

3D Pose Estimation of the Face from Video

Gaile G. Gordon

Interval Research Corporation¹
1801 Page Mill Road, Bldg. C
Palo Alto, CA 94304
gggordon@interval.com

Abstract Face pose information is valuable for a variety of applications including unconstrained face recognition, natural human computer interfaces, and video database indexing. 3D pose estimation is a critical requirement for accurate face recognition using view varying representations, such as 2D intensity images. 3D pose extraction in this context requires 3D information, which is present in image sequences if we assume the moving objects in the sequence are primarily rigid. This paper presents a motion based pose estimation system which computes the 3D pose of the head in each frame of a video sequence. Generic low level features, such as corners, are identified and tracked in the video stream. The feature tracks are processed by a shape from motion algorithm which produces estimates of 3D geometry and pose. The geometry and pose estimates are considered together with facial structure constraints, temporal constraints, and initial pose estimates to refine knowledge of the specific face structure and its pose. We describe this system and show that it works on human faces. This is significant because the face has many smooth surfaces which make it difficult to extract dense intensity features.

1 Introduction and Motivation

Knowing the pose of the face is important in many applications including interactive human computer interfaces, and video database indexing. However, we highlight in particular the importance of pose in the context of facial recognition. In the context of recognition, knowing the pose of the face is important if the representation of the face used for comparison is not view invariant. For instance, 2D intensity data is not view invariant, whereas surface curvature data is.

In this paper, we first discuss in detail the importance of face pose in 2D face recognition. Pose estimation from a single 2D image is ill-posed, however, several authors propose pose estimation methods using a single 2D image and a general 3D face model. In section 1.2, we provide several examples showing that the error source this approach introduces can be significant in the context of face recognition. The remainder of the paper describes a pose estimation method based on structure from motion. The 3D information present in a sequence of video images of a moving face is sufficient to compute an accurate pose estimate consistent with the 3D structure of an individual's face. The algorithm is described in section 2, stressing novel contributions involving the elimination of non-rigid feature points. Results are presented in section 3 which show that this structure from motion technique is effective for human faces despite the fact that it requires trackable intensity features which can not be easily extracted on smooth surfaces. Conclusions and future work are presented in section 4.

¹Much of this work was done while the author was at TASC in Reading, MA. and was supported by the FERET program of the Army Research Lab under contract DAAL01-93-C-0118.

1.1 Face pose in 2D face recognition

The term *pose* is given different meanings in many different contexts, so we begin by defining pose in the context of human faces. In general, face pose provides the position and orientation of the face. This is also analogous to the *view* of the face if one thinks of the face as stationary. More precisely, we define the 3D face pose as the six degrees of freedom which specify the relationship between the camera or sensor coordinate system and a face centered coordinate system. A face centered coordinate system is defined in terms of locations of specific facial landmarks. The choice of landmarks is not critical, except that their location must be reliable so different datasets of the same individual can be compared in a repeatable way.

Recognition from 2D intensity data is an attractive idea. The data is easy to obtain, and in many scenarios is already available. There are also several popular recognition methods using intensity data which work well under certain conditions [Moghaddam et al., 1996, Pentland et al., 1994, Wiskott et al., 1997]. However, one common property of these 2D methods is that similarity measures decrease as the difference in pose between two faces increases (see Figure 1). Correct recognition occurs when the relative difference in similarity scores for different individuals is larger than for the same individual with different poses. However, as pose misalignment increases, more recognition errors are likely to occur. Additional factors which also degrade similarity scores, such as lighting variation, and expression variation, only compound the problem. Currently we have no way of knowing what degree of pose mismatch exists, so we can't detect whether a low recognition score is due to pose misalignment error or actual identity mismatch.

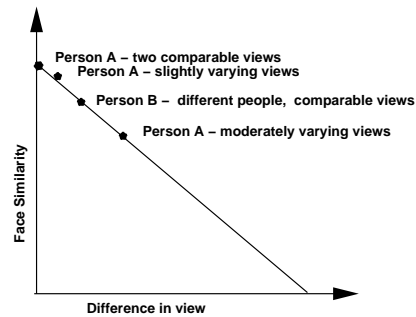


Figure 1. Similarity scores are a function of view similarity for 2D intensity face recognition systems. If the relative difference between scores of different individuals is sufficiently large compared to scores of the same individual over different views, correct recognition occurs (first three examples will produce correct ranking). Larger view misalignment or the presence of other factors which also degrade recognition (lighting variation, expression variation, crowded database) create a higher number of recognition errors.

