Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2010
August 15-18, 2010, Montreal, Quebec, Canada

DETC2010-29192

HUE ASSISTED REGISTRATION OF 3D POINT CLOUDS

Hao Men
Graduate Research Assistant

Kishore Pochiraju
Associate Professor

Department of Mechanical Engineering
Stevens Institute of Technology, Hoboken, New Jersey 07030, U.S.A.

ABSTRACT

This paper presents a variant of the Iterative Closest Point (ICP) algorithm for merging multiple color point clouds generated from a mobile 3D Light Detection and Ranging (LIDAR) System. This algorithm uses hue information generated from a camera along with the coordinates of the scan points and enables high accuracy registration of point clouds. A k-d tree based nearest neighbor search associates corresponding colored points in 4-D space between data and model point clouds. Singular Value Decomposition (SVD) method solves for the rigid rotation and translation. Experimental results illustrate that 3D color point clouds accelerate the 3D map registration if the hue data and model point clouds have sufficient hue distribution and the imaging sensor robustly captures the hue.

1. INTRODUCTION

Mobile mapping requires registration of map segments from various vantage points or those taken at various times. Due to the distance restrictions of range sensors and robot positioning constraints, 3D point clouds obtained from a single vantage point are seldom adequate to construct complete maps. Therefore, it is important to register point clouds obtained from different vantage points together to construct a large-scale global 3D map [1-4]. Registering the map segments is trivial if precise position and orientation of the sensor are known about a global reference frame. However, sensing the position and orientation of the robot accurately is challenging and use of registration algorithms based on identifying common features and geometries in the two map segments may be accurate, efficient and economical. A rigid both transformation with translation and rotation is obtained from the map registration process determining the map sensor position in 3D space and its orientation [5, 6]. The map registration quality can vary depending upon the accuracy required by the application. While high definition surveying may need sub-centimeter or millimeter accuracy, robotic exploration may only require much coarser registration.

Generation of point clouds is the most common 3D map format in mobile robotic mapping. Discrete range points received from range sensor describe spatial information about the environment. Different techniques exist for merging 3D maps together by exploiting geometric features and measuring surfaces. The most popular registration algorithm for point cloud map registration is iterative closest point (ICP) algorithm [2], in which the corresponding closest points in different point clouds are associated and optimal rigid transformation to minimize a mean-square error of separation between associated points of the two data sets [3] is iteratively found. Upon convergence, ICP algorithm terminates at a minimum [2]. Several algorithms are in existence for calculating the minimum average distance between two point clouds. Singular Value Decomposition (SVD) method [3], eigen-system methods that exploit the orthonormal properties of the rotation matrices, and unit and dual quaternion techniques were adopted in ICP techniques [9]. Quaternion based algorithms have been used in ICP for map fusion in [2], SVD based algorithms are widely used in ICP and 6DOF SLAM [6, 7] as they are robust [3] to reach local minimum and easy to implement.

Different variants of ICP have been investigated [8]. Corresponding points sampling, matching, weighting and rejecting are some methods used to accelerate the ICP algorithm. In the ICP algorithm, associating corresponding points in two point cloud data sets is the most critical step. Nearest neighbor search in 2D or 3D space is commonly used for associating the corresponding points. Parallel ICP algorithms have been developed [10] to accelerate computation speed.

As vision sensors are now integrated into laser ranging systems [11], 3D point clouds also contain the color properties in the scene. In this effort, the color attributes of the range point is utilized in ICP progress to increase computational speed and provide higher accuracy. The color attribute of the scanned point from Red-Green-Blue (RGB) space is translated into Hue-Saturation-Lightness (HSL) space and the hue value is used along with the coordinate data during the corresponding point association step. Prior work on hue association is based on filtering the point set on hue before ICP [11]. Some preliminary work on processing images to extract corresponding visual features for registration has also been reported [12].

