3D head tracking using non-linear optimization

James Paterson, Andrew Fitzgibbon
Department of Engineering Science
University of Oxford
jamie@robots.ox.ac.uk, awf@robots.ox.ac.uk

Abstract

Accurate and reliable tracking of the 3D position of human heads is a continuing research problem in computer vision. This paper addresses the specific problem of *model-based* tracking with a generic deformable 3D head model. Following the work of Vetter and Blanz, a collection of head models is obtained from a 3D scanner, registered and parameterized to give a *generic* head model which is linearly parameterized by a small number of parameters. This is the 3D analogue of Cootes and Taylor's active appearance models. We cast tracking as a parameter estimation problem, and note that many existing solutions to the problem—such as CONDENSATION and Kalman filtering—are analogous to nonlinear optimization strategies in numerical analysis. We show how careful analysis of the error function, parameterization of the model pose parameters, and choice of optimizer allows us to robustly track 3D head pose in digital video camera footage of quickly moving heads.

1 Introduction

The detection and tracking of human faces within a video sequence is a common objective in computer vision. Accurate and reliable head tracking has applications in animation, in user interface design, for biometrics, and as a measurement modality for the physiological sciences. Of current techniques, the most successful at recovering the 3D position are model-based—the tracking problem is cast as one of estimating the parameters of a deformable model which best fits the input video sequence. In this paper, we develop a morphable face model which encodes both the 3D shape and the 2D texture of a range of faces, and show how accurate and robust tracking can be achieved by combining an illumination-invariant image comparison metric with well-engineered nonlinear optimization techniques.

The morphable model we use is similar in nature to active appearance models [9], as specialized to 3D face models by Vetter and Blanz [24]. This model parameterizes full texture mapped 3D models of human faces, the parameterization learned from a collection of 3D scans of real faces. By performing an offline registration process on a set of 3D scanned faces we obtain a set of basis models and texture maps which can be linearly blended to produce new models not in the original set. In contrast to $2\frac{1}{2}$D models, for example coupled [8, 5], or view-based [21] models, these are full 3D models which allow 3D manipulations such as relighting, collision detection, and change of viewpoint. Section 3 describes how we acquire these models. To the best of the authors' knowledge, this

BMVC 2003 doi:10.5244/C.17.63

is the first replication of the Vetter and Blanz results, and the exposition of the techniques we have employed in order to do so is one contribution of this paper.

The second contribution is in the use of the model to track the 3D position and orientation of heads in video. We define tracking as the problem of determining the model parameters (both 3D pose and the parameters of the linear combination of basis faces) which best describe each image of a video sequence. By using mutual information [25, 15] as the similarity metric which compares the predicted and imaged face, robustness to changes in lighting and colour are immediately achieved. We experimentally compare mutual information as a metric to simple comparison of pixel difference, and to Roche's correlation ratio [20]. Finally, the use of a general-purpose nonlinear optimizer in preference to gradient-descent or random search strategies provides for robust tracking even with significant movement of the head from frame to frame.

We compare several nonlinear optimization strategies and similarity metrics, and demonstrate robust head tracking on video sequences of two subjects.

2 Background

Beymer, Shashua and Poggio were among the first develop a system for the alignment and synthesis of 2D images of faces [3], later being extended by Jones and Poggio to the matching of faces as well as other classes of objects such as cars [18]—more recently Ezzat, Geiger, and Poggio used a variation on this technique to provide generation of realistic 'newsreader' type footage, synchronized with audio [13]. Cootes et al's active shape models [10] are the basis for many of these techniques, and the recent development of 'coupled-view' active appearance models [8] allows "$2\frac{1}{2}$D" modelling of heads using correlated 2D views. As an example of the application of this technology, Devin and Hogg combine an Active Appearance Model with audio recognition to create an 'interactive talking head' [12]. Another example of fully 3D face modelling is the popular CANDIDE family of parameterized meshes. Much matching related work has been carried out using these systems, with Ahlberg and Forchheimer providing a recent example [1].

