International Journal of Scientific Research in Information Systems and Engineering Volume 1, Issue 2, December - 2015
ISSN 2380-8128

Estimation of Camera Ego-Motion for Real-Time Computer Vision Applications

Yusuf Sait Erdem[1], Feyza Galip[2], Ibrahim Furkan Ince[3*], Md. Haidar Sharif[4]

Department of Computer Engineering
Gediz University
Seyrek, Izmir 35665 – TURKEY

E-mail[1]: yusuf.erdem2011@ogr.gediz.edu.tr
E-mail[2]: feyza.galip@gediz.edu.tr
E-mail[3]: furkan.ince@gediz.edu.tr
E-mail[4]: md-haidar.sharif@gediz.edu.tr

*Corresponding author: Ibrahim Furkan Ince

Abstract— how can we distinguish the scene motion from the camera motion? In most of the computer vision applications, camera movements avoid true detection of events. For instance, if the camera is trembling due to wind in outdoor environments, it creates high frequency motion in the scene motion as well. To detect flame, we use high frequency motion. If camera trembles, then non-flame regions can also be detected as flame due to high frequency camera motion. Consequently, it is essential to detect the camera motion and avoid event detection (e.g., flame detection) when the camera is moving. In this paper, we have suggested a camera ego-motion estimation algorithm, which uses a feature based approach for motion estimation. Feature selection limits the number of feature points to be tracked and henceforth lowers the computational complexity. Besides, it has a low dependency on structure in the video frame. For video frames with unknown motion fields, displaced frame difference has been used as the criteria for evaluating the algorithm, whereas vector difference between known and estimated displacements has been operated as an error measure for video frames with known motion fields.

Keywords — Camera Ego-Motion Estimation, False Alarm Reduction, Computer Vision, Surveillance Systems.

1 Introduction

THE motion in video frames can be divided into global motion and local motion. Motion induced in the video frames due to camera movement is called global motion, whereas small moving objects in the scene result in local motion. If the moving object is large enough to occupy the complete image, it will produce the same effect as camera movement, resulting in global motion in the video frames. The aim of a motion estimation technique is to assign a motion vector (displacement) to each pixel in a video frame. The choice of a motion estimation approach strongly depends on the target of application. A key issue when designing a motion estimation technique is its degree of efficiency with enough accuracy to serve the purpose of intended application. Difficulties in motion estimation arise from unwanted camera motion, occlusion, noise, lack of image texture, and illumination changes.

Camera based computer vision applications e.g., video object detection, human detection, and tracking mostly require stationary camera to avoid false alarms. Camera shaking due to wind or camera movements e.g., zoom in, zoom out, tilt and pan may cause errors in computer vision algorithms. At the same time, speed of algorithm is an essential factor for real-time applications. In the literature, there exist various algorithms but they are considered as they are not applicable in real-time applications which require 30 frames per second. Consequently, it is required to develop computationally low-cost and highly efficient algorithms for camera zoom/tilt/pan detection. What is assumed that computer vision algorithms will skip their algorithmic steps in case of zoom/tilt/pan is detected. Henceforth, computationally low-cost and highly accurate zoom/tilt/pan detection algorithm is required for most of the computer vision algorithms.

Our aim is to develop a computationally low-cost and accurate zoom/tilt/pan detection algorithm for real-time computer vision applications. With this end, we have suggested a camera ego-motion estimation algorithm. The algorithm uses a feature based approach for motion estimation. Feature selection limits the number of feature points to be tracked and hence lowers the computational complexity. In addition, it has a low dependency on structure in the video frame. For video sequences with unknown motion fields, displaced frame difference was used as the criteria for evaluating the algorithm, whereas vector difference between known and estimated displacements was used as an error measure for video sequences with known motion fields.

