2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

VICP: Velocity Updating Iterative Closest Point Algorithm

Seungpyo Hong Heedong Ko Jinwook Kim
Imaging Media Research Center
Korea Institute of Science and Technology
Seoul, Korea
{junon,ko,jwkim}@imrc.kist.re.kr

***Abstract*— We propose a novel method to enhance a family of ICP(iterative closest point) algorithms by updating velocity. Even though ICP algorithms play a dominant role in a model based tracking, it is difficult to avoid an accumulated tracking error during a continuous motion. It is because that typical ICP algorithms assumes that each of the point in one scan are measured simultaneously while most of the available rangefinders measure each point sequentially. Hence conventional ICP algorithms are prone to be erroneous under a fast motion and an accumulated error during the motion cannot be ignored in many cases. In our approach, we estimate a velocity of a rangefinder numerically over ICP iterations. As a result, distortion of a scan due to the motion can be compensated using estimated velocity. In addition, outliers are effectively rejected during the iteration of velocity update, which means that more accurate and robust motion is trackable. Also we verify a performance and an accuracy of our method by demonstrating simulation and real-world experiment results.**

I. INTRODUCTION

Tracking a mobile robot's position is a key issue in Robotics area and has been intensively studied for many years. A number of sensors are available for tracking themselves or tracking moving objects in an environment. Laser rangefinder is one of the most widely used sensors for the indoor position tracking. Position of a rangefinder is estimated by matching two different scans[1]. ICP(Iterative Closest Point) algorithm is one of the dominant solutions for scan matching problem by iteratively finding the closest points. ICP algorithms are originally used for geometric alignments of three-dimensional data from 3D scanner[2]. Regardless of a dimension of the data, ICP algorithms are known to find the local minimum efficiently[3].

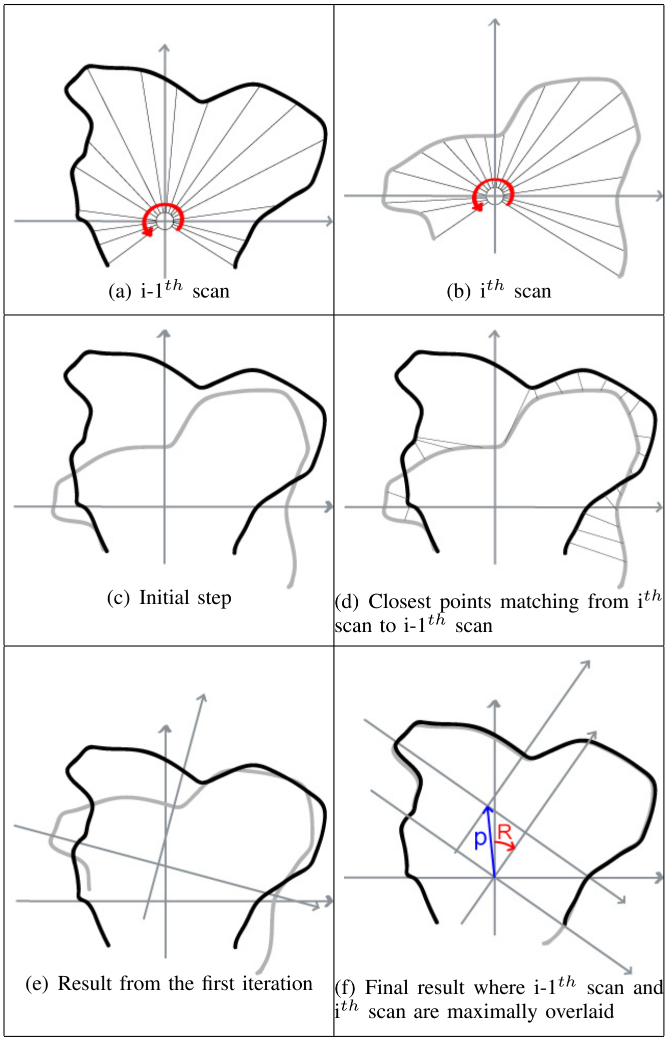
In this paper, we introduce the *VICP* algorithm, an innovative method to track a motion of a laser rangefinder by applying ICP algorithm with velocity update. Our contribution is to estimate the velocity of the rangefinder and compensate a distortion due to a scanning time difference between measurements among a set of scan data. Hence more accurate and up-to-date position can be tracked as a result. Next section, we summarize several previous research related to ICP algorithms with applications of 3D scan data alignment and 2D scan matching for localization. Then we introduce an original idea to estimate a rigid transformation(Section 3), and present a novel method with velocity update(Section 4). Finally, we compare our approach with the original method in simulated and real experimental setups(Section 5, 6).

