ARIZONA STATE UNIVERSITY,
ELECTRICAL & COMPUTER ENGINEERING DEPARTMENT,
EEE 508, IMAGE PROCESSING & COMPRESSION

Comparison of Motion estimation using (FAST & FREAK) , (FAST & LUCID) feature detctor and descriptor with Lucas-Kanade Optical flow method

Amith A Lawrence, Abhijith V Kumar

November 8, 2019

Abstract

In video sequences, motion is a key source of information and it arises due to moving objects in the 3D scene, as well as camera motion. Apparent motion, also known as optical flow, captures the resulting spatial-temporal variations of pixel intensities in successive frames in a video sequence. The purpose of motion estimation techniques is to recover this information by analyzing the image content. Efficient and accurate motion estimation is an essential component in the fields such as image sequence analysis, computer vision and video communication. Motion Estimation can be performed by either Optical flow or Descriptor matching.

Key words: Apparent motion, Pixel intensities, Motion estimation, Optical flow, Descriptor matching

1 INTRODUCTION

This report deals with Motion Estimation implemented by using 2 different methodologies:

1. Lucas-Kanade Optical Flow
2. Descriptor matching

Optical flow algorithms focuses on pixel patches around features and tries to match those patches instead. Optical flow methods in contrast rely on the minimization of the brightness constancy and additional constrain e.g. smoothness etc. Thus the motion vector is derived based on spatial and temporal image gradients of a sequence of consecutive frames. This method is therefore is more suited for image sequences rather than image pairs that are captured from very different view points.
The optical flow methodology assumes brightness constancy, that is, that pixel brightness doesn't change between frames. Also, involves the assumption that neighboring points to your features move similarly to your feature.

Feature matching uses the feature descriptors to match features with one another (usually) using a nearest neighbor search in the feature descriptor space. The basic idea is to utilize descriptor vectors, and the same feature in two images provided they both are close to each other in the descriptor space, for matching the points.

A feature detection step which returns a set of so called feature points. These feature points are located at positions with salient image structures, e.g. edge-like structures when you are using FAST or blob like structures if you are using SIFT or SURF.

Keypoints are the points in an image that are interesting or stands out in an image which are likely to be recognized in another image. Keypoints/interest points constitute to edges, corners, blobs or ridges based on the type of application. For different images of the same scene, key points should have repeatability. A descriptor is a vector of values, which describes the image patch around an keypoint. Descriptors are the way to compare keypoints. An interest point and its descriptor is commonly used together.

The second step is the matching. The association of feature points extracted from two different images. The matching is based on local visual descriptors, e.g. histogram of gradients or binary patterns, that are locally extracted around the feature positions. The descriptor is a feature vector and associated feature point pairs are pairs a minimal feature vector distances.

Most feature matching methods are scale and rotation invariant and are robust for changes in illuminations (e.g caused by shadow or different contrast).

2 DESCRIPTION OF ADOPTED APPROACHES

2.1 FEATURE DETECTOR

2.1.1 FAST (Features from Accelerated Segment Test)

The reason behind the work of the FAST algorithm was to develop an interest point detector for use in real time frame rate applications. In this detector a circle of a specific number of pixels and the intensities of the pixels within are compared to I1, I5, I9, I13 pixels and if found to be less than a particular value. Detection of multiple interest points adjacent to one another is one of the other problems of the initial version of the algorithm. This can be dealt with by applying non maximal suppression after detecting the interest points. The limitation of the detector is that the speed with which the pixels are queried in other words the manner in which the pixels are searched.

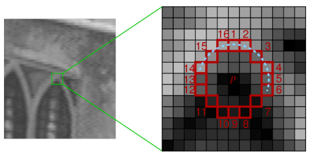

Figure 2.1: Bresenham Circle used for FAST

2.2 FEATURE DESCRIPTORS

2.2.1 FREAK (Fast Retina Keypoint)

The binary descriptor algorithm uses the retinal sampling grid which is circular and consists of inner circles symmetrically distributed, having higher density near the center. The retinal sampling grid to compute a sequence of one-bit Difference of Gaussians. The FREAK algorithm is based on a simple computation, selecting two circles randomly and then looking for the pairs that give more information. In combination with the sampling pattern it is also comprised of orientation compensation and pairing of sampling points. The descriptor also includes a mechanism to measure the orientation of the keypoint and compensate the difference.

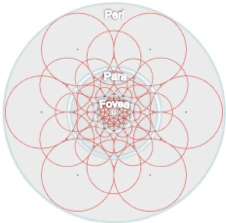

Figure 2.2: FREAK Sampling pattern

2.2.2 LUCID (Locally Uniform Comparison Image Descriptor)

A simple descriptor based on permutation distances between the ordering of intensities of RGB values between two patches. Image patches of size nxn are selected from a color image and the descriptors are computed for both patches.The hamming distance between the patches is calculated and the order permutations(mapping of a finite set onto itself) of the descriptors LUCID is computable in linear time with respect to patch size and does not require floating point computation.

