2014 5th IEEE RAS & EMBS International Conference on
Biomedical Robotics and Biomechatronics (BioRob)
August 12-15, 2014. São Paulo, Brazil

Design, Implementation and Evaluation of a Motion Control Scheme for Mobile Platforms with High Uncertainties *

Mustafa Mashali[1],Redwan Alqasemi[2],Sudeep Sarkar[3],Rajiv Dubey[4]

Abstract— **In this work, we present a motion control scheme for a robotic mobile platform using low-cost vision sensor to update encoder values. We track the pose of a power wheelchair using wheel encoders along with a Microsoft Kinect camera. Two methods of pose estimation are implemented and tested. These methods are a) encoder-based odometry and b)ICP(Iterative Closest Point)-based updated odometry. We evaluate the performance of each method using precise wheelchair pose ground truth data acquired via a state-of-the-art VICON® system with eight motion capture cameras. Offline data processing is performed to refine the ICP parameters and estimate the covariance matrices of the Kalman filter. The offline data processing results demonstrate that our ICP-based updated odometry has very accurate pose tracking. By implementing our control scheme, the position error is improved by a factor of 15 and the localization orientation error is improved by a factor of 13. In online implementation, there was 4 times improvement for both position and orientation angle estimation. To demonstrate the robustness of our approach, we apply it for online obstacle avoidance. A wheelchair-mounted robotic arm (WMRA) is also included in this platform and will be used for future work on combined mobility and manipulation control with sensor assistance.**

I. INTRODUCTION

Localization, i.e. estimating the position and orientation (pose) of a mobile robot from sensory data, is an active problem in autonomous mobile robots. A mobile robot has to accurately localize itself relative to its surrounding environment at all times in order to navigate safely and efficiently. Without an accurate localization, autonomous navigation and obstacle avoidance will be impossible [1], [2]. In literature, there are varieties of sensors, techniques and models that have been employed to handle this problem.

The dead reckoning method, commonly referred to as odometry, is the common practice for localizing mobile robots. In this method, the current robot pose is computed incrementally by knowing the previous pose and a measure of the movement that is carried out by the robot. It is well known that odometry is subject to many sources of measurement errors which make it impossible to maintain an accurate estimate of robot pose over long paths. Therefore, measuring and correcting systems inaccuracies and sensors errors is crucial for increasing the precision of the localization procedure.

Maintaining a good localization precision for a moving platform with the presence of high inaccuracies in the kinematic model and the motion control is a challenge, especially if this mobile platform is used for assisting individuals with disabilities. In this paper, we present a comprehensive study of modeling, measuring and correcting localization errors by using two inexpensive sensors. These sensors are wheel encoders and Microsoft Kinect camera. We also design and implement a corrected encoder-based motion control scheme and present its effectiveness in compensation of encoder localization errors. They are used to control the wheelchair mounted Robotic arm (WMRA) system [3]. The WMRA system is an assistive device mainly used for helping individuals with limited upper mobility to perform activities of daily living (ADL). This device is a 7 degree of freedom (DoF) robotic arm attached to a power wheelchair. Fig. 1 shows the first prototype WMRA I. Throughout this paper, mobile platform and wheelchair is used interchangeably.

II. BACKGROUND

As mentioned earlier, localization is a key problem in mobile robot navigation. This problem has gained a lot of interest in recent years. Generally the localization problem is solved by relative or absolute techniques [4]. The absolute positioning methods use features from the environment such as navigation beacons, landmarks, and GPS to determine the mobile platform location. The relative positioning methods use measurements from sensors that do not use any environment cues such as wheel encoders, accelerometers, and gyroscopes [5]. Relative positioning is simple, inexpensive and easy to achieve in realtime. However, it suffers from

Fig. 1: Wheelchair Mounted Robotic Arm (WMRA I).

*Research supported by NSF, award no. 0713560.

[1]Mustafa Mashali is a doctoral candidate at University of South Florida, Tampa, Florida 33620 USA. mmashali@mail.usf.edu

[2]Redwan Alqasemi is the research lead at Center for Assistive, Rehabilitation and Robotics Technologies at University of South Florida Alqasemi@usf.edu

[3]Sudeep Sarkar is a Professor in Department of Computer Science and Engineering at University of South Florida. Sarkar@usf.edu

[4]Rajiv Dubey is a Professor and Chair of Department of Mechanical Engineering at University of South Florida. Dubey@usf.edu

accumulating errors without bound over time and/or distance. These errors are due to navigation on irregular ground or smooth floor which causes the wheels to slip or slide. Localization using encoders readings can cause 20%-25% error in pose estimation [6]. However, in case of absolute position estimation, the error accumulation rate will be eliminated when the measurements are available due to the fact that the pose is externally determined. As a result, the error is not accumulated while the robot travels [6]. One example of the absolute position estimation is visual odometry.