II. RELATED WORKS

The ICP algorithm has become the dominant method for aligning three dimensional models based on the geometry[4]. A rigid transformation between a model and data is acquired by iteratively finding the closest points. Some variants of ICP use more sophisticated distance metric instead of Euclidean distance[5][6] to determine the closest points. Another variants try to find corresponding points rather than the closest points[2][7]. To accelerate the closest points searching process which is one of the bottle necks of ICP algorithm, k-d tree and closest point caching are used[8]. ICP algorithm is also used for localization of robot by matching current scan with the scan gained previously[1]. ICP algorithm have a same role in both cases of scan data alignment and motion tracking. The role is to estimate the rigid transformation between two scans so that the transformation is directly used for tracking the center point of scanning device or transforms one scan to be aligned to the other scan.

Javier Minguez, *et al.*[9] suggested a new distance metric which is suitable for minimizing rotational and translational error concurrently. As a result, their new metric distance contributes better convergence rate and more accurate correspondence matching. In addition, new distance enables detect faster motion which cannot be captured using traditional ICP algorithm. A. Diosi and L. Kleeman[10] present improved scan matching method named *Polar Scan Matching*(PSM). Their work is based on the truth that laser scan data does not use Cartesian coordinate system but polar coordinate system natively. The direct use of range and bearing measurements coupled with a matching bearing association rule and a weighted range residual minimization. PSM improves processing speed and ability to converge to a correct solution from a larger range of initial guess. E. Menegatti, et al.[11] suggest scan matching with omnidirectional camera instead of laser scan sensor. The omnidirectional vision system finds the distances of the closest transitions in the environment rather than measuring point to point distance.

In the case of 3D scan data alignment, scanning device is usually assumed to be fixed. In contrast, rangefinder usually moves during scanning in the case of motion tracking, and it causes the scan distortion that will be explained at Section 4.1. Previous researches [1][9][10] are limited to finding closest point pairs effectively with the belief that scan reflects surroundings correctly only with the white noise, although it



(a) i-1th scan (b) ith scan (c) Initial step (d) Closest points matching from ith scan to i-1th scan (e) Result from the first iteration (f) Final result where i-1th scan and ith scan are maximally overlaid

Fig. 1. ICP algorithm example.

is not true. Scan needs distortion correction process before finding the corresponding points.

Scan distortion problem was addressed by O. Bezet, V. Cherfaoui[12] in terms of time error correction. Their solution was interpolating two values which are distances to the same object and scanned at different frame. They assume that at a same angle θ, the sensor measures the distance to the same object between two frame. The drawback becomes great in a case that laser rangefinder rotates and moves fast so that their assumption becomes false. For the applications of indoor motion tracking, more accurate and fast-motion tolerant correction method is required.

III. Estimating Rigid Transformation using ICP

ICP starts with two scans and an initial guess for their relative rigid transformation, and iteratively refines the transformation by repeatedly generating pairs of closest points and minimizing an error metric[4] rather than finding corresponding points at once. ICP algorithm converges monotonically to the nearest local minimum of mean square distance metric[13]. If the initial guess of transformation is not close enough to ground truth, ICP algorithm will converge to local minimum which is not global minimum.

