Local Features based Facial Expression Recognition with Face Registration Errors

Tommaso Gritti, Caifeng Shan, Vincent Jeanne, Ralph Braspenning
Philips Research, High Tech Campus 36, Eindhoven 5656 AE, The Netherlands

Abstract

In this paper, we extensively investigate local features based facial expression recognition with face registration errors, which has never been addressed before. Our contributions are three fold. Firstly, we propose and experimentally study the Histogram of Oriented Gradients (HOG) descriptors for facial representation. Secondly, we present facial representations based on Local Binary Patterns (LBP) and Local Ternary Patterns (LTP) extracted from overlapping local regions. Thirdly, we quantitatively study the impact of face registration errors on facial expression recognition using different facial representations. Overall LBP with overlapping gives the best performance (92.9% recognition rate on the Cohn-Kanade database), while maintaining a compact feature vector and best robustness against face registration errors.

1. Introduction

Driven by its important applications and theoretical interests, automatic facial expression recognition has attracted much attention in the last two decades [9, 17, 18]. However, although the human cognitive process appears to detect and interpret facial expressions with little or no effort, it is still difficult to develop a computer vision system capable of recognizing facial expressions effectively.

A vital step for successful facial expression recognition is deriving an effective facial representation from original face images. Two types of features can be extracted [22]: geometric features and appearance features. Geometric features deal with the shape and locations of facial components (including mouth, eyes, brows, and nose), which are extracted to represent the face geometry. Appearance features present the appearance changes (skin texture) of the face, including wrinkles, bulges and furrows, extracted by applying image filters such as Gabor wavelets [5] to either the whole face or specific facial regions. The geometric feature based facial representations commonly require accurate and reliable facial feature detection and tracking, which is difficult to accommodate in many situations. Moreover, geometric features usually cannot encode changes in skin texture that are critical for facial expression modeling. In contrast to geometric features, appearance features suffer less from issues of initialization and tracking errors, and can encode changes in skin texture. Appearance features include Gabor wavelets [5, 14], Haar-like wavelets [25], the learned statistical image filters [6] (such as Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis and Local Feature Analysis), and temporal templates [23]. Combining appearance and geometric features [3] is a promising way for better facial representation.

Recently Histogram of Oriented Gradients (HOG) [4] descriptors have been introduced for human detection. The method counts occurrences of gradient orientation in local image regions, and the local histograms can be normalized using the overlapping local contrast for improved performance. Human faces are characterized by local appearance and shape which can be effectively represented by the HOG descriptors. However, the HOG descriptors have not been exploited for face image analysis. In this paper, we exploit HOG features for facial expression recognition. We systematically study the effects of various implementation choices on facial expression recognition performance. We further compare HOG descriptors with other widely used local features including Local Binary Patterns (LBP) [1] and Gabor wavelets, and more recent Local Ternary Patterns (LTP) [20].

In most of the existing work on facial expression recognition, the recognition performance is assessed on face images that are normalized based on the eyes positions (manually annotated). However, in the context of a fully-automatic expression recognition system, the faces presented to the recognizer may not be normalized perfectly. With current face detection [12] and eye localization [8] methods, face registration errors always occur, which can grow significantly in real-world unconstrained lighting and head pose conditions. Therefore, it is necessary to inves-
tigate the above facial representations under conditions of face registration errors. In this paper, we assess the impact of face registration errors on facial representations, including HOG descriptors, LBP and its variants, and Gabor wavelets, for facial expression recognition.

Our contributions are three fold: firstly we extensively study the HOG descriptors for facial representation; secondly, we present LBP/LTP with overlapping, which is much robust to face registration errors; thirdly, we quantitatively investigate the impact of face registration errors on facial expression recognition using different facial representations. The paper is organized as follows. We first introduce the data set and classifier used in this study (Section 2). In Section 3, we experimentally study different facial representations including HOG, LBP/LTP, LBP/LTP with overlapping and Gabor. We then present experiments on facial expression recognition with face registration errors in Section 4. Section 5 concludes this paper.

