Motion Detection Based on Optical Flow Technique with Moving Camera

Tiago Ruas da Gama Cerqueira
Computer Vision, DCC, UFMG
Belo Horizonte, Brazil

***Abstract*—Motion detection techniques are really useful to improve autonomy of surveillance systems. One of the techniques is known as optical flow and it's found in a range of implementations that focus on specific conditions.**

This paper proposes a solution to the use of optical flow in the condition of continuous camera motion, still being able to detect motion apart from the already existing camera movement. That way it will be possible to filter camera motion from other motion at the scene and extract only what is interesting to a surveillance system.

***Keywords*-optical flow; motion detection; camera motion;**

I. Introduction

On my undergraduate final project a map input is studied to be partitioned and give a series of regions that must be surveyed. One of the parameters is a radius representing the farthest point that a guard can view in front of him. That way we assume a circular range that a guard can see.

The idea here is to guard this area by detecting motion, and a interesting way to do that is by using the optical flow technique, which is the 2-D velocity field induced in an image due to the projection of moving objects onto the image plane. An optical flow field shows the velocity of each pixel in the image. [1].

However optical flow is not limited only to object movements in the scene, and can also be perceived by the observer's motion [2]. Since we expect to use an Unmanned Aerial Vehicle (UAV) as guard we have to consider that it will not be perfectly stable while using it's camera and also, just as a human, an UAV has limited field of view, and to successfully watch the entire circle around him it's necessary to rotate to gradually cover the entire desired region.

It is possible to perform small rotations based on the division of 360 degrees by the field of view and for every stop perform a motion detection maintaining the camera still. However it would require more time to scan an region at individual portions.

Performing a full rotation and scanning at the same time would be faster, but the motion from the camera would always trigger the motion threshold, giving unwanted results and detecting the kind of motion that is not interesting to the surveillance.

To properly detect motion in a scene we must separate camera motion from real motion and this paper will focus on techniques to improve the optical flow performance in a situation of constant camera motion to be used on UAVs, thus filtering only what is really interesting to a motion detection surveillance system.

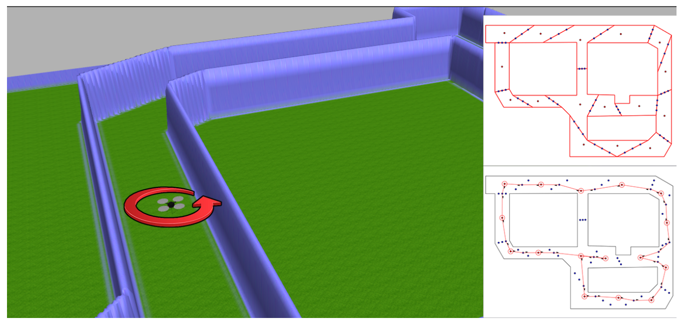


Fig. 1. Image showing a resulting map that contains regions of interest. The UAV must follow the circuit and for every red point it must survey the region using it's camera.

There are techniques that separate the background and foreground effectively and detect the object in motion accurately [3], but it will behave properly in continuous camera motion? That is what will be tested and modified to fit this paper's objective.

A. Related work

A number of optical flow techniques like "Lucas and Kanade", "Nagel", "Horn and Schunck" and others have been described and tested on [4] giving reference to a better choice to be used in this problem.

Also [3] proposes a real time motion detection algorithm that is based on the integration of accumulative optical flow and double background filtering method (long-term background and short-term background), that can be helpful to reduce the influence of background motion.

For moving camera, [5] proposes a clustering approximation to classify motion, but the techniques were found to only work on certain types of movements, encouraging this paper to follow the proposition on [6], that relies on homography-based motion detection.

With all that being said, the objective of this paper is to develop and test an optical flow technique tweaked to properly function in a camera movement setting. As metric, a number will be returned to represent the level of action detected on a sweep of the region and we must test if this number makes sense having it compared with a human description of the real movement that happened in the scene.

