Abstract
Mean Shift Tracking, one of the Kernel Tracking approach [1, 5, 6], is used to track by computing the motion of the kernel in consecutive image frames. The tasks of detecting the object and tracking are pretty much easier to understand. The object region, which contains the object we are interested, is estimated by iteratively updating object location and region information obtained from previous frames.

In this report, mean shift algorithm is studied and implemented to tracking of objects. The main problem in using mean shift to track objects is estimating the location of the objects. And I use probability density function (PDF) to estimate the new location of objects. Then I use a color video to perform a color tracking. The results of experiment show the approach has good performance of object tracking. But still, there are some problems, such as tracking through scale space.

1. Introduction
Mean Shift[1, 5], a simple iterative procedure that shifts each data point to the average of data points in its neighborhood is recently widely used in object tracking. Mean shift, which was developed by Fukunaga and Hostetler [2], is an efficient approach to tracking objects whose appearance is defined by histograms, and does not limited to only grayscale image.

To track the objects, we compute likelihood maps where the value at a pixel is proportional to the likelihood that the pixel comes from the object we are tracking. Computation of likelihood can be based on color, texture, shape (boundary), predicted location. So far, my main approach to achieve the mean shift object tracking is by finding modes in a set of data samples, manifesting an underlying probability density function. The data points are sampled from an underlying PDF. Data point density implies PDF value.

Many trackers just give a grayscale video or images by tracking objects. Here, I will use mean shift on color modes. My approach is changing color modes to HSV modes and represents it with a histogram. Then, use mean shift to find region that has most similar distribution of colors.

2. Mean Shift algorithm
To find the new location of the objects we are interested in, we need to find a vector which can indicate the direction of the movement of objects. This vector is called mean shift vector [1, 3].

First, we allocate the object and get data points and approximate location of the mean of this data. Then, we need to estimate the exact location of the mean of the data by determining the shift vector from initial mean. Mean shift vector example:\[ M_k(y) = \frac{1}{n_k} \sum_{i=1}^{n_k} (X_i - y_0) - y_0 \] (1)

(1) Show mean shift vector always points towards the direction of the maximum increase in the density. There is a modified mean shift:
\[ M_k(y_0) = \frac{\sum_{i=1}^{n_k} w(y_0) x_i}{\sum_{i=1}^{n_k} w(y_0)} - y_0 \] (2)

Where:
- \( n_k \): number of points in the kernel
- \( y_0 \): initial mean location
- \( x_i \): data points
- \( h \): kernel radius

Form (1) (2), we can see, mean shift vector has the direction of the gradient of the density estimate. It is computed iteratively for obtaining the maximum density in the local neighborhood. Automatic convergence speed – the mean shift vector size depends on the gradient itself. Near maxima, the steps are small and refined.

---

1 Data comes from PPT of mean shift object tracking given by Taoran Lu, ECE at University of Florida.
More knowledge about mean shift algorithm can be found in paper [6].

3. Probability density function

Given the AVI video sequence of a car,

Figure 1(a) shows the histogram of the selected region in the first frame. We can find the pixels of the car are definitely around 50. After some iterative times, the next location is found. Figure 1(b) and (c) shows the iterative times in process of finding the car from the fifth frame to 10th frame. We can see the histograms of the selected region are all near 50. It means that the car is always in the selected region. Then we track the car successfully. (Different number of pixels in different figures is caused by the slightly different size of selected regions or the change of the object.)

There is a problem. When you choose a region, which contains the object, you have to choose an exact region to represent the histogram of the object. Otherwise, it may make a wrong track.

Let me use another video sequence to illustrate this problem.

Figure 2(a). Soccer match (original)\(^2\)

When I choose a region to track the player, it may loss the player sometimes, if I use a bit larger window size.

\(^2\) Video comes from [http://staff.science.uva.nl/~vnedovic/MMIR2004/meanShift.html](http://staff.science.uva.nl/~vnedovic/MMIR2004/meanShift.html). The green ellipse is the tracker which the author did [4].
From Figure 2(b) (c), we can see that the track region cannot track the player exactly from the second frame. The region lost the player in the 10\textsuperscript{th} frame. Because I choose a big region containing the player and background, so the algorithm cannot tell which one is the object I want to track.