2. Data Set and Methodology

Data Set — We carry out facial expression recognition experiments on the Cohn-Kanade database [13], one of the most comprehensive databases in the current facial expression research community. The Cohn-Kanade database consists of 100 university students aged from 18 to 30 years, of which 65% were female, 15% were African-American, and 3% were Asian or Latino. Subjects were instructed to perform a series of 23 facial displays, six of which were based on description of prototypic emotions (i.e. Anger, Disgust, Fear, Joy, Sadness, and Surprise). Image sequences from neutral to target display were digitized into 640×490 pixel arrays with 8-bit precision for grayscale values. We selected 310 image sequences from the database. The only selection criterion was that a sequence could be labeled as one of the six basic emotions. The sequences come from 95 subjects, with 1 to 6 emotions per subject. For each sequence, the neutral face and three peak frames were used for prototypic emotional expression recognition.

Following Tian [21], we scaled the faces to a fixed distance between the two eyes. Face images of 108×147 pixels were cropped from original frames based on the location of the two eyes that were manually labeled.

Methodology — Support Vector Machines (SVM) [24] are a popular technique for facial expression recognition [2]. In this study, we adopt SVM as the classifier for expression recognition. SVM allows domain-specific selection of the kernel function. For simplicity and speed, we use linear SVM in this study.

As SVM makes binary decisions, the multi-class classification here is accomplished by using the one-against-rest technique, which trains binary classifiers to discriminate one expression from all others, and outputs the class with the positive output of binary classification. Each dimension of the training and testing vector was scaled to be between 0 and 1 [11]. We used the SVM implementation in the public available machine learning library SPIDER 1.

To evaluate the generalization performance to novel subjects, we adopted a 10-fold cross-validation testing scheme in our recognition experiments. The average recognition rate plus the standard deviation are reported. To fairly compare all reported features, cross validation sets are defined once and maintained the same for all features.

3. Facial Representation Methods

3.1. Histogram of Oriented Gradients

The HOG descriptors were introduced by Dalal and Triggs [4], and have shown superior performance to other existing feature sets including wavelets for human detection. The essential idea of the HOG descriptors is that local object appearance and shape can be described rather well by the distribution of local intensity gradients or edge directions. The method uses normalized local histograms of image gradient orientations as image features. In practice the HOG features are extracted by dividing the image window into small spatial regions called “cells”, for each cell accumulating a local 1-D histogram of gradient directions over the pixels of the cell. The combined histograms then form the image representation. For improved performance, the local histograms can be normalized to local contrast by calculating a measure of the intensity over larger spatial regions called “blocks” and using the result to normalize all cells within the block. The normalization results in a better invariance to changes in illumination and shadowing.

Taking human detection as a test case, Dalal and Triggs made a detailed study of the HOG implementations. They concluded that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good performance. The HOG descriptors have mainly been investigated for human detection. Here we propose to exploit the HOG representation for facial expression recognition. As the HOG descriptors are new for facial representation, in this section we systematically study the effects of various implementation choices on facial expression recognition performance. Throughout this section we refer results to our default implementation which has the following properties: gray-scale space with no gamma correction; [-1, 0, 1] gradient filter without smoothing; linear gradient voting into 9 orientation bins in 0° – 180°; 16×16 pixel blocks of four 8×8 pixel cells; L2-norm block normalization; block spacing stride of 8 pixels; linear SVM classifier. These default settings are based on Dalal and Triggs’s study [4]. Before computing the HOG, images are resized

---

to $112 \times 160$ pixels using bilinear interpolation in order to fit an integer number of blocks.

We now report in detail the effect of the various HOG parameters on facial expression recognition performance. All parameters not explicitly mentioned are set to the default setting.

**Gradient Computation** — The HOG descriptors depend on the way in which gradients are computed. We tested gradients computed using different discrete derivative masks, including simple 1-D masks $[-1, 0, 1]$, 1-D cubic-corrected $[-1, -8, 0, 8, 1]$, $2 \times 2$ diagonal filters [4], Sobel filters and Prewitt filters. We find that Prewitt filters perform best, at a slightly higher computational cost.

