Head pose estimation without manual initialization

Paul Fitzpatrick

AI Lab, MIT, Cambridge, USA
paulfitz@ai.mit.edu,

Abstract. This paper describes a fully automatic system for recovering the rigid components of head pose. It follows the conventional approach of tracking pose changes relative to a reference configuration. Rather than being supplied manually, here the reference configuration is determined by directly estimating pose from the image when the head is close to a frontal presentation. Tracking of pose is done in an intermediate coordinate system which minimizes the impact of errors in estimates of the 3D shape of the head being tracked.

1 Introduction

Unlike face detection or expression recognition, it is easy to imagine recovering head movement without knowing much in advance about head structure. Rigid body motion is the same for the head as it is for anything else; rotations and translations do not become somehow mysterious because they are biological in origin. Nevertheless, there are a number of advantages to be gained by considering the structure of the head :-

- ▷ The head is overlaid with non-rigid features, either actuated (the face) or just passively non-rigid (long hair). It is important to consider these features, if only to avoid them.
- ▷ To recover rigid motion from visual information, some knowledge of the shape of the head will have to recovered. To do so, it helps to have a parameterized model to match against.
- ▷ There must be a way to relate the pose of the head to the actual surface features that are important for interface purposes – for example, the eyes and mouth. This requires either manual initialization, or some automatic replacement that will necessarily depend on special knowledge of the head's structure.

It is possible to imagine ways to do without the first two points, but since this project aimed at a fully automatic system, the last point made at least some special consideration of the head's structure unavoidable. The next section looks at how and when other groups have used special knowledge of the head to aid in estimating its pose. The remainder of the paper is devoted to developing and evaluating a complete head pose estimation system, decomposed into a number of relatively independent components. The first is a head tracker, which tries to determine the current outline of the head. The second module is a pose recognition module, which replaces manual initialization. Both of these are very much dependent on a prior model of the head and face respectively. The next module, which is less model-dependent, is a mesh-based pose tracker that estimates how the head is moving from frame to frame, working in a coordinate system designed to limit the effect of inaccuracies in estimates of the head's dimensions. All the modules are combined to give fully automatic pose estimation, which can recover the position (up to a distance ambiguity) and 3D orientation of the head. The nomenclature used in this paper for talking about orientations is shown in Figure 1.

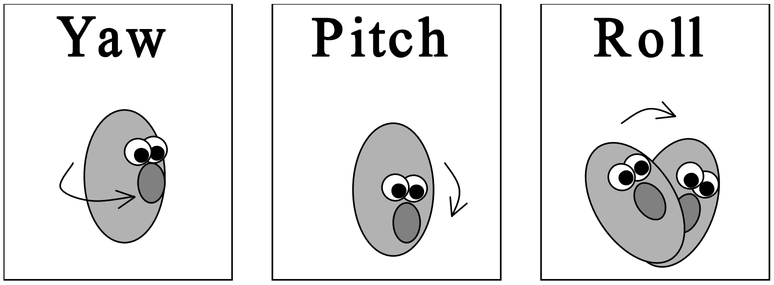

Fig. 1. Orientation of the head is described in terms of yaw, pitch, and roll as shown. To fully specify what these are being taken to mean, their order of application would need to be made clear. Since this coordinate system will not used internally for estimation, the details of this choice do not matter and will be left unspecified.

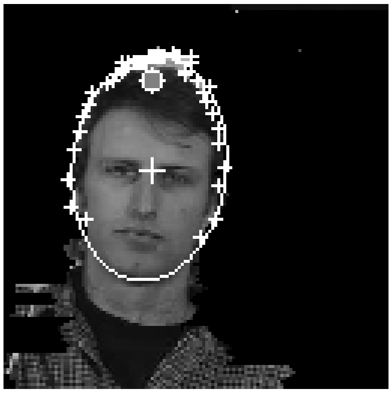

Fig. 2. Finding the outline of the head once it has been detected. Probes (shown as small crosses) radiate out from a location near the top of the head/body (shown as a circle). Across a large range of scales, the silhouette of the head can be extracted from the contour points encountered by the probes, and matched against an elliptic model.

2 Background

I will just make a brief review since this topic was covered extensively in class, and I am aware that my audience knows more about the subject than I do. There are many possible approaches to head pose estimation. At one end of the scale, there are techniques that rely on very strong models of head shape, such as the work of Horprasert and Yacoob [4]. The shape of the human head is broadly similar across the species. Anthropometry characterizes the distribution of face length scales and ratios within different sub-groups. These distributions are quite narrow for a subject whose gender, race, and age are known. Horprasert and Yacoob make use of this to estimate head orientation from monocular images. They show that pose can be recovered by tracking just five points on the face (four at the eye corners and a fifth at the tip of the nose), given that the necessary anthropometric data is available. They propose a two stage system that estimates a subjects gender, race and age first, indexes into the appropriate table of anthropometric data, and then performs the pose estimation.

