International Journal of Control and Automation
Vol.8, No.5 (2015), pp.99-116
http://dx.doi.org/10.14257/ijca.2015.8.5.10

A Vision-Aided Motion Detection: 3-DOF Orientation Estimation Algorithm Using Mobile Robot

Mamadou Diop, Lim Chot-Hun, Lim Tien-Sze and Ong Lee-Yeng

Faculty of Engineering & Technology
Multimedia University
75450 Melaka, Malaysia
mdiop86@gmail.com

Abstract

The problem of estimating motions of a mobile robot has been studied extensively in the past. In this paper, a new Vision-Aided Motion Detection Algorithm (VAMD) for orientation sensing in an unknown environment is proposed. This algorithm aims to estimate a mobile vehicle orientation, based on a single camera pose, while reducing accumulation error. The VAMD makes use of feature points, detected in a 2D scene, to recognize any rotational motion of a camera, placed on a mobile platform, and estimate the orientations in 3 Degree of Freedom (pitch, yaw, roll angles) from its initial position. The robustness, accuracy and flexibility of this algorithm will be put through several experiments on real scenarios aided with mobile robot, and the results are compared with data obtained from the mobile robot optical encoder.

Keywords: *VAMD, Orientation Estimation, Mobile robot, Optical flow, Image processing*

1. Introduction

In Navigation, knowing or estimating positions and orientations of a mobile robot, at any time, has been a key focus for these past years. The growth of autonomous mobile robot platform has created the need to come out with techniques, based on sensors and satellites, able to provide information on mobile platforms' positions and motions within short time intervals. One of the techniques being considered for orientation and position estimation is the INS (inertial navigation system), which is widely used in navigation applications. The INS make used of Inertial Measurement Units (IMUs) to estimate each of the Euler's angles (yaw, pitch roll) and provide their velocity and acceleration by time intervals. However, low cost INS suffers from the error accumulations which tend to increase over time. To overcome that issue, most practical systems make use of GPS (Global Positioning System) receivers to recalibrate the INS periodically. Even though GPS receiver can provide quite an accurate position information in real-time, it faces signal losses due to jamming, interference, and terrain geometry (such as hill, tall buildings or indoor environments).

Alternatively, vision sensor has been adopted for position and orientation estimation. It offers richness of information extracted from perceived environment while being robust against signal lost and low cost in price. Recent works using vision have achieved up to 6-DOF motion state estimation using the fusion of IMU/camera, monocular and stereo camera based techniques. Existing work regarding IMU/camera make use of several numbers of estimators such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), Batchleast Squares (BLS) and Particle Filter (PF) to estimate 6-DOF motion. However, few issues have been raised with this technique regarding its high computational cost, its delayed

ISSN: 2005-4297 IJCA

measurements, the accuracy of feature initialization and estimation, and the robustness to estimator initialization errors [1]. Using monocular vision, some recent works rely on known objects or reference image to recover 6DOF motion. For examples, [2] recover motions from the combination of the set of pose estimates and the positions of the reference cameras in a robust manner and [3] use known landmarks (fiducial markers) position and orientation. In the other hand, without any artificial landmarks on the environment, 3DOF orientations were able to be recovered by minimizing image projection error and 1 DOF translation by solving an NP-hard optimization problem through an approximation [4]. However, motion estimation using a monocular vision is limited either by a prior knowledge of the environment and landmarks' position, or by employing inferred measurement model (assuming the camera observes its pose in the world frame, up to scale). Monocular vision limitation can be solved by stereo vision systems; where the baseline between the two cameras can define the range scale of observation without any assumption. With this model, 6DOF motion could be recovered in example of [5]. However stereo vision limited itself by the baseline, its computational time and its setting complexity [4]. Most of the proposed techniques for vision-aided motion estimation are able to provide good accuracy measurement and robustness. But at some points, most of them are facing limitation either in their operating scale, their cost in price and computation time, or sometimes require prior knowledge and or modification of the environment. Those limitations could lead to wrong estimation where, respectively, whenever there is absence of features within the scale range, features' lost due to the significant delay caused by the computational time and where motions are operated in a unknown environment. Moreover, most of existing techniques in visual odometry were implemented in outdoor environments. They are more tolerant to processing delay since most of features are usually far from the camera and therefore still visible on successive image scene in high motion speed. Whereas in an indoor environment, object are usually near from the camera and tend to phase out of the scene or scale up quite fast. Therefore long processing delay of the algorithm might causes features losses and lead to wrong estimations.

In this paper, an initial phase of a new vision-aided motion detection algorithm, which estimates a 3-DOF camera orientation, is presented. This algorithm requires a single camera and is built to overcome the mentioned issues on existing vision-based techniques, while providing robustness and acceptable accuracy in measurement when operating in an unknown indoor environment. This paper is organized as follows. Section II outlines the fundamental approach in VAMD and its associated theories. Section III presents the experimental studies with detail discussions on the collected results, and lastly, Section IV concludes the paper.

2. Vision-Based Motion Detection Algorithm (VAMD)

2.1. VAMD Flowchart and Framework

VAMD is a vision based algorithm that recognizes and estimates motions of any vision-based hardware. Given a live or recorded video, VAMD identifies features points from each captured frame. Feature points are small windows of pixels, evaluated based on their textures uniqueness and traceability [6]. Once a set of feature points are selected, they are matched to those identified in the next captured frame by using optical flow technique (Lucas Kanade algorithm). Optical flow can be defined as the relative motion of features, edges objects, surfaces or image intensity patterns from one scene to another. Among the existing optical flow techniques developed over these past three decade, Lucas-Kanade algorithm stand out to be more robust and reliable than others [7]. Therefore, this algorithm is based on a recursive loop which consists of detecting good

features using Shi&Tomasi technique [6], then tracks and matches them on the next captured image using iterative Lucas-Kanade method with pyramids [8]. Between each capture, processes are executed to estimate orientations and recognize directions based on the features' motion magnitudes. The developed VAMD algorithm is a real-time algorithm. Figure 1 illustrates the flowchart of the VAMD algorithm.

