# A Multi-View Nonlinear Active Shape Model Using Kernel PCA

Sami Romdhani † , Shaogang Gong ‡ and Alexandra Psarrou †

† Harrow School of Computer Science, University of Westminster, Harrow HA1 3TP, UK [rodhams|psarroa]@wmin.ac.uk

‡ Department of Computer Science, Queen Mary and Westfield College, London E1 4NS, UK sgg@dcs.qmw.ac.uk

**Abstract**

Recovering the shape of any 3D object using multiple 2D views requires establishing correspondence between feature points at different views. However changes in viewpoint introduce self-occlusions, resulting nonlinear variations in the shape and inconsistent 2D features between views. Here we introduce a multi-view nonlinear shape model utilising 2D view-dependent constraint without explicit reference to 3D structures. For nonlinear model transformation, we adopt Kernel PCA based on Support Vector Machines.

## 1 Introduction

Modelling the 3D shape of any rigid or non-rigid object in principle requires the recovery of its 3D structure from 2D images. However, accurate 3D reconstruction has proved notoriously difficult to achieve and has never been fully implemented in computer vision. It can be shown that under certain restrictive assumptions, it is possible to faithfully represent the 3D structure of objects such as human faces and bodies without resorting to explicit 3D models. Such a representation would consist of multiple 2D views together with dense correspondence maps between these views, although practically only sparse correspondence can be established quickly for a carefully chosen set of feature points [5]. It should also be able to cope with object shape variations due to changes in viewpoint and self-occlusion, and in the case of an articulated object, changes in configuration [10, 11].

Cootes, Lanitis and Taylor *et al.* [2, 4, 8] have shown that the 2D shape appearance of objects can be modelled using Active Shape Models (ASM). An ASM consists of a Point Distribution Model (PDM) aiming to learn the variations of valid shapes, and a set of flexible models capturing the grey-levels around a set of landmark feature points. While this approach can be used to model and recover some changes in the shape of an object, it can only cope with largely linear variations. For instance, a single ASM is able to cope with shape variations from a narrow range of face poses (turning and nodding of $\pm 20°$). Nonlinear variations caused by changes in viewpoints and self-occlusions from different hand gestures had to be captured through the use of five different models [8]. Active shape models are based on a number of implicit but crucial assumptions: (i) the shape of the object of interest can be defined by a relatively small set of explicit view models, (ii) the grey levels around a particular landmark point are consistent for all the views of the

Figure 1: The typical 2D shape of a face across views from -90° to +90° can be given by a set of facial landmark feature points and their corresponding local grey-level structures.

object and can be used to find correspondences between these views and, (iii) the shapes at different views vary linearly. These assumptions are valid when the variations allowed are well constrained.

However, it is difficult for this approach to cope with largely nonlinear shape variations and the inconsistency introduced in the landmarks as a result. This is illustrated in Figure 1. A set of typical landmarks used for a 2D shape model of a face are superimposed on face images varying from the left to the right profile view. The local grey-levels around the landmarks vary widely. This is highlighted for landmark point $\mathcal{A}$ which clearly cannot be established across views solely based on local grey-levels. Due to self-occlusion, 2D local image structures correspond to different parts of the 3D structure of an object.

In this work, we describe a multi-view nonlinear active shape model that utilises 2D view-dependent contextual constraint without explicit reference to 3D structures. Such a model captures all possible 2D shape variations in a training set and performs a nonlinear transformation of the model during matching. The model therefore is able to cope with both nonlinear shape and grey-level variations around the landmarks. In particular, using a face database across the view sphere, we explicitly represent 2D view-based *context* spanned by the *yaw* variations of a face, referred to as the *pose*. For nonlinear model transformation, we adopt a nonlinear Principal Components Analysis (PCA), known as the Kernel PCA [13] based on Support Vector Machines [14].

The rest of this paper is arranged as follows. We first outline Kernel PCA in Section 2. In Section 3, we describe a new search algorithm that is used to simultaneously match nonlinear face shape variations and recover their poses. A set of experiments demonstrating the effectiveness of the approach are presented in Section 4 before conclusions are drawn in Section 5.

