CHAPTER 8

The final frontier: working in space, keeping track of objects

"I don't actually think," Ponder Stibbons said gloomily, "that I want to tell the Archchancellor that this machine stops working if we take its fluffy teddy bear away. I just don't think I want to live in that kind of world."
"Er, you could always, you know, sort of say it needs to work with the FTB enabled."
(Pratchett, 1996)

Objects live in 3D space. They move around and change pose. This chapter introduces an array of interconnected methods to track objects and their locations, from low-level visual attention to egocentric maps.

8.1 Overall approach

This work extends and overlaps with previous efforts on Cog and Kismet by Scassellati (2001), Breazeal (2000), and the author. Our work on attention has addressed active vision in a social setting (Breazeal et al., 2001) and a framework for using social cues to the advantage of the robot's vision system (Breazeal and Fitzpatrick, 2000). Breazeal and Scassellati (1999) implemented a model of low level visual attention based on measures of intrinsic salience (brightness, motion etc.) with some behavioral modulation. The author extended that work to introduce a mechanism for controlling persistence – to be able to deliberately hold gaze on an object, and dynamically control the trade-off between being responsive to changes in the environment and being a complete slave to fluctuations in it. This chapter goes further to encompass many more influences on attention (see Figure 8-1), such as object recognition and spatial memory. Figure 8-1) shows the basic architecture of how the robot's gaze is controlled. If influences are coming from modules that operate at different speeds and with different levels of noise versus stability, and the robot's head and eyes were under direct control of these modules, then its behavior will be very erratic indeed. Instead, all visual control of the robot's gaze direction is mediated by a tracking module which operates at the fastest rate possible (30Hz). This means that the robot can respond to fast moving objects even if not all parts of the vision system can react quickly.

Since this work is implemented on a humanoid robot, it is crucial that the robot carefully controls its own gaze to convey the most veridical impression of its state to the human. It helps if the robot moves its eyes in a manner consistent with human eye movement.

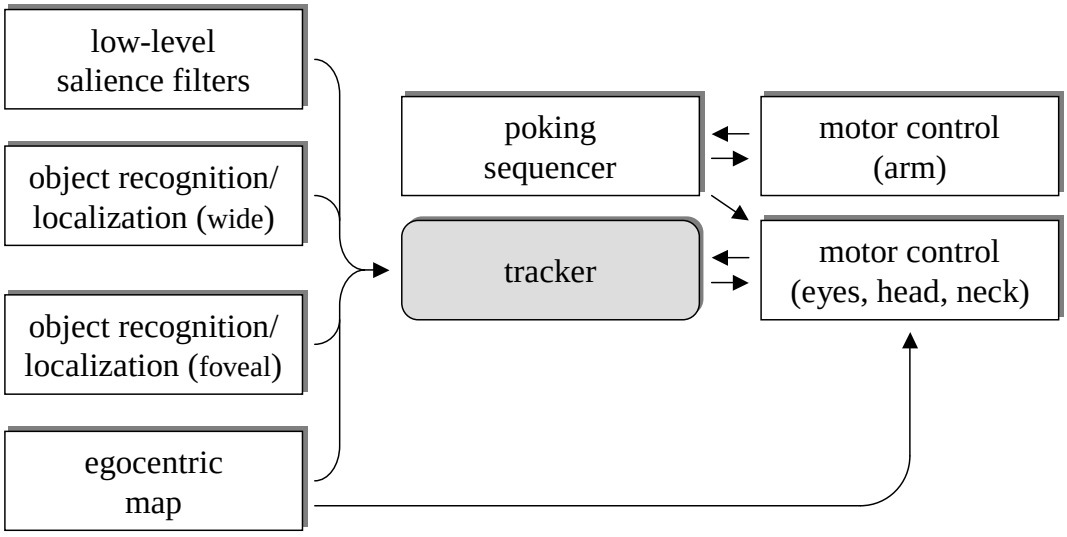


Figure 8-1: Influences on gaze. Object recognition subsumes the low-level salience filters.

Also, spatial 'awareness' is important for tasks, since even for adult humans cognition can often be traded off with physical space (Kirsh, 1995b; Pelz, 1995).

8.2 Human-like gaze

The eye movement primitives implemented on Kismet and Cog are modeled after the human ocular-motor system. The human system is so good at providing a stable percept of the world that we have little intuitive appreciation of the physical constraints under which it operates. Humans have foveate vision. The fovea (the center of the retina) has a much higher density of photoreceptors than the periphery. This means that to see an object clearly, humans must move their eyes such that the image of the object falls on the fovea. Human eye movement is not smooth. It is composed of many quick jumps, called saccades, which rapidly re-orient the eye to project a different part of the visual scene onto the fovea. After a saccade, there is typically a period of fixation, during which the eyes are relatively stable. They are by no means stationary, and continue to engage in corrective micro-saccades and other small movements. If the eyes fixate on a moving object, they can follow it with a continuous tracking movement called smooth pursuit. This type of eye movement cannot be evoked voluntarily, but only occurs in the presence of a moving object. Periods of fixation typically end after some hundreds of milliseconds, after which a new saccade will occur (Goldberg, 2000). The eyes normally move in lock-step, making equal, conjunctive movements. For a close object, the eyes need to turn towards each other somewhat to correctly image the object on the foveae of the two eyes. These disjunctive movements are called vergence, and rely on depth perception (see Figure 8-2). Since the eyes are located on the head, they need to compensate for any head movements that occur during fixation. The vestibulo-ocular reflex uses inertial feedback from the vestibular system to keep the orientation of the eyes stable as the eyes move. This is a very fast response, but is prone to the accumulation of error over time. The opto-kinetic response is a slower compensation mechanism that uses a measure of the visual slip of the image across the retina to correct for drift. These two mechanisms work together to give humans stable gaze as the head moves.