At the other end of the scale, there are pose tracking systems which do not require a prior model, and are therefore of more general application than systems that rely on special characteristics of the head – for example Harville et al [11].

Other points on the spectrum include the application of eigenspace techniques to directly recognize the pose of a specific user, as opposed to tracking changes in pose [14]. And then there are very many systems designed to run in real-time, using a wide variety of simple cues such as hair outline [22].

3 Head outline tracking

The outline of the head is tracked using a collection of techniques that seem to be typical of real-time systems [9], such as image differencing and ellipse-fitting. The implementation described here is qualitatively similar to that of Smith-Mickleson [20], and also traces back to Birchfield's work [3].

Before the head outline can be tracked, the head needs to be detected in the first place. Head movements often have a marked translational component. This is particularly the case when someone is walking into the scene (which is a perfect time to do initialization). Such movement makes it relatively easy to distinguish the head and body from the background using image differencing. A simple template tracker is assigned to the largest blob detected in this way. The image is then modeled as being generated by overlaying two layers, a "body layer" moving with the tracker, and a "background layer" that is stationary. Each pixel is independently assigned to one of the layers based on the intensity difference of that pixel with its predicted location in the previous frame for each layer. This gives a cleaner, more persistent outline for the body than raw image differencing, and discounts at least some fraction of pixels from moving objects in the background. The outline is cleaned up using various heuristics (implemented using Viterbi-based optimization across scan-lines). Probes are sent out in all directions from a point close to the top of the body to characterize the outline, and in particular identify the location of the head. The probes are filtered to eliminate those that wander back to the body, and an oriented ellipse is fit to the remainder[15]. Figure 2 shows an example of this.

If the ellipse is a good fit, its interior is used to initialize a color histogram. There is a tradeoff between trying to include the hair-color within the histogram while avoiding including the background. I found it necessary to err

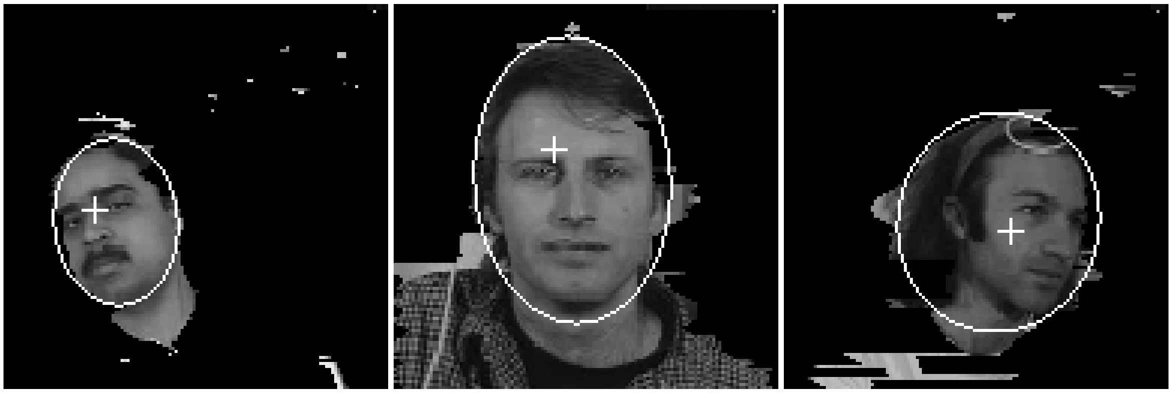

Fig. 3. Snapshots of the head tracker in action. Hair is problematic. Sometimes it will be included, sometimes not. In individuals with a great deal of hair (rightmost figure), the ellipse can deviate a great deal from the basic shape of the head itself.

on the side of caution, and not include the contents of the entire ellipse in the histogram, which often meant that hair color was not included. Figure 3 shows examples of the kind of variation this can lead to in what the putative “head outline” actually means.

4 Pose recognition

The pose recognition component is responsible for replacing manual initialization by automatically determining when the head is in a recognizable pose. The human face has a rich enough structure to admit of several possibilities for pose recognition :-

- ▹ Frontal pose. Because of the approximate bilateral symmetry of the human body, the projection of the face is close to symmetric in this pose. This, along with the relatively clear-cut features of the face such as the eyes and nose, makes the pose relatively easy to detect. This pose also has special behavioral status because of attentive orienting, and occurs very frequently during face-to-face human/robot interaction, which is my domain of interest.
- ▹ Profile view. The head has its most well-defined silhouette for 90° of yaw, particularly around the nose.
- ▹ Hair-line. Wang et al [22] argue that the hair-line can be indicative of pose.
- ▹ Recognition of trajectories. This is a rather different possibility, where the output of relative tracking is used as a feature in its own right. The movement of the head is strongly constrained by the neck, and it seems possible that those constraints may be tight enough to give unambiguous interpretations for certain types of head movement, particularly if enriched with some knowledge of the head outline.

