Multi-Camera person tracking using an extended Kalman Filter

B.Merven, F.Nicolls and G.de Jager

Department of Electrical Engineering, University of Cape Town,
Private Bag, Rondebosch, 7701, South Africa,
Email: {bruno,nicolls,gdj}@dip.ee.uct.ac.za

Abstract

This paper presents some 'work in progress' towards a multi-camera person tracking solution. The tracking system aims to combine observations obtained from one or more cameras and a simple motion model to optimally estimate the location of a person in the monitored scene using an Extended Kalman Filter. Observations are made in image space but the tracking takes place in world coordinates using the cameras' calibration information. The novelty of this implementation lies in the way the observations used by the Kalman filter are obtained. The observation in image space is done by finding the best match with a RGB-height histogram assuming an elliptical shape for the person. At this stage, once the system is initialised, a person can be tracked using two cameras at 4-5 frames per second in a Matlab implementation that is robust to prolonged partial occlusions in either or both views at the same time. Although further testing is required this implementation looks promising.

Keywords: Person tracking, Kalman filter

1 Introduction

Robust person tracking in real-time presents quite a difficult task that, if solved would find applications in surveillance and monitoring. There are different approaches to solving the problem, each one, making different assumptions about the tracked objects, the scenes and whether cameras are static or not. There is no real elegant solution at this point that has the speed and robustness that surveillance system requires. However tackling the tracking problem as a probabilistic estimation problem seems to be, judging from the literature, the most promising. Particle filters implementations such as ones by [2] and [6] and Kalman filter implementations such as ones by [1] and [10] seem to be the ones with most success.

In this implementation the tracker/estimator is an Extended Kalman Filter. The Kalman Filter is often referred to as an optimal estimator [5]. It is optimal in the sense that is lends itself very well to the problem of combining multiple observations and a dynamic model. The Extended Kalman Filter enables estimations when there are non-linearities in the way the observations and the system dynamics relate. This is done at relatively low computational cost compared with the particle filter, where more samples are required to be evaluated.

The other main divide in the different approaches is whether the tracking takes place in the 2-D image space (e.g. [1]) or in a 3-D world-view (e.g.[10]). The second approach is suitable when the cameras are static and calibration information is available. Since this is the case for the tracking problem tackled here, we can track in 3-D. This offers several advantages, namely:

(i) Motion models with various constraints are easier to construct in world coordinates;

(ii) Occlusions are easier to resolve;

(iii) The definition of a common coordinate system in the case of multi-camera tracking configurations is made simpler.

A person being tracked is assumed to be a 3D ellipsoid of known size with his/her feet on the ground plane. The ellipsoid is chosen because it is always projected onto image space as an ellipse, making things simpler.

Observations in the image plane of each of the cameras are taken by comparing ellipse shaped image samples with available models for each of the tracked subjects. The two most common approaches are: colour histograms (used by [1] and [2]) and appearance models (used by [10] and [3]). The first approach is scale and orientation invariant, but loses all spatial information. The second makes use of spatial information but adjusting for scale and orientation changes over time is difficult.

The approach used in this implementation is a sort of compromise between the two. We bin colour information in a $(10 \times 10 \times 10)$ RGB space but we also add a n bin discretised height (along the vertical axis of the ellipsoid) dimension. We hence make use of some of the spatial information, while retaining the 'nice' properties of a histogram.

Initialisation is a separate issue from the tracking and will not be dealt with in this paper. Hence the tracker presented here assumes that the initial 3-D location of the tracked subject is reasonably close to the truth and that a reasonably good colour-height model is available.

2 The Tracking method

2.1 Method overview

Once initialised, the tracking process runs as follows, each time a new image of the scene is received:

1. *Some basic segmentation*: A background model (image) is 'subtracted' from the new image to identify foreground regions.

2. *Prediction in world view*: The location of the tracked person is predicted using the previous estimate of his/her location.

3. *Projection to image space*: The 3-D ellipsoidal space occupied by the tracked person is projected to the corresponding ellipse in image space.

4. *Observations in image space*: Using this prediction as a starting point, a search of the best match of the elliptical-template-shaped image samples to an RGB-height histogram model is performed.

