International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 17 (2018) pp. 13288-13291

Motion Estimation Algorithm Based on Imaging Subtraction

Nicolas Ortiz Valencia

Master Student, Department of Mechatronics Engineering, Militar Nueva Granada University, Bogotá, Colombia.

Robinson Jimenez-Moreno

Professor, Department of Mechatronics Engineering, Militar Nueva Granada University, Bogotá, Colombia.

Ruben Dario Hernandez

Professor, Department of Mechatronics Engineering, Militar Nueva Granada University, Bogotá, Colombia.

Abstract

This paper presents a novel three frame motion estimation algorithm that uses information obtained from a single imaging subtraction to estimate the motion of a single object on a motionless image scene. The first step is to obtain two sets of new and old pixels in motion by subtracting three frames. Then, from defining the centroids of the given sets is derived and estimated displacement of the studied object. Evaluation was made in comparison with three optical ow estimation algorithms. Results shows low computational cost and higher precision.

Keywords: Three frame algorithm, motion estimation, imaging subtraction, motionless image scene.

INTRODUCTION

Motion is an intrinsic property of the world and an integral part of our visual experience [1], in this sense, motion has to be also an important information source for machines on certain tasks as object tracking [2], Motion information is obtained, traditionally, using optical flow estimation methods, unfortunately, many of these methods focuses on getting motion information by using variational models to estimate the currently position of every single pixel on the image scene, but not directly for a whole moving object [9].

In this document it is presented a novel motion estimation technique focused on getting motion data of a single object in motion. Unlike some optical flow estimation methods this technique does not try to obtain a good match, instead of this, it uses imaging subtraction between three frames to obtain a current and an old point to describe the vector of movement of a whole object, taking advantage of the fact that, ideally, if two frames are compared, pixels in the same position whose intensities are different have to be in motion if the image scene is motion less. It is briefly described also an applicable real model that takes into account light conditions to obtain motion data. Then, using the validation experiment methodology described on [2], it is compared the SB algorithm (proposed algorithm) accuracy with the pyramidal Lucas Kanade [4] [5] (LK), Farneback [6] (FB) and Phase Correlation [7] (PC) Algorithms. Finally, computational cost comparison was made using a 300 frames test to obtain the processing time average of each algorithm.

MOTION ESTIMATION ALGORTIHM SB

In this section are de_ned the basic sets needed for motion estimation taking into account the mathematical implications of doing an imaging subtractionand then it is described how to obtain a movement vector using this sets.

Set of Pixels in Motion

Let $I_1(i,j)$ and $I_2(i,j)$ be the intensities of the pixels in the position (i,j) of the _rst and second frame respectively. If it is considered that there is only one object in motion and camera is motionless, then, being O the set of pixel positions of the moving object and $x=(i,j)$any pixel position in the second frame:

$$x \in O \leftrightarrow I_2(i,j) - I_1(i,j) \neq 0 \qquad (1)$$

This means, that those pixels whose intensity changes from frame to frame are members of the moving object. Then O can be de_ned in a general way as follows:

$$O = \{(i,j) | |I_2(i,j) - I_1(i,j)| > 0\} \qquad (2)$$

However, this definition does not correspond to an applicable real model because light conditions can change from frame to frame causing that, for some pixels that do not belong to the moving object, the statement $|I_2(i,j) - I_1(i,j)| > 0$is true. Due to this is necessary to define a new parameter δ as a low threshold value assuming that light variations are low too. Then we redefine O as:

$$O = \{(i,j) | |I_2(i,j) - I_1(i,j)| > \delta\} \qquad (3)$$

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 17 (2018) pp. 13288-13291

The Centroid of O

Note that O is a set which members are the pixel positions of the studied object. Thus it is defined O^c as the centroid of O such that:

$$O^c = \frac{1}{n}\sum_{1}^{n} O_k \ , O_k \ \in O \qquad (4)$$

Where O_k is the kth element of the set O and n is the set length.

Motion Estimation

For motion estimation is necessary the inclusion of a third frame I_3 such that, if O_σ is the set of pixel positions obtained from the subtraction between I_2 and I_1 and O_η is the set of pixel positions obtained from the subtraction between I_3 and I_2 , the estimated movement vector $\vec{v}$ of the object from I_1 to I_3 is given as:

$$\vec{v} = \overrightarrow{O_\sigma{}^c O_\eta{}^c} \qquad (5)$$

If $O_\sigma{}^c = (i_\sigma, j_\sigma)$ and $O_\eta{}^c = (i_\eta, j_\eta)$ then:

$$|\vec{v}| = \sqrt{(i_\sigma - i_\eta)^2 + (j_\sigma - j_\eta)^2} \qquad (6)$$

And

$$\measuredangle\vec{v} = arctan\left(\frac{j_\sigma - j_\eta}{i_\sigma - i_\eta}\right) \qquad (7)$$

Note that $|\vec{v}|$ is the estimated displacement of the object.

The pseudocode of the implemented algorithm is shown on Table 1 and a graphical scheme of the algorithm is shown in Figure 1.