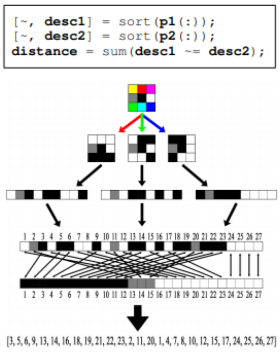

Figure 2.3: LUCID feature construction and matching

2.3 Pyramidal Implementation of the Lucas Kanade Feature Tracker

The basic concept of the Lucas Kanade algorithm is based on 3 assumptions:

1. Brightness constancy
2. Temporal persistence (small movements)

3. Spatial coherence

In certain images, if the new location is not far from the original feature location, using dense flow algorithm on small patches. Otherwise, the pyramid-resolution approach is recommended, in which the image is scaled down before tracking the features. The translation which spans over 16 pixels is now a 2 pixel translation, thereby scaling up with the detected transformation as your prior. The basic methodology of the pyramidal approach is as follows: The optical flow is computed at the deepest pyramid level A. Then, the result of that computation is propagated to the upper level B in a form of an initial guess for the pixel displacement (at level B). Given that initial guess, the refined optical flow is computed at level B The result is propagated to level B and so on up to the level 0 (the original image).
The clear advantage of a pyramidal implementation is that each residual optical flow vector d while computing a large overall pixel displacement vector d.

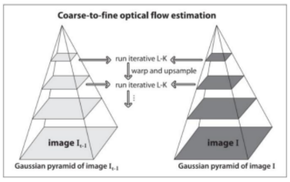

Figure 2.4: Pyramid Lucas Kanade Optical Flow

2.4 MATCHING METHOD

2.4.1 FLANN (Fast Library for Approximate Nearest Neighbors)

As the name implies this descriptor matcher finds the approximate nearest neighbors. It is important to note that the FLANN matcher will find a good match, but may not be the best. The BruteForce matcher will try for all possibilities but at the cost of slowing the algorithm. An example of the BruteForce matcher was implemented to gain a certain amount of understanding regarding feature matching [Project: Rough_work].

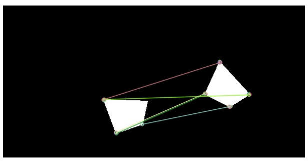

Figure 2.5: Feature matching using BruteForce matcher

3 ANALYSIS OF THE RESULTS

3.1 FEATURE TRACKING USING PYRAMID IMPLEMENTATION OF THE LUCAS-KANADE ALGORITHM

The features have been extracted from the frames of the video using FAST descriptor and using the Lucas-Kanade algorithm to track the corresponding features between the frames of the video in a sequential manner. A minimum of 2 features for each vehicle in the video have been tracked during the video. The same approach was tested on multiple videos and similar results were obtained.

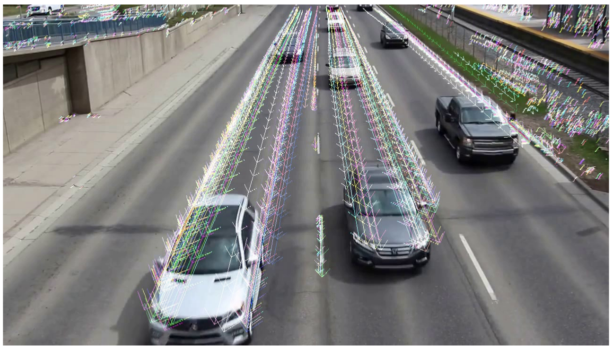

Figure 3.1: Result A

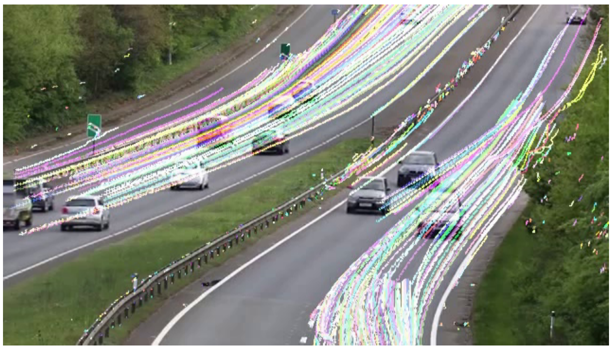

Figure 3.2: Result B

3.2 FEATURE DESCRIPTOR MATCHING:(DETECTOR & DESCRIPTOR)

3.2.1 FAST FREAK

The features have been extracted from the frames of the video using FAST descriptor and FREAK was used to generate the descriptors. The FLANN matcher is then used to get a good match between the features of consecutive frames of the video by using the descriptors as the criteria. The tracked features are limited for each vehicle in the video and the features are not tracked during the entire time the vehicle is in the video. The tracking starts only after the vehicle is visible for a few seconds. The same approach was tested on multiple videos and similar results were obtained. The features also appear to lose track and pick up with a intermittent delay as evidenced by the discontinuous arrowheads in both snapshots.