Many other cues are also practical, particularly if training for an individual is possible, but these were the ones considered for this project. Of the four, frontal pose was the one eventually developed. Profile was problematic for extracting an accurate orientation, since the silhouette changes too slowly with changes in roll and yaw. The hair-line is very variable between subjects. And trajectory recognition would in practice require a great deal of training data to learn the priors, and even then it is not clear whether it would actually do anything.

Frontal pose is by no means ideal either. Bilateral symmetry only constrains yaw; pitch can vary freely without affecting the symmetry of the face. The outline of the head from the tracker in the previous section can be used to constrain how the facial features should be distributed vertically, but this means that the initialization of pitch will suffer from the noise in the estimate of the outline. So be it; in actual use there are behavioral cues as to when the person is likely to be looking directly at the camera (a toy example is that the robot can suddenly shout HEY HUMAN! and if the yaw rotation seems to go to zero immediately thereafter, it is reasonable to assume that pitch is also zeroed). If yaw and pitch are indeed near zero, then any roll component in the pose corresponds to simple 2D rotation on the image plane and can be measured directly.

I had hoped to be able to base frontal pose recognition on an eye detector I had available [6]. But the requirements of pose recognition are different to feature detection :-

- ▹ A low false detection rate is arguably now much more important than a low missed detection rate, since a false detection, if trusted, will cause all following pose estimates to be inaccurate. There are ways to render this less of a problem, but nevertheless it is true that the trade-off will be different for this application.

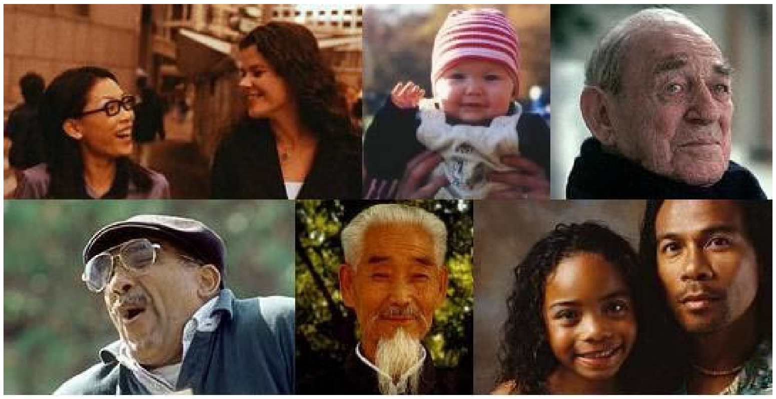

Fig. 4. Heads vary in dimensions, hair styles, clothing, expression, etc.

- ▹ There are 6 dimensions that need to be initialized. A detector that simply returns a single image plane coordinate is not going to do the job without a lot of extra machinery tacked on.
- ▹ Crucially, pose-invariance – usually so highly valued in a feature detector – is now a positive disadvantage and nuisance. Ideally we want a very sharply tuned response, limited only by the fact that the sharper the response is, the less likely we are to be presented with an instance of it during an interaction.

Informed by these requirements, I constructed a feature detector specifically designed for accurate localization. Looking at the variation in facial features from person to person (see Figure 4), there is very little that can be relied upon; the eyes and nose seem the best of a bad lot. At the low resolutions real-time performance often mandates, many attempt to locate eyes using the fact that since they are generally somewhat recessed, their associated pixels tend to be darker than the surrounding skin. But since the face may be unevenly illuminated, it is difficult to translate this model into a statement about pixel intensity or color thresholds (for example). A statement about intensity or color *differences* seems more tractable [19] but as a differential measure this is subject to noise for small regions such as the eyes.

My approach is to use a relative measure whose support extended across a large fraction of the face. Paths between different points on the face are considered, where each path is assigned a cost that sums the distance of its individual pixels from a simple model of skin color (and since as mentioned above eyes are often recessed and relatively poorly illuminated from overhead light, intensity is also factored in). Paths which pass through an eye will have a higher cost than paths that detour around the eye. A Viterbi-based calculation is used to assign optimal paths to pairs of end-points on opposite sides of the face, searching over all paths that remain within the face and don't exceed some maximum curvature (see Figure 5). Each of these paths is computed from about one half of the pixels on the face. The paths are then combined to localize the eyes, which correspond to regions avoided by the paths. A series of heuristic tests based on the paths around avoided regions serve to distinguish actual eyes from regions that are simply not quite as attractive as the neighboring area.

