J Intell Robot Syst (2015) 80:365–374
DOI 10.1007/s10846-014-0167-x

A Single Frame Depth Visual Gyroscope and its Integration for Robot Navigation and Mapping in Structured Indoor Environments

Cheng Chen · Wennan Chai · Hubert Roth

Received: 7 July 2014 / Accepted: 19 November 2014 / Published online: 20 January 2015

Abstract An accurate navigation system is an essential and important part for the mobile robot. The recent appearance of low cost RGBD cameras has made 3D point clouds together with RGB information easy accessible, and they have been widely applied in many applications. Relative poses of a mobile robot can be estimated from consecutive visual information. However, such incremental registration methods still suffer from accumulated errors which makes the estimated trajectory as weird as by only using wheel mounted encoders. In contrast, we introduce a novel and inexpensive sensor fusion based approach to solve the robot localization problem. The key idea is to use depth visual gyroscope as a complementary source for robot heading estimation. Aided with constraints, the unscented Kalman filter is used for robot pose estimation. A field experiment has been carried out in order to verify the introduced method. Accordingly, the 3D map of the environment is also presented based on the estimated robot trajectory.

C. Chen (✉) · H. Roth
Hoelderlinstr. 3, 57068 Siegen, Germany
e-mail: c.chen@zess.uni-siegen.de

H. Roth
e-mail: hubert.roth@uni-siegen.de

W. Chai
Center for Sensor Systems (ZESS), University of Siegen, Paul-Bonatz-Straße 9-11, 57076 Siegen, Germany

Keywords Robot localization · Depth visual gyroscope · Vanishing point · Orthogonal planes extraction · Kalman filter

1 Introduction

The advent of RGB-D sensors such as Microsoft Kinect has resulted in great progress in dense mapping and many researches have been focused on using RGBD images for camera pose estimation. One application of using a RGBD camera is to track its position according to its consecutive measurements. Such methods play an important role in the field of mobile robot localization. Especially for a robot moving in indoor environment, the lack of absolute positioning system such as global positioning system (GPS) results in difficulties in robot pose determination. Various approaches have been introduced to solve such problem such as gyrodometry [1] and stochastic state estimation based approaches [2]. A fruitful discussion of different simultaneous localization and mapping (SLAM) based approaches can be found in [3]. If a robot is equipped with a 3D sensor, the iterative closest point (ICP) [4] is normally used to register subsequential sensor measurements. By using a Kinect sensor, a stand-alone camera position tracking is also possible since it simultaneously provides RGB and depth images with a high frame rate. RGBD odometry is introduced recently for camera tracking [6]. By using ICP algorithm, Kinect fusion (KinFu) [7]

has demonstrated great results for indoor environment reconstruction. However, they both requires dense images as input and the camera should be moved slowly and steady. Thus, given a condition that the depth measurements can only be processed with low sampling rate and large area of depth image have missing values (Fig. 1b), such RGBD sensors show their limitations.

In order to address such a problem, a novel sensor fusion based navigation system is introduced in this paper. The key part of the system is the depth visual gyroscope (DVG). Visual gyroscope (VG) is introduced in [9] and [10], providing camera attitude estimation by using vanishing points (VP). For the indoor environments, corridors are one of the most common features which exists in nearly every building. Such kind of scenarios make Kinect depth measurement incomplete but ensures the vanishing point long time available and consistent. In a vivid metaphor, the vanishing point plays a role as a virtual beacon which is similar to lighthouse guiding a maritime pilot. Since VG directly estimates the camera heading angles from current image, the attitude estimation error does not suffer from previous measurement. However, in the absence of VP detection, visual gyroscope fails. The introduced system integrate VG with IMU to track the robot heading seamlessly. Moreover, further updated with ICP algorithm, the robot trajectory estimated by our system shows promising result. Figure 6 shows the environment 3D mapping result with KURT2 mobile robot and Kinect (Fig. 1a). With total traveling distance of around 61 meters, the point clouds shown in the figure have very good match with the environment, while, they are registered to the global frame by only using the estimated trajectory.

2 Relative Work

Scenes observed in urban and indoor environments consist mainly of straight lines in three orthogonal directions [11]. The intersection parts of those orthogonal structures formed up straight lines in real world. From the camera perspective, parallel lines projected in the captured image seem to intersect and the corresponding intersection points are called vanishing points (VP). The coordinates of the vanishing points in the image depend only on the direction of the parallel objects. Therefore, as long as the objects are parallel to each other, they all intersect at the same point after projective transformation. The coordinates of the vanishing points in the image plane are determined by the relative orientation between the camera and the observed scene. Given V as the location matrix of the vanishing points $[V_x \quad V_y \quad V_z]$, K as the 3×3 camera matrix containing intrinsic parameters and R as the 3×3 rotational matrix, the relation between locations of VP and camera orientation can be defined as Eq. 1:

$$V = KR \tag{1}$$

To be more specifically, V_x V_y and V_z are the 3 location vectors which represent horizontal, vertical and central vanishing point respectively. Camera matrix K consists of camera focal length, principal point and skew. If skew is ignored, K can be presented as Eq. 2:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The intrinsic camera matrix can be calibrated if the vanishing points are known [12]. On the other hand, R

