

# On Utilising Template and Feature-based Correspondence in Multi-view Appearance Models

Sami Romdhani<sup>1</sup>, Alexandra Psarrou<sup>1</sup> and Shaogang Gong<sup>2</sup>

<sup>1</sup> Harrow School of Computer Science, University of Westminster,  
Harrow HA1 3TP, United Kingdom  
[rodhams|psarroa]@wmin.ac.uk

<sup>2</sup> Department of Computer Science, Queen Mary and Westfield College,  
London E1 4NS, United Kingdom  
sgg@dcs.qmw.ac.uk

**Abstract.** In principle, the recovery and reconstruction of a 3D object from its 2D view projections require the parameterisation of its shape structure and surface reflectance properties. Explicit representation and recovery of such 3D information is notoriously difficult to achieve. Alternatively, a linear combination of 2D views can be used which requires the establishment of dense correspondence between views. This in general, is difficult to compute and necessarily expensive. In this paper we examine the use of affine and local feature-based transformations in establishing correspondences between very large pose variations. In doing so, we utilise a generic-view template, a generic 3D surface model and Kernel PCA for modelling shape and texture nonlinearities across views. The abilities of both approaches to reconstruct and recover faces from any 2D image are evaluated and compared.

## 1 Introduction

In principle, the recovery and reconstruction of a 3D object from any of its 2D view projections requires the parameterisation of its shape structure and surface reflectance properties. In practice, explicit representation and recovery of such 3D information is notoriously difficult to achieve. A number of shape-from-X algorithms proposed in the computer vision literature can only be applied on Lambertian surfaces that are illuminated through a single collimated light source and with no self-shadowing effects. Atick *et al.* [1] have applied such a shape-from-shading algorithm to the reconstruction of 3D face surfaces from single 2D images. In real-life environments, however, these assumptions are unlikely to be realistic.

An alternative approach is to represent the 3D structure of objects, such as faces, implicitly without resorting to explicit 3D models at all [3, 14, 15, 17]. Such a representation essentially consists of multiple 2D views together with dense correspondence maps between these views. In this case, the 2D image

coordinates of a point on a face at an arbitrary pose can be represented as a linear combination of the coordinates of the corresponding point in a set of 2D images of the face at different poses provided that its shape remains rigid. These different views span the space of all possible views of the shape and form a vector space. The shape of the face can then be represented by selecting sufficient local feature points on the face. Such representation requires the establishment of dense correspondence between the shape and texture at different views. These are commonly established by computing optical flow [3, 19]. In general, a dense correspondence map is difficult to compute and necessarily expensive. Besides, an optical flow field can only be established if the neighbouring views are sufficiently similar [3].

One can avoid the need of dense correspondence by considering a range of possible 2D representation schemes utilising different degrees of sparse correspondence. In the simplest case, transformations such as translation, rotation and uniform scaling in the image plane can be applied to a face image to bring it into correspondence with another face image. Such transformations treat images as holistic templates and do not in general bring all points on the face images into accurate correspondence. This transformation results in a simple template-based representation that is based only on the pixel intensity values of the aligned view images and does not take into account the shape information explicitly. Such representation, for example, was used by Turk and Pentland to model Eigenfaces [16].

Alternatively, a local feature-based approach can be used to establish correspondences only between a small set of salient feature points. Correspondences between other image points is then approximated by interpolating between salient feature points, such as corners of the eyes, nose and mouth. In Active Appearance Models (AAM) Cootes *et al.* bring two views into alignment by solving the correspondence problem for a selected set of landmark points [4]. The face texture is then aligned using a triangulation technique for 2D warping. In AAM, however, correspondences can only be established between faces of similar views.

Ultimately, modelling view-invariant appearance models of 3D objects, such as faces across all views relies on recovering the correspondence between local features and the texture variation across views. This inevitably encounters problems due to self occlusion, the non-linear variation of the feature positions and illumination change with pose. In particular, point-wise dense correspondence is both expensive and may not be possible across large view changes since rotations in depth result in self occlusions and can prohibit complete sets of image correspondence from being established. However, the template-based image representation such as [6, 16] did not address the problem of large 3D pose variations of a face. Recognition from certain views is facilitated using piece-wise linear models in multiple view-based eigenspaces [9]. Similarly Cootes *et al.* [4] do not address the problem of non-linear variation across views and aimed only at establishing feature-based correspondences between faces of very similar

views. In this case, small degrees of non-linear variations can also be modelled using linear piece-wise mixture models [5].