Explicit knowledge face pose makes it possible to detect pose misalignment error. Once detected, it is possible to minimize its effect by either choosing a closer matching pose, if alternatives are available (e.g. from a video stream), or, potentially, by transforming the available images to a better pose alignment.

1.2 Pose Computation from a Single 2D View

Computation of 3D pose requires 3D information. 3D pose can be computed from a single 2D image only when there is a set of landmarks visible in the image which have a known 3D relationship² (as is available in the case of model based recognition). This type of information is not available in the case of face recognition based on a single set of intensity data, which is inherently 2D. If 3D relationships on the individual's face are known from some other source, we can build a model specific to the person's face shape to compute pose in a novel 2D image.

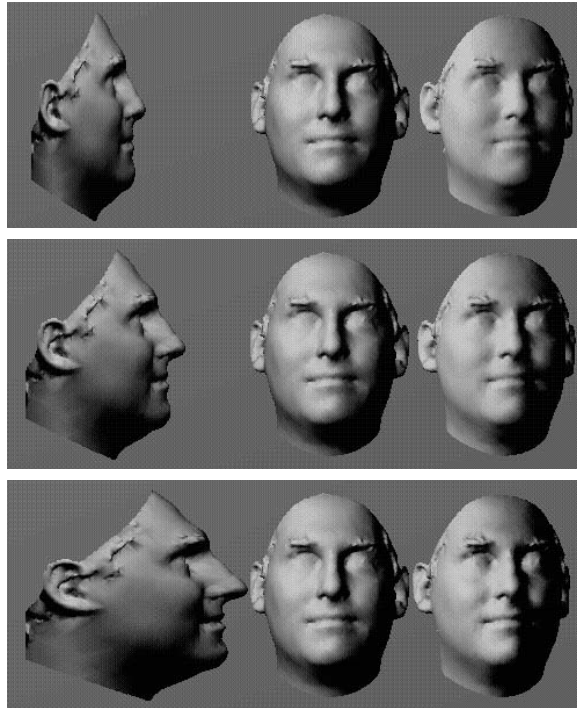


Figure 2. Illustrates the Bas Relief Ambiguity [Belhumeur et al., 1997] which confounds depth and view. The pose of the faces in the right column actually varies by several degrees (rotation about the vertical axis is 5.0° , 7.1° , 3.5° respectively). These tradeoffs between depth and view are impossible to distinguish from a single 2D image. (Reproduced by permission of Peter Belhumeur.)

Approaches have been suggested to deal with pose computation and construction of novel views from a single 2D view of the face using 3D information derived from *average* or *prototype* face shape [Aizawa et al., 1989, Beymer and Poggio, 1995, Vetter, 1997]. Since the shape of individual faces varies quite a bit, using a general face shape model introduces an

²The solution to the 3 point perspective pose estimation problem has been known since 1841 [Haralick et al., 1991]. This solution provides the 3D coordinates of each vertex of a known triangle from the perspective projection of the three points in a single 2D image.

inherent error into the pose computation. Although this source of error may be smaller when using a large set of prototype face models (assuming that there is truly a closer relationship between the actual face shape and one of the prototype face shapes), one would still expect pose computed using shape specific to the individual to be more accurate because it does not include this modeling error.

We provide two more visual arguments against the use of general face shape in computing face pose from a single image. The first is from [Belhumeur et al., 1997]. Figure 2 shows a nice example of how depth (and therefore shape) and view are confounded. The images in the middle and right columns all look exactly like the same face in the same pose, yet the pose of the faces in the right column actually varies by several degrees (rotation about the vertical axis is 5.0° , 7.1° , 3.5° respectively). The difference in pose is balanced by a difference in 3D shape (left column). These tradeoffs between depth and view are impossible to distinguish from a *single* 2D image.

The use of a general face shape to analyze the intensity data of a specific face is equivalent to mapping the intensity of one face onto the shape of another. Figure 3 shows a specific example of this type of mapping, with the bottom row of images being composed of the shape from the second row and the intensity from the first row. Does the last row represent either the first or second individual? It seems clear that a *perceptually independent* face can be produced in this manner. Novel views of a face generated using this sort of technique would therefore bring up many issues in the context of a recognition system. This effect is obviously subjective, but merits formal psychophysical testing.

