Active Appearance Models

Timothy F. Cootes, Gareth J. Edwards, and
Christopher J. Taylor

Abstract—We describe a new method of matching statistical models of appearance to images. A set of model parameters control modes of shape and gray-level variation learned from a training set. We construct an efficient iterative matching algorithm by learning the relationship between perturbations in the model parameters and the induced image errors.

Index Terms—Appearance models, deformable templates, model matching.

1 Introduction

The "interpretation through synthesis" approach has received considerable attention over the past few years [3], [6], [11], [14]. The aim is to "explain" novel images by generating synthetic images that are as similar as possible, using a parameterized model of appearance. One motivation is to achieve robust segmentation by using the model to constrain solutions to be valid examples of the class of images modeled. A suitable model also provides a basis for a broad range of applications by coding the appearance of a given image in terms of a compact set of parameters that are useful for higher-level interpretation of the scene.

Suitable methods of modeling photo-realistic appearance have been described previously, e.g., Edwards et al. [3], Jones and Poggio [9], or Vetter [14]). When applied to images of complex and variable objects (e.g., faces), these models typically require a large number of parameters (50-100). In order to interpret novel images, an efficient method of finding the best match between model and image is required. Direct optimization in this high-dimensional space is possible using standard methods but it is slow [9] and/or liable to become trapped in local minima.

We have shown previously [11], [3] that a statistical model of appearance can be matched to an image in two steps: First, an Active Shape Model is matched to boundary features in the image, then a separate eigenface model is used to reconstruct the texture (in the graphics sense) in a shape-normalized frame. This approach is not, however, guaranteed to give an optimal fit of the appearance model to the image because small errors in the match of the shape model can result in a shape-normalized texture map that cannot be reconstructed correctly using the eigenface model.

In this paper, we show an efficient direct optimization approach that matches shape and texture simultaneously, resulting in an algorithm that is rapid, accurate, and robust. In our method, we do not attempt to solve a general optimization problem each time we wish to fit the model to a new image. Instead, we exploit the fact that the optimization problem is similar each time so we can learn these similarities offline. This allows us to find directions of rapid convergence even though the search space has very high dimensionality. The approach has similarities with "difference decomposition" tracking algorithms [7], [8], [13], but rather than tracking a single deforming object we match a model which can fit a whole class of objects.

- *T.F. Cootes and C.J. Taylor are with the Division of Imaging Science and Biomedical Engineering, University of Manchester, M13 9PT, UK. E-mail: {t.cootes, chris.taylor}@man.ac.uk.*
- *G.J. Edwards is with Image-Metrics Ltd., Regent House, Heaton Lane, Stockport, Cheshire, Sk4 1BS, UK. E-mail: gareth.edwards@image-metrics.com.*

Manuscript received 5 Apr. 2000; revised 10 Nov. 2000; accepted 7 Jan. 2001. Recommended for acceptance by M. Shah.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 111825.

2 Background

Several authors have described methods for matching deformable models of shape and appearance to novel images. Nastar et al. [12] describe a model of shape and intensity variations using a 3D deformable model of the intensity landscape. They use a closest point surface matching algorithm to perform the fitting, which tends to be sensitive to the initialization. Lanitis et al. [11] and Edwards et al. [3] use a boundary finding algorithm (an "Active Shape Model") to find the best shape, then use this to match a model of image texture. Poggio et al. [6], [9] use an optical flow algorithm to match shape and texture models iteratively, and Vetter [14] uses a general purpose optimization method to match photorealistic human face models to images. The last two approaches are slow because of the high dimensionality of the problem and the expense of testing the quality of fit of the model against the image.

Fast model matching algorithms have been developed in the tracking community. Gleicher [7] describes a method of tracking objects by allowing a single template to deform under a variety of transformations (affine, projective, etc.). He chooses the parameters to minimize a sum of squares measure and essentially precomputes derivatives of the difference vector with respect to the parameters of the transformation. Hager and Belhumeur [8] describe a similar approach, but include robust kernels and models of illumination variation. Sclaroff and Isidoro [13] extend the approach to track objects which deform, modeling deformation using the low energy modes of a finite element model of the target. The approach has been used to track heads [10] using a rigid cylindrical model of the head.

