IDIAP RESEARCH REPORT

Active Shape Models Using Local Binary Patterns

Jean Keomany [a] Sébastien Marcel [b]

IDIAP–RR 06-07

February 2006

[a] Swiss Federal Institute of Technology (EPFL), Lausanne
[b] IDIAP Research Institute, Martigny

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 06-07

Active Shape Models Using Local Binary Patterns

Jean Keomany Sébastien Marcel

February 2006

Abstract. This report addresses the problem of locating facial features in images of frontal faces taken under different lighting conditions. The well-known Active Shape Model method proposed by Cootes *et al.* is extended in order to improve its robustness to illumination changes. For that purpose, we introduce the use of Local Binary Patterns (LBP). Three different approaches combining ASM with LBP are presented: profile-based LBP-ASM, square-based LBP-ASM and divided-square-based LBP-ASM. Experiments performed on the standard and darkened image sets of the XM2VTS database demonstrate that the divided-square-based LBP-ASM gives superior performance compared to the state-of-the-art ASM. It achieves more accurate results and fails less frequently.

Contents

List of Figures

Chapter 1

Introduction

Active Shape Model (ASM) is a popular statistical tool for locating examples of known objects in images. It was first introduced by Cootes *et al.* [5] in 1995 and has been developed and improved for many years. ASM is a model-based method which makes use of a prior model of what is expected in the image. Basically, the Active Shape Model is composed of a set of profile models and a deformable shape model. The shape model describes the typical variations of an object exhibited in a set of manually annotated images and the profile models give a statistical representation of the gray-level structures around each model point. Given a sufficiently accurate starting position, the ASM search attempts to find the best match of the shape model to the data in a new image using the profile models. ASM is thus fundamentally similar to Active Contour Model, or snake, proposed by Kass *et al.* [12]. However, ASM has global constraints that allow the shape model to deform only in ways found in the training set.

A direct extension of the ASM approach has lead to the Active Appearance Model [1]. Besides shape information, the textual information, i.e. the pixel intensities across the object, is included into the model. The AAM algorithm seeks to match both the position of the model points and a representation of the texture of the object to an image.

Although ASM and AAM can be used to find any object in an image, we focus this thesis on the detection of facial features such as eyes, nostrils, nose and mouth. Locating such features is an important stage in many facial image interpretation tasks such as face recognition, face tracking or face expression recognition. However, facial feature detection is a challenging task because human faces vary greatly between individuals. Faces can also appear at a wide range of sizes in images and facial hair or glasses can cause the facial features to be obscured. Although good results for facial feature localization using ASM and AAM have been reported [2, 13, 7], the ability of the model to perform well in different lighting conditions is still limited.

We propose in this thesis a novel approach combining ASM with Local Binary Patterns (LBPs). LBP is a powerful and computationally simple descriptor of local texture patterns. It expresses the difference of intensity between a given pixel and its

neighborhood. LBPs are therefore more robust to illumination changes.

This thesis is organized as follows: in the next chapter, we describe the original ASM method proposed by Cootes *et al.*. Chapter 3 introduces the different approaches combining ASM with LBPs that we have investigated during this work. Experiments and results are presented in Chapter 4 and a conclusion is drawn in Chapter 5.

Chapter 2

Active Shape Model

2.1 Building Models

Faces may vary from one image to another due to the identity of the individual, his facial expression, the lighting conditions and the 3D pose (both in plane and out of plane head rotation, scale variation, face location). In order to locate facial features in an image using Active Shape Models, we must first build a model that can describe shapes and typical variations of a face. To build a statistical model of a face, we require a set of training images reflecting possible variations.

In this section, we will present the steps needed to build a model from a set of training images. The first step is to annotate all the shapes in the training set, then align the labeled shapes, and finally capture the statistics of the variations.

2.1.1 Labeling the Training Set

The shape of a face is represented by a set of n landmark points or landmarks, which may be in any dimension d. Therefore, we must first decide upon a suitable set of landmarks which can be found reliably on every training image. The number of landmarks should be adequate to show the overall shape and also show details where it is needed. This number depends on the desired level of detail description.

Following Cootes *et al.* [4], good choices for landmarks in the two dimensional case, are points at corners of object boundaries, "T" junctions between boundaries or easily located biological landmarks such as the center of the eyes and the corners of the mouth. However, those points are usually not enough to provide a precise description of a human face. Therefore, we generally make use of intermediate points between well defined landmarks. Those points which are arranged to be equally spaced, describe most of the boundaries of the face. Figure 2.1 shows a face image from the XM2VTS database manually labeled with 68 landmark points.

Since the statistical model that will be used to describe faces is based on the variations of the coordinates of each landmark points within the training set, it is important to specify the landmarks positions as accurately as possible. The best method for gen-

erating a reliable training set is for a human expert to annotate each image with a set of corresponding points. In practice, this can be very time consuming, and automatic and semi-automatic methods are being developed to aid this annotation [4].

As a result, we end up with n landmark points in d dimensions for each shape. In order to have a mathematical representation, the coordinates of each point are concatenated to form a single vector of length $n \times d$. For instance, the n points of a planar shape ($d = 2$) can be represented by the vector $\mathbf{x}$:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)^T \tag{2.1}$$

where (x_j, y_j) are the coordinates of the j^{th} landmark. Given N training images, N such vectors $\mathbf{x_i}$ are then generated. Each vector is of length $2n$.

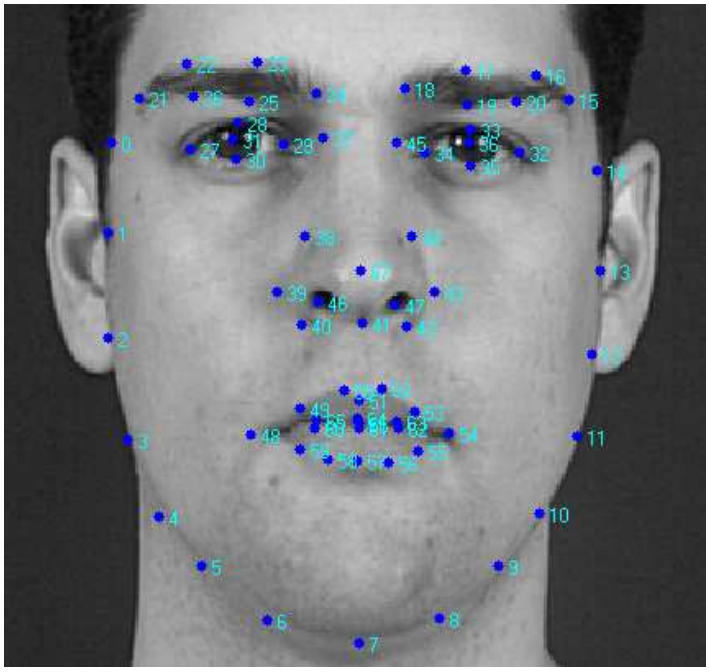


Figure 2.1: Example face image annotated with 68 landmarks

2.1.2 Aligning the Training Shapes

The shape of an object is normally considered to be independent of the scale, orientation and position of that object. Therefore, before any statistical analysis of the training shapes can be performed, variation due to scale, rotation and translation must first be removed from the shapes by aligning them into a common coordinate frame. This is achieved using Procrustes Analysis [9] which aligns each shape so that the squared distance to the mean ($\sum |\mathbf{x_i} - \bar{\mathbf{x}}|^2$) is minimized.

Although an analytical solution exists [8], the simple iterative approach proposed by Cootes *et al.* [4] is sufficient to align a set of shapes (see Algorithm 1).

Algorithm 1 Aligning the Training Shapes [4]

1. Translate each example so that its center of gravity is at the origin
2. Choose one example as an initial estimate of the mean shape (e.g. the first shape in the set) and scale so that $|\bar{\mathbf{x}}| = 1$
3. Record the first estimate as $\bar{\mathbf{x}}_\mathbf{0}$ to define the default reference frame
4. Align all shapes with the current estimate of the mean shape
5. Re-estimate mean from aligned shapes
6. Apply constraints on the current estimate of the mean by aligning it with $\bar{\mathbf{x}}_\mathbf{0}$ and scaling so that $|\bar{\mathbf{x}}| = 1$
7. If the mean $\bar{\mathbf{x}}$ has not changed significantly then STOP, else return to step 4

Step 1 of Algorithm 1 filters out variation between shapes due to variable face location in the training images. Hence, the transformations needed when aligning each shape to the mean in step 4, are only scaling and orientation. Suppose $T_{s,\theta}(\mathbf{x})$ scales the shape $\mathbf{x}$ by s and rotates it by θ. To align two 2D shapes, $\mathbf{x_1}$ and $\mathbf{x_2}$, each centered on the origin, we choose a scale s and rotation θ so as to minimize $|T_{s,\theta}(\mathbf{x_1}) - \mathbf{x_2}|^2$. The optimal solution is given in Appendix A. A weighting matrix can also be introduced in order to give more significance to the landmark points that tend to be more stable [5].

However, this approach introduces non-linearities. In order to improve the linearity of the aligned shape data, we project each shape $\mathbf{x_i}$ into the *tangent space* [8] of the mean shape $\bar{x}$ by scaling it by $1/(\mathbf{x_i} \cdot \bar{\mathbf{x}})$ after step 4. The tangent space of the mean shape is the hyperplane of vectors, normal to the mean shape, passing through it. In other words, it consists in all the vectors $\mathbf{x}$ such that $(\bar{\mathbf{x}} - \mathbf{x}) \cdot \bar{\mathbf{x}} = 0$.

The estimate of the mean shape is then computed in step 5 using:

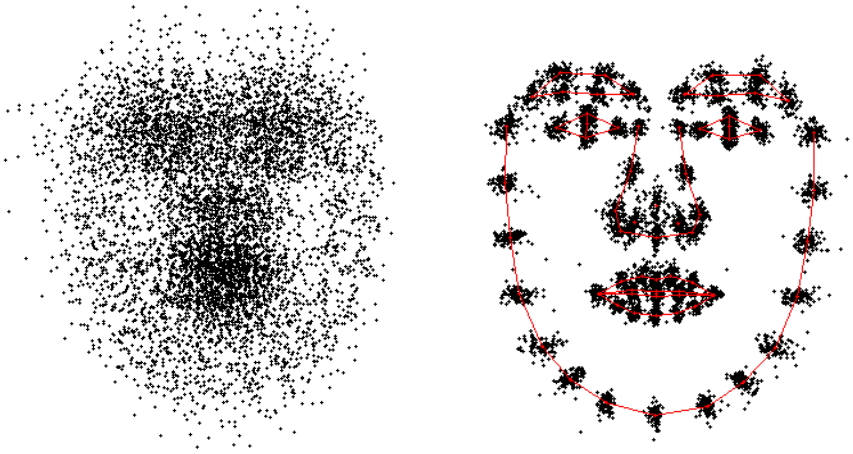
$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x_i} \tag{2.2}$$

where N denotes the number of training shapes. To avoid shrinking or drifting of the mean shape, size and orientation are properly fixed at each iteration by normalization (step 6).

In most cases, two iterations of the Algorithm 1 are sufficient to align all the shapes. Figure 2.2 illustrates a small training set before and after alignment.

2.1.3 Building a Shape Model

Each aligned shape can be considered as a single point in the nd dimensional space and the whole training set as a cloud of points in this space. In order to capture



(a) Unaligned shapes (b) Aligned shapes

Figure 2.2: Training set before and after alignment

the statistics of the shape variations, we apply Principal Components Analysis (PCA) to the data (see Algorithm 2). Basically, PCA computes the eigenvectors $\phi_{\mathbf{i}}$ of the covariance matrix $\mathbf{S}$ which correspond to the main axes of the cloud of points. Those eigenvectors define an orthogonal basis. Each axis gives a "mode of variation", a way in which the landmark points tend to move together as the shape varies. The eigenvectors $\phi_{\mathbf{i}}$ of the covariance matrix corresponding to the largest eigenvalues λ_i describe the most significant modes of variation.

