Review
Active Shape Models - Part I: Modeling Shape and Gray Level Variations

Ghassan Hamarneh, Rafeef Abu-Gharbieh and Tomas Gustavsson
Department of Signals and Systems, Imaging and Image Analysis Group,
Chalmers University of Technology, Göteborg, Sweden.

Abstract

In this paper we review and investigate a method capable of modeling the different appearance of objects in images that is due to natural shape variations, varying lighting conditions, 3D pose and others. Objects are represented by well-defined landmark points and shape variations are modeled using a principal component analysis. Also, gray level variations are being modeled. The first part of the paper describes the shape and gray scale modeling in some detail. The second part describes an iterative algorithm which deforms an initial model to fit data in ways that are consistent with shape variations found in previously acquired training data. An application to image classification is outlined.

1. Introduction

The purpose of this paper is to review and investigate a method referred to as Active Shape Models (ASM). ASM was originally proposed by Cootes *et al.* [1]. The modeling technique is similar in spirit to Active Contour Models, or snakes, proposed by Kass *et al.* [2], but the advantage of ASM is that instances of models can only deform in ways found in a training set. That is, they allow for considerable variability but are still specific to the class of objects or structures they intend to represent.

For several reasons, images of the same object class frequently display variation in appearance. For example, face images may vary due to individual look, facial expression, lighting conditions and 3D pose (see Figure 1)[3]. Robust recognition systems should be able to model these variations and reduce their effect in vision tasks such as search and classification.

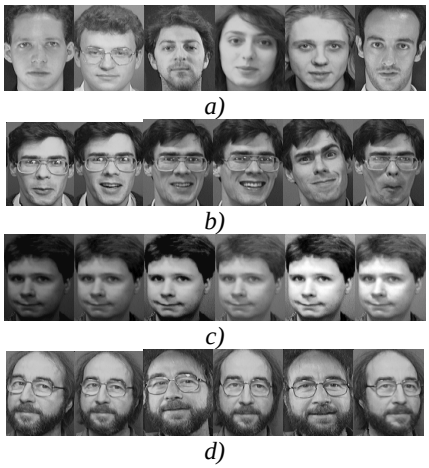


Figure 1. Variations due to a) individual look, b) facial expressions, c) lighting effects and d) 3D pose.

In ASM, an object shape is represented by a set of landmark points. Several instances of the same object class are included in a training set and in order to model the variations we need to align the set of shapes. Also, for extended application to image search and classification, we model the gray level information found in the training set.

2. Modeling Shape Variations

2.1. Training Set

In order to build a model that is flexible enough to cover the most typical variations of an object, a sufficiently large training set has to be used (see Figure 2). For the purpose of the investigation reported in this paper, a set of face images found in [3] were used.

Figure 2. Samples from a training set

2.2. Object Shape Representation

An object shape is represented by a set of labeled points or landmarks. The number of landmarks should be large enough to show the overall shape, and the details where it is needed (see Figure 3).

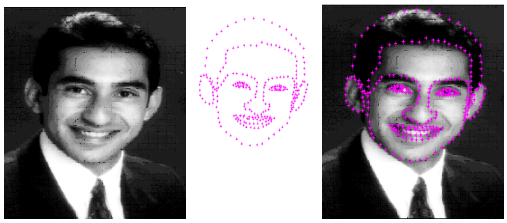


Figure 3. Example of a face image, its landmarks, and the image with landmarks overlaid.

2.3. Point Distribution Model

The model that will be used to describe a shape and its typical appearances is based on the variations of the spatial position of each landmark point within the training set. Each point will thus have a certain distribution in the image space and therefore the shape model is being referred to as a Point Distribution Model (PDM). In order to obtain the PDM, we first need to determine and label the landmarks, to align the shapes, and finally, to summarize the landmark variations in a compact form. In what follows, these steps are being described in some detail.

2.3.1. Labeling the Training Set

Before labeling the shapes of the training set, we need to determine the number of landmark points that can adequately represent the shape. In this study we apply a manual procedure. For each image of the training set, we locate the shape (by eye), and then identify significant landmarks on that shape. It is important that the landmarks are accurately located and that there is an exact correspondence between labels in different instances of training shapes.

There are at least three basic types of landmarks that can be used [1] :

1. Application-dependent landmarks.
2. Application-independent landmarks.
3. Landmarks interpolated from the two above.

Example of application-dependent landmarks in face images are the centers of the eyes. Application-independent landmarks may be the highest or lowest point of an object with a certain orientation. Interpolated landmarks can be points which are separated by equal distances and located along a certain path between two landmarks of type 1 or 2. Typically, type 3 will dominate and describe most of the boundary of the shape.

