The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Temporal and spatial 3D motion vector filtering based visual odometry for outdoor service robot

Giil Kwon, Yeong Nam Chae and Hyun S. Yang

Abstract— **This paper describes a visual odometry algorithm that deals with the nearly degenerated situation caused by a false motion vector generated by independently moving objects, repetitive patterns and wrong depth information that often arise in visual odometry for outdoor service robots. To filter out these false motion vectors, we use temporal and spatial motion vector filter. The temporal motion vector filter uses the previous motion models to filter out abruptly changed motion vectors, and the spatial motion vector filter uses the motion vector's length information and the motion vector's direction information. The direction information of the motion vectors generated by independently moving objects are different from the direction of the vector generated by camera movement in 3D space, and the length information of the motion vector caused by triangulation error is different from the correctly triangulated points. We uses voting scheme to determine primary motion vectors. This algorithm has been tested on a service robot that works in outdoor environment. By using our method, we can deal with independently moving objects and problem caused by repetitive patterns and triangulation errors.**

I. Introduction

LOCALIZATION is an important ability of autonomous mobile robots. Most mobile robots typically localize their position by using wheel odometry, inertial sensor (gyroscopes, and accelerometers) or GPS. However these approaches have some limitations. The result of localization using by wheel odometry is affected by ground conditions, and we cannot apply the wheel odometry technique to non-wheel type robots (walking robots, aero robots and underwater robots). Inertial sensors are prone to drift, and by using differential GPS, we can obtain accurate localization; however, this GPS is very expensive and has some error when GPS signals are blocked, which may occur in urban areas, forests and tunnels. Furthermore, GPS only provides location information, but it's not sufficient for service robot. To navigate the service robot, the service robot needs to know their pose information. Visual odometry, which estimates the ego motion from a sequence of images captured by moving cameras, provides the following advantages: it is not affected by ground conditions; it needs cameras but this method does not require any other sensors; it estimates full 6DOF motion; and it reduces error rates to levels lower than all but the most expensive IMU and GPSs. For this reason, visual odometry has received considerable attention in recent years. In most approaches [1][2][3][4] to visual odometry, a set of feature points is extracted from the images captured by cameras. Theses feature points are tracked over a sequence of images. In a stereo camera or multi-camera system, the 3D information of the feature points is calculated by triangulation. In order to detect and reject outliers in the matches and estimate the camera pose, the outlier removal methods are used, such as RANSAC [11] and LMedS[12]. To reduce drift, the iterative refinement technique is used. This approach assumes a static environment where the only moving object is the camera. However, in the area where the service robot navigates, there are independent moving objects, such as pedestrians, cars, and shadows. So if this approach is used in a dynamic environment like Fig.1, the error rate of this visual odometry approach will increase. This approach relies on an outlier removal method, but this is not insufficient to filter out an outlier caused by independently moving object. In [5], A. I. Comport et al. assumes that the frame rate of the images captured by cameras is sufficiently high to predict present frame image from the previous image by using estimated motion at previous frame. This predicted image of the present image is compared to the present image and this result, the difference between these two images, indicates an independently moving object. A robust M-estimator is used to

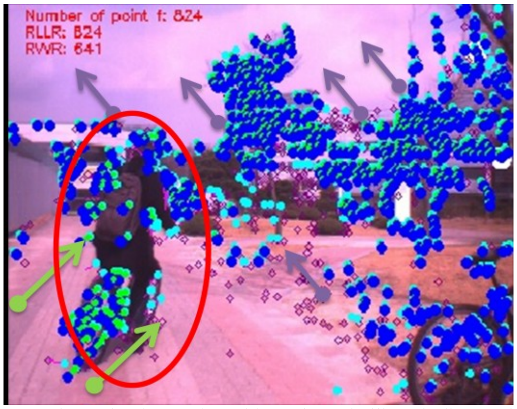


Fig.1 Independently moving object, the red ellipse indicates the motion vector that generated by independently moving object

Giil Kwon is with the Robotics Program, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea (corresponding author, phone: +82-42-350-8717; fax: +82-42-867-3567; e-mail: getupfor1@ paradise.kaist.ac.kr).