Let $X = \{x_i\}$ be a scan(a set of measurements), and $Y = \{y_i\}$ be a scan captured next to X. The objective of ICP is to find the rigid transformation that transforms Y to be maximally overlaid to X. The objective function is defined to minimize the objective function (1), where x_i is a point in X and y_i is a corresponding point to x_i in Y. By solving the objective function, a rotation matrix R and a translation vector p is computed. A resulting transformation matrix T can be described as (2).

$$f(R,p) = \sum_{i=1}^{n} \|Rx_i + p - y_i\|^2 \tag{1}$$

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \tag{2}$$

Because conventional ICP algorithm uses the closest point as a corresponding point, the initial result might not be close to the ground truth. However by repeating the process, a result converges to the ground truth [13]. Figure 1(a) shows X which is the $(i-1)_{th}$ scan, and Figure 1(b) shows Y which is the i_{th} scan. Figure 1(c) is the first step of ICP iteration. Iteration starts at Figure 1(c) and finds the closest points between X and Y as shown in Figure 1(d). After the first iteration, transformation T_1 is estimated and Y is updated by trasforming $y_i' = T_1 y_i$ in Figure 1(e). By repeating the process as shown in Algorithm 1, Y becomes close to X. Finally, transformation $T = T_n ... T_1$ is estimated through n times of repetition.

$$x_i = T_n T_{n-1} ... T_2 T_1 \; y_i \quad , \; i = 1, \cdots, n. \tag{3}$$

Algorithm 1 T = ICP(X, Y, T_0)

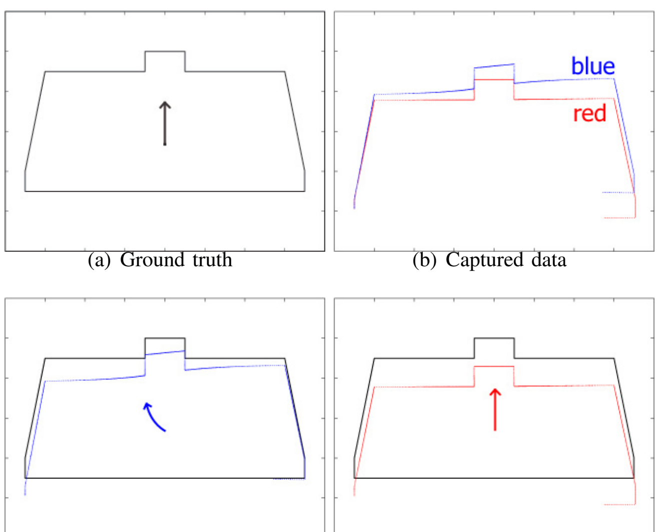
$T = T_0$
while $||T'|| > \epsilon$ **do**
  for k = 1 : n **do**
    $y_k = T y_k$
  end for
  for k = 1 : n **do**
    x_k = FindClosestPoint(X, y_k)
  end for
  $x_m = \frac{1}{n} \sum x_k$
  $y_m = \frac{1}{n} \sum y_k$
  for k = 1 : n **do**
    $x_k' = x_k - x_m$; $y_k' = y_k - y_m$
  end for
  $[U, S, V]$ = SVD($\sum y_k' \otimes x_k'$)
  $R = UV^T$
  $p = x_m - R y_m$
  $T' = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$
  $T = T'T$
end while
return T

IV. VIPC: VELOCITY UPDATING ICP

In this section, we first explain scan distortion, and then we present a methodology to compensate the distortion by estimating a velocity of a rangefinder and correcting the scan distortion at the same time.

A. Scan Distortion

Laser scan consists of a number of distance measurements because a laser rangefinder scans point by point in a discrete manner. Each of the point data in one scan is not measured at the same time. Laser scan might be distorted when a rangefinder moves during scanning. Figure 2 explains how scan is distorted while a rangefinder moves. A black line at Figure 2(a) represents an environment and the rangefinder starts moving in a direction indicated by the arrow. Blue points at Figure 2(b) represent a raw scan data. Note that it is distorted because a scanning component rotates in a counter-clockwise direction during measurements. If we try to estimate the transformation without compensating the distortion, a wrong transformation will be estimated as shown in Figure 2(c). By correcting the scan, we obtain a rectified data(red points at Figure 2(b)). Finally, a more accurate transformation is estimated from a rectified data as shown in Figure 2(d).



(a) Ground truth (b) Captured data

(c) Estimation from distorted data (d) Estimation from rectified data

Fig. 2. (a) is a given environment. Blue points in (b) shows distortion of the scan, and red points in (b) show compensated scan. Transformation estimated using distorted data includes inevitable errors(c). Transformation estimated from the rectified scan gives us more accurate results(d).

