Image Tracking using Mean Shift Algorithm

Haeyoon Chang
Computer Science Department
Portland State University
Portland, OR, USA
haeyoon@pdx.edu

***Abstract*—Mean Shift tracking algorithm is one of the basic feature tracking algorithms using an iterative method. This technique searches for the most probable target position in a subsequent frame in an iterative manner, based on the color histogram of the initial target object. Object tracking using Mean Shift tracking algorithm is suitable for tracking an object in a video where movement from one frame to the next frame is relatively small. Each step of this algorithm was demonstrated in two different videos. This experiment also showed that fine tuning of the number of iterations and convergence criteria affected how accurately the algorithm tracked the target object.**

***Index Terms*—Mean Shift, Object Tracking**

I. Introduction

There were four key problems discussed during the class: (1) feature detection, (2) feature description, (3) feature matching, and (4) feature tracking. The goal of this project was to tackle the last problem and Mean Shift tracking algorithm solves feature tracking problem quite well. Object tracking (or feature tracking) is closely related to object detection where Harris corner detection and scale invariant feature transform (SIFT) [1] were used. Once the object of interest was identified whether manually or using one of the object detection techniques, Mean Shift tracking algorithm was used to match the same object in a following images. Application of the object tracking algorithms such as Mean Shift tracking algorithm is endless from surveillance camera, robotics, and autonomous driving as time matters in a real world.

II. Methodology

There are two main approaches to find feature points and their correspondence.

First approach is to detect features in one image, then match the features in the other image based on their local descriptors. Towards the end of SIFT paper, this approach was used to find the matching key point in the other image. The algorithm picks the key point with the minimum Euclidean distance for the invariant descriptor vector. Feature matching is generally more useful when two images differ by large amount of motion such as panoramic images.

The second approach is to find the initial features in the next image using a local search technique. Compared to the first one, this approach is suitable when images are taken from nearby viewpoints or in rapid succession such as video sequences. Mean Shift algorithm belongs to this approach.

A. Mean shift tracking algorithm

Mean Shift tracking algorithm is an iterative method for locating the maxima of a density function given discrete data sampled from that function [2]. The mathematical equations and steps are directly from Comaniciu [4]. It starts from the initial point $\mathbf{x}$. The multivariate kernel density estimate with kernel $K(\mathbf{x})$ and window radius (band-width) h are computed in the point $\mathbf{x}$ and are defined as:

$$f(\mathbf{x}) = \frac{1}{nh^2}\sum_{i=1}^{N} K\left(\frac{\mathbf{x}-\mathbf{x_i}}{h}\right) \tag{1}$$

Normal kernel function is commonly used for $K(\mathbf{x})$ and it determines the weight of nearby points.

Let's define profile notation as $K(\mathbf{x}) = k(||x||^2)$.

$$f(\mathbf{x}) = \frac{1}{nh^2}\sum_{i=1}^{N} K\left(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2\right) \tag{2}$$

Let's define $g(x) = -k'(x)$. A kernel G is defined as

$$G(x) = Cg(||x||^2) \tag{3}$$

Then, we can re-write the gradient of the density function as

$$\begin{aligned}\nabla f(\mathbf{x}) &= \frac{2}{nh^4}\sum_{i=1}^{N}(\mathbf{x}-\mathbf{x_i})k'\left(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2\right)\\ &= \frac{2}{nh^4}\sum_{i=1}^{N}(\mathbf{x}-\mathbf{x_i})g\left(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2\right)\\ &= \frac{2}{nh^4}\left[\sum_{i=1}^{N} g\left(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2\right)\right]\times\\ &\left[\frac{\sum_{i=1}^{n}\mathbf{x_i}g(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2)}{\sum_{i=1}^{N}g(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2)}-\mathbf{x}\right]\end{aligned} \tag{4}$$

The last term of the bracket contains the mean shift vector and defined as:

$$M_{h,G(x)} \equiv \left[\frac{\sum_{i=1}^{n}\mathbf{x_i}g(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2)}{\sum_{i=1}^{N}g(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2)}-\mathbf{x}\right] \tag{5}$$