Romdhani *et al.* [10, 11] have shown that a *View Context-based Nonlinear Active Shape Model* by utilising Kernel Principal Components Analysis [13, 12] can locate faces and model *shape* variations across the view-sphere from profile to profile views. This approach is extended here in a Face Appearance Model of *both Shape and Texture* across views. We introduce two different methods in establishing correspondences between views. The first method uses affine transformation to register any view of a face with a *generic view shape template*. An alternative feature-based approach is examined that utilises a *generic 3D surface model*. We present the two approaches and examine the ability of the two correspondence methods to reconstruct and recover face information from any 2D view image.

In Section 2 of this paper, we introduce a generic-view shape template model and a generic 3D surface model to be used for establishing feature-based correspondences across poses. A Pose Invariant Active Appearance Model using Kernel PCA is discussed in Section 3. In Section 4, we present experimental results and comparative evaluations before we conclude in Section 5.

## 2 Feature Alignment Across Very Large Pose Variations

Accurately modelling the texture of an object requires the corresponding features to be aligned. However, achieving this geometric normalisation across views under large pose variation is nontrivial. Cootes *et al.* [4] and Beymer [2] both align the features of face images on the mean shape of a fixed pose. While this technique is valid when dealing with faces at the same or very similar pose, it is clearly invalid for faces which vary from profile to profile views as illustrated in Fig. 1.



**Fig. 1.** Left: Examples of training shapes. Right: The average shape, and the average shape overlapping the frontal view.

A new correspondence and alignment method is required that must address the following issues:

1. Due to self occlusion, some features visible at one pose are hidden at another pose (e.g. the left eye is hidden from the left profile). This problem can be addressed possibly by two methods: (a) Hidden features can be made explicit to the model without regenerating their texture by utilising a *generic-view shape template* and establishing affine correspondence between views. (b)

A *generic 3D face surface model* can be utilised to establish feature-based correspondence. The hidden features are regenerated using the information of the visible features, based on the bilateral symmetry of faces.

2. Pose change is caused by head rotation out of the image plane. This means that the features' positions vary nonlinearly with the pose and a feature alignment algorithm must be able to cope with nonlinear deformations. We use Kernel PCA to model nonlinear deformations of both the shape and the texture of a face across pose.

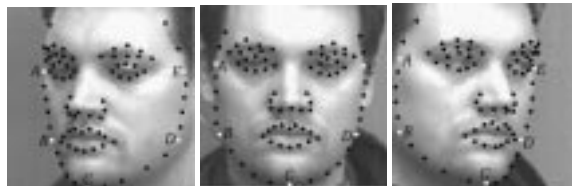
In this paper we discuss two correspondence and alignment methods and evaluate their ability to recover and reconstruct faces from any 2D image. The first alignment technique establishes affine correspondences between views by utilising a generic-view shape template. The second approach uses a local feature-based approach to establish correspondences between views by utilising a generic 3D surface model.

First let us define some notations. A shape  $\mathbf{X}$  is composed of a set of  $N_s$  landmark points  $\mathbf{x}_i$  and the texture  $\mathbf{v}$  of a set of  $N_t$  grey-level values  $v_i$ :

$$\mathbf{x}_i = (x_i, y_i)^T, \mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_{N_s})^T, \mathbf{v} = (v_1, \dots, v_{N_t}) \quad (1)$$

The shape  $\mathbf{X}$  of any single view is composed of two types of landmark points: (a)  $\mathbf{X}_{out}$ , the outer landmark points which define the contour of the face and, (b)  $\mathbf{X}_{in}$ , the inner landmark points which define the position of the features such as mouth, nose, eyes and eyebrows.

In particular, 25 outer landmark points define the contour of the face and 55 inner landmark points define the position of the features such as mouth, nose, eyes and eyebrows. The landmarks that correspond to salient points on the faces are placed manually on the training images whereas the remaining landmarks are evenly distributed between them. This is illustrated in Fig. 2. First, landmarks A, B, C, D and E that are selected to correspond to points on the contour at the height of the eyes, the lips and the chin-tip were set. Then the remaining outer landmarks are distributed evenly between these 5 points. Note that as the view changes the positions of the outer landmarks change accordingly.



**Fig. 2.** Shapes overlapping faces at pose  $-30^\circ$ ,  $0^\circ$  and  $30^\circ$ . The salient outer landmarks (in white) are first manually set then the other outer landmarks (in black) are distributed evenly between the salient ones.

