Bayesian Mean Shift Face Tracker

Robin Hewitt

March 2006

Algorithm Summary

Mean-Shift tracking plus Birchfield's constant-velocity-based location predictor [1].

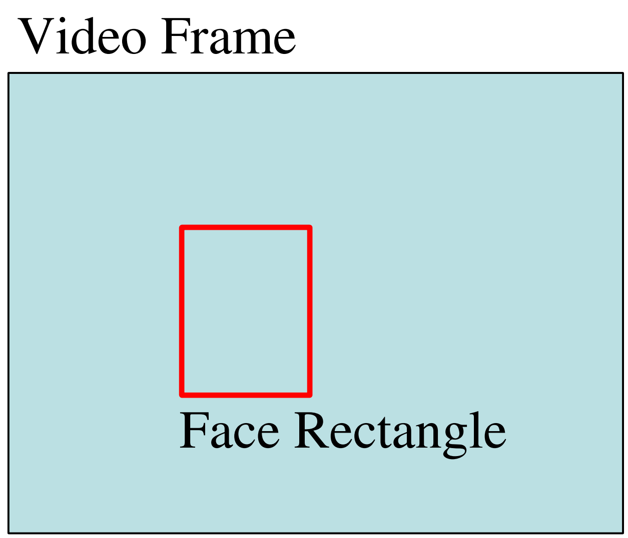
As in Bradsky's Camshift method [3], the tracker uses face probability values based on a color histogram.

Bradsky uses a heuristic (the ratio histogram described in [5]) to compute face probability. Instead, I computed probability in a more principled way. Doing that improved robustness against drift.

Algorithm Details

Inputs:

- The start frame
- Face-location rectangle



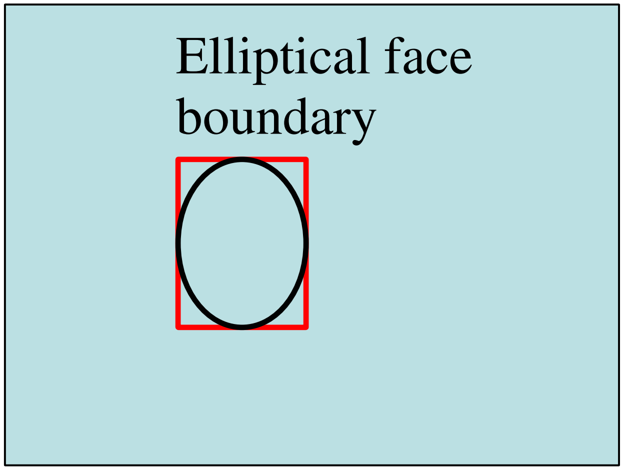
Algorithm Details

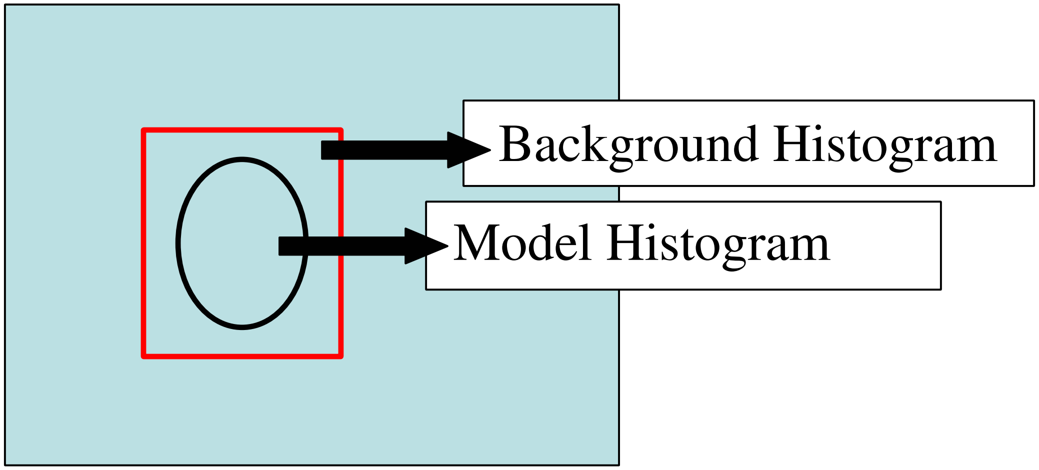
Initialization:

1. Compute an elliptical boundary based on input rectangle

2. Compute the model histogram, *M*, for the ellipse's interior

3. Compute a background histogram, *B*, for a region between the ellipse and a rectangle that's slightly larger than the ellipse





Algorithm Details

For each subsequent video frame:

1. Predict face location assuming constant velocity

2. Compute histogram for a search region centered on the predicted location

3. Create the lookup table for face-pixel probability

4. Assign probability values to each pixel in search region

5. Locate nearest mode in the probability density using Mean Shift

6. Update the background histogram (to use for the next frame)

Algorithm Details

Main Parameters:

- Color model (used hue x saturation)
- Number of histogram bins per dimension (used 15)
- Min and max intensity cutoffs for calculating hue

Computing a Pixel's Face Probability

Start with Bayes Rule: $P(f \mid i) = \frac{P(i \mid f)P(f)}{P(i)}$,

where

i is the histogram bin for a pixel, and

f represents face class.