Our labeled training set is denoted $\mathcal{S}$. It contains N shapes, each of which has n landmarks. Put in another way, we have N coordinate points for each landmark of the shape. We denote the j^{th} landmark coordinate point of the i^{th} shape of the training set by (x_{ij}, y_{ij}), and the vector describing the n points of the i^{th} shape in the training set by:

$$\mathbf{x}_i = \left[x_{i0}, y_{i0}, x_{i1}, y_{i1}, \ldots, x_{in-1}, y_{in-1}\right]^T ;\ 1 \le i \le N .$$

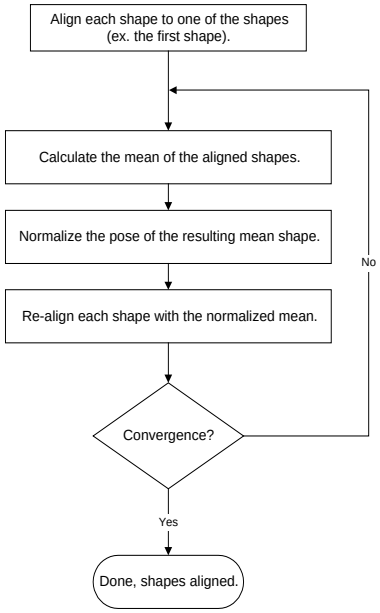
2.3.1.1. Aligning Two Shapes

In order to study the variations of the position of each landmark throughout the set of training images, all shapes, each of which is represented by its corresponding landmark vector $\mathbf{x}$, must be aligned to each other. This is done by changing the pose (scale, rotation, and translation) of consecutive shapes until the complete set is properly aligned.

We first introduce the problem of aligning two shapes or two landmark vectors. Given two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, we find the scaling value s, the rotation angle θ, and the value of translation in both dimensions (t_x, t_y), operating on the vector $\mathbf{x}_j$ so as to align it to $\mathbf{x}_i$ in a weighted least-squares sense. Weighting is applied to give more significance to the landmark points that tend to be more stable. The stability of a point is measured by the amount of variation in the distance between that point and the other points.

2.3.1.2. Aligning All Shapes

The following algorithm is used for aligning the set of N shapes to each other [1]:



The pose of a shape is described by its scaling, rotation, and translation, with respect to a known reference.

Normalization of the pose means a) *scaling* of the shape so that the distance between two points becomes a certain constant, b) *rotating* the shape so that the line joining two pre-specified landmarks is directed in a certain direction, and c) *translating* the shape so that it becomes centered at a certain coordinate. Normalization is carried out in order to force the process to converge, otherwise the mean shape may translate or expand (or shrink) indefinitely.

Convergence is established if the shapes are not changing more than a pre-defined threshold.

2.3.2. Capturing the Statistics

The i^{th} aligned shape of the training set of images is represented by the vector $\mathbf{x}_i$, where $\mathbf{x}_i$ now contains the new coordinates resulting from alignment. This vector is of dimension $2n$, so it can be represented by a point in a $2n$-dimensional space. The N vectors representing the N aligned shapes will then map to a 'cloud' of N points in the same $2n$-D space. It is assumed that these N points are contained within a region of this $2n$-D space referred to as the 'Allowable Shape Domain' (ASD). Every point in this region contributes to a shape that is similar to the other shapes in this ASD. As a measure of similarity, we

state that the shorter the Euclidean distance between two points (that represent two shapes) the more similar the shapes. The weighted Euclidean distance d between the two points representing the two shapes $\mathbf{x}_i$ and $\mathbf{x}_k$ is given by:

$$d_{ik} = \sqrt{(\mathbf{x}_i - \mathbf{x}_k)^T \mathbf{W} (\mathbf{x}_i - \mathbf{x}_k)}$$

where

$$\mathbf{x}_i = \left[x_{i0}, y_{i0}, x_{i1}, y_{i1}, \ldots, x_{in-1}, y_{in-1} \right]^T$$

$$\mathbf{W} = diag(w_o, w_o, w_1, w_1, \ldots, w_{n-1}, w_{n-1})$$

The weighting matrix $\mathbf{W}$ is used to give more importance to those landmark points that vary less in the training set.