Model-based tracking—the problem of determining the parameters of some model which best "predict" a sequence of observations such as a video sequence—has been a topic of perennial interest throughout computer vision. An incomplete selection of examples includes the tracking of vehicles [14] and the human body [17, 23, 16]. In this paper, we use a texture-mapped 3D model as the data representation, so our "predictions" are synthetic 2D renderings of the head, rendered using desktop OpenGL hardware. These are compared to the input video via an *similarity metric* which measures the similarity between the rendered and target images. Simple metrics such as the sum of pixel-by-pixel intensity differences used by Vetter and Blanz [24] are very sensitive to differences in lighting between the synthetic rendering and the target scene, so a more robust measure is required. Viola and Wells first introduced the mutual information metric with the aim of aligning similar images, including obtaining the pose of a human head in an image [25]—this was then extended to full multi-modal 2D image registration [26]. Gilles provides an excellent technical report detailing the implementation of mutual information [15]. With the success of mutual information came much analysis of its performance and potential problems—the "correlation ratio" suggested by Roche et al [20] takes advantage of spatial information in the *intensity space* of the images being compared to provide favourable

behaviour on multi-modal data sets.

Much work has been conducted by the computer vision community on face tracking, with emphasis gradually moving from 2D matching approaches to full 3D model based systems. Schödl and Haro [22] provide an early example of the use of a rigid, texture mapped model. DeCarlo and Metaxas [11] were among the first to apply a deformable 3D model based on anthropometric data, extracting shape and motion via optical flow. Other authors have used coarse head models such as cylinders [7, 6] or superquadrics [28].

The tracking strategy in this paper is an explicit nonlinear minimization of the error metric over the model parameters. This is a natural generalization of "predictor-corrector" strategies such as the extended Kalman filter [23] or CONDENSATION [17]. By choosing explicit minimization strategies which have been well honed by the numerical analysis community, we benefit from wide convergence and robust operation of standard numerical algorithms [19]. In this paper, the multi-dimensional function we wish to optimize is not easily differentiated analytically, so derivative-free optimizers such as the downhill simplex method introduced by Nelder and Mead, and Powell's direction set method will be used. Such techniques have been employed previously for tracking [27] but their equivalence to more traditional tracking methodologies is rarely emphasised, and the use of algorithms more sophisticated than steepest descent is rare.

3 Building the morphable model

The 3D model we use is a triangulated mesh, as is widely used in computer graphics and accelerated in modern graphics hardware. The model is defined by a collection of F 3D vertices $\{\mathbf{X}_0 \dots \mathbf{X}_{F-1}\}$, and a single 2D texture image $T(i,j)$ of N^2 RGB pixels. Each vertex $\mathbf{X}_i$ has a corresponding 2D *texture coordinate* $\mathbf{U}_i$. The 3D triangles are defined as triples of vertex indices, and all models have the same triangulation topology, although different models have different vertices and texture maps. A 3D model is represented as a point in a vector space by simply concatenating the vertices, their texture coordinates, and the pixels of the texture map into a long vector denoted $\mathbf{E} = [\mathbf{X}_0 \dots \mathbf{X}_{F-1}, \mathbf{U}_0 \dots \mathbf{U}_{F-1}, T(0,0) \dots T(N,N)]$. For a typical model, $F \approx 5000$, $N = 512$, so the dimensionality of $\mathbf{E}$, at $5F + 3N^2$, is of the order of 10^6. We define a morphable model as a linear combination of *basis* models $\mathbf{E}_0 ... \mathbf{E}_B$, with model coefficients $\alpha_{0..B}$, given by

$$\mathbf{E}(\alpha_0, .., \alpha_B) = \sum_{q=0}^{B} \alpha_q \mathbf{E}_q \quad \text{where} \sum_{q=0}^{B} \alpha_q = 1 \tag{1}$$

Clearly, for this linear combination to produce sensible novel faces, the triangulations of the basis models must correspond. This means that a vertex with a given index must always represent the same facial feature, for example the tip of the nose. In practice, models scanned from real-world subjects (such as the output from a cylindrical Cyberware head scanner) are supplied with an arbitrary triangulation which varies from model to model, making registration of the models essential.

