Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces

John G. Allen, Richard Y. D. Xu, Jesse S. Jin

School of Information Technologies
University of Sydney
Madsen Building F09, University of Sydney, NSW 2006

jallen@it.usyd.edu.au

Abstract

The Continuously Adaptive Mean Shift Algorithm (CamShift) is an adaptation of the Mean Shift algorithm for object tracking that is intended as a step towards head and face tracking for a perceptual user interface. In this paper, we review the CamShift Algorithm and extend a default implementation to allow tracking in an arbitrary number and type of feature spaces.

In order to compute the new probability that a pixel value belongs to the target model, we weight the multidimensional histogram with a simple monotonically decreasing kernel profile prior to histogram back-projection.

We evaluate the effectiveness of this approach by comparing the results with a generic implementation of the Mean Shift algorithm in a quantized feature space of equivalent dimension.

The aim if this paper is to examine the effectiveness of the CamShift algorithm as a general-purpose object tracking approach in the case where no assumptions have been made about the target to be tracked.

Keywords: object tracking, video, mean shift

1 Introduction

The Intel Open Source Computer Vision Library (Intel Corporation, 2003) contains an implementation of the CamShift algorithm that tracks head and face movement using a one-dimensional histogram consisting of quantized channels from the HSV color space.

Since the algorithm is designed to consume the lowest number of CPU cycles possible, a single channel (hue) is considered in the color model.

This heuristic is based on the assumption that flesh color has the same value of hue. Furthermore, a bandwidth of acceptable color values is defined to allow the tracker to compute the probability that any given pixel value corresponds to flesh color.

Difficulty may arise when one wishes to use CamShift to track objects where the assumption of single hue cannot be made. In particular, the algorithm may fail to track multi-hued objects or objects where hue alone cannot allow the object to be distinguished from the background and other objects.

In a sequence of simple experiments it has been observed that an increase in the number of quantized feature spaces used to generate the target probability distribution function (PDF) during histogram back-projection can lead to improved target localization when a range of acceptable HSV values cannot be determined.

Furthermore, since the multidimensional histogram back-projection is essentially linear with the number of feature spaces, the modifications can be applied with a trivial amount of additional overhead.

2 The Mean Shift Algorithm

The Mean Shift algorithm is a robust, non-parametric technique that climbs the gradient of a probability distribution to find the mode (peak) of the distribution (Fukunaga, 1990). Mean Shift was first applied to the problem of mode seeking by Cheng (1995).

Particle filtering based on color distributions and Mean Shift is described by Isard and Blake (1998) and extended by Nummiaro *et al.* (2002).

Kernel based object tracking (including adaptive scale and background-weighted histogram extensions) is described by Comaniciu *et al.* (2003).

CamShift is primarily intended to perform efficient head and face tracking in a perceptual user interface (Bradski, 1998). It is based on an adaptation of Mean Shift that, given a probability density image, finds the mean (mode) of the distribution by iterating in the direction of maximum increase in probability density (Intel Corporation, 2001).

The primary difference between CamShift and the Mean Shift algorithm is that CamShift uses continuously adaptive probability distributions (that is, distributions that may be recomputed for each frame) while Mean Shift is based on static distributions, which are not updated unless the target experiences significant changes in shape, size or color.

Since CamShift does not maintain static distributions, spatial moments are used to iterate towards to mode of the distribution. This is in contrast to the conventional implementation of the Mean Shift algorithm where target and candidate distributions are used to iterate towards the maximum increase in density using the ratio of the current (candidate) distribution over the target.

3 Object Tracking Using CamShift

3.1 The CamShift Algorithm

The CamShift algorithm can be summarized in the following steps (Intel Corporation, 2001);

1. Set the region of interest (ROI) of the probability distribution image to the entire image.
2. Select an initial location of the Mean Shift search window. The selected location is the target distribution to be tracked.
3. Calculate a color probability distribution of the region centred at the Mean Shift search window.
4. Iterate Mean Shift algorithm to find the centroid of the probability image. Store the zeroth moment (distribution area) and centroid location.
5. For the following frame, center the search window at the mean location found in Step 4 and set the window size to a function of the zeroth moment. Go to Step 3.

