

# Real-time Visual Odometry Estimation Based on Principal Direction Detection on Ceiling Vision

Han Wang<sup>1</sup> Wei Mou<sup>1</sup> Gerald Seet<sup>2</sup> Mao-Hai Li<sup>3</sup> M. W. S. Lau<sup>4</sup> Dan-Wei Wang<sup>1</sup>

<sup>1</sup>School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore

<sup>2</sup>School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore

<sup>3</sup>School of Mechanical and Electrical Engineering, Soochow University, Suzhou 215006, China

<sup>4</sup>School of Mechanical and System Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, UK

**Abstract:** In this paper, we present a novel algorithm for odometry estimation based on ceiling vision. The main contribution of this algorithm is the introduction of principal direction detection that can greatly reduce error accumulation problem in most visual odometry estimation approaches. The principal direction is defined based on the fact that our ceiling is filled with artificial vertical and horizontal lines which can be used as reference for the current robot's heading direction. The proposed approach can be operated in real-time and it performs well even with camera's disturbance. A moving low-cost RGB-D camera (Kinect), mounted on a robot, is used to continuously acquire point clouds. Iterative closest point (ICP) is the common way to estimate the current camera position by registering the currently captured point cloud to the previous one. However, its performance suffers from data association problem or it requires pre-alignment information. The performance of the proposed principal direction detection approach does not rely on data association knowledge. Using this method, two point clouds are properly pre-aligned. Hence, we can use ICP to fine-tune the transformation parameters and minimize registration error. Experimental results demonstrate the performance and stability of the proposed system under disturbance in real-time. Several indoor tests are carried out to show that the proposed visual odometry estimation method can help to significantly improve the accuracy of simultaneous localization and mapping (SLAM).

**Keywords:** Visual odometry, ego-motion, principal direction, ceiling vision, simultaneous localization and mapping (SLAM).

## 1 Introduction

Localization is a key issue for the autonomous navigation of robots. It is also a prerequisite for many applications like obstacle detection, augmented reality and many other tasks. For ground vehicles, it is typically performed by combining the wheel odometry (from wheel encoders) and the inertial sensing (gyroscopes and accelerometers). However, this approach has several limitations. First, the precision depends on very exact measurements of the robot wheels, wheel base and slip conditions. Second, the inertial sensors are prone to drift. Moreover, the wheel odometry is unreliable in rough terrain (wheels tend to slip and sink) or it may be lifted and moved. In recent years, camera systems are getting cheaper, and more compact and computational ability increase dramatically even on standard PC. These developments make it possible to process high quality images at a high frame rate. The information implied in sequential images can be used to estimate the camera motion accurately. Visual odometry is the process of estimating the pose of a mobile robot incrementally using only camera input. It has been shown that compared to wheel odometry, visual odometry provides more accurate trajectory estimations<sup>[1]</sup>.

A variety of approaches for visual odometry tasks have been introduced in the past. Based on the sensor type, these approaches can be divided into two categories: Monocular camera and stereo camera. Nister et al.<sup>[2, 3]</sup> demonstrated the successful results of visual odometry estimation over long distances using perspective and omnidirectional cameras, respectively. Davison<sup>[4]</sup> also presented a

real time monocular system using the extended Kalman filter. Related work of monocular camera based approaches can be further divided into two categories: Feature based approaches and appearance based approaches. For the first category, Nister<sup>[5]</sup> presented one of the first visual odometry system using a monocular camera. They used a five-point solver<sup>[6]</sup> to estimate ego-motion using random sample consensus (RANSAC). For the second category, Yu<sup>[7]</sup> mounted a downward looking camera on the vehicle and presented a method to compute visual odometry using template matching. Besides monocular camera, additional sensors can be utilized to improve the estimation accuracy. Yang et al.<sup>[8]</sup> used both monocular camera and single-axis gyroscope to produce better rotation estimation results compared to using a single sensor. In order to have an accurate visual odometry calibration, the relative pose between cameras and other sensors needs to be calibrated. Xu et al.<sup>[9]</sup> offered an effective way to calibrate the relative poses between inertial and camera mounted on a mobile robot system.

Compared to the techniques using monocular cameras, motion estimation from stereo cameras is relatively easy and tends to be more stable and well-behaved. The major advantage of using stereo cameras is that one does not need to worry about the scale ambiguity present in monocular camera techniques<sup>[10]</sup>. It can be further separated into feature tracking over a sequence of images<sup>[5, 11, 12]</sup>, or feature matching between two consecutive images<sup>[13, 14]</sup>. Given a calibrated stereo camera, the 3D scene can be reconstructed and the iterated closest point (ICP) algorithm<sup>[15]</sup> can estimate the ego-motion between two consecutive images. Additional sensors can be combined with cameras to improve the accuracy of the visual odometry. Dornhege

and Kleiner<sup>[12]</sup> used an inertial measurement unit (IMU), while Agrawal and Konolige<sup>[11]</sup> used the global positioning system (GPS) together with wheel encoders and IMU to complement the visual odometry system. Compared to these approaches, our method focuses on estimating the robot motion with high accuracy completely based on visual inputs.