Since the landmarks positions are always partially correlated, most of the variation exhibited in the training set can usually be explained by a small number of modes, t. Hence, the dimension of the model can be reduced. The proportion of the total variance explained by each eigenvector is equal to its corresponding eigenvalue. The number of eigenvectors to retain can then be chosen so that the model represents a certain percentage p (e.g. 98%) of the total variance given by the sum of all the eigenvalues λ_i:

$$V_T = \sum_{i=1}^{nd} \lambda_i \tag{2.3}$$

Therefore, t can be chosen as the smallest number such that,

$$\sum_{i=1}^{t} \lambda_i \geq p \cdot V_T \tag{2.4}$$

where the eigenvalues are sorted into descending order ($\lambda_i \geq \lambda_{i+1}$).

Algorithm 2 Principal Components Analysis

1. Compute the mean of the data using Equation 2.2
2. Compute the covariance of the data

$$\mathbf{S} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^T$$

3. Compute the eigenvectors ϕ_i and corresponding eigenvalues λ_i of $\mathbf{S}$
4. Sort the eigenvectors so that $\lambda_i \geq \lambda_{i+1}$
5. Store the first t eigenvectors as a matrix $\mathbf{\Phi} = (\phi_1|\phi_2|\ldots|\phi_t)$

PCA allows then each shape $\mathbf{x}$ in the training set to be approximated using the mean shape $\bar{x}$ and a small number of parameters $\mathbf{b}$:

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{\Phi b} \tag{2.5}$$

where $\mathbf{b}$ is a vector of dimension $t(< nd)$, obtained by projecting $\mathbf{x}$ into the subspace defined by the mean shape and the matrix $\mathbf{\Phi}$:

$$\mathbf{b} = \mathbf{\Phi}^T(\mathbf{x} - \bar{\mathbf{x}}) \tag{2.6}$$

Equation 2.5 allows us to generate new examples by varying the vector $\mathbf{b}$. The parameters b_i are assumed to be independent and Gaussian. The variance across the training set of an individual parameter b_i is given by λ_i. We can thus ensure that the shape generated is similar to those in the original training set by applying limits of $\pm 3\sqrt{\lambda_i}$ to the parameter b_i.

Figure 2.3 shows the effect of varying the first three shape parameters in turn between ± 3 standard deviations from the mean value, leaving all other parameters at zero.

2.2 Image Search

Having generated flexible models in the previous section, we would like to use them in image search, to find new examples of face in images. This involves finding the set of parameters which best match the model to the image. Using the model we described before, the parameters we can vary are the shape parameters $\mathbf{b}$ and the pose parameters (X_t, Y_t), θ and s, defining respectively the position, the orientation and the scale of the model in the image.

In this section, we describe an iterative method proposed by Cootes *et al.* [4] for finding the appropriate shape model given a very rough starting approximation. This

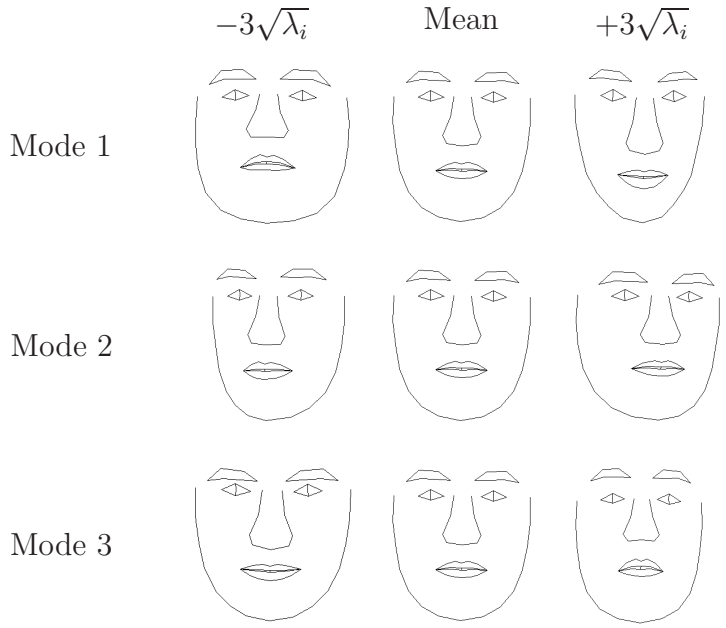


Figure 2.3: First three modes of the human face shape model. Images of mode i are generated using the shape parameter vector $\mathbf{b}$ where $b_j = 0$ for $j = 1, \ldots, t; j \neq i$ and $b_i = \{-3\sqrt{\lambda_i}, 0, +3\sqrt{\lambda_i}\}$.

approach moves each landmark to a position where the gray-level structure around the point is the most similar to that occurring at the given model point in the training images.

2.2.1 The Algorithm

Given an unlabeled face, the ASM search is required to match the shape model to the face image automatically. We assume in what follows that we know roughly the position in which the model should be placed.

An initial face model which is generally the mean shape model is first projected into the image being searched. Using the iterative approach explained by Algorithm 3, shape and pose parameters are altered such that the model moves and evolves in the image plane, hopefully converging to the best possible match of the model to the face image.

More specifically, the algorithm examines at each iteration a region of the image around each model point to determine a displacement which moves it to a better location. Although we could consider a region of any shape, we look in practice along profiles normal to the model boundary, passing through each model point (Figure 2.4). When the model boundaries correspond to edges, the points are just moved to the strongest edge along the profiles. However, this is not always the case. The model points may sometimes represent a weaker secondary edge or some other image structure,

Algorithm 3 Active Shape Model Algorithm [4]

1. Examine a region of the image around each point x_i to find the best nearby match for the point x_i'
2. Update the parameters $(X_t, Y_t, s, \theta, \mathbf{b})$ to best fit the new found points $\mathbf{x}'$
3. Apply constraints to the parameters $\mathbf{b}$ to ensure plausible shapes (e.g. limit so $|b_i| < 3\sqrt{\lambda_i}$)
4. Repeat until convergence

e.g. eye center, tip of nose. It is then necessary to have a more general model of the gray-level structures. The best approach according to Cootes is to learn from the training set what to look for in the target image. Since a given point corresponds to a particular part of the face, the gray-level patterns about that point in images of different examples will often be similar. A statistical model describing the gray-level structures around each landmark point in the training images can then be used to find the best movement.

Once a new position is found for each landmark point, the shape and pose parameters are adjusted so as to fit the model as close as possible to the suggested new points. By applying constraints to the shape parameters, we force the model to deform only in ways found in the training set.

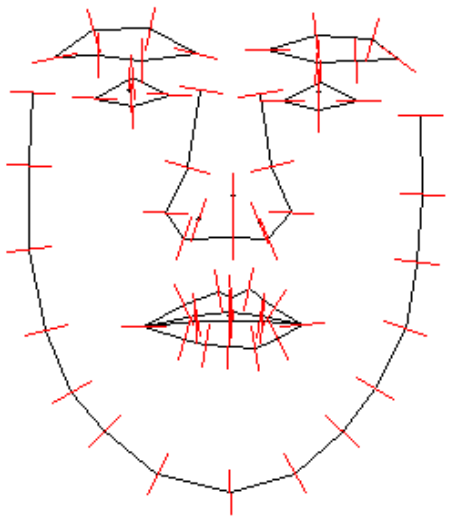


Figure 2.4: Profiles normal to the model boundary

Finding the Best Movement for each Point

For every landmark point i in the image j of the training set, we extract a profile $\mathbf{g_{ij}}$ of length n_p pixels, centered at the point. To reduce the effects of global intensity changes, we sample the derivative along the profile, rather than the absolute gray-level values (Cootes *et al.* [3]). The k^{th} element of the derivative profile is given by:

$$g_{ijk} = I_j(y_{i(k+1)}) - I_j(y_{i(k-1)}) \tag{2.7}$$

where y_{ik} is the k^{th} point along the i^{th} profile and $I_j(y_{ik})$ is the gray level in image j at that point.

This profile is then normalized by dividing each element by the sum of absolute element values:

$$\mathbf{g'_{ij}} = \frac{\mathbf{g_{ij}}}{\sum_{k=1}^{n_p} |g_{ijk}|} \tag{2.8}$$

For each landmark point i, we get a set of N normalized samples $\{\mathbf{g'_{ij}}\}$. Assuming that these are distributed as a multivariate Gaussian, we can compute the mean normalized derivative profile,

$$\mathbf{\bar{g}_i} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{g'_{ij}} \tag{2.9}$$

and the $n_p \times n_p$ covariance matrix,

$$\mathbf{S_{g_i}} = \frac{1}{N-1}\sum_{j=1}^{N}(\mathbf{g'_{ij}} - \mathbf{\bar{g}_i})(\mathbf{g'_{ij}} - \mathbf{\bar{g}_i})^T \tag{2.10}$$

This gives one gray-level model for each point.

Given a new profile $\mathbf{g_s}$, the quality of fit of that profile to its corresponding model $\mathbf{\bar{g}}$ can then be estimated using the following square error function which decreases as the fit becomes better:

$$f(\mathbf{g_s}) = (\mathbf{g_s} - \mathbf{\bar{g}})(\mathbf{g_s} - \mathbf{\bar{g}})^T \tag{2.11}$$

This fit function can also be weighted by the inverse of the covariance matrix $\mathbf{S_g}$ giving the Mahalanobis distance between the profile and the model:

$$f(\mathbf{g_s}) = (\mathbf{g_s} - \mathbf{\bar{g}})\mathbf{S_g}^{-1}(\mathbf{g_s} - \mathbf{\bar{g}})^T \tag{2.12}$$

In both cases, minimizing $f(\mathbf{g_s})$ is equivalent to maximizing the probability that $\mathbf{g_s}$ comes from the distribution.

During search, we extract for each landmark point, a search profile $\mathbf{g}$ from the current image, of some length l ($> n_p$) and centered at the point. We take the derivative and normalize it as we did in the training process. We then test the quality of fit of the corresponding gray-level model to the $l - n_p + 1$ possible sub-profiles (of length n_p) along the sample (Figure 2.5). The center of the sub-profile giving the best match indicates the new location of the landmark point.

In this way, we get a new position for each point. In order to ensure that the shape defined by the new points is similar to those in the training set, we then update the current pose and shape parameters to best match the shape model to the suggested new points.

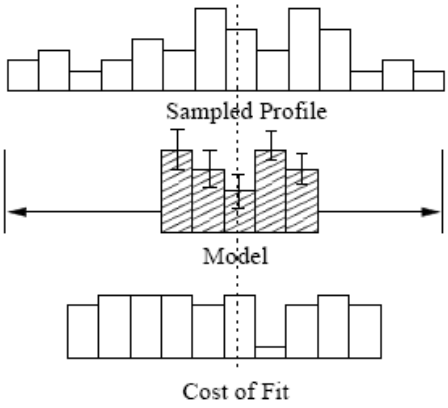


Figure 2.5: Search along sampled profile to find best fit of gray-level model [4]

Fitting the Model to New Points

The positions of the shape model points in an image are given by:

$$\mathbf{X} = T_{X_t,Y_t,s,\theta}(\bar{\mathbf{x}} + \mathbf{\Phi b}) \tag{2.13}$$

where the elements of vector $\mathbf{b}$ correspond to the shape parameters and $T_{X_t,Y_t,s,\theta}$ performs a rotation by θ, a scaling by s and a translation by (X_t, Y_t).

Given a new set of points $\mathbf{Y}$, the shape model can be fitted to the new shape by finding a suitable set of shape and pose parameters. The transformation $T_{X_t,Y_t,s,\theta}$ and $\mathbf{b}$ are chosen to minimize the sum of square error between the new set of points $\mathbf{Y}$ and the model instance $\mathbf{X}$:

$$|\mathbf{Y} - T_{X_t,Y_t,s,\theta}(\bar{\mathbf{x}} + \mathbf{\Phi b})|^2 \tag{2.14}$$

Cootes et al. [4] define an algorithm to iteratively minimize the approximation error (see Algorithm 4). In Algorithm 4, convergence is declared when applying an iteration produces no significant change in the pose and shape parameters. Convergence usually takes only a few iterations.

