OPTICAL FLOW FOR MOTION DETECTION WITH MOVING BACKGORUND

Research · October 2015

DOI: 10.13140/RG.2.1.2206.5363

1 author:

Amisha J. Shah
C.K. Pithawalla College Of Engineering & Technology
9 PUBLICATIONS **42** CITATIONS

OPTICAL FLOW FOR MOTION DETECTION WITH MOVING BACKGORUND

Amisha Shah
ECED Department,
CKPCET,Surat,India,
Amisha_janak@yahoo.co.in

Dr. Suprava Patnaik
Professor,
SVNIT,Surat,India
Suprava_patnaik@yahoo.com

Chirag Paunwala
Assistant Professor
SCET,Surat,India
cnpaunwala@gmail.com

ABSTRACT

Each application that benefit from smart video processing has different needs, thus requires different treatment. However, they have something in common: moving objects. Thus, detecting regions that correspond to moving objects such as people and vehicles in video is the first basic step of almost every vision system since it provides a focus of attention and simplifies the processing on subsequent analysis steps. This paper represents optical flow based object detection. Optical flow can detect motion in video stream obtained from a moving camera. The motion between two frames is commonly described by a dense displacement vector field which links the location of each point in a given frame to its location in the next frame. Optical flow gives a description of this motion. Before moving object detection, if temporal movement (e.g. leaves movement) is present then preprocessing is required. Non-linear bilateral filter is used to remove the temporal movement. The rationale of bilateral filtering is that two pixels are close to each other not only if they occupy nearby spatial locations but also if they have some similarity in the photometric range.

Keywords— Bilateral filter, motion vector, optical flow, optical flow field.

1. INTRODUCTION

Extracting moving objects from a video sequence is the first step of many video processing applications like video compression, navigation systems, surveillance systems etc. If the camera is fixed in the scene, we can detect moving objects using different techniques like background subtraction, statistical methods, temporal differencing etc. However, for a moving camera, the problem is more complex because of the possibility that every object in the environment may be moving with respect to both the camera and the environment. We can reduce the problem by rectifying the image using the camera control signal. Before the object detection pre-processing is required. It may be necessary to enhance image intensities using smoothing or deblurring operations. Image smoothing reduces noise but blurs the image. Deblurring, on the other hand, reduces blur but enhances noise. The size of the filter selected for smoothing or deblurring determines the amount of smoothing or sharpening applied to an image. Bilateral filter is used to remove temporal movement like leaves movement. Bilateral filter is an edge-preserving smoothing filter. It replaces a pixel's value by a weigthed average of its neighbours in both space and range (pixel value). This preserves sharp edges by systematically excluding pixels across discontinuities from consideration. Moving objects are determined using intensity based optical flow. Optical flow methods make use of the flow vectors of moving objects over time to detect moving regions in an image. Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. Sequences of ordered images allow the estimation of motion as either instantaneous image velocities or discrete image displacements.

2. PRE-PROCESSING

Filtering is perhaps the most fundamental operation of image processing and computer vision. In the broadest sense of the term "filtering," the value of the filtered image at a given location is a function of the values of the input image in a small neighborhood of the same location. In particular, Gaussian low-pass filtering computes weighted average of pixel values in the neighborhood, in which, the weights decrease with distance from the neighborhood center [1].
The assumption of slow spatial variations fails at edges, which are consequently blurred by low-pass filtering. Bilateral filter is an edge-preserving smoothing filter. Whereas many filters are convolutions in the image domain, a bilateral filter also operates in the image's range-pixel value.

2.1 Bilateral filter

Bilateral filtering is a non-linear filtering technique introduced by Tomasi et al.. It extends the concept of Gaussian smoothing by weighting the filter coefficients with their corresponding relative pixel intensities. Pixels that are

very different in intensity from the central pixel are weighted less even though they may be in close proximity to the central pixel. This is effectively a convolution with a non-linear Gaussian filter, with weights based on pixel intensities. This is applied as two Gaussian filters at a localized pixel neighborhood , one in the spatial domain, named the domain filter, and one in the intensity domain, named the range filter.

A very intuitive mathematical approach is given as follows [2].

