Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates

Paul Fieguth*
Dept. of Systems Design Engineering
University of Waterloo, Ontario, Canada
pfieguth@uwaterloo.ca

Demetri Terzopoulos
Dept. of Computer Science
University of Toronto, Ontario, Canada
dt@cs.toronto.edu

Abstract

We develop a simple and very fast method for object tracking based exclusively on color information in digitized video images. Running on a Silicon Graphics R4600 Indy system with an IndyCam, our algorithm is capable of simultaneously tracking objects at full frame size (640×480 *pixels) and video frame rate (30 fps). Robustness with respect to occlusion is achieved via an explicit hypothesis-tree model of the occlusion process. We demonstrate the efficacy of our technique in the challenging task of tracking people, especially tracking human heads and hands.*

1. Introduction

A variety of problems of current interest in computer vision require the ability to track moving objects [2], whether for purposes of surveillance [9], manufacturing, video compression [6], visually "aware" information kiosks [19], etc. The fundamental challenges that drive much of the research in this field are the enormous data bandwidths implied by high resolution frames at high frame rates, a desire for real-time, possibly interactive, performance, and a typically vaguely or ill-posed specification of the tracking problem itself. Numerous innovative methods have been proposed [4, 7, 12, 16, 17, 18, 20], however most of these are relatively sophisticated edge/snake/spline/template [3, 8] or eigenimage [10] based models; although these approaches are broad in their abilities (e.g., offering object recognition or pose estimation in addition to tracking), they are unable (yet) to run on full video resolution images at high frame rates.

The goal of the research described in this paper is to develop a tracking algorithm capable of tracking multiple objects in real time at full frame size and rate. We achieve this by relying heavily on color cues (as have others [14, 16, 20]) rather than tracking edges. Furthermore, we achieve some degree of robustness to occlusion by explicitly modeling the occlusion process, rather than relying implicitly on prior model fidelity (for which our simple prior model be inadequate).

Our tracking algorithm is intended to *complement* existing approaches. While our proposed algorithm is capable of simultaneously tracking multiple objects at video frame rates, we require the support of existing methods for two crucial steps:

- the detection and localization (e.g., [18]) of new objects to track;
- periodic reassessment of each object, to take into account possible color, shape, or scale changes.

For example, one might use our approach as a fast core program, with a more general (but slower) algorithm wrapped around this core providing periodic (e.g., every 10–60 frames) updates on object form and position.

Section 2 outlines a simple color-based model for tracking and demonstrates preliminary results. Section 3 addresses the question of robustness in the face of occlusion, and develops an explicit model for the occlusion process. Section 4 demonstrates our algorithm in tracking objects subject to occlusion. Section 5 draws conclusions about our approach.

2 Simple Color Tracking

Our simple tracking method is based on tracking regions of similar normalized color from frame to frame. Specifically, N regions $R_1, \ldots, R_N$ (taken to be rectangular for convenience) are defined within the extent of the object to be tracked; the size and relative position

*Supported by a postdoctoral fellowship of the Natural Sciences and Engineering Research Council of Canada.

1063-6919/97 $10.00 © 1997 IEEE

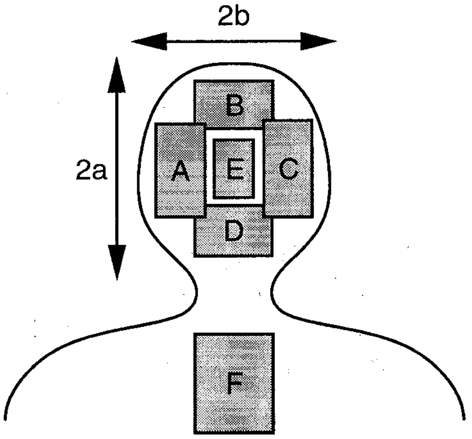


Figure 1. Example of six regions which might be used for head/torso tracking.

of these regions is assumed to be fixed. For example, in the context of head tracking, a possible assignment of five regions is shown in Figure 1; optionally a sixth region might be added to record clothing color below the head to provide some level of discrimination between individuals.