## 2.1 2D Generic-View Shape Template based Alignment

The 2D Generic-View Shape Template Alignment method uses affine transformations to establish correspondences across very large pose variations. It utilises a *generic-view shape template*, denoted by  $\mathbf{Z}$ , on which the landmark points of each view are aligned. The generic-view shape template  $\mathbf{Z}$  is computed based on  $M$  training shapes and the following alignment process:

1. The training shapes  $\mathbf{X}$  of each view are scaled and aligned to yield shape  $\tilde{\mathbf{X}}$ :

$$\tilde{\mathbf{X}} = \frac{(\mathbf{X} - \mathbf{x}_k)}{\|\mathbf{x}_k - \mathbf{x}_l\|} \quad (2)$$

where  $k$  refers to the landmark located on the nose-tip and  $l$  to the chin-tip.

2. These aligned shapes are superimposed.
3. The resulting *generic-view template shape* is formed by the mean of the inner landmark points and the extreme outer landmark points:

$$\mathbf{Z}_{in} = \frac{1}{M} \sum_{i=1}^M (\tilde{\mathbf{X}}_{in,i}) \quad (3)$$

$$\mathbf{z}_{out,j} = \tilde{\mathbf{x}}_{i,j} \quad \text{if } \tilde{\mathbf{x}}_{i,j} \notin \mathbf{Z}_{out} \quad (4)$$

$$\forall i = 1, \dots, M, j = 1, \dots, N_s$$

$$\mathbf{Z} = \mathcal{H}(\mathbf{Z}_{out}, \mathbf{Z}_{in}) \quad (5)$$

where  $\tilde{\mathbf{x}}_{i,j} \notin \mathbf{Z}_{out}$  is true if the point  $\tilde{\mathbf{x}}_{i,j}$  is not included in the area of the shape  $\mathbf{Z}_{out}$  and  $\mathcal{H}(\cdot)$  is the operator which concatenates an outer shape and an inner shape, yielding a complete shape.

The process for creating the *generic-view shape template* is illustrated in Fig. 3.



**Fig. 3.** Left: Left profile, frontal view and right profile shapes aligned with respect to the nose-tip. Right: A generic-view shape template which includes a set of inner feature points as illustrated.

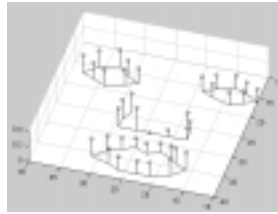
To align the shape and the texture to the generic-view shape template, a fast affine transformation is applied. Examples of aligned textures at different poses is shown in Fig. 4.

To utilise the generic-view shape template, all feature points including the hidden features are made explicit to the model all the time: A special value of grey-level is used to denote hidden points (0 or black).



**Fig. 4.** Example of aligned textures at different poses

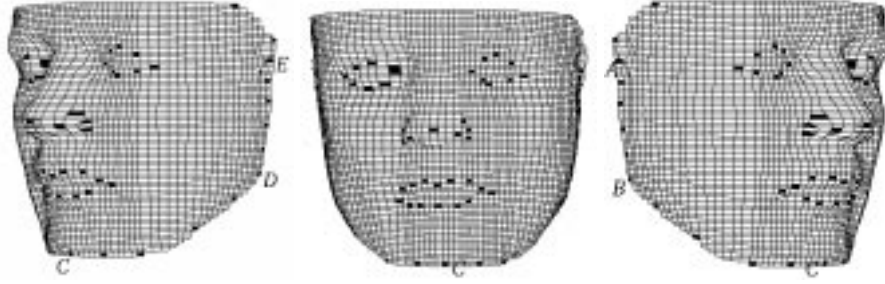
In addition, the initial alignment performed is coarse and is exact only for the nose-tip. The other features are only approximately aligned as illustrated in Fig. 5. The z axis of this 3D graph is proportional to the distance covered by a landmark point on the aligned shape as the pose vary from profile to profile. Ideally this distance for all landmark points should be null, as it is for the nose-tip. Once the initial bootstrapping of the texture alignment is performed, Kernel PCA is applied to minimise the error of the aligned shape  $\hat{\mathbf{X}}$  and the generic-view shape template  $\mathbf{Z}$ .



**Fig. 5.** Variance of the error for inner features alignment across pose. The z axis represents the distance covered by a landmark point as the pose vary from profile to profile relative to the face width.