Visual odometry, sometimes in literature referred to as ego-motion estimation, is a method in which the pose of a mobile robot is determined by using image information. In this method, computer vision algorithms [7], [8], [9] are used to estimate a 6 DoF pose of a moving camera frame by analyzing a sequence of video frames. It is primarily tracking visual features from one video frame to another and instantaneously determining the camera pose. By projecting the camera pose to the robot's coordinate frame, the pose of the robot based in a global coordinate frame can be estimated. One of these vision algorithms is ICP (Iterative Closest Point), which was introduced by Besl et al. [10], [11] in the 1990s. The ICP algorithm is a well-known algorithm for point set registration [12][13].To improve the ICP outcomes, Hervier et al, in [14] proposed to fuse the ICP with measurements from other motion sensors by using the Kalman filter [15].

One of the most widely used approaches for sensor fusion is the Kalman filter. In [16], Chen presents a review of contributions of Kalman filtering in solving mobile robot problems such as localization, mapping and navigation. In literature, many authors have fused motion sensor measurements with vision sensor data for mobile robot localization using the Kalman filter [14], [17], [18], [19]. In these works, often different types of motion sensors (i.e. encoders, accelerometers, and gyroscopes) are combined with vision sensors.

Recently, the assistive navigation systems for individuals with disabilities became a new area of research in mobile robotics [20]. Many studies have been undertaken to design smart wheelchairs with different levels of autonomy to assist people with disabilities in performing their ADL.

Power wheelchairs are designed for manual operation which depends on human control and perception [21]. These wheelchairs lack precise motion hardware, such as built-in encoders, and precise motion controllers. As stated by Simpson et al, in[22], most smart wheelchairs that have been developed from power wheelchairs need major modifications to operate properly. These modifications involve adding sensors and by-passing the wheelchair's controller to directly control the wheelchair's motors. As a result, a wheelchair without any major modification is determined to be a mobile platform that is highly inaccurate for precise motion or autonomous operation.

Although there are some similarities between our approach and the aforementioned works, there are major differences. We do not manipulate the environment by using artificial landmarks for localization [17], [23], and we do not modify the mobile platform (power wheelchair) to be more suitable for autonomous motion. In addition, we not only detect the localization error, but also design a control scheme to correct it. Furthermore, we test and evaluate our design of motion control scheme in a real-world mobile platform application, which demonstrates the robustness of our approach. Moreover, our approach evaluations are compared against a highly accurate measurement of wheelchair motion ground truth using a state-of-the-art VICON® system with eight motion capturing cameras.

The remainder of this paper is organized as follows: The localization methods used in this paper are described in section III. In section IV, we introduce the motion control scheme that utilizes the localization methods for compensating odometry errors. Experimental procedures and discussion are introduced in section V. Finally, the conclusions are presented in section VI.

III. WHEELCHAIR POSE ESTIMATION METHODS

Two methods are used to estimate the wheelchair location and heading: encoder-based odometry and ICP-based odometry. These methods are implemented using two inexpensive sensors which are wheel encoders and vision sensor. The wheel encoders are ball bearing optical shaft encoders H5 from US DIGITAL [24]. The vision sensor is Microsoft XBOX 360 Kinect camera [25]. This camera is able to capture 30 frames per second with a resolution of 640X480 pixels. For each pixel, the Kinect measures the associated depth information by projecting a pattern of infrared lights and use stereo triangulation to calculate the depth. Fig.2 shows a flowchart that illustrates the steps of the two pose estimation methods.

In the encoder-based odometry, measurements of optical encoders mounted on each driving wheel were used for calculating the transformation matrix between two consecutive wheelchair frames ${}^{w_{i-1}}_{w_i}T$ (refer to Fig.3). We refer to this transformation matrix as the local transformation matrix. The same transformation matrix was determined by applying the ICP algorithm on two consecutive overlapped Kinect point clouds. The Extended Kalman Filter (EKF) was used to fuse

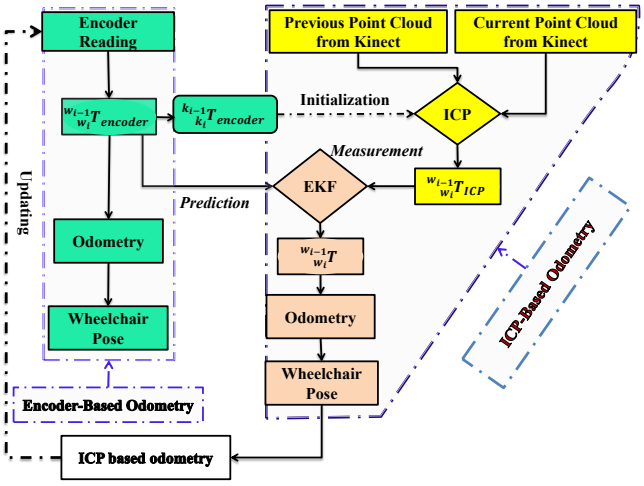


Fig. 2: Flowchart of two estimation methods.

