Algorithms for 3D Map Segment Registration

Hao Men
Graduate Research Assistant
Kishore Pochiraju
Associate Professor

Design & Manufacturing Institute & Department of Mechanical Engineering
Stevens Institute of Technology, Hoboken, NJ 07030, USA

ABSTRACT

Many applications require dimensionally accurate and detailed maps of the environment. Mobile mapping devices with laser ranging devices can generate highly detailed and dimensionally accurate coordinate data in the form of point clouds. Point clouds represent scenes with numerous discrete coordinate samples obtained about a relative reference frame defined by the location and orientation of the sensor. Color information from the environment obtained from cameras can be mapped to the coordinates to generate color point clouds. Point clouds obtained from a single static vantage point are generally incomplete because neither coordinate nor color information exists in occluded areas. Changing the vantage point implies movement of the coordinate frame and the need for sensor position and orientation information. Merging multiple point cloud segments generated from different vantage points using features of the scene enables construction of 3D maps of large areas and filling in gaps left from occlusions. Map registration algorithms identify areas with common features in overlapping point clouds and determine optimal coordinate transformations that can register or merge one point cloud into another point cloud's coordinate system. Algorithms can also match the attributes other than coordinates, such as optical reflection intensity and color properties, for more efficient common point identification. The extra attributes help resolve ambiguities, reduce the time and increase precision for point cloud registration. This chapter describes a comprehensive parametric study on the performance a specialized Iterative Closest Point (ICP) algorithm that uses color information. This Hue-assisted ICP algorithm, a variant developed by the authors, registers point clouds in a 4-D (x, y ,z, hue) space. A mobile robot with integrated 3D sensor generated color point cloud used for verification and performance measurement of various map registration techniques. The chapter also identifies various algorithms required to accomplish complete map generation using mobile robots.

1. INTRODUCTION

Complete and dimensionally accurate maps of the environments are of interest to many domains including surveying, search and rescue, security, defense and construction. Laser based scanning devices (Light Detection And Ranging-LIDAR) are generally used to generate point clouds that describe spatial information in the form of numerous discrete point coordinate measurements. Point data are acquired by measuring time of flight of scattered light or phase shift between incident and reflected light to find the distance between the object surface and the scanning device (Blais, 2004). The speed of scanning discrete points can be enhanced by pulse and phase based measurement technologies (Blais, 2004). Precise rotation mechanisms with high-resolution encoders spin a 2D LIDAR device to generate a 3D point cloud. Point cloud scanners have been mounted on airplanes (Browell et. al. 1990) and ground vehicles (Gebre, et al. 2009) to create large area terrain maps. When vision sensors are integrated with the laser ranging systems, point clouds can also contain the color information of the scene. Optical imagery from the camera is associated with point coordinates to produce color point clouds (Andresson, 2007).

A 3D point cloud obtained from a single vantage point is seldom adequate to construct a complete map. Generation of a complete map of an environment requires merging or registration of map segments taken from various vantage points. The registration enables construction of large-scale global 3D maps (Thrun, 2003). Registering the map segments is trivial if precise position and orientation of the sensor are accurately known about a global reference frame. Position sensors such as inertial measurement units or those relying on global positioning systems are prone to errors and can be highly inaccurate under certain conditions. The map registration process determines the rigid body translation and rotation of the sensor as its output (Thrun, 1993, 2003). The map registration quality varies depending upon the sensor resolution and the extent of overlap between the map segments. Different techniques exist for merging 3D maps by exploiting geometric features and measuring surfaces. The most popular registration algorithm for point cloud registration is the iterative closest point (ICP) algorithm (Thrun 2003). In ICP, the corresponding closest points in different point clouds are associated and optimal rigid transformation required to minimize a mean-square error of separation between the associated points (Bsel, 1992) is iteratively found. The color attributes of the sampled point can be utilized in ICP progress to increase computational speed and provide higher accuracy. Anderson (2007) filtered the point set data based on hue before conducting traditional ICP. Houng et al. (2009) processed images to extract corresponding visual features that are used in registration process.

In this chapter, we examine the algorithms required for a mobile robot to generate a dimensionally accurate and complete map of an area without prior information about the area. We focus particularly on the techniques for registration of map segments taken from various vantage points. The chapter also describes a mobile robotic system with a color point cloud scanner and various algorithms required for accomplishing the mission of generating a complete and dimensionally accurate map of an area.

2. MOBILE MAPPING WITH COLOR POINT CLOUD SCANNERS

Color point clouds are created by synchronizing range sensors such as the LIDAR with video/still cameras. LIDAR devices discretely measure the distance between a light source and a reflection target at a high frequency. By changing the path of the light through mirrors and actuators, a point cloud of a 3D space is produced. A calibrated vision sensor maps the color

information to the sampled points. Installing such a scanning sensor on a mobile platform extends its range and enables mapping of large areas.

2.1 3D Color Scanner

The 3D color scanner used in this effort consists of a 2D LIDAR and two 1.3 megapixel high-frame rate video cameras installed on the LIDAR scanning plane. The LIDAR and the cameras moved such that the scan plane is rotated about an axis within the plane, thus generating 3D color point clouds. Figure 1 shows that the LIDAR consists of a rotating mirror which is driven about Y axis (degree of freedom: θ) and the scan plane is rotated about Z axis (φ: degree of freedom). The rotations are controlled by servomotors installed on the axes. The cameras are calibrated to be on the LIDAR scan plane and a forty-pixel wide image stripe is extracted from the cameras. The color information is then matched, in real-time, to the points ranged by the LIDAR. The relative distance between cameras and LIDAR is pre-configured and images are pre-aligned. The 2D LIDAR generates scans at a frequency of 38 Hz and the cameras provide imagery at 60 frames per second. Time synchronization establishes that the pixel color is mapped to each ranged point. Use of two cameras reduced occlusions due to the offset between the LIDAR mirror and the camera lens. All areas visible to the LIDAR are visible to one of the two cameras. The 2D range measurement along with the scanner rotation position (φ) is used to generate the coordinate in a spherical coordinate system, which is transformed to Cartesian system as necessary. Figure 1 also shows a picture of a compact version of the system.

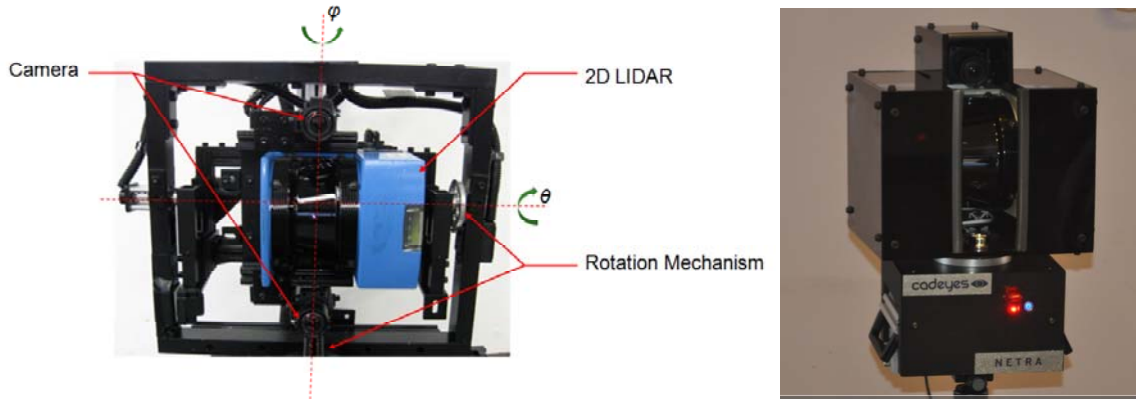


Figure 1 3D Scanning Devices built with 2D commercial scanners.

The 3D color scanner is mounted on a mobile vehicle for mapping large areas. This mobile mapping system generates color point cloud data. Figure 2 shows the mobile system with the scanner installed on top of the vehicle. The vehicle has no global positioning devices other than wheel encoders. Cameras and short-range inferred sensors enable observation of terrain conditions, collision avoidance and allow a remote operator to drive the vehicle. Map data and video feeds are transmitted using an on-board wireless communication system. This mapping system performs scans only when it is stopped. The vehicle can localize itself from the map observations and moves directly from one vantage point to the next and acquires additional map information. This system can generate color point cloud maps with 0.25° angular resolution in the vertical scanning direction with a coverage angle of 100°. In the rotation (ϕ) direction, the resolution is at 0.1° with coverage angle 300°. The map segment from one vantage point covers a maximum radius of 80 meters.

The data elements produced by the scanner are shown in Figure 3. Figure 3(a) shows the camera image taken from the vantage point depicting scene visible to the scanner. The 3D color point cloud generated at that vantage point is shown in figure 3(b). In this figure, the coordinate (x,y,z) and the color (r,g,b) for all the pixels known. The point density (spatial resolution of the point cloud) varies on the left and right sides of the color scan scene depending upon the distance between the scanned point and the scanner. The closest area to the scanner has the highest density of points. The scanner also records the optical reflection intensity of laser beam. The intensity information is combined with range measurement data and shown in figure 3(c). The object surface material, color and distance towards scanner cause variations in intensity data. Similarities between intensity point cloud and color point cloud can be observed between figure 3(b) and (c) on edges, doors, and windows.