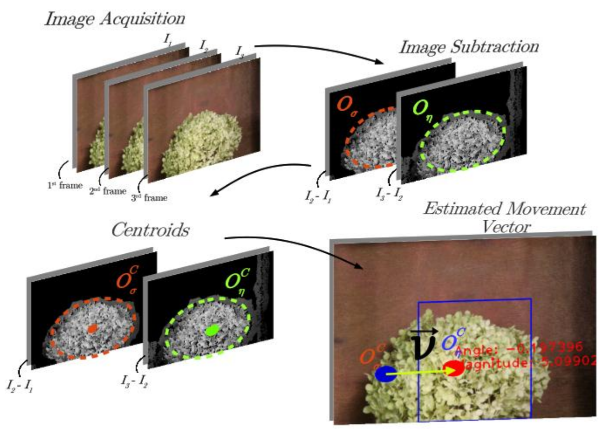


Figure 1. Graphical Scheme of the Algorithm (The Images were taken from [3]).

Table 1. Algorithm SB: Pseudocode

Algorithm
1. Image Acquisition: I_1, I_2, I_3
2. Get $I_2 - I_1$ and $I_3 - I_2$
3. Thresholding of $I_2 - I_1$ and $I_3 - I_2$ using δ
4. Obtain O_σ and O_η (Equation 3)
5. Obtain Centroids: $O_\sigma{}^c$ and $O_\eta{}^c$ (Equation 4)
6. Estimate $\vec{v}$ (Equations 5, 6, 7)

RESULTS

Accuracy

As it was mentioned, algorithm evaluation was made using the validation experiment proposed on [2] with $\delta = 10$ (Equation 3), which consist of creating a set of three frames sequences with a constant displacement to calculate the error between the movement vector estimated by the algorithm and the preset movement vector. In this paper, it is evaluated the algorithms using preset displacements of 10, 20, 30, 40, 50 and 65 pixels. Figure 2 show the validation frames.

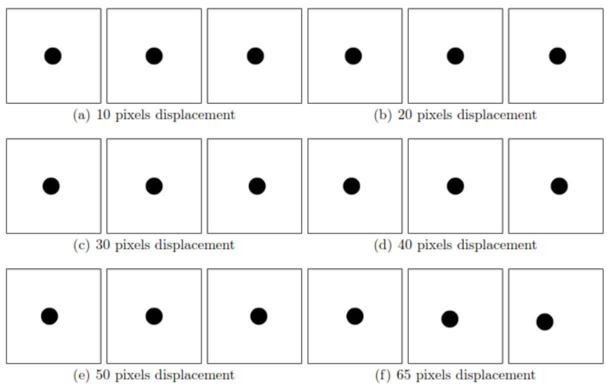


Figure 2. Validation Frames

Figures 3, 4, 5, 6, 7 and 8 show, in a graphic manner, the estimated displacement and on tables 2, 3, 4, 5, 6 and 7 are contained the calculated errors for each set of validation frames.

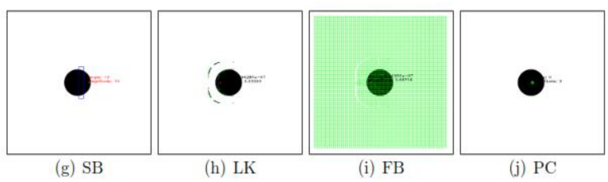
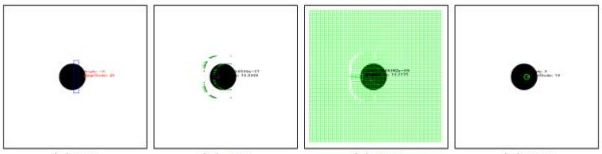


Figure 3. Displacement field validation for each algorithm for a 10 pixels Displacement.

Table 2. Calculated Errors for a 10 pixels displacement

Algorithm	Calculated displacement (pixels)	Error %
SB	10.00	0.000
LK	9.999	0.010
FB	9.647	3.530
PC	9.000	10.00

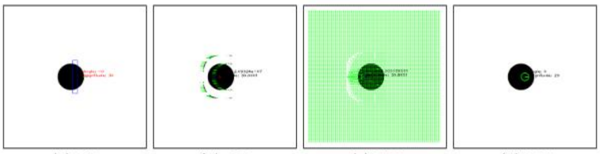


(a) SB (b) LK (c) FB (d) PC

Figure 4. Displacement field validation for each algorithm for a 20 pixels Displacement

Table 3. Calculated Errors for a 20 pixels displacement

Algorithm	Calculated displacement (pixels)	Error %
SB	20.000	0.000
LK	19.999	0.005
FB	19.517	2.415
PC	19.000	5.000

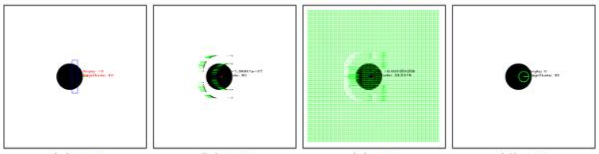


(a) SB (b) LK (c) FB (d) PC

Figure 4. Displacement field validation for each algorithm for a 30 pixels Displacement