the two local transformation matrices to get the optimized local transformation matrix which was used to calculate the ICP-based odometry. Then the encoder-based odometry was updated with ICP-based odometry. The following is a detailed explanation of these two methods:

A. Encoder-based odometry

Encoders are used to track the wheelchair global pose $[X_i, Y_i, \gamma_i]^T$ by measuring the angular displacement of the right wheel θ_{r_i} and the left wheel θ_{l_i}, where X_i and Y_i are the X and Y global coordinates of the wheelchair respectively, and γ_i is the wheelchair orientation angle. Throughout this paper, the subscript i means the i^{th} instance in the wheelchair motion. These two angular displacements are computed using the encoders readings from both wheels. The distance traveled by the left and right wheels are $L_i = wr\theta_{r_i}$ and $R_i = wr\theta_{l_i}$ respectively, where wr is the wheel radius in meters. The pose of the wheelchair $[X_i, Y_i, \gamma_i]^T$ relative to a global frame as can be computed by using (1)

$$\begin{bmatrix} X_i \\ Y_i \\ \gamma_i \end{bmatrix} = \begin{bmatrix} X_{i-1} \\ Y_{i-1} \\ \gamma_{i-1} \end{bmatrix} + \begin{bmatrix} r_i\left[sin\gamma_{i-1} - sin\left(\gamma_{i-1} + \frac{R_i - L_i}{wb}\right)\right] \\ r_i\left[cos\left(\gamma_{i-1} + \frac{R_i - L_i}{wb}\right) - cos\gamma_{i-1}\right] \\ \frac{R_i - L_i}{wb} \end{bmatrix} \tag{1}$$

the symbol r_i represents the instantaneous radius of curvature, where $r_i = \frac{wb}{2}\left(\frac{L_i + R_i}{L_i - R_i}\right)$ and wb is the wheel base. This model is similar to the model represented in [26]. For further details on the wheelchair kinematic model, refer to [3].

B. ICP-based odometry

As it is illustrated in Fig.2, the ICP-based odometry was obtained in two steps:

1) ICP-based local transformation matrix: The alignment of two point clouds, also referred to as registration, means finding the transformation matrix $\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$ (rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$) that will transform one data set to the other. Thus, given two data sets, one is a target data set denoted as $M \triangleq \{\overrightarrow{m_i}\}_{i=1}^{N_m}$ with N_m points, and the other is a source data set denoted as $P \triangleq \{\overrightarrow{p_i}\}_{i=1}^{N_p}$ with N_p points. The goal is to find the transformation parameters between the two point sets in which the error between the transformed source data and the closest points in the target data will be minimum.

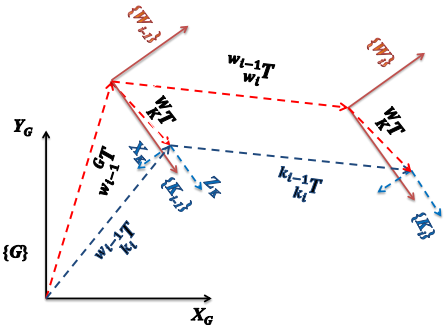


Fig. 3: Wheelchair and Kinect coordinates frame reative to the Global coordinate frame.

Knowing the transformation matrix between each two consecutive frames (see Fig 3) by applying a registration process, the pose of a mobile platform can be tracked. Using the ICP algorithm, the registration process is usually composed of two stages: coarse and fine alignments. The coarse alignment is implemented for roughly aligning the two frames by using, for example, feature matching or encoder measurements. This makes the ICP algorithm faster for the fine alignment and avoids local minima.

In this work, registration using the ICP algorithm was applied to the two consecutive point clouds captured by the Kinect. The following is to illustrate how the local transformation matrix was determined based on the ICP algorithm. Refer to Fig. 3 for the equations' variables.

First, point clouds associated with Kinect coordinate frames $K_{i-1}$and K_i were captured. They were initially aligned using the local transformation matrix of the current Kinect frames relative to the previous Kinect frames ${}^{k_{i-1}}_{k_i}T_{encoder}$ which can be determined using (2). ${}^{w_{i-1}}_{w_i}T_{encoder}$ can be calculated form (1) by assuming $X_{i-1} = Y_{i-1} = \gamma_{i-1} = 0$.