Our proposed approach includes the following algorithmic steps: (i) Convert the video frames from RGB scale to gray scale for easier processing; (ii) Apply background subtraction to every frame to extract the foreground objects from the video frame; (iii) Apply frame difference to every consecutive frame to recognize the moving object between frames; (iv) Compare the pixel location of the moving object obtained from frame differencing with the labels generated now, and find out the label in which they exist; (v) Determine the velocity vectors for the motion occurring in the frame; (vi) Calculate the threshold velocity, the velocity which marks the minimum velocity to be perceived as a gesture; (vii) Segment the object of interest from the frame based on the calculated threshold velocity; (viii) Calculate the area of every segmented object and draw a bounding box around it; (ix) Calculate the center of the bounding box to give the coordinates of the detected motion; (x) Estimate the pan, tilt, and zoom of the camera based on the computed coordinates and the range of the camera parameters. The flow chart of the proposed method is shown in Figure 1 as follows:

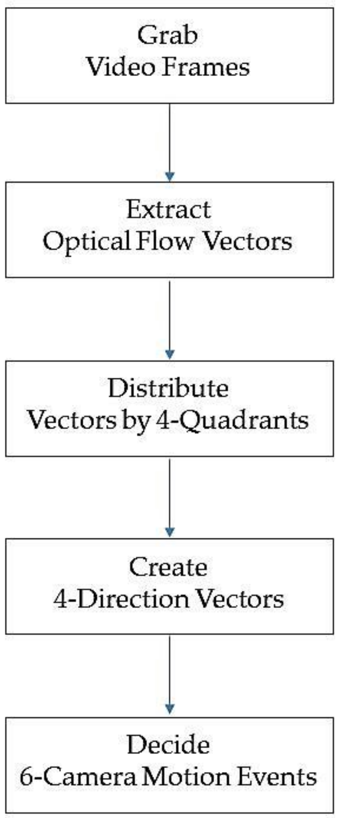


Fig. 1. Flow-Chart of the Proposed Method

Experimental results demonstrate that our proposed algorithm estimates camera ego-motion even for video frames having small object motion. If there is a big moving object near the image center, the algorithm is sometimes fooled by the object motion and considers it as camera motion. The exactitude of the algorithm for static scenes was resulting favorably, and the obtained contraction in computational cost made the trade off well vindicated.

The remaining part of this paper has been organized as follows: Section 2 accords with the most fashionable techniques. Section 3 describes in vivid detail of our proposed framework; Section 4 reports some experimental results; finally, Section 5 presents the conclusion of the work with few glimpses for further study.

2 State-of-the-art Methods

Yuan et al. [1] addressed the problem of ego-motion estimation for a monocular moving camera, which is under arbitrary translation and rotation. They determined the range of each rotational parameter roughly. They searched for the ground truth of each rotational parameter in the corresponding range. Once the ground truth of rotational parameters was determined, the location of the focus of expansion is determined simultaneously, which gave the direction of the camera linear velocity. Ratshidaho et al. [2] approached for ego-motion estimation using time-of-flight camera data. Ego-motion was defined as a process of estimating a camera's pose relative to some initial pose using the camera's image sequence. The time-of-flight camera was characterised with a number error models. These models were used to design several filters that were applied on point cloud data. Iterative closest point is applied on the consecutive range images of the time-of-flight camera to estimate relative pose transform which is used for ego-motion estimation. Ohr et al. [3] presented an image feature based method to estimate ego-motion and relative pose to the road plane for a road vehicle-mounted camera. They introduced a restrictive motion model exploiting several constraints arising from the construction of road vehicles, thus significantly reducing parameter space and increasing overall estimation robustness. Yamaguchi et al. [4] estimated ego-motion from the correspondences of feature points extracted from various regions other than those in which objects are moving. After estimating the ego-motion, the three dimensional structure of the scene was reconstructed and any moving objects were detected. Effendi et al. [5] proposed a new approach by altering the motion estimation problem into a 2D image registration problem. Li et al. [6] proposed a method for estimating pitch angle with a non-occurrence of cumulative error. They used the Harris-corner algorithm and the pyramid Lucas-Kanade method to detect the optical flow of feature points between adjacent frames from the monocular camera. Sturzl et al. [7] presented a generalization of the Koenderink-van Doorn algorithm that allowed robust monocular localization with large motion between the camera frames for a wide range of optical systems including omnidirectional systems and standard perspective cameras. The Koenderink-van Doorn algorithm estimated simultaneously ego-motion parameters, i.e. rotation, translation, and object distances in an iterative way. A variational approach for ego-motion estimation and segmentation based on 3D camera was proposed by Shiyan et al. [8]. Horn et al. [9] proposed a method to reliably estimate the motion of a dynamic stereo camera system in the three dimensional world where observations are disturbed by high portions of independently moving objects.

