

Pose-Independent Face Identification from Video Sequences

Michael C. Lincoln and Adrian F. Clark
VASE Laboratory, University of Essex
Colchester CO4 3SQ, UK
{mclinc,alien}@essex.ac.uk

Abstract

A scheme for pose-independent face recognition is presented. An “unwrapped” texture map is constructed from a video sequence using a texture-from-motion approach, which is shown to be quite accurate. Simple lighting normalization methods improve robustness to directional and/or varying lighting conditions. Recognition of single frames against calculated unwrapped textures is carried out using principal component analysis. The system is typically better than 90% correct in its identifications.

1 Introduction

Face recognition is currently a particularly active area of computer vision. Although work on face analysis was performed as long ago as the 1970s [1], current interest was arguably inspired by the “eigenfaces” technique [2]. Subsequent workers have applied a wide variety of approaches, including various types of neural networks [3], hidden Markov models [4] and shape analysis [5].

The vast majority of face recognition techniques, including all those listed above, concentrate on full-face imagery. This is partly because such a constraint simplifies the problem and partly because typical current applications are for situations in which the subject is cooperative. There has been work on face recognition from profile imagery [6] but the more general problem in which the head orientation is unknown remains relatively unexplored. Full 3D face recognition is touched on in [7] and considered in more detail in [8]. The area in which face recognition technology arguably has the most potential is in policing, where full-face imagery is rarely available. Hence, *pose-independent* schemes are of practical value.

To be able to perform pose-independent face recognition, one ideally would have images of subjects captured at all possible orientations. This is not a tractable solution; but it is easy to consider an “image” that is a projection of the head shape onto a notional *cylinder* rather than onto a plane. We term this an *unwrapped texture map*. Our scheme involves taking each image (planar projection) in a video sequence, tracking the head from frame to frame and determining the head orientation in each frame, then merging the appropriate region of the image into the unwrapped texture map. If the head exhibits a reasonable amount of motion, a fairly complete texture map can be accumulated.

There are similarities between the texture tracker described herein and that reported in [9], though the two were developed independently. However, there are some impor-

tant differences. [9] used a *difference decomposition* optimization method for determining pose changes between frames. Iterative simplex optimization is used here; although computationally more expensive, results suggest that it is more robust. Furthermore, our scheme, although sub-real-time on current PC-class hardware, does not have to pause for lengthy off-line calculations at any stage during tracking. Finally, [9] did not attempt identification. This paper extends our earlier work [10] by including a correction for the effects of illumination.

Identification schemes applied to conventional, planar images can be exploited on unwrapped texture maps, though care is needed. For example, Kanade-like distances between interior features can be used [1], as can eigen-based approaches, as used here. Most importantly however, one can compare a single frame of a person’s head with a portion of a texture map to achieve identification.

The remainder of this paper is organized as follows. The construction of an unwrapped texture map, the most important component of the scheme, is described in Sec 2. Normalization to accommodate changes in illumination is discussed in Sec 3. The use of these textures in an eigenfaces-like identification scheme is discussed in Sec 4. Conclusions are drawn in Sec 5.

2 Construction of an Unwrapped Texture Map

2.1 Preliminaries

A 3D surface model of the head being tracked is required in order to evaluate the corresponding texture. An accurate model of the head is not required, though poor models are likely to affect the accuracy and stability of tracking. This work employs a tapered ellipsoid as a user-independent head model; this is a simple shape to control and, as it is convex, means that hidden surface removal can be accomplished by back-face culling [11].

In computer graphics, a 2D texture is normally applied to a 3D model. Associated with each vertex in each facet of a 3D model is a 2D texture coordinate. The rendering process then determines the appearance of each screen pixel for each facet by interpolating between the texture coordinates of the vertices. However, this technique requires the reverse operation: values are inserted *into* the texture map when the image positions of the projections of vertices of the head model have been determined. Our implementation of this uses OpenGL, which allows this process to be carried out in hardware, even on PC-class systems.

As explained above, not every pixel in the texture map will have the same accuracy. Hence, each pixel in the constructed texture map has a corresponding confidence value (forming a *confidence map*). This is modelled as the ratio between the area of a pixel in texture space and the area in screen space that gave rise to it. These confidence values are central to the way in which image data are merged into the unwrapped texture map.