2.2.2 Multi-Resolution Active Shape Models

An important parameter that affects considerably the image search is the length of the search profile. On one hand, the search profile should be long enough to contain within it the target point, but on the other hand, it has to be short so as to avoid the landmark point moving to far away from the target and missing it. To solve this problem, Cootes *et al.* [6] proposed a multi-resolution approach where the search is first performed in a coarse image, then refined in a series of finer resolution images. This improves the efficiency, the robustness and the speed of the algorithm while making it less likely to get stuck on wrong image structures.

For each training and test image, a Gaussian image pyramid is built. The base image (level 0) is the original image. Each level is then formed by smoothing the

Algorithm 4 Iterative Fitting to New Points [4]

1. Initialize the shape parameters $\mathbf{b}$ to zero
2. Generate the model instance $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi b}$
3. Find the pose parameters (X_t, Y_t, s, θ) which best map $\mathbf{x}$ to $\mathbf{Y}$ (see Appendix A)
4. Invert the pose parameters and use to project $\mathbf{Y}$ into the model co-ordinate frame:
$$\mathbf{y} = T^{-1}_{X_t,Y_t,s,\theta}(\mathbf{Y})$$
5. Project $\mathbf{y}$ into the tangent plane to $\bar{\mathbf{x}}$ by scaling by $1/(\mathbf{y} \cdot \bar{\mathbf{x}})$
6. Update the model parameters to match to $\mathbf{y}$
$$\mathbf{b} = \mathbf{\Phi}^T(\mathbf{y} - \bar{\mathbf{x}})$$
7. Apply constraints on $\mathbf{b}$ (see Section 2.1.3)
8. If not converged, return to step 2

image at the level below with a 5×5 Gaussian mask (see Figure 2.6) and then subsampling every other pixel to obtain an image with half the number of pixels in each dimension. Figure 2.7 shows a three level pyramid.

0.125	0.625	1	0.625	0.125
0.625	3.125	5	3.125	0.625
1	5	8	5	1
0.625	3.125	5	3.125	0.625
0.125	0.625	1	0.625	0.125

Figure 2.6: Gaussian mask

During training, a statistical model of the gray-level structures around each landmark point is built at each level of the pyramid using the method described in Section 2.2.1. We usually use the same number of pixels in each profile model, regardless of the level.

During search, we only need to search a few pixels either side of the current point position at each level. This allows quite large movements at coarse levels whereas the shape model is just slightly modified in the finer resolution.

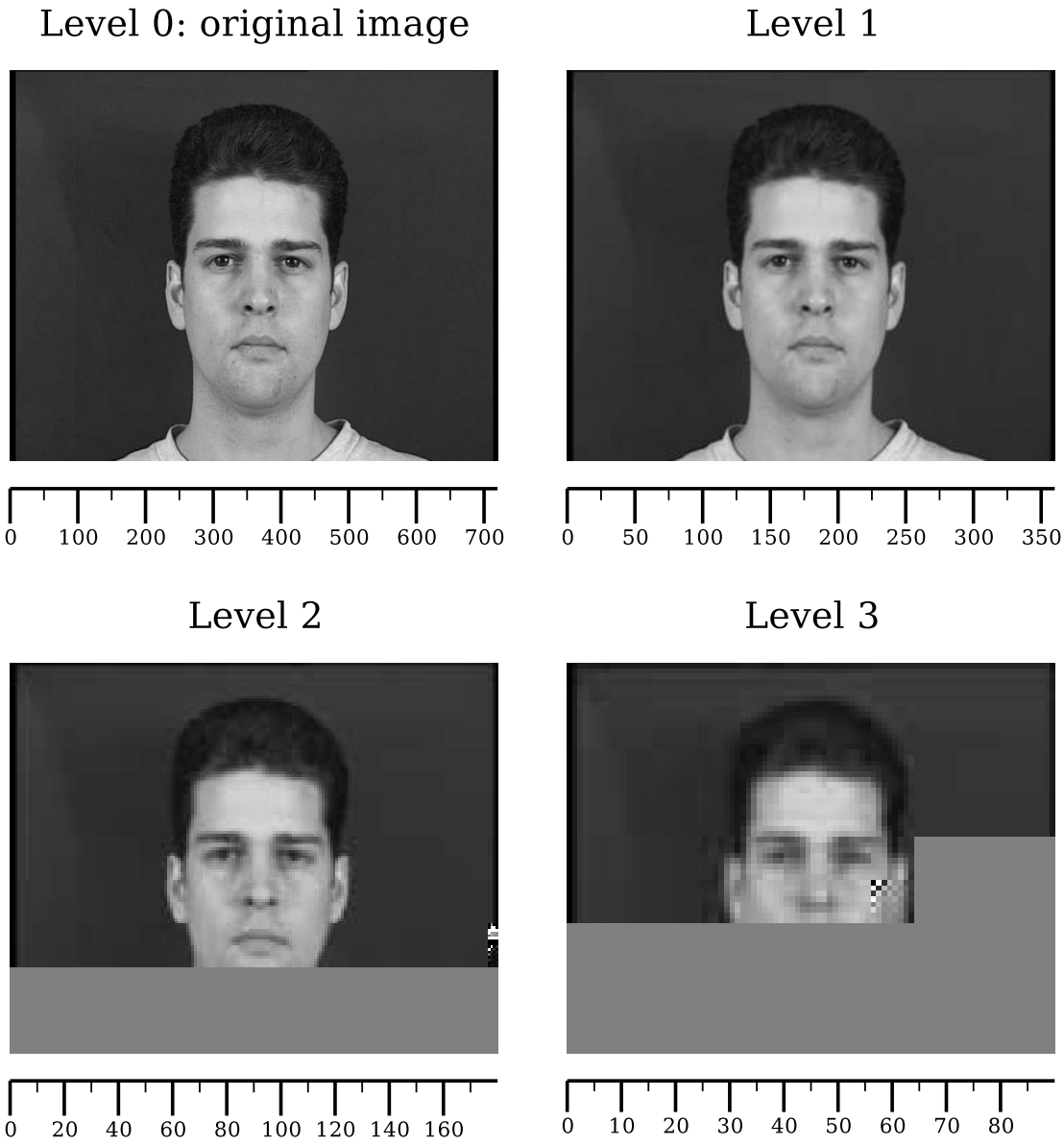


Figure 2.7: Pyramid of images

The search is performed by first searching at the top level of the pyramid. When the position of a certain percentage of landmark points does not change significantly, the algorithm is declared to have converged at that resolution. For instance, when 95% of the new points are within the central 50% of the search profile, the current shape model is projected into the next image and run to convergence again. The search is stopped when convergence is reached on the lowest level of the pyramid. Algorithm 5 summarizes the full multi-resolution ASM search algorithm.

Algorithm 5 Multi-Resolution ASM search algorithm [4]

1. Set $L = L_{max}$
2. While $L \geq 0$
   (a) Compute model point positions in image at level L
   (b) Search at n_s points on profile either side each current point
   (c) Update pose and shape parameters to fit model to new points
   (d) Return to (2a) unless more than 95% of the points are found within $n_s/2$ pixels of the current point, or N_{max} iterations have been applied at this resolution
   (e) If $L > 0$ then $L \rightarrow (L - 1)$
3. Final result is given by the parameters after convergence at level 0

2.2.3 Examples of search

Figure 2.8 illustrates two examples of face feature localization using Active Shape Model. The shape model is trained with 600 images annotated with 68 landmark points. 58 modes of variation are retained and the normal profiles used to build the local structure models are 25 pixels long.

Face detection is first performed. Then the shape model is initialized according to the estimated eye positions output by the face detector. The search starts at level 3 of the pyramid and moves to the next level when 95% of the new points are within the central 50% of the search profile or 20 iterations have been already performed.

Figure 2.8 shows that large movements are made in the first few iterations. In finer resolutions, adjustments are made and the shape model converges to a good match.

Figure 2.8: Examples of search using Active Shape Model of a face

Chapter 3

Active Shape Model using Local Binary Patterns

Active Shape Model is a powerful tool for face alignment. However, the features used to model the local gray-level structures are very sensitive to illumination, particularly when the lighting conditions during search are significantly different from the lighting conditions used to train the shape model.

In this master thesis, we introduce Local Binary Patterns (LBP) as novel features for local appearance representations. LBPs are powerful texture descriptors which are much more robust to illumination changes. So far, only Huang *et al.* [10] proposed an improved ASM method based on this idea but they used extended local binary patterns which encode not only the original image but also the gradient magnitude image.

In this chapter, we first introduce the LBP operator and present the approach proposed by Huang *et al.* We then describe the different methods for modeling the local structures using LBPs that we have investigated during the work.

3.1 Local Binary Patterns

The LBP operator, first introduced by Ojala *et al.* [15], is a powerful method of analyzing textures. The operator labels the pixels of an image by thresholding the 3×3 neighborhood of each pixel with the center value and considering the result as a binary number (see Figure 3.1). At a given pixel position (x_c, y_c), the decimal form of the resulting 8-bit word can be expressed as follows:

$$LBP(x_c, y_c) = \sum_{n=0}^{7} s(i_n - i_c)2^n \tag{3.1}$$

where i_c corresponds to the gray value of the center pixel (x_c, y_c), i_n to the gray values of the 8 surrounding pixels and function $s(x)$ is defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{3.2}$$

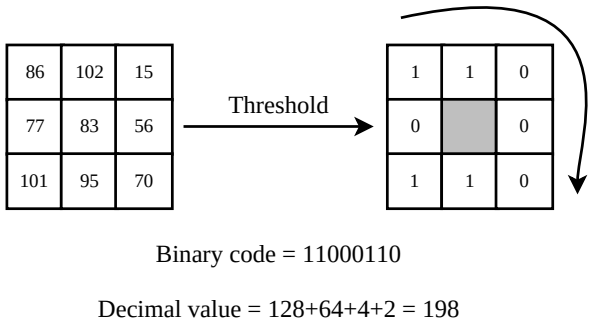


Figure 3.1: Calculating the original LBP code

The operator is therefore invariant to monotonic changes in gray scale and can resist illumination variations as long as the absolute gray value differences are not badly affected. However, the limitation of the original LBP operator comes from its small 3×3 neighborhood which can not capture features with large scale structures. Hence, Ojala *et al.* [16] extended their original LBP operator to a circular neighborhood of different radius size. Figure 3.2 illustrates examples of extended LBP operators where (P, R) refers to P equally spaced pixels on a circle of radius R. The value of neighbors that do not fall exactly on pixels, are estimated by bilinear interpolation.

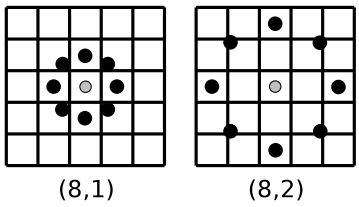


Figure 3.2: Examples of extended LBP operators

Further extension of LBP is to use uniform patterns [16]. A Local Binary Pattern is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. For instance, 00000000, 11111001 or 00011111 are uniform patterns. It has been observed that uniform patterns contain most of the texture information. They mainly represent primitive micro-patterns such as spots, lines, edges, corners.

The notation $LBP^{u2}_{P,R}$ denotes the extended LBP operator in a (P, R) neighborhood. The superscript ^{u2} indicates that only uniform patterns are used, labeling all remaining patterns with a single label.

Since each bit of the LBP resulting code has the same significance level, two successive bit values may have a totally different meaning. That is the reason why histograms of the labels are used to describe textures.

3.2 Extended Local Binary Patterns ASM

Recently, Huang *et al.* proposed in [10] an ASM method framework, ELBP-ASM, in which local appearance patterns of facial landmark points are modeled using *extended* local binary pattern.

Huang *et al.* pointed out in their paper that LBP can only reflect the first derivation information of images, but could not present the velocity of local variation. To solve this problem, they proposed an Extended version of Local Binary Patterns (ELBP) that encodes the gradient magnitude image in addition to the original image. Moreover, in order to retain spatial information, sub-images of landmark points are divided into small regions from which the LBP histograms are extracted and concatenated into a single feature histogram representing the local appearance patterns. Algorithm 6 describes the method for building the ELBP histogram associated to a given landmark point.

The mean ELBP histogram of each landmark can then be computed using:

$$\bar{H}_{i,j} = \frac{1}{N} \sum_{n} H_{n,i,j} \tag{3.3}$$

where N is the number of training images.