Yeong Nam Chae is with the Department of Computer Science, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea (corresponding author , phone: +82-42-350-8717; fax: +82-42-867-3567; e-mail: ynchae@ paradise.kaist.ac.kr).

Hyun S. Yang is with the Department of Computer Science, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea (corresponding author, phone: +82-42-350-8717; fax: +82-42-867-3567; e-mail: hsyang@ paradise.kaist.ac.kr).

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

reject outliers corresponding to moving objects within the scene. However, if the image changes abruptly between frames, such as when the robot moves abruptly or illumination changes between buildings, shadows, the difference between the previous images and predicted image is too big. Therefore, the stationary image area also detects independently moving object area, and the error of pose estimation increase as the visual odometry algorithm works. The error of the visual odometry algorithm has a tendency to accumulate. This may be a critical problem for visual odometry for outdoor service robots.

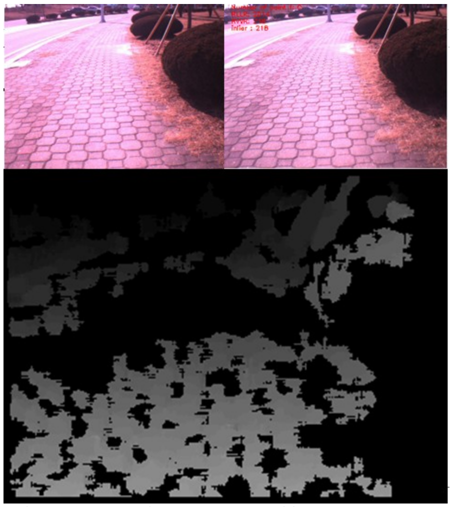


Fig. 2 (Upper image) Stereo images captured by a stereo camera and (lower image) a disparity image calculated by the dense stereo matching algorithm.

To deal with this problem, we present a spatial filtering method that uses the direction of the motion vectors and the length of the motion vectors as well as temporal filtering. The motion vector is the vector that connects the feature point extracted from the previous image to the feature point extracted from the present image. The motion vector indicates the movement of the feature point from the previous image to the present image. We can obtain a motion vector group that has the principal direction and principal length of vectors that indicate the camera motion. However, when the independently moving object takes most of the image's area (more than 50% of the image area), the obtained principle motion vector direction and length do not indicate the camera motion. These motion vectors are generated by an independently moving object.

We present the robust temporal filtering method that detects the correct motion vector that contains camera pose information using the history of the motion model $(\mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3, \cdots, \mathrm{H}_{t-1})$ to complement the flaws of spatial filtering methods.

In most approaches [1][2][3][4][5], researchers developed visual odometry for ground vehicles. So these algorithms use cameras that have wide baseline. (In [1], the camera base line is 21cm.) However, the size of outdoor service robots is limited, as service robots are required to co-exist with people. Therefore, we tested our algorithm using a small baseline stereo camera that can be installed on service robots. To minimize the effect of bad triangulation results, we use spatial and temporal motion vector filtering methods.

It is generally admitted that getting a good precision in depth from stereo vision demands a large base lane. Therefore, if the baseline is small, the triangulation result is poor. If the baseline is wide, the triangulation result is good.

To minimize matching errors, we use SURF [7]. Because the ground outdoors is uneven, for outdoor service robot application, rotations around the optical axis are generally large. If we cannot deal with these motions, the matching result will be poor; consequently, we use SURF. This feature is a scale and rotation invariant interest point detector and descriptor that also has good repeatability. By using this feature, we can deal with large motions around or along the optical axis and can also match points that locate far frame.

II. Visual Odometry for Outdoor Service Robots

A. Dense stereo algorithm

Our proposed visual odometry algorithm assumes that the outdoor service robot is equipped with a stereo camera. The motion of the robot determined from the disparity images processed by dense stereo matching algorithm. The dense stereo matching algorithm performs the following three steps on each new stereo pair:

1) *Low-pass filtering:* A low-pass filter averages out rapid changes in intensity. Noise always changes rapidly from pixel to pixel because each pixel generates its own independent noise. To filter out this noise, we utilize the Laplacian-of-Gaussian filter.