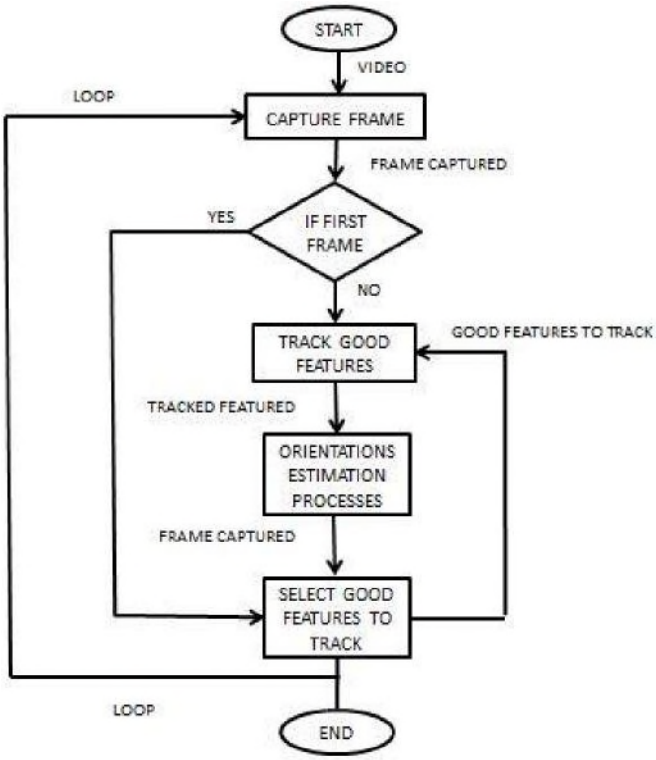


Figure 1. VAMD Flowchart

Orientations estimation process makes uses of the pairs of features points matched between images. The coordinates (x,y) of the features points, that relate its position on the image frame, are extracted. Following each rotation axis, the magnitude values from the displacements of each pair of features are extracted and placed into containers; where theirs corresponding angles are calculated using information of the camera (such as the field of view and the image resolution). In each container (or rotation axis), recursive processes will be triggered for the estimation accuracy and the algorithm robustness against errors:

- For the estimation accuracy, the average of the most common estimated angles up to a certain precision is calculated; that way to reduce and compensate error in estimation.
- The algorithm robustness will make use of features' magnitude pattern to avoid drift at a stationary sate and tolerate moving object and translation up to some extent.

The final orientations values (yaw, pitch, roll angles) will result the sum of all estimated angles in between images. The Figure 2 describes the VAMD framework that performed the orientation estimations.

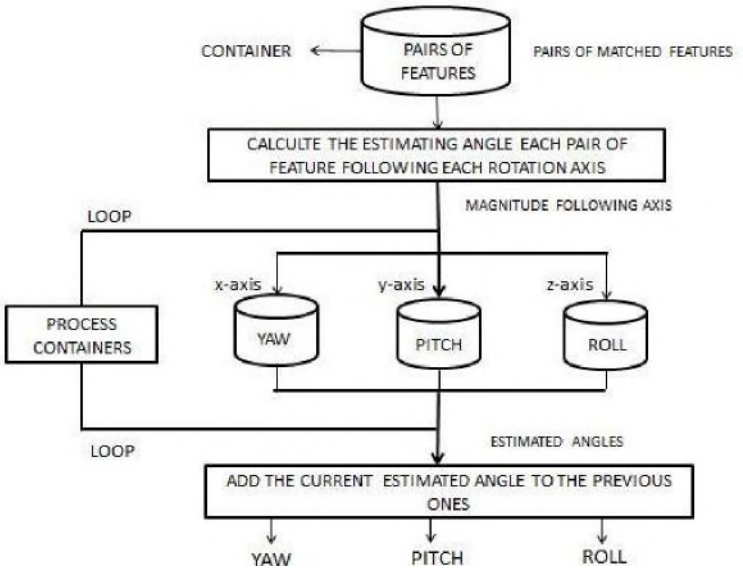


Figure 2. VAMD Framework

2.2. Orientation Estimation

The orientation of an object can be defined as the direction of the object in terms of the combination of three rotations angles (Euler angles) from an initial position (fixed, global, or world coordinate system). Euler angles represent a sequence of three rotations about different axes of a coordinate system: a rotation about the z-axis called yaw angle (ψ), a rotation about the x-axis is called roll angle (□), and a rotation about the y-axis is called pitch angle (θ). Figure 3 illustrates the representation of those rotation angles on a Three-Dimensional (3D) point of view.

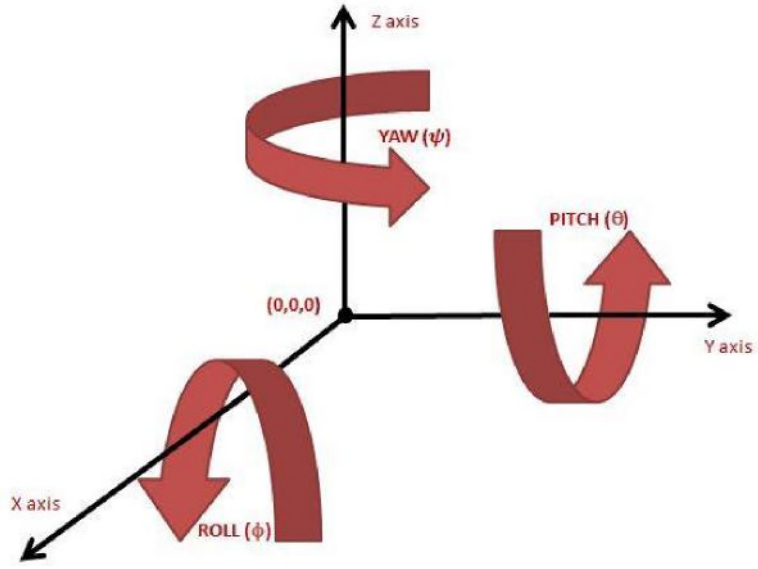


Figure 3. Euler Angles Representation on 3D View

However, representing the Euler angles in images requires different approach. Two-Dimensional (2D) images are represented in two axis coordinates system(X, Y) where the x-axis follows the height of the image and the y-axis follows the width. Therefore, representing Euler angle in a 2D image would imply that the yaw angle (ψ) would be represented through x-axis (height) or a rotation on the y-axis, with

the pitch angle (□) represented through y-axis (width) or a rotation on the x-axis, and the roll angle (□) represented by a rotation around the center of the image, as illustrated in Figure 4.

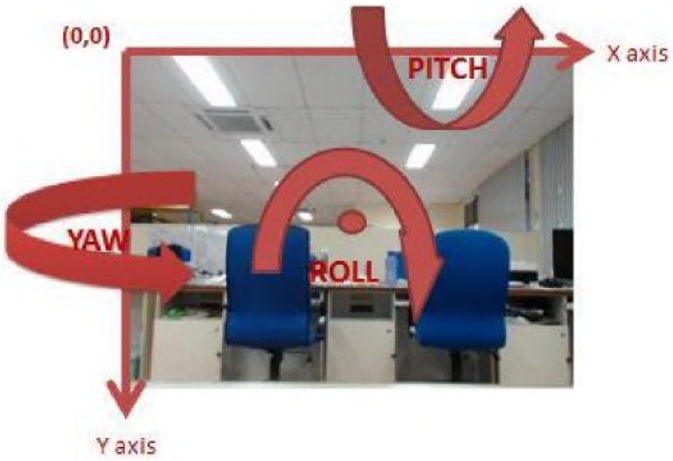


Figure 4. Euler Angles Representation on a 2D Image

Estimating orientation from a motion between 2D images requires information on pairs of features points coordinate, the camera resolution in pixel and its field of view (FOV). The field of view is the extent of the observable world that is seen through a camera. It is a value expressed in degrees and can have the same or different value on the height (x-axis) or width (y-axis) of the image, depending on the type of camera. Information about the FOV can be obtained either along with the hardware specifications or by performing a manual calibration. Manual camera calibration, to obtain the field of view, makes use of a protractor or a motorized rotator. As shown in Figure 5, it is performed by identifying one object, edge or point from one extremity of the captured scene, then rotate the camera until the same feature is located to the opposite extremity from its initial position.