Given the mean shift vector and the density estimate at $\mathbf{x}$, it becomes

$$\nabla f(\mathbf{x}) = f(x)\frac{2/C}{h^2}M_{h,G(x)} \tag{6}$$

$$M_{h,G(x)} = \frac{h^2}{2/C}\frac{\nabla f(\mathbf{x})}{f(\mathbf{x})} \tag{7}$$

The final expression showed that mean shift with kernel G moves the center of target window along the direction of the gradient of density estimate $f(\mathbf{x})$ with kernel K [4]. The center of kernel recursively moves by $M_{h,G(x)}$ (mean shift). In each step, the dissimilarity in color histogram between the target and the target candidates in the following frame is calculated using Bhattacharyya coefficient [4]. The iteration stops earlier of when the number of maximum iteration is reached or when the center of the kernel starts to converge.

B. Mean shift algorithm pseudo-code

Initially, the algorithm was provided with the location and color histogram of the target. Then, the algorithm conducted periodic analysis of each object to account for possible updates of target models due to changes in color [4].

Algorithm 1: Mean shift tracking

Result: Tracks the target object throughout the video
`// color representation`
a. the algorithm is given a target to track
b. initialize track-window size around target
c. initialize the position of the track-window
d. calculate the color histogram of the track-window
while *frames left* **do**
  e. read in new current frame
  while *True* **do**
    `// search for new target location in current frame that minimizes the dissimilarity`
    f. initialize location of the target with target location in previous frame
    g. derive the the weights based on the color probability of the target candidate at new location in the current frame
    h. derive new location of the target using mean shift vector
    i. update the color probability and evaluate the dissimilarity metric ρ
    while ρ *new loc* $<$ ρ *prev loc* **do**
      `// large` ρ `means good color match`
      j. new location = $\frac{1}{2}$(prev location + new location)
    end
    if $\|newloc - prevloc\| < \epsilon$ **then**
      `// loc converges, stopping criteria`
      k. Stop
    end
    l. assign new location as the target location
  end
end

III. Experiment Results

Mean Shift tracking algorithm is applied to two different video clips. Before running the algorithm, the location of target object was identified and the program calculated the color probability of the target.

The first video had 263 frames with frame size (480, 640). The time length of the video was 14 seconds, thus 35 frames were taken every second.

Experiment with toy train sequence in Fig. 1 showed that the algorithm worked well under the relatively controlled environment. The background for this experiment was plain without much of color mix while the target object was red, easily distinguishable from the background or neighboring objects (i.e. hands or train track). The last image of Fig. 1 showed that the track window stayed at the last position the train was detected since the localized search failed to find any other objects around with similar color histogram.

The second video had 492 frames with frame size (1080, 1920). The time length of the video was 17 seconds, thus 30 frames were taken every second.

This experiment is done in more natural setting compared to the first experiment. The target object color was yellow cone. The frames 40, 120, and 200 showed that the algorithm tracked the yellow cone, but the frames 320 and 400 showed that it tracked the boy's hand wrapping around the cone and sand ice cream instead of the yellow cone. See Fig. 2. The number of maximum iteration was 10 and convergence criteria was tight with epsilon 1. In the third experiment, the number of iterations and convergence criteria were fine tuned to see if it improved tracking precision.

Fig. 3 showed the effect of adjusting the two aforementioned parameters, the number of iterations and convergence criteria. The first two images had tighter convergence criteria (smaller epsilon) failed to track the yellow cone even with more iterations. On the other hand, when I set lenient convergence criteria (bigger epsilon), the algorithm stops before finding the center-mass and settles at the patch with color histogram that seemed good enough, leading to lower precision. In this specific video, the algorithm tracked yellow cone better when the epsilon was 10 instead of 1 or 20.