(a) (b)

Fig. 1 Kurt2 mobile robot moving through the corridor and the depth image it sensed

can be estimated if K is known. Using Euler angles, the attitude of the camera regarding to the observed scene can be represented with roll, pitch and yaw. For a camera fix mounted on a robot which works under indoor scenario, it is safe to approximate the camera attitude with pitch and yaw by omitting the roll angle. Where pitch should also be a fixed value depending on the mechanical alignment between camera and the robot and yaw represented camera heading direction. The corresponding rotational matrix R can be therefore written as Eq. 3:

$$R = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ \sin\beta\sin\gamma & \cos\beta & -\sin\beta\cos\gamma \\ -\cos\beta\sin\gamma & \sin\beta & \cos\beta\cos\gamma \end{bmatrix} \tag{3}$$

where β and γ are the pitch and yaw (heading) angle of the camera. By knowing the third column of KR is equal to V_z and substituting Eqs. 2 and 3 into Eq. 1, camera pitch and yaw can be estimated if central vanishing point is known:

$$\gamma = \arcsin\left(\frac{V_{zx} - c_x}{f_x}\right) \quad \beta = \arcsin\left(\frac{V_{zy} - c_y}{-f_y\cos\gamma}\right) \tag{4}$$

Based on aforementioned method, visual gyroscope (VP) algorithm is finalized and its integration with inertial navigation system have shown promising results in attitude estimation [14, 15]. VG relies on using RGB image as input and the foremost task is to localize the vanishing points. One of the methods to locate the vanishing points with a RGB image is using a vote scheme of the detected lines [13]. Edges are firstly needed to extracted [16], Hough Lines algorithm [17] is used afterwards to separate straight lines from the edges. The intersections of those straight lines are then voted and the most voted points are chosen as the vanishing points.

3 Introduced Method

It is common that human made indoor environments are well constructed and organized: corridors and walls are the most common features. The scene observed by a mobile robot is dominated by such orthogonal structures. Given a depth image obtained with a RGBD camera, such as Kinect, the corresponding point cloud is therefore geometrically constrained to such property. Parallel lines in real world are possible to be detected from the measured 3D point cloud and the location of the corresponding VPs can be calculated in its corresponding depth image. Based on this fact, depth visual gyroscope (DVG) is introduced in this paper. Similar to VG, DVG only uses current depth image. Comparing with IMU heading estimation, camera heading estimated with DVG does not suffer from accumulated errors. However, as constrained by the observed scene, DVG can easily fail if no parallel structures exist in the scene. In our introduced method, a IMU is used as an additional source to estimate robot heading continuously. The moved distance can be predicted with wheel mounted encoders, by using the current robot heading angle, its position in global coordinate can be further calculated. However, due to the wheel slippage, the true distance that the robot has moved is hard to be estimated. Therefore, ICP is used here to update robot position accordingly. In the following parts of this section, we will first present DVG and immediately after, the introduced navigation system is presented.

3.1 Depth Visual Gyroscope

Having a Kinect looking forward into the corridor, although much of the depth information is missing due to limited measurement range, it still captures the information of the orthogonal structures from the scene. The orthogonal structures form up parallel lines in real world which explicitly matches the definition of the vanishing point. Since lines are preserved in perspective transform, the coordinates of the vanishing points can be calculated by using the intersections of the projected lines on the image plane. For a wheel mounted mobile robot, the camera is normally fixed, thus the roll rotation of the camera can be safely ignored. Thus, according to previous introduction, once the central vanishing point is known, camera pitch and yaw angles can be estimated.

In order to find central vanishing point, parallel lines within ground plane (plane from the world coordinate) have to be found (Fig. 2). In indoor environments, the floor is generally by default defined as the ground plane which naturally provides the basis for parallel lines detection. The next step is to identify which parallel plane models fit the rest of the scene and orthogonal to the floor. If such planes exist,

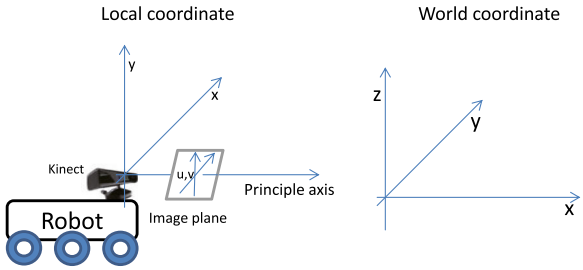

Fig. 2 World coordinate and robot local coordinate

the parallel lines can be found. Consequently, central vanishing point can be calculated.

A line function in 3D space can be defined as the combination of two 3D plane equations. In order to find these plane equations, we apply orthogonal multi plane extraction algorithm from the point cloud using Kinect depth image. Before generating points from depth images, the bilateral filter [18] is applied for smoothing the surface while preserving the geometrical edges. The voxel filter is used afterwards over the points generated from filtered depth image to reduce the total amount of points. From our experiments, the number of points for each frame is reduced to around 2 % from its original size.