## 2 Kernel Principal Components Analysis

The Active Shape Model (ASM) can only be used to model faithfully objects whose shape variations are linear [2, 8]. When the Valid Shape Region (VSR) in the shape space is nonlinear, as in the case when large pose variations are allowed, the PDM of an ASM requires nonlinear transformations. If a linear PDM is used, the model would suffer from poor specificity and compactness (see experiments shown in Section 4). The problem can be addressed to some extend by approximations using a combination of linear components [3, 7]. However, the use of linear components increases the dimensionality of the model and also allows for non valid shapes [1]. Although nonlinear shape variation can be captured by a set of structured linear models using hierarchical principal components [10], this requires a very large database for learning the distribution of the linear subspaces.

Kernel Principal Components Analysis (KPCA) is a nonlinear PCA method recently introduced by Sholkopf *et al.* [13], based on Support Vector Machines (SVM) [14]. The

essential idea of KPCA is both intuitive and generic. In general, PCA can only be effectively performed on a set of observations that vary linearly. When the variations are nonlinear, they can always be mapped into a higher dimensional space which is again linear. If this higher dimensional linear space is referred to as the *feature space* ($\mathcal{F}$), Kernel PCA utilises SVM to find a computationally tractable solution through a simple kernel function which intrinsically constructs a nonlinear mapping from the input space to $\mathcal{F}$. As a result, KPCA performs a nonlinear PCA in the input space.

More precisely, if a PCA is aimed at decoupling nonlinear correlations among a given set of shape vectors $\mathbf{x}_j$ through diagonalising their covariance matrix, the covariance can be expressed in a linear feature space $\mathcal{F}$ instead of the nonlinear input space, i.e.

$$\mathbf{C} = \frac{1}{M} \sum_{j=1}^{M} \Phi(\mathbf{x}_j) \Phi(\mathbf{x}_j)^{\mathsf{T}} \tag{1}$$

where $\Phi(\cdot)$ is a nonlinear mapping function which projects the input vectors from the input space to the $\mathcal{F}$ space. To diagonalise the covariance matrix, the eigen-problem $\lambda \mathbf{p} = \mathbf{C}\mathbf{p}$ must be solved in the $\mathcal{F}$ space. As $\mathbf{C}\mathbf{p} = \frac{1}{M} \sum_{j=1}^{M} (\Phi(\mathbf{x}_j) \cdot \mathbf{p}) \, \Phi(\mathbf{x}_j)^{\mathsf{T}}$, all nonsingular solutions $\mathbf{p}$ with $\lambda \neq 0$ must lie in the span of $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_M)$. This eigen-problem is equivalent to

$$\lambda(\Phi(\mathbf{x}_k) \cdot \mathbf{p}) = (\Phi(\mathbf{x}_k) \cdot \mathbf{C}\mathbf{p}) \tag{2}$$

for all $k = 1, \ldots, M$ and there exists coefficients $\alpha_i$ such that

$$\mathbf{p} = \sum_{i=1}^{M} \alpha_i \Phi(\mathbf{x}_i). \tag{3}$$

Substituting Equation (2) with (1) and (3) gives

$$\lambda \sum_{i=1}^{M} \alpha_i (\Phi(\mathbf{x}_k) \cdot \Phi(\mathbf{x}_i)) = \frac{1}{M} \sum_{i=1}^{M} \alpha_i (\sum_{j=1}^{M} (\Phi(\mathbf{x}_k) \cdot \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i))) \tag{4}$$