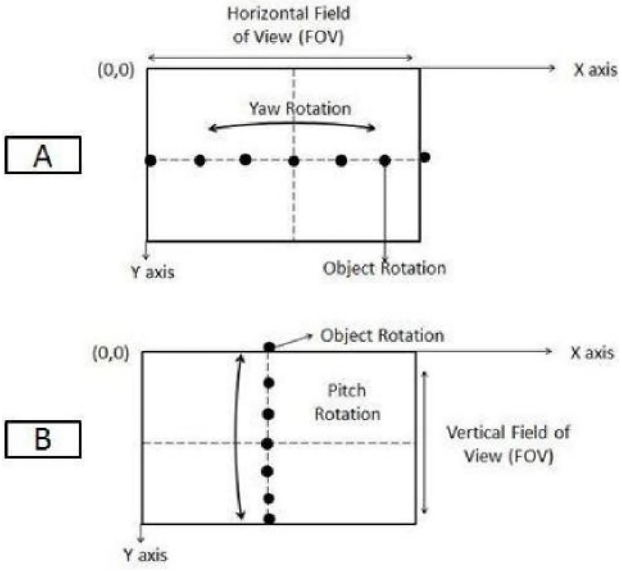


Figure 5. Field of View Calibration with a Yaw Rotation

As illustrated in Figure 6, the magnitude ( ) from a displacement of a feature point ( ), between a previous image (p) and a current image (), is obtained from the coordinate points (x,y) and (x,y). It can be expressed, based on each axis (X, Y) for the Euler angles calculation, as follows:

$$M = \quad - \quad \quad (1)$$

$$M = \quad - \quad \quad (2)$$

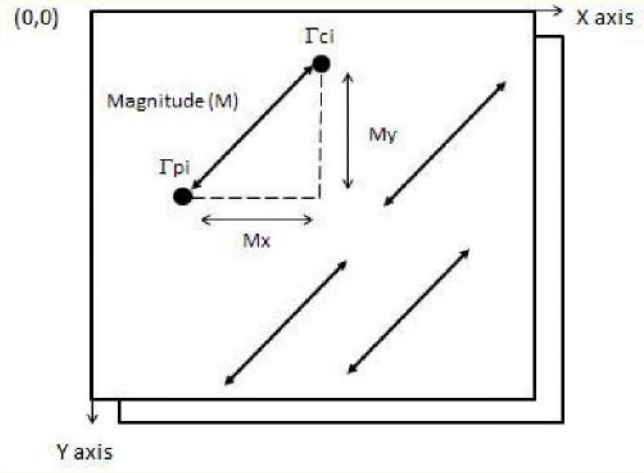


Figure 6. Optical Flow Magnitude Represented Following x-axis and y-axis

Since it is assumed that:

Height of the image (in pixel) is equivalent to the field of view following the x-axis (FOV in degree), and the Width of the image (W in pixel) is equivalent to the field of view following the y-axis (FOV in degree)

The formulas to calculate the yaw (Ψ_i) and pitch (ϕ_i) rotation angles in between frames are expressed as follows:

$$= \quad (\quad * \quad) \quad (3)$$

$$= \quad (\quad * \quad)_{w} \quad (4)$$

$$= \sum_{=1} \quad \text{and} = \sum_{=1} \quad (5)$$

where n is the number of pairs of images.

Figure 7 illustrates the roll estimation that consists of a rotational angle about the z-axis, passing through the center of the image.

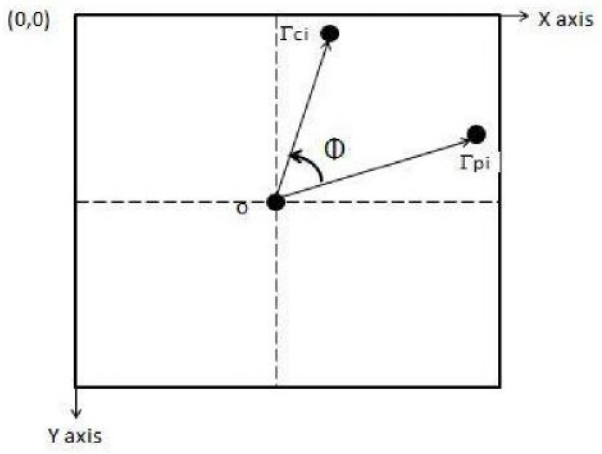


Figure 7. Representation of Roll Angle Estimation on a 2D Image

Thus, Roll angle $_{(\emptyset)}$ between pair of feature points $_{(1,1)}$ and center point of the image ( ) is calculated using dot product (*dot*) and cross product (*cross*) formulated as follows:

$$\emptyset = {}^{-1}(,) * 180 \quad (6)$$

$$\emptyset = \sum_{=1} \emptyset \quad (7)$$

where,

$$= \|1\| * \|1\| * \emptyset \quad (8)$$

$$= \|1\| * \|1\| * \emptyset \quad (9)$$

3. Experimental Studies

3.1. Experimental Setup

In this section, the VAMD algorithm is put under several experiments in order to test its robustness and accuracy. The experiments are conducted in an indoor environment. The equipment used for the experiments consisted of the PIONEER P3-AT from Adept Mobile Robot, a Logitech camera HD c920 and a laptop that host the VAMD algorithm. The camera is set with a 320x240 (width x height) pixels resolution, 52x42 degrees (horizontal x vertical) field of view after calibration and capturing 30Fps (frames per second). VAMD is developed on visual C++ with OpenCV library, running at most 2ms (milliseconds). It uses a set of maximum 100 features to track, selecting only the most salient ones that can be observed on a scene.

The experiments will be divided in 3 sections. In the first experiment, rotation and translation motions are recognized and differentiate based on the optical flow magnitude patterns. Then, the accuracy of the 3-DOF rotation (yaw, pitch roll) estimations will be tested with different angles at different rotational velocity. Finally, the robustness of the algorithm is evaluated when the mobile robot runs through a trajectory and when moving object occurs on the scene.

3.2. Optical Flow Pattern over Rotational and Translational Motions

In this first section, optical flow patterns will be evaluated when rotation or translation motion occurs. Since only 3-DOF rotation motion is estimated in this

paper, it is important for the algorithm to differentiate them and avoid error in estimation as both motions lead to features' displacement. To study optical pattern on both motions, successive captured images with incremented translation (centimeters) and rotation (degrees) units are observed in different circumstances.

