Abstract—In this paper, a novel algorithm based on the Active Shape Model (ASM) for locating landmarks on human faces is proposed. A challenge for detecting facial features is that faces may be under different poses; this makes the local appearance of each facial landmark vary greatly. To account for these variations, we propose an adaptive-profile scheme for ASM so that facial landmarks can be detected reliably and accurately under different poses. In our algorithm, a 2D profile is used for each landmark, and the 2D profiles of each landmark of the training face images are grouped to form a number of clusters. The corresponding shape vector for each of the clusters is then learned. For a query face image, the profiles to be used to locate the respective facial landmarks will be selected according to the face-shape vector in the current iteration. In other words, adaptive profiles are used in the search for landmarks. Face images from two subsets of the IMM Face Database are used for training, and the other two subsets are used for testing. The performance of our proposed algorithm is also evaluated using another dataset, namely the Bosphorus Dataset. Experiment results show that our proposed Adaptive-Profile Active Shape Model (APASM) can locate facial landmarks accurately under different face shapes, expressions, and poses.

I. INTRODUCTION

Detecting and locating facial features are vital to real-life applications like face recognition, facial-expression analysis, 3D face reconstruction, and medical-image interpretation. However, unlike face detection, which only aims to find faces in images, facial-feature detection or landmark localization requires a much more accurate position for each of the facial features. The variations caused by facial expressions, poses, occlusion and lighting make this task complicated and challenging.

The deformable template [1] is one of the earliest approaches to facial-feature extraction; it uses parametric models to represent facial features, such as the eyes and mouth. The active contour model [2] has also been used to detect and locate face boundaries. However, these approaches consider local facial features individually, which give them limited robustness and accuracy. The Active Shape Model (ASM) [3] and The Active Appearance Model (AAM) [4] consider the facial features holistically, while AAM also includes the texture information about faces. Hence, these two methods can achieve a more accurate performance, and have become the state-of-the-art methods for facial-feature localization. These two methods also define a number of labeled facial landmarks in faces, and are based on the correlation to the facial information from training samples. ASM only uses gray-scale or texture profile models around each facial landmark to locate the landmark positions whereas AAM considers the appearance of the whole face region as well. Furthermore, ASM seeks to minimize the distance between the model profiles and the corresponding detected landmarks, while AAM attempts to minimize the difference between the target image and the synthesized model image. As stated in [5], AAM can lead to a better match in terms of facial textures, but ASM can achieve a faster and better performance in terms of facial-feature localization. Therefore, in this paper we propose a more efficient and accurate ASM for locating facial landmarks in face images under different poses and expressions.

In the original ASM, two approaches have been proposed to better locate landmark locations. The first is to move each landmark towards the strongest edge within its neighborhood. This approach can perform well in locating the boundary of an object. However, the profiles around most of the facial features are much more complicated than a simple edge. The second approach is to incorporate gray-level information [6] for extracting a one-dimensional gradient profile around each landmark to describe the local texture. The gray-level approach is suitable for a boundary, but not for complex shape structures like the eyes and mouth. This is because the direction of the profile for some inner facial landmarks, like the nose tip or the mouth corners, is difficult to define.

A lot of literature has proposed improving the accuracy of the ASM method. In [7], three facial-feature models and one face contour were proposed to represent a face, and the genetic algorithm (GA) was used for facial-feature extraction. The drawback with this algorithm is that man-made constraints are introduced to separately model the shapes of those inner facial features (eyes, eyebrows, nose, mouth) and that of the face contour. In this way, different definitions are required for different landmarks; this makes the overall algorithm inconsistent. The two-dimensional (2D) profile model was first proposed in [8]; it captures more information around a landmark and can achieve better results than a one-dimensional (1D) profile can. Furthermore, the work has proposed performing searching in two steps, which can obtain a better starting position by sacrificing time efficiency. This 2D profile-based method is an important contribution to ASM and has been evaluated as the most efficient algorithm in a recent comparative study of face landmarking techniques.
Further extended work on using 2D profiles include [10] and [11]. The method in [10] uses the Scale-Invariant Feature Transform (SIFT) descriptor in the facial-feature model, while [11] employs Principal Component Analysis (PCA) to model 2D profiles for each landmark. However, these 2D profile methods still use fixed profiles to describe the texture around the landmarks; they can perform well for frontal-view face images, but their performances are sensitive to changes in pose and expression.