5. *Update*: The best match and the associated quality of the match, together with the predicted 3-D location is used to compute the Kalman gain. This Kalman gain weighs the contribution of the measurement to make a new estimate of the person's location.

2.2 Segmentation

In this step a very basic background subtraction using an adaptive background model is performed. The difference is thresholded and foreground regions are labelled as shown in figure 1. Since the tracker does not rely only on this segmentation, it is not crucial that good segmentation is achieved but the information obtained here improves the measurement substantially.

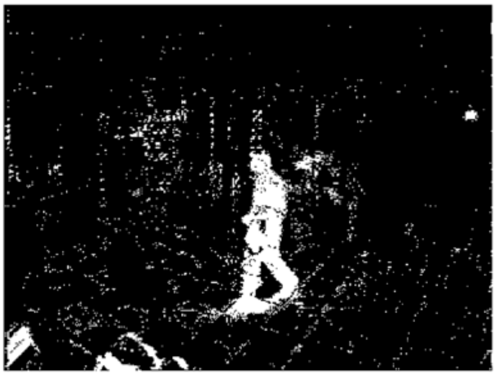


FIGURE 1: Thresholded background subtraction. The foreground pixels are highlighted.

2.3 Prediction in world-view

For each person being tracked, the system uses a single world-view model. This model describes the x and y position and velocity (a 4-D state vector: $\mathbf{x} = (x, y, \dot{x}, \dot{y})^T$), together with a measure of the uncertainty in this vector (a 4×4 covariance matrix: $\mathbf{N} = diag(\sigma_x, \sigma_y, \sigma_{\dot{x}}, \sigma_{\dot{y}})$) in the chosen 3-D world coordinate space. Each time a new frame is received from one of the cameras, $\mathbf{x}$ follows a transitional relationship of the form:

$$\mathbf{x}(t + \Delta t) = \mathbf{A}(\Delta t) \cdot \mathbf{x}(t) + |\Delta t| v(t) \quad (1)$$

And the observations $\mathbf{y}$:

$$\mathbf{y}(t) = b(\mathbf{x}(t)) + e(t) \quad (2)$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

b is the ground-plane to image-plane transformation function, Δt is the time that has elapsed since the model was last updated (when the previous frame, from the same or a different camera, was received by the system), $v(t)$ and $e(t)$ are noise sequences with zero mean and covariance matrix:

$$\mathbf{E}\begin{pmatrix} v(t) \\ e(t) \end{pmatrix}\begin{pmatrix} v(t) & e(t) \end{pmatrix} = \begin{pmatrix} \mathbf{N}(t) & 0 \\ 0 & \mathbf{W}(t) \end{pmatrix}$$

This formulation allows asynchronous predictions to be made and increases the uncertainty of the prediction with Δt. Hence, given an initial or a previous estimate of the state vector $\hat{\mathbf{x}}(t-\Delta t|t-\Delta t)$ and the associated uncertainty $\mathbf{P}(t-\Delta t|t-\Delta t)$, the prediction and associated uncertainty is given by

$$\hat{\mathbf{x}}(t|t-\Delta t) = \mathbf{A}(\Delta t)\cdot\hat{\mathbf{x}}(t-\Delta t|t-\Delta t) \quad (3)$$

$$\mathbf{P}(t|t-\Delta t) = \mathbf{A}(\Delta t)\mathbf{P}(t-\Delta t|t-\Delta t)\mathbf{A}^T(\Delta t)+|\Delta t|\mathbf{N}(t) \quad (4)$$

2.4 Projection to image space

2.4.1 Quadrics and conics

An ellipsoid is a particular configuration of a quadric. A quadric is represented in homogeneous coordinates by a symmetric 4×4 matrix $\mathbf{Q}$ such that points in space that are inside the ellipsoid will satisfy:

$$\mathbf{X}^T\mathbf{Q}\mathbf{X} > 0 \quad (5)$$

Where $\mathbf{X} = (x, y, z, 1)^T$ — the 3-D homogeneous coordinates of points in world view.
It is shown in [8] that for a normalised projective projective camera $\mathbf{P}_n = \begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix}$, the profile of a quadric $\mathbf{Q}_n = \begin{pmatrix}\mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & c\end{pmatrix}$ is a conic $\mathbf{C}$ described by:

$$\mathbf{C} = c\mathbf{A} - \mathbf{b}\mathbf{b}^T \quad (6)$$

Hence the points in the image, $\mathbf{Y}$ that satisfy (7) will lie inside the projected ellipse.

