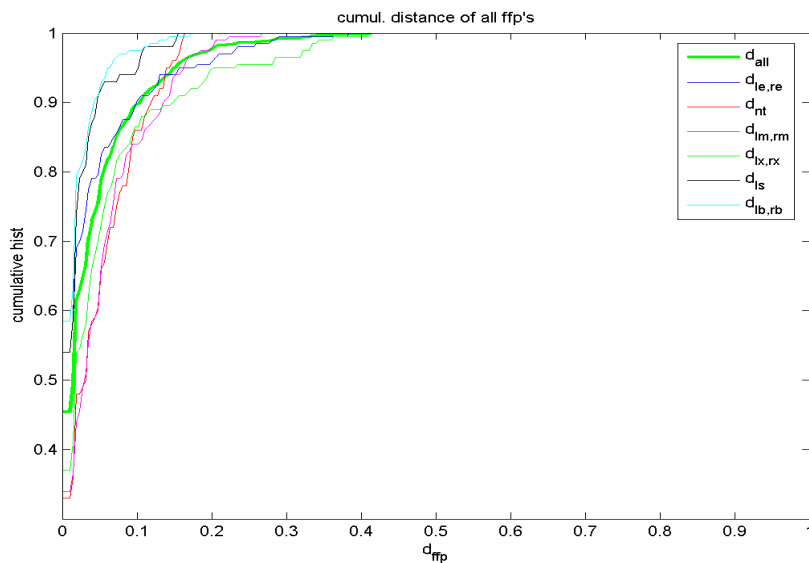


Figure 3-37: Facial feature set localisation accuracy
 The curves show the cumulative histogram of d_{ffp} (3.18) in percentage for all features (see Table 3-5 for the definition of the features) and d_{all} is the average over all features.

Chapter 4

Applications

In the previous chapters, we developed an efficient classifier and applied it to face and facial feature detection. For a first unification of the 3D Morphable Model (top-down approach) and the 2D image-based Wavelet Approximated Reduced Vector Machine (bottom-up approach), the 3D MM is used for the training of the W-RVM classifiers and to reduce the FAR. This unification will be developed further in Chapter 5. However, while developing the W-RVM, we applied the first unification stage for applications taking advantage of face and facial feature detection or the trained classifiers.

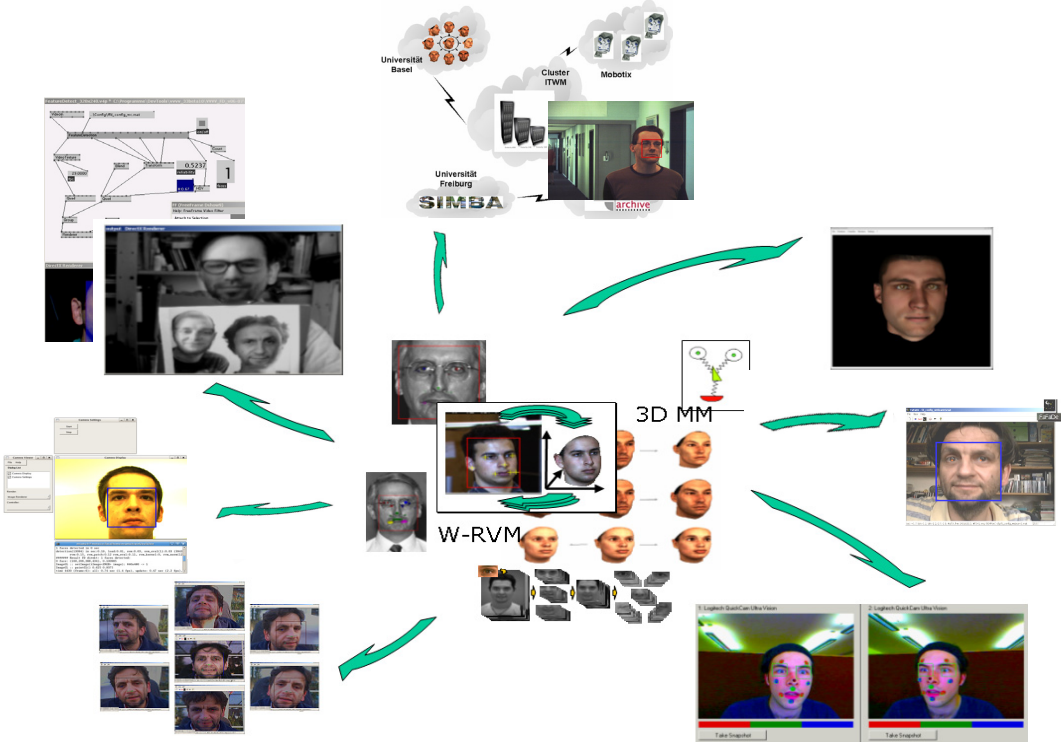


Figure 4-1: Applications taking advantage of the unification of the 3D and 2D model
The loops of unification of the 2D image-based classifier (W-RVM, centre left) and the 3D face model (Morphable Model, centre right) form the general background of this thesis. While developing the first unification stage, we applied the system to applications using face or facial feature detection (outer circle).

4.1. Applications Demonstrating the W-RVM

4.1.1. Application for W-RVM's – FD_FFpDetectApp

FD_FFpDetectApp is a command-line application and can be used to detect faces or other single features, but also for the detection of a set of facial features. The application uses the FD_Detect library, which realises the W-RVM facial feature detection approach proposed in the earlier chapters. The W-RVM approach and the performance of the classifiers and detectors are detailed there. In the configuration file, the size and content of the feature set, the classifiers, and threshold sets can be controlled. A command-line help for starting the application and a detailed description is available at the source documentation (see also Appendix A). Key features of the implementation are:

- Fully configurable via configuration files; to change from one to another project environment only the main configuration file needs to be replaced, without re-compilation. The parameter of the W-RVM approach can be optimised in the configuration file.
- Portable for Linux and Windows platforms,
- The application can be used in a batch mode on a list of images,
- For labelled data a statistic is automatically produced showing the FAR, FRR and runtime performance,
- A visualisation of the results of the W-RVM stages can be saved into debug images. The amount of saved images and what to visualise can be controlled in the configuration files,
- The amount of debug output at the console is adjustable.

4.1.2. Fast Face Detection – FaFaDe

FaFaDe (Fast Face Detection) is an application using a standard webcam and is implemented to demonstrate the efficient and accurate face detection algorithm. Accurate face detection is obtained in real time by 25 fps (on a Intel Pentium M Centrino 1600 CPU, at a resolution of 320×240 , step size 1 pixel, 5 scales). Using three resolution scales FaFaDe achieves more than 50 fps.

The GUI can be used for presentations but also to optimise the parameters online and to learn their influence, e.g. concerning the trade-off between accuracy and runtime performance.

Figure 4-2 shows the display of the application, visualising the detected face by a blue (high), or red box (uncertain) detection probability.

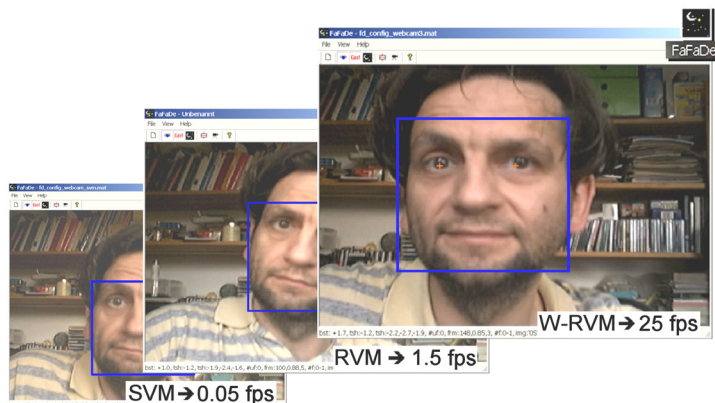


Figure 4-2: W-RVM face detection integrated into FaFaDe

FaFaDe (Fast Face Detection) is an application using a standard webcam and demonstrates the efficient and accurate face detection algorithm. FaFaDe can be used to optimise the parameters of the W-RVM (*right*) and to demonstrate the improved run-time performance in comparison to the standard Reduced Support Vector Machine (*RVM, middle*) and to the Support Vector Machine (*SVM, left*).

In Figure 4-2, right is a classifier used, which was trained on face patches with fixed eye coordinates. The fixed coordinates are marked by red crosses. What they can naturally not handle is in-plane rotation and the inter-eye distance is only as accurate as the resolution of the pyramid images enables it. In our experiments, it turned out, that the localisation accuracy of the face is higher by using fixed eye coordinates. In addition to the final detection result, all pre-stages of the detector can be visualised.

A detailed description and the usage of FaFaDe can be seen in the source documentation (see Appendix A for a documentation of FaFaDe and Appendix C for the trained classifiers).

4.1.3. Fast Facial Feature Detection – FaFaFeDe

The Fast Facial Feature Detector (FaFaFeDe) application is built to expose the Wavelet Approximated Reduced Vector Machine (W-RVM). The live video application is based on FaFaDe for detecting a single feature or for face detection. FaFaFeDe can run several facial feature detectors. Only detection is used; that means no information is used from the past frames to speed up the detection by tracking methods. The first W-RVM classifier is used to detect faces. In Figure 4-3 the found face is visualised by a surrounding box like by FaFaDe. Then an FOI is defined relative to the detected face, and the W-RVM stages of the detectors of the features are applied (top row of Figure 4-3, the FOI can be seen in the top-middle image by bright boxes). The results of the detectors (bottom left) are classified by a 3D model-based correlation classifier using the Prior Shape Model (bottom middle; here the PSM for five facial feature points is used; the new PSM function is not yet integrated to FaFaFeDe). By combining the 3D shape likelihood and the 2D appearance certainty, the final

feature assortment is found (bottom right). All other detection candidates are rejected. The application marks the most-likely feature assortment at the last three stages by circles.

A detailed documentation of the theoretical background of the approach, pseudo code how to train, and how the W-RVM detector works, is given in Chapter 2, 3 and in [70], [68], [71]. There is also the accuracy and runtime performance verified. The FaFaFeDe application is detailed at the software packed by a HTML documentation. In Appendix A the usage of FaFaFeDe is documented, e.g. how to start the application and the command-line parameters. It is explained which detection results are visualized, the used colours and markers for the features, or how the frame rate can be controlled. The stages of the realised detection are detailed and a guide is given on how to obtain optimal detection results and the best runtime performance. In addition, the parameters from the configuration files are explained showing how to adjust them in the configuration files and live at detection.

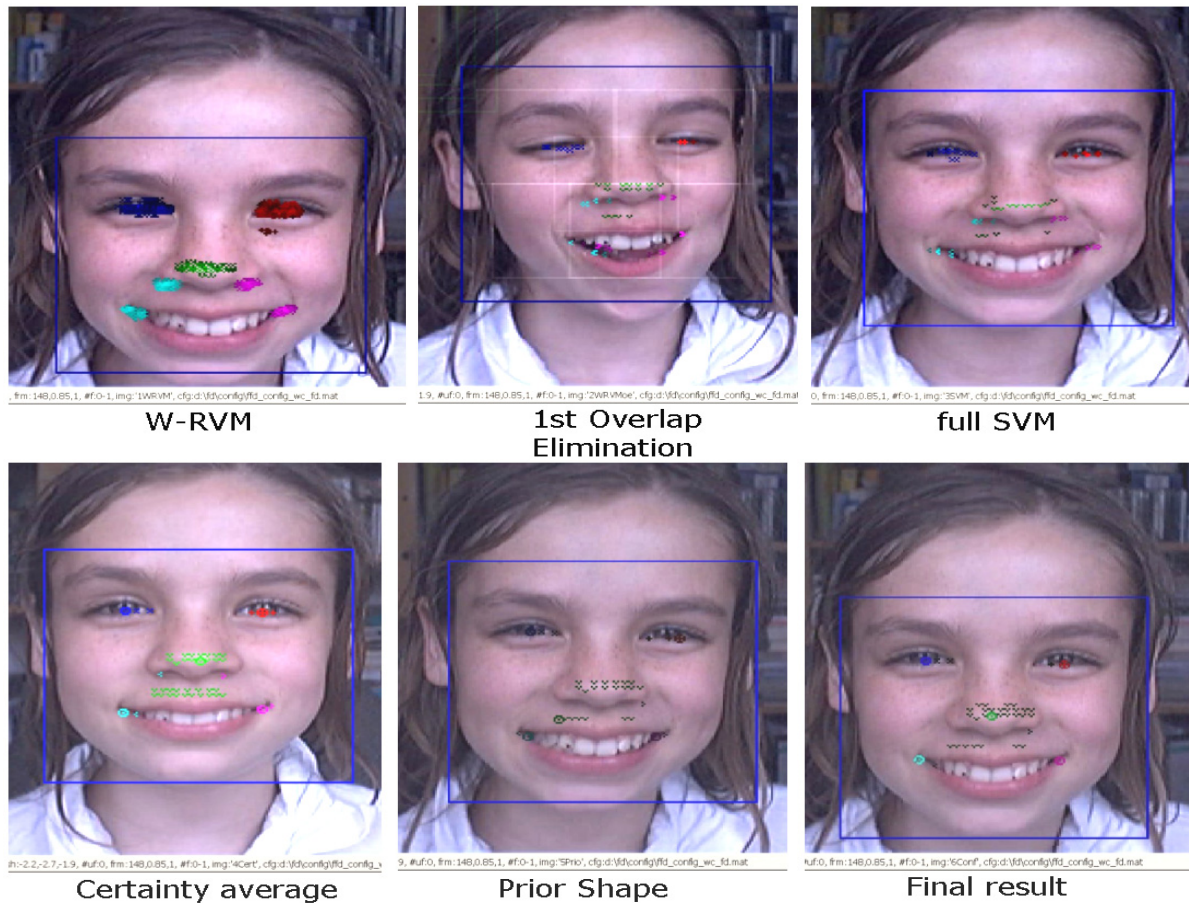


Figure 4-3: Facial feature set detection integrated into FaFaFeDe
The stages of the W-RVM feature set detection in combination with the PSM for five facial feature points are shown.

4.1.4. Pose Estimation integrated to FaFaFeDe

The W-RVM features set detection delivers an estimation of facial feature points. This set of facial points can be used to estimate the pose of the face. This could be achieved by training a Support Vector Machine [46]. However, a pose estimation is already integrated into the first fitting stages of the 3D MM. Moreover, by taking advantage of the MM as a three-dimensional model, we can estimate the pose without much effort. As detailed in Section 3.6, we use the 3D MM to build a correlation classifier. The 2D positions in the image of the feature point candidates are obtained by multiplying the projection matrix \mathbf{P} with the 3D shape matrices of the MM as seen in (3.10). The by the PSM function used projection matrix \mathbf{P} (3.11) is exploit for a pre-selection stage. All combinations of detection candidates are rejected if their projection matrix \mathbf{P} is not valid with respect to the constraints (3.14). From the projection matrices computed here, the three angles (roll, yaw and pitch; see (3.11)) can be used as pose estimation. First, we selected the projection from the combination of feature candidates with the highest shape likelihood. However, the best results were obtained using the angles computed from the projection matrix from the final feature assortment, which combines the shape and appearance likelihoods. Therefore, we could improve the pose estimation, but more than five facial features are needed, since the estimation is too sensitive for the detection of the nose tip. To evaluate pose estimation for more facial feature points is one of the next goals.

For visualisation purposes, we integrated the pose estimation to FaFaFeDe. In Figure 4-4 the three angles roll, yaw, and pitch are shown within the status bar of the application for the live video stream.



Figure 4-4: Pose Estimation integrated in FaFaFeDe
 For visualisation purposes, we integrated the pose estimation to the FaFaFeDe live application. The three angles p : ($\langle roll \rangle$, $\langle yaw \rangle$, $\langle pitch \rangle$) are shown within the status bar of the application.

4.1.5. Fd_camFFDViewer

We implemented a new portable application for FFD, which is not limited to Windows like FaFaDe and FaFaFeDe because of the usage of the MFC. This application has fewer features but is a starting point for further extensions of the face and facial features detection approach. The Fd_camFFDViewer is limited to face detection or one facial feature but uses the identical classes as the facial feature set detection, only the GUI has to be expanded and the interface adapted.

This application takes advantage of the libraries of our research group, developed in the last few years, like librabbit, librender, or libcam. It uses the new standard interface for cameras and for GUI applications. As seen in Figure 4-5, it applies the standard display window, the window controller, and option dialogs. Therefore, it has a good opportunity to be used by related bachelor and master theses. The Fd_camFFDViewer could already be adapted for new applications. For instance, the projects “Face and Facial Feature Point Tracking” in Section 4.3.1, “Avatar Following with Eye and Head Motion” in Section 4.3.2, and the “Tracking of Higher Feature Parameters” in Section 5.2.4 took advantage of Fd_camFFDViewer as a starting point.

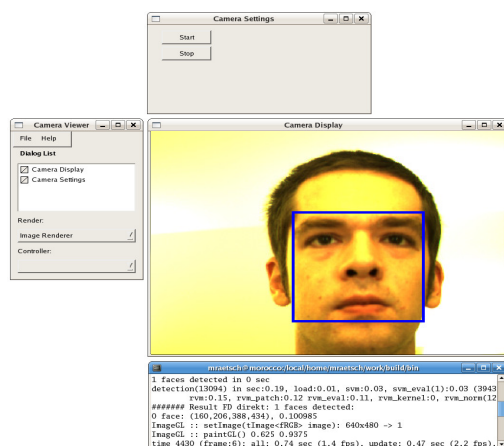


Figure 4-5: Face detection integrated into Fd_camFFDViewer

4.2. I-Search project

We also proved the performance and detection accuracy under real-life conditions at the “Institut für Techno- und Wirtschaftsmathematik” (ITWM) in Kaiserslautern.

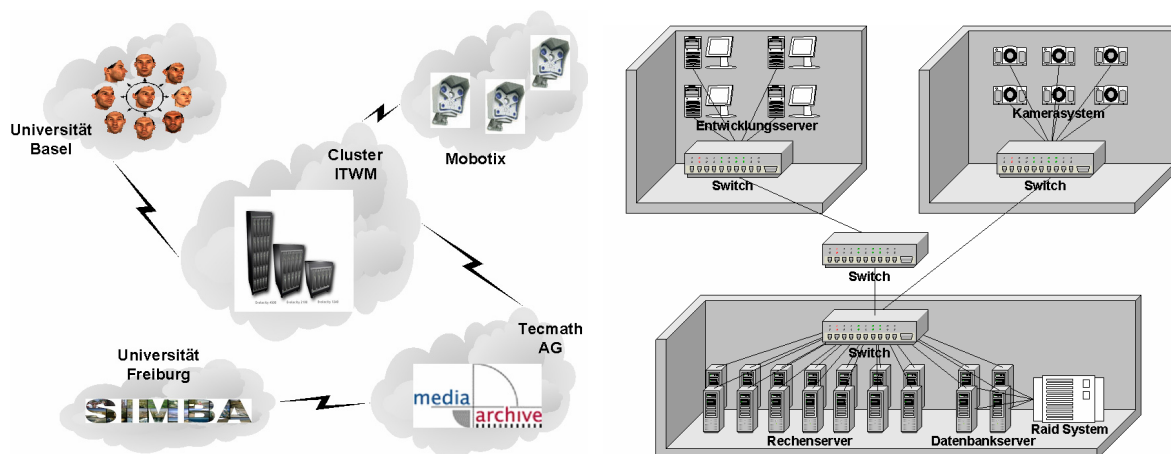


Figure 4-6: The I-Search project

The I-Search project [54] was a joint project by the following partners (Figure 4-6, left):

- Image processing: Graphics and Vision Research Group, University of Basel, Department computer science [37],
- Image processing: Albert-Ludwigs-Universität Freiburg, Institut für Informatik, Lehrstuhl für Mustererkennung u. Bildverarbeitung,
- Cluster optimisation and hardware installation: Fraunhofer Institut für Techno- und Wirtschaftsmathematik (ITWM),
- Data base applications: tecmath AG, GB Content Management Systems,
- Camera hardware: Mobotix AG.

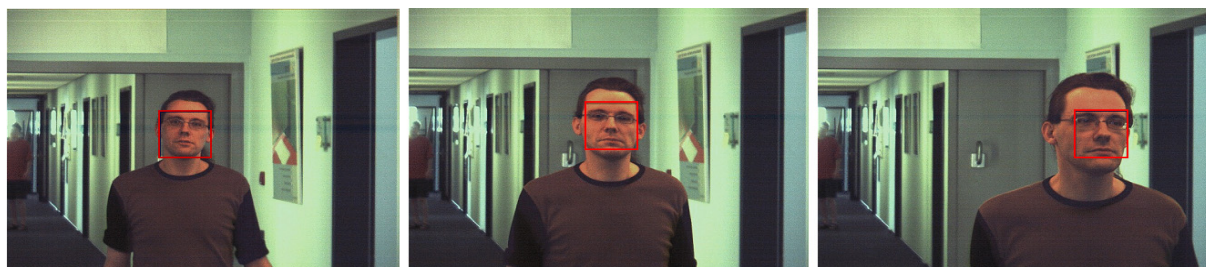


Figure 4-7: Example ITWM video sequence with results of the face detection

I-Search was the first project using an API of our face detector applying the W-RVM classifier introduced in Chapter 2. One intention of the project was to use a web crawler to search through the WWW for faces. A cluster architecture as seen in Figure 4-6, right was set up and the computational last optimised for each cluster.

The second intention was to build up a live detection system, as seen in Figure 4-7. One of the applications for this setting was our face detection approach. The API was applied by a web-service application.

At the final presentation, we could show that our W-RVM face detector was able to detect all of the persons, who passed by the installation in the demonstration.

A further documentation of the project and detailed descriptions of the results are given in our final report [69].

4.3. HCI, CHIL Applications using W-RVM

One of our intentions at the research group GraVis [37], is to apply the Morphable Model to face recognition and real-time facial feature or expression tracking, so that it can be used for Avatar Technology, Human in Computer Interaction and Computer in Human Interaction Loops (HCI, CHIL, see Figure 4-8). To be able to build up an interaction between the computer (e.g. using an avatar) and the person in front of it, the machine must be able to localise the person and track the emotion of its “conversation partner” in real time. To come closer to this intention, we can apply of the efficient W-RVM classifier and the introduced face and facial feature detection. Some applications following this intention by applying first HCI and CHIL aspects will be introduced in the next sections.

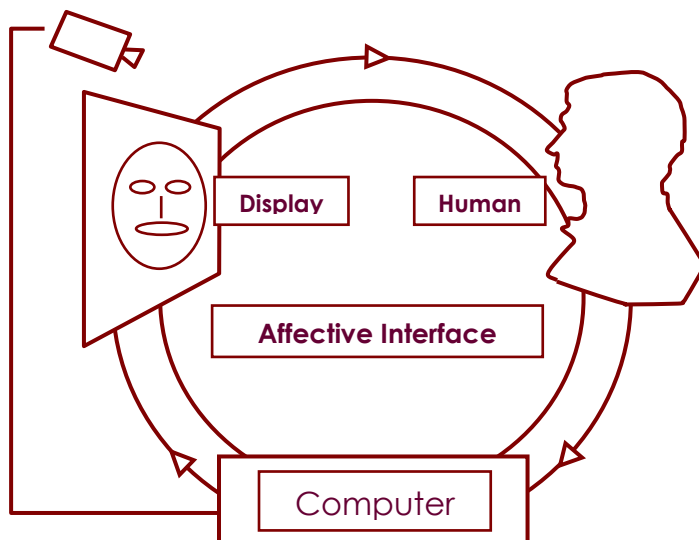


Figure 4-8: Schemata for Human in Computer Interaction (HCI, [37])

4.3.1. Face and Facial Feature Point Tracking

Yet image-based detection tasks are time consuming. For instance, detecting a specific object in an image, such as a face, is computationally expensive, as all the pixels of the image are potential object centres. Hence, all the pixels must be classified for all possible object sizes.

Up to this point, only detection algorithms are mentioned in this thesis. Detection uses a sliding observation window strategy. In a brute-force search for each column and each row of the entered image, patches are cut out and classified. To detect larger objects an image pyramid is used by down-sampling the image several times (see Figure 2-2). In Chapter 3 we obtain real time performance on VGA video streams using detection and the W-RVM as classifier. However, for video streams with high-resolution cameras or if we want to detect more objects at the same time (e.g. up to ten facial features) the sliding observation window strategy quickly becomes intractable.

It is obvious that the object's position and size vary only slightly from one video frame to the next. It is therefore possible to use information from the last step to speed up the search in the next frame. The process of seeking and following objects is called tracking. A method that is capable of using information of the previous iterations is the Condensation algorithm. This was proposed by Isard and Blake [53]. Condensation is even able to track objects in a highly cluttered background. The tracking method is a good alternative to the Kalman Filter [95]:

- Condensation can estimate the unknown a-posteriori probability function and does not need the assumption of a Gaussian distribution,
- The estimated function is multi-modal, i.e. it can have several maxima,
- System and measurement dynamics can be nonlinear,
- They are suited for parallelisation.

Condensation is a very stable and high-performance procedure that makes use of stochastic techniques. A probability distribution of possible image locations is represented by a randomly generated set of particles (also called samples). The idea is to estimate the probability function densely for areas of the images with a high a-priory likelihood and only roughly for the background. The prior obtained from the last frame is used to control the density of the samples over the model space at the current frame. Regarding this probability distribution (coded as size of the radii in Figure 4-9, left), new samples are chosen. It follows a prediction step by applying a translation of the samples as drift and stochastic noise as diffusion. The prediction is verified by a measurement function. The obtained probability of the samples is used for the new probability distribution, and the procedure starts again as schematised in Figure 4-9, left.