This paper introduces a hue assisted ICP algorithm for registration of color point clouds. The criteria for association are defined on a 4D space rather than 3D geometric space. The 4^{th} dimension selected is the hue value representing the intrinsic color values of the scene. While achieving the effect of a hue-based filter, hue-association also reduces the nearest neighbor search burden considerably. The remaining sections of the paper describe the approach and the performance of the algorithm under several hue distributions in the scene.

2. APPROACH

2.1 Introduction to ICP Algorithm

ICP algorithm is an iterative process that calculates rigid transformation matrix based on associating two point clouds. The point cloud defined about a known reference frame is termed as the model point cloud and that being registered as the data point cloud. We use a SVD based mean square error minimization method to solve for the transformation matrix that registers the data point cloud into the model reference frame [3]. A transformation error function, E(R,T) is defined as Eq. (1) that defines the distance between associated points.

$$E(R,T) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} \mathrm{W_{ij}} \left\| \overline{\mathrm{m_i}} - \left(\mathrm{R}\overline{\mathrm{d_j}} - \mathrm{T}\right) \right\|^2 \quad (1)$$

R and T are the relative 3D rotation and translation matrix for data point cloud transformation into the reference frame. N_m and N_d are the number of related points in model and data point clouds, respectively. $\overrightarrow{m_i} = \{m_{ix}, m_{iy}, m_{iz}\}$ represents the coordinates of the i^{th} point in the model point cloud and $\overline{d_j} = \{d_{jx}, d_{jy}, d_{jz}\}$ is the j^{th} point in data point cloud. W_{ij} is weight for the association, which is assumed unity in this effort indicating that each associated point contributes equally to the error.

The point association or correspondence between the data and model point clouds is based on a nearest neighbor search using a k-d tree. The transformation matrix R and T are then updated by minimizing Eq. (1). The point association and update of the transformation matrices, R and T is iteratively performed until the specified convergence criterion is reached. This variant of the ICP algorithm has been proved to be convergent [2].

2.2 Point Cloud Association in 4-D space

The ICP computation speed and precision are highly dependent on association process. Use of a k-d tree for closest point search and association or the Nearest Neighbor Search (NNS) problem increases the speed and efficiency of the search. The k-d tree is a spatial partitioning data structure that stores and organizes data in a k dimensional space. The k-d tree is a generalized type of binary tree, with every leaf node is a k-dimensional data point that splits the hyperspace into two subspaces. Splitting is done sequentially from the first dimension to the k^{th} dimension. A typical k-d tree in 2D space is shown in Fig. 1. Each point in the 2D space divides the space sequentially into a left-right spaces (about x-axis) or into a top-bottom spaces (about y-axis).

Nearest neighbor search can be done very efficiently on k-d trees. For a given point with known coordinates in the data point cloud and a search radius, the algorithm recursively moves down the tree and follows the same procedure as insertion. Search stops at a leaf node of the tree and the points in the model tree within the search radius are identified. The nearest point is obtained using distance computation. Figure 2 shows the nearest neighbor (red square) for the search point at the center of the circle. The nearest point is then regarded as the point associated with the search point.

