Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

journal homepage: www.elsevier.com/locate/jvci

# RGB-D face recognition under various conditions via 3D constrained local model[☆]

Nastaran Nourbakhsh Kaashki, Reza Safabakhsh[*]

*Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran*

A R T I C L E I N F O

*Keywords:*
3D face recognition
Depth image
Face model
3D constrained local model
Head pose
Lighting
Facial expression
Kinect
Feature descriptor

A B S T R A C T

This research proposes a method for 3D face recognition in various conditions using 3D constrained local model (CLM-Z). In this method, a combination of 2D images (RGBs) and depth images (Ds) captured by Kinect has been used. After detecting the face and smoothing the depth image, CLM-Z model has been used to model and detect the important points of the face. These points are described using Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and 3D Local Binary Patterns (3DLBP). Finally, each face is recognized by a Support Vector Machine (SVM). The challenging situations are changes of lighting, facial expression and head pose. The results on CurtinFaces and IIIT-D datasets demonstrate that the proposed method outperformed state-of-the-art methods under illumination, expression and pitch pose conditions and comparable results were obtained in other cases. Additionally, our proposed method is robust even when the training data has not been carefully collected.

## 1. Introduction

Human face recognition has been one of the earliest computer vision problems. Recently, it has received more attention due to its emerging applications in electronics such as cameras, cell phones, and in security applications. The main aspects affecting the performance of a face recognition system are image acquisition, feature representation and classification method. We focus on the last two factors in this research.

The major challenges in acquisition are variations in lighting conditions, facial expression, and head pose. A real-world face recognition system has to be insensitive to these variations in order to be effective. The conventional methods usually try to solve these challenges having a 2D image. An in-depth review on 2D face recognition methods is outside the scope of this paper. One can refer to [1,2] for good surveys on the subject.

Information about the human face in two-dimensional images may be insufficient for face recognition. Therefore, three-dimensional information has been employed for this purpose. As researchers revealed [3], combining depth and texture information leads to increasing the efficiency of face recognition. The depth information is invariant to lighting changes and image capturing conditions. Lighting conditions directly affect two-dimensional images, unlike three-dimensional ones. While the sensors for capturing depth information could be affected by lightening conditions, the 3D data is inherently invariant to illumination. In [4], for example, researchers have employed Gabor filters in order to describe the face images produced by 3D digitizers by considering pose and expression in faces. To construct a strong classifier, Linear Discriminant Analysis (LDA) and AdaBoost learning has been combined.

In spite of the good accuracy of the early 3D scanners and depth sensors, they are expensive and infeasible to use in common applications. Introducing new and cheap depth sensors like Kinect, capturing depth image (D) and 2D color image (RGB) simultaneously, can contribute to obtain RGB-D images in an easier and more affordable way. Recently, RGB-D face images have been used for face recognition [5,6,3,7–10]. In [11], RGB-D images were employed by an algorithm to recognize faces under the effect of various covariates. In this study, the employed algorithm involved the Discriminant Color Space transform with sparse coding for recognition. Additionally, researchers introduced an algorithm based on which Kinect possesses a face recognition capability [8]. Despite the low resolution of the depth images captured by Kinect, these images can be used to increase face recognition under challenging changing conditions.

In [9], a landmark and 3D reconstruction-based face recognition method has been proposed. SRC [12] has been used for classification. In this method, by detecting the pose in the probe face image and rotation of existing images in gallery to the detected pose, the authors overcome the challenge of pose changes. In [13], which is useful for gaining more information about RGB-D face recognition, the authors have presented an overview of existing RGB-D face recognition algorithms and

[☆] This paper has been recommended for acceptance by Zicheng Liu.

[*] Corresponding author.

*E-mail addresses:* nastaran_nourbakhsh@aut.ac.ir (N. Nourbakhsh Kaashki), safa@aut.ac.ir (R. Safabakhsh).

https://doi.org/10.1016/j.jvcir.2018.02.003
Received 14 December 2016; Received in revised form 8 July 2017; Accepted 3 February 2018
Available online 10 February 2018

available RGB-D face datasets.

In this paper, we propose a face recognition method designed particularly for the data obtained by low resolution RGB-D sensors such as Kinect. The contribution of this research is employing CLM-Z for 3D modeling, aligning the face images, and selecting the key points on the face image. A combination of histogram of oriented gradients (HOG) and 3D local binary patterns (3DLBP) is employed for describing the features around landmark on the intensity and depth images. The face recognition is performed by support vector machines (SVMs) as the classifier. The proposed method can effectively perform the face recognition on various head pose, facial expression, and illumination conditions.

The paper is organized as follows: Section 2 discusses the related work. The proposed method is explained in Section 3 in detail. In this section, first we briefly review the construction of 3D constrained local model (CLM-Z) (Section 3.1). Then, we introduce the calculation of patch experts (Section 3.2) and estimation of initial model for the face image (Section 3.3). The model fitting procedure (Section 3.4) and the feature extraction based on HOG, LBP, and 3DLBP is explained thereafter (Sections 3.5 and 3.6). We finish this section by discussing the method for the classification and recognition (Section 3.7). Experimental results are reported in Section 4. Finally, the paper is concluded in Section 5.

## 2. Related works

Face recognition methods, whether 2D, 3D or multimodal, can be classified into the following two main categories [1]:

### 2.1. Holistic matching algorithms

Holistic matching algorithms use the face as a whole for recognition. Based on the idea of the employing PCA for face representation [14], Turk and Pentland proposed to use this representation in human face recognition [15]. In this approach, the face images are mapped to the feature space, a.k.a facial space. The basis of this space is the set of eigenfaces constructed in the training phase. Each face image in the training set can be reconstructed by a linear combination of these eigenfaces. The number of eigenfaces is further reduced to avoid the curse of dimensionality. While PCA can be effective when a single image is available for each subject, it faces difficulties in the presence of several images with lighting variations. Since the approach in the PCA is to find a mapping that maximizes the variance in the feature space, this leads to preserving the undesired variations in the lighting. The variations in the lighting of a single subject can be usually stronger than the variations in the face of different subjects. Therefore, a subject can be misclassified with variation in the lighting. In [16], a similar method for face recognition based on the Fisher's Linear Discriminant Analysis (LDA) is proposed. The methods based on this feature space are known as fisherface methods. The eigenface and fisherface methods presume that there is an optimal mapping which maps the face images into non-overlapping regions in the feature space while reducing the data dimension. This assumption is not always held as there is the possibility of mapping the face images of different subjects to the same region.

One of the proposed workarounds to this problem is using difference of images [17]. A difference image for two face images is obtained by calculating the signed difference of intensity of corresponding pixels in the two images. The difference image obtained for the two images from same subject is called intra-personal and the difference image for two different subjects is called extra-personal. The eigenfaces constructed in PCA are based on the pairwise relation of pixels in the training set. However, there is the possibility that some useful information can be extracted by considering higher order relations, leading to better basis images. Independent Component Analysis (ICA) which is an extension of PCA is one of these methods [18].

In [19], the Basel Face Model (BFM), which is essentially a generative 3D shape and texture model, has been introduced. This model aims to recognize face under various lighting conditions and various poses and is a generalized version of the three-dimensional Morphable model (3DMM) [20]. BFM has improved the process of 3DMM construction using more advanced scanner devices. In [21], the authors proposed a modeling approach by combining Shape-From-Shading (SFS) and Local Morphable Model (LMM) which can quickly make a 3D face model.

Guo et al. [22] mapped 3D facial surface onto a 2D lattice to overcome the intrinsic complexity in representing 3D facial data. The 2D lattice is then converted to a 2D facial image. Each facial image is identified by using its sparse coefficient. Lu et al. [23], and Mohammadzade et al. [24] employed Iterative Closest Point (ICP) and its variants for matching face surfaces. Generally, ICP is highly affected by variations in illumination, pose, scale and facial expressions [25].

One of the main advantages of holistic methods is exploiting the whole image data for face recognition. However, this is also a downside of these methods as they weight the image pixels equally [26]. In addition, these methods usually have a high computation complexity. The performance of these methods is dependent on the correlation between training and test datasets and is hindered by large changes in the head pose, image scale and illumination. To improve these methods, dimension reduction tools have been employed. By using these tools, the generalization of holistic methods can be improved, but reducing the dimensions also can lead to losing some discriminative information.

### 2.2. Feature-based methods

The feature-based methods extract features from the face regions that have high discriminative capability like the eyes, mouth, and nose. These methods then compute the geometric relation between these features. Therefore, these methods essentially map the image face to a vector of geometric features. In the adaption of these method, the face features was being marked manually. In one of the early works [27], for each face image, 16 parameters were extracted which specified the distances and angles between features. The Euclidean distance is then used for the classification, leading to 75% accuracy on the test dataset. Later, more advanced methods were proposed for feature extraction [28,29]. Some of these methods employed deformable models and morphological operations. In [30], the feature points are manually set on the face image. Based on the distance of these 35 points, a feature vector with dimensionality of 30 is extracted. The performance of this method is reported to be 95% on the test dataset with 95 face images, having 685 face images in the training dataset. The early methods for automatic feature extraction did not lead to high accuracy and had high computational complexity.

The method presented in [31] is also a feature-based method which generates a graph for a face image. In this method, predefined points are selected on the face image. Each point is a node in this graph that is labeled based on the response of applying Gabor filters applied in its neighborhood. Each edge of graph is weighted by the distance of its vectors. The resulting graph is known as Face Bunch Graph (FBG). The fitting procedure starts with an initial FBG for the face image. This graph is then adapted automatically using Elastic Bunch Graph Matching (EBGM). The EBGM procedure is a coarse-to-fine approach in which the graph of a face image in manually defined. In the next step, this graph is fitted to each face image and its misalignments are manually corrected. This iterative procedure is continued until the graph fits to all image with an acceptable margin of error. To recognize a face image, its graph is compared to the FBGs available in the training set. The method has a performance of 98% on a dataset with 250 subjects. Since reaching to the optimal graph requires transforming FBG to various scales and positions over the image, this method has high computational complexity in the test phase. Another drawback of this method is its dependency on the predefined feature points.

Other models have been proposed that can automatically determine face feature points while having lower computational requirements. These methods model the appearance [32] and shape [33] of the face. The main problem with the methods based on the rigid model is that they ignore the deformability of the face; hence these methods cannot effectively model the shape. Using a deformable model can enhance the modeling and further improve the face recognition performance. The Point Distribution Model (PDM) is proposed [34]. This model aims to model the shape and its deformation using a set of points. To generate this model, the face images in the training set are labeled on their landmarks either automatically or manually. This labeling is crucial in training the model correctly. Mislabeling a feature point causes it to be placed in various positions in the training face images, preventing the model from effectively capturing the shape deformations.

The Active Shape Model (ASM) which is based on PDM is presented in [35]. To calculate the rigid and deformable model parameters, this method exploits geometrical relations between the landmarks. In [36], the ASM is employed to generate the shape for the frontal face images. As stated before, the initial estimation of the model is crucial as an inaccurate estimate makes it impossible to improve and fit the shape to the face image. Therefore, in addition to region containing the face, this study tries to improve the fitting by using the position of the eyes, nose, and lips as hints to the model fitting. One method for extending the ASM for other head poses is to train a separate ASM for each head pose. In [37], three models for left, right and frontal head pose are trained. The genetic algorithm is used to optimize the model parameters. To improve the ASM fitting, the important face regions like mouth, eyes, and nose can be determined using the Focus Of Attention (FOA) method [38]. This region is then refined using the Active Contour Model (ACM). These features are used for initially estimating the model. This model is further refined by ASM, and the Mahalanobis distance of the model landmarks in each face image is used for feature recognition.

In [39], the PDM is employed for face recognition. In this method, the frontal face image and the fitted model are exploited to generate synthetic face images for other head poses. This helps to have various head poses for the face in the training set and improves the face recognition performance for non-frontal head poses. Gabor filters are used for feature extraction from the face images.

In [40], the ASM is used for feature extraction. Using a manually labeled set of images as the training dataset, the gradient image around the neighborhood of each landmark is calculated. The mean and variance of gradient for each landmark is then calculated in the training dataset. In the model fitting phase, the gradient is calculated in the neighborhood of each landmark. A candidate point with the smallest Mahalanobis distance to the mean of gradient calculated in the training phase is chosen as the new position for that landmark. In [41], the deformable model fitting methods are studied. In this study, ASM is classified as one of the deformable model fitting methods. In addition, PDM-based models are generally known as Constrained Local Models (CLM). One of the CLM models is also abbreviated as CLM [42] which uses Subspace Constrained Mean-Shift (SCMS) for the model fitting. This study shows that SCMS outperforms previous methods in optimizing model parameters.

Recently, a Curvelet-based method relying on a multimodal keypoint detector has proposed [43]. For 3D face recognition, local surface descriptors around each landmark in facial images have been used.

The main advantage of feature-based methods is their robustness in the presence of changes in facial expression and scale since they extract the landmarks and face shape before the classification and face recognition. This is because using the key features such as the eyes for the face recognition is less prune to illumination and facial expression changes, compared to using the whole face image. In addition, by locating the landmarks and extracting the shape faces, the face images are implicitly aligned and the comparison of the face images are more accurate, compared to the holistic methods. Another advantage of the feature-based methods is compactness of feature vector which leads to faster face recognition. The main disadvantage of the feature-based methods is the need for accurately labeling the landmarks in face images for training the model. In fact, all feature-based methods require labeling the landmarks in the images of the training dataset by some means.

## 3. Proposed method

Fig. 1 shows the framework of the proposed method. Face detection,

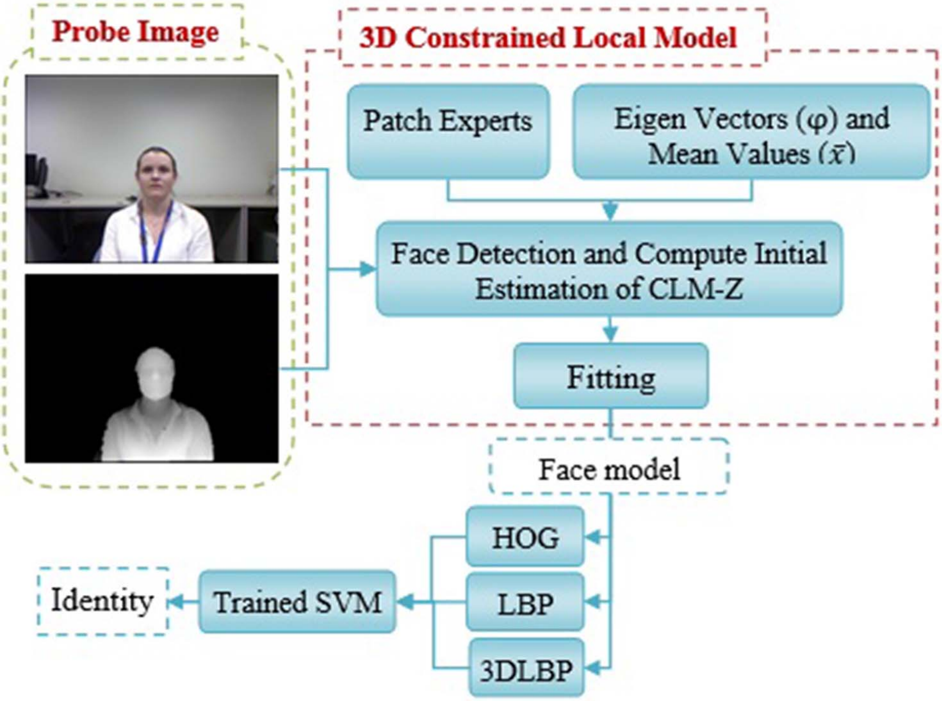

Probe Image
3D Constrained Local Model
Patch Experts
Eigen Vectors (φ) and Mean Values ($\bar{x}$)
Face Detection and Compute Initial Estimation of CLM-Z
Fitting
Face model
HOG
LBP
3DLBP
Trained SVM
Identity

**Fig. 1.** The framework of the proposed method.

3D modeling, and feature extraction are the main three parts of this framework. In the training phase, first a 3D constrained local model (CLM-Z) is calculated [44]. In the next step, the position of face in each training image is determined. The depth information is extracted from the face region and is smoothed. The 3D face model is then fitted on the image. After that, we employ Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and 3D Local Binary Patterns (3DLBP) as feature extraction tools around the landmarks in the model. These extracted features construct the training data for an SVM classifier.