The core issue for 3D visual odometry is to register point clouds from various time instants into one reference coordinate system to obtain 3D motion parameters. The common method for point cloud registration is ICP. The general idea is to iteratively assign correspondences between the points of the two point clouds and to update motion parameters until the system converges. However, ICP is prone to local minimum as the registration tends to reinforce a possibly suboptimal initial point correspondence. Alternatively, monocular images can be used to extract key-points and match them with previous frames using descriptor matching to roughly align two point clouds<sup>[16]</sup>. The correct data correspondences of ICP are much more likely to be found in pre-aligned point clouds. As a result, the registration performance can be improved. However, the runtime for key point detection and descriptor extraction is unable to reach real-time performance without using a graphics processing unit (GPU). Moreover, descriptor based methods suffer from the drawback that they cannot distinguish the repeated patterns.

All frame-to-frame based algorithms suffer from error accumulation problem. Although motion estimation error between two consecutive frames can be small enough, this error accumulates over time and leads to a bad estimation of robot position after a long run. The accumulated error can only be corrected by loop closing technique. However, besides additional computation power and sensor requirements, loop closure detection can introduce new error and result in an even worse position estimation.

The main contribution of this paper is that we propose a real-time principal direction detection approach that acts as a global reference to reduce accumulated error in frame-to-frame based methods. Also, it can perform well with repeated patterns since the direction detection is not completely based on local intensity changes. In many buildings, ceilings contain vertical and horizontal lines, and the orientation of these lines can be used as a global reference for robot's orientation. Even in the case where no typical lines or markings on the ceiling, the edges between walls and ceiling can be also utilized as the principal direction.

We used a Microsoft Xbox Kinect camera in our proposed approach. The richness of their data and the recent development of low-cost sensors have combined to present an attractive opportunity for mobile robotics research. We mounted the Kinect on the robot and pointed it to the ceiling. Ceiling vision has the advantage of no or little obstruction. Also, compared to the frontal view in which the scale of images may change, ceiling vision only involves rotation and affine transformation. The main steps for the proposed principal direction detection approach are as follows. First, local linear features are extracted from monocular images grabbed by the Kinect camera pointing up to the ceiling. Second, the ceiling plane is detected using depth map obtained by the Kinect, and local features not belonging to

the ceiling are eliminated to improve the accuracy. Then hough transform is performed to extract lines and the principal direction is determined by a voting scheme. After identifying the principal direction, visual odometry can be obtained by using ICP to register two consecutive frames.

The structure of this paper is as follows. In Section 2, more details for principal direction detection are described. Section 3 shows how the detected principal direction is used to estimate the relative transformation between two consecutive frames. The current robot position in global coordinate can be retrieved by simply accumulating the relative transformation matrix. In Section 4, real world experiments are carried out to show the stability of the proposed visual odometry estimation approach and its application to simultaneous localization and mapping (SLAM) problem.

## 2 Principal direction detection

The principal direction detection refers to the detection of yaw ( $\theta_z$ ) value which is the rotation around the normal of the plane that the robot is currently on. The non-stationary objects, such as moving people, cannot be used to detect the principal direction of the image. In order to have a better principal direction estimation, we mounted the Kinect on the robot and pointed it to the ceiling. Calibration and rectification are performed using the toolbox proposed in [17] to achieve undistorted frames.

Although the principal direction is a global feature of the environment, its calculation is based on local evidence extracted from monocular images. Edges have been recognized as critical features in image processing since the beginning of computer vision. Compared to point features, edges are less distinctive, which makes it more challenging for data association problem. However, the invariance of edges to orientation and scale makes them good candidates for tracking. Moreover, the runtime for edge detection is much faster compared to scale-invariant feature transform (SIFT)<sup>[18]</sup> or speeded up robust features (SURF)<sup>[19]</sup>. Canny's algorithm<sup>[20]</sup> for choosing edges in an image has emerged as the standard technique, and consistently ranks well in comparisons<sup>[21]</sup>. We use it for our edge feature selection algorithm. We first smooth the data using a symmetric 2D Gaussian blur kernel to remove some noise from the data (Fig. 1 (a)). After that, the Canny algorithm is adopted to generate an edge image (Fig. 1 (b)).

We mounted a Kinect pointing up towards the ceiling. However, the data acquired may still contain parts from walls or lamps. These undesirable parts decrease the accuracy of principal direction detection. We eliminate this effect by extracting points only from the ceiling plane which cannot be correctly detected by using monocular images alone. As our data are grabbed from a Kinect, every point has a 2D position (pixel coordinates in the image) and its corresponding 3D position (the measured position in the world coordinate system). However, the ceiling plane cannot be determined through simple height filtering because the robot may have a non-zero roll ( $\theta_x$ ) or pitch ( $\theta_y$ ) angle. Our ceiling plane detection approach has two assumptions: First, the robot starts with  $\theta_x = 0$  and  $\theta_y = 0$ , which means that at the start point, points from the ceiling can be determined through simple height filtering. Second, the slope

change between two consecutive frames is small, which is reasonable if the overall processing time for each frame is short. At the first frame, 3 non-collinear points at certain height are selected as points from the ceiling and a plane equation is estimated from these points. Then, for each following frame, the plane detection and outlier removal steps are listed as follows:

- 1) Use the plane calculated in the previous frame as initial hypothetical plane for the current plane estimation.
- 2) Points that are close enough to this initial hypothetical plane are selected as set  $S$ .
- 3) A RANSAC-fashioned approach<sup>[22]</sup> is applied to  $S$  to determine the current ceiling plane and eliminate all points that do not belong to this plane.
- 4) All the remaining 3D points are projected back to 2D image pixels, as shown in Fig. 1 (c).

