VIDEO OBJECT TRACKING USING TWO METHOD: MEAN SHIFT AND LUCAS KANADE

Le Pham Tuyen (ID: 2014311082)

December 12, 2014

Final project of Computer Vision Course

Lecture by Professor Seungkyu Lee

Contents

Chapter 1

Introduction

In this chapter, I want to introduce something about two popular tracking methods: Mean Shift and Lucas Kanade. Nowadays, many researches about object tracking still based on two these methods. They are also background for my implementation in this project.

1.1 Mean Shift Method

Mean Shift is a analysis technique for locating the maxima of a density function. Application domains include cluster analysis in computer vision and image processing. In computer vision, video object tracking is a topic applied Mean Shift and the first proposal is from [3]. Particularly, object given from a frame image is analyzed as a histogram which is distribution of colors in that object template. After that, in the next frames of video (sequence of images), mean shift method is used to find the next center of tracked object. The next center of tracked object have the same/similar distribution of color as object template. A advantage of Mean Shift algorithm is that it can track a non-rigid objects, (like a walking person).

There have some improvements based on Mean Shift. Mean Shift from paper [4] involves image template with a kernel (Gaussian, Triangle,...) to make template more smoothly. Tracking a scaled object (object moves from near to far) is proposed in paper [2]. In order to reduce effect of background around object template, paper from [4] has introduced Background-Weighted Histogram (BWH) and so on.

In my project, mean shift is implemented as an original version of Mean Shift with histogram of image template. However, it also use BWH in [4] to make result better by reducing effect of object's background.

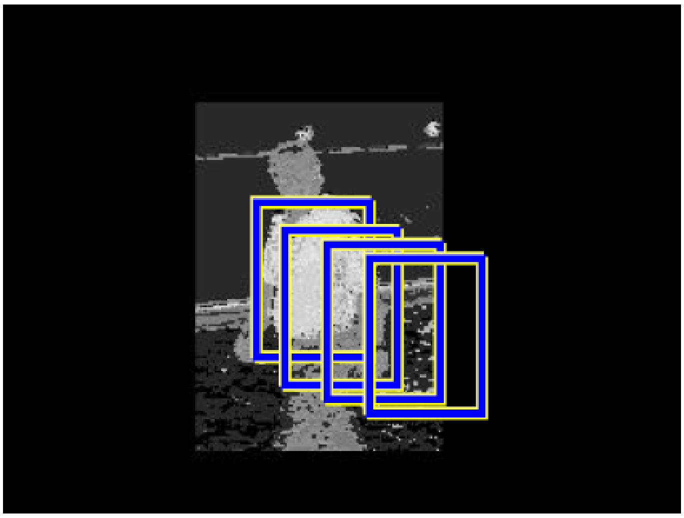

Figure 1.1: Mean Shift Algorithm

1.2 Lucas Kanade Method

Lucas Kanade method is widely used differential method for optical flow estimation. It assumes that the flow is essentially constant in a local neighborhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighborhood. There also have many variants of Lucas Kanade. Popularly as Kanade-Lucas-Tomasi (KLT) [6] which using Lucas-Tomasi feature for tracking. Another paper from [5] introduces many strategies to update template which is also applied in my project. Paper from [1] develops a completely framework to track object from warping to update template. The framework is described in figure 1.2. By the way, Lucas Kanade algorithm only works well on rigid object (cars, ...) and become worse if object have many changes from frame-by-frame.

In my project, a simple algorithm based o Lucas Kanade is implemented to compare with algorithm's Mean Shift. Although it still has many limitations, it can also track object which explicitly displays without covering of other object/background.