We will exploit both the depth and intensity information for the face recognition. Microsoft Kinect, which is considered an inexpensive and affordable image sensor, is employed to record the information. As the feature-based methods generally outperform the global methods in face recognition, we have employed a 3D face model to determine shape of the faces and to align them. The feature descriptors we have used are invariant to lighting conditions.

In the test phase, the position of the face in the input image is first determined using a face detection method. The depth information in the corresponding region is smoothed and the trained 3D model is fitted to the face. The features around each landmark are extracted and fed to the trained SVM for recognizing face identity. We will discuss the details of each step as follows.

### 3.1. Generation of the 3D constrained local model

We have employed a 3D constrained local model [44] to model the face. This model is represented with a Point Distribution Model (PDM):

$$\mathbf{x}_i = s\cdot R(\overline{\mathbf{x}}_i + \varphi_i q) + t \tag{1}$$

where $s$ is the scaling factor, $R$ is the 3D rotation matrix, $q$ is the non-rigid deformation vector, and $t$ specifies the translation. Therefore, the model parameters are $p = [s;R;q;t]$. Here $\mathbf{x_i}$ is the average of coordinates $(x_i,y_i,z_i)$ of the $i$th point in the training set and $\mathbf{x_i}$ is coordinates of the $i$th point after applying the rigid and non-rigid transforms.

A manually labeled dataset of faces is needed to train this model. Accurate labeling is crucial to train a successful model and each landmark must be labeled at the same location in all the images. There are 66 landmark in the CLM-Z model which we have chosen for our method.

All the landmarks are represented by the a vector $\mathbf{X}$ as

$$\mathbf{X} = (x_1,\ldots,x_{66},y_1,\ldots,y_{66},z_1,\ldots,z_{66}) \tag{2}$$

where $\boldsymbol{X}$ is an instance of the 3D model, $x_i$ is the coordinates of $i$th landmark on the $x$-axis, $y_i$ is its coordinates on $y$-axis, and likewise $z_i$ is the coordinates of $i$th landmark on the $z$-axis.

To calculate the average for the model landmarks and non-rigid deformations, the 3D face dataset BU-4DFE [45] is used. The average coordinate of landmarks can be calculated after manual labeling:

$$\overline{\mathbf{x}}_{\mathbf{i}} = \frac{1}{M}\sum_{j=1}^{M}\mathbf{x_{ij}} \tag{3}$$

where $M$ is the number of training images.

The deformations resemble how the landmarks displace with respect to each other. These deformations model the non-grid transformations and are calculated by applying PCA to the average deviation. Therefore, for each shape in the training set, we calculate its difference to the average:

$$\mathrm{d}\mathbf{X}_{\mathrm{j}} = \mathbf{X}_{\mathrm{j}} - \overline{\mathbf{X}} \tag{4}$$

in which $\mathbf{X_j}$ is shape vector for the $i$th sample, $\mathbf{X}$ is their average, and $dX_j$ is the deviation of the $i$th shape from the average. The covariance is calculated by

$$S = \frac{1}{M}\sum_{j=1}^{M}\mathrm{d}\mathbf{X}_{\mathrm{j}}\mathrm{d}\mathbf{X}_{\mathrm{j}}^{\mathrm{T}} \tag{5}$$

The covariance matrix resembles the distribution of non-rigid deformations in the training set. The eigenvalues and eigenvectors of the covariance matrix is calculated. The eigenvectors show the direction of deformations. Specifically, the $k$th eigenvector moves the $i$th landmark along with the vector $(dx_{ik},dy_{ik},dz_{ik})$ where $(dx_{ik},dy_{ik},dz_{ik})$ is the $i$th row of the $k$th vector. Generally, the main contribution to the deformation are made by a few of the eigenvectors. We select a subset of $\mathbf{T}$ eigenvectors with the largest eigenvalues. The sum of their eigenvalues is indeed a fraction of eigenvalues total sum. This fraction is the desired accuracy in the deformation. In the BU-4DFE dataset, the value of 27 is chosen for $\mathbf{T}$.

The model is constructed by having the average and eigenvectors calculated from the training samples. By varying the model parameters, face samples can be generated from the model. An extreme value of q can lead to a distorted face sample. To prevent this situation, it has been suggested to pick its value from $-3\sqrt{\lambda_i} \leqslant q_i \leqslant 3\sqrt{\lambda_i}$ where $\lambda_i$ is the $i$th eigenvalue of covariance matrix.

### 3.2. Computation of patch experts

In addition to the shape average deformation eigenvectors, patch experts should be computed for shape alignment. The patch experts are employed to calculate the patch response in the neighborhood of each point. The response of the $i$th patch is the probability of alignment for the $i$th landmark in the candidate positions. These probabilities are used to derive the values of parameters for the fitting model using the maximum a posteriori probability (MAP):

$$p(\mathbf{p}|\{l_i = 1\}_{i=1}^{n},\mathrm{I},\mathrm{Z}) \propto p(\mathbf{p})\prod_{i=1}^{n} p(l_i = 1|\mathbf{x}_i,\mathrm{I},\mathrm{Z}) \tag{6}$$

where $\mathbf{p}$ gives the model parameters, $l_i \in \{-1,1\}$ is a discrete variable indicating the alignment (aligned/misaligned) of the $i$th point, $p(\mathbf{p})$ is prior probability for the model parameters, and $\prod_{i=i}^{n} p(l_i = 1|x_i,I,Z)$ is the joint probability of alignment of all the model points.

A training set consisting of intensity and depth images is used for training the patch experts. In this paper, we have employed the Multi-PIE dataset [46] for the intensity images and the 4D-BUFE dataset for the depth images. As in the model training phase, the landmarks on the images are manually labeled. However, in this case, some images have aligned landmarks while the landmarks in the other images are misaligned. Therefore, the point alignment problem becomes a binary classification problem. A subset of the dataset is used as the training data for the linear SVM classifier. For each landmark and each angle, an SVM classifier is trained to distinguish the aligned candidate points from the misaligned candidate points. The angles accounted for the horizontal rotation of head are $\{0,\pm15,\pm30,\pm45,\pm60,\pm75\}$ and the angles for vertical rotation are $\{0,\pm30,\pm60\}$. Separate SVM classifiers are trained for intensity and depth images:

$$\mathscr{C}_{\mathrm{I,i}}(\mathrm{x_i};\mathrm{I}) = \mathrm{w}_{\mathrm{I,i}}^{\mathrm{T}}\mathscr{P}(\mathscr{W}(\mathrm{x_i};\mathrm{I})) + \mathrm{b}_{\mathrm{I,i}} \tag{7}$$

$$\mathscr{C}_{\mathrm{Z,i}}(\mathrm{x_i};\mathrm{Z}) = \mathrm{w}_{\mathrm{Z,i}}^{\mathrm{T}}\mathscr{P}(\mathscr{W}(\mathrm{x_i};\mathrm{Z})) + \mathrm{b}_{\mathrm{Z,i}} \tag{8}$$

where $C_{I,i}$ and $C_{Z,i}$ are the outputs of intensity and depth classifiers for the $i$th landmark respectively, $\{w_{I,i}^{T},b_{I,i}\}$ and $\{w_{Z,i}^{T},b_{Z,i}\}$ are their weight matrix and bias; $P(c)$ is a function which normalizes $c$ to have mean zero and variance 1; $W(x_i;I)$ and $W(x_i;Z)$ are vectorized versions of the candidate points around landmark $i$. The set of candidate points around the $i$th landmark is called an image patch.

Since we need the output of the SVM classifiers as the probability of alignment, we apply a logistic function on their output for mapping them to the probabilities. To learn the parameter of these logistic functions, we use a separate subset of the training dataset. First, we apply the calculated weights of SVMs to the all samples in this subset. Then we use output of SVM to derive the logistic approximation functions:

$$p(l_i|\mathbf{x}_{\mathrm{i}},\mathrm{I}) = \frac{1}{1 + e^{-(\beta_0+\beta_1\mathscr{C}_{\mathrm{I},i}(\mathbf{x}_i;\mathrm{I}))}} \tag{9}$$

$$p(l_i|\mathbf{x}_{\mathrm{i}},\mathrm{Z}) = \frac{1}{1 + e^{-(\beta_0+\beta_1\mathscr{C}_{\mathrm{Z},i}(\mathbf{x}_i;\mathrm{Z}))}} \tag{10}$$

in which $\beta_0$ is the $y$-intercept of linear regression equation and $\beta_1$ is the regression coefficient. The patch experts are computed for each landmark and both for intensity and depth images. The patch experts are employed later in the model fitting stage.

### 3.3. Face detection, smoothing, and initial model estimation

The face detection is a crucial step in determining the initial position and scale of the face. A poor face detection will lead to failure of CLM-Z in fitting the face model. Since the intensity-based face detection methods generally outperform depth-based methods, we perform the face detection on the intensity images using Viola-Jones method. The intensity and depth images captured by the Kinect sensor are aligned; hence the position of the face found in the intensity image will be the same in the depth image.

The Kinect sensor sets the value of pixels with unknown depth to zero. To resolve this issue, we perform smoothing in the neighborhood of these pixels. Specifically, we apply a median filter on each pixel with zero depth value in the region which the face is detected. The depth value of pixels in the sub-image is then linearly scaled to [0,255]. Fig. 2 depicts the effect of this filter on a depth image.

The first step in the model fitting process is to have an initial estimation of the position and scale of the model for the detected face. For this purpose, we set $s = 1$, $R$ to an identity matrix, $t = 0$, and $q$ to zero in Eq. (1) to get an instance of model. This 3D instance is $\overline{\mathbf{X}}$, the average of landmarks. Now the ratio of 3D instance size to the detected face in the image gives us an estimate of scale parameter $s$. Likewise, the displacement in the detected face and 3D model instance is an estimate of translation $t$. Having an initial estimate of the model parameters, they are further refined during model fitting process which is discussed in the following.

### 3.4. Model fitting

An adaptive two-stage fitting procedure is commonly used for locally constrained models. In this procedure, first the response patch is calculated for each landmark. The optimization is then performed using the response patches. For the adaption stage, here we utilize the Subspace Constrained Mean-Shifts (SCMS) [47]. Regarding Eq. (6), the goal is to compute the optimal model parameters. The final position of model landmarks are determined by using these optimized parameters.

In Fig. 3, the current position of landmark $i$ is $x_i^c$. The parameters should be adjusted such that $x_i^c$ reach its best position. As some information on the model deformation is lost by applying PCA, there will be some error in the landmarks. This error is determined by the candidate points. Therefore, we can rewrite Eq. (6) as

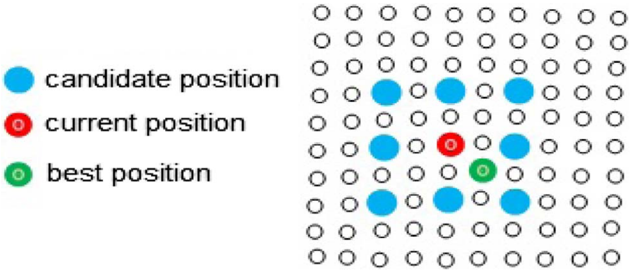


**Fig. 3.** An example of landmark and its candidate points.

$$p(\mathbf{p}|\{\mathrm{l_i} = 1\}_{\mathrm{i}=1}^{\mathrm{n}},I,Z) \propto p(\mathbf{p}) \prod_{i=1}^{\mathrm{n}} \sum_{y_i\in\psi_i} \pi_{y_i}\mathscr{N}(\mathbf{x}_i;\mathbf{y}_{\mathrm{i}},\rho\mathbf{I}) \tag{11}$$

where $\pi_{y_i} = p(l_i = 1|y_i,I,Z)$ is the probability of alignment for the $i$th landmark in the candidate position $y_i$ in the intensity and depth images; $N(x_i;y_i,rI)$ is the alignment error for the $i$th landmark in candidate position $y_i$; $p(\mathbf{p})$ is assumed to have a uniform distribution.