<table>
<thead>
<tr>
<th>Mask Type</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D mask</td>
<td>88.9±3.8</td>
</tr>
<tr>
<td>1D cubic corrected</td>
<td>86.3±7.0</td>
</tr>
<tr>
<td>Diagonal filters</td>
<td>89.5±4.0</td>
</tr>
<tr>
<td>Sobel filters</td>
<td>89.4±4.0</td>
</tr>
<tr>
<td>Prewitt filters</td>
<td>89.7±4.5</td>
</tr>
</tbody>
</table>

Table 1. The effect of different gradient computation on recognition performance (%).

**Orientation Binning** — Each pixel within the cell, which can be either rectangular or radial, casts a weighted vote for an orientation histogram channel based on the orientation of the gradient element centered on it. The orientation bins are evenly spread over $0^\circ - 180^\circ$ ("unsigned" gradient) or $0^\circ - 360^\circ$ ("signed" gradient).

We tested both signed and unsigned gradients with 9, 18 and 36 uniform bins. As observed in Table 2, signed gradients perform best, with a constant increase of performance as the number of bins grows. It is important to point out that the feature length is proportional to the number of bins though.

<table>
<thead>
<tr>
<th>Binning Type</th>
<th>9 bins</th>
<th>18 bins</th>
<th>36 bins</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0^\circ-180^\circ$</td>
<td>88.9±3.6</td>
<td>90.4±3.8</td>
<td>90.5±3.5</td>
</tr>
<tr>
<td>$0^\circ-360^\circ$</td>
<td>90.5±5.3</td>
<td>90.8±4.6</td>
<td><strong>91.7±4.6</strong></td>
</tr>
</tbody>
</table>

Table 2. The effect of different orientation bins and orientation ranges on recognition performance (%).

**Histogram Voting** — The weight with which each pixel contributes to the histogram can be a function of the gradient magnitude at the pixel, either the magnitude itself, its square, its square root, or a binary voting representing presence/absence of an edge at the pixel. We tested different voting methods as showed in Table 3, and found that the magnitude square gives best performance, while the binary edge presence voting is worst.

<table>
<thead>
<tr>
<th>Voting Method</th>
<th>Performance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magnitude</td>
<td>88.9±3.8</td>
</tr>
<tr>
<td>Magnitude Square</td>
<td><strong>89.5±4.7</strong></td>
</tr>
<tr>
<td>Magnitude Square Root</td>
<td>88.7±3.5</td>
</tr>
<tr>
<td>Binary voting</td>
<td>86.3±3.7</td>
</tr>
</tbody>
</table>

Table 3. The effect of different voting methods on recognition performance (%).

**Descriptor Blocks** — In order to account for local variations in illumination and contrast, it is essential to normalize the gradient strengths to local contrast. The normalization schemes group cells into larger spatial blocks and normalize each block separately. We study the influence of cell and block size by testing different combinations. Results are reported in Table 4, where the block overlap is kept constant at half of block size [4] $1$. Considering the image window size, for the large cell size like $24 \times 24$, we do not test with large block size. Grouping cells of $6 \times 6$ or $8 \times 8$ pixels into a block of $3 \times 3$ cells performs best. It is evident that a coarse spatial binning with cells of $16 \times 16$ pixels or more reduces the performances considerably.

<table>
<thead>
<tr>
<th>Cell Size (pixels)</th>
<th>1×1 cells</th>
<th>2×2 cells</th>
<th>3×3 cells</th>
<th>4×4 cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>4×4</td>
<td>89.8±3.4</td>
<td>89.7±3.5</td>
<td>89.9±3.8</td>
<td>90.6±3.7</td>
</tr>
<tr>
<td>6×6</td>
<td>90.7±3.5</td>
<td>90.8±3.3</td>
<td><strong>91.4±3.7</strong></td>
<td>90.8±3.9</td>
</tr>
<tr>
<td>8×8</td>
<td>89.7±3.0</td>
<td>89.9±3.9</td>
<td><strong>91.6±3.2</strong></td>
<td>89.5±4.1</td>
</tr>
<tr>
<td>12×12</td>
<td>89.2±3.4</td>
<td>89.1±3.6</td>
<td>89.1±3.6</td>
<td>-</td>
</tr>
<tr>
<td>16×16</td>
<td>86.7±3.5</td>
<td>86.4±4.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>24×24</td>
<td>83.9±2.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>32×32</td>
<td>80.7±3.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4. The effect of the cell and block sizes on recognition performance (%).