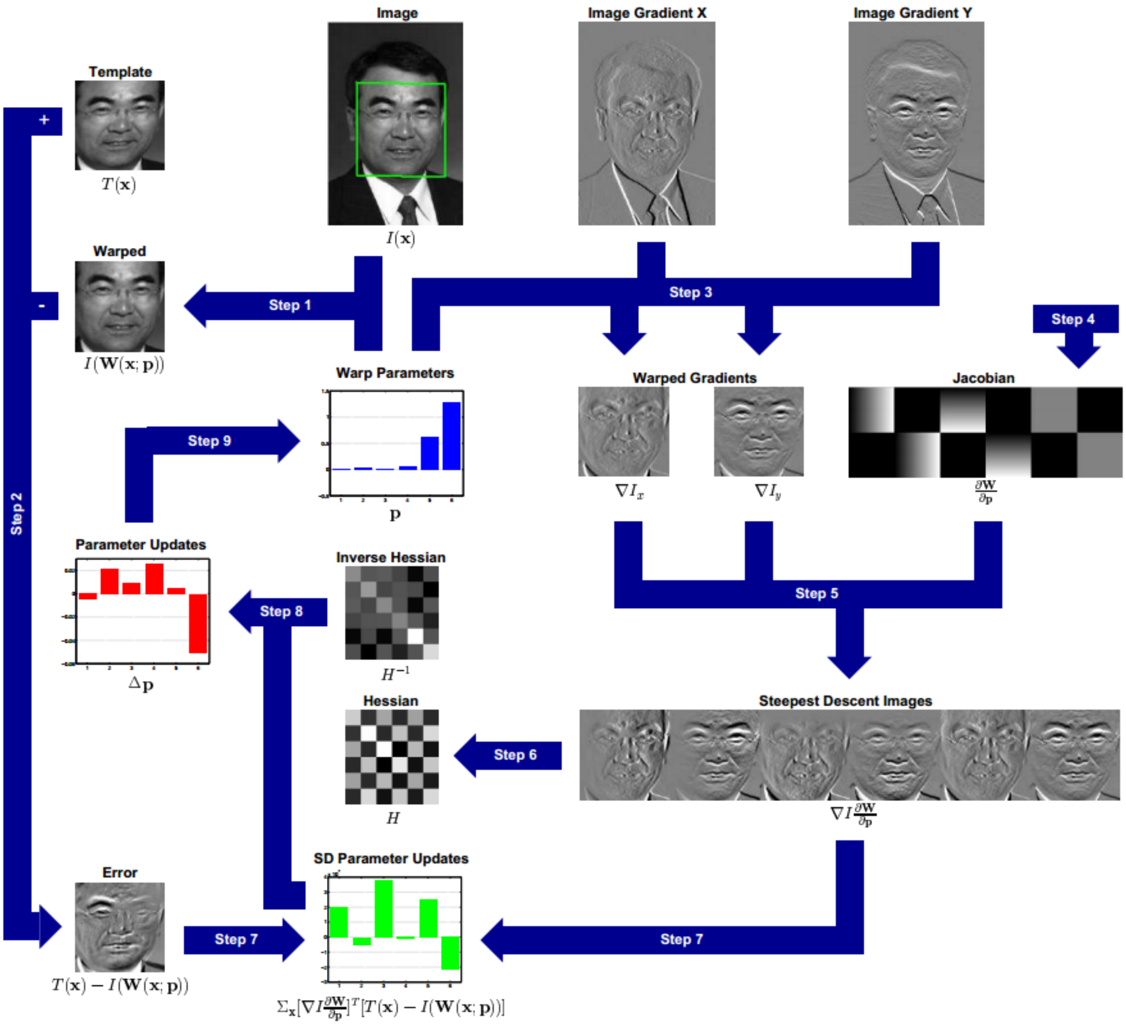

Figure 1.2: A Lucas Kanade framework in [1]

Chapter 2

Main components in my implementation

My project is implemented using MATLAB software and is divided into two algorithms separately: Mean Shift and Lucas Kanade. Besides, reading file and getting object template is common components used in both algorithms.

2.1 Reading File

This procedure is used to read the name of all images in sequence. In order to save memory, I only read the name of all images but not their data. The data will be read later when tracking method is really running. User must enter the path to folder containing images. Besides, I also need user to enter the index of started image which is used to get object template.

2.2 Getting Object Template

After reading name of all images and reading index of the started image, I allow user to get template of object by select a bounding box around that object. Getting object template is made on started image. Template of object is specified by its extrema points (top, left, right, bottom), its center point and the its half-size. After getting object template, that template will be used as input in object tracking method (Mean Shift and Lucas Kanade).

2.3 Algorithm based on Mean Shift

The main ideal of Mean Shift algorithm implemented in my code is computation of an offset from the current location to a new location based on distribution of color in object template. The new location of tracked object must have the same/similar distribution of colors as object template. Basically, flow of Mean Shift algorithm in my project can be summarized as follows:

$$wi = \sqrt{\frac{pdf}{currPdf}} \tag{2.1}$$

$$newCenter = \frac{\sum_{i=1}^{n} x_i w_i g_i}{\sum_{i=1}^{n} w_i g_i} \tag{2.2}$$

1. Calculate the distribution of colors in template (**pdf**) at started frame.

   pdf of object template is combined with its background (**pdfPg**) to reduce effect of background on that object template. This is called Background-Weighted Histogram (**BWH**) and is introduced in [4].