1.3 Pose Computation from 3D Data

When an object is represented by 3D data, accurate estimation of pose is a simple geometric computation. The only requirement is that the features which define the object coordinate system can be identified in the data.

There are quite a number of methods to obtain 3D description of a face. On one end of the spectrum, there are laser range scanners (such as those made by Cyberware which were used to generate the data in Figure 3). These collect high spatial resolution data (in an absolute scale), but require expensive and non-portable sensing equipment. In past work [Gordon, 1991b, Gordon, 1992, Gordon, 1991a] we have used automatic feature extraction to perform pose computation and normalization on these types of data sets. Figure 4 shows the points which define the face centered coordinate system in this work.

At the other end of the spectrum, three dimensional descriptions can be computed from 2D intensity images using stereo or structure from motion techniques. These techniques provide lower resolution depth, balanced by the advantage of less expensive and more portable sensors. In particular, structure from motion is attractive because it requires only a single uncalibrated camera. The remainder of the paper describes a pose extraction algorithm using structure from motion on video sequences.

2 Pose Computation from Video Sequences

Pose extraction from video has been demonstrated in the general case. Our algorithm is based on the work of [Tomasi and Kanade, 1992]. Faces, with their smoothly varying surfaces, are not at the outset ideal candidates for these algorithms, which are based on tracking features defined by contrast events. Their surfaces can also exhibit elastic deformation, which violates the rigid motion assumptions. Similarly, the head and the torso can move independently. We use domain specific knowledge to successfully compute face pose from video despite these potential problems. The contribution of this work involves automatically limiting the feature selection to the central portion of the head through independent initial pose estimation, and

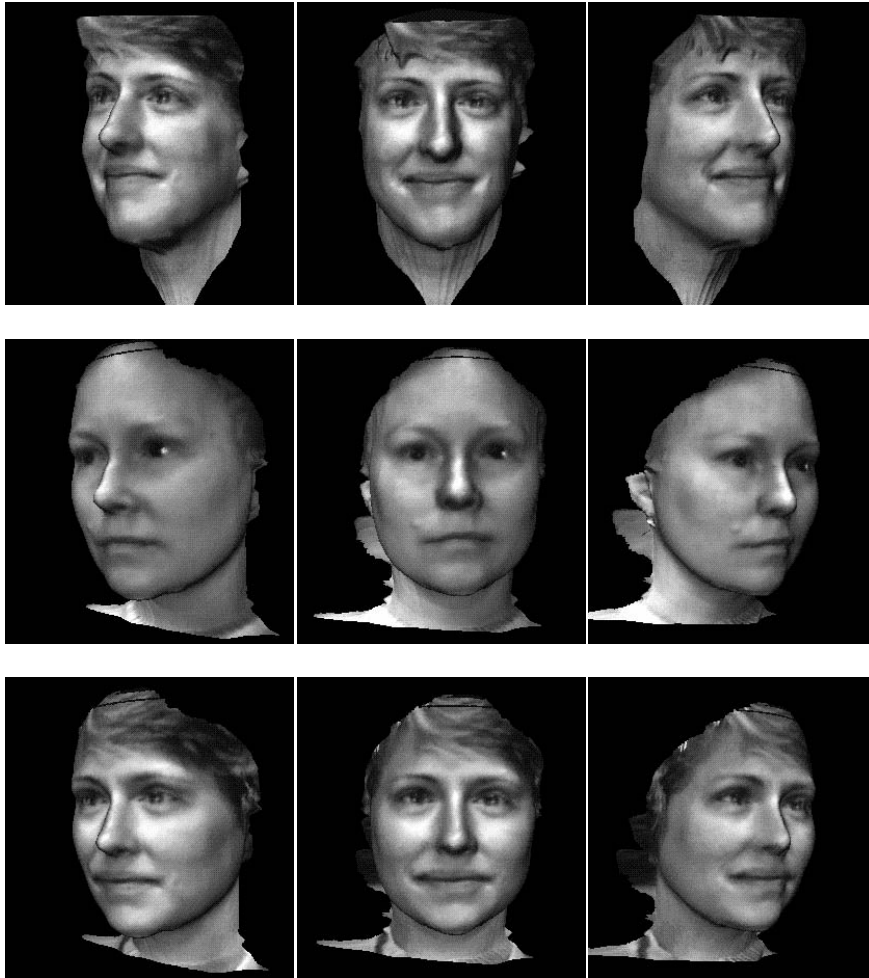


Figure 3. The top two rows of images are rendered 3D models of two different people based on actual shape and color scans (using a Cyberware scanner) of their heads. The bottom row shows the intensity data from the first person, mapped onto the shape data of the second person. The new synthetic face is perceived as a different (new) person.