Orthogonal multi plane extraction algorithm is based on MSAC [19]. Since ground plane is already known (the points belongs to the ground floor are clustered by setting a threshold with premeasured height of the camera and MSAC is used to estimate its mathematical model), we only need to extract planes orthogonal to the ground from the rest points. Unlike multi plane extraction algorithm based on multi-stage random sample consensus (RANSAC) [20, 22], MSAC based algorithm is used here since it takes Gaussian noise model into account which reduces incorrect rejection of the inliers in a great amount. Since the only the orthogonal plane models are interested, an orthogonal error function is added in addition to the cost function (5) from MSAC (details about C_2 can be found in [19]):

$$C_\alpha = \cos(\Theta_o) \tag{5}$$

where Θ_o is the angle between the plane model defined by randomly selected points and the ground plane. As Θ_o is more close to 90°, C_α is more close to zero. Therefore, planes which are more orthogonal to the ground will have higher score with respect to C_α. Meanwhile, the minimum percentage of inliers is carefully chosen in order to avoid false plane detection. The overview of the algorithm works as follows (Algorithm 1):

Algorithm 1: Orthogonal plane extraction

input : Ground plane function, Point cloud
output: orthogonal plane function
1 denote PC to be the target point cloud, M to be the maximum number of iterations and N to be the minimum number of inliers;
2 denote P to be plane function, n to be the calculated number of inliers and denote P^* $C_2{}^*$ n^* and $C_\alpha{}^*$ to be the parameters of the final estimation;
3 initialization;
4 **for** $i = 1, 2, \ldots, M$ **do**
5 randomly select three points from PC;
6 calculate P C_2 C_α and n;
7 **if** $n \geq N$ **then**
8 **if** $n \geq n^*$ & $C_2 \leq C_2{}^*$ & $C_\alpha \leq C_\alpha{}^*$ **then**
9 $P^* = P$;
10 $C_2{}^* = C_2$;
11 $C_\alpha{}^* = C_\alpha$;
12 $n^* = n$;
13 **end**
14 **end**
15 **end**

In order to extract multi planes, the above algorithm is repeated with multi stages. For each stage of plane extraction, threshold for selecting inliers should be carefully set. According to the Kinect depth image accuracy analysis ([21]), we select the average value of 5 *cm* as the threshold since depth information in the range from 50 *cm* and 500 *cm* is used in our system. The extracted planes are grouped according to their parallelism, and $\Theta_p = \pi/80$ is set as the angular threshold as the maximum angular difference of two plane normals. By the explicit definition of central vanishing point, parallel structures which are more along the principal axis of the camera is interested. Therefore, according to the field of view of the camera, unnecessary parallel planes can be discarded. Together with ground plane function, the parallel lines in real world can be found and the corresponding VP location in the depth image can be calculated.

Given a set of parallel lines in 3D defined by ground plane function and its orthogonal plane function (6):

$$\begin{aligned} a_g \cdot X + b_g \cdot Y + c_g \cdot Z + d_g = 0 \\ a_o \cdot X + b_o \cdot Y + c_o \cdot Z + d_o = 0 \end{aligned} \tag{6}$$

where X, Y, Z is the position of the points belonging to the lines in 3D space and its line function projected of the depth image plane (7):

$$\alpha_l \cdot x_{image} + \beta_l \cdot y_{image} + \eta_l = 0 \tag{7}$$

where x_{image}, y_{image} are the pixel coordinates in the depth image with the camera projection function (8):

$$\begin{aligned} x_{image} &= \frac{X \cdot f_x}{Z} + o_x \\ y_{image} &= \frac{Y \cdot f_y}{Z} + o_y \end{aligned} \tag{8}$$

The parameter of the line function $(\alpha_l, \beta_l, \eta_l)$ can be solved using Eqs. 6, 7 and 8. By dividing both plane functions in Eq. 6 with d_g and d_o respectively, the substraction of the two plane functions can be written in the form of $A \cdot X + B \cdot Y + C \cdot Z = 0$. Accordingly, by substituting Eq. 8 into Eq. 7, the derived equation holds exactly the same form. Since (X, Y, Z) in both functions represent the same points in 3D space, the coefficients $(\alpha_l, \beta_l, \eta_l)$ should match and can be calculated as:

$$\begin{aligned} \alpha_l &= \left(\frac{a_g}{d_g} - \frac{a_o}{d_o}\right)/f_x \\ \beta_l &= \left(\frac{b_g}{d_g} - \frac{b_o}{d_o}\right)/f_y \\ \eta_l &= \frac{c_g}{d_g} - \frac{a_o}{d_o} - \alpha_l \cdot o_x - \beta_l \cdot o_y \end{aligned} \tag{9}$$

Consequently, given a set of lines functions, coordinate of VP in the depth image can be calculated and the camera heading can be estimated by using Eq. 4.

Depth camera matrix is obtained from calibration. Since the working principle of depth measurement by Kinect is based on structured light. The projector is manually blocked and extra infrared lighting source is used so that the pattern of the chessboard is clean and visible in the infrared image. In total, 110 images of the chessboard were collected and the camera matrix is estimated by using the calibration toolbox [23].