Each region R_i is characterized by a color vector

$$(r_i, g_i, b_i) = \sum_{(x,y)\in R_i} (r(x,y), g(x,y), b(x,y)) / |R_i| \quad (1)$$

which represents the averaged color of pixels within R_i, and where the notation $|R|$ measures the number of elements in set R (i.e., the number of pixels). If the camera possesses a favorable signal-to-noise ratio and the tracker is used in circumstances where the strongly colored features in each region R_i are relatively large, then the average in (1) will be relatively insensitive to subsampling and can be well approximated by computing the average over a regularly gridded subset of pixels as follows:

$$(r_i, g_i, b_i) = \sum_{(\hat{x},\hat{y})=(x_o+a\Delta x, y_o+b\Delta y)\in R_i} (r(\hat{x},\hat{y}), g(\hat{x},\hat{y}), b(\hat{x},\hat{y})) \frac{\Delta x \cdot \Delta y}{|R_i|} (2)$$

Typically we sample about 50–200 pixels per region; this subsampling yields a significant (order of magnitude) computational benefit.

For each region R_i, we assume the existence of an ideal or target color vector $(\bar{r}_i, \bar{g}_i, \bar{b}_i)$, which is computed at initialization. The details of assessing an accurate initial estimate of the position and shape of the object of interest may be relegated to any of several allegedly capable methods [10, 18, 20].

Goodness of Fit

The closeness in color space of a measurement vector (r_i, g_i, b_i) and its target $(\bar{r}_i, \bar{g}_i, \bar{b}_i)$ is assessed by the following goodness of fit criterion. To reduce sensitivity to shading or changes in illumination we want to eliminate multiplicative factors in intensity. Let

$$\gamma_{r_i} = \frac{r_i}{\bar{r}_i},\ \gamma_{g_i} = \frac{g_i}{\bar{g}_i},\ \gamma_{b_i} = \frac{b_i}{\bar{b}_i} \quad (3)$$

Ideally these ratios are equal; deviations from equality are assessed by the goodness of fit

$$\Psi_i = \frac{\max\{\ \gamma_{r_i}, \gamma_{g_i}, \gamma_{b_i}\ \}}{\min\{\ \gamma_{r_i}, \gamma_{g_i}, \gamma_{b_i}\ \}} \quad (4)$$

i.e., $\Psi = 1$ implies a perfect fit (modulo normalization), and Ψ increases as the fit becomes poorer. Clearly Ψ_i is a function of two parameters: $\Psi_i(x_o, y_o)$ is the goodness of fit of the hypothesis that R_i is centered on coordinate (x_o, y_o).

Two properties of Ψ are worth mentioning:

1. Ψ is insensitive to camera noise.

   For common CCD imaging systems, the low standard deviation of the pixel noise (on the order of 2–7 units), further reduced by a factor of 10 to 15 by the pixel averaging process (2), reduces the variability in Ψ due to pixel noise to a negligible value (about 0.02 in our setup).

2. Ψ is insensitive to scaling the object of interest, as opposed to the considerable sensitivity in edge-based approaches (which has motivated the development of more complicated deformable models [3, 4, 8] in order to retain robustness). In our tests (see below), if the regions R_i were chosen to be non-negligible in size (e.g., as suggested by the proportions in Figure 1), then changes in scale of $\approx \pm 50\%$ can be tracked without any special accommodation.

Tracking

Denote by (x_i, y_i) the center of region R_i relative to some natural origin of the object of interest. Then the hypothesis that (the origin of) the object is at location (x_H, y_H) is tested using

$$\Psi(x_H, y_H) = \sum_{i=1}^{N} \frac{\Psi_i(x_H + x_i, y_H + y_i)}{N} \quad (5)$$

Based on Ψ, the best estimate for the object's location is just

$$(\hat{x}, \hat{y}) = \arg \min_{(x_H, y_H)} \{\Psi(x_H, y_H)\} \quad (6)$$

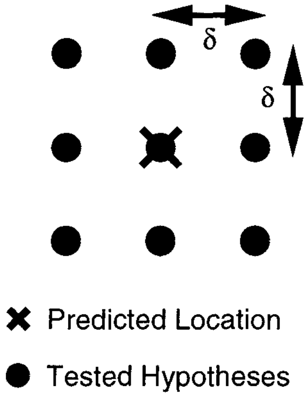


Figure 2. The nine lattice points which determine the local hypotheses to be tested for each object at each frame.