It is important to note that this eigen-problem only involves dot products of mapped shape vectors in the feature space $\mathcal{F}$. This is the *raison d' être* of this method. Indeed, the nature of structural risk minimisation suggests that mapping $\Phi(\cdot)$ may not always be computationally tractable although exists. However, it needs not be explicitly computed either. Only dot products of two vectors in the feature space are needed. Even so, since the feature space has high dimensionality, computing such dot products could still be computationally expensive if at all possible. A support vector machine can be used to avoid the needs for either explicitly performing mappings $\Phi(\cdot)$ or dot products in the high dimensional feature space $\mathcal{F}$. Let us define an $M \times M$ matrix $\mathbf{K}$ where $k_{ij} = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, Equation (4) can then be rewritten as

$$M\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha} \tag{5}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]^{\mathsf{T}}$. Now, performing PCA in the feature space $\mathcal{F}$ amounts to resolving the eigen-problem of (5). This yields eigenvectors $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^M$ with eigenvalues $\lambda^1 \geq \lambda^2 \geq \ldots \geq \lambda^M$. Dimensionality can be reduced by retaining only the first

$L$ eigenvectors. The principal components $\mathbf{b}$ of a shape vector $\mathbf{x}$ are then extracted by projecting $\Phi(\mathbf{x})$ onto eigenvectors $\mathbf{p}^k$ where $k = 1, \ldots, L$

$$b_k \equiv \mathbf{p}^k \cdot \Phi(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) \tag{6}$$

To solve the eigen-problem of Equation (5) and to project from the input space to the KPCA space using Equation (6), one can avoid the needs for computing both the dot products in the feature space and performing the mappings through constructing a SVM (Figure 2). This is achieved by finding a *kernel function* when applied to a pair of shape vectors in the input space, it yields the dot product of their mapping in the feature space:

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) \tag{7}$$

There exists a number of kernel functions which satisfy the above criterion [14]. This includes the Gaussian kernel we have adopted where $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$.

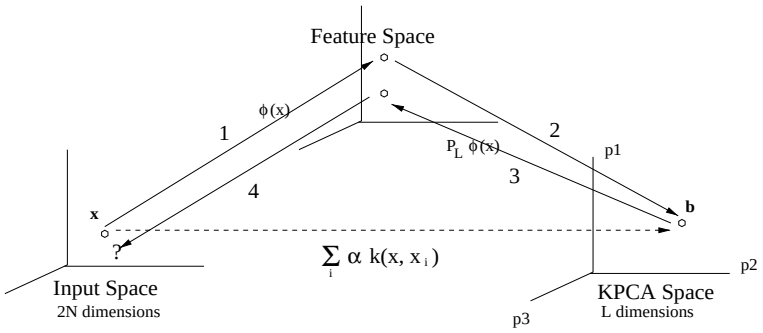


Figure 2: *Conceptually*, KPCA performs a nonlinear mapping $\Phi(\mathbf{x})$ to project an input vector to a higher dimensional feature space $\mathcal{F}$ (step 1). A linear PCA is then performed in this feature space giving a lower dimensional KPCA space based representation (step 2). To reconstruct an input vector from the KPCA space, its KPCA representation is projected into the feature space (step 3) before an inverse $\Phi(\mathbf{x})$ mapping is performed (step 4). *Computationally*, however, none of the four steps is performed. The mapping is in fact carried out directly by kernel functions $\sum_i \alpha k(\mathbf{x}, \mathbf{x}_i)$ between the input space and its KPCA space, shown as the dashed line in the diagram. For reconstruction, this kernel-based mapping is only approximated. Optimisation is required in the KPCA space in order to find a best match between the model and the KPCA representation of the input vector.

This SVM based kernel function effectively provides a low dimensional Kernel-PCA subspace which represents the distribution of the mapping of the training vectors in the high dimensional feature space $\mathcal{F}$. As a result, nonlinear shape transformation in the input space can be performed by reconstructions from the KPCA subspace. However, this process can be problematic [9, 12]. The vectors in the feature space $\mathcal{F}$ which have a pre-image in the input space are the ones which can be expressed as a linear combination of the vectors $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_M)$. However, if the reconstruction in $\mathcal{F}$ is not perfect, there is no guarantee to find a pre-image of the reconstruction in the input space (Figure 2). Indeed, if dimensionality reduction is used, then the reconstruction from the KPCA space to $\mathcal{F}$

can only be an approximation. Therefore the reconstruction ($\hat{\mathbf{x}}$) of an input observation vector ($\mathbf{x}$), whose principal components is truncated to the first $L$ components, must be approximated by minimising

$$\|\Phi(\hat{\mathbf{x}}) - P_L\Phi(\mathbf{x})\|^2 \tag{8}$$

where $P_L$ is a truncation operator. To solve this minimisation problem, there exists optimisation techniques tailored to particular kernels [9].