3 Implementation Steps

3.1 Extraction of Optical Flow Vectors

Optical flow arises from the relative motion of objects and the viewers and consequently it is useful for detection of moving objects in video frames. Two basic assumptions are made for computation of optical flow: (i) Image brightness constancy: The brightness of any observed object point remains constant over time; (ii) Surface reflectance does not contain highlights. There exist many algorithms for extraction optical flows (e.g., OpenCV library [10]). Figure 2 gives an example of optical flow.

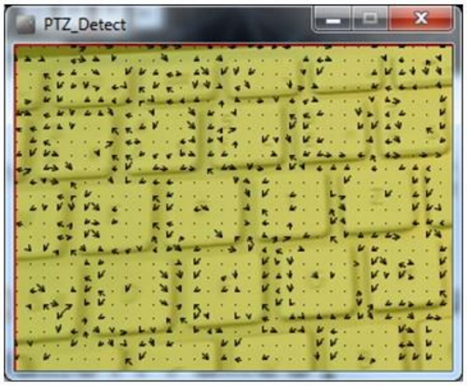


Fig. 2. An Example of Optical Flow

3.2 Block-based Optical Flow Vectorization

Let us consider two consecutive video frames f_i and f_{i+1}. Figure 3 illustrates the block based optical flow vectorization between two consecutive video frames f_i and f_{i+1}.

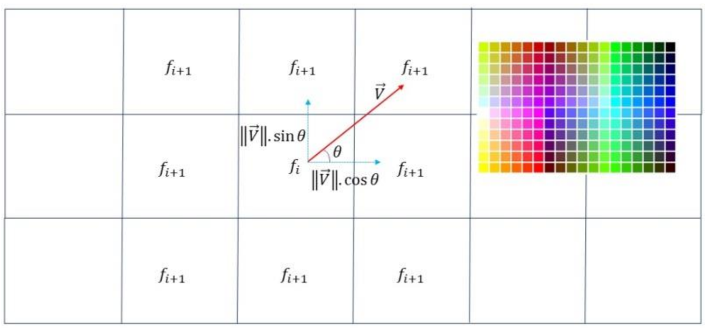


Fig. 3. Block-based Optical Flow Vectorization between Two Consecutive Video Frames f_i and f_{i+1}

3.3 Optical Flow Estimation

Block-based optical flow is calculated by minimizing the sum of squared differences or sum of absolute differences, or maximizing normalized cross-correlation among the 8 neighborhood of each block (patch) in terms of color intensities as formulated in Equations 1, 2, and 3 as follows:

$$min\left[\sqrt{\sum_{0}^{N-1}(x_i - y_i)^2}\right] \tag{1}$$

$$min\left[\sum_{0}^{N-1}|x_i - y_i|\right] \tag{2}$$

$$max\left[\frac{1}{N}\sum_{0}^{N-1}\frac{(x_i-\overline{x})(y_i-\overline{y})}{\sigma_x\sigma_y}\right] \tag{3}$$

where $0 \leq x < width$, $0 \leq y < height$, σ be the standard deviation, and N be the number of optical flow vectors inside the corresponding quadrant.