This paper presents a novel approach, namely Adaptive-Profile Active Shape Model (APASM), whose profile models used for each landmark are adaptive to the face shape, i.e. according to pose and expression variations. In other words, the profile models used for searching the landmark locations change with the current face-shape vector. Thus, our method can achieve a more accurate and reliable result when faces are under pose and expression variations. The rest of the paper is organized as follows. In Section II, the training phase and the searching phase of the APASM algorithm will be presented. In Section III, we evaluate our proposed APASM method, and compare it with an state-of-the-art ASM method. Finally, in Section IV, a conclusion will be drawn and the direction of future work will be discussed.

II. THE APASM ALGORITHM

In this section, a brief overview of ASM [3] will first be given. Then, we will present our APASM algorithm, in particular, our new contributions and the details of both the training phase and the searching phase.

In the training of ASM, each training face image is labeled with a predefined number of landmarks on the facial features. The collection of the landmark points for each training sample forms the point-distribution model (PDM). After these PDMs are aligned and demeaned to form the training-sample matrix, PCA is applied to the matrix so that the face shape to be synthesized can only be deformed in controlled ways. Then, by projecting the PDM of a face to the eigenspace, the corresponding shape vector can be calculated and the landmarks can be reconstructed. By varying each of the parameters in the shape vector, new face shapes can be constructed. It should be noted that the magnitude of each parameter in the shape vector should not exceed three times the standard deviation of the training samples.

To extract the facial features in a face image using ASM, an initial model, which is the mean face of the training samples, is placed near the face under consideration. Then, the landmarks move to better positions based on the learned profiles. In order to guarantee that the new landmarks form a face shape, they are aligned to the mean face using an affine transformation. Then, the changed shape is projected onto the eigenspace in order to make sure that the new shape is like those in the training set. Finally, the new shape vector is transformed back to fit the face image. The above steps are repeated until the model converges.

Having introduced ASM, we will, in the following sections, present our proposed APASM algorithm in terms of its training and testing phases.

A. Training Phase for APASM

In this paper, the IMM face dataset [12] is used, and 58 landmarks are labeled in each image. It should be noted that images in the IMM dataset have been annotated for AAM. Since AAM also considers the whole face texture, the accuracy requirement of the annotated landmarks is not as high as that for ASM. Consequently, we have annotated the face images again manually, following the original labeling scheme. Fig. 1 shows the re-annotated landmarks of a face subject at three different views (frontal, left, and right), which are slightly different from their original ones.

![Fig. 1: The landmarks labeled for a face subject at (a) right view (b) frontal view, and (c) left view.](image)

We apply PCA to the training set in the same way as ASM does, and the dimension of the shape vector is reduced from 116 (58 landmarks, with $x$ and $y$ coordinates for each landmark) to 11 which accounts for 95% of total variation. It should also be noted that including too many common faces in the training set should be avoided, otherwise ASM will be over-trained leading to the infrequent shapes being neglected.

1) Profile Model: The profiles used for searching the landmark positions in our proposed APASM method is based on the 2D profile models in [8]. The use of a 2D profile can result in locating landmarks more accurately than using a 1D profile. In our algorithm, we further improve the 2D profile used for searching landmarks. The first improvement is that we calculate two gradient maps for each face image in the $x$ and $y$ directions. As mentioned in [13], the simplest filter [-1 0 1] can provide the best result for human detection. However, this simple filter preserves some high frequency content outside the normal frequency band of images. Therefore, some unnecessary noise will be kept in the gradient maps. In order to reduce the noises, we combine the original band-pass filter with a Gaussian (low-pass) filter, and use the new, narrower, band-pass filter to extract the gradient maps. It can be observed from the experiment that the high-frequency components have been removed efficiently, so the noises in the two gradient maps are also reduced.

The second improvement is that the profile window for matching is rotated and scaled according to the relative orientation of the sample face and the mean face. The angle and scale difference between the two faces can be computed using similarity transform. As illustrated in Fig. 2, the profile
window around Landmark 1 is used as an example. It should be noted that resizing the profiles is very time-consuming. In practice, we only resize the profile models in the training phase using bicubic interpolation to down-sample or up-sample the original profile model. In the searching phase, profile models are extracted directly without performing any rescaling.