Finally, a measure of the similarity between two textures is required. The measure used here is

$$\sqrt{\frac{\sum_{uv} C_{min}^2(u, v) d^2(u, v)}{\sum_{uv} C_{min}^2(u, v)}} \quad (1)$$

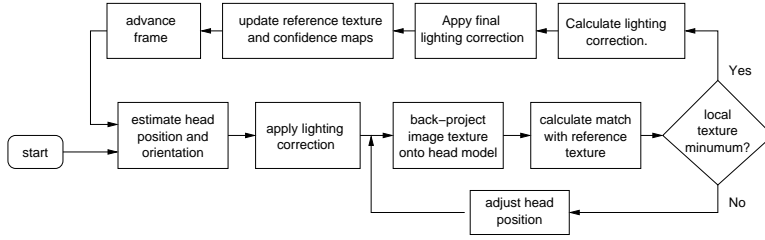


Figure 1: Procedure for constructing unwrapped texture map

where $d(u, v)$ is the sum-squared difference between textures for the pixel at (u, v) and $C_{min}(u, v)$ is the minimum confidence value for the same pixel.

2.2 Tracking by Optimization

The position and orientation of the head in the first frame of a sequence is currently specified manually, though this could be automated. As outlined above, the “reference” unwrapped texture and confidence maps are initialized from the image. The procedure for accumulating the texture and confidence maps from subsequent frames is illustrated in Fig 1. An estimate for the head’s new position and orientation is made; this can be simply the same as in the previous frame, though some prediction scheme (*e.g.*, Kalman filtering) is probably better. The head model is transformed to this new position and the image texture *back-projected* onto it, facet by facet. A match with the reference head texture is then performed. The six position and orientation parameters of the head model are adjusted using a simplex optimization scheme until the best (smallest) match value is obtained.

With the optimum parameters found, the back-projected texture for the current frame is merged into the reference texture map. A pixel in the texture map is updated only if it will result in a higher confidence value: if C_r is the confidence of a pixel in the reference image and C_i the corresponding value for the current image, then

$$W_r = \frac{C_r}{C_r + C_i} \quad W_i = \frac{C_i}{C_r + C_i} \quad (2)$$

and, providing $C_i > C_r$, the texture map value V_r is updated to

$$V_r = V_r W_r + V_i W_i \quad (3)$$

where V_i is the value of texture map for the current image.

This procedure is illustrated in Fig 2, which shows a single frame of a video sequence. Superimposed at the top right of the frame are the unwrapped texture and confidence maps extracted from that frame, and superimposed at the top left are the corresponding maps accumulated over the entire sequence to date. Note that, for performance reasons, the accumulated texture is constructed without back-face culling and hence appears twice at slightly differing magnifications. The confidence map, however, does employ back-face culling. Fig 3 illustrates the accuracy of the tracker with and without lighting correction (see next section) on the uniformly lit “Boston” dataset used in [9]. They captured position

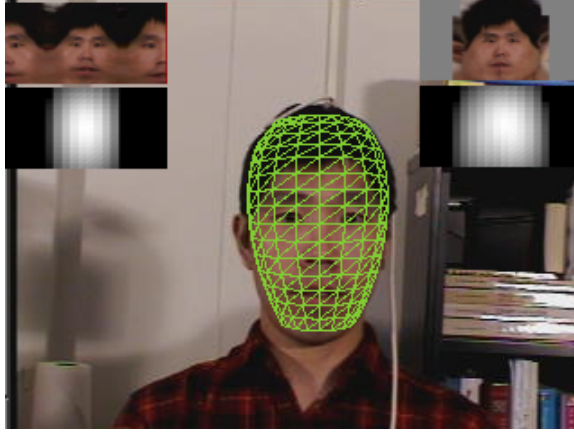


Figure 2: The construction of texture maps by tracking

and orientation information of subjects' heads as they moved using a magnetic tracker. As expected lighting correction makes little difference to uniformly lit video sequences.

Compared to the magnetic tracker data, the RMS positional and orientation errors from this texture tracker are 3.5 cm and 2.8° respectively. These compare well with previous results on the Boston dataset. (Of course, there is no guarantee that the data from the magnetic tracker are correct.)

3 Lighting Normalization

Strong directional and varying light sources can adversely affect tracking. Two steps are used to normalize the luminosity of the textures. These can be done in a computationally efficient manner by making the assumption that illumination varies slowly compared to the frame rate of the video. This assumption is easily justified by plotting the correction parameters against time.

