International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 02 Issue: 08 | Nov-2015 www.irjet.net p-ISSN: 2395-0072

Motion Detection Using Optical Flow on Raspberry PI

[1]C. Sureindar, [2]Dr. P. Saravanan

[1]PG- Scholar, Department of Electrical and Electronics, Sriram Engineering College, Tamil Nadu, India

[2] HOD, Department of Electrical and Electronics, Sriram Engineering College, Tamil Nadu, India

---***---

Abstract – This paper presents to implements the motion detection algorithm through optical flow using Raspberry Pi. The idea behind this project is to find the displaced object from an image using Lucas-Kanade algorithm. This algorithm works by comparing the first two successive image frame with that it guess the direction of the displaced object. The main advantage of this algorithm it doesn't need to scan the next image for matching the pixel or neighbouring pixel of the image. This process can be achieved through optical flow vector the vector will be similar to small neighbourhood surrounding pixel. This algorithm successfully implemented on using Python OpenCV.

Key Words: **Motion Detection; Optical Flow; Lucas-kanade**

1. INTRODUCTION

The process of image processing is mainly depends on the process of "illuminations" from source and "reflectance" from the object or some absorbtion from scene we can acquire an image. In modern technology motion detection plays a vital role in the field of real time application for an artificial intelligence. Motion detection process is used to find out the displaced object from a relative scene. The motion we get from that process may be a salient one, or it may be distractive one. More number of approaches are followed to find out displaced object through motion detection on a real time video streams. The displaced object from a scene is identified by the help of optical flow vector method. The motion flow is used for monitoring the 2D environment while the advanced 3D technology is process with the help of optical flow field.

The optical flow algorithm will make several number of assumption before getting in to calculation. Changes in the illumination of a respective scene and surface reflectance will cause some violation to our assumptions. Inconsistencies in the optical flow field are possible through occlusion effect.

The best choice for the optical flow method is Lucas-Kanade algorithms. This algorithm works by comparing the two successive image frame based on that it estimates the displaced object from that two successive image scene. The moved object from a scene is highlighted by the

optical flow vector. Lucas-kanade algorithm doesn't need to scan the next image for matching the pixel of image or neighbourhood pixel.

2. 2. MOTION DETECTION USING OPTICAL FLOW

3. The distribution of apparent velocities of brightness pattern movements in an image is termed as Optical flow[2]. It can arise from relative movements between objects and the viewer (or camera). Optical flow provides a description of motion helpful for image interpretation even if there are no quantitative parameters obtained from motion analysis.

4. 2.1 Optical Flow Constraint Equation

5.

6. The basic assumption made in optical flow calculations image brightness constancy. This is simply the assumption that while an object may change position from a short interval t1 to t2, the reflectivity and illumination will remain constant. Mathematically, it can be expressed as:

7.

$$f(x+\Delta x; y + \Delta y; t + \Delta t) \approx f(x; y; t) \quad (1)$$

where, $f(x; y; t)$ is the intensity of the image at position (x,y)and at time t, and Δx, Δy is the change in position and Δt is the change in time. When Taylor series expansion is applied to equation 1, optical flow constraint equation is found by ignoring the higher order terms.

$$\nabla I.V+ I_t = 0 \quad (2)$$

where, $\nabla I = (I_x; I_y)$ is the spatial gradient, V= (u,v) = (Δx, Δy) is the optical flow vector and It is the temporal gradient. As single time displacement between two frames is considered, $\Delta t = 1$ and therefore disappears. The spatial and temporal gradients are easy to calculate using derivative operators. The optical flow vector (u,v) represents the displacement of pixels. When a flow vector is applied to the spatial gradient of the image it would be exactly cancelled by the temporal gradient. This makes sense since it is assumed that there will be no change in the brightness of the image.

3. LUCAS-KANADE ALGORITHM

Lucas and Kanade's method [1] involves solving for the optical flow vector by assuming that the vector will be similar to a small neighbourhood surrounding the pixel. The Lucas-Kanade optical flow algorithm offers a simple technique which can provide an estimate of the movement of interesting features in successive image frames of a scene. A movement vector (u,v), obtained by comparing the two consecutive image frames, is associated to every such "interesting" pixel in the scene.