First we explain a details of scan. X^i is a scan at time t_i. Similarly, X^{i-1} is a scan at time t_{i-1}. A time difference between two scan, X^i and X^{i-1}, is Δt. Each of the scan has its own local coordinate frame and a transformation T_i represents a coordinate frame of X^i. Therefore, relative transformation from the coordinate frame of X^i to the coordinate frame of X^{i-1} can be given as $T_{i-1}^{-1}T_i$. Then a relation between X^i and X^{i-1} is represented as

$$x_k^{i-1} = T_{i-1}^{-1}T_i x_k^i \quad , k = 1, \cdots, n, \tag{4}$$

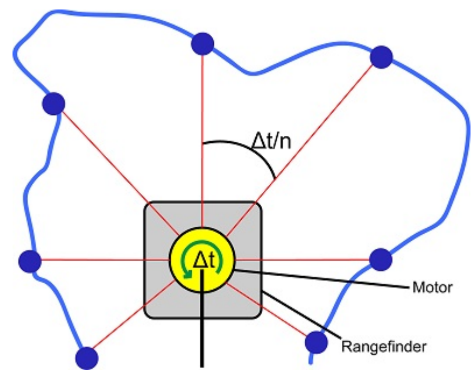


Fig. 3. Simplified structure of typical laser rangefinder

where two scans are sorted so as to find their correspondences easily.

B. Velocity Estimation and Data Compensation

In order to compensate the scan, velocity of the rangefinder has to be estimated. We assume that the velocity is constant during the scanning time. Let V_i represent a body velocity of the coordinate frame of the rangefiner at time t_i. First, V_i is approximated from the relative transformation between X^i and X^{i-1} using a backward difference;

$$V_i = T_i^{-1}\dot{T}_i \approx \frac{1}{\Delta t}\log T_{i-1}^{-1}T_i. \tag{5}$$

Then approximated V_i is used to transform further each point in X^i. Let n denotes a number of points in X^i. Then a time difference between adjacent points is Δt_s(=$\Delta t/n$) as shown in Figure 3. To be specific, $x_0, x_1, ..., x_n$ are points in X^i, and $t_{x_j} - t_{x_{j-1}} (j = 0, 1, ..., n-1)$ equals to Δt_s. Each point in X^i has its own local coordinate frame, and a transformation $T(t_i + j\Delta t_s)$ is illustrated in the equation (6).

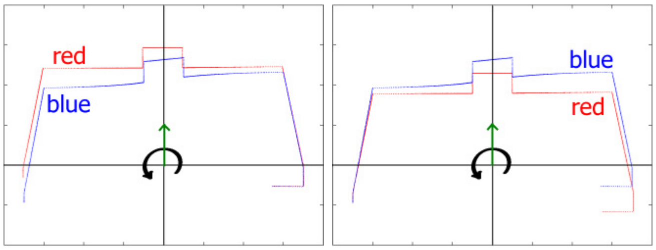
$$T(t_i + j\Delta t_s) = T_i e^{j\Delta t_s V_i} \tag{6}$$

Pluging the equation (6) to the equation (4), X^i is converted into $\bar{X}^i$ which is compensated considering the velocity as described in equation (7). Original X^i and compensated $\bar{X}^i$ is visualized in the Figure 4(a).

$$\bar{X}^* = \{e^{j\Delta t_s V_i} p_j \mid j = 0, ..., n\} \tag{7}$$

Through the equation (6) and (7), X^i can be transformed as if it is measured at the time when the first point of X^i is scanned. As a result of compensation, a tracked motion will be delayed by an amount of the time gap between the first and the last point of X^i. For a certain type of sensor which takes 200ms for one scanning with 360° scan angle, the motion will be tracked 200ms later than its actual motion. To prevent this delay, we use a backward compensation scheme. By using the time when the last point is measured as a reference time(t_i), each of the points($x_0, x_1, ..., x_{n-1}, x_n$) has its own time instance(t_i-$n\Delta t_s$, t_i-(n-1)$\Delta t_s, \cdots, t_i$-Δt_s, t_i). Therefore equation (6) is modified as

$$T(t_i - (n-j)\Delta t_s) = T_i e^{(n-j)\Delta t_s(-V_i)} \tag{8}$$



(a) Forward Compensation (b) Backward Compensation

Fig. 4. Blue and red points represents X and $\bar{X}$ respectively. A green arrow indicates a motion of the rangefinder. $\bar{X}$ is compensated as if it is measured simultaneously at the time when the first(a) and the last(b) point is scanned.