Computing a Pixel's Face Probability

$$P(f \mid i) = \frac{P(i \mid f)P(f)}{P(i)} \quad \text{(Bayes Rule)}$$

Use $P(i \mid f) = M_i$, $P(f) = \frac{A_{\text{int}}}{A_{\text{int}} + A_{ext}}$, and $P(i) = \frac{I_i A_{\text{int}} + B_i A_{ext}}{A_{\text{int}} + A_{ext}}$,

where,

M_i is the model histogram,

I_i is an image-region histogram,

B_i is the background histogram from the previous frame,

A_{int} and A_{ext} are interior and exterior areas in pixels.

Computing a Pixel's Face Probability

Repeating: $P(f \mid i) = \dfrac{P(i \mid f)P(f)}{P(i)}$ (Bayes Rule)

$$P(i \mid f) = M_i, \quad P(f) = \frac{A_{\text{int}}}{A_{\text{int}} + A_{ext}}, \quad P(i) = \frac{I_i A_{\text{int}} + B_i A_{ext}}{A_{\text{int}} + A_{ext}}$$

Then,

$$P(f \mid i) = \frac{M_i A_{\text{int}}(A_{\text{int}} + A_{ext})}{(A_{\text{int}} + A_{ext})(I_i A_{\text{int}} + B_i A_{ext})} = \frac{M_i A_{\text{int}}}{(I_i A_{\text{int}} + B_i A_{ext})}.$$

Computing a Pixel's Face Probability

$$P(f \mid i) = \frac{M_i A_{\text{int}}}{(I_i A_{\text{int}} + B_i A_{ext})} \qquad \text{(Face probability)}$$

To find a pixel's face probability,

1. Determine its histogram bin, i
2. Look up $P(f \mid i)$ for bin i

Computational Overhead

Without the constant-velocity prediction, Mean Shift typically takes 5-6 iterations to converge.

With this prediction, Mean Shift typically converges in 1-2 iterations.

Most of the computation is in updating histograms, so Using the background histogram is roughly equivalent to one additional Mean Shift iteration.

Net result: Bayesian tracker is faster than Camshift.