$$\mathbf{Y}^T\mathbf{C}\mathbf{Y} > 0 \quad (7)$$

where $\mathbf{Y} = (j, i, 1)$ - homogeneous pixel coordinates of points in the image space.

To obtain the image $\mathbf{Q}_n$ of a quadric $\mathbf{Q}_w$ in an arbitrary projective camera $\mathbf{P} = \mathbf{K}\begin{pmatrix}\mathbf{R} & \mathbf{t}\end{pmatrix}$, one has to first compute $\mathbf{H}$ such that $\mathbf{PH} = \begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix}$, where $\mathbf{R}$ and $\mathbf{t}$ define rotation and translation transformations, and $\mathbf{K}$, sometimes described as the intrinsic matrix, defines the scaling from metric dimensions to pixel dimensions.

Once $\mathbf{H}$ is computed, $\mathbf{Q}_n$ is calculated as follows:

$$\mathbf{Q}_n = \mathbf{H}^T\mathbf{Q}_w\mathbf{H} \quad (8)$$

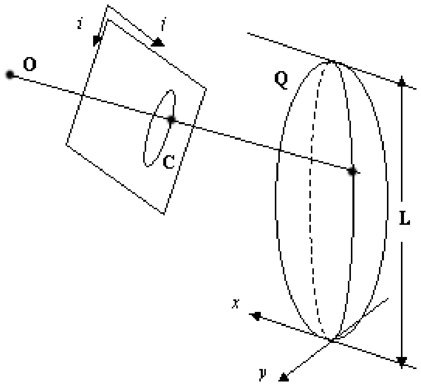


FIGURE 2: A quadric Q with its projection C on the image plane

2.4.2 From a 3-D position to an ellipse in image space

This process is broken down in the following steps:

1. Given the predicted position $\hat{\mathbf{x}}(t|t-\Delta t)$ and the height of the person L, generate $\mathbf{Q}_w(t|t-\Delta t)$ that will describe the vertical ellipsoid with centre at $(\hat{x}(t|t-\Delta t), \hat{y}(t|t-\Delta t), L/2)$ and dimensions $(L/4, L/4, L)$.
2. Calculate $\mathbf{H}$ and $\mathbf{Q}_n$ using $\mathbf{P}$ and $\mathbf{Q}_w(t|t-\Delta t)$.
3. Calculate the corresponding conic $\mathbf{C}(t|t-\Delta t)$ using (6).
4. Find the points (j_u, i_u) in the image plane that will be inside the ellipse by applying (7).
5. These are ideal, distortion-free pixel image coordinates. The real, distorted pixel image coordinates (j_d, i_d) are calculated as follows:

$$j_u = j_d + j_d(\kappa_1 j_d^2 + \kappa_1 i_d^2) \quad (9)$$

$$i_u = i_d + i_d(\kappa_1 j_d^2 + \kappa_1 i_d^2) \quad (10)$$

where κ_1 is the first order distortion coefficient of the lens.

2.5 Observations in image space

The predicted image location obtained in the previous step together with the foreground regions identified in step 1 are not sufficient for robust tracking. To distinguish between the different foreground objects we make use of the colour information contained in the image together with some colour reference for each of the tracked subjects.

2.5.1 Template matching

As mentioned in the introduction, we bin colour information in a $(10 \times 10 \times 10 \times n)$ RGB-height histogram. In the binning process only foreground pixels within the ellipse shaped samples are counted. Figure 3 shows an example of such a sample and the corresponding RGB-height histogram. The colours shown are the calculated mean RGB values for each of the n segments. A value $n = 6$ has been found to be satisfactory for a number of tracking sequences.