3.1 Registration of raw models

The input to model construction is a set of five Cyberware scanned 3D head models, obtained from the Max-Planck-Institut, as used by Vetter and Blanz [4, 24]. These are

supplied as triangulated meshes with single texture maps, and are in roughly the same 3D coordinate systems, but with vertices which are not in correspondence. To find this correspondence, we take advantage of the fact that each 3D vertex has a corresponding 2D texture map coordinate. To produce a set of vertices which are in correspondence, we first place the texture maps in correspondence, and then define a new 2D triangulation in texture coordinates. For each vertex $\mathbf{U}_i$ of this 2D triangulation, its 3D coordinates $\mathbf{X}_i$ can be found by interpolation of the original 3D model, as discussed below.

To place the texture maps in correspondence, one model is chosen as the reference. The texture map of each other model is then warped so that it is in correspondence with that of the reference model. While previous work [24] achieved such correspondence using a modified optical flow algorithm, our experience has been that such techniques can be unreliable, and require significant manual optimization to produce useful results. Instead, we formally require manual input: a set of 30 feature points is defined on the human face, and these points manually marked on each input texture map. This allows for consistency across a variety of faces and is easily done for each scanned face. A 2D warp using radial basis function interpolation [2] between these control points is then used to bring each texture map into alignment with the reference texture map.

With all the texture maps in correspondence, a new triangulation of the 3D vertices is defined by selecting F points (2D) on the reference texture map and computing the Delaunay triangulation. This triangulation is defined just once, so can be optimized manually if required. The final step is to generate 3D vertices for each of these 2D points. The 3D coordinates corresponding to each 2D point can be found by inverse-warping the texture coordinates into the texture map of the original model. Each point on the original texture map is associated with a 3D point on a triangle of the original model, so its 3D position can be easily determined.

Combining the above steps gives a set of texture-mapped 3D models which are in vertex-for-vertex correspondence. This allows a large range of faces to be generated via equation 1. The next step is to determine the parameters of this generic model which best match specific examples which are observed in new 2D images.

4 Using the model for tracking

The task of tracking is to take input video footage of a moving face, and to determine, for each frame of the video, the model parameters which best match the face's position and shape in that video frame. Here the degrees of freedom are the 3D position and orientation of the head, the five model shape parameters $\alpha_0, \ldots, \alpha_B$, and the focal length of the camera.

Our aim is to construct an off-line tracking system which is automated except for some manually defined starting conditions. The tracking system should work on 'real-life' footage, such as the first two sequences shown in figure 2. Here the sequences were captured at normal PAL resolution using a digital camcorder, having various uncalibrated lighting sources, including fluorescent and natural light, and a complex background setting.

The general tracking framework is to maximize the similarity between a rendered model image and the target video. The rendering process takes a vector of model parameters, denoted $\mathbf{S}$, which encodes all the degrees of freedom of the model and produces a

rendered image $R(\mathbf{S})$. The key to successful tracking is then to define a *similarity function* $\varepsilon(R,I)$ which measures the similarity between the rendered image R and the target video image I. Several choices of ε are possible, and these are now discussed. In the following, all images are assumed to be grayscale luminance images for ease of exposition.

4.1 Image similarity metrics

The most straightforward metric is **pixel difference** (PD), defined simply as the summed difference in per-pixel luminance, defined as

$$PD(R,I) = \kappa - \sum_{\mathbf{x}\in\mathbb{R}^2} (I(\mathbf{x}) - R(\mathbf{x}))^2$$

where κ is a large positive constant. This metric, as used in [24], is high when the images are similar, and low otherwise. However, it is extremely sensitive to lighting variation—two renderings of the same face under different lighting conditions can have very different luminances at any given pixel.

Mutual information (MI) is more resistant to lighting changes, and is calculated in terms of individual and joint entropy between two images and as such is an information-theoretic approach. The entropy of a probability distribution $P(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^d$, is computed from a discrete histogram. If the histogram bins are $\{h(\mathbf{x}_i)\}_{i=1}^{N^d}$, the entropy is defined as $E(h) = -\sum_i h(\mathbf{x}_i)\log h(\mathbf{x}_i)$. To compare images R and I, we compute the 1D histograms h_R and h_I for each image and the 2D histogram $h_{R,I}$ of pixels from the two-channel image obtained by superimposing R and I. Then the mutual information [15] is

$$MI(R,I) = E(h_R) + E(h_I) - E(h_{R,I}) \tag{2}$$

Correlation ratio (CR), introduced by Roche et al [20], works on the basis that a *functional* relationship exists between luminance values in the two images so that $I = \Phi(R)$ for some unknown scalar function Φ. Consider again the comparison of two images R and I. If the images are correctly registered, we expect each different intensity in R to map to a small cluster of intensities in I. CR works by first finding the function Φ^* which best fits I to R, then evaluates the quality of the fitting. For brevity we omit a full derivation (discussed in detail in [20]), and simply state the definition:

$$CR(I|R) = 1 - \frac{Var(I - \Phi^*(R))}{Var(I)} \tag{3}$$

where $Var(K)$ denotes the variance of an image K.