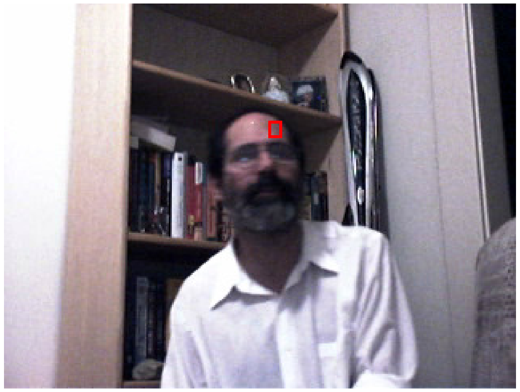
Testing

Overall accuracy comparisons

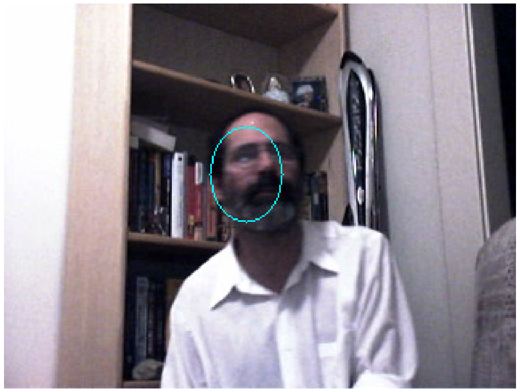
Camshift | Birchfield | Bayesian

Testing

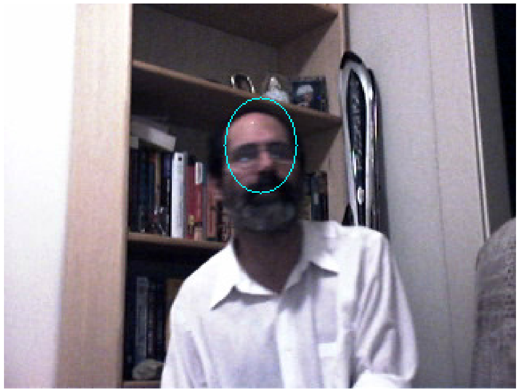
Camshift



Birchfield



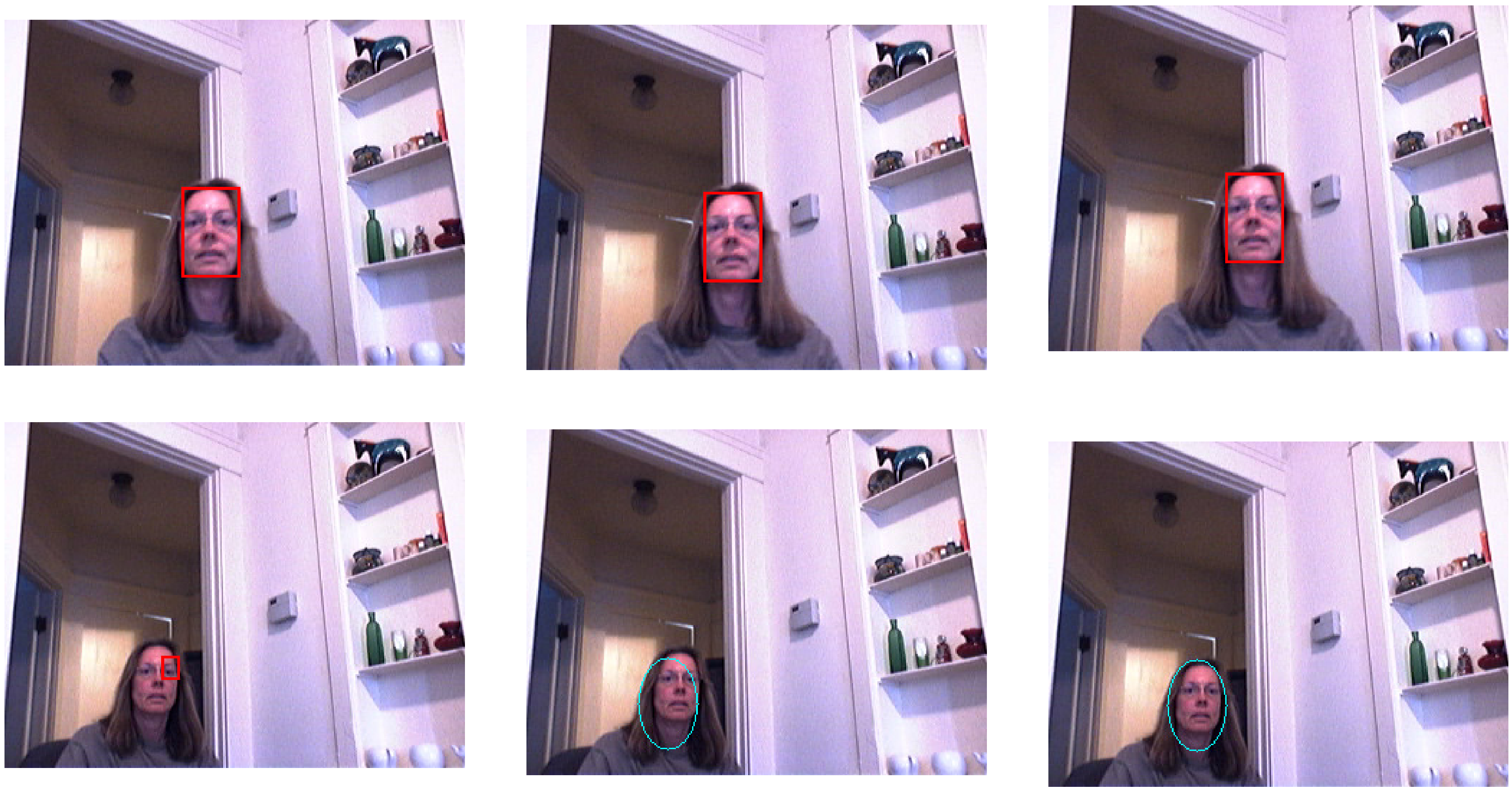
Bayesian



Bayesian tracker's worst localization in this sequence was in frame 21.