with a negative velocity. Also the equation (7) is modified as

$$\bar{X}^i = \{e^{(n-j)\Delta t_s(-V_i)} x_j \mid j = 0, ..., n\}. \tag{9}$$

The compensated scan through equation (8) and (9) are visualized in the Figure 4(b). The former compensation method is named a *Forward Compensation*, and the latter method is named a *Backward Compensation*. Note that red points in Figure 4(b) is more up-to-date than ones in Figure 4(a) where a rangefinder is assumed to move upward along the y-axis. Similarly the backward-compensated scan is more up-to-date than the original scan.

Obviously the velocity V_i includes an error because it is approximated using the distorted scan. However we can refine V_i and $\bar{X}_i$ iteratively using the previous results as shwon in Algorithm 2. Even though we do not prove the convergence, it was observed that the iteration converges always.

C. Accelerating Convergence Speed

Algorithm 2 requires nested while loops. An inner loop is for ICP and an outer loop is to approximate the velocity and refine X.

Algorithm 2 Velocity Updating ICP

$V_i = V_{i-1}$
while $||V - V_i|| > \epsilon$ **do**
  $T_{\Delta ts} = e^{\Delta ts(-V_i)}$
  for j = n : 1 **do**
    $T_{j\Delta ts} = T_{(j-1)\Delta ts} T_{\Delta ts}$
    $\bar{x}^i_j = T_{j\Delta ts}\, x^i_j$
  end for
  $T = \text{ICP}(\bar{X}^{-1}, \bar{X}^i, T)$
  $V = V_i$
  $V_i = 1/\Delta \log T$
end while

If the velocity approximation requires n iteration, total computational complexity may grow up to n times of the computational complexity of the original ICP algorithm. Fortunately, convergence of ICP algorithm is accelerated dramatically during the outer loop iteration by reusing the previous results as an initial guess of the current estimate. The iteration counts is illustrated in Figure 5. We observed that a total complexity of VICP algorithm increases less than twice of the original one.

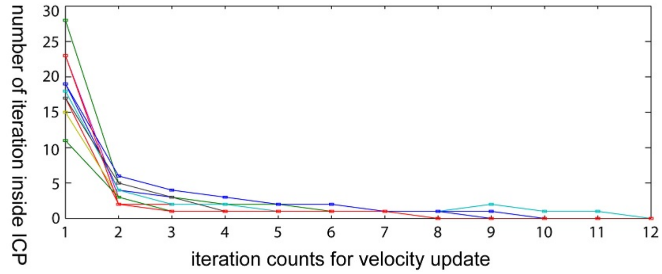


Fig. 5. A number of iteration for ICP decreases as the velocity converges. This data is sampled from the simulation shown in Figure 7(c).

Moreover, an effective outlier rejection is possible during the iteration of velocity update. During the iteration, an estimated transformation from the previous step is used as an initial guess of the current step. By comparing the initial guess and a sensing area, points in X^i which has no correspondence to X^{i-1} can be rejected during ICP. A typical sensing area of a laser rangefinder can be modeled as a pie shaped area defined by a sensing radius and sensing angle as illustrated in Figure 6.

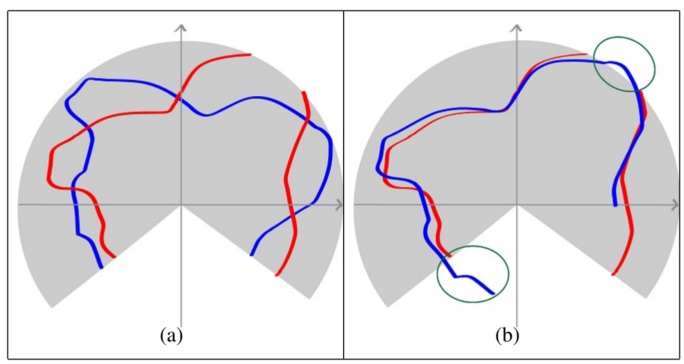


Fig. 6. (a) A blue and a red line represent X and Y, respectively. A sensing area of scanning sensor is painted in a gray color. (b) Some points(circled area) of X have no corresponding points in Y. Hence they can be ignored during ICP process.