The ocular-motor system implemented on Cog and Kismet is an approximation of the human

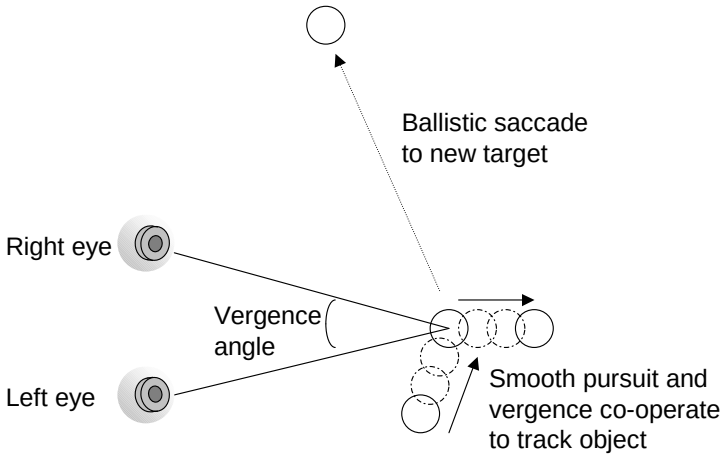


Figure 8-2: Humans exhibit four characteristic types of eye motion. Saccadic movements are high-speed ballistic motions that center a target in the field of view. Smooth pursuit movements are used to track a moving object at low velocities. The vestibulo-ocular and opto-kinetic reflexes act to maintain the angle of gaze as the head and body move through the world. Vergence movements serve to maintain an object in the center of the field of view of both eyes as the object moves in depth. The robotic eye movements are controlled to mirror this pattern.

system. Kismet's eyes periodically saccade to new targets chosen by the targeting system, tracking them smoothly if they move. On Kismet, vergence eye movements are not made, since with the scale of the head the robot ends up looking disturbingly cross-eyed even if it is correctly verged. On Cog, a disparity measure computed by comparing left and right camera views drives vergence movements which simply add to whatever tracking movements are being made – the conjunctive and disjunctive channels operate independently. An analogue of the vestibular-ocular reflex has been developed for Cog using a 3-axis inertial sensor. A crude approximation of the opto-kinetic reflex is performed by our implementation of smooth pursuit.

8.3 Social gaze

Apart from functional constraints, eye movements in humans also have communicative value. This needs to be considered when designing humanoid robots. To a human observer, the robot's eyes indicate its locus of attention. The robot's degree of engagement can also be conveyed, to communicate how strongly the robot's behavior is organized around what it is currently looking at. If the robot's eyes flick about from place to place without resting, that indicates a low level of engagement, appropriate to a visual search behavior. Prolonged fixation with smooth pursuit and orientation of the head towards the target conveys a much greater level of engagement, suggesting that the robot's behavior is very strongly organized about the locus of attention. Kismet makes a great deal of use of gaze and head/neck posture to convey state, the goal being to give a cooperative human natural cues as to how they could help the robot. For example, Kismet has a number of coordinated motor actions designed to deal with various limitations of Kismet's visual perception (see Figure 8-3). If a person is visible, but is too distant for their face to be imaged at adequate resolution, Kismet engages in a calling behavior to summon the person closer. People who come too close to the robot also cause difficulties for the cameras with narrow fields of view, since only a small part of a face may be visible. In this circumstance, a withdrawal response is invoked, where Kismet draws back

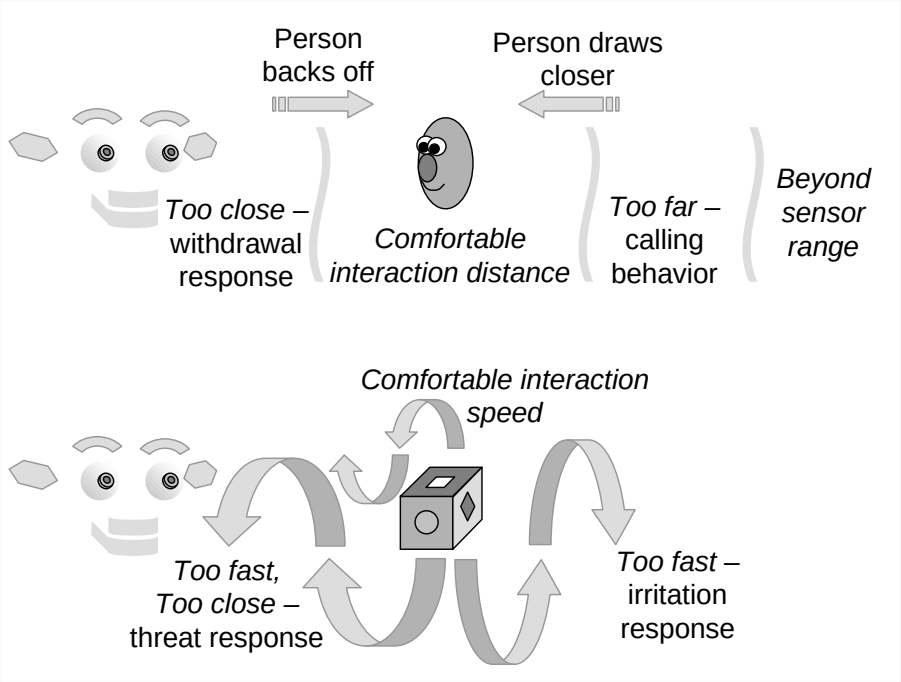


Figure 8-3: Regulating interaction. People too distant to be seen clearly are called closer; if they come too close, the robot signals discomfort and withdraws. The withdrawal moves the robot back somewhat physically, but is more effective in signaling to the human to back off. Toys or people that move too rapidly cause irritation.