3.2 Continuously Adaptive Distributions

The probability distribution image (PDF) may be determined using any method that associates a pixel value with a probability that the given pixel belongs to the target. A common method is known as Histogram Back-Projection. In order to generate the PDF, an initial histogram is computed at Step 1 of the CamShift algorithm from the initial ROI of the filtered image.

The histogram used in Bradski (1998) consists of the hue channel in HSV color space, however multidimensional histograms from any color space may be used.

The histogram is quantized into bins, which reduces the computational and space complexity and allows similar color values to be clustered together. The histogram bins are then scaled between the minimum and maximum probability image intensities using Equation 2.

3.3 Histogram Back-Projection

Histogram back-projection is a primitive operation that associates the pixel values in the image with the value of the corresponding histogram bin.

The back-projection of the target histogram with any consecutive frame generates a probability image where the value of each pixel characterizes probability that the input pixel belongs to the histogram that was used.

Given that m-bin histograms are used, we define the n image pixel locations $\{x_i\}_{i=1\ldots n}$ and the histogram $\{\hat{q}\}_{u=1\ldots m}$. We also define a function $c:\Re^2 \rightarrow \{1\ldots m\}$ that associates to the pixel at location x_i^* the histogram bin index $c(x_i^*)$. The unweighted histogram is computed as

$$\hat{q}_u = \sum_{i=1}^{n} \delta\left[c(x_i^*) - u\right] \qquad (1)$$

In all cases the histogram bin values are scaled to be within the discrete pixel range of the 2D probability distribution image using

$$\left\{ \hat{p}_u = \min\left(\frac{255}{\max(\hat{q})} \hat{q}_u, \quad 255 \right) \right\}_{u=1\ldots m} \qquad (2)$$

That is, the histogram bin values are rescaled from [0, max(q)] to the new range [0, 255], where pixels with the highest probability of being in the sample histogram will map as visible intensities in the 2D histogram back-projection image.

3.4 Mass Centre Calculation

The mean location (centroid) within the search window of the discrete probability image computed in Step 3 is found using moments (Horn, 1986; Freeman et al., 1996; Bradski, 1998). Given that $I(x, y)$ is the intensity of the discrete probability image at (x, y) within the search window.

a) Compute the zeroth moment

$$M_{00} = \sum_x \sum_y I(x, y)$$

b) Find the first moment for x and y

$$M_{10} = \sum_x \sum_y xI(x, y)$$

$$M_{01} = \sum_x \sum_y yI(x, y)$$

c) Compute the mean search window location

$$x_c = \frac{M_{10}}{M_{00}}; \; y_c = \frac{M_{01}}{M_{00}}$$

Problems with centroid computation for face tracking have been identified (Bradski, 1998; McKenna, 1999; Comaniciu *et al.*, 2003). The direct projection of the model histogram onto the new frame is known to introduce a large bias in the estimated location of the target and the measurement is known to be scale variant.

3.5 Mean Shift Convergence Criteria

The Mean Shift component of the algorithm is implemented by continually recomputing new values of (x_c, y_c) for the window position computed in the previous frame until there is no significant shift in position.

The maximum number of Mean Shift iterations is usually taken to be 10-20 iterations.

Since sub-pixel accuracy cannot be visually observed, a minimum shift of one pixel in either of the x and y directions is selected as the convergence criteria.

Furthermore, the algorithm must terminate in the case where M_{00} is zero, which corresponds to a window consisting entirely of zero intensity.