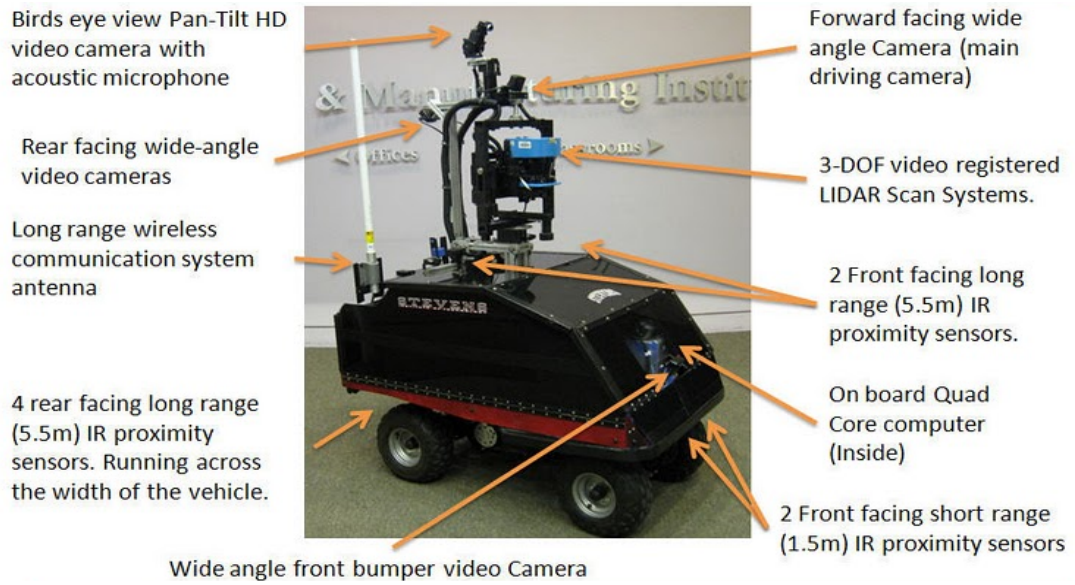


Figure 2 Mobility platform used for 3D color map construction in large area (Gebre et al., 2009).

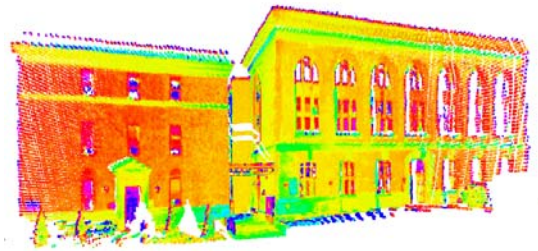
2.2. ALGORITHMS FOR COMPLETE MAPPING

An autonomous robot with the color point cloud scanner can reduce the surveying and map building cost and time. However, several methodologies for robust self-localization, map completeness evaluation, map based navigation and 3D map registration must evolve before a high degree of autonomy can be achieved.

A mapping robot deployed at initial start position must go through the four phases of the mapping processes as shown in Figure 4. The robot must be able to localize itself so it can navigate the scene. This can be accomplished by 2D SLAM (Simultaneous Localization and Mapping) techniques or other methods. Methodologies for establishing the map completeness and detection of occluded areas are necessary. Determination of the optimal vantage point for filling in the occluded areas and exploring unmapped areas is also a critical step. As the navigation is based on imprecise mapping and localization information, the map segment

registration based on 3D color point clouds is the last but crucial step in building the complete map of a given area. In this subsection, we discuss the algorithms that address each of these tasks.

(a) Image of an urban building. (b) Color point cloud map.



(c) Laser reflection intensity map.

Figure 3 High dimensional point cloud map segment taken from a single vantage point.

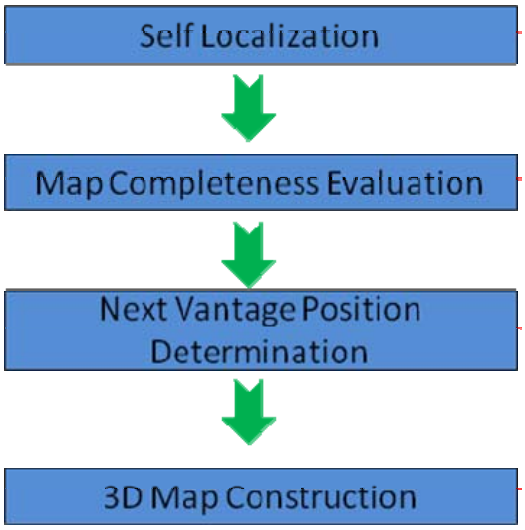


Figure 4 Map completeness orientated robotic mapping process.

2.2.1 Robot Self-localization

The self-localization problem requires mobile mapping robot to determine its location in an unknown environment. Localization is critical because robot cannot effectively navigate to the next waypoint without the location information. Map registrations require location and pose estimates. Usually robot is equipped with multiple position and orientation sensors like GPS, Inertial Measurement Unit (IMU), odometer, and wheel encoders to measure real-time pose and position. Multiple position and location sensors return robot position information with certain level of error due to reasons like sensor precision, GPS signal noise and errors, sensor drift for IMU and inaccurate measurements from other sensors.

The main challenge for robot localization is to escape location sensor noise, drift errors, and constantly provide accurate location and position reference for the robot. Probabilistic self-localization techniques based on maximum-likelihood estimation have been applied to address this problem. These techniques assume that the noise of position sensor follows certain probabilistic distribution, which can be described mathematically. They also assume that two subsequent map results are highly comparable to each other and several landmarks can be quickly identified. Therefore, accurate relative position and location can be solved by comparing current map with a previous map in short time intervals, and probabilistically maximizing similarity between two maps (Olson, 2000). Map could be generated by different sensors like stereo cameras, sonar or laser range finders. Landmarks extracted from maps are commonly applied in the self-localization process to reduce computation cost. Whyte and Bailey (2006) utilize the relative localization results between two neighbor vantage points to merge the two maps.

A two-step process, termed as Simultaneous Localization and Mapping (SLAM), typically localizes the robot. The robot position is established from multiple but imprecise sensor measurements and comparison of landmarks in the scene. The position sensor data is improved using sensor fusion techniques by Spletzer (2003). Location information is estimated based on previous location, driving command information and current sensor measurements. In SLAM, probabilistic methods are applied to reduce sensor noise effects. Extended Kalman filter and particle filters and noise models improve the location estimates (Montemerlo, et al. 2003). The SLAM solution has been expanded into 3D space with a six degree of freedom (6DOF) SLAM which applies sensor measurement and robot kinematics models (Nücher, 2005). Landmark extraction and map comparison entail the major computation effort during the SLAM progress. Real-time SLAM has been demonstrated with stereovision sensors (Davison, 2003).

The SLAM technique simultaneously considers the localization and mapping mission (Thrun, et al., 2000). The SLAM problem can be described by a joint posterior:

$$P(x_t, m \mid z_{0:t}, u_{0:t}, x_0) \qquad (1)$$

Where, $\boldsymbol{x_t}$ is the state vector representing the robot location and orientation, $\boldsymbol{m_i}$ is the vector representing the i^{th} landmark location, $\boldsymbol{z_{it}}$ is the robot mapping measurement about i^{th} landmark at time t, and $\boldsymbol{u_t}$ is the control vector applied at t-1 time to drive robot to state $\boldsymbol{x_t}$ at time t.

The SLAM problem requires that equation (1) be solved for the time, t, and the latest robot state vector $\boldsymbol{x_t}$ be computed. Solving the joint posterior from, 0-t requires an observation

model and a motion model based on Bayes Theorem. (Whyte and Bailey, 2006). The observation model determines the probabilistic distribution of observation z_t with known vehicle state and landmarks location as:

$$P(z_t \mid x_t, m) \tag{2}$$

The robot motion model describes probability on state transition of robot state vector, x_t with known previous state x_{t-1} and control input u_t

$$P(x_t \mid x_{t-1}, u_t) \tag{3}$$

The transition of state vector is assumed as a Markov process, implying that the next robot state x_t can only be determined on previous state x_{t-1} and latest control input u_t and not the history of states. The state of robot is independent of both observations and landmarks. Equation (1) can be recursively solved in a Prediction (time update) and Correction (Measurement update) form.

Prediction:

$$P(x_t, m \mid z_{0:t-1}, u_{0:t}, x_0) = \int P(x_t \mid x_{t-1}, u_t) P(x_{t-1}, m \mid z_{0:t-1}, u_{0:t-1}, x_0) dx_{t-1} \tag{4}$$

Correction:

$$P(x_t, m \mid z_{0:k}, u_{0:t}, x_0) = \frac{P(z_t \mid x_t, m) P(x_t, m \mid z_{0:t-1}, u_{0:t}, x_0)}{P(z_t \mid z_{0:t-1}, u_{0:t})} \tag{5}$$

Equation (4) and equation (5) recursively solve latest robot state joint posterior. Robot state can be predicted from the motion model $P(x_t \mid x_{t-1}, u_t)$ and control input at time *t*. The observation model $P(z_t \mid x_t, m)$ is applied to correct state prediction with observation and mapping at time *t*.