The Active Appearance Models described below are an extension of this approach [4], [1]. Rather than tracking a particular object, our models of appearance can match to any of a class of deformable objects (e.g., any face with any expression, rather than one persons face with a particular expression). To match rapidly, we precompute the derivatives of the residual of the match between the model and the target image and use them to compute the update steps in an iterative matching algorithm. In order to make the algorithm insensitive to changes in overall intensity, the residuals are computed in a normalized reference frame.

3 Modeling Appearance

Following the approach of Edwards et al. [3], our statistical appearance models are generated by combining a model of shape variation with a model of texture variation. By "texture" we mean the pattern of intensities or colors across an image patch. To build a model, we require a training set of annotated images where corresponding points have been marked on each example. For instance, to build a face model, we require face images marked with points defining the main features (Fig. 1). We apply Procrustes analysis to align the sets of points (each represented as a vector, $\mathbf{x}$) and build a statistical shape model [2]. We then warp each training image so the points match those of the mean shape, obtaining a "shape-free patch" (Fig. 1). This is raster scanned into a texture vector, $\mathbf{g}$, which is normalized by applying a linear transformation, $\mathbf{g} \rightarrow (\mathbf{g} - \mu_g \mathbf{1})/\sigma_g$, where $\mathbf{1}$ is a vector of ones, μ_g and σ_g^2 are the mean and variance of elements of $\mathbf{g}$. After normalization, $\mathbf{g}^T\mathbf{1} = 0$ and $|\mathbf{g}| = 1$. Eigen-analysis is applied to build a texture model. Finally, the correlations between shape and texture are learned to generate a combined appearance model (see [3] for details).

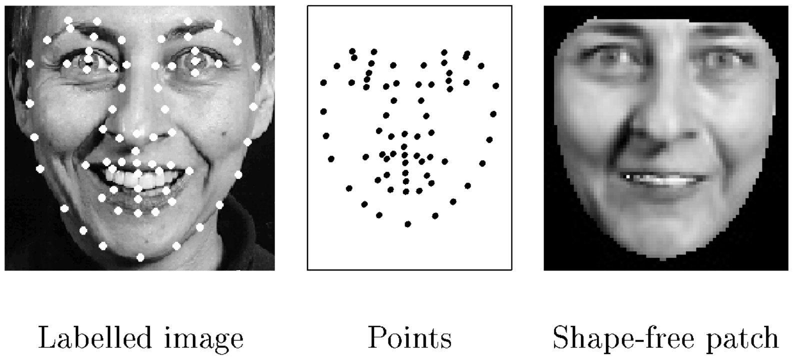

Fig. 1. A labeled training image gives a shape free patch and a set of points.

The appearance model has parameters, $\mathbf{c}$, controlling the shape and texture (in the model frame) according to

$$\begin{aligned}\mathbf{x} &= \bar{\mathbf{x}} + \mathbf{Q}_s\mathbf{c}\\ \mathbf{g} &= \bar{\mathbf{g}} + \mathbf{Q}_g\mathbf{c},\end{aligned} \tag{1}$$

where $\bar{\mathbf{x}}$ is the mean shape, $\bar{\mathbf{g}}$ the mean texture in a mean shaped patch, and $\mathbf{Q}_s$, $\mathbf{Q}_g$ are matrices describing the modes of variation derived from the training set.

A shape, $\mathbf{X}$, in the image frame can be generated by applying a suitable transformation to the points, $\mathbf{x} : \mathbf{X} = S_{\mathrm{t}}(\mathbf{x})$. Typically, S_{t} will be a similarity transformation described by a scaling, s, an in-plane rotation, θ, and a translation (t_x, t_y). For linearity, we represent the scaling and rotation as (s_x, s_y), where $s_x = (s\cos\theta - 1)$, $s_y = s\sin\theta$. The pose parameter vector $\mathbf{t} = (s_x, s_y, t_x, t_y)^T$ is then zero for the identity transformation and $S_{\mathrm{t}+\delta\mathrm{t}}(\mathbf{x}) \approx S_{\mathrm{t}}(S_{\delta\mathrm{t}}(\mathbf{x}))$.