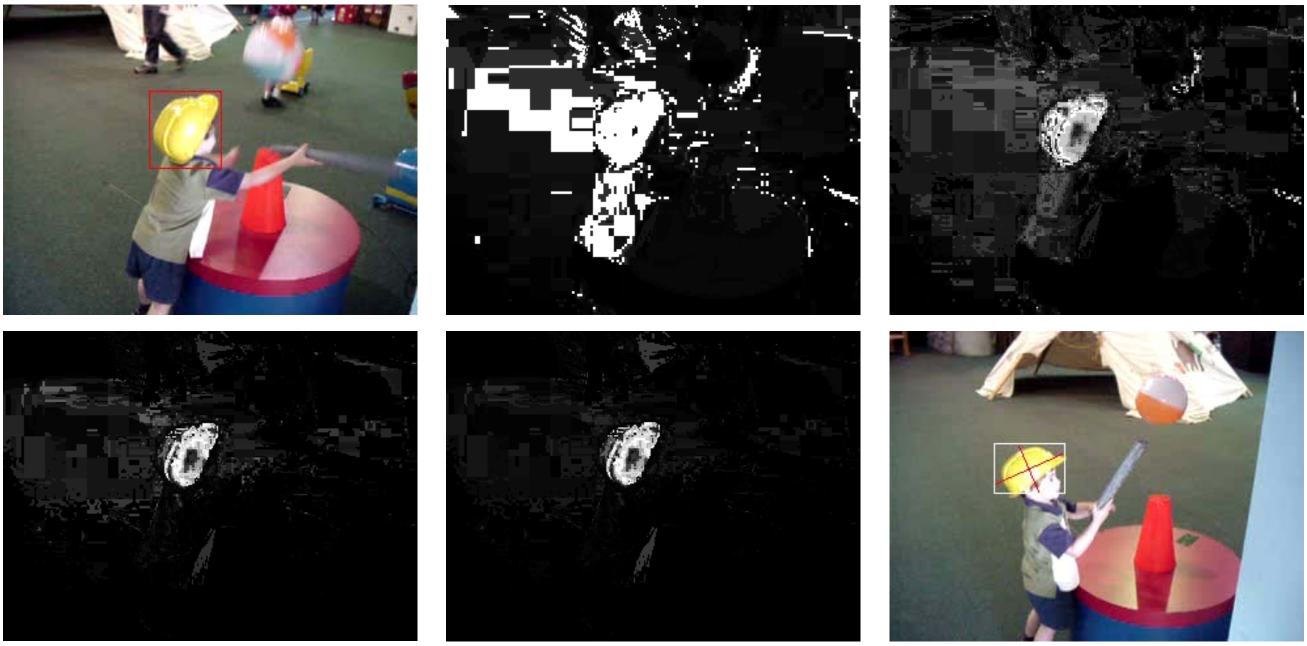


Figure 1: (a) Original image and ROI (b) 1-D projection (H) (c) 2-D projection (HS) (d) 3-D projection (HSV) (e) 3-D projection of weighted histogram (e) A tracked frame

3.6 Target Model for Localization

3.6.1 Weighted Histogram

When the initial selected region contains some pixels from outside the object (background pixels), our 2D probability distribution image will be influenced by their frequency in the histogram back-projection. In order to assign higher weighting to pixels nearer to the region center, a weighted histogram may be used to compute the target histogram (Comaniciu *et al*., 1996)

$$\hat{q}_u = \sum_{i=1}^{n} k\left(\left\|x_i^*\right\|^2\right)\delta\left[c(x_i^*)-u\right] \tag{3}$$

Where the resulting histogram is scaled using Equation 2 for the discrete quantities we are using and $k(x)$ is any convex, monotonically decreasing kernel profile that assigns higher weight to pixels near the centre of the normalized search window. The most simple kernel profile used to generate the background-weighted histogram in our experiment is shown in Equation 4

$$k(r) = \begin{cases} 1-r & r \le 1 \\ 0 & otherwise \end{cases} \tag{4}$$

It is worth noting that since the Mean Shift iterations are based on moment calculations and do not require an estimate of the probability density gradient, the selected kernel profile does not need to be differentiable or have a constant derivative (kernels with Epanechnikov profile, for instance).

3.6.2 Ratio Histogram

The weighted histogram selected in 3.6.1 is not sufficient to localise the target when histogram back-projection is used to generate the 2D probability distribution image. In a sequence of experiments it has been observed that if the target histogram contains a significant number of features that belong to the background image or neighbouring objects, target localisation and scale cannot be accurately determined.