## 2.2 3D Generic Surface Model based Alignment

We introduce a second feature alignment technique based on a *generic 3D surface model* shown in Fig. 6. It is composed of facets and vertices and constructed using the average of the 3D surface of training faces. A feature-based approach is used to establish correspondence between the 3D model and the 2D image views of a face. Landmarks on the 3D model are placed in the same manner to that of the face images described earlier. In total 64 facets are selected to correspond to the 64 landmarks registered on the images (the eyebrows' landmarks were not used for the alignment). The inner landmark points are placed in facets that correspond to features such as eyes or nose, whereas the outer landmark points are placed on facets that correspond to the extreme outer boundaries of the face model. Examples of the outer landmark points placed on the 3D model is shown in Fig. 6. A property of the outer landmark points of the generic 3D surface model is that their position can vary to outline the outer boundaries of any 2D projected view of the 3D model.

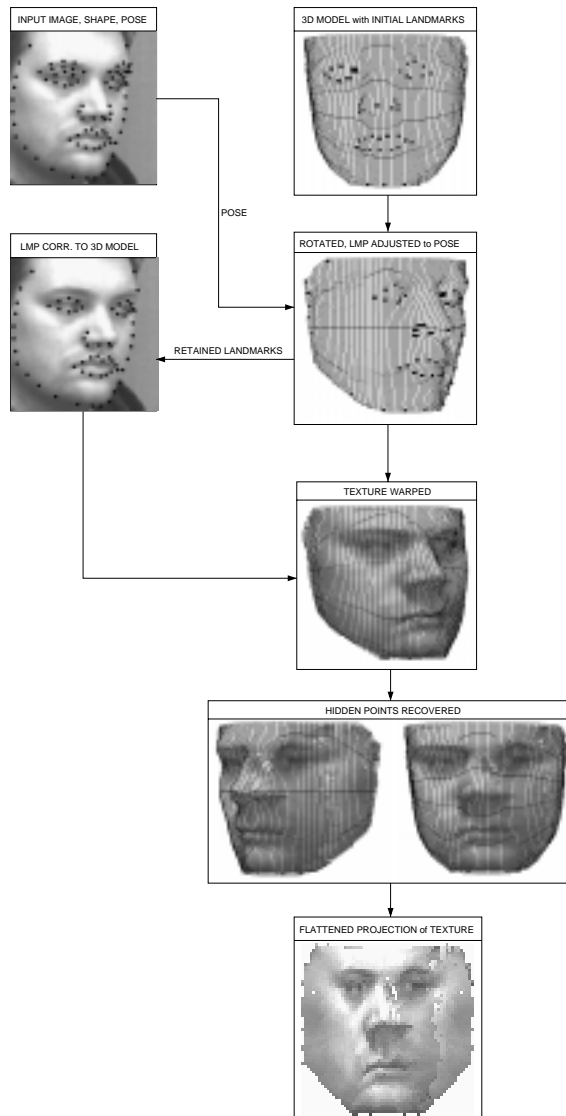


**Fig. 6.** Examples of landmarked facets on the generic 3D surface model.

The feature-based alignment algorithm used to establish the correspondences between the 3D surface model and a 2D face image is outlined in Fig. 7, and consists of the following steps:

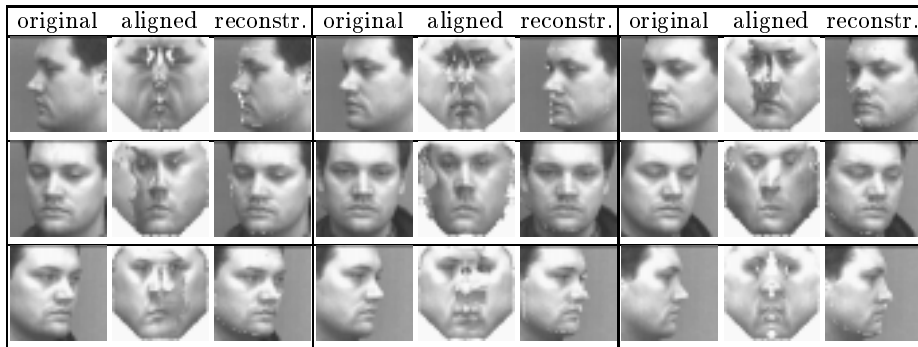
1. **3D Model Rotation:** First the generic 3D surface model is rotated to reflect the same pose as that of the face in the image.
2. **3D Landmarks Recovery:** The position of the landmarks on the 3D generic model relative to the rotated pose are examined in order to determine: (a) which inner landmark points are visible at this pose and therefore can be used for alignment and, (b) the new position of the outer landmark points so that the current visible outer boundary of the generic 3D model is outlined. This process ensures that the landmarks on the generic 3D model correspond to the face image landmarks at that pose.
3. **2D Projection of the Generic 3D Model:** Once the new position of the landmark points of the 3D model has been established, the 2D projection of the generic 3D model at that pose is computed.
4. **2D Texture Warping:** A triangulation algorithm is used to warp the face image on the 2D projection of the 3D model using the landmarks recovered at step 2.
5. **Hidden Points Recovery:** The grey level values of the hidden points are recovered using the bilateral symmetry of faces.
6. **Aligned Texture:** Our aligned texture is a flattened representation of the 3D texture.