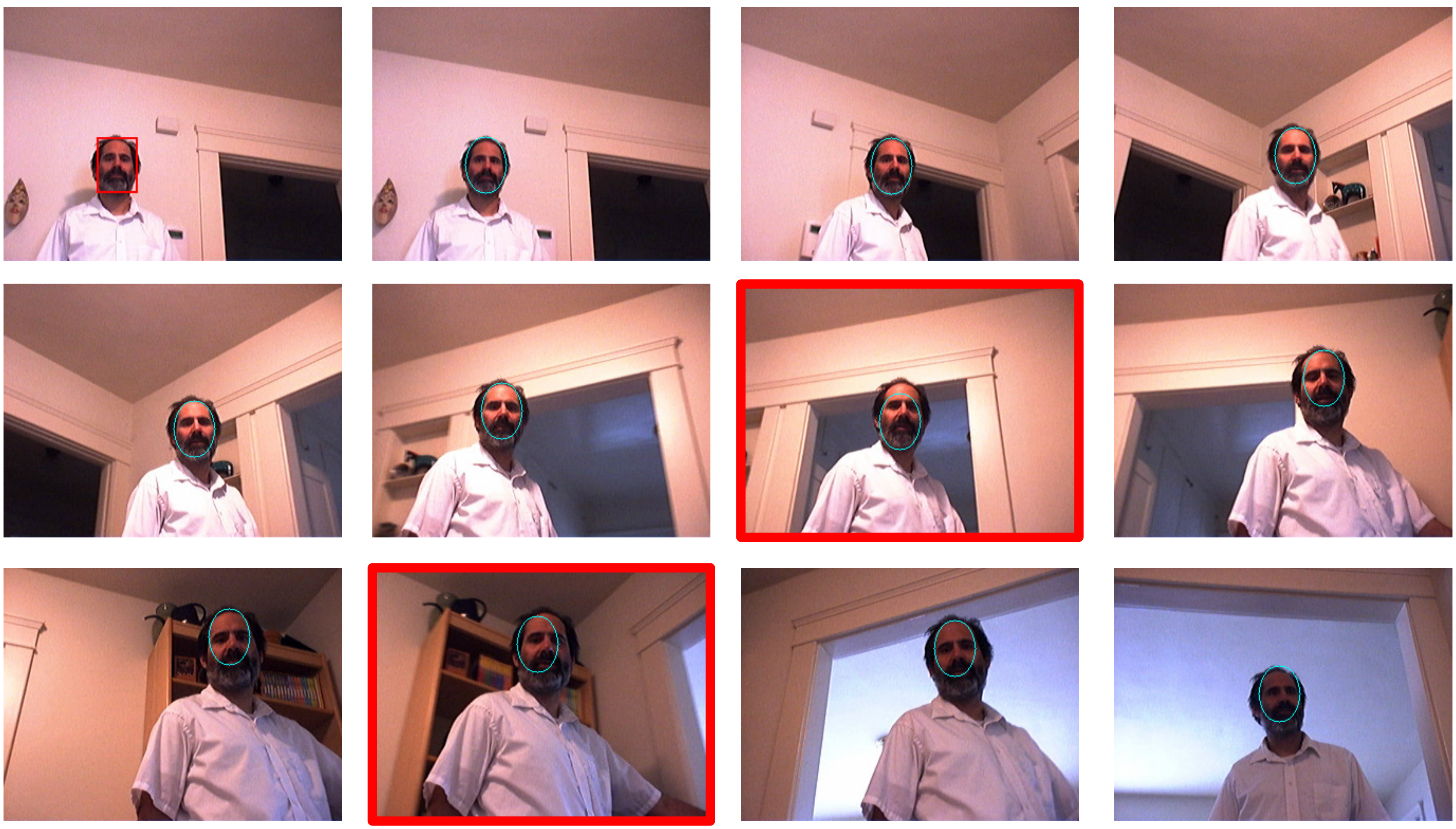
Testing

Camshift Birchfield Bayesian



Test Results

Stress test: a moving camera, moving subject, and lighting changes. Localization slips then recovers in two places (red).



Conclusions

Bayesian Tracking method is fast.

BT was more robust than either Camshift or Birchfield's elliptical head tracker during testing.

Tracking errors consist of small drifts that correct themselves.

References

1. S. Birchfield, “Elliptical Head Tracking Using Intensity Gradients and Color Histograms,” *CVPR,* 1998.
2. A. Blum and T.Mitchell, “Combining Labeled and Unlabeled Data with Co-training,” *Proceedings of the Workshop on Computational Learning Theory*, Morgan Kaufmann, 1998.
3. G.R. Bradski, “Computer video face tracking for use in a perceptual user interface,” *Intel Technology Journal*, Q2, 1998.
4. D. Comaniciu, V. Ramesh, and P. Meer, “Real-Time Tracking of Non-Rigid Objects using Mean Shift,” *CVPR2000,* Vol. 2, pp. 142-149, 2000.
5. M.J. Swain and D.H. Ballard, “Color Indexing,” *International Journal of Computer Vision*, Vol. 7:1, 1991.