A ratio histogram can help to solve the background problem by assigning color features that belong to the background with lower weights (Swain and Ballard, 1991; Comaniciu *et al*., 1996). In our experiment, we compute a histogram for a region outside the normalized target location using a kernel a with the following profile

$$k(r) = \begin{cases} ar & 1 < r \le h \\ 0 & otherwise \end{cases} \tag{5}$$

Where *a* is a scaling factor and *h* is the bandwidth of the new search window. A background region that is three times as large as the target region (*h* = 4) was used in the experiment. A histogram $\{\hat{O}\}_{u=1...m}$ is computed using Equation 3 with a bandwidth *h* and then weighted using Equation 6 where $\hat{O}^*$ is the smallest nonzero entry

$$\left\{\hat{w}_u = \min\left(\frac{\hat{O}^*}{\hat{O}_u},\ 1\right)\right\}_{u=1...m} \tag{6}$$

The background-weighted histogram used in our experiment is therefore given as

$$\hat{q}_u = \hat{w}_u \sum_{i=1}^{n} k\left(\left\|x_i^*\right\|^2\right)\delta\left[c(x_i^*)-u\right] \tag{7}$$

3.7 Orientation and Scale Calculation

The use of moments to determine the scale and orientation of a distribution in robot and computer vision is described in Horn (1986) and has been used for vision in computer games (Freeman *et al*., 1996) and for head and face orientation and tracking (Bradski, 1998).

The orientation (θ) of the major axis and the scale of the distribution are determined by finding an equivalent rectangle that has the same moments as those measured from the 2D probability distribution image (c.f. Horn, 1986). Defining the first and second moments for x and y

$$M_{20} = \sum_x \sum_y x^2 I(x, y)$$

$$M_{02} = \sum_x \sum_y y^2 I(x, y)$$

$$M_{11} = \sum_x \sum_y xyI(x, y)$$

The first two eigenvalues (the length and width of the probability distribution) are calculated in closed form as follows. From the intermediate variables a, b and c

$$a = \frac{M_{20}}{M_{00}} - x_c^2$$

$$b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right)$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2$$

We find the orientation of the equivalent rectangle

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{b}{a-c}\right)$$

The distances l_1 and l_2 from the distribution centroid (the dimensions of the equivalent rectangle) are given by,

$$l_1 = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}$$

$$l_2 = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}$$

Where the extracted parameters are independent of the overall image intensity.

4 Results

Initial regions for the video sequences in Appendix A and B were manually selected by drawing a rectangle around the target area of interest. Three-dimensional histograms were selected in the HSV colour space with bins of size 32x32x16. The target histograms were initialised using Equation 7 and scaled to image intensity range using Equation 2. A region three times larger (h = 4) than the target was used to compute the weights with a = 1.

The synthetic video sequence shown in Appendix A was used to demonstrate object tracking through occlusion as well as the scale and orientation estimation. Appendix B demonstrates person-tracking using a multi-colour shirt region giving scale and orientation estimates for the torso region. If the histogram values used in Figure 1 are included, then the resultant tracking will include the head and face region. When weighted histograms are not used, the CamShift based tracker fails due to influence of background pixels included in the target region. We note that the assumption of single hue used in the CamShift face tracker does not apply to the tracking presented in this paper.

Appendix C demonstrates target region selection and the resulting target localization. Note that the background-weighted histogram allowed the snow and partially selected sleigh colors to be eliminated from the target histogram. These results suggest that CamShift can be used for general-purpose object tracking using a background- weighted histogram and arbitrary quantized color features of the target, however the tracking performance was inferior to the Mean Shift tracker implementation in equivalent quantized feature spaces.

5 References

Intel Corporation (2001): Open Source Computer Vision Library Reference Manual, 123456-001

Comaniciu, D. and Meer, P. (1996): Robust Analysis of Feature Spaces: Color Image Segmentation. *CVPR'97, pp. 750-755.and Gesture Recognition*, pp. 176-181, 1996.