Global lighting changes are accommodated by calculating the mean and standard deviation of the texture's luminosity component, given by equations 4, 5, and 6 below. These are used to adjust the image of the next frame. By adjusting the whole image, not just the texture, this need be done only once per frame rather than for every iteration of the head location phase.

After the new head location has been found, the texture is regenerated from the original image, new statistics are calculated, and these are used to adjust the texture's mean and contrast (standard deviation) prior to merging with the reference texture map. The use of standard deviation rather than range provides robustness to the "salt and pepper" noise common in video signals.

$$i_{(u,v)} = \frac{r_{(u,v)} + g_{(u,v)} + b_{(u,v)}}{3} \quad (4)$$

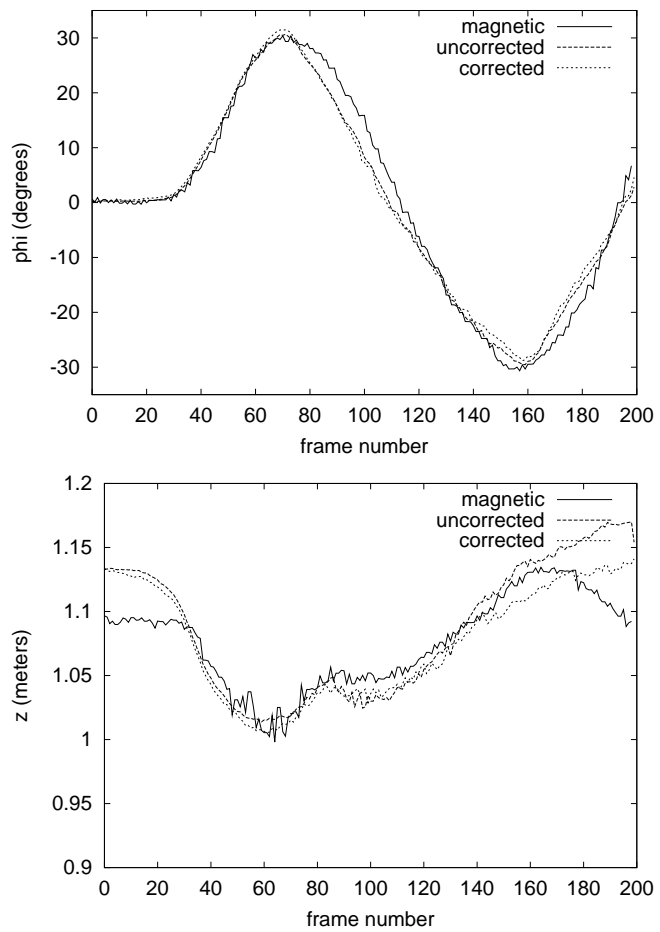


Figure 3: Head-tracking accuracy under uniform lighting

correction	frames tracked	sequences completed	distance error	angular error
None	84.5	3	0.158	0.367
global only	140.8	11	0.062	0.172
global + local	172.8	17	0.050	0.092

Table 1: Average lighting correction performance

$$m = \frac{1}{uv} \sum_{uv} i_{(u,v)} \tag{5}$$

$$sd = \sqrt{\frac{1}{uv - 1} \sum_{uv} (m - i_{(u,v)})^2} \tag{6}$$

Local lighting changes are then normalized through a similar process. Local changes in lighting, when a directional light source is used to illuminate the face, can be simply modelled if the head is assumed to be a smooth Lambertian surface. Once these lighting effects are modelled, they can simply be removed by subtraction.

Planar, bilinear and bicubic *lighting maps* could all be used to approximate these effects to varying degrees of accuracy. Bilinear lighting maps as defined by (7) were found to work well. These capture the lighting trend across the texture and have only four parameters to find.

$$f(u, v) = p_1 + p_2u + p_3v + p_4uv \tag{7}$$

The parameters p_1 – p_4 are found by minimizing the RMS error between the reference and current texture’s luminosities. Alternatively, the reference and a synthetic, flat-valued image can be used. The light map can then be subtracted from the textures to remove the lighting effects.

As before, the light map from the previous frame is used to correct the textures in the current frame to reduce computation. Again, to move computation outside the head-locating optimization loop, the negative of the light map is used to adjust the reference texture map temporarily rather than the current texture map.