Algorithm 6 Building an ELBP histogram [10]

1. Extract from the original image a disk of radius 15 pixels, centered at the landmark point
2. Apply a low-pass Gaussian filter to the sub-image in order to reduce noise impact
3. Generate the gradient magnitude image using Sobel filter operators, h_x and h_y:

$$|\nabla I| = \sqrt{(h_x \otimes I)^2 + (h_y \otimes I)^2}$$

4. Divide the original image and the gradient magnitude image into four regions
5. Build five histograms corresponding to the whole image and four regions using (see Figure 3.3):

$$H_{i,j} = \sum_{im,fl} \sum_{x,y} I\{f_l(im(x,y)) = i\} I\{(x,y) \in R_j\}$$

   where $im \in \{$*original image, magnitude image*$\}$, $f_l \in \{LBP^{u2}_{8,1}, LBP^{u2}_{8,2}, LBP^{u2}_{8,3}\}$, $R_j \in \{$*region1, region2, region3, region4, whole image*$\}$ and I is the indicator function

6. Concatenate the histograms to get the ELBP histogram

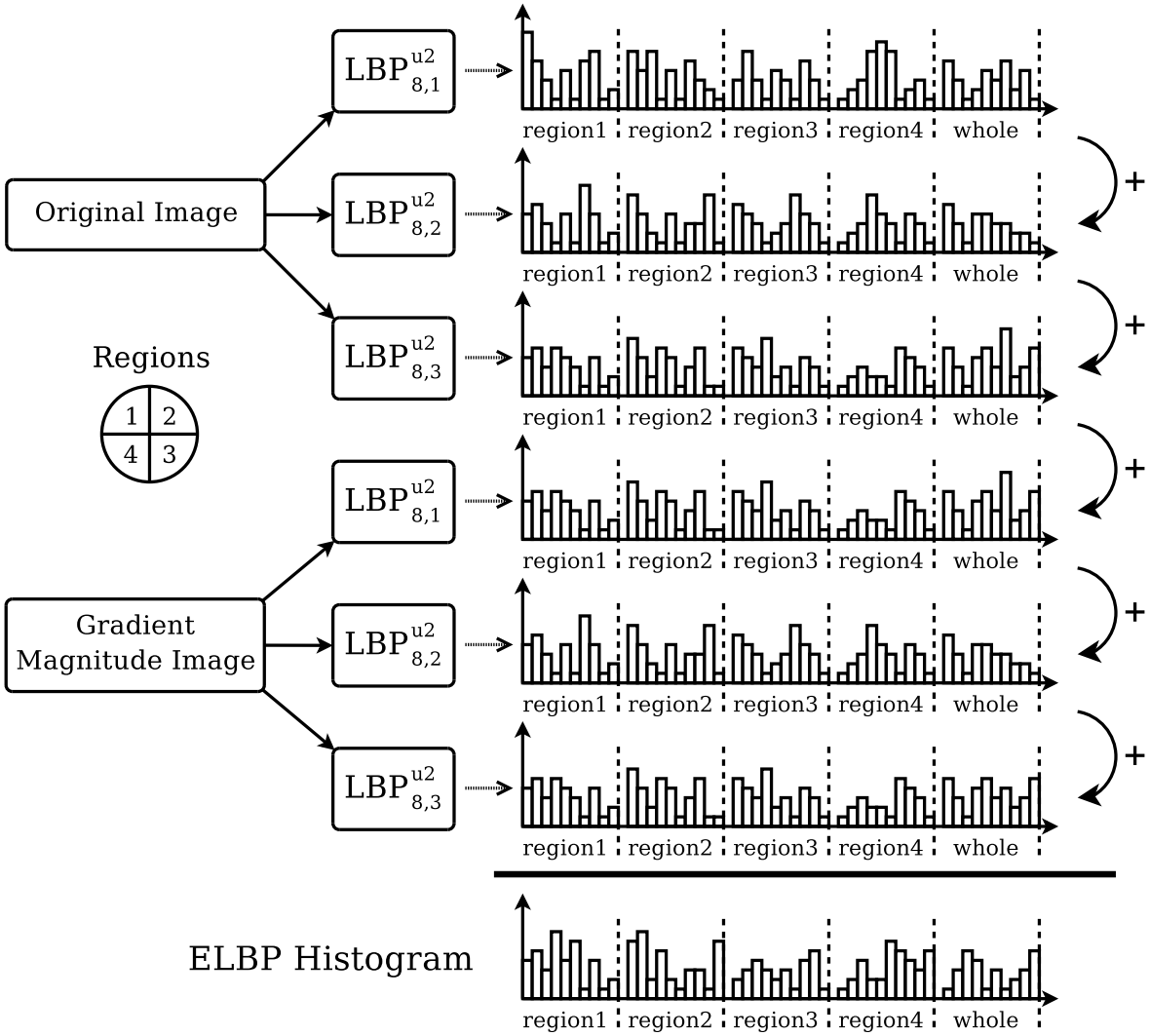


Figure 3.3: Building an ELBP histogram

During search, the ELBP histogram corresponding to each point located on the normal profile (see Section 2.2.1), is built using Algorithm 6. They are then compared to the mean histogram. The similarity between the testing point's histogram H and the mean histogram $\bar{H}$ is calculated using Chi square statistic:

$$\chi^2(\mathbf{H}, \bar{\mathbf{H}}) = \sum_i \frac{(H_i - \bar{H}_i)^2}{(H_i + \bar{H}_i)} \tag{3.4}$$

The smaller the distance is, the more similar the histograms are. The landmark point is thus moved to the profile point whose ELBP histogram is the closest to the mean histogram. Similarly to the original ASM, the pose and shape parameters of the shape model are then adjusted to fit the new suggested points, before starting a new iteration.

Huang *et al.* reported that ELBP-ASM achieves more accurate results than the original ASM. However, summing up the original image and the gradient magnitude image histograms might not be the most efficient way to take advantage of all the information available. Indeed, the features specific to each image histogram are lost when they are summed together. Moreover, using multi-scale LBP allows to capture the gray-level structures at different scales but it also adds computational load. Consequently, we believe that even better results can be achieved using simpler methods.

3.3 Proposed Approaches

During this work, we investigated new methods for modeling the local structures using LBPs. The following subsections describe the different approaches.

3.3.1 Profile-based LBP-ASM

We first propose to use a local appearance descriptor based on the LBP values extracted from the normal profile of each landmark point. In this method, $LBP^{u2}_{8,2}$ operator is used.

During training, we extract a profile of length n for every point of every training image and build the associated histogram of LBP values. We then compute the mean histogram of each landmark point.

During search, we extract for each landmark point, a search profile which is longer than the training profile. For each sub-profile of length n contained in the search profile, we build a histogram. The obtained histograms are compared to the corresponding mean histogram using the Chi square dissimilarity measure given by equation 3.2 (Figure 3.4). The landmark point is then moved to the center of the sub-profile which produces the most similar LBP histogram.

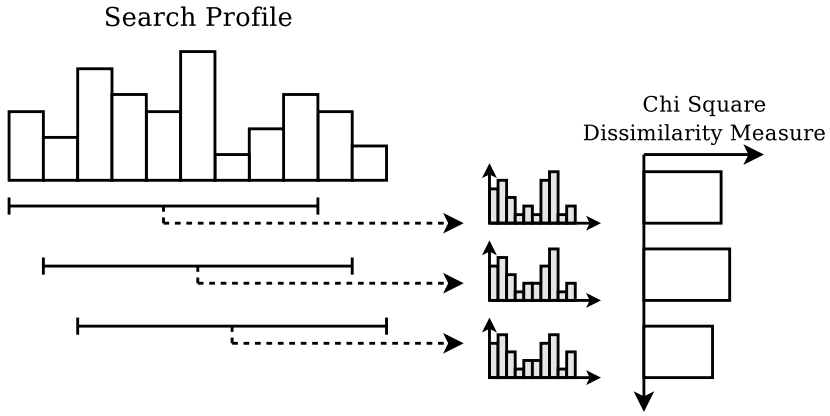


Figure 3.4: Search using histograms extracted from a profile

This approach is very simple but limited. First, the training profile has to be long enough to provide a sufficient number of points in order to build a reliable histogram.

Since the $LBP^{u2}_{8,2}$ operator produces 59 different labels, the profile has to be at least 59 pixels long to fill the histogram with in average one pixel per bin. However, this condition can hardly be satisfied. Second, comparing the histograms of two consecutive points along the profile does not make any sense since only one point has been replaced from one histogram to the other. These histograms can be considered to be almost identical. To cope with these problems, we propose to build the histogram with the points contained in a square centered at the landmark point.

3.3.2 Square-based LBP-ASM

The local appearance patterns are complex and it is hard to model them well only using simple profiles. In order to acquire more information related to the local gray-level structures, we use the points which are located within a square centered at a given landmark point to build the LBP histogram.

Basically, the training part is very similar to the one described before but sampling the points in a square region instead of a profile. During search, a LBP histogram is computed in the same manner for each point located on the search profile (Figure 3.5). The length of the search profile depends in this case only on the distance we allow the landmark point to move at each iteration (a few points). The similarity between the testing point's histograms and the mean histogram is also measured using the Chi square distance.

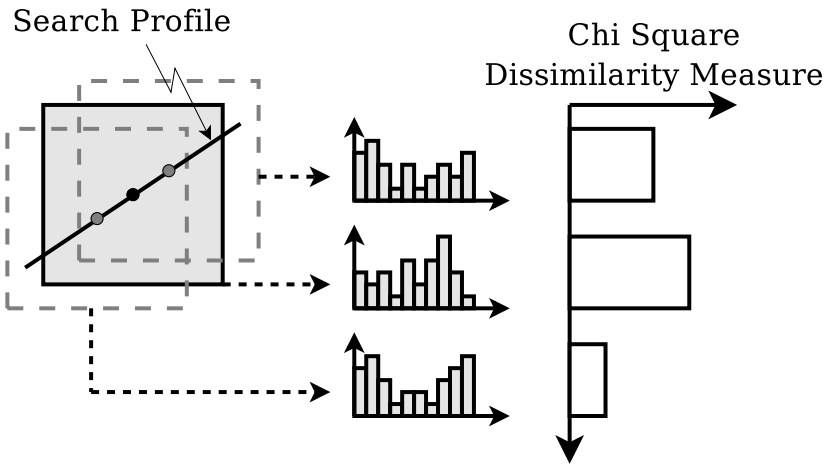


Figure 3.5: Search using histograms extracted from a square

Hence, this method allows us to model larger structures and fill the histograms with much more LBP values. However, this approach still suffers from the lack of spatial information. Indeed, the main pattern we want to detect could be anywhere in the square, the resulting histogram will always look similar. In order to retain spatial information, we divide the square into small regions as Huang *et al.* did in their algorithm.

3.3.3 Divided-Square-based LBP-ASM

The square used in the previous method is divided into four regions from which the LBP histograms are extracted and concatenated into a single feature histogram representing the local appearance patterns (Figure 3.6).

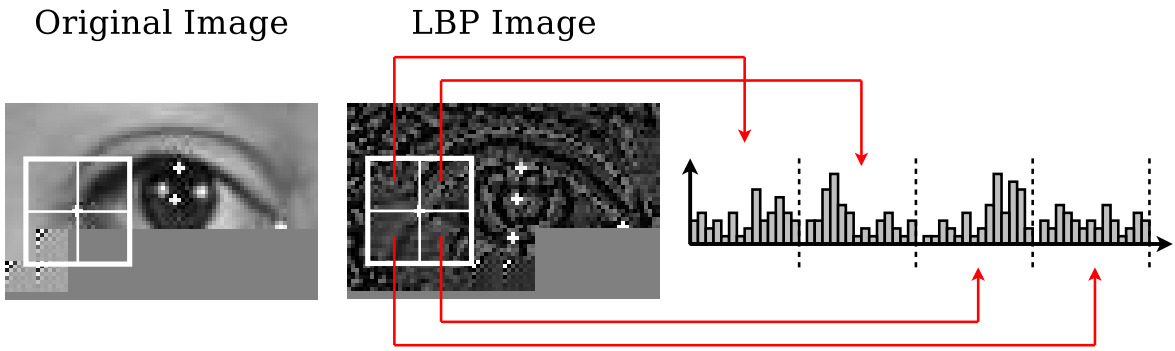


Figure 3.6: Local appearance representation using a divided square

This representation has many interesting properties. First, it is robust to illumination changes since it uses LBPs. Second, it allows us to capture appearance patterns of any size since the square's dimension can easily be changed. Third, it contains information on three different levels: the LBP labels describe the pixel-level patterns, the histograms extracted from the small regions provide information in a regional level and the regional histograms are concatenated to build a global description of the gray-level structures around each landmark point. And last but not the least, it is easy to compute.