3.2.1. Optical Flow Pattern over Rotational Motions: In this section, rotation motions are study, through several yaw rotations (1, 2, 3, 4 degrees), on scenes with different features' distance (more than 3 meters, more than 0.5 meters and more than 0.1 meter) as illustrated on Figure 8, The values of the features' displacement magnitude, from each case scenario, are display and interpreted on Figure 9 and Figure 10.

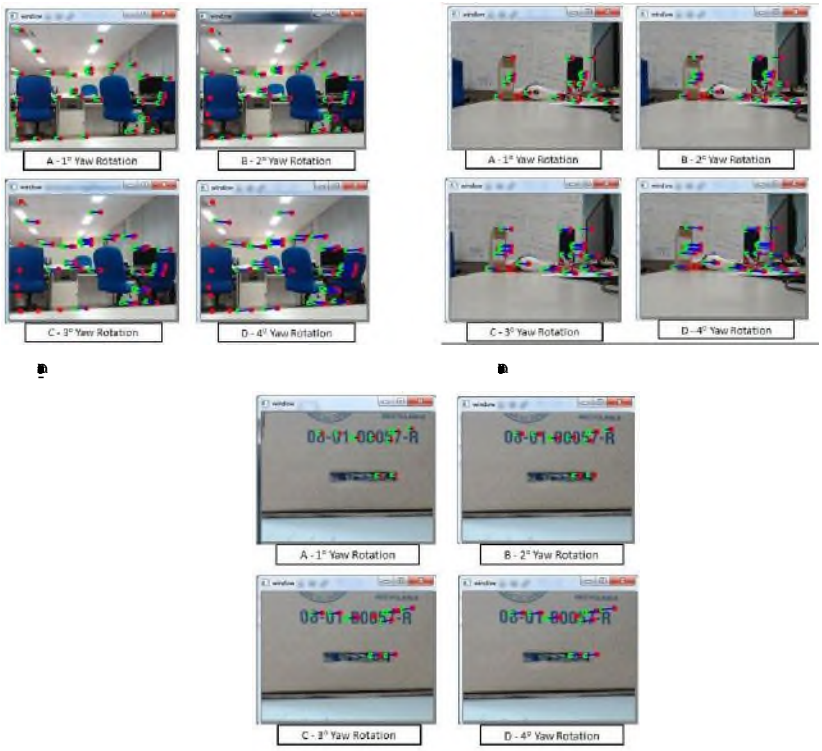


Figure 8. Rotation: Optical Flow Pattern for Feature at Different Distances

		1° 2°				1° 2°				1° 2°	
7.32596	12.1992	19.7395	26.7418	7.36523	12.9313	19.0147	26.6308	7.66855	13.4065	21.0901	27.0467
7.32764	13.9202	19.9735	25.9494	7.03024	12.9294	19.014	26.6229	7.16624	13.4788	20.0714	26.1754
6.26741	13.7119	19.4357	25.96	7.36526	13.6745	19.9705	27.6463	7.37926	12.9015	20.1945	
7.12338	11.9683	19.4349	24.4297	7.04369	12.8786	18.7657	26.3528	7.49146	12.9818	20.9873	
6.84096	11.9683	18.3511	25.2775	7.32079	12.9339	19.0149	26.631	7.4935	12.9078	20.9863	
6.40833	12.4412	19.4344	25.9503	7.34596	13.6732	19.9707	27.645	7.16589	12.9061	20.0712	
6.68808	11.9683	19.6065	25.8736	6.82008	12.8874	18.7657	26.3447	7.16629	13.4974	20.0715	

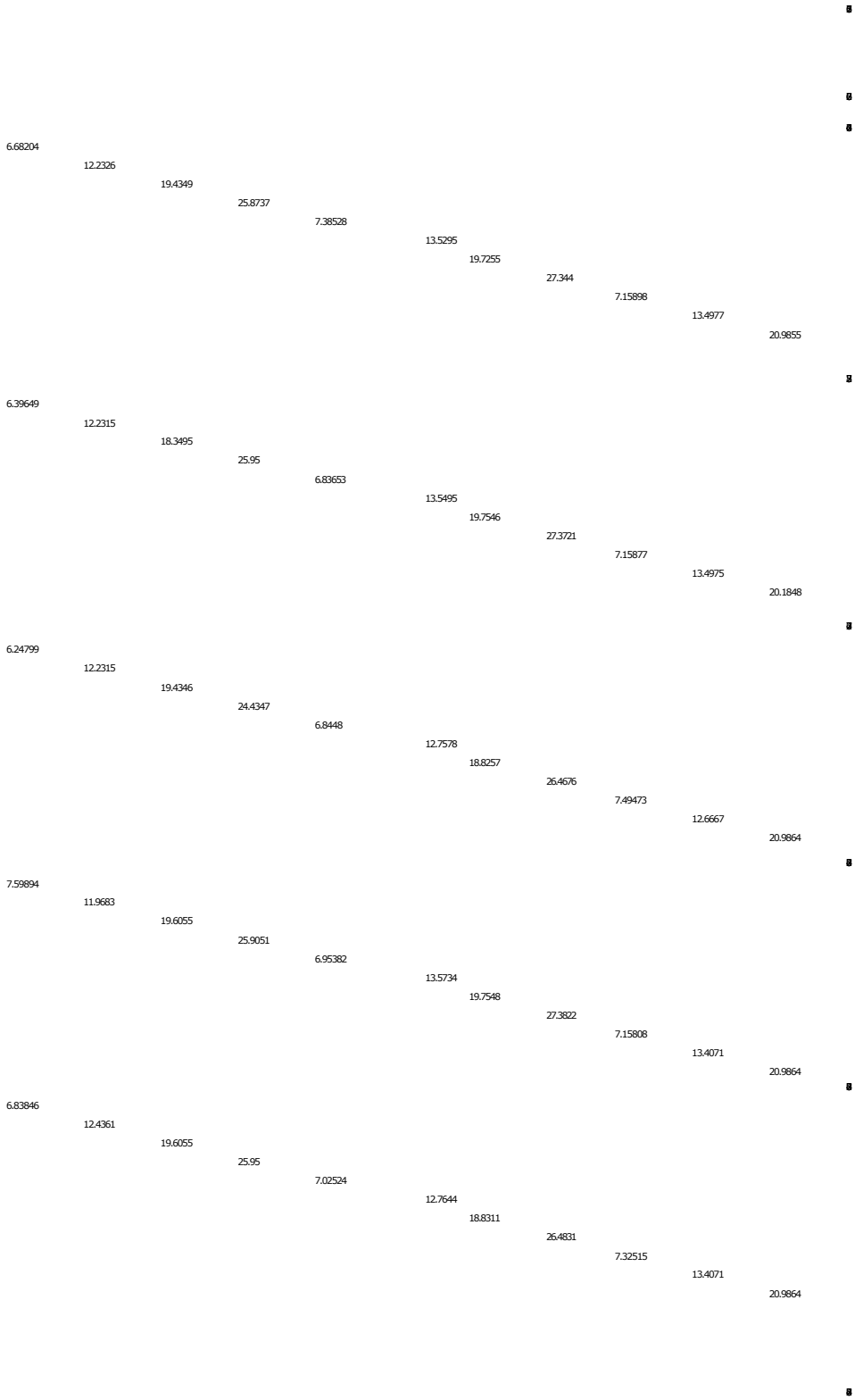


Figure 9. Rotation: Features' Displacement Magnitude Values at different Distances