2) *Rectification:* Lenses often cause distortions in raw images. For example, straight lines in the scene often appear curved in the raw images. This effect is particularly evident in the corners of the images. Furthermore, rectified images are corrected so that the rows of images digitized from horizontally displaced cameras are aligned. Without this feature, searching along the rows and columns will not produce the correct results.

3) *Stereo matching*: The stereo processing module applies the Sum of Absolute Differences algorithm.

The inputs for our algorithm are the pre-filtered stereo images and disparity images.

B. Feature detection and matching

The feature points extracted from images are used to estimate the robot pose. In the outdoor environment, the camera on the service robot rocks from side to side, and the camera rotates around the optical axis; therefore, the result of feature matching is not good. In urban environments repetitive patterns such as brick pavement, fences, etc. can be found. These repetitive patterns cannot be distinguished by correlation based methods, and this causes a bad match result.

Although some of these correlation mistakes can be detected using techniques such as the mutual consistency check or the unique maximum criterion, the number of mis matched points will increase [8].

In order to match the stereo images robustly, the speeded-up robust feature (SURF, [9]) is adopted. By using SURF, we can handle these mismatched problems. SURF is robust to scale, viewpoint, and illumination changes. We extract SURF features from key-points and then match this feature to a feature on the other side of the image using the best-bin-first search algorithm [10], which is designed to efficiently find an approximate solution to the nearest neighbor search problem in high dimensional spaces based on the kd-tree search strategy.

The error in the 3D reconstruction of these points is not large enough to be rejected by the RANSAC algorithm, so they will corrupt the final solution.

C. Spatial and temporal motion vector filtering

We detect correctly matched motion vectors based on their temporal and spatial information in estimating the robot motion model. Based on spatial and temporal information of motion vectors, we establish the spatial motion vector filter and temporal motion vector filter.

1) Spatial motion vector filter: we detect spatially principle motion vectors based on their direction and length.

We can obtain the vector directional vector's end point:

$$(x_{N,}y_N,z_N)=\frac{(x_p-x_v,y_p-y_v,z_p-z_v)}{\sqrt{(x_p-x_v)^2+(y_p-y_v)^2+(z_p-z_v)^2}}$$

(1)

The directional vectors are normalized with their lengths and are moved to the origin. These points have the same length. As shown Fig. 3(a), the end points of directional vectors are on a sphere that has a unit length radius, so this directional vector has only the directional information of the motion vector. We use the voting method to obtain principle motion vector group based on the directional vector, so we can convert the 3D point into two angle values:

$$\varphi=\arctan\frac{y_N}{x_N} \qquad (2)$$
$$\theta=\arctan\frac{z_N}{\sqrt{{x_N}^2+{y_N}^2}}$$

As shown in Fig.3 (c) , θ and φ are two angles that represent the direction of the motion vector. We use the voting algorithm to obtain the principle motion vector directions. We divide the angle into equally spaced bin, and we count the number of point included in each bin. The bin that has the maximum number of points is the principle bin, and the points included in this bin represent the principle direction of the motion vectors. These motion vectors represent the principle motion shown in the images, and if a static scene occupies most of the image area, these motion vectors represent the camera motion.

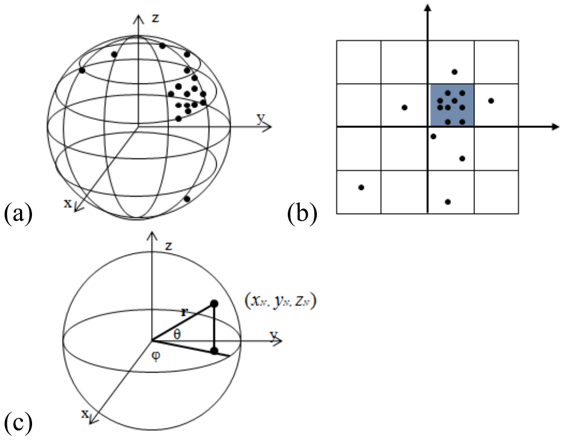


Fig. 3 (a) The end points of directional vectors are located above the surface of the sphere that has unit length radius. (b) We can obtain the primary bin that indicates the principle direction of the motion vector. (c) The end points of the directional vectors are converted into two angle pairs (φ, θ) Stereo images are captured by a stereo camera and the disparity image is calculated by the dense stereo matching algorithm.