$${}^{k_{i-1}}_{k_i}T_{encoder} = {}^{w}_{k}T^{-1} \ {}^{w_{i-1}}_{w_i}T_{encoder} \ {}^{w}_{k}T \tag{2}$$

where ${}^{w}_{k}T$ is the transformation matrix of the Kinect frame relative to the wheelchair frame. The perfection of the initial alignment depends on how accurate the wheel encoders are. The initial misalignment was obviously noticed during rotation motion more than the translation. This indicates that the encoder-based odometry is more accurate during translation compared with rotation. Second, the ICP algorithm was applied on the encoder-aligned Kinect point clouds to get a 6 DoF fine-alignment transformation matrix, T_{ICP}. This transformation matrix is a compensation for any error in the local transformation matrix determined from encoders measurements. The overall transformation matrix between the previous and current Kinect point cloud is:

$${}^{k_{i-1}}_{k_i}T_{ICP} = T_{ICP} \ {}^{k_{i-1}}_{k_i}T_{encoder} \tag{3}$$

The local transformation matrix between two consecutive wheelchair frames W_{i-1} and W_i can then be determined as follows (refer to figure 3):

$${}^{w_{i-1}}_{w_i}T_{ICP} = {}^{w}_{k}T \ {}^{k_{i-1}}_{k_i}T_{ICP} \ {}^{w}_{k}T^{-1} \tag{4}$$

The accuracy of the local transformation depends on how accurate the registration process is. ICP algorithm sometimes fails in perfectly aligned Kinect point cloud, which leads to inaccurate estimation of the local transformation matrix. To smooth the pose estimation of the wheelchair, the Kalman filter is used to fuse the measurements from wheel encoders and ICP algorithm.

2) ICP-based odometry: As shows in Fig.2, the Extended Kalman Filter (EKF) is applied to fuse the two measurements of the local transformation matrices determined from encoder-based odometry, ${}^{w_{i-1}}_{w_i}T_{encoder}$, and the ICP algorithm, ${}^{w_{i-1}}_{w_i}T_{ICP}$. These transformation matrices are relative to the wheelchair local frame. This means that the previous state vector has no effect on the current state vector. In this work,

EKF filter deals with the following model:

$$\begin{cases} S_i = f(0, u_i, w_i) \\ y_i = h(S_i) + v_i \end{cases} \quad (5)$$

where $S_i = [\Delta X_i \ \Delta Y_i \ \Delta Z_i \ \Delta\alpha_i \ \Delta\beta_i \ \Delta\gamma_i]^T$ is the state vector at instance i. w_i and v_i are supposed to be zero-mean Gaussian noises for the system and measurements respectively. $f(.)$ and $h(.)$ are the models of the system and the measurements respectively. y_i is the vector of measurements returned by the sensor. For each iteration, the EKF calculates the best estimate of the state vector in two stages:
Prediction stage:

$$\begin{cases} \hat{S}_i = f(0, u_i, 0) \\ \hat{P}_i = (\nabla_{s_i} \mathbf{F}) \ P_i \ (\nabla_{s_i} \mathbf{F}^T) + W_i \end{cases} \quad (6)$$

where $\mathbf{F}(.)$ is the Jacobian matrix that linearizes the system model $f(.), \hat{P}_i$ is the covariance matrix for predicted state vector $\hat{S}_i$, and W_i is the covariance matrix of the system noise w_i. The predicted state vector is calculated from system model (1) by eliminating the previous state vector variable which will yield (7). From (7), we can calculate the Jacobian $\nabla_{s_i}\mathbf{F} = \frac{\partial f(0,u_i,0)}{\partial s}$

$$\hat{S}_i = \begin{bmatrix} \Delta X_i \\ \Delta Y_i \\ \Delta Z_i \\ \Delta\alpha_i \\ \Delta\beta_i \\ \Delta\gamma_i \end{bmatrix}_{encoder} = \begin{bmatrix} -r_i sin\left(\frac{R_i - L_i}{w_b}\right) \\ r_i\left(cos\left(\frac{R_i - L_i}{w_b}\right) - 1\right) \\ 0 \\ 0 \\ 0 \\ \frac{R_i - L_i}{w_b} \end{bmatrix} \quad (7)$$

In the case when $\Delta\gamma_i = 0$, which means that the wheelchair has translation with no rotation, $\Delta X_i = \frac{(R_i + L_i)}{2}, \Delta Y_i = \Delta\gamma_i = 0$. Then the *update stage* is:

$$\begin{cases} K_i = \hat{P}_i \boldsymbol{H} \ (\boldsymbol{H}\hat{P}_i\boldsymbol{H}^T + V_i)^{-1} \\ S_i = \hat{S}_i + K_i(y_i - h(\hat{s}_i)) \\ P_i = (I_{n\times n} - K_i\boldsymbol{H})\hat{P}_i \end{cases} \quad (8)$$

where K_i is the Kalman gain at instance $i, \boldsymbol{H}(.)$ is the Jacobian matrix that linearizes the measurement model $h(.)$, $y_i = [\Delta X_i \ \Delta Y_i \ \Delta Z_i \ \Delta\alpha_i \ \Delta\beta_i \ \Delta\gamma_i]^T_{ICP}$ and V_i is the covariance matrix of the measurement noise v_i. The error covariance matrices were determined by calculating the error between the ground truth and the estimated wheelchair pose in both the encoder-based and visual odometry cases as will be explained in section V-B.