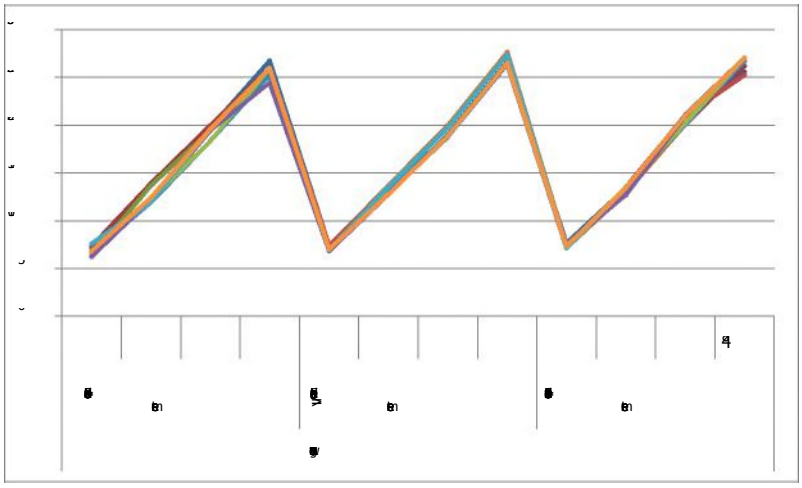


Figure 10. Graph of the Features' Displacement Magnitude

Based on this experiment results, features depth seems to have no effect on a rotation motion. Values extracted from features' displacement magnitude appear almost constant on each rotation value; regardless of the features distance from the camera (see Figure 10). This shows that features displacement can be used to estimated rotation motions without any prior knowledge on the operating environment.

1 °
2 °
3 °
4 °
1 °
2 °
3 °
4 °
1 °
2 °
3 °

3.2.2. Optical Flow Pattern over Translational Motions: In this section, forward translation motions are studied. Pairs of image, taken with one centimeter gap forward, are used to describe the optical flow pattern on scenes with different features' distance (more than 3 meters, more than 0.5 meters and more than 0.1 meter) as illustrated on Figure 11. Features' Displacement magnitude values, following x and y-axis, are illustrated in Figure 12.

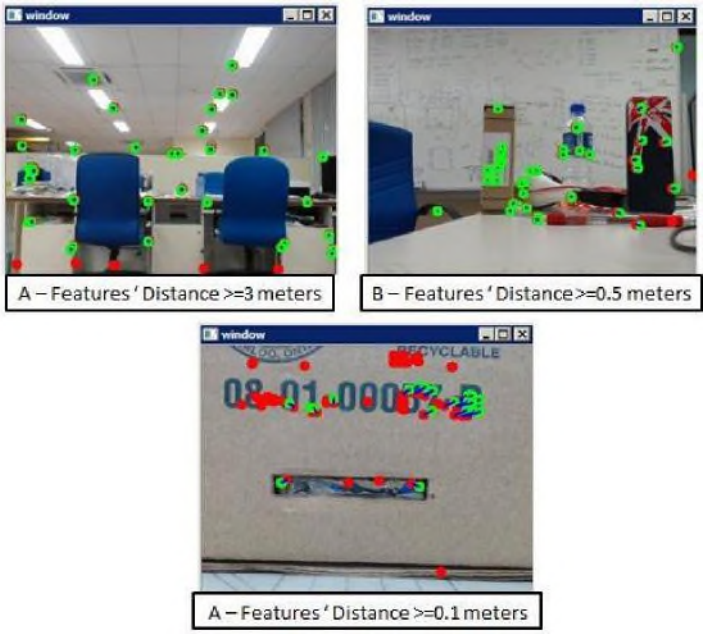
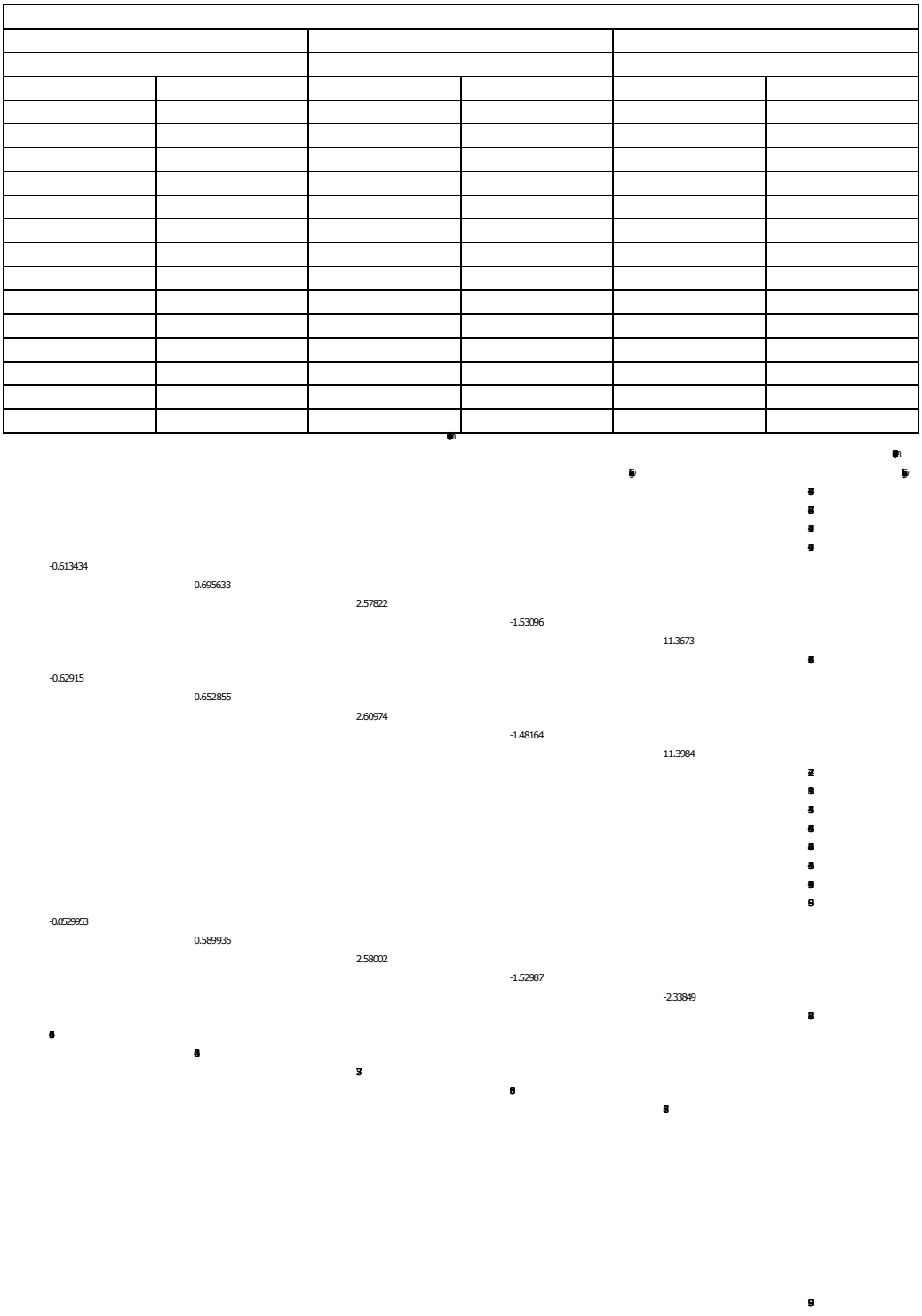