Examples of alignment and reconstruction using the generic 3D surface model are shown in Fig. 8. The difference of texture between the visible region and the hidden region is often contrasted due to the lighting conditions. It can be noted that the aligned profile view of an individual is different from the aligned frontal view of the same individual. A more accurate alignment can be obtained using a 3D model containing more facets and higher resolution images.



**Fig. 7.** Overview of the algorithm for aligning the texture of a face based on its shape and its pose using a landmarked 3D model. After rotation of the 3D model, its landmarks are adjusted: the hidden inner points (3 landmarks on the bridge of the nose, here) are dropped and the outer landmarks are moved to be visible. Then a 2D warping is performed from the image to the 2D projection of the 3D model. Next, the grey-level values of the hidden points are recovered using the symmetry of faces and the texture is projected onto 2D yielding a flattened representation of the texture.





**Fig. 8.** Example of alignment of face images at different poses using the 3D Generic Surface Model and their reconstruction.

### 3 Pose Invariant Appearance Model using Kernel PCA

Our process for constructing a pose invariant shape and texture model is illustrated in Fig. 9. The shape is represented as a vector containing the  $x_i$  and  $y_i$  coordinates of  $N_s$  landmarks augmented with the pose angle  $\theta$  as in the *View Context-based Nonlinear Active Shape Model* [10]:  $(x_1, y_1, \dots, x_N, y_N, \theta)$ . Cootes *et al.* [4] used a linear PCA to model the shape and texture of faces. However, under very large pose variations the shape and texture vary nonlinearly despite our texture alignment.

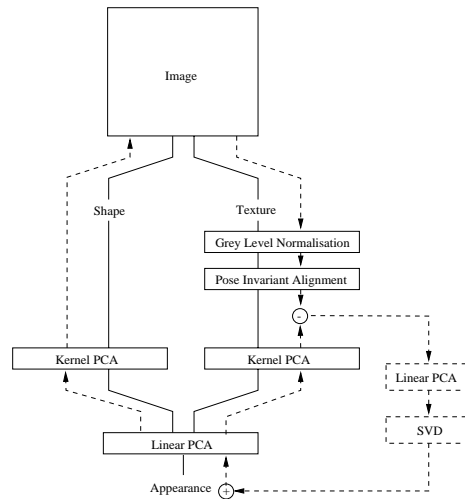
Kernel Principal Components Analysis (KPCA) [13] is a nonlinear PCA method, based on the concept of Support Vector Machines (SVM) [18]. Kernel PCA can also be regarded as an effective nonlinear dimensionality reduction technique which benefits from the same features of PCA. KPCA does not require more training vectors than normal PCA as opposed to mixture models. However, there is one major drawback of KPCA. The reconstruction of a vector from the KPCA space to the original space requires to solve an optimisation problem and it is computationally expensive [8].

Romdhani *et al.* [10,11] successfully used KPCA to model shape and variations from profile to profile views. KPCA is also used to model the aligned texture. However, the combined shape and texture model is built with a linear PCA. This is because our experiments verified that the correlation between the shape and the texture is linear *after* KPCA has been applied to both shape and texture individually.

As explained in Section 2, the model must be constructed using manually landmarked training images. The projection of a *landmarked* new face image to our model can be computed in a single step by computing (1) the projection of its shape (defined by its landmarks), (2) the projection of the underlying texture and (3) the projection of the combined shape and texture. However in the most general case a new face image does not possess landmarks. Hence a *fitting* algorithm is used which recovers the shape and computes the projection of a novel face. This is achieved by iteratively minimising the difference between the image

under interpretation and that synthesised by the model. Instead of attempting to solve such a general optimisation problem for each fitting, the similar nature among different optimisations required for each fitting is exploited. Hence, directions of fast convergence, learned off-line, are used to rapidly compute the solution. This results into a linear relationship between the image space error and the model space error. Before this linear relationship is learned by an SVD regression, a linear PCA is performed to reduce the dimensionality of the image space error and ease the regression. The iterative fitting algorithm described in the following is similar to that used by Cootes *et al.* [4]:

1. Assume initial shape and pose. In the next Section we will detail the constraints set on the starting shape and pose for the algorithm to converge.
2. Compute a first estimation of the projection using the shape and pose from step 1 and the texture of the image underlying the current shape.
3. Reconstruct the shape (along with its pose) and the aligned texture from the current projection.
4. Compute the image space error between the reconstructed aligned texture obtained in step 3 and the aligned texture of the image underlying the reconstructed shape obtained in step 3.
5. Estimate the projection error using the image space error computed in step 4 along with the known linear correlation between the image space error and the model space error computed off-line. This projection error is then applied to the current projection.
6. Go back to step 3 until the reconstructed texture does not change significantly.



**Fig. 9.** An algorithm for constructing a Pose Invariant AAM. The projection and back-projection to and from the model are outlined in plain line. The generation of model parameters for a novel image for which the shape is unknown (the model fitting process) is outlined in dashed line.

## 4 Experiments

To examine and compare the ability of the two approaches to reconstruct and recover faces from any 2D image views we use a face database composed of images of six individuals taken at pose angles ranging from  $-90^\circ$  to  $+90^\circ$  at  $10^\circ$  increments. During acquisition of the faces, the pose was tracked by a magnetic sensor attached to the subject's head and a camera calibrated relative to the transmitter [7]. The landmark points on the training faces were manually located. In the case of the *generic 3D surface model* we used a 3D surface model provided by Michael Burton of the University of Glasgow. We trained three Pose Invariant AAM (PIAAM) on the images of faces of six individuals at 19 poses. The first PIAAM used a 2D generic-view shape template, the second a generic 3D surface model containing 3333 facets and the third a generic 3D surface model containing 13328 facets. In the three cases ten, forty, and twenty eigenvectors were retained to describe the shape, the texture and the combined appearance respectively.

### 4.1 Face Reconstruction

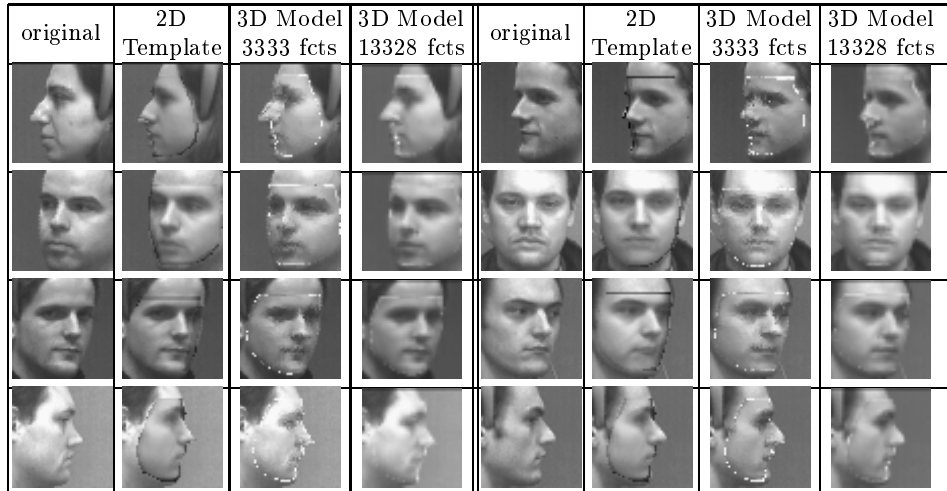
Fig. 10 shows examples of reconstruction of the three PIAAM when the shape and pose of the faces is known. The PIAAM using a generic 3D surface model containing 3333 facets exhibits a "pixelised" effect. The accuracy of the reconstruction of the PIAAM using a 2D generic-view shape template and of the PIAAM using a generic 3D surface model containing 3333 facets is similar while that of the PIAAM using a generic 3D surface model containing 13328 facets is superior. However, the experiments of the next section show that 3333 facets is sufficient to produce a good fitting.

### 4.2 Face Recovery Using the Pose Invariant AAM

**2D Generic-view Shape Template** Fig. 11 illustrates examples of recovering the shape and texture of any 2D image view using a 2D generic-view shape template-based PIAAM trained on five individuals.

While the shape (both pose and feature points) can be recovered adequately, this is not the case for texture. Whilst the pose of the face can be recovered correctly, the intensity information for all pixels is not always recovered. The reason for such effect is that the alignment is only approximate and the variation in the aligned texture due to the pose change overwhelms the variation due to identity difference.