Only pairs of avoided regions roughly aligned horizontally are considered. The regions give a reasonable estimate of the deviation from the horizontal of the angle between the eyes, which will be useful for initializing the roll. The location of the bridge of the nose serves as a useful origin for the surface of the face. The degree of symmetry can be estimated and used to see if the pose is close enough to zero yaw for initialization to be practical. Pitch is initialized by comparing the bridge location with the head outline as determined by the head tracker. The estimated size of the eyes, the distance between them, and the dimensions of the head outline all can contribute to an estimate of the size of the head (code for: none of them are at all reliable). The size estimate is only a relative measure across the interaction, since there is a scale factor that can't be recovered.

5 Pose tracking

This section develops a 6-dimensional coordinate system that is convenient for tracking, deferring until the next section how those coordinates relate to the rigid body parameters we really want. The goal was to develop a coordinate system that isolates the impact of estimates of the shape of the head as much as possible.

If an object being viewed does not rotate in depth but is otherwise free to explore a 3D space, there are four numbers that both describe the object's pose completely and are relatively easy to recover from the perspective projection of the object on the image plane. These are :-

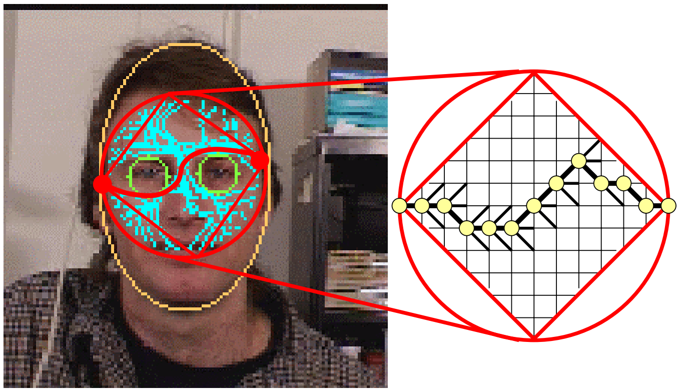

Fig. 5. How a path is found between two example end-points. A grid is laid down within an area between the end-points as shown. For the orientation shown in the figure, a path starts at the left, and moves through successive grid intersections as shown, always moving right and moving at most one step up or down (giving a maximum slope of 45°). By analogy with HMMs, each step to the right is a time step, each possible vertical level is a state, and the transition matrix is sparse with three entries per state. With the simple cost function described in the text, the optimal path can be efficiently computed with a Viterbi lattice structure. The grid resolution shown is artificially low for clarity. The path picked out on the left might just as easily have gone under or over both eyes, that is not relevant to localization.

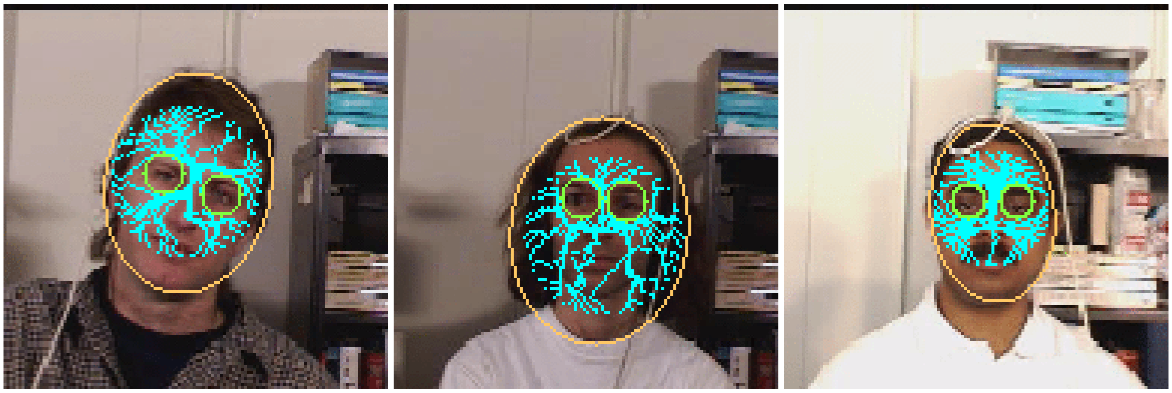

Fig. 6. Frontal pose being recognized for a number of individuals. As noted in the text, roll of the head is not problematic since it will be parallel to the image plane and so can be recovered directly from the angle the eyes make with the horizontal.