# 3 View-Context Based Nonlinear Active Shape Model

The Active Shape Model applied to the modelling of faces exhibits only limited pose variations. One implicit but crucial assumption of the existing method is that correspondences between landmark points of different views can be established solely based on the grey-level information. However, when large nonlinear shape variations are introduced due to changes in object pose, the grey-level values around the landmarks are also view dependent. In general, 2D image structures do change according to 3D context. In order to find correspondences between landmarks across large variations of shape, we make explicit use of this contextual information in the model.

In the case of face varying from profile to profile, this contextual information is indexed by the pose angle itself. Consequently, the shape vector for the PDM is augmented by the pose angle $\theta$: $(x_1, y_1, \ldots, x_N, y_N, \theta)$ where $(x_i, y_i)$ are the coordinates of the i$^{\text{th}}$ landmark. Similarly, a model for the Local Grey-Levels (LGLs) around each landmark is a concatenation of the grey-levels along the normal to the shape contour and the pose of the face. Both the PDM and the LGLs are built using Kernel PCA. Given that view-contextual based constraints are built into the models, these models are used to match novel images of faces. It is assumed that a rough position of a face in the image is known. However the pose is unknown and the matching of the models with the target image recovers both the shape of the face and its pose. The computation is performed as follows:

1. An iterative process starts from the frontal view of the shape located near the object on the image. Notice that it is better to start from a specific view rather than the average shape, as was adopted in [4]. This is because as we are dealing with large shape variations, the average shape is not a valid shape anymore, as illustrated in Figure 3.

2. To find plausible correspondences of landmarks between views, augmented local grey-level models are used. To this end, the KPCA reconstruction of the grey-level vector is minimised along the normal to the shape. To compute the KPCA reconstruction of a vector, one first projects this vector to the KPCA space using Equation (6), obtaining the kernel principal components ($\mathbf{b}$). The reconstruction is then performed by minimising the norm given in Equation (8). During the first iteration the pose of the object is unknown. Therefore the reconstruction error must also be minimised with respect to poses. This process yields an estimation of both the landmark positions and the pose for each landmark. The newly estimated pose is then the average pose of all the landmarks. This pose is to be used to constrain the shape within the Valid Shape Region (VSR) at step 3.

3. The estimated shape is aligned as explained in [4].

4. To constraint the estimated shape within the VSR, it is projected to the shape space using the view-context based nonlinear PDM given by Equation (6), constrained to lie

within the VSR by limiting the values of **b** [4] and projected back to the input space using Equation (8). This yields a new estimated shape. Its pose will be used to locate the correspondence of the landmark points at the next iteration (step 3).

5. Repeat step 3 until convergence.

Figure 3: Examples of training shapes (top) and the average shape at the frontal view (bottom).

# 4 Experiments

To illustrate our approach, we use a face database composed of images of 6 individuals taken at pose angles ranging from $-90°$ to $+90°$ at $10°$ increments. The pose of the face is tracked by means of a magnetic sensor attached to the subject's head and a camera calibrated relative to the transmitter [6]. The landmark points on the training faces were manually located. An example of such a sequence was shown in Figure 1.

**On linear ASM coping with pose change**: A linear PDM trained to capture face shape variation between a small range of poses ($\pm 20°$) was compared to a PDM trained for a full range of poses between $\pm 90°$. Figure 4 shows the 2 main modes of variation for each of these linear PDMs. The Valid Shape Range (VSR) for training the $\pm 20°$ PDM was set to $\pm 3\sqrt{\lambda_i}$ and the PDM for across $\pm 90°$ views was limited to $\pm 0.2\sqrt{\lambda_i}$.