**Block Overlap** — In Dalal and Triggs’s study it is reported that overlapping between blocks improves the performance significantly. With block overlap, each cell contributes more than once to the final descriptor, each normalized with respect to a different block. As shown in Table 5, performance obtained by varying the amount of overlap confirm what has been presented in [4]. A drawback of overlap is the increase in feature length by a factor of 4 (Overlap=1/2) or 16 (Overlap=3/4).

**Normalization** — As mentioned above, a key step in computing the HOG is block normalization. The norm of the vector containing the histogram contributions of all

---

$1$To assure an integer number of blocks in the images, face images are resized from default $112 \times 160$ pixels to $120 \times 168$, $108 \times 144$ or $128 \times 160$ pixels depending on the block side in use.
cells belonging to a given block is computed. Each block’s histogram is then normalized with its respective value. Following [4], we evaluated three different normalization schemes: L2-norm, L1-norm, and L1-sqrt. We also tested the case without normalization. L2-norm performs slightly better. We did not observe the large drop in performance reported in [4], most likely because face images in our data set are captured under constant illumination.

<table>
<thead>
<tr>
<th></th>
<th>L1</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overlap=0, Stride 16</td>
<td>87.2±3.9</td>
<td>87.4±4.5</td>
</tr>
<tr>
<td>Overlap=1/2, Stride 8</td>
<td>88.6±4.1</td>
<td>88.9±3.8</td>
</tr>
<tr>
<td>Overlap=3/4, Stride 4</td>
<td>91.2±3.8</td>
<td>91.7±3.4</td>
</tr>
</tbody>
</table>

Table 5. The effect of overlap on recognition performance (%).

We observe that, by fine tuning HOG parameters, it is possible to increase performances considerably for a specific problem. It is thus recommended to extensively study HOG performance for any new given problem.

### 3.2. Local Binary Patterns

The original LBP operator was introduced by Ojala et al. [15], and has proven to be a powerful texture descriptor. The operator labels the pixels of an image by thresholding a 3 x 3 neighborhood of each pixel with the center value and considering the results as a binary number. Formally, given a pixel at \((x_c, y_c)\), the resulting LBP can be expressed in the decimal form as

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

where \(n\) runs over the 8 neighbors of the central pixel, \(i_c\) and \(i_n\) are the gray-level values of the central pixel and the surrounding pixel, and \(s(x)\) is 1 if \(x \geq 0\) and 0 otherwise.

Ojala et al. [16] later made two extensions of the original operator. Firstly, the operator was extended to use neighborhood of different sizes, to capture dominant features at different scales. Using circular neighborhoods and bilinearly interpolating the pixel values allows any radius and number of pixels in the neighborhood. The notation \((P, R)\) denotes a neighborhood of \(P\) equally spaced sampling points on a circle of radius of \(R\). Secondly, they proposed to use a small subset of the \(2^P\) patterns, produced by the operator \(LBP_{P,R}\), to describe the texture of images. These patterns, called uniform patterns, contain at most two bitwise transitions from 0 to 1 or vice versa when considered as a circular binary string. For example, 00000000, 001110000 and 11100001 are uniform patterns. The uniform patterns represent local primitives such as edges and corners. It was observed that most of the texture information was contained in the uniform patterns. Labeling the patterns which have more than 2 transitions with a single label yields an LBP operator, denoted \(LBP_{P,R}^{u}\), which produces much less patterns without losing too much information.