Because the ceiling detection for the current frame is based on the last detected ceiling plane, this plane detection approach is adaptive to small changes in ceiling slope. If more than one plane (such as ceiling and walls) exists in the same frame, this approach can still recognize the ceiling correctly even when only a small portion of points are from the ceiling.

After projecting the 3D ceiling points back to the 2D image, the hough transform is used to extract lines from these back-projected edge pixels. To reduce the effect of noisy data and to improve the algorithm stability, only those lines whose hough votes are higher than a threshold are selected, as shown in Fig. 1 (d). For hough transform, lines are expressed in polar system as

$$r = x \cos \theta + y \sin \theta \quad (1)$$

where  $r$  is the distance from origin to the line and  $\theta_z$  indicates the orientation of the line. We set angular resolution of  $\theta_z$  as 1 degree. We made an assumption that the current detected principal direction is close to the previous detected

one. This is a reasonable assumption since we use only two consecutive frames to calculate the transformation and the frame rate of Kinect camera can be up to 30 Hz. As a result, the searching space for  $\theta_z$  is much narrower and the line extraction process is faster. Finally, the most voted angle value is selected as the principal directions.

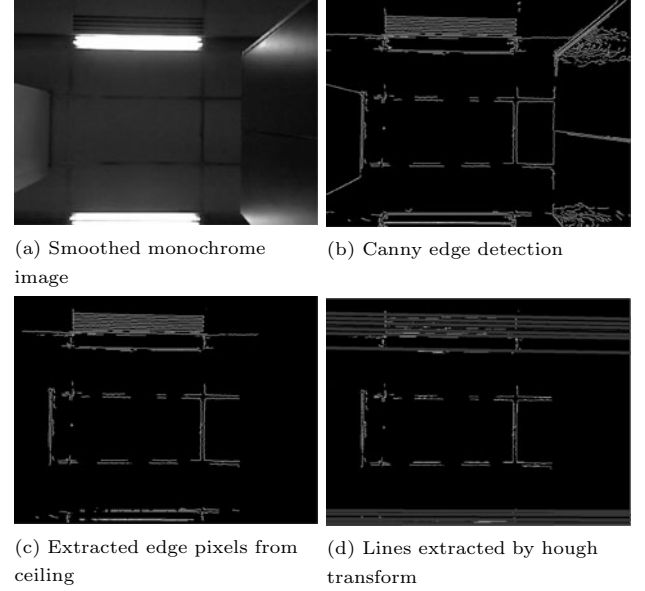


Fig. 1 Line extraction of ceiling image

However, if the robot has a non-zero roll or pitch angle, the perspective distortion will lead to a wrong estimation for the principal direction as shown in Figs. 2 (a)–(d). Lines that are parallel in the real world tend to intersect at some point. This error will lead to a wrong estimation of the principal direction. Fortunately, it can be corrected by utilizing the plane detected in the previous step.