FIGURE 3: Ellipse shaped image sample and corresponding RGB-height histogram

The similarity measure between the model distribution $p(u)$ (generated on initialisation and updated throughout the tracking process) and an image sample distribution $q(u)$ uses the popular Bhattacharyya coefficient ([1] and [6])

$$\rho[p, q] = \int \sqrt{p(u)q(u)} du \tag{11}$$

The larger ρ is, the better the match. The next step is to find the best match ρ_{peak}, and its corresponding position $\mathbf{y}(t) = (j_{peak}, i_{peak})$.

2.5.2 Finding the best match

Since the computation involved in the histogram representation and matching is the bottleneck of the whole tracking process, one would like to keep the number of image samples required to find $\mathbf{y}$ as low as possible.

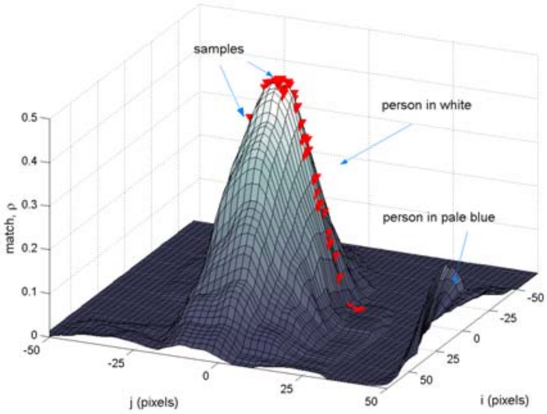


FIGURE 4: Surface plot of ρ and samples points

Mean-shift type searches [1], which usually don't require many iterations, cannot be implemented using the chosen model representation because of the spatial information it contains. So the 3 approaches that were considered were:

(i) Multi-scale exhaustive search within $\mathbf{P}_p(t|t - \Delta t)$

(ii) n random samples from the normal distribution $N(I, \mathbf{P}_p(t|t - \Delta t))$

(iii) The Simplex search method [4], implemented by the Matlab *fminsearch* function with limited iterations.

Good results were obtained using approach (iii), with a 20 iterations limit and the projected predicted location picked as the starting point. But however, in more cluttered scenes, approach (ii) with as few as 20 samples was found to be robust to mixing up people. Approach (i) proved to be the most robust but required a lot more than 20 samples.

Figure 4 shows the template matching output $\rho(j, i)$ in the neighbourhood of the image of a tracked subject. Also shown are the samples that were taken from the distribution $(\hat{x}_p(t|t - \Delta t), \mathbf{P}_p(t|t - \Delta t))$.

2.5.3 Calculating the projected prediction uncertainty $\mathbf{P}_p(t|t - \Delta t)$

The prediction uncertainty P, is projected to image space $\mathbf{P}_p(t|t - \Delta t)$ by the Jacobian of the image to world transformation:

$$\mathbf{P}_p(t|t-\Delta t) = \begin{pmatrix} \frac{\partial j}{\partial x} & \frac{\partial j}{\partial y} \\ \frac{\partial i}{\partial x} & \frac{\partial i}{\partial y} \end{pmatrix} \times \mathbf{P}(t|t-\Delta t) \times \begin{pmatrix} \frac{\partial j}{\partial x} & \frac{\partial j}{\partial y} \\ \frac{\partial i}{\partial x} & \frac{\partial i}{\partial y} \end{pmatrix}^T \tag{12}$$

The uncertainty associated with the measurement $\mathbf{y}(t)$ is scaled by the quality of the match found by *fminsearch*:

$$W(t) = \frac{1}{\rho(\mathbf{y}(t))} \times \begin{pmatrix} \sigma_j & 0 \\ 0 & \sigma_i \end{pmatrix} \tag{13}$$

where a value of 5 pixels for σ_j and σ_i seems to be good for this particular configuration.

