Proceedings of the 2012 IEEE
International Conference on Robotics and Biomimetics
December 11-14, 2012, Guangzhou, China

Visual Odometry using RGB-D Camera on Ceiling Vision

Han Wang, Wei Mou, Hendra Suratno, Gerald Seet, Maohai Li, M.W.S. Lau[1] and Danwei Wang
School of Electrical & Electronic Engineering
Nanyang Technological University, Singapore
Email:{mouwei, hw, hstju, limaohai, mglseet, edwwang}@ntu.edu.sg
michael.lau@ncl.ac.uk[1]

***Abstract*— In this paper, we present a novel algorithm for odometry computation based on ceiling vision. The main contribution in this algorithm is the introduction of principal direction detection that can greatly reduce error accumulation problem present 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 and these lines 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 current camera position by calculating the translation and rotation to the previous frame. However, its performance suffers from data association problem or it requires pre-alignment information. Unlike ICP, the performance of the proposed approach does not rely on data association knowledge. Using this method, two point clouds are pre-aligned. Hence, we can use ICP to fine-tune the transformation parameters and to minimize registration error. Experimental results demonstrate the performance and stability of the proposed system under disturbance in real-time.**

***Index Terms*— Visual Odometry, principal direction, Ceiling vision, real-time**

I. INTRODUCTION

Localization is a key issue for the autonomous navigation of robots. 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, namely monocular camera and stereo camera. Nister [2] presented one of the first visual odometry system using a monocular camera. Davison [3] also presented a real time monocular system using the Extended Kalman Filter. Compared to techniques incorporating 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 needs not worry about the scale ambiguity present in monocular camera techniques. It can be further separated into feature tracking over a sequence of images ([4], [6], [1]), or feature matching between two consecutive images ([5], [7]). Given a calibrated stereo camera, the 3D scene can be reconstructed and the Iterated Closest Point (ICP) algorithm [8] can be applied to estimate the ego-motion between two consecutive images. Additional sensors can be combined with cameras to improve the accuracy of the visual odometry. In [6], the authors use an IMU while in [4], the GPS together with wheel encoders and IMU are used to complement the visual odometry system. Compared to these approaches, our method focuses on estimating the robot motion 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 the 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 converge. 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 [9]. 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 points detection and descriptors extraction are unable to reach real-time performance without using a GPU. Moreover, descriptor based methods suffer from the drawback that they cannot distinguish 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 sensors 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 also be utilised as the principal direction.

The main steps for the proposed principal direction detection approach are as follows. First, local linear features

978-1-4673-2126-6/12/$31.00 © 2012 IEEE

are extracted from monocular image grabbed by a Kinect pointing up to the ceiling. Second, the ceiling plane is detected using depth map obtained by a Kinect and local features that does not belong to the ceiling are eliminated to improve accuracy. Then Hough Transform is then applied to extract lines and the principal direction is determined by a voting scheme.

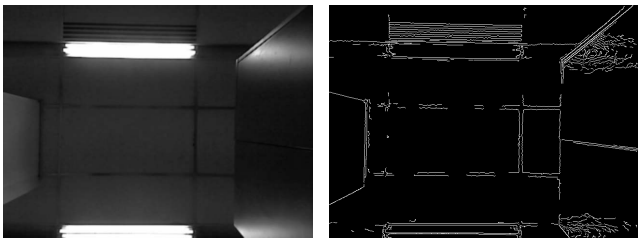
The structure of this paper is as follows. In section II, more details for principal direction detection are demonstrated. After having principal direction, relative transformation between two consecutive frames are determined by using ICP in section III and the current robot position in global coordinate can be retrieved by simply accumulating the relative transformation matrix. In section IV, real world experiments are carried out to show how the detected principal directions can be used to guide the frame-to-frame based algorithm (ICP) to improve generated visual odometry significantly.

II. Principal direction detection

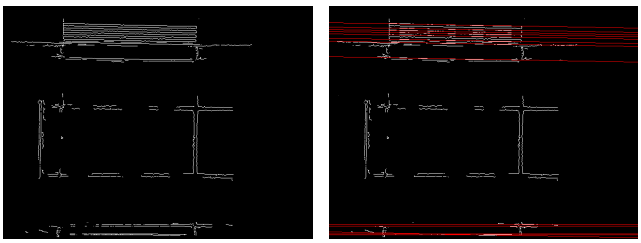
The principal direction detection refers to the detection of Yaw (θ) value which is the rotation around the normal of the plane that the robot is currently on. In order to have better principal direction estimation, we mounted the Kinect on the robot and pointed up to the ceiling. Ceiling vision has an advantage of no or little obstruction, as well as only involving rotation and affine transformation without scale changes compared to the frontal view. Also in ceiling vision, the field of view of the camera is not likely filled with non-stationary objects, such as moving people, which cannot be used as reference to determine principal direction. Calibration and rectification are performed to achieve undistorted frames [10].