In order to find solutions to the SLAM problem, proper practical descriptions about motion and observation model in equation (2) and equation (3) should be provided with reliability and efficiency. Extended Gaussian Filter (EKF) is applied to represent these models on state-space model with additive Gaussian noise (Welch and Bishop, 1995). The EKF based SLAM simplifies motion model as:

$$x_t = f(x_{t-1}, u_t) + w_t \tag{6}$$

$f(x_{t-1}, u_t)$ is the robot kinematics model and $\mathbf{w_t}$ is the additive uncorrelated Gaussian disturbances with zero mean and covariance Q_t. The observation model can be described as:

$$z_t = h(x_t, m) + v_t \tag{7}$$

In which, $h(x_t, m)$ is the observation geometry description and $\boldsymbol{v}_t$ is the additive uncorrelated Gaussian disturbance with zero mean and covariance R_t. Eqs. (6) and (7) can be applied to the SLAM prediction and correction. In EKF-SLAM process, the mean and covariance of both motion model and observed motion should be updated at every time *t*. Other probabilistic methods such as Particle Filter (PF) (Montemerlo et al., 2003) and Graph Filter (GF) are used to solve the SLAM problem. A typical SLAM method is implemented on 2D space,

however, SLAM in 3D space with 6 Degree of Freedom (6DOF) on robot kinematics have been implemented by expanding landmarks state, motion model and observation model into 3D space (Nücher, 2005).

2.2.2 Map completeness evaluation

The map completeness problem can be addressed with several methodologies including grid occupancy, obstacle recognition and object view completion detection. The completeness of map is calculated by occupancy grid map (Thrun, 2003), which entails projecting the acquired map on an occupancy grid and calculating the occupancy level. Possible mapping area is determined based on the contour of the objects and separating the map into areas that can be potentially mapped or impossible to map (Oh et al., 2004). Terrains are extracted from current incomplete map for possible paths for navigation. The map evaluation also returns possible explorative area that is accessible to the mobile robot but not mapped. If map completeness is the most important factor for the mission, algorithms that evaluate latest exploration status after every scan may require assessment of the complete map and not just the current map segment. There are many techniques to evaluate the completeness of mapping, namely, grid based occupancy map (Thrun, 2003), network/graph, cell based map (Zelinsky, 1994) and template based completeness evaluation (Oh et al, 2004).

The occupancy grid map is one of the most commonly used methods to determine map completeness. Area of interest is gridded and acquired maps from different vantage position are transferred into or projected onto the grid. Grid is marked as occupied when data exists on this grid, every grid should be represented with certain level of occupancy, which is computed by density of point cloud map on this grid. Map can be assumed as complete all the mapped objects form self-closed contours or closed contours with the boundaries of the mapped area.

A major challenge in map completeness evaluation is deciding whether an area can be mapped. For example, when mapping robot is performing indoor exploration, space behind wall of the hallway may not be accessible. Contours extracted from latest global map may be used to determine possible navigation paths. Possible mapping area exists for contours with gaps. Ascertaining that the gaps in map contours are indeed traversable paths requires discerning traversable pathways in the map.

2.2.3 Map based navigation

Determination of the next vantage point may depend upon several criteria: best view, coverage of unmapped areas, areas of overlap with current map, localization, accessibility and traveling costs. Two steps are required for determination of the next vantage position. The first step is the generation of candidate positions and second step is the selection of optimal vantage point from the list. The candidate vantage positions can be created based on frontier exploration algorithm (Basilico and Amigoni, 2009) considering obstacles, position and terrain conditions. The vantage position is selected between candidate positions that have the best view coverage and shortest traveling cost. Next vantage point should be decided based on the best view to fill occluded regions and cover as much new area as possible. Frontier based exploration algorithm provides vantage point candidates for the best view point, these candidate points are evaluated to determine best vantage point for next mapping.

Computing vantage position for mapping based on previous vantage positions and incomplete map is known as the Next Best View (NBV) problem (Yamauchi, 1997; Basilico and Amigoni, 2009) NBV algorithms navigate robot to acquire maximum uncovered area. A certain level overlapped area ensures that the robot has enough landmarks to navigate between the current and the next best view vantage point. Frontier based algorithm can be applied to provide candidate positions for the next best view point. Based on the regions on the boundary between mapped and unmapped space, the frontier can be extracted. Considering the range for mapping sensor constraints, next mapping position on the frontier can then be generated. Current frontier should be evaluated in occupancy grid map so that the frontier grid positions that cover more unoccupied can be selected to accelerate the coverage of the area. These candidate points can be evaluated based on the criteria for the exploration and time and power requirements for reaching the vantage point.

The map data acquired from various vantage points must be registered into global map space using various registration algorithms. Although this section describes the various algorithms required for complete map generation, the focus of this chapter remains on the registration aspect of the mapping exploration.

3. ALGORITHMS FOR REGISTERING MAP SEGMENTS

Three-dimensional point cloud segments acquired from different locations have to be combined together as complete large-scale map. Position and orientation information required for registration can be provided directly by mobile platform sensors such as GPS and IMU (Thrun 1993). In most cases, position information acquired from sensor is reasonably accurate. However, the orientation information is costly and relatively imprecise because orientation sensor measurement can be affected by external disturbances like magnetic field variations and sensor integration drift with time. Position and orientation information can also be provided by indirect techniques based on both rough position sensor measurement and common geometric feature identification. Figure 5 shows two maps generated from separate vantage points. The left map on the top row shows map generated with robot facing towards one side of the building, the right map shows the map generated from the second vantage point. The bottom figure shows the map data from the first vantage point registered into the coordinate system of the second location. Registering the two segments produces the complete map of the façade of the building.

Comparing with the SLAM algorithm, map registration techniques focus on generating accurate map details rather than localization of the robot in a global coordinate system (Arun, 1987; Bsel, 1992; Lorusso, 1995; Rusinkiewucz, 2001). Discrete range points received from color point cloud sensor contain detailed spatial information about the environment. Different techniques exist for merging such point clouds together by exploiting geometric features and measuring surfaces. Map registration techniques such as Iterative Closest Point (ICP) algorithm proposed by Bsel (1992) has been applied to stitch two neighbor 3D point cloud maps together into one map based on their common coverage area. Upon convergence, ICP algorithm terminates at a minimum. Several algorithms are in existence for calculating the minimum average distance between two point clouds. Singular Value Decomposition (SVD) method by Arun (1987), eigen-system methods that exploit the orthonormal properties of the rotation matrices, and unit and dual quaternion techniques were adopted in ICP process. Quaternion based algorithms have been used in ICP for map fusion by Bsel (1992), SVD based algorithms are widely used in ICP and 6DOF SLAM (Arun 1987, Nucher, 2005, Joung et al., 2009) as they

are robust to reach local minimum and easy to implement. Several variants of ICP are reported by Rusinkiewucz (2001) to increase the speed and precision. 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 by Robertson (2002) to accelerate computation speed. Point to plane registration method (Lorusso, 1995, Rusinkiewucz, 2001, Salvi et al., 2007) accelerates the ICP iteration and convergence.

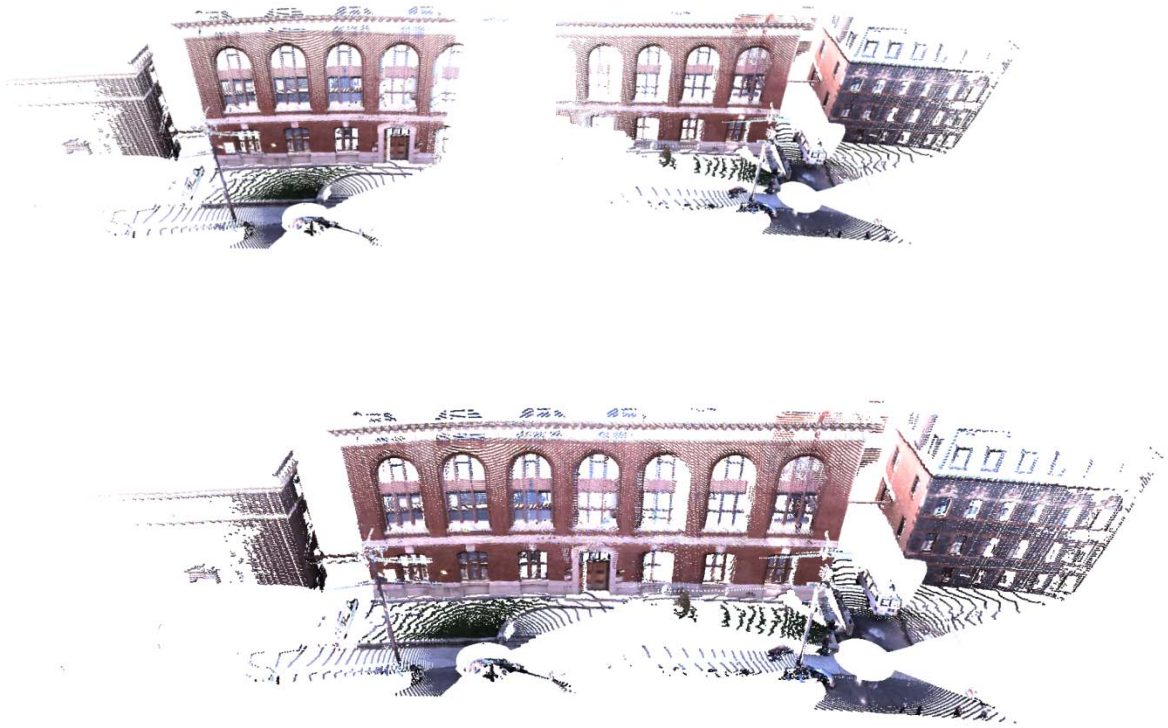


Figure 5 Map segments generated from two vantage points (Top) and registered map (Bottom).