The Taylor expansion of Eq. (1) is utilized to compute the parameters

$$\mathbf{x}_i \approx \mathbf{x}_i^c + \mathbf{J}_i\Delta\mathbf{p} \tag{12}$$

in which $J$ is Jacobian matrix. This Jacobian matrix is the matrix of partial derivatives of CLM-Z model with respect to its parameters. $\Delta\mathbf{p}$ is the adjustment to the parameters to obtain the best position. The optimal parameters can be computed by:

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} \tag{13}$$

$\Delta\mathbf{p}$ can be computed from Eq. (12):

$$\Delta\mathbf{p} = \mathbf{J}^{\dagger}\mathrm{v} \tag{14}$$

where $v = [v_1;...;v_n]$ is concatenation of mean-shift vectors of landmarks. $v_i$ is the difference between the best position among candidate points and, $x_i^c$, is current position of $i$th landmark:

$$\mathrm{v}_i = \left(\sum_{\mathbf{y}_i\in\psi_i} \frac{\pi_{y_i}\mathscr{N}(\mathbf{x}_i^c;\mathbf{y}_i,\rho\mathbf{I})}{\sum_{\mathbf{z}_i\in\psi_i}\pi_{z_i}\mathscr{N}(\mathbf{x}_i^c;\mathbf{z}_i,\rho\mathbf{I})}\mathbf{y}_i\right) - \mathbf{x}_i^c \tag{15}$$

In (15) the best position is assumed to be the weighted average of candidate points. Its difference to the current position gives the mean-shift vector. Substituting it in Eqs. (14) and (12), the new parameter values are obtained. This procedure is iterated until the model converges. To calculate $\pi_{y_i}$, the patch experts are employed in this procedure. For each landmark, we have a set of SVMs and their corresponding logistic regressor and the $R$ angle determines which one is applicant for the current model orientation. The selected SVM and

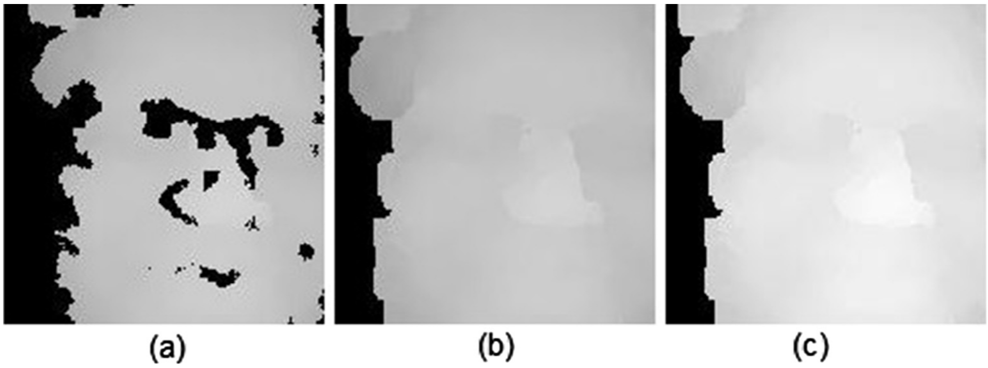


**Fig. 2.** Smoothing a depth image from IIIT-D dataset. (a) The depth image from the Kinect cropped by face region. (b) The depth image after applying the smoothing (c) image (b) after scaling.

```
The model fitting algorithm
0.  Input: the intensity and depth images, the average and eigenvectors of covariance matrix, patch experts consisting of the SVM weights and coefficients of the logistic regressor).
1.  Set the loop counter to zero
2.  Face detection is performed on the intensity image. The face region in the depth image is smoothed.
3.  A 3D sample from the model is generated by setting s=1, R=I, q=0, and t=0
4.  Initial estimation of CLM-Z model: The s and t parameters are estimated using the position and ratio of 3D sample with respect to the detected face. The model is created using equation (1) and stored in currModel.
5.  An estimation of model angle is computed using R and stored in currAngle
6.  $oldAngle \leftarrow currAngle$
7.  $oldModel \leftarrow currModel$
8.  Computing patch experts: Repeat the steps 8-1 and 8-2 for each landmark
    8-1-  Using currModel, a set of candidate points are selected around $i$th landmark.
    8-2-  Based on the angle determined by $R$, the corresponding SVM and logistic regressor is selected. Using the equation (9) and (10) the response of candidate patch in the intensity and depth images are computed and their average is stored in $p_{y_i}$
9.  The Jacobian matrix $\mathbf{J}$ and mean-shifts vector matrix $\mathbf{v}$ are computed by (15)
10. Using equation (14), the $\Delta\mathbf{p}$ is computed.
11. The model parameters are updated according to (13)
12. A sample from the model is created by substituting new parameter values in (1) and stored in currModel.
13. Check the model convergence in the current angle: if the distance between currModel and oldModel exceeds a predefined threshold, go to step 7.
14. The model angle is estimated based on matrix $R$ and is stored in currAngle.
15. Check the termination criteria: if the angle of model is changed ($currAngle \neq oldAngle$), go to step 6.
16. Output: Optimal parameters of the CLM-Z model for input face.
```

**Fig. 4.** The model fitting algorithm.

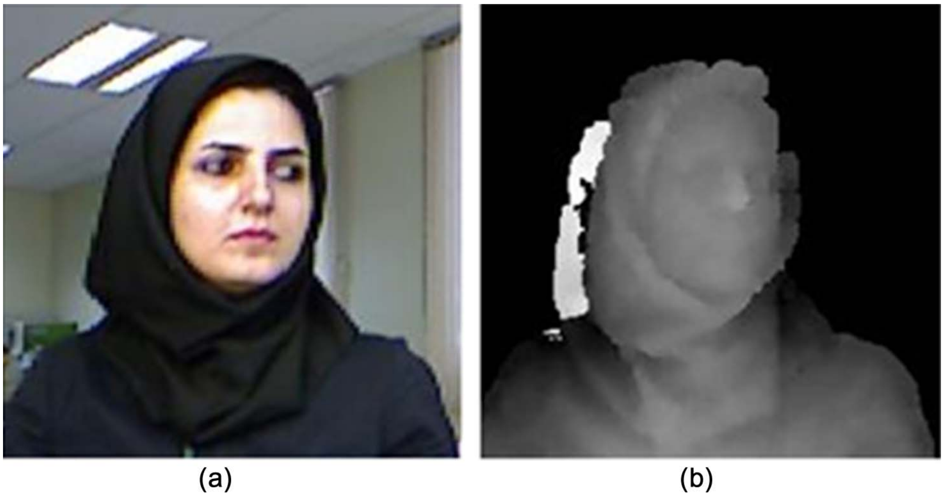


**Fig. 5.** A sample face image captured by Kinect sensor. (a) The color image, (b) The depth image.

logistic regressor are employed by the mean-shift vectors for the optimization. The procedure of the model fitting is depicted in Fig. 4. A sample color image and its corresponding depth image is shown in Fig. 5. The model configuration during model fitting is shown on the 2D image and the 3D space in Figs. 6 and 7 respectively.

### 3.5. Feature extraction using Histogram of Gradients (HOG)

Histogram of Gradients is a feature descriptor for digital images which counts the occurrence of gradients in the neighborhood of a pixel. The philosophy behind this descriptor is that the appearance of an object can be described by its gradient distribution of intensities and the edge orientations. In [48] the procedure for computing HOG is presented. The first step in computing HOG is to calculate gradient values for the input image. The trivial method for calculating gradients is applying 1D derivative vectors along horizontal and vertical directions. These vectors are

$$K_v = [-1,0,1]^T \tag{16}$$

$$K_H = [-1,0,1] \tag{17}$$

It has been suggested that applying smoothing filters like the Gaussian smoothing filter reduces the performance of the descriptor [48].

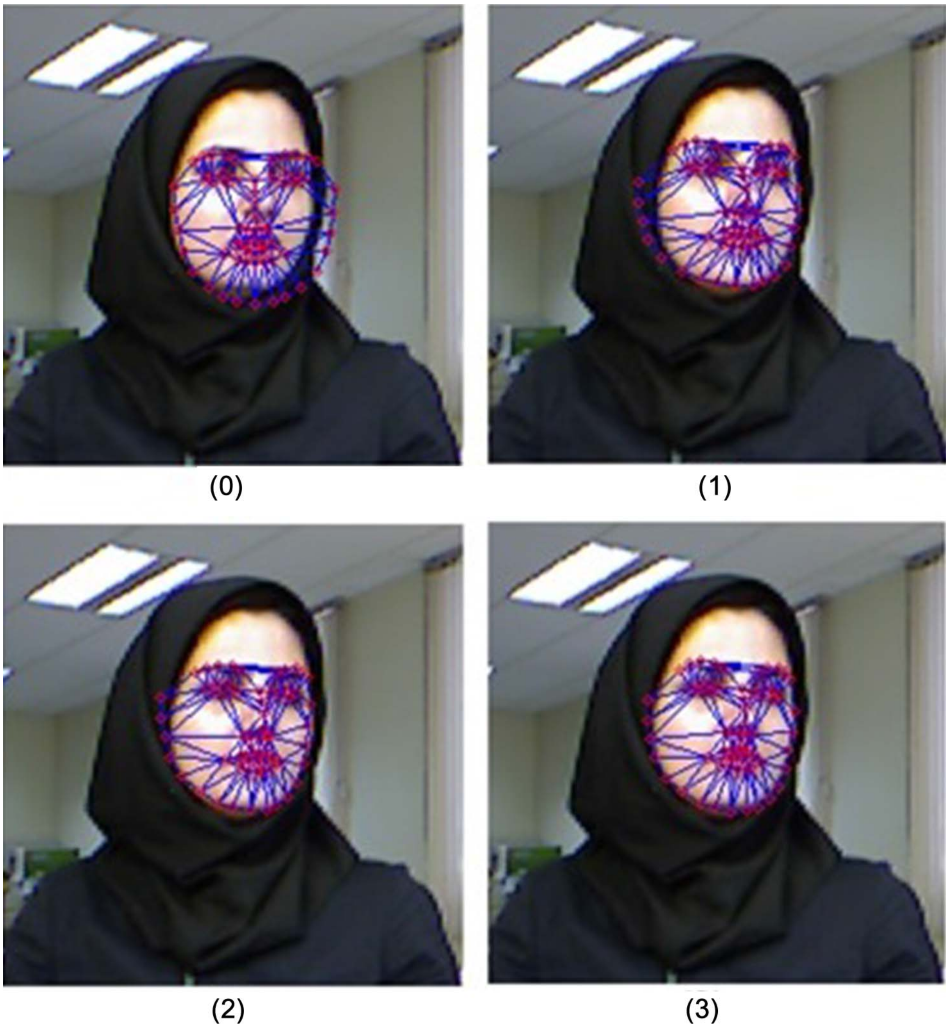


**Fig. 6.** Fitting the CLM-Z model to the face image. (0) initial estimation of model based on the position and size of the detected face. (1) Model fitting after optimization with initial angle of zero. The final angle for this iteration is −45. (2) Model fitting after optimization with initial angle of −45. The final angle for this iteration is −15. (3) Model fitting after optimization with initial angle of −15. The final angle for this iteration is also −15 hence the optimization is terminated.

For the second step, the image is partitioned into non-overlapping cells. These cells can be square or circular. In each cell, the weighted direction of histogram is computed. Specifically, for each pixel in the cell, a weight is assigned that is the value (or a function) of gradient at that pixel. The image is then divided into blocks, each block containing several cells and the blocks can share the cells. To make the descriptor more robust to lighting variations, the histogram of cells is normalized in each block. The *L-2* norm with clipping is utilized for normalization. This normalization is known as *L2-Hys*:

$$L2{-}norm\colon\ v = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon^2}} \tag{18}$$

$$v = \begin{cases} 0.2 & v > 0.2 \\ v & otherwise \end{cases} \tag{19}$$

Firstly, the *L-2* norm is calculated for the block using Eq. (18) and the feature vector $v$ which is the concatenation of cell histograms for that block is obtained. The $v$ values are then clipped using Eq. (19). Finally, the *L-2* norm of the clipped values is computed to get the final normalized vector. Experiments [48] have shown that utilizing *L2-Hys* has a better performance compared to other normalization methods and the value of $\varepsilon$ does not have a significant effect on the performance.

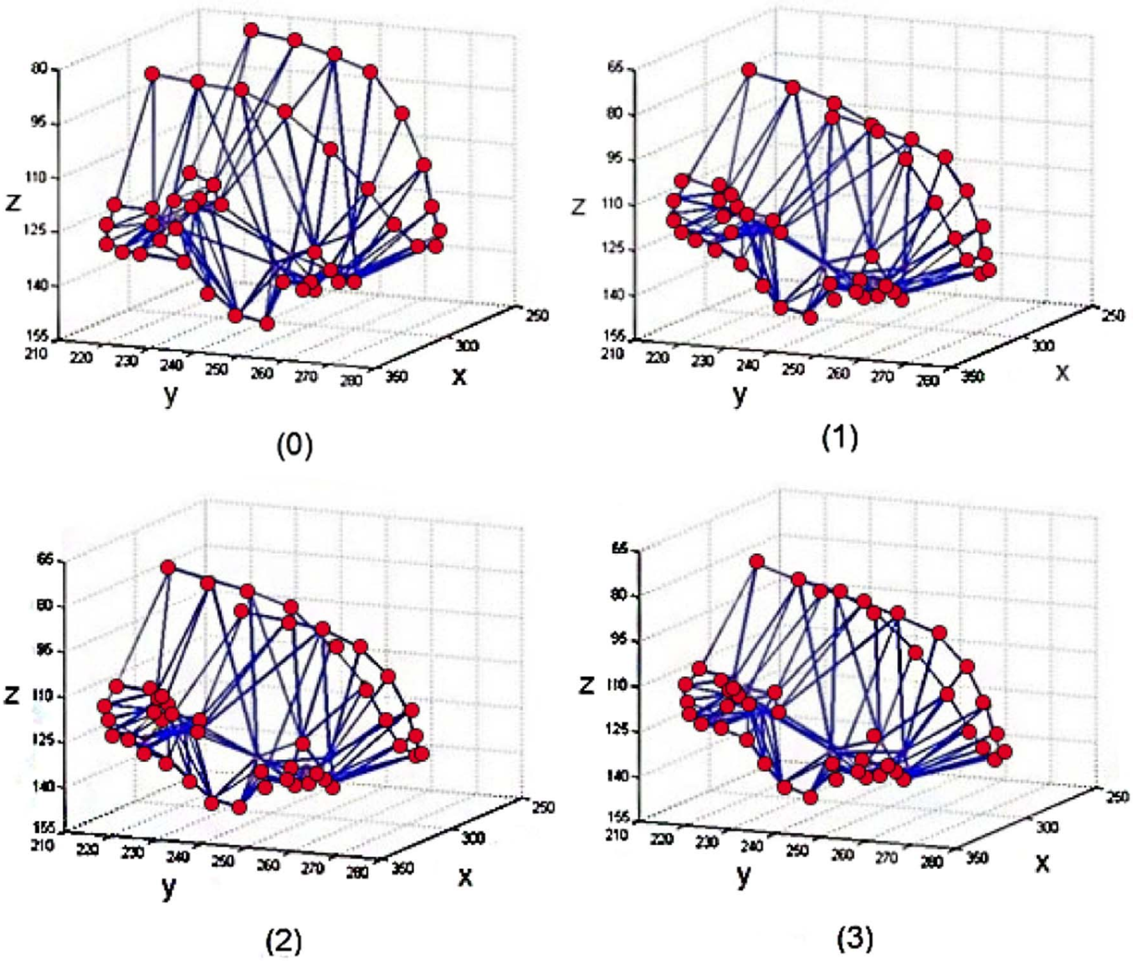
A cell can belong to multiple blocks and is normalized in each block differently. This leads to redundancy in the final feature vector, but improves the performance. The resulting feature vectors from the blocks are concatenated to construct the histogram of oriented gradients feature vector.

In this paper, HOG is utilized to describe the face features around the model landmarks. For this goal, a block is placed around each landmark and partitioned into cells. Therefore, the size of neighborhood around each landmark is the number of pixels in each cell multiplied by the number of cells in each block. The face feature vector is obtained by concatenating the feature vector of the block of each landmark. Fig. 8 illustrates the result of this method on the intensity and depth images.

### 3.6. Feature extraction using Local Binary Patterns (LBP) and 3D Local Binary Patterns (3DLBP)

The Local Binary Patterns (LBP) descriptor is based on the difference between the value of a pixel and the value of its neighbor pixels [49]. For each pixel, a neighborhood centered on that pixel is considered and the value of each pixel in the neighborhood is subtracted from the origin pixel. If this difference is negative, the value of zero is placed in the binary pattern; otherwise, a one is placed in the binary pattern. Finally, this binary pattern is converted to a decimal number which represents the local binary pattern of that pixel.