Fig. 2: Examples of the rotating and scaling of profiles (the profile window size is pre-set at 25 × 25).

The third improvement is that the profiles are normalized so as to reduce the effect of illumination variations. In [13], the histogram of the cells inside a block is normalized. The reason for this normalization is to remove the variations in local luminance. In our APASM, a similar normalization procedure is applied. Each profile is normalized using the Frobenius norm

\[ FN = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} |p_{ij}|} \]

where \( p_{ij} \) is an element in a 2D profile.

Having described the major improvements for the profiles used in APASM as compared to previous ASM algorithms, we will present our method in detail as illustrated in Fig. 3. In particular, we will now introduce how a nonlinear relationship between the profile models and the face-shape vectors can be established.

![Diagram](image)

Fig. 3: The training phase of APASM.

2) **Unsupervised Learning of Profile Vectors:** As described in the introduction, we aim to establish a relationship between the profile models and the corresponding face-shape vectors. Therefore, with the shape vector of a face, our algorithm will choose a suitable profile model for searching the landmark locations. Two problems remain to be solved before the association between the profile and the shape vector can be established. The first problem is to cluster the profile models according to their similarity. The profile models for those faces at a similar pose and expression should also be similar. If a profile model is being chosen correctly, the corresponding shape vectors will then be assigned the same cluster of profile models. The second problem is how to set the shape vector for each cluster of profile models.

In the APASM algorithm, the \( k \)-means clustering algorithm [14] is applied to cluster the profile models of the training samples, due to its simplicity and good performance. After clustering, the mean of the profile models in each of the clusters can be used for representation. However, the mean is easily affected by outliers or by wrongly clustered samples. Therefore, in our algorithm, a weighted mean of the profile models is used to represent a cluster. Each sample in a cluster is weighted by its distance to the cluster mean as follows:

\[ \mu_{\text{new}} = \frac{\sum_{i=1}^{n} w_{i} p_{i}}{\sum_{i=1}^{n} w_{i}} \]  

where \( p_{i} \) is the \( i \)th sample in the cluster under consideration, \( w_{i} = e^{-|p_{i} - \mu_{\text{old}}|} \) is the corresponding weight assigned to the samples, and \( n \) is the number of samples. This weighted \( k \)-means algorithm is iterated until the weighted mean converges. In this paper, the number of clusters is determined to be 3 depending on the variations of face poses and expressions.

3) **Supervised Learning of the Shape Vector for each Cluster:** Through the above unsupervised learning, profiles or shape vectors are assigned to different clusters. To learn the rule for classifying the shape vectors, supervised learning algorithms can be applied. For simplicity, the Bayes classifier [15] is employed to classify the shape vector for each landmark of a query face into one of the clusters. Based on the clustering results of the training samples, the probability and the density function of each cluster can be derived. The probability of a query shape vector \( x \) belonging to cluster \( C_{i} \) can be computed as follows:

\[ p(C_{i}|x) = \frac{p(C_{i})p(x|C_{i})}{p(x)} = p(C_{i})p(x|C_{i}) \sum_{j=1}^{k} p(C_{j})p(x|C_{j}) \]  

The sample will be assigned to the cluster with the highest probability. In our algorithm, the dimensionality of the face shape vector is 11. More information on the kind of density function of the shape vectors in each cluster is needed. We assume that, within each cluster, the shape vectors fit a multivariate normal distribution. Through the multivariate normality statistical test, it is found that 48.23% of the 174 clusters (58 landmarks × 3 clusters/landmark) fit the multivariate normal distribution. Considering a small number of samples within each cluster, it is reasonable to accept that the data within each cluster follow a multivariate normal distribution.

2851
1. $p(x|C_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$, where $|\Sigma_i|$ represents the determinant of the covariance matrix.

Given that the probabilities of the different clusters are equal, maximizing $p(C_i|x)$ is equivalent to minimizing the distance $D^2 = \ln|\Sigma_i| + (x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i)$. For a given new shape vector, its distance to each cluster is calculated according to this measurement. The shape vector will be classified to the cluster with the minimum $D$.