The assumptions made by Lucas-Kanade algorithm:

1) The two images are separated by a small time increment Δt, in such a way that objects have not displaced significantly (that is, the algorithm works best with slow moving objects)

2) The images depict a natural scene containing textured objects exhibiting shades of gray (different intensity levels) which change smoothly

The algorithm does not use colour information in an explicit way. It does not scan the second image looking for a matchfor a given pixel. It works by trying to guess in which direction an object has moved so that local intensity changes can be explained. It uses a weighted least squares method to approximate the optical flow at pixel (x,y).

$$Ev = \Sigma p \in \Omega[\nabla I(p).v + It(p)] \quad (3)$$

Where, $\delta I(p)$ and $It(p)$ represents the spatial gradient and temporal gradient at neighbouring pixel 'p' respectively. The optical flow vector for pixel (x,y) is 'v'. For each pixel, an optical flow vector consistent with the neighbouring spatial and temporal gradients is found. This is done by considering a surrounding neighbourhood 'Ω', where each neighbour is represented as 'p'. The error of applying the flow vector 'v' to the spatial and temporal gradients of all the surrounding neighbours is summed using the optical flow equation. The error 'E_v' will be higher if the considered flow vector is inconsistent with the spatial and temporal gradients of some neighbours.

$$A^T Av = A^T b \qquad (4)$$

where, 'A 'is a vector of all the spatial gradients of all then neighbours of the neighbourhood 'Ω', and 'b' Is a vector of the temporal gradients. Solving for 'v' gives us:

$$v = (A^TA)^{-1}A^T b \qquad (5)$$

Gaussian pyramid levels [1] were used to detect movement of both large and small objects in the scene. At each successive

Level, the size of image gets reduced by a constant ratio. For each level of the Gaussian pyramid, the optical flow was computed for each pair of frames.

4. SIMULATION AND HARDWARE PLEMENTATION

The Lucas-Kanade algorithm was simulated using Python OpenCV. The optical flow vectors shows the direction of The Python OpenCV program was run on Raspberry Pi. The output window displayed the processed result of real-time video stream captured using Raspberry Pi camera.

Fig.1. Block Diagram

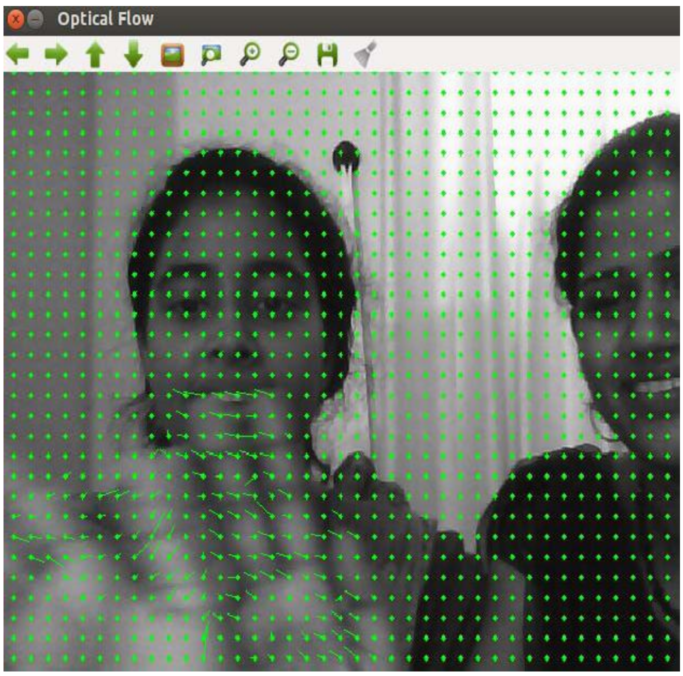

Fig.2. Simulated Results

5. CONCLUSIONS

The Lucas-Kanade algorithm estimates the motion of objects without relying on the entire image. Therefore the advantage is that for a neighbourhood of fixed size, the total number of operations needed to compute the optical low vector remains constant. The algorithm worked well for moderate object speeds

REFERENCES

[1] N. Amamoto and K. Matsumoto, "Obstruction Detector by Environmental Adaptive Background Image Updating," In ERTICO, editor, 4th World Congress on Intelligent Transport Systems, No. 4, pp1-7, Berlin, Oct.1997. Traffic Technology International.

[2] S. Huwer and H. Niemann, "Adaptive Change Detection for Real-time Surveillance applications," In the Proc. of the 3rd IEEE Workshop on Visual Surveillance, 2000, pp.37-45.

[3] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background Modeling and Subtraction of Dynamic Scenes", In Proc. of International Conference on Computer Vision (ICCV), 2003, Pages 1305–1312.

[4] Y. Ren, C. Chua, and Y. Ho, "Motion Detection with Non-stationary Background," In Proc. Of 11th Int'l Conf. Image Analysis and Processing, 2001, 78-83.

[5] C. Stauffer and W.E.L. Grimson, "Adaptive Background mixture Models for Real-time Tracking", CVPR99, June, 1999.

[6] K. Toyama, J. Krumm, B. Brumitt and B. Meyers, "Wallflower: Principles and practice of background Maintenance." In Proc. International Conference on Computer Vision, 1999, 255-261.

[7] R. P. Wildes, "A Measure of Motion Salience for Surveillance Applications," In Proc. Of IEEE International Conference on Image Processing, p183-187, 1998.

[8] L. Wixson, "Detecting Salient Motion by Accumulating Directionally Flow," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22. No. 8. pp774-779, August, 2000.