Equation (6) is essentially a measurement, and as such it can be employed in the context of a variety of established estimation/tracking algorithms: a Kalman filter [1, 2], Blake's condensation algorithm [7], a multiple-hypothesis tree approach [5, 12], a maximum likelihood approach [13] etc. We will be using a simple implementation of the latter two methods.

The optimization in (6) in principle implies a search over the entire plane (x, y). In practice, because our tracker is operating at video frame rate (30 frames/second), the prediction error in the object's position is likely to be small, so the search can be restricted to a local region. Specifically, at each frame f, we hypothesize M different positions of the object encircling the predicted location $(x(f|f-1), y(f|f-1))$ of the object at frame f based on observations up to frame $(f-1)$. The best hypothesis is then selected as the estimate for frame f (i.e., classical M-ary hypothesis testing [13]):

$$\begin{aligned}(x(f|f), y(f|f)) &= (x(f|f-1), y(f|f-1)) + \\ &\quad \delta \cdot (\rho_x(f), \rho_y(f)) \qquad (7)\\ (\rho_x(f), \rho_y(f)) &= \underset{(x,y)\in S}{\arg\ \min} \Big\{ \Psi \Big[x(f|f-1) + \\ &\quad \delta \cdot x, y(f|f-1) + \delta \cdot y \Big] \Big\} \qquad (8)\end{aligned}$$

where δ is a predefined step size, limiting the tracking resolution, and where S, $|S| = M$, is a set of lattice points in the plane including the origin. For example, the simplest choice of S is

$$S = \{\ (0,0)\ (0,1)\ (0,-1)\ (1,0)\ (-1,0)\ \} \qquad (9)$$

The frame-to-frame prediction is the usual linear one,

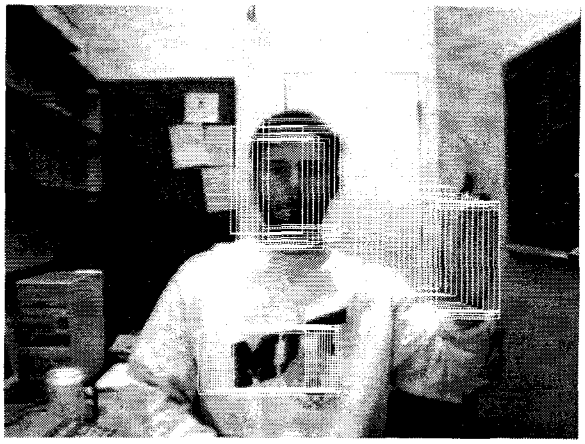


Figure 3. An example showing the simultaneous real-time tracking of three objects: a head, a hand, and the 'MIT' logo on the sweater. The effective motion can be inferred from the tracked positions in earlier frames indicated by the white rectangles.

$$x(f|f-1) = x(f-1|f-1) + \Delta t \cdot v_x(f-1) \qquad (10)$$

where $\Delta t = 1/30$ second is the frame separation time. The estimation of the velocity (v_x, v_y) is typically easily accomplished using a Kalman filter [3, 4]. In the context of our color-based tracker, the velocity estimation step is made difficult because the Kalman filter is, strictly speaking, inapplicable, because the noise in the "measurements" (6) or (7) is not Gaussian and its uncertainty is unknown (and variable). Using a Kalman filter to estimate (v_x, v_y) by assuming a variety of measurement noise uncertainties led to poor tracking results in all cases. Instead, we use a simple, nonlinear estimator which estimates the velocity based on trending:

$$\begin{array}{llrcl} & & v(f) & = & v(f-1) \\ \text{if} & (\rho(f) \cdot \rho(f-1) > 0) & v(f) & += & \delta \frac{\mathrm{sgn}(\rho(f))}{\Delta t} \\ \text{if} & (\rho(f) \cdot v(f-1) < 0) & v(f) & += & \delta \frac{\mathrm{sgn}(\rho(f))}{\Delta t} \\ \text{if} & (\rho(f) = 0) & v(f) & -= & \delta \frac{\mathrm{sgn}(v(f))}{2\Delta t} \end{array} \qquad (11)$$

where the last three equations respectively represent velocity correction on the basis of an accelerating trend, a decelerating trend, and a damping term to avoid oscillations.