Other techniques include the point signature method by Chua (1997), which uses signature points to describe curvature of point cloud and matches corresponding signature points during the registration process. Spin image based methods compute 2D spin image to represent surface characterization and solve the registration problem by finding best correspondence between two different scan spin images (Johnson 1997). Other methods like principle component analysis (Chung and Lee, 1998) and algebraic surface model (Tarel et al., 1998) are based on the point cloud surface geometrical features. The normal vector distribution can be translated into an orientation histogram in an Extended Gaussian Image (EGI) (Makadia and Daniilidis, 2006). Rigid motion required to register two point clouds is solved from the cross covariance function (Chibunichev and Vilizhev, 2008) of the two EGI images. Rigid motion could also be solve in Fourier domain by computing Discrete Fourier Transform on Rotation Group on SO(3) (SOFT) (Joistekecm and Ricjnirem, 2008).

Registration of color point clouds has been considered (Ferbabdez, et al., 2007; Druon, 2007; Newman et al., 2006; Anderson, 2006, 2007). By applying proper calibration on the hybrid sensor system (Joung et al., 2009; Newman, Cole, Ho, 2006), range measurement and visual information can be integrated together to construct a visually accurate representation of the scene. Color mapped 3D data was used in map registration by weighted red, green, blue data. The corresponding point search during the ICP is conducted on both the coordinate and color data (Johnson, Kang, 1997). Hue filters were also used to constrain the closest point search in every ICP iteration (Druson, 2007). Color data can be used to estimate initial alignment of pair wise scans using Scale Invariant Feature Transform (SIFT) techniques. Color attributes transferred in YIQ color model can also be weighted to construct new variant together with range information for ICP fine registration. Depth-interpolated Image Feature (DIFT) algorithm solves corresponding points between two images and registers color point clouds based on extracted correspondences (Anderson, Lilienthal, 2010).

In this chapter, we introduce 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 fourth dimension selected is the hue, representing the intrinsic color values of the pixel. While achieving the effect of a hue-based filter, hue-association reduces the nearest neighbor search burden considerably (Men and Pochiraju, 2010). The remaining sections of the paper describe the approach and the performance of the algorithm under several hue distributions in the scene.

4. Hue-Assisted Iterative Closest Point (HICP) Algorithm

The primary hypothesis of this algorithm is that the hue value can be applied to increase the accuracy of point association and accelerate the registration process. The major time and computation cost during ICP is finding the correct points pairs. Closest spatial distance is typically applied in 3D ICP method. The distance value in 3D space can be expanded into 4D space by adding weighted hue value as the 4^{th} dimension. By integrating hue value into the closest point search, accuracy of point association can then be improved.

4.1 Hue invariance with vantage point

Hue value remains consistent about the same point between images taken from two vantage points, while the color values represented in red, green and blue quantities usually differ because of variation in light conditions. In order to apply color to improve the association process, lighting effect should be removed. Color raw data are transformed into representation of separate chroma, lightness and brightness value. Figure 6 shows two camera images of different angles of a color palette on a Rubik's cube, four colors are used on the same surface. Figure 6 also shows the color pixels with the background and black frame removed. Histograms showing the red, green and blue value in RGB space for all the pixels are shown in figure 7. In the RGB histogram, R, G, and B distributions of the image vary considerably with the vantage point. When the RGB color space is transformed into HSL space and histograms of hue, lightness and saturation are plotted figure 8, the hue values remain relatively invariant with the position of the camera. Therefore, hue value of the pixel, taken from the Hue-Saturation-Lightness (HSL) model, is used as the fourth dimension in the point association process. In Figure 9, the hue range map

of color range map in Figure 3(b) is shown. Hue values are normalized between 0 and 1. The hue distribution is typically similar to the color distribution in Figure 3(b).

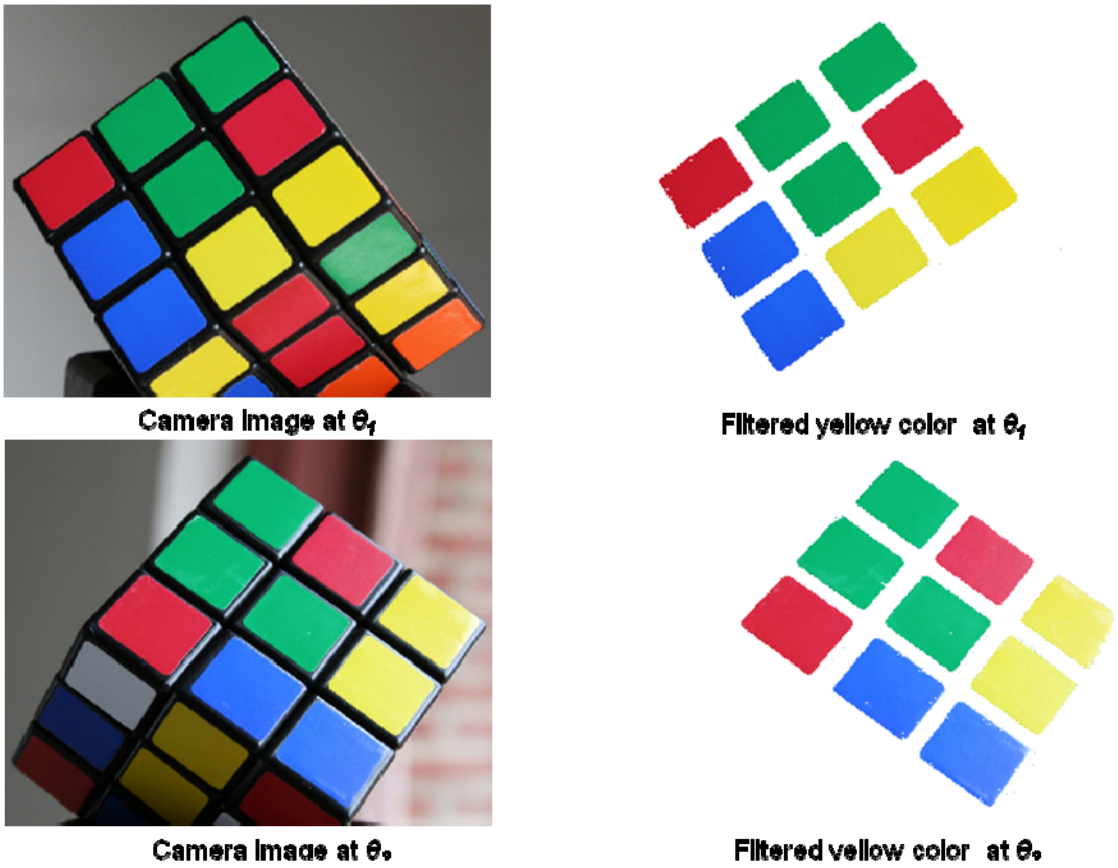


Figure 6 Rubik's cube camera images take from two vantage points.

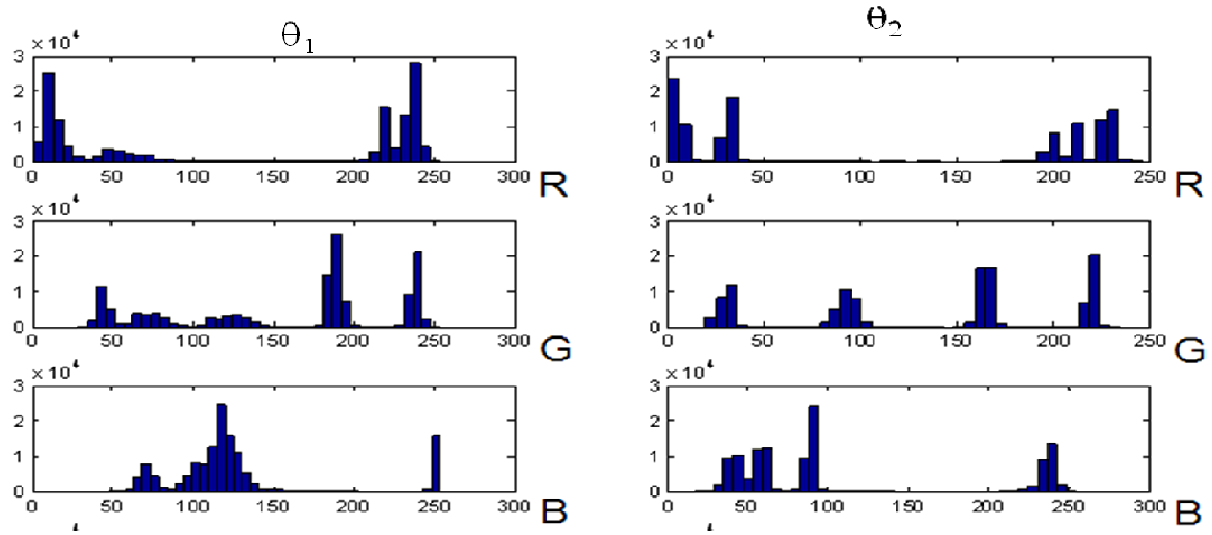


Figure 7 RGB distribution varies with camera positions.