2. Process i^{th} frame image.

3. Let **iter** = 0. Iteration is repeated until the max iteration (**maxIter**).

4. Calculate the distribution of colors in template at current frame image (**currPdf**).

5. Calculate the weights image (**wi**) according to Eq. 2.1.

6. Calculate new location of center point (**newCenter**) according to Eq. 2.2. In here, g_i is Epanechnikov kernel.

7. Check if Mean Shift distance (**ms**) is convergence or not.

   If convergence $\Rightarrow$ stop iteration and go to step 2 for the next frame image.

   Otherwise, update information of object template and goto step 3 for the next iteration (**iter** = **iter** + 1).

8. The algorithm also stops if object goes outside the frame image. (tend to disappear in current image)

2.4 Algorithm based on Lucas Kanade

$$C * [u; v] = D \tag{2.3}$$

$$C = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{2.4}$$

$$D = \begin{bmatrix} I_t I_x \\ I_t I_y \end{bmatrix} \tag{2.5}$$

Same as Mean Shift Algorithm, Lucas Kanade also repeats operations on a frame image until getting a acceptable position. A strategy to update template is also applied in my implementation. Particularly, I summarize Lucas Kanade as bellow:

1. Let previous frame be started frame (specified by user)

   (**prevFrame = startedFrame**)

2. Process i^{th} frame image.

3. Let current frame (**currFrame**) be i^{th} frame image

4. Let offset $u = 0$ and $v = 0$. **u** and **v** are offset of object template from previous image to current image.

5. Calculate C according to Eq.2.4. C is matrix in Lucas Kanade equation according to Eq. 2.3

6. Let **iter** = 0. Iteration is repeated until the max iteration (**maxIter**).

7. Warping current image (**WCurrFrame**) in system of previous image.

8. Calculate I_t which is different between warping image and previous image.

9. Calculate D according to Eq.2.5.

10. Resolve Lucas Kanafde equation to find **u** and **v**.

11. Update value of u and v. Then

    If offset at current iteration is less than threshold (**offsetThresh**) $\Rightarrow$ Stop iteration, update template, update previous frame ($prevFrame = currFrame$) and go to step 2 for the next frame image.

    Otherwise, goto step 6 for the next iteration (**iter** = **iter** + 1)

12. The algorithm also stops if object goes outside the frame image. (tend to disappear in current image).

Chapter 3

Result and Analysis

3.1 Result

3.1.1 Mean Shift

Mean Shift algorithm gives very good results when tracking object which in environment of sparse objects and it is not covered by other object. Experiment on sequence of image as Figure 3.1 gives the good result. Object is given template at frame 50^{th} and is tracked from frame 50^{th} to the end of images sequence frame 1821^{th}. It means that ratio of tracked frame is 100%

Figure 3.1: Tracking a car in frame 50th, 250th, 500th, 800th, 1100th and 1500th using Mean Shift

However, if object is covered by other objects, mean shift method tends to lose tracking on this object. Figure 3.2 describes a failure case which tracking is lost by the car object is disappeared under the tree. In this images sequence, ratio of tracked frame is 83%

Figure 3.2: A failure case which the car is disappeared under the tree

3.1.2 Lucas Kanade

In other to compare with Mean Shift, an experiment of Lucas Kanade is also made. Generally, Lucas Kanade in my implementation also can track well a object in sparse environment. However, its result is not better than Mean Shift (the ratio of tracked frame is from 50% to 70%).

In addition, performance of Lucas Kanade is poor. When make experiments of two algorithms on the same images sequence, I see that Mean Shift need average 0.04 seconds to work on each frame. In case of Lucas Kanade is 0.93 seconds (equal 23 times of Mean Shift). Mean Shift can track object in real time but not Lucas Kanade. The reason will be explained in the next section.

3.2 Analysis

As result above, performance of Mean Shift is better than Lucas Kanade. The reason I can list as below:

1. Mean Shift only processes on a part of frame image which equals to size of object template. That means that if size of object template increases, time to run Mean Shift also increase and conversely. In contrast, Lucas Kanade must process on whole frame image. Especially, warping operation takes long time when processing on whole frame image.

2. Mean Shift algorithm convergence faster than Lucas Kanade. With the same maximum of iteration number (**maxIter**), Mean Shift only need a small iteration number to lead next position, but Lucas Kanade tends to running until reaching **maxIter**.

Although Mean Shift have a little bit better than Lucas Kanade, both of them seems not good in case of crowded environment where object is covered by other objects. In order to solve this issue, we maybe need to apply other techniques. For example, Kalman Filter for predicting moving direction or using Data Association on each moving object.

Bibliography

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[2] R. Collins. Mean-shift blob tracking through scale space. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–234–40 vol.2, June 2003.

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

[4] D. Comaniciu, V. Ramesh, P. Meer, S. Member, and S. Member. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:564–577, 2003.

[5] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 26:810–815, 2003.

[6] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.