The texture in the image frame is generated by applying a scaling and offset to the intensities, $\mathbf{g}_{im} = T_{\mathrm{u}}(\mathbf{g}) = (u_1 + 1)\mathbf{g}_{im} + u_2\mathbf{1}$, where $\mathbf{u}$ is the vector of transformation parameters, defined so that $\mathbf{u} = \mathbf{0}$ is the identity transformation and $T_{\mathbf{u}+\delta\mathbf{u}}(\mathbf{g}) \approx T_{\mathbf{u}}(T_{\delta\mathbf{u}}(\mathbf{g}))$.

A full reconstruction is given by generating the texture in a mean shaped patch, then warping it so that the model points lie on the image points, $\mathbf{X}$. In experiments described below, we used a triangulation-based interpolation method to define a continuous mapping.

In practice, we construct a multiresolution model based on a Gaussian image pyramid. This allows a multiresolution approach to matching, leading to improved speed and robustness. For each level of the pyramid, we build a separate texture model, sampling a number of pixels proportional to the area (in pixels) of the target region in the image. Thus, the model at level 1 has one quarter the number of pixels of that at level 0. For simplicity, we use the same set of landmarks at every level and the same shape model. Though strictly the number of points should reduce at lower resolutions, any blurring of detail is taken account of by the texture model. The appearance model at a given level is then computed given the global shape model and the particular texture model for that level.

3.1 An Appearance Model for Faces

We used the method described above to build a model of facial appearance. We used a training set of 400 images of faces, each labeled with 68 points around the main features (Fig. 1), and sampled approximately 10,000 intensity values from the facial region. From this, we generated an appearance model which required 55 parameters to explain 95 percent of the observed variation. Fig. 2 shows the effect of varying the first four parameters from $\mathbf{c}$, showing changes in identity, pose, and expression. Note the correlation between shape and intensity variation.

4 Active Appearance Model Search

We now turn to the central problem addressed by this paper: We have a new image to be interpreted, a full appearance model as described above and a rough initial estimate of the position, orientation, and scale at which this model should be placed in an image. We require an efficient scheme for adjusting the model parameters, so that a synthetic example is generated, which matches the image as closely as possible.

4.1 Modeling Image Residuals

The appearance model parameters, $\mathbf{c}$, and shape transformation parameters, $\mathbf{t}$, define the position of the model points in the image frame, $\mathbf{X}$, which gives the shape of the image patch to be represented by the model. During matching, we sample the pixels in this region of the image, $\mathbf{g}_{im}$, and project into the texture model frame, $\mathbf{g}_s = T_{\mathbf{u}}^{-1}(\mathbf{g}_{im})$. The current model texture is given by $\mathbf{g}_m = \bar{\mathbf{g}} + \mathbf{Q}_g\mathbf{c}$. The current difference between model and image (measured in the normalized texture frame) is thus

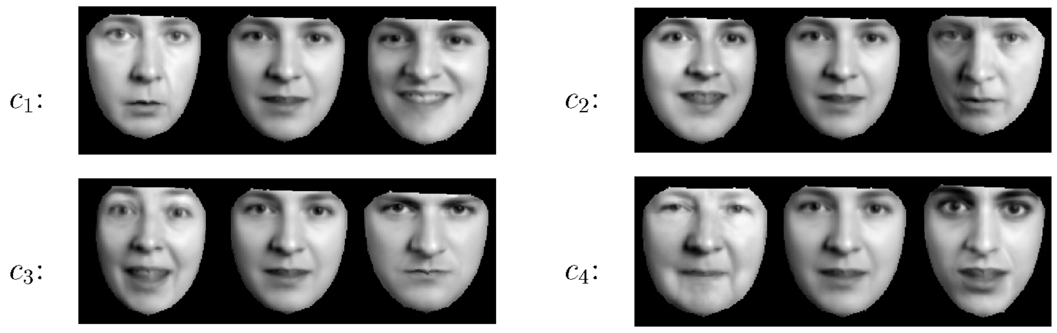

Fig. 2. Effect of varying first four facial appearance model parameters, $c_1 - c_4$ by ± 3 standarad deviations from the mean.

