FUN ROBOTIC
OUTDOOR GUIDE

Deliverable: D3.1

Final feature extraction component

Consortium

UNIVERSITEIT VAN AMSTERDAM (UvA)
YDREAMS - INFORMATICA S.A. (YD)
IDMIND - ENGENHARIA DE SISTEMAS LDA (IDM)
UNIVERSIDAD PABLO DE OLAVIDE (UPO)
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE (ICL)
UNIVERSITY OF TWENTE (UT)

Grant agreement no. **288235**
Funding scheme **STREP**

UNIVERSITEIT TWENTE.

DOCUMENT INFORMATION

Project acronym:	FROG
Project full title:	Fun Robotic Outdoor Guide
Grant agreement no.:	288235
Funding scheme:	STREP
Project start date:	1 October 2011
Project duration:	36 Months
Call topic:	ICT-2011.2.1 Cognitive Systems and Robotics (a), (d)
Project web-site:	www.frogrobot.eu

Document	
Deliverable number:	D3.1
Deliverable title:	Final feature extraction component
Due date of deliverable:	M24 – 30 September 2013
Actual submission date:	M25 – 8 October 2013
Editors:	Imperial
Authors:	Imperial
Reviewers:	UvA
Participating beneficiaries:	UvA, Imperial
Work Package no.:	3
Work Package title:	Behaviour detection and human-aware guidance
Work Package leader:	Imperial
Work Package participants:	1,4,5
Estimated person-months for deliverable:	33
Dissemination level:	Public
Nature:	Demonstrator
Version:	
Draft/Final:	Final
No of pages (including cover):	31
Keywords:	Pedestrian Detection, Pedestrian Orientation, Facial Feature Extraction, Facial Landmarks

Summary:

A stereo vision system for pedestrian detection and orientation estimation was successfully developed, speed-optimized and integrated onto the FROG robot. Stand-alone evaluation suggests that perception performance is good; this remains to be confirmed in connection with the other FROG modules, especially with the module on navigation. Some further parameter optimization of the joint system will be beneficial.

A 'Visual face analysis' module was developed for FROG project. The function of this module is to detect facial landmarks. The module is based on a novel discriminative regression based approach for the Constrained Local Models (CLMs) framework, referred to as the Discriminative Response Map Fitting (DRMF) method. The method shows impressive performance in the generic face fitting scenario. It is suitable for handling dynamic backgrounds, significant amount of clutter/ occlusions and changing illumination conditions. This module also supports the further development of a set of visual methods for detecting human affective states including users' positive and negative reactions to FROG robot and their overall level of interest and engagement when interacting with the robot. These visual methods are to be developed for future deliverables in FROG project.

Fun Robotic Outdoor Guide

Work package 3.1 description

J. Shen, I. Marras and M. Pantic
Imperial College London

M. C. Liem, J. F. P. Kooij and D. M. Gavrila
University of Amsterdam

Chapter 1

Introduction

In FROG project, the robot should be able to navigate outdoors, in typical tourist scenarios. There are two main tasks to be performed:

- Pedestrian detection provides the necessary initialization for person tracking, at larger distances to the robot (up to about 25 m). At closer distance (up to about 15m), the overall person body orientation is estimated. This provides important cues regarding the focus of attention and willingness to interact. FROG uses stereo-vision, for its accurate object localization and discrimination capability. The person detection and body orientation estimation component is described in Chapter 2.

- When the persons are quite close to the robot, facial expression analysis has to be performed. To do this, a method for feature extraction suitable for handling dynamic backgrounds, significant amount of clutter/ occlusions and changing illumination conditions, is needed. For the face and the head, to be observed with a high-resolution camera, multi-person face detection, face tracking, and head pose estimation, regardless of head pose, clutter, and variations in lighting conditions need to be solved. The problem is difficult due the fact that not only the observed persons move and may occlude each other, but that the observation camera moves as well and may jitter due to the movements of the robot. Robust, fast and effective image registration should be developed. To enable feature-based facial gesture and gaze recognition, facial feature points tracking in the detected face region are attempted as well. This task aims further to develop a set of visual methods for detecting human affective states including user' positive and negative reactions to FROG robot and their overall level of interest and engagement in the current interaction with FROG. The feature extraction method developed for FROG project, is based on a novel discriminative regression based approach for the Constrained Local Models (CLMs) framework, referred to as the Discriminative Response Map Fitting (DRMF) method, which shows impressive performance in the generic face fitting scenario. Face analysis is described in Chapter 3.

Chapter 2

Person detection and body orientation estimation module

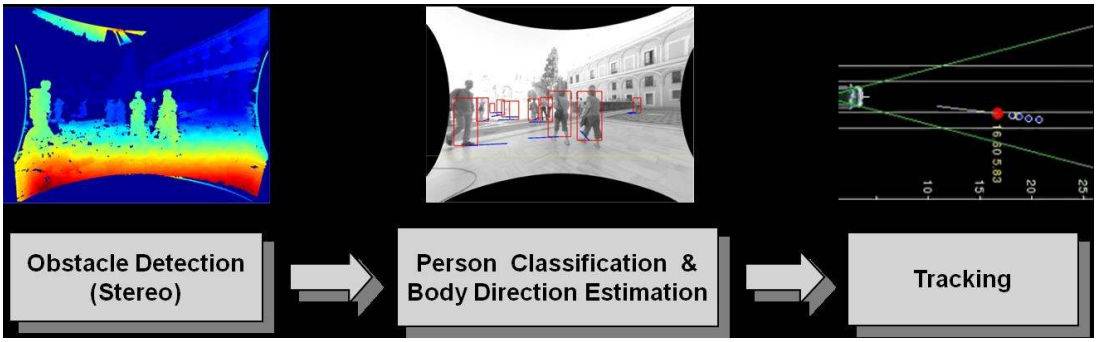


Figure 2.1: Three step person detection and orientation estimation. (1) Obstacle detection using stereo information. (2) Person classification and orientation estimation using HOG features and SVM classifier. (3) Person position and orientation tracking.

An important part of the FROG system is the detection of potential persons of interest in front of the robot. Much research has been done on the subject of person detection, tracking and pose estimation. Much previous work however assumes stationary cameras, controlled backgrounds, single persons in the scene, several overlapping cameras or a combination thereof. Extensive overviews of the field have been provided in [22] and [12].

For the FROG project, the challenge consists of dealing with a moving platform in dynamic, outdoor environments with uncontrollable illumination. These environments can contain many persons, giving rise to a significant amount of occlusions. Persons are to be located at distances between 3 and 25 meters in front of the robot. This is done in three steps: obstacle detection, person classification and tracking. Furthermore, together with person classification, the person orientation is determined. This can be

used to determine if the persons of interest are facing the robot or facing away from the robot. Furthermore, this information can be used to determine if multiple people standing close to each other are likely to be interacting with each other. This kind of information is useful in order to determine if the robot should approach the persons and offer help. An overview of the three steps is given in Figure 2.1.

Data is captured using two calibrated Dalsa HM1400 XDR cameras in a small baseline stereo setup, using a baseline of 22 cm. We use a 6 mm lens for a field of view of 82^o horizontally and 61^o vertically. The stereo rig is placed at about 1.2 m height. The stereo cameras are aligned such to have their optical axes approximately parallel to each other and to the ground plane. The cameras record 10 bit grayscale images at 1400 x 1024 pixels. Automatic gain and shutter control were implemented to control image contrast and illumination while recording, without halting person detection. Camera calibration is performed offline using OpenCV stereo calibration tools. Using the calibration information, a fixed ground plane is computed using markers put on the floor in front of the cameras. Stereo computation is performed using the Library for Efficient Large-scale Stereo Matching (LIBELAS) [14], which offers high speed high quality stereo disparity map computations. An example of a stereo disparity map can be found in the leftmost image in Figure 2.1. A picture of the stereo setup used can be found in Figure 2.2.

Figure 2.2: The stereo setup used for the FROG platform.

The use of stereo information is beneficial for selecting ROI to be processed by the person classifier. Instead of using a calibrated stereo rig, other options would be to use a time of flight camera or a structured light solution like offered by Microsoft's Kinect sensor. While these alternative methods offer fast and accurate depth reconstructions,

the major issue with these systems lies in the fact that they rely on active illumination. Time of flight cameras measure the time it takes to receive the reflections of a light pulse emitted by the camera, while structured light systems project a regular pattern onto the scene and record the pattern's distortion. When using such systems outside, the light emitted by the cameras is quickly overpowered by the environment lighting. This is especially the case in outside environments. Furthermore, the depth range for these sensors is much more limited than when using a stereo setup, since the projected light dissipates at distances further away from the camera. Depending on the baseline used for a stereo setup, a stereo system can see further away from the camera and is only disturbed by illumination conditions when the camera's gain and shutter settings are incorrect.