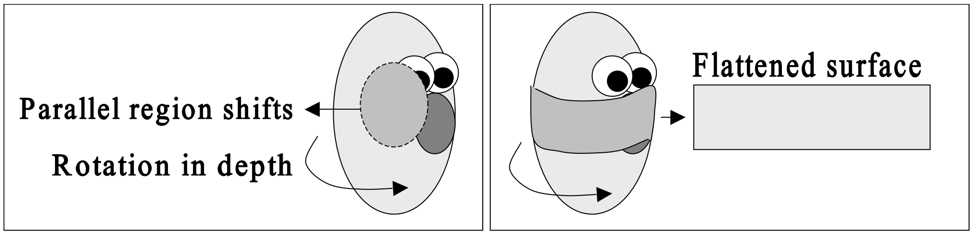

Fig. 7. Left: when the head rotates in depth, a different region on the head will become parallel to the image plane. Right: if the regions of the head that become parallel during a movement of the head do not explore the surface in 2D, then the surface they do explore can be thought of as Euclidean without running into contradictions (except for full 360° excursions).

- A position on the image plane, specified by two coordinates, giving a ray from the camera to the object.
- A coordinate specifying any in-plane rotation of the object, which is the only rotational freedom it has.
- A coordinate specifying a scaling of the projection of the object on the image plane.

These coordinates completely describe the object's pose in the sense that if the camera configuration is known, and *if the shape of the object is known*, the full 3D pose of the object can be recovered from these parameters. The need for the shape of the object arises from the implicit reference points of the coordinates – for example, the scaling coordinate can be converted to distance only with knowledge of the size of the object.

Once the object starts rotating in depth, there are two more degrees of freedom to factor in. The goal here to introduce them without destroying the simplicity of the image plane coordinates defined above. Importing some domain knowledge, assume the object being tracked is basically convex. Then at any moment there will be a unique region on the surface of the object that is close to parallel to the image plane. As the object rotates in depth, this region will shift to another part of the surface. We can parameterize where this region lies on the surface of the object using two dimensions. And since the region is (by construction) parallel to the image plane, the four coordinates developed earlier can be recast as follows :-

- Two coordinates that specify where the projection of the parallel region lies on the image plane.
- A coordinate specifying how the parallel region is rotated with respect to the image plane. This is the only rotational degree of freedom the parallel region has, by construction.
- A coordinate specifying a scaling of the parallel region (or equivalently of the projection of the entire object, as before).

Combined with two coordinates that determine what part of the surface of the object is currently parallel to the image plane, we have a 6-dimensional coordinate system that fully specifies the 3D pose of the object (if the shape of the object is known). This choice of coordinates has some virtues. In contrast to Euler angles, for example, the coordinates can be considered separately and in any order. This is least obvious for the rotation coordinate, but becomes clear if that coordinate is thought of as a counter-rotation of the camera about its optical axis.

A crucial issue that has not yet been addressed is what kind of coordinates are used to span the surface of the object being tracked. There are many possible coordinate systems for specifying a location on a convex surface – for example, latitude and longitude angles. The challenge here is to use coordinates that can be related to the projection of the object without knowledge of its 3D shape. There is no such magical coordinate system, so technically at this point the dimensions of the head have to be estimated before proceeding any further. But suspending disbelief for a moment, consider setting up a Euclidean coordinate system on the surface (which can be thought of as flattening the surface out onto a plane and then using standard rectangular coordinates). Of course, it isn't possible to flatten out the surface in this way without introducing inconsistencies. But if we do so anyway, then coordinates on the surface of the object that lie within the parallel region will map on to the image plane very simply, with just a scaling and an in-plane rotation. If we only ever try to relate coordinates within this region, then we can relate small steps in the image plane to small steps on the surface of the object, and so integrate the surface coordinates without needing to know the actual shape of the object.

The above discussion imposes two conditions :-

1. We must be able to determine what part of the projection of the object originated from a surface parallel to the image plane.
2. The path the parallel region traces across the surface of the object must lie within a strip that is thin relative to the curvature of the object. The wider the strip, the less Euclidean it is. The strip also must not make a full 360° excursion, no matter how thin it is.

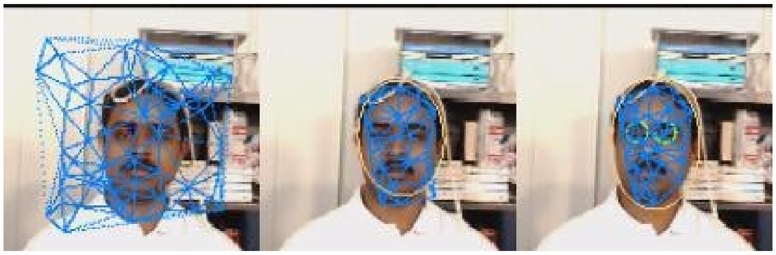

Fig. 8. Mesh initialization (apologies if you are viewing this in grey-scale). The mesh is initially distributed arbitrarily. It is pruned by the head outline when it is detected, and by heuristics based on the relative motion of different parts of the mesh. When frontal pose is detected, surface coordinates of the mesh can be initialized within the parallel region (that part of the face that is parallel to the image plane).