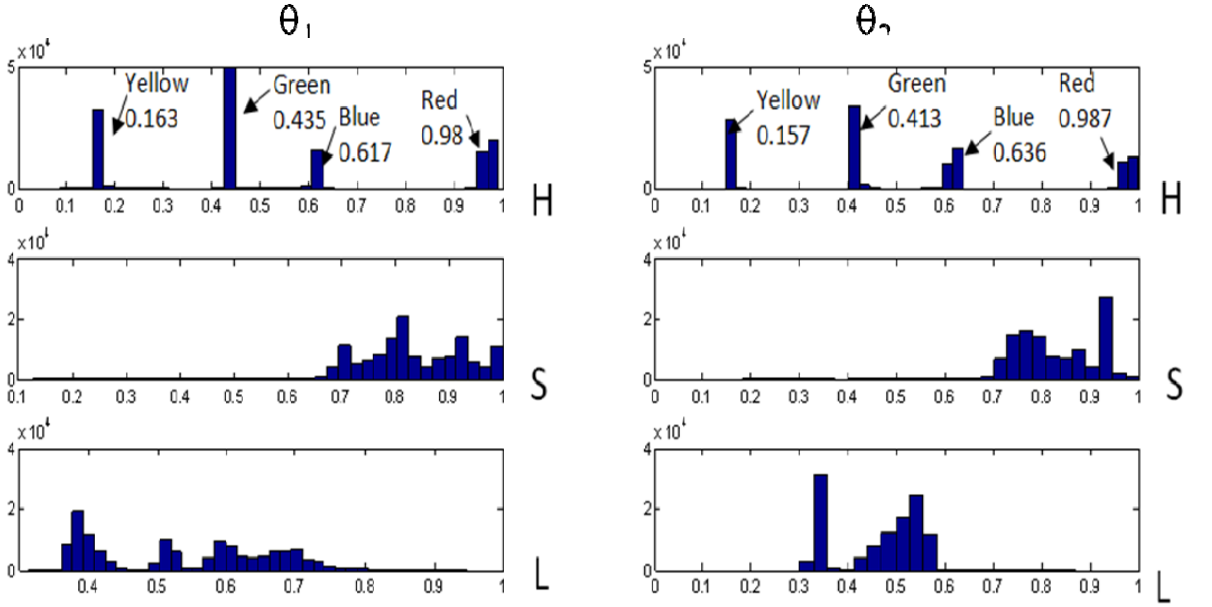


Figure 8 HSL distribution: hue remains invariant.

Figure 9: Hue map of the scene shown in Figure 3.

4.2 Construction of a weighted 4D search space

Both hue and range value have to be combined together in the HICP variant as $\{x_o, y_o, z_o, h_w\}$ for point association. x_o, y_o, z_o are the original coordinate values with distance units and h_w is the weighted hue value. Hue values are normalized to a 0-1 range and must be weighted during the closest point search in the four-dimensional space. In order to normalize the coordinates, we find the bounding box for each map segment and the coordinate space is rescaled to a 0-1 range. The normalized variant for point association is $\{x, y. z, h_w\}$, where $x=x_o/r_x$, $y=y_o/r_y$, $z=z_o/r_z$. r_x, r_y, r_z are the dimensions of the bounding box in x, y, z directions.

The weight value for the hue dimension should be properly selected for point association. Since both range and hue value are normalized from 0 to 1. Weight for hue represents its influence in the nearest neighbor search process. Low weight biases the point association towards the range data and a high weight towards the hue values. Small weight values for the hue correspond to the traditional 3D-ICP. Hue weight should be selected between 10% and 35% for accurate point association. Error in 4D ICP will be evaluated by the average mean square root distance of normalized associated point pairs.

4.3 k-d Tree Based Point Cloud Association

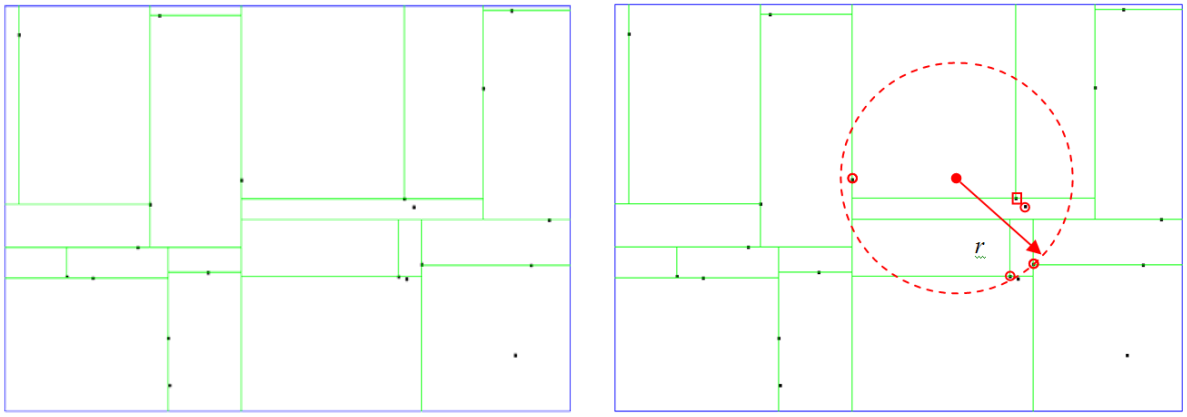
In 3D ICP algorithm, corresponding points are searched according to the closest distance rule. This may cause incorrect matching during single iteration loop as Figure 10. Dashed line circle illustrates range based nearest point association results, in which all points in data set look for nearest neighbor in 3D space. It takes more than one iteration to pair correct nearest neighbor points for given data points set. Grey circle denotes the HICP nearest point search that also uses the correct hue property in finding the best neighbor in the model. Depending on the correct color information, corresponding point can locked with less iteration.

Figure 10 Point association based on nearest distance (dotted) and nearest distance and hue(solid)

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 figure 11(a). 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 11(b) 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.



(a) k-d tree construction in 2D space. *(b) 2D space nearest neighbor search in k-d tree.*

Figure 11 k-d tree construction and nearest neighbor search in 2D space.

In 3D closest point search, the distance between 2 points between 2 point clouds is:

$$r_{ij} = \sqrt{(m_{ix} - d_{jx})^2 + (m_{iy} - d_{jy})^2 + (m_{iz} - d_{jz})^2} \tag{8}$$

In which, $\boldsymbol{d_i}\{d_{ix},d_{iy},d_{iz}\}$ and $\boldsymbol{m_j}\{m_{jx},m_{jy},m_{jz}\}$ are point spatial coordinates in data and model range map respectively.

In 4D space, the 4th dimension for each point should be weighed hue value d_{hw} or $m_{hw.}$ The spatial value of points should be normalized by 3D search radius r_{ij} as mentioned in section 4.1. In order to accomplish closest point search in 4D space, the distance between two normalized points $\boldsymbol{d_i}\{d_{ix},d_{iy},d_{iz},,h_{ihw}\}$ and $\boldsymbol{m_j}\{m_{jx},m_{jy},m_{jz},m_{jhw}\}$ should be:

$$r_{ij}^{'} = \sqrt{(m_{ix} - d_{jx})^2 + (m_{iy} - d_{jy})^2 + (m_{iz} - d_{jz})^2 + (m_{ihw} - d_{jhw})^2} \tag{9}$$

or

$$r_{ij}^{'} = \sqrt{r_{ij}^2 + \Delta h_{ijw}^{\ 2}} \tag{10}$$

In the ICP process, search radius effects the computation time and final result. A constant search radius is applied for all iteration loops. Once the search radius is large, too many points will be included as candidates and increases the computational burden. Candidate points cloud be missed if search radius is too small. The search radius is determined by the density of point cloud. In 4D k-d tree search, the search radius comprises of two parts -- a distance part and weighted hue part as seen in equation (9). The search range for 3D distance is selected such that it ensures about 50 candidate points within search radius. As hue value is not transformed at iteration, hue search is analogous to filtering. If the weight for hue is high, k-d tree search will bias toward hue dimension. Therefore, appropriate hue weighting ensures that spatial search dominates over hue filtering.

The ICP algorithm iteratively converges at minimum error, which is described by mean square root of the spatial distance between paired points. At each iteration, a rigid transformation matrix is computed so that the distance error metric between the associated points is minimized. Data point cloud is transformed into the model space using the computed transformation matrix. This iteration continues until error metric converges.

Use of hue as a fourth dimension is significant in those instances where the coordinate based matching results in a 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 may result in unique registration of the points.

The color assisted ICP algorithm in this paper can be described as follows.