In the first step, obstacle detection, stereo information is leveraged to determine which locations in the scene have sufficient depth support for a person. This is done similar to the method described in [17]. Regions of interest (ROIs) are computed for fixed locations on the ground plane. ROIs are placed on the ground plane at 46 equally spaced intervals between 2 and 25 meters from the camera. Multiple ROI scales are used, representing 16 different ROI sizes between 1.2 and 2 meters high. ROIs are created in image coordinates positioned from the left edge of the image to the right edge at equal intervals, ensuring sufficient overlap such that people in the scene won't fall in between ROIs. These values represent parameters that can be easily configured to the FROG scenario at hand; their setting involves a trade-off between computational effort on one hand, and detection and orientation estimation performance, on the other hand.

For each of the 46 depth levels, a binary image is created representing all pixels that have stereo support at that depth level. Integral images are used to determine for each ROI whether or not sufficient depth support is available within that ROI at that depth level to possibly contain a person. A ROI is accepted for further processing when at least 30% of its area is supported at that level.

After filtering the ROIs using the stereo information, the remaining ROIs are used for person classification and body orientation estimation. Person classification is based on Histogram of Oriented Gradients (HOG) features and a linear Support Vector Machine (SVM) classifier, as proposed by [11]. For efficiency reasons, all regions of interest are re-scaled such that the ratio between ROI width and height equals 1:2. The classifier is trained on a dataset containing 48 x 96 pixel images, split up into two different classes containing either persons or background clutter. This training dataset was kindly supplied by the authors of [13]. The HOG features of all ROIs of the same scale are computed efficiently making use of integral images. The image is scaled such that the ROI to be evaluated are 48 x 96 pixels and classified using the trained SVM. ROIs at multiple scales are processed in parallel for more efficient processing.

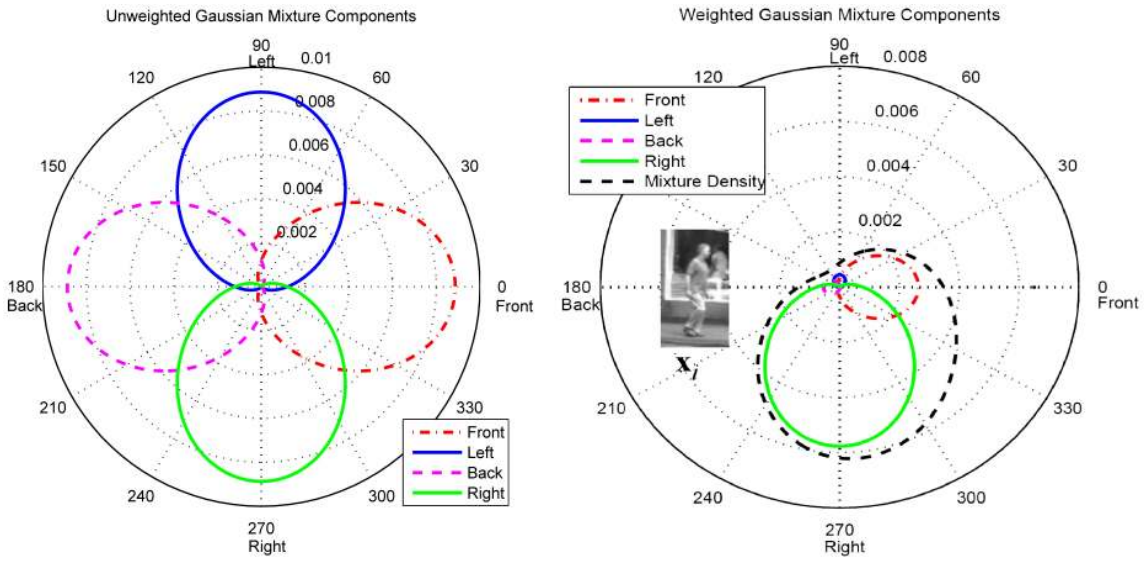


Figure 2.3: Polar plots of the Gaussian mixture model used for orientation estimation. Left: unweighted components, Right: weighted components, accumulated mixture density and classified image. This figure was taken from [13] with the authors' permission.

Retrieving the person body orientation is done simultaneously with person classification, according to the method described in [13]. Instead of training the SVM to only classify person or non-person ROIs, 4 experts are trained on 4 person orientations: front facing, back facing, left facing and right facing. In order to get the actual body orientation instead of just the classifier result, the posterior probabilities for each of the 4 orientations, resulting from the classification, are used to create a mixture of experts. This is done by using the posterior probabilities of each class as mixing weights in a Gaussian mixture model. The Gaussian components have means at 0^o, 90^o, 180^o and 270^o and all have a standard deviation of 45^o. The maximum likelihood orientation can now be determined by locating the maximum of the Gaussian mixture model. An example of the non-weighted and weighted mixture models, visualized in a polar plot, can be found in Figure 2.3.

The final step is tracking the person detections and the orientation estimates. Because the ROIs are positioned close to each other, having significant overlap, multiple ROIs remaining after stereo filtering will cover the same person. In order to reduce these ROIs to only 1 ROI per person, mean shift based non-maxima suppression will be applied as suggested in [10]. Based on the position, scale and person probability, ROIs in close proximity of each other are clustered using the mean shift algorithm and a new ROI representing this cluster is used as the detected person location. The maximum person likelihood over all ROIs in the cluster is used as the new ROI person likelihood.

Person positions are tracked in the 2D image plane using a constant velocity Kalman filter, based on the ROI scale and (x, y) position in the image. The person probability is

filtered over time using a low-pass filter. When the filtered person probability exceeds a value of 0.4, a new track is accepted. When no new detections are found near the tracker and the filtered person probability becomes less than 0.2 the track is deleted again. The assignment of tracks to detections is based on the Mahanalobis distance between the detections and the Kalman filter predictions of the track positions. The Hungarian algorithm [23] is used to find an optimal assignment between all tracks and detections.

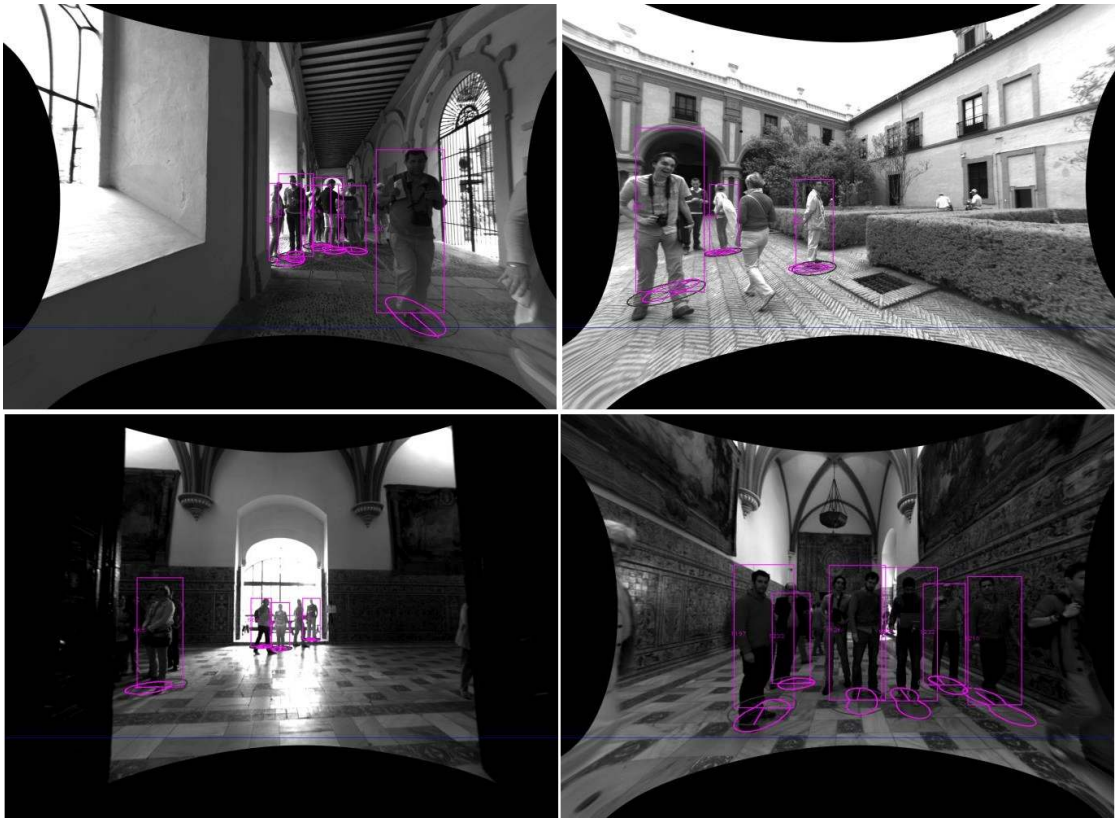


Figure 2.4: Tracking results on data recorded at the Real Alcázar test site in Sevilla. Ellipses below the pink ROI show the full orientation estimation pdf as a polar plot. The line inside each ellipse represents the current maximum likelihood orientation.

The person orientation is also filtered over time using a Kalman filter. However, instead of directly filtering the estimated orientation, the full orientation probability density function (pdf) is filtered. This is done by Kalman filtering the mixture weights of the 4 experts in the mixture model. This allows us to retain more information on the full pdf over time and make a more accurate orientation estimate. Figure 2.4 shows some examples of frames with tracking results on data recorded at the Real Alcázar test site in Sevilla.