IV. WHEELCHAIR MOTION CONTROL SCHEMES

The main aim of this work is to design and implement a control motion scheme for the power wheelchair that has high uncertainty. This is a step towards making the WMRA system capable of performing high precision tasks such as "Go and Pick up" and "Open the door". The existing motion control depends solely on wheel encoders which makes the system unreliable. However, encoder-based odometry has a higher frequency rate than the ICP-based odometry. That is because the latter uses a computer vision algorithm which needs more computational power. Therefore, the idea is

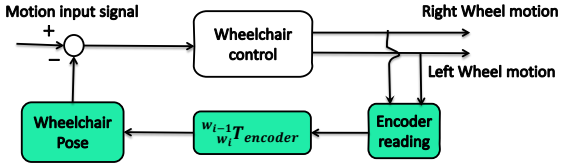


Fig. 4: Schematic diagram of the wheelchair encoder-based odometry motion control .

using the encoder-based odometry to control the wheelchair, and then update it once the ICP-based odometry output is available. The update rate should be fast enough to guarantee the accuracy of position estimation. The following two types of motion control are used:

A. Encoder only motion control

The wheelchair pose is estimated according to the encoder-based odometry method explained in section III-A. Then the wheelchair pose is compared with the motion reference input to calculate the motion error. Based on the error, a signal for rotation and/or translation is sent to the wheelchair controller to minimize this error. Fig.4 shows a schematic diagram of this controller.

B. ICP-based updated odometry motion control

In this motion control, the two methods of wheelchair pose estimation are running at the same time. Because the encoder-based odometry is faster than the ICP-based odometry, it is used to control the wheelchair similar to previous control scheme. The difference is using the ICP-based odometry to correct the encoder-based odometry (red updating link in Fig. 5) . As shown in Fig. 5, the EKF is used to fuse the local transformation matrices computed by the ICP algorithm and encoders to obtain the optimized pose estimation. Then the encoder-based odometry is updated with the ICP-based odometry. The resultant pose estimation is called ICP-based updated odometry.

V. EXPERIMENTS AND DISCUSSION

In this work, the experiments were performed in two stages: offline data processing, and online wheelchair application. In both stages, the wheelchair ground truth was captured using a state-of-the-art VICON®(Oxford, UK) system with eight motion capture cameras. The cameras use infrared lights to detect passive reflective markers attached to the WMRA system and the Kinect.The markers' locations

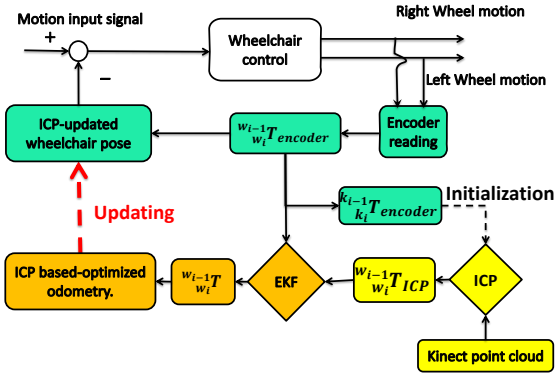


Fig. 5: Schematic diagram of the wheelchair ICP-based updated odometry motion control.

were captured at a frequency of 120 Hz. A Matlab® program was used for post motion data processing to compute the wheelchair and Kinect frame poses. An accurate transformation matrix of the Kinect coordinate frame with respect to the wheelchair coordinate frame is determined from that data.

A. *Offline data processing*

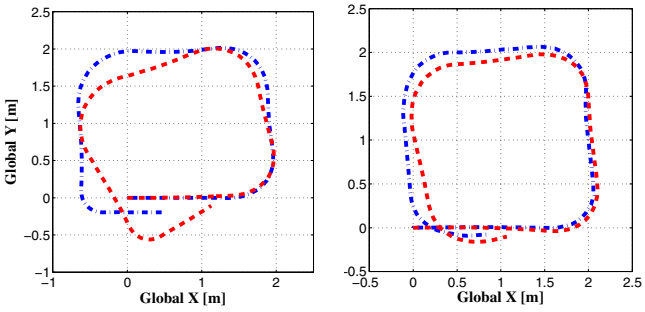
The purposes of this stage are: to refine the ICP parameters and to determine the covariance matrices of the Kalman filter. In this stage, the wheelchair was commanded to move in a square motion for five loops. It was controlled using the encoder only motion control scheme (section IV-A). Kinect point clouds associated with the encoders reading and time stamp were stored using an onboard laptop. At the same time, the ground truth of the wheelchair was captured. The synchronization process between the ground truth wheelchair pose and ICP-based updated wheelchair pose is accomplish by using time stamp saved in both programs. The open source Point Cloud Library (PCL) [27] was used to process the Kinect point clouds and to implement the ICP algorithm. For the Kalman filter covariance matrices, four covariance matrices were estimated according to the case of the wheelchair motion (translation or rotation): one translation error and one rotation error covariance matrices for encoder-based odometry and one translation error and one rotation error covariance matrices for ICP algorithm.