Experiments

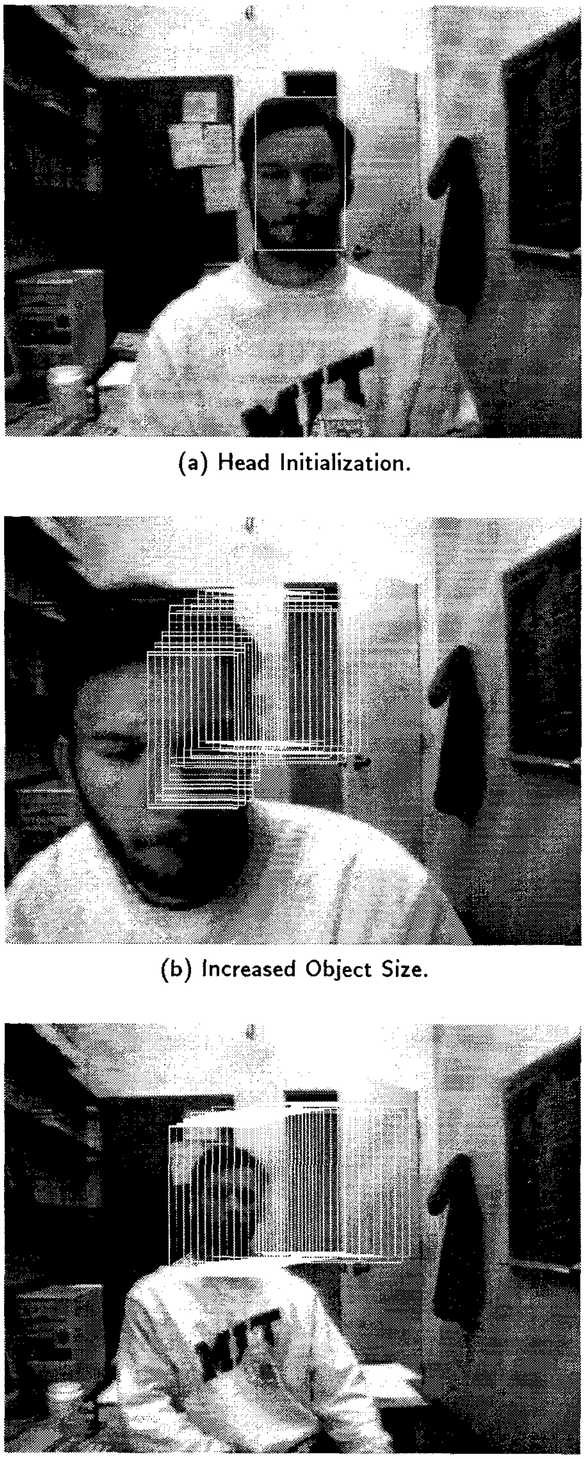
The tracker described above was implemented on a Silicon Graphics R4600 Indy system for testing. The tracker is based on the $N = 4$ regions (A,B,C,D) shown in Figure 1 and the $M = |S| = 9$–ary hypothesis cluster shown in Figure 2. The video source was the Indy camera operating at the full frame size of 640 × 480 RGB pixels. In this configuration, up to about ten objects can simultaneously be tracked and displayed at the full frame rate (30Hz) of the IndyCam camera.[1] For each of the following examples, the initial location of the object and its dimensions (a, b in Figure 1) are assumed known (e.g., provided by some independent algorithm); in this case, the initialization was performed manually.

Figure 3 demonstrates the results of tracking three objects: a head, hand, and the colored logo on a shirt. The sequence of rectangular frames in the figure indicates the tracked motion. The tracker is robust in the sense that normal head and hand motions are tracked without failure. Although it is possible to cause the tracker to fail with sufficiently vigorous motion, the robustness can be increased by increasing M (values of $M \approx 60$ were tested on a more powerful DIGITAL Alpha workstation with remarkably robust results).