The first condition is tractable and will be addressed shortly. With regard to the second condition: in practice, the estimated curvature of the object should be factored in to the surface coordinate system, and this becomes an argument about what kinds of movements the accuracy of the estimate actually matters for. The answer is as might be expected: tracking accuracy is insensitive to the estimate of shape for movements combining in-plane translation, scaling (translation in depth), in-plane rotation, and rotation in depth for which all the surface patches made successively parallel to the image plane lie within a strip. This includes the important case of turning away, then turning back in an approximately symmetric manner.

To actually implement a pose tracking system based on this coordinate system, a mesh is laid down on the projecting of the head as illustrated in Figure 8. Nodes on the mesh are kept in correspondence to the face using simple template trackers, which are destroyed if they misbehave (measured by a set of consistency checks) and recreated elsewhere. Scaling, in-plane rotation, and in-plane translation are straightforward to compute from deformations of this mesh. As the head rotates in depth, some trackers will lose the support of the surface they are tracking as it becomes occluded, and so be destroyed. New parts of the head will become visible and have trackers assigned to their surface.

The mesh is used to maintain the surface coordinate system as follows. First, the parallel region is determined heuristically. If the translational component of motion can be eliminated, the parallel region can be identified easily because the flow due to rotation peaks there (since the motion of that surface is completely parallel parallel to the image plane). Translational motion can be accounted for by normalizing flow relative to the outline of the head. This crude procedure works better than it should because in practice translations and rotations of the head are often coupled so as to sum within the parallel region rather than cancel. Exceptions include pure rolls and translations in depth. The extent of the parallel region is chosen to scale in a heuristic way with the head outline, since in theory it should be infinitesimally small but in practice it has to be assigned some extent to be useful. And luckily, surface distortions such as the nose don't seem to cause trouble.

The parallel region can be seen as a mask overlaid on the image, within which it is safe to relate image coordinates and surface coordinates. Pose recognition events, detected in the manner described in the previous section, are used to choose an origin on the surface, and an initial translation, scaling and (in-plane) rotation of the surface coordinate system with respect to the image plane. This association is represented by assigning surface coordinates to points on the mesh that lie within the parallel region, augmenting the image plane coordinates they jointly possess. As the parallel region shifts during a rotation in depth, new points entering the region are assigned surface coordinates based on their image plane coordinates, with the transformation between the two easy to maintain using the rotation, scaling, and translation of the mesh already recovered.

Independently of the argument given earlier for the types of movements that can be tracked without accurate knowledge of the shape of the head, the mesh allows a new set of trajectories to be tracked: those which leave some portion of the face visible throughout. The surface coordinates of points on the mesh covering that part of the face can be used as landmarks.

6 3D pose recovery

Recovery of the 3D location of the head is straightforward, given knowledge of the camera's parameters, although there is of course a scale/depth ambiguity since no absolute depth information is recovered.

Recovery of 3D orientation is equally straightforward, but shape dependent. The output of the tracker is effectively a procedure for turning a specified point on the surface of the object towards the camera and then rotating

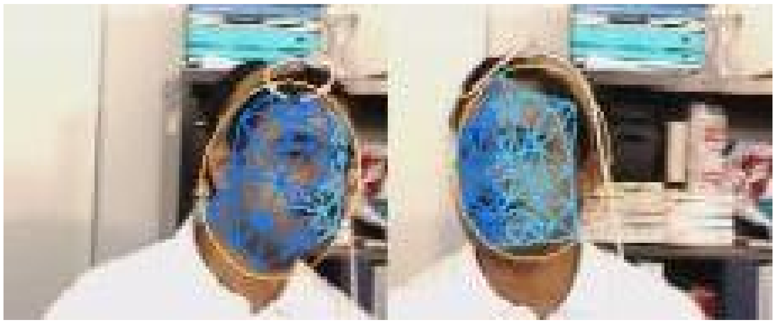

Fig. 9. A visualization of surface coordinates on the mesh. The mesh has been colored here based on the sign of a surface coordinate, so that it appears as two halves locked onto either side of the face.

it to a specified degree. To convert this into Euler angles, for example, requires knowledge of the shape of the object so that surface points can be associated with vectors from wherever the center of the head is taken to be. At this point, we must make use of the estimates for the dimensions of the head from the head tracker and make the conversion using a simple ellipsoidal model. The crucial point is that inaccuracies in this process do not feed back to the tracker itself.

7 Results

The system was implemented to run on a network of computers associated with the robot Kismet [5], under the QNX real-time operating system. Three 800MHz machines were required to run the system at frame rate (which reflects a sublime indifference to optimization on the part of the programmer rather than fundamental algorithmic complexity). The system also ran on a single 400MHz Linux box at about 2 frames per second.