3.2 The Navigation System

Briefly, the introduced navigation system is based on using the unscented Kalman filter (UKF) [24]. In the system prediction model, robot heading is estimated with IMU using Eular angles. Based on the moved distance calculated from wheel mounted encoders, robot position is further predicted. In the measurement update model, VG is used to update robot yaw, and ICP is used to update robot position. Since the precise alignment between IMU and robot is not known, we model robot attitude with roll, pitch and yaw predicted by IMU. Additionlly, based on the scenarios that vehicles travel on the ground, the angular constraints are taken into measurement model: among the three angular velocities, only the change in the heading angle is not zero. The state vector of the system model is therefore defined as (10):

$$x_k = \left[p_{x,k}, p_{y,k}, \alpha_k, \beta_k, \gamma_k, \omega_{\alpha,k}, \omega_{\beta,k}, \omega_{\gamma,k}\right]^\mathsf{T} \tag{10}$$

and the corresponding system prediction model is shown in Eq. 11:

$$\begin{aligned} p_{x,k+1} &= p_{x,k} + d_k \cdot \cos\gamma_k \\ p_{y,k+1} &= p_{y,k} + d_k \cdot \sin\gamma_k \\ \psi_{k+1} &= \psi_k + E_{b2n,k} \cdot (\widetilde{\Omega}_k - \Omega_k) \cdot \Delta t \\ \Omega_{k+1} &= \Omega_k \end{aligned} \tag{11}$$

$$E_{b2n,k} = \begin{bmatrix} 1 & \sin\alpha\tan\beta & \cos\alpha\tan\beta \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha/\cos\beta & \cos\alpha/\cos\beta \end{bmatrix} \tag{12}$$

where $\left[\, p_x, p_y\right]^\mathsf{T}$ is robot position vector, ψ_k is robot attitude vector presented with Euler angles (roll, pitch and yaw) which equals to $[\,\alpha_k, \beta_k, \gamma_k]^\mathsf{T}$, $\widetilde{\Omega}_k$ is the IMU measurement input and Ω_k is the gyro bias vector which equals to $\left[\,\omega_{\alpha,k}, \omega_{\beta,k}, \omega_{\gamma,k}\right]^\mathsf{T}$. $E_{b2n,k}$ is the rotation rate transformation matrix from body to global frame.

In the measurement model, DVG heading estimation, position refinement using ICP and angular constraints are modeled in the system observation vector: $y_k = \left[\,\widehat{p}_{x,icp,k}, \widehat{p}_{y,icp,k}, \widehat{\gamma}_{dvg,k}, \widehat{\alpha}_k, \widehat{\beta}_k\right]^\mathsf{T}$. The corresponding measurement update equation is written as:

$$\text{DVG ICP update} \begin{cases} \widehat{p}_{x,icp,k} = p_{x,k} \\ \widehat{p}_{y,icp,k} = p_{y,k} \\ \widehat{\gamma}_{dvg,k} = \gamma_k \end{cases}$$

$$\text{Angular constraints} \begin{cases} \widehat{\alpha}_k = \alpha_k \\ \widehat{\beta}_k = \beta_k \end{cases}$$

In total, the complete system can be written as:

$$\begin{aligned} x_k &= A_{k-1}x_{k-1} + w_k \\ y_k &= Hx_k + v_k \end{aligned} \tag{13}$$

where w and v are process and measurement noise respectively.

Due to high order of nonlinearity of the system process model, UKF is used in this paper. For nonlinear system models, UKF provides a balance between computational effort and estimation accuracy. Unlike extended kalman filter (EKF), UKF use signam points to yield the transformed state vectors along with the state covariance. It preserves the states' probabilistic model even without linearize the propagation func-

tion. A detailed description of UKF is shown in Algorithm 2.

