Project Report: Implementation of Laplace Based Image Hashing for Object Tracking on Traffic Scene Videos

Rezka Bunaiya Prayudha
ENGR7761-Image Processing
Flinders University
Due date: 02-June-2020
Date Submitted: 02-June-2020

***Abstract*—One of important aspect in Computer Vision domain is video object tracking. In this report, result of three perceptive hashing template matching for object tracking development and experiment is described. The tracking process involves three hashing methods, namely Average Hash (AHash), Perceptive Hash (PHash), and Laplace Based Hash (LHash). The hashing methods is applied and evaluated on challenging traffic scene video sequences. Based on qualitative comparison, experimental results show promising performance of three hashing methods were they able to track objects under various constraints.**

***Index Terms*—computer vision, image hashing, object tracking, Laplace**

I. Introduction

This report is developed to present experiment and result of implementation of image hashing for object tracking. Object tracking plays important roles in computer vision field which many researches recently has taken object tracking as their focus. Object tracking provides promising performance in many application such as robot navigation, intelligent transport system and medical diagnosis.

There are two steps for object tracking based on visual recognition. First, detection on interesting objects and tracking particular object frame by frame. In recent development, there is a need to implement more advanced technique for object recognition and detection by employing artificial intelligence such as convolution neural network. In this experiment, more conventional approach is used by implement hashing code that generated from object's image. The hash code is object's "personal identification" that will be used to track object frame by frame by matching previous and next frame hash code.

In this project there are three hash code methods that will be used for object tracking which are Average Hash (AHash), Perceptive Hash (PHash), and Laplace Based Hash (LHash) method. The qualitative comparison is conducted to measure performance of three hashing methods demonstration to three videos of traffic scene.

II. Object Tracking Based on Hashing Methods

Hashing method is used in many application. Most common application of hashing method is in authentication where it try to find similarities between contents. Unlike cryptographic hash that generates random codes for same data, perceptive hashing methods that used in this project generates same codes for the same data. The similarity between the codes will determine image accuracy. In this section, three hashing methods algorithms are described.

A. Average Hashing Method

Average Hash (AHash) works by utilise frequency differences in the image. Low frequency in the image contains information such as luminance change in certain areas and high frequency contains dramatic changes such as edge of an object. AHash mainly uses low frequency parts of an image. Details of how to construct hash codes based on AHash method is described below:

- Resize the image to 8x8 pixels size. This step allows fastest way to remove high frequency contents.
- Convert the resized image to grayscale value to make total color of 64 pixels (64 RGB color) to only 64 color
- Find the mean value of constructed 64 color image. This step will be used for determine the threshold color of the image. For color larger than threshold will be set 1, and for color lower than threshold will be set 0.
- Construct 64 bit integer hash code based on binarization value of previous step.

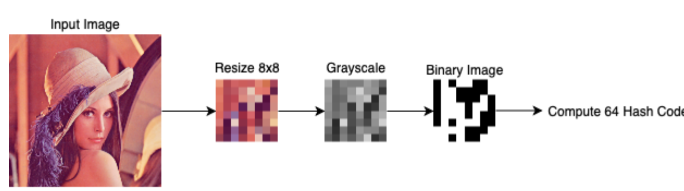

Fig. 1. Process of Average Hashing Method.

B. Perceptive Hashing Method

Average Hashing result is based on the threshold or average value of color. This value could be changed if color correction is applied to the image. Hence, it will change the binary value between below and above threshold depends on location of average value.

Perceptive Hashing is used to overcome this constraint by applying Discrete Cosine Transform (DCT). DCT transforms image from pixel domain to frequency domain by separates it into collection of frequencies and scalars. DCT tends to show that most information concentrated on low frequency. The implementation of PHash includes the following steps:

- Resize the image into 8x8 image and convert to grayscale value.
- Apply DCT method to 8x8 grayscale image to obtain 8x8 DCT coefficient matrix.
- Calculate the mean value of the DCT coefficients.
- Compute the binary hash code with respect to the mean value and convert the binary value to 64-bit integer. The result is the fingerprint of the image.

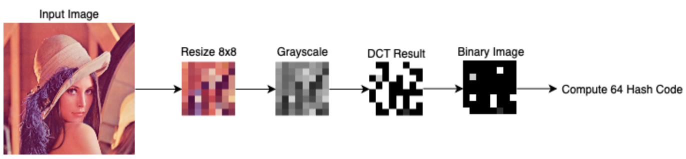

Fig. 2. Process of Perceptive Hashing Method.

C. Laplace Hashing Method

In the traffic scene video the object tends to move very fast and create a motion blur. To outperform image blurring, the differential operator could be used to highlight image details. Laplace transform as one of differential operator could enhance edge regions and weakens the varying gray areas of an image.