The “Boston” varying light dataset was used to verify that these simple techniques work. Some 27 sequences of three subjects with strongly-varying side illumination were tracked. Fig 4 shows how the robustness of the tracker is significantly improved by lighting correction.

A criterion for tracking failure is helpful in evaluating performance. When the estimated velocity exceeds a realistic value this often signifies the onset of failure (as the simplex optimizer jumps into a local rather than the desired solution). It cannot detect slow drift of accuracy, but this rarely leads to catastrophic failure. This criterion has proven a very useful measure of robustness as it requires no ground truth data. Table 1 shows four such measures: the average number of frames tracked before failure, the number of sequences tracked to completion without failure; and the RMS positional and angular errors between re-aligned magnetic and optical tracking. Care should be taken comparing these results to that of [9] as their sequences are longer and different failure criteria are employed.

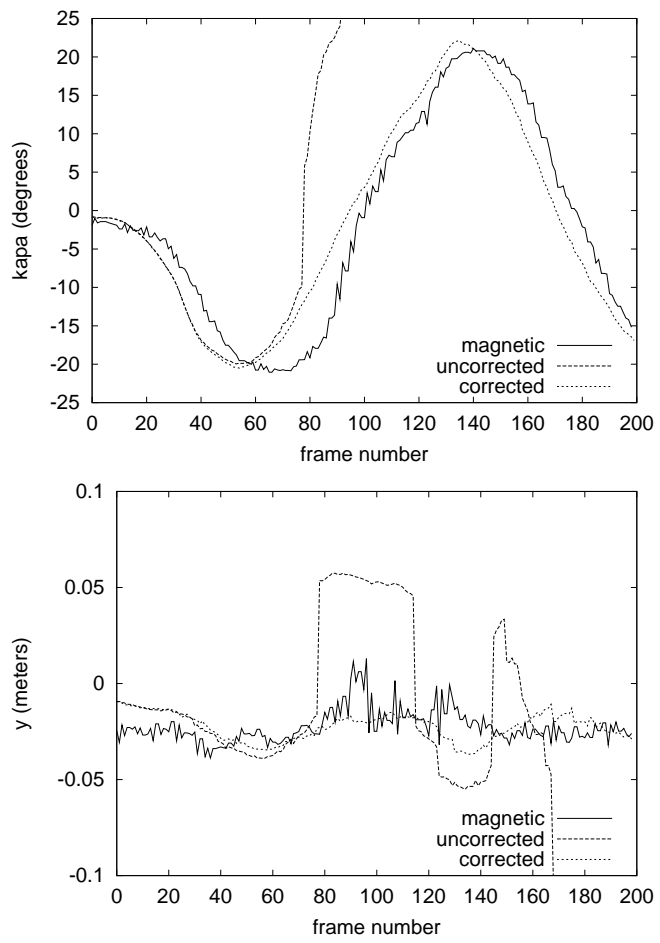


Figure 4: Improvement in robustness though lighting normalization

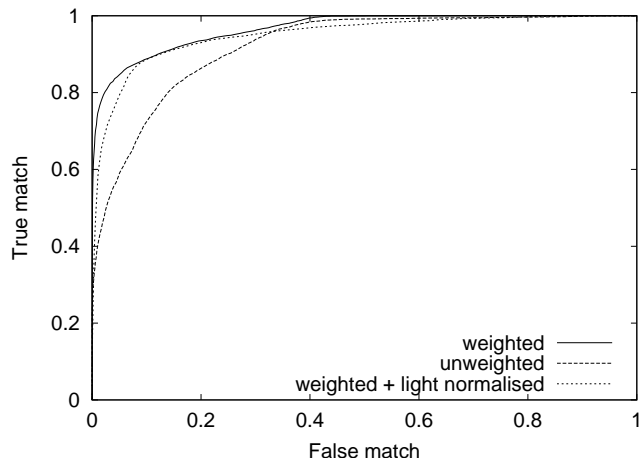


Figure 5: Receiver Operating Characteristic

4 Recognition Using Derived Texture Maps

To explore recognition, the authors used the “Boston” dataset, which consists of nine 200-frame video sequences for each of five subjects. This is admittedly a small dataset, so the results presented here should be treated with some caution. (A somewhat larger dataset is in the process of being collected by the authors.)