Let *f* be the original brightness function of an image which maps the coordinates of a pixel *(x, y)* to a value in light intensity. Then for any given pixel *a* at *(x,y)* within a neighborhood of size n, which has a_0 as its centre, its coefficient assigned by the range filter *r(a)* is determined by the following function:

$$r(a_i) = e^{\left(-\frac{f(a_i)-f(a_0)^2}{2\times\sigma_r^2}\right)} \quad (1)$$

Where, the range standard deviation σ_r defines the influence of neighboring pixels according to their intensity difference from center pixel.

Similarly, its coefficient assigned by the domain filter *g(a)* is determined by the closeness function below:

$$g(x,y) = e^{\left(-\frac{x^2+y^2}{2\times\sigma_d^2}\right)} \quad (2)$$

Where the range standard deviation σ_d defines the influence of neighboring pixels according to their distance to center pixel in the image lattice.

For the central pixel of the neighborhood a_0 , its new value, denoted by $h(a_0)$:

$$h(a_0) = k^{-1}\sum_{i=0}^{n-1} f(a_i)\times g(a_i)\times r(a_i) \quad (3)$$

where, k is the normalization constant to maintain zero-gain and is defined as follows:

$$k = \sum_{i=0}^{n-1} g(a_i)\times r(a_i) \quad (4)$$

Pixels close to the central pixel a0 in both space and intensity contribute more than those further away in space and intensity.

The bilateral filter is controlled by two parameters: σ_r and σ_d .

• As the range parameter σ_d increases, the bilateral filter becomes closer to Gaussian blur because the range Gaussian is flatter i.e., almost a constant over the intensity interval covered by the image.

• Increasing the spatial parameter σ_r smooths larger features.

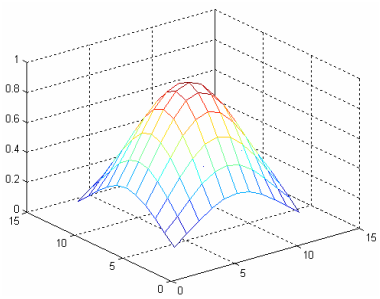

Fig: 1 spatial Gaussian kernel of 11×11 with σ = 3.76

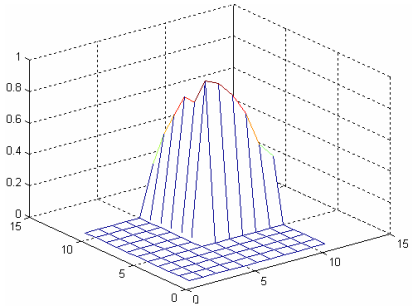

Fig: 2 spatial representation of the normalization constant k for a center pixel

The bilateral filter has several qualities that explain its success:

• Its formulation is simple: each pixel is replaced by an average of its neighbors. This aspect is important because it makes it easy to acquire intuition about its behavior, to adapt it to application-specific requirements, and to implement it.

• It depends only on two parameters that indicate the size and contrast of the features to preserve.

• It can be used in a non-iterative manner. This makes the parameters easy to set since their effect is not cumulative over several iterations.

3. OPTICAL FLOW IN MOTION ANALYSIS

Detecting changes in image sequences is of significant interest due to a large number of applications in several disciplines. Video surveillance is among the important applications, which require reliable detection of changes in the scene. There are several different approaches for such a detection problem. Frequently used techniques for moving object detection are background subtraction, statistical methods, temporal differencing and optical flow.

Background subtraction is particularly a commonly used technique for motion segmentation in static scenes. It attempts to detect moving regions by subtracting the current image pixel-by-pixel from a reference background image that is created by averaging images over time in an initialization period. The pixels where the difference is above a threshold is classified as foreground. The statistical

methods are mainly inspired by the background subtraction methods in terms of keeping and dynamically updating statistics of the pixels that belong to the background image process. Foreground pixels are identified by comparing each pixel's statistics with that of the background model. The detection performance of temporal differencing is usually quite poor in real-life surveillance applications.

Optical flow can be used for a large variety of motions-moving observer and static objects, static observer and moving objects, or both moving.

Motion, as it appears in dynamic images, is uaually some combination of four basic elements [5]:

- Translation at constant distance from the observer.
- Translation in depth relative to the observer.
- Rotation at constant distance about the view axis.
- Rotation of a planar object perpendicular to the view axis.