B. *Offline data processing results and discussion*

To obtain the covariance matrices of the EKF, the encoder-based odometry and ICP-based odometry were calculated for the five loops. By calculating the error in the local transformation matrix estimated by these methods, we were able to determine the covariance matrices of the encoder-based odometry and the ICP algorithm. This was accomplished by calculating the error in the local transformation matrix estimated by the encoder-based odometry and the ground truth. A similar calculation was performed for the ICP registration process. As stated before, the offline data processing is for evaluating the performance of our approach in terms of error detection and correction. Fig.6(a) shows the wheelchair pose estimated by the encoder-based odometry (blue-dashed line) with the corresponding wheelchair ground truth motion (red-dashed line). Only one square loop is shown for the clarity of the figure. Fig.6(a) demonstrates the high angle drift between the wheelchair pose that was estimated by the encoder-based odometry and the ground truth. The localization error was mainly due to high orientation error which made the position error accumulate faster as the wheelchair moved. This justifies the assumption we made earlier about this mobile platform high uncertainty. It is worth mentioning that we did not calibrate the encoders and the odometry equations with any odometry calibration procedure, such as UMBmark [28]. This is because we need to determine how effective our approach is in detecting and compensating the localization error in the case of platforms with high inaccuracies.

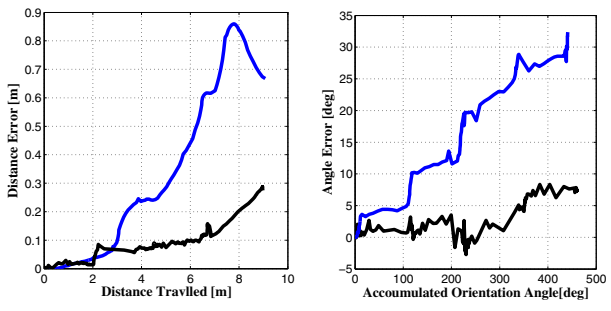
For the online implementation, the rate of updating the encoder-based odometry with the ICP-based odometry should be fast enough to guarantee the Kinects scenes have sufficient overlap, which is crucial for a successful registration process. As a result, the wheelchair pose error is corrected regularly. We achieved this by relaxing the ICP's parameters, which made the online results differ from the offline results. Fig.6(b) shows online implementation results with the wheelchair ground truth of the proposed algorithm for one square motion. The figure demonstrates 4 times improvement in wheelchair pose estimation. It shows that the wheelchair ground truth and estimated wheelchair pose are almost identical in orientation angle with some position error at the end of the loop.

For a complete comparison, Fig.7 shows the global position and orientation error for the encoder-based and ICP-based updated odometry in online implementation. These errors are the difference between the wheelchair pose estimation using the these methods and the ground truth for a one square loop. The encoder-based odometry (blue line) has larger error in both cases (position and orientation) compared with the other method. After the wheelchair moved approximately 8 meters, the encoder-based odometry has almost 0.8m position error, which is 10% of the traveled distance, while ICP-based updated odometry, the position error was 0.2m which is 2.5% of the traveled distance. This is an improvement of 4 times. More improvement can be



(a) Encoder only odometry control (b) ICP-based updated odometry control

Fig. 6: Encoder-based and ICP-based updated odometry(blue lines) for online implementation with wheelchair ground truth (red line).



(a) Position error (b) Orientation Error

Fig. 7: Global position and orientation angle error for encoder-based (blue line) and ICP-based updated (black line) odometry for online implementation.

achieved by using high computational power. The offline data processing results showed an improvement of about 15 times.

The same outcome is observed with the orientation angle; after the wheelchair rotated for approximately 360 degrees, the encoder-based odometry had almost 27 degrees of error, which is 7.5% of the rotated angle, while the ICP-based updated odometry had around 7 degrees which is around 2% of the rotated angle. This is an improvement of about 4 times. The offline data process results showed that the encoder-based odometry can be improved by using ICP-based updated odometry for about 13 times. Even though the online application improvement is less than the offline data processing, we could improve that by using more computational power such as using GPU which is considered as a future work.

Although, the Kinect cannot operate in outdoor environment, this technique can be applied in outdoors by using other sensors that suit these environments, such as 3D laser scanner. The important issue here is that the registration process needs 3D static features within the depth range of the sensor to get good alignment results. Therefore, this technique is not applicable in outdoor or indoor environments that lack these features. Using probabilistic model to fuse the measurements can cope with some failures of the alignment process. In the worst case scenario in which the registration always fails, the wheelchair localization will be solely dependent on the wheel encoders.