Currently, the pedestrian detection rate is about 90% with about 1 false positive every 15-20 mins, at track level. The trade-off between detection rate and the number of false positives is controlled by underlying thresholds, and can be modified. The error in orientation estimation, and its increase with larger pedestrian distance, is analyzed in Figure 2.5.

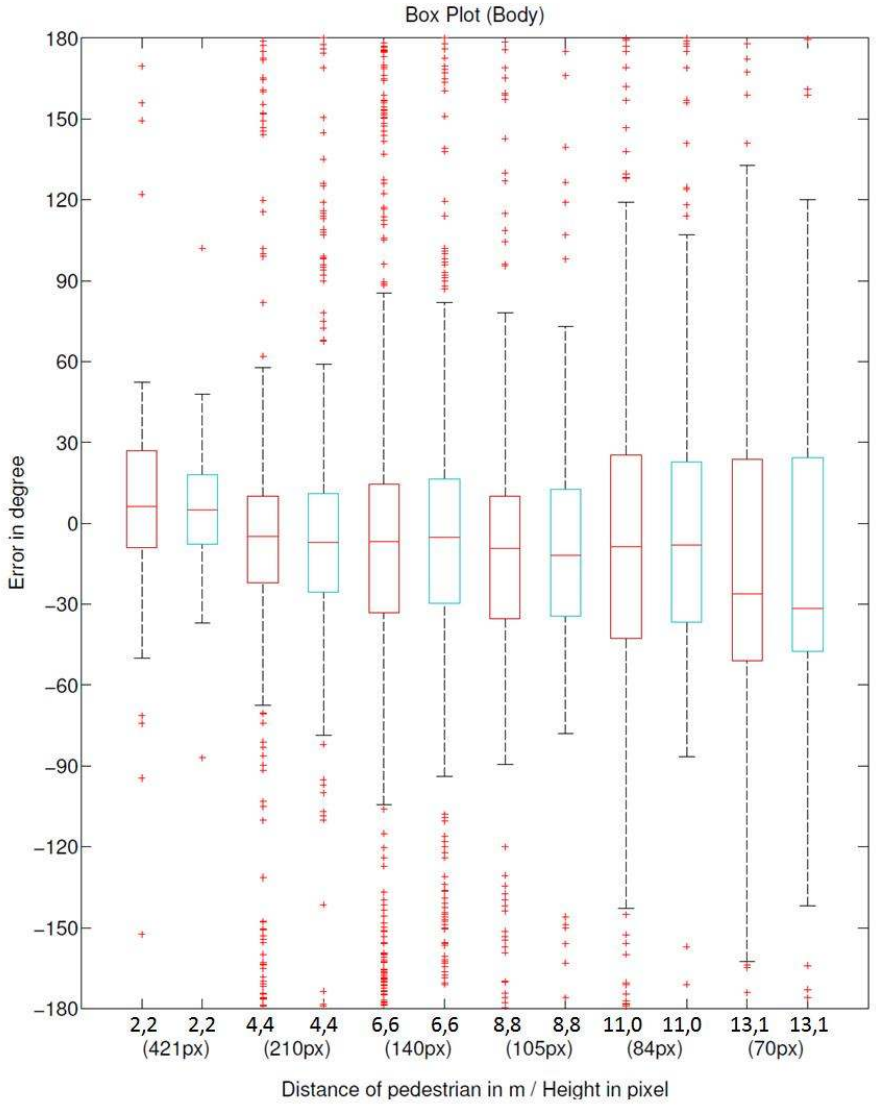


Figure 2.5: Orientation estimation error over distance (ground truth visually estimated from images). Boxplot shows median inside box containing 50% of data points, and whisker length containing 99.3% of data, if normally distributed. In red: per frame, in blue: after tracking.

In order to boost computational speed, the first two steps of the process have been parallelized. Besides parallelizing the person classification at different scales, as discussed earlier, the complete first and second step are run in parallel. While one thread

is classifying the stereo filtered ROI for person presence, another thread is already computing the stereo information for the next frame to be processed. This parallelization allows running the complete system at 10 Hz using a 3.7 Ghz quad core Intel Core i7 processor and 16 GB RAM. This speed is sufficient for real-time system operation on the FROG platform.

Finally, the person detection, tracking and orientation estimation system has been integrated into the FROG robot platform. Stereo information is shared with the localization and path planning module developed by UPO. Raw data of the JPEG compressed left stereo image and the stereo disparity map in sent over a standard TCP/IP connection. This information also includes the image timestamps and stereo calibration matrices. Detection data is coded using JSON and sent over a Websockets connection. This information contains a timestamp, the ROI found, their location on the ground plane with respect to the camera, their person detection probability and the maximum likelihood orientation estimate. To allow further orientation analysis, the full orientation pdf, discretized over 360^o, is sent as well.

Chapter 3

Visual face analysis module

In FROG, given that the focus lies on locating persons in groups in outdoor spaces and to perform facial expression analysis, the method for feature extraction should be suitable for handling dynamic backgrounds, significant amount of clutter/ occlusions and changing illumination conditions. For the face and the head, to be observed with a high-resolution camera, multi-person face detection, face tracking, and head pose estimation, regardless of head pose, clutter, and variations in lighting conditions need to be solved. The problem is difficult due the fact that not only the observed persons move and may occlude each other, but that the observation camera moves as well and may jitter due to the movements of the robot. Robust, fast and effective image registration should be developed. To enable feature-based facial gesture and gaze recognition, facial feature points tracking in the detected face region are attempted as well. This task aims further to develop a set of visual methods for detecting human affective states including users' positive and negative reactions to FROG robot and their overall level of interest and engagement in the current interaction with FROG.

The 'Visual face analysis' module takes as an input an image taken by the DALSA camera and provides as an output the 66 facial landmarks depicted in Figure 3.1, as well as the pitch, yaw and roll angles that define the face pose estimation. The module is based on a novel discriminative regression based approach for the Constrained Local Models (CLMs) framework, referred to as the Discriminative Response Map Fitting (DRMF) method, which shows impressive performance in the generic face fitting scenario. The motivation behind this approach is that, unlike the holistic texture based features used in the discriminative AAM approaches, the response map can be represented by a small set of parameters and these parameters can be very efficiently used for reconstructing unseen response maps. Furthermore, by adopting very simple off-the-shelf regression techniques, it is possible to learn robust functions from response maps to the shape parameters updates. The detailed experiments in a generic face fitting scenario on the databases with images captured under both the controlled (Multi-PIE and XM2VTS) and uncontrolled natural setting (LFPW Database), show that the DRMF method outperforms state-of-the-art algorithms for the task of generic face fitting. Moreover, the DRMF method is computationally very efficient and is real-time capable. The current

MATLAB implementation is a real time procedure. In the next sections, more details about the algorithm that it is used in this module, are provided.

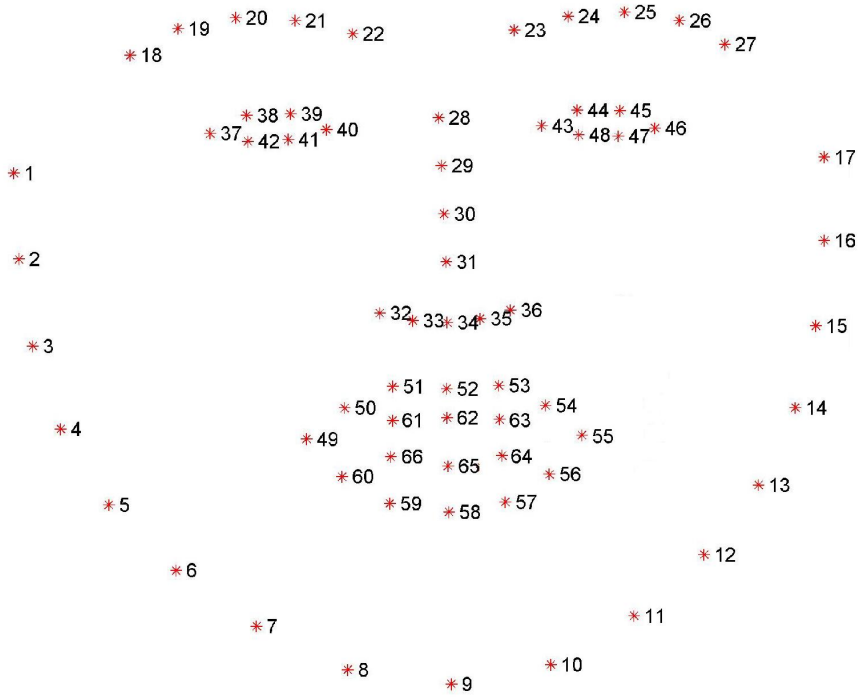


Figure 3.1: The 66 points mark-up provided by the module.