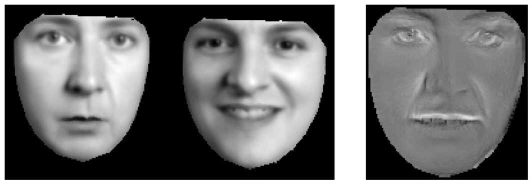
Fig. 3. First mode and displacement weights.

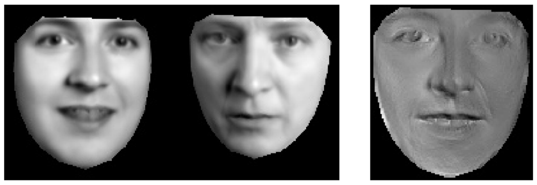
Fig. 4. Second mode and displacement weights.

$$\mathbf{r}(\mathbf{p}) = \mathbf{g}_s - \mathbf{g}_m, \tag{2}$$

where $\mathbf{p}$ are the parameters of the model, $\mathbf{p}^T = (\mathbf{c}^T|\mathbf{t}^T|\mathbf{u}^T)$.

A simple scalar measure of difference is the sum of squares of elements of $\mathbf{r}$, $E(\mathbf{p}) = \mathbf{r}^T\mathbf{r}$. A first order Taylor expansion of (2) gives

$$\mathbf{r}(\mathbf{p}+\delta\mathbf{p}) = \mathbf{r}(\mathbf{p}) + \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\delta\mathbf{p}, \tag{3}$$

where the *ij*th element of matrix $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$ is $\frac{dr_i}{dp_j}$.

Suppose during matching our current residual is $\mathbf{r}$. We wish to choose $\delta\mathbf{p}$ so as to minimize $|\mathbf{r}(\mathbf{p}+\delta\mathbf{p})|^2$. By equating (3) to zero, we obtain the RMS solution,

$$\delta\mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p}) \;\; where \;\; \mathbf{R} = \left(\frac{\partial \mathbf{r}^T}{\partial \mathbf{p}} \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\right)^{-1} \frac{\partial \mathbf{r}^T}{\partial \mathbf{p}}. \tag{4}$$

In a standard optimization scheme, it would be necessary to recalculate $\frac{\partial \mathbf{p}}{\partial \mathbf{p}}$ at every step, an expensive operation. However, we assume that since it is being computed in a normalized reference frame, it can be considered approximately fixed. We can thus estimate it once from a training set. We estimate $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$ by numeric differentiation. The *j*th column is computed by systematically displacing each parameter from the known optimal value on typical images and computing a weighted average of the residuals. To improve the robustness of the measured estimate, residuals at displacements of differing magnitudes, δp_{jk}, are measured (typically up to 0.5 standard deviations, σ_j, for each parameter, p_j) and summed with a weighting proportional to $\exp(-\delta p_{jk}^2/2\sigma_j^2)/\delta p_{jk}$. We then precompute $\mathbf{R}$ and use it in all subsequent searches with the model.

Images used in the calculation of $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$ can either be examples from the training set or synthetic images generated using the appearance model itself. Where synthetic images are used, one can either use a suitable (e.g., random) background, or can detect the areas of the model which overlap the background and remove those pixels from the model building process. This latter makes the final relationship more independent of the background. Where the background is predictable (e.g., medical images), this is not necessary.