V. Simulation

The original(ICP) and a proposed(VICP) algorithms are tested in a simulated environment. We assume to use a HOKUYO URG04-LX[14] rangefinder which has 4 meter sensing radius, 240° sensing angle, 0.36° angular resolution and 100Hz sampling rate(10 scan/sec) in a simplified environment as shown in Figure 7(a). The virtual rangefinder moves along a curved path given as an equation (10), where the parameter θ is defined as a sinc function of time t.

$$T = \begin{bmatrix} \sin(\theta) & \cos(\theta) & 0 & 1000\sin(\theta) \\ -\cos(\theta) & \sin(\theta) & 0 & -1000\cos(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Five scans($X^0, \cdots, X^4$) are generated over a time interval $[0, 0.466]$, which means the first point of X^0 is scanned at t = 0sec and the last point of X^4 is scanned at t = 0.466sec, because a time difference between each X^i is 0.1sec and a time difference between the first and the last point in one X^i is 0.066sec. To verify how efficiently VICP can track a fast motion, we compare a reference motion(Figure 7) and a two times faster motion(Figure 8). As shown in the figure, VICP shows a better result than ICP algorithm when the rangefiner moves faster.

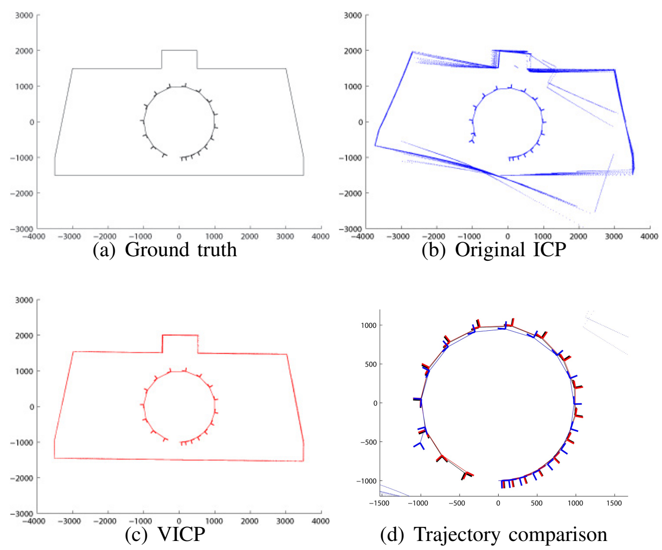


Fig. 7. A rangefinder moves along a curved trajectory where $\theta = 2\pi(1 - \mathrm{sinc}(t/2))$. (a) shows a given environment and a motion of the rangefinder. (b) shows a tracked motion and scan using ICP. (c) shows a tracked motion and refined scan using VICP. (a), (b) and (c) are merged in (d) for a comparison.

VI. Experiment

Experiments were made in an office-scale real environment with the HOKUYO URG04-LX rangefinder(Table I) All the algorithms are implemented in C++, and executed in a laptop computer with Intel dual core 2.0Gz CPU. It is difficult to know the ground truth in a real environment, thus experiments are done by returning to the starting point and measuring the drift errors. The rangefinder and the laptop are carried by a cart and driven by person along the predefined path.