B. Technique overview

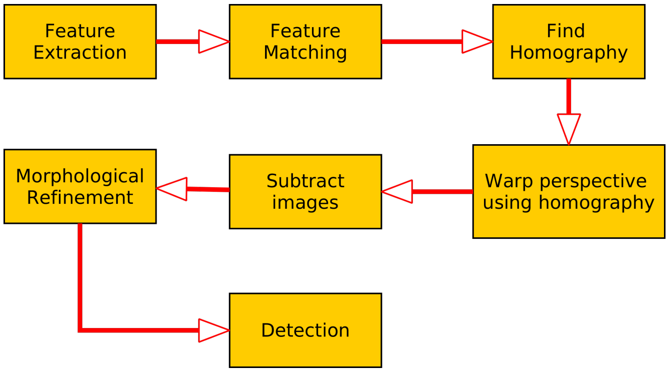


Fig. 2. Technique Scheme.

In order to achieve our goal, we'll utilize OpenCV to calculate the optical flow between two frames, representing the feature extraction and matching process. By having two sets of corresponding points we'll be able to find the perspective transformation of the background plane between the frames, transforming the perspective of the previous frame to match the current frame. That way it is possible to subtract the current frame from the transformed one and find residual pixels that correspond to the areas that were not bound to the same movement as the background. Applying morphological transformations we're then able to create a mask to consider only optical flow vectors in areas that are moving differently from the background, what means that the motion from the camera is now filtered.

II. Implementation

We utilize OpenCV routines to accomplish our methodology. By using Good Features to Track we are able to extract features and using the implementation of the Pyramidal Lucas-Kanade we can match those features between two frames. At this point, we have the normal visualization of the Lucas-Kanade Optical Flow and it contains both motion from a interest subject and motion caused by the rotation of the camera.

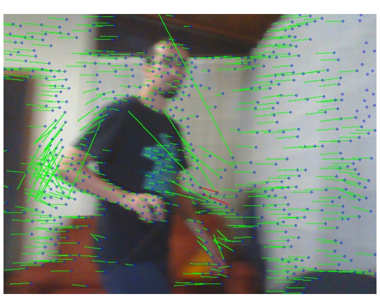
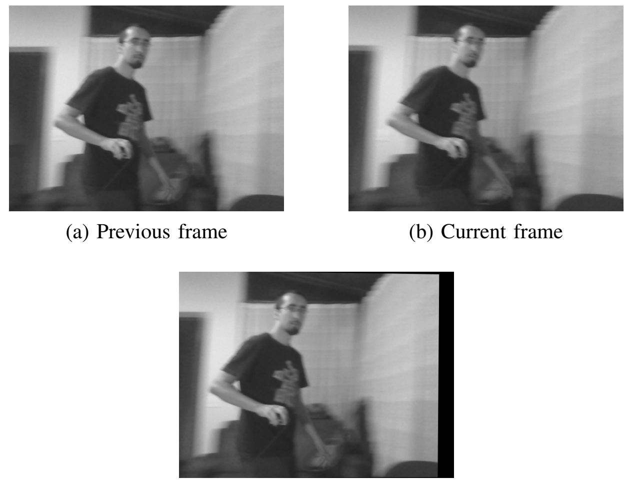


Fig. 3. Optical Flow visualization.

To find the homography between the previous and the current frame, we utilize the two sets of points returned by the optical flow and pass it to a procedure called findHomography, using RANSAC to remove outliers. In that case, we assume that most of the points are correspondent to the background plane, helping RANSAC to better remove outliers.

Once the homography is found, we apply it to the previous frame by using warpPerspective, and it is warped to match the current frame. Since the background plane was supposedly used as the homography base, everything that doesn't match the background movement will be distorted by the warp.



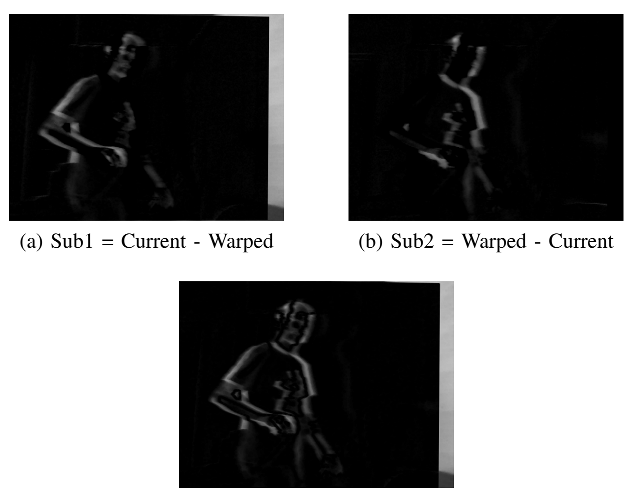
(a) Previous frame

(b) Current frame

(c) Previous frame warped by the homography

Fig. 4. Frames

Now we subtract the current frame from the warped frame and find the residual pixels of the transformation. Since the subtraction will create residual pixels on the edges of where something is moving differently from the background, we need to subtract in both ways to get all the edges and in the end add the two subtractions to get one final image representing residual pixels.