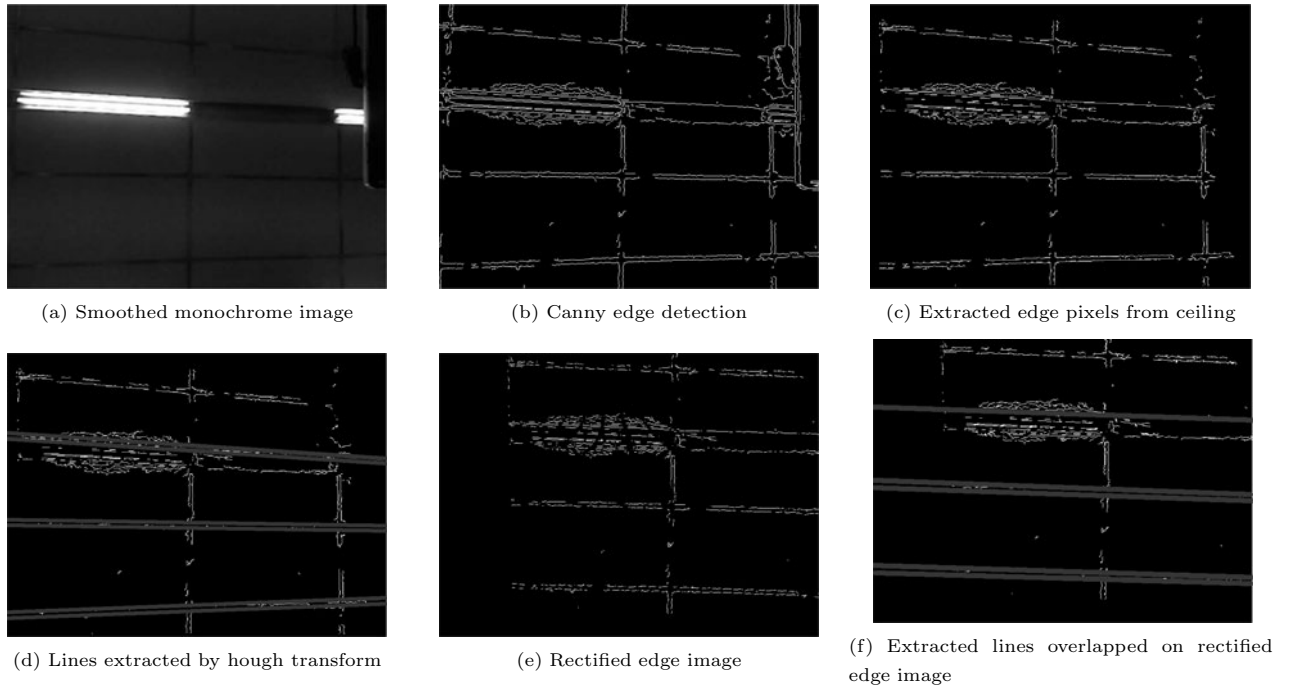


Fig. 2 Lines extraction under perspective distortion

Given detected ceiling plane  $ax + by + cz + d = 0$ , the plane normal can be expressed as

$$\vec{n} = \left( \frac{a}{\sqrt{a^2 + b^2 + c^2}}, \frac{b}{\sqrt{a^2 + b^2 + c^2}}, \frac{c}{\sqrt{a^2 + b^2 + c^2}} \right). \quad (2)$$

Roll ( $\theta_x$ ) and pitch ( $\theta_y$ ) of the ceiling plane can be calculated using normal  $\vec{n}$ :

$$\begin{aligned} \theta_x &= \arctan \frac{\vec{n}(1)}{\vec{n}(3)} \\ \theta_y &= \arctan \frac{\vec{n}(2)}{\vec{n}(3)} \end{aligned} \quad (3)$$

where  $\vec{n}(i)$  denotes the  $i$ -th element in vector  $\vec{n}$ .

Given a 3D point in camera coordinate  $X$ , its corresponding pixel  $x$  in image is

$$x = K [I | 0] \begin{bmatrix} X \\ 1 \end{bmatrix} = KX \quad (4)$$

where  $K$  is the calibration matrix obtained in a previous calibration step. Rotation matrix  $R$  can be calculated by the estimated  $\theta_x$  and  $\theta_y$  as

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix}. \quad (5)$$

Because the principal direction is the estimation for the robot's yaw rotation ( $\theta_z$ ), the rectification process needs to make the ceiling in the monocular image horizontal to the image plane ( $\theta_x = \theta_y = 0$ ). The principal direction can be determined using the algorithm described before. The actual translation of camera does not affect the detection of principal direction. Thus, we assume there is no translation or  $\theta_z$  changes between the distorted image and rectified image. The rectified pixel  $x'$  can be expressed as

$$x' = K [R | 0] \begin{bmatrix} X \\ 1 \end{bmatrix} = KRX. \quad (6)$$

From (4) and (6), we get

$$x' = KRK^{-1}x \quad (7)$$

where  $KRK^{-1}$  is a  $3 \times 3$  homography matrix. The rectified edge image and its extracted lines can be seen in Figs. 2(e) and 2(f), respectively.

The detected principal direction  $\theta \in [0, 2\pi]$ . Thus, one principal direction represents two possible headings of the robot. However, this orientation ambiguity does no harm to visual odometry estimation. Although the principal direction is used to guide frame-to-frame based visual odometry estimation approach like ICP, the relative rotation changes between two consecutive frames cannot exceed  $\pi$ . Therefore, the ambiguity can be resolved while determining the relative transformation.  $\theta_z$  is simply calculated as the relative change between the two consecutive principal directions.

### 3 Visual odometry implementation

Given  $\theta_x$ ,  $\theta_y$  and  $\theta_z$ , we can roughly align two consecutive point clouds filled with edge points from the ceiling. After that, ICP is used to fine-tune the transformation parameters and provide 6-DOF (degree of freedom) motion estimation using these two pre-aligned point clouds. The system overview is shown in Fig. 3.