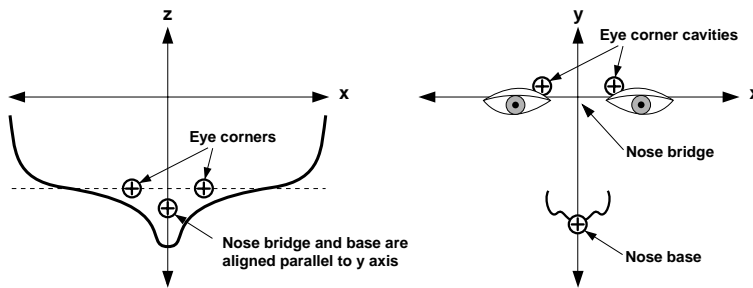


Figure 4. The face centered coordinate system defined for pose computation with range data based on features automatically extracted from surface curvature.

through careful elimination non-rigid feature points via modeling of the occluding contour and residual error analysis. Both of these issues are discussed in section 2.1.

[Azarbayejani and Pentland, 1995] presents related work on pose computation of faces from video sequences. This work represents an interactive rather than batch mode algorithm, but is based on similar rigid motion assumptions. The emphasis of this work is on stability in the domain of recursive estimation, whereas our emphasis is on automated feature selection to avoid non-rigid phenomena. Their face pose estimates include only 6 or 7 features such as the eyes centers, nose base, etc.

The pose extraction algorithm includes these key steps:

- Computation of initial view (frontal view)
- Feature selection
- Feature tracking
- Estimation of occluded feature positions
- Computation of structure and pose at each frame

The last step, a structure from motion algorithm, is addressed first because its requirements drive the design of the other steps. The structure from motion algorithm used is based on the factorization method presented by [Tomasi and Kanade, 1992]. The algorithm takes as input a set of feature tracks. Each track consists of an array of 2D image locations, one for each frame, showing the path which a 3D point on the surface of a rigid object takes from frame to frame. These feature tracks are assembled into a $2N \times F$ matrix, where N is the number of frames and F is the number of features. After adjustment to separate the effects of translation, the track matrix is factored into two matrices, one providing the 3D location of the feature points (structure) and other providing the rotational relationship between the object and the camera at each frame (full pose results from the combination of the rotation and translation data). The pose given at each frame is relative to the first frame of the sequence. The center of mass of all feature points is used as the origin of the coordinate system. This algorithm uses orthogonal projection and assumes that 1) the correspondence of the features points from frame to frame is known, 2) occlusion events are known, and 3) the motion represented by the feature tracks is that of a rigid object.

The job of *automatically* selecting and tracking the points for input to the factorization algorithm is complex. The features tracked must correspond to surface points on a rigid object. Image features corresponding to occluding boundaries, shadows, and motion independent from the head (e.g., torso or other objects in the scene) violate this assumption and act as sources of error. This becomes even more critical when using this algorithm on faces, because the total number of features which can be extracted in low contrast areas is small, and there

are large low contrast regions on the face. We discuss the feature selection first, followed by the feature tracking. The following section discusses avoiding non-rigid features.

The basic feature selection method [Shi and Tomasi, 1994] identifies local image patterns containing edges in more than one direction. These are identified by examining the eigenvalues of the G matrix used in tracking (equation 3). Two strong eigenvalues of G are a good indication of strong gradients in multiple directions and hence corner-like features. A threshold on the minimum eigenvalue of G is used as a specific selection criteria.

Once the features have been selected, they are tracked by modeling the motion of the feature window as a simple translation. The displacement of the feature window, d , is computed by minimizing the error residue, e , between frames based on this model [Lucas and Kanade, 1981].

$$Gd = e \quad (1)$$

$$G = \sum_W \begin{bmatrix} x'^2 & x'y' \\ x'y' & y'^2 \end{bmatrix} \quad (2)$$

$$e = \sum_W \begin{bmatrix} \frac{dI}{dt} x' \\ \frac{dI}{dt} y' \end{bmatrix} \quad (3)$$

where W is a small local image region (15x15 pixels). If this solution does not converge quickly, tracking stops for that specific feature. Although the translation model is more simplistic than the actual image change expected over time (e.g. it doesn't include rotation or scale changes), it is quite effective over the small inter-frame distances typical in video rate sequences.