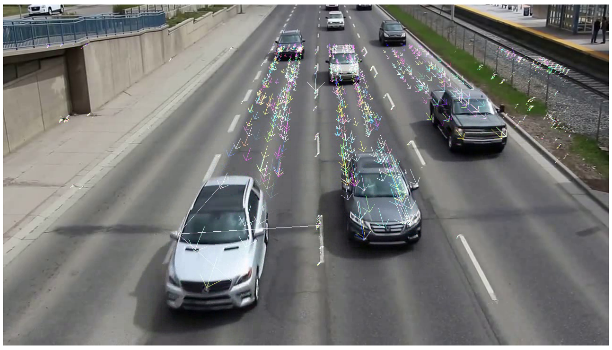

Figure 3.3: Result A

Figure 3.4: Result B

3.2.2 FAST LUCID

The features have been extracted from the frames of the video using FAST descriptor and LUCID was used for generating descriptors. A minimum of 2 features for each vehicle in the video have been tracked during the video. The same approach was tested on multiple videos and similar results were obtained. This combination is faster than the latter. There is no gap in tracking of features. There are a few contradicting outliers that have also crept into the output as seen in the snapshots.

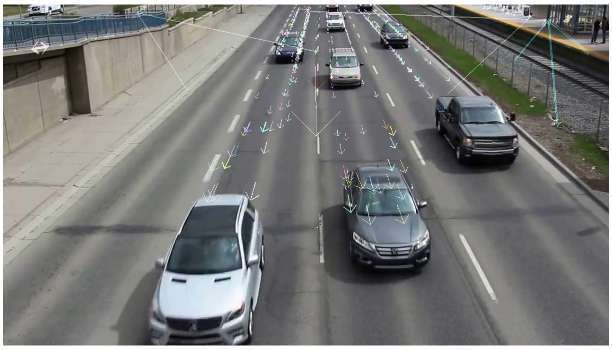

Figure 3.5: Result A

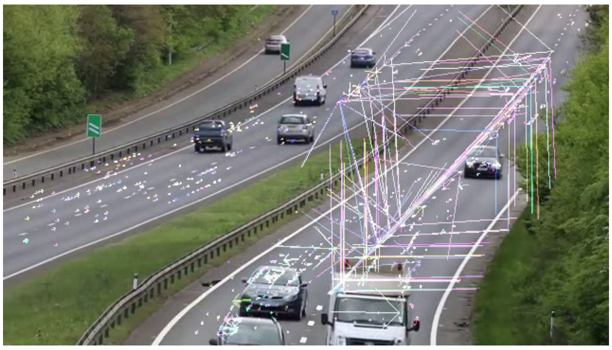

Figure 3.6: Result B

4 CONCLUSION

The motion estimation performed using the Pyramidal implementation of the Lucas-Kanade algorithm is more robust and produces better results without any significant delay being observed in the output. The efficiency of the descriptor matching approach is lower and may not work best with different video samples. The camera being used for the video capture also needs to be stabilied in order to avoid the algorithm to latch onto irrelavant movements. Therefore, a better result can be obtained if the video is stabilized before our motion estimation approach is applied.

The surrounding environment also tends to contaminate the feature set by creeping in due to minor movement being registered. A good example of such a situation is the movements of the branches observed in one of the test videos, [clip2.mp4].

5 REFERENCES

1. B. D. Lucas and T. Kanade,An Iterative Image Registration Technique with an Application to Stereo Vision,International Joint Conference on Artificial Intelligence, pp. 674-679, 1981.
2. C. Tomasi and T. Kanade,Detection and Tracking of Point Features,Carnegie MellonUniversity Technical Report CMUCS91132, April 1991.J.
3. Shi and C. Tomasi,Good Features to Track, IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, 1994.
4. https:/docs.opencv.org
5. Alexandre Alahi, Raphael Ortiz, Pierre Vandergheynst , FREAK: Fast Retina Keypoint,Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland
6. Andrew Ziegler, Eric Christiansen, David Kriegman, Serge Belongie, Locally Uniform Comparison Image Descriptor, Department of Computer Science and Engineering, University of California, San Diego
7. Jean-Yves Bouguet, Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm, Intel Corporation