After applying the LBP operator on all pixels and obtaining their binary pattern, the histogram of these numbers presents the LBP feature vector for that image.



(0)

(1)

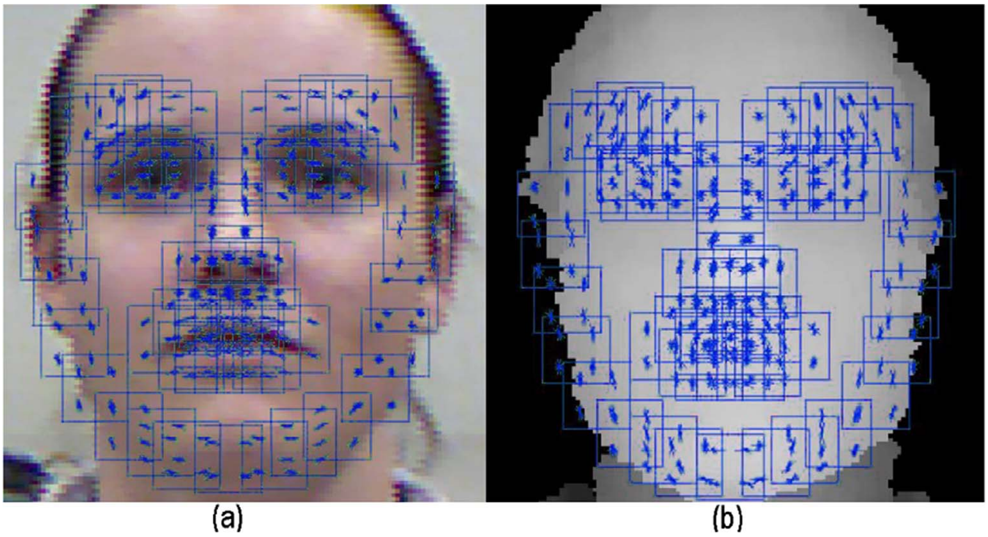
(2)

(3)

**Fig. 7.** Fitting the CLM-Z model in the 3D space. (0) Preliminary estimation of model based on the position and size of the detected face. (1) Model fitting after optimization with initial angle of zero. The final angle for this iteration is −45. (2) Model fitting after optimization with initial angle of −45. The final angle for this iteration is −15. (3) Model fitting after optimization with initial angle of −15. The final angle for this iteration is also −15 hence the optimization is terminated.

LBP is a powerful tool to represent the texture in the image. However, this feature descriptor ignores the magnitude of difference between the pixel values. This magnitude can have important information about the face, especially on the depth images. For instance, the tip of the nose has a small depth and its LBP value will be zero in all the face images. However, the depth difference between the tip of the nose and its neighborhood varies for different faces.

A generalization of LBP which preserve the magnitude of difference is 3D Local Binary Pattern (3DLBP) [49]. This method keeps sign and magnitude of difference between pixels. The authors have shown that statistically, more than 93% of differences in neighborhood of 2 pixels are below 7. The reason is that the depth image of face is generally



(a)

(b)

**Fig. 8.** An example of applying HOG around landmarks. The blocks centered on the landmarks are shown by green squares. The distribution of oriented gradients in each cell is shown by asterisks. The number of arms for an asterisk indicates the number of intervals for the histogram of oriented gradients. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

smooth. This is not true for the intensity images where the value of pixels can have large variations due to color, texture, and lighting conditions. Therefore, 3DLBP is utilized for the depth images and three bits are used to represent the magnitude of the difference between pixel values. These three bits can encode the difference of zero to 7. A difference greater than 7 is also encoded by 7. For the sign and magnitude of difference between two pixels, 4 bits are needed, $\{i_1,i_2,i_3,i_4\}$. $i_1$ is the sign and $i_2i_3i_4$ is the magnitude of difference. This concepts can be formulated as

$$i_1 = \begin{cases} 1 & if\ \ DD \geqslant 0 \\ 0 & otherwise \end{cases} \tag{20}$$

$$|DD| = i_2*2^2 + i_3*2^1 + i_4*2^0 \tag{21}$$

where $DD$ is the difference between the value of the central pixel and a pixel in the neighborhood. These four bits are divided into four layers. The bits of neighbor pixels in each layer construct an LBP code. Therefore, for each pixel four LBP codes are generated where the code from the first layer is the standard LBP code. These four codes are called 3D Local Binary Pattern (3DLBP).

By applying 3DLBP to an image, four LBP images are obtained. The histogram of each image is calculated independently and the final feature vector is constructed by concatenating the histograms. In Fig. 9, the four LBP images extracted for a smoothed depth image is shown.

As it can be seen, the feature image of layer 4 that corresponds to the least significant bit of the magnitude of the difference contains the details of the depth image while the feature image of layer 1 contains the general outline of the depth image. The final feature vector of 3DLBP is the combination of histogram of these four images. Therefore, the general depth information is preserved.

After obtaining the optimal parameters of the CLM-Z model for the input face image, the face model is obtained. Using this model, the location of landmarks on the input image is determined. In the next step, the feature descriptors are applied on these landmarks to obtain suitable features. For this purpose, LBP is applied to the intensity image, 3DLBP is applied to the depth image, and HOG is applied to the both intensity and depth images. The vector is normalized to a mean of zero and a variance of one and then combined to build the final feature vector.

### 3.7. Classification

After obtaining the feature vector for a face image, the face image is recognized using a multi-class classifier. The classifiers like artificial neural networks and Support Vector Machines can be employed for this classification. The classifier must be robust to the large number of classes, have low computation complexity, and lead to a reasonable accuracy. We used a SVM classifier for this purpose. SVM classifiers can generate non-linear decision boundaries and can be extended to be used in multi-class problems. Unlike artificial neural networks, their computation complexity does not depend on the dimension of the input vector. They are also less susceptible to overfitting.

The choice of SVM kernel and its parameters has a significant effect on its performance. In this study, we used a polynomial kernel. The value of parameters for this kernel is determined by a grid search using training samples. In the face recognition problem, the face feature vector is an input sample and the face identity is the sample label. Thus, the number of classes is equal to the number of persons present in the dataset. The trained SVM is used to determine the face identity of a test image.

## 4. Experimental results

In this section we will evaluate the proposed method. We use two face datasets for the evaluation. First, these datasets will be introduced. Next, the experiments and the results will be presented. We conclude this section by analyzing the results.

### 4.1. CurtinFaces dataset

CurtinFaces is a dataset of more than 5000 pair of intensity and depth images [50]. These images are captured from 52 individuals with the Kinect sensor under various illuminations, facial expressions, poses, and obstacles (e.g. wearing sunglasses). We use a subset of this dataset with variations in pose, illumination, and facial expression. In some images of this subset, the person is wearing glasses. Fig. 10 shows an example of color images in this dataset and Fig. 11 shows its corresponding depth image. The depth images in this dataset has a 12-bit resolution for each pixel.

For capturing the images in this dataset, the distance between the Kinect sensor and the person was around 70 cm and the distance between Kinect sensor and wall was around 3 m. To represent the distance of each point with millimeter precision, 12 bits are needed. In Fig. 12, the distance between 700 mm and 1500 mm is mapped to 8 bits.

#### 4.1.1. Selection of training images

We select 18 captured images for each subject as the training samples. In each image, only one of the three variations (illumination, pose or expression) is present. Notably, using multiple training images is practically feasible in the case of Kinect, since it can instantly obtain RGB-D data at 30 frames per second. The training images with the fitted model for a subject are presented in Fig. 12.

#### 4.1.2. Selection of training images for evaluating robustness against variations in illumination and facial expression

For this experiment, 30 images are chosen for each person. An example of selected images for an individual is given in Fig. 13. The images have 5 different illuminations. For each illumination, six facial expressions consisting of laughing, disgust, anger, sadness, surprise, and fear is selected. By choosing these images, we can evaluate the robustness of the proposed method against simultaneous variation in facial expression and illumination.

#### 4.1.3. Selection of training images for evaluating robustness against variations in head pose and facial expression

For this experiment, 30 non-frontal images are chosen for each

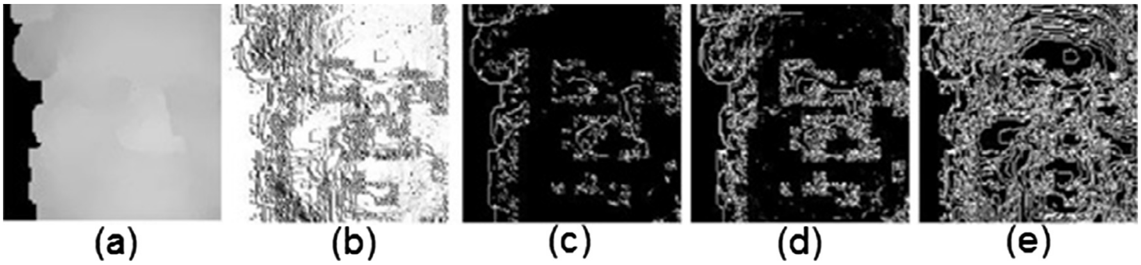


**Fig. 9.** An example of applying 3DLBP on a depth image. (a) The smoothed depth image. (b) The feature image of layer 1 (standard LBP). (c) The feature image of layer 2. (d) The feature image of layer 3. (e) The feature image of layer 4.

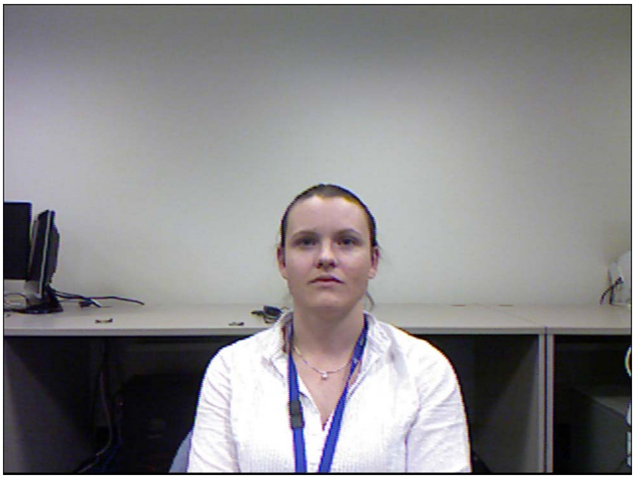


**Fig. 10.** An intensity RGB image from CurtinFaces dataset.

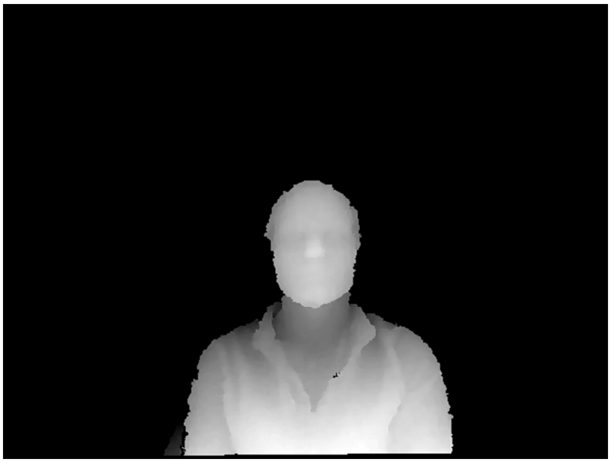


**Fig. 11.** The depth image from CurtinFaces scaled to 8 bit.

subject. An example of selected images for an individual is given in Fig. 14. Five different head poses are considered. For each head pose, six facial expression consisting of laughing, disgust, anger, sadness, surprise, and fear are selected. By choosing these images, we can evaluate the robustness of the proposed method against simultaneous variations in head pose and illumination.

### 4.2. IIIT-D dataset

We also evaluate our proposed method using IIIT-D dataset [6]. This dataset contains color and depth images for 106 individuals. The images are captured by the Kinect sensor. The number of captured images for each person varies between 11 and 254. The total number of images is 4056. Variations in head pose and facial expression are present in the images. Fig. 15 shows an example of IIIT-D images.

#### 4.2.1. Training/test dataset

In the experiments, we use a method similar to K-fold cross-validation with K = 5. In 5-fold cross-validation, the training dataset is partitioned randomly into 5 equal-sized subsets. In each run, one of the subsets is selected for the testing and the other four subsets are used for training. Therefore, five experiments is performed and the overall performance of the method is calculated by the averaging the performance over these five experiments. In the experiments, we also perform five experiments. In each experiment four images are randomly selected for a subject as the training samples and the other images of the subject are treated as the test images. The average performance in these five experiments gives the overall performance.

### 4.3. Evaluating the effect of HOG parameters on the performance

In this experiment, we only apply HOG feature descriptor around landmarks of the fitted model to extract the feature vector. We evaluate the effect of HOG parameters on the performance and try to find their optimal value. These parameters are the cell size (in pixels), the block size (in cells), and the number of histogram bins which must be determined depending on the application.