Figure 11. 1cm Forward Translation over Different Features' Distance Scenes



-0.613434
0.695633
2.57822
-1.53096
11.3673
-0.62915
0.652855
2.60974
-1.48164
11.3984
-0.0529953
0.589935
2.58002
-1.52987
-2.33849

Figure 12. Translation: Features' Displacement Magnitude Values at Different Distances

Features distance >=3 meters

Following x-axis Following y-axis

-0.71875
0.640026
1.12857
-0.294468
6.85445
-0.508835
0.427673
1.11606
-0.313915
7.52579

Features distance >= 0.5 meters

Following x-axis Following y-axis

-0.719322 0.635818 1.11606 -0.313915 5.85083
-0.688896 0.656559 2.62271 -1.48339 8.06799

Based on this experimental study, optical flow pattern over forward translation motions doesn't recorded any constant pattern following x and y-axis. However, it is noticeable that features' displacement magnitude increased on nearer distance from the camera with different scale factor; even though they seem barely moving when they are far enough from the camera.

-0.538498 0.615547 2.57816 -1.53023 0.370125
-0.51369 0.871056 1.05573 -0.578583 11.7205
-0.534737 0.699432 1.05574 -0.578514 10.4415
-0.534599 -0.699417 -0.483192 -1.39017 11.3867
-0.905266 0.702515 2.62473 -1.4839 -3.82469
-0.905319 0.702568 2.63763 -1.48595 11.415
-0.534798 0.699455 0.460571 -1.3903 -2.41111

3.3. VAMD Accuracy

In this section, experiments will consist of evaluating the VAMD measurement accuracy by comparing its estimation to the one stated by the mobile robot optical encoder. The Data provided from the mobile robot and the VAMD algorithms are displayed using the same GUI developed in Visual basic as shows in Figure 13.

Figure 13. GUI of VAMD and Mobile Robot Data Output

Since the mobile robot is restricted to moving on the flat floor where only the yaw rotation could be performed (with an error more or less equal to 1.31 degrees), the camera will be set at its best on different dispositions (see Figure 14, Figure 15, Figure 16, sections A), while being parallel (for yaw, pitch and roll) and centered (in addition for roll) with the robot rotation, to estimate all three rotation angles (yaw,

pitch, roll) by selecting the average of the most common value with 10^1 precision. Therefore, this experiment is realized as follow:

□ A set of 10 yaw rotation angle (approximatively 25, 50 and 90 degrees) is performed by the mobile robot.

□ Each set of rotation angle is performed at rotation speed of 10 and 50 degrees per second.

□ From each set, the measurement obtained from VAMD (VO) will be compared the ones obtained from the mobile robot (WO).

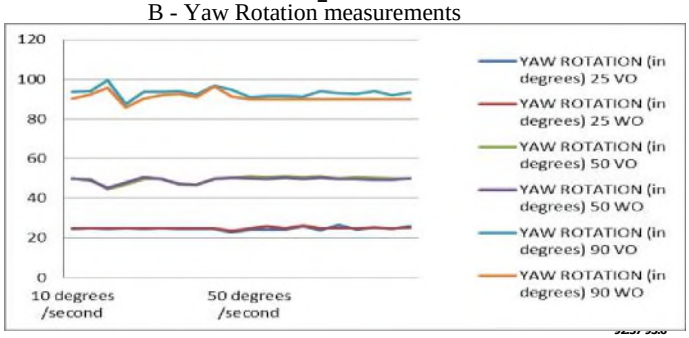
Figure 20 Looking-Forward Camera for Yaw Estimation

A - Looking-Forward Camera for Yaw Estimation

	VO	WO	VO	WO	VO	WO
	24.44	24.86	49.55	49.83	93.44	90.31
	24.77	24.86	49.58	49.05	94.03	
	24.63	24.86	44.4	45.22	99.35	
	24.82	24.86	46.95	47.77	87.32	
	24.61	24.86	49.69	50.66	93.56	
	24.82	24.86	49.99	49.48	93.75	
	24.54	24.86	46.89	47.29	93.77	
	24.63	24.86	46.35	46.78	92.13	
	24.48	24.86	49.98	49.57	96.67	
	22.95	23.37	50.13	50.27	94.71	
	max_error = 0.42		max_error = 0.97			

25.21
25.83
50.74
49.66
24.22
24.7
51.01
50.19

B - Yaw Rotation measurements



85.66
90.14
91.93
92.59
90.87
96.34

C - Graph of the Yaw Rotation Measurements

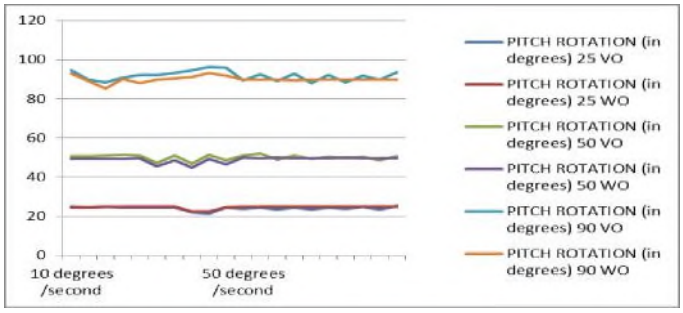
Figure 14. Experiments: Yaw Rotation Estimation

A – Flip-Sided Looking Camera for Pitch Estimation

				[illegible]		
		2		50		0
		WO		WO		WO
	24.92	24.69	50.62	49.39	94.55	92.76
	24.57	24.69	50.75	49.39	89.8	
	24.83	24.79	50.99	49.39	88.46	
	24.43	24.79	51.31	49.39	90.82	
	24.47	24.8	51.09	49.48	92.17	
	24.48	24.8	47.06	45.35	92.08	
	24.59	24.8	50.84	48.5	93.33	
	21.95	22.5	46.89	44.74	94.58	
	21.49	22.3	51.48	49.39	96.21	
	24.43	24.69	48.72	46.4	96.07	
	max_error = 0.81		max_error = 2.34			