Chapter 4

Experiments and Results

This chapter describes the experiments we did in order to compare the performances of the different approaches presented in the previous chapters: original ASM, ELBP-ASM, profile-based LBP-ASM, square-based LBP-ASM and divided-square-based LBP-ASM. Each algorithm has been implemented using Torch3vision[1] which is a machine vision library written in C++ and developed at IDIAP.

The tests have been carried out using the standard and darkened image sets of the XM2VTS database. Comparative results are presented and discussed.

4.1 Dataset

The XM2VTS database[2] [14] consists in face images of 295 subjects collected over four sessions, at one mouth intervals. It was originally designed for research and development of personal identity verification systems but it has been used to evaluate performances of facial feature detection algorithms as well. In this work, we use the frontal image set and the darkened frontal view images.

The frontal image set contains two frontal views for each of the 295 subjects and each of the four sessions. The 2360 images are at resolution 720×576 pixels. They have been taken under controlled conditions against a flat blue background. The face is large in the image and there is no background clutter. The subjects were volunteers of both sexes and many ethnical origins. Since the data acquisition was distributed over a long period of time, significant variability of appearance of individuals, e.g. changes of hair style, facial hair shape and presence or absence of glasses, is present in the recordings. Some examples are shown in Figure 4.1.

The darkened image dataset contains four frontal views for each of the 295 subjects taken from the final session. In two of the images, the studio light illuminating the left side of the face was turned off. In the other two images, the light illuminating the right side of the face was turned off. See Figure 4.2.

[1] Torch3vision; `http://www.idiap.ch/~marcel/en/torch3/torch3vision.php`

[2] XM2VTSDB; `http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/`

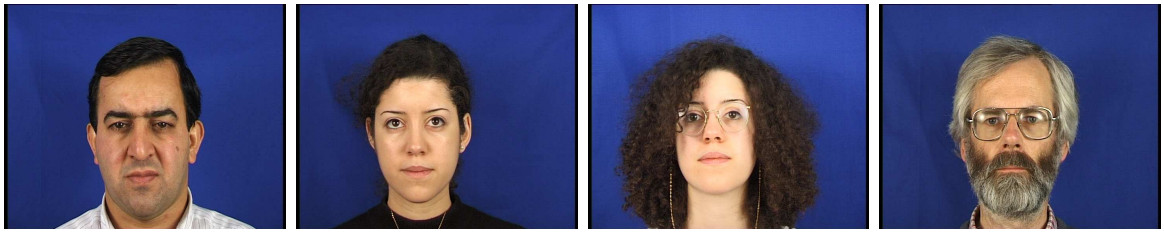


Figure 4.1: Sample images from the standard image set

Figure 4.2: Sample images from the darkened image set

The standard and darkened image sets are both supplied with manually located eye center positions. However to enable more detailed testing and model building, the XM2VTS markup[3] has been expanded to landmarking 68 facial features on each face of the standard image set. The 68 points chosen are shown in Figure 2.1. Since the ground-truth position of these landmark points are not available for the darkened image set, tests on this dataset will essentially be based on the eye locations.

4.2 Experimental Setup

From the standard dataset, training set, evaluation set and test set are built according to the Lausanne protocol [14]. The Lausanne protocol was originally defined for the task of person verification. The standard image set is randomly divided into 200 clients, 25 evaluation impostors and 70 test impostors. It exists two configurations that differ in the distribution of client training and client evaluation data. For our experiments, we use *configuration I* which is illustrated in Figure 4.3.

The training set is used to build the face shape model and the local gray-level structures models. The evaluation set is then used to find the optimal search parameters. Finally the test set is selected to evaluate the performance of the facial feature detection algorithms. In order to test the system robustness to illumination changes, the detection is performed on the darkened images using the shape model and search

[3]Available on Tim Cootes' web site:
`http://www.isbe.man.ac.uk/~bim/data/xm2vts/xm2vts_markup.html`

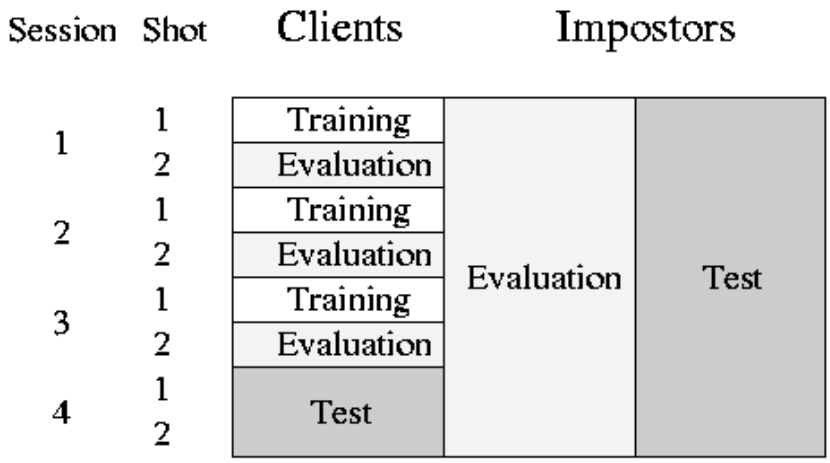


Figure 4.3: Partitioning of the XM2VTS database according to Lausanne protocol Configuration I

parameters obtained with the standard image set.

We assume that the facial feature detection follows a face detection step. The shape model is then initialized according to the estimated eye positions output by the face detector.

4.3 Training Part

From the training set, we build a statistical model for each method described in the previous chapters: original ASM, ELBP-ASM, profile-based LBP-ASM, square-based LBP-ASM and divided-square-based LBP-ASM. The building process of each model requires the choice of three parameters:

- the number of landmark points
- the number of modes to use
- the size of the local appearance pattern descriptor

The number of landmark points is equal to 68 and the number of modes is chosen so that the model represents 98% of the variance. As a result, 58 modes are retained. For the original ASM and the profile-based LBP-ASM, 12 pixels along the normal profile are sampled either side of the landmark point in order to build the local structure model. To simplify the implementation, the ELBP histogram is built using the LBP values contained within a square instead of a disk. The size of the square used in the ELBP-ASM, the square-based LBP-ASM and the divided-square-based LBP-ASM, is set to 25 pixels (12 pixels from the landmark point to each side).

4.4 Evaluation Part

The evaluation set is then used to find the optimal search parameters. Each search algorithm requires the choice of four parameters:

- L, the coarsest level of the Gaussian pyramid to search
- n_s, the longest displacement the landmark point can make along the search profile
- it_{max}, the maximum number of iterations allowed at each level
- q, the proportion of points found within the central 50% of the search profiles determining when to change pyramid level

However, we noticed during experiments that the choice of parameters it_{max} and q does not affect significantly the final shape compared to parameters L and n_s. Therefore, in the following tests, the maximum number of iterations allowed at each level is set to 20 and the shape model is projected to a lower level when 95% of points are found within the central 50% of the search profiles.

In order to measure the quality of fit of the resulting shapes to the ground-truth model, we compute the mean square error and estimate the point location accuracy.

4.4.1 Mean Square Error

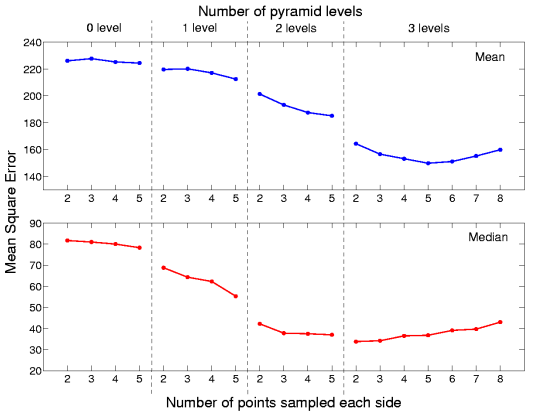
The mean square error (MSE) is given by:

$$MSE = \frac{1}{2n}\sum_{i=0}^{2n}(x_i - gt_i)^2 \tag{4.1}$$

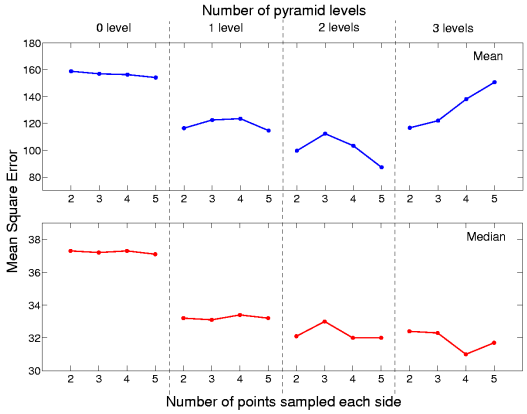
where n is the number of landmark points ($n = 68$), $\mathbf{x}$ is the search vector and $\mathbf{gt}$ is the ground-truth vector.

Figure 4.4 shows the MSE mean and median measured for each algorithm given different combinations of L and n_s. The median is the value in the middle of the MSE distribution: half the MSE measures are above the median and half are below it. The variances have also been calculated but are not represented on the graphs due to their large values.

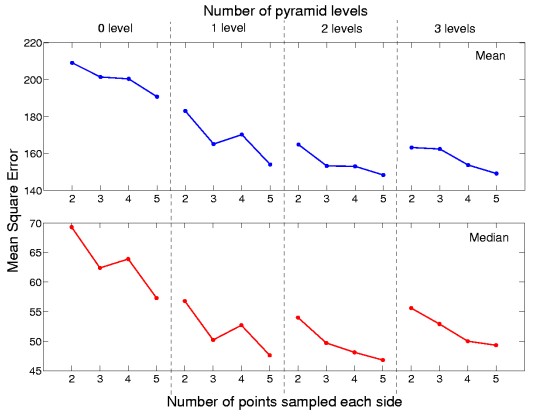
We observe that the median is always much smaller than the mean. This indicates that the MSE distribution is highly skewed. MSEs are typically close to 10 when the system converges to a good solution, whereas they can go up to 2000 when the detection fails. Therefore, a small MSE median indicates that the facial feature detection succeeded in most images of the evaluation set. On the other hand, a mean value greater than the median, involves that some large values caused by detection failures, have affected the mean MSE. The median is therefore more appropriate to evaluate the algorithm performances since it is less sensitive to extreme values. The optimal search parameters are consequently given by the combination which produces the smallest MSE median. In order to validate the choices, we measure the point location accuracy.



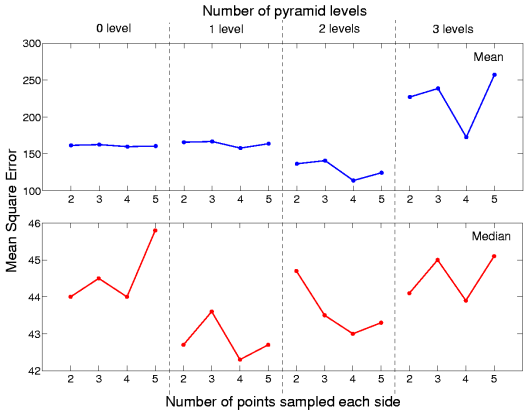
(a)



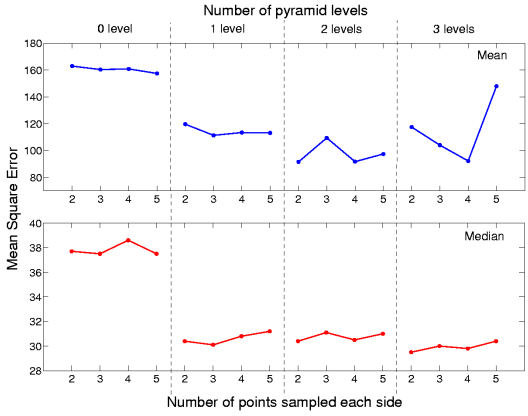
(b)



(c)



(d)



(e)

Figure 4.4: Mean MSE and median of the evaluation set (a) original ASM, (b) ELBP-ASM, (c) profile-based LBP-ASM, (d) square-based LBP-ASM and (e) divided-square-based LBP-ASM

4.4.2 Point Location Accuracy

After search, we measure the distance between the found points and their associated ground-truth position. We then build a frequency histogram for the resulting point-to-target errors. For each algorithm, the frequency histograms of the four best configurations suggested by MSE statistics are compared in Appendix B. The histograms show the proportion of found points whose point-to-target error lies from 0 (perfect match) to 14 pixels. Any point located further than 14 pixels from its corresponding ground-truth position is considered as a failure. Therefore, we want to maximize the proportion of points close to the target while minimizing the number of detection failures. This method is then more reliable than the MSE median since it provides more information on the whole set of shapes and it is not influenced by convergence failures. As a result, the optimal parameters are chosen based on this method. Most of the time, they correspond to the combination selected with the MSE median.