Although 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. Compare 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 SIFT [15] or SURF [14]. Canny's algorithm [11] for choosing edgels in an image has emerged as the standard technique, and consistently ranks well in comparisons [12]. 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 (Figure 1a). After that, the Canny algorithm is applied to generate an edge image (Figure 1b).

We mounted a Kinect pointing up towards the ceiling. However, data can 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 ceiling plane. However, the ceiling plane cannot be correctly detected by using monocular images alone. As our data are grabbed from a Kinect, every point has a 2D position (pixel coordinate in the image) and its



(a) Smoothed monochrome image (b) Canny edge detection



(c) Extracted edge pixels from ceiling (d) Lines extracted by Hough Transform

Fig. 1: Lines extraction of ceiling image

corresponding 3D position (the measured position in the world coordinate system). Given all edgels extracted in the previous step, we apply a Ransac-fashioned approach on their corresponding 3D points to determinate the ceiling plane and eliminate all points that does not belong to this plane as shown in Figure 1c. Then, all the remaining 3D points are projected back to 2D image pixels. Finally, the Hough Transform is used to extract lines from these pixels. To reduce the effect of noisy data and improve algorithm stability, only those lines whose Hough votes higher than a threshold are selected as shown in Figure 1d. For Hough Transform, lines are expressed in polar system as:

$$r = x\cos\theta + y\sin\theta$$

where r is the distance from origin to the line and θ indicates the orientation of the line. We set angular resolution of θ as 1 degree. Finally, the most voted angle value is selected as the principal direction.

However, if robot has $roll$ or $pitch$ angle, the perspective distortion will lead to wrong estimation for principal direction as shown in Figure 2a – Figure 2d. This error can be corrected by using the detected plane in previous step.

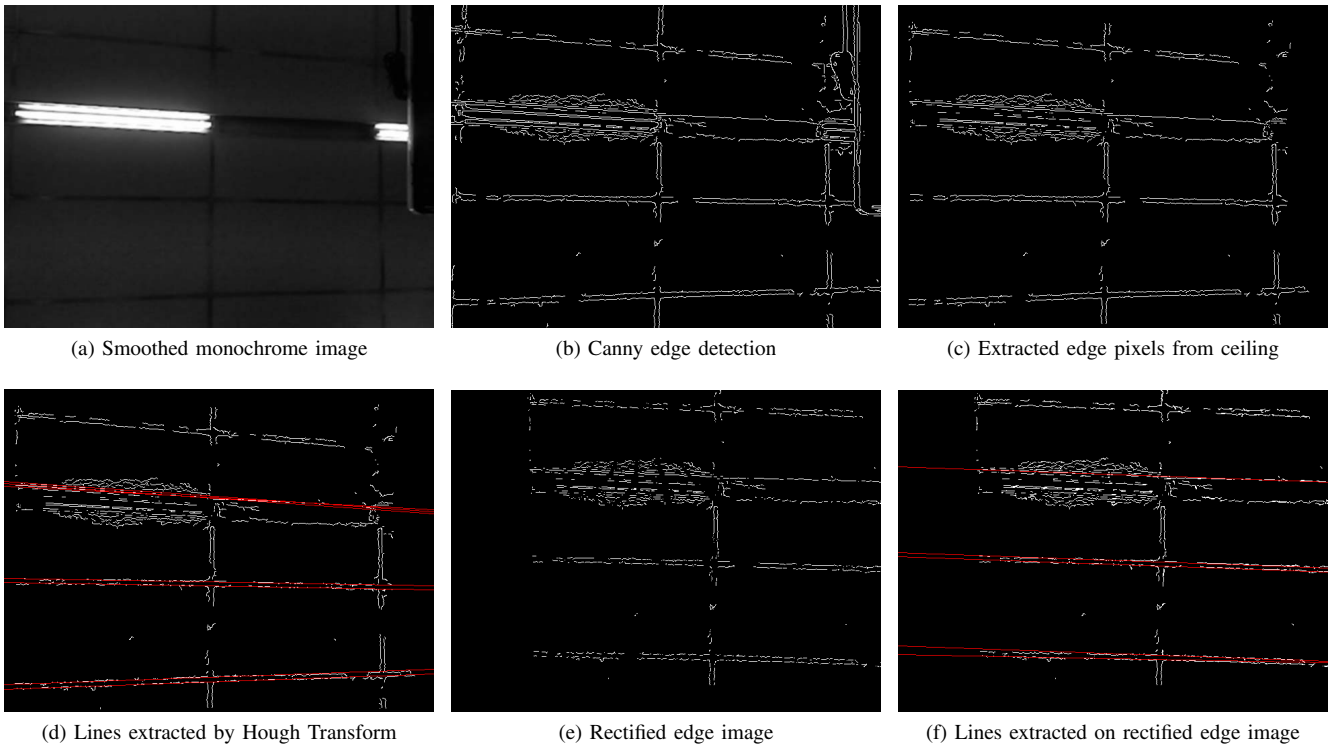
Given detected ceiling plane $ax + by + cz + d = 0$, plane normal can be expressed as:

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

$Roll$ (α) and $Pitch$ (β) of the ceiling plane can be calculated using normal $\vec{n}$:

$$\alpha = \arctan\frac{\vec{n}(1)}{\vec{n}(3)} \qquad \beta = \arctan\frac{\vec{n}(2)}{\vec{n}(3)}$$

where $\vec{n}(i)$ denotes the i^{th} element in vector $\vec{n}$. Given a 2D coordinate in image plane (X), its corresponding pixel (x)



(a) Smoothed monochrome image (b) Canny edge detection (c) Extracted edge pixels from ceiling

(d) Lines extracted by Hough Transform (e) Rectified edge image (f) Lines extracted on rectified edge image

Fig. 2: Lines extraction under perspective distortion

in image is:

$$x = K\,[I|0]\begin{bmatrix} X \\ 1 \end{bmatrix} = KX \tag{1}$$

where K is the calibration matrix obtained in a previous calibration step. Rotation matrix R can be calculated given estimated $roll$ and $pitch$. Because the principal direction is the estimation for the robot's yaw rotation, the rectification process needs to make the ceiling in the monocular image appears to be horizontal to the image plane ($\alpha = \beta = 0$). The principal direction can be determined using the algorithm described before. Thus, we assume there is no translation or yaw changes between the distorted image and rectified image. The rectified pixel x' can be expressed as:

$$x' = K\,[R|0]\begin{bmatrix} X \\ 1 \end{bmatrix} = KRX \tag{2}$$

From equation (1) and (2) we get:

$$x' = KRK^{-1}x \tag{3}$$

where KRK^{-1} is a 3×3 homography matrix. The rectified edge image and its extracted lines can be seen in Figure 2e and 2f respectively.

The detected principal direction $\theta \in [0, 2\pi]$. Thus, one principal direction represents two possible headings of 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 2π. Therefore, the ambiguity can be resolved while determining the relative transformation.

III. Motion estimation using ICP

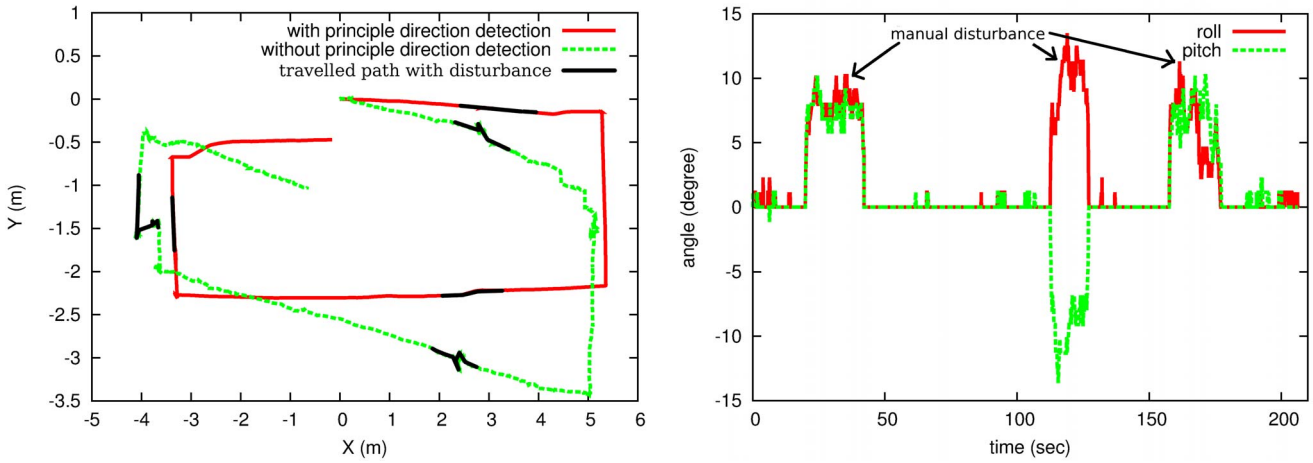
After having $roll$, $pitch$ and yaw angles, we can roughly align two point clouds from consecutive frames. We choose ICP as frame-to-frame based alignment approach for its simplicity and robustness. Given two pre-aligned point clouds, the correct data association is more likely to be found by ICP, hence, reducing the risk that ICP is stuck in a local minimum. Also, the translation parameters can be determined during aliment.

