Boston College
Computer Science Department

Senior Thesis 2002

Christopher Fagiani

An Evaluation of Tracking Methods for Human-Computer Interaction

Prof. James Gips

Evaluation of Tracking Methods for Human-Computer Interaction

Camera Mouse [1,2,21], a system developed at Boston College that uses video input to control the mouse cursor. Upon start-up, a feature is selected for tracking by clicking on it in the vision window. The system then tracks that feature in real time, moving the mouse cursor accordingly.

The purpose of introducing the Kalman Filter [19] is to reduce the amount of processor time needed by the tracking application, thereby allowing for third-party applications to have a greater share of CPU time. The Kalman Filter was applied in two distinct ways: estimating the feature location in every frame and in every other frame. Using the filter every frame allowed for a smaller search area for the tracking algorithm whereas, when using the Kalman filter every other frame, the Kalman estimate was used instead of running the tracking algorithm on that frame. Both methods significantly reduce the CPU time needed by the system, but the scheme in which the Kalman filter is used in every frame is more efficient in terms of CPU load. It is anticipated that tracking systems that are less taxing upon CPUs will eventually enable vision-based tracking systems to be installed in a wide variety of technology, from intelligent vehicles, to almost any other application where head tracking and feature analysis is required.

This paper describes the algorithms and computer vision techniques used to refine a real time feature tracking system in terms of demand on the CPU in such a manner that does not significantly impact tracking accuracy.

Among the methods presented, one employs visual information about the motion of a feature in conjunction with the mathematics of the Kalman Filter to alternately measure and estimate the location of the feature. This method did reduce CPU load, but the reduction in tracking accuracy is prohibitive to using this approach.

The normalized correlation coefficient tracking without any Kalman Filters proved to be the most accurate algorithm. This method, in addition to being fairly accurate, was slightly more demanding on the CPU than the next-best method, the Lucas-Kanade (LK) tracker [18].

Utilizing the normalized correlation coefficient tracker on a smaller search area while using the Kalman filter to estimate feature position every frame proved to be the least CPU intensive method tested. While this was not as accurate as the normalized correlation method without Kalman filters, the processing time was considerably less.

Though stil

2.1 Normalized Correlation Tracking

The normalized correlation coefficient tracking method computes the correlation coefficient for each point a search area that is centered around the position of the feature in the previous frame [See Equation 2.1.1].

(Eq 2.1.1)

$$R(x,y) = \frac{n \sum_{i=0}^{n} x_i y_i - \sum_{i=0}^{n} x_i \sum_{i=0}^{n} y_i}{\sqrt{\left(n \sum_{i=0}^{n} x_i^2 - \left(\right.\right.}}$$

If u is defined as dx/dt and v is defined as dy/dt, then one may derive the optical flow constraint equation:

(Eq 2.2.3)

$$- \#I/\#t = \#I/\#x\ u + \#I/\#y\ v$$

Feature position estimate is carried out by

(Eq. 2.3.5)

$$X_k = \bar{X}_k + K_k(z_k - H_k \bar{X}_k)$$

Where each term represents the same matrix as in above equations.

(Eq. 2.3.6)

$$P_k = (I - K_k H_k) \bar{P}_k$$

Where each term represents the same matrix as in above equations and I is the identity operator.

The movement of the feature within the image window is translated to screen coordinates through a function that takes into account the screen resolution, the size of the image being analyzed and the apparent motion of the feature (See Equations 2.4.1 and 2.4.2). Before translating image plane coordinates to screen coordinates, the image window is first reduced so that only the interior three quarters of the image is considered (i.e. the edge of the image window does not correspond to the edge of the screen).

(Eq 2.4.1)

$$c_x = \begin{cases} \dfrac{((0.5\, d_x - p_{x,i})m_x)}{0.5\, d_x} + 0.5\, m_x \text{ if } d_x/4 < p_{x,I} < 3d_x/4 \\ 0 \text{ if } p_{x,I} < d_x/4 \\ 1 \text{ if } p_{x,I} > 3d_x/4 \end{cases}$$

Where d_x is the width of the image (in pixels), m_x is the width of the monitor (in pixels), and $p_{x,i}$ is the x coordinate of the predicated feature location in frame i. The image is mirrored in the horizontal direction to make it easier for users to coordinate head movement with screen movement, therefore $0.5d_x - p_{x,I}$ in the above equation is negative when the feature

movements and in the other two, the subject used the system to control the Midas Touch spelling application.

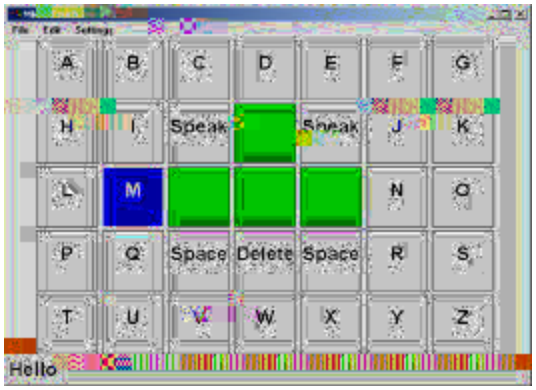


Figure 4.1 The Midas Touch spelling application. Users spell words by moving the mouse cursor over a character and then dwelling within a small radius of pixels for a second to generate a mouse click. Subjects were asked to spell the word "hello" during trials.