IV. Discussion

Overall Mean Shift tracking algorithm tracked the object pretty well. However, Mean Shift tracking algorithm poses some other limitations due to its localized search. It could not detect the object if it was coming from unexpected side. For example, when the ice cream cone disappeared from the frame and reentered from the top side of the frame, it did not detect well because the algorithm would have already found something similar in color around the region the ice cream cone was detected last time. Also, there was a scaling issue with Mean Shift tracking algorithm, but this problem can be

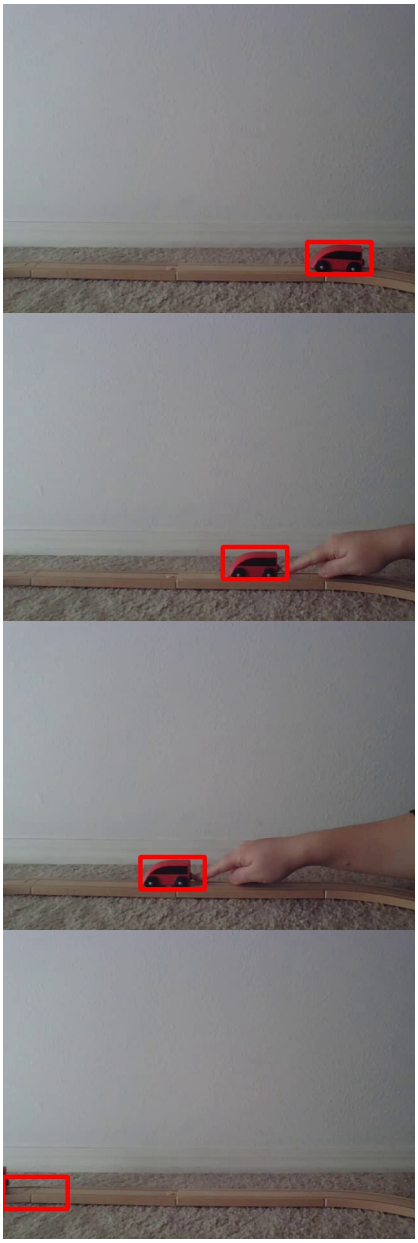

Fig. 1. Toy train sequence: Tracking a small red train with initial track-window 50 x 100. The frames 40, 120, 160, and 200 were shown. The termination criteria for Mean Shift was set up, either 10 iteration or move by at least 1pt

Fig. 2. Sand ice cream sequence: Tracking a small yellow cone with initial track-window 200 x 200. The frames 40, 120, 160, 200, 320, and 400 are shown. The termination criteria for Mean Shift was set up, either 10 iterations or 1 epsilon.

solved with Continuously Adaptive Mean Shift (CAMShift) [5] algorithm.

Despite the small number of experiments, implementation of Mean Shift algorithm on real video showed that the algorithm, like many other techniques, is not one-size-fit-all solution and requires fine tuning of parameters depending on cases. For example, the movement of yellow cone in this experiment was fairly slow, and when the convergence criteria was too tight, the algorithm overshoot and the track window ended up locating the hand or sand ice cream. It was hard for the track window to get out of the sand ice cream or the boy's hand because localized search does not search outside the target window. When the convergence criteria was too lenient, the algorithm would eventually fail to keep track of the object's movement because the track window's momentum would not keep up with the target movement. In the end, good solution

Fig. 3. Sand ice cream sequence: Tracking a small yellow cone with initial track-window 200 x 200. The frame 360 is shown. The termination criteria (iteration, epsilon) is (10, 1) for upper-left corner, (1000, 1) for upper-right corner, (1000, 10) for lower-left corner, and (1000, 20) for lower-right corner.

to the object tracking problem would come from combination of knowledge in camera specifications (e.g. frame per second) and the object pose estimation (e.g. calculating the velocity of the object using Kalman Filter), in addition to tracking algorithms like Mean Shift tracking algorithm.

REFERENCES

[1] Lowe, David G. "Distinctive image features from scale-invariant key points," in International Journal of Computer Vision, vol. 60, 2004, pp. 91–110

[2] Cheng, Yizong, "Mean shift, mode seeking, and clustering," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, number 1, 1995.

[3] K. Fukunaga and L. Hostetler, "The estimation of gradient of a density function, with application in pattern recognition," in IEEE Transactions on Information Theory, vol. 21, number 1, 1975, pp. 32–40

[4] D. Comaniciu and V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in Proceedings IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 142–149

[5] John G. Allen, Richard Y.D. Xu, Jesse S. Jin, "Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces," in VIP '05: Proceedings of the Pan-Sydney area workshop on Visual information processing, June 2004, pp.3-7