After labeling an image with a LBP operator, a histogram of the labeled image \(f_i(x, y)\) can be defined as

\[
H_i = \sum_{x,y} I(f_i(x, y) = i), \quad i = 0, \ldots, L - 1
\]  

where \(L\) is the number of different labels produced by the LBP operator and

\[
I(A) = \begin{cases} 
1 & \text{if } A \text{ is true} \\
0 & \text{if } A \text{ is false}
\end{cases}
\]  

The derived LBP histogram contains information about the distribution of local micro-patterns, such as edges, spots and flat areas, over the image, so can be used to statistically describe image characteristics.

Each face image can be seen as a composition of micro-patterns which can be effectively detected by the LBP operator, so it is intuitive to use LBP features for face representation. Ahonen et al. [1] introduced a face recognition method using LBP based face representation. To also consider the shape information of faces, in their method face images are equally divided into \(M\) small non-overlapping regions \(R_0, R_1, \ldots, R_M\) to extract LBP histograms (as shown in Fig. 1). The LBP histograms extracted from each region are then concatenated into a single, spatially enhanced feature histogram defined as:

\[
H_{i,j} = \sum_{x,y} I(f_i(x, y) = i)I((x, y) \in R_j)
\]  

where \(i = 0, \ldots, L - 1, j = 0, \ldots, M - 1\). The extracted feature histogram describes the local texture and global shape of face images. It contains facial information on three different levels [1]: the labels for the regional histogram (pixel-level), the regional histogram (regional-level) and the concatenated histogram which builds a global description of the face (image level). Using a nearest-neighbor classifier with Chi square as a dissimilarity measure, their scheme achieved superior recognition performance on the FERET database. The LBP based face representation has also been used with success for facial expression recognition [10, 19].
Figure 1. A face image is divided into small regions from which LBP histograms are extracted and concatenated into a single, spatially enhanced feature histogram.

Some parameters can be optimized for LBP-based facial representation. One is the LBP operator, and another is the number of regions divided. Following the existing work [1, 19], we selected the $LBP_{u,c,t}^{n}$ operator, which has 59 labels, and divided the $108 \times 147$ pixels facial images into 18×21 pixels regions, giving a good trade-off between recognition performance and feature vector length. Thus the facial image was divided into 42 (6×7) regions as shown in Figure 1. The performance of this method is reported in the top row of Table 7.

**Local Ternary Patterns (LTP)** — As the LBP operator thresholds the neighborhood with exactly the center value, a bit of noise to the center value may change the LBP code dramatically. To address this, more recently Tan and Triggs [20] introduced an extension of LBP called Local Ternary Patterns (LTP) —

![Image](https://via.placeholder.com/150)

that is less sensitive to noise in near-uniform regions. In LTP, neighboring pixel values in a zone of width ±1 around $i_{c}$ are quantized to zeros, ones above this are quantized to +1 and ones below it to −1:

$$s'(i_{n}, i_{c}, t) = \begin{cases} 
1, & i_{n} \geq i_{c} + t \\
0, & |i_{n} - i_{c}| < t \\
-1, & i_{n} \leq i_{c} - t 
\end{cases}$$

where $t$ is a user-specified threshold. So a ternary code instead of a binary code is produced by LTP. They then adopted a coding scheme that splits each ternary pattern into its positive and negative parts, subsequently treating these as two separate channels of LBP descriptors. To evaluate this approach we adopt the same parameters chosen for LBP, and set the needed threshold value to 6 (which we found provides the best performance on the given dataset). The performance of this method is reported in the top row of Table 7.

### 3.3. LBP/LTP with Overlapping

In the above LBP/LTP implementation, face images are divided into non-overlapping regions for calculating local LBP/LTP histograms. However, as observed in Table 5, overlapping between blocks can increase performance significantly for the HOG descriptors. Thus it is interesting to investigate LBP/LTP histograms extracted from overlapping regions, which can potentially provide better performance.