Two forms of the well-established “eigenfaces” recognition scheme [2] have been examined. The first form simply uses contrast normalization before principal component analysis (PCA) but assumes each texture pixel has the same accuracy, which is not the case. The second form weights the variance of each element of each face vector by its corresponding confidence value. Using the notation of [2]:

$$\Psi = \left(\frac{1}{1 + \sum_{j=1}^M \omega_j} \right) \sum_{n=1}^M \Gamma_n \omega_n \quad (8)$$

$$\Phi_i = (\Gamma_i - \Psi_i) \omega_i \quad (9)$$

where ω is the confidence vector for an image.

The heuristic weighting attempts to take into account pixel variance that is caused by low confidence pixel noise rather than true inter-face variation. A simple Euclidian nearest neighbour classifier is used to identify subjects in the seven-dimensional weight space.

The test procedure adopted is in accordance with [12], which describes the distinction between “verification” and “identification” in Table 2. Each video sequence was treated as a separate sample, and used only for training, testing, or imposter samples. A “leave-one-out” testing methodology was used, resulting in about 1,800 separate tests. The result of the testing is shown in Fig 5 for both conventional and weighted PCA with and without lighting correction. Overall performance is summarized in Table 2, which is commensurate with front-face-only techniques. Indeed, the error rate is similar to that of [2], so

Database	Equal error rate (verification)	Error rate (identification)
Boston (tracked, PCA)	16%	6%
Boston (tracked, weighted PCA)	10%	2%
Boston (tracked, lighting normalized, WPCA)	10%	4.2%

Table 2: Recognition performance

the extension of the technique to accommodate unwrapped texture maps is introducing no new significant sources of error. It is also apparent that recognition performance is appreciably better using the weighted PCA, justifying the weighting terms introduced in the calculation of the covariance matrix.

5 Conclusions and Further Work

This paper presents initial results from a “texture-from-motion” scheme that is being developed for pose-independent face recognition. The approach to building texture maps appears to be reasonably effective, as demonstrated here. We have demonstrated that classical front-face schemes can be adapted for use on texture maps. Recognition performance is promising, though not yet comparable to the best front-face-only schemes. Although only single frame identifications are presented in this paper to aid comparison with other work, identifications made from successive frames can be used in a voting scheme to accumulate improved accuracy.

Face-feature normalization and a more sophisticated classifier have not yet been included in this scheme. It is anticipated that these will improve recognition rates, just as it does with conventional, front-face schemes. In the longer term, it is intended to extract shape information at the same time as texture information, and our expectation is that this will lead to a further improvement in performance.

References

- [1] T. Sakai, M. Nagao, and T. Kanade. Computer analysis and classification of photographs of human faces. In *Proc. First USA-Japan Computer Conference*, pages 55–62, 1972.
- [2] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [3] S. Lawrence, C. Giles, A. Tsoi, and A. Back. Face recognition: A convolutional neural network approach. *IEEE Trans. Neural Networks*, 8:98–113, 1997.
- [4] F. Samaria. Face segmentation for identification using hidden markov models. In *Proceedings of the 1993 British Machine Vision Conference*, pages 399–408, University of Surrey, 1993.

- [5] A. Lanitis, T. J. Cootes, and C. J. Taylor. An automatic face identification system using flexible appearance models. In *Proceedings of the 1994 British Machine Vision Conference*, pages 65–74, University of York, 1994.
- [6] A. Samal and P. A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25(1):65–77, January 1992.
- [7] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, October 1993.
- [8] S. McKenna, S. Gong, and J. J. Collins. Face tracking and pose representation. In *Proceedings of the 1996 British Machine Vision Conference*, pages 755–764, University of Edinburgh, 1996.
- [9] Marco La Cascia and Stan Sclaroff. Fast, reliable head tracking under varying illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1999.
- [10] Michael C. Lincoln and Adrian F. Clark. Pose-independent face identification from video sequences. In *Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, volume LNCS 2091, pages 14–19, Halmstad, Sweden, June 2000. Springer-Verlag.
- [11] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Systems Programming Series. Addison-Wesley, 1990.
- [12] Association for Biometrics. Best practices in testing and reporting performance of biometric devices, January 2000. <http://www.afb.org.uk/bwg/bestprac10.pdf>.