#### 4.3.1. Evaluating the effect of cell size and block size on the performance

To evaluate the effect of cell and block size, we vary these

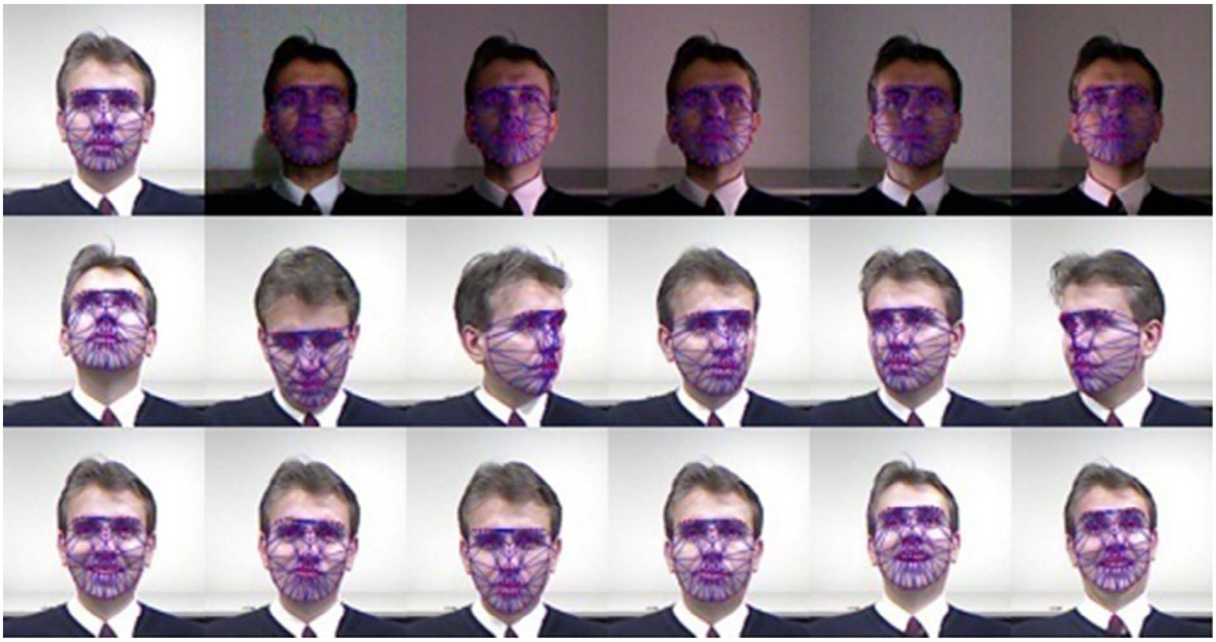


**Fig. 12.** The sample images for a subject from CurtinFaces dataset overlaid by the fitted CLM-Z model.

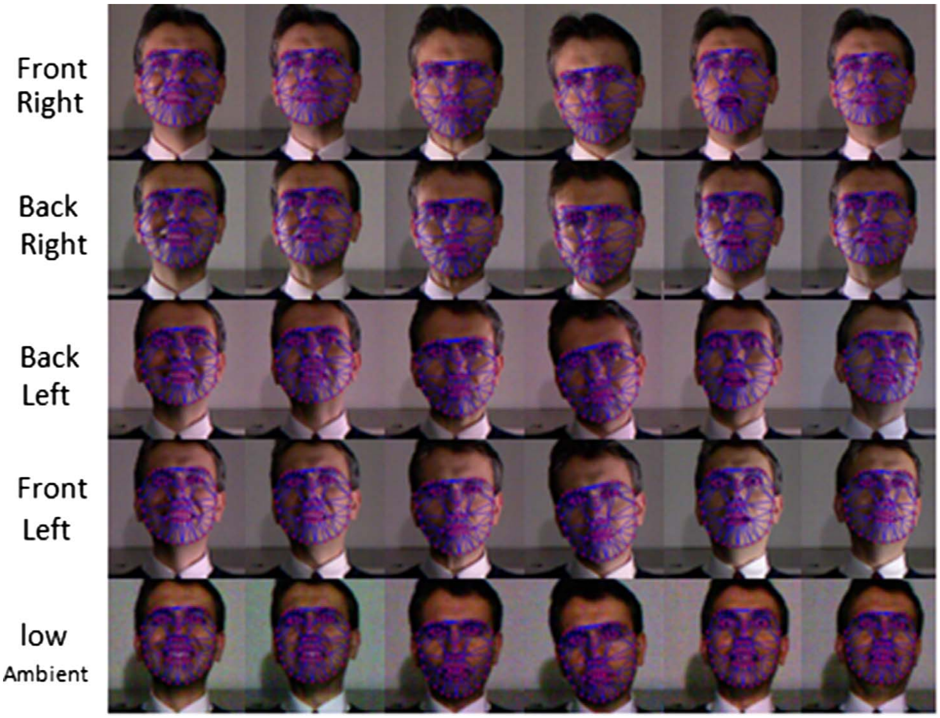


**Fig. 13.** An example of selected images with variation in illumination and facial expression for an individual from CurtainFaces dataset, overlaid by fitted CLM-Z model.

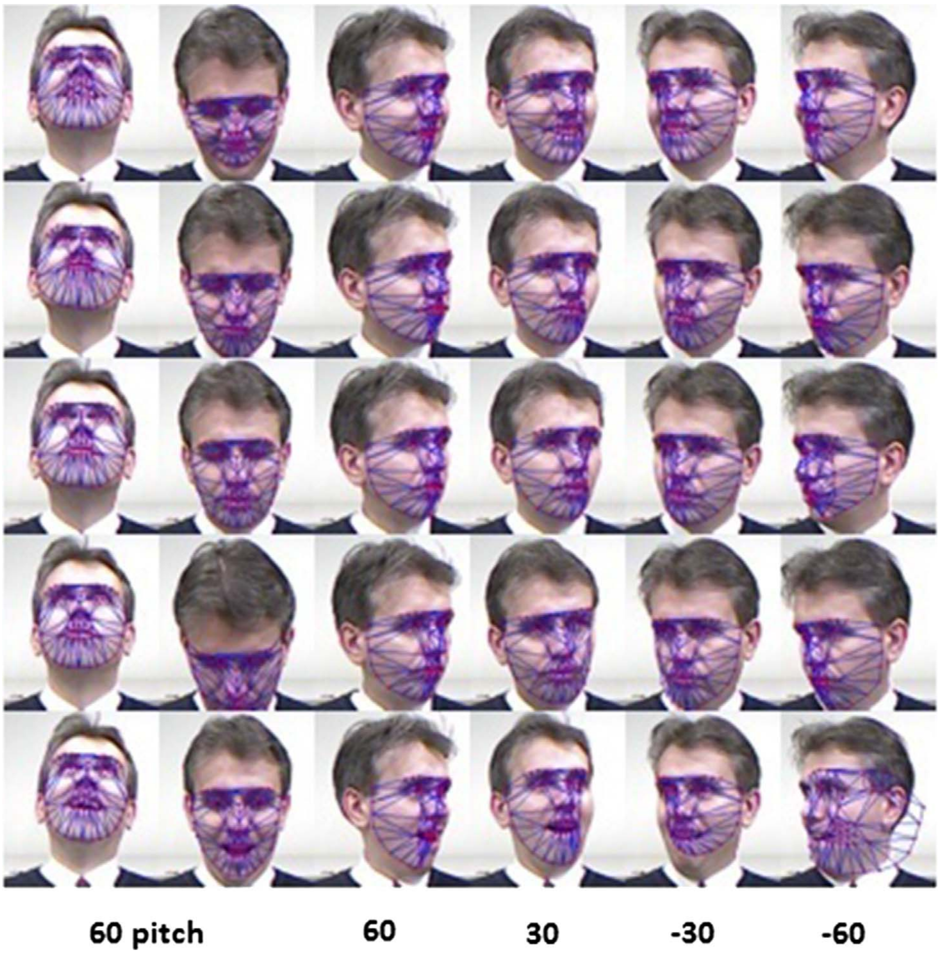


**Fig. 14.** An example of selected images with variation in head pose and facial expression for an individual from CurtainFaces dataset, overlaid by fitted CLM-Z model.

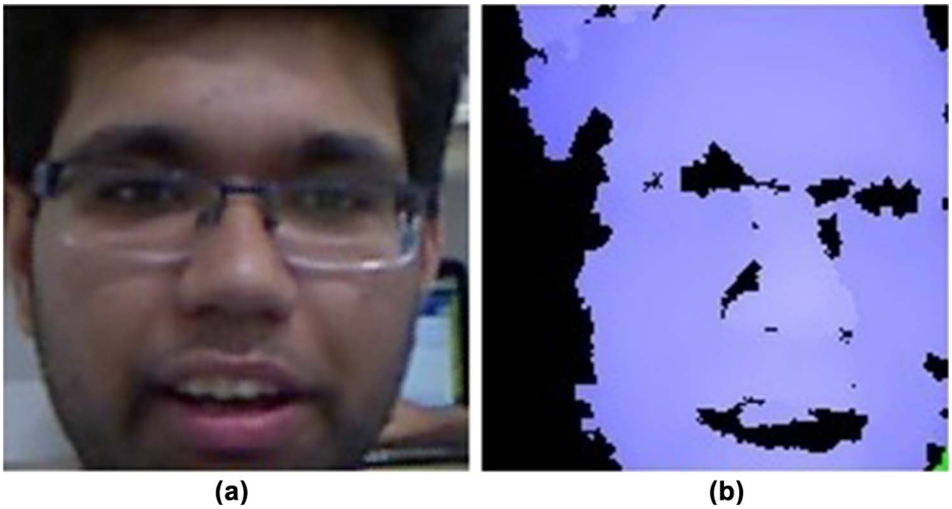


**Fig. 15.** (a) An example of IIT-D images. (a) Color image. (b) Depth image.

**Table 1**
HOG parameters for evaluating the effect of block/cell size.

| Parameter | Value |
|---|---|
| Gradient calculation | [−1,0,1] and $[-1,0,1]^T$ without Gaussian smoothing |
| Cell size (in pixels) | 6 × 6, 8 × 8, 10 × 10, 12 × 12, and 16 × 16 |
| Block size (in cells) | 1 × 1, 2 × 2, 3 × 3, 4 × 4 |
| Number of histogram bins | 9 |
| Orientations of histogram | 0–180° |
| Neighborhood | Rectangular |
| Normalization method | L2-hys |

parameters while the other parameters are fixed. The values are given in Table 1. We perform the experiment using the CurtinFaces dataset. The face recognition accuracy for different cell and block sizes in presence of variation in facial expression and illumination is depicted in Fig. 16. Furthermore, Fig. 17 shows the face recognition performance for different cell and block sizes in presence of variation in head pose and facial expression.

Based on the results presented Fig. 16, the recognition performance on the intensity images increases for the larger cells and blocks. It should be noted that block size multiplied by cell size is basically the area of neighborhood around a landmark. Thus, by increasing the neighborhood area in the RGB image, more discriminative features are extracted from the face which lead to higher recognition performance. In the depth images, however, this relation between neighborhood size and recognition performance is not in effect. In the depth images, an excessively large neighborhood causes increase in unrelated features and loss of useful information for face recognition. This contrast is because of intrinsic difference between intensity and depth information. This is also can be observed in Fig. 17.

Considering the results presented in Figs. 16 and 17, the optimal block and cell sizes are 3 × 3 and 16 × 16 in the RGB images, respectively; and 2 × 2 and 10 × 10 in the depth images, respectively. The results show that these values are robust to variations in illumination, head pose, and facial expression.

#### 4.3.2. Evaluating the effect of number of histogram bins

We evaluate the effect of number of histogram bins on the performance. We vary it while the other HOG parameters are fixed. The values for HOG parameters in this experiment are shown in Table 2. These parameters are set based on the Dalal study [48]. This experiment is performed on the CurtinFaces dataset. Fig. 18 depicts the face recognition performance with respect to number of histogram bins in the presence of variations in illumination and facial expression.

Based on Fig. 18, a small number of bins leads to a poor performance due to loss of histogram of gradients information. On the other hand, increasing the number of bins beyond 9 does not have significant effect on the performance, while it increases the size of the feature vector. Therefore, we choose 9 bins for the histogram to have a tradeoff between the performance and size of the feature vector.

The plots in Fig. 18 shows the performance in different lighting conditions for both intensity and depth images. As we can see, the number of histogram bins has the similar effect in all these conditions. We can conclude that the optimal number of bins is independent of illumination. This is due to calculating the gradient and normalization in HOG which make the calculated features invariant to the illumination.

### 4.4. Evaluating the effect of LBP parameters on the performance

In this experiment, we only apply LBP feature descriptor around landmarks of the fitted model to extract the feature vector. We evaluate the effect of LBP parameters on the performance and try to find their optimal value. These parameters are the neighborhood size around each landmark and the number of histogram bins which must be determined depending on the application.

#### 4.4.1. Evaluating the effect of neighborhood size

In this experiment, we fix the number of histogram bins and vary neighborhood size to evaluate its effect on the performance. The parameter values are presented in Table 3. This experiment is performed on the CurtinFaces dataset.

Fig. 19 shows the face recognition performance versus neighborhood size in the presence of variation in illumination and facial expression. Based on the results, the recognition performance on the color images improves as the neighborhood size increases, up to a neighborhood size of around 17. Further increase of the neighborhood size causes a drop in the performance. On the other hand, the recognition performance for the depth images rapidly increases as the neighborhood size increases, up to the neighborhood size of 41.

This dissimilarity can be explained by the intrinsic difference between intensity and depth images. The LBP feature descriptor encodes the difference in pixel values in the defined neighborhood. In the depth images, the pixel values also represent their depth distance to a reference plane. Consequently, by increasing the neighborhood size, the LBP descriptor can deduce a feature vector that represents the depth information more accurately. However, in the color images a pixel value represents its color and calculating their difference in farther neighborhood does not lead to a better feature vector for them.

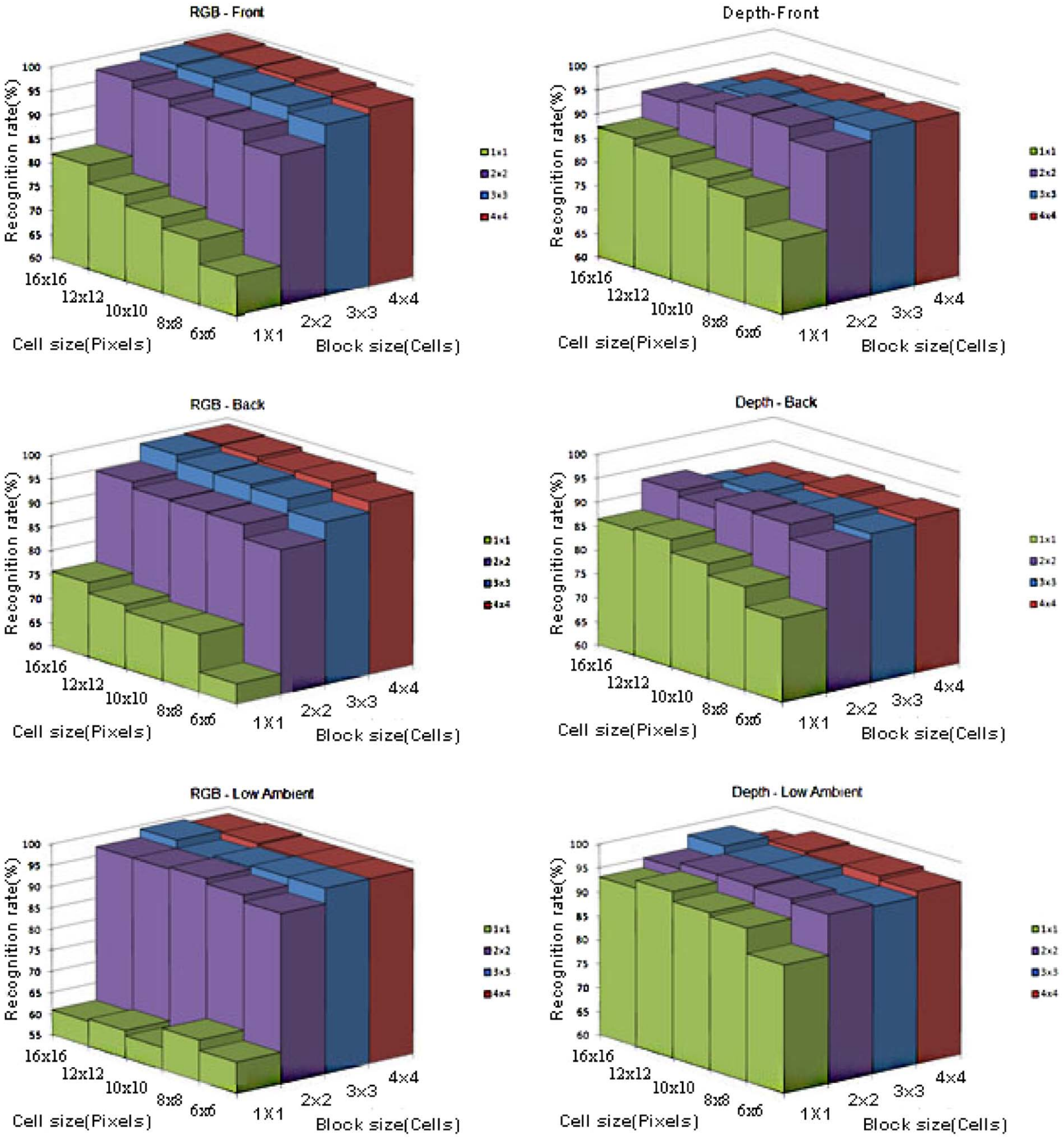


**Fig. 16.** The performance of face recognition in the presence of variation in illumination and facial expression versus different cell/block sizes for HOG.