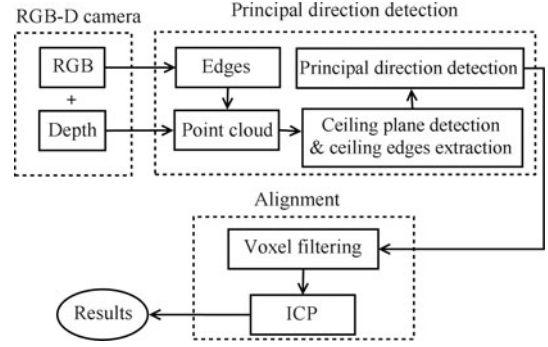


Fig. 3 System overview

In order to reach real-time performance, the size of point cloud needs to be reduced. We divided the space into a set of tiny 3D boxes and all points inside each box are approximated with the coordinates of their centroids. By default, we choose the voxel size to be of 0.05 m which allowed us to down-sample the point cloud by an order of magnitude, reaching real time requirement and having enough data for performing the alignment procedure later. Moreover, after this operation, points density no longer depends on their distances from the sensor.

ICP has the drawback of local minima. Hence, a reasonable initial guess of transformation is needed in order to have a good estimation result. The detected principal direction  $\theta_z$  together with  $\theta_x$  and  $\theta_y$  can be used as initial guess for rotation parameters. For translation, a simple alpha-beta filter based on the previous motion is applied. All points in the voxel grid are transformed into point cloud  $P_s$  based on this initial guess.

Given two pre-aligned point clouds, ICP is used to fine-tune the motion estimation. In ICP, points in a source cloud  $P_s$  are matched with their nearest neighboring points in a target cloud  $P_t$  and a rigid transformation is calculated by minimizing the  $n$ -D error between associated points. This transformation may change the nearest neighbors for points in  $P_s$ , so the two steps of association and optimization are alternated until convergence.  $P_s$  is transformed into  $P_s'$  using the pre-calculated initial guess. For each point in  $P_s'$ , data association is determined by finding the nearest neighbor in target cloud  $P_t$ . While it is possible to compute associations between points based on a combination of Euclidean distance, color difference, and shape difference, we found Euclidean distance along with a fast kd-tree search is sufficient in most cases.

Point-to-plane ICP has been shown to generate more accurate alignments than point-to-point ICP due to an improved interpolation between points<sup>[23]</sup>. However, since almost all points in  $P_s'$  and  $P_t$  come from the ceiling, point-to-plane error metric is not feasible in our case and we em-

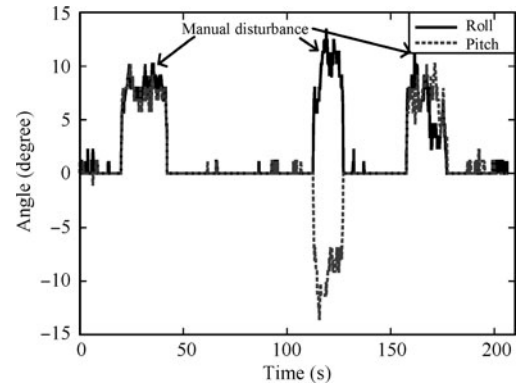
ploy the point-to-point error metric instead. The current robot location and orientation in global coordinate can be retrieved by accumulating the relative rotation and translation matrix.

## 4 Experiments

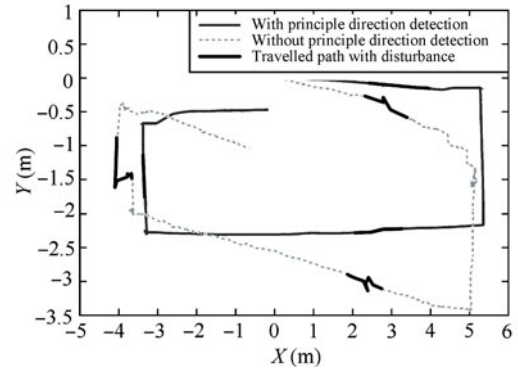
In this section, several experiments are conducted to demonstrate the effectiveness and the stability of our principal direction detection approach for visual odometry estimation. The experiments were carried out using an iRobot Create platform. We used the Kinect camera which is pointed up to the ceiling to test our visual odometry system. The robot was manually controlled with a wireless game pad and the testing data were collected by driving the robot in the Robotic Research Center of Nanyang Technological University.

In each frame, around 8000 edge points were detected. Only about 3000 points were left after voxel filtering, and they were used to calculate the relative transformation between two consecutive frames. On a notebook with an Intel i7-2710-QE 2.10G Hz processor with 4 GB of memory, the average number of frames per second for our visual odometry algorithm is about 10. Edge detection takes about 0.01 s. It takes 0.009 s and 0.007 s for ceiling plane detection and principal direction detection respectively, while 0.07 s is required for ICP. Fig. 4 shows the stability of our system under disturbance. During the test, the robot started at position (0, 0) following a clockwise path and returned to the start point at the end. We manually disturbed the robot during the test. Fig. 4(a) shows the roll and pitch angles at different times. In Fig. 4(b), the black solid line is the robot trajectory using the principal direction to guide ICP while the dotted one is without it. Although under obvious disturbance, the path of robot is still smooth and almost closed loop at the end. The bold parts of the paths are places where disturbance occurs. The test emphasizes the effectiveness of the principal direction detection.