3.4 Accumulation of Motion Vectors into 4 Quadrants

In order to make a true decision of camera event, motion vectors in each block is accumulated in each frame to compose a resulting vector in 4 quadrants. Figure 4 depicts the accumulation of motion vectors into 4 quadrants as follows:

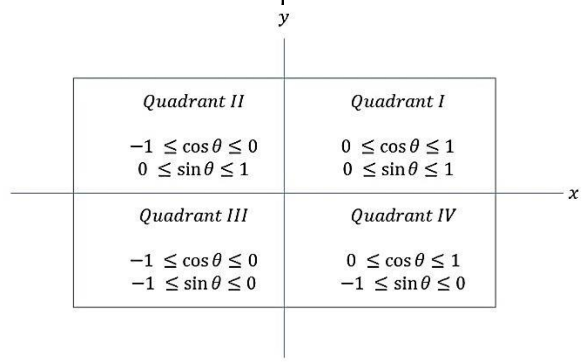


Fig. 4. Accumulation of Motion Vectors into 4 Quadrants

The analysis of accumulated vectors is performed according to the Equation 4, Equation 5, Equation 6 and Equation 7 as follows:

$$\|\vec{R}\|_I = \frac{1}{N_I}\left(\int_0^{\pi/2}\sqrt{(\|\vec{V_\theta}\|\cos\theta)^2 + (\|\vec{V_\theta}\|\sin\theta)^2}\,d\theta\right) \tag{4}$$

$$\|\vec{R}\|_{II} = \frac{1}{N_{II}}\left(\int_{\pi/2}^{\pi}\sqrt{(\|\vec{V_\theta}\|\cos\theta)^2 + (\|\vec{V_\theta}\|\sin\theta)^2}\,d\theta\right) \tag{5}$$

$$\|\vec{R}\|_{III} = \frac{1}{N_{III}}\left(\int_{\pi}^{3\pi/2}\sqrt{(\|\vec{V_\theta}\|\cos\theta)^2 + (\|\vec{V_\theta}\|\sin\theta)^2}\,d\theta\right) \tag{6}$$

$$\|\vec{R}\|_{IV} = \frac{1}{N_{IV}}\left(\int_{3\pi/2}^{2\pi}\sqrt{(\|\vec{V_\theta}\|\cos\theta)^2 + (\|\vec{V_\theta}\|\sin\theta)^2}\,d\theta\right) \tag{7}$$

where N is the number of optical flow vectors inside the corresponding quadrants. Later, resulting vectors are employed in camera event decision.

3.4 Thresholding on the Magnitude of Vectors

Resulting vectors obtained in 4 quadrants are thresholded to minimize the error rate and maximize the true detection of camera events. To detect camera motion, Equation 8 is taken into account as follows:

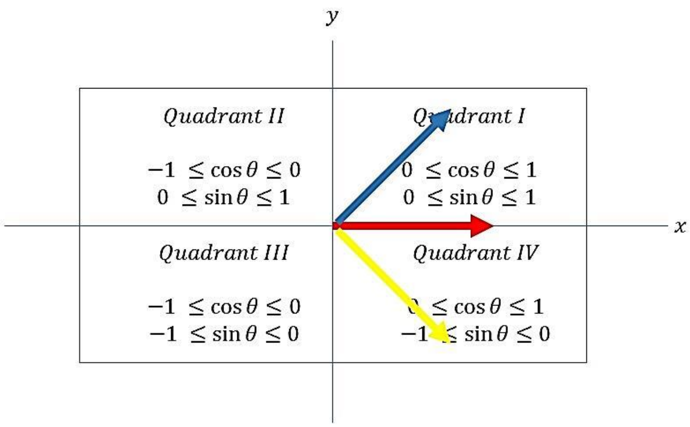
$$IF(\|\vec{R}\| > Threshold) \overset{THEN}{\Longrightarrow} Detect\ Camera\ Motion \tag{8}$$