4.1.1 Comparison of similarity metrics

The success of our tracking strategy depends on choosing an image similarity metric which is high when the model parameters, $\mathbf{S}$, are correct, independent of lighting changes. It is also necessary that when the model parameters are slightly different from the correct value, the metric should degrade gracefully, reducing slowly for larger deviations from the true position. Figure 1 compares the three metrics on a synthetic test. In this test, a face is rendered at the identity set of model parameters to make the "target" image I. Then the two elements of the model parameter vector $\mathbf{S}$ corresponding to X and Y rotation

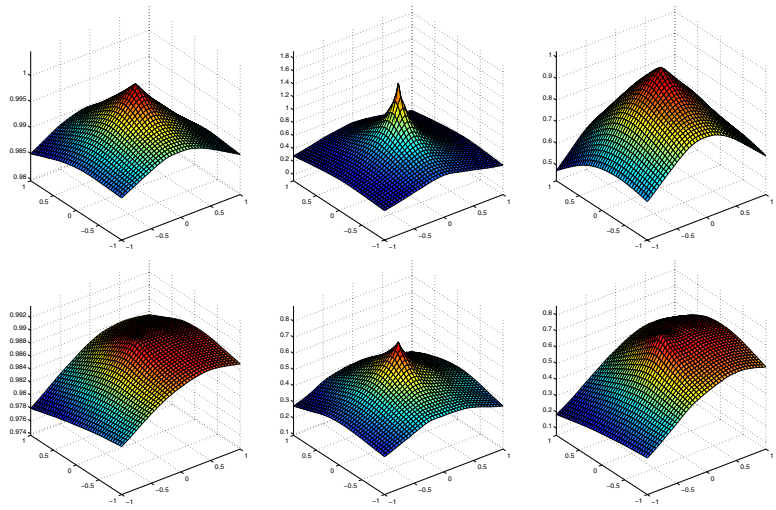

Figure 1: **Metric behaviour under different lighting conditions** (Left to right) Plots of Pixel Difference, Mutual Information and Correlation Radio metric's behaviour, comparing an identity image of the model versus renderings of the model with varying rotation about the X and Y axis. The top row is under 'correct' lighting conditions, the bottom with 'incorrect'. Under correct lighting, all perform well. Under incorrect lighting, only MI falls off uniformly as the rotation increases.

(nodding and shaking of the head) are varied systematically. For each value of R_x and R_y, a rendered image $R(\mathbf{S})$ is computed and the value $\varepsilon(R,I)$ is computed for each of the three metrics. Only MI has the desired properties of a unique global maximum and graceful decay with increasing deviation from the true position under general lighting conditions.

4.2 Tracking as nonlinear optimization

Tracking is the problem of determining, for each new frame of a sequence, the set of model parameters which best describes the new frame. Tracking assumes that the estimate of model parameters in the last frame can be used to predict the model position in the current frame, and that this prediction can be corrected by minimizing an error criterion which measures the deviation between the predicted model position and the observed one. With minimization of error replaced by maximisation of a similarity function, this is a similar problem to the goal of nonlinear optimization: to maximise a multidimensional function, $f(\mathbf{S})$, given an initial starting position $\mathbf{S}^0$. This is an enormous research field, and we refer the reader to [19] for an accessible introduction. In this work we restrict our attention to optimization strategies which do not require the computation of function derivatives, as the similarity metrics we optimize do not permit analytic derivatives to be computed. Of the many methods available, Powell's conjugate direction method and the Nelder-Mead simplex method (sometimes called "Amoeba") are powerful, well understood techniques. On functions with long narrow "valleys" in parameter space, they require fewer function evaluations in general than gradient descent, even when derivatives

are available.