**Generic 3D Surface Model** Fig. 12 shows the recovery of faces from varying poses using generic 3D surface model-based Pose Invariant AAM. The model contained 3333 facets. The linear PCA used in the fitting regression was configured to retain 99% of information yielding 626 eigenvectors. The iterative fitting starts always from a frontal pose shape located near the face on the image and can recover the shape and texture of any 2D image face view. Each iteration takes about 1 sec. (on a normal Pentium II 333 MHz) and the convergence is reached after an average of 4 iterations.

































**Fig. 10.** Example of reconstruction produced by three Pose Invariant AAM. The first image is the original image, the second, the third and the fourth images are its reconstruction yielded by the Pose Invariant AAM using the 2D generic-view shape template, the 3D generic surface model containing 3333 facets and the 3D generic surface model containing 13328 facets, respectively. The reconstructions are computed using the manually generated shape.

### 4.3 On Model Convergence

The AAM introduced by Cootes *et al.* requires a good starting shape to reach convergence [4]. That is, an estimation of the position of the face and of its pose must be known. Fig. 13 depicts the dependency on this requirement for the Pose Invariant AAM using the 2D generic-view surface template and the generic 3D surface model by showing the proportion of searches which converged for different initial displacement and pose offset. The 2D generic-view shape template-based PIAAM is very constrained by its initial pose and location : if the pose is known within  $10^\circ$  accuracy, it has 80% chances to reach convergence if the  $x$  offset is within 4 pixels. However, the generic 3D surface model-based PIAAM has 80% chances to reach convergence if the pose offset is within  $50^\circ$  and the  $x$  offset within 4 pixels (Note that the faces have average of 30 pixels in  $x$ ). This is because the 3D surface model alignment is more accurate than the 2D generic-view shape template alignment. As expected, the better the pose is known, the lower the dependency on the estimation of the face location.

## 5 Conclusions

We have presented a novel approach for constructing a *Pose Invariant Active Appearance Model* (PIAAM) able to capture both the shape and the texture of faces across large pose variations from profile to profile views. We illustrated why

































































original	start	iterations	conv.	shape		
						pose: $-90^\circ$ , init. pose: $-50^\circ$ , init. offset: (0, 0) shape error: (2.09, 0.76)
						pose: $-50^\circ$ , init. pose: $0^\circ$ , init. offset: (-4, -3) shape error: (3.67, 0.82)
						pose: $-90^\circ$ , init. pose: $0^\circ$ , init. offset: (-6, 0) shape error: (2.43, 0.97)
						pose: $90^\circ$ , init. pose: $50^\circ$ , init. offset: (-6, 0) shape error: (0.87, 0.65)
						pose: $-40^\circ$ , init. pose: $0^\circ$ , init. offset: (0, -6) shape error: (6.62, 2.84)

**Fig. 11.** Face recovery of a 2D generic-view shape template-based Pose Invariant AAM trained on five individuals. Each row is an example of texture and shape fitting. The first image is the original image, the following images are obtained at successive iterations. The penultimate image shows the converged fitting of both shape and texture and the last image overlaps the recovered shape on the original image.

the key to effective Pose Invariant AAM is the choice of an accurate but also computationally viable alignment model and its corresponding texture representation. To that end, we introduced and examined quantitatively two alignment techniques for the task: (a) A *2D Generic-view Shape Template* using affine transformations to bootstrap the alignment before it is further refined by the use of Kernel PCA. (b) A *Generic 3D Surface Feature Model* using projected dense 3D facets to both establish local feature-based correspondence between facial points across pose and recover the grey level values of those points which are hidden at any given view. Our extensive experiments have shown that whilst the reconstruction accuracy of the 2D generic-view template-based PIAAM is similar to that of a generic 3D surface feature-based PIAAM using 3333 facets, the reconstruction performance of a generic 3D feature-based PIAAM using 13328 is superior. Furthermore, good fitting was produced using a PIAAM based on a generic 3D surface model containing 3333 facets. On the other hand, the fitting of a 2D generic-view shape template-based PIAAM was shown to have a greater degree of dependency on the initial positions before fitting.

