Object tracking using histogram matching and particle filter

Report Author: Kildo Alkas Alias
Group Partner: Oskar Erlandsson

***Abstract*—Tracking objects in images has many useful applications ranging from security to commercial. There are many different ways to detect and track an object in images with different advantages and drawbacks. In this paper, a sliding window histogram has been implemented for detection and a particle filter is implemented for tracking. The performance is evaluated by attempting to track a ball. The conclusion is that the particle filter offers good performance in tracking but is highly dependent on the performance of the histogram matching.**

***Index Terms*—Histogram matching; Tracking; Particle Filter**

I. Introduction and problem statement

To be able to reliably predict where an object has been, where it is and where it might go the object has to be detected and tracked. Tracking objects using vision is something humans do effortlessly and reliably every day. For a computer using cameras as sensors, this is a much greater challenge. The object has to first be detected, its position calculated and then tracked.

Object tracking using computer vision has useful applications in many different fields. It can, for example, be used to track people using security cameras as a security measure. It can also be used to track an object in commercial applications such as tracking a table tennis ball in a game to either aid or replace the referee or to provide visual aids to the audience. Using cameras to aid in refereeing sports games are gaining popularity. It is already used in hockey and in soccer but with human operators. Object tracking could be used to automate these kinds of aids.

There are different ways to detect and track an object in an image. Some methods are semi-autonomous, i.e require human intervention such as drawing a rectangle around the object that you want to track and those methods in many cases involve some kind of image segmentation such as mean-shift, Ncut or graph cut [1]. Our implementation is fully autonomous, it will detect an object and track it without any human intervention.

Object tracking in our application is a two-part problem, the first being to be able to detect the object and gain a measurement of its position and then to be able to track it. For detection, we use a sliding histogram method where the histograms from the image are compared to a reference. After the object is detected a measurement from the sliding histogram algorithm is used in a particle filter to track the object.

Our contributions are

- Implementing existing histogram detection algorithms
- Implementing existing particle filter algorithm for tracking
- Testing and analysing the combination of detection and tracking on balls used in sports to test the filters capacity to track objects

The next section will highlight other works related to histogram matching and particle filters. In section three the implementation of object detection is explained as it plays the important role of providing measurements to the tracking algorithm and the implementation of the tracking using particle filter is explained. The result is shown in section four and discussed in section five. A short conclusion is given in section six.

II. Related works

Sliding windows is a concept that has been used for a long time but the formal definition of a sliding window model was given by Datar et al. in 2002 [2]. They proposed a way of maintaining aggregates and statistics over data streams, for example, approximate histograms. F.S. Mohammed et al. show in their 2010 study how histograms can be used for color detection similarly to other methods as for example euclidean distance and they show that histogram matching is more useful than other methods in certain applications [3]. Histograms are now commonly used to detect an object in an image. In their report published in 2010 Mehta et al. uses histogram matching and the Kalman filter in a real-time tracking system to be used in a static camera such as a surveillance camera [1]. Our implementation is inspired by their work but since the use case is different the implementation differs slightly. We use the particle filter which can be traced back to a paper published in 1949 by Metropolis and Ulam where the method is presented [4]. The particle filter has proven successful for non-linear and non-Gaussian estimation problems such as an adaptive color-based particle filter used to track a non-rigid object as presented by K. Nummiaro et al. in their 2003 report [5]. How to use the particle filter in non-linear and non-gaussian applications is described well in a paper written in 2002 by M. S. Arulampalam et al. with the descriptive title "*A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*". They review some different optimal and suboptimal filters and also suggest some different resampling methods.

Many resampling methods exist for the particle filter and a comparison of these are made by R. Douc et al. in their 2005 paper [6]. A common resampling method for the particle filter is systematic resampling but it has some drawbacks. X. Fu and Y. Jia suggests a method to improve the resampling method in a 2010 paper [7].

III. Implementation and Theory

A. Implementation structure

The measurement from the image pre-processing is used in a particle filter for tracking. The particle filter uses a fixed motion model. The fixed motion model is chosen because the path of the object can be unpredictable. By previous experiments done a fixed motion model appears to also be the least sensitive to the choice of noise parameters which is an advantage in an autonomous tracking system. Because the object is moving but is modeled with a fixed motion model it is important to have a high process noise. When the object detection algorithm finds the object the measurements are often on target and therefore a smaller measurement noise is used. Systematic resampling is used as it is favorable in computational complexity [8] and because it is a variance reducing method that is simple to implement [6].