Therefore, as it can be observed in Fig. 19, an excessively large neighborhood size can hinder the recognition. This is in contrast to HOG feature descriptor in which a larger neighborhood size (i.e. cell size and block size) improved the recognition performance on the color images. In a nutshell, HOG is a feature descriptor that encodes the edges and orientation of gradients while LBP encodes difference of pixel values in a neighborhood. Thus, a larger neighborhood size in the color images is desirable for HOG feature descriptor while causes information loss in the depth images. Likewise, increasing the neighborhood size improves the LBP performance on the depth images while hindering it on the color images.

#### 4.4.2. Evaluating the effect of number of histogram bins

We evaluate the effect of number of histogram bins in LBP. We vary it while keeping the other LBP parameters fixed. The values for the LBP parameters are shown in Table 4. This experiment is performed on the CurtinFaces dataset. Fig. 20 plots the face recognition performance versus the number of histogram bins in the presence of variations in illumination and facial expression.

Fig. 20 shows that a low number of histogram bins leads to a poor performance as the facial information is lost. On the other hand, having more than 24 histogram bins does not have a significant effect on the performance. Therefore, we choose 24 bins for histograms to have a compact feature vector without compromising the recognition performance.

### 4.5. Evaluating the performance under variations in illumination and facial expression

We have used CurtinFaces dataset to evaluate the performance of the proposed method in the presence of illumination and facial expression variations. The parameter values for HOG and LBP methods are presented in Tables 5 and 6, respectively.

Table 7 shows the overall recognition performance for the proposed

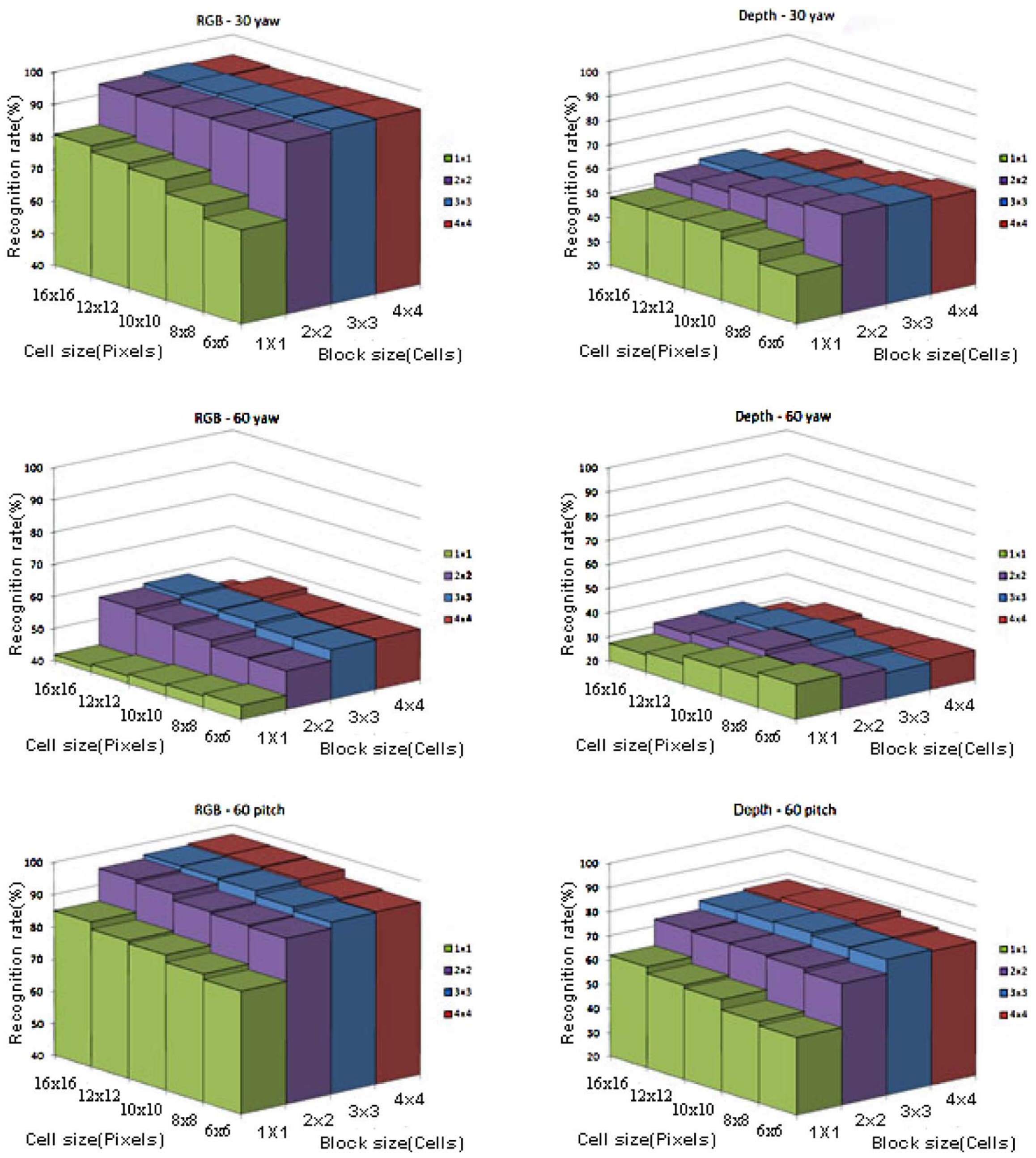


**Fig. 17.** The performance of face recognition in the presence of variation in head pose and facial expression versus different cell/block sizes for HOG.

**Table 2**
HOG parameters for evaluating the effect of number of histogram bins.

| Parameter | Value |
|---|---|
| Gradient calculation | [−1,0,1] and $[-1,0,1]^T$ without Gaussian smoothing |
| Cell size (in pixels) | 8 × 8 |
| Block size (in cells) | 2 × 2 |
| Number of histogram bins | 3, 6, 9, 12, and 32 |
| Orientations of histogram | 0–180° |
| Neighborhood | Rectangular |
| Normalization method | L2-hys |

method, for different cases. For readability, a number is assigned to each method. The best performance in each case is highlighted with gray color.

The first rows of Table 7 shows the performance of the Li et al. method [3]. This method exploits intensity and depth images and use a combination of symmetric filling and sparse coding for face recognition. The results for this method show that using both texture and depth information can improve the performance, particularly in a low illumination environment (from 91.3% to 97.1%). This is caused by the information loss in RGB images due to low illumination, while the depth images are not affected by variations in illumination. Moreover, the symmetric filling has improved the overall performance of the method.

Based on the results presented in Section 4.4, the optimal block size and cell size for RGB images are assumed to be 3 × 3 and 16 × 16 respectively and 2 × 2 and 10 × 10 for the depth images. The number of histogram bins are set to 9 for HOG and 24 for LBP.

In Table 7, method 2 uses CLM-Z model and the HOG feature descriptor with parameter values optimized for the RGB images. Method 3 is similar to method 2 with the difference that it is uses the optimal parameter values corresponding to the depth images. By comparing the results of these two methods, once more we can observe that a larger

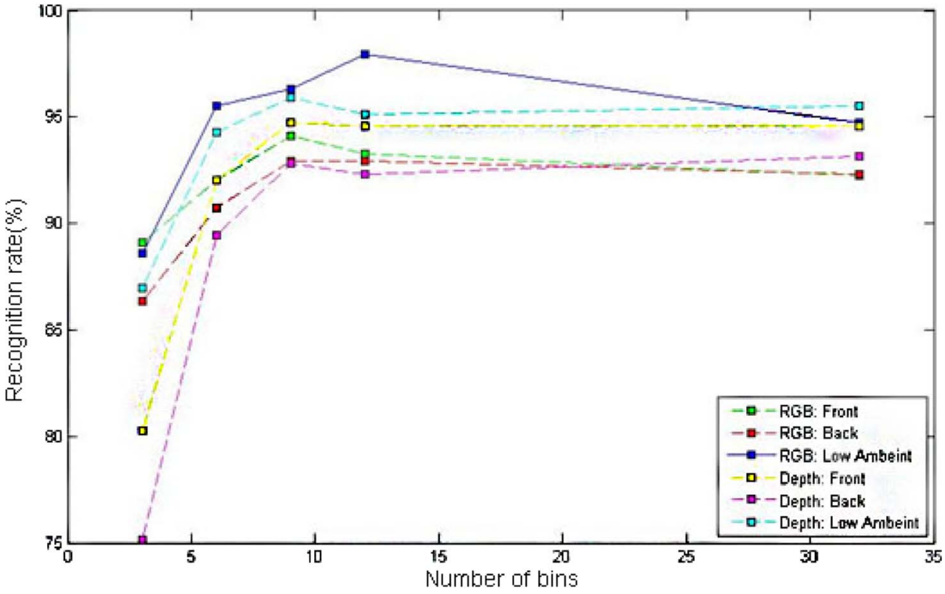


**Fig. 18.** The performance of face recognition in the presence of variation in illumination and facial expression versus the number of histogram bins for HOG.

**Table 3**
The value of LBP parameters for evaluating the effect of neighborhood size.

| Parameter | Value |
|---|---|
| Neighborhood type | Rectangular |
| Number of histogram bins | 32 |
| Neighborhood size around each model landmark | 5 × 5, 7 × 7, 9 × 9, 11 × 11, 15 × 15, 17 × 17, 19 × 19, 23 × 23, 25 × 25, 27 × 27, 31 × 31, 35 × 35, 41 × 41, 57 × 57, 71 × 71 |

**Table 4**
The value of LBP parameters for evaluating the effect of number of histogram bins.

| Parameter | Value |
|---|---|
| Neighborhood type | 3 × 3 |
| Number of histogram bins | 3, 6, 9, 12, 21, 24, 32, 39, 48 |
| Neighborhood size around each model landmark | 15 × 5 for color images<br>41 × 41 for the depth images |

neighborhood size around each landmark can improve the performance for HOG features (from average of 95.1% to 98.4%) on the intensity images. On the other hand, the performance on the depth images is decreased (from average of 95.1% to 93.2%) because an excessively large neighborhood size causes loss in the depth details.

In method 4, the optimal block size and cell size is chosen for the depth and intensity images. Compared to the case that the block size and cell size is the same for depth and RGB images, the performance has been improved (from an average of 98% and 96.8% to 98.7%).

Method 5 is the result of using the CLM-Z model and LBP feature descriptor with the optimal parameter values as feature extraction method. Generally, using LBP feature descriptor has given lower performance compared to the case that HOG feature descriptor is employed for feature extraction. LBP only retains the sign of difference between two pixel values and disregard its magnitude. However, this magnitude can contain significant information in some cases. For instance, as the tip of the nose has the lowest depth in its neighborhood, its LBP value will be the same (zero) in all depth images. Therefore, some discriminative information that can assist in face recognition will be lost.

Nevertheless, LBP still can obtain useful information about the face for RGB images. Therefore, we apply it to the RGB images in method 6. For the depth images, we use 3DLBP feature descriptor to obtain more

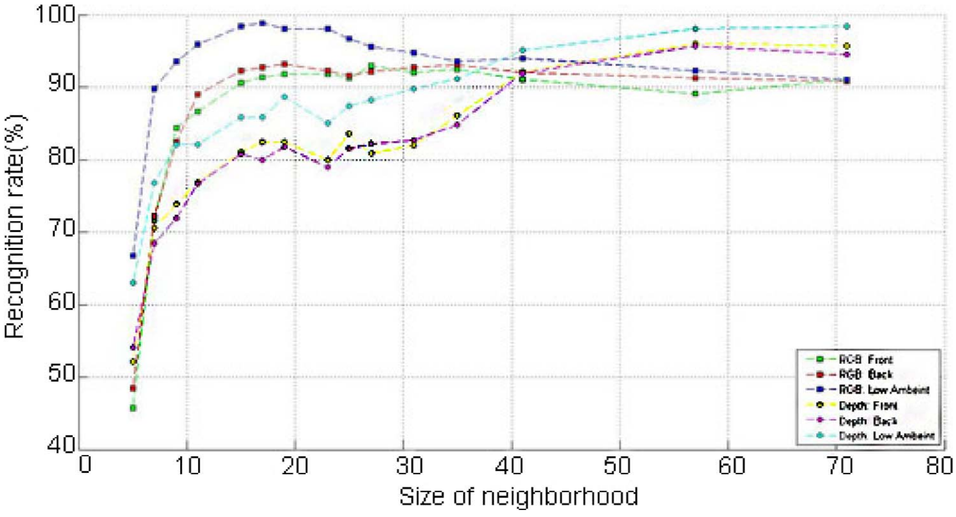


**Fig. 19.** The face recognition performance with respect to neighborhood size in LBP.

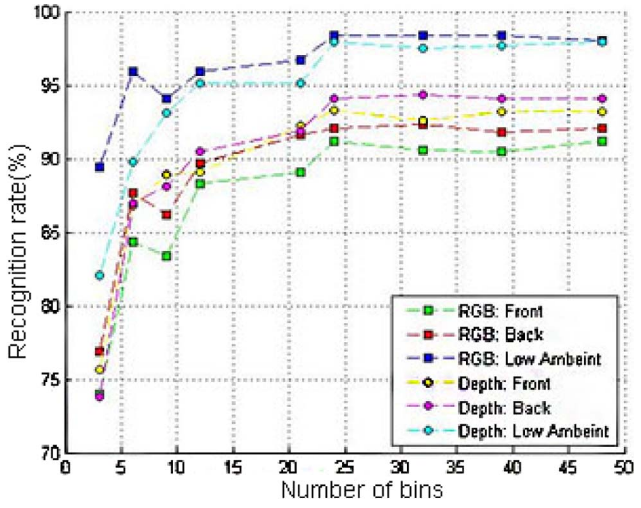


**Fig. 20.** The face recognition performance with respect to number of histogram bins in LBP.

**Table 5**
The optimal value for HOG parameters, based on previous experiments.

| Parameter | Value |
|---|---|
| Gradient calculation | [−1,0,1] and $[-1,0,1]^T$ without Gaussian smoothing |
| Cell size (in pixels) | 10 × 10, 16 × 16 |
| Block size (in cells) | 2 × 2, 3 × 3 |
| Number of histogram bins | 9 |
| Orientations of histogram | 0–180° |
| Neighborhood | Rectangular |
| Normalization method | L2-hys |

**Table 6**
The optimal value for LBP parameters, based on previous experiments.

| Parameter | Value |
|---|---|
| Neighborhood type | 3 × 3 Matrix |
| Number of histogram bins | 24 |
| Neighborhood size around each model landmark | 15 × 15 for the intensity images<br>41 × 41 for the depth images |

descriptive features. 3DLBP preserves the magnitude of difference between the pixel value and its neighborhood pixels, leading to features that help to discriminate between face images more accurately. The results for this method presented in Table 7 show that using 3DLBP has improved the recognition performance on the depth images (from an average of 95.1% to 96.1%).

