Image Tracking using Mean Shift Algorithm

Haeyoon Chang

***Abstract*—Mean Shift tracking algorithm is one of the basic feature tracking algorithms using iterative method. The technique allows to find the most probable target position[2] in a frame without computing the complete function. Object tracking using Mean Shift tracking algorithm is demonstrated on several image sequences with fine tuning of the number of iteration and allowable error rate to improve tracking accuracy.**

I. Introduction

The goal of the project was to understand Mean Shift tracking algorithm and to implement it in some video clips. Object tracking is closely related to object detection where Harris corner detection and scale invariant feature transform techniques are used. Once the object of interest is identified in one image using object detection techniques, tracking algorithm comes in to see if the same feature exists in the the another image or in a sequence of images (video). The main purpose of object tracking algorithm is to connect target objects in successive video frames and its applications are prevalent in the area of surveillance camera and autonomous driving. MeanShift is one of basic tracking algorithms, and thus I decided to explore Mean Shift tracking algorithm and implement it in my own video clips.

II. Methodology

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

First approach is to detect features in the image, then match features based on their local descriptors. Towards the end of scale-invariant feature transform (SIFT) paper, we used this approach to find the matching keypoint in the other image. The algorithm picks the keypoint with the minimum Euclidean distance for the invariant descriptor vector. This approach is more suitable for the large amount of motion or appearance change such as panoramic images.

The second approach is to find features in one image that can be accurately tracked using a local search technique, such as correlation and least square. 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. MeanShift 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[1].

It starts from the initial point $\mathbf{x}$. The multivariate kernel density estimate with kernel $K(\mathbf{x})$ and window radius (bandwidth) h, computed in the point $\mathbf{x}$ is defined as:

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

where $K(\mathbf{x})$ is commonly used kernel multivariate normal kernel function 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

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

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

$$M_{h,G(x)} \equiv \left[\frac{\sum_{i=1}^{n}\mathbf{x_i}g(\left(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2\right)}{\sum_{i=1}^{N}g(\left(\left\|\frac{\mathbf{x}-\mathbf{x_i}}{h}\right\|^2\right)} - \mathbf{x}\right] \tag{4}$$

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

$$\nabla f(\mathbf{x}) = f(x)\frac{2/C}{h^2}M_{h,G(x)} \tag{5}$$

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

The final expression shows that the sample mean shift vector obtained with Kernel G is an estimate of the normalized density gradient obtained with kernel K[2].

A mean shift procedure is defined recursively by computing the mean shift vector $M_{h,G(x)}$ and translating the center of kernel G by $M_{h,G(x)}$. The dissimilarity between the target and the target candidates in the following frame is expressed by a metric derived from Bhattacharyya coefficient[2]. The iteration stops when the dissimilarity metrics become smaller than predetermined level.

B. MeanShift algorithm pseudocode

The algorithm assumes that the initial frame of the target is detected and given. Then, the algorithm conducts periodic analysis of each object to account for possible updates of target models due to changes in color[2, 4, 3].

Algorithm 1: Mean shift tracking

```
Result: Write here the result
// 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 descriptor 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 = 1/2(prev location + new
             location);
        end
        if ||newloc − prevloc|| < ε then
            // loc converges, stopping
             criteria ;
            k. Stop ;
        end
        l. assign new location as the target location in
         prev frame ;
    end
end
```

III. Experiment Results

Mean Shift tracking algorithm is applied to three different video clips. Before running the algorithm, the location of target object is identified so that the program can calculate 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 showed that the algorithm works well under the relatively controlled environment. The background for this experiment was plain without many color mix while the target object was red, easily distinguishable from background or neighboring objects (i.e. hands or train track). See Figure 1.