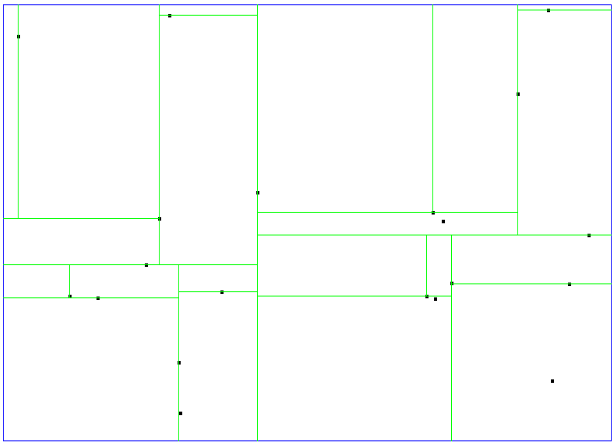


Figure 1 k-d tree construction in 2D space

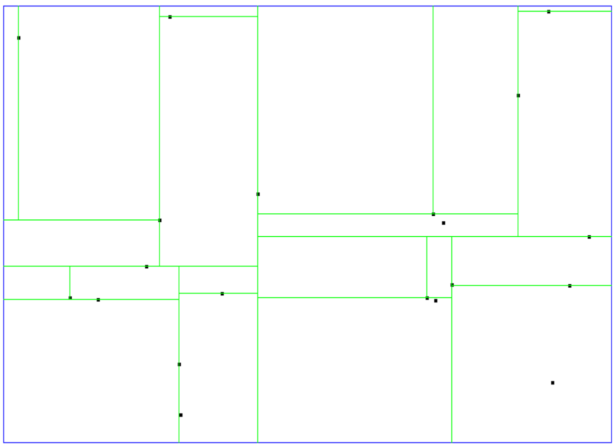
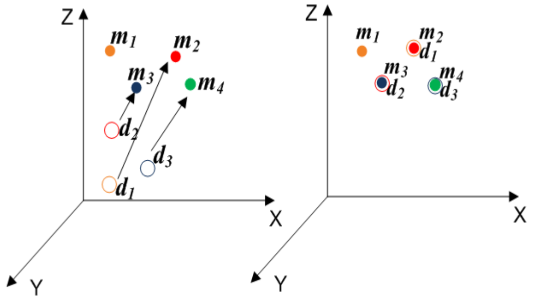


Figure 2 2D space nearest neighbor search in k-d tree

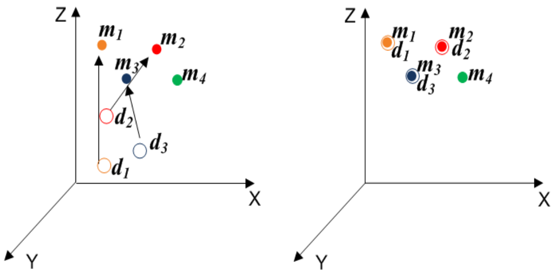
2.3 Color Assisted ICP Algorithm

Object color described in the RGB (red, green, blue) space is easily affected by light and camera performance. In order to ensure robustness to color search, only hue component of the color in the HSL (hue, saturation and lightness) space are used. Hue is used as the fourth dimension in the point association process as it is regarded to be independent of light condition and robustly represents the object color property. Hue value is weighted to construct a 4D k-d tree along with *x y z* range value for point association. Compared to 3D range point association (Fig. 3), more accurate corresponding point search can be expected from the 4D color point association when the object has a non-homogeneous hue distribution as shown in Fig. 4.



(a) 3D point association (b) 3D matching result

Figure 3 3D ICP point association



(a) Point association in 4D space (b) 4D matching result

Figure 4 Color ICP point association

A typical range based nearest point association process is shown in Fig. 3. The data point cloud $\boldsymbol{D}\{\boldsymbol{d}_i, i=1,2,3\}$ must be registered into model point cloud $\boldsymbol{M}\{\boldsymbol{m}_i, i=1,2,3,4\}$. The registration process is based on spatial information in 3D space, the closest point search results in association as shown in Fig. 3(a). After matching and transformation, data point cloud is then shown in Fig3 (b). Color ICP takes into account the hue information from color space during the point association process. Fig. 4(a) illustrates the initial position of data and model point sets. Each point now has the hue information and has the coordinates $\{x, y, z, h\}$ in a 4D space. Compared to typical ICP, registration process in color ICP associates points based on the closest distance by combining the coordinate and hue value.

Addition of hue in the nearest neighbor match is especially significant in those instances where the coordinate based matching results in non-unique registration. For example, if the points in the model and the data point clouds belong to a plane, traditional coordinate based ICP results in non-unique association of points. In such cases using the hue value (if differences exist in this dimension in the matched features) may result in unique registration of the points

The computational complexity of associating range data ICP algorithm is $O(n^2)$. This hue value based color assisted ICP algorithm holds the same computational complexity with range data ICP algorithm. As the hue value of a point remains invariant with the registration transformation during the ICP iteration loop, the algorithm acts as filter in the nearest point search process.