physically from the person. This behavior, by itself, aids the cameras somewhat by increasing the distance between Kismet and the human. But the behavior can have a secondary and greater effect through "social amplification" – for a human close to Kismet, a withdrawal response is a strong social cue to back away, since it is analogous to the human response to invasions of personal space. Similar kinds of behavior can be used to support the visual perception of objects. If an object is too close, Kismet can lean away from it; if it is too far away, Kismet can crane its neck towards it. Again, in a social context, such actions have power beyond their immediate physical consequences. A human, reading intent into the robot's actions, may amplify those actions. For example, neck-craning towards a toy may be interpreted as interest in that toy, resulting in the human bringing the toy closer to the robot. Another limitation of the visual system is how quickly it can track moving objects. If objects or people move at excessive speeds, Kismet has difficulty tracking them continuously. To bias people away from excessively boisterous behavior in their own movements or in the movement of objects they manipulate, Kismet shows irritation when its tracker is at the limits of its ability. These limits are either physical (the maximum rate at which the eyes and neck move), or computational (the maximum displacement per frame from the cameras over which a target is searched for). Here we see the visual-motor system being driven by the requirements of a nominally unrelated sensory modality, just as behaviors that seem completely orthogonal to vision (such as ear-wiggling during the call behavior to attract a person's attention) are nevertheless recruited for the purposes of regulation. These mechanisms also help protect the robot. Objects that suddenly appear close to the robot trigger a looming reflex, causing the robot to quickly withdraw and appear startled. If the event is repeated, the response quickly habituates and the robot simply appears annoyed, since its best strategy for ending these repetitions is to clearly signal that they are unde-

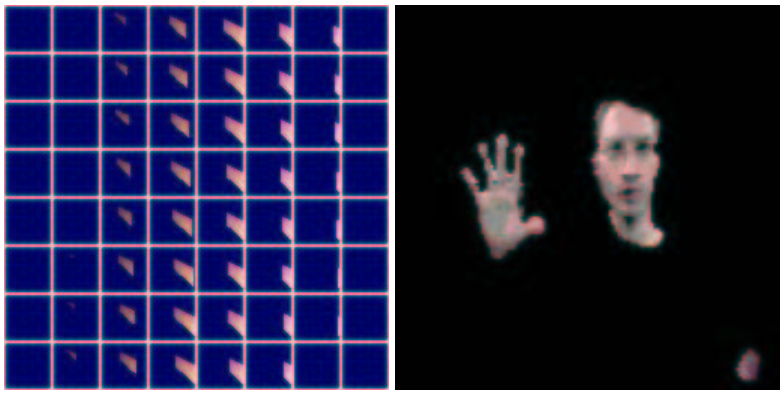


Figure 8-4: The skin tone filter responds to 4.7% of possible (R, G, B) values. Each grid in the figure to the left shows the response of the filter to all values of red and green for a fixed value of blue. The image to the right shows the filter in operation. Typical indoor objects that may also be consistent with skin tone include wooden doors, cream walls, etc.

sirable. Similarly, rapidly moving objects close to the robot are threatening and trigger an escape response. These mechanisms are all designed to elicit natural and intuitive responses from humans, without any special training. But even without these carefully crafted mechanisms, it is often clear to a human when Kismet's perception is failing, and what corrective action would help, because the robot's perception is reflected in behavior in a familiar way. Inferences made based on our human preconceptions are actually likely to work.

In general, motor control for a humanoid robot poses challenges beyond issues of stability and accuracy. Motor actions will be perceived by human observers as semantically rich, regardless of whether the imputed meaning is intended or not. This can be a powerful resource for facilitating natural interactions between robot and human, and places constraints on the robot's physical appearance and movement. It allows the robot to be readable – to make its behavioral intent and motivational state transparent at an intuitive level to those it interacts with. It allows the robot to regulate its interactions to suit its perceptual and motor capabilities, again in an intuitive way with which humans naturally cooperate. And it gives the robot leverage over the world that extends far beyond its physical competence, through social amplification of its perceived intent.

8.4 Visual attention

Cog and Kismet have attention systems that are designed to attract the robot towards features that humans find visually salient (Nothdurft, 1993), such as motion, the presence of skin tone, bright colors, and size. The advantage of doing so is that in interactive situations people may intuitively provide the right cues to direct the robot's attention, such as shaking an object, moving it closer, waving their hand, and so on (see Figures 8-5, 8-6 and 8-7). The attention system can also take care of tasks that need very fast response times. For example, on Kismet looming objects are detected pre-attentively using a measure of optic flow expansion, to facilitate a fast reflexive withdrawal. The output of low-level feature detectors for color and motion are combined through a weighted average to produce a single attention map. This combination allows the robot to select regions that are visually salient and to direct its computational and behavioral resources towards those regions.

The operation of a representative low level feature detector is shown in Figure 8-4). This filter is a simple skin tone detector, a computationally inexpensive means to find regions which may

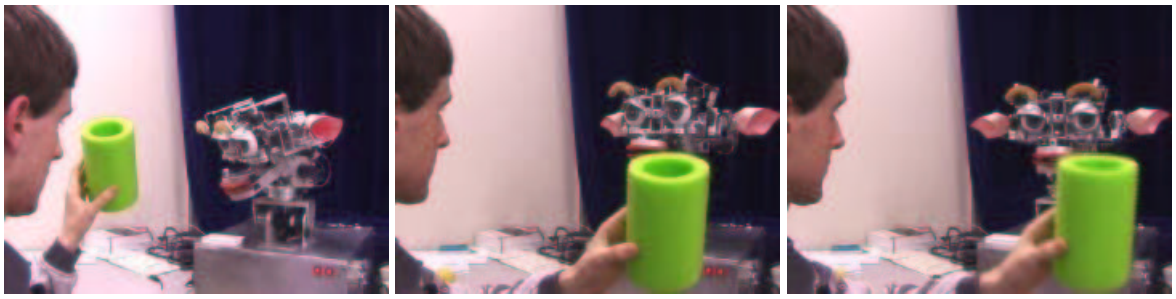


Figure 8-5: These images are from a sequence in which the instructor wanted the robot to attend to the green object as it moved away from a central location. In the first image the robot is clearly attending; in the second it just as clearly has become fixated on the instructors face. Knowing this prompted the instructor to wave the object a little until it regained the robot's attention.