C. Online wheelchair application

This experiment stage was designed to test and evaluate the proposed motion control scheme. Obstacle avoidance was chosen to be the online application. This application involves many crucial tasks which need faster, more accurate motion control. These tasks are: detecting an obstacle, mapping the environment, path planning, and path execution. Another Microsoft Kinect camera was used to detect the obstacle and create a 2D-map of the environment. The A-star algorithm [29] combined with a path smoother algorithm was used for path planning [30]. At the beginning of this experiment, the wheelchair was at a zero-orientation angle relative to the global coordinate frame, and the obstacle was at an angle of around +135 degrees relative to the global coordinate frame as illustrated in Fig.8. The wheelchair was commanded to move from a point (0, 0) to a point (-2.5, 2.5) with respect to the global coordinate frame. Therefore, the wheelchair had to rotate 135 degrees first, and then the wheelchair faced the obstacle. Once the wheelchair detected the obstacle, the path planning process started. The wheelchair stopped moving and the Kalman and the ICP Algorithms were paused until the waypoints of the path were determined. The wheelchair environment was then mapped by the second Microsoft Kinect camera. The environment's 2D map was uploaded to the planning algorithm. The output of the path planning algorithm was the waypoints of the wheelchair trajectory. The wheelchair was then commanded to move from one waypoint to the other in two consecutive movements which are rotation and translation until it reached the final waypoint.

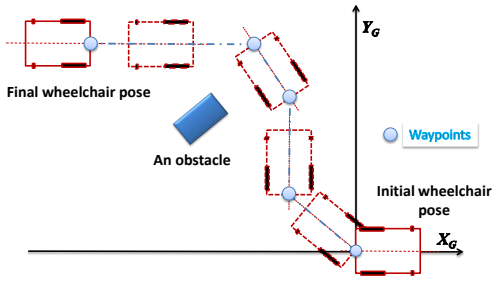


Fig. 8: Initial and final wheelchair poses and path planning.

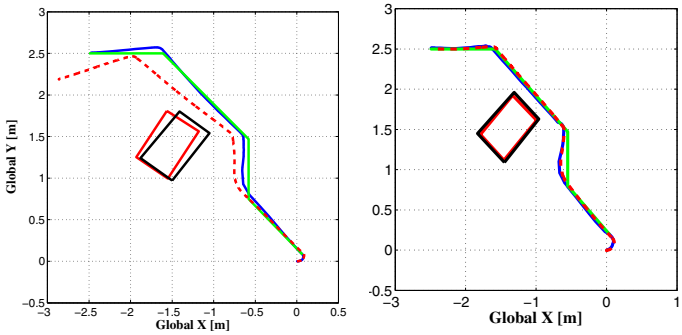
D. Online application results and discussion

The wheelchair and the obstacle ground truth were captured using the motion capture system. The obstacle was also mapped using the second Microsoft Kinect camera(black line). Fig.9(a) and Fig.9(b) show the results of the encoder only and ICP-based updated motion control schemes respectively with the ground truth of the wheelchair and the obstacle. In Fig. 9(a), the Kinect-obstacle coordinates did not match the obstacle ground truth. This was due to the encoder-based odometry orientation error.

This introduced obstacle localization error which increased the possibility of collision with the obstacle. In addition, the wheelchair pose estimated by the encoder based odometry had a considerable deviation from the wheelchair ground truth at the end of the path execution, due to the starting orientation angle error. However, in case of ICP-based updated odometry motion control scheme (refer to Fig.9(b)), the obstacle mapping was more accurate compare to the former method. This was due to the pose error correction of the ICP-based updated odometry motion control. It was also observed that the wheelchair ground truth had a near perfect match with the ICP-based updated odometry.

VI. CONCLUSION

Two methods were used to estimate the wheelchair pose using two inexpensive sensors, which were wheel encoders



(a) Encoder-based odometry control (b) ICP-based updated odometry control

Fig. 9: Wheelchair pose estimation(blue line) with the ground truth(red-dashed line for the wheelchair and red-solid line for the obstacle), Kinect-mapped obstacle(black line) and commanded trajectory (green line) for both control schemes.

and Microsoft Kinect sensor. Two wheelchair motion control schemes were designed and implemented based on two methods. These methods were encoder-based odometry and ICP-based updated odometry. Encoder-based odometry suffers from error accumulation. The ICP-based odometry can detect the orientation angle with a good accuracy, but it suffers from lacking a mechanism to cutoff the error propagation from inaccurate pose estimation. Kalman filter provides a mechanism to optimize any imprecise measurement from the ICP algorithm. This gives a smooth wheelchair pose estimation compared to the ICP algorithm. EKF was used to fuse the transformation matrices obtained from the two sensors measurements. Two stages of experiments were performed: offline data processing and online wheelchair application. In the stage of offline data processing, Kinect point clouds associated with the encoders reading were stored for five square loops. ICP algorithm parameters were refined and the covariance matrices for EKF were determined. There was a significant improvement of 15 times in position estimation and 13 times in orientation angle estimation compared to encoder-based odometry when it was corrected with ICP-based odometry. In the second stage, the proposed motion controls were tested and evaluated using online wheelchair application which was obstacle avoidance. Using ICP-based updated odometry, the wheelchair was able to correctly map and avoid the obstacle. There was an improvement of 4 times in both position and orientation angle estimation. Despite the fact that online implementation needs more computational power to obtain the same result as the offline implementation; the online application proved that the ICP-based updated odometry motion control scheme gives good results in filtering out the systematic and nonsystematic error of the encoder-based odometry.