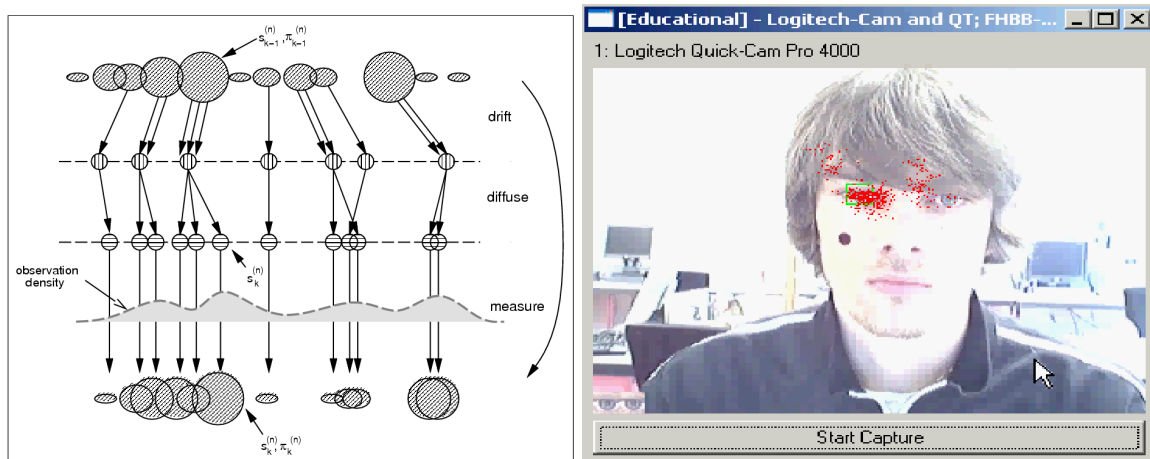


Figure 4-9: Condensation for face and facial feature tracking
 The Condensation tracking (schematised *left*) is applied for tracking the left eye feature (*right*). The density distribution is high for samples (*red points*) near the eye. Samples are wide distributed over the image if they have obtained a low probability (e.g. from the Probabilistic W-RVM classifier (Section 3.4)).

The original Condensation approach [53] is used to track contours of objects. This makes it difficult to implement features like the recognition of individual objects or the determination of orientation. Chappuis and Blanc [13] enhanced the Condensation algorithm. The goal of this cooperation with FHNW Basel (Department computer science, Prof. Hudritsch) was to build up a facial motion tracking system using a stereo webcam installation. Facial feature points should be tracked to expand the former project [12] using coloured markers. To track several markers at the same time and additionally the eyes without markers was not possible in real time using standard methods. Taking advantage of the efficient W-RVM classifier introduced in Chapter 2 and using a tracking approach (instead of the detection used in Chapter 3) enables to use more features. Figure 4-10 shows the installation tracking two times eleven features in real time based on Condensation tracking.

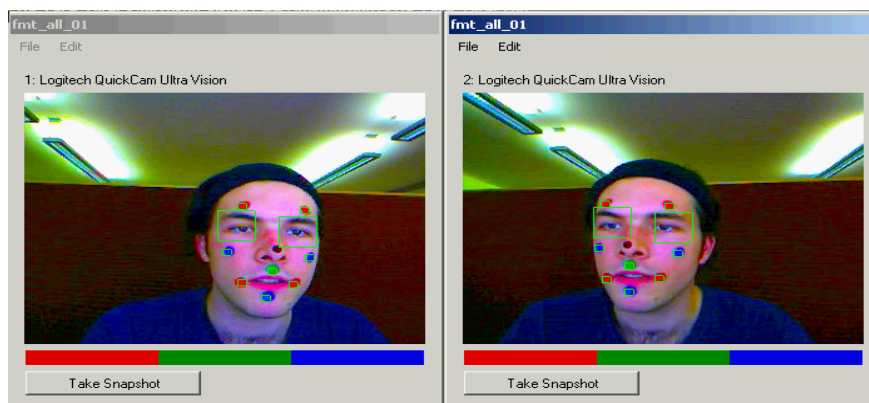


Figure 4-10: Facial motion tracking using a stereo camera installation
 Using Condensation instead of detection (sliding observation window strategy) enables tracking of two times eleven features parallel on one standard PC in real time.

4.3.2. Avatar Following with Eye and Head Motion

Tracking of emotions as discussed in the previous section is an important aspect to start to tackle HCI and CHIL aspects. In the future, an interaction between humans and computers should be as natural as a conversation between humans. Before the avatar can get in contact to its “conversation partner”, it must be able to localise the person in front of it. Eye contact is an important aspect of conversations in the field of perception psychology. Starting with this intention, we can apply the real-time W-RVM classifier and the proposed face and facial feature detection.

In the project 'I can see you', C. Horisberger [45] developed a 3D animated avatar reacting to head movement of a person in front of the computer using a face detector. In addition, the avatar should react to the position and distance by moving the eyes.

The project explores the various steps needed to create an animated 3D model, acting in this HCI framework. A video camera constantly captures images of the viewer, sitting in front of the monitor and the W-RVM face detection algorithm introduced in Chapter 3 and summarised in Table 3-6, is used to locate the viewer's face. The coordinates of the detected face on the camera picture allow the calculation of the viewer's three-dimensional position in relation to the camera. Using a sequence of geometric transformations, the viewer position is evaluated in relation to the virtual space, where a direction vector is calculated from the avatar to the viewer's position. The direction vector defines the required head and eyes rotation to align the gaze of the avatar directly to the viewer. This will create the impression of the avatar's gaze following the viewer.

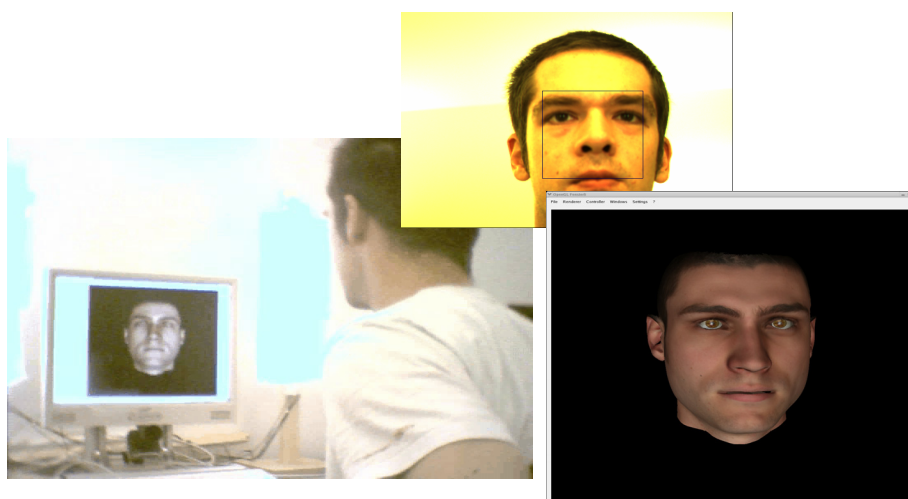


Figure 4-11: Avatar following with eye and head motion

A person is sitting in front of the monitor (*left image*). A video camera below the monitor constantly captures images of the viewer and the W-RVM face detector locates the person (*top*) in real time. An animated 3D model of a human head (*avatar right*) follows with the eyes and head automatically the face of the human viewer. The human viewer has the impression to be “watched”.

Figure 4-11 shows the framework of our first project that achieves a basic inaction loop. The human viewer moves in front of a camera, which is placed below the display, and the monitor shows the 3D modelled avatar face. The computer extracts information from the human. The avatar is reacting to the face of the person by moving the head and eyes and the human “reacts back” by the impression of being “watched”.

The 3D face model of the avatar is created taking advantage of the 3D MM and uses the W-RVM on the 2D images to locate the human viewer. Therefore, this project demonstrates how to create applications taking advantage of the unification of a 3D face model and a 2D appearance model, as described at the beginning of this chapter (Figure 4-1).

4.3.3. Switching Faces – a Perception Psychological Installation

InFaFeDe – Interactive Fast Face Detection

We implemented a Free Frame interface for our W-RVM detector. Free Frame [33] is an open-source cross-platform for real-time video effects. With Free Frame, we use a plug-in system, which is a common interface for visual programming languages, e.g. Free Frame interfaces can be used for Adobe plug-ins.

The Free Frame interface was implemented for the graphical programming language VVVV. The intention of the cooperation with J. Diessl and B. Groß was to realise a media installation (proposed at the next section and [25]).

The Interactive Fast Face Detection application (InFaFeDe) uses the Free Frame interface of the W-RVM detector and VVVV as programming language. As seen in Figure 4-12, VVVV can be used for graphical or experimental programming or rapid prototyping (<http://www.vvvv.org>). The face detection interface can be used as a graphical node and can be joined with other function nodes, e.g. In/Output nodes to control the parameter of the face detection function, to colour the face by the detection certainty (blue rectangle), or to display the number of current detected faces (box with “1”). VVVV is convenient for real-time video applications: camera inputs or displays are simple nodes and thus easy to use (Figure 4-12,

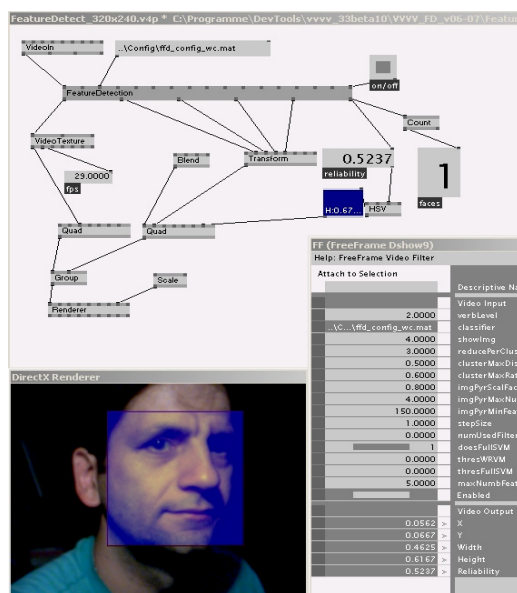


Figure 4-12: IaFaFeDe - Interactive Fast Face Detection

bottom left). The parameters for each function can be comfortably controlled by the inspector view (bottom right).

The advantage of the interactive interface is that it is applicable, as in this installation, for projects not in the field of computer science. For VVVV no imperative programming knowledge is needed. In addition, the interface can be used for experimental programming or rapid prototyping. The parameters of the W-RVM approach can be demonstrated: For instance, non-experts can optimise the detection with respect to the trade-off between accuracy and runtime performance for specific environments.

Switching Faces – HCI Application in the Field of Perception Psychology

The Free Frame interface of the W-RVM face detector, used in InFaFeDe, could further be used for an installation, demonstrating aspects in the field of HCI and Perception Psychology.

The project was developed in cooperation with the Academy of Art and Design Basel, University of Applied Sciences Northwestern Switzerland [57], [38]. It was first demonstrated at the opening of the Eikones Building in Basel, Switzerland as part of the NCCR project: "Iconic Criticism – The Power and Meaning of Images". In the century of "the digital revolution" which has created a "new, image-based society", the NCCR project tries to answer: "How do images create meaning – in science, in everyday life or in the arts? How are they influenced by them and how, conversely, do they influence them? What is their inherent vital power?" For these intentions, our face detection installation tried to demonstrate that vision is not only an optical process, but also rather a perception process, realised in the human brain. The relatively simple installation demonstrates this aspect in a paradoxical or even perplexing way. This perception part of the human brain is taken over by the computer, to confuse or even manipulate this process. If a viewer steps into the observation area of the camera, he is recognised as having a human face and caught by zooming into the face on a large screen, e.g. beamed on a wall. He is tackled by the mechanism as long, as he does not leave the area or a second person is coming in. The installation relieves the single person, but is now isolating the faces and switching them between all human viewers within the observation area. By placing a photograph in the camera view, the faces are also switched between viewers and photographs. Thus, the viewer can feel its "identity" switched to a well-known person or take over their appearance.

We presented the installation in public, e.g. at the Berlin Long Night of Science at the Konrad Zuse Institute and at the Open Day at the University of Basel, Department Computer Science, incorporated with the informatica08 (Year of Informatics, <http://www.informatica08.ch>). The popularity of the installation, the responses of enjoyment, and the reactions, which are sometimes perplexing or even reflective, are surprising.

Some snapshots are shown in Figure 4-13. The viewers enjoy switching into a “new identity” or going on a “time travel”, if e.g. a father gets a younger face like that of his son, and the son gets an idea of how he could look in the future (top right). The left bottom image shows the developers of this project. This example demonstrates that it is confusing to point to the person who is the expert for perception psychology, who for VVVV, and who for machine learning. The question appears: What determinates an individual, the face or the body? “Where” or “what” is the identity of a person?

If a person comes too close or wants to touch the installation hardware, its face is replaced by the “Laughing Man” mask. This is an aphorism from the classic piece of science fiction “Ghost in the Shell”⁹. A criminal, called “Laughing Man”, hacks the perception of the eye implants of all humans and androids. He always digitally replaces his face with the mask, so that the mask becomes his only known appearance. Questions are discussed like, is manipulating perception or an identity allowed?



Figure 4-13: Switching Faces – HCI application in the field of perception psychology

The installation shows how faces can be detected in real time, tracked and randomly switched. At the same time, this HCI installation demonstrates playful and irritating aspects from the field of perception psychology.

The interesting question from the field of computer sciences is, how to detect faces at real time. This can be realised by our proposed face detection system. However, this installation demonstrates playful and irritating aspects at the same time from the field of perception psychology and brings up interesting questions. Maybe that is the reason why it catches attention at presentations.

⁹ see e.g. [http://en.wikipedia.org/wiki/Laughing_Man_\(Ghost_in_the_Shell\)](http://en.wikipedia.org/wiki/Laughing_Man_(Ghost_in_the_Shell))

Chapter 5

Perspective and Conclusion

5.1. Further Unification of W-RVM and 3D MM

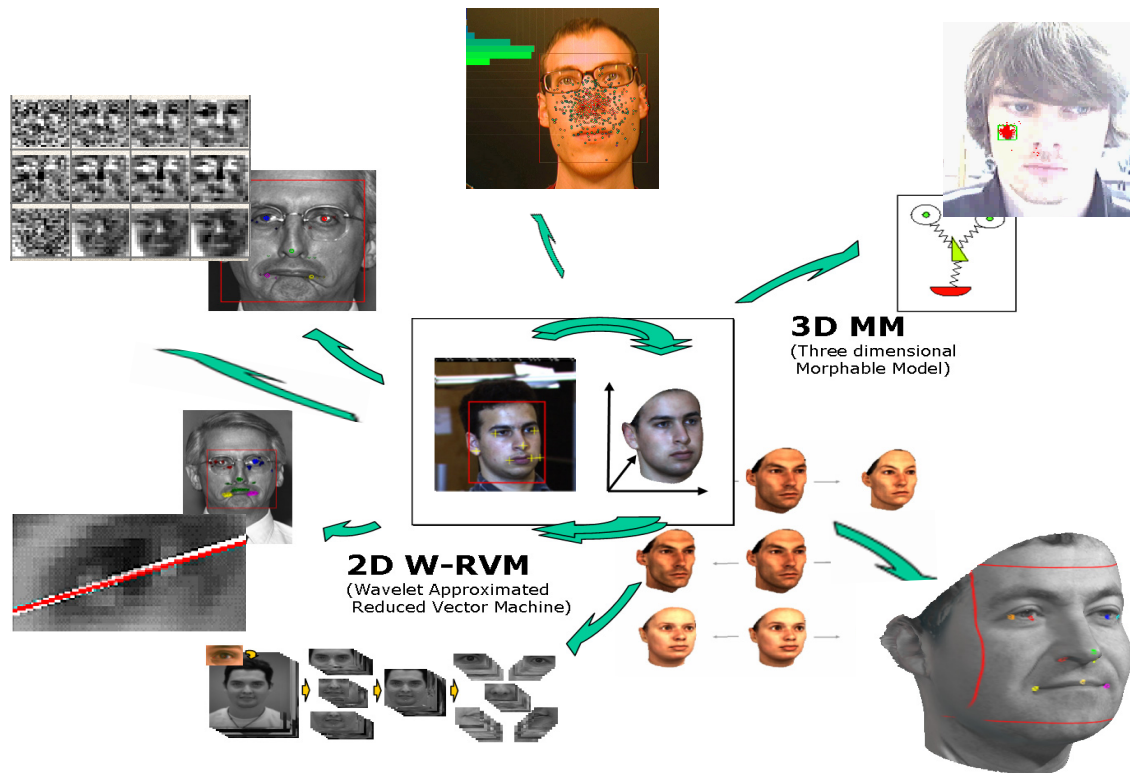


Figure 5-1: Further unification of the W-RVM and the 3D MM-Fitting

The loops of unification of the 2D image-based classifier (W-RVM, centre left) and the 3D face model (3D MM, centre right) form the general background of this thesis. This process will be continued by improving the automatic 3D MM fitting (bottom right); improving the W-RVM, e.g. by a Single-Stage approximation (top left), or adapting the W-RVM approach to Support Vector Regression (bottom left); or by tracking and real-time learning of model parameters (top left).

5.1.1. Morphable Model

In Chapter 2 we proposed the W-RVM; before we describe the unification we will briefly introduce the Morphable Model. The 3D Morphable Model [101], [5], [6], [7] is the core

competence of the Graphics and Vision Research Group (GraVis, University of Basel, [37]). We used in this thesis the Morphable Model Toolbox (MMT) from Romdhani et al. [76], [77].



Figure 5-2: Morphable Model face database

The construction of a 3D Morphable Model requires a set of example 3D faces (e.g., laser scans see Figure 5-2). The Morphable Model used for this thesis was constructed with 200 laser scans acquired by a Cyberware 3030PS laser scanner. The construction is performed in three steps: First, the laser scans are pre-processed. This semi-automatic step aims to remove the scanning artefacts and to select the part of the head that is to be modelled (from one ear to the other and from the neck to the forehead). In the second step, the correspondences are computed between one scan chosen as the reference scan, and each of the other scans. Then a principal components analysis is performed to estimate the statistics of the 3D shape and texture of the faces.

The correspondences enable the formulation of a face space. The face space is constructed by putting a set of M example 3D laser scans into correspondence with a reference laser scan. This introduces a consistent labelling of all N_v 3D vertices across all the scans. The shape and texture surfaces are parameterised in the (u, v) reference frame, where one pixel corresponds to one 3D vertex (Figure 5-3). The 3D position in Cartesian coordinates of the N_v vertices of a face scan are arranged in a shape matrix, \mathbf{S} ; and their colour in a texture matrix, \mathbf{T} .

$$\mathbf{S} = \begin{pmatrix} x_1 & x_2 & \dots & x_{N_v} \\ y_1 & y_2 & \dots & y_{N_v} \\ z_1 & z_2 & \dots & z_{N_v} \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} r_1 & r_2 & \dots & r_{N_v} \\ g_1 & g_2 & \dots & g_{N_v} \\ b_1 & b_2 & \dots & b_{N_v} \end{pmatrix} \quad (5.1)$$

Having constructed a linear face space, we can make linear combinations of the shapes, \mathbf{S}_i , and the textures, \mathbf{T}_i of M example individuals to produce faces of new individuals.

$$\mathbf{S} = \sum_{i=1}^M \alpha_i \cdot \mathbf{S}_i, \quad \mathbf{T} = \sum_{i=1}^M \beta_i \cdot \mathbf{T}_i \quad (5.2)$$

The knowledge about the appearance of faces is represented by our Morphable Model, and can be used to modify novel image about which no information of the 3D structure of the face is available.

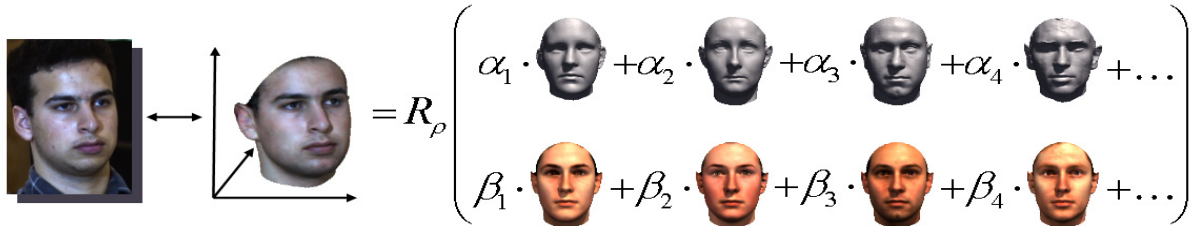


Figure 5-3: Morphable Model – Learning from examples

Let us assume we are able to reconstruct a face in an image through our Morphable Model. That means we find the coefficients in our Morphable Model on the right side of Figure 5-3 in that way that the generated face looks identically to the novel face on the left. Then we can map all the valid face variations learned from our example faces onto the novel image.

This automated matching procedure is an optimisation problem. In order to reconstruct the face we not only have to find the model coefficients α, β we also need to estimate all the rendering parameters, R_ρ . The rendering parameters model the head orientation, the projection into the image plane, and the illumination conditions.

To solve this optimisation task we use a stochastic gradient decent algorithm. To stabilise the optimisation we not only minimise the difference between reconstruction (right-hand side of Figure 5-3) and the target image (left-hand side), in parallel we also maximise the posterior probability of the model coefficients.

To evaluate the linear combination of face examples we could just simply add the texture and radius values point by point. Depending on the location of the features in the scans, we would end up with two mouths and four eyebrows, see Figure 5-4. However, if we always add 3D coordinates and texture values of points that belong together, such as the corner of the mouth, we obtain a proper human face as a 3D morph of other faces.

How can we find corresponding points in a pair of face scans? The simplest way is that a user manually clicks on a number of points, and the correspondence in between is found by interpolating. The Morphable Model dense correspondence is found automatically, but prior

to this thesis, a user had to manually click on a number of landmark points to initialise the fitting procedure.

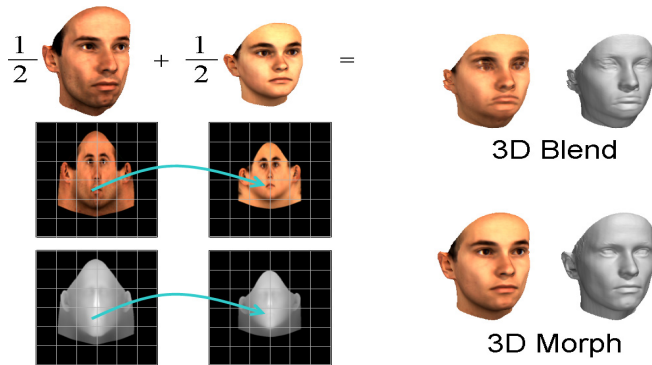


Figure 5-4: Registration

Many changes in the appearance, such as pose or illumination, require some kind of depth information of the scene. The Morphable Model represents faces explicitly in 3D, using textured 3D models in high resolution. In most applications a 3D scan of the person is not available, so a crucial element of the Morphable Model algorithm is automatically to estimate 3D shape from a single image.

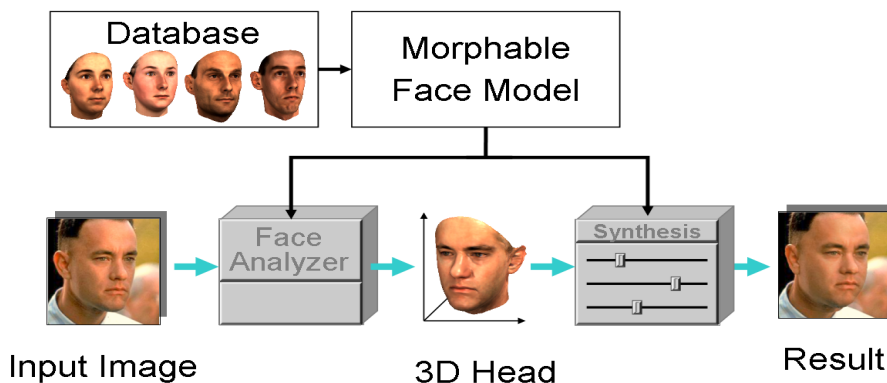


Figure 5-5: Analyse by Synthesis

The Morphable Model can be used for an interactive modeller tool where a wide range of relevant attributes of faces can be controlled, given nothing more than a photograph or a video frame. This model can be used for synthesis of many representations of variations within the class of human faces or to change the appearance, such as pose or illumination. The renderings can be used as an example set for the training of a 2D appearance model, like the W-RVM introduced in Chapter 2. On the other hand, the W-RVM can be used to overcome the manual labelling of the landmark points. How this unification can be obtained is detailed in the following sections.

5.1.2. Automatic Anchor Point Detection for the MM-Fitting

The fitting function of the MM is fully automatic except the manual labelling of the anchor points. If we were able with help of facial feature detection to find landmarks usable as anchor points, it would be a significant improvement. The detection can be used to enrich the fitting algorithm of the Morphable Model by providing the locations of facial features, like top of the nose, mouth corners, eyes, bridge of the nose. This could be applied to automate the fitting algorithm and to improve the starting-point approximation. That would facilitate to use larger non-labelled database, where the manual labelling would take too much effort. The manual initialisation also needs in some degree expert knowledge about the MM fitting.

The framework of the MM fitting has at the moment three steps: The first is the interactive anchor point setting with the help of a GUI, then the fitting function is called and at the end, some renderings of the fitting result are done for verifying the fitting [77]. This framework will not change significantly. Only the step of the manual anchor point clicking is exchanged by the facial feature set detection based on the W-RVM detectors and the PSM for finding the final feature assortment. The framework for the facial feature set detection approach has three stages. In the first stage, the W-RVM face detector is applied for the full image (Table 3-6). In the second stage, we define within the detected face areas a FOI for the facial features (Section 3.5.2) and apply all single facial feature detectors (Table 3-6) within their regions of interest. Figure 5-6 shows the stages of the feature detectors. In difference to Chapter 2 and 3, we use lower threshold sets for the first classifier stages to provide more candidates for the PSM function. Therefore, the clusters (Figure 5-6, first and second image, from left to right) are larger and a higher FAR is obtained for the last detector stage (right).