Optical-flow based motion analysis can recognize these basic elements by applying a few relatively simple operators to the flow. Motion form recognition is based on the following facts:

- Translation at constant distance is represented as a set of parallel motion vectors.
- Translation in depth forms a set of vectors having a common focus of expansion.
- Rotation at constant distance results in a set of concentric motion vectors.
- Rotation perpendicular to the view axis forms one or more sets of vectors starting from straight line segments.

These facts are shown in the Fig. 3.

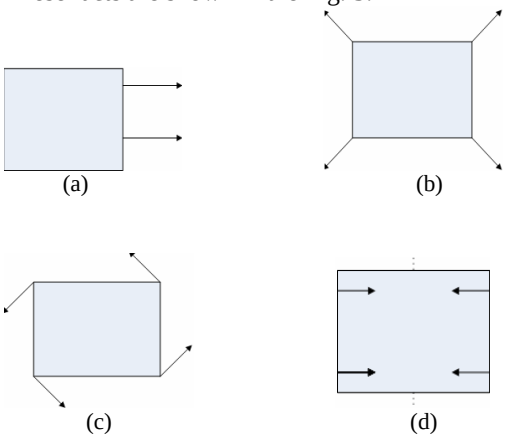

Fig : 3 Motion form recognition.
(a) translation at constant distance (b) translation in depth
(c) rotation at constant distance (d) planar object rotation perpendicular to the view axis.

3.1 Optical flow computation

Optical flow reflects the image changes due to motion during a time interval *dt* and the optical flow field is the velocity field that represents the 3 dimensional motion of the object points across a two dimensional image. Optical flow is an abstraction typical of the kind that computational methods are trying to achieve. Optical flow computation is based on two assumptions:

1. The observed brightness of any object point is constant over time.
2. Nearby points in the image plane move in a similar manner (the velocity smoothness constraint).

Suppose having a continuous image; *f(x, y, t)* refers to the gray-level of *(x, y)* at time t. Representing dynamic image as a function of position and time permits it to be expressed as a Taylor series:

$$f(x+dx, y+dy, t+dt) = f(x,y,t) + f_x dx + f_y dy + f_t dt + O(\partial^2)$$

(5)

where, f_x , f_y, f_l denote the partial derivatives of *f*. If the immediate neighbourhood of *(x, y)* is translated some small distance *(dx, dy)* during the interval *dt*, the higher-order terms in equation vanish and

$$-f_t = f_x \frac{dx}{dt} + f_y \frac{dy}{dt} \qquad (6)$$

The motion velocity can then be estimated as

$$-f_t = f_x u + f_y v = \nabla f \times c \qquad (7)$$

where, $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$, $c = \left(\frac{dx}{dt}, \frac{dy}{dt}\right) = (u, v)$ and ∇f is a two-dimensional image gradient [5].

To estimate motion correctly, the motions need to be small. However, if the pixel correspondence cannot be found in a small neighborhood, we need to reduce the size of the image so that the bigger motion now becomes a smaller one.

Consequently, we employ a hierarchical intensity-based technique to efficiently distinguish foreground regions from the background. The pyramids at each level are created by applying the Gaussian filter to the images and then taking alternative rows and columns.

An iterative implementation of the Lucas-Kanade optical flow computation provides objects having motion.

4. IMAGE PYRAMID REPRESENTATION

Let us define the pyramid representation of a generic image I of size $n_x \times n_y$ .Let $I^0 = I$ be the "zeroth" level image. This image is essentially the highest resolution image (the raw image). The image width and height at that level are defined as $n_x^0 = n_x$ and $n_y^0 = n_y$ .The pyramid representation is then built in a recursive fashion: compute I^1 from I^0 , and then compute I^2 form I^1 and so on... Let $L = 1, 2, 3, ...$ be a generic pyramidal level, and let I^{L-1} be the image at level $L-1$. Denote n_x^{L-1} and n_y^{L-1} the width and height of I^{L-1}.The image I^{L-1} is then defined as

$$I^L(x,y) = \frac{1}{4} I^{L-1}(2x, 2y) +$$

$$\frac{1}{8}\left(I^{L-1}(2x-1,2y)+I^{L-1}(2x+1,2y)+I^{L-1}(2x,2y-1)+I^{L-1}(2x,2y+1)\right)+$$
$$\frac{1}{16}\left(I^{L-1}(2x-1,2y-1)+I^{L-1}(2x+1,2y+1)+I^{L-1}(2x-1,2y+1)+I^{L-1}(2x+1,2y+1)\right) \quad (8)$$