The color assisted ICP algorithm in this paper is as follows:

1. Estimate the initial transformation matrix $\boldsymbol{R}$ and $\boldsymbol{T}$;

*2.*Construct k-d tree of model point cloud $\boldsymbol{M\{m_1,m_2,m_3 \ldots m_M\}}$, hue value has been weighted as the 4^{th} dimension;

3. **While** merging error ε >*preset error*

Use $\boldsymbol{R}$ *and* $\boldsymbol{T}$ *to transfer data point cloud* $\boldsymbol{D\{d_1,d_2 \ldots d_N\}}$.

$$\overline{D} = R\overline{D} + T$$

4. ***For*** *i*=1 ***to*** length of data point cloud

Search closest point for point $\boldsymbol{d_i}$ $\{d_{ix},d_{iy},d_{iz},d_{ih}\}$ in model k-d tree

If closest point $\boldsymbol{m_j}$ exists in search range r

Pair $\boldsymbol{d_i}$ and $\boldsymbol{m_j}$ as $\{\boldsymbol{d_k},\boldsymbol{m_k}\}$;

k++;

End If

End For

5. Acquire paired point cloud $\boldsymbol{D_p}$ and $\boldsymbol{M_p}$, contain N points, calculate merging mean square ε as error:

$$\varepsilon = \sum_{i=1}^{N}[(d_{ix}-m_{ix})^2+(d_{iy}-m_{iy})^2+(d_{iz}-m_{iz})^2]$$

6. Calculate mean value in paired point cloud $\boldsymbol{D_p}$ and $\boldsymbol{M_p}$,

$$\overline{c_d} = \{\overline{d_{px}}, \overline{d_{py}}, \overline{d_{pz}}\}$$

$$\overline{c_m} = \{\overline{m_{px}}, \overline{m_{py}}, \overline{m_{pz}}\}$$

Construct new data set $\boldsymbol{D'}$ and model set $\boldsymbol{M'}$, in which, $\boldsymbol{d_i'=d_{pi}-c_d}$, $\boldsymbol{m_j'=m_{pj}-c_m}$;

7. Construct *H* matrix for singular value decomposition,

$$H = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix}$$

$$S_{xx} = \sum_{i=1}^{N} m'_{ix} d'_{ix}$$

$$S_{yy} = \sum_{i=1}^{N} m'_{iy} d'_{iy}$$

$$S_{xx} = \sum_{i=1}^{N} m'_{ix} d'_{ix}$$

$$S_{xy} = \sum_{i=1}^{N} m'_{ix} d'_{iy} \ldots$$

8. Solve ***R*** *&* ***T*** *using SVD*

SVD(H),

$$H = U \Lambda V^T$$

$$R = VU^T$$

$$\boldsymbol{T = C_m - RC_d}$$

End While

The differences between this color ICP algorithm and a typical ICP algorithm are in Step 2 and Step 4.

3. ALGORITHM PERFORMANCE

In this section, we describe the performance of the algorithm under various hue distribution scenarios on the same geometric model, the Stanford bunny point cloud. In HSL color space, hue value varies from 0- 360. The color correspondence between RGB and hue is given in Table 1.

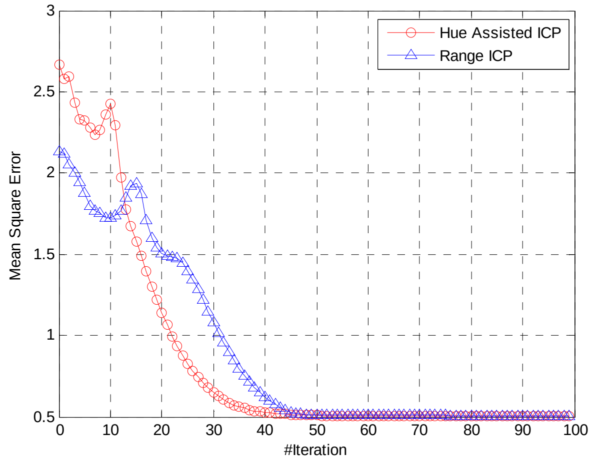
3.1 Hue Varied Color Point Cloud Map Registration

For the first experiment, we colorized he Stanford bunny point cloud model as shown in Figure 5. In this model, the hue varies from 0 to 360 with from bottom to top at Z direction in seven segments. Figure 5 also shows the initial registration of the model and data point clouds used for this simulation.