Algorithm 2: Unscented Kalman Filter

denote X to be system state vector with its size n and covariance matrix P ;
denote Q and R to be process and measurement noise covariance matrix ;
initialization:
$\widehat{x}_0^+ = E(x_0)$;
$P_0^+ = E\left[(x_0 - \widehat{x}_0^+)(x_0 - \widehat{x}_0^+)^T\right]$;
for $k = 1, 2, \ldots, \infty$ **do**
  Time update:
  Draw state sigma points: $\widehat{x}_{k-1}^{(i)} = \widehat{x}_{k-1}^+ + \widetilde{x}^{(i)} \quad i = 1, 2, \ldots, 2n$
  $\widetilde{x}^{(i)} = \left(\sqrt{nP_{k-1}^+}\right)_i^T \quad i = 1, 2, \ldots, n$
  $\widetilde{x}^{(n+i)} = -\left(\sqrt{nP_{k-1}^+}\right)_i^T \quad i = 1, 2, \ldots, n$
  update with system propagation model:
  $\widehat{x}_k^{(i)} = f(\widehat{x}_{k-1}^{(i)}) \quad \widehat{x}_k^- = \frac{1}{2n}\sum_{i=1}^{2n} \widehat{x}_k^{(i)}$
  $P_k^- = \frac{1}{2n}\sum_{i=1}^{2n}(\widehat{x}_k^{(i)} - \widehat{x}_k^-)(\widehat{x}_k^{(i)} - \widehat{x}_k^-)^T + Q_{k-1}$
  Measurement Update:
  Draw measurement sigma points:
  $\widehat{x}_k^{(i)} = \widehat{\mathbf{x}}_k^- + \widetilde{\mathbf{x}}^{(i)} \quad i = 1, 2, \ldots, 2n \quad \widetilde{x}^{(i)} = \left(\sqrt{n\mathbf{P}_k^-}\right)_i^T \quad i = 1, 2, \ldots, n$
  $\widetilde{x}^{(n+i)} = -\left(\sqrt{n\mathbf{P}_k^-}\right)_i^T \quad i = 1, 2, \ldots, n$
  measurement prediction:
  $\widehat{y}_k^{(i)} = h(\widehat{x}_k^{(i)}) \quad \widehat{y}_k = \frac{1}{2n}\sum_{i=1}^{2n} \widehat{y}_k^{(i)}$
  covariance update:
  $P_k^{yy} = \frac{1}{2n}\sum_{i=1}^{2n}(\widehat{y}_k^{(i)} - \widehat{y}_k)(\widehat{y}_k^{(i)} - \widehat{y}_k)^T + R_k$
  $P_k^{xy} = \frac{1}{2n}\sum_{i=1}^{2n}(\widehat{x}_k^{(i)} - \widehat{x}_k^-)(\widehat{y}_k^{(i)} - \widehat{y}_k)^T$
  state update:
  $K_k = P_k^{xy}(P_k^{yy})^{-1} \quad \widehat{x}_k^+ = \widehat{x}_k^- + K_k(y_k - \widehat{y}_k) \quad P_k^+ = P_k^- - K_k P_k^{yy} K_k^T$
end

Additionally, the robot motion is constrained on a 2D plane, therefore, the roll and pitch angle of the robot should equality be zero (attitude constraints). Comparing the IMU propagation model without using constrains, the attitude of IMU is constrained with one possible rotational movement. In other words, the unknown bias of IMU pitch and roll measurement can be compensated by using the constrains. Constraints in UKF can be handled with various approaches [25, 26]. As the angular constraints directly cope with system states, they are treated as linear equality constraints. Written as a part of the system observation function, their covariance should be set as zero accordingly. Another issue is how to use ICP algorithm output as position update in the Kalman filter update. Since the quality of ICP algorithm strongly depends on the initial alignment, bad initial alignment will result in local minimum and thus the alignment result diverges. A simple mechanism to detect false registration of ICP is used (Algorithm 3):

Algorithm 3: ICP update Rejection

input : relative moving distance estimated from encoders and IMU, alignment estimated by ICP using depth image
output: ICP validation result
denote $\delta P_x \sim N(\mu_{p,x}, \sigma_{p,x})$ and $\delta P_{p,y} \sim N(\mu_{p,y}, \sigma_{p,y})$ to be estimated robot positions with gaussian noise model; denote δICP_x and δICP_y to be the translation for aligning two frames estimated with ICP;
if $|\delta ICP_x| > 2\sigma_{p,x}$ *or* $|\delta ICP_y| > 2\sigma_{p,y}$ **then**
  Reject ICP update;
end

The exact values in $\delta P_x \sim N(\mu_{p,x}, \sigma_{p,x})$ and $\delta P_{p,y} \sim N(\mu_{p,y}, \sigma_{p,y})$ can be obtained using UKF time update, in other words, running the UKF without measurement update. The only difference is that the covariance of the states are initialized with zeros for each new ICP registration process. The overall system has multi observation models to handle the issue of availabilities with respect to DVG and ICP output. In a way that states are updated according to the corresponding observations. Furthermore, if the robot is moving under the same scenario such as the same straight corridor, the detected vanishing point is consistent and its corresponding DVG output can therefore be used for continuous robot heading update. However, once the robot has made a sharp turn or moved into a new scenario, the reference direction for DVG should be changed accordingly. For example, the robot may running to the end of one corridor and turning into another one. So far, the method to handle such situation is not implemented. Since our experiment scenario is mainly consisted of orthogonal corridors, the reference direction for DVG will be set with the offset ($\pi/2$ or $-\pi/2$ etc.) if a sharp turn of the robot is detected by IMU.

4 Experiment Results

The introduced method is verified with a field experiment (Fig. 3). The place for carrying out the experiment is inside a university building. The robot used for the experiment is KURT2 mobile robot with wheel mounted encoders. Sensors employed are wheel mounted encoders, Microsoft Kinect and a MEMS IMU (Xsens MTi). The sampling rates for IMU and encoder are 100Hz while Kinect depth image is sampled with 5Hz. The total travelling distance of the robot is about 61 meters with 937 pairs of RGB and depth images collected. As shown in Fig. 3, the robot is mainly travelling in the corridors. In order to verify the estimated trajectory of the robot, the turning point of the robot is marked manually and collected as the ground truth.