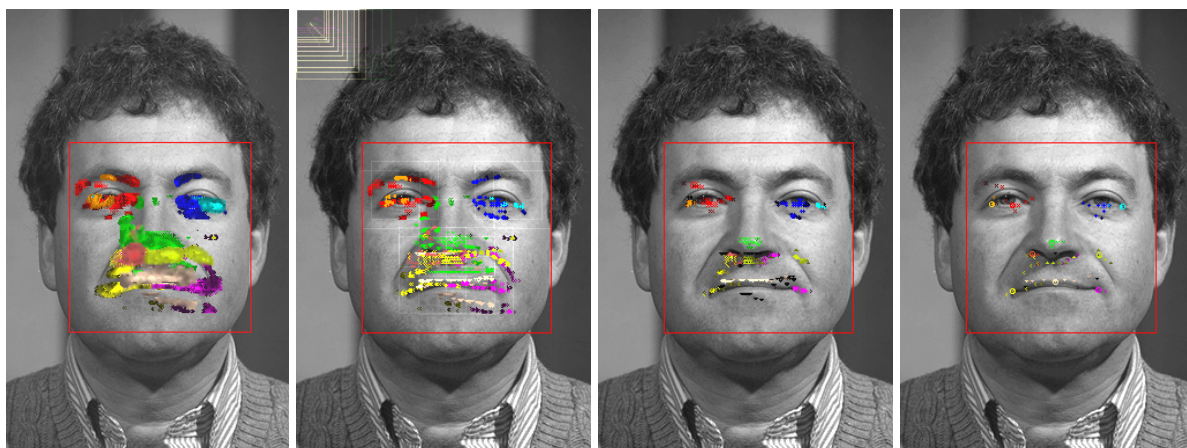


Figure 5-6: Second stage of the W-RVM facial feature set detector for ten features

The face detector result (*red box*) and the four stages are shown for ten facial features (The ten features, their *colours*, and *marker* are defined in Table 3-5). The *1st* (from left to right) image shows the result after the first fast filter stage of the W-RVM's, the *2nd* the results after the *1st* overlap-elimination, the *3rd* after applying the full SVM for the remaining patches and the *4th* the final result after the *2nd* overlap-elimination. The detections with the highest certainty per feature are *circled* (notice, that we use lower threshold sets for the first classifier stages to obtain more candidates for the PSM function).

In the third stage of the facial feature set detection, we apply as correlation classifier the PSM function (Section 3.6). The function takes as input the lists of coordinates and probabilities of the candidates of all features and the learned model obtained from the 3D MM. As input, the last stage of the facial feature detectors is used as seen in Figure 5-6, right and zoomed in Figure 5-7, left. After applying the PSM, we obtain as output new probabilities. The output of the PSM function is seen in Figure 5-7, middle and the maximum of the probabilities (circled) is used as final feature assortment of the W-RVM facial feature set detection (Figure 5-7, right image; see Section 3.5 for details and Table 3-7 for a survey of the approach).

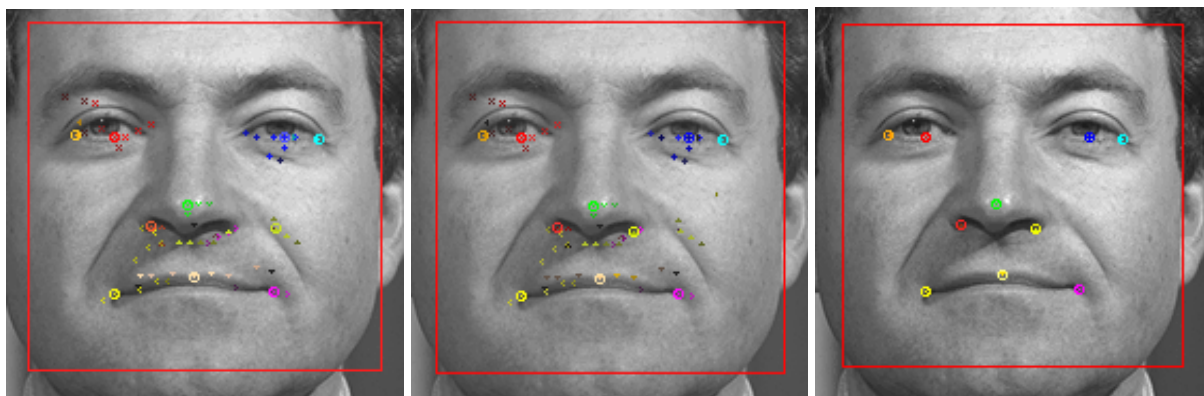


Figure 5-7: Third stage of the W-RVM facial feature set detector using the PSM

The list of candidates of all features from the last stage of the W-RVM feature detectors (*left image*) are the inputs of PSM. The correlation classifier is the last stage of the W-RVM facial feature set detection. After applying the PSM, we obtain as output new probabilities taking advantage of the 3D MM. The maximum of the probabilities (*middle image, encircled points*) is used as final feature assortment of the W-RVM facial feature set detection (*right*). These points can be used as anchor points for the MM-fitting.

The final feature assortment found by the feature set detection can be used as anchor points for the MM-fitting. In Figure 5-8, top row, we show the first results on the example image and compare the fitting results with a MM-fitting based on interactively set anchor points (middle row). We take advantage here of the Morphable Model Toolbox (MMT) from Romdhani et al. [77] and the Multiple Feature Fitting (MFF) proposed there. The first column shows the anchor points found by the W-RVM facial feature set detector (top) and the manual-set anchor points (bottom). A difference is visible for the labels of the eyes: these points are the average points of the left and right eye corners, and not the middle point of the eye area, or of the centre of the pupil. For that reason, they are difficult to set. In the following columns, the five MMF stages can be compared. The bottom row shows renderings the automatic and interactive fitting. Already after the second fitting stage (edge fitting), the results are similar.

The application of the proposed facial feature detection for automatic anchor point detection is not yet finished. Some optimisation issues are not integrated and a suitable statistic validation has to be processed. For instance, the runtime of the PSM function can be improved by using a cascade of the reference sets (see Section 3.6). Furthermore, the usage of the occlusion likelihood is not finished.



Figure 5-8: Comparison of the MM-fittings stages

The *top row* shows the MM-fitting stages on anchor points set by the W-RVM facial feature set detection and the *middle row* the on manually clicked landmarks. The bottom row shows the input image (*left*) and renderings of estimated texture blended into the input image. We demonstrate the automatic fitting results on W-RVM landmarks and results based on manual initialisation with the estimated illumination (*2nd and 3rd images from left to right*) and with a normalized illumination for the estimated texture (*4th and 5th*). Comparing the five MM-fitting stages and the renderings shows that the automatic and interactive fitting have similar results already after the second fitting stage (edge fitting).

However, the examples in Figure 5-8 and 5-9 show that the fitting results are robust to small displacements. This is expected, because the anchor points are mainly used at the first fitting stages.

Labelling the anchor points manually will always be the most accurate method. However, a statistic would be interesting for the fitting of a large dataset using an automatic W-RVM anchor point detection in comparison to fitting results based on manual labelling. In Figure 5-10 first results on example images taken from the FERET database [61] are shown. Each column shows an example of MM-fittings performed on anchor points found by the W-RVM facial feature set detection using the PSM (Figure 5-10 a-c)) and manually clicked landmarks (Figure 5-10, d-f)).

The first example (from left to right) has small differences for the lighting estimation. The lighting estimation is also done in the first fitting stages. Hence, differences at the anchor points have more influence in this phase. The second and fourth example show problems to fit

the ears, because of occlusions (hair style). This problem is not influenced by the anchor points, but can be solved using contour points or an outlier mask. In Section 3.3.1 we chose a set of facial features for the training using a multi-criteria catalogue. One criterion was if the features are suitable for MM-fitting. However, as discussed there the W-RVM facial feature detector will not be the appropriate method for contour points. The first experiments show as expected, that they are important for good fitting results. Hence, contour-based methods, which are more suitable for contour points, should be applied to use not only anchor points at the face. Another opportunity is to optimise the edge fitting at the first fitting stages, because there is already an edge- and model-based search included. The problem to find the contour of the face for the initialisation of the fitting could also be solved by the advanced skin segmentation by Pierrard et al. [60].



Figure 5-9: Comparison of the renderings from the obtained fittings

The *top row* shows the renderings from the fitting based on anchor points set by the W-RVM facial feature set detection and the *bottom row* the fitting results based on manually clicked landmarks. The *1st* (from left to right) *column* shows the fitted 3D shape, the *2nd to 4th* *columns* renderings using the extracted textures with different poses. The anchor points are visualised on the 3D shape with the *colours* and *markers* defined in Table 3-5. Only very small differences can be seen on the shape between the fitting based on automatic found and manual clicked landmarks. The landmarks are mainly used at the first fitting stages. Hence, the 3D MM-fitting is robust to small displacements of the landmarks.

In Figure 5-10, fourth example face, the left eye point (le) is not detected accurately. However, the MM-fitting is robust to small displacements of the anchor points. Even if one anchor point is missing (like the right nose point (rb) at the third example), the fitting result is stable. In our experiments, we figured out that if at least seven anchor points are given, comparable fitting accuracy is obtained to manually set anchor points.

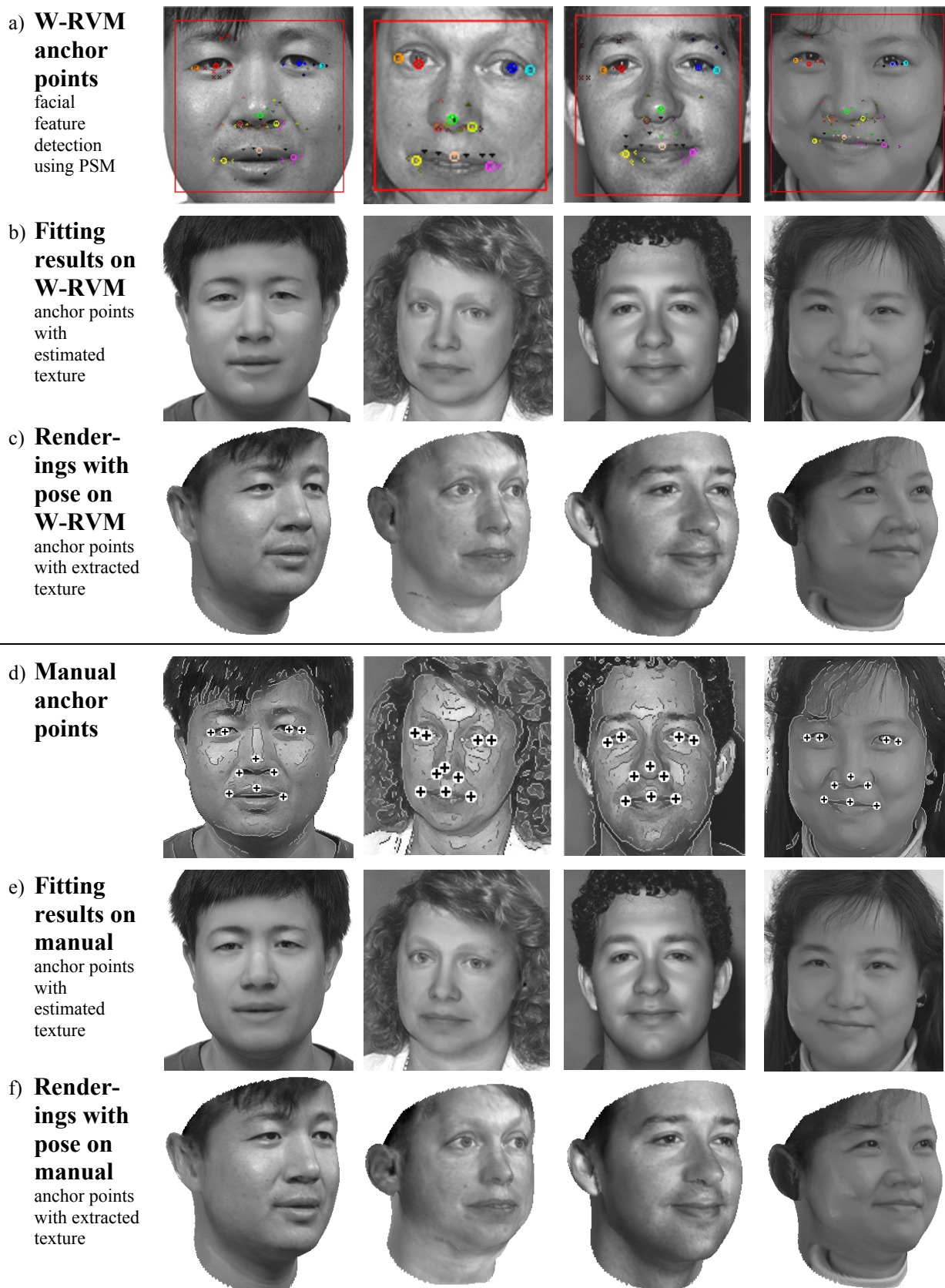


Figure 5-10: Comparison of MM-fittings using anchor points set by W-RVM and manually. Each column shows an example for MM-fittings performed on anchor points found by the W-RVM facial feature set detection using the PSM (a-c) and manually clicked (d-f). The 1st example (from left to right) has small differences for the light estimation. The 2nd and 4th ex. show fitting problems because of occlusions, this problem is not influenced by these anchor points, and can be solved using contour points or an outlier mask. In the 3rd ex. the right nose point (rb) and in the 4th ex. the left eye (le) are not detected accurately; however the fitting is robust. For the first set of images, we obtained a comparable fitting accuracy.

5.1.3. Further Unification W-RVM and 3D MM

The PSM function is applied in Section 3.6 as correlation classifier. The input parameter of the function are the image likelihood given by the 2D appearance model (W-RVM), the shape likelihood based on the 3D MM, and the occlusion likelihood (see Equation (3.16)). Furthermore, in [78] an orientation and size likelihood is provided by the introduced PSM, which are estimated from the output of the SIFT key point detector. As expected, to use the size and orientation of the features improves the recognition rate of the optimal features assortment within all combinations. If, for example, two eye features within one combination have different sizes or opposite orientations, than this is not a likely feature assortment. From experiments by Romdhani et al. an improvement about five percent of the recognition rate of the final assortment can be expected.

For this reason, we started a project applying the W-RVM approach for the approximation of regression functions (see Section 5.2.3). For the estimation of the orientation of the eyes (in-plane rotation angle) we gained promising results. These regression functions have to be very efficient, because the regression must be applied for the size and the orientation of about ten features (for all candidates per feature included in at least one combination with a valid projection matrix). Therefore, standard Support Vector Regression would not be sufficient. In Section 5.2.3 it is shown that using the W-RVM approach efficient regression is gained. Taking advantage of the 2D appearance model the 3D model function (PSM) can be improved as a further loop of unification of the 2D W-RVM and 3D MM-based approach.

Another opportunity to take advantage of further unification of the two models was mentioned in Section 3.5. The FOI for the feature detectors is defined empirically. The idea is first to transform the area of the detected face to the 3D face space. Then to evaluate the probability, with the help of the 3D MM, that a feature could be located at each of the points within that area. After projecting back the probability map for the detected face area to the 2D coordinates, we can define for a given probability limit a more precise ROI for each feature. This would optimise the efficiency and improve the accuracy by reducing the FAR.

Several opportunities appear if we use the 2D appearance and 3D model in more than one loop. With at least four points of the face, we can evaluate areas where missing facial feature points can be expected. For instance, features not detected at the first loop because of poor image quality or lighting conditions. A more complex detection can be applied in rather small uncertainty areas with reduced size (see Section 3.6 and uncertainty areas in Figure 3-35).

In an improved framework, the features could be detected in loops. After, e.g., the first ten features are found using the introduced FOI, uncertainty areas with reduced size can be

obtained by the PSM. Then the next set of features like *sci*, *g*, *li*, *en*, and *sba* (see Table C-4 and Section 3.3.1) can be detected more efficiently or more-complex classifiers can be used for these more problematically or more ambiguously to detect features.

However, not only the search areas can be reduced: If the pose or the light is estimated after detecting the first feature points and using the first fitting stages of the MM, a normalisation could be applied at the image before searching for the next portion of features. For instance, after the focus length is estimated (based on the projection matrix by the PSM), not all scales of the pyramid image have to be considered for the next features. Additionally, if the in-plane rotation angle is estimated or the light source then the image can be normalised by the orientation or the illumination. The advantage is that the feature classifiers can be trained on normalised data, and therefore smaller training sets. The reduction of the hypothesis space, e.g. concerning the in-plane rotation angle or the variety of light conditions, improves significantly the complexity and generalisation performance of the classifiers.

On the other hand, with a higher number of facial features or more precise facial anchor points the fitting of the MM can be improved. Already the initialisation of the fitting or the first fitting stage is more accurate, yielding better fitting results. Alternatively, for the PSM more PC for the specific subject can be estimated (see M in Equation (11) in [78]), which improves the finding of the assortment of the final features points, again.

These are some opportunities of the ongoing process of the unification of the two models symbolised by circled arrows in Figure 5-1.

5.2. Relevance of the W-RVM Hyper-plane Approximation

5.2.1. Single-Stage W-RVM

This is a project in cooperation with Gerd Teschke at Konrad Zuse Institute Berlin, Germany, Numerical Analysis and Modelling Department and University of Applied Sciences Neubrandenburg, Germany, Department for Signal and Image Processing, Ill-posed and Inverse Problems, Geomathematics.

In Chapter 2, we introduced the W-RVM as a two-stage approach. The machine is trained by a hyper-plane approximation from the original SVM to the Reduced SVM (RVM) and from the RVM to the Wavelet Approximated Reduced Vector Machine (W-RVM). The first reduction step reduces the number of vectors and the second the complexity of the vectors. The idea is now a single-stage computation of the Wavelet Approximated Reduced Vectors

(W-RSV's) as a complexity reduction of Support Vector Machines based on a simultaneous sparse approximation of hyper-plane and set vectors.

The core idea is described in the following: The W-RVM approach is concerned with Support Vector Machines (SVM) and its minimal and sparse approximation. Such a sparse approximation may accelerate the classification process impressively. The algorithm allows a simultaneous computation of both a sparse version of the underlying set vectors and an associated minimal approximation of the hyper-plane (which induces the classification). We proved that this algorithm will always converge in norm to one optimal reduced SVM.

In accordance with the goal above, we aim to minimise the following cost functional

$$J(\mathbf{x}, \mathbf{y}) = \|\Psi(\mathbf{x}, \alpha) - \Psi(\mathbf{y}, \beta)\|_F^2 + 2\gamma_0 \|\beta\|_{l_p, \sigma_0} + 2 \sum_{i=1}^N \gamma_i \|\mathcal{W}\mathbf{y}_i\|_{l_p, \sigma_i} \quad (5.3)$$

where $\mathcal{W}\mathbf{y}_i$ denotes the wavelet expansion of \mathbf{y}_i . The subsequent analysis is applicable for arbitrary one homogeneous and convex penalty terms. But typically sparsity is achieved when $p < 2$. The case $p < 1$ amounts to non-convex penalties. The resulting non-uniqueness does not cause problems, but when dealing with iterative methods, convergence is an open problem. To this end, we limit the analysis to the case $p = 1$. In the field of image patches, this makes sense since it allows the use of very fast image integrators (as long as we search for sparse rectangular, Haar-like representations of the patches). Since the minimisation is also with respect to the set vectors \mathbf{y}_i , the data misfit term is non-convex (since Ψ acts nonlinear on \mathbf{y}). This requires the application of minimisation techniques for nonlinear operator equations. Here we adapt a scheme, quite recently developed in the habilitation (postdoctoral lecture qualification thesis) of Gerd Teschke that fits into our framework with sparsity constraints.

The general idea goes as follows: construct a sequence of surrogate functionals from which we know that they provide unique solutions that are relatively easy to compute (assuming twice Frechét differentiability on Ψ , the surrogate functionals are even strictly convex). Moreover, it is shown that under certain assumptions on the operator $\Psi : X^N \times \mathbb{R}^N \rightarrow F$ and properly chosen iteration parameters, the sequence of minimisers converges at least to one critical point, i.e. it converges towards one optimal Reduced Support Vector Machine.

The iterative method is constructed by defining the surrogate functional

$$J^S(\mathbf{y}, \beta; \mathbf{a}, \eta) := J(\mathbf{y}, \beta) + C \|\beta - \eta\|^2 + C \sum_{i=1}^N \|\mathbf{y}_i - \mathbf{a}_i\|^2 - \|\Psi(\mathbf{y}, \beta) - \Psi(\mathbf{a}, \eta)\|_F^2 \quad (5.4)$$

The sequence of iterates tending to one optimal/minimal Support Vector Machine is then created by

$$(\mathbf{y}^{n+1}; \beta^{n+1}) = \arg \min_{(\mathbf{y}; \beta)} J^s(\mathbf{y}; \beta; \mathbf{y}^n; \beta^n) \quad (5.5)$$

The goal is to improve the performance of SVM classifiers. Computational speed is for example in the field of face detection in image movies required. At first experiments, we reduced a small SVM trained on faces. In Figure 5-11 we show some examples of the fix-point iterations of the first four W-RSV's. The algorithm converges to a reduced SVM, where the W-RSV's are not longer a subset of the Support Set Vectors.

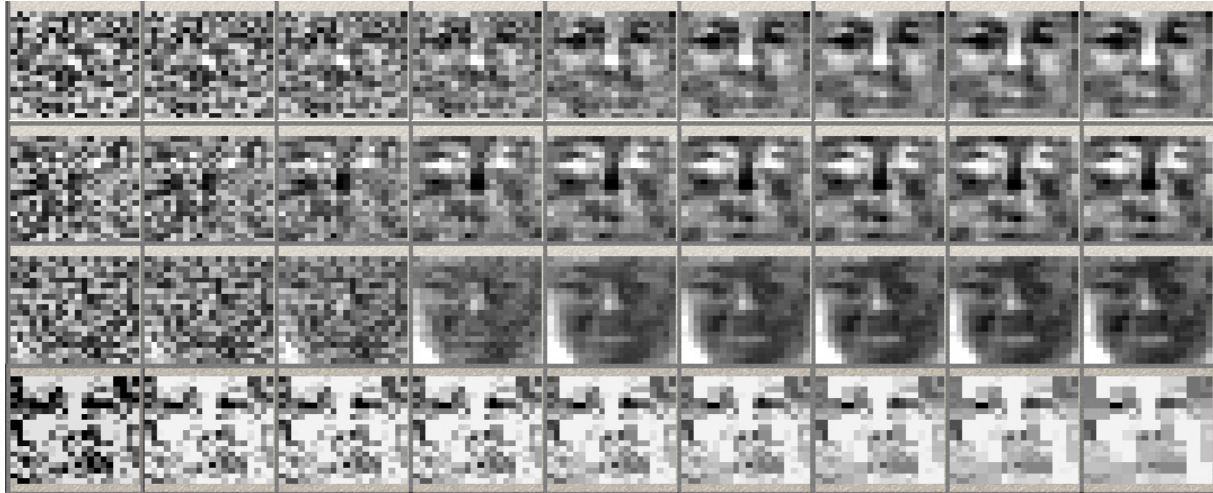


Figure 5-11: Fix point iteration for Single-Stage W-RVM

5.2.2. Multi-feature and Multi-invariant W-RVM

One of the principles of the W-RVM is the early rejection of easy-to-discriminate vectors. It is obtained by double-cascaded evaluations over coarse-to-fine Wavelet Approximated Reduced Set Vectors (W-RSV's). The used cascade for an early rejection of non-objects is a specific, maximal deep tree. However, a well-balanced tree would be much more efficient. The idea is to use general and more invariant W-RSV's in upper nodes of the tree and more specific for deeper nodes.

For example, we want to apply a classifier tree for the left and right mouth corners as seen in Figure 5-12, left. It is achievable to train W-RSV's, able to discriminate left and right mouth corners from vectors not located at these feature points. To build a tree of classifiers is proposed, e.g. in [81]. The improvement would be to generate the classifier tree automatically. One strategy is to compute W-RSV's (using Table 2-2) for an upper node as long as the number of operations per rejection of non-feature vectors is lower than the number of operations per rejection needed by the more specific deeper nodes; otherwise, the generalisation of the upper node is finished. This bottom-up strategy is finished when no upper node (with at least one W-RSV) can be extracted anymore. Instead of the rejection rate, also the

decrement of the hyper-plane distance per operation can be used. Figure 5-12 shows how W-RSV's of the upper node would look like for this example.

The generalisation performance of an SVM is correlated to the size of the hypothesis space. For instance to classify all (right and left) mouth corners by only one SVM is not achievable. This is handled in praxis by applying different classifiers sequential. However, training a tree of classifiers (one for the left, one for the right and one for both) is more efficient and enables to build a multi-class classifier.