The Laplace transform is used in this hashing method to solve motion blur problem and track object more accurately. The specific process to generates hash code based on Laplace transform is describe below with details result shows in Fig. 3.

- Convert the image into grayscale value and resize the image to 8x8 pixel size.
- Apply Laplace transform to resized grayscale image.
- Calculate the mean value of Laplace step result and used it as threshold for binarisation.
- Compute the 64-bit hash code.

Fig. 3. Process of Laplace Hashing Method.

D. Hamming Distance

In this project, successful object tracking is determined by similarity between object recognition in first frame and the next frame. To measure the similarity, the Hamming Distance technique is used. Hamming Distance is the number of position at which corresponding symbols are different in two equal length of strings. In this project, Hamming distance measure similarity of object hash codes in two different frames. The smaller the hamming distance, the more similar the two images is. A Hamming distance with result 0 indicates the images is identical.

III. Experiments and Results

To test and evaluate the proposed methods, the experiments were conducted to three traffic scene video dataset on a personal computer with specification Intel 2.3 GHz Dual-Core Intel Core i5, RAM 8GB under MacOS environment combined with OpenCV libraries. The dataset is based on highway scenario, where it contains challenging sequences such as the object get smaller or larger due to frame changing. In this section, the result is present based on qualitative comparison of three hashing methods over some representative frames in six video sequences.

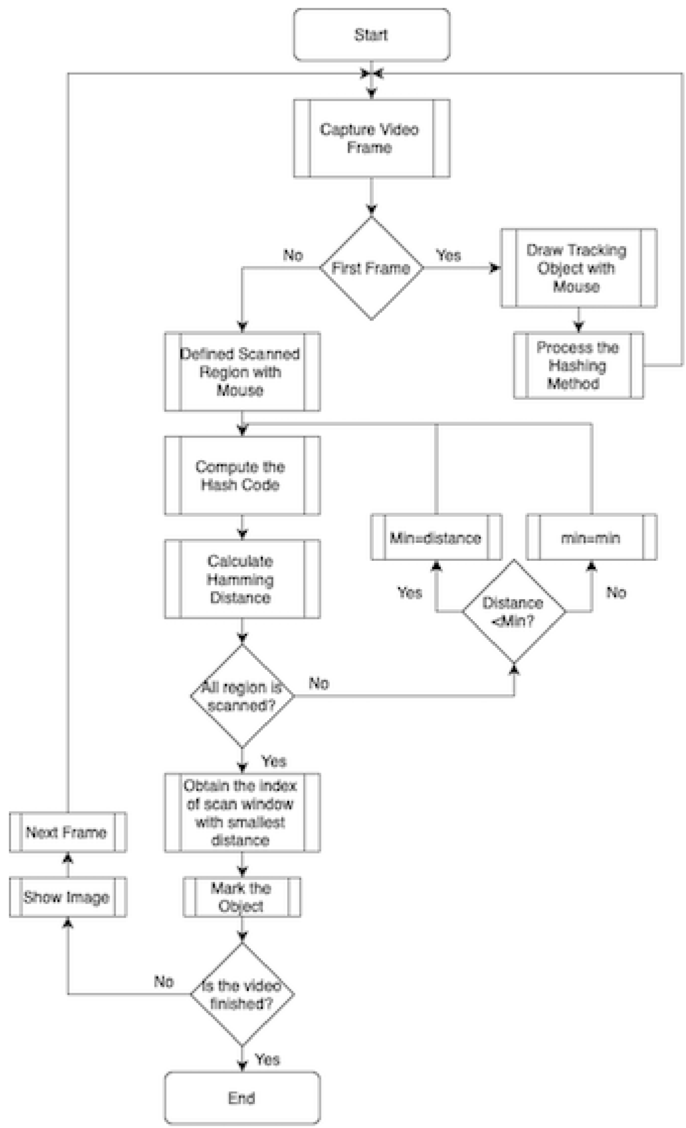

Fig. 4. Flowchart of Object Tracking.

A. Experiment Methods

The experiment perform object tracking using template matching method to computing image similarity in the first frame and the next frame. The experiments steps are, first select the object in the first frame with the mouse. This selected object image will be applied with the hash operators to generate the codes. The same steps is applied to the next frame and the distance between consequences frames is computed.

Based on Fig.4, the process are:

1) Capture the first frame and draw rectangle box around object with mouse r_0. Compute the hash code H_0 of the object;
2) Capture the next frame and select image area r_s around the object;
3) Using sliding window technique, scan each part of r_s, and separately calculate the hash code H_s;
4) Calculate the hamming distance between H_0 and H_s. Compared each calculated distance and find the smallest distance;
5) Update the location of object in r_s based on selected smallest distance;
6) End the cycle.