4.1 Results of Depth Visual Gyroscope

Figure 4a shows both depth and RGB measurement of the Kinect camera mounted on KURT2 mobile robot (Fig. 1a) looking at the travelling direction under the experiment scenario. The depth image has large area of missing values since the object in the front is

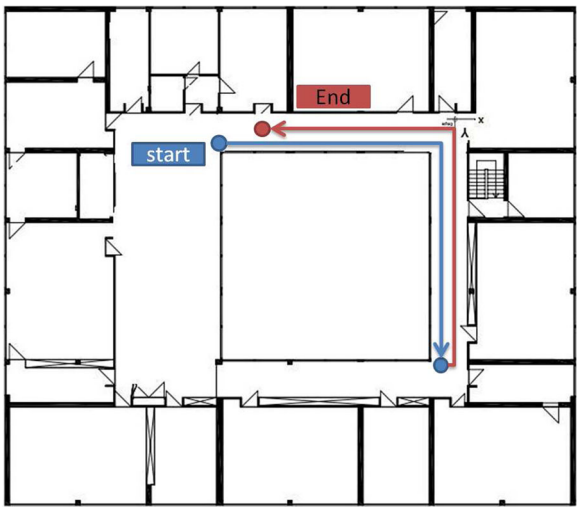

Fig. 3 Experiment scenario

beyond the camera's measurement range. Figure 4c shows the orthogonal multi plane segmentation results based on the introduced method. The red cross in the depth image (Fig. 4b) shows the estimated coordinate of the vanishing point.

Figure 5 shows the robot heading estimation with DVG and IMU. The discontinuous red line is the output of DVG, the missing part means that DVG does not have a valid estimation of current robot heading. It is clear that the most missing part of red lines are located shortly before and after robot turns, where the robot is running to the end of the corridor. For the heading estimation before 30 meters, the error between DVG and IMU is smaller than the measured information afterwards because heading estimation with IMU has drifted away in long time running situation.

4.2 Results of the Navigation System and Map Building

Robot trajectory estimation results are shown in Fig. 6d. The trajectory estimation using wheel mounted encoders is as bad as expected. Additionally, the trajectory estimation with similar system model