When using the Midas Touch spelling program, the subjects were instructed to spell the word "Hello." The subjects did not make any movements that were not associated with performing this task (i.e. subjects did not look away from the screen during input capture), except in the case of the sequence with random movement where the subject was asked to move his head in any direction he wished. The subjects were positioned approximately two feet from the camera and they remained at this distance for the duration of the input sequences.

Evaluation of algorithms was performed in terms of accuracy of the tracking. Accuracy was measured by comparing $P_{d,i}$, a vector

4.2 Qualitative Experiments

Input data of a much longer duration than the hand-classified sequences was recorded on videotape. Video was recorded while the subject used the system to control a number of applications including Midas Touch, Eagle Aliens, Speech Staggered, and Eagle Paint [See Figure 4.2]. Nine minutes of video input was captured in this manner.

Figure 4.2 The applications used during video capture. From top to bottom: Eagle Aliens, Eagle Paint, and Speech

Figure 5.1 In this image sequence, the LK tracker was used without Kalman Filters. The region containing the tracked feature is blown up in each image to make it easier to see. The lighter square denotes the ideal feature location whereas the darker square denotes the location returned by the tracking algorithm. The area tracked in this sequence is the left side of the bridge of the glasses.

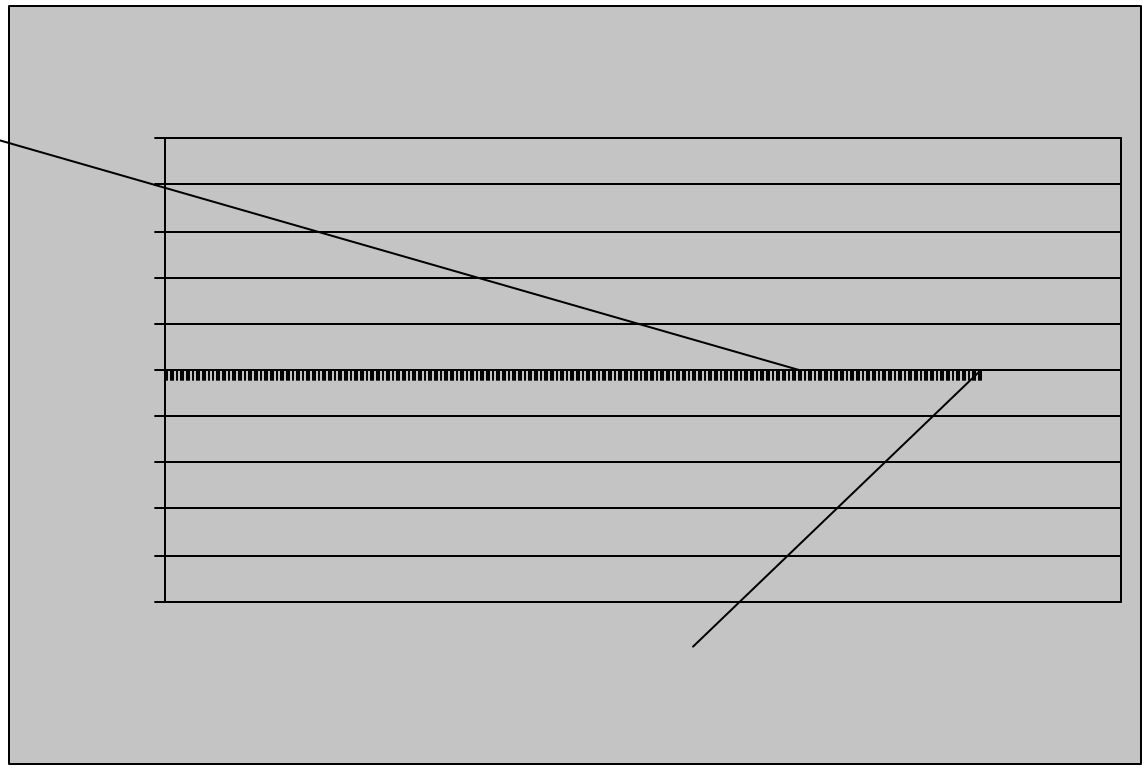
Features tracked were deemed suitable for tracking if they were easily differentiable from their immediate surroundings. This quality may be observed by examining the correlation coefficients observed when a template is compared with itself [See Figure 5.2]. Even