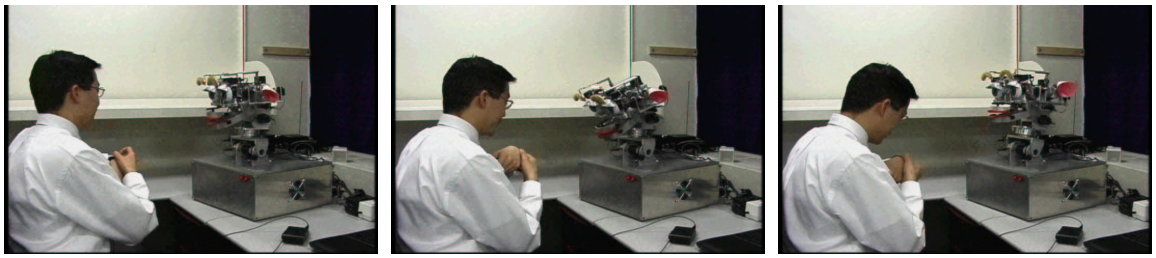


Figure 8-6: Kismet interacting with a test subject unacquainted with the details of the robot's implementation and behavior. The subject decides to show Kismet his watch, so he brings it forward and taps it (left). Motion plus size plus skin-color makes this an attractive target and Kismet looks at it (center). Once the motion stops, Kismet looks away (right).

contain faces or hands. Such filters are of course very imperfect, but if no other information is available they are better than simply staring at the wall or ceiling for hours on end (which seems to be what all robots do be default). Higher level modules can compensate for their limitations when the opportunity arises – for example, both Cog and Kismet use the frontal face detector developed by Viola and Jones (2001).

Figure 8-7: Manipulating low-level attention. Images on the top row come directly from the robot's camera. Images on the bottom summarize the contemporaneous state of the robot's attention system. Brightness in the lower image corresponds to salience; rectangles correspond to regions of interest. The thickest rectangles correspond to the robot's locus of attention. In the first pair of images, the robot (Kismet) is attending to a face and engaging in mutual regard. By shaking the colored block, its salience increases enough to cause a switch in the robot's attention. The third pair of images shows that the head tracks the toy as it moves, giving feedback to the human as to the robot's locus of attention. The eyes are also continually tracking the target more tightly than the neck does. In the fourth pair of images, the robot's attention switches back to the human's face, which is tracked as it moves.

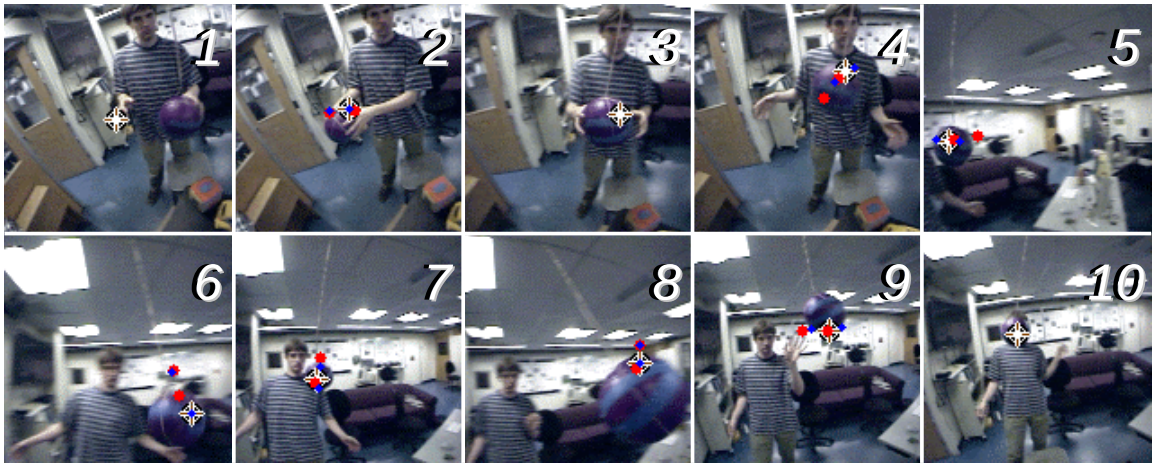


Figure 8-8: The tracking system has an inbuilt preference for moving objects. This mean that if the robot is staring at something, and you want to bring the robot's attention to it, then you can simply move the object in front of the robot (frames 1 to 2). This figure shows frames from a 9 second sequence of the robot (Cog) tracking a rapidly moving ball suspended from the ceiling by elastic until it crashes into the author's face. The crosshair shows the current tracked target. Notice the wide range of motion (compare frames 1, 5, 10) and distance to the ball (compare frames 7 and 8) over this short sequence. The tracker continually looks several pixels forward and backward along its direction of motion and then shifts to whichever location is moving most rapidly. This keeps the tracker on the target, although it will not stay locked to a specific part of the target.

8.5 Maintaining gaze

In the presence of multiple salient objects, it is useful be able to commit attention to one of the objects for a period of time. This gives time for post-attentive processing to be carried out on the object, and for downstream processes to organize themselves around the object. On Kismet, an example of this is visual search – the robot scans across the visual field, dwelling long enough at each point of fixation to decide whether the fixated object is behaviorally relevant (for example, it may lack eyes, which are searched for post-attentively). On Cog, the robot needs to keep an object fixated for some time for vergence to stabilize and to get an accurate measure of its distance. Committing to an object is also useful for behaviors that need to be atomically applied to a target. On Kismet, an example is a calling behavior where the robot needs to stay looking at the person it is calling for the gesture to be socially readable. On Cog, while poking an object the robot needs to maintain fixation.

To allow such commitment, the attention system is augmented with a tracker. The tracker follows a target in the visual field, using simple correlation between successive frames. Usually changes in the tracker target will be reflected in movements of the robot's eyes, unless this is behaviorally inappropriate. If the tracker loses the target, it has a good chance of being able to reacquire it from the attention system. Figures 8-8 and 8-9 shows the tracker in operation on Cog and Kismet respectively.