5 Implementation

The tracking system can now be summarized. At each new video frame, I, we have an initial estimate of the face position, parameterized by a vector of parameters $\mathbf{S}_0$. Defining the objective function $f(\mathbf{S})$ as the similarity between a rendered image $R(\mathbf{S})$ and I, we nonlinearly optimize f over the model parameters $\mathbf{S}$. The optimal value of $\mathbf{S}$ is then used as the initial estimate for the next frame. The next section shows that this is a very successful strategy on long, fast-moving video sequences. Several additional details of the system were important to its success, and these are described here.

Texture extraction: Although the linear combination as specified above (eq 1) blends both the 3D shape and the 2D texture map of each basis model, for the purposes of this work we use a fixed texture map mapped onto the shape created via linear combination of $\mathbf{E}(\alpha_0, .., \alpha_B)$. The fixed texture map is acquired from three reference images of the person, by first manually positioning (i.e: rotating, translating and scaling) the 3D model to align with each image, and then backprojecting from the reference images onto the 3D model. Noting each triangular face of the 3D model has a corresponding 2D triangle in the texture map, a mapping can be constructed between the face pixels in each reference image and the pixels in the texture map, using the position of the image projected 3D triangles of the aligned mesh. Thus, each texture map pixel is simply a blend of the pixel colours from each reference image in which the projected position of the texture map pixel is visible.

Parameterization: The explicit parameters being optimised by our tracking system are as follows: 3 for translation, 4 rotational (a quaternion so as to avoid gimbal lock), 5 model parameters $\alpha_0, \ldots, \alpha_4$, and a parameter to match the view angle of the renderer to that of the capture camera. This parameter is multiplied by the translation Z in order to decouple view angle and zoom, which creates narrow valleys in the error surface.

Narrow valleys also result if the model origin is placed unwisely. We improved performance by scaling the Z translational parameter, and moving the centre of the model such that the rotation was about the nose rather than the centre of mass.

Background modelling: The morphable model itself can only model the area of the input images containing face, and so during non-linear optimisation we draw a background image behind the rendering of the morphable model, so that the rendered images R more closely resemble the target image I. This background frame can be easily acquired by having the subject move out of the view of the camera. Note that background subtraction is not a good solution as it will tend to confuse the rendering both inside and outside the head area.

6 Experiments

Four sequences of head motion were captured and processed, all from real-life subjects but two under highly controlled conditions (bottom row, figure 2). Each sequence began with a fronto-parallel view and so, after defining poses for left, right and first frames, texture maps could be extracted and the sequence run though the various combinations of image metrics and optimizers using the texture mapped model.

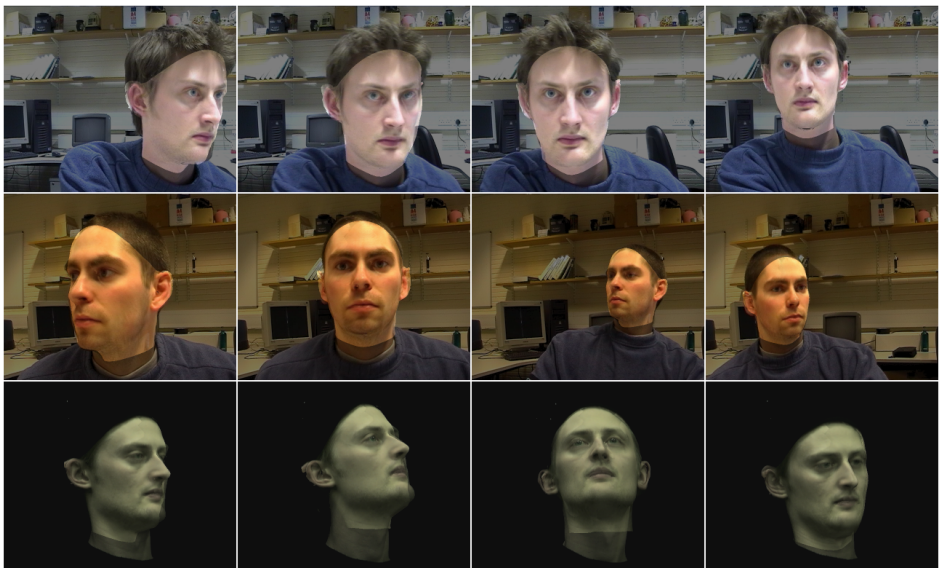

Figure 2: **Example frames from successful tracks** The morphable model is rendered overlaid on the target frame using the pose and model output from the tracker. Top and middle sequences were tracked using MI and Powell's method, the bottom using downhill simplex method and CR, a combination which offers better performance in controlled environment.