1. Estimate the initial transformation matrix ***R*** and ***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 $\boldsymbol{\varepsilon} > preset\ error$

Use ***R*** and ***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 $\mathbf{d_i}$ and $\mathbf{m_j}$ as $\{\mathbf{d_k}, \mathbf{m_k}\}$;

k++;

End If

End For

5. Acquire paired point cloud $\boldsymbol{D_p}$ and $\boldsymbol{M_p}$, contain N Points, calculate normalized mean square root distance $\boldsymbol{\varepsilon}$ *as* error,

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

6. Construct orthonormality matrix ***H*** (Equation14) and solve rigid rotation ***R*** and translation ***T*** (Equation15, 16) for next iteration;

End While

4.4 Solving Rigid Transformation

ICP algorithm is an iteration process to calculate rigid transformation matrix based on associated point clouds. $m_i = \{m_{ix}, m_{iy}, m_{iz}\}$ represent the coordinates of the i^{th} point in the model point cloud and $d_j = \{d_{jx}, d_{jy}, d_{jz}\}$ is the j^{th} point in data point cloud. Rigid transformation (R) that minimizes the error measure $E(R,T)$ shown in Equation (11) is determined.

$$E(R,T) = \frac{1}{N}\sum_{i=1}^{N} \left\| m_i - (Rd_i + T) \right\| \tag{11}$$

A centroid for the associated points is calculated as the first step (Equation12) and associated points are translated into centroid relative coordinates (Equation13). Orthonomal matrix of associated points can then be constructed as shown in Equation14. Rotation $\boldsymbol{R}$ and translation $\boldsymbol{T}$ are decoupled based on the gravity center of associated points. Using Singular Value Decomposition (SVD) methods, $\boldsymbol{R}$ can be determined as shown in Equation15. Translation $\boldsymbol{T}$ is computed using Equation16.

$$\overline{m} = \frac{1}{N}\sum_{i=1}^{N} m_i \quad , \quad \overline{d} = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{12}$$

In which, $\overline{m} = \{\overline{m}_x, \overline{m}_y, \overline{m}_z\}$ and $\overline{d} = \{\overline{d}_x, \overline{d}_y, \overline{d}_z\}$ are the center points of associated points in model and data point clouds. N is the amount of point pairs.

The coordinated of associated point in center point relative space should be

$$m_i' = m_i - \overline{m} \quad , \quad d_i' = d_i - \overline{d} \tag{13}$$

In which, $m_i' = \{m_{ix}', m_{iy}', m_{iz}'\}$ and $d_i' = \{d_{ix}', d_{iy}', d_{iz}'\}$ are the i^{th} associated point with center relative coordinates.

The orthonormality matrix H can be constructed based on $m' \{m_i', i=1 \ldots N\}$ and $d' \{d_i', i=1 \ldots N\}$.

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

Where

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

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

$$S_{zz} = \sum_{i=1}^{N} m'_{iz}\, d'_{iz}$$

$$S_{xy} = \sum_{i=1}^{N} m'_{ix} \, d'_{iy} \tag{14}$$

Singular value decomposition is performed on constructed ***H*** matrix for optimal rotation **R**

$$H = U \Lambda V^T \tag{15}$$

Where optimal rotation $R = VU^T$.

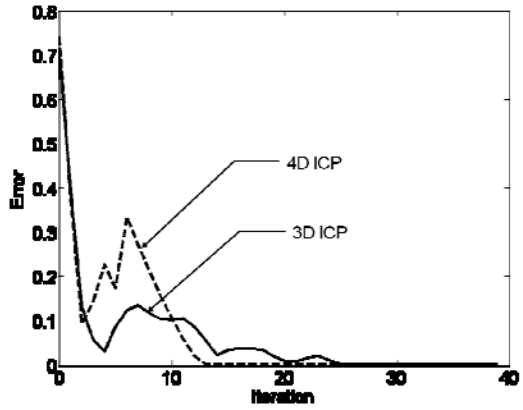
The translation ***T*** can be calculated as

$$T = \overline{m}^T - R\overline{d}^T \tag{16}$$

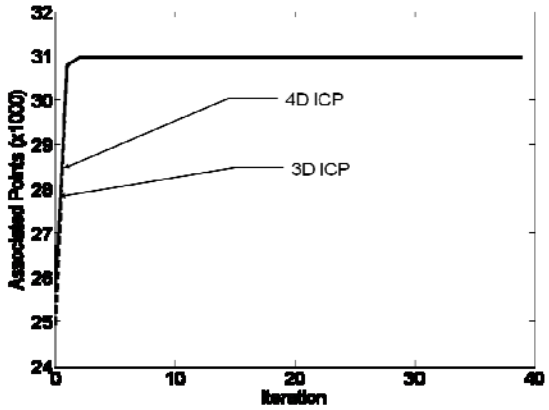
4.5 Convergence Criteria

Convergence and stopping criterion for the HICP iteration are discussed in this sub-section. An association stability criterioa is introduced as the one of the convergence criteria. Association stability, denoted as *S*, is defined as the for the number of points which changed their paired point in any iteration. If a point comes into association or changes its nearest neighbor, S is incremented. Large value of *S* signifies that point association has not stabilized. HICP iteration is terminated when *S* vanishes and the distance error converges.

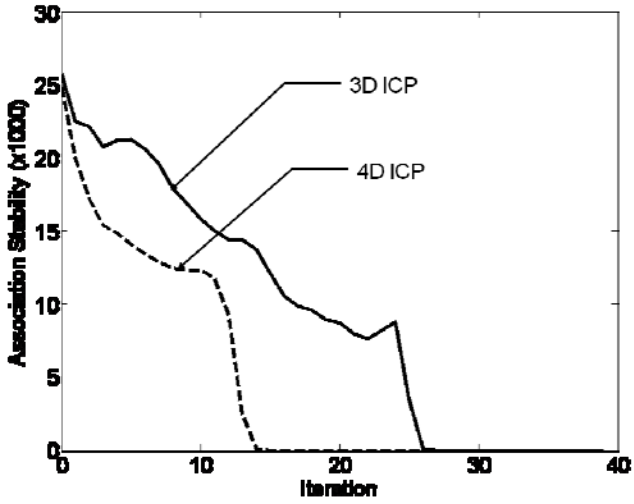
A pair wised color point HICP registration was accomplished based on above criteria. Model point cloud contained 122,409 points with color attributes. Data point cloud is extracted from model point cloud with a known rotation ($\theta_z=5^o$). The HICP registration process is compared with 3D ICP, error as shown in Figure 12(a). The associated point number reaches maximum after the 5^{th} iteration (Figure 12(b)), but error has not converged. From Figure 12(c) the association stability (S) reaches 0 after 15^{th} and 26^{th} iteration for HICP and 3D ICP respectively. Error and rigid transformation are shown in Figure 12(a) and Figure 13. The known transformation ($\theta_z=5^o$) is recovered by the HICP and ICP algorithms.



(a) Comparison of error convergence



(b) Association number convergence



(c) Association stability convergence

Figure 12 Building color range map registration comparison between H-ICP and Range ICP algorithm.

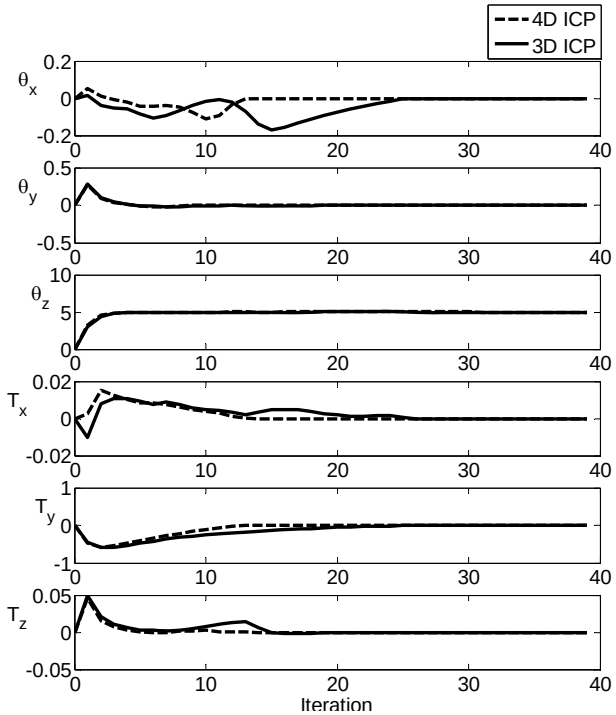


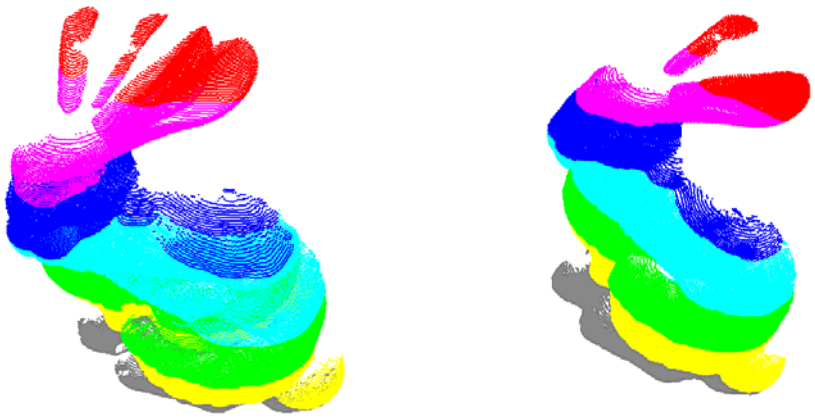
Figure 13 Convergence of translation and rotation estimates during registration

5. MAP REGISTRATION WITH ICP AND HICP

The hue distribution or the color of the model is generally independent of the geometry. If the entire body is painted with a color of a single hue, HICP is as effective as the traditional ICP. In this section, we describe the performance of the algorithm under various hue distribution scenarios. The Stanford bunny point cloud is considered as the benchmark data set. In HSL color space, hue value varies from 0- 360. The color correspondence between RGB and hue is given in Table 1.