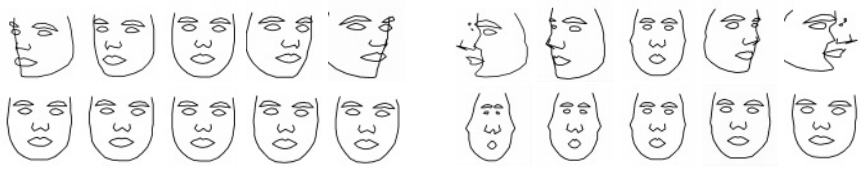

Figure 4: The first (top) and second (bottom) modes of shape variation for a linear PDM covering $50°$ views (left) and across $180°$ views (right). The range of variation for the $50°$ views PDM was set to $\pm 3$ times of standard deviation whilst $\pm 0.2$ times of standard deviation was limited for the model covering $180°$ views.

The two PDMs in Figure 4 and their corresponding LGL models were used to fit ASMs to face images, as shown in Figure 5. Using the model trained for the $50°$ pose range, an ASM was able to fit shapes to face images quite well (left). However, when the PDM for the full pose range was used, an ASM was only able to fit shape satisfactorily near the frontal view. At most of the other poses, the ASM was unable to recover the shape within the Valid Shape Range. This is mainly because both the shape of the face and the local grey-levels at the landmarks vary significantly and nonlinearly across views.

Figure 5: Fitting shapes to images using linear ASMs trained across $\pm 20^\circ$ (left) and $\pm 90^\circ$ (right).

**On nonlinear ASM coping with pose change**: Kernel PCA was used to train a nonlinear PDM and capture shape variations of a face across views ($\pm 90^\circ$). Figure 6 shows the three main modes of variation and illustrates that the nonlinear PDM succeeds in capturing valid variations of shape and extends the VSR of linear PDMs shown in Figure 4.

Figure 6: First three modes of shape variation for a Kernel PCA based PDM. The range of variation is set to $-1.5\sqrt{\lambda_i} \leq b_i \leq 1.5\sqrt{\lambda_i}$.

The nonlinear PDM and its corresponding LGL models were used to fit a nonlinear ASM on face images, as shown in Figure 7. The nonlinear ASM converges and recovers shapes within the VSR but not to the right shape. This is because sometimes the background grey-levels are very similar to the grey-levels around certain landmarks at specific poses, as can be seen from the examples in Figure 1. In such cases, using grey-levels alone will fail to find correspondences between views. To better discriminate object foreground and background, we use pose to impose a view-context based constraint.

Figure 7: Examples of fitting shapes to images at different views using a nonlinear ASM.

**On view-context based nonlinear ASM**: We used the view-context based nonlinear PDM to capture the shape variation across the full range of poses. Figure 8 shows the 3 main modes of variation similar to those of a nonlinear PDM shown in Figure 6.

Figure 8: First three modes of shape variation for a view-context based nonlinear PDM. The range of variation is set to $-1.5\sqrt{\lambda_i} \leq b_i \leq 1.5\sqrt{\lambda_i}$.

Figure 9: Fitting shapes to images at different views using a view-context based nonlinear ASM.

A view-context based nonlinear PDM and its corresponding LGL models were used to fit a view-context based nonlinear ASM to face images, as shown in Figure 9. The ASM converges to the right shape and is able to recover the pose. We used the frontal view shape to start fitting. For the first iteration, the landmarks were allowed to move along the normals to the shape contour for up to a distance of 12 pixels on each side. This was then adjusted proportionally to the fitting error after each iteration. A LGL model was built using 3 pixels on both sides of a landmark along the normal to the shape. Both the PDM and the LGLs were restrained to ten dimensional eigenspaces.

Figure 10 illustrates an example of fitting a shape to a face image. From left to right, the top row depicts the shape transformation in the process. The bottom row shows both pose recovery (convergence towards $-80°$) and shape fitting errors in pixels. Figure 11 compares fitting errors of different ASMs. A linear ASM performs better at mean poses than at extreme poses. A nonlinear ASM exhibits similar results except at mean poses. For all poses, a view-context based nonlinear ASM performs significantly better.