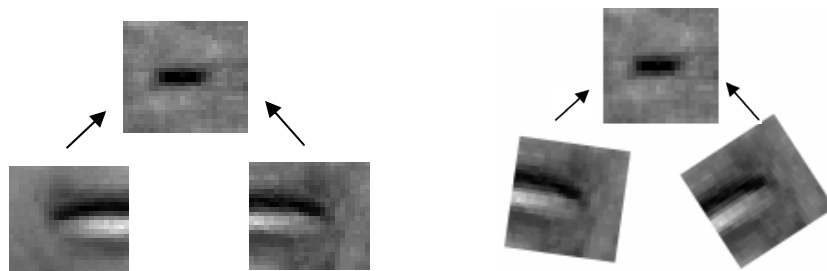


Figure 5-12: Multi-feature and multi-invariant W-RVM

Using W-RVM's for each feature sequentially is not efficient. We can evaluate Reduced Set Vectors for some features jointly, e.g. for both corners of the mouth (*left*), before the specific classifiers are used. The same strategy can be used for multi-invariant classifier, e.g. for a large range of in-plane rotation, *right*.

We can use the same approach for multi-class or multi-invariant classifier. For instance in Figure 5-12 right, if the range of the in-plane rotation angle of the feature is too large for one classifier, different classifiers are trained to handle this classification problem. Again, an upper more general classifier node can be extracted. This way more-invariant classifier for different poses, rotations, or scales can be build, working efficiently by starting with more invariant classifier nodes.

At our experiments, we tried to train such more general classifiers as seen in Figure 5-12 and discussed the results. The approach used for a multi-invariant classifier is implemented by one of our cooperation partners using one general classifier before three classifiers with different ranges of in-plane rotation. This improved the runtime performance significantly, because the upper node is rejecting 94% of the non-feature vectors and only 6% have to be evaluated by the deeper nodes.

5.2.3. Wavelet Approximated Vector Regression – W-RVR

Support Vector Regression [88], [86] can be used to estimate functions. For example, we want to train a regression function for the in-plane rotation (roll) angle of the eyes within a face. In Section 3.6, it is figured out that the detection rate for the correlation classifier could be improved using additional an orientation and size likelihood for the PSM. Therefore, we want to estimate the orientation and size of the facial features and use as further input

parameters of the PSM. The evaluation function for regression (5.7) is similar to the decision function of the SVM

$$y(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^{N_x} \alpha_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b \right) \quad (5.6)$$

$$\tilde{y}(\mathbf{x}) = \sum_{i=1}^{\tilde{N}_x} (\tilde{\alpha}_i - \tilde{\alpha}_i^*) k(\mathbf{x}, \tilde{\mathbf{x}}_i) + \tilde{b} \quad (5.7)$$

Identically we have to compute the kernel function for the current patch \mathbf{x} and all Support Set Vectors \mathbf{x}_i ((5.6) or (2.1) in Section 2.1) or respective with all Support Regression Vectors $\tilde{\mathbf{x}}_i$. With the same theoretical background, an approximation of the hyper-plane function for regression can be used as introduced in Chapter 2. That means we substitute the Support Regression Vectors (SRV's) by Wavelet Approximated Reduced Regression Vectors (W-RRV's). For regression, we cannot use an early rejection rule as for the W-RVM. Nevertheless, for the training stage we can use the same residual approach by stepwise reducing the hyper-plane distance. The only difference is that in the working stage we always use the full set of approximated vectors. The learning stage will be without changes and it can be used identically as in Section 2.6.

Performing the training, Frank et al. [31] was able to take advantage again from the 3D MM by generating synthetic training and validation data. This was substantial because labelled data would not be available for this issue. In addition, we improved the generation of synthetic data and scrutinised the difference between the classes of synthetic data (see Section 3.1).

We trained the Support Vector Regression (SVR) function for the roll angle, then evaluated for these vectors a reduced set of vectors by the RVM approach by Schölkopf et al. [87] and approximated these vectors by the W-RVM method summarised in Table 2-2.

In the experiments, Frank et al. was able to prove the accuracy and efficiency of the novel Wavelet Approximated Reduced Vector Regression (W-RVR) algorithm, applying the W-RVM approach. In Figure 5-13 results of the roll angle estimation are seen. An error between the estimation of the roll angle (red lines) and the ground truth (white) lower than one degree cannot be recognised manually. For an error less than three degree, it is not trivial to decide if the label or the estimation is more realistic. Thus, we defined five degree as an error bound for a valid estimation.

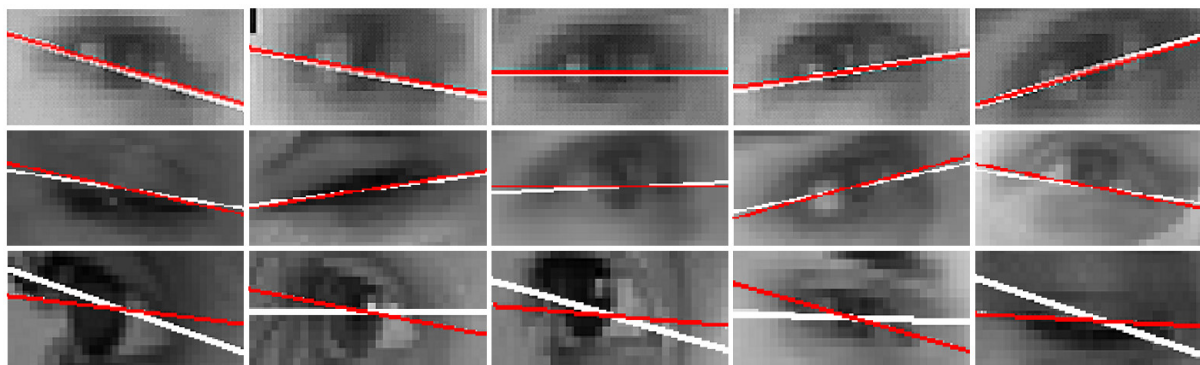


Figure 5-13: Accurate W-RVR results estimating the roll angle of eyes
 The top row shows examples where the error of the Wavelet Reduced Vector Regression estimation of the roll angle (red line) and the ground truth (white) is less than 1° . In the middle row are examples with an estimation error less than 3° , obtained even for closed eyes or examples where the pupil is not centred. For an error less than 3° , it is not trivial to decide if the label or the estimation is more realistic. The bottom row shows examples with an estimation error of 5° . Most of these errors came from too large yaw angles (first three examples from left) or imprecise ground truths (two examples right).

In Figure 5-14, it is shown that we can approximate the SVR function for no significant loss of accuracy by a 11- to 28-fold speed-up. To achieve the error bound of 5° for at least 95% of the estimated roll angles Wavelet Approximated Vector Regression based on the W-RVM approach provides a 75-fold speed-up over Support Vector Regression.

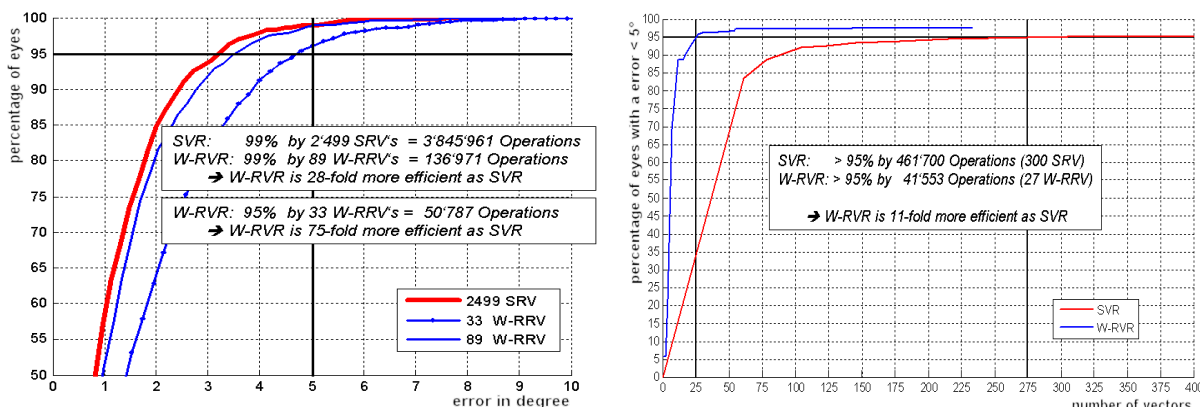


Figure 5-14: Wavelet Appr. Vector Regression is 11- to 75-fold more efficient as SVR
 Left: To gain for 99% of the features an estimation error less than 5° the W-RVR needs only 89 Wavelet Appr. Regression Vectors (W-RRV's: left, plain blue curve, $1.4e5$ operations) instead of 2,499 Support Regression Vectors (SRV's: thick red curve, $3.8e6$ opr.). To estimate 95% of the roll angles with less than 5° error only 33 W-RRV's (blue curve, dot marker, $5e4$ opr.) are used.
 Right: If we reduce the SRV's (by adjusting the generalisation parameter), the SVR needs 300 vectors to gain for 95% of the eyes an error less than 5° (SVR: red curve right, $4.6e5$ opr.). The W-RVR uses only 27 vectors to gain the same accuracy (W-RVR: blue curve, $4.2e4$ opr.). The W-RVR achieves a speed-up by a factor of 11 to 28 compared to the SVR.

Our objective was to train efficient regression functions to improve the PSM function used in Section 3.6. Our experiments above for the orientation estimation for the eyes are promising. We can improve the PSM and therefore the detection rate of the facial feature set by training the W-RVR estimation for the orientation and scale parameters. Even using the regression functions several times (on all candidates for all features) can be done in real time by applying the W-RVM approach to Support Vector Regression.

Another interesting application is to use the Wavelet Approximated Vector Regression approach to learn an aging function based on the regression method proposed in [82].

Adapting the W-RVM approach to regression shows impressive the relevance of the W-RVM for hyper-plane approximations. This gives the thesis an interesting novel theoretical background not limited to the field of Support Vector Machines and classification problems.

5.2.4. Tracking of Higher Feature Parameters

In Section 4.3.1 we introduced face tracking and tracking of a facial feature using the W-RVM approach for the measurement function of the Condensation [55], [52], [53] algorithm. This project was realised in cooperation with the FHNW Basel, Computer Science Department, Prof. Hudritsch [13] and based on "Face Motion Tracking with web cams" [12].

At the end of this project, we could track a feature vector including the x and y coordinates (translation of the object). In addition, the size (or respective the distance to the camera), the rotation angles, or even higher feature like a lightning estimation or parameter describing roughly the subject could be tracked in real time. Therefore, we realise the 3D tracking project. Julian Batliner [3] re-implemented the Condensation with an open architecture. To ensure extensibility and adaptation of the framework, the classes for Prediction and Measurement were designed as open interfaces. Therefore, the tracking is unproblematically applied for other objects, prediction, or new selection policy. As an example, we used as a measurement function the W-RVM for faces and for eyes. In addition, the feature vector can be extended. Currently, the three-dimensional vector of the x and y coordinates, and the distance of the object to the camera are tracked.

As a novelty to the Condensation approach, we found new strategies for the prediction stage through a dynamical and adaptive evaluation of the prediction. We realised the constant diffusion matrix as described in [53], and a novel dynamic diffusion matrix computed from the covariance matrix of the sample feature vectors each frame new. In addition our prediction is adaptive, because it diffuses samples with a high object probability (obtained by the W-RVM classifier) less than samples with a low probability. Comparing the approaches, we obtained a more stable tracking.

Tracking a higher-dimensional feature vector speeds up the tracking. We obtained a speed-up to a detection method up to factor ten. In addition, the localisation accuracy improved, because the density near the object is higher, so more effort for accuracy is spent there. Therefore the core idea is the same as for the W-RVM approach to spend high computational complexity only for difficult-to-discriminate areas of the hypothesis space. This idea can be continued by adding new features to the tracking procedure, e.g. the in-plane rotation, or the

pose of the head. It would be also possible to track abstract algorithm parameters (like the order of the used W-RSV's or parameter of the MM fitting) and end up with a real-time learning system (see next section).

Here we demonstrate an expansion to the tracked feature vector for a third dimension, which is an improvement to standard methods tracking only object translations. A histogram over distance (respective the samples per scale of the used image pyramid, see [79] or Figure 2-2) visualises in Figure 4-14 the distribution of the samples on the third dimension.

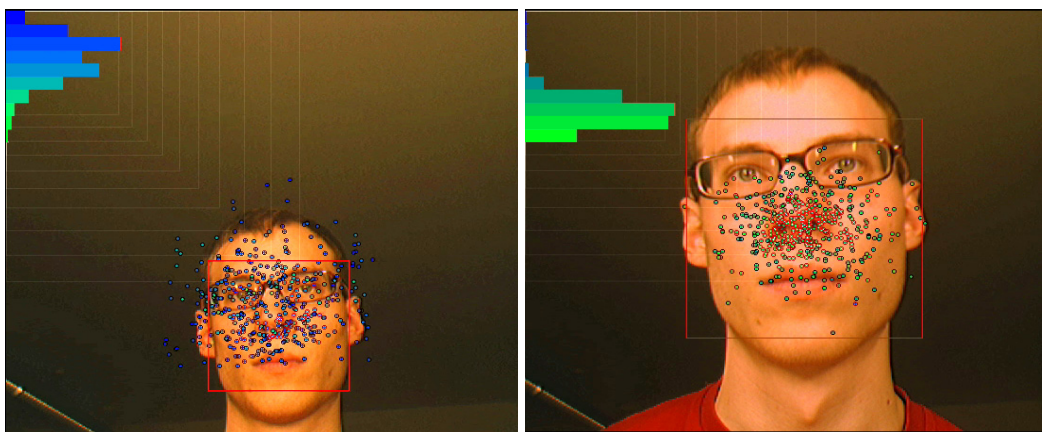


Figure 4-14: Tracking of higher feature parameters

Tracking of faces with a three-dimensional vector is demonstrated. The x and y coordinates and the distance to the camera are tracked. *Blue* circles (*left*) visualise face-centre locations for samples with a high and *green* (*right*) with a small distance to the camera. The distribution of the samples on the third dimension is also shown with a *histogram* of the samples over the distance. The *red box* shows the object used to compute the drift.

Open issues are multi-object tracking (track more than one object of the same object class, e.g. more persons within the image area, or two eyes within a face) and multi-class tracking (tracking of different objects, like the facial features of the face). A new project tackling these open issues has already started and is ongoing. This project uses a dual-camera system able to track a larger area of interaction in front of the camera using a Pan/Tilt/Zoom camera. Thus, a high resolution for the object of interest is obtained.

5.2.5. W-RVM Real-time Learning

As discussed in the previous sections one core idea of this thesis is to spend less computational effort in easy-to-discriminate feature space areas and only a high complexity for areas where the objects of interest are supposed to be located with a higher probability.

The Wavelet Approximated Reduced Vector Machine uses more resolution levels (cascade over approximation levels for the vectors to describe the decision hyper-plane) for patches close to the decision hyper-plane. The second cascade of the W-RVM contracts the number of these incorporated vectors (cascade over number of used W-RSV's). This optimises the

computational complexity per pixel location. The tracking contracts the density of the probability distribution to pixel areas near the observed object. This is an improvement to detection methods, where the probability is evaluated on a grid with constant resolution. Additionally, the 3D tracking contracts the distribution on scales close to the observed object. At the next abstraction level, we worked on the requirement to use several classifiers sequentially, if more than one object class is searched in images (or respectively if more than one classifier must be used, because of the limited invariance, e.g. a too-large range of in-plane rotation). To contract the computational effort near the objects of interest over a set of classifiers, we introduced multi-class and multi-invariant classification trees (Section 5.2.2). As mentioned in the previous section a multi-dimensional feature tracking tackles the same problem. The density distribution estimation is contracted, e.g. on in-plane rotation angles similar to the observed object.

The introduced tracking and the multi-feature tracking approaches already start to learn from the previous frames where to spend more computation cost and where less effort is sufficient. However, the used classifier do not work dynamically and still search, e.g., for the full range of faces under all conditions. Real-time learning in this sense would mean to contract the computation of the hyper-plane on locations of interest, learned from the history. That means to reduce dynamically the number of incorporated W-RSV's and their resolution levels using the last frames.

Our purpose is not only a speed-up of the algorithms, but to concentrate computational power, where complex decisions are required. This makes the algorithms faster for easy to discriminate areas of the hypothesis space, but at the same time more accurate at sensitive areas.

For a reduced hypothesis space, the classifier does not have to tackle so much invariance and is thus faster or more accurate by real-time learning methods. The hypothesis space can be reduced, e.g. by tracking additional the orientation of the face, the lighting condition, or other environment conditions, and further by learning individual parameters of the observed subjects. Model parameters obtained by first fitting stages of the 3D MM fitting can be used to concentrate on W-RSV's correlated to that subclass. Alternatively, the cascade of used W-RSV's could be reordered taking in account the learned relevance from the history.

To apply real-time learning strategies is inspired by the project in cooperation with the FHNW Basel [13]. To speed up their tracking algorithm they implemented in the former project [12] a simple L2-norm classifier. The robustness of that simple but dynamic measure is demonstrated in the left image at Figure 4-15. For a drawing application only a black round marker stuck on the fingernail is used. Clicking once at the image a template is cut out (Toolbox "Set Capture"). A Condensation tracking is applied using as simple measurement function the L2-norm of the particles (red dots) and the template. The maximal response

(green box around the fingertip) is used as coordinates for the virtual red pen on the drawing image (top left). The right image shows the real-time learning for tracking a round marker stuck on a face. If the response of the L2-norm is higher as a threshold the template is updated by the current underlying image area, the last two dynamic templates are shown at the left bottom corner of the display.

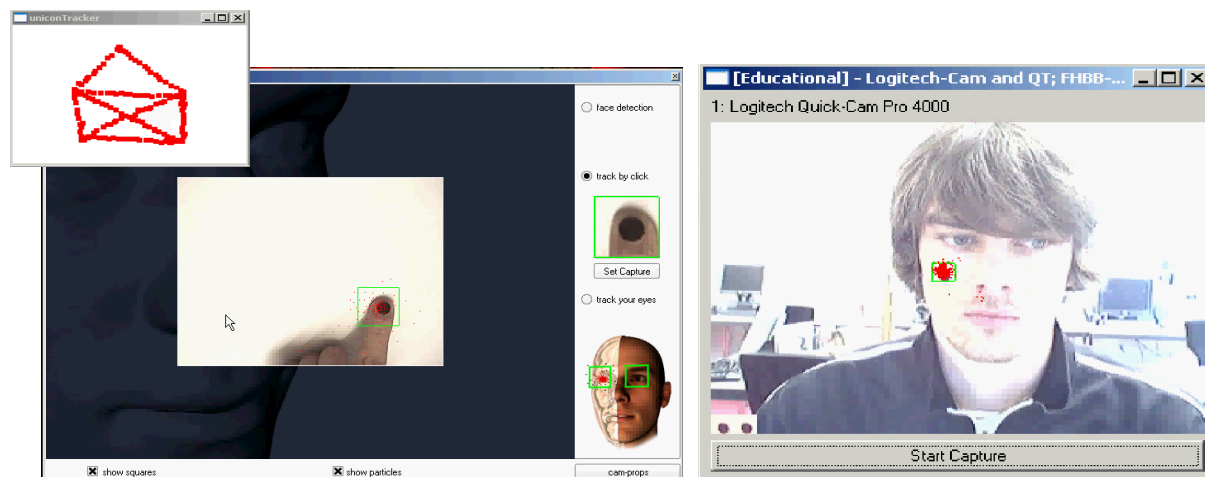


Figure 4-15: Adaptive real-time learning using Condensation and L2-norm

The *left image* demonstrates the real-time learning for a drawing application using only a black round marker stuck on the fingernail. Clicking once at the image a template is cut out (*Toolbox "Set Capture"*). Condensation tracking is applied using as simple measurement the L2-norm of the particles (*red dots*) and the template. The maximal response (*green box around the fingertip*) is used as coordinates for the virtual red pen on the drawing image (*top left*). The *right image* shows the real-time learning for tracking a round marker, stuck on a face. If the response of the L2-norm is higher as a threshold, the template is updated by the current underlying image area. The last two adaptive templates are shown at the *left-bottom corner* of the display.

To use only one template and only the L2-norm is not suitable for features that are more complex or in case of cluttered background. We would apply a FIFO stack of templates (verified by a complex classifier) and a dynamic learned W-RVM from these templates. Thus, we could realise the above-proposed contraction of the evaluation of the hyper-plane near the observed subject. The learning of small W-RVM classifiers on only a few training vectors is achievable in real time. We can take advantage of the automatic and straightforward learning stage of the W-RVM approach, in opposite to classifiers based on an optimisation of weak features, like [102].

5.2.6. Adaptive and Invariant Kernel

Up to now, we have always used RBF kernels (see discussion in Section 2.1). The question arises, of whether we could reduce the hypothesis space by altering the kernel function.

When we train a classifier like for faces or features, such as the nose tip, we enrich the training set by adding the mirrored positive examples, because if a face is within the positive class so is the mirrored face as well. For the pairwise used classifiers, such as the left and

right eyes or mouth corners, we train only one classifier. For instance, to train the left eye classifier we add the mirrored positive examples from the right eyes to the training set (and use for the detection of right eyes the (back) mirrored W-RSV's). Both strategies exploit the symmetric characteristic of the features; on the other hand they enlarge the training sets, and so the hypothesis space and computational complexity of the classifiers.

The question arises, as to whether we can enrich the kernel function, so that we only need to use one of the mirrored examples, without losing classification performance. In cooperation with B. Haasdonk, University of Freiburg [40] we discussed different kinds of invariant kernels. Most promising are reflection invariant kernels like:

$$1. \text{ IDS-Kern: } k(\mathbf{x}_j, \mathbf{x}_i) = \exp\left(-\frac{\min(\|\mathbf{x}_j - \mathbf{x}_i\|^2, \|s(\mathbf{x}_j) - \mathbf{x}_i\|^2)}{2\sigma^2}\right), \quad (5.8)$$

$$2. \text{ TI-Kern: } k(\mathbf{x}_j, \mathbf{x}_i) = 0.5\left(\exp\left(-\frac{\|\mathbf{x}_j - \mathbf{x}_i\|^2}{2\sigma^2}\right) + \exp\left(-\frac{\|s(\mathbf{x}_j) - \mathbf{x}_i\|^2}{2\sigma^2}\right)\right), \quad (5.9)$$

where $s(\mathbf{x})$ is the reflection of \mathbf{x} . In experiments, we have to prove if the more expensive kernels reduce the hypothesis space adequately. That means if the kernels reduce the complexity of the classifiers so that we end up with fewer operations for a comparable classification performance.

Another idea to exploit directly the symmetry of the pairwise trained classifier, for example for the left and right mouth corner, is to use this characteristic for the correlation classifier. The correlation classifier finds the final feature assortment within all combinatory possible combinations of detection candidates per facial feature point (Section 3.6). The classifier could be improved by the following strategy. A combination of feature points, where the difference of the patches of the mirrored left feature (e.g. the left mouth corner) and the right-side feature (right mouth corner) is high, gets a smaller correlation likelihood. To handle in-plane rotations the features could also be mirrored horizontally, and the minimum of differences would be used. This symmetry likelihood could be added as a term to the PSM function (3.16) introduced in Section 3.6.

In the previous section, we discussed that the core idea of this thesis is to spend less computation effort in easy-to-discriminate feature space areas and only a high complexity for areas where the objects of interest are supposed to be located. Following this strategy, we attempted to alter the kernel function in this section. But up to now, we have always used a identical kernel, optimised for all W-RSV's. For RBF kernels, used in our case, that means all kernels use the same "radius", regardless if this part of the hyper-plane is smooth or rather "winding". To put it differentially, we use the identical kernel parameters σ in (2.2),

regardless if the hyper-plane has to be described in more detail or can be evaluated more roughly for the area around the W-RSV. To tackle this issue we want to use adaptive kernels. They use instead of constant kernel parameters, adapted parameters per Support Set Vector. Our intention is to combine the theory of adaptive kernels with our W-RVM approach.

5.2.7. Further Optimisation of the W-RVM

Optimisations directly correlated to the learning stage of the W-RVM classifier were figured out in one of the cooperations with our partners; some of the ideas are listed in the following:

- Integral Image Representation: Instead of using the coordinates of the rectangle structure of the W-RSV's, it is more efficient to use a sparse matrix per W-RSV counting the number of used Integral Images pixels per location. This will reduce the computational cost between 30 and 50 percent.
- Soft-Shrinkage is the superior way to reduce the wavelet coefficient. In some cases, the hard-shrinkage is closer to the RSV's. Therefore, an optimisation considering both is optimal. In addition, we can test if the shrinkage for the average wavelet coefficients improves the efficiency.
- Optimisation of the evaluation strategy for the W-RSV's
 - Using an optimal number of W-RSV's per approximation level and an optimal number of levels per vector, or even an open order of the W-RSV's.
 - Using a constant number of wavelets coefficients, instead of a percentage of the highest coefficient per filter.
 - We could apply the rejections, instead of the decrement of the distance of the hyper-plane per operations in the cost function.
- Bundle W-RSV's into sets instead of rejecting patches after each W-RSV (needs fewer operations by unifying residuals using the Integral Image Representation)
- We could scale the size of the W-RSV's (observation window size), instead of the input image. The pyramid image does not have to be computed anymore.
- Zero mean and unit variance normalisation of the feature space with a pre-stage: Up to now, we have used no normalisation of the feature space, because this was not the focus of the thesis (see Chapter 1). A zero mean and unit variance normalisation would reduce the hypothesis space, and therefore the complexity of the classifier. We derived a method to compute the transformation by taking advantage of the Integral Image method with less than twenty operations per image patch. The contrast and the mean of an image patch are on the other hand a criterion for rejecting image areas. As a pre-stage, a threshold could be trained to reject image location where the contrast (variance) is too small, e.g. homogeneity background in the image.

5.3. Conclusion

The research of the proposed thesis is based on the optimal generalisation capabilities of Support Vector Machines [100] and the sequential evaluation of Reduced Set Expansions introduced by Schölkopf et al. [87] and applied for face detection by Romdhani et al. [73]. We also use the Integral Image approach by Viola and Jones [102]; used there for an AdaBoost learning algorithm.