4 Experimental Results

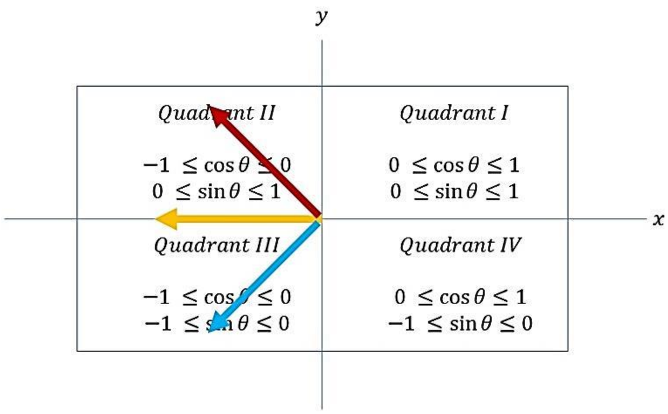
4.1 Experimental Setup

We have tested our algorithm using standard laptop with its built-in camera. Figure 5 and Figure 6 illustrate the mathematical modeling of camera Pan/Tilt/Zoom and sample snapshots of camera events, respectively as follows:

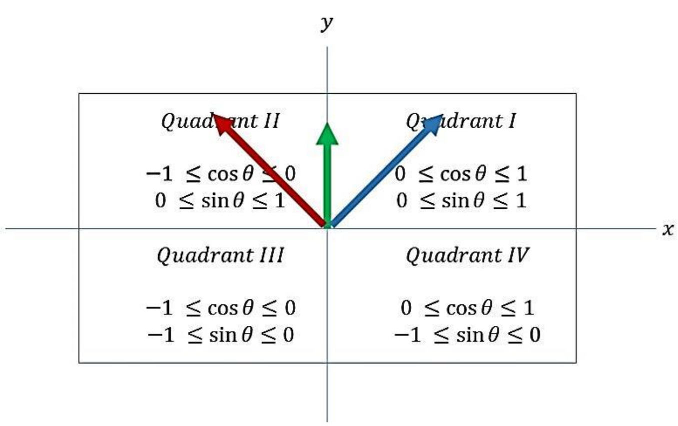
ISSN 2380-8128



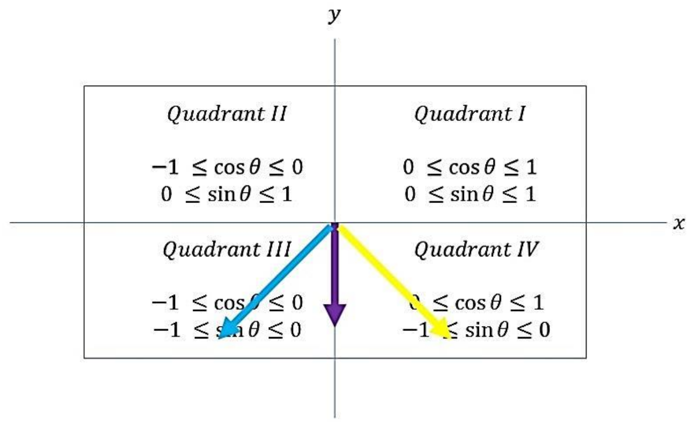
a) Pan-Right Detection



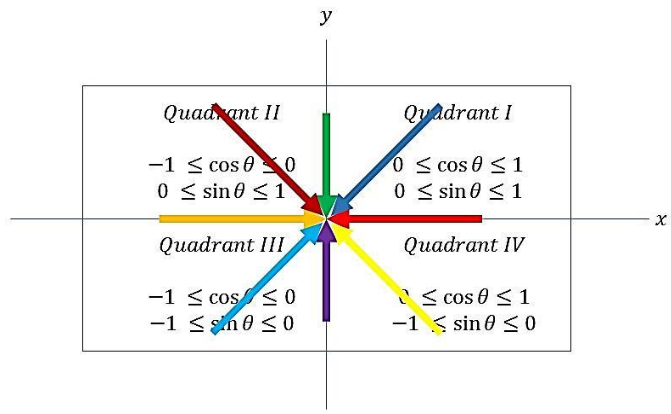
b) Pan-Left Detection



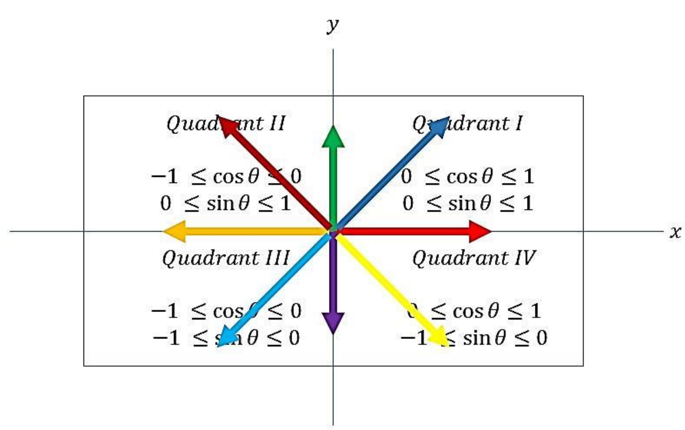
c) Tilt-Up Detection



d) Tilt-Down Detection



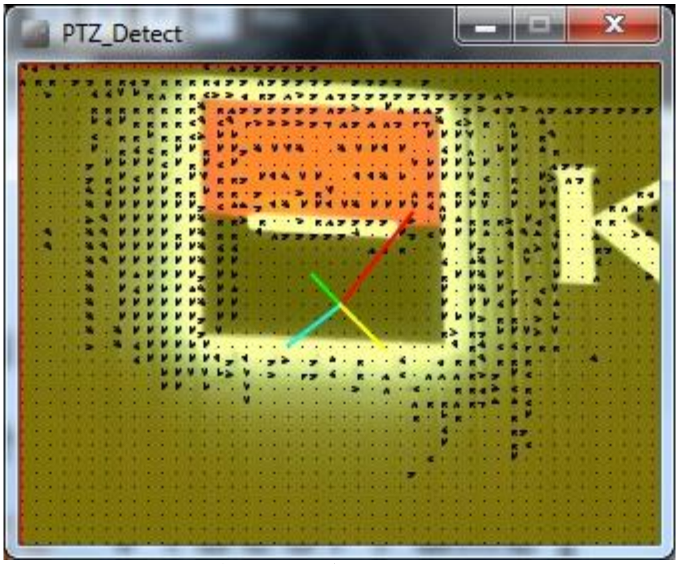
e) Zoom-In Detection



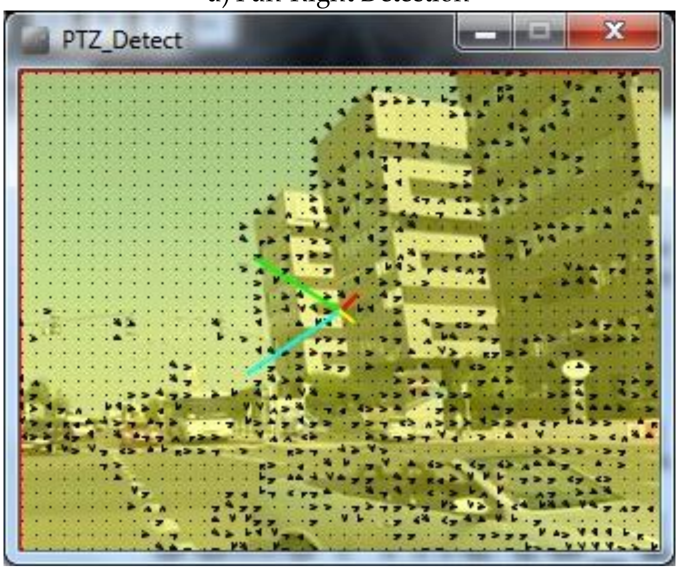
f) Zoom-Out Detection

Fig. 5. Mathematical Modeling of Camera Events

ISSN 2380-8128

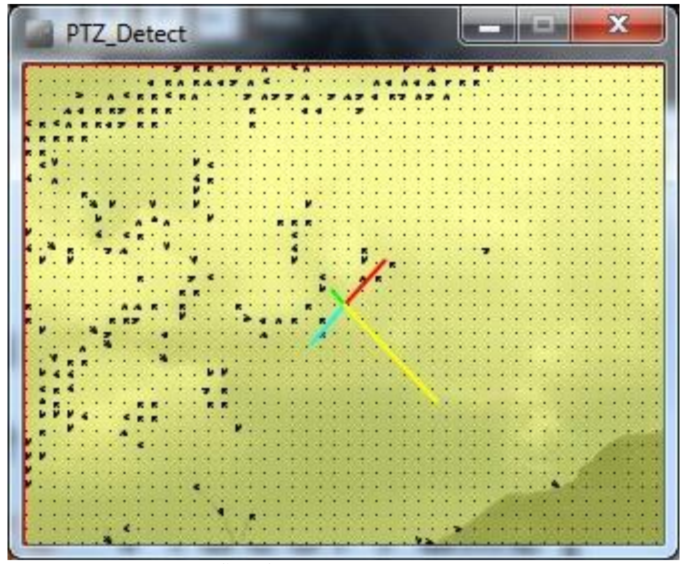


a) Pan-Right Detection

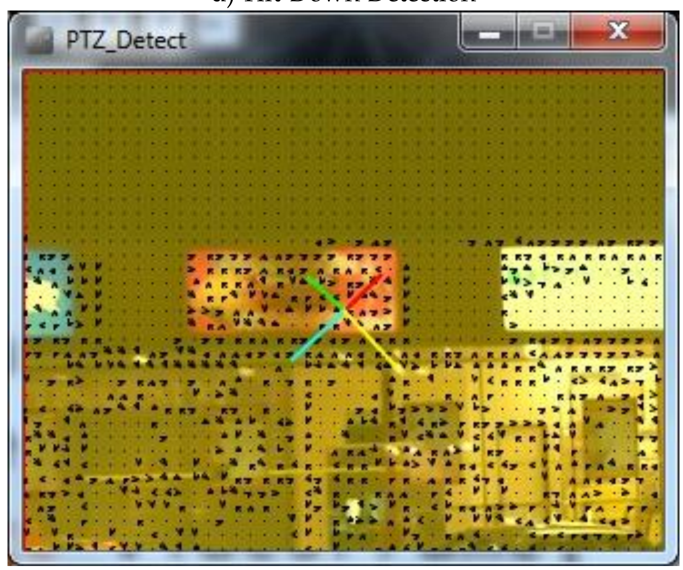


b) Pan-Left Detection

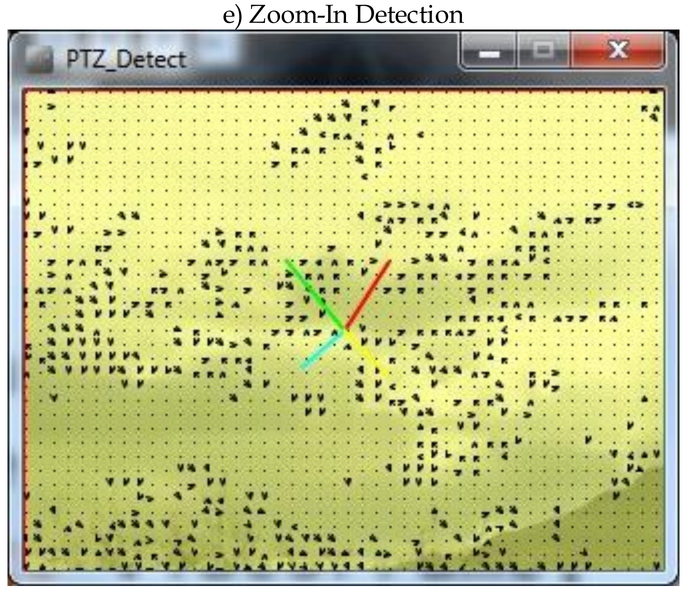
c) Tilt-Up Detection



d) Tilt-Down Detection



e) Zoom-In Detection



f) Zoom-Out Detection

Fig. 6. Sample Snapshots for Detected Camera Events

ISSN 2380-8128

4.2 Dataset

We did not use any benchmark dataset to conduct experiments. However, we have used our own dataset.