2.1 Avoiding Non-rigid Features

This low level feature selection and tracking algorithm has several additional steps aimed at reducing the number of features selected and tracked which do not correspond to points on a rigid surface. In some cases, we take advantage of our knowledge of the application domain.

Identifying where the head is in the image is an important step. An initial pose selection algorithm provides an estimate of likely frontal views in the image stream, and also identifies the center and scale of the face. This estimate is used to locate the head initially, and limit the feature selection to the central head region, rather than the torso or the background.

Initial frontal pose is selected via a multi-resolution template matching algorithm. A small number of templates are used consisting of the central portion of generic frontal view faces at varying scales. The matching is performed at a reduced spatial and temporal resolution to lower computation time. The maximum of all correlation scores in any frame or scale is used to determine the face scale. Frames which represent local maxima in the correlation score for the selected template are identified and sorted in descending order. The highest value in the list determines frame number of the frontal views from which pose computation begins. Figure 5 shows an example of typical correlation scores over an image sequence of a rotating head. The head passes through frontal view twice.

Object occluding contours are a common source of bad feature points. Many corner features which are not true surface features can be found along the occluding contour, and can even be consistently tracked. Often these are caused by an intersection between an edge in the background and the object, or a concavity in the occluding silhouette which is visible over several frames (e.g. the intersection of the base of the ear and the neck). Assuming that the background is not moving, we compute a mask based on local frame differences indicating a liberal region contiguous to the occluding contour. We then insure that new features are not selected within this region, and similarly we stop tracking existing features when we reach this zone.

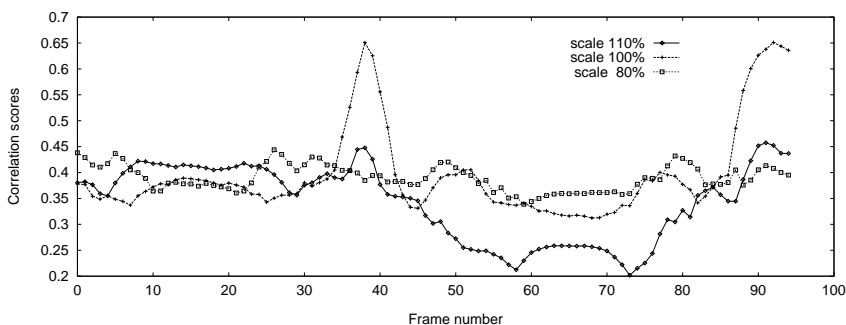


Figure 5. The selection of frontal view frame based on multiresolution correlation with generic frontal view face.

Two further steps are taken to minimize the inclusion of points not representative of rigid surface motion. The first is an affine criteria used to monitor longer term changes in the feature window [Shi and Tomasi, 1994]. Including affine transformations allows for a more realistic tracking model. To examine the likelihood that a feature window has been tracked correctly, we can compare its original values with those of its final tracked position. The affine transform which best models the relationship between the two windows is computed. This transform is applied to the original window, and the result is compared with the window in the actual tracked position. Low similarity is indicative of occlusion. A threshold can be used to remove these features from further consideration.

In addition to the affine similarity criteria, nonrigid features are identified by comparing the actual observed tracks with those estimated by the computed pose and structure [Gordon and Lewis, 1995]. Average residual for feature f is

$$\epsilon(f) = \sum_{i=1}^N \frac{\|observation(f, i) - prediction(f, i)\|}{N}$$

The predicted location at frame i is the orthogonal projection of the 3D feature location transformed by $pose(i)$. Features with large position residuals are removed and the pose and structure are recomputed. This method is effective at removing features not well modeled by the computed structure and pose parameters. It requires one additional recomputation of the tracking matrix factorization.

2.2 Occlusion Events

In the above description, a number of different methods are discussed to identify features which are no longer being tracked successfully. This will often occur because a point being tracked becomes occluded. Since the structure from motion algorithm requires a complete feature track matrix, we must either eliminate a partially tracked feature from the matrix in all frames or attempt to fill in the undefined feature locations via estimation. It is clearly an advantage to be able to keep as many features in the tracking matrix as possible. Also, with an effective estimation procedure, new feature sets can also be selected during the tracking process to incorporate new portions of the object which come into view during the sequence. When new feature sets are selected, their locations can only be tracked forward in time from

the point where they were identified. Thus, feature locations must be estimated for the frames before the selection occurred.