We tested different overlaps between local regions and report the results in Table 7. We see that overlap between regions indeed achieves better results. In the case of LBP no improvement is achieved by using overlap=3/4 compared to overlap=1/2. In the case of LTP, the performance increases constantly from no overlap to overlap=3/4. However, the improved performance comes at the cost of increased feature length (by a factor of 4 for overlap=1/2 or a factor of 16 for overlap=3/4).

<table>
<thead>
<tr>
<th>Overlap</th>
<th>LBP</th>
<th>LTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>90.9±5.6</td>
<td>90.9±4.9</td>
</tr>
<tr>
<td>1/2</td>
<td>92.9±5.0</td>
<td>91.7±5.5</td>
</tr>
<tr>
<td>3/4</td>
<td>92.6±5.0</td>
<td>92.6±4.7</td>
</tr>
</tbody>
</table>

Table 7. The recognition performance (%) of LBP/LTP with overlapping.

### 3.4. Gabor Filters

Gabor wavelet features have been widely used in face image analysis [14]. Here we also study Gabor features for facial expression recognition to provide a comparison to the other presented features, being aware that, as a main drawback, Gabor features require a very high computational cost. Following Bartlett et al. [2], we converted images into a Gabor magnitude representation using a bank of Gabor filters at 8 orientations and 5 spatial frequencies (9.36 pixels per cycle at 1/2 octave steps$^2$). To reduce the length of the feature vector, the outputs of the 40 Gabor filters were downsampled by a factor of 16 [6], so the dimensionality of the Gabor feature vector is $39,960(40 \times 110/4 \times 150/4)$. The performance of this method is reported in the bottom row of Table 8.

### 3.5. Comparison of Feature Performances

We report the performance of different facial representations in Table 8 and Figure 2. In the case of HOG, we show results of the default implementation and two implementations that perform best for facial expression recognition after tuning of the parameters (resulting from the experiments in Section ). HOG-1 is implemented with the following properties: signed gradients with 18 bins, 16×16 pixel blocks of four 8×8 pixel cells, block spacing stride of 4 pixels, and HOG-2 is implemented with the properties: signed gradients with 18 bins, 24×24 pixel blocks of nine 8×8 pixel cells, block spacing stride of 12 pixels, gradient computed with Prewitt filters. With regards to both LBP and LTP we compare both the original implementation and overlapping approach. The performance of Gabor

---

$^2$ i.e. $9, 9\sqrt{2}, 18, 18\sqrt{2}, 36$ pixels per cycle, so the frequencies used $\approx 12, 6\sqrt{2}, 6, 3\sqrt{2}, 3$ cycles/image-width.
is reported as a reference given the extensive study in literature. Given the influence of feature length on both training time and memory requirements, we also report the length for every feature. It is clear that LBP maintains an edge in performance over other methods, especially when feature length is kept in consideration.

<table>
<thead>
<tr>
<th>Feature Length</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG</td>
<td>88.9±3.8</td>
</tr>
<tr>
<td>HOG-1</td>
<td>92.4±3.4</td>
</tr>
<tr>
<td>HOG-2</td>
<td>92.7±3.4</td>
</tr>
<tr>
<td>LBP</td>
<td>90.9±5.6</td>
</tr>
<tr>
<td>LBP-Overlap</td>
<td>92.9±5.0</td>
</tr>
<tr>
<td>LTP</td>
<td>90.9±4.9</td>
</tr>
<tr>
<td>LTP-Overlap</td>
<td>91.7±5.6</td>
</tr>
<tr>
<td>Gabor</td>
<td>89.9±6.2</td>
</tr>
</tbody>
</table>

Table 8. Recognition performance (%) of different facial representations with the linear SVM.

Figure 2. Recognition performance (%) of different facial representations with the linear SVM.

Figure 3. The effect of $\sigma = 5\%$ on randomly selected faces from several subjects.

However, the results reported here assume a perfect face registration using the manual annotation of the eye centers, which is certainly not realistic when facial expression recognition is applied to real world scenarios. In the next section we will analyze the performance of these methods when registration errors exist.