## Acknowledgment

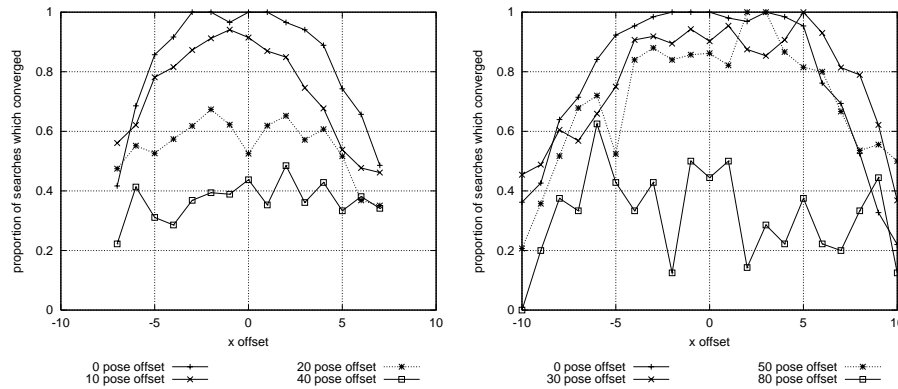
We would like to thank Mike Burton of the University of Glasgow who provided us with the 3D dense facets-based face surface model.

original	start	iterations					conv.	information
								init. offset: (10, 0) shape error: (0.92, 0.47) pose error: -1.98°
								init. offset: (10, 0) shape error : (1.01, 0.74) pose error: -0.20°
								init. offset: (7, 0) shape error: (0.94, 0.51) pose error: -1.52°
								init. offset: (-10, -5) shape error : (1.06, 0.49) pose error: -7.92°
								init. offset: (-8, -5) shape error : (0.94, 0.51) pose error: 2.09°
								init. offset: (-10, -5) shape error : (1.3, 0.71) pose error: 11.49°
								init. offset: (7, 2) shape error : (0.79, 0.85) pose error: -3.34°
								init. offset: (7, 2) shape error : (0.93, 0.55) pose error: -1.89°
								init. offset: (3, 0) shape error : (1.24, 0.74) pose error: -2.9°

**Fig. 12.** Face recovery of a PIAAM using the generic 3D surface model containing 3333 facets trained on six individuals. Each row is an example of texture and shape fitting. The first image is the original image, the followings images are obtained at successive iterations until convergence. Each fitting started from the frontal pose ( $0^\circ$ ).

## References

1. J. J. Atick, P. A. Griffin, and A. N. Redlich. Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Computation*, 8(6):1321–1340, 1996.
2. D. Beymer. Feature correspondence by interleaving shape and texture computations. In *cvpr*, pages 921–928, 1996.
3. D. J. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272:1905–1909, 28 June 1996.
4. T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *ECCV98*, pages 484–498, 1998.
5. T.F. Cootes and C.J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17:567–573, 1999.



**Fig. 13.** Proportion of searches which converged for the Pose Invariant AAM using the 2D generic-view shape template (on the left) and using the generic 3D surface model (on the right) from different initial displacement (in pixel) and pose offset. Note that the faces have average of 30 pixels in width. Note the different pose offsets of the two graphs.

6. F. de la Torre, S. Gong, and S. J. McKenna. View-based adaptive affine tracking. In *ECCV*, pages 828–842, Freiburg, Germany, June 1998.
7. S. Gong, E. J. Ong, and S. McKenna. Learning to associate faces across views in vector space of similarities to prototypes. In *BMVC*, pages 54–63, 1998.
8. S. Mika, B. Schölkopf, A. Smola, G. Ratsch, K. Müller, M. Scholz, and G. Rätsch. Kernel pca and de-noising in feature spaces. In *NIPSS*, 1998.
9. A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *CVPR*, pages 84–91, Seattle, July 1994.
10. S. Romdhani, S. Gong, and A. Psarrou. Multi-view nonlinear active shape model using kernel pca. In *BMVC*, pages 483–492, September 1999.
11. S. Romdhani, A. Psarrou, and S. Gong. Learning a single active shape model for faces across views. In *IEEE International Workshop on Real Time Face and Gesture Recognition*, pages 31–38, September 1999.
12. B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K. Müller. Kernel pca pattern reconstruction via approximate pre-images. In *ICANN*. Springer Verlag, 1998.
13. B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
14. A. Shashua. *Geometry and Photometry in 3D Visual Recognition*. PhD thesis, MIT, AI Lab., 1992.
15. A. Shashua. Algebraic functions for recognition. A. I. Memo 1452 (C.B.C.L. Paper 90), MIT, January 1994.
16. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
17. S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE PAMI*, 13(10):992–1006, October 1991.
18. V. Vapnik. *The nature of statistical learning theory*. Springer Verlag, 1995.
19. T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. Technical Report 16, Max Planck Inst. fur Bio. Kybernetik, 1995.