We conducted more tests in common office scene to demonstrate the performance of our visual odometry system and made comparison with other conventional odometry options. The result is shown in Fig. 5. The robot started at position (0, 0), heading towards the positive  $X$  direction. Raw wheel odometry is the accumulated position changes over time calculated using only wheel encoders. Dead reckoning means to incorporate raw wheel odometry with IMU sensing. The visual odometry is the result of our proposed algorithm. In Fig. 5(a), the path of raw wheel odometry and dead reckoning deviates a lot from the straight line because of the wheel slip and shaking of the robot, while the visual odometry can successfully maintain on the straight line path regardless of the slip. In Fig. 5(b), the robot started at (0, 0), heading towards the positive  $X$  direction. We drove the robot along a “ $\infty$ ” shaped loop and came back to the starting point in the end. Because errors are accumulated over time for both visual and wheel odometry, none of the odometry approaches managed to close the loop. However, visual odometry still outperforms conventional odometry methods.

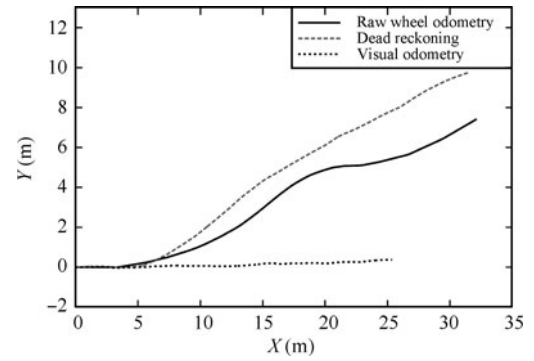


(a) Roll and pitch angle during robot's movement

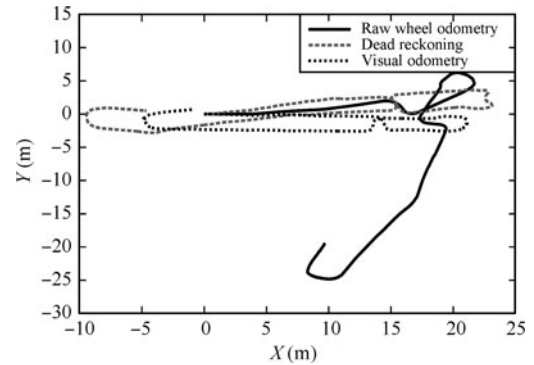


(b) Robot trajectory

Fig. 4 Robot trajectory with disturbance



(a) Robot move straight forward



(b) Robot move along “ $\infty$ ” shaped loop

Fig. 5 Comparison of visual odometry and conventional visual odometry



Fig. 6 Kinect mounted on iRobot Create platform

Odometry is commonly used in different SLAM algorithms. A more precise odometry can produce a better map. We fed the three odometry results mentioned above into the Gmapping algorithm<sup>[24]</sup>. A Hokuyo Laser Range Finder (Fig. 6) is used to get measurements from surround-

ings and the produced map can be seen in Fig. 7. The first row is the Gmapping result based on raw wheel odometry while the second and third rows are based on dead reckoning and visual odometry, respectively. Because of the wheel slip, the maps are distorted or have some false edges for raw wheel odometry and dead reckoning cases in varying degrees. Gmapping based on our visual odometry approach generates the best map. In Fig. 7(b), the bottom one shows that the loop successfully closed using Gmapping based on our visual odometry. Although our approach can generate satisfactory results, the small errors still accumulate over time leading to the failure of loop closing. There are two main reasons for these failures. First, there are some long gaps in the logs (each lasting 1 s or more) when the logging disks are unable to sustain the frame rate. Not all of these gaps are successfully bridged by the algorithm, particularly when the image overlap is small. This is more of a system (hard disk) failure rather than a visual odometry algorithm failure. The second failure occurs when the robot enters a narrow corridor. In this case, only a small portion of the image is from the ceiling and the visual odometry works