2.6 Update

The update step can be summarised as follows:
Given an observation $\mathbf{y}(t)$, the predicted state $\hat{\mathbf{x}}(t|t-\Delta t)$, and their respective uncertainties $\mathbf{W}(t)$ and $\mathbf{P}(t|t-\Delta t)$, make an optimal estimate of the location $\hat{\mathbf{x}}(t|t)$ and its associated uncertainty $\mathbf{P}(t|t)$. This is done using the Kalman filter formulation as treated in [9] and [7]:

$$\hat{\mathbf{x}}(t|t) = \hat{\mathbf{x}}(t|t-\Delta t) + \mathbf{K}(t)[\mathbf{y}(t) - b(t, \hat{\mathbf{x}}(t|t-\Delta t))] \tag{14}$$

$$\mathbf{P}(t|t) = \mathbf{P}(t|t-\Delta t) - \mathbf{K}(t)\mathbf{B}(t)\mathbf{P}(t|t-\Delta t), \tag{15}$$

where $\hat{\mathbf{x}}(t|t-\Delta t)$ and $\mathbf{P}(t|t-\Delta t)$ were calculated in the prediction step using (3) and (4).

Since $b(t, \mathbf{x})$ is non-linear, $B(t)$ is calculated by locally linearising b at $\mathbf{x} = \hat{\mathbf{x}}(t|t-\Delta t)$:

$$B(t) = \left.\frac{\partial b(t, \mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(t|t-\Delta t)} \tag{16}$$

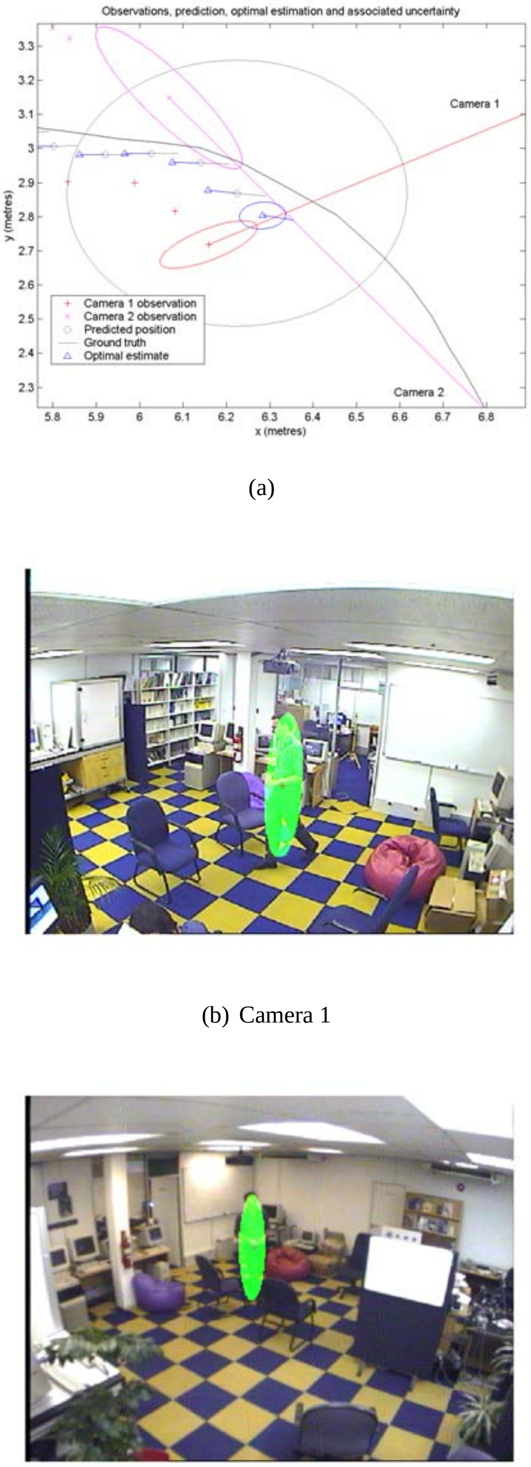
This gives:

$$B(t) = \begin{pmatrix} \frac{\partial j}{\partial x} & \frac{\partial j}{\partial y} & 0 & 0 \\ \frac{\partial i}{\partial x} & \frac{\partial i}{\partial y} & 0 & 0 \end{pmatrix} \tag{17}$$

The Kalman gain is calculated as follows:

$$\mathbf{K}(t) = \mathbf{P}(t|t-\Delta t)\mathbf{B}^T(t) \cdot [\mathbf{B}(t)\mathbf{P}(t|t-\Delta t)\mathbf{B}^T(t) + \mathbf{W}(t)]^{-1} \tag{18}$$

Figure 5(a) shows a close-up view, as seen from the top, of the predicted person position, the two observations, and the optimal estimate together with their respective uncertainties. Figure 5(b) and 5(c) show the 2 frames from which the observations were made and the projected ellipse at the estimated location.