4. Recognition with Face Registration Errors

In most of the research on facial expression recognition, including the study above, the recognition performance is accessed on faces normalized based on the position of the eyes (usually manually labeled). However, in the context of a fully-automatic facial expression recognition system, the faces presented to the recognizer are not normalized perfectly due to face/eye detection errors. Therefore, it is necessary to investigate the performance of the above facial representations with face registration errors.

To simulate registration errors, Gaussian noise is added to the co-ordinates of the manually annotated eye centers. Then the noisy co-ordinates are used for face normalization. The added noise in each co-ordinate is controlled so that the standard derivation $\sigma$ of the Gaussian noise is a given percentage of the eye distance [7]. In this way, different versions of the normalized database are obtained, with the eye perturbation ranging from $\sigma = 0$ (ideal normalization) to $\sigma = 7\%$ of the eye distance. The state-of-the-art face/eye localization systems have the eye perturbation falling in the selected testing range.

Facial images of $108 \times 147$ pixels\(^3\) were cropped from original frames based on the two eyes centers. The effect of the eye perturbation on the faces is to shift, scale and rotate them in a random manner. Figure 3 shows examples of the erroneously registered faces.

4.1. Perturbation on Testing

We first consider eye perturbation in testing sets only, to investigate the impact of face registration errors on the classifiers trained on ideally registered face images. This set of experiment mimics the performance of a system trained

\(^3\)Size used by LBP/LTP, LPB/LTP-Overlap and Gabor. 112×160 pixels for HOG and HOG-1. 120×168 for HOG-2.
assumption a perfect face registration which encounters registration errors in real world scenarios. As shown in Figure 4, HOG is the most sensitive to face registration errors, followed by HOG-1 and HOG-2. In contrast, LBP-Overlap and LTP-Overlap are the most robust to face registration errors followed by LBP. The performance of most methods are stable up to 2% eye perturbation, a value optimistic for current face/eye detector for high-resolution face images. With large eye perturbations above 3%, such as those expected in face/eye detector operating on low-resolution face image, the performance of most methods start deteriorating greatly. At 7% eye perturbation, the recognition performance decreases from 90-93% to 82-84% for LBP-Overlap, LTP-Overlap, LTP, and Gabor, 77% for LBP, 75% for HOG-1 and HOG-2, and 73% for HOG.

4.2. Perturbation on Both Training and Testing

We expect that the classifier trained on the data set with similar registration errors is much more robust to face registration errors. Therefore, we consider eye perturbation in both testing and training sets. As shown in Figure 5 and Figure 6, although the recognition performance of all methods still deteriorates with eye perturbation, the performance is improved by 3-6% compared to the results in Figure 4. This is because the classifiers are trained on face images with similar eye perturbation, and therefore they can better cope with face images with registration errors. With regard to different methods, similarly LBP-Overlap, LTP-Overlap, and LTP are less sensitive to face registration errors, while HOG is the most sensitive to registration errors, followed by HOG-1 and HOG-2. The performance of most methods are also stable up to 2% RMS eye perturbation.

5. Conclusions

In this paper we have compared the performance of different local features, such as Local Binary and Ternary Patterns (LBP, LTP), Histogram of Oriented Gradients (HOG) and Gabor, with varying parameters for facial expression recognition. Furthermore, we introduce overlapping for
LBP and LTP. Especially for HOG we tested many different parameter settings and their effect on the recognition performance. Compared to other applications, different parameter settings were found to be optimal. It is thus recommended to extensively study HOG parameter settings for any new given problem. Overall LBP with overlapping gives the best performance (92.9% recognition rate on the Cohn-Kanade database), while maintaining a compact feature vector.

Another contribution is a study on the robustness of the different features against errors in the face registration. LBP with overlapping is most robust, with the overlapping consistently increasing performance by 1-2%. By taking possible registration errors into account during training better robustness can be achieved, i.e. 3-6% in recognition performance can be gained. Furthermore, the performance of LBP with overlapping is unaffected up to 4% registration error, compared to 2% originally.

Acknowledgements

We would like to thank Prof. Jeffery Cohn for granting access to the Cohn-Kanade database.

References