Now, we wish to find the driving principals governing the behavior of the variations of the N points in the $2n$-D space defined by the $2n$ variables of $\mathbf{x}$. Applying Principal Component Analysis (PCA), we generate a new set of variables called the principal components. Each principal component is a linear combination of the original variables (Standardized Linear Combination (SLC) [4]). All the principal components are orthogonal to each other so there is no redundant information. The principal components as a whole form an orthogonal basis for the space of data [5]. The first principal component is a single axis in space. When projecting each of the observations (N vectors representing the shapes) on that axis, the resulting values form a new variable. The variance of this variable is the maximum among all possible choices of the first axis, *i.e.* represents the maximum shape variation. The second component is another axis in space, perpendicular to the first. Projecting our N observations on this axis generate another new variable. The variance of this variable is the maximum among all possible choices of this second axis. The dimension of both the full set of principal components and the original set of variables is the same hence both sets contain $2n$ variables.

In many applications it can be assumed that the first few principal components describe a high percentage of the total variance of the original data. Hence, the dimension of the model can be reduced and the variations described by a less number of variables (less than $2n$), that is performing what is referred to as "parsimonious summarization" [4], [5].

Now, we can express each landmark vector as a linear combination of the principal components. Moreover, we can express the difference between each vector and the mean of all vectors as a linear combination of the principal components, because this difference vector will also lie in the $2n$-D space spanned by the principal components.

Denoting the mean vector by $\bar{\mathbf{x}}$, and the difference vector between the vector $\mathbf{x}_i$ and $\bar{\mathbf{x}}$ by $d\mathbf{x}_i$, we have

$$d\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}} \quad \text{and} \quad \bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$$

The covariance matrix for the landmarks of the shapes is given by

$$\mathbf{C}_x = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

Representing the difference $d\mathbf{x}_i$ as a linear combination of the principal components, we write

$$d\mathbf{x}_i = b_{i0}\mathbf{p}_0 + b_{i1}\mathbf{p}_1 + \ldots + b_{i2n-1}\mathbf{p}_{2n-1}$$

where $\mathbf{p}_l$ is the l^{th} principle component axis or vector and b_{il} is a scalar that weighs $\mathbf{p}_l$. We normalize to unit length, *i.e.* $\mathbf{p}_l{}^T \mathbf{p}_l = 1$, and because the principal components are also mutually orthogonal, they are orthonormal, *i.e.*

$$\mathbf{p}_l{}^T \mathbf{p}_m = \begin{cases} 1 & l = m \\ 0 & l \neq m \end{cases}$$

Equivalently, we can write

$\mathbf{x}_i = \bar{\mathbf{x}} + d\mathbf{x}_i$ and rewrite $d\mathbf{x}_i = \mathbf{P}\mathbf{b}_i$, where

$$\mathbf{b}_i = \begin{bmatrix} b_{i0} & b_{i1} & \ldots & b_{i2n-1} \end{bmatrix}^T \text{ and}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \ldots & \mathbf{p}_{2n-1} \end{bmatrix}$$

This yields $\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}_i$, and $\mathbf{b}_i$ can be found as $\mathbf{b}_i = \mathbf{P}^{-1}(\mathbf{x}_i - \bar{\mathbf{x}})$.

With $\mathbf{P}$ being an orthogonal matrix (a square matrix with orthonormal columns) [6], we have $\mathbf{P}^{-1} = \mathbf{P}^T$, and so $\mathbf{b}_i = \mathbf{P}^T(\mathbf{x}_i - \bar{\mathbf{x}})$.

To summarize, we have N landmark vectors having a mean $\bar{\mathbf{x}}$. Each such vector can be expressed as the sum of the mean vector and a weighted sum of the principle components. If we choose the weights as: $\mathbf{b}_i = \mathbf{P}^T(\mathbf{x}_i - \bar{\mathbf{x}})$; $1 \le i \le N$, we will end up with the known shape $\mathbf{x}_i$, but if we choose other weights, such as $\mathbf{b} = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}})$ where $\mathbf{x} \notin \{\mathbf{x}_1 \quad \mathbf{x}_2 \quad \ldots \quad \mathbf{x}_N\}$, we will get a shape not found in the training set. The degree of similarity between this shape and those in the training set is determined by the distance d_{ik}. If we want the vector of weights $\mathbf{b}$ to be chosen so that the resulting shape can be considered an acceptable or allowable shape, then we must put certain constraints on the these weights. If we limit b_l to: $b_{l\,\min} \le b_l \le b_{l\,\max}$, for $0 \le l \le 2n-1$, then it is suitable to choose $b_{l\,\min} = -b_{l\,\max}$, and $b_{l\,\max}$ proportional to the variance of the training set's projection along the l^{th} principle component.

The principle components can be obtained by eigen value decomposition of the covariance matrix $\mathbf{C}_x$ [1], or by singular value decomposition (SVD) of the observation matrix containing the difference vectors $d\mathbf{x}_i$ [7].