3.1 The Problem

The aim of a facial deformable model is to infer from an image the facial shape (2D or 3D, sparse [8, 4] or dense [6]), controlled by a set of parameters. Facial deformable models can be roughly divided into two main categories: (a) Holistic Models that use the holistic texture-based facial representations; and (b) Part Based Models that use the local images patches around the landmark points. Notable examples of the first category are AAMs [8, 4] and 3D deformable models [6]. While the second category includes models such as Active Shape Models (ASMs) [9], Constrained Local Models (CLMs) [27] and the tree-based pictorial structures [29].

3.1.1 Holistic Models

Holistic models employ a shape model, typically learned by annotating n fiducial points $\mathbf{x}_j = [x_j, y_j]^T{}_{j=1}^n$ and, then, concatenating them into a vector $\mathbf{s} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$. A statistical shape model $\mathcal{S}$ can be learned from a set of training points by applying PCA. Another common characteristic of holistic models is the motion model, which is defined using a warping function $\mathcal{W}(\mathbf{x}; \mathbf{s})$. The motion model defines how, given a shape, the image should be warped into a canonical reference frame (usually defined by the mean shape). This procedure is called shape-normalization and produces shape-free textures. Popular motion models include piece-wise affine and Thin-Plate Splines [4, 2].

The holistic models can be further divided according to the way the fitted strategy is designed. In *generative holistic models* [3, 19], a texture model is also defined besides the shape and motion models. The fitting is performed by an analysis-by-synthesis loop, where, based on the current parameters of the model, an image is rendered. The parameters are updated according to the residual difference between the test image and the rendered one. In probabilistic terms, these models attempt to update the required parameters by maximizing the probability of the test sample being constructed by the model. In *discriminative holistic models*, the parameters of the model are estimated by either maximizing the classification score of the warped test image, so that it belongs to the class of the shape-free textures [18], or by finding a set of functions that map the holistic texture features to the shape model parameters [24, 25].

Drawbacks of Holistic Models: **(1)** For the case of the generative holistic models, the task of defining a linear statistical model for the texture that explains the variations due to changes in identity, expressions, pose and illumination is not an easy task. **(2)** Similarly, due to the numerous variations of facial texture, it is not easy to perform regression from texture features to shape parameters (in a recent methodology whole shape regression is performed from randomly selected texture samples [7]). **(3)** Partial occlusions cannot be easily handled. **(4)** The incorporation of a 3D shape model is not easy due to the need of defining a warping function for the whole image; inclusion of a 3D shape model can be performed by sacrificing efficiency [1] (there is not an inverse compositional framework for the 3D case [20]) or by carefully incorporating extra terms in the cost function (which again is not a trivial task [20]).

3.1.2 Part Based Models

The main advantages of the part-based models are (1) partial occlusions can be easier to handled since we are interested only in facial parts, (2) the incorporation of a 3D facial shape is now straightforward since there is no warping image function to be estimated. In general, in part-based representations the model setup is $M = \{\mathcal{S}, \mathcal{D}\}$ where $\mathcal{D}$ is a set of detectors of the various facial parts (each part corresponds to a fiducial point of the shape model $\mathcal{S}$). There are many different ways to construct part-based models [27, 29], however we will focus only on ASMs and CLMs [27].

The 3D shape model of CLMs can be described as:

$$s(\mathbf{p}) = s\mathbf{R}(\mathbf{s}_0 + \mathbf{\Phi}_s\mathbf{q}) + \mathbf{t}, \tag{3.1}$$

where $\mathbf{R}$ (computed via pitch r_x, yaw r_y and roll r_z), s and $\mathbf{t} = [t_x; t_y; 0]$ control the rigid 3D rotation, scale and translations respectively, while $\mathbf{q}$ controls the non-rigid variations of the shape. Therefore the parameters of the shape model are $\mathbf{p} = [s, r_x, r_y, r_z, t_x, t_y, \mathbf{q}]$. Furthermore, $\mathcal{D}$ is a set of linear classifiers for detection of n parts of the face and is represented as $\mathcal{D} = \{\mathbf{w}_i, b_i\}_{i=1}^{n}$, where $\mathbf{w}_i, b_i$ is the linear detector for the i^{th} part of the face (e.g., eye-corner detector). These detectors are used to define probability maps for the i^{th} part and for a given location $\mathbf{x}$ of an image $\mathcal{I}$ being correctly located ($l_i = 1$) as:

$$p(l_i = 1 \mid \mathbf{x}, \mathcal{I}) = \frac{1}{1 + e^{\{l_i(w_i^T \mathbf{f}(\mathbf{x};\mathcal{I}) + b_i)\}}}. \tag{3.2}$$

where $\mathbf{f}(\mathbf{x};\mathcal{I})$ is the feature extracted from the patch in image $\mathcal{I}$ centered at $\mathbf{x}_i$. The probability of not being correctly spotted at $\mathbf{x}$ is simply $p(l_i = -1 \mid \mathbf{x}, \mathcal{I}) = 1 - p(l_i = 1 \mid \mathbf{x}, \mathcal{I})$.

In ASM and CLMs, the objective is to create a shape model from the parameters $\mathbf{p}$ such that the positions of the created model on the image correspond to well-aligned parts. In probabilistic terms, we want to find the shape $\mathbf{s}(\mathbf{p})$ by solving the following:

$$\begin{aligned} \mathbf{p} &= \arg\max p(\mathbf{s}(\mathbf{p}) \mid \{l_i = 1\}_{i=1}^n, \mathcal{I}) \\ &= \arg\max p(\mathbf{p}) \;\; p(\{l_i = 1\}_{i=1}^n \mid \mathbf{s}(\mathbf{p}), \mathcal{I}) \\ &= \arg\max p(\mathbf{p}) \prod_{i=1}^{n} p(l_i = 1 \mid \mathbf{x}_i(\mathbf{p}), \mathcal{I}). \end{aligned} \tag{3.3}$$

In [27], by assuming a homoscedastic isotropic Gaussian kernel density estimate in a set of fixed locations $\{\mathbf{\Psi}_i\}_{i=1}^n$ for every part i (i.e., $p(l_i = 1|\mathbf{x}_i(\mathbf{p}, I) = \prod_{i=1}^n \sum_{\mathbf{y}_i \in \mathbf{\Psi}_i} p(l_i = 1 \mid \mathbf{y}_i, I)\mathcal{N}(\mathbf{x}_i(\mathbf{p}) \mid \mathbf{y}_i, \rho\mathcal{I})$, the above optimization problem can be reformulated as:

$$\begin{aligned} \mathbf{p} = \arg\max p(\mathbf{p}) \prod_{i=1}^{n} \sum_{\mathbf{y}_i \in \mathbf{\Psi}_i} \\ p(l_i = 1 \mid \mathbf{y}_i, I)\mathcal{N}(\mathbf{x}_i(\mathbf{p}) \mid \mathbf{y}_i, \rho\mathcal{I}). \end{aligned} \tag{3.4}$$

For the case of the prior $p(\mathbf{p})$, which acts as a regularization term, the standard choice is a zero mean Gaussian prior over $\mathbf{q}$ (i.e., $p(\mathbf{p}) = \mathcal{N}(\mathbf{q} \mid \mathbf{0}, \mathbf{\Lambda})$). The above optimization problem was solved in [27] using an Expectation-Maximization (EM) algorithm. The Expectation step concerns the computation of $p(\mathbf{y}_i|l_i = 1, \mathbf{x}_i, I)$, given the parameters $\mathbf{p}$, while the Maximization step involves the minimization of:

$$Q(\mathbf{p}) = ||\mathbf{q}||_{\mathbf{\Lambda}}^{-1} + \sum_{i=1}^{n} \sum_{\mathbf{y}_i \in \mathbf{\Psi}_i} \frac{p(\mathbf{y}_i|l_i = 1, \mathbf{x}_i, I)}{\rho} ||\mathbf{x}_i(\mathbf{p}) - \mathbf{y}_i||^2$$

which can be solved using a Gauss-Newton optimization. This method is known as Regularized Landmark Mean-Shift (RLMS) [27] fitting. Even though it has been shown that the above optimization problem can produce state-of-the-art results it can also suffer from local minimum problem, as all Gauss-Newton optimization methodology.

3.2 Discriminative Response Map Fitting

The module follows a different direction to the RLMS approach for the part-based models discussed in the above Section 3.1.2. Instead of maximizing the probability of a reconstructed shape, given that all parts are correctly located in the image, (i.e., $p(\mathbf{s}(\mathbf{p}) \mid \{l_i = 1\}_{i=1}^n, \mathcal{I})$), we propose to follow a discriminative regression framework for estimating the model parameters $\mathbf{p}$. That is, we propose to find a mapping from the response estimate of shape perturbations to shape parameter updates. In particular, let

us assume that in the training set we introduce a perturbation $\Delta\mathbf{p}$ and around each point of the perturbed shape we have response estimates in a $w \times w$ window centered around the perturbed point, $\mathbf{A}_i(\Delta\mathbf{p}) = [p(l_i = 1 \mid \mathbf{x} + \mathbf{x}_i(\Delta\mathbf{p})]$. Then, from the response maps around the perturbed shape $\{\mathbf{A}_i(\Delta\mathbf{p})\}_{i=1}^n$ we want to learn a function f such that $f(\{\mathbf{A}_i(\Delta\mathbf{p})\}_{i=1}^n) = \Delta\mathbf{p}$. We call this the *Discriminative Response Map Fitting (DRMF)* method. The motivation behind this choice was the fact that, contrary to texture features in holistic regression based AAM frameworks [24, 25], response maps (1) can be very well represented by a small set of parameters and (2) learned dictionaries of probability response maps could very faithfully reconstruct response maps in unseen images.