4.2 Iterative Model Refinement

Using (4), we can suggest a correction to make in the model parameters based on a measured residual $\mathbf{r}$. This allows us to construct an iterative algorithm for solving our optimization problem. Given a current estimate of the model parameters, $\mathbf{c}$, the pose $\mathbf{t}$, the texture transformation $\mathbf{u}$, and the image sample at the current estimate, $\mathbf{g}_{im}$, one step of the iterative procedure is as follows:

1. Project the texture sample into the texture model frame using $\mathbf{g}_s = T_{\mathbf{u}}^{-1}(\mathbf{g}_{im})$.
2. Evaluate the error vector, $\mathbf{r} = \mathbf{g}_s - \mathbf{g}_m$, and the current error, $E = |\mathbf{r}|^2$.
3. Compute the predicted displacements, $\delta\mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p})$.
4. Update the model parameters $\mathbf{p} \rightarrow \mathbf{p} + k\delta\mathbf{p}$, where initially $k = 1$.
5. Calculate the new points, $\mathbf{X}'$ and model frame texture $\mathbf{g}'_m$.
6. Sample the image at the new points to obtain $\mathbf{g}'_{im}$.
7. Calculate a new error vector, $\mathbf{r}' = T_{\mathbf{u}'}^{-1}(\mathbf{g}'_{im}) - \mathbf{g}'_m$.
8. If $|\mathbf{r}'|^2 < E$, then accept the new estimate; otherwise, try at $k = 0.5$, $k = 0.25$, etc.

This procedure is repeated until no improvement is made to the error, $|\mathbf{r}|^2$, and convergence is assumed. In practice, we use a multiresolution implementation, in which we start at a coarse resolution and iterate to convergence at each level before projecting the current solution to the next level of the model. This is more efficient and can converge to the correct solution from further away than searching at a single resolution. The complexity of the algorithm is $O(n_{pixels}n_{modes})$ at a given level. Essentially, each iteration involves sampling n_{pixels} points from the image then multiplying by a $n_{modes} \times n_{pixel}$ matrix.

In earlier work [4], [1], we described a training algorithm based on regressing random displacement vectors against residual error vectors. Because of the linear assumptions being made, in the limit this would give the same result as above. However, in practice, we have found the above method to be faster and more reliable.

5 An Active Appearance Model for Faces

We applied the Active Appearance Model training algorithm to the face model described in Section 3.1. The elements of a row of the matrix $\mathbf{R}$ give the weights for each texture sample point for a given model parameter. Figs. 3 and 4 show the first two modes of the appearance model and the corresponding weight images. The areas which exhibit the largest variations for the mode are assigned the largest weights by the training process.

We used the face AAM to search for faces in previously unseen images. Fig. 5 shows frames from an AAM search for two faces, each starting with the mean model displaced from the true face center. In each case, a good result is found. The search takes less than one second on a 400MHz Pentium II PC.

6 Quantitative Evaluation

To obtain a quantitative evaluation of the performance of the AAM algorithm, we trained a model on 100 hand-labeled face images and tested it on a different set of 100 labeled images. Each face was about 200 pixels wide. A variety of different people and expressions were included. The AAM was trained by systematically displacing the model parameters on the first 10 images of the training set.

On each test image, we systematically displaced the model from the true position by ± 10 percent of the face width in x and y, and changed its scale by ± 10 percent. We then ran the multiresolution search, starting with the mean appearance parameters. 2,700 searches were run in total. Of these, 13 percent failed to converge to a satisfactory result (the mean point position error was greater than 5 percent of face width per point). Of those that did converge, the RMS error between the model points and the target points was about 0.8 percent of face width. The mean magnitude of the final texture error vector in the normalized frame, relative to that of the best model fit given the marked points, was 0.84 (sd: 0.2), suggesting that the algorithm is locating a better result than that provided by the marked points. This is to be expected since