The system was tested on the data-set made available by Sclaroff et al [7], consisting of video of head movements with ground truth measured by a Flock of Birds sensor on the subjects' heads. These sequences are 200 frames in duration. To test the stability of the tracker over long intervals, the Sclaroff sequences are here artificially extending by looped them forward and back for twenty iterations. Figure10 shows tracking results for the sequence which appeared to have the largest rotation in depth (in no case unfortunately did the eyes become occluded, which would have made for a better demonstration of the advantages of the system developed in this paper). Angular measurements are limited by the accuracy with which they can be initialized, which turns out to be to within about 5°for roll and yaw, and about 10° for pitch. Because of re-initialization events, estimates of pose will contain discontinuities when drift is corrected, which is not brought out in the figure. This could be dealt with for estimation of pose across a pre-recorded video sequence like this one, but for use in a vision interface it seems the discontinuities are unavoidable. This is because the best estimate of the current pose does truly change instantaneously when an initialization even occurs, and there is no point propagating information backwards to previous frames during real-time interaction unless there is some background processing going on that can have high latency.

8 Discussion and Conclusions

This paper has demonstrated that limited mechanisms of pose recognition can be integrated with a pose tracker to give stable pose estimates over long periods of time. It has also examined how accurately the estimated dimensions of the head need to be for tracking to succeed, and found tractable classes of head motion that in fact span most of those seen in face-to-face human/robot interaction. Counter-examples where accurate knowledge of dimensions is vital would be any 360° excursion, or a very large "roll" of the head on its neck, for which very little of the head remains in view at all times and which will not be correctly identified as a closed path if the dimensions of the head are not accurately known. It is plausible that for large excursions, since such widely varying views of the head are presented, enough information is available visually to accurately pin down the shape of the head.

The project diverged from what was proposed in many ways. I had hoped to use stereo information, but this turned out to be problematic for purely mechanical reasons. I also proposed using an ellipsoidal model for the head, but it seems difficult to really make that work for people with arbitrarily voluminous hair-styles. And finally, I had planned to reuse existing code available for the Kismet platform, but in the end all the code described in this project was developed specifically for it, beyond a basic image processing library.

