Lucas-Kanade using MATLAB

Akshat Bajpai[#1]

[#]*School of Engineering and Computer Science , University of the Pacific*

3601 Pacific Avenue, Stockton, CA

[1]a_bajpai@u.pacific.edu

***Abstract*— This document contains MATLAB implementation of an image processing algorithm known as Lucas-Kanade Segmentation. The algorithm is used to track different feature points in an image. The whole program comprises 2 sections. The first section deals with Lucas Kanade using Cherry Picked 21 feature points and the second section deals with Lucas-Kanade using Haris corner algorithm. The program takes in a sequence of images as an inputs and generates an output in the form of red points signifying feature points.**

***Keywords*— Lucas-Kanade algorithm, Haris corner algorithm, MATLAB, Feature Point tracking**

I. Implementation

This program is an MATLAB implementation of the Lucas-Kanade algorithm. The algorithm involves finding a feature of interest and tracking it through an image sequence. These could be used to detect facial landmarks, corners of a moving object, traffic estimation, etc. In an image, these points are usually corners, which are an intersection of two edges.

The basic idea behind the algorithm is to capture a single point in previous image and relocate it in the current one. This would be easiest to spot if the point is distinct in the previous image. The feature that makes it distinct is the sudden change of pixel intensity(color) when compared to its neighboring pixels. This is why corners are preferred as good feature points. A good corner point's change in the intensity level when compared to its neighbors is really high. This is what Harris corner algorithm tries to figure out. So based on this, the image on the left(Fig.1.) with more distortions would have more good keypoints to follow than the one on the right(Fig.2), which is a plain image.

Fig.1 More Keypoints

Fig.2. Less Keypoints

II. Methodology

We utilize most of the functions from the earlier labs such as gaussian, gaussian derivative, convolve, etc. The program begins by taking in an input as the first image from a sequence

("flowergarden", "statue_seq"). As an example, we'll select the "flowergarden" image sequence. The program then gives us an option to choose from Cherry-picked or Haris corner.

If the user chooses cherry-picked, the program prompts to select the first image from the sequence. Then it asks the user to mark 21 feature points. Preferably, these points should be corner points. The program then runs the Lucas-Kanade algorithm on the image with the selected feature points. The output of the lucas-kanade function generates the coordinates for the next frame of the feature point compared to the current one. If the user chooses the Haris-corner option, the program picks the top feature points and runs the algorithm similar to the previous option. The program also makes sure that the next feature point placed is atleast 2-city block distant from another ones.

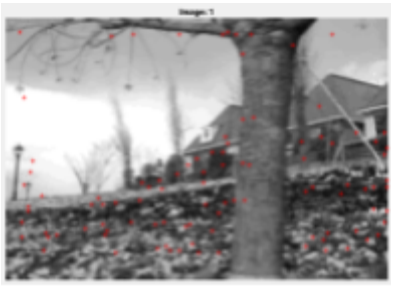

Fig.3. Keypoints detected in image 1

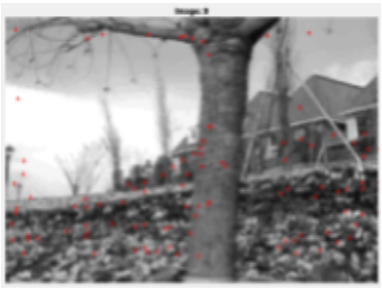

Fig.4. Keypoints detected in image 9

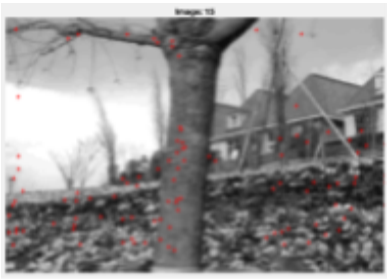

Fig.5. Keypoints detected in image 15

Fig.6. Keypoints detected in image 21

Fig.7. Keypoints detected in image 23

Fig.8. Keypoints detected in image 25

Fig.9. Keypoints detected in image 27

Fig.10. Keypoints detected in image 29

IV. Conclusions

The program successfully implements the Lucas-Kanade algorithm using MATLAB using both Cherry-picked and Haris controller methods. The points are tracked more efficiently with smoothing applied to the image before running the Lucas-Kanade algorithm.