Table 4.1 summarizes the parameters selected for each algorithm.

Method	L	n_s	it_{max}	q
original ASM	3	3	20	0.95
ELBP-ASM	3	4	20	0.95
profile-based LBP-ASM	2	5	20	0.95
square-based LBP-ASM	1	4	20	0.95
divided-square-based LBP-ASM	2	2	20	0.95

Table 4.1: Optimal search parameters. it_{max} and q are fixed.

4.5 Test results and Discussion

4.5.1 Mean Square Error

The image search is performed on each image of the test set using the parameters chosen in the evaluation part. Figure 4.5 shows the MSE mean and median obtained with each algorithm.

The divided-square-based LBP-ASM seems to give better results than the other approaches since it has the smallest median. However, due to the reasons explained in Section 4.4.1, this test cannot be used to draw conclusion on the performances of each algorithm. It only gives a first insight.

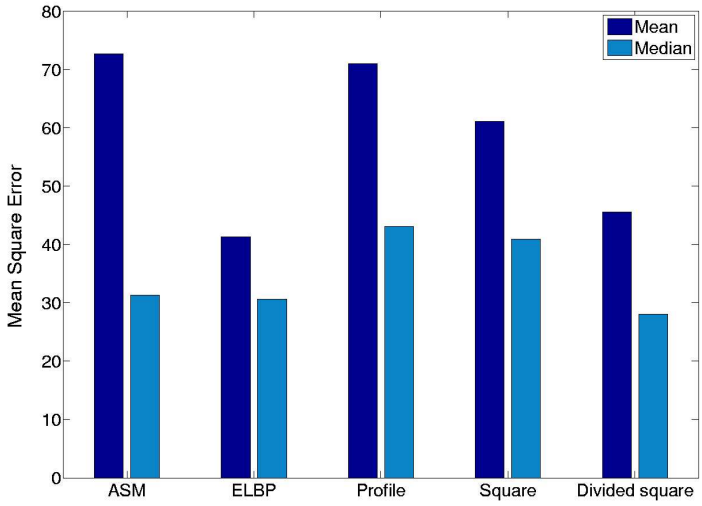


Figure 4.5: Mean MSE and median of the test set

4.5.2 Point Location Accuracy

The frequency histograms of the point-to-target errors described in Section 4.4.2 are compared in Figure 4.6.

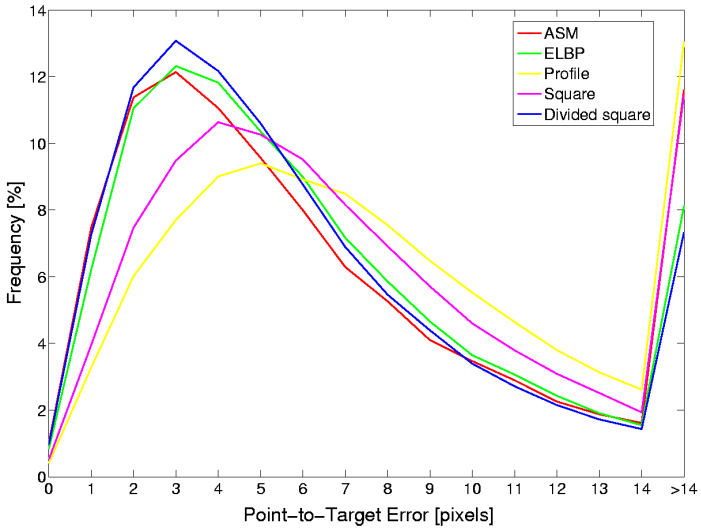


Figure 4.6: Frequency histograms of point-to-target errors of the test set

As expected, the performance of the profile-based LBP-ASM is very limited. LBP histograms extracted from a profile are not reliable local appearance descriptors due to the small number of points they are made of. Using a square region instead of a profile is a good idea but the results of the square-based LBP-ASM show the relevance of retaining spatial information. Indeed, we observe that our proposed method based on a divided square gives much more accurate results and less detection failures than

the other approaches. The ELBP-ASM locates the points slightly less accurately than the original ASM but fails less frequently. The small failure rate of the divided-square-based LBP-ASM and the ELBP-ASM is due to the good ability of a square to catch the target gray-level structure within it. Theses two algorithms are then less likely to diverge. We can also notice from Figure 4.6 the difference of accuracy between our approach and Huang *et al.*'s one. The ELBP histogram gathers too much information that can not be totally exploited during search. As a result, this affects the ELBP-ASM's performance.

4.5.3 Robustness to illumination

In order to test the system robustness to illumination changes, the detection is performed on the darkened images using the shape model and search parameters obtained with the standard image set. Facial feature localization is particularly difficult in this case because the lighting conditions during search are considerably different from the lighting conditions used to train the shape model. Since only the ground-truth eye center positions are available for this set of images, the quality of fit is assessed using the eye location accuracy and the Jesorsky's measure [11].

Let C_l (respectively C_r) be the true left (resp. right) eye coordinate position and let $\tilde{C}_l$ (resp. $\tilde{C}_r$) be the left (resp. right) eye position estimated by the facial feature detector. Jesorsky's measure can be written as

$$d_{eye} = \frac{max(d(C_l, \tilde{C}_l), d(C_r, \tilde{C}_r))}{\|C_r - C_r\|} \tag{4.2}$$

where $d(a, b)$ is the Euclidian distance between positions a and b. A successful localization is accounted if $d_{eye} < 0.25$ (which corresponds approximately to half the width of an eye).

Figure 4.7 presents the mean Jesorsky's measure and the median derived from the standard test image set and the darkened image set. Figure 4.8 shows the frequency histogram of the point-to-target errors corresponding to the eye center positions computed on the darkened images.

In Figure 4.7 and 4.8, the detector's values correspond to the measures obtained after the face detection stage (before facial feature detection). As expected, the original ASM, the ELBP-ASM and the divided-square-based LBP-ASM improve significantly the Jesorsky's measure for the standard test images. However, we can see that ELBP-ASM completely fails on darkened images. The ELBP histogram is based on 6 images: the $LBP^{u2}_{8,1}$, $LBP^{u2}_{8,2}$, $LBP^{u2}_{8,2}$ of the original image and the $LBP^{u2}_{8,1}$, $LBP^{u2}_{8,2}$, $LBP^{u2}_{8,2}$ of the gradient magnitude image. When lighting conditions change, each image is degraded in a different way. Therefore, the ELBP histogram obtained by summing up the six LBP histograms is considerably different from the mean histogram trained on standard images. The algorithm diverges then more frequently. This problem is not present in the approaches we propose. We observe in Figure 4.8, that the square-based LBP-ASM and the divided-square-based LBP-ASM are more robust to illumination

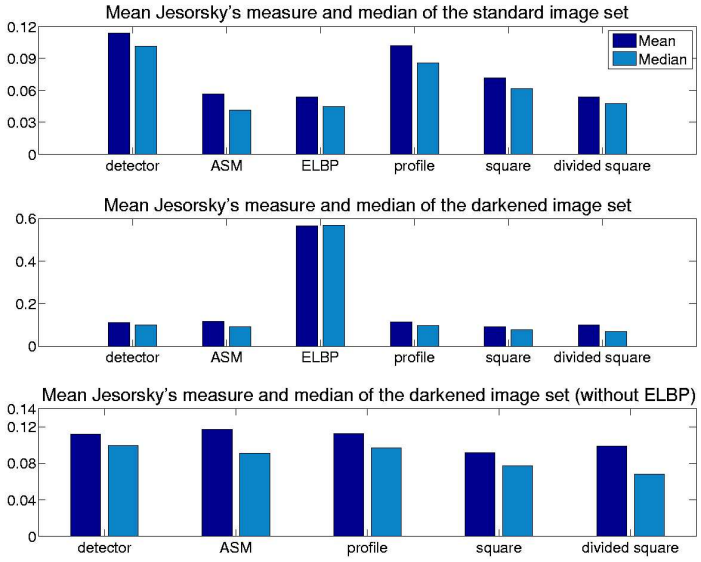


Figure 4.7: Mean Jesorsky's measure and median of the standard test image set and darkened image set

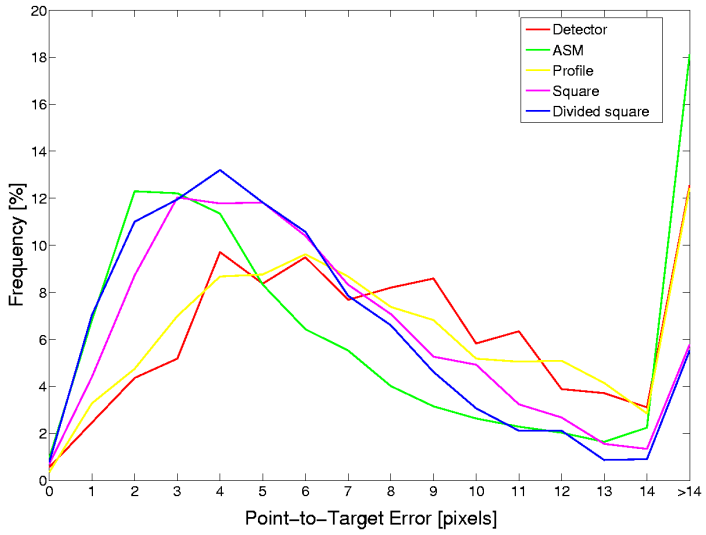


Figure 4.8: Frequency histograms of point-to-target errors corresponding to the eye center positions computed on the darkened image set

changes than the original ASM. Indeed, the eye localization failure rates are much lower.

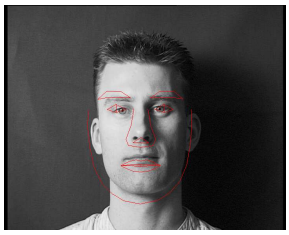
When the facial feature localization is used for face recognition, it is important to locate accurately the eye center positions. However, in other applications, minimizing the Jesorsky's measure is not sufficient. Indeed, the Jesorsky's measure expresses only partially the quality of fit. The system can properly locate the eye center and fail on the other facial features. In order to perform more detailed tests, it would have been useful to annotate the 1160 darkened images with the same 68 landmark points. Unfortunately, it could not be done during this work due to time constraints. Figure 4.9 shows examples of search on a darkened image using the original ASM, the ELBP-ASM and the divided-square-based LBP-ASM. We can observe that the facial feature localization performed by the divided-square-based LBP-ASM is the most accurate whereas the Jesorsky's measure is not the lowest.

4.5.4 Computation Times

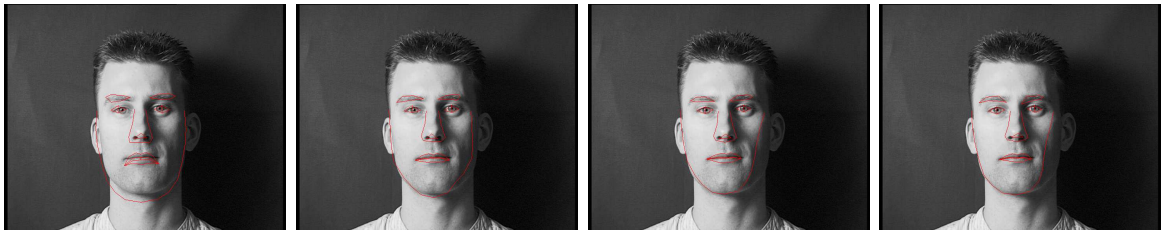
Table 4.2 summarizes the computation times and the average number of iterations that the five algorithms need to converge. Experiments were performed on a 1GHz PC with 1GB memory.