REFERENCES

[1] J. S. Gutmann, W. Burgard, D. Fox, and K. Konolige, "An experimental comparison of localization methods," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 2, 1998, pp. 736–743 vol.2.

[2] J. S. Gutmann and D. Fox, "An experimental comparison of localization methods continued," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, 2002, pp. 454–459 vol.1.

[3] R. M. Alqasemi, "Maximizing manipulation capabilities of persons with disabilities using a smart 9-degree-of-freedom wheelchair-mounted robotic arm system," 2007.

[4] G. Ippoliti, A. Manna, and S. Longhi, "Robust robot localization by sensors with different degree of accuracy," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 3, pp. 259–276, 2009.

[5] J. Borenstein and L. Feng, "Gyrodometry: a new method for combining data from gyros and odometry in mobile robots," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1, 1996, pp. 423–428 vol.1.

[6] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme, "Robust localization using relative and absolute position estimates," in *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 2, 1999, pp. 1134–1140 vol.2.

[7] W. L. D. Lui, T. J. J. Tang, T. Drummond, and L. Wai Ho, "Robust egomotion estimation using icp in inverse depth coordinates," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 1671–1678.

[8] J. Martinez-Carranza and A. Calway, "Efficient visual odometry using a structure-driven temporal map," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 5210–5215.

[9] A. Milella, G. Reina, and R. Siegwart, *Computer Vision Methods for Improved Mobile Robot State Estimation in Challenging Terrains*, ser. 2006, 2006, vol. 1.

[10] P. J. Besl and H. D. McKay, "A method for registration of 3-d shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, 1992.

[11] Z. Zhang, "Iterative point matching for registration of free-form curves," Tech. Rep., 1992.

[12] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, 2001, pp. 145–152.

[13] G. Dissanayake, H. Shoudong, W. Zhan, and R. Ranasinghe, "A review of recent developments in simultaneous localization and mapping," in *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on*, 2011, pp. 477–482.

[14] T. Hervier, S. Bonnabel, and F. Goulette, "Accurate 3d maps from depth images and motion sensors via nonlinear kalman filtering," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 5291–5297.

[15] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[16] S. Y. Chen, "Kalman filter for robot vision: A survey," *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 11, pp. 4409–4420, 2012.

[17] B. Bischoff, N.-T. Duy, F. Streichert, M. Ewert, and A. Knoll, "Fusing vision and odometry for accurate indoor robot localization," in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*, 2012, pp. 347–352.

[18] M. Polanczyk, P. Baranski, M. Strzelecki, and K. Slot, "The application of kalman filter in visual odometry for eliminating direction drift," in *Signals and Electronic Systems (ICSES), 2010 International Conference on*, 2010, pp. 131–134.

[19] L. Tae-jae, B. Wook, J. Byung-moon, S. Ho-Jeong, and C. Dong-il Dan, "A new localization method for mobile robot by data fusion of vision sensor data and motion sensor data," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, 2012, pp. 723–728.

[20] O. Horn and M. Kreutner, "Smart wheelchair perception using odometry, ultrasound sensors, and camera," *Robotica*, vol. 27, no. 02, pp. 303–310, 2009.

[21] C. De la Cruz, T. F. Bastos, F. A. A. Cheein, and R. Carelli, "Slam-based robotic wheelchair navigation system designed for confined spaces," in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, 2010, pp. 2331–2336.

[22] R. Simpson, E. LoPresti, S. Hayashi, I. Nourbakhsh, and D. Miller, "The smart wheelchair component system," *Journal of Rehabilitation Research and Development*, vol. 41, no. 3B, pp. 429–442, 2004.

[23] Q. Zeng, T. Chee Leong, B. Rebsamen, and E. Burdet, "A collaborative wheelchair system," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 16, no. 2, pp. 161–170, 2008.

[24] "Us digital h5 ball bearing optical shaft encoder." [Online]. Available: http://www.usdigital.com/products/encoders

[25] "Kinect sensor." [Online]. Available: www.xbox.com/en-us/kinect

[26] C. Kok Seng and L. Kleeman, "Accurate odometry and error modelling for a mobile robot," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, 1997, pp. 2783–2788 vol.4.

[27] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1–4.

[28] J. Borenstein and L. Feng, "Umbmark: a benchmark test for measuring odometry errors in mobile robots," vol. 2591, 1995, pp. 113–124.

[29] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, 1968.

[30] P. Lester, "A* pathfinding for beginners," *online]. GameDev WebSite. http://www.gamedev.net/reference/articles/article2003.asp (Acesso em 08/02/2009)*, 2005.