B. Image pre-proccessing

The histogram matching is done by taking the sum of the absolute difference of the grayscale histograms of two images, similar to the method described in [1]. One of the images is a reference image, it contains the object that we want to track and the other is an image where we want to find the object. A window slides over the image and at every pixel, a histogram of the window is created and matched with the reference histogram. The window histogram that has the lowest sum of absolute difference from the reference is the best matching window. The corresponding image from the window is then thresholded to create a binary image and extract the center of the object. The center of the object in the image frame is then used as a measurement in the tracking.

C. Particle filter

Particle filters are sequential Monte Carlo methods based on particle representation of probability densities. The main advantage is they can be applied to any state-space model and is convenient for handling multivariate data and nonlinear processes.[7, 9]. Nonlinear Bayesian tracking, which the particle filter is a variant of, essentially consists of two stages that repeat recursively; predict and update. However, since the particle filter suffers from a degeneracy problem it is practical to add resampling as a third stage. The degeneracy phenomenon is that after a few iterations all but one particle will have negligible weight. The basic idea is to concentrate on particles with larger weight and eliminate particles with smaller weight [7].

To initialize the particle filter M particles are randomly sampled from the valid positions in the image and given equal weights. Until a first measurement is given the object is equally likely to be anywhere in the image. The position of the particles is used in the predict step to calculate new positions for all the particles. Since a fixed motion model has been chosen the prediction step consists of only diffusion of the x and y positions. When a measurement is given the new weights are then calculated by first calculating the innovation between the particle positions and the given measurement and then using a maximum likelihood function to calculate the new weights. Since the measurement noise is modeled as Gaussian noise the likelihood function is given by equation 1. Outliers are detected by thresholding the mean of the weights and weights are set to equal values if an outlier is detected. The weights are then normalized by the sum of the weights.

$$p\left(z|x,\Sigma_Q\right) = \frac{1}{2\pi|\Sigma_Q|^{\frac{1}{2}}} e^{-\frac{1}{2}(z_t-x_t)^T \Sigma_{Q_t}^{-1}(z_t-x_t)} \tag{1}$$

The filter is then resampled using the systematic resampling algorithm, sometimes also called low variance resampling. The resampling algorithm computes a single random number and selects samples according to the number and with a probability proportional to the weights. A random number is drawn in the interval of [1, 1/M], where M is the number of particles. Then at every 1/M interval, a particle is sampled [10]. After resampling the belief is finally calculated as the mean of the particles.

IV. Results

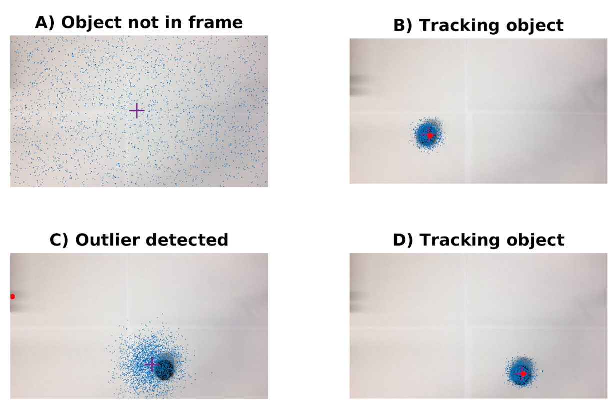

Fig. 1. A somewhat blurry data set where the object detection is not optimal. Some parts of the image are mistaken as the ball because of shadows. The red dot is the measurement, purple cross is the belief and the blue dots are the particles.

Two different scenarios are presented here. Both are simulated with 3000 particles. The red dot is the measurement provided by the object detection algorithm, the purple cross is the calculated belief and the blue dots are the particles of the particle filter. Note that the ball is rolling from left to right in Figure 1 but right to left in Figure 2

The first in Figure 1 is a sub-optimal data set where the camera was handheld and some shadows were present. This makes the images more blurry and can confuse the object detection somewhat. In frame A we can see that the

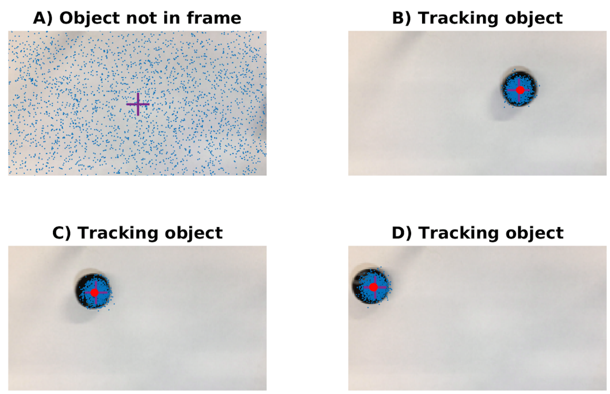

Fig. 2. A data set where the camera is more steady and less shadows are present. The ball is tracked for most of its path. The red dot is the measurement, purple cross is the belief and the blue dots are the particles.

particles are spread evenly over the image as no measurement has yet been provided. In frame B we can see that the object has been detected and is being tracked by the particle filter. In frame C a shadow on the upper left side of the image is mistaken as the ball but since the particles are far away from the measurement it is detected as an outlier. In frame D we can see that the object detection has found the ball again and the particle filter converges to its position.