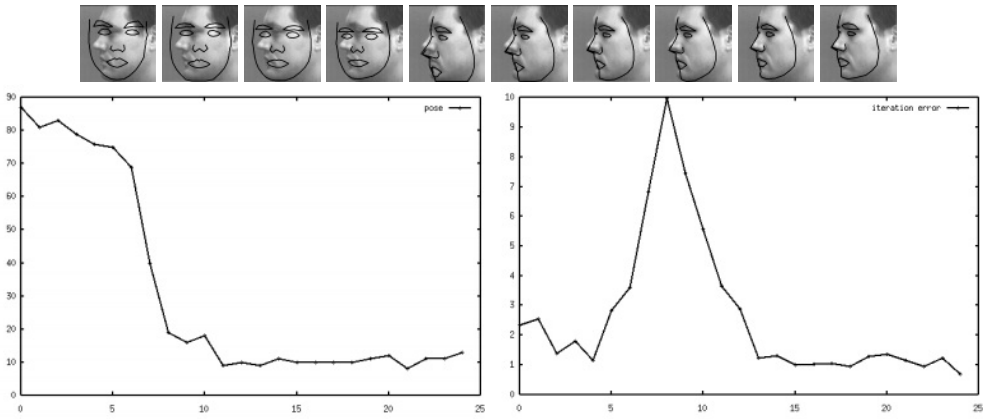


Figure 10: An example of fitting a shape to a face image and recovering its pose at $-80°$. The top row shows estimated shape after iterations 0, 1, 4, 6, 12, 13, 15, 16, 20 and 25. The recovered pose and the fitting errors (in pixels) are shown at the bottom left and right respectively.

**Generalisation to novel views and novel faces**: Two more experiments were conducted to evaluate the capability of the view-context based nonlinear ASM for interpolating shape of novel faces not in the training set and recovering poses at novel views. A view-context based nonlinear ASM was first trained at $20°$ pose intervals between $\pm 90°$. The model was then used to recover both the shape and pose of faces at novel views. Here the number of eigenvectors was increased to 20 and the Valid Shape Region was extended to 10 times the standard deviation. Examples of shape fitting at novel views between known poses are shown in Figure 12.

A view-context based nonlinear model was also trained to recover both the shape and pose of novel faces not in the training set. A model was trained on all but one of the faces in a database and was then tested on all poses of an unknown face. The experiment was performed for a number of unknown faces and an example is shown in Figure 13.

A comparison of the fitting errors of both a model trained on all the poses and a model trained only on half of the poses in a database is shown on the left in Figure 14. Both

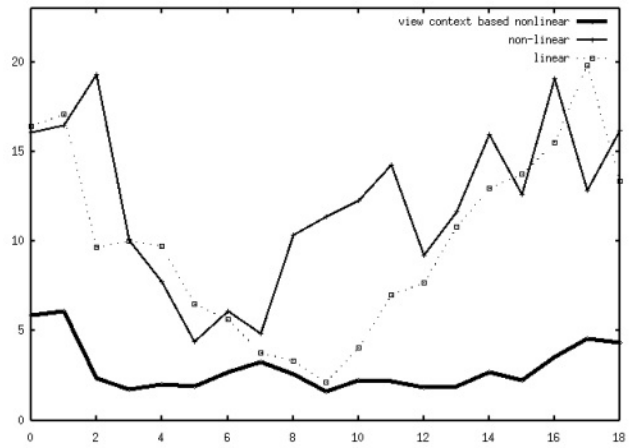


Figure 11: Comparing shape fitting errors across views. Typical fitting errors of different ASMs in pixels are drawn against pose. Whilst the dashed-line represents a linear ASM, the plain-line is for a nonlinear ASM and the bold-line for a view-context based nonlinear ASM.

Figure 12: Examples of fitting shapes to images at novel views using a view-context based nonlinear ASM.

Figure 13: An example of fitting shapes to images of an unknown face across views using a view-context based nonlinear ASM.

ASMs exhibit similar results which shows the generalisation ability of a view-context based nonlinear ASM to novel poses. A similar error comparison for generalisation to novel faces can be seen on the right in Figure 14.