Careful analysis of the function being optimized on these sequences yielded some important observations. Whilst globally smooth, the function is very noisy at a micro scale, confusing numerical derivative calculation. In addition, long valleys could be seen which would lead to many iterations of gradient-based or random-search-based strategies.

The results of tracking on selected frames from the example sequences are shown in figure 2, and the reader is strongly encouraged to consult the attached video files in order to confirm that these are representative frames. Note that these sequences were tracked offline, so the user is not obtaining tracking feedback during capture, and that one of the sequences is of a subject who had not used the system before.

Local optima: The most striking performance characteristic of the system is the effect of local maxima in the objective function. Occasionally the model will remain at the same place for a number of frames, and then snap back onto the correct track. This shows two things: first, that under strong variations in lighting or head pose (examples are shown in fig 3), the similarity metrics will be higher for incorrect model positions than for the true position. These incorrect interpretations tend to be stable, so the tracking degrades gracefully in difficult situations. Encouragingly, when the head is returned to a less difficult position, tracking resumes without the need for specialized restart strategies.

Comparison of similarity metrics: In the context of our tracking application, we found that the pixel difference metric generally leads to poor performance. The two more sophisticated metrics gave varying results depending on the specific sequence, with CR being more suited to the controlled environment sequences, and MI to the real life ones.

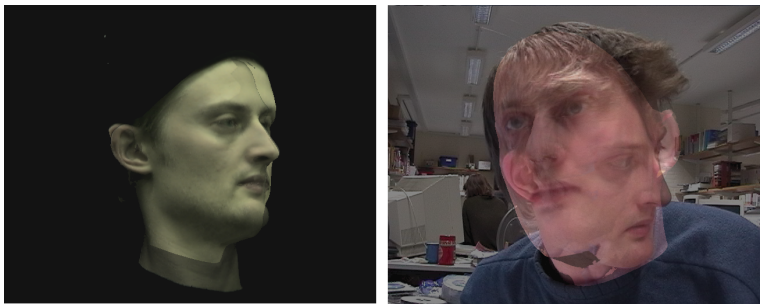

Figure 3: **Failure cases for MI and CR metrics** In each image, the morphable model is rendered overlaid on the target frame using the pose and model parameters output from the tracker. (Left) MI tends to lead to poor performance on 'controlled environment' sequences at profile views—note the zoomed-in views, especially the double forehead! (Right) CR attempts to match hair and neck areas on real-world sequences.

MI has difficulty with cases such as profile views, whilst CR attempts to scale the model to include hair and neck from the subject. Figure 3 gives examples. We observe that the greatest challenge for a comparison metric is handling changes in appearance due to lighting — even having the subject look up or down can change the surface illumination of the face a great deal. Applying an appropriate extracted texture to the model was essential to achieving good performance, perhaps partly making up for the lack of rigorous lighting estimation.

Rendering speed: A major issue is the time required per function evaluation. Currently all metric calculation is performed in the main CPU, requiring grabbing each rendering into main memory from the graphics card. With this expensive overhead, typical execution time is of the order of five to ten minutes *per frame*. This is a common issue with current graphics hardware, but is expected to improve with driver technology.

6.1 Conclusions and extensions

This paper has shown that the use of gradient-free nonlinear optimizers as the compute engine of a model-based tracker allows for robust and stable tracking which can recover well from local optima. The experimental results show sequences with significant translation, rotation, and depth variation. In addition to pose estimation, we obtain a set of model parameters in each frame giving shape information. Another positive feature of our approach is that the tracker can often recover from badly tracked frames, as long as the model is kept within a searchable distance of the subject.