(a) Sub1 = Current - Warped

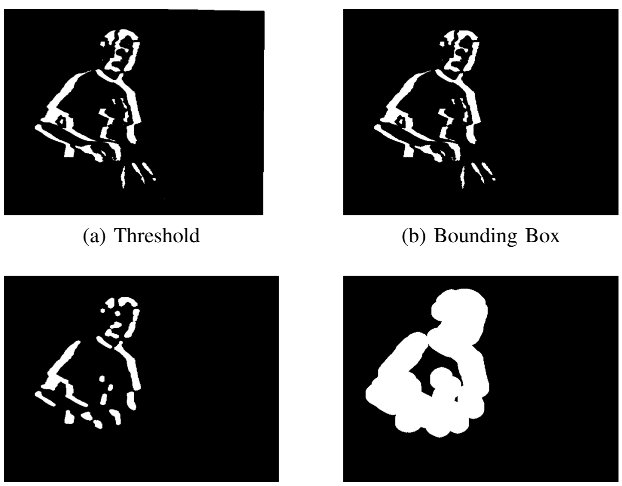
(b) Sub2 = Warped - Current

(c) Sub = Sub1 + Sub2

Fig. 5. Subtraction

For the morphological refinement we first apply a threshold to binarize the subtraction. Then a bounding box is applied to remove the residual pixels that are just caused by the warping. This method limits our process to identify movement only inside the bounding box.

Now we have a well defined mask, but it still have some outliers and is not robust enough, clearing only the edges as mentioned before. To fix that, we apply an opening morphological process, that helps to eliminate some outliers and opens the mask by eroding and then dilating the image. Finally we dilate the image lots more, to fill as much as possible the inside of the mask we are trying to produce.



(a) Threshold

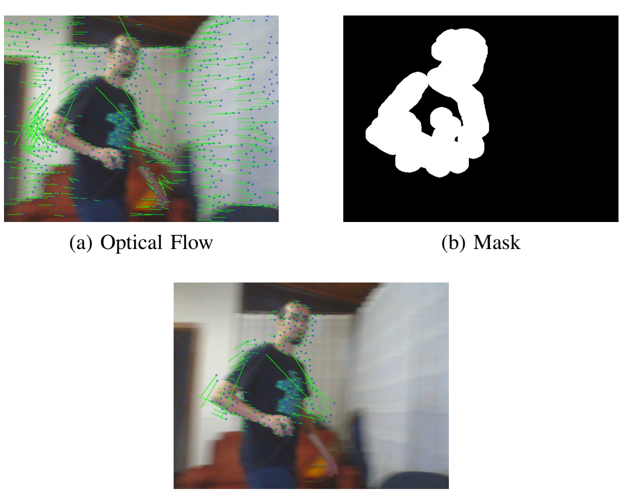
(b) Bounding Box

(c) Opening

(d) Dilation

Fig. 6. Morphological Refinement

This process now give us a mask to be used as a filter of optical flow vectors, considering only vectors that finish (we are at the current frame) at the mask.



