MultiBlob Particle Filter

Markus Louw Fred Nicolls Gerhard de Jager

Abstract

Particle filters can be used for motion tracking in monocular image sequences. This tracker uses the multiple bayesian tracking approach (BraMBLe), with the foreground image obtained through background subtraction, as well as prior colour information in the observation stage. With each particle representing the world coordinates of approximation ellipsoids of all of the people in the scene, it takes approximately six hundred particles per frame to track four people in a known environment. The tracking is stable for a single tracking object, and progressively less reliable as more tracking objects are added. The tracker survives short periods of occlusion with fewer than four people .

1 Introduction

This paper is loosely based on the work of M. Isard and J. MacCormick [1]. Particle filters can be used to propagate weighted estimates of parameter values representing object world-coordinates in a video sequence. In [1], a single parameter set representing a set of ellipsoids, as well as the visibility of any ellipsoid were propagated. Occlusions, where one person obscures another, are modelled by the corresponding occlusion of one ellipsoid by another, or by the setting of the non-visibility parameter of the occluded ellipsoid. In this scheme, we do not randomly vary the visibility of the ellipsoid, since a person being occluded should ideally be perfectly represented by the occlusion of the ellipsoid representing him by the ellipsoid of another person. The framework allows for static occlusion by inactive objects in the scene, but this has not been implemented.

The choice of using an ellipsoid to model an individual is arbitrary, any 3D geometric shape could have been chosen, and an ellipse is a simple shape.

2 Observation likelihood

2.1 Image data

The parameter vector of each particle is represented as:

$$X = (X_1, X_2, X_3, X_n) \tag{1}$$

with

$$X_n = (x_n, y_n, z_n) \tag{2}$$

the spatial coordinates of the bottom of the ellipse. Note that the ellipsoids themselves are of constant size, since we assume the size of the people in the sequence to remain reasonably constant.

For each multi-blob parameter set which is propagated, a set of ellipsoids are generated. These ellipsoids are tested against the corresponding image frame for that particle. The conditional probability,

$$P(Z|X)=P(im_z(Z),f(Z)|im_x(X))=$$
$$G(\sum_{i=0}^{N} d(p(im_z(Z),\, i),p(f(Z),\, i),p(im_x(X),\, i)),p(im_x(X),E_d) \tag{3}$$

with im_z a function returning the current image frame from image data Z, and im_x a function returning the virtual image generated by the ellipsoid configuration, f (foreground) a function returning the foreground image, obtained by subtracting the current image from a known background image, and p (pixel) is a function which returns the pixel 3-value of a particular pixel in an image. The distance function, d, measures the distance between the particle generated image and the image data of a particular frame. The pixel function p returns the 3-value of the pixel at the specified index i, given an image. The value E_d is to assist the algorithm, to give it an expected value for the distance of a good particle for that image. It is used for particle weight adjustment.

The function G is there to adjust the resulting value to accentuate low distances, and to normalize the total image distance, by dividing it by the number of ellipse pixels (non-zero pixels) in the virtual image generated by a particle, and this is the purpose of the second parameter. Ideal particle weights in a particle filter are very small (10^{-20}) for inappropriate particles, and larger (10^{-1}) for better particles, and the method for achieving this range is explained in the next subsection.

For the total set of ellipsoids generated per particle, we then scan each pixel of the image data, and depending on which ellipsoid from the generated ellipsoid set was nearest in world coordinates (an ellipsoid in the same area, but behind another such ellipsoid, would be occluded), that pixel is assigned a "distance" measure. This procedure is a way of assigning an observation value to individual pixels.

The distance measure is used to exploit the known colour information of the individuals in the scene. The people in this sequence are in fact highly colourized: one is wearing orange, another yellow, etc. A pixel generated by a particular ellipse will correspond to the colour model of a particular person, and here, the distance measure associated with the pixel i is

$$\begin{aligned} d(p_1, p_2, p_3) = \sqrt{\Delta R^2 + \Delta G^2 + \Delta B^2} \\ + foreground(p_2) \end{aligned} \tag{4}$$

where

$$\begin{aligned} \Delta R &= (R_m(p_1) - R_e(p_3)) \\ \Delta G &= (G_m(p_1) - G_e(p_3)) \\ \Delta B &= (B_m(p_1) - B_e(p_3)) \end{aligned} \tag{5}$$

with R_m, G_m, B_m the expected colour likelihoods of a pixel in the elliptical region around the target person in image space (these values can be accessed through a table, at run-time, containing the expected colour values for each person/ellipse), and reasonable values for these can be estimated or computed. R_e, G_e, B_e are the actual values in the image data for the pixel values in the region of one of the ellipsoids projected onto the image space. If no ellipse is projected onto a particular pixel, that pixel does not contribute to the distance measure. The $foreground$ function returns a high value if the p_2 pixel 3-value is zero, as it will be returned by the function f if the pixel is in the background.

If a pixel does not fall in the foreground region generated by the image background subtraction, it has an additional very high distance value added to it. With this type of observation, the distance of a virtual image comprising the hypothetical ellipsoid shapes can be compared very quickly with the image data, since each pixel is visited once only per particle.