Overall, the training procedure for the DRMF method has two main steps. In the first step, the goal is to train a dictionary for the response map approximation that can be used for extracting the relevant feature for learning the fitting update model. The second step involves iteratively learning the fitting update model which is achieved by a modified boosting procedure. The goal here is to learn a set of weak learners that model the obvious non-linear relationship between the joint low-dimensional projection of the response maps from all landmark points and the iterative 3D shape model parameters update ($\Delta\mathbf{p}$).

3.2.1 Training Response Patch Model

Before proceeding to the learning step, the goal is to build a dictionary of response maps that can be used for representing any instance of an unseen response map. In other words, our aim is to represent $\mathbf{A}_i(\Delta\mathbf{p})$ using a small number of parameters. Let us assume we have a training set of responses $\{\mathbf{A}_i(\Delta\mathbf{p}_j)\}_{j=1}$ for each point i with various perturbations (including no perturbation, as well). A simple way to learn the dictionary for the i-th point is to vectorize the training set of responses, stack them in a matrix $\mathbf{X}_i = [\text{vec}(\mathbf{A}_i(\Delta\mathbf{p}_1)), \ldots, \text{vec}(\mathbf{A}_i(\Delta\mathbf{p}_n))]$ and since we deal with non-negative responses, the natural choice is to perform Non-negative Matrix Factorization (NMF) [28]. That way the matrix is decomposed into $\mathbf{X}_i \approx \mathbf{Z}_i\mathbf{H}_i$ where $\mathbf{Z}_i$ is the dictionary and $\mathbf{H}_i$ are the weights. Now, given the dictionary $\mathbf{Z}_i$, the set of weights for a response map window $\mathbf{A}_i$ for the point i can be found by:

$$\mathbf{h}_i^o = \arg\max_{\mathbf{h}_i} ||\mathbf{Z}_i\mathbf{h}_i - \text{vec}(\mathbf{A}_i)||^2, \text{ s.t } \mathbf{h}_i \geq 0 \tag{3.5}$$

which can be solved using NMF strategies [28]. Then, instead of finding a regression function from the perturbed responses $\{\mathbf{A}_i(\Delta\mathbf{p})\}_{i=1}^n$, we aim at finding a function from the low-dimensional weight vectors $\{\mathbf{h}_i(\Delta\mathbf{p})\}_{i=1}^n$ to the update of parameters $\Delta\mathbf{p}$.

For practical reasons and to avoid solving the optimization problem (3.5) for each part in the fitting procedure, instead of NMF we have also applied PCA on $\{\mathbf{A}_i(\Delta\mathbf{p}_j)\}_{j=1}^N$. Using PCA, the extraction of the corresponding weigh vector $\mathbf{h}_i$ can be performed very efficiently by just a simple projections on the PCA bases. An illustrative example on how effectively a response map can be reconstructed by as small number of PCA components (capturing 85% of the variation) is shown in Figure 3.2. We refer to this dictionary

as *Response Patch Model* represented by:

$$\{\mathcal{M}, \mathcal{V}\} : \mathcal{M} = \{\mathbf{m}_i\}_{i=1}^{n} \text{ and } \mathcal{V} = \{\mathbf{V}_i\}_{i=1}^{n} \tag{3.6}$$

where, $\mathbf{m}_i$ and $\mathbf{V}_i$ are the mean vector and PCA bases, respectively, obtained for each of the n landmark points.

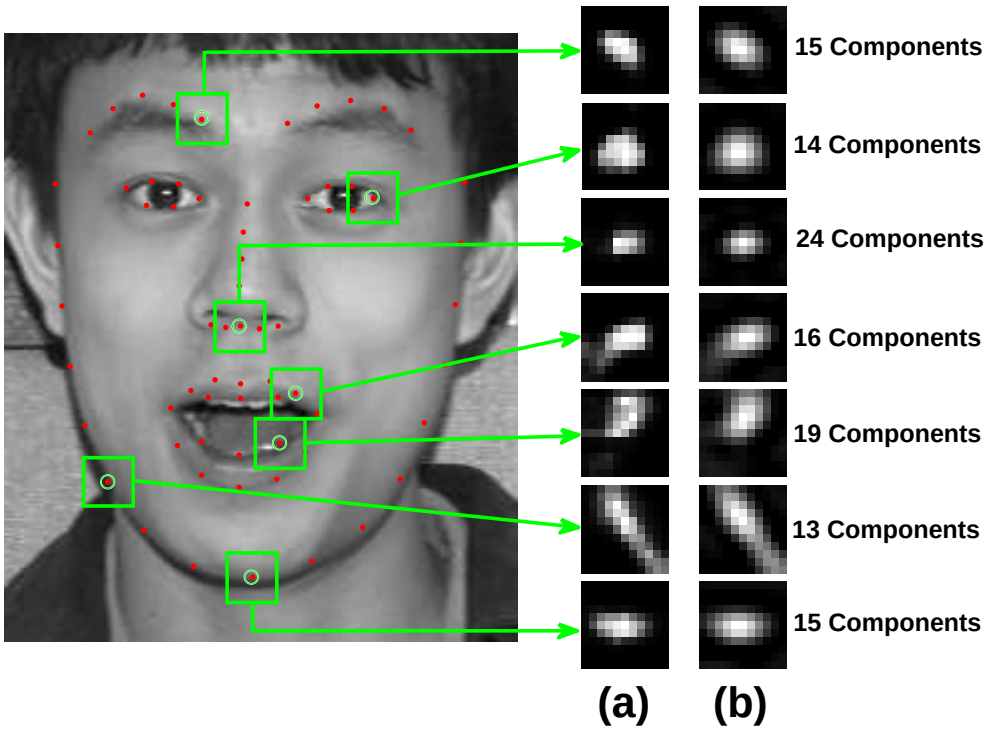


Figure 3.2: Overview of the response patch model: (a) Original HOG based response patches. (b) Reconstructed response patches using the response patch model that captured 85% variation.

3.2.2 Training Parameter Update Model

Given a set of N training images $\mathcal{I}$ and the corresponding shapes $\mathcal{S}$, the goal is to iteratively model the relationship between the joint low-dimensional projection of the response patches, obtained from the response patch model $\{\mathcal{M}\ ,\ \mathcal{V}\}$, and the parameters update ($\Delta\mathbf{p}$). For this, we propose to use a modified boosting procedure in that we uniformly sample the 3D shape model parameter space within a pre-defined range around the ground truth parameters $\mathbf{p}_g$ (See Eqn. 3.1), and iteratively model the relationship between the joint low-dimensional projection of the response patches at the current sampled shape (represented by t^{th} sampled shape parameter $\mathbf{p}_t$) and the parameter update $\Delta\mathbf{p}$ ($\Delta\mathbf{p} = \mathbf{p}_g - \mathbf{p}_t$). The step-by-step training procedure is as follow:

Let T be the number of shape parameters set sampled from the shapes in $\mathcal{S}$, such that the initial sampled shape parameter set is represented by $\mathcal{P}^{(1)}$:

$$\mathcal{P}^{(1)} = \{\mathbf{p}_j^{(1)}\}_{j=1}^{T} \quad \text{and} \quad \psi^{(1)} = \{\Delta\mathbf{p}_j^{(1)}\}_{j=1}^{T} \tag{3.7}$$

'1' in the superscript represents the initial set (first iteration). Next, extract the *response patches* for the shape represented by each of the sampled shape parameters in $\mathcal{P}^{(1)}$ and compute the low-dimensional projection using the response patch model $\{\mathcal{M}, \mathcal{V}\}$. Then, concatenate the projections to generate a joint low-dimensional projection vector $\mathbf{c}(\Delta\mathbf{p}_j^{(1)}) = [\mathbf{h}_1(\Delta\mathbf{p}_j^{(1)}), \ldots, \mathbf{h}_n(\Delta\mathbf{p}_j^{(1)})]^T$, one per sampled shape, such that:

$$\chi^{(1)} = \{\mathbf{c}(\Delta\mathbf{p}_j^{(1)})\}_{j=1}^{T} \tag{3.8}$$

where, $\chi^{(1)}$ represents the initial set of joint low-dimensional projections obtained from the training set. Now, with the training set $\mathcal{T}^{(1)} = \{\chi^{(1)}, \psi^{(1)}\}$, we learn the fitting parameter update function for the first iteration i.e. a weak learner $\mathcal{F}^{(1)}$:

$$\mathcal{F}^{(1)} : \psi^{(1)} \leftarrow \chi^{(1)} \tag{3.9}$$

We then propagate all the samples from $\mathcal{T}^{(1)}$ through $\mathcal{F}^{(1)}$ to generate $\mathcal{T}_{new}^{1}$ and eliminate the converged samples in $\mathcal{T}_{new}^{(1)}$ to generate $\mathcal{T}^{(2)}$ for the second iteration. Here, convergence means that the shape root mean square error (RMSE) between the predicted shape and the ground truth shape is less than a threshold (for example, set to 2 for the experiments). Any regression method can be employed in our framework. We have chosen a simple Linear Support Vector Regression (SVR) [16] for each of the shape parameters. In total, we used 16 shape parameters i.e. 6 global shape parameters and the top 10 non-rigid shape parameters. Structured regression based approaches can also be employed but we opted to show the power of our method with a very simple regression frameworks.

