Real-Time Object Tracking from a Moving Video Camera: A Software Approach on a PC

Yoav Rosenberg **Michael Werman**
Institute of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
{yoavr, werman}@cs.huji.ac.il

Abstract

We demonstrate a real time system for image registration and moving object detection. The algorithm is based on describing the displacement of a point as a probability distribution over a matrix of possible displacements. A small set of randomly selected points is used to compute the registration parameters. Moving object detection is based on the consistency of the probabilistic displacement of image points with the global image motion.

1 Background

Detecting moving object in a video sequence is usually done by aligning pairs of images and creating the difference image. Moving objects can be found in the difference image using algorithms such as clustering [1, 2, 5, 3]. Unfortunately image alignment is usually not exact at every image point, and the difference image is noisy, causing false alarms. Another problem with image alignment and subtraction is execution time, as every image point is examined in both the registration, the alignment, the subtraction, and the clustering. Therefore, only special harward could be used if a real-time system was needed [1, 3].

As an alternative, we introduced in [4] a method to describe the displacement of a point p as a discrete probability distribution matrix $Y_p(u,v)$. The matrix is computed using the sum of squared difference (SSD) values: $Y_p(u,v) = K * exp\left(-SSD_p(u,v)/\sigma\right)$, where $SSD(\mathbf{d}) = \sum_{i,j \in \mathcal{W}} \left(\Psi_2(i+\mathbf{d}_x, j+\mathbf{d}_y) - \Psi_1(i,j)\right)^2$ for a window $\mathcal{W}$ around the point p in image Ψ_1.

2 Image Registration

Consider the simple case where the motion model is a uniform image-plane translation. Given the matrices $Y_p(u,v)$ at N points, the distribution of the displacement of all the points is: $P_{sum}(u,v) = \sum_{i=1..n} Y_i(u,v)$. $P_{sum}(u,v)$ is the **expected** number of points with a displacement of $d = (u,v)$. Normalizing P_{sum} gives a probability matrix approximating the distribution of the displacement of the image points. P_{sum} is very robust for outliers and for image noise. The robustness applies also when the image motion is not an exact translation. In this case the maximal entry in the sum matrix belongs to the dominant translation and the matrix as a whole shows the deviation from the dominant translation.

Registration of Translation and Rotation: Assume that the rotation θ is known and the image-plane translation should be found. In this case, each matrix is shifted with the offset induced by the rotation and the sum matrix P^{θ}_{sum} is computed. To find an unknown rotation angle, different angles are searched with steps of $\Delta\theta$. For each angle θ in the range, the matrix $P^{\theta_j}_{sum}$ is computed, and the chosen angle $\hat{\theta}$ is the one which maximizes a measure of the sharpness of the matrix P^{θ}_{sum}.

Registration of an Affine Transformation: A similar approach can be developed to compute an affine transformation: $\Delta x_i = a_{11}x_i + a_{12}y_i + w_x - x_i$, $\Delta y_i = a_{21}x_i + a_{22}y_i + w_y - y_i$.

Theoretically, a sequential search can be done for the parameters set $A = \{a_{11}, a_{12}, a_{21}, a_{22}\}$. However, searching in a 4-dimensional space is very slow. An alternative method is to perform a separate search for the two pairs: $\{a_{11}, a_{21}\}$ and $\{a_{12}, a_{22}\}$, reducing the problem into a double 2-dimensional search.

Consider the case where all the points have the same value of y: $\mathbf{p}_i = (x_i, y)$ so that the transformation is reduced to: $\Delta x_i = a_{11}x_i + k_1 - x_i$, $\Delta y_i = a_{21}x_i + k_2 - y_i$ where: $k_1 = a_{12}y$ and $k_2 = a_{22}y$. In this case, the relative shift between the matrices depends only on a_{11}, a_{21}. Nevertheless, not all the points have the same y, therefore, the image is divided into M horizontal strips S_j, $j = 1..M$ so that all the points $\mathbf{p}_i \in S_j$, have roughly the same y. We search sequentially for a_{11}, a_{21} and for each guess a sum