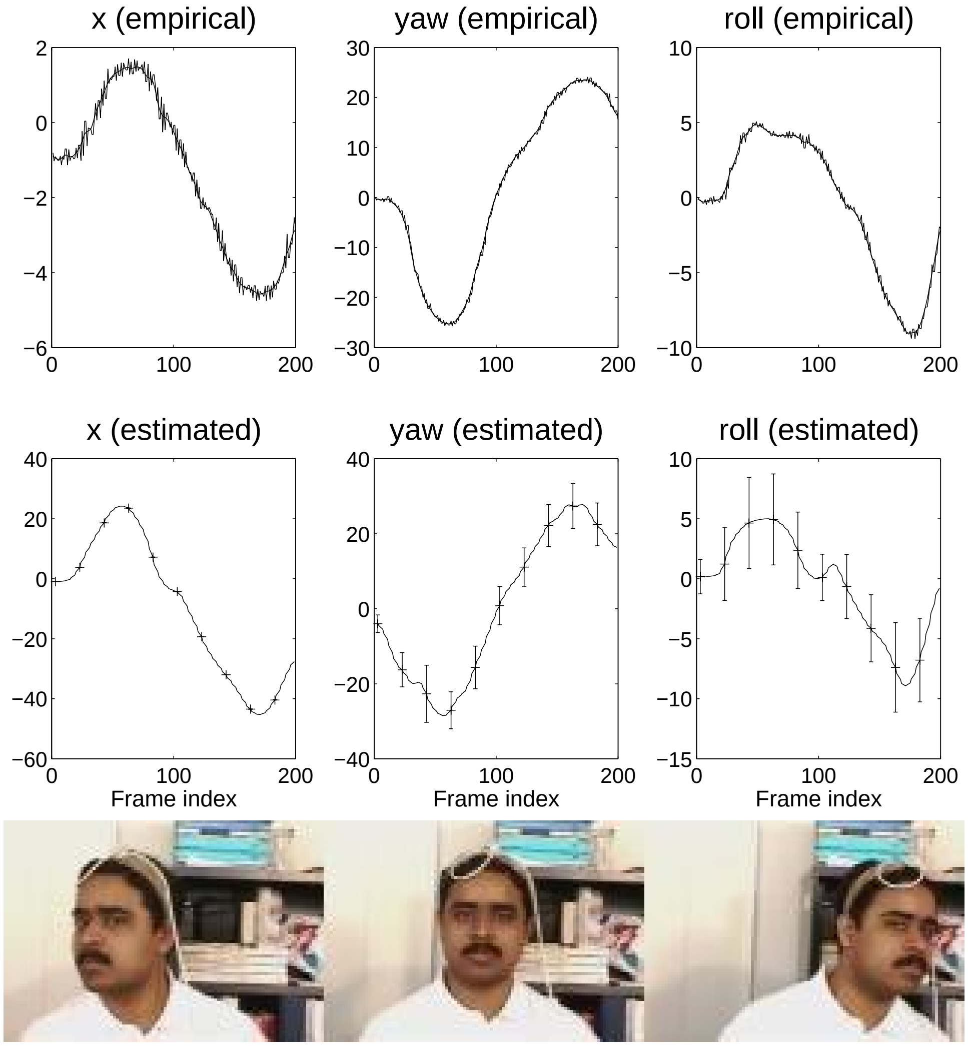

Fig. 10. Results for a sequence containing a yaw movement and horizontal translation, with all other parameters remaining basically unchanged except for a slight roll. The top row shows ground truth. The second row shows the estimated pose parameters that change significantly during the sequence. The estimated x coordinate is left in terms of the image plane. Values plotted are averaged for each occurrence of a particular frame over a *single tracking run* constructed from a sequence being played, then played in reverse, then repeated again for twenty iterations. Error bars show the standard deviation of estimates for each frame. There is about a 5° error in angles, which in this case means the roll estimate is mostly noise.

References

[1] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *Intl. Conf. on Pattern Recognition*, Vienna, Austria, 1996.

[2] P. A. Beardsley. A qualitative approach to classifying head and eye pose. In *IEEE Workshop on Applications of Computer Vision*, pages 208–213, Florence, Italy, October 1998.

[3] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *CVPR*, pages 232–237, 1998.

[4] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *ICCV*, pages 374–381, 1995.

[5] C. Breazeal. *Sociable Machines: Expressive Social Exchange Between Humans and Robots*. PhD thesis, MIT Department of Electrical Engineering and Computer Science, 2000.

[6] C. Breazeal, A. Edsinger, P. Fitzpatrick, B. Scassellati, and P. Varchavskaia. Social constraints on animate vision. *IEEE Intelligent Systems*, 15, July/August 2000.

[7] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 22(4), April 2000.

[8] Q. Chen, H. Wu amd T. Shioyama, and T Shimada. 3d head pose estimation using color information. In *6th IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, June 1999.

[9] M. Cordea, E. Petriu, N. Georganas, D. Petriu, and T. Whalen. Real-time 2.5d head pose recovery for model-based video-coding. In *IEEE Instrumentation and Measurement Technology Conference*, Baltimore, MD, USA, May 2000.

[10] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *CVPR*, pages 231–238, 1996.

[11] M. Harville, A. Rahimi, T. Darrell, G. Gordon, and J. Woodfill. 3d pose tracking with linear depth and brightness constraints. In *ICCV*, pages 206–213, 1999.

[12] J. Heinzmann and A. Zelinsky. Robust real-time face tracking and gesture recognition. In *Proc. International Joint Conference on Artificial Intelligence*, volume 2, pages 1525–1530, August 1997.

[13] T. Horprasert, Y. Yacoob, and L. S. Davis. An anthropometric shape model for estimating head orientation. In *3rd International Workshop on Visual Form*, Capri, Italy, May 1997.

[14] S. McKenna and S. Gong. Real-time face pose estimation. *International Journal on Real Time Imaging, Special Issue on Real-time Visual Monitoring and Inspection.*, 4:333–347, 1998.

[15] M. Pilu, A. Fitzgibbon, and R. Fisher. Ellipse-specific direct least-square fitting. In *IEEE International Conference on Image Processing*, Lausanne, September 1996.

[16] J. Sherrah and S. Gong. Fusion of perceptual cues for robust tracking of head pose and position. In *to appear in Pattern Recognition*, 2000.

[17] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593 – 600, 1994.

[18] L. Sigal, S. Sclaroff, and V Athitsos. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In *CVPR*, volume 2, pages 152–159, 2000.

[19] Pawan Sinha. Object recognition via image invariants: A case study. *Investigative Ophthalmology and Visual Science*, 35:1735–1740, May 1994.

[20] J. Smith-Mickelson. Design and application of a head detection and tracking system. Master's thesis, MIT, 2000.

[21] J. Strom, T. Jebara, S. Basu, and A. Pentland. Real time tracking and modeling of faces: An ekf-based analysis by synthesis approach. In *Modelling People Workshop, ICCV*, 1999.

[22] C. Wang and M. Brandstein. A hybrid real time face tracking system. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing.*, Seattle, 1998.

[23] C. Wren, A. Azarbayejani, T. Darrell, and P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 1997.

[24] K. Toyama Y. Wu and T. S. Huang. Wide-range, person- and illumination-insensitive head orientation estimation. In *Proc. of Int'l Conf. on Face and Gesture Recognition*, Grenoble, France, March 2000.