In Figure 2 a more optimal data set has been tested. The camera is on a mount and fewer shadows are present. As in Figure 1, frame A shows the particles prior to object detection. In frames B, C, and D we can see that the ball is being tracked for most of its path as the object is easier to detect by the object detection algorithm.

V. Discussion

The tracking works well for some scenarios. When good measurements are provided the ball is reliably tracked by the particle filter. Even if the object detection misses some frames or mistakenly detects another object the particle filter still provides a reasonable belief. The outlier detection discards most of the incorrect measurements. Because the process noise has to be relatively high due to the fixed motion model the filter can not go too long without a measurement. If too many frames lack a correct measurement the filter will eventually lose track of the ball. A remedy to this issue could be to instead of a fixed motion model a linear motion model can be applied. If a good motion model is used a lower process noise can be used. A linear motion model could be calculated by using the difference in frames to estimate the trajectory.

For a simple case of a black ball on a white background the object detection algorithm works well and provides correct measurements most of the time. For more complex scenes with many different colors, or even just some shadows, the detection algorithm can struggle to find the correct object. For such scenes, the modeled measurement noise must be increased and a good reference image must be provided.

The object detection was done here with a grayscaled image. Detection using RGB images were also tested but only showed a marginal improvement in measurements at the cost of significant computational cost. The detection and tracking algorithms can be improved to include more measurements than one to be able to track multiple hypotheses or even multiple objects in the same frame. The issue of only using one measurement is that if the object detection mistakenly gives the first measurement far away from where the object might be in the frame then it will take some time before the object is tracked by the filter as many of the subsequent measurements will be classified as outliers even though they are correct measurements.

3000 particles proved sufficient in this application but can be easily increased if needed. More particles are needed in the case of a data set with many incorrect measurements and fast-moving objects.

The detection algorithm is too slow to be used in live applications but can be improved. Instead of calculating an entire new histogram for each window only a histogram for the new column is calculated. The new column histogram is then added and the old column histogram is removed from the total as the window slides over the image. This method should provide sufficient speed to be used in live applications [11].

VI. Conclusion

We have shown that the particle filter can be used to track an object in an image and works well given good measurements. The particle filter can track fast-moving objects and is suited to track for example a ball given that a good reference image is provided. With improved computer vision algorithms for better measurements, the particle filter can achieve good accuracy and reliability. A motion model can be calculated by for example taking the difference in frames to further improve the particle filter tracking. The use of grayscale images for object detection was sufficient and was a good trade-off between accuracy and speed. Speed improvements can be made to the object detection algorithm by using efficient histogram methods if the tracking needs to be used in live applications.

References

[1] Madhur Mehta, Chandni Goyal, M. C. Srivastava, and R. C. Jain, "Real time object detection and tracking: Histogram matching and kalman filter approach," in *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, vol. 5, Feb 2010, pp. 796–801.

[2] M. Datar, A. Gionis, P. Indyk, and R. Motwani, "Maintaining stream statistics over sliding windows: (extended abstract)," in *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '02. Philadelphia, PA, USA: Society for

Industrial and Applied Mathematics, 2002, pp. 635–644. [Online]. Available: http://dl.acm.org/citation.cfm?id=545381.545466

[3] F. S. Mohamad, A. A. Manaf, and S. Chuprat, "Histogram matching for color detection: A preliminary study," in *2010 International Symposium on Information Technology*, vol. 3, June 2010, pp. 1679–1684.

[4] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, 1949, pMID: 18139350. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1949.10483310

[5] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99 – 110, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885602001294

[6] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering," in *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, Sep. 2005, pp. 64–69.

[7] X. Fu and Y. Jia, "An improvement on resampling algorithm of particle filters," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5414–5420, Oct 2010.

[8] J. D. Hol, T. B. Schon, and F. Gustafsson, "On resampling algorithms for particle filters," in *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, Sep. 2006, pp. 79–82.

[9] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb 2002.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2005.

[11] Y. Wei and L. Tao, "Efficient histogram-based sliding window," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3003–3010.