Fukunaga, K. (1990): Introduction to Statistical Pattern Recognition, 2nd Edition. Academic Press, New York, 1990.

Cheng, Y. (1995): Mean shift, mode seeking, and clustering, *IEEE Trans. Pattern Anal. Machine Intell.*, 17:790-799, 1995.

Comaniciu, D., Ramesh, V. and Meer, P: (2003): Kernel-Based Object Tracking, *IEEE Trans. Pattern Analysis Machine Intell.*, Vol. 25, No. 5, 564-575, 2003

Nummiaro, K., Koller-Meier, E. and Gool, L. V. (2002): A Color-Based Particle Filter *Proceedings of the 1st International Workshop on Generative-Model-Based Vision,in conjunction with ECCV02*, Denmark, pp. 53-60, Jun 2002

Isard, M. and Blake, A. (1998): Condensation -- conditional density propogation for visual tracking. *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5--28, 1998.

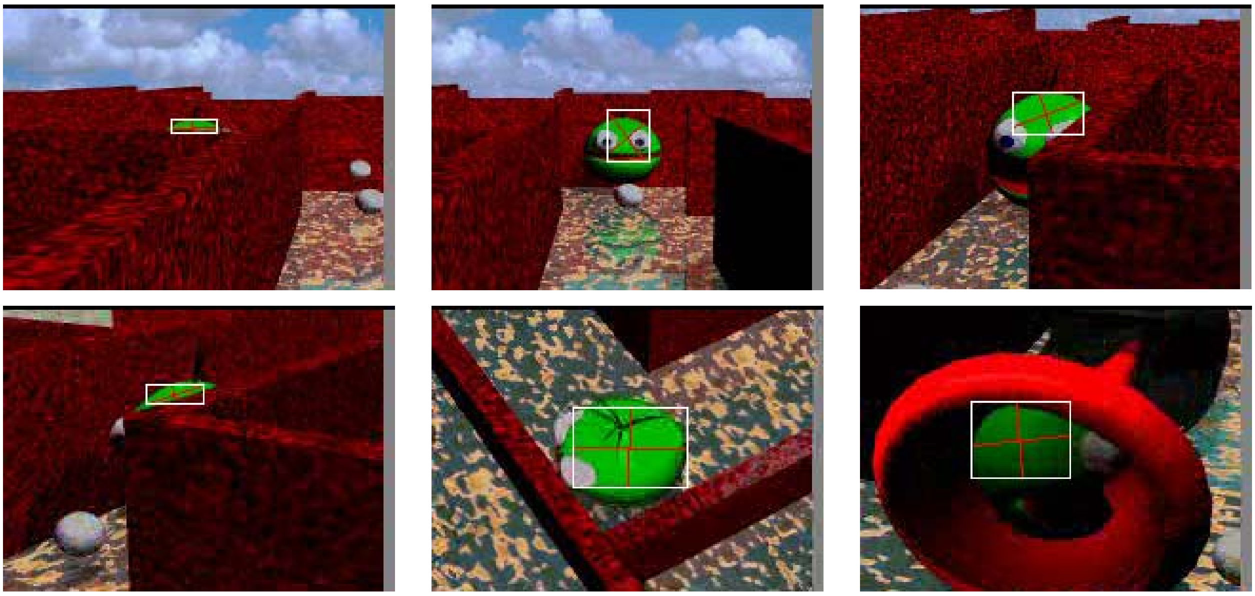
Bradski , G. R. (1998): Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal, 2nd Quarter*, 1998.

Freeman, W. T., Tanaka, K., Ohta, J. and Kyuma, K. (1996): Computer Vision for Computer Games. *Int. Conf. On Automatic Face and Gesture Recognition*, pp.100-105, 1996.