The novelty of our approach proposed in this thesis is to combine the Reduced Set method with the Integral Image method. We adapted the Integral Image method to Support Vector Machine based learning. As major novelties of our approach, we invented a Double Cascade for an optimal trade-off between accuracy and speed. It is obtained by the cascaded evaluations over the number of incorporated Wavelet Approximated Reduced Set Vectors and additionally by a coarse-to-fine cascade of resolution levels for each of these vectors. As opposed to the RVM, the sparseness of operations required for classification is not only controlled by the number of Reduced Set Vectors but also by the number of wavelet basis functions used to approximate a Reduced Set Vector. Hence, negative examples can be rejected with a much less number of operations, making the runtime of the algorithm very efficient.

A novelty of this thesis is also the approximation of the Reduced Set Vectors by a Haar wavelet transform. Thus, a block structure is obtained and the Integral Image method can be used for their extremely efficient evaluation. The wavelet frame approach provides an upper bound of the hyper-plane approximation error. Exploring this characteristic, the training of the Wavelet Approximated Reduced Vector Machine works without heuristics and is fast. We also detail the relation between the hyper-plane approximation error of the decision functions and a training parameter to control the trade-off between sparsity and approximation.

Additionally we expanded the wavelet transform in our approach by an over-complete wavelet system to find the best representation of the Reduced Set Vectors.

The optimal approximation of the classification hyper-plane yields a very-fast-working classifier. The invention of an easy-to-train and fast-working classifier was the main goal of this thesis.

Taking advantage of the above-mentioned novelties the learning stage of our proposed Wavelet Approximated Reduced Vector Machine is fast, straightforward, automatic, and does

not require the manual selection of ad-hoc parameters and is therefore simple. This is the main advantage of this algorithm compared to other approaches.

The approach is straightforward because of our paradigm to avoid a complex search over the parameter space, by starting with the proved optimal performance of an SVM. Then we reduce the complexity by a Reduced Vector Set and the Over-complete Wavelet Approximation. The W-RVM is simple to re-implement. In Section 2.6, we propose a detailed pseudo code and the only input is the SVM and RVM. The used matrix notation makes the double-cascaded structure visible, supports vectorised code and reduces the update rule. This speeds up the training significantly. The parameter are adjusted automatically by the algorithm, e.g. for the number of resolution levels and the number of approximated vectors per level (Section 2.8). Also the thresholds b in (2.38) are obtained automatically. These thresholds are set to yield a given False Rejection Rate (FRR). The trade-off between FRR and FAR is the only parameter of our algorithm to be set by the user, because it depends on the requirements of the application (Section 2.7). All other parameters are automatically adjusted. The learning stage is fast, because the training of the W-RVM takes about two hours instead of weeks.

The performance of the classifier was first demonstrated on the task of face detection. Then the identical training algorithm was used for several facial features. This enables an automatic training of new objects and is an advantage over other techniques using a specific method for each feature [11].

For advanced post-processing we extended our classifier to a Probabilistic W-RVM. We introduced and compared different non-parametric techniques to evaluate the PDF. The best results we obtained by a Sigmoid Functions Fitting in Section 3.4.

Facial features are ambiguous, so we use a cascaded approach to apply the facial feature detectors in a FOI within the detected face area and a correlation classifier to find the final feature assortment within all combinatory possible configurations of detection candidates. We adapted two methods and got the best results using the advanced Prior Shape Model function using ten facial feature points.

For the problem of face and facial feature classification, we demonstrated that the decision hyper-plane of an SVM can be approximated by a much smaller number of vectors. Moreover, we show that the number of operations to evaluate the distance to the decision hyper-plane is drastically decreased. Together with the double-cascaded strategy, we gained that large parts of the images can be rejected with only few operations.

This theoretical improvement could be verified in the experiments. We used for the validation the common FERET database and compared the results to other state-of-the-art detection methods. On the validation sets, we compared the accuracy and the average time required to

evaluate the patches. The novel W-RVM algorithm provides a 530-fold speed-up over the Support Vector Machine and more than a 15-fold over the Reduced Support Vector Machine, for no substantial loss of accuracy.

Our proposed classifier is more efficient for detection than the most common state-of-the-art AdaBoost method [4]. The main advantage to [4] is the significantly improved training time by the paradigm to start with an SVM with proved classification performance instead with a complex search over weak classifiers. The W-RVM classifies about 25 times faster than the Rowley-Baluja-Kanade detector [1] and about 1000 times faster than the Schneiderman-Kanade detector [19]. To demonstrate the efficiency and accuracy of the detection algorithm, we implemented an application using a standard webcam. Accurate face detection is obtained in real time by 25 fps on a standard PC.

The essential ingredient was the development of an adequate classifier for face and facial feature detection. This is the main contribution of this thesis.

The performance of the facial features detection system was obtained exploring the 3D Morphable Model in first unifications loops with the proposed 2D image-based classifier. We took advantage of the 3D Morphable Model by generating synthetic training and validation data with a large variability concerning lighting, noise, pose, texture, and shape. We discussed the advantage of different classes of synthetic data. The use of synthetic data was indispensable, because labelled databases for arbitrary facial features were not accessible. The availability of such diverse data is an enormous improvement. Training now the W-RVM on these diverse data sets, yield classifiers that detect candidates for all feature points, which we then use as input for the 3D MM model again. The 3D Prior Shape Model function uses the 3D MM to find the final feature assortment within all combinations of the candidates for the feature points. At the resulting unification loop, the facial feature assortment can be used as initialisation for an automatic fitting of the 3D Morphable Model.

The loops of unification of the 2D image-based classification and the 3D Morphable Model form the general background of this thesis. This will be continued as discussed in Section 5.1.3. The practical use of unification is illustrated by a number of possible applications in the field of face analysis and synthesis.

Beside the publication of the new approach [70], [68], [71], a plug-in for the I-Search project [54], webcam applications for live presentations and an API for several HCI and CHIL projects were provided. While working on the thesis the new method is used in praxis and in cooperation with companies and institutes, for example the Cognitec GmbH, the Konrad-Zuse-Institute Berlin (ZIB), the University of Applied Sciences Neubrandenburg [99], or in a tracking application in cooperation with the FHNW Basel [13]. For the face detector a Free

Frame plug-in is available and can be used as common interface for visual programming languages, e.g. as Adobe plug-in [33]. We used it as interface for the visual programming languages VVVV [25] in cooperation with the HGK Basel in the field of perception psychology [57], [38]. This amount of projects verifies the efficiency and robustness in real-life environments. The applicability of the invented approach is also shown within bachelor, diploma, and master theses using the detection algorithm, the classifiers, or adapting the approach for other functional approximations like the approximation of regression functions.

Appendix A

UML Documentation

The following section gives a summary of the components of the developed software and some introductions how to use it. For a comprehensive documentation and implementation details, see the HTML documentation at the software.

W-RVM Project Report

Root Package

Component Diagrams

diagram FFDTraining

diagram FFDWorking

Component Diagrams

Component Diagram *FFDTraining*

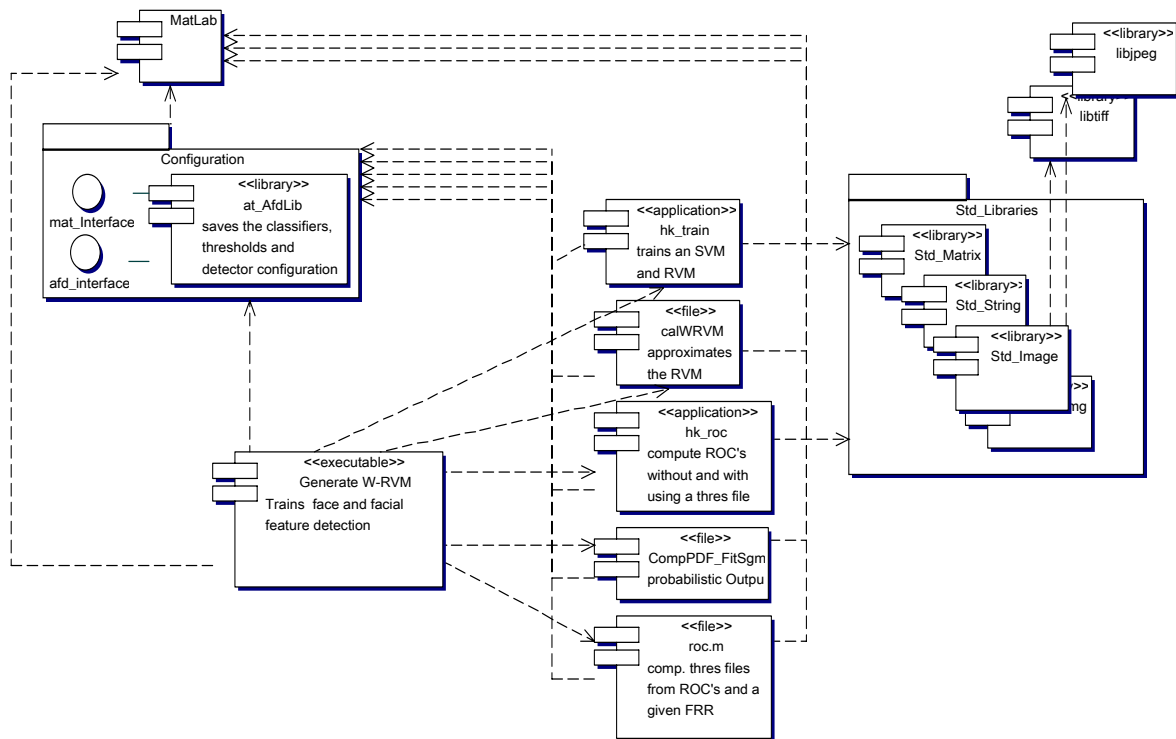


Figure A-1: UML diagram of the W-RVM Training Component UML view for all modules for the training of a W-RVM.

Component Detail **Component** *Generate W-RVM***Stereotype:** MatLab script**author:**

Matthias Raetsch

explanation:

Trains face and facial feature classifiers

The MatLab script for automatic training, adjusting, and validating of all facial feature classifiers and the face classifier has following stages:

0. INI for all facial features and for the face classifier
1. TRAIN SVM (see [hk_train](#))
2. TRAIN RVM (see [hk_train](#))
3. TRAIN W-RVM (see [calWRVM](#))
4. VALIDATE on test set (see [hk_roc](#))
 - R.O.C. on test set of the single SVM
 - R.O.C.'s for each W-RSV filter separately
 - R.O.C. of the last W-RSV filter (not cascaded W-RVM)
5. COMPUTE the PDF using histogram, knn and sigmoid function fitting to obtain a calibrated posterior probability to enable post-processing (see [CompPDF_FitSg.m](#))
6. ADJUST the thresholds for each W-RSV for a set of given FRR's (see [roc.m](#))
7. VALIDATE on test set with a set of threshold sets (see [hk_roc](#))
 - R.O.C. of the W-RVM only the final SVM stage, cascaded
 - PLOT Rejections over number of used operations and number of vectors
 - R.O.C. of the cascaded W-RVM + final SVM stage

Dependency Linksto **Subsystem** [Configuration](#)to **Component** [MatLab](#)to **Component** [hk_train](#)to **Component** [calWRVM](#)to **Component** [hk_roc](#)to **Component** [CompPDF_FitSg.m](#)to **Component** [roc.m](#) **Component** *hk_roc***Stereotype:** application

author:

Matthias Raetsch

explanation:

Application for computing R.O.C.'s without and with using a threshold file to validate a trained classifier

Dependency Links

to Subsystem [Std_Libraries](#)

to Subsystem [Configuration](#)

**Component** *hk_train*

Stereotype: application

author:

Matthias Raetsch

explanation:

Application for the training of an SVM and RVM on a set of patches

Dependency Links

to Subsystem [Std_Libraries](#)

to Subsystem [Configuration](#)

**Component** *roc.m*

Stereotype: MatLab script

author:

Matthias Raetsch

explanation:

Script for computing threshold files from an R.O.C. and a given FRR

Dependency Links

to Component [MatLab](#)

to Subsystem [Configuration](#)

**Component** *caWRVM*

Stereotype: MatLab script

author:

Matthias Raetsch

explanation:

Script for the approximation of an RVM (see Chapter 2, Table 2-1)

Dependency Links

to Component [MatLab](#)

to Subsystem [Configuration](#)

Component *CompPDF_FitSg.m*

Stereotype: MatLab script

author:

Matthias Raetsch

explanation:

Script for the training of a Probabilistic W-RVM

The PDF is computed using histogram, knn and sigmoid function fitting to obtain a calibrated posterior probability to enable post-processing.

Sigmoid function: $p(\text{ffp}|t) = 1 / (1 + \exp(A t + B))$

Dependency Links

to Component [MatLab](#)

to Subsystem [Configuration](#)

Component *MatLab*

Stereotype: library

explanation:

The standard libraries are used to handle MatLab data files

Component *libjpeg*

Stereotype: library

explanation:

The standard library is used for jpeg images

Component *libtiff*

Stereotype: library

explanation:

The standard library is used for tiff images

Subsystem Detail

Subsystem *Configuration*

Dependency Links

to Component [MatLab](#)

Contained Elements

Component *at_AfdLib*

Stereotype: library


explanation:

The library is used for the handling of the classifiers, thresholds and detector configuration files

"Supports" links

to **Interface** [mat_Interface](#)


to **Interface** [afd_interface](#)

 **Interface** *afd_interface*

Stereotype: interface

explanation:

This intern interface realisation is used to handle configuration files, if MatLab is not available

 **Interface** *mat_Interface*

Stereotype: interface

explanation:

This interface realisation is used to handle configuration files in MatLab format

 **Subsystem** *Std_Libraries*
Contained Elements
 **Component** *Std_Image*

Stereotype: library

author:

Matthias Raetsch

explanation:

The Library contains the handling of images

Dependency Links

to **Component** [libjpeg](#)

to **Component** [libtiff](#)

 **Component** *Std_IntImg*

Stereotype: library

author:

Matthias Raetsch

explanation:

The Library contains the handling of Integral Images

 **Component** *Std_Matrix*

Stereotype: library

author:

Matthias Raetsch

explanation:

The Library contains the handling of matrices

 **Component** *Std_String***Stereotype:** library**author:**

Matthias Raetsch

explanation:

The Library contains handling of strings

 **Component Diagram *FFDWorking***

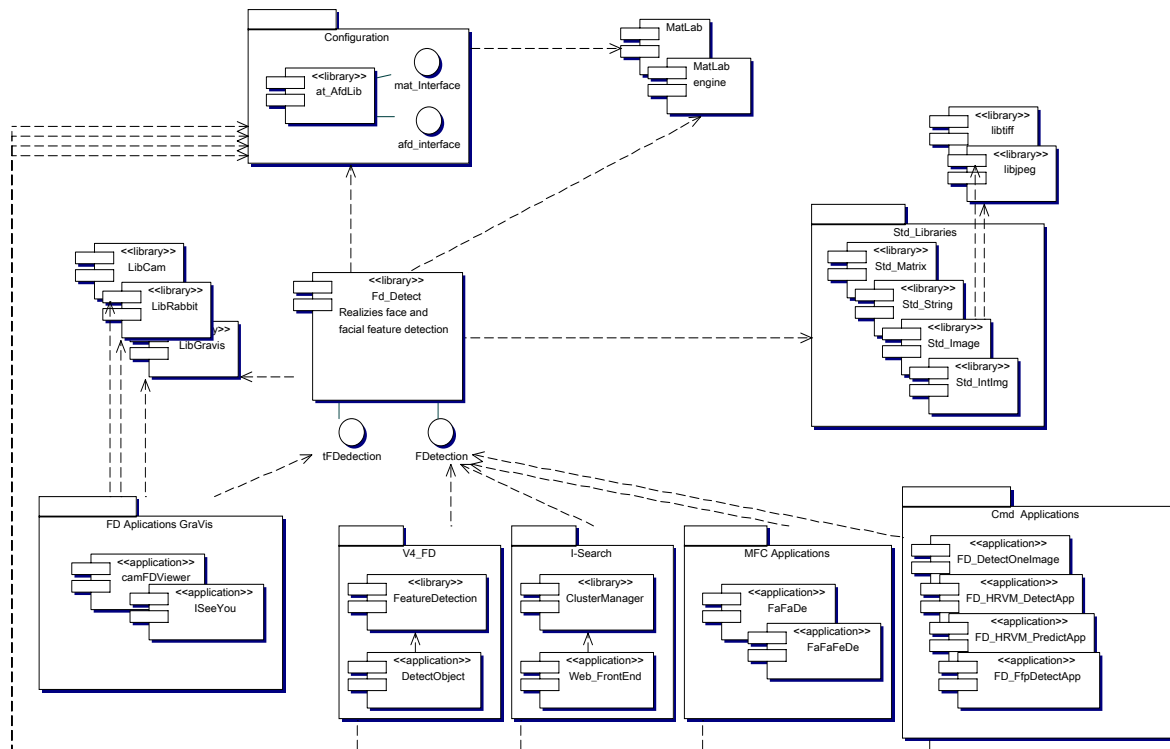


Figure A-2: UML diagram of the W-RVM Detection
Component UML view of all modules for the detection using a W-RVM.

Subsystem Detail


 **Subsystem Configuration**

author:

Matthias Raetsch

explanation:

System for the handling of the configuration of all components, see FFDTraining for more details

 **Subsystem Std_Libraries**

author:

Matthias Raetsch

explanation:

System for the handling of images, Integral Images, matrices, and strings, see FFDTraining for more details

 **Subsystem Cmd Applications**

Dependency Links

to Interface [FDetection](#)

to Subsystem [Configuration](#)

Contained Elements**Component** *FD_DetectOneImage*

Stereotype: application

author:

Matthias Raetsch

explanation:

The application runs face detection method for one image

```
usage: FD_DetectOneImage [<cfg_file.afd>] [<image_list.lst>]
      <cfg_file.afd>: Configfile, default: "..\config\fd_config_fd.mat"
      <image.tif>: Image, default: ".\tst.tif"
example: FD_DetectOneImage ..\config\fd_config_fd.mat .\tst.tif
```

Component *FD_FfpDetectApp*

Stereotype: cmdlinie application

author:

Matthias Raetsch

explanation:

The application runs the facial feature detection algorithm based on W-RVM detectors and the PSM correlation classifier on a list on images. Which features to use and the configuration of the application is stored in the config file.

```
usage: FD_FFpDetectApp [<cfg_file.afd>] [<image_list.lst>]
      <cfg_file.afd>: Configfile, default: "..\config\fd_config_ffd.mat"
      <image_list.afd>: Imagelist, default: ".\tst_i-search.lst"
example: FD_FFpDetectApp ..\config\fd_config_ffd.mat .\tst_i-search.lst
```

Component *FD_HRVM_DetectApp*

Stereotype: application

author:

Matthias Raetsch

explanation:

The application runs the face detector for a list of images using the Haar-like approximated vectors trained with Morphological Filters.

Usage see "FD_HRVM_DetectApp -h".

Component *FD_HRVM_PredictApp*

Stereotype: application

author:

Matthias Raetsch

explanation:

The application runs the face detector for a set of patches using the Haar-like approximated vectors trained with Morphological Filters.

Usage see “FD_HRVM_PredictApp -h”.

 **Subsystem** *MFC Applications*
Dependency Links

to **Interface** FDetection

to **Subsystem** Configuration

Contained Elements
 **Component** *FaFeDe and FaFaFeDe*

Stereotype: application

author:

Matthias Raetsch

explanation:

Live Video Application FaFaFeDe – Fast Facial Feature Detector

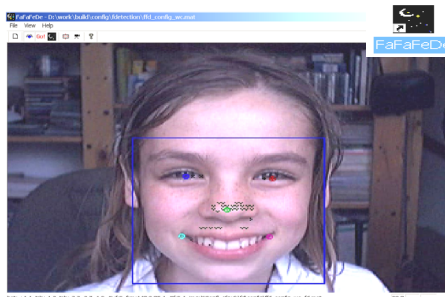


Figure A-3: The Fast Facial Feature Detector (FaFaFeDe)

1. Goal

For demonstrating the Fast Facial Feature Detection an application is implemented based on the face detection application FaFaDe for detecting a single feature or single face detection.

FaFaFeDe can run several facial feature detectors, use ROI's for each feature relative to the detected face and show the stages of the detectors.

The results of the detectors are combined and using additional a 3D correlation model for the feature points the final feature assortment is found. All other detections are rejected as False Acceptances.

After showing the stages of the single W-RVM detectors a combination of the detection results is evaluated, if at least five facial feature points are detected. A 3D Prior Shape Model is used to estimate the 3D correlations of the features. Finally the unification of results from the appearance model (combinations of the outputs of the detectors) and the shape model is computed. The detection certainties and the final confidence, combining the detection and prior shape results, can be visualized. The best feature assortment is marked by circles at these stages.

The here described version is developed for using the first PSM using five facial features.

2. FaFaFeDe GUI

2.a. The FaFaFeDe Toolbar

- From left to right can here be chosen (Figure A-4): the video source, to pause the video, to start or stop the detector with “Go!” (to load the configuration and classifiers takes some seconds, since of the size; restarting the detection sets back all parameters to the values from the configuration files), switch off the visualization, open the video format dialog (e.g. to change the video resolution), open the video source dialog to adjust the parameters of the camera, like brightness, shutter time, gain etc. and to open the about dialog.

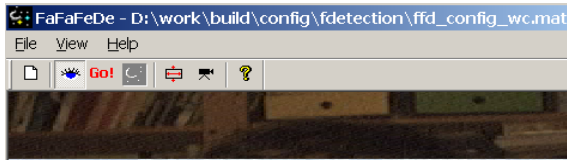


Figure A-4: FaFaFeDe Toolbar

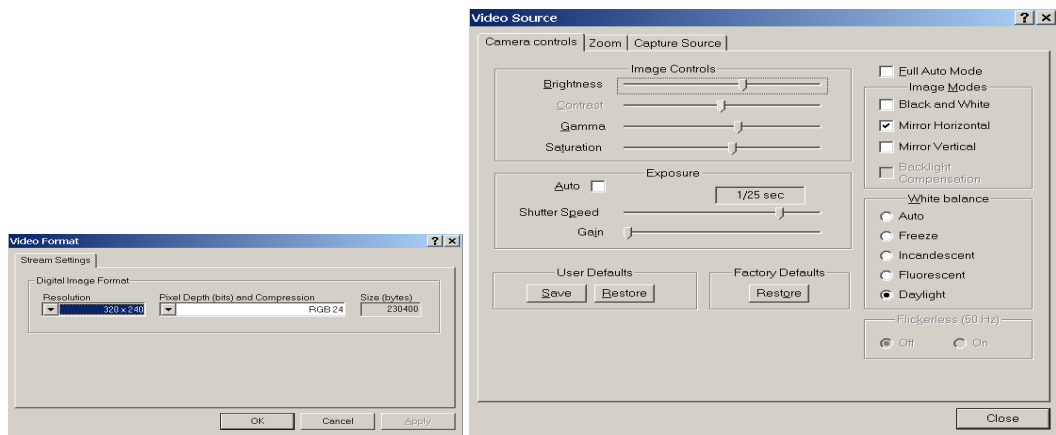


Figure A-5: FaFaFeDe video format and source dialog

2.b. The FaFaFeDe status bar

- The status bar is used to watch the most important outputs of the detector, like the certainty of the detectors, the frame rate or a pose estimation. Also the current adjusted values of the most significant method parameters are shown, like the number of used scales, or which stage is visualized at the display window.

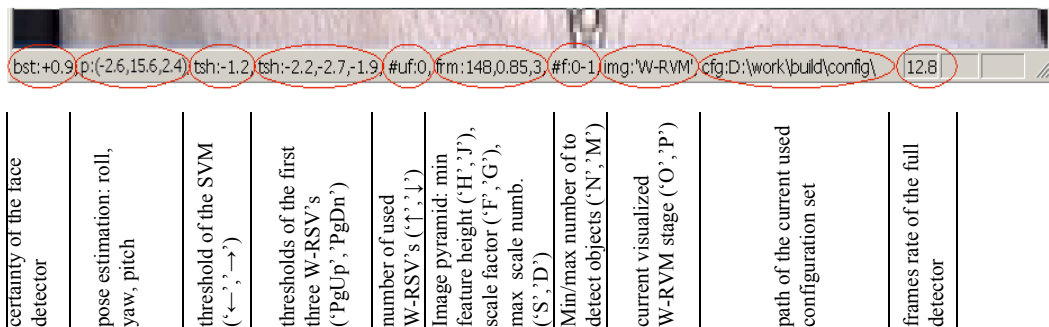


Figure A-6: FaFaFeDe status bar and keys for adjusting some of the control parameter

2.c. The visualisation of the W-RVM stages by FaFaFeDe

- The W-RVM detection is a multi-stage approach. Each stage rejects detected locations of the features from the output of the previous stage. The algorithm starts from a core and fast to a fine and complex filter. The output of each stage can be visualized by FaFaFeDe for each detector into the same view by different colors and different markers (blue 'x': left eye, red '+': right eye, green 'v': nose tip, cyan '<': left mouth corner, and magenta '>': right mouth corner). A dark color shows always a low

detection probability of the current filter stage and a bright color a high probability. The detections of the face classifier are marked by the bounding box and bright red for uncertain and bright blue for certain detections.