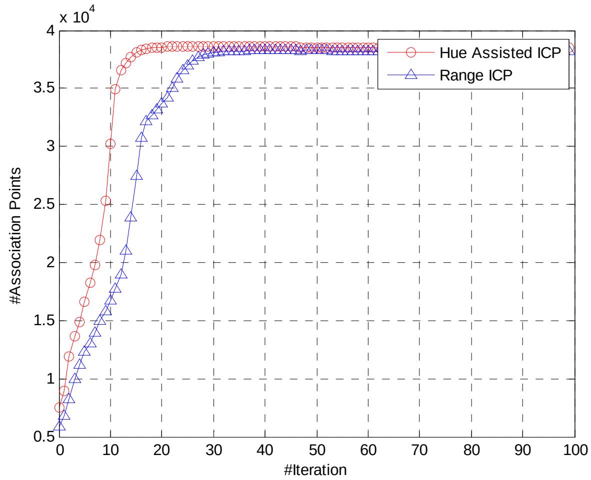
Color	R	G	B	Hue
Gray	128	128	128	0
Yellow	255	255	0	60
Green	0	255	0	120
Cyan	0	255	255	180
Blue	0	0	255	240
Magenta	255	0	255	300
Red	255	0	0	360

Table 1 Varied hue and corresponding color in RGB space

Figure 5 Varied hue rendered Stanford bunny



(a) Mean square error comparison in ICP progress



(b) Associated point number comparison in ICP progress

Figure 6. Hue assisted ICP results compare with range ICP

Figure 7 Registered 7 segment hue color point cloud

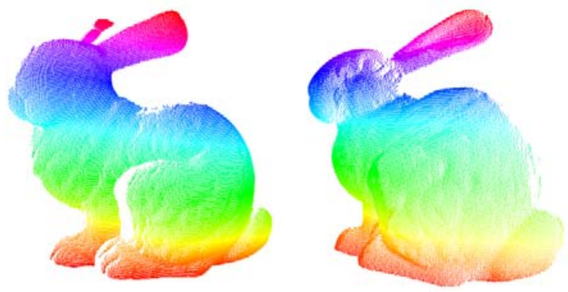
Hue assisted ICP registration progress is shown in Figure 6. Figure 6(a) shows the mean square error during the ICP process and Figure 6(b) shows the number of points associated during iteration loops. Both data and model point cloud after registration is shown in Figure 7. The hue-assisted ICP registers the point and data clouds faster than the traditional coordinate based ICP.

3.2 Continuously Varied Hue along One Dimension

In the second simulation, a continuous hue distribution is assigned to the bunny model. The hue value is varied from 0 to 360, smoothly, along the z (vertical) direction. The resultant model and data clouds are shown in Figure 8. Saturation and lightness value have been set as constant at every point inside dataset. Hue value can be calculated by Eq. (2).

$$h = 360 \frac{z_i - z_{min}}{z_{max} - z_{min}} \tag{2}$$

h is the hue value at range point i, z_i is the coordinate distance for i^{th} point at z direction, z_{max} and z_{min} are maximum and minimum coordinate of the point cloud at z direction.