# 5 Conclusion

In this work, we presented a novel approach to modelling nonlinear 2D shapes of non-rigid 3D objects and simultaneous recovering of object pose at multiple views and across the view sphere. Large pose variation in 3D objects such as human faces raise two difficult problems. First, shape variations across views are highly nonlinear. Second, correspondences of landmark points across views cannot be reliably established based solely on local grey-levels. The first problem was addressed by performing nonlinear shape transformation across views using Kernel PCA based on the concept of Support Vector Machines. The second problem was tackled by augmenting a nonlinear 2D active shape model with pose constraint.

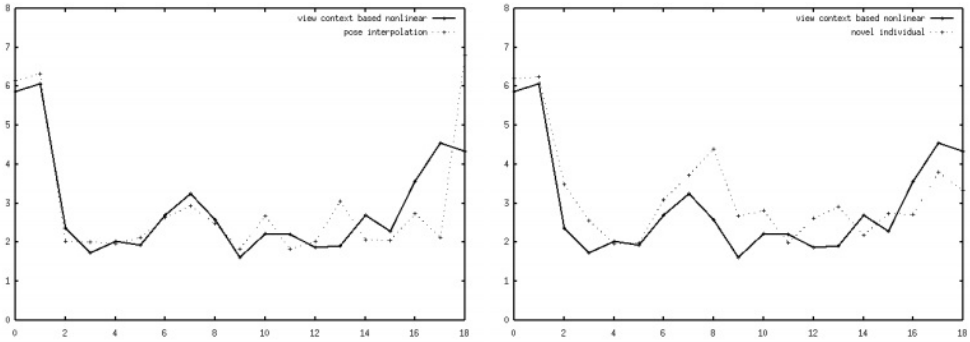


Figure 14: Left: Comparable fitting errors between a view-context based nonlinear ASM trained on all poses (plain-line) and a model trained only on half of the poses (dashed-line). Right: Comparable fitting errors for a model trained on all faces (plain-line) and a model trained only on some of the faces and tested on a novel face (dashed-line). The horizontal axis shows the pose and the vertical axis shows fitting errors in pixels.

# References


[1] R. Bowden, T. A. Mitchell, and M. Sarhadi. Reconstructing 3d pose and motion from a single camera view. In *BMVC*, pages 904–913, Southampton, UK, 1998.

[2] T. Cootes, A. Hill, C. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12:355–366, 1994.

[3] T. Cootes and C. Taylor. A mixture model for representing shape variation. In *BMVC*, pages 110–119, Essex, UK, 1997.

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

[5] F. de la Torre, S. Gong, and S. McKenna. View-based adaptive affine tracking. In *ECCV*, volume 1, pages 828–842, Freiburg, Germany, 1998.

[6] 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.

[7] T. Heap and D. Hogg. Improving specificity in pdms using a hierarchical approach. In *BMVC*, pages 80–89, Essex, UK, 1997.

[8] A. Lanitis, C. Taylor, T. Cootes, and T. Ahmed. Automatic interpretation of human faces and hand gestures using flexible models. In *FG*, pages 98–103, Zurich, 1995.

[9] S. Mika, B. Scholkopf, A. Smola, G. Ratsch, K. Muller, M. Scholz, and G. Ratsch. Kernel pca and de-noising in feature spaces. In *NIPSS*, 1998.

[10] E-J. Ong and S. Gong. A dynamic human model using hybrid 2d-3d representations in hierarchical pca space. In *BMVC*, Nottingham, UK, September 1999.

[11] E-J. Ong and S. Gong. Tracking hybrid 2d-3d human models through multiple views. In *IEEE International Workshop on Modelling People*, Corfu, Greece, September 1999.

[12] B. Scholkopf, S. Mika, A. Smola, G. Ratsch, and K. Muller. Kernel pca pattern reconstruction via approximate pre-images. In *ICANN*. Springer Verlag, 1998.

[13] B. Scholkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[14] V. Vapnik. *The nature of statistical learning theory*. Springer Verlag, 1995.