2.2 Adjusting the distance measure to obtain an appropriate particle weight

The particle-probability configuration of an image sequence changes over time, so although relative measures for particle suitability should always remain in ordered relation to one another, the actual probability measures found for appropriate particles changes over time. This is usually dealt with in the normalization stage of particle propagation. However, if we need to place an accentuation function on the weights of appropriate particles we need to maintain a measure of a reasonable distance for appropriate particles, for an arbitrary point in the sequence.

The accentuation function is defined as follows:

$$\begin{aligned} G(dist, im_x(X), dist_{exp}) = \\ N(L(\tfrac{dist}{count(im_x(X))}, dist_{exp}); 0, 1) \end{aligned} \tag{6}$$

where $N(\cdots; 0, 1)$ is a Gaussian curve with mean zero and standard deviation of one, and the function L linearly separates (preprocesses) the distances. The parameter $dist$ is the distance of a particle, and the $dist_{exp}$ parameter is the expected distance for a particle. The $count$ function takes the image data returned by the $im_x(X)$ function, and returns the number of non-zero pixels in the image of the ellipsoids generated by the particle (see Fig 4.). The separation function L was used with continually updated values for the expected distance value ($dist_{exp}$) for particles in a frame. This value was updated by assuming the shortest particle distance from the previous frame to be the same as the expected distance particle in the next frame.

3 Initialization

The tracker was initialized with prior information concerning the time and location of arrival of new people in the tracking scene. Therefore the number of tracking objects is known to be dynamically variable. In the original

Figure 1: **Image data frame with ellipsoids of particle superimposed**

BraMBLe implementation [1], ellipsoids had associated 'survival' and 'arrival' probabilities. While this is an alternative, searching for and distinguishing new objects is not in the scope of this implementation.

4 Results

The Multi Blob tracker of this implementation was able to track a single person past occlusions indefinitely. The number of particles per frame required for successful tracking varies with the number of the people in the sequence. For a single person, thirty particles are required. For four people, six hundred particles suffices, however occlusion can become a problem when a person occupying a small region in the image is covered by a much larger person, who is closer to the camera. The tracking in this implementation does not occur in real time. The platform is not optimized for speed, and on a P3 approximately ten particles can be evaluated per second.

5 Discussion

The algorithm itself is appropriate for tracking multiple objects in a robust fashion. There are many possible areas of development in this system which could improve its performance. The observation method was hand-optimized for this scene, and while it is reasonable to assume the foreground image should contribute heavily

Figure 2: **Background of scene**

Figure 3: **Foreground of scene (image data)**

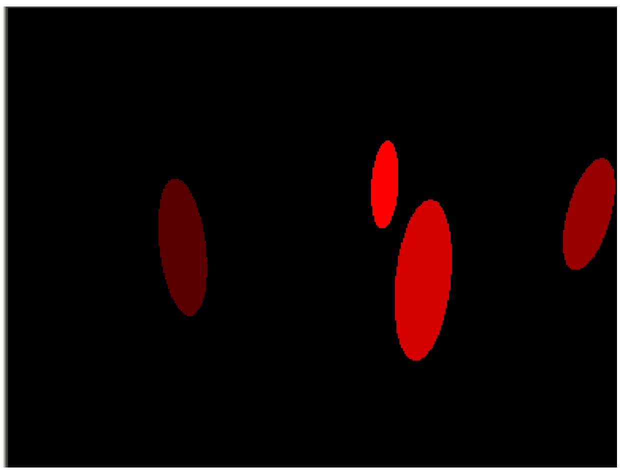

Figure 4: **Virtual scene comprising ellipsoids generated by particle. This virtual scene was hand-calibrated.**

to the selection of any particle, and the colour data should contribute less, this may not always be the case, and other ways of observing a particle could be employed.

An interesting effect was that when the people were lined up at the center of the image, a particle would evolve in such a way that it minimizes the size of the ellipsoid for that person, by moving the ellipsoid further away from the camera. Thus, a very tiny, distant, ellipse has a better observation value than a large one which includes parts of the background around its tracking target.

A way of preventing this would be to restrict the available parameter space of any particle according to the known dimensions of the scene. Particle dynamics, which take into account the momentum of a particle at any time could be used to sample more effectively from the prior. Parameter restrictions and dynamics were not implemented here.

6 Conclusion

The BraMBLe algorithm, with foreground and colour based observations, is a suitable algorithm for tracking multiple people occluding one another in a sequence. The number of particles required for tracking varies with the number of tracking objects in the scene.

There is room for for future research in sampling methods, particle dynamics modelling methods, parameter restraints, and particle observations.

References

[1] M. Isard, J. MacCormick. BraMBLe: A Bayesian Multi Blob Tracker Proc. Int. Conf. Computer Vision. Vol 2, 2001, pages 34-41

[2] D. Reynard, A. Blake, M. Isard. Learning to track the visual motion of contours Artificial Intelligence vol 78 pages 101-133, 1995

[3] S. Kirkpatrick, C. Gelatt, M. Vecchi. Optimizing by Simulated Annealing Science May1983 nr 4598 vol 220 pages 671-680, 1983

[4] Michael Isard. Visual motion analysis by probabilistic of conditional density University of Oxford, Department of Engineering Science, 1998

[5] P. Fearnhead, J. Carpenter, P. Clifford. An improved particle filter for non-linear problems. Technical Report. Vol 220, 1983

[6] G. Kitigawa. Monte carlo filter and smoother for non-gaussian non-linear state space models Journal of Computational and Graphical Statistics, vol 1, 1996 pages 1-25