Recalling our aim of this analysis, which is to reduce the dimension of the data and describe the variations with a fewer number of variables, we now express the N landmark vectors as the sum of their mean $\bar{\mathbf{x}}$ and a weighted sum of *some* of the principal components. We assume that the first t (out of $2n$) principal components explain a sufficiently high percentage of the total variance of the original data. By this we are saying that

the *2n*-D cloud of shapes has a 'small' width in the direction of the $t+1$ and higher principle components.

Considering only *t* principle components, our basic equation becomes, $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pb}$, where

$$\mathbf{b} = \begin{bmatrix} b_0 & b_1 & \dots & b_{t-1} \end{bmatrix}^T$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \dots & \mathbf{p}_{t-1} \end{bmatrix}$$

and the limits on $\mathbf{b}$ become
$b_{k\,\min} \le b_k \le b_{k\,\max}$, for $0 \le k \le t-1$.

3. Modeling the Gray Level Appearance

Here we will discuss how we can model the gray level information in the training set of images. The main idea is to examine the gray levels in a region around each landmark throughout the training set. In general, any region around a landmark can be studied, but here we will concentrate on the gray levels along a line passing through the landmark.

For every landmark point *j* in the image *i* of the training set, we extract a gray level profile $\mathbf{g}_{ij}$, of length n_p pixels, centered around the landmark point. We do not use the actual gray level profile but its normalized derivative. This gives invariance to the offsets and uniform scaling of the gray levels [8].

The gray level profile of the landmark *j* in the image *i* is a vector of n_p values,

$$\mathbf{g}_{ij} = \begin{bmatrix} g_{ij0} & g_{ij1} & \dots & g_{ijn_p-1} \end{bmatrix}^T,$$

and the derivative profile of length $n_p - 1$ becomes

$$d\mathbf{g}_{ij} = \begin{bmatrix} g_{ij1} - g_{ij0} & g_{ij2} - g_{ij1} & \dots & g_{ijn_p-1} - g_{ijn_p-2} \end{bmatrix}$$ The normalized derivative profile is given by

$$\mathbf{y}_{ij} = \frac{d\mathbf{g}_{ij}}{\sum_{k=0}^{n_p-2} \left| d\mathbf{g}_{ijk} \right|}$$

Now, we calculate the mean of the normalized derivative profiles of each landmark throughout the training set, and we get for landmark *j*

$$\bar{\mathbf{y}}_j = \frac{1}{N} \sum_{i=1}^{N} d\mathbf{g}_{ij}$$

The covariance matrix of the normalized derivative is given by

$$\mathbf{C}_{\mathbf{y}_j} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{y}_{ij} - \bar{\mathbf{y}}_j)(\mathbf{y}_{ij} - \bar{\mathbf{y}}_j)^T$$

With this we obtain a model for the gray levels around any landmark *j* represented by $\bar{\mathbf{y}}_j$ and $\mathbf{C}_{\mathbf{y}_j}$.

4. Conclusions

Active Shape Models can be used for accurate modeling of shape and gray level appearance. Although the model has much less dimensions as compared to the original shape data it can still allow for considerable amount of variability while at the same time being specific to the class of objects or structures to be represented.

Acknowledgment

This study was supported by the Swedish Foundation for Strategic Research under the VISIT program.

References

[1] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active Shape Models - Their Training and Application. Computer Vision and Image Understanding, January 1995, Vol. 61, No. 1, pp. 38-59.

[2] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active Contour Models, In Proceedings, First International Conference On Computer Vision, pp. 259-268, IEEE Comput. Soc. Press, 1987.

[3] The ORL Database of Faces, http://www.cam-orl.co.uk/facedatabase.html.

[4] K. Mardia, J. Kent, J. Bibby, Multivariate Analysis, Academic Press, 1995, pp. 213-254.

[5] J. Bradley, MATLAB Statistics Toolbox - User's Guide, The MathWorks Inc., 1997, Tutorial pp. 77-87.

[6] G. Strang. Linear Algebra And Its Applications, Saunders 1988, pp. 167.

[7] MATLAB version 5.1, and the 'rudimentary statistics toolbox' more specifically the functions 'princomp(…)' and 'svd(…)'. MATLAB's reference: J. Edward Jackson, A User's Guide to Principal Components, John Wiley & Sons, Inc. 1991 pp. 1-25, B. Jones 3-17-94.

[8] T. Cootes, C. Taylor, A. Hill, J. Halsam, The Use of Active Shape Models for Locating Structures in Medical Images. Proceedings of the 13th International Conference on Information Processing in Medical Imaging, (Eds. H.H.Barrett, A.F.Gmitro) Springer-Verlag, 1993, pp. 33-47.