Figure 4 illustrates the insensitivity of the proposed tracking method to changes in scale. The tracker is initialized to a head as in Figure 4(a). In Figure 4(b) the head (or some portion of it) has been tracked over about 40 frames while approaching the camera, growing in linear dimension by about 75%. Similarly, in Figure 4(c) the head moves away from the camera, to the point where it is now smaller than its initialized counterpart by 50% in linear dimension. In each case, the size of the head *model*, implied by the plotted rectangles in the figures, remains constant. To be fair, it should be pointed out that an insensitivity to scale *does* imply an inability on the part of the tracker to achieve an accurate (e.g., pixel level) target lock, which may be a liability in certain circumstances; using the tracker in the context of a more robust shell, as envisaged in the Introduction, would mitigate this liability somewhat.

3. Modeling Occlusion

Many robust tracking methods have been developed; however, the question of occlusion is addressed infrequently [11, 17], and then usually implicitly rather than by virtue of an explicit model. Typically, some measure of occlusion robustness is realized via a sufficiently detailed model of the object being tracked, with a limited rate of deformability. With such a detailed model, it is possible to discriminate between good and spurious measurements, and therefore to ignore measurements taken while the object of interest is occluded.



(a) Head Initialization.

(b) Increased Object Size.

(c) Diminished Object Size.

Figure 4. An example illustrating the insensitivity of color-region-based tracking to moderate changes in scale of the object being tracked: the regions are initialized to "truth" in (a). The head is then tracked reliably despite increases (b) and decreases (c) in size.

[1] The computational complexity of the tracker scales in general by a factor proportional to $MN/(\Delta x \cdot \Delta y)$.

Our simple color-based model, developed in the previous section, offers several motivations for looking at the question of occlusion more explicitly:

1. The tracker proposed in the previous section is *not* occlusion-robust: The prior model (i.e., N color vectors) is not highly tuned or specific and, more significantly, an estimate (6) is computed in every frame, regardless of how well the image fits the prior color model. Occlusion forces (6) to track a different (incorrect) local minimum of Ψ.

2. We are interested in fast (real-time) tracking. Although maintaining a very large set of hypotheses (such as in [7]) could address the above limitations, it is unlikely that this could be accomplished within the desired computational limits.

3. The simplicity of our model provides an excellent opportunity for explicitly modeling the state of occlusion with relatively few hypotheses. If each region R_i is assumed to be either visible or occluded, then there are 2^N possible occlusion states (quite tractable for the modest N considered in this paper). If we also assume that the occluding object is comparable or larger in size than the object being tracked, then the number of hypotheses can be reduced much further. This forms the basis for the occlusion modeling described below.

The $N = 5$ regions (A,B,C,D,E) of Figure 1 will be used to model the head for tracking purposes. For occluding objects larger than the tracked object, the occluding hypotheses can reasonably be limited to those shown in Figure 5. We limit ourselves to the $H = 10$ hypotheses drawn in bold; a variety of logical extensions are possible, such as the four additional dashed hypotheses. The arrows in the figure identify the permitted successive hypothesis states.

Each occlusion class is represented by a set $\mathcal{O}$ which is a set of integer indices corresponding to visible regions; that is,

$$i \notin \mathcal{O} \Rightarrow R_i \text{ is occluded} \qquad (12)$$

Each hypothesis, then, is identified in terms of its state $(x, y, v_x, v_y, \mathcal{O})$. We denote by $\mathcal{N}(\mathcal{O})$ the set of occluding states adjacent to $\mathcal{O}$ as implied by Figure 5. In considering the transition of a hypothesis from one class $\mathcal{O}_1$ to another $\mathcal{O}_2$, the quantity of primary interest is the state of the disputed region $\mathcal{O}_1 \oplus \mathcal{O}_2$ (where $\oplus$ is the set exclusive-or operator). With reference to (5), we now define

$$\Psi_{\mathcal{O}}(x_H, y_H) = \sum_{i\in\mathcal{O}} \frac{\Psi_i(x_H + x_i, y_H + y_i)}{N} \qquad (13)$$