B. Results and Discussion

The hashing methods were applied to three different videos. The videos is come from [3] which shows highway situation with vehicles. It is worth noting that the conducted experiment did not apply background subtraction to increase detection accuracy. The experiment only focused on detection based on captured image similarity. The result of experiment is show in Fig. 5.

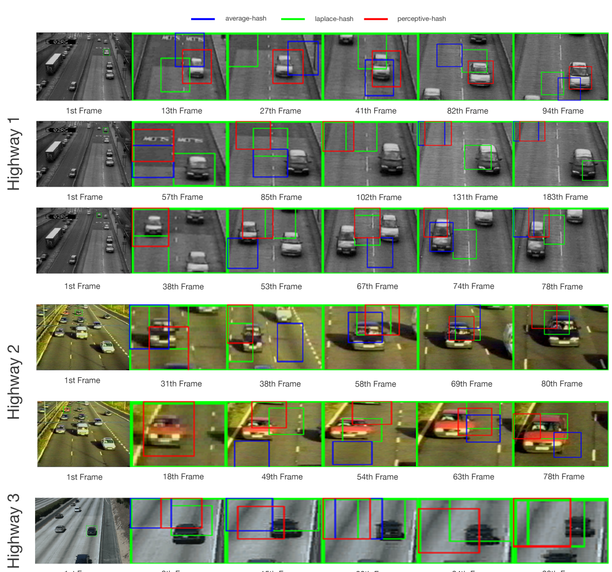

Fig. 5. Qualitative Comparison of Three Hashing Methods.

In Highway 1 first row frame sequences, it is shown that perceptive hash outperform other hashing methods. PHash can tracked object from 13^{th} frame to 94^{th} frame. In the second row of Highway 1 video sequences, the LHash method perform better than other hashing methods which it could tracked object at 57^{th} frame, 131^{th} frame, and 183^{th} frame. However, at 85^{th} frame and 102^{th} frame all the methods lost track of the object. In the last row of Highway 1 video sequence, the PHash perform better at 38^{th} frame than other methods but then it lost track. At 74^{th} frame AHash perform perfectly than other methods. In this sequence, the LHash method could not tracked object accurately.

The first row of Highway 2 video sequence shows that LHash method performed better than other methods throughout the whole video sequence. In the second row of Highway 2 video sequence, both LHash and PHash method perform quite better than AHash Method. In Highway 3 video sequence LHash and PHash successfully track the target throughout the video even although the methods did not accurately track in the center of the object.

From the result shows that some video sequences encounter challenges such as target size changes and complex background where in some frame in contains edge caused by other object such as roadway or other vehicle. Throughout the whole video sequences, the three methods present well performance object tracking although in some frame, one or two methods failed to track the object. The challenge also lay on determine the image area r_s in the next frame. The image area should be draw precisely to enhance the accuracy of tracking. At first the size of window for the sliding window search is based on r_0 or the size of object size in the first frame. However, the distance computation will be not accurate if r_s is too small or too large. To maximize the search accuracy, the size of $32 * 32$ is chosen for sliding window size.

In Highway 1 and Highway 2 video sequences the tracking most affected by object size changes where the object is become larger in the sequences. For Highway 2 video sequence, the challenging part is complex background where it contains highway separator and other vehicles. Foreground detection and background subtraction will be beneficial to be implemented to overcome complex background problem. Furthermore, by comparing three different methods for object tracking, the result shows promising performance of the three methods based on experiment on three different video sequences.

IV. Conclusion

In this report, a visual tracking method based on perceptual hashing is present. Three hashing methods namely AHash, PHash and LHash, were applied and evaluated on challenging video sequences. Experimental results indicates that the three methods are able to track object in three different videos. Despite constraints such as object size changes and features complexity, the hashing methods shows promising results. For future topics, it is beneficial to combine the hashing methods with object recognition technique for object re-identification.

References

[1] M. Fei, Z. Ju, X. Zhen, and J. Li, "Real-time visual tracking based on improved perceptual hashing," Multimed Tools Appl, vol. 76, no. 3, pp. 4617–4634, Feb. 2017, doi: 10.1007/s11042-016-3723-5.

[2] M. Fei, J. Li, and H. Liu, "Visual tracking based on improved foreground detection and perceptual hashing," Neurocomputing, vol. 152, pp. 413–428, Mar. 2015, doi: 10.1016/j.neucom.2014.09.060.

[3] http://ftp.pets.rdg.ac.uk

Appendix

For details implementation and code download or clone this GitHub repository: GitHub Repo