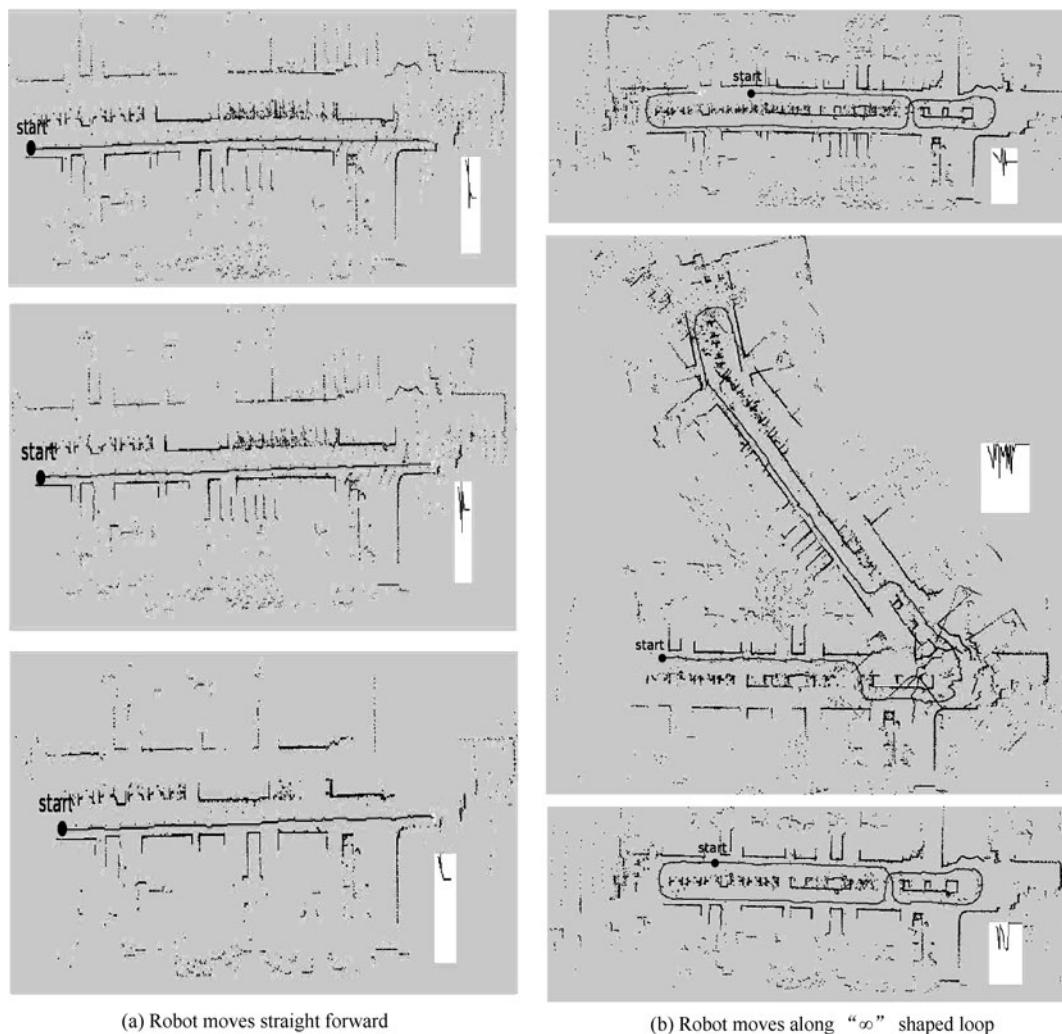


Fig. 7 Gmapping results using different odometry (From top to bottom are results based on raw wheel odometry, dead reckoning and visual odometry, respectively. Black circles indicate the robot's starting point)

with few pixels or not enough overlap area with previous frames. These failures highlight a limitation of our approach. Visual odometry will only work in environments where enough ceiling overlap can be obtained between two consecutive frames.

## 5 Conclusions

In this paper, we presented a real-time method for determining the robot odometry only using ceiling vision. It is clear that the introduction of principal direction detection has greatly improved the visual odometry performance and makes it robust and insensitive to camera disturbance. The method has been evaluated using the Kinect camera system. In an indoor environment with uneven floor, the precision of this visual odometry system is demonstrated to be better than wheel odometry and even better than the combination of wheel odometry and an IMU sensor, especially when wheel slip occurs. It also offers the prospect of substantially reducing the sensing cost. By using this approach, we can replace very expensive inertial measurement units with a much cheaper Kinect camera and produce even better odometry results under many conditions. We also showed in experiments that the proposed visual odometry can significantly improve the SLAM performance.

After presented the capabilities and limitations of pure visual odometry, it should be noted that even the most minimal robot is likely to have some form of proprioceptive sensing (including encoders and/or IMU). Therefore, in future work, we intend to augment rather than replace these sensors, and to work with higher-level pose estimators that fuse data from multiple sources.