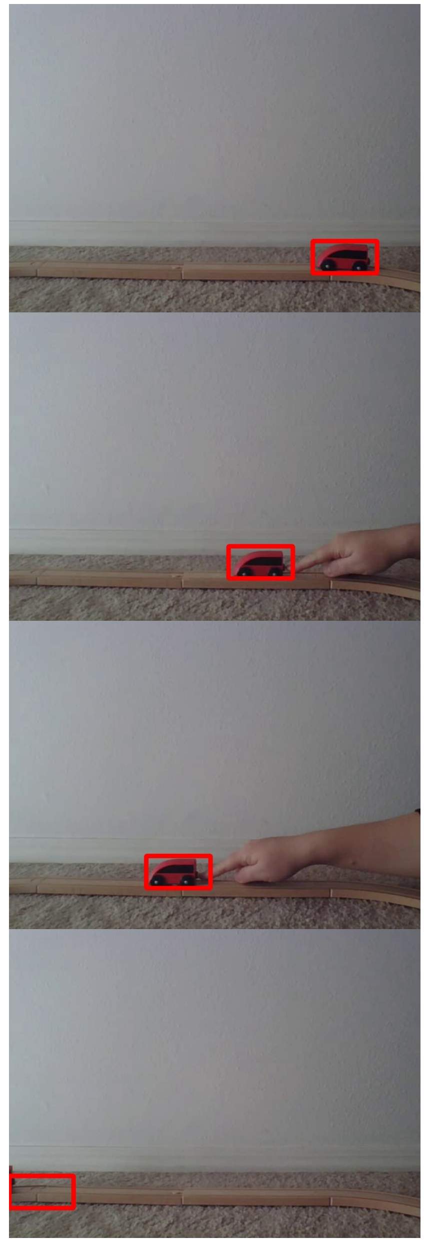

Fig. 1. Toy train sequence: Tracking a small red train with initial track-window 50 x 100. The frames 40, 120, 160, and 200 are shown. The termination criteria for Mean Shift was set up, either 10 iteration or move by at least 1pt

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 and the target object color was yellow.

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.

The frames 40, 120, and 200 show that the algorithm tracked the yellow cone whereas the frames 120 and 200 show that it tracked a boy's hand wrapping around the cone and sand ice cream instead of the cone, respectively. They are both similar in color. See Figure 2.

The third experiment was to adjust two parameters, the number of iterations and size of error the algorithm tolerates. Depending on the parameters, the algorithm performs better on the image.

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.

Figure 3 shows that the first two images with less tolerance to the error failed to track the ice cream cone even with more iterations, while the algorithm successfully tracked the ice cream cone with 1000 iteration and larger error tolerance. However, when the error tolerance was too high (e.g. the lower right corner image) the algorithm stops before finding the target or settles at the feature that seems good enough, leading to lower accuracy.

IV. Discussion

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 suitable depending on the case. For the experiments done for this project demonstrated that Mean Shift algorithm performs better when the movement from one frame to the other was small, the target has unique color probability, and find tuning of parameters such as iteration and error tolerance level.

Mean Shift tracking algorithm poses some limitations. It cannot detect the object if it is coming from unexpected side. For example, if the ice cream cone disappeared from the frame and reentered from the top side of the frame, it may not be able to detect well because the algorithm would have already found something similar in color around the last time the ice cream cone is detected. Also, there is a scaling issue with Mean Shift tracking algorithm, but this problem can be solved with Continuously Adaptive Mean Shift (CAMShift) algorithm.

References

[1] Yizong Cheng. "Mean Shift, Mode Seeking, and Clustering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.1 (1995).

[2] D. Comaniciu, V. Ramesh, and P. Meer. "Real-time tracking of non-rigid objects using mean shift". In: 2 (2000), 142–149 vol.2. DOI: 10.1109/CVPR.2000.854761.

[3] Leow Wee Kheng. "Lecture note, Mean Shift Tracking". In: CS4243 Computer Vision and Pattern Recognition ().

[4] Richard Szeliski. "Computer Vision: Algorithms and Applications". In: Chapter 5, Segmentation (2010), pp. 289–295.