To verify the effectiveness of our proposed method, we have conducted an experiment which includes 68 training images of 17 distinct subjects. These training samples are divided into 3 clusters based on the profile models $(25 \times 25)$ of each landmark (58 landmarks in total). In other words, for each landmark, 68 shape vectors are classified into 3 clusters based on the profiles around the landmark under consideration. The result shows that for all the landmarks, their overall classification rate is over $85\%$. This proves that the change of the profile models can be reflected by the change of the shape vectors. Thus, the rules developed to classify the shape vectors are effective.

**B. Searching Phase for APASM**

To locate the landmarks of a query face image, the initial face-shape model is placed roughly around the possible face position, as in [3]. In our algorithm, the initial model, i.e. the mean face model, is placed at the center of the square located by the face-detection algorithm of Viola and Jones [16]. In each iteration, the resulting shape vector is used to determine the profile-model cluster to be employed in the next iteration. The selected cluster of profile models should now be more adapted to the pose and expression of the face, and hence the landmarks can be located more accurately. The procedures of the searching phase for APASM is illustrated in Fig. 4. In the following sub-sections, we will discuss how the landmarks can be searched and located.

1) **Profile Matching:** Most of the previous ASM algorithms use the Mahalanobis distance to measure the similarity between a model profile and a testing profile. However, there are two problems with this measurement. First, only a limited number of training samples is usually available; it is impractical to compute the inverse of the covariance matrices. The profile models are always of a high dimension. For example, a profile model of size $15 \times 15$ has a dimension of 450 (the gradients in both the $x$ and $y$ directions at each pixel position), which requires at least 451 training profile samples to make the covariance matrix nonsingular. The second problem, which is more important, can be explained from the probability point of view. The Mahalanobis distance is a special case of the Bayes theorem, when the density functions of all the clusters follow the multivariate normal distribution, and the covariance matrix of each cluster is the same. Comparing two profiles involves one cluster only. As a result, if the profiles fit a multivariate normal distribution, the determinant of the covariance matrix will be a constant, and can be neglected in the comparison. However, through the multiple multivariate normality test, none of profiles within each cluster agree with the multivariate normal distribution (the $p$-value is much smaller than the significance level 0.01). The abovementioned problems with the Mahalanobis distance mean that it may not be a suitable metric to measure the similarity between two profiles.

In our proposed APASM algorithm, we employ a simple but efficient metric to measure the similarity which is the correlation of two profiles under consideration:

$$S(g_1, g_2) = g_1^T \cdot g_2.$$  \hspace{1cm} (3)

Compared to the original Mahalanobis distance, this dot product measurement can significantly reduce the computation complexity, and can also enhance the searching accuracy.

2) **Shrinking the Searching Window:** In APASM, we also take advantage of using multi-level profiling, as in [11], which uses different searching window sizes to avoid rescaling the image. In the experiments, the searching window is set at $21 \times 21$ initially. When the face model converges, the window size is reduced to $17 \times 17$. The whole process will stop until the window size reaches $1 \times 1$. Empirically, if the distance moved between the current model and the previous model is within 2 pixels, the model can be considered achieving convergence, and it will then proceed to the next level. Even though the image is not scaled, the computational complexity is still very high. During the first level, a window size of $21 \times 21$ contains 441 pixels, which means that 441 profile samples are compared with the learned profile model. These comparisons are performed for the 58 landmarks one by one. To further reduce the computational complexity of the searching phase, downsampling is applied. According to experiments, sampling every other pixel will not affect the searching accuracy. However, further reducing the sampling rate will significantly degrade the overall performance of the algorithm.
III. EXPERIMENT

In the experiments, 156 gray-scale face images of 39 distinct subjects in the IMM dataset are selected to form the test set. There are 4 images of each subject, with neutral-frontal, smiling-frontal, neutral-left, and neutral-right views, respectively. Face images with other expressions are not included in the testing due to the limited number of training samples. The size of the profile model is $25 \times 25$, while the number of clusters is set at 3. In the searching phase, the initial window size is set at $21 \times 21$, and is decreased by 4 horizontally and vertically for each level (i.e., $17 \times 17, 13 \times 13$, etc.). To measure the performance of our APASM algorithm, we use the ground-truth-based localization error, as used in [9]. This is a straightforward way to assess landmark-detection and landmark-localization performances, as the ground-truth positions are available. The performance can be expressed in terms of the normalized root-mean-squared error (NRMSE), which is averaged over all the landmarks to produce an overall precision measurement. Given the ground-truth landmarks, the normalized error is computed as follows:

$$e_k^i = \frac{d((x_k^i, y_k^i), (\tilde{x}_k^i, \tilde{y}_k^i))}{IOD}, \quad (4)$$

where $d(\cdot, \cdot)$ indicates the Euclidean distance between the $k^{th}$ ground-truth landmark and the estimated landmark of the $i^{th}$ person. $IOD$ here represents the Inter-Ocular Distance, which is defined as the distance between two eye pupils. According to [9], $e_k^i < 0.1$ can be taken as an acceptable error criterion; this means that a landmark is considered to be detected correctly if its normalized error is within the threshold.

By varying the threshold, we plot the performance results for each pose and thus the overall detection rate, as shown in Fig. 5. From the results, we can see that, if we set the threshold at 0.1, our method can achieve a detection rate over 80% under all poses, and the overall performance can reach a detection rate of 90%, as shown by the dark dashed line.

Since we employ the 2D profile framework from [8], we compare our method with the method in [8] in terms of the localization performance, as shown in Fig. 6. In the comparison, the new version [17] of the method in [8] is compared; this new version has a more efficient and better performance. Our localization results are illustrated using undashed yellow lines, while the results based on [17] are shown using undashed red lines. From the results, we can see that after improving the original 2D profile model, we can achieve much better results under different poses, while the method in [17] is sensitive to pose, as can be seen in Fig. 6(b). Furthermore, we note that, even with the frontal-expression faces, the method in [17] cannot locate the chins very well while our proposed model can fit the chin much better, as shown in Fig. 6(a).

B. Performance of Locating Local Facial Features

Having shown the overall performance in terms of locating all the landmarks, we also measure the individual performances of landmark localization for four different facial-feature regions, namely those landmarks around (1) the eyes

A. Overall Performance

First, we evaluate the overall performance of the proposed APASM algorithm using the following measurement:

$$P = \frac{\sum_{k=1}^{K} \sum_{i=1}^{N} (i : e_k^i < Th)}{K \times N}, \quad (5)$$

where $K$ is the total number of landmarks per face, and $N$ denotes the number of test images. $Th$ is the threshold which controls the acceptable error criterion, as mentioned above.
and eyebrows, (2) the nose, (3) the mouth, and (4) the face contour. We set the threshold for the acceptable error criterion at 0.08 this time, and we show the detailed comparison in Fig. 7. Our proposed APASM algorithm achieves almost the same performance for all the inner facial landmarks (i.e. the first three groups of facial features); this proves that our method can achieve a consistent performance on the different facial features in a range of faces. The chin-contour detection is more difficult than the inner-feature detection since the edge information is easily affected by the background or by clothing occlusion. The performance decreases slightly for the contour part, but can still achieve a detection accuracy of 50% under the threshold. Actually, we found in the experiment that, the detected landmarks are still located on the chin contour, but at different positions from those in the ground-truths.

Fig. 7: Comparison of the accuracy of landmark localization under different groups of facial features

Fig. 8: Localization performance of our algorithm on the Bosphorus dataset.

C. Performance on a Novel Dataset

In order to examine the robustness of our proposed APASM algorithm, we have also applied it to another database, namely the Bosphorus dataset [18]. None of the images in this database has been used in training. Furthermore, the faces in this dataset varies greatly in facial expression and pose, but have been cropped to include only the face region, i.e. to exclude the face contour. We apply our algorithm to locate the inner facial features; some of the results are shown in Fig. 8. We see that APASM can still deal with both expression and pose variations well.

IV. CONCLUSION AND FUTURE WORK

In this paper, a more efficient and accurate active shape model (ASM), namely Adaptive-Profile ASM (APASM), has been proposed, which can improve facial-feature detection performance in several aspects. These include the use of a more effective filter to extract gradient maps for generating the landmark profiles, and correlation to measure the similarity between profiles. Among these improvements, the major contribution is that the proposed APASM algorithm attempts to establish a relationship between the face-shape vectors and the profile models so that the set of model profiles used for searching is adaptive to the faces’ shapes and poses. Therefore, our method can locate facial features more robustly and accurately when a face is under the variations of pose, expression, and shape. Our future work will focus on further improving the robustness of our model against illumination variations and occlusions by facial hair and glasses.

REFERENCES