## References

- [1] D. M. Helmick, Y. Cheng, D. S. Clouse, L. H. Matthies, S. I. Roumeliotis. Path following using visual odometry for a mars rover in high-slip environments. In *Proceedings of the IEEE Aerospace Conference*, IEEE, Big Sky, MT, vol. 2, pp. 772–789, 2004.
- [2] D. Nistér, O. Naroditsky, J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.
- [3] D. Scaramuzza, R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1015–1026, 2008.
- [4] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the 9th IEEE International Conference on Computer Vision*, IEEE, Nice, France, pp. 1403–1410, 2003.
- [5] D. Nistér, O. Naroditsky, J. Bergen. Visual odometry. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Washington, DC, USA, vol. 1, pp. 652–659, 2004.
- [6] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.
- [7] Y. Yu, C. Pradalier, G. H. Zong. Appearance-based monocular visual odometry for ground vehicles. In *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics* IEEE, Budapest, pp. 862–867, 2011.
- [8] K. H. Yang, W. S. Yu, 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, pp. 292–298, 2012.
- [9] D. Xu, H. W. Wang, Y. F. Li, M. Tan. A new calibration method for an inertial and visual sensing system. *International Journal of Automation and Computing*, vol. 9, no. 3, pp. 299–305, 2012.
- [10] M. Agrawal, K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive GPS. In *Proceedings of the 18th International Conference on Pattern Recognition*, IEEE, Hong Kong, pp. 1063–1068, 2006.
- [11] M. Agrawal, K. Konolige. Rough terrain visual odometry. In *Proceedings of the International Conference on Advanced Robotics*, Daegu, South Korea, vol. 25, pp. 28–30, 2007.
- [12] C. Dornhege, A. Kleiner. Visual odometry for tracked vehicles. In *Proceedings of the IEEE International Workshop on Safety Security and Rescue Robotics*, IEEE, Gaithersburg, Maryland, USA, 2006.
- [13] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Nice, France, pp. 3946–3952, 2008.
- [14] A. Talukder, S. Goldberg, L. Matthies, A. Ansar. Real-time detection of moving objects in a dynamic scene from moving robotic vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Las Vegas, NV, USA, vol. 2, pp. 1308–1313, 2003.
- [15] P. J. Besl, N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–265, 1992.
- [16] P. Henry, M. Krainin, E. Herbst, X. F. Ren, D. Fox. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Proceedings of the International Symposium on Experimental Robotics*, ISER, Delhi, India, vol. 25, pp. 22–25, 2010.
- [17] H. C. Daniel, J. Kannala, J. Heikkilä. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 10, pp. 2058–2064, 2012.
- [18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [19] H. Bay, T. Tuytelaars, L. J. Van Gool. Surf: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision*, Graz, Austria, pp. 404–417, 2006.
- [20] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [21] M. Heath, S. Sarkar, T. Sanocki, K. Bowyer. Comparison of edge detectors: A methodology and initial study. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, San Francisco, CA, USA, pp. 143–148, 1996.
- [22] S. Se, M. Brady. Ground plane estimation, error analysis and applications. *Robotics and Autonomous Systems*, vol. 39, no. 2, pp. 59–71, 2002.
- [23] A. V. Segal, D. Haehnel, S. Thrun. Generalized-ICP. In *Proceedings of Robotics: Science and Systems*, RSS, Seattle, USA, vol. 2, pp. 1–8, 2009.

- [24] G. Grisetti, C. Stachniss, W. Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

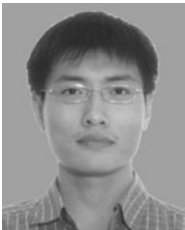


**Han Wang** received his bachelor degree in computer science from Northeast Heavy Machinery Institute, China, and Ph. D. degree from University of Leeds, UK. He is currently an associate professor with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. He has published over 120 top quality international conference and journal papers. He has been invited as a member

of Editorial Advisory Board, the Open Electrical and Electronic Engineering Journal. He is a senior member of IEEE.

His research interests include computer vision and robotics.

E-mail: hw@ntu.edu.sg



**Wei Mou** received his B.Sc. degree in software engineering from Shandong University, China in 2007, and M. Sc. degree in applied computer science from University of Freiburg, Germany in 2010. He is currently a Ph. D. candidate of Nanyang Technological University, Singapore.

His research interests include visual navigation and 3D environment reconstruction.

E-mail: mouwei@ntu.edu.sg (Corresponding author)



**Gerald Seet** received his B.Sc. degree from Council for National Academic Awards in 1979, his M. Sc. degree from University of Dundee, UK, his Ph. D. degree from Aston University, UK in 1985. He is currently an associate professor with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. He holds a concurrent appointment as the director of the Robotics Research Centre.

search Centre.

His research interests include mechatronics and field robotics, with specific interest in underwater mobile robotics.

E-mail: mglseet@ntu.edu.sg



**Mao-Hai Li** received his bachelor, M. Sc. and Ph. D. degrees in computer application from Harbin Institute of Technology China. He is currently an associate professor with the School of Mechanical and Electric Engineering, Soochow University, China. He has published over 40 international conference and journal papers.

His research interests include computer vision and robotics.

E-mail: limaohai@163.com



**M. W. S. Lau** received his B. Eng. degree from Singapore University (now NUS), M. Sc. degree from NUS, and Ph. D. degree from Aston University, UK. He is currently with Newcastle University UK, and teaching mechanical design and manufacturing engineering in Singapore Institute of Technology, Singapore.

His research interests include mechatronics, control of mechanical systems, motion control, active vibration control, underwater robotic vehicles and the design and control of unmanned ground robotic vehicles.

E-mail: michael.lau@ncl.ac.uk



**Dan-Wei Wang** received his B. Eng. degree from the South China University of Technology, China in 1982, his Ph. D. and M. Sc. degrees from the University of Michigan, USA in 1989 and 1984, respectively. Currently, he is professor of control and instrumentation. He is a senator in Nanyang Technological University Academics Council. He has served as general chairman, technical chairman in international conferences, such as *International Conference on Control, Automation, Robotics and Vision (CARCVs)* and *IEEE International Conference on Intelligent Robots and Systems*. He is an associate editor for the *International Journal of Humanoid Robotics* and served as an associate editor of Conference Editorial Board, IEEE Control Systems Society from 1998 to 2005.

His research interests include robotics, control theory and applications.

E-mail: Edwwang@ntu.edu.sg