(a)

(b) Camera 1

(c) Camera 2

FIGURE 5: Views from camera 1 and 2 at that particular instant with estimated position of ellipsoid projected back onto the image

3 Some results

Figure 6 shows the result of the tracking of one person inside a room using two cameras. The ground truth (done by hand) is shown as the dark line and the estimates are shown by the triangles. This particular sequence has an instance of complete occlusion from camera 1 and partial occlusion from camera 2. Once initialised, the tracker currently tracks a single person, from one view, at roughly 4-5 frames a second. This result is achieved on a Pentium 2.4 GHz, with image size of 384×288 with ellipses containing 1000-4000 pixels (roughly 1-4$\%$ of the total image area).

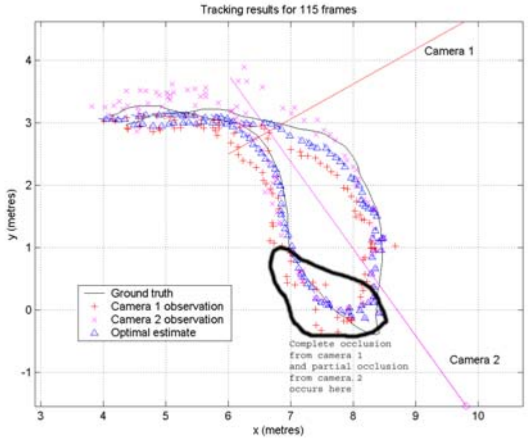


FIGURE 6: Tracking results for 115 frames

4 Conclusions

In this paper a 3D tracker making use of multiple observations was presented. The use of an Extended Kalman filter as the optimal estimator/tracker was demonstrated. A novel 4 dimensional RGB-height (along projected ellipse's longer axis) histogram for matching ellipsoidal-shaped templates was tested and seems to be a good compromise between colour histograms and appearance models offering the advantages of each approach for representing the subject being tracked. Further testing of the algorithm still needs to be performed. There is a lot of scope to improve on the current processing speed.

Acknowledgements

The author would like to thank NRF and TSS Technology, De Beers for financial support.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Porc. of IEEE Conf. on Comp. Vis. and Pat. Rec (CVPR00), pages 142–151, San Francisco*, CA, 2000.

[2] M.Isard and J.MacCormick. BraMBLe: a Bayesian multiple-blob tracker. In *International Conference on Computer Vision*, 2001.

[3] A. Jepson, D. Fleet and T. E1-Maraghi, Robust Online Appearance Models for Visual Tracking, *CVPR, pp. 415-422, Vol. 1*, 2001

[4] Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright, "onvergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *SIAM Journal of Optimization*, Vol. 9 Number 1, pp. 112-147, 1998.

[5] Peter S.Maybeck. "Stochastic models, estimation and control — Volume 1". Academic Press, 1979.

[6] K. Nummiaro and E. Koller-Meier and L.Van Gool. "Color-based real-time recognition and tracking", International Symposium on Mixed and Augmented Reality (ISMAR 02), 2002.

[7] Söderström. "Discrete-time Stochastic Systems". Springer, 2002.

[8] B.Stenger, P.R.S.Mendonça, and R.Cipolla. Model-based 3D tracking of an articulated hand. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume II, page 310-315, December 2001.

[9] Greg Welch and Gary Bishop. "An Introduction to the Kalman Filter", http://www.cs.unc.edu/ [welch,gb].

[10] Tao Zhao, Ram Nevatia and Fengjun Lv. "Segmentation and Tracking of Multiple Humans in Complex Situations" (*CVPR01*).