(a) Optical Flow

(b) Mask

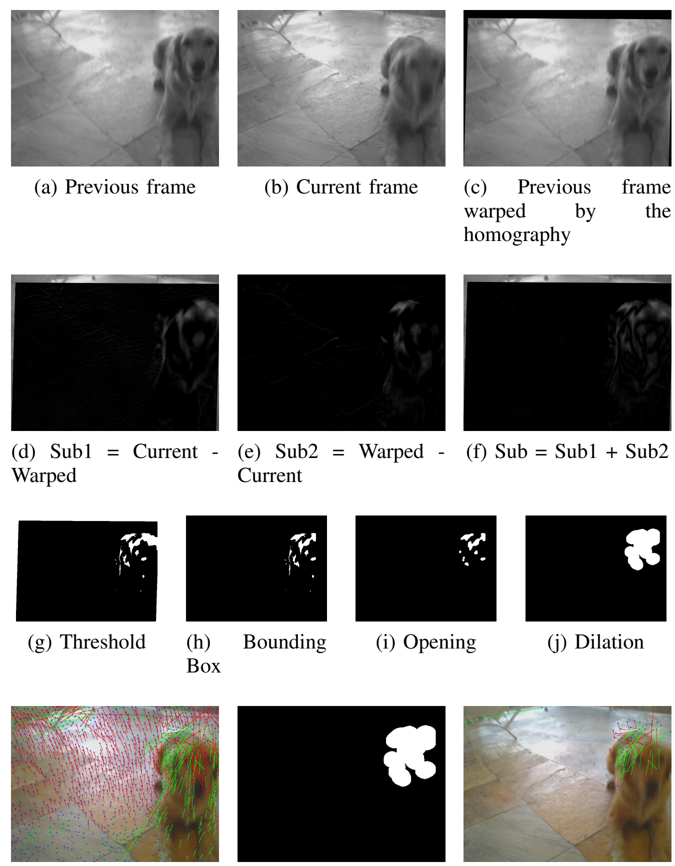
(c) Filtered Optical Flow

Fig. 7. Final Result

III. Experiments

Here are some experiments showing the process for different types of camera movement and the results.

First experiment: The first experiment shows how this technique behaves under general camera movement. The camera had a forward and upward motion, and it was able to capture only the dog's head movement.



(a) Previous frame

(b) Current frame

(c) Previous frame warped by the homography

(d) Sub1 = Current - Warped

(e) Sub2 = Warped - Current

(f) Sub = Sub1 + Sub2

(g) Threshold

(h) Bounding Box

(i) Opening

(j) Dilation

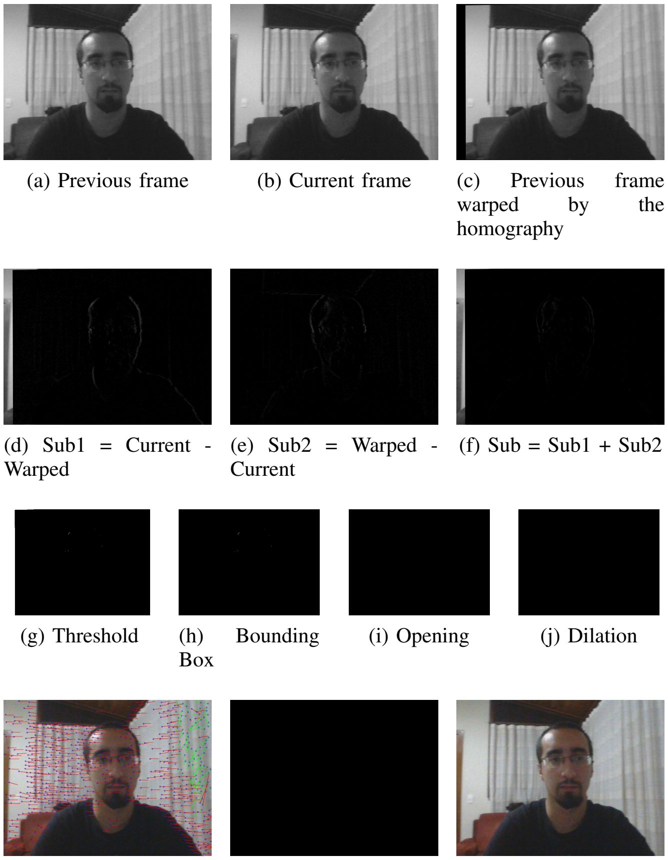
(k) Optical Flow

(l) Mask

(m) Filtered Optical Flow

Fig. 8. First experiment

Second experiment: The second experiment shows how static images creates a blank mask, filtering every motion that was created by the camera.