Figure 8-9: Behavior of the tracker over a longer period (on Kismet). Frames are taken at one second intervals. The white squares indicates the position of the target. The target is not centered in the images since they were taken from a camera fixed with respect to the head. On the third row, the face slips away from the tracker, but it is immediately reacquired through the attention system. The images are taken from a three minute session during which the tracker slipped five times. This is typical performance for faces, which tend not to move too rapidly.

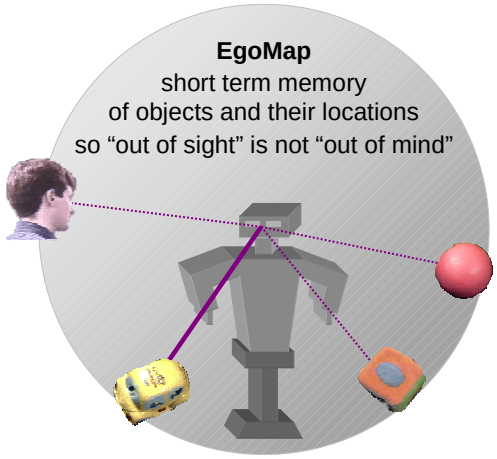


Figure 8-10: The egocentric map or ‘egomap.’ Locations of objects are tracked with respect to the robot.

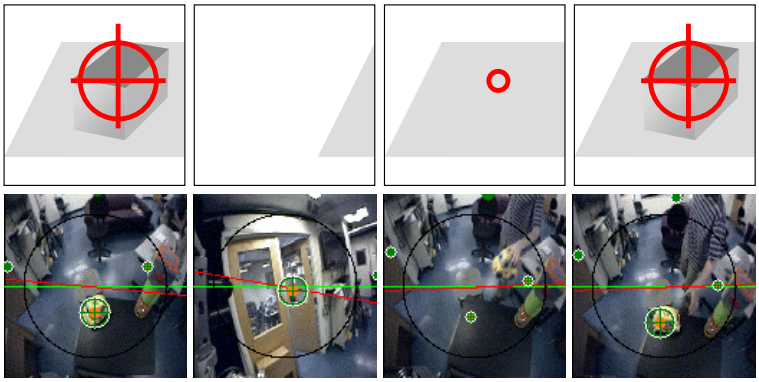
8.6 Maintaining an egocentric map

Low-level attention and visual tracking have no memory for objects – once something goes out of view, it is completely forgotten. This does not make for a very useful robot. To rectify this deficit, a map of objects the robot has observed recently is maintained. The map is egocentric; locations are expressed relative to the robot’s own position. In fact, the distance to objects is ignored and only the direction to them is stored. The purpose of the map is to allow the robot to return its gaze to an object it has previously observed (even if that object has left the field of view), and this does not require knowledge of distance unless an object is extremely close. A kinematic model of the head is used to computed eye gaze relative to the robot’s torso when it is fixating an object. If the object is familiar to the robot from poking, its identity is stored in a two-dimensional grid of bins indexed by spatial dimensions similar to latitude and longitude (see Figure 8-10), along with the time of observation. Later, if the robot needs to refixate an object, it consults the grid for the last observation of the object, and then directs the robot’s gaze to turn to approximately the right direction. Once there, the object recognition module is requested to locate the object in question, and the robot will fixate it precisely.

8.7 Flat-track compound coordinate system

Another important aspect of objects is their pose. This section develops a pose tracking mechanism designed for objects that can be recognized in one pose but not more generally. This is an important scenario for this thesis since poking only offers segmented views of the part of an object facing the camera. Head tracking will initially be treated. It is a useful case in and of itself, and is suitable because face detection is currently better developed for frontal presentations than arbitrary pose. Another reason to work with head tracking is that there are data sets available for evaluating the fidelity head pose estimation.

In 3D space, there are six dimensions associated with the pose of a rigid object – three translational, and three rotational. When tracking objects using a camera, changes in some of these pose dimensions can be difficult to recover accurately. One approach to deal with this is to have a strong model of the object being tracked, which has been particularly successful for head tracking (see for example Black and Yacoob (1995)). The shape of the human head is broadly similar across the



Location marked, Target present | *Robot looks away* | *Robot looks back, Target is gone* | *Target reappears*

Figure 8-11: Keeping track of locations. Circles with cross-hairs represent locations that contain a particular object. If the object is removed, this is detected using color histograms (Swain and Ballard, 1991), and is indicated by a small circle without a cross-hair. The upper row is a cartoon sequence to illustrate what is happening in the views below, which are taken directly from Cog's egocentric map. Initially a yellow car is present on the table in front of Cog. The robot looks away to the door, and when it looks back, the car is no longer present. It then reappears and is immediately detected. This behavior, along with object tracking (which has also been implemented), give the basics of a representation of the robot's workspace.