Estimating an Individual Point Location

The estimation procedure repeatedly examines complete submatrices of the sparsely filled track matrix [Lucas and Kanade, 1981]. To estimate any given feature location, the algorithm starts with a complete submatrix, $2F$ rows \times P columns, and two partial rows each $P - 1$ in length. These partial rows represent tracks for the first $P - 1$ points in a new frame, $F + 1$. From this information, the coordinate of the point P in frame $F + 1$ is estimated.

The first step in the estimation process is to perform factorization of the dense submatrix, which establishes shape information for all P points, and the rotation and translation information for F frames.

To compute the rotational axes for frame $F + 1$, we use the $2(F + 1) \times (P - 1)$ system for which, between the matrix and partial rows data, all point locations are available. The computation of the shape coordinates was originally based on the centroid of all P points and must be first adjusted with respect to the centroid the first $P - 1$ points in each frame. As long as there are at least 4 total points, the traditional $\tilde{W} = RS$ equations will then provide an over-determined system (solved with least squares) for the two missing rotation vectors.

The translation vector for frame $F + 1$ is computed algebraically, using the original shape coordinates for $P - 1$ features, the rotation information for the new frame, and the tracking coordinates of the $P - 1$ points in the new frame.

The missing track coordinates for the point P in frame $F + 1$ are now a straight forward solution of the $W = R * S + T$ equation.

Fill Order

The control structure of our estimator is responsible for selecting the order of estimation of new points. The estimator is invoked once in each frame for each point which has an unknown location. New features are selected in groups every 8 or 10 frames. The general looping scheme processes the sparse matrix by feature selection group in several passes. Before the first pass, the columns (points) in each feature selection group are shuffled so that the number of valid frames is decreasing as point index increases. Features which are detected in less than four total frames are eliminated. An example this operation is shown graphically in Figure 6(A) and (B). Figure 6(A) represents the fill pattern of the original sparse matrix. In this example features were selected twice after the initial feature selection. Figure 6(B) shows the same information with a different point index order.

During the first pass, each of the feature selection groups is processed, filling forward in time and in increasing order of point index. For this example, the points in the original feature selection group have values over all frames at the completion of this pass. However, for any subsequent feature selection groups, there are invalid values in all frames earlier than the selection frame. The resulting track matrix is shown in Figure 6(C). The second pass fills backward in time and in increasing order of point index, as shown by the completely filled matrix in Figure 6(D). In this example there were a number of features which were valid in all frames. If this were not the case, the columns would be reshuffled and a third pass would fill in the remaining points, filling forward in time and in increasing order of point index.

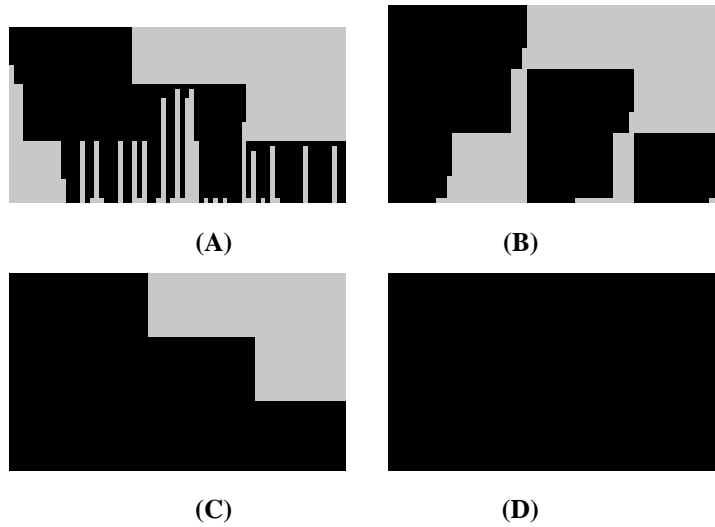


Figure 6. Filling in the track matrix by feature point estimation. Example shows the fill pattern of a track matrix at various stages of processing. Black squares indicate valid feature locations. Gray squares indicate unknown values. Each row represents a frame (first at the top) and each column represents a feature points. **(A)** After the tracking is completed, **(B)** after change of feature point order, **(C)** after first stage of estimation, and **(D)** after second stage of estimation.