- The stages of the W-RVM approaches are (see Figure 4-3):
 - i) **"WRVM"**: default visualization of the full approach, same as last stage "6Conf"
 - ii) **"1WRVM"**: The first stage, the Wavelet Approximated Reduced Vector Machine classifier runs in a sliding window method over each column, each row and each scale of the pyramid images and classifies the underlying patch. Over 95% of the locations are rejected as non feature and it will not further evaluate. The W-RVM is also a Double Cascaded classifier of the number of used Wavelet Approximated Reduced Set Vectors (W-RSV's) and over the approximation resolution levels. Hence easy to classify image location (like homogeneous background) are rejected very efficient with less as fifty computational operations (see Chapter 2, 3 for details).
 - iii) **"2WRVMoe"** (W-RVM overlap elimination): All detections per cluster are rejected except the top n best detections (see key in Config). At this stage also the ROI's are shown by bright boxes and the image pyramid of each feature in the top left image corner by a brighter color of each feature.
 - iv) **"3SVM"** Full Support Vector Machine Stage: The full classification power with the well-known best generalization performance of the SVM for most accurate detection and certainty outputs.
 - v) **"4Cert"** Certainty of the combinations of all possible combinations of detections. Each feature point is shown with the max certainty of all combination where it is incorporated. Only all detections after an overlap elimination of the SVM detections (similar to the overlap elimination after the "1WRVM" stage) are considered.
 - vi) **"5Prio"**: Probability of the Prior Shape Model for each combination of detections. It shows the probability that this set of features has a valid correlation in a 3D facial space.
 - vii) **"6Conf"**: Final confidence of each combination of features as function of the certainty from the appearance model (see "4Cert") and from probability (see "5Prio") of the 3D Prior Shape Model.

The Stages v) until viii) can only be computed by at least five feature points, (otherwise nothing is shown here). The combination of feature points with the highest likelihood to be the final assortment is marked by circles. You can switch between the stages with the keys 'O' and 'P' and check at the status bar which stage current is shown.

2.d. Optimization parameter of the W-RVM approach

- All used parameter to control the W-RVM algorithm can be adjusted in configuration files. The format of these files is similar to Windows Ini files. The parser is MatLab and can also used as editor. At the command prompt the parameter '-i <config_file>' can be used to call FaFaFeDe with a configuration file (see usage). This general configuration file includes the description witch feature points are to detect and default parameter for each detector. This parameter can be over loaded by the specific config files of the features. The set of configuration files of the single feature detectors are expected at the same directory as the <config file> or their path must be defined in the MatLab environment variable MATLABPATH. The filename syntax is <ffp_config_file>=<config_file>_<ffp>.mat, where <config_file> is the name of the general configuration file (without extension) and <ffp> the abbreviation of the feature ('fd' face, 'le' left eye, 're' right eye, 'nt' nose tip, 'lm' left and 'rm' right mouth corner detector).
- The most relevant parameters can also be live adjusted by short cuts keys in FaFaFeDe

Short cut key	Config section and key name	Description
	ALLGINFO.configname	Path and name of the configuration file (after changes MatLab will save the file there). Running the general config files saves all config files of the defined features. Syntax of the file names see above.
	ALLGINFO.verboselevel level9o9o9o oi0l0ööp.öp.- 7 6u	Verbose level for debug outputs at the console and into the event log (use e.g. http://www.download.com/DebugView/3000-2218_4-10213956.html) (0: no, 1: most important, 2: many outputs)
	ALLGINFO.dumping	List of bool values which debug information and images should be saved (InImg, GreyImg, OverElmImg, DrawScalings, DrawROI, OutImg, drawBB, BBList, facc, tacc, frej, trej; eg. [0 0 0 0 0 1 0 0 0 0]: only draw detection into the image (fast!) [0 0 1 1 1 1 1 1 0 0 0 0]; save images of all stages, draw image pyramid, draw ROI's and draw the detections and save a list in a ascii file of them). Use this to report questions or bugs, be careful of HHD space and slow performance)
	ALLGINFO.outputdir	Directory for debug outputs (see ALLGINFO.dumping)
	FFD.detec_features and FFD.num_features	This key is only set in the general config file and defines the set of to detect facial feature points (at the moment 0=face, 1=left eye, 2=right eye, 3=nose tip, 4=left and 5=right mouth corner are defined) eg. only face detector: [0]; eg. face and eye detectors: [0 1 2]; eg. face and nose tip detectors: [0 3]; The face detector must be the first, if the other detectors shall run only into the detected face area. num_features is defined by the length of the vector.
	FD.scenario	This key can be used to define configs for specific scenarios like webcams, FireWireCams or working at a data base.
	FD.ffp	Key to define the kind of feature point for that detector (at the moment 0=face, 1=left eye, 2=right eye, 3=nose tip, 4=left and 5=right mouth corner are defined)
'O', 'P'		Visualized W-RVM stage in FaFaFeDe (see Section 3c)
'S', 'D'	FD.maxscales	Max number of scales for the image pyramid
'F', 'G'	FD.scalefactor	Scaling factor for sup sampling to the next smaller image in the pyramid
'H', 'J'	FD.face_size_min	Min. height of the feature in pixel
'N', 'M'	FD.expected_number_faces	Max number of expected objects (if n=1,2 or 3 then always only the best n detection are not reject, if n>3 all object with a higher certainty output as the threshold are not rejected)
'←', '→'	FD.limit_reliability	Threshold offset for final SVM stage. All feature detections with a smaller output are rejected
'↑', '↓'	FD.limit_reliability_filter	Threshold offset for all W-RSV's. All feature detections with a smaller output at the cascaded filters over the W-RSV's are rejected (e.g. -0.1, than less patches are rejected at each filter, since less false rejections, but slower performance, since more have to be evaluated by more complex stages)
'PgUp', 'PgDn'	FD.numUsedFilter	Number of used W-RSV's. Only one until the full number of trained W-RSV's can be used before the W-RVM cascade stops and the SVM

		stage is used (Default is 0 = all trained W-RSV's)
	FD.expected_face_orientation	Excepted orientation of the feature, at moment only frontal, upright features are trained, but the classifiers are about +/-15° in plane, +/-45° yaw und 30° pitch angle rotation invariant
	FD.roi	ROI within the feature will be detected eg.: [0 0 0 0] full image eg.: [-1 -1 -1 -1] full bounding box of the face detector eg.: [l t r b] l pixels border from the left, t from top, r from right, and b from the bottom of the detected face box, or of the full image in the case it is a face detector or no face detector defined.
	FD.only_full_svm	Only full SVM (no W-RVM), SLOW!!!
	FD.doesPPOverlapElimination	Mode for overlap elimination eg. 0: only after SVM, 1: only after W-RVM stage or n: Reduce each cluster to the n top best after W-RVM and to the best after the SVM stage.
	FD.distOverlapElimination	Overlap elimination distance [and ratio] that two detection belong to the same cluster. eg. [d r]: If more overlap (smaller dist as d of the centres and smaller ratio r of the feature areas), than detections with lower likelihood will be rejected (d <=1: distance is measured relative to the feature widths, d >1: distance measured in pixels)
	FD.distOverlapElimination	Overlap elimination distance [and ratio] % if more overlap (smaller dist as [0] and ratio [1]), than the all detections with the lower likelihood will be deleted % Werte: [0]: float ([0]<=1 rel. to patch width, [0]>1 in pixel), [1]: float (-1.0 ignore, 0.0<[1]<=1.0); Default: 5,-1
	FD.stepsize	Step size of the detector in pixel. The step size depends at the image scale coef and is defined by max(1,int(coef*FD.stepsize+0.5)) eg. 3.0: step size of 3 pixel for the full resolution image (smallest feature), for the other resolution the step size is smaller 3.0 but at least 1.0 pixel.
	FD.classificator	Path and name of the file which contains the trained W-RVM and SVM classifier in *.mat format
	FD.threshold	Path and name of the file which contains the trained thresholds of the W-RSV's in *.mat format

3. How to start and optimize the detection results by FaFaFeDe

3.a. FaFaFeDe can be started at the command prompt as following

Usages: fafafede.exe [-i <config file>]

-i <config file>: configuration file which includes the description witch feature points are to detect and default parameter for each detector (Default ,..\config\ffd_config_wc.mat'

Example: fafafede.exe -i ..\config\fdetection\ffd_config_wc.mat

3.b. How to optimize detection results and solving problems

- o Read the documentation carefully
- o By problems look at the system log, since FaFaFeDe hast no console (e.g. use http://www.download.com/DebugView/3000-2218_4-10213956.html)

- Start the application with default parameter (see usage.) or use an optimized configuration file.
- Check if the camera and the light conditions are good adjusted. The detector is sensitive to the brightness. Do don't start with light mostly from one side, spectacular highlights within the face and other strange lighting conditions.
- It is always better to use more light and longer shutter times than to turn the gain control of the camera higher (noisy images)
- Start the detection with "Go!" at the tool bar. First the classifier is loaded, that can tack some seconds, since of the size of the trained SVM.
- Switch to the first stage with the key 'P' for initial optimization.
- Start with an as frontal as possible face detection in ALL three axes. The detectors are rotation invariant until about $\pm 15^\circ$ in plane, $\pm 45^\circ$ yaw und $\pm 30^\circ$ pitch angle. But best results are obtained by frontal, upright and not tilt features.
- Optimize orientation, distance to the camera and brightness of the camera/lighting source until the bounding box of the face detector is bright blue and you get as bright and as large as possible detection clusters for the feature detectors at the first W-RVM stage.
- Now change to the next detector stage "2WRVMoe" using the key 'P'. Here you can optimize the image pyramid (size of the detector patches), the ROI, thresholds of the classifier and other parameter (see above). Find an optimal trade off between speed (frame rate) and detection certainty (see both at the status bar).
- If you changed parameter and don't get a detection anymore go back to the default parameter set (e.g. by restarting the detectors clicking "Go!" twice.)
- After optimization copy your best parameters to the configuration files (see documentation of the status bar).
- If you need support or have suggestions don't hesitate sending your comments to matthias.raetsch@web.de (please with a copy of the debug infos from the system log, saved debug images (see config file), and the used set of config files)

Subsystem *V4_FD*

Dependency Links

to **Interface** [FDetection](#)

to **Subsystem** [Configuration](#)

Contained Elements

Component *DetectObject*

Stereotype: application

author:

Matthias Raetsch in cooperation with HGK Basel

explanation:

The component realises the application InFaFeDe – Interactive Fast Face Detection (see Section 4.3.3).

The application runs the face detection algorithm using the graphical programming language VVVV. This language is convenient for real-time video applications: camera inputs or displays are simple graphical nodes and thus easy to use.

The advantage of the interactive interface is that it is applicable, for projects not in the field of computer science. For VVVV no imperative programming knowledge is needed. In addition, the interface can be used for experimental programming or rapid prototyping. The parameters of the W-RVM approach can be demonstrated: For in-

stance, non-experts can optimise the detection with respect to the trade-off between accuracy and runtime performance for specific environments.

Dependency Links

to Component [FeatureDetection](#)

Component *FeatureDetection*

Stereotype: library

author:

Matthias Raetsch

explanation:

The library realises the Free Frame interface for the W-RVM detector. Free Frame is an open-source cross-platform for real-time video effects. With Free Frame, we use a plug-in system, which is a common interface for visual programming languages, e.g. Free Frame interfaces can be used for Adobe plug-ins (see Section 4.3.3).

Subsystem *I-Search*

Dependency Links

to Interface [FDetection](#)

to Subsystem [Configuration](#)

Contained Elements

Component *ClusterManager*

Stereotype: library

author:

Matthias Raetsch in cooperation with ITWM Kaiserslautern

explanation:

The library realises the API for the I-Search project (see Section 4.2).

I-Search was the first project using an API of our face detector applying the W-RVM classifier introduced in Chapter 2. One intension of the project was to use a web crawler to search through the WWW for faces. A cluster architecture was set up and the computational load optimised for each cluster. The W-RVM detector is one of the functions running at the cluster.

A further documentation of the project and detailed descriptions of the results are given in our final report [69].

Component *Web_FrontEnd*

Stereotype: application

author:

Matthias Raetsch in cooperation with ITWM Kaiserslautern

explanation:

The component realises a web application for the I-Search project (see Section 4.2).

The intention of this application was to build up a live detection system, as seen in Figure 4-7. One of the applications for this setting was our face detection approach. The API was applied by a web-service application.

At the final presentation, we could show that our W-RVM face detector was able to detect all of the persons, which passed the installation during the demonstration.

A further documentation of the project and detailed descriptions of the results are given in our final report and Section 4.2.

Dependency Links

to Component [ClusterManager](#)

Subsystem *FD Applications GraVis*

author:

Matthias Raetsch in cooperation with diploma, master, and bachelor students

explanation:

This is a platform where projects can be developed using the W-RVM approach. The purposes of the projects are HCI and CHIL applications (see Section 4.1.5, 4.3). Standard libraries of our research group like libCam (camera interface), libRabbit (GUI library), and libGravis (standard image and matrix library) are used.

Dependency Links

to Interface [tFDetection](#)

to Component [LibGravis](#)

to Component [LibCam](#)

to Component [LibRabbit](#)

Contained Elements

Component *camFDViewer*

Stereotype: application

author:

Matthias Raetsch in cooperation with Pascal Paysan

explanation:

This is a portable application for FFD, which is not limited to Windows like FaFaDe and FaFaFeDe because of the usage of the MFC. This application is a starting point for further extensions of the face and facial features detection approach. The application camFDViewer is limited to face detection or one facial feature but uses the identical classes as the facial feature set detection, only the GUI has to be expanded and the interface adapted.

The application camFDViewer could already be adapted for new applications. For instance, the projects “Face and Facial Feature Point Tracking” in Section 4.3.1, “Avatar Following with Eye and Head Motion” in Section 4.3.2, and the “Tracking of Higher Feature Parameters” in Section 5.2.4 took advantage of camFDViewer as a starting point.

 **Component** *ISeeYou*

Stereotype: application

explanation:

The project “I See You” is one example for an HCI application see “Avatar Following with Eye and Head Motion” in Section 4.3.2

Component Detail

 **Component** *Fd_Detect*

Stereotype: library

author:

Matthias Raetsch

explanation:

The library realises the face and facial feature classifiers, detectors and the facial feature set detection as introduced in Chapter 2 and 3. Table 2-2 can be used for a summary of the W-RVM classifier, Table 3-6 for the W-RVM detectors, and Table 3-7 for a summary of the W-RVM facial feature set detection using the PSM.

"Supports" links

to **Interface** [tFDetection](#)

to **Interface** [FDetection](#)

Dependency Links

to **Subsystem** [Std_Libraries](#)

to **Subsystem** [Configuration](#)

to **Component** [MatLab engine](#)

to **Component** [LibGravis](#)

 **Component** *LibCam*

Stereotype: library

 **Component** *LibGravis*

Stereotype: library

author:

GraVis research group

explanation:

The library realises the handling of images, vectors, arrays and matrices

 **Component** *LibRabbit*

Stereotype: library

author:

GraVis research group

explanation:

The library realises the handling of GUI displays and dialogs

 **Component** *libjpeg*

Stereotype: library

explanation:

The standard library realises the handling of jpeg images

 **Component** *libtiff*

Stereotype: library

explanation:

The standard library realises the handling of tiff images

 **Component** *MatLab*

Stereotype: library

explanation:

The MatLab libraries realise the handling of MatLab data files

 **Component** *MatLab engine*

Stereotype: library

explanation:

The MatLab libraries realise the handling of a MatLab engine used to run the PSM functions

Interface Detail

 **Interface** *FDetection*
author:

Matthias Raetsch

explanation:

The component realises an interface for all applications using libraries of the subsystem Std_Libraries and Std_Image as image format.

 **Interface** *tFDedection*
author:

Matthias Raetsch

explanation:

The component provides an interface for all applications using standard libraries of the GraVis research group, like libRabbit, libGravis and tImage as image format.

Appendix B

Used Data Sets

Description of some of the used data sets for training and validation of face and facial feature classifiers:

Sets Patches:

Set Patches	neg	pos	pose	Comment
patches_ind_front_faces.txt p_front_all_rand_#F18213.txt	0	18.213	frontal	Web <root_data>..\Sets
patches_ind_nonface.txt	93.630	0	most frontal	Web,
good_#F4911_#nF5604.txt	5.604	4.911	most frontal	Manually sortet from Web
patches_faces-and-nonfaces_tr_3194f_20275nf.txt f-nf-train.mat	20.275 taken randomly from 100 Corel images	3.194 80% of the 300 faces from Manchester, 100 female models of Corel and 4000 faces of Penev	most frontal	from former project train-set <root_data>\Sets\
patches_faces-and-nonfaces_test.txt f-nf-test.mat	5.375	828 20 remaining % of the training set.	most frontal	from former project test-set <root_data>\Sets\
patches_faces-and-nonfaces_101_1000f-vontr_5x20000nf-vontr.csv f-nf-tr_001.mat	100.000 5x20.000nf von f-nf-tr.mat	1000 1.000f von f-nf-tr.mat	most frontal	
patches_vir-2_train.txt	13.890	19.377 3194 from above and 5 virtual per face	most frontal	from former project
patches_vir-5_train.txt	13.504	20.504 from above + faces detected in the 600 Corel images + webimages	most frontal	from former project
patches_vir-2_test.txt	28.080	5.003	most frontal	from former project test-set zu vir-5/2
bdisc-20x20-testset.mat	100.000 <root_data>\bdisc-20x20_testset	7.742	frontal	pos. from feret (all frontal with mirrored) <root_data>\Sets\
<root_data>\Sets\b2_1920\le_tr.mat training set	100800	2880	frontal and pose	pos. from feret (all frontal with mirrored re, see Section 3.1)
<root_data>\Sets\b2_1920\le_ts.mat test set	33600	960	frontal and pose	pos. from feret (all frontal with mirrored re, see Section 3.1)
<root_data>\Sets\b2_1920\nt_tr.mat training set	109432	2880	frontal and pose	pos. from feret (all frontal with mirrored, see Section 3.1)
<root_data>\Sets\b2_1920\nt_ts.mat test set	36476	960	frontal and pose	pos. from feret (all frontal with mirrored, see Section 3.1)

<root_data>\Sets\b2_1920\lm_tr.mat training set	92160	2878	frontal and pose	pos. from feret (all frontal with mirrored rm, see Section 3.1)
<root_data>\Sets\b2_1920\lm_ts.mat test set	30720	960	frontal and pose	pos. from feret (all frontal with mirrored rm, see Section 3.1)
<root_data>\Sets\b2_1920\lx_tr.mat training set	119280	22812	randomly	pos. from feret (all frontal with mirrored rx) mixed with synthetic data (see Section 3.1)
<root_data>\Sets\b2_1920\lx_ts.mat test set	59640	11406	randomly	pos. from feret (all frontal with mirrored rx) mixed with synthetic data (see Section 3.1)
<root_data>\Sets\b2_1920\ls_tr.mat training set	119280	22812	randomly	pos. from feret (all frontal with mirrored) mixed with synthetic data (see Section 3.1)
<root_data>\Sets\b2_1920\ls_ts.mat test set	59640	11406	randomly	pos. from feret (all frontal with mirrored) mixed with synthetic data (see Section 3.1)
<root_data>\Sets\b2_1920\lb_tr.mat training set	119280	20238	randomly	pos. from feret (all frontal with mirrored rb) mixed with synthetic data (see Section 3.1)
<root_data>\Sets\b2_1920\lb_ts.mat test set	59640	10119	randomly	pos. from feret (all frontal with mirrored rb) mixed with synthetic data (see Section 3.1)

Table B-1: Sets patches**Syntax sets for patches:**

- each row is one patch ($h \times w$ image vector, where h is the height and w the width of a patch)
- first number is the label (1: feature, -1: non-feature)
- followed by ($h \times w$, e.g. $20 \times 20 = 400$) grey values.

Sets Images:

Set	Numb	Number faces per image	pose	Comment
Frontal FERET data base feret-frontal.eye	3.876	1	frontal	All frontal faces form FERET see Table B-3 <root_data>\Feret\ground_truths\
feret-frontal_m2.lst	74	1	frontal	Subset of FERET
feret-frontal_m3.lst	33	1	frontal	ditto
feret-frontal_m4.lst	30	1	frontal	ditto
BioId.lst	100	1	frontal	Subset from BioID see Table B-5
cmu.lst	30	1-n	frontal and pose	Subset from MIT-CMU see Table B-4

Table B-2: Sets of images used for validation and experiments**Syntax sets for images:**

- each row is one image, given by the path and filename followed optional by labels.

name	Face Recognition Technology (FERET) program database	
description	images	The Colour FERET database contains a total of 11338 facial images. They were collected by photographing subjects over the course of 15 sessions between 1993 and 1996. This database is largely a colour version of the original Facial Recognition Technology (FERET) Database, which was released in 2001 and consisted of 14051 greyscale images of human heads with views ranging from frontal to left and right profiles.
	number subjects	994 subjects (Colour FERET database), 1209 subjects (Grey FERET database)
	colour depth	24-bit colour images and eight-bit greyscales images
	resolution	512 by 768 pixels (Colour FERET database), 256 by 384 pixels (Grey FERET database)

	labels	four (eyes, nose tip, mouth centre)
	comment	The FERET program ran from 1993 through 1997. Sponsored by the Department of Defence's Counterdrug Technology Development Program through the Defence Advanced Research Products Agency (DARPA), its primary mission was to develop automatic face recognition capabilities that could be employed to assist security, intelligence, and law enforcement personnel in the performance of their duties. The program consisted of three major elements: sponsoring research, collecting the FERET database and performing the FERET evaluations.
URL	http://www.nist.gov/humanid/colorferet/home.html (Colour FERET database) http://www.itl.nist.gov/iad/humanid/feret/ (Grey FERET database)	

Table B-3: Face Recognition Technology (FERET) program database

name	MIT-CMU Face Detection Database	
description	images	130 images including images from the World Wide Web, scanned from photographs and newspaper pictures, and digitized from broadcast television. The image dataset was first used by the CMU Face Detection Project and is provided for evaluating algorithms for detecting frontal views of human faces. It combines images provided by K.-K. Sung and T. Poggio at MIT (Test Set B), and by H. Rowley, S. Baluja, and T. Kanade (Test Sets A,C and the rotated test set) at CMU.
	number subjects	507 frontal faces
	colour depth	grey scale
	resolution	large range
	labels	six (eyes centres, nose tip, mouth centre and corners)
	comment	poor resolution, difficult illumination, pose, rotation, oclusions, structural components, line drawings
URL	http://vasc.ri.cmu.edu/~fdb/html/face/frontal_images/index.html	

Table B-4: MIT-CMU Face Detection Database

name	BioID Face Database	
description	images	The BioID Face Database is a test set of 1521 images. Each one shows the frontal view of a face of one out of 23 different test persons. For comparison reasons the set also contains manually set eye positions. During the recording special emphasis has been laid on "real world" conditions. Therefore the test set features a large variety of illumination, background and face size.
	number subjects	23
	colour depth	grey scale
	resolution	384x286
	labels	20 (eyes centres, nose tip, mouth centre and corners, etc.)
	comment	The BioID Face Database is being used within the FGnet project of the European Working Group on face and gesture recognition. D. Cristinacce and K. Babalola, (University of Manchester) selected several additional feature points.
URL	http://www.bioid.com/ (BioID database) http://www-prima.inrialpes.fr/FGnet/html/home.html (BioID used by European Working Group on face and gesture recognition)	

Table B-5: BioID Face Database

Appendix C

Trained W-RVM Classifiers

Tables with some of the trained SVM's, RVM's, and W-RVM's

WRVM	#W-RSVs	#Lev	#RSVs	final Stage	Used RVM	threshold ed with	Comment
hrvm_ss_ts3_b1-4x1-4_8_bkorr_c0.2_2_20_With-v5-SVM.mat	90	1	90	v5-SVM	f-nf-hk90.mat	f-nf-tr	fd with MorFilter generated. only one level
wrvm_o8x8_n7l20_thr0.7-0.3.mat	140	7	20	f-nf-hk90.mat	f-nf-hk90.mat	f-nf-tr	fd, no thresholds trained
fd_fnf_wvm_r0.06_c1_o8x8_n14l20t10_hcthr0.84-0.42,0.36-0.18.mat	80	14	20	f-nf-hk90.mat	f-nf-hk90.mat	f-nf-tr	
fd_rfb2_20x20_wvm_r1e-006_c2_o8x8_n30l7t5_hcthr0.84-0.49,0.60-0.35.mat	210	7	30	bdisc-20x20-rfb1_svm_org.mat	bdisc-20x20-rfb2_rvm.mat	bdisc-20x20-testset.mat	fixed eye coordinates
le_wvm_r0.1_c4_o8x8_n30l7t5_thr0.91-0.49,0.65-0.35.mat	210	7	30	le_wvm_r0.1_c4_o8x8_n30l7t5_thr0.91-0.49,0.65-0.35--le_ts_thres_0.01.mat	le_rvm_r0.1_c4.mat	b2_1920\le_tr.mat	
nt_wvm_r0.2_c1_o8x8_n30l7t5_hcthr0.84-0.49,0.60-0.35.mat	210	7	30	nt_wvm_r0.2_c1_o8x8_n30l7t5_hcthr0.84-0.49,0.60-0.35--nt_ts_thres_0.01.mat	nt_rvm_r0.2_c1.mat	b2_1920\nt_tr.mat	
lm_wvm_r0.1_c4_o8x8_n30l7t5_thr0.91-0.49,0.65-0.35.mat	210	7	30	lm_wvm_r0.1_c4_o8x8_n30l7t5_thr0.91-0.49,0.65-0.35--lm_ts_thres_0.01.mat	lm_rvm_r0.1_c4.mat	b2_1920\lm_tr.mat	
lx_wvm_r0.4_c10_o8x8_n50l7t5_hcthr0.84-0.49,0.60-0.35.mat	350	7	50	lx_wvm_r0.4_c10_o8x8_n50l7t5_hcthr0.84-0.49,0.60-0.35--lx_ts_thres_0.01.mat	lx_rvm_r0.4_c10.mat	b2_1920\lx_tr.mat	
ls_wvm_r0.3_c10_o8x8_n30l7t3_thr0.84-0.56,0.60-0.40.mat	210	7	30	ls_wvm_r0.3_c10_o8x8_n30l7t3_thr0.84-0.56,0.60-0.40--ls_ts_thres_0.01.mat	ls_wrvm_r0.3_c10.mat	b2_1920\ls_tr.mat	
lb_wrvm_r0.5_c8_o8x8_n30l7t5_thr0.84-0.49,0.60-0.35.mat	210	7	30	lb_wrvm_r0.5_c8_o8x8_n30l7t5_thr0.84-0.49,0.60-0.35--lb_ts_thres_0.01.mat	lb_rvm_r0.5_c8.mat	b2_1920\lb_tr.mat	