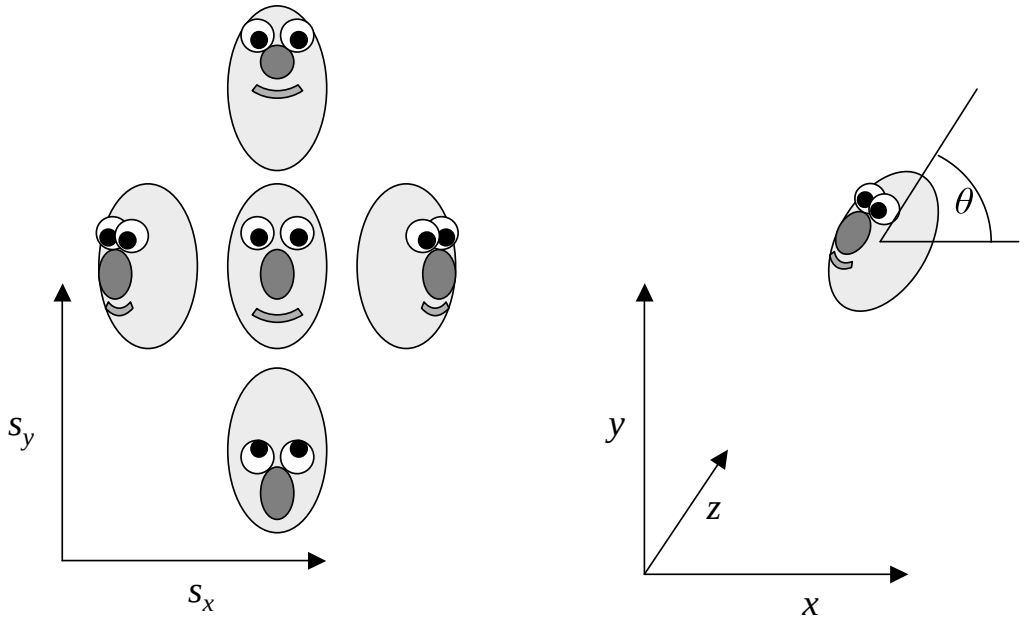
species. Anthropometry characterizes the distribution of face length scales and ratios within different sub-groups. These distributions are quite narrow for a subject whose gender, race, and age are known. Horprasert et al. (1997) make use of this to estimate head orientation from monocular images. They 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 (Harville et al., 1999). 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 (McKenna and Gong, 1998). And then there are very many systems designed to run in real-time, using a wide variety of simple cues such as hair outline (Wang and Brandstein, 1998).

Some representations are more prone to accumulated errors than others, and there has been considerable research on good coordinate systems for tracking under various situations. Yet another one is introduced here, with the goal being to minimize the effect of mis-estimation of object shape on tracking, and to allow for opportunistic calibration whenever a known view of the object is observed. If a rigid object being viewed by a camera does not rotate in depth but is otherwise free to explore a 3D space, the object's pose can be specified completely with just four numbers :-

- ▷ A position on the image plane, specified by two coordinates, giving a ray from the camera to a particular point on the object.
- ▷ A coordinate specifying any rotation of the object in the plane parallel to the camera, which is the only rotational freedom the object has, given that it cannot (for now) rotate in depth.

▷ A coordinate specifying a scaling of the projection of the object on the image plane, or any other scalar such as size or metric distance to the object.

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.



Non-planar component:
What part of the object is facing towards the camera?

Surface coordinate s_x, s_y (2D)

Planar component:
Where is the camera-facing surface located, and how is it oriented?

Retinotopic coordinate x, y (2D)
Distance/size measure z (1D)
In-plane orientation θ (1D)

Figure 8-12: The 'flat-track' compound coordinate system. One component is a surface coordinate on the object being tracked (left). The second component are the degrees of freedom of a flat surface facing the camera – x and y position, a measure of scale or distance, and an in-plane rotation.

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. Suppose the object being tracked is basically convex (if it has a more awkward shape, there is a good chance that pose can be recognized from its silhouette directly). Then at any moment there will ideally 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

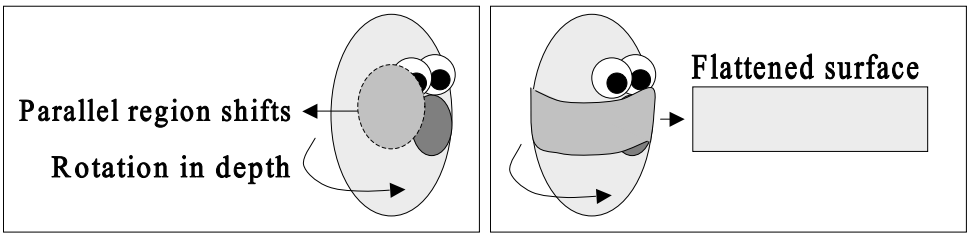


Figure 8-13: Left: when an object rotates in depth, a different region of the object will become parallel to the image plane. Right: if the regions of the object that become parallel during a movement of the head do not explore the surface in 2D, then the surface they do explore may be thought of as Euclidean without running into serious contradictions (except for full 360° excursions).

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 objects 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.

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.

8.7.1 Pose detector

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 (Wang and Brandstein, 1998) 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.

Frontal pose is the option adopted here. 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.

The outline of the head is tracked using a collection of techniques that seem to be typical of real-time systems (Cordea et al., 2000), such as image differencing and ellipse-fitting. The implementation described here is qualitatively similar to that of Smith-Mickleson (Smith-Mickelson, 2000), and also traces back to Birchfield's work (Birchfield, 1998).

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 scanlines). 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 (Pilu et al., 1996). Figure 8-14 shows an example of this.

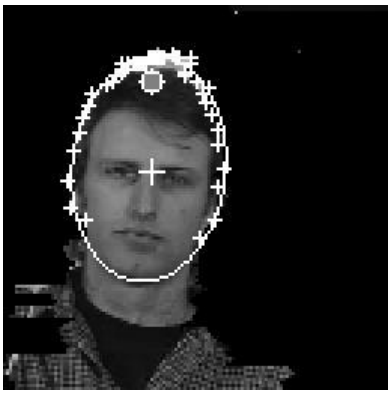


Figure 8-14: Finding the outline of the head once it has been detected. Probes (shown as small crosses) radiate out from a location near the top of the head/body (shown as a circle). Across a large range of scales, the silhouette of the head can be extracted from the contour points encountered by the probes, and matched against an elliptic model.