Table 4. Calculated Errors for a 30 pixels displacement

Algorithm	Calculated displacement (pixels)	Error %
SB	30.000	0.000
LK	30.000	0.000
FB	30.865	2.883
PC	29.000	3.333

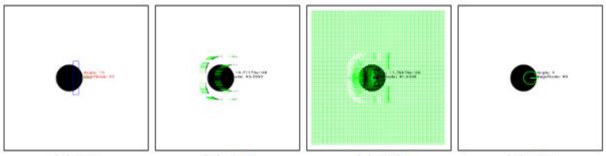


(a) SB (b) LK (c) FB (d) PC

Figure 5. Displacement field validation for each algorithm for a 40 pixels Displacement

Table 5. Calculated Errors for a 40 pixels displacement

Algorithm	Calculated displacement (pixels)	Error %
SB	40.000	0.000
LK	40.000	0.000
FB	33.591	16.02
PC	39.000	2.500

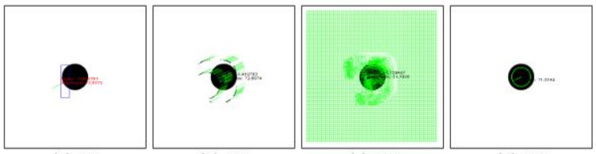


(a) SB (b) LK (c) FB (d) PC

Figure 6. Displacement field validation for each algorithm for a 50 pixels Displacement

Table 6. Calculated Errors for a 50 pixels displacement

Algorithm	Calculated displacement (pixels)	Error %
SB	50.000	0.000
LK	49.999	0.002
FB	41.033	17.93
PC	49.000	2.000



(a) SB (b) LK (c) FB (d) PC

Figure 7. Displacement field validation for each algorithm for a 65 pixels Displacement

Table 7. Calculated Errors for a 65 pixels displacement

Algorithm	Calculated displacement (pixels)	Error %
SB	72.897	12.149
LK	72.897	12.149
FB	51.789	20.324
PC	71.554	10.083

Validation experiment was stopped here, because for a 65 pixels displacement the errors obtained, it was considered, are in the maximum admissible range

Computational Cost

Computational cost was evaluated on a 300 frame sequence on an Intel® Core i5-5200U processor. Average processing times are enclosed in the table below:

Table 8. Average processing time for each algortithm

Algorithm	Average processing time (ms)
SB	36.920
LK	45.833
FB	107.932
PC	31.620

CONCLUSIONS

Experimental Results shows that the proposed algorithm is more accurate than the LK, FB and PC algorithms using the methodology presented by Ortiz et al in [2]. Additionally, it was observed that from tables 2 to 6 the estimated error was always 0%. It also presents low computational cost in comparison with the LK and FB algorithms. Nevertheless, more accuracy and speed tests are needed, since the other algorithms shows very accurate performances too. However, in this first development stage, algorithm performance proofs that it can compete with one the most used optical ow algorithm, LK [8]. Furthermore, the proposed algorithm is able to manage with a whole moving object instead of single motion pixel information.

ACKNOWLEDGMENTS

The research for this paper was supported by Davinci research Group of Nueva Granada Military University.

REFERENCES

[1] David J. Fleet and Yair Weiss," Optical Flow Estimation. Mathematical models for Computer Vision: The Handbook," 2005. p. 239-257. ISBN 0387263713.

[2] N. O. Valencia, L. F. V. Londoño, M. A. Jinete and R. Jimenez, "Movement detection for object tracking applied to the InMoov robot head," 2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA), Bucaramanga, 2016, pp. 1-6. doi: 10.1109/STSIVA.2016.7743328.

[3] Simon Baker, Daniel Scharstein, J. P. Lewis Stefan Roth, Michael J. Black and Richard Szeliski, "A Database and Evaluation Methodology for Optical Flow," International Journal of Computer Vision, 2011. Vol 92. p. 131.

[4] Bruce D. Lucas and Takeo Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," Proceedings of Imaging Understanding Workshop, 1981. p. 121-130.

[5] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm," Intel Corporation, Microprocessor Research Labs, 2000.

[6] Gunnar Farneback, "Two-Frame Motion Estimation Based on Polynomial Expansion," Image Analysis: 13th Scandinavian Conference, 2003. pp. 363-370. doi: 10.1007/3-540-45103-X 50.

[7] E. De Castro and C. Morandi, "Registration of Translated and Rotated Images Using Finite Fourier Transforms," IEEE Transactions on pattern analysis and machine intelligence, 1987.

[8] Z. Yan and X. Xiang, "Scene Flow Estimation: A Survey," pp. 151, 2016. [Online]. Available: http://arxiv.org/abs/1612.02590.

[9] M. Burger, H. Dirks, and L. Frerking, "On Optical Flow Models for Variational Motion Estimation," pp. 127, 2015. [Online]. Available: http://arxiv.org/abs/ 1512.00298.