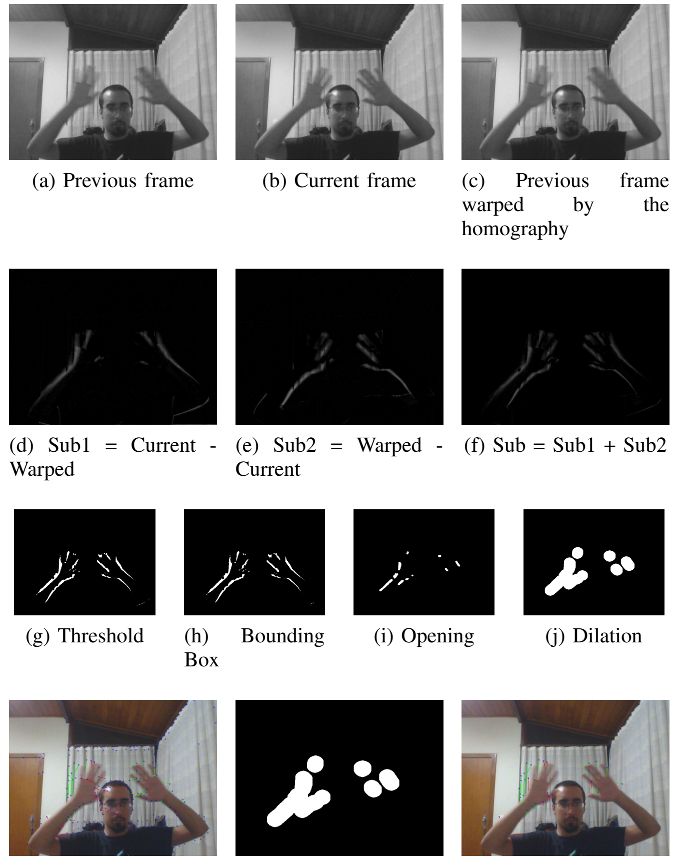
(a) Previous frame (b) Current frame (c) Previous frame warped by the homography

(d) Sub1 = Current - Warped (e) Sub2 = Warped - Current (f) Sub = Sub1 + Sub2

(g) Threshold (h) Bounding Box (i) Opening (j) Dilation

(k) Optical Flow (l) Mask (m) Filtered Optical Flow

Fig. 9. Second experiment



(a) Previous frame (b) Current frame (c) Previous frame warped by the homography

(d) Sub1 = Current - Warped (e) Sub2 = Warped - Current (f) Sub = Sub1 + Sub2

(g) Threshold (h) Bounding Box (i) Opening (j) Dilation

(k) Optical Flow (l) Mask (m) Filtered Optical Flow

Fig. 10. Third experiment

Third experiment: The last experiment shows how even on non rotating camera, the homography is found and is able to warp the scene enough to detect what is really moving.

IV. Conclusion

In this paper, the technique discussed is efficient to point interest regions on optical flow scenarios that involves movement of the camera.

The convex hull of the residual pixels was pointed out to be used as a limiting region instead of the morphological refinement, however it would not distinguish fingers on a hand for example since it would create a full region around the residual pixels. Also it would require to know how many objects are being detected, or it would return a convex hull that contains both regions and also the background in between.

This technique is then considered the best option to represent the motion detection system we are trying to achieve.

References

[1] L. Fang, Z. Meng, C. Chen, and Q. Hui, "Smart motion detection surveillance system," in *Education Technology and Computer, 2009. ICETC'09. International Conference on*. IEEE, 2009, pp. 171–175.

[2] W. H. Warren and D. J. Hannon, "Direction of self-motion is perceived from optical flow," *Nature*, vol. 336, no. 6195, pp. 162–163, 1988.

[3] N. Lu, J. Wang, L. Yang, and Q. H. Wu, "Motion detection based on accumulative optical flow and double background filtering." in *World Congress on Engineering*. Citeseer, 2007, pp. 602–607.

[4] J. Barron, D. Fleet, and S. Beauchemin, "Systems and experiment performance of optical flow techniques," *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.

[5] J. Kim, X. Wang, H. Wang, C. Zhu, and D. Kim, "Fast moving object detection with non-stationary background," *Multimedia tools and applications*, vol. 67, no. 1, pp. 311–335, 2013.

[6] W. J. Kim and I.-S. Kweon, "Moving object detection and tracking from moving camera," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*. IEEE, 2011, pp. 758–759.