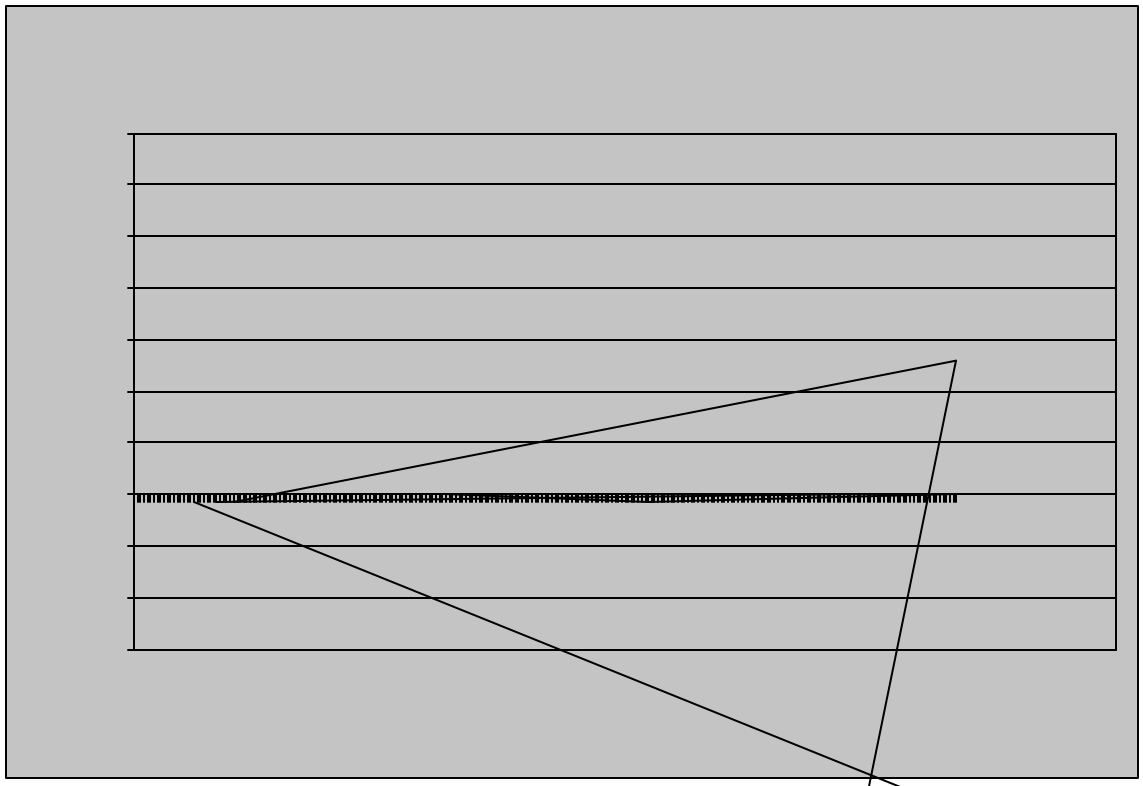
In terms of processing time, the normalized correlation coefficient tracker with no Kalman Filters was the most demanding. The least demanding was the normalized correlation coefficient with a 4 dimensional Kalman Filter [See Figure 5.4]. Despite the large time required for the normalized correlation tracker, the system was still able to operate at approximately 30

without Kalman Filtering performed similarly on some input, the version without Kalman Filtering worked better on erratic movement. Mean feature velocity in sequence with random movement was 110.2 pixels per frame and the mean velocity in the bottom sequence was 55.8 pixels per frame. Each entry in the graph represents the same tracking method as in Figure 5.3.

On the image series in which the subject was using the Camera Mouse to manipulate the mouse cursor in a third-party application, the normalized correlation coefficient tracker performed with the highest level of accuracy [See Figures 5.7 and 5.8]. Though some drift was observed, the motion recorded was

have occurred due to the possibility that lighting changes may occur more rapidly with vertical movement than horizontal. Also, because of the





exhibited slightly more pronounced drift, as the feature drifted from one eyebrow to the other and then to the eyelid. This corresponds to approximately 10 millimeters. The drift was observed while the subject was using Eagle Aliens. This behavior is to be expected since the Eagle Aliens application is characterized by much more rapid movements than the spelling applications. Since the subject may be changing direction of movement suddenly, it is not surprising that the Kalman Filter yields poor estimates of feature location.

When the LK tracker was used in conjunction with Kalman Filtering, the tracker lost the feature multiple times. In the 2-D and 4-D cases, the feature was lost twice per sequence tested and, in the 6-D case, the feature was lost an average of nine times on each sequence tested. In each of the cases where the feature was lost, the

References

[1] J. Gips, M. Betke, and P. Fleming. The Camera Mouse: Preliminary investigation of automated visual tracking for computer access. In
, Orlando, FL, July 2000.

[2] The Camera Mouse Project at Boston College. http://www.cs.bc.edu/~gips/CM

[3] M. Otte and H.H. Nagel. Optical Flow Estimation: Advances and Comparisons.

[21] M. Betke, J. Gips, and P. Fleming. The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access For People with Severe Disablilities.

.

In press, April 2002.