For simplicity in the notation, let us define dummy image values one pixel around the image I^{L-1} (for $0 \le x \le n_x^{L-1}-1$ and $0 \le y \le n_y^{L-1}-1$) :

$$I^{L-1}(-1,y) \doteq I^{L-1}(0,y),$$
$$I^{L-1}(x,-1) \doteq I^{L-1}(x,0),$$
$$I^{L-1}(n_x^{L-1},y) \doteq I^{L-1}(n_x^{L-1}-1,y),$$
$$I^{L-1}(x,n_y^{L-1}) \doteq I^{L-1}(x,n_y^{L-1}-1),$$
$$I^{L-1}(n_x^{L-1},n_y^{L-1}) \doteq I^{L-1}(n_x^{L-1}-1,n_y^{L-1}-1).$$

Then, equation (4) is only defined for values of x and y such that $0 \le 2x \le n_x^{L-1}-1$ and $0 \le 2y \le n_y^{L-1}-1$. Therefore, the width n_x^L and height n_y^L of I^L are the largest integers that satisfy the two conditions:

$$n_x^L \le \frac{n_x^{L-1}+1}{2} \quad (9)$$

$$n_y^L \le \frac{n_y^{L-1}+1}{2} \quad (10)$$

Equations (8), (9) and (10) are used to construct recursively the pyramidal representations of the two images I and J: $\{I^L\}_{L=0,\ldots,Lm}$ and $\{J^L\}_{L=0,\ldots,Lm}$. The value Lm is the height of the pyramid (picked heuristically). Practical values of Lm are 2, 3, 4. For typical image sizes, it makes no sense to go above a level 4 [4].
A multiresolution pyramid is depicted in Fig.4.

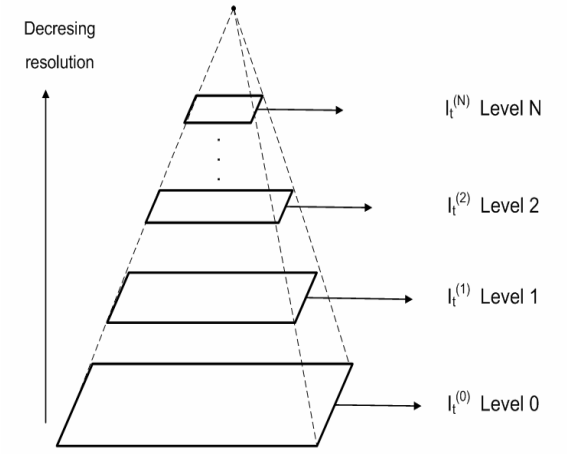

Fig : 4 Multiresolution pyramid [3]

5. PYRAMIDAL LUCAS KANADE ALGORITHM

Let u be a point on image I. Find its corresponding location v on image J [2].

- Build pyramid representations of I and J: $\{I^L\}_{L=0,\ldots,L_m}$ and $\{J^L\}_{L=0,\ldots,L_m}$
- Initialization of pyramid guess: $g^{Lm} = [g_x^{Lm} g_x^{Lm}]^T = [0\ 0]^T$

For L=Lm down to 0 with step of -1

- Location of point u on image I^L: $u^L = [p_x\ p_y]^T = u/2^L$
- Derivative of I^L with respect to x: $I_x(x,y) = \frac{I^L(x+1,y)-I^L(x-1,y)}{2}$
- Derivative of I^L with respect to y: $I_y(x,y) = \frac{I^L(x,y+1)-I^L(x,y-1)}{2}$
- Spatial gradient matrix: $G = \sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y}\begin{bmatrix} I_x^2(x,y) & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y^2(x,y)\end{bmatrix}$
- Initialization of iterative L-K: $\bar{v}^0 = [0\ 0]^T$

For k=1 to K with step of 1

- Image Difference: $\delta I_k(x,y) = I^L(x,y) - J^L(x+g_x^L+v_x^{k-1}, y+g_y^L+v_y^{k-1})$
- Image mismatch vector: $\bar{b}_k = \sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y}\begin{bmatrix}\delta I_k(x,y)I_x(x,y) \\ \delta I_k(x,y)I_y(x,y)\end{bmatrix}$
- Optical flow (Lucas –Kanade): $\bar{\eta}^k = G^{-1}\bar{b}_k$
- Guess for next iteration: $\bar{v}^k = \bar{v}^{k-1} + \bar{\eta}^k$