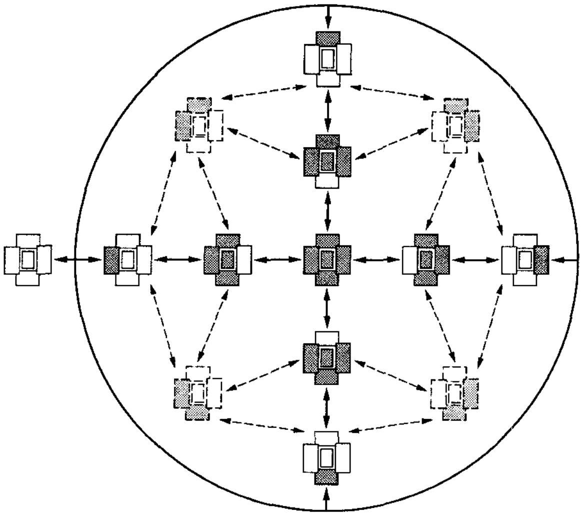


Figure 5. A set of occlusion classes and the permitted transitions of a hypothesis between these classes. The ten classes in bold represent those used in the example of this paper; the dashed classes are a possible reasonable extension.

Then it is $\Psi_{\mathcal{O}_1\oplus\mathcal{O}_2}$ which is of interest, and which determines the relative probabilities of hypothesis creation and destruction. Specifically, for each hypothesis $(x, y, v_x, v_y, \mathcal{O})$ and for each $\bar{\mathcal{O}} \in \mathcal{N}(\mathcal{O})$, we solve (7)–(11), but with Ψ replaced by $\Psi_{\bar{\mathcal{O}}}$. Then, with the estimated position $x(f|f), y(f|f)$ from (7), we calculate the discriminant

$$\phi = \Psi_{\mathcal{O}\oplus\bar{\mathcal{O}}}(x(f|f), y(f|f)) \qquad (14)$$

Low values of ϕ (near unity) imply that the contested regions $\mathcal{O} \oplus \bar{\mathcal{O}}$ are not occluded; larger values suggest increasing probabilities of occlusion. Consequently if $\bar{\mathcal{O}} \subset \mathcal{O}$,

$$\Pr(\text{Create Hypoth}(x(f|f), y(f|f), v_x(f), v_y(f), \bar{\mathcal{O}})) = \begin{cases} 0 & \phi < \phi_1 \\ \frac{\phi-\phi_1}{\phi_2-\phi_1} & \phi_1 \le \phi \le \phi_2 \\ 1 & \phi_2 < \phi \end{cases} \qquad (15)$$

$$\Pr(\text{Delete Hypoth}(x, y, v_x, v_y, \mathcal{O})) = \begin{cases} 0 & \phi < \phi_2 \\ 1 & \phi_2 \le \phi \end{cases} \qquad (16)$$

for appropriate thresholds ϕ_1, ϕ_2. The probabilities are reversed for the opposite case in which $\mathcal{O} \subset \bar{\mathcal{O}}$. Reasonable threshold values were selected empirically as $\phi_1 = 1.1, \phi_2 = 1.3$.

Having repeated the above procedure for each hypothesis, a pruning step is required to prevent exponential growth of the tree. Specifically, we wish to limit the number of active hypotheses to $H \leq P$. Choosing the P most likely hypotheses is complicated by the difficulty in comparing the relative likelihoods of hypotheses belonging to different occlusion classes. Consequently, we limit ourselves to pruning only *within* occlusion classes, for which comparing relative likelihoods $\Psi_{\mathcal{O}}$ is straightforward (with the exception of the totally occluded class, in which case the more recently fully-occluded hypotheses are considered to be more likely).

Of course, there are some limitations to this scheme. The simplicity of the prior model in our particular head tracking application leads to ambiguity as to whether the head is being occluded from the left or from the right (similarly for top and bottom). However, to the extent that our goal is robust tracking, and not so much an understanding of the occlusive behavior, this is not a significant drawback.