Further work is required in expanding our model database and providing automatic initialization of the tracker. Although the current system performs well given that only five scanned models are used, shape estimation remains inaccurate. With a larger morphable model containing more faces there is evidence that this estimation of shape could be quite accurate, and separate texture extraction might no longer be necessary [4]. Initialization via the detection of facial features and the use of a generic face detector is also an area where significant increases in robustness can be gained. The speed of the system is limited

by slow transfer rates between the graphics hardware and main memory, however it may be possible to compute the similarity metric on the programmable hardware of modern graphics cards. A small empirical test of rendering a large number of model poses (with and without copying the image back into main memory) suggests eliminating this transfer would provide a speedup of two orders of magnitude.

References

[1] J. Ahlberg and R. Forchheimer. Face tracking for model-based coding and face animation. In *IJIST*, 2003.

[2] N. Arad, N. Dyn, D. Reisfeld, and Y. Yeshurun. Image warping by radial basis functions: Application to facial expressions. *Computer Vision, Graphics and Image Processing*, 56(2):161–172, 1994.

[3] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. Technical Report AIM-1431, MIT AI Lab, 1993.

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, 1999.

[5] R. Bowden, T. A. Mitchell, and M. Sarhadi. Non-linear statistical models for the 3d reconstruction of human pose and motion from monocular image sequences. *Image and Vision Computing*, 18(9):729–737, 2000.

[6] L. Brown. 3D head tracking using motion adaptive texture-mapping. In *CVPR*, pages 998–1005, 2001.

[7] M. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. In *IEEE PAMI*, volume 22, pages 322–336, 2000.

[8] T. Cootes, G. Wheeler, K. Walker, and C. Taylor. Coupled-view active appearance models. In *BMVC*, volume 1, pages 52–61, 2000.

[9] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. ECCV*, volume 2, pages 484–498, 1998.

[10] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *CVIU*, 61(1):38–59, 1995.

[11] D. DeCarlo and D. N. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *IJCV*, 38(2):99–127, 2000.

[12] V. Devin and D. Hogg. Reactive memories: An interactive talking-head. In *BMVC*, 2001.

[13] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic facial animation. In *SIGGRAPH*, pages 388–398, 2002.

[14] J. M. Ferryman, A. D. Worrall, and S. J. Maybank. Learning enhanced 3D models for vehicle tracking. In *BMVC*, pages 187–196, 1998.

[15] S. Gilles. Description and experimentation of image matching using mutual information. Technical Report, Dept. of Engineering Science, University of Oxford, 1996.

[16] T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *Intl. Conf. on Automatic Face and Gesture Recognition*, pages 140–145, 1996.

[17] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. ECCV*, pages 343–356, 1996.

[18] M. J. Jones and T. Poggio. Model-based matching by linear combinations of prototypes. In *Proceedings of the 1997 Image Understanding Workshop*, pages 1357–1365, 1997.

[19] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[20] A. Roche, G. Malandain, X. Pennec, and N. Ayache. The correlation ratio as a new similarity measure for multimodal image registration. *Lecture Notes in Computer Science*, 1496:1115–1124, 1998.

[21] S. Romdhani, S. Gong, and A. Psarrou. Multi-view nonlinear active shape model using kernel PCA. In *BMVC*, pages 13–16, 1999.

[22] I. Schödl and A. Haro. Head tracking using a textured polygonal model. In *In Proceedings of Workshop on Perceptual User Interfaces*, 1998.

[23] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model based 3D tracking of an articulated hand. In *CVPR*, pages 310–315, 2001.

[24] T. Vetter and V. Blanz. Estimating colour 3D face models from a single image: An example based approach. In *Proc. ECCV*, pages 499–513, 1998.

[25] P. Viola and W. Wells. Alignment by maximization of mutual information. In I. C. S. Press, editor, *Proc. ICCV*, pages 16–23, 1995.

[26] W. Wells, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. In *Medical Image Analysis*, volume 1, pages 35–51, 1996.

[27] A. D. Worrall, G. D. Sullivan, and K. D. Baker. Pose refinement of active models using forces in 3D. In *Proc. ECCV*, pages 341–350, 1994.

[28] Y. Zhang and C. Kambhamettu. Robust 3D head tracking under partial occlusion. *Pattern Recognition*, 35:1545–1557, 2002.