Table C-1: Trained W-RVM's

RVM	#RSVs	final Stage	Used SVM	Thres	Comment
bdisc-20x20-rfb1_rvm.mat	100	bdisc-20x20-rfb1_svm_org.mat	bdisc-20x20-rfb1_svm_org.mat <root_data>\SVM	Sets\bdisc-20x20-testset.mat	RVM: <root_data>\RVM \bdisc-20x20-rfb1_rvm_n
bdisc-20x20-rfb2_rvm.mat	100	bdisc-20x20-rfb2_svm_org.mat	bdisc-20x20-rfb2_svm_org.mat	Sets\bdisc-20x20-testset.mat	RVM: <root_data>\RVM \bdisc-20x20-rfb2_rvm_n
f-nf-hk90_With-v5-SVM.mat	90	v5-SVM	f-nf	threshold_ss_ts3_b1-4x1-4_8_bkorr_c0.2_2_20_f-nf-tr_001.mat	Large SVM
f-nf-hk90.mat	90	f-nf	f-nf	threshold_ss_ts3_b1-4x1-4_8_bkorr_c0.2_2_20_f-nf-tr_001.mat	
le_rvm_r0.1_c4.mat	100	le_svm_r0.1_c4.mat	le_svm_r0.1_c4.mat	le_rvm_r0.1_c4-le_ts_thres_0.05.mat	
nt_rvm_r0.2_c1.mat	100	nt_svm_r0.2_c1.mat	nt_svm_r0.2_c1.mat	nt_rvm_r0.2_c1-nt_ts_thres_0.05.mat	
lm_rvm_r0.1_c4.mat	100	lm_svm_r0.1_c4.mat	lm_svm_r0.1_c4.mat	lm_rvm_r0.1_c4-lm_ts_thres_0.05.mat	
lx_rvm_klr_r0.3_c50p_10n.mat	100	lx_svm_klr_r0.3_c50p_10n.mat	lx_svm_klr_r0.3_c50p_10n.mat	lx_rvm_klr_r0.3_c50p_10n-lx_ts_thres_0.05.mat	
ls_rvm_r0.3_c10.mat	100	ls_svm_r0.3_c10.mat	ls_svm_r0.3_c10.mat	ls_rvm_r0.3_c10--ls_ts_thres_0.05.mat	
lb_rvm_r0.5_c8.mat	100	lb_svm_r0.5_c8.mat	lb_svm_r0.5_c8.mat	lb_rvm_r0.5_c8--lb_ts_thres_0.05.mat	

Table C-2: Trained RVM's

SVM	from Set	Numb SSVs	pos	neg	Comment
bdisc-20x20-rfb2_svm_org.mat	Cog.	11.307 <root_data>\bdisc-20x20-rfb2_ssv_allv	5.679	5.628	best without bootstrapping gamma=1e-6 (bei 0-255), 0.065(0-1) alphabound=2: On Bound = 8832 bdisc-rfb2-1e-6-alpb2-20x20.zip
bdisc-20x20-rfb1	Cog.	16.892 <root_data>\bdisc-20x20-rfb1	7.674	9.218	gamma=1e-6 alphabound=10: On Bound = 6883 Bootstrap bdisc-rfb1-1e-6-alpb10-20x20.zip
bdisc-20x20-rfb2	Cog.	22.130 <root_data>\bdisc-20x20-rfb2	10.591	11.539	gamma=1.5e-6 alphabound=2 On Bound = 13811 Bootstrap bdisc-rfb2-1.5e-6-alpb2-20x20.zip
f-nf-SVM	f-nf-train.mat	769	324	445	gamma=0.06 alphabound=1
v5-SVM	patches_vir-5_train.txt	11831	5242	6589	gamma=0.14 alphabound=1
le_rvm_r0.1_c4.mat	b2_1920\le_tr.mat	1609	557	1052	gamma=0.1

					alphabound=4
nt_svm_r0.2_c1.mat	b2_1920\nt_tr.mat	3238	931	2307	gamma=0.2 alphabound=4
lm_svm_r0.1_c4.mat	b2_1920\lm_tr.mat	1454	610	844	gamma=0.1 alphabound=4
lx_wvm_r0.4_c10.mat	b2_1920\lx_tr.mat	16932	4815	12108	gamma=0.4 alphabound=10
ls_rvm_r0.3_c10.mat	b2_1920\ls_tr.mat	9184	3358	5826	gamma=0.3 alphabound=10
lb_svm_r0.5_c8.mat	b2_1920\lb_tr.mat	16353	4424	11929	gamma=0.5 alphabound=8

Table C-3: Trained SVM's**Syntax W-RVM's:**

MatLab file containing the following matrices:

- num_hk: 1×1 scalar containing the number of all W-RSV's (in general num_hk_wvm*num_lev_wvm).
- num_hk_wvm: 1×1 scalar containing the number of W-RSV's per level.
- num_lev_wvm: 1×1 scalar containing the number of appr. levels per W-RSV's.
- param_nonlin1: 1×5 matrix, see SVM
- area: $1 \times \text{num_hk}$ matrix of structures
 - area(1,i): representation of the i -th W-RSV ($i = (l-1) \text{ num_hk_wvm} + n$, where l is the appr. level and n the n -th W-RSV at this level)
 - val_u: [$1 \times \text{num_v_u}$] vector of num_v_u grey values of the residual $u(n,l)$ (the number of grey values can be smaller than the number of rectangles)
 - val_a: [$1 \times \text{num_v_a}$] vector of num_v_a grey values of the appr. vector ($a(n,l)$ is the sum of residuals over the appr. levels: $u(n,j)$, with $j=1, \dots, l$)
 - cntrec_u: [$1 \times \text{num_v_u}$] vector of number of rectangles per num_v_u grey values of $u(n,l)$
 - cntrec_a: [$1 \times \text{num_v_a}$] vector of number of rectangles per num_v_a grey values of $a(n,l)$
 - rec_u: [$\text{num_v_u} \times \max(\text{cntrec_u})$] matrix of structures of rectangles coordinates of $u(n,l)$
 - coordinates (up/left and down/right point) of the rectangle: $x1, x2, y1, y2$ (if for this grey values exist less than $\max(\text{cntrec_u})$ rectangles the coordinates are set to [])
 - rec_a: [$\text{num_v_a} \times \max(\text{cntrec_a})$] matrix of structures of rectangles coordinates of $a(n,l)$
 - coordinates (up/left and down/right point) of the rectangle: $x1, x2, y1, y2$
 - cntallrec_u: 1×1 scalar with number of all rectangles of $u(n,l)$
 - cntallrec_a: 1×1 scalar with number of all rectangles of $a(n,l)$
 - cntallopr_u: 1×1 scalar with number of all operations to compute the kernel function for $u(n,l)$
 - cntallopr_a: scalar with number of all operations to compute the kernel function for $a(n,l)$
 - cntallval_sum_u: 1×1 sum over the numbers of all grey values (num_v_u) of $u(n,j)$, with $j=1, \dots, l$
 - cntallrec_sum_u: 1×1 sum over the numbers of all rectangles of $u(n,j)$, with $j=1, \dots, l$
 - cntallopr_sum_u: 1×1 sum over the numbers of all operations to compute the kernel function for $u(n,j)$, with $j=1, \dots, l$
 - crec: [$\text{num_v_u} \times \max(\text{cntrec_u})$] same as rec_u, but all coordinates a in C-syntax (i.e. first row and columns are zero not one)

- `app_rsv_convolve`: $1 \times \text{num_hk}$ vector of square values ($u(n,l)' u(n,l)$) of the residual $u(n,l)$ for all W-RSV's, used for fast evaluation of the norm in a RBF kernel based on Integral Images (see $x'x$ in Eq. (2.7) in Section 2.3.1)
- `rvmfile`: sting with name of the used RVM
- `statistics`: structure with many statistics of the W-RVM approximation
- `support_hk<i>`: where $\langle i \rangle$ is a number which runs from 1 to `num_hk`. These 20×20 matrices store the Wavelet Appr. Reduced Set Vectors (as double)
- `weight_hk<i>`: where $\langle i \rangle$ is a number which runs from 1 to `num_hk`. These $1 \times i$ matrices store the weights of the `num_hk` W-RSV's.
- `support_nonlin1`: $h \times w \times N_x$ matrix storing the N_x Support Set Vectors of the full SVM, with h is the height and w the width of a vector.
- `weight_nonlin1`: $1 \times N_x$ matrix storing their weights.

note: images in MatLab run over the columns not over the rows.

Syntax RVM's:

MatLab file containing the following matrices:

- `num_hk`: 1×1 matrix containing the number of RSV's.
- `param_nonlin1`: 1×5 matrix, see SVM
- `support_hk<i>`: where $\langle i \rangle$ is a number which runs from 1 to `num_hk`. These 20×20 matrices store the Reduced Set Vectors (as double)
- `weight_hk<i>`: where $\langle i \rangle$ is a number which runs from 1 to `num_hk`. These $1 \times i$ matrices store the weights of the `num_hk` RSV's
- `support_nonlin1`: $h \times w \times N_x$ matrix storing the N_x Support Set Vectors of the full SVM, with h is the height and w the width of a vector
- `weight_nonlin1`: $1 \times N_x$ matrix storing their weights.

note: images in MatLab run over the columns not over the rows.

Syntax ROC Files:

MatLab file containing the following matrices:

ROC: 3(FRR, FAR, Thres) \times NumbROCPnts).

These files are generated by the program `hk_roc`. It contains the matrices `roc_hk<i>` where i runs from 1 to `num_hk` (the number of sequential RSV's from which this file originates). These matrices are of dimension $3 \times T$, where T is the number of points of that ROC. The first row contains the false negative ratio, the second, the false positive ratio and the third row, the threshold of the RSM for which this point was obtained (or rather the delta threshold, i.e. that number must be added to the full SVM threshold to obtain that point on the ROC).

Syntax SVM's:

MatLab file containing the following matrices:

- `param_nonlin1`: 1×5 matrix
 - element 1 \times 1: threshold of the full SVM (this threshold is subtracted from the sum of kernels to have the SVM evaluation of a patch).

- element 1×2: the type of threshold according to John Platt's notation (for Gaussian kernel, the value is 2).
- element 1×3: this is the Gaussian parameter $b : k(x, y) = \exp(-b\|x - y\|^2)$ (be careful if you compute the kernel at the range [0,1] or [0,255])
- element 1×4 and
- element 1×5: would be used for polynomial kernels
- support_nonlin1: $h \times w \times N_x$ matrix storing the N_x Support Set Vectors of the full SVM, with h is the height and w the width of a vector
- weight_nonlin1: $1 \times N_x$ matrix storing their weights.

note: images in MatLab run over the columns not over the rows.

note: kernel parameter depends on used range:

$\sigma : k(\mathbf{x}_i, \mathbf{z}_i) = \exp(-\sigma\|\mathbf{x}_i - \mathbf{z}_i\|^2)$. Important in which range one works. Training sets are stored as integer in the range: 0-255, but the SSV's are trained and stored as double: 0-1. At classification the range is again: double, 0-255, therefore σ is divided by 255^2 (because $\sigma = 1/\gamma^2$).

$\sigma : 0.01-1$ for range 0-1 (e.g. ls about 0.3, rbf2 0.065; e.g. for training and stored in *.mat)

$\sigma : 1.7e-7 - 1.5e-5$ for range 0-255 (e.g. ls about 4.6e-6, rbf2 1e-6; for classification)

RVM Training, mat: 0-1: $k(\mathbf{x}_i, \mathbf{z}_i) = \exp(-\sigma\|\mathbf{x}_i - \mathbf{z}_i\|^2)$

SVM Classification: 0-255: $k(\mathbf{x}_i, \mathbf{z}_i) = \exp(-\sigma / 255^2 (255\|\mathbf{x}_i - \mathbf{z}_i\|)^2)$

Point ID	MPEG4	Anthropometric point name	1) Invariance of position wrt. to expression, subj. and aging	2) Invariance of the area wrt. scale, pose and illumination	3) Visibility (as percentage of images) 5 all, 4 ¾, 3 ½, 2 ¼, 1 0	4) Reliability of finding the point (depends on the classifier)	5) Saliency (in a certain neighbourhood)	6) Estimability (using locations of other feature points)	7) ISO compatibility (point listed in MPEG4 ISO standard)	8) Accuracy of manual labelling	9) Usefulness for pose estimation	10) Usefulness for the MM fitting.	Average	Resume -Rank for FFD	Comment	How to point
v	11.4	Vertex	-	-	-	-	-	-	-	-	-	-	-	outside of the face	The highest point of head when the head is oriented in FH	
g		Glabella	5	2	5	3	1	4	4	1	2	2	2,9	3		The most prominent middle point between the eyebrows
op		Opisthocranium	-	-	-	-	-	-	-	-	-	-	-	outside of the face	Situated in the occipital region of the head is most distant from the glabella	
eu		Eurion	-	-	-	-	-	-	-	-	-	-	-	outside of the face	The most prominent lateral point on each point of the skull in the area of the parietal and temporal bones	
ft		Frontotemporale	4	2	3	1	1	1	2	1	4	2	2,1	4		The point on each side of the forehead, laterally from the elevation of the linear temporalis
tr	11.1	Trichion	1	4	4	1	2	3	4	2	3	3	2,7	4		The point on the hairline in the midline of the forehead
zv		Zygion	4	1	3	2	2	1	2	1	1	1	1,8	4		The most lateral point of each of the zygomatic
go	2.15 2.16	Gonion	3	2	4	2	2	2	4	2	2	5	2,8	3		The most lateral point on the mandibular angle close to the bony gonion
sl		Sublabiale	2	2	4	2	2	3	2	3	3	3	2,6	4		Determines the lower border of the lower lip or the upper border of the chin
pg	2.10	Pogonion	1	3	5	3	4	2	4	2	3	2	2,9	3		The most anterior midpoint of the chin, located on the skin surface in the front of the identical bony landmark of the mandible
gn	2.1	Menton (or gnathion)	2	2	3	2	3	2	4	2	2	4	2,6	3		The lowest median landmark on the lower border of the mandible
cdl		Condylion laterale	4	3	3	1	2	1	2	1	4	1	2,2	4		The most lateral point on the surface of the condyle of the mandible
en	3.8 3.11	Endocanthion	4	4	4	4	4	4	4	5	4	5	4,2	3	(label available!) ex better	The point at the inner commissure of the eye fissure
ex	3.7 3.12	Exocanthion (or ectocanthion)	4	4	4	4	4	4	4	5	5	5	4,3	2	(label available!) ex better	The point at the outer commissure of the eye fissure
p	3.5 3.6	Center point of pupil	5	4	4	5	5	5	4	5	4	4	4,5	1	label available!	Is determined when the head is in the rest position and the eye is looking straight forward
or	3.9 3.10	Orbitale	3	3	4	4	1	4	4	4	3	4	3,4	4	saliency!	The lowest point on the lower margin of each orbit
ps	3.1 3.2	Palpebrale superius	3	3	4	4	1	4	4	3	3	4	3,3	4	saliency!	The highest point in the midportion of the free margin of each upper eyelid
pi	3.3 3.4	Palpebrale inferius	3	3	4	4	1	4	4	4	3	4	3,4	4	saliency!	The lowest point in the midportion of the free margin of each lower eyelid
os		Orbitale superius	3	3	4	4	1	4	2	2	3	4	3	4	sci better	The highest point on the lower border of the eyebrow
sci	4.3 4.4	Superciliare	2	5	3	4	3	3	4	3	4	5	3,6	3	no analogue for 4.5/6 by Farkas, sci nearest	The highest point on the upper border in the midportion of each eyebrow
n		Nasion	4	4	5	2	1	3	2	1	3	3	2,8	4	g better	The point in the middle of both the nasal root and nasofrontal suture
se		Sellion (or subnasion)	4	4	5	2	1	3	2	1	3	3	2,8	4	g better	Is the deepest landmark located on the bottom of the nasofrontal angle
mf		Maxillofrontale	?	?	?	?	?	?	?	?	?	?	0,2	?		The base of the nasal root medially from each endocanthion
al	9.1 9.2	Alare	2	2	4	3	4	4	4	4	3	4	3,4	3	sbal better	The most lateral point on each alar contour
prn	9.3	Pronasale	2	5	5	2	3	5	4	2	5	5	3,8	1	label available!	The most protruded point of the apex nasi
sn	9.15	Subnasale	5	3	3	4	4	5	4	5	4	4	4,1	3	ls better	The midpoint of the angle at the columella base where the lower border of the nasal septum and the surface of the upper lip meet
sbal		Subalare	4	4	3	4	3	4	2	5	3	5	3,7	2		The point at the lower limit of each alar base, where the alar base disappears into the skin of the upper lip
ac	9.1 9.2	Alar curvature (or alar crest) point	4	2	2	1	2	3	4	2	4	3	2,7	4		The most lateral point in the curved base line of each ala
cph	8.9 8.10	Christa philtri landmark	2	3	4	3	3	4	4	4	3	4	3,4	3	ls better	The point on each elevated margin of the philtrum just above the vermillion line
ls	8.1	Labiale (or labrale) superius	3	4	4	4	4	5	4	5	4	5	4,2	2		The midpoint of the upper vermillion line
li	8.2	Labiale (or labrale) inferius	2	4	4	3	1	2	4	2	3	5	3	3	(label available!)	The midpoint of the lower vermillion line
ch	8.3 8.4	Cheilion	2	2	3	4	5	4	4	5	4	5	3,8	1	(label available!)	The point located at each labial commissure
sa	10.1 10.2	Superaurale	4	2	2	1	2	3	4	2	5	4	2,9	4		The highest point of the free margin of the auricle
sba	10.5 10.6	Subaurale	4	2	3	2	3	4	4	4	5	5	3,6	3		The lowest point of the free margin of the ear lobe
pra	10.9 10.10	Preaurale	4	2	2	1	2	4	4	4	4	3	3	3		The most anterior point on the ear, located just in front of the helix attachment to the head
pa		Postaurale	4	2	2	1	1	2	2	2	4	3	2,3	4		The most posterior point on the free margin of the ear
obs	10.3 10.4	Otobasion superius	4	2	2	1	1	2	4	2	4	4	2,6	4	pra better	The point of attachment of the helix in the temporal region
obi		Otobasion inferius	4	2	3	2	4	4	2	4	5	5	3,5	3	sba better	The point of attachment of the ear lobe to the cheek
po		Porion (soft)	5	2	2	3	4	4	2	4	5	3	3,4	3		The highest point of the upper margin of the cutaneous auditory meatus
t		Tragion	5	2	3	2	2	3	2	4	5	4	3,2	4	po better	The notch on the upper margin of the tragus

Table C-4: Multi-criteria evaluation of optimal facial features

Catalogue of criteria for discussion which facial features to choose for the W-RVM training (see Section 3.3.1; for the localisations of the MPEG-4 and Farkas landmarks see Figure 3-16).

¹⁰ see: Heisele et al. [44], [41], [42], [43]

¹¹ 1: first, 2: second,... 5: last

List of Figures

FIGURE 1-1:	IMAGES ANALYSIS BY UNIFYING A 2D IMAGE-BASED CLASSIFIER AND A 3D FACE MODEL	14
FIGURE 2-1:	TOY EXAMPLE DEMONSTRATING THE SEQUENTIAL RVM.....	22
FIGURE 2-2:	OBJECT DETECTION AS AN EXAMPLE FOR COMPLEX CLASSIFICATION TASKS	24
FIGURE 2-3:	DEFINITION AND ADVANTAGE OF INTEGRAL IMAGES	25
FIGURE 2-4:	USE OF INTEGRAL IMAGES FOR FAST KERNEL EVALUATION	26
FIGURE 2-5:	TEMPLATES CORRESPONDING TO ALGEBRAIC MOMENTS	27
FIGURE 2-6:	EXAMPLES FOR POLYNOMIAL APPROXIMATION OF REDUCED SET VECTORS	28
FIGURE 2-7:	POLYNOMIAL APPROXIMATION OF THE RVM.....	28
FIGURE 2-9:	STAGES OF THE HAAR-LIKE APPROXIMATION USING MORPHOLOGICAL FILTER	30
FIGURE 2-10:	EXAMPLES FOR HAAR-LIKE APPROXIMATIONS	34
FIGURE 2-11:	EXAMPLE OF CASCADED APPROXIMATING A RSV	41
FIGURE 3-1:	RANDOM FACE AND CORRESPONDING VERTEX NUMBERS	46
FIGURE 3-2:	DEFINITION OF THE FACIAL FEATURE AREAS	47
FIGURE 3-3:	GENERATION OF TRAINING SETS FROM THE FERET DATABASE	47
FIGURE 3-4:	GENERATION OF TRAINING SETS WITH SYNTHETIC ENVIRONMENT	49
FIGURE 3-5:	GENERATION OF TRAINING SETS WITH SYNTHETIC TEXTURE	50
FIGURE 3-6:	SYNTHETIC TEXTURE AND SHAPE OF NOT EXISTING INDIVIDUALS	50
FIGURE 3-7:	GENERATION OF NEGATIVE TRAINING SETS	52
FIGURE 3-8:	DISTANCE OF THE HYPER-PLANES AND REJECTIONS OVER NUMBER OF OPERATIONS	53
FIGURE 3-9:	PERCENTAGE OF REJECTION OVER NUMBER OF OPERATIONS	53
FIGURE 3-10:	R.O.C.'S FOR THE SVM, THE RVM AND THE W-RVM	54
FIGURE 3-11:	EXAMPLE DEMONSTRATING FAST REJECTION OF LARGE IMAGE AREAS.....	55

FIGURE 3-12: STANDARD FACIAL FEATURES IN COMPUTER VISION AND MEDICINE	57
FIGURE 3-13: MEDICAL CRITERIA FOR THE EVALUATION OF OPTIMAL FACIAL FEATURES	57
FIGURE 3-14: ANCHOR-POINTS OPTIMAL AS INITIALISATION FOR THE 3D MM FITTING	58
FIGURE 3-15: OPTIMISATION OF THE POSITION AND SIZE OF FACIAL FEATURES BY HEISELE ET AL.....	58
FIGURE 3-16: DEFINITION OF MPEG-4 AND ANTHROPOLOGY LANDMARKS	60
FIGURE 3-17: TRAINED SUPPORT VECTOR MACHINES FOR ALL FEATURES.....	63
FIGURE 3-18: DECREASE OF THE HYPER-PLANES DISTANCE OVER NUMBER OF USED OPERATIONS	64
FIGURE 3-19: TRAINING STAGES OF THE W-RVM FOR THE MOUTH CORNER.....	65
FIGURE 3-20: TRAINED W-RVM'S FOR ALL FACIAL FEATURES	65
FIGURE 3-21: REJECTION RATE OF THE W-RVM _{LS} AND RVM _{LS} OVER NUMBER OF USED OPERATIONS	66
FIGURE 3-22: PDF ESTIMATION BY NON-PARAMETRIC HISTOGRAM METHOD	68
FIGURE 3-23: HISTOGRAM METHOD IS SENSITIVE TO THE SIZE OF THE BINS	68
FIGURE 3-24: PDF ESTIMATION BY NON-PARAMETRIC K-NN ESTIMATION.....	69
FIGURE 3-25: PROBABILISTIC W-RVM USING SIGMOID FUNCTION FITTING	70
FIGURE 3-26: STABLE PROBABILISTIC W-RVM FOR ALL FACIAL FEATURES.....	71
FIGURE 3-27: EXAMPLE FOR DETECTION PROCESS OF THE LEFT EYE FOR ONE IMAGE.....	73
FIGURE 3-28: EXAMPLES FOR DETECTION RESULTS OF THE LEFT MOUTH CORNER	73
FIGURE 3-29: REDUCING FRR BY A CASCADED FRAMEWORK AND USE OF A FOI.....	74
FIGURE 3-30: STAGES OF THE W-RVM WITH TEN FACIAL FEATURES.....	75
FIGURE 3-31: GOOD EXAMPLES FOR FINAL FEATURE ASSORTMENTS BY MAXIMUM RULE	75
FIGURE 3-32: NOT SATISFYING EXAMPLES FOR THE FINAL FEATURE ASSORTMENTS	76
FIGURE 3-33: FINAL FEATURE ASSORTMENT USING SOLELY THE 2D APPEARANCE MODEL	77
FIGURE 3-34: UNIFYING 3D PRIOR SHAPE MODEL AND W-RVM AS 2D APPEARANCE MODEL	77
FIGURE 3-35: PROBABILITY OF A FACIAL FEATURE POINT GIVEN MODEL PARAMETERS.....	80
FIGURE 3-36: ESTIMATION OF THE FINAL FEATURE ASSORTMENT USING THE PSM	82
FIGURE 3-37: FACIAL FEATURE SET LOCALISATION ACCURACY	83
FIGURE 4-1: APPLICATIONS TAKING ADVANTAGE OF THE UNIFICATION OF THE 3D AND 2D MODEL	85