4.3 Findings

Limited experimental results show that our proposed algorithm with few mistakes estimates camera ego-motion even for video sequences having small object motion. If there is a big moving object near the image center, the algorithm is sometimes fooled by the object motion and considers it as camera motion. The reported accuracy of the algorithm for static scenes was quite good, and the achieved reduction in computational cost made the trade off well justified.

4.4 Shortcomings

The proposed system was tested with a small scale setup. The efficiency of the algorithm has not yet known. It is important to test the algorithm with benchmark datasets.

5 Conclusion

We addressed a camera ego-motion estimation algorithm, which used a feature based approach for motion estimation. Feature selection limited the number of feature points to be tracked and thus lowered the computational complexity. Experimental results with limited dataset demonstrated the potential usages of our proposed approach. Yet the efficacy of the algorithm would be studied with benchmark datasets.

Acknowledgment

This paper was extracted from an undergraduate course entitled: "COM497 - Senior Design Project" held in 2014-2015 academic period at Gediz University, Izmir, Turkey. Previous version of this study was presented in the "Third International Symposium on Engineering, Artificial Intelligence and Applications (ISEAIA 2015)" held in between 4 to 6 of November, 2015, in Girne American University, North Cyprus. This is the extended full version of the paper already published in the conference proceedings and selected to be published in this journal by the conference committee.

References

[1] Yuan, D., Yu, Y., "A new method on camera ego-motion estimation", *In 6th International Congress on Image and Signal Processing (CISP)*, vol.2, pp. 651-656, 2013.

[2] Ratshidaho, T., Tapamo, J., Claassens, J., Govender, N., "Tof camera ego-motion estimation", *In 5th Robotics and Mechatronics Conference of South Africa (ROBOMECH)*, pp. 1-6, 2012.

[3] Ohr, F., Parakrama, T., Rosenstiel, W., "Model based estimation of ego-motion and road plane using vehicle-mounted camera", *In 16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 602-607, 2013.