In order to replace the *eliminated* converged samples, we generate a new set of samples (Eqn. 3.7 and Eqn. 3.8) from the same images in $\mathcal{I}$ whose samples converged in the first iteration. We propagate this new sample set through $\mathcal{F}^1$ and eliminate the converged samples to generate an additional *replacement* training set for the second iteration $\mathcal{T}_{rep}^{(2)}$. The training set for the second iteration is updated:

$$\mathcal{T}^{(2)} \leftarrow \{\mathcal{T}^{(2)}, \mathcal{T}_{rep}^{(2)}\} \tag{3.10}$$

and the fitting parameter update function for the second iteration is learnt i.e. a weak learner $\mathcal{F}^{(2)}$. The sample elimination and replacement procedure for every iteration have two-fold benefits. Firstly, it plays an important role in insuring that the progressive fitting parameter update functions are trained on the tougher samples that have not converged in the previous iterations. And secondly, it helps in regularizing the learning procedure by correcting the samples that diverged in the previous iterations due to overfitting.

The above training procedure is repeated iteratively until all the training samples have converged or the maximum number of desired training iterations (η) have been reached. The resulting fitting parameter update model $\mathcal{U}$ is a set of weak learners:

$$\mathcal{U} = \{\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(\eta)}\} \tag{3.11}$$

The training procedure is outlined in Algorithm 1.

Algorithm 1: Training Parameter Update Model

Require: PDM (Eqn. 3.1), $\mathcal{I}$, $\mathcal{S}$, $\{\mathcal{M}, \mathcal{V}\}$ (Eqn. 3.6).

Get initial shape parameters sample set (Eqn. 3.7). ;
Get initial joint low-dimensional projection set (Eqn. 3.8). ;
Generate training set for first iteration $\mathcal{T}^{(1)}$. ;
for $i = 1 \rightarrow \eta$ **do**
    ;
    Compute the weak learner $\mathcal{F}^{(i)}$ using $\mathcal{T}^{(i)}$. ;
    Propagate $\mathcal{T}^{(i)}$ through $\mathcal{F}^{(i)}$ to generate $\mathcal{T}^{(i)}_{new}$. ;
    Eliminate converged samples in $\mathcal{T}^{(i)}_{new}$ to generate $\mathcal{T}^{(i+1)}$. ;
    if $\mathcal{T}^{(i+1)}$ *is empty* **then**
        All training samples converged. *Stop Training*. ;
    else
        Get new shape parameters sample set (Eqn. 3.7) from images whose samples are eliminated in Step 7. ;
        Get new joint low-dimensional projection set (Eqn. 3.8) for the samples generated in Step 11. ;
        Generate new *replacement* training set $\mathcal{T}^{(i)}_{rep}$. ;
        for $j = 1 \rightarrow (i-1)$ **do**
            ;
            Propagate $\mathcal{T}^{(i)}_{rep}$ through $\mathcal{F}^{(j)}$. ;
            Eliminate converged samples in $\mathcal{T}^{(i)}_{rep}$. ;
        Update $\mathcal{T}^{(i+1)} \leftarrow \{\mathcal{T}^{(i+1)}, \mathcal{T}^{(i)}_{rep}\}$;

Output : Fitting Parameter Update Model $\mathcal{U}$ (Eqn. 3.11).

3.2.3 Fitting Procedure

Given the test image $\mathcal{I}_{test}$, the fitting parameter update model $\mathcal{U}$ is used to compute the additive parameter update $\Delta\mathbf{p}$ iteratively. The *goodness of fitting* is judged by the fitting score that is computed for each iteration by simply adding the responses (i.e. the probability values) at the landmark locations estimated by the current shape estimate of that iteration. The final fitting shape is the shape with the highest fitting score.

3.3 Experiments

We conducted generic face fitting experiments on the Multi-PIE [15], XM2VTS [21] and the LFPW [5] databases. The Multi-PIE database is the most commonly used database for generic face fitting and is the best for comparison with previous approaches. Moreover, its consists of thousands of images with combined variations of identity, expression, illumination and pose, making it a very useful database for highlighting the ability of the proposed DRMF method (Section 3.2) to handle all these combined variations accurately in the generic face fitting scenario. The XM2VTS database focuses

mainly on the variations in identity and is a challenging database in a generic face fitting scenario because of the large variations in facial shape and appearance due to facial hair, glasses, ethnicity and other subtle variations. Unlike the Multi-PIE and the XM2VTS, the LFPW database is a completely *wild* database, i.e. consists of images captured under uncontrolled natural settings, and is an extremely challenging database for the generic face fitting experiment.

For all the experiments, we consider the independent model (p1050) of the tree-based method [29], released by the authors, as the baseline method for comparison. For the multi-view CLM approach, the pose range of $\pm 30^\circ$ in yaw (i.e. with pose code 051, 050, 140, 041 and 130) is divided into three view-based CLMs with each covering -30° to -15°, -15° to 15° and 15° to 30° in yaw, respectively. Other non-frontal poses have been excluded from our experiment for the lack of ground-truth annotations.

Another consistent aspect for all the following experiments is the initialization of the fitting procedure. For CLMs, we directly used the off-the-shelf OpenCV face detector. However, this face detector often fails on the LFPW dataset and for several images with varying illumination and pose in Multi-PIE and XM2VTS database. Therefore, for the images on which the face detector failed, we used the bounding box provided by our own trained tree-based model p204 (described in the following section) and perturbed this bounding box by 10 pixels for translation, 5° for rotation and 0.1 for scaling factor. We then initialized the mean face at the centre of this perturbed bounding box.

Overview of Results:

[1] The Multi-PIE experiment focuses on accessing the performance with combined identity, pose, expression and illumination variation. The results show significant performance gain for the proposed DRMF method over all other methods. Furthermore, the results show that the CLMs outperform the equivalent tree-based model for the task of landmark localization. We believe this is due to the use of tree-based shape model that allows for non-face like structures to occur making it hard to accurately fit the model, especially for the case of facial expressions.
[2] XM2VTS experiment, performed in an out-of-database scenario, highlights the ability of the DRMF method to handle unseen variations and other challenging variations like facial hair, glasses and ethnicity.
[3] LFPW experiment further verifies the generalization capability of the DRMF method to handle challenging uncontrolled natural variations. The results show that DRMF outperform RLMS and the tree-based method [29] convincingly on this wild database.
[4] The results on XM2VTS and LFPW database also validate one of the main motivations behind the DRMF method i.e. the response maps extracted from an unseen image can be very faithfully represented by a small set of parameters and are suited for the discriminative fitting frameworks, unlike the holistic texture based features.
[5] FROG experiments include the testing of the DRMF method using challenging videos that represent the realistic scenarios of the FROG project. The results show

that DRMF is suitable for handling dynamic backgrounds, significant amount of clutter/ occlusions and changing illumination conditions.

The fitting procedure of the DRMF method is highly efficient and is real-time capable. The current MATLAB implementation of the Multiview DRMF method, using the HOG feature based patch experts, takes 1 second per image on Intel Xeon 2.30 GHz processor. The compiled MATLAB source code and the pre-trained models are available for research purposes in *http://ibug.doc.ic.ac.uk/resources/frog-facial-tracking-component-code/*.

3.3.1 Multi-PIE Experiments

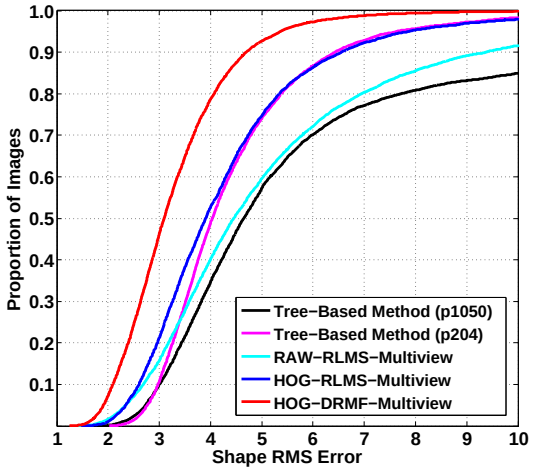


Figure 3.3: Experiment on Multi-PIE database.