In ICP algorithm, 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. 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 to be sufficient in most cases.

After having the relative transformation parameters between two consecutive frames. The current robot location and orientation in global coordinate can be retrieved by accumulating the relative rotation and translation matrix.

IV. Experiments

In this section, several experiments are conducted to demonstrate the effectiveness and the stability of our principal direction detection approach in visual odometry esti-



(a) robot trajectory.

(b) roll and pitch angle during robot's movement.

Fig. 3: robot trajectory with disturbance

mation problem. The experiments were carried out using an iRobot Create platform. We use the Kinect camera which pointed it up to the ceiling to test our visual odometry system. The robot was manually controlled with a wireless game pad and testing data was 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 these 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 4GB of memory, the average number of frames per second for our visual odometry algorithm is about 10. Edge detection takes about 0.01 sec. It takes 0.009 sec and 0.007 sec for ceiling plane detection and principal direction detection respectively while 0.07 sec is required for ICP.

Figure 3 demonstrates 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. Figure 3b shows the $roll$ and $pitch$ angles at different time. In Figure 3a, the red path is the robot trajectory using principal direction to guide ICP while the green one is without it. Although under obvious disturbance, the path of robot is still smooth and almost closed the loop at the end. The black parts of the paths are places where disturbance occurs. The test emphasizes the effectiveness of principal direction detection.

Although our approach can generate satisfactory results, the small errors still accumulate over time leading to failure of loop closing. There are two main reasons for these failures. First, there are some long gaps in the logs (each lasting a second or more) when the logging disks is unable to sustain the frame rate. Not all of these gaps are successfully bridged by the algorithm, particularly when the image overlap is small. While this is more of a system failure than a visual odometry failure, it is worth noting that the latter reports a failure rather than returning an incorrect motion estimate. 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 leaving the visual odometry with few pixels to work with and 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.

V. Conclusion

It is clear that the introduction of principal direction detection has greatly improved the performance. The proposed method is robust and insensitive to camera disturbance. Experimental results have demonstrated its strength and this method can tolerate substantial disturbances in real run such as uneven floors. The future work is to reconstruct the 3D environment in full three dimensional view.

References

[1] D. Nister and O. Naroditsky and J. Bergen, "Visual Odometry", *in IEEE Computer Society Conference Computer Vision and Pattern Recognition*, 2004, volume 1, pp. 652–659.

[2] D. Nister, "An efcient solution to the ve-point relative pose problem", *in IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2004, volume 26, pp. 756–770.

[3] A. Davison, "Real-time simultaneous localisation and mapping with a single camera", *in Proc. International Conference on Computer Vision (ICCV)*, 2003, pp. 1403–1410.

[4] Motilal Agrawal and Kurt Konolige, "Rough Terrain Visual Odometry", *in Proceedings of the International Conference on Advanced Robotics*, 2007.

[5] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles", *in IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3946–3952.

[6] C. Dornhege and A. Kleiner, "Visual odometry for tracked vehicles", *in Proceedings of the IEEE International Workshop on Safety Security and Rescue Robotics*, 2006.

[7] A. Talukder and S. Goldberg and L. Matthies and A. Ansar, "Real-time detection of moving objects in a dynamic scene from moving robotic vehicles", *in IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, volume 2, pp. 1308–1313.

[8] P. J. Besl and D. McKay, "A method for registration of 3-d shapes", *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[9] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments", *in Proc. of the Intl. Symp. on Experimental Robotics (ISER)*, Delhi, India, 2010.

[10] Nicholas Burrus, "Kinect RGB Demo v0.5.0", url=http://nicolas.burrus.name/index.php/Research/KinectRgbDemoV5.

[11] J. F. Canny, "A computational approach to edge detection", *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 8. pp. 679-678, 1986.

[12] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "Comparison of edge detectors: A methodology and initial study", *in CVPR'96*, pp. 143-148, 1996.

[13] A. Segal D. Haehnel and S. Thrun, "Generalized-ICP", *in Proc. of Robotics: Science and Systems*, 2009.

[14] Herbert Bay and Tinne Tuytelaars and Luc Van Gool, "Surf: Speeded up robust features", *in European Conference on Computer Vision*, 2006, pp. 404–417.

[15] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *in International Journal of Computer Vision*, 2004, volume 20, pp. 91–110.