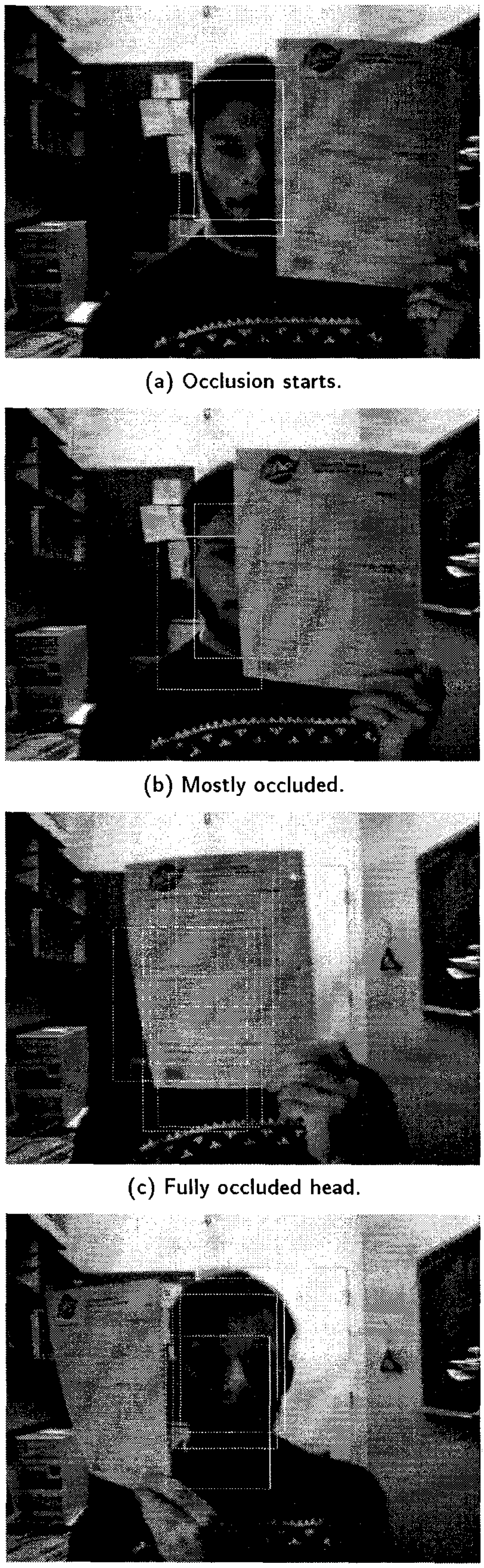
4. Results

We will limit ourselves here to a single demonstration of tracking in the presence of occlusion, for an example in which the tracker of Section 2 alone would fail. Because of the large number of hypotheses of varying likelihood in different occlusion classes which may be present in each frame, to simplify the presentation we will limit ourselves to showing only the least-occluded hypothesis subset.

As before, we start by initializing our head model as in Figure 4(a), except now based on the 5 regions (A,B,C,D,E) depicted in Figure 1. The underlying tracker is that described in Section 2, with the five point search space S given in (9). The upper bound on the number of hypotheses kept after the pruning step is set to $P = 12$; with this setting real-time tracking can still be achieved.

Figure 6 contains a sequence of four images which show the least-occluded hypotheses as the tracked head is occluded. In each case, the line-style of the plotted rectangle determines the occlusion state of the associated hypothesis; that is, □, □, ⬚, ⬚ respectively imply fully visible, right 30% occluded, right 70% occluded, and fully occluded states.

Figure 6(a) shows the occlusion process just begin-



(a) Occlusion starts.

(b) Mostly occluded.

(c) Fully occluded head.

(d) Head re-exposed after occlusion.

Figure 6. Four images illustrating the progression of occluding a tracked head from right to left. In each image, the rectangles show the position and occlusion type of the least occluded hypotheses.

ning; the tracker correctly has no hypotheses in the fully-visible state, and the two hypotheses shown represent left and right occlusion (consistent with the left–right ambiguity discussed in the previous section). As the occlusion progresses, in Figure 6(b), the least occluded hypotheses are those in which only one of the five modeled regions is visible. When the actual occlusion is total, Figure 6(c), the tracker correctly does not have a single hypothesis less than fully occluded, and seven fully occluded hypotheses are shown. As the tracked head is re-exposed, Figure 6(d), several of the hypotheses successfully reacquire the tracked object.