We also use the length of the motion vector to obtain primary motion vectors. We didn't use the voting method to obtain primary length of the motion vector. In this case, if we divide bins equally, we cannot filter out false match effectively. If the maximum length of motion vectors is too big, the size of the bin also increases, and the false motion vector filtering ability will be decrease, so we cluster the motion vectors by their length and we chose the bin that has maximum number of motion vectors.

2) Temporal motion vector filter: We can obtain the motion vector group that has principal direction and principal length of vectors that indicate the camera motion. However, when the independently moving object takes up most of the image area (more than 50%), the obtained principle motion vector direction and length do not indicate the camera motion. These motion vectors are generated by the independently moving object. We assume that the frame rate is sufficiently high, so the movements from each frame are small and the movement is smooth. The temporal filter uses previous motion models. We use a temporal filtering method that detects the correct motion vector that contains camera pose information using the history of motion model $(H_1, H_2, H_3, \cdots, H_{t-1})$ to complement the flaws of spatial filtering methods.

$$\varepsilon_H^i=\sum_{j\in Q}(P_r^i-H_jP_v^i)^2 \qquad (3)$$

where Q is the group of previous motion models $(H_1, H_2, \cdots, H_{t-1})$, obtained in the previous frame, and the points P_v, P_r are the matched points that is filtered by the spatial motion vector filtering method.(P_v is the matched point of P_r in the previous frame) ε_H^i is the error that represents the difference of the present point i and predicted previous frame point using the history of the motion model.(ε_H^i is the error of motion vector i, obtained by Equation (3)). If ε_H^i is greater than some fixed threshold, this motion vector is rejected, which means the selected motion vector is not related to the previous motion model. By using the temporal motion vector, we can reject the abrupt motion vector and obtain smooth motion vectors.

By using spatial and temporal motion vector filters, we can obtain primary and smooth motion vectors.

D. Motion estimation

The input of our motion estimation algorithm is matched points computed by feature matching algorithms. These have matching points extracted from the left and right image and have inter-frame matching points. By using these matches, our motion estimation algorithm estimates the pose of the stereo camera. Our approach uses RANSAC to estimate the camera pose. We select three points in the image. For each of the three points, we estimate H and evaluate H to obtain motion model H and inliers (the group of three 3D points). To minimize the error of H, a non-linear minimization method is applied for the inlier sets [2].

1) *Select three points:* Three points are required to generate a motion hypothesis. To achieve reliable motion, we must ensure that these three points are spaced out well in the image. If these points are too close together in the image then the estimation results will be unsatisfactory. Therefore, we divide the image into equally spaced areas and select points in these different areas each time. By using matching information, we triangulate these three points, and we can obtain their 3D locations Pi and Pi'

2) *Estimate hypothetical motion model (H):* H (Hypothetical motion model) represents the camera pose. We then seek an H that satisfies Equation (4).

$$P_r = HP_v \tag{4}$$

P_r denotes the matrix of the three selected 3D points acquired at some time r and P_v denotes the matrix of three 3D points acquired at some time v < r. We estimate H through a closed-form solution of the absolute orientation problem[9].

3) *Evaluate hypothetical motion model:* For all matches M, we evaluate this hypothetical motion model H to find the inliers and H. These inliers refer to correspondences between the two sets of 3D points (P_v, P_r) that make re-projection error ϵ smaller than the threshold that we have determined:

$$\varepsilon_i = (P_r^i - HP_v^i)^2 + (P_v^i - H^{-1}P_r^i)^2 \tag{5}$$

$$\textit{Where, if the } \epsilon_i < t\hbar et\hbar old \textit{ than } i \in I \textit{ else } i \notin I$$

In Equation (5), ϵ means re-projection error. We can determine inlier set I. The RANSAC is applied for a fixed number of samples. For each of the selected three points, we can estimate motion model H and the number of inliers calculated by H. The H that has the largest number of inliers is the best motion model. To obtain more accurate motion model, we correct H through a closed-form solution of the absolute orientation problem [10]. In that case, we estimate the motion model by using all inliers matches.