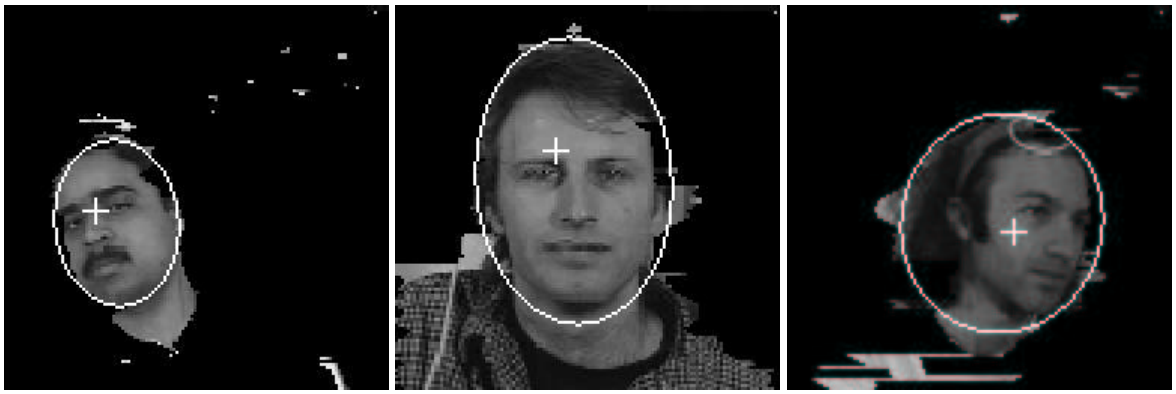


Figure 8-15: Snapshots of the head tracker in action. Hair is problematic. Sometimes it will be included, sometimes not. In individuals with a great deal of hair (rightmost figure), the ellipse can deviate a great deal from the basic shape of the head itself.

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 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 8-15 shows examples of the kind of variation this can lead to in what the putative "head outline" actually means.

At the low resolutions real-time performance mandates, many researchers 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 (Sinha, 1994) but as a differential measure this is subject to noise for small regions such as the eyes.

The approach adopted here 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

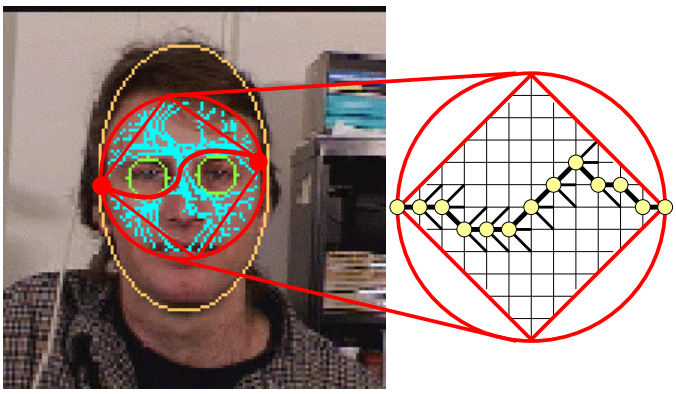


Figure 8-16: How a path is found between two example end-points. A grid is laid down within an area between the end-points as shown. For the orientation shown in the figure, a path starts at the left, and moves through successive grid intersections as shown, always moving right and moving at most one 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.

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 8-16). 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.

To actually implement a pose tracking system based on this coordinate system, a mesh is laid down on the projection of the head as illustrated in Figure 8-18. 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.

Figure 8-17: 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.

Figure 8-18: Mesh initialization. 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).

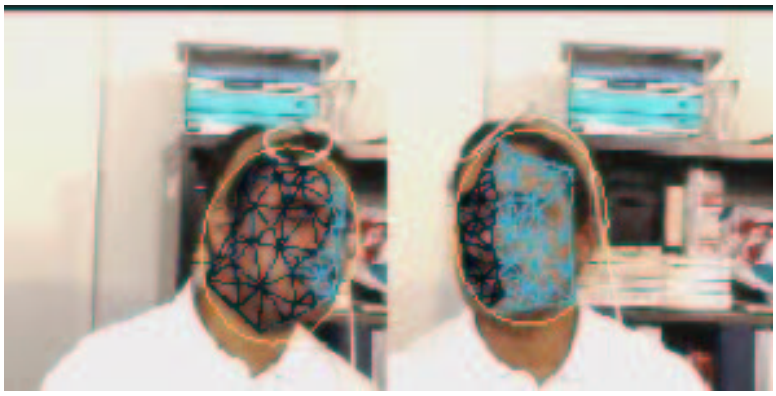


Figure 8-19: A visualization of surface coordinates on the mesh. The mesh has been colored here based on the sign of a surface coordinate, so that it appears as two halves locked onto either side of the face.

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.

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 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.

8.7.2 An evaluation

The system was tested on a data-set made available by Sclaroff et al (La Cascia et al., 2000), 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. Figure 8-20 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.