An easiest way to enhance the tracking is that we choose an exactly closed region to track the object which we are interested in.

Figure 2(d) show the tracker traces the player in 10\textsuperscript{th} frame.

While, there is another problem, which is the scaling. If we choose a region contains the most of the information that comes from the object, we have to restrict the window size. It is good for tracking, but is not good for human to see. Sometimes, we want to see the whole object, not just one part of it. This problem will be discussed in section 5.

4. Color models

Usually, we use grayscale image or video to do object tracking. Because we just need the histogram to estimate and track the object, we are not care about the color. So in my project, I first translate all the images into HSV images, no matter what the images are RGB or YIQ. Then, I do the processing through the HSV images. Finally, I merge all the images into a video sequence. Commonly, it is enough. But, sometime we don’t want to change the color of the image, or we will loss some information about the object. So it is necessary to product a tracking algorithm which maintains the color information.

There are two commonly approaches to use mean shift for tracking in color images

i). Create a color “likelihood” image, with pixels weighted by similarity to the desired color (best for unicolored objects).

ii). Represent color distribution with a histogram. Use mean shift to find region that has most similar distribution of colors.

To achieving color image tracking, I change my MATLAB code a little. My approach is keeping the PDF algorithm while change the output image into 3 images, which are R, G, B related images. Then, I merge these 3 images into one color image to create a video sequence. Here are some results from the color video, soccer match.
Figure 3(a). Soccer match, 5th frame, color

Figure 3(b). Soccer match, 15th frame, color

From 3(a) (b), we can see the original grayscale image becomes a color image. I use the blue rectangle to track the player. Of course, you can use red ellipse to track the player. It depends on you.

Although I achieve the object tracking in color model, this method is not exactly a color-based object tracking approach. Its main idea is computed iteratively for obtaining the maximum density in the local neighborhood using histogram. The ways to achieve color tracking are mentioned above. It is one of my future attentions.

5. Related work

As I mentioned before, if I select a big region to track an object, it will loss the object sometimes. The easiest way is to choose a restrict region which contains mainly the object. The other way to achieve a successful tracking is to decrease the window size (region firstly use a larger one in order to search the object in the frame), while increase the count of iterations to achieve convergence of the window center in one image. I achieve the change in the code. It performs well but needs much more time to realize convergence.

Now, let me serve the problem given in section 3-scaling. As I mentioned, we have to restrict the window size to achieve tracking. But sometimes, we need a bigger window which contains the object, so that we can see much more clearly. In figure 2(d), the window covers some part of the player; we can not see him very clearly. To enlarge the window size, I reset the window size as a square.

Figure 4(a). Soccer match, 5th frame, color, enlarge

Figure 4(b). Car, 2nd frame

Figure 4(a) shows the enlarged window size. It gives clear information about the player. No covered part occurs.

It seems that this problem has been solved. But we haven’t consider the object itself yet. If the object itself change the size, whether the algorithm can also work well. I use car.avi to check it.
From figure 4 (b) (c) (d), we can see the algorithm can slightly scale the window size, while the car is becoming bigger. But the true is that, the window can not wholly contain the car. We need the window change when the car size changed. It is difficult and is one of the important things I take attention to in the future. I think the best way to solve the problem is image segmentation and recognition. But it is another topic and there will be other problems.

6. Conclusions

This report primarily studies the mean shift algorithm, solves some problems when doing experiment on tracking based on mean shift. At first, this report review the mean shift algorithm, then gives the idea of tracking the object by computing the PDF, finding the centre of the new window by some iterative times. There are some experimental results provided. However some problems are presented and discussed. I consider how to select a window, and how to change the size of the window. Also, I realize the color video object tracking, though it does not depend on distribution of colors. But it looks good. Some other problems I haven’t solved yet. Such as, the code should load whichever format of images (CIF, QCIF) or videos (MPG, WMA) it can; scale the window size adaptively to catch the object when the object itself change the size. Generally, this kernel object tracking using mean shift algorithm in my project has proven to perform well.

7. References


Acknowledgements

Thanks for Taoran Lu’ advice and help.