4) *Non-linear minimization*: We use the Levenberg -Marquardt algorithm for nonlinear least squares minimization. The estimated H is used as the initial point for a non- linier minimization algorithm. For inlier set I, the object function to be minimized is given by Equation (3).

$$H = \arg\min_H \sum_{i \in I} \varepsilon_i \tag{6}$$

The non-linear minimization process converges to a local minima within five iterations, because the initial points of the non-linear minimization process are good estimated values.

III. Experiment

As shown in Fig 4, our system consists of a 2 wheel mobile robot, which is equipped with stereo camera and a laptop PC. The primary vision sensor on this mobile robot is a downward looking stereo camera pair from Point Grey Research (Bumblebee 320X240 monochrome with 40 degree horizontal field-of-view and a baseline of 0.12m).

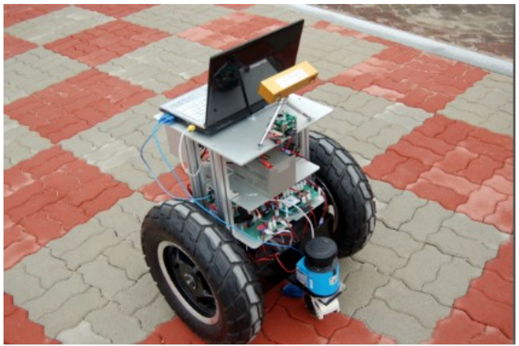
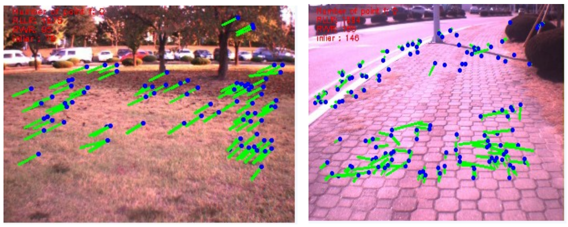


Fig. 4 The experiment environment and the two wheel mobile robot used our experiment

In order to evaluate the performance of our visual odometry algorithm, we applied it to a set of collected video sequences. Specifically, each video sequence was recorded in real-time while a mobile robot was traveled along a predefined trajectory.

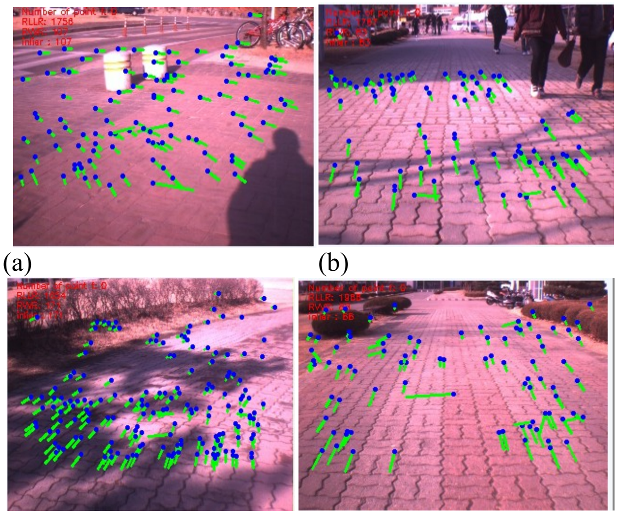


(a) (b)

Fig. 5 (a) our matching algorithm works well. (b)In an urban area, the matching result is unsatisfactory

To obtain the ground-truth trajectory, we utilize the trajectory measured by GPS and manually correct this trajectory based on a predefined location on the map.