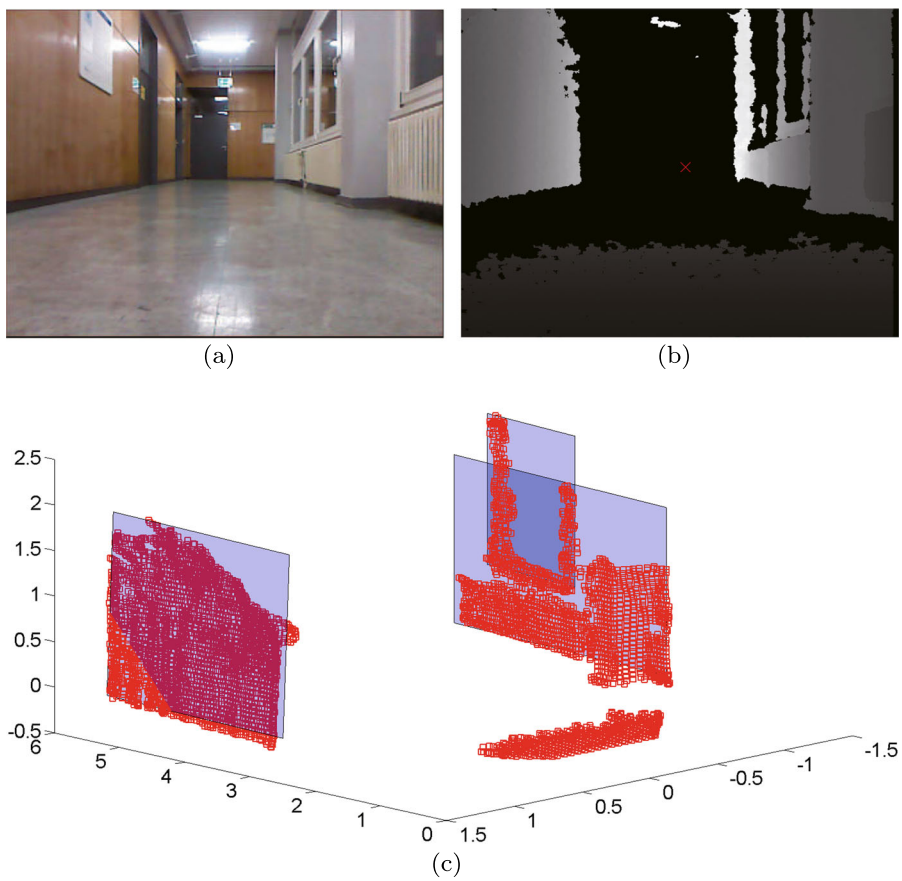

Fig. 4 Vanishing point detection using depth image and multi plane segmentation

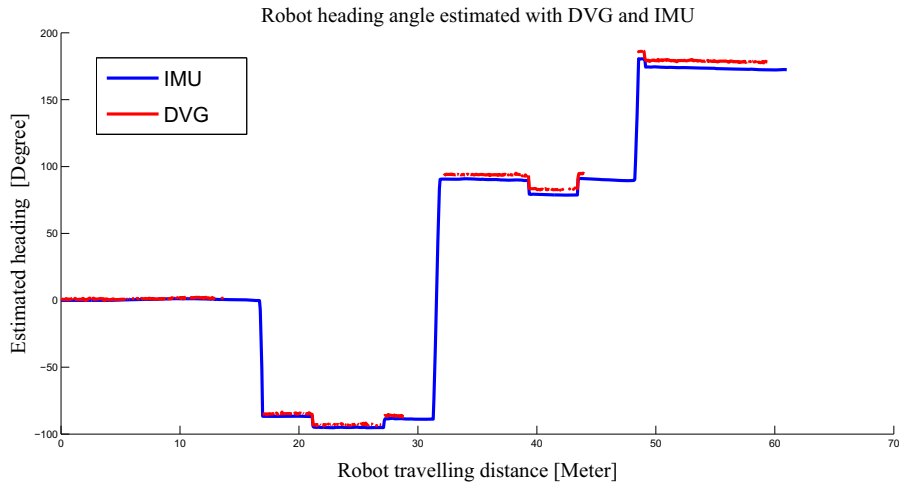

Fig. 5 Robot heading estimation with DVG and IMU

which dose not use DVG heading estimation is plotted for comparision purpose. The comparison between them is obvious: our introduced system outperforms the others and even after 61 meters running, the positioning error at the end point is only around 20 cm, which is even only around half of the length of the robot (40 cm). Using the estimated robot trajectory directly, the mapped 3D point cloud of the environment is presented (Figs. 6a–6c). We did not use any post processing to merge different sensor scans or eliminate false measurement, the presented map is clean and consistent to the real world model.

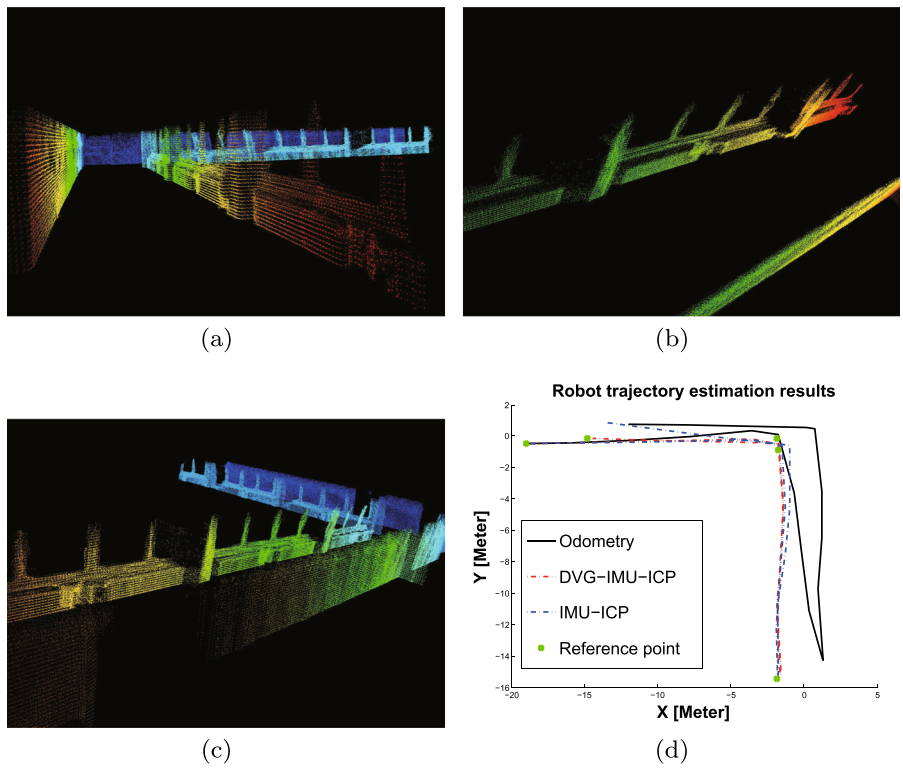

Fig. 6 Environment3D using estimated robot trajectory

5 Conclusion and discussion

A single frame depth visual gyroscope algorithm is presented in this paper with its integration to the robot navigation and mapping system. Using depth image, the detected vanishing point is restricted with the observed geometrical information of the observed scene. Both trajectory and mapping results are presented, it is a promising method for indoor mobile robot localization and mapping especially for corridor scenarios. As long as the environment fills the assumptions and is consistent, vanishing points detected in subsequential measurement is also consistent. In other words, it can be treated as virtual heading landmark which can be used for robot heading update. Therefore, formalize the presented method shows high potential for further implementation as a SLAM approach. The advantage is obvious: there is no need to dynamically add landmarks into the system states which results in extra computational effort. However, our experiment scenario is a unique case, there is further work to be carried out for implementing this method for scenario with complex structures. On the other hand, it solves the mapping problem of the corridors effectively without considering landmark detection and association which is the key issue of EKF SLAM. Another perspective is to study map representation and scene understanding. For example, the map can be modeled with the reference angle with respect to the virtual landmarks as long as the scene is detected to be corridor like scenario. Comparing with heading estimation using RGB images, depth image does not suffer from the light conditions of environment and the color patterns of the observed scene. Another part of our future work is to use this method for large scale environment reconstruction and test it for outdoor applications.