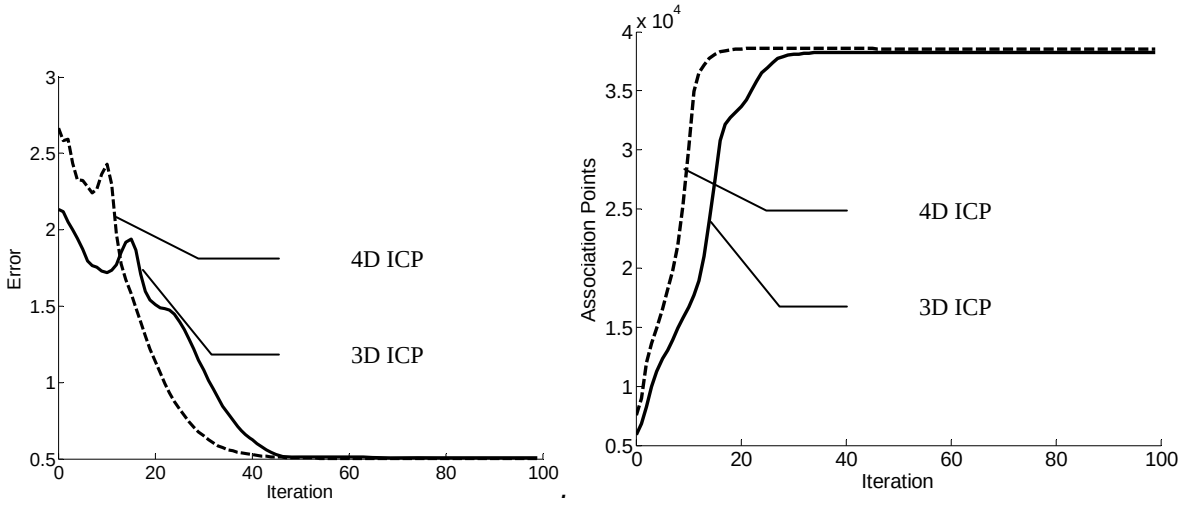
Color	R	G	B	Hue
Gray	128	128	128	0
Yellow	255	255	0	60
Green	0	255	0	120
Cyan	0	255	255	180
Blue	0	0	255	240
Magenta	255	0	255	300
Red	255	0	0	360

Table 1: Hue and RGB values for several common colors.



(a) *(b)*

Figure 14: Registration of point clouds with uniformly distributed hues(a) Stanford Bunny point clouds with hue distributed as seven distinct stripes (b) Registered color point cloud.



(a) Mean square error comparison. *(b) Associated point number comparison.*

Figure 15 Registration comparisons between H-ICP and Range ICP algorithm.

5.1 Environments with Fixed Hue Distributions

For the first experiment, we textured the Stanford bunny point cloud model as shown in Figure 14(a). In this model, the hue varies from 0 to 360 with from bottom to top at Z direction in seven segments. Figure 14(b) also shows the initial registration of the model and data point clouds used for this simulation.

The HICP registration progress is shown in Figure 15(a) and Figure 15(b). Figure 15(a) shows the mean square error during the ICP process and Figure15 (b) shows the number of points associated during iteration loops. Both data and model point cloud after registration is shown in Figure 14(b). The hue-assisted ICP registers the point and data clouds faster than the traditional coordinate based ICP.

5.2 Continuously Varied Hue along One Dimension

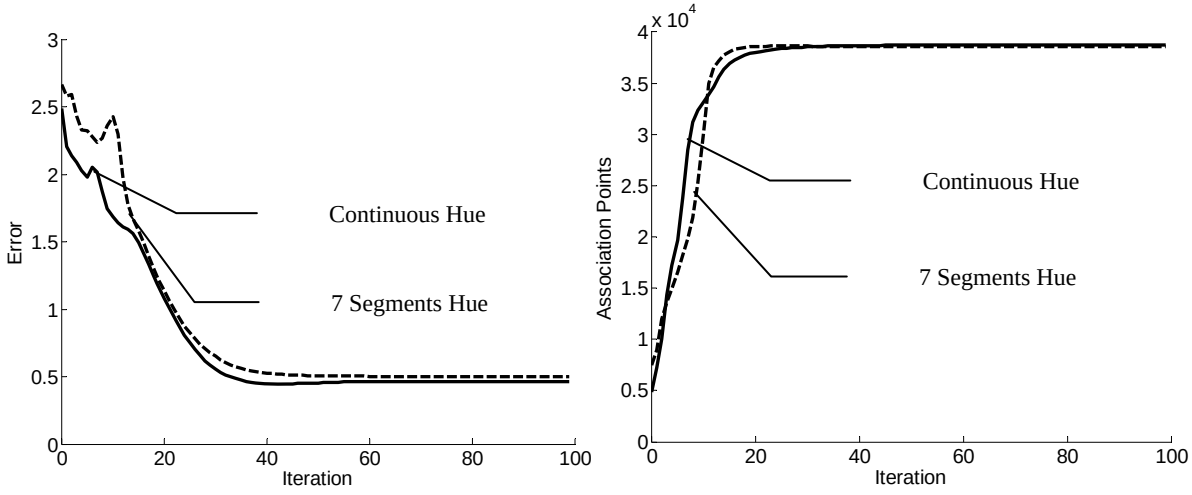
In the second simulation, a continuous hue distribution is assigned to the bunny model. The hue value is varied from 0 to 360, smoothly, along the z (vertical) direction. The resultant model and data clouds are shown in Figure 16 (a), (b). Saturation and lightness value have been set as constant at every point inside dataset. Hue value can be calculated by equation (17).

$$h = 360\frac{z_i - z_{min}}{z_{max} - z_{min}} \tag{17}$$

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. (c) Merged View.
Figure 16 Bunny model with continuous hue variation in one axis.

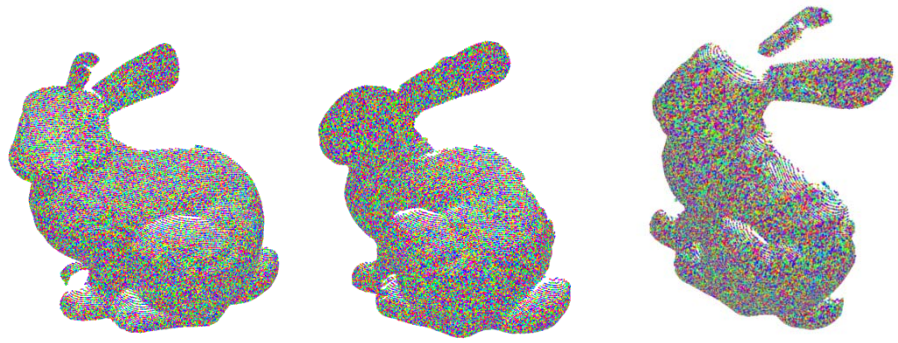
Continuous hue distribution on point cloud data is registered together (Figure 16 (c)) and the results are shown in Figure 17. A comparison of model performance on discrete and continuous distribution of hue on the same model shows the expected acceleration in performance due to uniform distribution of hue on the model.



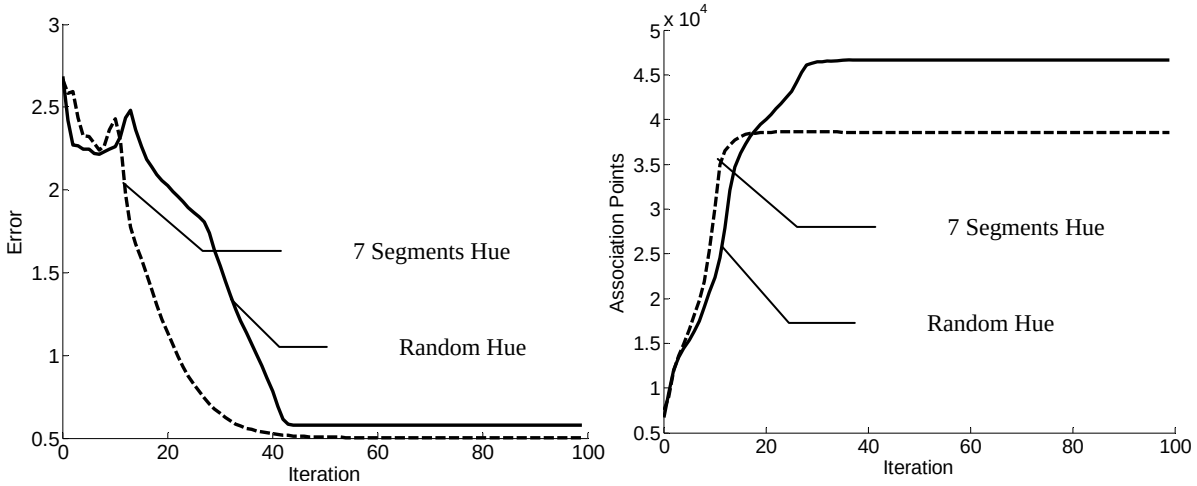
(a) Mean square error comparison. (b) Associated point number comparison.
Figure 17 Registration comparisons between 7 segment hue model and continuous hue model.