[4] Yamaguchi, K., Kato, T., Ninomiya, Y., "Vehicle ego-motion estimation and moving object detection using a monocular camera", *In 18th International Conference on Pattern Recognition (ICPR)*, vol.4, pp. 610-613, 2006.

[5] Effendi, S., Jarvis, R., "Camera ego-motion estimation using phase correlation under planar motion constraint", *In International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 158-165, 2010.

[6] Li, B., Zhang, X., Sato, M., "Pitch angle estimation using a vehicle-mounted monocular camera for range measurement", *In 12th International Conference on Signal Processing (ICSP)*, pp. 1161-1168, 2014.

[7] Sturzl, W., Burschka, D., Suppa, M., "Monocular ego-motion estimation with a compact omnidirectional camera", *In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 822-828, 2010.

[8] Wang, S., Yu, H., "A variational approach for ego-motion estimation and segmentation based on 3d tof camera", *In 4th International Congress on Image and Signal Processing (CISP)*, vol.3, pp. 1160-1164, 2011.

[9] Horn, J., Bachmann, A., Dang, T., "Stereo vision based ego-motion estimation with sensor supported subset validation", *In IEEE Intelligent Vehicles Symposium*, pp. 741-748, 2007.

[10] "OpenCV: Open Source Computer Vision", *Opencv.org*, 2015.