The goal of the this experiment is to compare the performance of the HOG feature based CLM framework, using the RLMS [27] and the proposed DRMF (Section 3.2) method, with the tree-based method [29] under combined variations of identity, pose, expression and illumination. For this, images of all 346 subjects with all six expressions at frontal and non-frontal poses at various illumination conditions are used. The training set consisted of roughly 8300 images which included the subjects 001-170 at poses 051, 050, 140, 041 and 130 with all six expressions at frontal illumination and one other randomly selected illumination condition. For this experiment, we train several versions of the CLMs described below. The multi-view CLMs trained using the HOG feature based patch experts and the RLMS fitting method is referred as HOG-RLMS-Multiview. Whereas, the multi-view CLMs trained using the HOG feature based patch experts and the DRMF fitting method (Section 3.2) is referred as as HOG-DRMF-Multiview. Moreover, we also trained RAW-RLMS-Multiview which refers to the multi-view CLM using

the RAW pixel based patch experts and the RLMS fitting method. This helps in showing the performance gained by using the HOG feature based patch experts instead of the RAW pixel based patch experts.

For the tree-based method [29], we trained the tree-based model p204 that share the patch templates across the neighboring viewpoints and is equivalent to the multi-view CLM methods, using exactly the same training data for a fair comparison with CLM based approaches. We did not train the independent tree-based model (equivalent to p1050) because of its unreasonable training requirements, computational complexity and limited practical utility. Basically, training an independent tree-based model amounts to training separate models for each variation present in the dataset i.e. different models for every pose and expression. For our dataset that consists of five poses with all six expressions, an independent tree-based model will require training 2050 part detectors (i.e. 68 points $\times$ 5 poses $\times$ 6 expressions = 2050 independent parts). With preliminary calculations, such a model will require over a month of training time and nearly 90 seconds per image of fitting time.

The test set consisted of roughly 7100 images which included the subjects 171-346 at poses 051, 050, 140, 041 and 130 with all six expressions at frontal illumination and one other randomly selected illumination condition. From the results in Figure 3.3, we can clearly see that the HOG-DRMF-Multiview outperforms all other method by a substantial margin. We also see a substantial gain in the performance by using the HOG feature based patch experts (HOG-RLMS-Multiview) instead of the RAW pixel (RAW-RLMS-Multiview). Moreover, the HOG-RLMS-Multiview also outperform the equivalent tree-based model p204 for the task of landmark localization. The qualitative analysis of the results suggest that the tree-based methods [29], although suited for the task of face detection and rough pose estimation, are not well suited for the task of landmark localization. We believe, this is due to the use of tree-based shape model that allows for the non-face like structures to occur frequently, especially for the case of facial expressions. See the sample fitting results in Figure 3.6.

3.3.2 XM2VTS Experiments

All 2360 images from XM2VTS database [21] were manually annotated with the 68-point markup and are used as the test set. This experiment is performed in an out-of-database scenario i.e. the models used for fitting are trained entirely on the Multi-PIE database. We used the HOG-DRMF-Multiview, HOG-RLMS-Multiview and the tree-based model p204, used for generating results in Figure 3.3, to perform the fitting on the XM2VTS database. Note that this database consists of only frontal images. Nonetheless, the results from Figure 3.4 show that the HOG-DRMF-Multiview outperforms all other methods again. Moreover, the HOG-RLMS-Multiview outperforms the tree-based model p204 and the baseline p1050 convincingly.

This results is particularly important because it highlights the capability of the DRMF method to handle unseen variations. The generative model based discriminative approaches [18, 24, 25] have been reported to generalize well for the variations present on the training set, however, the overall performance of these discriminative fitting methods have been shown to deteriorate significantly for out-of-database experiments [26]. The results show that not only does DRMF outperform other state-of-the-art approaches in an out-of-database experiment but also handles the challenging variations in the facial shape and appearance present in the XM2VTS database due to facial hair, glasses and ethnicity. This result validates one of the main motivations behind the DRMF method i.e. the response maps extracted from an unseen image can be very faithfully represented by a small set of parameters and are suited for the discriminative fitting frameworks, unlike the holistic texture based features.

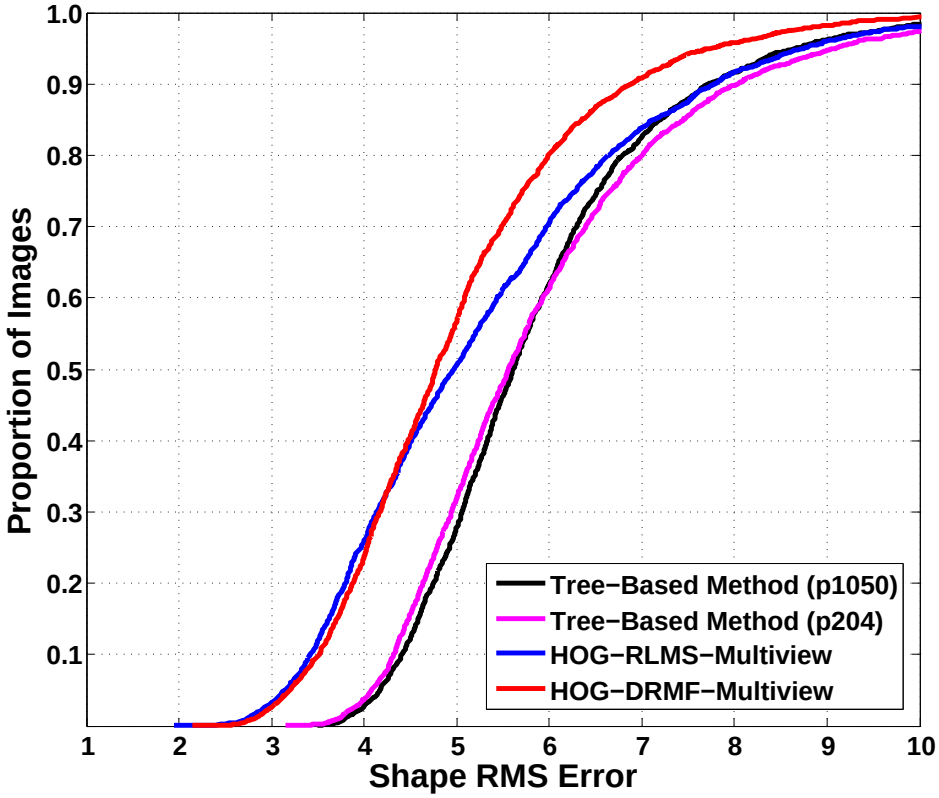


Figure 3.4: Out-of-database experiment on XM2VTS database.

3.3.3 LFPW Experiments

For further test the ability of the DRMF method to handle unseen variations, we conduct experiments using the database that presents the challenge of uncontrolled natural settings. The Labeled Face Parts in the Wild (LFPW) database [5] consist of the URLs to 1100 training and 300 test images that can be downloaded from internet. All of these images were captured *in the wild* and contain large variations in pose, illumination, ex-

pression and occlusion. We were able to download only 813 training images and 224 test images because some of the URLs are no longer valid. These images were manually annotated with the 68-point markup to generate the ground-truths used in this section.

We used the HOG-DRMF-Multiview, HOG-RLMS-Multiview and the tree-based model p204 trained only on the Multi-PIE database (used previously for generating results in Figure 3.3) to perform fitting on the LFPW test set. We then augmented the Multi-PIE training set with the LFPW training set and re-trained the CLM and tree-based models. We refer to these methods as HOG-Wild-DRMF-Multiview, HOG-Wild-RLMS-Multiview and the tree-based model p204-Wild. These wild models were then used to perform fitting on the LFPW test set and the results are reported in Figure 3.5. Note that the size of the faces in these images vary greatly because of the wild nature of this dataset. Therefore, we normalized the shape RMSE by the distance between the eye-corners which we believe is the best way to show unbiased results. From these results, we can clearly see the dominance of the HOG-Wild-DRMF-Multiview over other methods.

Firstly, this result clearly show that the proposed response map based discriminative fitting methodology can handle *wild face* and further emphasises the suitability of the parameterized response map models for the discriminative fitting frameworks. Secondly, an interesting result is the performance gain achieved by augmenting the Multi-PIE training set with the LFPW training set. Notice that in Figure 3.5, the accuracy of HOG-Wild-DRMF-Multiview increases consistently in comparison to the HOG-DRMF-Multiview (for example, by over 13% for the cases with Shape RMSE below 0.05 fraction of inter-ocular distance). Whereas for the same scenario, HOG-Wild-RLMS-Multiview show little improvement in performance over HOG-RLMS-Multiview (for example, increases by a little over 2% for the cases with Shape RMSE below 0.05 fraction of inter-ocular distance). This shows the advantage of the proposed response map based discriminative fitting approach that uses the available training data in a more useful way by learning the fitting update model as compared to the RLMS that rely entirely on the gauss-newton optimization based methodologies.

The results show that the proposed DRMF method outperforms the state-of-the-art RLMS fitting method [27] and the recently proposed tree-based method [29] consistently across all databases. See the sample fitting results in Figure 3.6.