23.7
24.87
51.12
49.83
24.39
24.87
52
49.74
23.45
24.87
48.76
49.83
2
8
5
6
3
8
9
6
4
8
0
7
9
8
0

B - Pitch Rotation measurements



89.03 85.31
90.17 87.98
89.75 90.36
91.14 93.25
max_error = 4.82 89.65

C - Graph of the Pitch Rotation Measurements

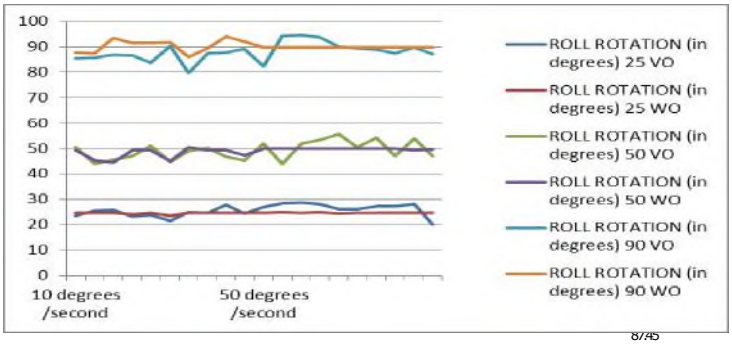
Figure 15. Experiments: Pitch Rotation Estimation

A - Looking-Upward Camera for Roll Estimation

				[illegible]		
		2		50		0
	WO	WO	WO	WO	WO	WO
	23.71	24.79	50.51	49.49	85.41	87.65
	25.62	24.7	43.96	45.46	85.65	
	25.88	24.78	45.59	44.47	86.92	
	23.25	24.25	47.19	49.4	86.54	
	23.81	24.69	50.99	49.4	83.82	
	21.64	23.64	44.88	45	90.24	
	24.92	24.79	49.02	50.52	79.75	
	24.69	24.79	50.24	49.39	87.48	
	27.93	24.79	46.91	49.48	87.7	
	24.53	24.7	45.35	47.37	89.2	
	max_error = 3.14		max_error = 2.57			

26.99
24.7
51.96
49.83
28.54
24.96
43.95
49.83
28.87
24.7
51.93
49.83
8
2
3
0
8
0
6
0
0
2
0
0
2
2
0

B - Roll Rotation measurements



C - Graph of the Roll Rotation Measurements

Figure 16. Experiments: Roll Rotation Estimation

Based on this experiment results in Figure 14, Figure 15, Figure 16 sections B, VAMD algorithm shows acceptable accuracy in yaw, pitch and roll estimations, compared to results obtained in [9] and [10]. However, these experimental results show as well errors that increased depending on the camera disposition, the speed and larger rotation angles.

Indeed estimating yaw, pitch, and roll rotation, in this context, requires the camera to be disposed into a specific manner while being parallel to the motion or the scene perpendicular to it. Therefore, a small inclination from that particular disposition would generate errors that are accumulated over time. That would explain the accuracy degradation on some set of measurement compared to other; especially in roll estimations (see Figure 16), where the camera was supposed to be placed at the center of the robot rotation for higher accuracy.

Also, the rotation speed of the robot happens to be a factor that can impact on the estimation accuracy. It usually shows the limitations of the implemented camera. Using a low cost camera, higher motion speeds are most likely to generate blurrier image frame, which caused feature losses and tracking errors, especially when camera lens is not sharp enough to detect salient feature points. That would explain the better accuracy on most of 10 degrees rotation speed measurement sets over the 50 degrees ones. That fact is more obvious on roll estimations, where the camera, looking-upward, did not detect much of a traceable features points as illustrated in Figure 17.

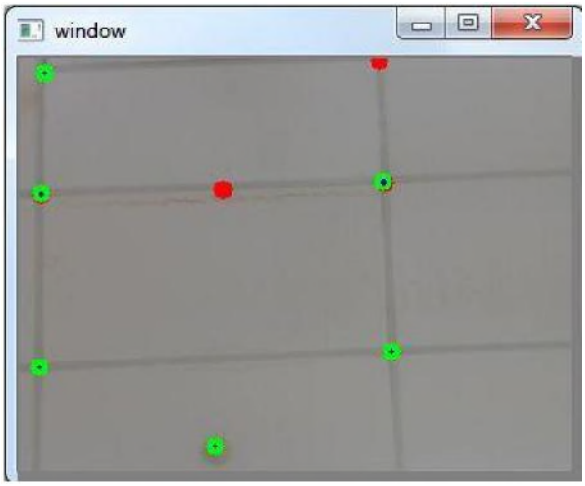


Figure 17. Ceiling View for Roll Measurements

Another factor impacting the accuracy in estimation would be larger rotation angle performed by the mobile robot. Since the robot's rotations are measured based on the number of round produced by its wheels, it often encounter wheel slips which tend to generate errors that are accumulated over time. An experimental study (see Figure 18), conducted to compare wheel odometry of the robot with real measurement over 10 runs of 100cm distance and 50mm/s velocity, clearly shows difference (significant at some runs) in both measurements. That would explain higher errors in estimation occurring on higher rotation angles; even though they are relatively small on some measurements within sets.

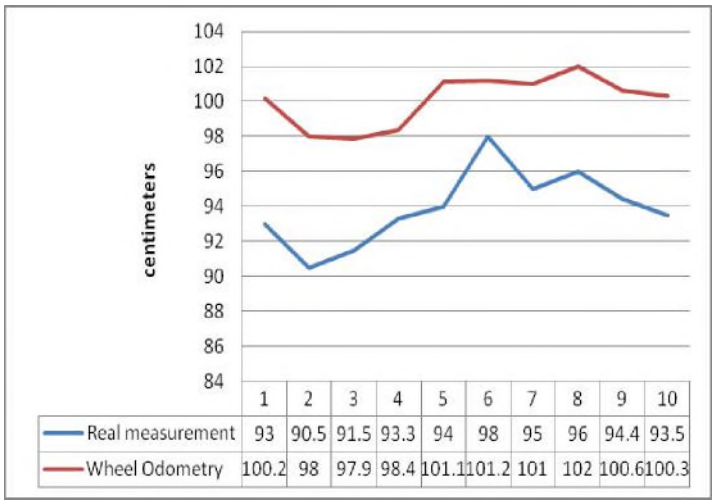


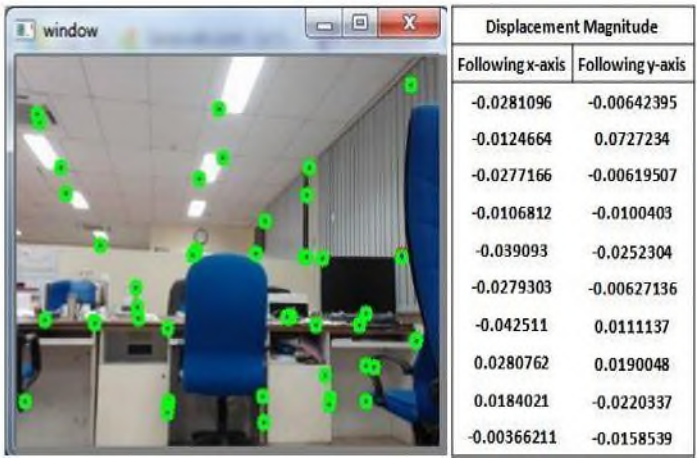
Figure 18. Graph of Wheel Slip Estimation

In conclusion, the maximum errors recorded on sets of measurement don't necessarily relate the accumulated errors, all coming from the angle estimation approach of VAMD algorithm. Instead, the quality of the camera, as well as wheel slips, impacts significantly the accuracy in estimation.