Light source	λ=785nm
Accuracy	$\pm$10mm
Resolution	$1mm$
Scan Angle	240°
Range	4000mm
Angular Resolution	0.36°
Scan Time	100 msec/scan
External dimension(W*D*H)	50 * 50 * 70mm

TABLE I
Spec. of URG04-LX laser scan sensor

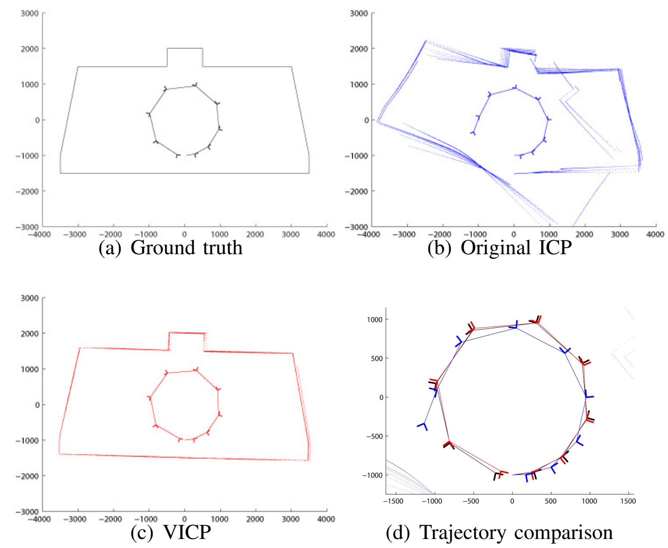


Fig. 8. A rangefinder moves faster long a curved trajectory where $\theta = 2\pi(1 - \mathrm{sinc}(t))$. (a) shows a given environment and a motion of the rangefinder. (b) shows a tracked motion and scan using ICP. (c) shows a tracked motion and refined scan using VICP. (a), (b) and (c) are merged in (d) for a comparison.

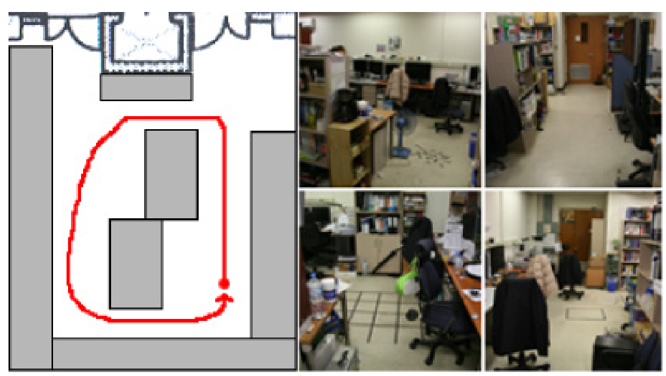


Fig. 9. Floor plan and pictures of test environment(W x H = 7.2m x 7.8m) with several desks, bookshelves ,chairs and many stuff

Experiments in Figure 10 are tested along the red line of Figure 9) while experiments in Figure 11 are tested by going straight 4.5 meter forward and turning around then coming back to the starting position. Experiment 2 and experiment 4 were tested with a faster motion than experiment 1 and experiment3. Average velocities of the expereiment 1 and 2 are approximately 1.2 m/s and 2.7 m/s respectively. VICP algorithm shows a more stable tracking result for a faster motion than ICP algorithm. Through the experiments 1-4, VICP algorithm gives better results consistently as listed in Table II.

VII. Conclusion

In this paper, we suggest *VICP*, an innovative scan matching method by updating a velocity and compensating scan simultaneously. In our approach, distortion which comes from time difference during scanning is compensated by iteratively refining a velocity of a rangefinder. Although velocity

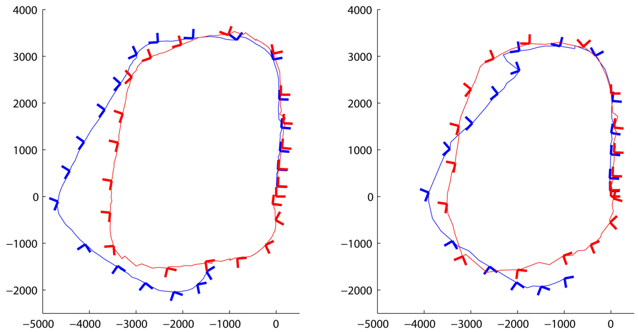


Fig. 10. Experiments with a curved motion: blue(ICP), red(VICP)