As shown Fig. 5(b), the experiment area is urban. There are many independently moving objects, such as cars, peoples and bicycles. In urban environments, repetitive patterns such as brick pavement, fences, etc. can be found. The feature points in these repetitive patterns do not match well, so the visual odometry in this environment is very challenging and practical problem. But most of other approaches test their algorithms in outdoor areas, such as plains and forest. These areas do not contain repetitive patterns, and the number of extracted feature points is also large. Therefore, the matching rate of the feature points extracted in these areas is better than that of the feature points of urban areas.



(a) (b)

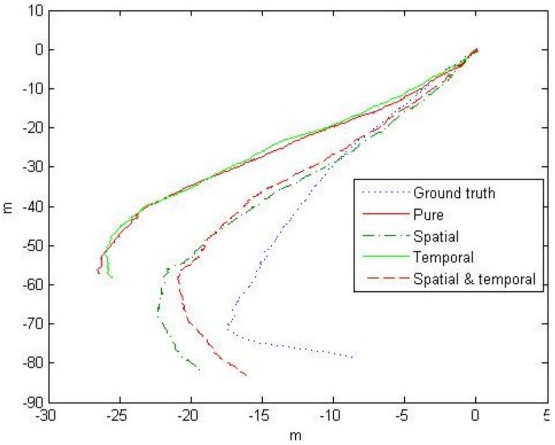
(c) (d)

Fig. 6 (a) When we applied our approach, the spatial and temporal filtering method, the motion vectors of moving shadows are rejected.(b) The motion vectors of pedestrians are also rejected (c) Although there are illumination changes , the matching result is good. (d) In an urban. Area our approach works well

To filter out false matches generated by repetitive patterns, independently moving objects, and triangulation errors, we developed the spatial and temporal filter algorithm. As shown Fig 6, our approach rejects falsely matched motion vectors and maintains well-matched motion vectors.

Fig .7 shows the result of the visual odometry algorithm. We tested our algorithm on the sidewalk. The raw data was logged at a 320x240 resolutions at 30Hz. The accuracy of the visual odometry is indicated by the estimation error, defined as the root-mean-squared distance between the estimated points and the key points of the ground truth. In Fig.7, we can see that the RMS error of the visual odometry is reduced from 14.2624m to 6.037m. There are many independently moving objects and many bricks that create a repetitive pattern. The result of the spatial and temporal filtering method is better than the visual odometry algorithm, which does not apply our algorithm. The trajectory generated by the spatial and temporal filtering method is especially accurate.

In Fig.8 we can see that the error rate of the spatial filtering method is increased at the front part and then drop to zero. In this area, because the number of extracted feature point is small. So the estimated robot trajectory is unstable. Besides the spatial filtering method also decrease the number of feature point. Therefore the estimated result is unstable in this area. But the error rate of the spatial and temporal method is smaller than that of original visual odometry approach, and the spatial filtering method also works well. The error increasing rate of our approach is slower than pure visual odometry.



Distance	150m	Rate	10Hz
Frame	1500	Resolution	320x240

	Original	Spatial	Temporal	Spatial & temporal
RMS Error(meter)	14.2624	6.3522	13.6644	6.0737

Fig. 7 Experiment results

As shown in Fig.9 the maximum error is suppressed. The maximum error of pure visual odometry is 0.5, but the maximum error of spatial and temporal filtering visual odometry is 0.3. By using our algorithm, we can minimize the maximum error rate and also minimize the overall size of the error generated at each frame. Fig.9 shows us the size of the error occurring at each frame. We calculate this error by subtracting the previous error from the present error.

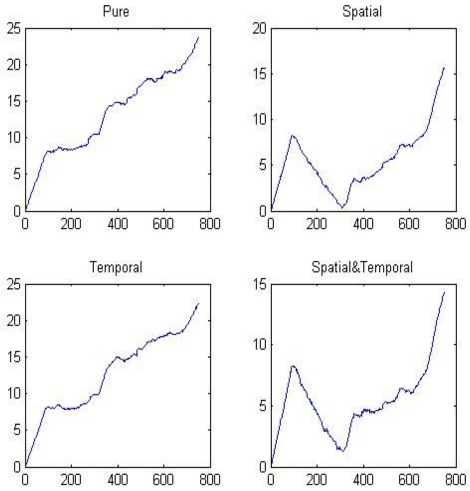


Fig.8 This graph indicates the error rate of the experiments. The pure visual odometry algorithm’s error rate increases very rapidly, but the error rate of the spatial filtering approach and spatial and temporal filtering increase slowly.