matrix is computed separately for the points belonging to each strip S_j:

$$P^A_{s_j}(u,v) = \sum_{\mathbf{p}_i \in S_i} Y_i(u - \Delta x_i(A), v - \Delta y_i(A))$$

where: $A = \{a_{11}, 0, a_{21}, 1\}$. The best choice for a_{11}, a_{21} is the one that maximizes:

$$\sum_{j=1..M} Q(P^A_{s_j}(u,v))$$

where $Q(\mathbf{B})$ is the quality measure of the matrix $\mathbf{B}$.

After finding the best pair $\{\hat{a}_{11}, \hat{a}_{21}\}$, the remaining two parameters $\{a_{12}, a_{22}\}$ can be found with a simple two dimensional search. Let $A = \{\hat{a}_{11}, a_{12}, \hat{a}_{21}, a_{22}\}$. We search for $\{a_{12}, a_{22}\}$, choosing $\hat{A} = maxarg\{Q(P^A_{sum}(u,v))\}$.

After $\hat{A}$ is found, $P^{\hat{A}}_{sum}(u,v)$ can be used as the estimation of the probability distribution of the displacement of a pixel under the computed registration parameters and the maximum likelihood choice for $\{w_x, w_y\}$ will be: $\{\hat{w}_x, \hat{w}_y\} = maxarg_{u,v}(P^A_{sum}(u,v))$

3 Coarse to Fine Registration

The registration algorithm is based on the probability distribution matrices at N points. For real-time implementation, matrix sizes should be kept small. To enlarge the dynamic range of the registration, image pyramid can be used. However, the implementation in real-time is very hard. As an alternative we use a more efficient approach especially tailored for our registration implementation.

Currently, our system uses a three levels coarse to fine implementation with a matrices size of $(-3..3, -3..3)$. At the top level the matrices at N points are computed with jumps of 9 pixels, so that the displacement range covered by the matrix is $-27..27$ pixels, The registration parameters are computed at this level and are used as the base registration for the next level. At the next level, N matrices are computed with a jump of 3 pixels. At the bottom level, N matrices are computed with a jump of one pixel.

4 Moving Object Detection

After global image registration, moving objects can be detected. A uniform grid of "detectors" are deployed on the image. Each detector is an image point whose displacement probability is computed. The displacement probability matrix of each detector is compared to that of the background. These comparison gives two measures: P_m - The confidence that the point moves differently than the background. P_b - The confidence that the point moves as the background. These two measures are not reciprocals, as a flat matrix has no confidence of any type. This measures are more robust to noise than image difference and they take into account the uncertainty of the image registration.

Figure 1. Examples of moving objects detected from a moving camera.

Clustering points into moving objects Moving object are detected and tracked using the detectors grid. A moving object is defined as a region R where $\sum(P_m - P_b) > 0$. The boundaries of the object should be those who maximize $\sum(P_m - P_b)$. This reflects the assumption that a moving object contains points with a motion different than the background. The measures are further used to decide if a cluster belongs to a single moving object or to multiple objects.

References

[1] P. Burt, J. Bergen, R. Hingorani, R. Kolczynski, W. Lee, A. Leung, J. Lubin, and H. Shvaytser. Object tracking with a moving camera. In *IEEE Workshop on Visual Motion*, pages 2–12, 1989.

[2] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In G. Sandini, editor, *Second European Conference on Computer Vision*, pages 282–287, Santa Margherita, Italy, May 1992. Springer.

[3] P. Nordlund and T. Uhlin. Closing the loop: Detection and pursuit of a moving object by a moving observer. *Image and Vision Computing*, 14(4):265–275, May 1996.

[4] Y. Rosenberg and M. Werman. Representing local motion as a probability distribution matrix and object tracking. In *DARPA Image Understanding Workshop*, pages 153–158, New Orleans, Louisiana, May 1997. Morgan Kaufmann.

[5] H. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):814–830, August 1996.