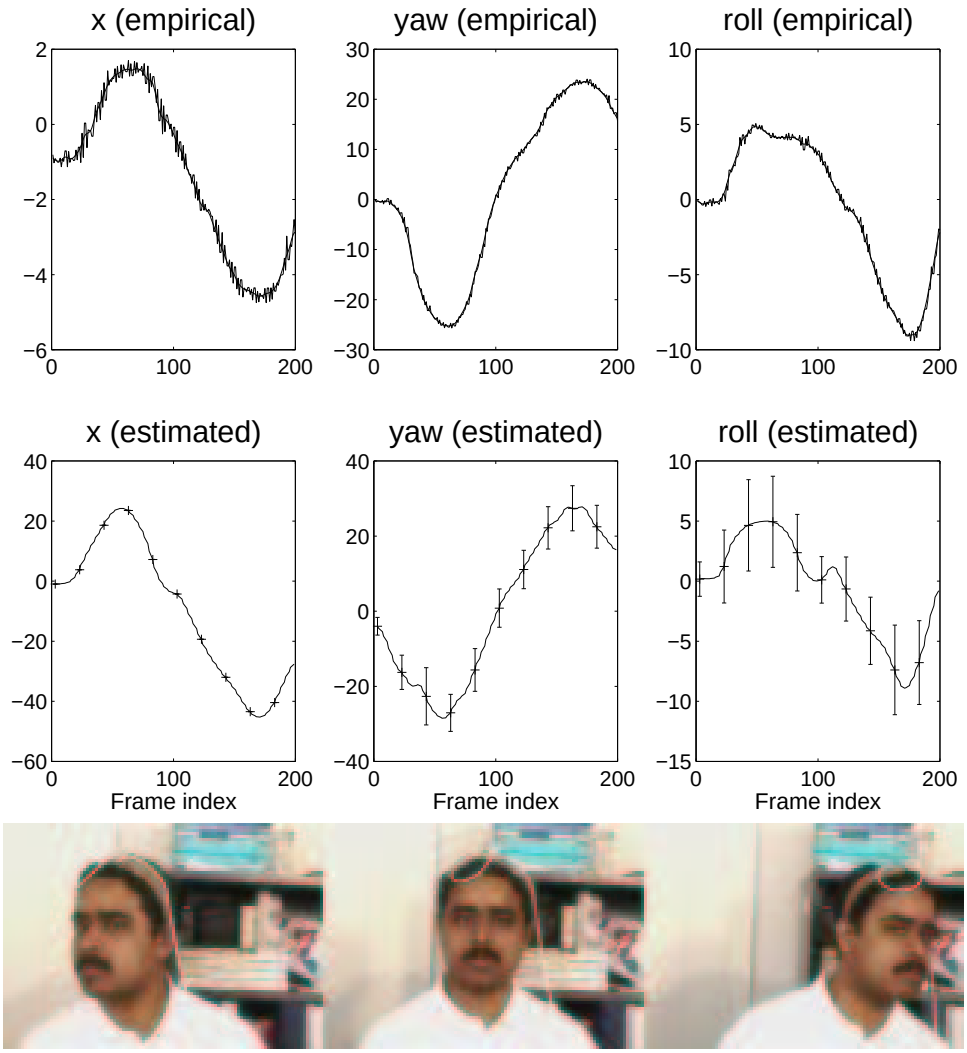


Figure 8-20: Results for a sequence containing a yaw movement and horizontal translation, with all other parameters remaining basically unchanged except for a slight roll. The top row shows ground truth. The second row shows the estimated pose parameters that change significantly during the sequence. 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.

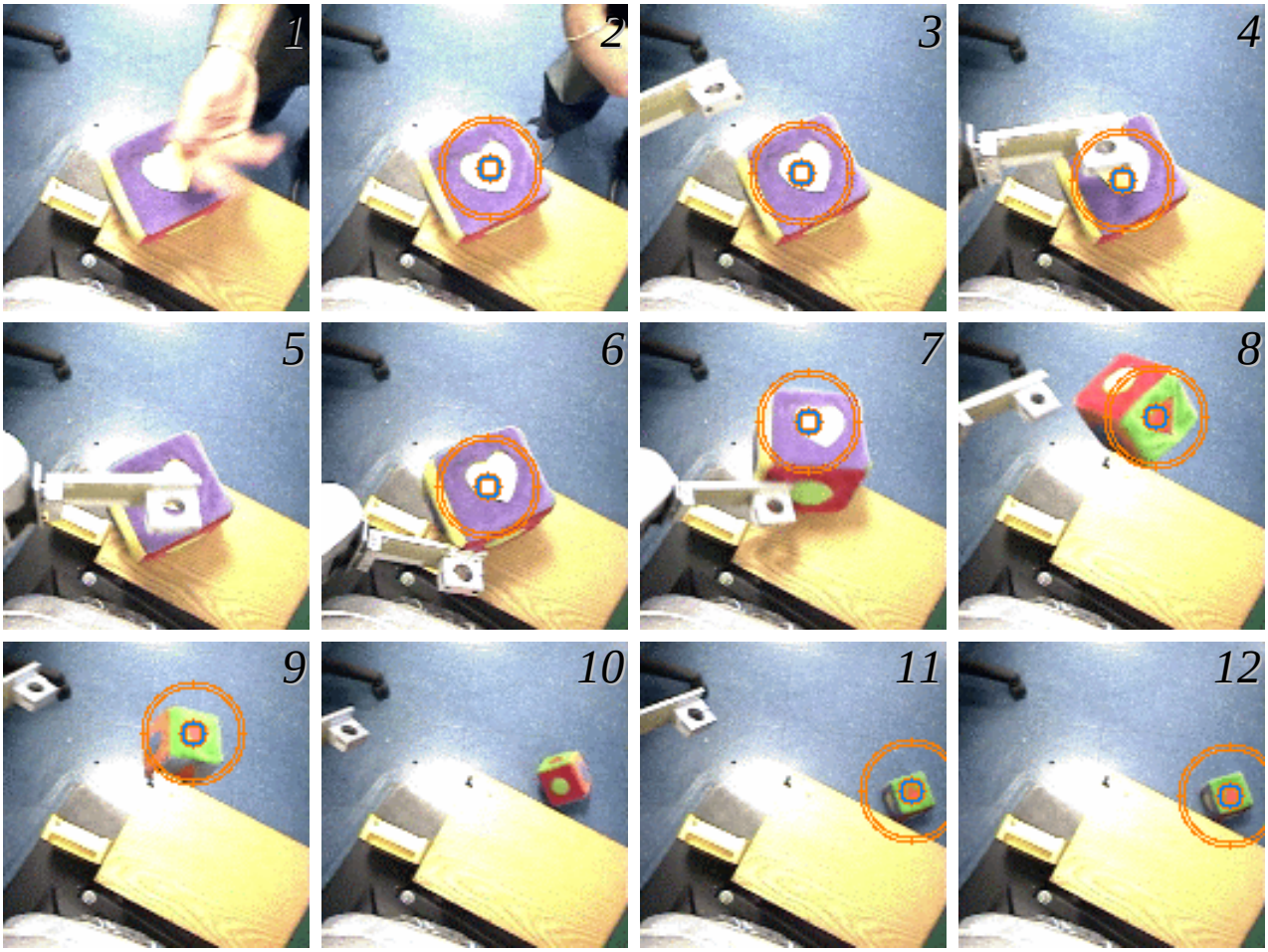


Figure 8-21: Tracking a poked object (a cube). The robot is familiar with the purple and green face of the cube. By tracking the cube using flat-track,the two faces are revealed to be views of the same object.