Method	Computation time (s)	# of iterations
original ASM	2.3	12.6
ELBP-ASM	29	13.4
profile-based LBP-ASM	5.3	38.9
square-based LBP-ASM	4.4	14.6
divided-square-based LBP-ASM	7.4	23.4

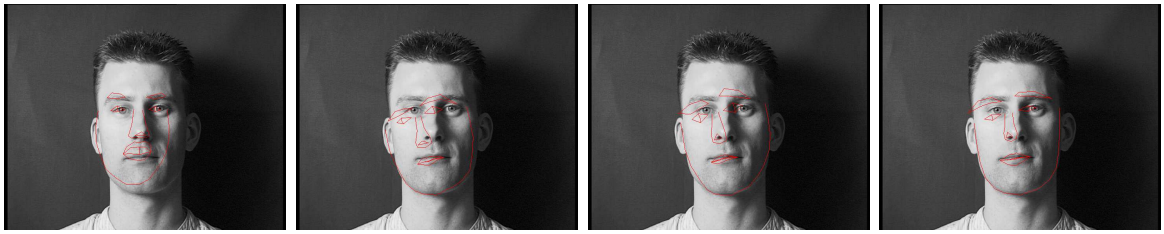
Table 4.2: Computation times and average numbers of iterations



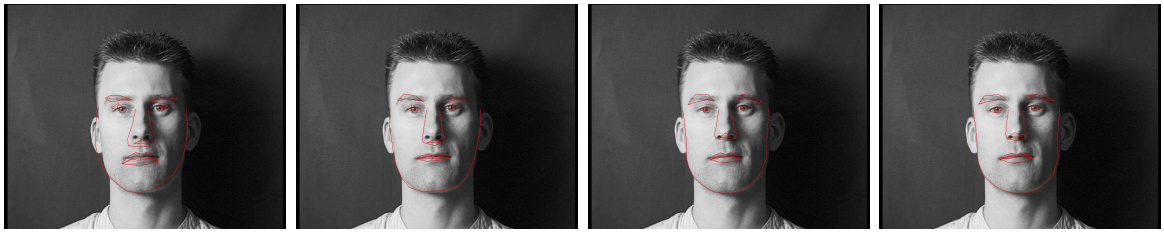
(a) Initial Condition. Jesorsky's measure before facial feature detection = 0.181623



(b) ASM: iteration 1, 4, 8 and 13. Jesorsky's measure = 0.023976



(c) ELBP: iteration 1, 16, 25 and 32. Jesorsky's measure = 0.241385



(d) divided-square-based LBP-ASM: iteration 1, 5, 10 and 19. Jesorsky's measure = 0.039618

Figure 4.9: Example of search on a darkened image using the original ASM, the ELBP-ASM and the divided-square-based LBP-ASM

Chapter 5

Conclusion and Future Work

In this thesis, we extended the Active Shape Model method proposed by Cootes *et al.* in order to improve its robustness to illumination changes. Three different approaches using Local Binary Patterns to model the structures around each landmark point were proposed:

Profile-based LBP-ASM The local appearance patterns are described using LBP histograms extracted from the normal profile of each landmark point. Similar to the original ASM, this method suffers from the limited ability of normal profiles to describe complex structures.

Square-based LBP-ASM The local structures are modeled using LBP histograms extracted from a square region around each landmark point. This method acquires more information related to the local appearance patterns but does not retain spatial information.

Divided-square-based LBP-ASM The square region used in the square-based LBP-ASM is divided into four regions from which the LBP histograms are extracted and concatenated into a single feature histogram representing the local appearance patterns.

Although this thesis focused on facial feature detection, the proposed algorithms can be used to find any deformable object in an image. A prerequisite is only to collect a training set of images containing instances of the object to be modeled.

Experiments were performed in order to compare those three approaches with the original ASM and the only method combining ASM and LBP exiting so far, ELBP-ASM. The tests were carried out using the standard and darkened image sets of the XM2VTS database.

Experiments on the standard image set demonstrated that the divided-square-based LBP-ASM achieves more accurate results and fails less frequently than the other approaches. The accuracy can still be improved by using more landmark points. Indeed,

68 landmarks were used whereas facial feature localization is usually performed using at least 133 landmark points.

Experiments on darkened images only gave us an insight into the robustness to illumination changes of the proposed algorithms. Since only the eye center ground-truth positions were available, tests were based on the Jesorsky's measure. As expected, the divided-square-based LBP-ASM is the most robust to illumination changes. However, we showed through an example that a large Jesorsky's measure does not mean that the facial feature detection failed completely. Therefore, although the results look very promising, more experiments still have to be done before drawing any final conclusion.

Since good results could be achieved by combining ASM with LBP, a logical continuation of this project would be to extend the divided-square-based method to Active Appearance Model.

Appendix A

Aligning Two 2D Shapes

Given two 2D shapes, $\mathbf{x}$ and $\mathbf{x}'$, we wish to find the similarity transformation $T(\cdot)$ which, when applied to $\mathbf{x}$, minimizes the least squares distance between the two shapes, as follows.

$$E = |T(\mathbf{x}) - \mathbf{x}'|^2 \tag{A.1}$$

The two dimensional similarity transformation is define as

$$T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{A.2}$$

Without loss of generality, both shapes are first translated so that their center of gravity is on the origin. Thus, $t_x = t_y = 0$.

We wish to find then the scale and rotation which best aligns $\mathbf{x}$ with $\mathbf{x}'$, i.e. minimizes

$$\begin{aligned} E(a,b) &= |T(\mathbf{x}) - \mathbf{x}'|^2 \\ &= \sum_{i=1}^{n}(ax_i - by_i - x'_i)^2 + (bx_i + ay_i - y'_i)^2 \end{aligned} \tag{A.3}$$

Differentiating with respect to both a and b and equating to zero gives

$$\sum_{i=1}^{n} ax_i^2 + ay_i^2 - x_i x'_i - y_i y'_i = 0 \tag{A.4}$$

$$\sum_{i=1}^{n} bx_i^2 + by_i^2 - x_i y'_i + y_i x'_i = 0 \tag{A.5}$$

This implies

$$a = (\sum_{i=1}^{n} x_i x'_i + y_i y'_i)/(\sum_{i=1}^{n} x_i^2 + y_i^2) = \mathbf{x}\cdot\mathbf{x}'/|\mathbf{x}| \tag{A.6}$$

$$b = (\sum_{i=1}^{n} x_i y'_i - y_i x'_i)/(\sum_{i=1}^{n} x_i^2 + y_i^2) = (\sum_{i=1}^{n} x_i y'_i - y_i x'_i)/|\mathbf{x}| \tag{A.7}$$

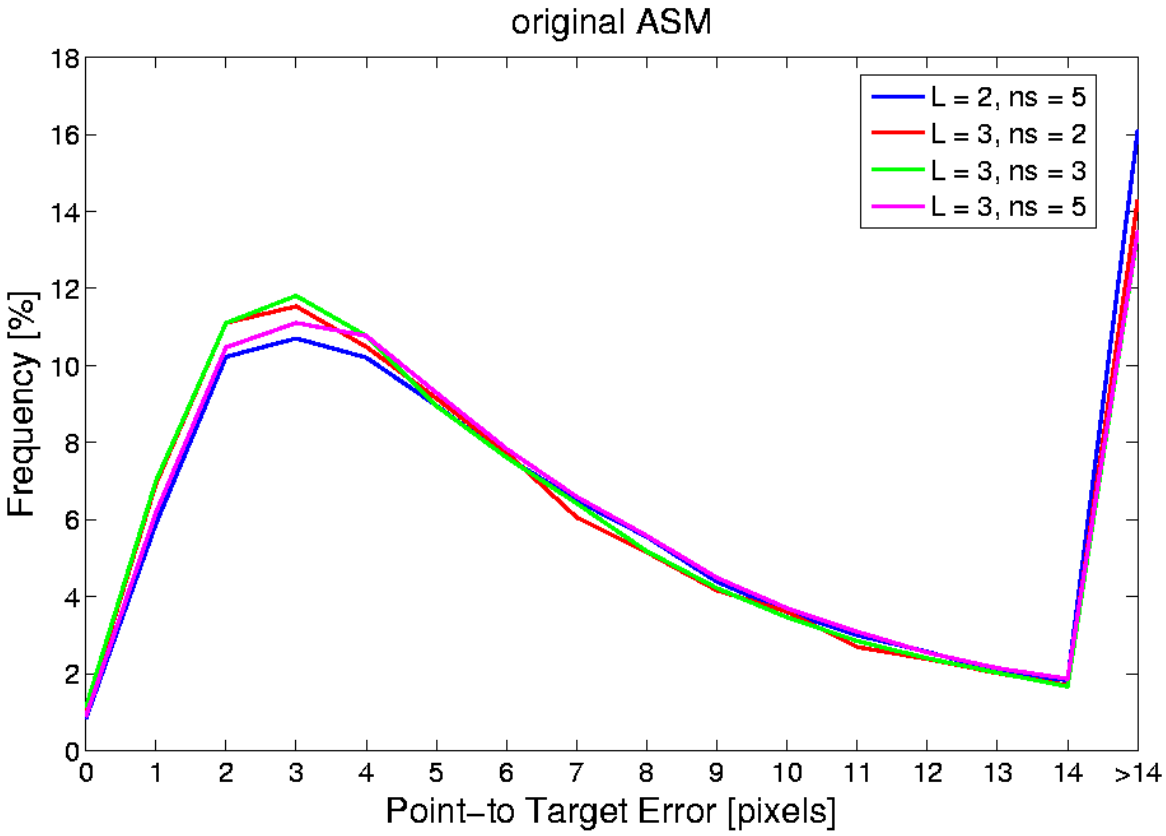
Given a and b, a shape $\mathbf{x}$ can be approximately mapped to a shape $\mathbf{x}'$ as follows

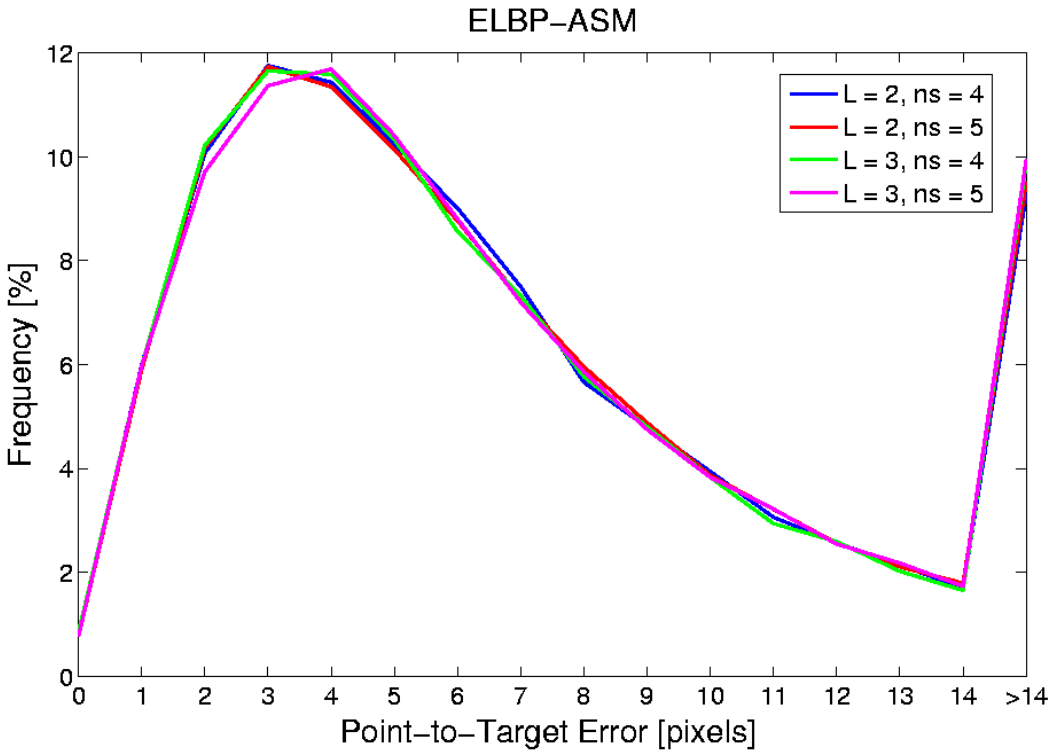
$$\mathbf{x}' \simeq \mathbf{x}'_{\mathbf{c}} + T(\mathbf{x} - \mathbf{x}_{\mathbf{c}}) \tag{A.8}$$