3.3.4 FROG Experiments

DRMF was created in order to improve the performance in the generic face fitting scenario. This is a very challenging problem in outdoor spaces, while it is crucial for the FROG project to fulfill the goal of developing a set of visual methods for detecting human affective states including users' positive and negative reactions to FROG robot and their overall level of interest and engagement. In total, 15 videos were recorded

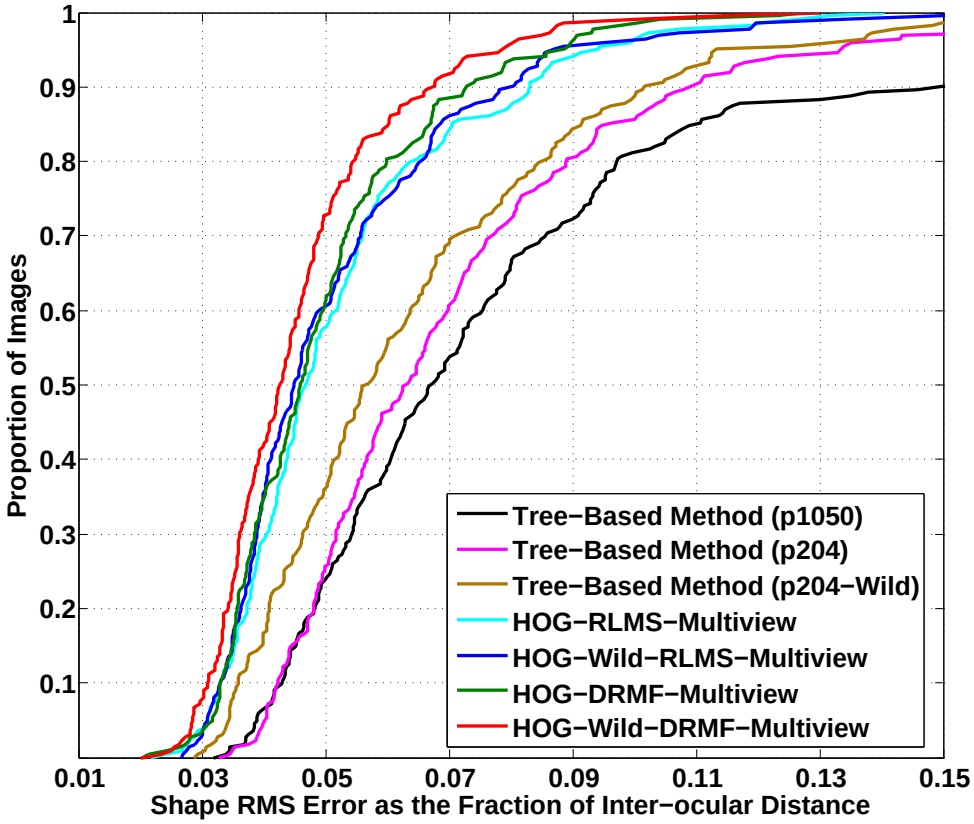
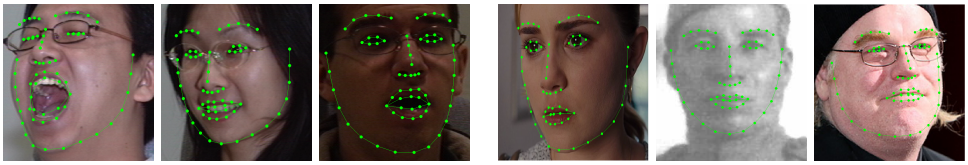
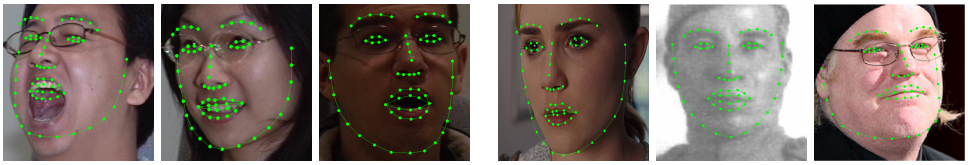


Figure 3.5: Wild experiments on LFPW database.

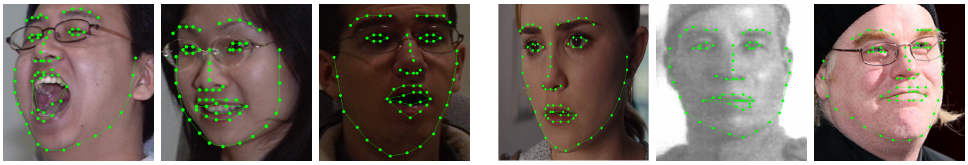
for these experiments using the DALSA camera. We used the HOG-DRMF-Multiview, HOG-RLMS-Multiview and the tree-based model p204 trained on both Multi-PIE and LFPW databases. In Figure 3.7 sample fitting results using the videos recorded for the FROG project, are depicted. Moreover, the DRMF method is computationally very efficient and real-time capable. The current C++/CUDA implementation is a real-time procedure with NVIDIA Quadro K1000M graphic card on an Intel Core^{TM} i5-3360M.



(a) DRMF Fitting Results



(b) RLMS Fitting Results



(c) Tree-based Model (p204) Fitting Results

Figure 3.6: Sample Fitting Results. Column 1-3: Multi-PIE Results. Column 4-7: LFPW Results.

Figure 3.7: Sample fitting results using videos recorded for the FROG project.

Chapter 4

Conclusion

A stereo vision system for pedestrian detection and orientation estimation was successfully developed, speed-optimized and integrated onto the FROG robot. Stand-alone evaluation suggests that perception performance is good; this remains to be confirmed in connection with the other FROG modules, especially with the module on navigation. Some further parameter optimization of the joint system will be beneficial.

A 'Visual face analysis' module was developed for FROG project. The function of this module is to detect facial landmarks. The module is based on a novel discriminative regression based approach for the Constrained Local Models (CLMs) framework, referred to as the Discriminative Response Map Fitting (DRMF) method. The method shows impressive performance in the generic face fitting scenario. It is suitable for handling dynamic backgrounds, significant amount of clutter/ occlusions and changing illumination conditions. This module also supports the further development of a set of visual methods for detecting human affective states including users' positive and negative reactions to FROG robot and their overall level of interest and engagement when interacting with the robot. These visual methods are to be developed for future deliverables in FROG project.

Bibliography

[1] T. Albrecht, M. Lüthi, and T. Vetter. A statistical deformation prior for non-rigid image and shape registration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[2] B. Amberg, A. Blake, and T. Vetter. On compositional image alignment, with an application to active appearance models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[3] S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 3. Technical report, RI, CMU, USA, 2003.

[4] S. Baker and I. Matthews. Equivalence and Efficiency of Image Alignment Algorithms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[5] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[6] V. Blanz and T. Vetter. Face Recognition Based on Fitting a 3D Morphable Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, Sept. 2003.

[7] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[8] T. Cootes, G. Edwards, and C. Taylor. Active Appearance Models. In *European Conference on Computer Vision (ECCV)*, 1998.

[9] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and applications. *CVIU*, 1995.

[10] N. Dalal. Finding People in Images and Videos. In *PhD Thesis, Institut National Polytechnique de Grenoble, INRIA Rhone-Alpes*, 2006.

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[12] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, 2009.

[13] M. Enzweiler and D. M. Gavrila. Integrated pedestrian classification and orientation estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 982–989, 2010.

[14] A. Geiger, M. Roser, and R. Urtasun. Efficient Large-Scale Stereo Matching. In *Computer Vision (ACCV), Springer Berlin Heidelberg*, pages 25–38, 2011.

[15] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. In *IEEE FG*, 2008.

[16] C. Ho and C. Lin. Large-scale linear support vector regression. Technical report, Technical report, NTU, 2012.

[17] C. G. Keller, M. Enzweiler, M. Rohrbach, D. F. Llorca, C. Schnorr, and D. M. Gavrila. The benefits of dense stereo for pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1096–1106, 2011.

[18] X. Liu. Discriminative face alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1941–1954, Nov. 2009.

[19] I. Matthews and S. Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60(2):135–164, Nov. 2004.

[20] I. Matthews, J. Xiao, and S. Baker. 2D vs. 3D Deformable Face Models: Representational Power, Construction, and Real-Time Fitting. *International Journal of Computer Vision*, 75(1):93–113, Oct. 2007.

[21] K. Messer, J. Matas, J. Kittler, J. LG'Ottin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *AVBPA*, 1999.

[22] T. B. Moeslund, A. Hilton, and V. KrG'Oger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006.

[23] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[24] J. Saragih and R. Goecke. Iterative Error Bound Minimisation for AAM Alignment. In *International Conference on Pattern Recognition (ICPR)*, 2006.

[25] J. Saragih and R. Goecke. A Nonlinear Discriminative Approach to AAM Fitting. In *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[26] J. Saragih and R. Goecke. Learning AAM fitting through simulation. *Pattern Recognition*, 42(11):2628–2636, Nov. 2009.

[27] J. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, Jan. 2011.

[28] D. Seung and L. Lee. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13:556–562, 2001.

[29] X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.