5. Conclusion

We have illustrated the development and operation of a real-time feature tracker based entirely on image color information that possesses an explicit occlusion model. Future work will include examining questions of color-based tracking in contexts where the tracked object and the background may have similar normalized colors, tracking in contexts where multiple similar objects (e.g., several people) are simultaneously present in the scene in possibly close proximity, the tracking of articulated objects (e.g., fingers) whose members allow several degrees of freedom of motion with respect to one another (as opposed to the rigidity assumption made of the colored regions in the model of this paper), and looking at the introduction of edge-based cues to allow finer tracking and possibly to resolve the left/right top/bottom occlusion ambiguity.

References

[1] B. Anderson, J. Moore, *Optimal Filtering*, Prentice-Hall, New Jersey, 1979

[2] Y. Bar Shalom, T. Fortmann, *Tracking and Data Association,* Academic Press, New York, 1988

[3] A. Blake, A. Yuille (ed.s), *Active Vision*, MIT Press, 1992

[4] A. Blake, R. Curwen, A. Zisserman, "A Framework for Spatiotemporal Control in the Tracking of Visual Contours," *Int. J. Computer Vision* (11) #2, pp.127–145, 1993

[5] S. Blostein, T. Huang, "Detecting Small, Moving Objects in Image Sequences Using Sequential Hypothesis Testing," *IEEE Signal Processing* (39) #7, pp.1611-29, 1991

[6] A. Eleftheriadis, A. Jacquin, "Automatic Face Location Detection and Tracking for Model-Assisted Coding of Video Teleconference Sequences at Low Bit Rates," *Signal Processing – Image Communication* (7) #3, pp.231–248, 1995

[7] M. Isard, A. Blake, "Contour tracking by stochastic propagation of conditional density," *Proc. 4th European Conf. on Computer Vision*, Cambridge, UK, April 1996

[8] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active Contour Models," *Int. J. of Computer Vision* (1) #4, pp.321–331, 1988

[9] A. Nagai, Y. Kuno, Y. Suirai, "Surveillance System Based on Spatio-Temporal Information," *ICIP'96* (II) pp.593–6, 1996

[10] A. Pentland, B. Moghaddam, T. Starner, "View-Based and Modular Eigenspaces for Face Recognition," *CVPR'94*, pp.84–91, 1994

[11] J. Rehg, T. Kanade, "Model-based tracking of self-occluding articulated objects," *Proc. Fifth ICCV*, pp. 612-617, Boston, MA, 1995

[12] D. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Auto. Control* (24) #6, pp.843–854, 1979

[13] K. Shanmugan, A. Breipohl, *Random Signals*, John Wiley & Sons, New York, 1988

[14] M. Swain, D. Ballard, "Color indexing," *Int.J. Comput. Vision* (7), pp.11-32, 1991

[15] D. Terzopoulos, R. Szeliski, "Tracking with Kalman Snakes," In [3], pp.3–20, 1992

[16] D. Terzopoulos, T. Rabie, "Animat vision: Active vision with artificial animals," *Proc. Fifth ICCV*, Cambridge, MA, pp.801–808, 1995

[17] C. Toklu, A. Tekalp, A. Erden, M. Sezan, "2-D Mesh-Based Tracking of Deformable Objects with Occlusion," *ICIP'96* (I) pp.933–6, 1996

[18] C. Vieren, F. Cabestaing, J. Postaire, "Catching Moving Objects with Snakes for Motion Tracking," *Pattern Recognition Letters* (16) #7, pp.679–685, 1995

[19] K. Waters, J. Rehg, M. Loughlin, S. Kang, D. Terzopoulos, "Visual Human Sensing for Active Public Interfaces," in *Computer Vision in Man-Machine Interfaces* (S. Pentland & R. Cipolla ed.s), Cambridge University Press, 1996

[20] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," SPIE Vol. 2615, 1995