3 Results

The pose extraction algorithm has been tested using the FERET Program video database. Each video sequences is 5 to 10 seconds and shows approximately 180° of rotation of the head primarily about the vertical (Y) axis, e.g. including both profiles and frontal views.

A frontal view frame is identified automatically as described above, and a segment of the sequence is chosen for processing such that the front view frame is essentially in the middle. Subsequences ranged from 30 to 60 frames.

Sequences were processed to produce pose estimates at each frame, as well as the 3D locations of all the feature points. A polygonal model can be generated from this data using 2D Delaunay triangulation (we used the frontal view frame) to provide the topology of the 3D points. Image data from the frontal view frame was texture mapped onto the polygonal model for visual evaluation. An example of this type of model is shown in Figure 7 at different viewing angles. Since we have computed 3D coordinates only at the extracted feature locations, there are several areas (e.g. along the bridge of the nose) which will not be described well. However, this shows that even a sparse model is useful to provide the general structure of the face. In particular, the eye sockets are set back with respect to the nose, and the overall face shape is gently curved.

To evaluate pose we visually compare our computed model transformed by a specific computed pose to the actual image in the corresponding frame. Figure 8 shows this comparison at the start and end of a sequence covering 36 frames and consisting of 40 tracked features. Relative pose between the two frames shown is 2.4° rotation about the x axis (horizontal), 70.3° rotation about the y axis (vertical), and 0.7° rotation about the z axis (out of the image plane). Visual alignment between the frames and modeled pose is very close, despite

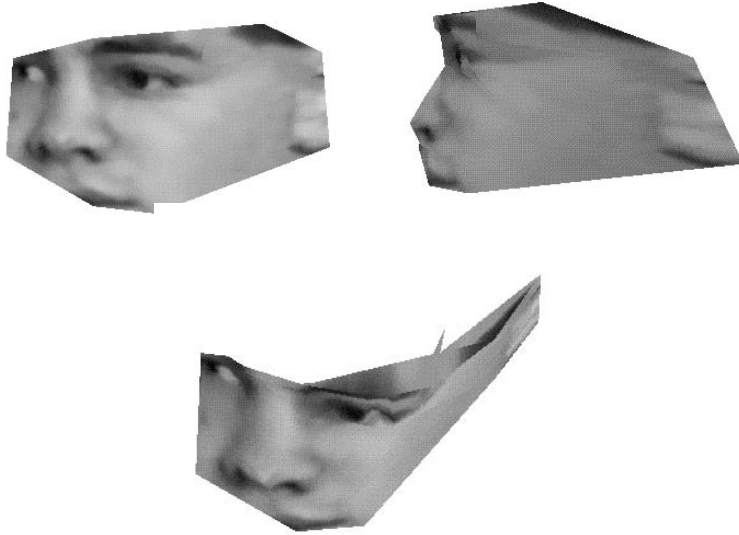


Figure 7. Different views of the same face model.

the fact that the texture which is mapped onto the model is from the frontal view frame, and is not expected to be a good map for a frame $\pm 30^\circ$ away from this view.

4 Conclusions and Future Work

This paper presents a concrete demonstration of the use of structure from motion algorithms to compute the pose change and 3D structure of a human face as seen in a video sequence. This successful demonstration is significant particularly because of the many challenges presented by facial images with respect to structure from motion computation. These challenges include an uneven distribution of intensity features, many smooth surfaces, potential elastic movement, and the existence of multiple motion fields. The use of this capability enables accurate face recognition based on representations which are not view invariant, such as 2D intensity images.

In the context of face recognition, pose information can be used for extracting key poses from video (e.g. building reference databases), and, similarly, to select views during comparison whose pose best match those in the reference database. Since pose mismatch is a key source of error in recognition systems based on view variant representations, the comparison of pose provides an explicit confidence measure in the recognition process. It may also be possible to use the pose information to further transform available images to better match existing pose examples.

Face pose can be valuable in many other application domains including video database management, facial expression analysis, and human computer interface.