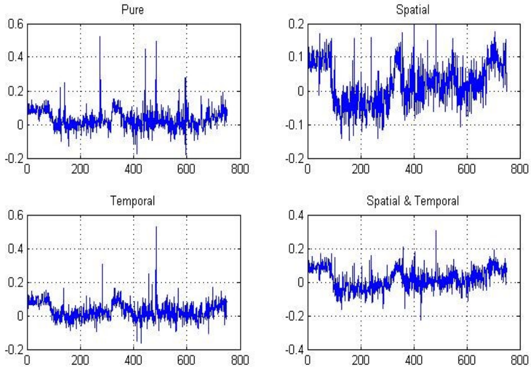


Fig.9 This graph indicates the difference between errors, error(t) and error(t-1)

IV. Conclusion

We have presented a novel approach to stereo visual odometry that successfully deals with the problems encountered in outdoor visual odometry for service robots. The key component of our approach is a spatial motion vector filter and temporal motion vector filter. We obtain the spatially primary motion vector based on voting methods, and we obtain the temporally primary motion vector by using previous motion models. We implemented our filtering method and compared it to a normal visual odometry method that does not apply our method.

By using our method, we can minimize the error occurrence in each frame and can deal with independently moving objects and problems occurring due to repetitive patterns. The stability of the spatial filtering method can be enhanced by applying the three-point algorithm [1]. And the accuracy of the presented system can be also improved by applying the three-point algorithm, by using landmarks to provide longer range constraints, and by using IMU or DGPS.

Acknowledgment

This research is supported by Foundation of Healthcare Robot Project, Center for Intelligent Robot, the Ministry Of Knowledge Economy (MKE)

References

[1] D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry for ground vehicle applications”, Journal of Field Robotics, 23(1):3-20,Jan 2006

[2] M. Agrawal and K. Konolige. “Real-time localization in outdoor environments using stereo vision and inexpensive GPS”,*In International Conference on Pattern Recognition(ICPR)*,volume 3, pages 1063-1068,2006.

[3] C. Olson, L. Matties , M. Schoppers, and M. Maimone, “Rover navigation using stereo ego-motion”, *Robotics and Autonomous Systems,* vol. 43, no. 4, pp. 215-229, 2003.

[4] Howard, A., “Real-time stereo visual odometry for autonomous gound vehicles”*, IEEE/RSJR International Conference on Intellignet Robots and Systems,* Nice, France.

[5] A.I.Comport, E.Malis, P.Rives, “Real-time Quadrifocal Visual Odometry”, *The International Journal of Robotics Research*, Jan 5, 2010.

[6] H. Hirschmuller, PR Innocent, JM Garibaldi, “Fast, unconstrained camera motion estimation from stereo without tracking and Robust statistics”, In *Control, Automation, Robotics and Vision (ICARCV’02),* pages 1099-1104,2002.

[7] Er H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features(SURF)”, International Journal on Computer Vision, pp. 404-417,2006..

[8] I Parra, MA Sotelo, DF Llorca, M Ocaña, , “Robust visual odometry for vehicle localization in urban environment”, *Robotica*, 2009

[9] BKP Horn. “Closed-form solution of absolute orientation using unit quaternions”, Journal of the Optical Society of America A,1987.

[10] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns, *IEEE Trans. Pattern Analysis and Machine Intelligence, 13(4),* April 1991.

[11] R. Bolles and M. Fischler, “A RANSAC-based approach to model fitting and its application to finding cylinders in range data”, in *Intl. Joint Conf. on AI(IJCAI),* Vancouver, BC, Canada, 1981, pp. 637-643.

[12] P. Rousseeuw and A. Leroy.” Robust Regression and Outlier Detection.”, Wiley Series in Probability and Mathematical Statistics, 1987.