FIGURE 4-2:	W-RVM FACE DETECTION INTEGRATED INTO FAFaDe.....	87
FIGURE 4-3:	FACIAL FEATURE SET DETECTION INTEGRATED INTO FAFaFeDe	88
FIGURE 4-4:	POSE ESTIMATION INTEGRATED IN FAFaFeDe	89
FIGURE 4-5:	FACE DETECTION INTEGRATED INTO Fd_CAMFFDVIEWER.....	90
FIGURE 4-6:	THE I-SEARCH PROJECT	91
FIGURE 4-7:	EXAMPLE ITWM VIDEO SEQUENCE WITH RESULTS OF THE FACE DETECTION	91
FIGURE 4-8:	SCHEMATA FOR HUMAN IN COMPUTER INTERACTION (HCI, [37]).....	92
FIGURE 4-9:	CONDENSATION FOR FACE AND FACIAL FEATURE TRACKING.....	94
FIGURE 4-10:	FACIAL MOTION TRACKING USING A STEREO CAMERA INSTALLATION.....	94
FIGURE 4-11:	AVATAR FOLLOWING WITH EYE AND HEAD MOTION.....	95
FIGURE 4-13:	SWITCHING FACES – HCI APPLICATION IN THE FIELD OF PERCEPTION PSYCHOLOGY	98
FIGURE 4-14:	TRACKING OF HIGHER FEATURE PARAMETERS.....	116
FIGURE 4-15:	ADAPTIVE REAL-TIME LEARNING USING CONDENSATION AND L2-NORM.....	118
FIGURE 5-1:	FURTHER UNIFICATION OF THE W-RVM AND THE 3D MM-FITTING	99
FIGURE 5-2:	MORPHABLE MODEL FACE DATABASE.....	100
FIGURE 5-3:	MORPHABLE MODEL – LEARNING FROM EXAMPLES.....	101
FIGURE 5-4:	REGISTRATION.....	102
FIGURE 5-5:	ANALYSE BY SYNTHESIS	102
FIGURE 5-6:	SECOND STAGE OF THE W-RVM FACIAL FEATURE SET DETECTOR FOR TEN FEATURES	103
FIGURE 5-7:	THIRD STAGE OF THE W-RVM FACIAL FEATURE SET DETECTOR USING THE PSM.....	104
FIGURE 5-8:	COMPARISON OF THE MM-FITTINGS STAGES.....	105
FIGURE 5-9:	COMPARISON OF THE RENDERINGS FROM THE OBTAINED FITTINGS	106
FIGURE 5-10:	COMPARISON OF MM-FITTINGS USING ANCHOR POINTS SET BY W-RVM AND MANUALLY	107
FIGURE 5-11:	FIX POINT ITERATION FOR SINGLE-STAGE W-RVM	111
FIGURE 5-12:	MULTI-FEATURE AND MULTI-INVARIANT W-RVM	112
FIGURE 5-13:	ACCURATE W-RVR RESULTS ESTIMATING THE ROLL ANGLE OF EYES	114
FIGURE 5-14:	WAVELET APPR. VECTOR REGRESSION IS 11- TO 75-FOLD MORE EFFICIENT AS SVR.....	114

FIGURE A-1: UML DIAGRAM OF THE W-RVM TRAINING	125
FIGURE A-2: UML DIAGRAM OF THE W-RVM DETECTION	131
FIGURE A-3: THE FAST FACIAL FEATURE DETECTOR (FAFAFEDE)	133
FIGURE A-4: FAFAFEDE TOOLBAR	134
FIGURE A-5: FAFAFEDE VIDEO FORMAT AND SOURCE DIALOG	134
FIGURE A-6: FAFAFEDE STATUS BAR AND KEYS FOR ADJUSTING SOME OF THE CONTROL PARAMETER	134

List of Tables

TABLE 2-1:	SUMMARY OF THE TRAINING OF THE W-RVM CLASSIFIER	40
TABLE 2-2:	SUMMARY OF THE WORKING STAGES OF THE W-RVM CLASSIFIER	42
TABLE 3-1:	COMPARISON OF THE EFFICIENCY OF THE APPROACHES.....	54
TABLE 3-2:	OPTIMAL FEATURES IN THE SCENES OF RECOGNITION RATE AND INVARIANCE	59
TABLE 3-3:	CRITERIA FOR EVALUATION OF OPTIMAL FACIAL FEATURES.....	59
TABLE 3-4:	PART OF THE EVALUATION TABLE FOR EVALUATION OF OPTIMAL FACIAL FEATURES	60
TABLE 3-5:	FACIAL FEATURE POINTS CHOSEN FOR THE THESIS	61
TABLE 3-6:	STAGES OF A SINGLE W-RVM DETECTOR	72
TABLE 3-7:	STAGES OF THE W-RVM FACIAL FEATURE SET DETECTOR.....	76
TABLE B-1:	SETS PATCHES	144
TABLE B-2:	SETS OF IMAGES USED FOR VALIDATION AND EXPERIEMENTS.....	144
TABLE B-3:	FACE RECOGNITION TECHNOLOGY (FERET) PROGRAM DATABASE.....	145
TABLE B-4:	MIT-CMU FACE DETECTION DATABASE.....	145
TABLE B-5:	BIOID FACE DATABASE	145
TABLE C-1:	TRAINED W-RVM'S	147
TABLE C-2:	TRAINED RVM'S	148
TABLE C-3:	TRAINED SVM'S.....	149
TABLE C-4:	MULTI-CRITERIA EVALUATION OF OPTIMAL FACIAL FEATURES	152

Curriculum Vitae

Address	Lindstedter Straße 15 14469 Potsdam, Germany
Tel.	+ 49 176 61 29 61 33
Email	matthias.raetsch@unibas.ch
Date of birth	12 December 1966
Marital status	Single
Children	Darja Soleil (12), Jannes Danjar (5)
Citizenship	Germany

Education

1974 – 1984	Polytechnic Secondary School (degree 'with honors') in Lychen
1984 – 1986	Extended Secondary School (A-levels diploma 'with honors') in Templin
1988 – 1994	Master of Education studies for mathematics, physics and later for computer science at the University of Potsdam (German State Exam thesis in the field of numerical mathematics)
1992 – 1993	Assistant Teacher for conversation and computer science at the Chrieff and the Auchterarder High School in Perthshire, UK, and 'Cambridge First Certificate in English' at the Perth College, UK
1994 – 2002	Diploma studies for computer science at the University of Potsdam (diploma-degree 'with honors'), thesis on segmenting Touching Characters for Optical Character Recognition (Patent No: 195 33 585).
1996 – 1999	Research Fellowship, Industry Research Foundation Köln, Digital Image Processing for the Pre-processing of Optical Character Recognition Systems.
since 2005	Philosophical Doctorate at the Graphics and Vision Research Group (GraVis) at the University of Basel, Department Computer Science, Switzerland.

Employment

- Apr. 1994 – Jan. 1999 Scientific Assistant by Prof. P. Maaß, University of Potsdam, Germany in cooperation with WiSenT GmbH in the field of Machine Learning and Optical Character Recognition
- Feb. 1999 – Sep. 1999 Software Engineer at WiSenT GmbH in Potsdam, Germany
- Oct. 1999 – Apr. 2001 Research Associate by Prof. P. Maaß at the Bremen University, Germany in cooperation with CONTEX GmbH, Machine Learning, Computer Vision, Director Algorithm Development
- May 2001– Mar. 2003 Project Manager and Director Algorithm Development at WiSenT GmbH in Potsdam, Germany
- Apr. 2003 – June 2005 Research Associate at the University of Freiburg, Germany, BMBF Joined Project: High-Performance Online Image Search. I-Search, Kaiserslautern, Machine Learning, and Face Detection
- since July 2005 Phd Student and Assistant Lecturer at University of Basel, Computer Science Department, Switzerland.
- Apr. 2007 – Nov. 2007 Research Associate at Konrad Zuse Institute Berlin, Germany, Numerical Analysis and Modelling by Prof. Teschke
- since Dec. 2007 Research Associate at University of Applied Sciences Neubrandenburg, Germany, Signal and Image Processing, Ill-posed and Inverse Problems, Geomathematics by Prof. Teschke

Patent

Title: *Approaches for Segmentation of Characters* (Verfahren zur Segmentierung von Zeichen). German Patent Office, Patent number: 195 33 585. January 1996.

Publications

M. Rättsch, G. Teschke, S. Romdhani, and T. Vetter. *Wavelet Frame Accelerated Reduced Support Vector Machines*, IEEE Transactions on Image Processing, Vol. X, No. XX. (accepted), Sep. 2006

M. Rättsch, S. Romdhani, G. Teschke, and T. Vetter. *Over-complete Wavelet Approximation of a Support Vector Machine for Efficient Classification*. DAGM'05: 27th Pattern Recognition Symposium, Vienna. 2005

M. Rättsch. *Fast Detection Module for Face Detection in Images*. High Performance Online Image Search. I-Search: BMBF, DLR: 01 IR B02 B. Kaiserslautern, Oct. 2005.

M. Rättsch, S. Romdhani, and T. Vetter. *Efficient Face Detection by a Cascaded Support Vector Machine using Haar-like Features*. DAGM'04: 26th Pattern Recognition Symposium, Tübingen. pp 62-70, 2004

M. Rättsch. *Digital Image Processing to Eliminate Bondings by Touching Characters*. Diploma thesis, Computer Science Department, University of Potsdam, 2002.

M. Rättsch. *Digital Image Processing for the Pre-processing of Images for Optical Character Recognition Systems*. Research fellowship thesis, Industry Research Foundation Köln, 1997.

M. Rättsch. *Pre-processing of Images to Improve Optical Character Recognition Systems*. Intermediate diploma thesis, Computer Science Department, University of Potsdam, November 1996.

M. Rättsch. *Numerical Methods for Evaluation of Equilibrium Constants of Electron-Donator-Acceptor Complexes*. German State Exam thesis (Master of education for math and physics), Department of Numerical Mathematics and Physics, University of Potsdam, January 1994.

Bibliography

- [1] Y. Andreopoulos. A new method for complete to overcomplete discrete wavelet transform. In Proc. *International Conference on Digital Signal Processing*, Santorini, Greece, pp. 501-504, 2002.
- [2] H. Bässmann, P. W. Besslich. *Ad Oculos*, Heidelberg: *Springer Verlag*, 1991.
- [3] J. Batliner. Multidimensional Face Tracking using Condensation methods. *Bachelor thesis, University of Basel, GraVis*, Basel, 2007.
- [4] R. Bellman. On the Theory of Dynamic Programming. *Proceedings of the National Academy of Sciences*, 1952.
- [5] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. *SIGGRAPH '99*, Conference Proceedings, pp. 187-194 (Impact Paper Award), 1999.
- [6] V. Blanz and T. Vetter. Reconstructing the complete 3d shape of faces from partial information. *it+ti – Informationstechnik und Technische Informatik*, 44:295-302, 2002.
- [7] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *PAMI*, 2003.
- [8] C. Burrus, R. Gopinath, H. Guo, Introduction to Wavelets and Wavelet Transforms: A Primer, *Prentice Hall*, New Jersey, 1998
- [9] C. Burges, Simplified support vector decision rules. In *13th International Conference on Machine Learning*, pp. 71-77, 1996.
- [10] B. E. Boser, I.M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifier. In D. Haussler, editor, Proc. of the *5th ACM Workshop on Computational Learning Theory*, pages 144-152, Pittsburgh, PA, 1992. ACM Press.
- [11] P. Campadelli, R. Lanzarotti. Localization of Facial Features and Fiducial Points. In Processings of the *International Conference Visualization, Imaging and Image Processing*, Malaga (Spagna), pp. 491-495, 2002.
- [12] P. Chappuis, D. Blanc, L. Mäglin, and L. Oldani. Face Motion Tracking with web cams. Department computer science, *FHNW*, Muttenz, 2006.
- [13] P. Chappuis and D. Blanc. Realtime face tracking. Diploma Theses, Department computer science, *FHNW*, Muttenz, 2007.
- [14] L. Chen, L. Zhang, H. Zhang, and M. Abdel-Mottaleb. 3D Shape Constraint Facial Feature Localization Using Probabilistic-like Output. In Proc. of the *6th IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

- [15] A. Cohen, R. DeVore, P. Petrushev, and H. Xu. Nonlinear Approximation and the Space $BV(\mathbb{R}^2)$. *American Journal of Mathematics*, (121):587-628, 1999.
- [16] R. Coifman and D. Donoho. Translation invariant de-noising. In *Lecture Notes in Statistics: Wavelets and Statistics*, New York: Springer-Verlag, pp. 125-150, 1995.
- [17] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models – their training and application. *CVIU*, 1995.
- [18] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR*, 2005.
- [19] F. Crow. Summed-area tables for texture mapping. In Proc. of *SIGGRAPH*, 18(3):207-212, 1984.
- [20] W. Dahmen. Stability of multiscale transformations. *The Journal of Fourier Analysis and Applications*, 2, 341-361, 1996.
- [21] I. Daubechies. Ten Lectures on Wavelets. *SIAM*, Philadelphia, 1992.
- [22] I. Daubechies: Wavelet transforms and orthonormal wavelet bases. *Proceedings of Symposia in Applied Mathematics*, (47), 1993.
- [23] I. Daubechies and G. Teschke. Variational image restoration by means of wavelets: simultaneous decomposition, deblurring and denoising. *Applied and Computational Harmonic Analysis*, 19(1):1-16, 2005.
- [24] I. Daubechies, G. Teschke and L. Vese, On some iterative concepts for image restoration, *Advances in Imaging and Electron Physics*, Vol. 150, 2008.
- [25] J. Diessl and B. Groß. ATTACK + SWAP media installation, Free Frame plug-in for *VVVV*. http://www.vvvv.org/tiki-index.php?page=benefit_projects01, Juli 2006
- [26] R. DeVore, B. Jawerth, and V. Popov. Interpolation of besov spaces. *Trans. Math Soc.*, 305, pp. 397-414, 1988.
- [27] R. A. DeVore, B. Jawerth, and V. Popov. Compression of wavelet decompositions. *American Journal of Mathematics*, Vol. 114, No. 4, pp. 737-785, Aug., 1992.
- [28] L. G. Farkas. Anthropometry of the Head and Face. Second edition, *Raven Press*, New York, 1994.
- [29] P. Felzenszwalb and D. Huttenlocher. Pictorial structure for object recognition. *IJCV*, 61(1), 2005.
- [30] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005.
- [31] T. Frank. Wavelet Approximated Reduced Vector Machine for Regression. *Bachelor thesis, University of Basel, GraVis*, Basel, 2007.
- [32] M. Frazier, and B. Jawerth. A discrete transform and decompositions of distribution spaces. *Journal of Functional Anal.*, 93, 34-170, 1990.

- [33] Free Frame. Open-source cross-platform for real-time video effects plug-in systems, <http://freeframe.visualvinyl.net>
- [34] R. Frischholz, U. Dieckmann. BioID: A Multimodal Biometric Identification System. In *IEEE Computer*, Vol. 33, No. 2, February 2000.
- [35] C. Garcia, G. Zikos, and G. Tziritas. Face detection in color images using wavelet packet analysis. *IEEE Int. Conf. on Multimedia Computing and Systems*, 1999.
- [36] V. K. Goyal, M. Vetterli, and N. T. Thao. Quantized Overcomplete Expansions in RN: Analysis, Synthesis, and Algorithms, *IEEE Trans. Inform. Theory*, vol. 44, pp. 16-31, Jan. 1998.
- [37] GraVis: Graphics and Vision Research Group, University of Basel, Switzerland, <http://gravis.cs.unibas.ch/>.
- [38] B. Groß. Attack & Swap, FaceDetection Installation for the new Eikones Building in Basel, Switzerland, <http://www.looksgood.de/log/2006/10/20/attack-swap>, Oct 2006.
- [39] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, H. Kälviäinen, and J. Matas. Feature-Based Affine-Invariant Localization of Faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 9, 1490-1495, 2005.
- [40] B. Haasdonk. Transformation Knowledge in Pattern Analysis with Kernel Methods. *PhD thesis, Computer Science Department, University of Freiburg*, Freiburg, May 2005.
- [41] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. *AI Memo 1687*, Massachusetts Institute of Technology, 2000.
- [42] B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 657-662, Hawaii, 2001.
- [43] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by Learning and Combining Object Parts. *NIPS*, 1239-1245, 2001.
- [44] B. Heisele and T. Koshizen. Components for Face Recognition. *FGR*, 153-158, 2004.
- [45] C. Horisberger. I SEE YOU – 3D Head and Gaze Simulation for Visual Tracking. *Bachelor thesis, University of Basel, GraVis*, Basel, 2006.
- [46] J. Huang, X. Shao, and H. Wechsler. Face Pose Discrimination Using Support Vector Machines (SVM), *IEEE proc. of 14th International Conference on Pattern Recognition*, Brisbane, Queensland, Australia, pp.154-156, 1998.
- [47] J. Huang, H. Wechsler. Eye Detection Using Optimal Wavelet Packets and Radial Basis Functions (RBFs). *IJPRAI* 13(7): 1009-1026, 1999.
- [48] D.A. Karras. Improved defect detection in textile visual inspection using wavelet analysis and support vector machines. *ICGST International Journal on Graphics, Vision and Image Processing*, 2005.

- [49] D. Keren, M. Osadchy, and C. Gotsman. Antifaces: a novel, fast method for image detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:747-761, July 2001.
- [50] W. Kienzle, G. H. Bakir, M. O. Franz, and B. Schölkopf. Efficient approximations for support vector machines in object detection. Proc. *DAGM'04*, pages 54-61, 2005.
- [51] K. Kirchberg, O. Jesorsky, R. Frischholz. Genetic Model Optimization for Hausdorff Distance-Based Face Localization. In *International ECCV Workshop on Biometric Authentication, Springer, Lecture Notes in Computer Science, LNCS-2359*, pp. 103-111, Copenhagen, Denmark, June 2002.
- [52] E. B. Koller-Meier. Extending the Condensation Algorithm for Tracking Multiple Objects in Range Image Sequences. *Phd thesis, Swiss Federal Institute of Technology*, 2000.
- [53] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking, *International Journal of Computer Vision*, 1998.
- [54] I-Search cooperation project, High Performance Online Image Search: <http://www.itwm.fhg.de/bv/projects/I-SEARCH/>.
- [55] P. Li, T. Zhang, A. E. C. Pece. Visual contour tracking based on particle filters. *Image Vision Comput.* 21(1): pp. 111-123, 2003.
- [56] S. Mallat. A Wavelet Tour of Signal Processing 2nd edition, *Academic Press*, 1999.
- [57] A. M. Müller, Academy of Art and Design Basel, University of Applied Sciences Northwestern Switzerland. HGK: <http://www.fhnw.ch/hgk/ivk/>.
- [58] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio. Full-body person recognition system. *Pattern recognition*, 36(9):1997-2006, 2003.
- [59] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *CVPR*, pp. 130-136, 1997.
- [60] J. Pierrard and T. Vetter. Skin Detail Analysis for Face Recognition. IN: *Proceedings of CVPR'07*, Minneapolis, USA, 2007.
- [61] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi. The feret evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, October 2000.
- [62] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, Overview of the face recognition grand challenge, *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.
- [63] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. in *Advances in Large Margin Classifier*, MA: MIT Press, Cambridge, 2000.
- [64] M. Pontil and A. Verri. Support vector machines for 3d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637-646, 1998.

- [65] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. Numerical Recipes in C. The Art of Scientific Computing, Second Edition, *Cambridge University Press*, 1999.
- [66] M. Rättsch. Digital Image Processing for the Pre-processing of Images for Optical Character Recognition Systems. *Research fellowship thesis, Industry Research Foundation Köln*, 1997.
- [67] M. Rättsch. Digital Image Processing to Eliminate Bondings by Touching Characters. *Diploma thesis, University of Potsdam*, 2002.
- [68] M. Rättsch, S. Romdhani, G. Teschke, and T. Vetter. Over-Complete Wavelet Approximation of a Support Vector Machine for Efficient Classification. Proc. *DAGM'05: 27th Pattern Recognition Symposium*, Vienna, 2005.
- [69] M. Rättsch. Fast Detection Module for Face Detection in Images. High Performance Online Image Search. *I-Search*. Oct. 2005.
- [70] M. Rättsch, S. Romdhani, and T. Vetter. Efficient Face Detection by a Cascaded Support Vector Machine Using Haar-Like Features. Proc. *DAGM'04: 26th Pattern Recognition Symposium*, pages 62-70, 2004.
- [71] M. Rättsch, G. Teschke, S. Romdhani, and T. Vetter. Wavelet Frame Accelerated Reduced Support Vector Machines, *IEEE Transactions on Image Processing*, (submitted Sept. 2006).
- [72] M. Rautenberg. MPEG-4: Ein neuer Standard für mehr Funktionalität in Multimedia Anwendungen, in *Informatik Spektrum* 22, pp. 82-87, 1999.
- [73] S. Romdhani, P. Torr, B. Schölkopf, and A. Blake. Computationally efficient face detection. In Proceedings of the *8th International Conference on Computer Vision*, July 2001.
- [74] S. Romdhani, N. Canterakis, and T. Vetter. Selective vs. global recovery of rigid and non-rigid motion. *Technical report, CS Dept, University of Basel*, 2003.
- [75] S. Romdhani, P. Torr, B. Schölkopf, A. Blake. Efficient face detection by a cascaded support-vector machine expansion In *Proceedings of The Royal Society A*, 460(2501):3283-3297, November 2004.
- [76] S. Romdhani and T. Vetter. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *CVPR*, 2005.
- [77] S. Romdhani. Face Image Analysis using a Multiple Feature Fitting Strategy. *PhD thesis, University of Basel*, January 2005.
- [78] S. Romdhani and T. Vetter. 3D Probabilistic Feature Point Model for Object Detection and Recognition. In *CVPR*, 2007.
- [79] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *PAMI*, 20:23-38, 1998.

- [80] J. Rurainsky and P. Eisert. Eye Center Localization Using Adaptive Templates. Proc. of the *First IEEE CVPR Workshop on Face Processing in Video*, Washington, USA, June 2004.
- [81] H. Sahbi, D. Geman. A Hierarchy of Support Vector Machines for Pattern Detection. in: *Machine Learning Research*, 2005.
- [82] K. Scherbaum, M. Sunkel, H.-P. Seidel, V. Blanz. Prediction of Individual Non-Linear Aging Trajectories of Faces. In *Computer Graphics Forum, Eurographics*, 26(3), 2007.
- [83] H.-J. Schmeisser and H. Triebel. Topics in Fourier Analysis and Function Spaces. *John Wiley and Sons*, New York, 1987.
- [84] H. Schneiderman and T. Kanade. Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition. In *IEEE CVPR*, July, 1998, pp. 45-51.
- [85] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. *IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 1:746-751, 2000.
- [86] B. Schölkopf, C. J. C. Burges, and A. J. Smola. Advances in Kernel Methods – Support Vector Learning. *MIT Press*, Cambridge, MA, 1999.
- [87] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000-1017, 1999.
- [88] B. Schölkopf and A. J. Smola. Learning with Kernels. *MIT Press*, Cambridge, MA, 2002.
- [89] H. Schweitzer, J. Bell, and F. Wu. Very Fast Template Matching. Proc. *Seventh European Conf. Computer Vision*, pp. 358-372, 2002.
- [90] T. Sim, S. Baker, and M. Bsat, The CMU Pose, Illumination, and Expression Database *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, pp. 1615-1618, December, 2003.
- [91] E. Stollnitz, T. DeRose, and D. Salesin, Wavelets for Computer Graphics: A Primer, Part 1, *IEEE Computer Graphics and Applications*, 15(3), pp. 76-84, May 1995.
- [92] E. Stollnitz, T. DeRose, and D. Salesin, Wavelets for Computer Graphics: A Primer, Part 2, *IEEE Computer Graphics and Applications*, 15(4), pp. 75-85, July 1995.
- [93] K.-K. Sung and T. Poggio at MIT (Test Set B), and H. Rowley, S. Baluja, and T. Kanade (Test Sets A,C and the rotated test set) at CMU, MIT-CMU Face Detection Database, http://vasc.ri.cmu.edu/idb/html/face/frontal_images/index.html.
- [94] A. Teolis. Computational Signal Processing With Wavelets (Applied and Numerical Harmonic Analysis). *Birkhauser Boston*, March 1, 1998.
- [95] D. Terzopoulos and R. Szeliski. Tracking with Kalman snakes. In *Active Vision*, 3-20, MIT, 1992.

- [96] T. Thies, and F. Weber, Optimal reduced-set vectors for support vector machines with a quadratic kernel. *Neural Computation*, vol. 16, no. 9, pp. 1769-1777, Sep. 2004.
- [97] A. Thomson, Issues at stake in eighteenth-century racial classification, *Cromohs*, 8: pp. 1-20, 2003.
- [98] H. Triebel. Interpolation Theory, Function Spaces, Differential Operators. *Verlag der Wissenschaften*, Berlin, 1978.
- [99] University of Applied Sciences Neubrandenburg, Dept. of Landscape Architecture, Geoinformatics, Geodesy and Civil Engineering, Mathematics, Geometry and Applied Computer Sciences: <http://www.fh-nb.de/geoinf/lehre/index.asp>.
- [100] V. Vapnik. Statistical Learning Theory. *John Wiley*, New York, 1998.
- [101] T. Vetter. Synthesis of novel views from a single face image. *International Journal of Computer Vision*, 28:2, pp. 103-116. 1998
- [102] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
- [103] P. Wang, M. Green, Q. Ji and J. Wayman. Automatic Eye Detection and Its Validation. *IEEE Workshop on Face Recognition Grand Challenge Experiments (with CVPR)*, San Diego, CA, June 2005.