Method 7 is using CLM-Z model with both HOG and LBP feature descriptors. Method 8 is the method proposed in this paper. In this method, the CLM-Z model is employed for finding landmarks on the detected face image. For feature extraction, HOG is applied to both intensity and depth images, LBP is applied on the RGB images, and 3DLBP feature descriptor is applied to the depth images. The concatenation of extracted features forms the final feature vector. Based on the results presented in Table 7, this method has higher performance over Li et al. method [3] in all illumination conditions. There are three points in support of this improvement: firstly, using the CLM-Z model in the early stages helps to focus on the areas (i.e. neighborhood of landmarks) on the face image that contain essential information and discard the irrelevant areas of face image. By using CLM-Z, the face images are also aligned and consequently, they can be compared more accurately. Secondly, by using HOG as the feature descriptor on the intensity images, the method becomes invariant to illumination variations because of calculating oriented gradients and normalization. The third point is the use of LBP and 3DLBP feature descriptors which are invariant to changes in the illumination and facial expression.

For the sake of thoroughness, in method 9 CLM-Z is employed with Gabor filters as the feature descriptor. Gabor filters are a set of linear filters typically used for edge detection. We generated Gabor filters in 8 directions and 5 scales, resulting in 40 Gabor filters in a filter bank. These filters are applied to neighbor pixels around each landmark of CLM-Z model and their combination gives the final feature vector for the face. A neighborhood of 15 × 15 pixels is set. As Gabor filters produce a high dimensional feature vector, PCA is used for dimension reduction. Based on the results in Table 7, the Gabor filter is more sensitive to illumination and facial expression changes compared to HOG and LBP; noticeably in the depth images.

To assess the role of modeling in the overall recognition performance, methods 2, 3, and 4 in the absence of CLM-Z modeling stage are evaluated on the test images and their results are reported as methods 10, 11, and 12, respectively. Comparing these results, we conclude that CLM-Z modeling has a crucial role in improving the face recognition performance (from the average of 95.6% to 99.5%). This can be explained by the role of CLM-Z in aligning the face images.

### 4.6. Evaluating the performance under variations in head pose and facial expression

We have used CurtinFaces dataset to evaluate the performance of the proposed method in the presence of head pose and facial expression variations. The parameter values for HOG and LBP methods are the same as those in previous section, presented in Tables 5 and 6, respectively. Table 8 shows the overall recognition performance for the proposed method, accompanied by the other cases. The best performance in each case is highlighted with gray color.

In Table 8, method 1 shows the results for the Li et al. method [3]. As we have stated, this method exploits RGB and depth images and uses a combination of symmetric filling and sparse coding for face recognition. The results for this method show that symmetric filling has significantly improved the performance, particularly in the depth images. This is due to the substantial changes in the depth images under different head poses.

Similarly, the optimal block size and cell size for the intensity images are assumed to be 3 × 3 and 16 × 16, respectively; and 2 × 2 and 10 × 10 for the depth images. The number of histogram bins is set to 9.

As before, method 2 is the method in which the CLM-Z model and HOG feature descriptor with optimal parameter values are used on the intensity images. Method 3 is similar to Method 2, except that it uses depth images. Comparison of these two methods' results shows again that a larger neighborhood size improves the performance of HOG (from an average of 85.1% to 87.1%) on the intensity images while decreasing its performance on the depth images (from an average of 66.3% to 66.2%).

Method 5 is the result of using the CLM-Z model and LBP feature descriptor as feature extraction method. These results also shows that the LBP feature descriptor leads to lower performance compared to HOG. Consequently, 3DLBP is employed in method 6 on the depth images. While 3DLBP improves the performance (from an average of 66.7% to 69.7%), the performance on the depth image is still inferior to the performance on the intensity images. A significantly low performance on depth images hinders the overall performance on the intensity and depth images. For this reason, method 7 combines CLM-Z model with HOG for RGB and depth images and LBP for intensity images.

Method 8 is the proposed method in this paper. As we have stated before, this method employs the CLM-Z model to detect landmarks and uses HOG on both RGB and depth images, LBP on the intensity images, and 3DLBP feature descriptor on images. By comparing the result of this

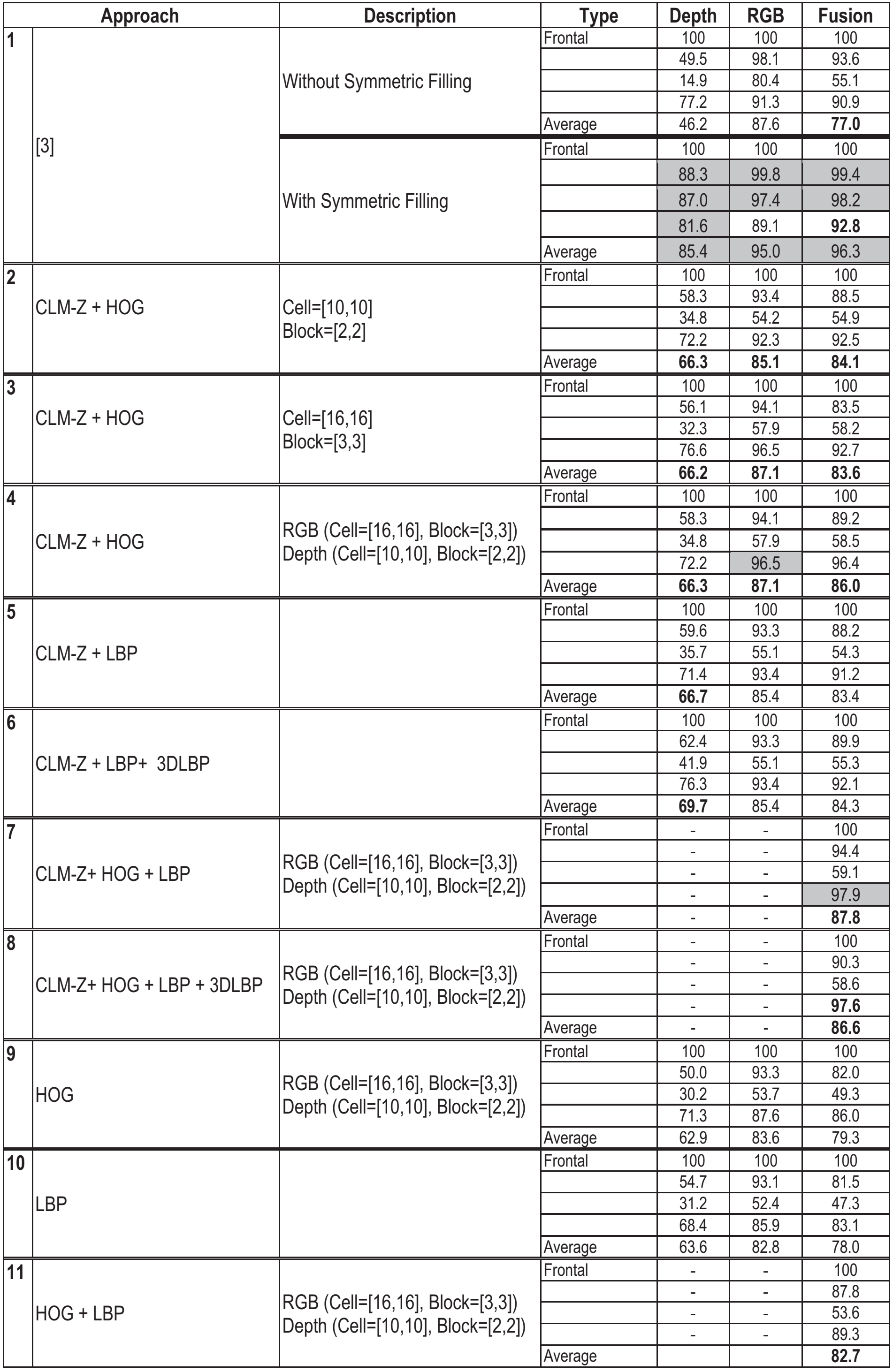
**Table 7**
The face recognition performance in presence of illumination and facial expression variation.

| Approach | | Description | Type | Depth | RGB | Fusion |
|---|---|---|---|---|---|---|
| 1 | [3] | Without Symmetric Filling | Front | 89.1 | 96.8 | 98.4 |
| | | | Back | 89.4 | 96.6 | 97.6 |
| | | | Low Ambient | 87.2 | 91.0 | 95.8 |
| | | | Average | 88.8 | 95.6 | 97.6 |
| | | With Symmetric Filling | Front | 92.5 | 97.1 | 98.9 |
| | | | Back | 93.8 | 96.5 | 98.6 |
| | | | Low Ambient | **91.3** | 91.0 | **97.1** |
| | | | Average | 92.8 | 95.6 | 98.4 |
| 2 | CLM-Z + HOG | Cell=[10,10]<br>Block=[2,2] | Front | 95.0 | 94.5 | 97.7 |
| | | | Back | 93.9 | 93.2 | 97.6 |
| | | | Low Ambient | 96.3 | 97.6 | 98.8 |
| | | | Average | **95.1** | **95.1** | **98.0** |
| 3 | CLM-Z + HOG | Cell=[16,16]<br>Block=[3,3] | Front | 91.6 | 99.2 | 96.4 |
| | | | Back | 90.8 | 98.5 | 96.5 |
| | | | Low Ambient | 97.2 | 97.6 | 97.6 |
| | | | Average | **93.2** | **98.4** | **96.8** |
| 4 | CLM-Z + HOG | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Front | 95.0 | 99.2 | 98.5 |
| | | | Back | 93.9 | 98.5 | 98.5 |
| | | | Low Ambient | 96.3 | 97.6 | 99.2 |
| | | | Average | 95.1 | 98.4 | **98.7** |
| 5 | CLM-Z + LBP | | Front | 93.3 | 91.2 | 93.5 |
| | | | Back | 94.1 | 92.1 | 94.7 |
| | | | Low Ambient | 97.9 | 98.4 | 98.8 |
| | | | Average | **95.1** | 93.9 | 95.6 |
| 6 | CLM-Z + LBP + 3DLBP | | Front | 94.5 | 91.2 | 95.3 |
| | | | Back | 95.3 | 92.1 | 95.8 |
| | | | Low Ambient | 98.7 | 98.4 | 99.3 |
| | | | Average | **96.1** | 93.9 | 96.8 |
| 7 | CLM-Z+ HOG + LBP | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Front | - | - | 99.4 |
| | | | Back | - | - | 99.3 |
| | | | Low Ambient | - | - | 99.6 |
| | | | Average | - | - | **99.4** |
| 8 | CLM-Z+ HOG + LBP + 3DLBP | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Front | - | - | 99.5 |
| | | | Back | - | - | 99.3 |
| | | | Low Ambient | - | - | 99.7 |
| | | | Average | - | - | **99.5** |
| 9 | CLM-Z + Gabor | | Front | 76.5 | 89.9 | 87.1 |
| | | | Back | 70.1 | 82.9 | 77.4 |
| | | | Low Ambient | 70.7 | 84.6 | 81.4 |
| | | | Average | 72.4 | 85.8 | 82.0 |
| 10 | HOG | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Front | 92.3 | 92.5 | 93.9 |
| | | | Back | 91.3 | 92.9 | 93.9 |
| | | | Low Ambient | 95.1 | 92.7 | 95.5 |
| | | | Average | 92.9 | 92.7 | 94.4 |
| 11 | LBP | | Front | 90.5 | 88.2 | 89.1 |
| | | | Back | 91.4 | 90.7 | 90.6 |
| | | | Low Ambient | 95.7 | 93.9 | 94.2 |
| | | | Average | 92.5 | 90.9 | 91.3 |
| 12 | HOG + LBP | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Front | - | - | 95.2 |
| | | | Back | - | - | 94.8 |
| | | | Low Ambient | - | - | 96.6 |
| | | | Average | - | - | **95.6** |

method to method 7, we see that adding 3DLBP feature vectors extracted from depth images to the final feature vector decreases the performance (from an average of 87.8% to 86.6%). This underlines the significant loss of depth information in the presence of head pose variation.

The proposed method outperforms method 1 when symmetric filling is not used (from an average of 77% to 86.6%). However, it underperforms method 1 with symmetric filling when the yaw of the head pose varies (horizontal movement). This can be explained by the lack of some area of the face image when the head pose is not full frontal. Despite the fact that CLM-Z still locates the key points on the face images, and this improves the performance compared to other methods, it cannot reconstruct the missing areas of the face image. Symmetric filling, however, can construct an estimate of those areas by exploiting the symmetric nature of the face image. Nevertheless, the symmetric filling is not effective when there are variations in the pitch of the head pose (vertical movement) while CLM-Z still can model the face image. This explains the superiority of proposed method to method 1 in the case of pitch variations in the head pose (average performance recognition of 97.6% compared to 92.8%).