(a) Data point cloud (b) Model point cloud

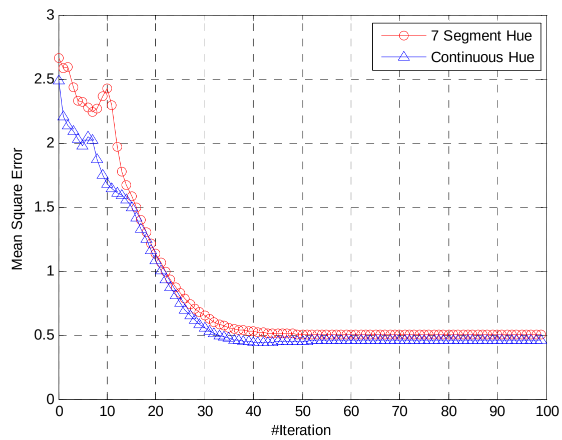
Figure 8: Bunny model with continuous hue variation in one axis

Figure 9 Merged Continuous Hue bunny

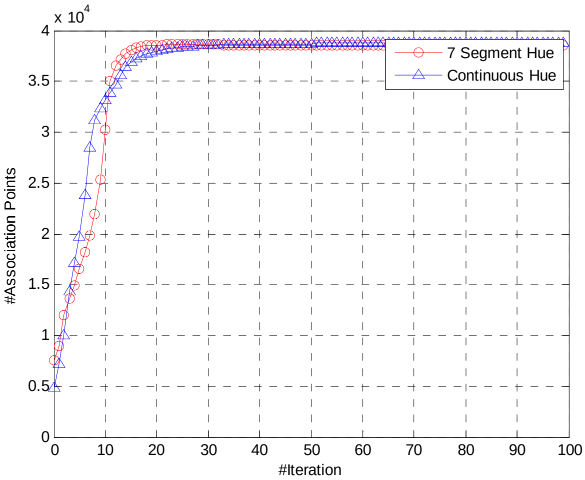
Continuous hue distribution on point cloud data is registered and the results are shown in Figure 10. A comparison of model performance on discrete and continuous distribution of hue on the same model shows the expected acceleration in performance due to uniform distribution of hue on the model.

3.3 Randomized Hue on the Model

In this case, the model considered has a continuously distributed hue but with a randomized and noisy pattern. In this case, there is no geometric pattern for the color on the object. The color point clouds are rendered in Figure 11. The merged cloud point cloud after registration is shown in Figure 12. Figure 13 shows the error minimization iteration and comparison with the seven-segment hue distribution model. In this case the hue confuses the nearest neighbor search. The registration accuracy is also not as good as a patterned hue case.

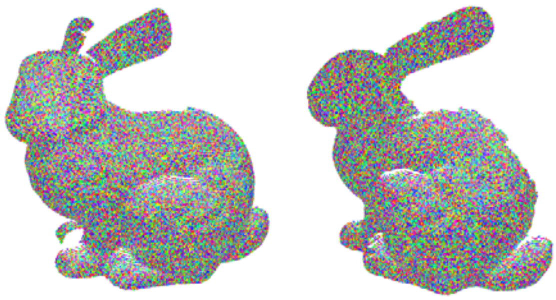


(a) Mean square error comparison in ICP progress



(b) Associated point number comparison in ICP progress

Figure 10. Comparison between discrete and continuous hue distribution cases



(a) Data point cloud (b) Model point cloud

Figure 11 Random Hue rendered bunny

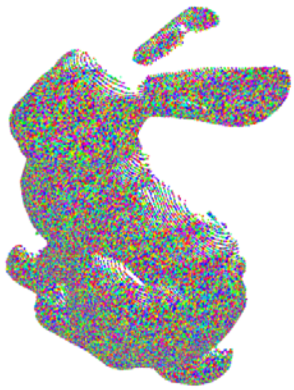
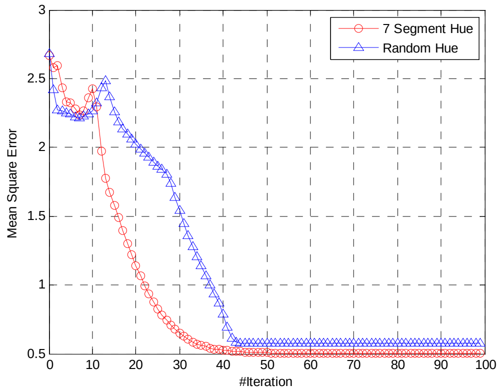
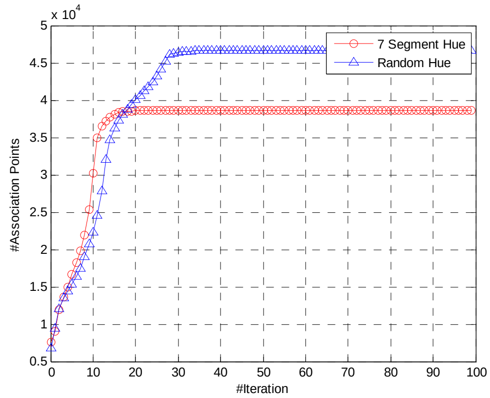