Despite this successful demonstration, there are remaining issues which form the basis of future work. A key issue is the use of *a priori* knowledge of face structure to register

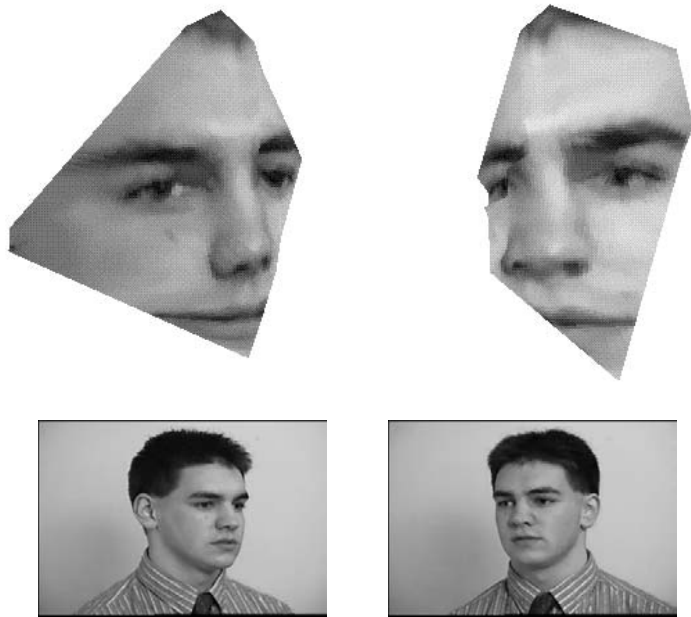


Figure 8. The model computed is shown in the computed pose of the first and last frames of the subsequence. This is compared to the actual video image at the corresponding frames demonstrating a close alignment.

the pose computed to a head centered coordinate system. At the moment, the algorithm tells us only pose relative to the frontal view, but the exact initial pose is not determined with respect to a face centered coordinate system. This can be done via combination of this structure from motion approach with feature location algorithms [Covell and Bregler, 1996], thus relating the generic corner features selected for tracking with the semantic understanding of the facial feature locations. Once this relationship is understood, the motion computed can be transformed to the face centered coordinate system. The goal of this registration is to produce pose estimates comparable across all data sets and repeatable within data sets of the same individual.

References

- [Aizawa et al., 1989] Aizawa, K., Harashima, H., and Saito, T. (1989). Model-based analysis synthesis image coding system for a person's face. *Signal Processing: Image Communications*, 1(2):139–152.
- [Azarbayejani and Pentland, 1995] Azarbayejani, A. and Pentland, A. (1995). Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):561–575.

- [Belhumeur et al., 1997] Belhumeur, P. N., Kriegman, D. J., and Yuille, A. L. (1997). The bas-relief ambiguity. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1060–1066, Puerto Rico.
- [Beymer and Poggio, 1995] Beymer, D. and Poggio, T. (1995). Face recognition from one example view. In *Proceedings of the International Conference on Computer Vision*, pages 500–507, Cambridge, MA.
- [Covell and Bregler, 1996] Covell, M. and Bregler, C. (1996). Eigen-points. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 471–474, Lausanne, Switzerland.
- [Gordon, 1991a] Gordon, G. G. (1991a). *Face Recognition from Depth and Curvature*. PhD thesis, Harvard University, Division of Applied Sciences.
- [Gordon, 1991b] Gordon, G. G. (1991b). Face recognition from depth maps and surface curvature. In *Proceedings of SPIE Conference on Geometric Methods in Computer Vision*, volume 1570, San Diego, CA.
- [Gordon, 1992] Gordon, G. G. (1992). Face recognition based on depth and curvature features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 808–810, Champaign, Illinois.
- [Gordon and Lewis, 1995] Gordon, G. G. and Lewis, M. E. (1995). Face recognition using video clips and mug shots. In *Proceedings of the Office of National Drug Control Policy (ONDCP) International Technical Symposium*, volume 2, pages 13.45–13.51, Nashua, NH.
- [Haralick et al., 1991] Haralick, R., Lee, C., Ottenberg, K., and Nolle, M. (1991). Analysis and solutions of the three point perspective pose estimation problem. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 592–598, Maui, Hawaii.
- [Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679.
- [Moghaddam et al., 1996] Moghaddam, B., Nastar, C., and Pentland, A. (1996). Bayesian face recognition using deformable intensity surfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 638–645, San Francisco, CA.
- [Pentland et al., 1994] Pentland, A., Moghaddam, B., and Starner, T. (1994). View-based and modular eigenspaces for face recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 84–91, Seattle, WA.
- [Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, WA.
- [Tomasi and Kanade, 1992] Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154.
- [Vetter, 1997] Vetter, T. (in press, 1997). Synthesis of novel views from a single face image. *International Journal of Computer Vision*.
- [Wiskott et al., 1997] Wiskott, L., Fellous, J.-M., Kruger, N., and von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779.