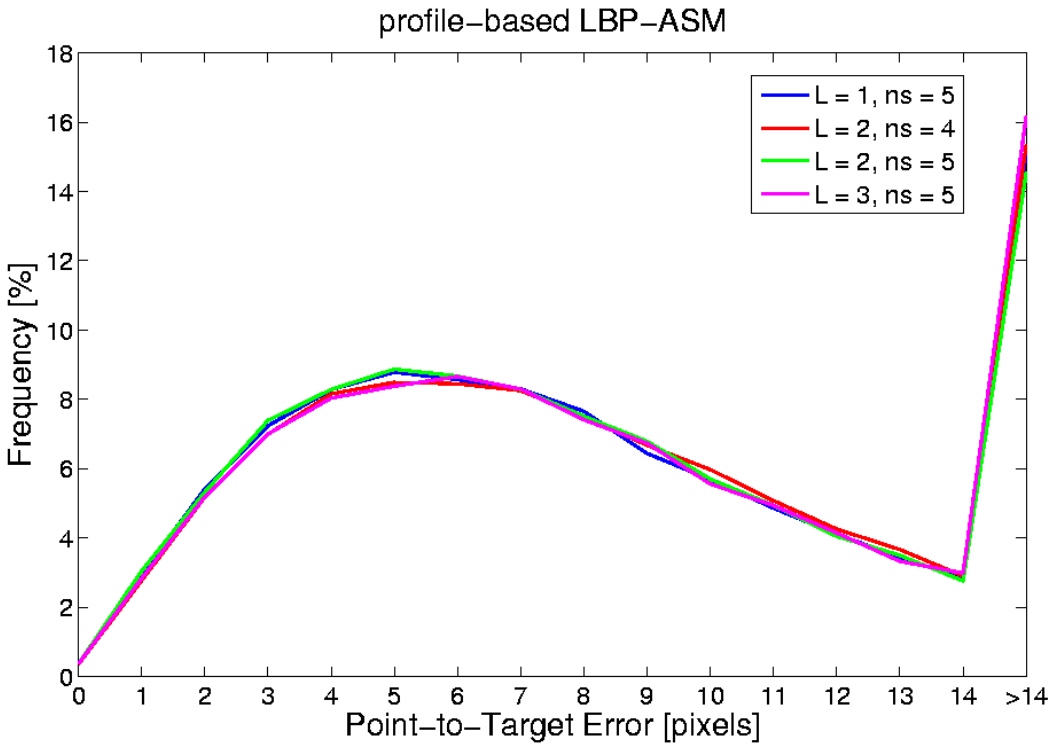
where $\mathbf{x}_{\mathbf{c}}$ is the center of gravity of $\mathbf{x}$ and $\mathbf{x}'_{\mathbf{c}}$, the center of gravity of $\mathbf{x}'$.

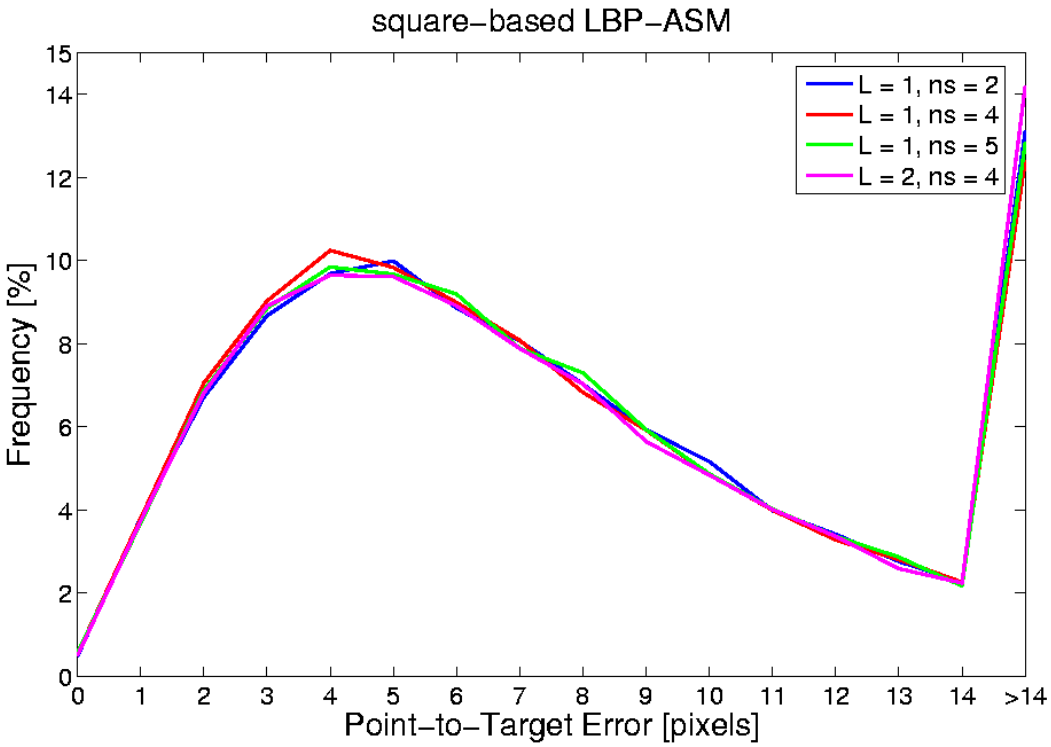
Appendix B

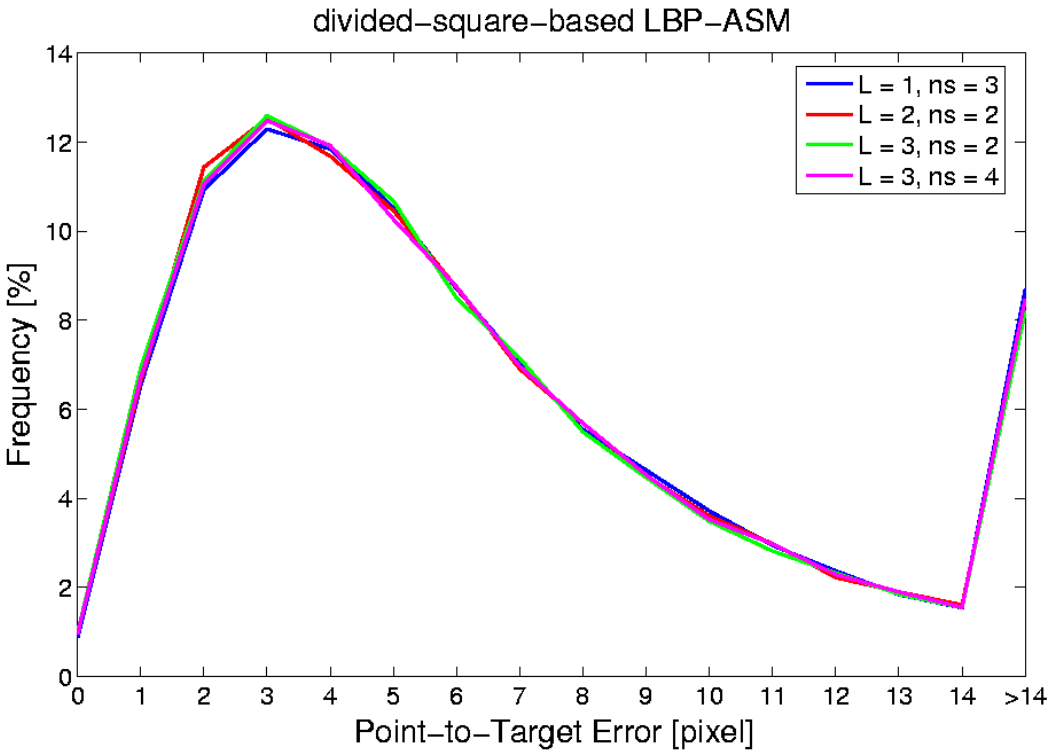
Frequency Histograms of Point-to-Target Errors of the Evaluation Set




ELBP–ASM
L = 2, ns = 4
L = 2, ns = 5
L = 3, ns = 4
L = 3, ns = 5
Frequency [%]
Point–to–Target Error [pixels]
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 >14
0 2 4 6 8 10 12


profile–based LBP–ASM
L = 1, ns = 5
L = 2, ns = 4
L = 2, ns = 5
L = 3, ns = 5
Frequency [%]
Point–to–Target Error [pixels]
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 >14
0 2 4 6 8 10 12 14 16 18


square–based LBP–ASM
L = 1, ns = 2
L = 1, ns = 4
L = 1, ns = 5
L = 2, ns = 4
Frequency [%]
Point–to–Target Error [pixels]
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 >14
0 2 4 6 8 10 12 14 15


divided–square–based LBP–ASM
L = 1, ns = 3
L = 2, ns = 2
L = 3, ns = 2
L = 3, ns = 4
Frequency [%]
Point–to–Target Error [pixel]
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 >14
0 2 4 6 8 10 12 14

Bibliography

[1] T.F. Cootes, G.J. Edwards, and Taylor C.J. Active appearance models. In *5th European Conference on Computer Vision*, volume 2, pages 484–498, Berlin, Germany, 1998. Springer.

[2] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Comparing active shape models with active appearance models. In *British Machine Vision Conference*, Nottingham, UK, September 1999.

[3] T.F. Cootes, A. Hill, C.J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. In *Proceedings of the 13th International Conference on Information Processing in Medical Imaging*, pages 33 – 47, London, UK, 1993. Springer-Verlag.

[4] T.F. Cootes and C.J. Taylor. Statistical models of appearance for computer vision. Technical report, Dept of Imaging Science and Biomedical Engineering, University of Manchester, March 2004.

[5] T.F. Cootes, C.J. Taylor, D. Cooper, and J. Graham. Active shape models – their training and applications. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[6] T.F. Cootes, C.J. Taylor, and A. Lanitis. Active shape models: evaluation of a multi-resolution method for improving image search. In *Proc. British Machine Vision Conference*, pages 327–336, 1994.

[7] D. Cristinacce. *Automatic Detection of Facial Features in Grey Scale Images*. PhD thesis, University of Manchester, 2004.

[8] I.L. Dryden and K.V. Mardia. *Statistical Shape Analysis*. Wiley Series in Probability and Statistics. Wiley and Sons, 1998.

[9] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society B*, 53(2):285 – 339, 1991.

[10] X. Huang, S. Li, and Y. Wang. Shape localization based on statistical method using extended local binary pattern. In *Third International Conference on Image and Graphics (ICIG)*, pages 184–187, Hong Kong, China, 2004.

[11] O. Jesorsky, K.J. Kirchberg, and R.W. Frischholz. Robust face detection using the hausdorff distance. In *Third International Conference on Audio and Video-Based Biometric Person Authentication*, volume 2091 of *Lecture Notes in Computer Science*, pages 90 – 95, Halmstad, Sweden, 2001. Springer.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259 – 268, London, UK, June 1987.

[13] A. Lanitis, C.J. Taylor, T.F. Cootes, and T. Ahmed. Automatic face identification system using flexible appearance models. *IVC*, 13(5):393–401, June 1995.

[14] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *Second International Conference on Audio and Video-based Biometric Person Authentication*, Washington D.C, USA, March 1999.

[15] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51 – 59, 1996.

[16] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with loval binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:971 – 987, 2002.