Figure 12 Merged model with randomized hue



(a) Mean square error comparison in ICP progress

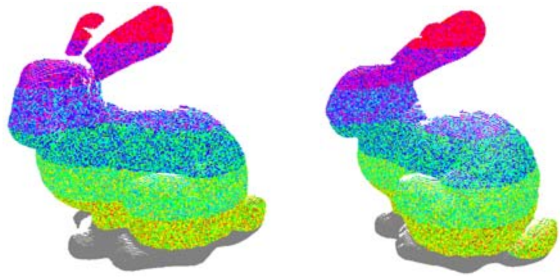


(b) Associated point number comparison in ICP progress

Figure 13. Comparison between discrete and random hue distribution case

3.4 Effect of Camera Noise

In the previous simulation, the imaging sensor is assumed perfect. The hue on a point is assumed to be recorded by the imaging sensor perfectly in both model and data clouds. Some noise in the color measurement can be expected when the point clouds are generated from two vantage points [13]. Considering this situation, we colorized the bunny model but with 50% noise in the sensor. The points in the model and data clouds differ in color by as much as 50%. The resulting point clouds are shown in Figure 14. The merged color point cloud is shown as Figure 15.



(a) Data point cloud (b) Model point cloud

Figure 14 Varied Hue with 50% noise rendered bunny model

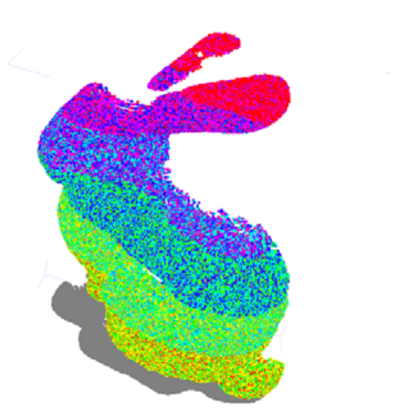
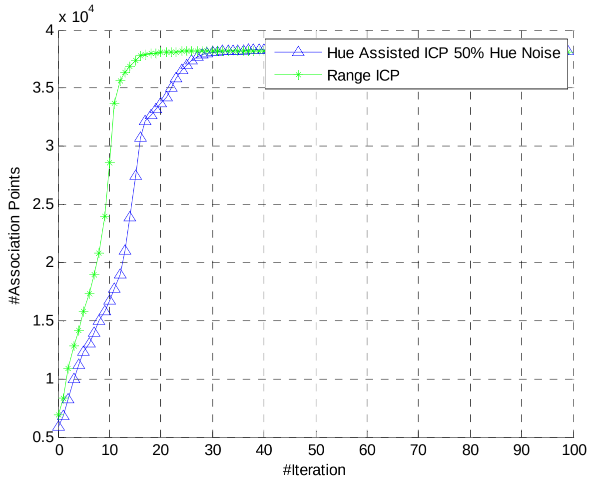
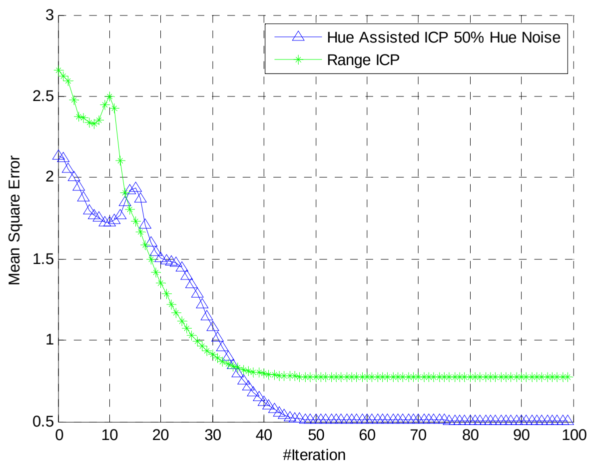