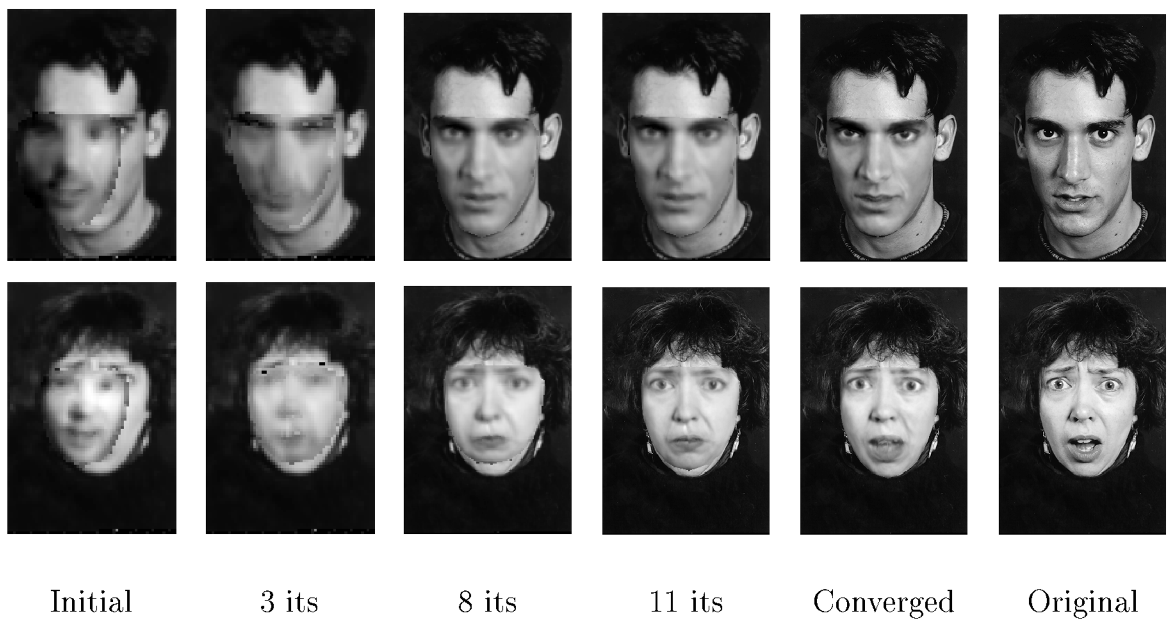

Fig. 5. Multiresolution search from displaced position using face model.

the algorithm is minimizing the texture error vector and will compromise the shape if that leads to an overall improvement in texture match.

Fig. 6 shows the mean intensity error per pixel (for an image using 256 gray-levels) against the number of iterations, averaged over a set of searches at a single resolution. The model was initially displaced by 5 percent of the face width. The dotted line gives the mean reconstruction error using the hand-marked landmark points, suggesting a good result is obtained by the search.

Fig. 7 shows the proportion of 100 multiresolution searches which converged correctly given various starting positions displaced by up to 50 pixels (25 percent of face width) in x and y. The model displays good results with up to 20 pixels (10 percent of the face width) initial displacement.

7 Discussion and Conclusions

We have demonstrated an iterative scheme for matching a Statistical Appearance Model to new images. The method makes use of learned correlation between errors in model parameters and the resulting residual texture errors. Given a reasonable initial starting position, the search converges rapidly and reliably. The approach is valuable for locating examples of any class of objects that can be adequately represented using the statistical appearance model described above. This includes objects which exhibit shape and texture variability for which correspondences can be defined between different examples. For instance, the method cannot be used for tree like structures with varying numbers of branches, but can be used for various organs which exhibit shape variation but not a change in topology. We have tested the method extensively with images of faces and with medical images [1], [5] and have obtained encouraging results in other domains. Although an AAM search is slightly slower than Active Shape Model search [2], the procedure tends to be be more robust than ASM search alone, since all the image evidence is used.

The algorithm has been described for gray-level images but can be extended to color images simply by sampling each color at each sample point (e.g., three values per pixel for an RGB image). The nature of the search algorithm also makes it suitable for tracking objects in image sequences, where it can be shown to give robust results [5].

Acknowledgments

Dr. Cootes is funded under an EPSRC Advanced Fellowship Grant. During much of this work, Dr. Edwards was funded by an EPSRC CASE award with BT Research Laboratories.