3.3. VAMD Robustness

In this section, the robustness of VAMD algorithm is studied. It consists of adjusting the framework (see Figure 2) by implementing extras conditions (if-statement) in between its phases. Those conditions highly depend on the camera and the mobile robot capabilities.

On a stationary state of the camera, an experiment, consisting of capturing image every 5 seconds interval, shows that some features points are not completely still. Some features points, after each captured, image recorded small scale displacement magnitudes, varying up to certain Threshold values () depending on the camera (see Figure 19). The threshold is the values interval ranging from the minimum to the maximum features displacement when the camera is at its stationary state.



Displacement Magnitude	
Following x-axis	Following y-axis
-0.0281096	-0.00642395
-0.0124664	0.0727234
-0.0277166	-0.00619507
-0.0106812	-0.0100403
-0.039093	-0.0252304
-0.0279303	-0.00627136
-0.042511	0.0111137
0.0280762	0.0190048
0.0184021	-0.0220337
-0.00366211	-0.0158539

Figure 19. Graph of Wheel Slip Estimation

Since rotation motions involve features to move in same direction and having approximatively the same estimated angle, any pair of feature having a displacement

magnitude (following the three rotation axis) below or equal to the threshold would result to a non-motion on that particular axis. That way, VAMD is able to eliminate drift when there is no motion and furthermore, it will be able to tolerate moving object on the scene as long as the object is not covering all feature points.

With this proceeding, VAMD robustness is pushed further more up to tolerating translation motion. That would involve the mobile robot, along with a defined camera frame rate and frame resolution, to run under a certain range of rotation and translation velocity. When motion occurs on a given a pair of images P, representing either a translation motion P_t (estimated in centimeters or pixels per second) or a rotation motion $P_,$ (estimated in degrees or pixels per second), the rotation velocity $V_,$ and translation velocity V_t, based on a specified camera frame rate FPS, are expressed in following equations 10 and 11:

$$V_, = P_, \times FPS \tag{10}$$

$$V_t = P_t \times FPS \tag{11}$$

Therefore, VAMD to be translation tolerant until specific distance to the nearest to avoid confusion between translation and rotation, a minimum $V_,$ (in Degrees or pixels per second or in) must be set with a maximum V_t (centimeters per second), while the threshold is ranging within ± $P_,$ (see Equation 12).

$$\begin{cases} V_, \geq P_, \times FPS \\ V_t \leq P_t \times FPS \\ -P_, < \quad < P_, \end{cases} \tag{12}$$

Based on experiments performed under section 3.1, the optical magnitude corresponding to 1□ rotation is greater than all magnitudes on 1cm translation where features' distance greater than 0.5 meters. Therefore, the translation tolerant, using a frame rate FPS = 30 frame per second and a frame resolution (320x240) will require the robot velocities to be set as follow:

$$\begin{cases} V_, \geq 30 \ deg/second \\ V_t \leq 30c/second \\ -1° < \quad < 1° \end{cases}$$

Using the above setting on the mobile robot has resulted to a non-motion; when it travels a straight forward distance, up to 0.5 meters to a wall.

Figure 20. Experimental Scene of Forward/Backward Translation Tolerant

4. Conclusion

This paper presented a vision-aided motion detection algorithm capable of estimating 3-DOF orientations. This algorithm can be considered as an alternative approach for motion sensing technology since it has shown being able to operate in narrows indoor and unknown environments while providing promising accuracy in estimations, robustness against undesirable motions (translation and moving objects) and flexibility in its capacity of adjusting the robustness and accuracy in estimation which depend on the available hardware and statistical analysis performed on the containers (see Figure 2). Moreover, the algorithm runs at a very low cost, in terms of hardware sensors and computation time, compared to other existing technology.

This algorithm will be further enhanced to estimated travelled distance and thus able to recover a full path or trajectory of a mobile robot while improving approaches in estimations accuracy and robustness. Furthermore its implementation on an aerial vehicle will be taken into consideration; where it will able recovered up to 6-DOF motion.

Acknowledgements

Authors would like to thank all CRSST (Center of Remote Sensing and Surveillance Technology) members for their support through the realization of this project.

References

[1] J. A. Hesch, D. G. Kottas, S. L. Bowman and S. I. Roumeliotis, “Camera-IMU-based localization: Observability analysis and consistency improvement”, The International Journal of Robotics Research, (**2013**).
[2] C. Arth, G. Reitmayr and D. Schmalstieg, “Full 6dof pose estimation from geo-located images”, In Computer Vision–ACCV 2012 (pp. 705-717). Springer Berlin Heidelberg, (**2013**).
[3] T. Amemiya and H. Gomi, “Camera pose estimation with a two-dimensional marker Grid for haptic navigation”, In Virtual Reality (VR), 2013 IEEE, IEEE (**2013**), pp. 143-144.
[4] J. Zhang, S. Singh and G. Kantor, “Robust Monocular Visual Odometry for a Ground Vehicle in Undulating Terrain”, In Field and Service Robotics, Springer Berlin Heidelberg (**2014**), pp. 311-326.
[5] K. Konolige, M. Agrawal and J. Sola, “Large-scale visual odometry for rough terrain”, In Robotics Research, Springer Berlin Heidelberg (**2011**), pp. 201-212.
[6] J. Shi and C. Tomasi, “"Good features to track”, Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on. IEEE, (**1994**).
[7] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques”, International journal of computer vision, vol. 12, no. 1, (**1994**), pp. 43-77.
[8] J. Y. Bouguet, “Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm”, Intel Corporation, vol. 5, (**2001**).
[9] K. H. Yang, W. S. Yu, and X. Q. Ji, “Rotation estimation for mobile robot based on single-axis gyroscope and monocular camera”, International Journal of Automation and Computing, vol. 9, no. 3, (**2012**), pp. 292-298.
[10] N. A. Karthik, “Vision System for Autonomous Navigation”, Doctoral dissertation in National Institute of Technology, Rourkela (**2014**).