References

1. Borenstein, J., Arbor, A.: Gyrodometry: a new method for combining data from gyros and odometry in mobile robots, IEEE international Conference on Robotics and Automation (1996)
2. Chung, H., Ojeda, L., Borenstein, J.: Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope. IEEE Trans. Robot. Autom. **17**(1), 80–84 (2001)
3. Quellette, R., Hirasawa, K.: A comparison of SLAM implementation for indoor mobile robots, IEEE/RSJ International Conference on Intelligen Robots and Systems (IROS) (2007)
4. Bsel, P., McKay, N.: A Method for registration of 3-D shapes. IEEE Trans. Pattern Analysis and Machine Intelligence **14**, 239–256 (1992)
5. Nchter, A., Surmann, H., Lingemann, K., Hertzberg, J., Thrun, S.: 6D SLAM with an Application in autonomous mine mapping. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1998–2003, New Orleans (2004)
6. kerl, C., Sturm, J., Cremers, D.: Robust Odometry Estimation for RGB-D Cameras. In: Proceedings Of the IEEE Int. Conf. on Robotics and Automation (ICRA) (2013)
7. Newcombe, R.A., Davison, A.J., Izadi, S., Kohli, O., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking, 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (2011)
8. Taguchi, Y., Jian, Y.-D., Ramalingam, S., Feng, C.: SLAM using both points and planes for hand-held 3D sensors, Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on (2012)
9. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision, 3rd edn, Stamford, Connecticut: Cengage Learning (2008)
10. Klein, G., Drummond, T.: A Single-frame visual gyroscope. In: Proceedings of British Machine Vision Conference (BMVC) (2005)
11. Coughlan, J.M., Yuille, A.L.: Manhattan World: Compass Direction from a Single Image by Bayesian Inference, In Internation Conference on Computer Vision (1999)
12. Caprile, B., Torre, V.: Using Vanishing Points for camera Calibration. The International Journal of Computer Vision **4**(2), 127–140 (1990)
13. Ruotsalainen, L., Bancroft, J., Lachapelle, G., Kuusniemi, H., Chen, R.: Effect of camera characteristics on the accuracy of a visual gyroscope for indoor pedestrian navigation, Ubiquitous Position, Indoor Navigation, and Location Based Service (UPINLBS), 2012, Helsinki, Finland
14. Ruotsalainen, L., Kuusniemi, H., Chen, R.: Visual-aided two dimensional pedestrian indoor navigation with a smartphone. J. Global Positioning Syst. **10**(1) (2011)
15. Hong, D., Lee, H., Cho, H., Park, Y., Kim, J.H.: Visual Gyroscope: Integration of Visual Information with Gyroscope for attitude Measurement of Mobile Platform, International Conference on Control, Automation and Systems, 2008, Seoul, Korea
16. Shrivakshan, G., Chandrasekar, C.: A comparision of various edge detection techniques used in image processing. IJCSI International Journal of Computer Science Issues **9**(1), 269–276 (2012)
17. Hough, P.V.C.: Method and means for recognizing complex patterns, U.S. Patent 3069654 (1962)
18. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images, Sixth International Conference on Computer Vision (1998)

19. Torr, P.H.S., Zisserman, A.: MLESAC:A new robust estimator with application to estimating image geometry. Comp. Vision Image Underst. **78**, 138–156 (2000)
20. Taguchi, Y., Jian, Y.-D., Ramalingam, S., Feng, C.: Point-plane SLAM for hand-held 3D sensors, IEEE International Conference on Robotics and Automation (ICRA), 2013, Karlsruhe, Germany
21. Khoshelham, K., Elberink, S.O.: Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. Sensors **12**(2), 1437–1454 (2012)
22. Nasir, A.K., Hille, C., Roth, H., Extraction, P.lane.: Plane Extraction and Map Building Using a Kinect Equipped Mobile Robot, Workshop on Robot Motion Planning: Online, Reactive and in Real-time, IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal (2012)
23. Lindner, M., Schiller, I., Kolb, A., Koch, R.: Time-of-Flight Sensor Calibration for Accurate Range Sensing. J. Comput. Vis. Image Underst. (CVIU) **114**(12), 1318–1328 (2010)
24. Simon, D.: Optime state estimation: Kalman, H, and Nonlinear Approaches. Wiley, Hoboken (2006)
25. Simon, D.: Kalman filtering with state constraints: a survey of linear and nonlinear algorithms, IET Control Theory and Applications (2009)
26. Teixeira, B.O.S., Chandrasekar, J., Toores, L.A.B., Dennis, L.A.A.: S.Bernstein, State estimation for linear and nonlinear equality-constrained systems, International Journal of Control (2009)