End of for-loop on k

- Final optical flow at level L: $d^L = \bar{v}^k$
- Guess for next level L-1: $g^{L-1} = [g_x^{L-1} g_y^{L-1}]^T = 2(g^L + d^L)$

End of for-loop on L

- Final optical flow vector: $d = g^0 + d^0$
- Location of point on J: v = u+d
- Get the magnitude value of a vector v at each pixel.
- the magnitude does not lie between the two thresholds then define that pixel as moving object's pixel, where two thresholds are defined as below: $th_1 = mean + c*\sigma$ and $th_2 = mean - c*\sigma$

Where, *mean* equal to global mean of the magnitude and σ is the standard deviation of the magnitude and c is a constant.

As the camera motion is removed from both frames it means that we are having frames which are having same background. If the object is not moving then it will have motion vector magnitude value in the close proximity of its mean.The graphical representation of Lucas-Kanade with pyramid is shown in Fig. 5.

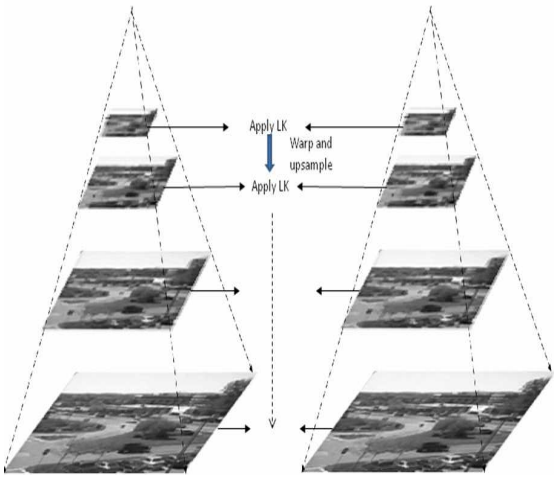

Fig : 5 Lucas-Kanade with pyramid

6. RESULTS

Results obtained with the bilateral filter of size 19×19 and σ_d=11.3526 for the above discussed method are compared with the results obtained without using the filter are shown below. Fig 6(b) and 7(b) shows detection of moving objects without bilateral filter. Here, leaves movement is also detected as moving object. This is overcome by the application of bilateral filter as shown in Fig 6(a) and 7(a). Moving object is detected with 5 iterations at each 5 levels of Gaussian pyramid after preprocessing. Gaussian mask used at each level of optical flow is [0.05 0.25 0.4 0.25 0.05]. More results using bilateral filter are shown in Fig. 8,9 and 10 where, it can be perceived that temporal movement is not detected.

7. CONCLUSION

Though optical flow based moving object detection is robust to intensity changes, it fails for temporal movements like swinging leaves in a video sequence. Detection of temporal movement like swinging leaves can be removed by using bilateral filter as a preprocessing block.

(a)

(b)

Fig: 6 moving object detection (a) with filter (b) without filter in frame-10 of a video sequence

(a)

(b)

Fig: 7 moving object detection (a) with filter (b) without filter in frame-220 of same video sequence

(a)

(b)

Fig: 8 moving object detection (a) without filter (b) with filter in frame-105 of a video sequence taken with handheld camera

(a)

(b)

Fig: 9 moving object detection (a) with filter (b) without filter in frame-120 of same video sequence

(a)

(b)

Fig: 10 moving object detection (a) with filter (b) without filter in frame-220 of same video sequence

REFERENCES

[1] C. Tomasi, R. Manduchi “Bilateral Filtering for Gray and Color Images”, Porceeding of IEEE 6th International Conference on Computer Vision, p.p. 839-847,1998.

[2] Qimei Hu, Xiangjian He, Jun Zhou “Multi-scale edge detection with Bilateral filtering in spiral architecture”, Univesity of Technology, Sydney, Australia.

[3] Chiou-Ting Hsu, Yu-Chun Tsan “Mosaics of video sequences with moving objects,” Signal Processing: Image Communication, vol. 19, pp. 81-98, and 2004.

[4] Jean-Yves Bouguet “Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm” Intel Corporation Microprocessor Research Labs.

[5] M Sonka,V Hlavac,R Boyle. “Digital Image Processing and Computer Vision,” Nelson Engineering.