Figure 15 Merged noisy color point with 50% hue noise



a) Mean square error comparison in ICP progress



(b) Associated point number comparison in ICP progress

Figure 16. Comparison between color ICP in noise hue data and range ICP results

Hue assisted color ICP matching result in camera noise color point cloud is compared with range ICP matching performance. From Figure 16, noise in hue decreases the matching accuracy and reduces the iteration efficiency.

4. CONCLUSION AND FUTURE WORK

The color ICP algorithm developed in paper was applied to match a standardized data set with several texture happing. Use of the hue value to assist the point association and error minimization is shown to be effective during the ICP iteration schemes when there is a patterned hue on the object and when the camera imaging noise is low. Work is in progress to use the luminosity and the camera, light and material interactions for faster scan registration performance. . However, in HSL data space, Lightness should change according to the view angle and light position. Corresponding point search using additional lightness value could be a further research field to increase Color ICP algorithm. The developed algorithms are being used to register large scale 3D map data obtained using a mobile mapping robot [13].

References

[1] S. Thrun, “Robotic Mapping: A Survey”, Exploring artificial intelligence in the new millennium, Morgan Kaufmann Publishers Inc, pp 1-35, 2003.

[2] PJ. Bsel, “A Method for Registration of 3D Shapes”, IEEE Trans on Pattern Analysis and Machine Intelligence, Vol. 14, pp239-256, 1992

[3] K. S. Arun, “Least Square Fitting of Two 3D-Point Sets”, IEEE Trans on Pattern Analysis and Machine Intelligence, Vol.9 No.5, pp698-700, 1987.

[4] J. Timotee, “Fast ICP Algorithms for Shape Registration”, Pattern Recognition, , Springer Berlin, pp 91-99, 2002.

[5] S. Thrun, “Exploration and Model Building in Mobile Robot Domains”, IEEE International Conference on Neural Networks, Vol.1 , pp 175-180,1993, San Francisco, USA.

[6] A. Nuchter, “6D SLAM with Approximate Data Association”, IEEE International Conference on Robotics and Automation, pp242-249, 2005, Barcelona, Spain.

[7] A. Nuchter, “Catched k-d tree Search for ICP Algorithms”, 6th International Conference on 3-D Digital Imaging and Modeling, 2007.

[8] S. Rusinkiewucz, “Efficient Variants of the ICP Algorithm” 3rd International Conference on 3-D Digital Imaging and Modeling, pp145-152, 2001.

[9] A. Lorusso, “A Comparison of Four Algorithms for Estimating 3D Rigid Transformations”, British Machine Vision Conference, 1995

[10] C. Robertson, “Parallel Evolutionary Registration of Range Data”, Computer Vision and Image Understanding, Vol 87, pp 39-50, 2002

[11] S. Druon, “Color Constrained ICP for Registration of Large Unconstructed 3D/Color Data Sets”, IEEE International Conference on Robotics and Automation, pp249-255, 2007.

[12] H. Andresson, “Vision Aided 3D Laser Scanner Based Registration”, roc. European Conference on Mobile Robots: ECMR, pp192-197, 2007.

[13] B. Gebre, H. Men, K. Pochiraju, “Remotely Operated and Autonomous Mapping System(ROAMS),” 2nd Annual IEEE International Conference on Technologies for Practical Robot Applications, Woburn, MA, 2009