**Table 8**
The face recognition performance in presence of head pose and facial expression variation.



| Approach | | Description | Type | Depth | RGB | Fusion |
|---|---|---|---|---|---|---|
| 1 | [3] | Without Symmetric Filling | Frontal | 100 | 100 | 100 |
| | | | | 49.5 | 98.1 | 93.6 |
| | | | | 14.9 | 80.4 | 55.1 |
| | | | | 77.2 | 91.3 | 90.9 |
| | | | Average | 46.2 | 87.6 | **77.0** |
| | | With Symmetric Filling | Frontal | 100 | 100 | 100 |
| | | | | 88.3 | 99.8 | 99.4 |
| | | | | 87.0 | 97.4 | 98.2 |
| | | | | 81.6 | 89.1 | **92.8** |
| | | | Average | 85.4 | 95.0 | 96.3 |
| 2 | CLM-Z + HOG | Cell=[10,10]<br>Block=[2,2] | Frontal | 100 | 100 | 100 |
| | | | | 58.3 | 93.4 | 88.5 |
| | | | | 34.8 | 54.2 | 54.9 |
| | | | | 72.2 | 92.3 | 92.5 |
| | | | Average | **66.3** | **85.1** | **84.1** |
| 3 | CLM-Z + HOG | Cell=[16,16]<br>Block=[3,3] | Frontal | 100 | 100 | 100 |
| | | | | 56.1 | 94.1 | 83.5 |
| | | | | 32.3 | 57.9 | 58.2 |
| | | | | 76.6 | 96.5 | 92.7 |
| | | | Average | **66.2** | **87.1** | **83.6** |
| 4 | CLM-Z + HOG | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Frontal | 100 | 100 | 100 |
| | | | | 58.3 | 94.1 | 89.2 |
| | | | | 34.8 | 57.9 | 58.5 |
| | | | | 72.2 | 96.5 | 96.4 |
| | | | Average | **66.3** | **87.1** | **86.0** |
| 5 | CLM-Z + LBP | | Frontal | 100 | 100 | 100 |
| | | | | 59.6 | 93.3 | 88.2 |
| | | | | 35.7 | 55.1 | 54.3 |
| | | | | 71.4 | 93.4 | 91.2 |
| | | | Average | **66.7** | 85.4 | 83.4 |
| 6 | CLM-Z + LBP+ 3DLBP | | Frontal | 100 | 100 | 100 |
| | | | | 62.4 | 93.3 | 89.9 |
| | | | | 41.9 | 55.1 | 55.3 |
| | | | | 76.3 | 93.4 | 92.1 |
| | | | Average | **69.7** | 85.4 | 84.3 |
| 7 | CLM-Z+ HOG + LBP | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Frontal | - | - | 100 |
| | | | | - | - | 94.4 |
| | | | | - | - | 59.1 |
| | | | | - | - | 97.9 |
| | | | Average | - | - | **87.8** |
| 8 | CLM-Z+ HOG + LBP + 3DLBP | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Frontal | - | - | 100 |
| | | | | - | - | 90.3 |
| | | | | - | - | 58.6 |
| | | | | - | - | **97.6** |
| | | | Average | - | - | **86.6** |
| 9 | HOG | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Frontal | 100 | 100 | 100 |
| | | | | 50.0 | 93.3 | 82.0 |
| | | | | 30.2 | 53.7 | 49.3 |
| | | | | 71.3 | 87.6 | 86.0 |
| | | | Average | 62.9 | 83.6 | 79.3 |
| 10 | LBP | | Frontal | 100 | 100 | 100 |
| | | | | 54.7 | 93.1 | 81.5 |
| | | | | 31.2 | 52.4 | 47.3 |
| | | | | 68.4 | 85.9 | 83.1 |
| | | | Average | 63.6 | 82.8 | 78.0 |
| 11 | HOG + LBP | RGB (Cell=[16,16], Block=[3,3])<br>Depth (Cell=[10,10], Block=[2,2]) | Frontal | - | - | 100 |
| | | | | - | - | 87.8 |
| | | | | - | - | 53.6 |
| | | | | - | - | 89.3 |
| | | | Average | | | **82.7** |

Similar to Section 4.5, methods 9, 10, and 11 demonstrate the role of CLM-Z modeling in improving the performance of the proposed method (improving the performance from an average of 82.7% to 87.8%).

### *4.7. Evaluating the performance on the random training and test data*

In the previous experiments, the training dataset was chosen in a way that all variations were covered, each image containing exactly one change. In some applications having such a dataset is impractical. To evaluate the effectiveness of the proposed method in such applications, in this section we assess the approach performance on a training dataset whose samples are randomly chosen. In this experiment, IIIT-D dataset is used. This dataset contains face images that have variations in facial expression and head pose. We have used Cumulative Match Characteristic (CMC) curve to evaluate the performance. For this purpose, first we select the training and test subsets for 5 experiments discussed in Section 4.2.1. In each experiment, we calculate the probability of association of each test image to all subjects. For each face, the probabilities are sorted in the descending order. Based on these

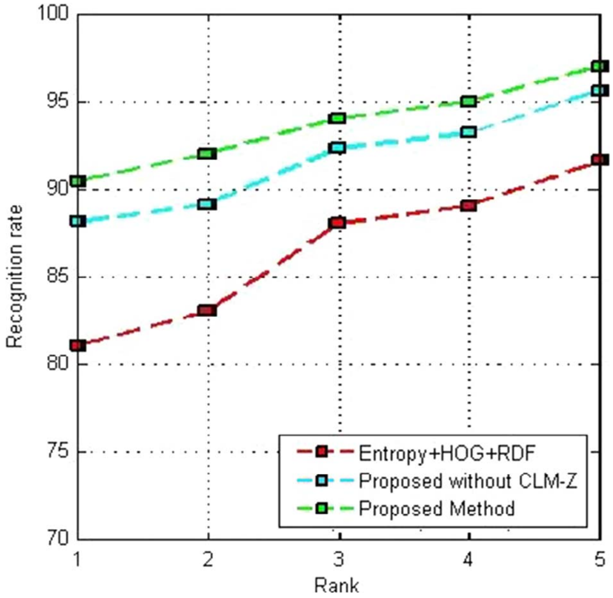


**Fig. 21.** Cumulative Match Characteristic curve for different methods on IIIT-D dataset.

probabilities, the face recognition accuracy is calculated for 5 ranks. In rank 1, the face identity is determined by the highest association probability. In rank 2, two identities with the highest probability are determined for the face image. Therefore, if any of these identities are the correct identity, we count it as a correct classification. Likewise, the performance recognition in rank 3, 4, and 5 is calculated. Finally, the overall recognition performance in each rank is calculated by averaging the performance in five experiments. The overall performance of the method in each rank is the basis for the CMC curve. Fig. 21 plots CMC curve for the proposed method and the method proposed by Goswami et al. [5].

As can be seen in Fig. 21, the proposed method has significantly outperformed the Goswami et al. method [5]. This advantage can be explained from three aspects. First, the face modeling aligns the face images, making their comparison more accurate. Second, the smoothing of depth images reduces the negative effect of missing information on the face recognition performance. The third point is the use of multiple feature descriptor which makes the method more invariant to face variations. Once again, the role of modeling can be observed in Fig. 21 where the proposed method without modeling gets inferior performance.

## 5. Conclusions

In this paper, we have discussed the problem of face recognition in presence of variations in illumination, facial expression, and head pose while the face images are captured by low-resolution intensity and depth sensors. We have tackled this problem by 3D face modeling and using effective feature descriptors on the intensity and depth information. In the proposed method, CLM-Z is employed for face modeling, the feature descriptors are HOG, LBP, and 3DLBP, and finally the SVM classifier is utilized for the recognition.

The experiments have shown that depth information can significantly improve the recognition performance in presence of illumination variation. Furthermore, CLM-Z improves the recognition performance by tracking the landmarks and aligning the face images. The crafted combination of feature descriptors also caused less sensitivity to the changes in illumination and head pose.

Nevertheless, CLM-Z is not very effective when the movement of the head is horizontal. We expect that adding a symmetric filling phase can improve the performance of the method in this cases. To be more precise, the symmetric filling can compensate for the lost information when the head movement is horizontal. Therefore, in the modeling phase the landmarks can be tracked more precisely. This can lead to better alignment and more accurate comparison between face images.

In the future work, the effect of various classifiers for depth and color images on efficiency enhancement can be explored. It is possible that depth and color information require different classifiers.

## References

[1] R. Jafri, H.R. Arabnia, A survey of face recognition techniques, J. Inform. Process. Syst. 5 (2009) 41–68.

[2] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surv. (CSUR) 35 (4) (2003) 399–458.

[3] B.Y. Li, A. Mian, W. Liu, A. Krishna, Using kinect for face recognition under varying poses, expressions, illumination and disguise, in: 2013 IEEE Workshop on Applications of Computer Vision (WACV), 2013.

[4] C. Xu, S. Li, T. Tan, L. Quan, Automatic 3D face recognition from depth and intensity gabor features, Pattern Recogn. 42 (2009) 1895–1905.

[5] G. Goswami, M. Vatsa, R. Singh, RGB-D face recognition with texture and attribute features, IEEE Trans. Inform. Forensics Secur. 9 (10) (2014) 1629–1640.

[6] G. Goswami, S. Bharadwaj, M. Vatsa, R. Singh, On RGB-D face recognition using Kinect, in: 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2013.

[7] N. Nourbakhsh Kaashki, R. Safabakhsh, 3D constrained local model-based face recognition using Kinect under variant conditions, in: 2014 7th International Symposium on Telecommunications (IST), 2014.

[8] Z. Cao, Q. Yin, X. Tang, J. Sun, Face recognition with learning-based descriptor, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[9] G.-S.J. Hsu, Y.-L. Liu, H.-C. Peng, P.-X. Wu, RGB-D-based face reconstruction and recognition, IEEE Trans. Inform. Forensics Secur. 9 (12) (2014) 2110–2118.

[10] M. Hayat, M. Bennamoun, A.A. El-Sallam, An RGB–D based image set classification for robust face recognition from Kinect data, Neurocomputing 171 (2016) 889–900.

[11] B.Y. Li, M. Xue, A. Mian, W. Liu, A. Krishna, Robust RGB-D face recognition using Kinect sensor, Neurocomputing 214 (2016) 93–108.

[12] A. Aldroubi, M. Unser, M. Eden, B-spline signal processing, IEEE Trans. Sign. Process. 41 (1993) 821–849.

[13] G. Goswami, M. Vatsa, R. Singh, Face recognition with RGB-D images using Kinect, in: T. Bourlai (Ed.), Face Recognition Across the Imaging Spectrum, Springer, Cham, 2016, pp. 281–303.

[14] L. Sirovich, M. Kirby, Low-dimensional procedure for the characterization of human faces, J. Opt. Soc. Am. A: Opt. Image Sci. Vis. 4 (1987) 519–524.

[15] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 3, 1991, 71–86.

[16] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 711–720.

[17] B. Moghaddam, C. Nastar, A. Pentland, A Bayesian similarity measure for direct image matching, in: Proceedings 13th International Conference on Pattern Recognition, 1996, pp. 350–358.

[18] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, Face recognition by independent component analysis, in: IEEE Transactions on Neural Networks, November 2002.

[19] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, T. Vetter, A 3D face model for pose and illumination invariant face recognition, in: Sixth IEEE International Conference on AVSS '09, 2009.

[20] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, in: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999.

[21] H.-B. Liao, Q.-H. Chen, Q.-J. Zhou, L. Guo, Rapid 3D face reconstruction by fusion of SFS and local morphable model, J. Vis. Commun. Image Represent. 23 (6) (2012) 924–931.

[22] Z. Guo, Y.-N. Zhang, Y. Xia, Z.-G. Lin, Y.-Y. Fan, D.D. Feng, Multi-pose 3D face recognition based on 2D sparse representation, J. Vis. Commun. Image Represent. 24 (2) (2013) 117–126.

[23] X. Lu, A.K. Jain, D. Colbry, Matching 2.5 D face scans to 3D models, IEEE Trans. Pattern Anal. Mach. Intell. 28 (1) (2006) 31–43.

[24] H. Mohammadzade, D. Hatzinakos, Iterative closest normal point for 3d face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2) (2013) 381–397.

[25] A.S. Mian, M. Bennamoun, R. Owens, Keypoint detection and local feature matching for textured 3d face recognition, Int. J. Comput. Vis. 79 (1) (2008) 1–12.

[26] R. Cendrillon, B. Lovell, Real-time face recognition using eigenfaces, in: Visual Communications and Image Processing 2000, 2000.

[27] T. Kanade, Picture Processing System by Computer Complex and Recognition of Human Faces, Kyoto University, Japan, 1973.

[28] C. Creusot, N. Pears, J. Austin, A machine-learning approach to keypoint detection and landmarking on 3D meshes, Int. J. Comput. Vis. 102 (1–3) (2013) 146–179.

[29] S. Berretti, N. Werghi, A.D. Bimbo, P. Pala, Selecting stable keypoints and local descriptors for person identification using 3D face scans, Vis. Comput. 30 (11) (2014) 1275–1292.

[30] I.J. Cox, J. Ghosn, P.N. Yianilos, Feature-based face recognition using mixture-distance, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1996.

[31] L. Wiskott, J.-M. Fellous, N. Krüger, C.V.D. Malsburg, Face recognition by elastic bunch graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 775–779.

[32] L. Teijeiro-Mosquera, J. Alba-Castro, D. Gonzalez-Jimenez, Face recognition across

pose with automatic estimation of pose parameters through AAM-based landmarking, in: 2010 20th International Conference on Pattern Recognition (ICPR), Istanbul, 2010.

[33] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active shape models – their training and application, Comput. Vis. Image Underst. 61 (1) (1995) 38–59.

[34] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Training models of shape from sets of examples, BMVC92, Springer, London, 1992.

[35] T.F. Cootes, C.J. Taylor, Active shape models—'smart snakes', BMVC92, Springer, London, 1992.

[36] W. Wang, S. Shan, W. Gao, B. Cao, B. Yin, An improved active shape model for face alignment, in: Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, IEEE Computer Society, 2002, p. 523.

[37] K.-W. Wan, K.-M. Lam, K.-C. Ng, An accurate active shape model for facial feature extraction, Pattern Recogn. Lett. 26 (15) (2005) 2409–2423.

[38] A. Faro, D. Giordano, C. Spampinato, An automated tool for face recognition using visual attention and active shape models analysis, in: 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2006, EMBS'06, 2006.

[39] J. González, Daniel, C.J.L. Alba, Toward pose-invariant 2-d face recognition through point distribution models and facial symmetry, IEEE Trans. Inform. Forensics Secur. 2(3) (2007) 413–429.

[40] K. Seshadri, M. Savvides, Robust modified active shape model for automatic facial landmark annotation of frontal faces, in: IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems, 2009, BTAS'09, 2009.

[41] J.M. Saragih, S. Lucey, J.F. Cohn, Face alignment through subspace constrained mean-shifts, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009.

[42] D. Cristinacce, T.F. Cootes, Feature detection and tracking with constrained local models, BMVC 17 (2006) 929–938.

[43] S. Elaiwat, M. Bennamoun, F. Boussaid, A. El-Sallam, A Curvelet-based approach for textured 3D face recognition, Pattern Recogn. 48 (4) (2015) 1235–1246.

[44] T. Baltrušaitis, P. Robinson, L.-P. Morency, 3D constrained local model for rigid and non-rigid facial tracking, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[45] L. Yin, X. Chen, Y. Sun, T. Worm, M. Reale, A high-resolution 3D dynamic facial expression database, in: 8th IEEE International Conference on Automatic Face & Gesture Recognition, 2008, FG'08, 2008.

[46] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Multi-PIE, Image Vis. Comput. 28 (5) (2010) 807–813.

[47] J.M. Saragih, S. Lucey, J.F. Cohn, Face alignment through subspace constrained mean-shifts, in: 2009 IEEE 12th International Conference on Computer Vision, 2009.

[48] N. Dalal, T. Bill, Histograms of oriented gradients for human detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005, 2005.

[49] T. Huynh, R. Min, J.-L. Dugelay, An efficient LBP-based descriptor for facial depth images applied to gender recognition using RGB-D face data, Computer Vision-ACCV 2012 Workshops, Springer, Berlin Heidelberg, 2013.

[50] B.Y. Li, W. Liu, S. An, A. Krishna, Tensor based robust color face recognition, 2012 21st International Conference on Pattern Recognition (ICPR), IEEE, 2012.