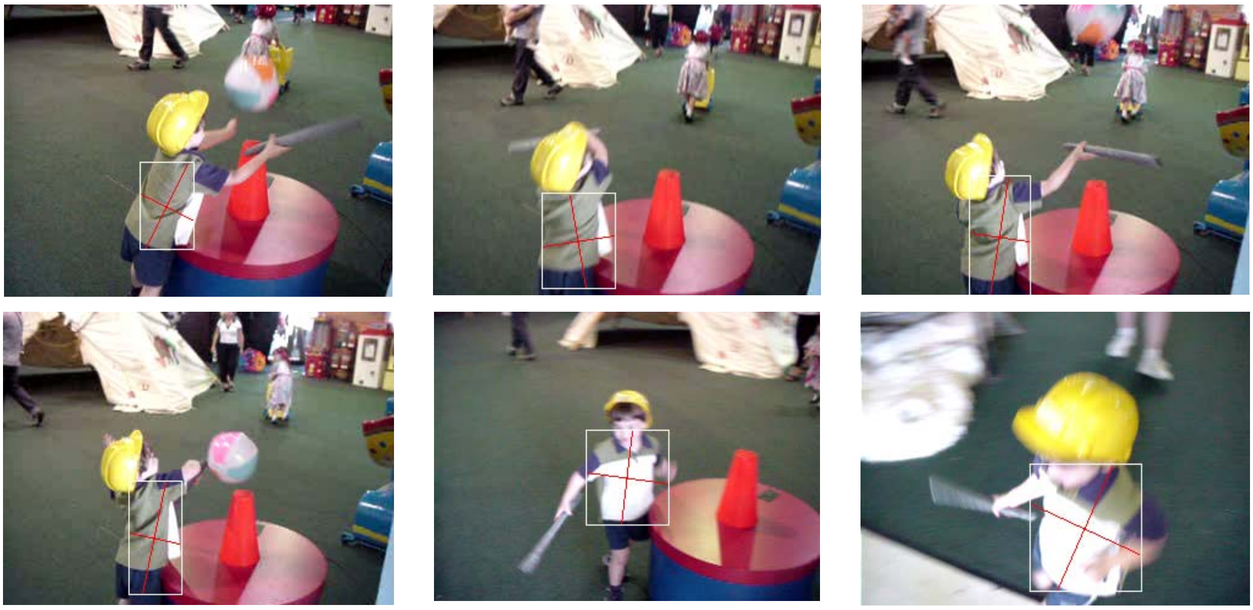
Horn, B. K. P. (1986): Robot vision. MIT Press, 1986.

McKenna, S., Raja, Y. and Gong, S. (1999): Tracking Colour Objects Using Adaptive Mixture Models. *Image and Vision Computing Journal*, vol. 17, pp. 223–229, 1999.

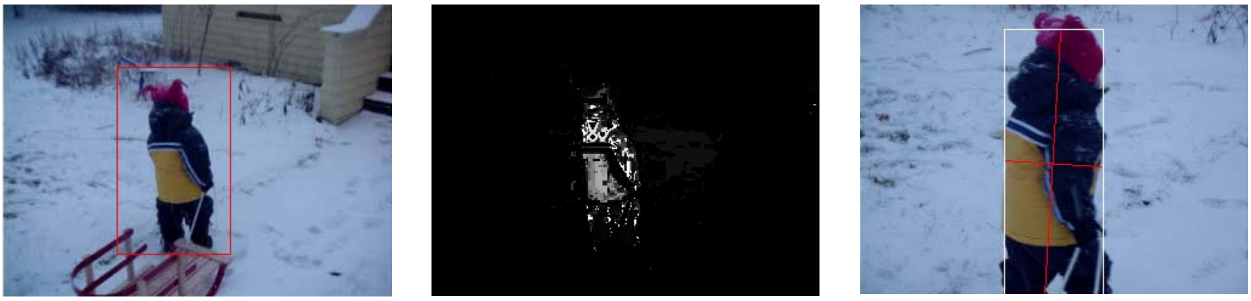
Appendix



Appendix A: Tracking Synthetic Object Through Occlusions



Appendix B: Tracking Shirt Color Region



Appendix C: (a) Target Region (b) Probability Distribution Image (c) A Tracked Frame