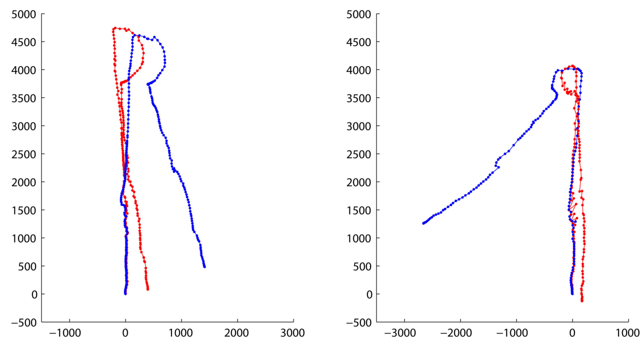


Fig. 11. Experiments with a back and forth motion: blue(ICP), red(VICP)

updating iteration nests ICP iteration, a total computational complexity increases less than twice of the original one, because convergence is accelerated dramatically by reusing the previous result as an initial guess. Also, more up-to-date position is acquired through a backward compensation and an effective outlier rejection is possible with velocity update by cropping sensor's scanning area. A simulation and real world experiments shows that VICP provides more accurate tracking especially for faster motion compared to the original ICP algorithms.

Acknowledgements. This research project was supported by the Sports Promotion Fund of Seoul Olympic Sports Promotion Foundation from Ministry of Culture, Sports and Tourism.

	Original ICP		VICP	
	R. Error	T. Error	R. Error	T. Error
Experiment 1	58.14°	2191mm	7.28°	177mm
Experiment 2	79.98°	2014mm	17.06°	65mm
Experiment 3	16.80°	1490mm	6.88°	408mm
Experiment 4	54.59°	2942mm	3.28°	210mm

TABLE II
COMPARISON OF DRIFT ERRORS BETWEEN ICP AND VICP ALGORITHMS. (R. ERROR AND T. ERROR REPRESENT ERRORN IN ROTATION AND TRANSLATION RESPECTIVELY)

REFERENCES

[1] F. Lu and E. Milios, "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans," *Journal of Intelligent and Robotic Systems*, vol. 18, no. 3, pp. 249–275, 1997.
[2] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," *Robotics and Automation, IEEE International Conference on*, pp. 2724–2729, 1991.
[3] C. Chen, Y. Hung, and J. Cheng, "A fast automatic method for registration of partially-overlapping range images," in *Proc. International Conference on Computer Vision*, 1998, p. 242.
[4] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proc. Intl. Conf. on 3D Digital Imaging and Modeling*, 2001, pp. 145–152.
[5] G. Godin, M. Rioux, and R. Baribeau, "Three-dimensional registration using range and intensity information," in *Proc. SPIE Conference*, 1994, pp. 279–290.
[6] K. Pulli, "Multiview registration for large data sets," *Proc. 3-D Digital Imaging and Modeling*, pp. 160–168, 1999.
[7] S. Weik, "Registration of 3-d partial surface models using luminance and depth information," in *Proc. International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, 1997, p. 93.
[8] D. A. Simon, "Fast and accurate shape-based registration," Ph.D. dissertation, Pittsburgh, PA, USA, 1996.
[9] J. Minguez, L. Montesano, and F. Lamiraux, "Metric-based iterative closest point scan matching for sensor displacement estimation," *Robotics, IEEE Transactions on*, vol. 22, no. 5, pp. 1047–1054, Oct. 2006.
[10] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," *Journal of Intelligent and Robotic Systems*, vol. 26, no. 10, pp. 1125–1153, 2007.
[11] E. Menegatti, A. Pretto, A. Scarpa, and E. Pagello, "Omnidirectional vision scan matching for robot localization in dynamic environments," *Robotics, IEEE Transactions on*, vol. 22, no. 3, pp. 523–535, 2006.
[12] O. Bezet and V. Cherfaoui, "Time error correction for laser range scanner data," in *Information Fusion, International Conference on*, July 2006, pp. 1–7.
[13] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal.*, vol. 14, no. 2, pp. 239–256, 1992.
[14] H. Kawata, A. Ohya, S. Yuta, W. Santosh, and T. Mori, "Development of ultra-small lightweight optical range sensor system," in *Proc. Intelligent Robots and Systems, IEEE/RSJ International Conference on*, 2005, pp. 1078–1083.