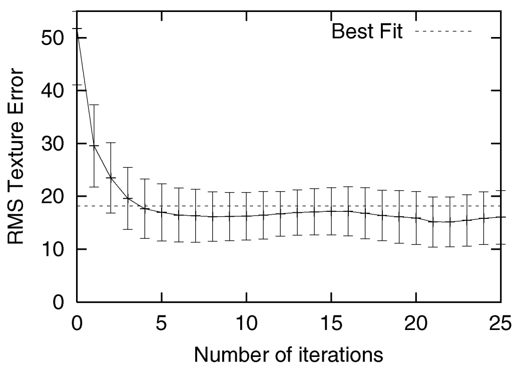

Fig. 6. Mean intensity error as search progresses. Dotted line is the mean error of the best fit to the landmarks.

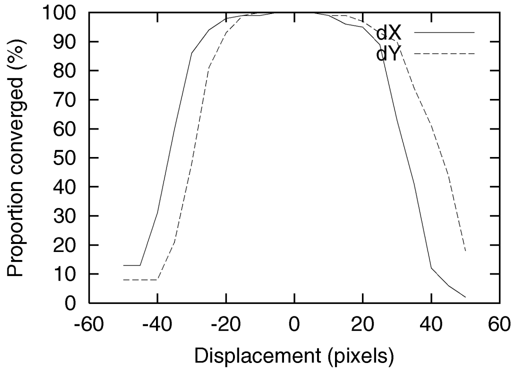

Fig. 7. Proportion of searches which converged from different initial displacements.

REFERENCES

[1] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Models," *Proc. Fifth European Conf. Computer Vision,* H. Burkhardt and B. Neumann, eds., vol. 2, pp. 484-498, 1998.

[2] T.F. Cootes, C.J. Taylor, D. Cooper, and J. Graham, "Active Shape Models—Their Training and Application," *Computer Vision and Image Understanding,* vol. 61, no. 1, pp. 38-59, Jan. 1995.

[3] G. Edwards, A. Lanitis, C. Taylor, and T. Cootes, "Statistical Models of Face Images—Improving Specificity," *Image and Vision Computing,* vol. 16, pp. 203-211, 1998.

[4] G. Edwards, C.J. Taylor, and T.F. Cootes, "Interpreting Face Images Using Active Appearance Models," *Proc. Third Int'l Conf. Automatic Face and Gesture Recognition,* pp. 300-305, 1998.

[5] G.J. Edwards, C.J. Taylor, and T.F. Cootes, "Face Recognition Using the Active Appearance Model," *Proc. Fifth European Conf. Computer Vision,* H. Burkhardt and B. Neumann, eds., vol. 2, pp. 581-695, 1998.

[6] T. Ezzat and T. Poggio, "Facial Analysis and Synthesis Using Image-Based Models," *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition 1997,* pp. 116-121, 1997.

[7] M. Gleicher, "Projective Registration with Difference Decomposition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1997.

[8] G. Hager and P. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 10, pp. 1025-1039, Oct. 1998.

[9] M.J. Jones and T. Poggio, "Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes," *Int'l J. Computer Vision,* vol. 2, no. 29, pp. 107-131, 1998.

[10] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture Mapped 3D Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 4, pp. 322-336, Apr. 2000.

[11] A. Lanitis, C.J. Taylor, and T.F. Cootes, "Automatic Interpretation and Coding of Face Images using Flexible Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 743-756, July 1997.

[12] C. Nastar, B. Moghaddam, and A. Pentland, "Generalized Image Matching: Statistical Learning of Physically-Based Deformations," *Computer Vision and Image Understanding,* vol. 65, no. 2, pp. 179-191, 1997.

[13] S. Sclaroff and J. Isidoro, "Active Blobs," *Proc. Sixth Int'l Conf. Computer Vision,* pp. 1146-1153, 1998.

[14] T. Vetter, "Learning Novel Views to a Single Face Image," *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition,* pp. 22-27, 1996.