5.3 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 18 (a, b). The merged cloud point cloud after registration is shown in Figure 18(c). Figure 19 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.



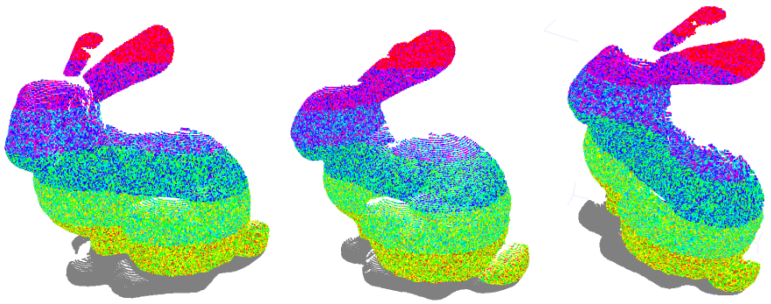
(a) Data point cloud (b) Model point cloud (c) Merged View
Figure 18: Bunny point cloud with randomized hue distribution



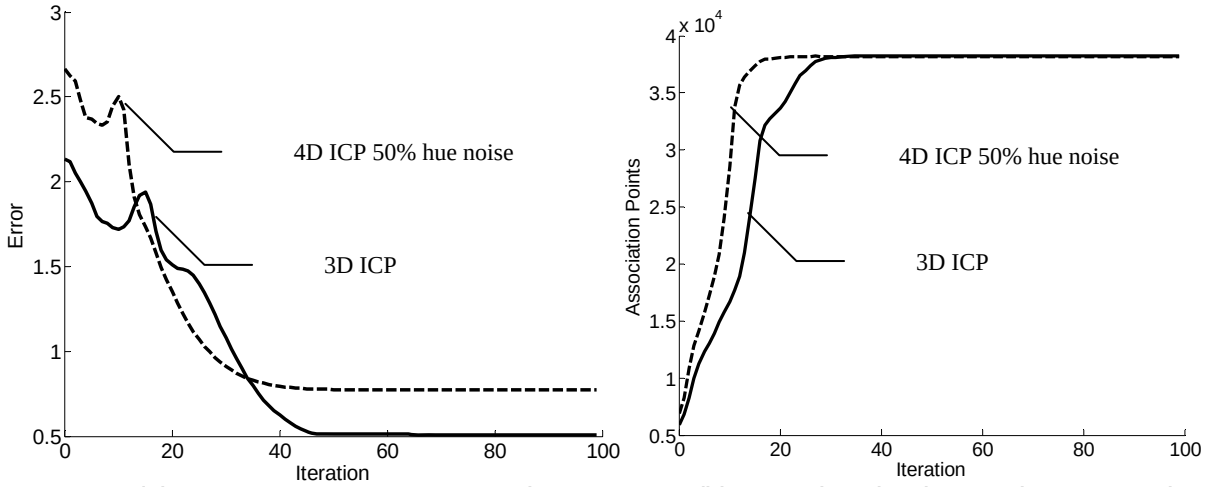
(a) Mean square error comparison (b) Associated point number comparison
Figure 19 Comparison between discrete and random hue distribution case

5.4 Effect of Imaging 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 (Gebre et al., 2009). 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 20(a, b). The merged color point cloud is shown as Figure 20(c).



(a) Data point cloud. (b) Model point cloud. (c) Merged View.
Figure 20 Hue mapped with noise.



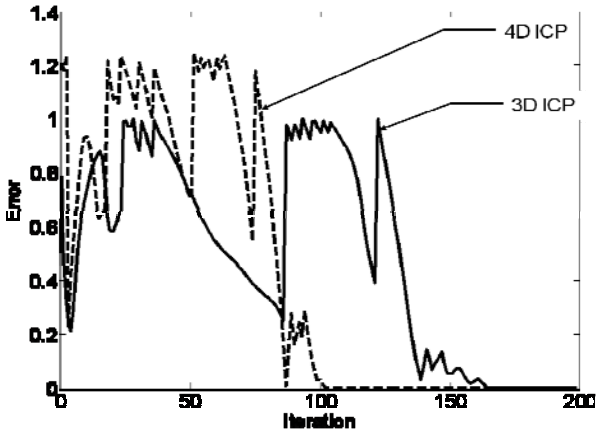
(a) Mean square error comparison. (b) Associated point number comparison.
Figure 21 Comparison between color ICP in and range ICP for noisy hue case

Hue assisted color ICP matching result in camera noise color point cloud is compared with range ICP matching performance. From Figure 21, noise in hue decreases the matching accuracy and reduces the iteration efficiency. Two groups of cloud point clouds are selected to evaluate the performance of 4D ICP algorithm compare with typical 3D ICP. A known transformation point cloud data pair was generated by transforming model point cloud at 6DOF to compare the convergence speed and registration accuracy as the rigid transformation is already known. Outdoor large scale area pair wised registration includes 8 pair wised data registration.

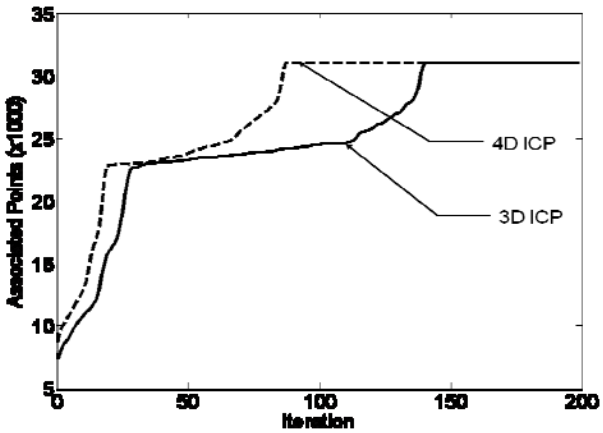
5.5 REGISTRATION WITH SIX-DOF ROTATION.

In this experiment, registration speed between 3D ICP and HICP are compared using data and model point clouds with known (and exact) registration transformation. Both HICP algorithm and 3D ICP algorithm have been applied on a building data set(Gebre et al., 2009). The data point cloud is taken from a view position that is 10° off in Y and Z axis from the model point cloud. Translation between the point clouds is known to be 2.46, 2.612 and 0.347 along the X, Y, and Z respectively. Same parameters for registrations are selected to be the same as in the previous 1-DOF registration. Error comparison and associated point number comparison are shown in Figure 22(a) and (b). Association stability is shown in Figure 22(c). The evolution of rigid transformation during ICP is shown in Figure 23. The HICP completes

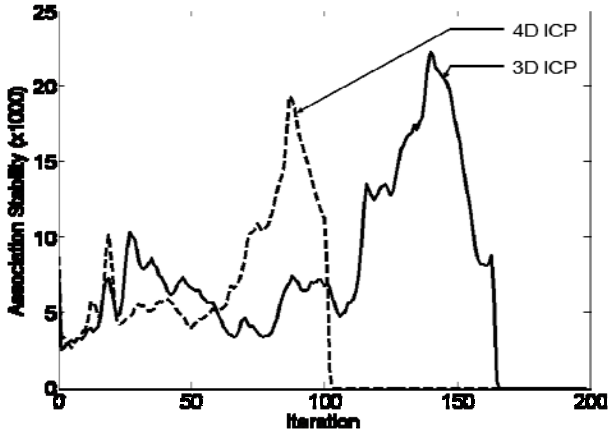
registration after 102th iteration and the traditional 3D ICP after the 164th iteration, which demonstrates the effectiveness of HICP for registering complex and realistic point clouds. The merged color point cloud about building is shown in Figure 24.



(a) Mean square error comparison.



(b) Associated point number comparison.



(c) Stability Comparison.

Figure 22 Registration comparisons between 3D ICP and 4D ICP algorithm.

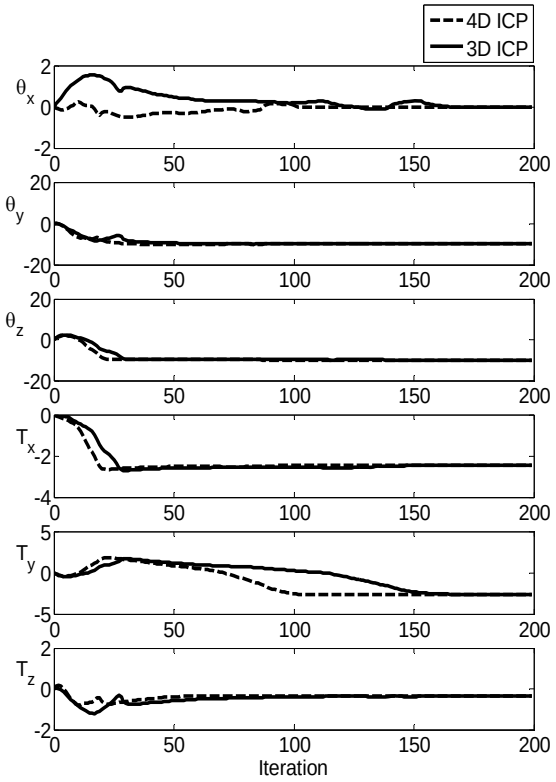


Figure 23 Convergence of translation and rotation estimates during registration

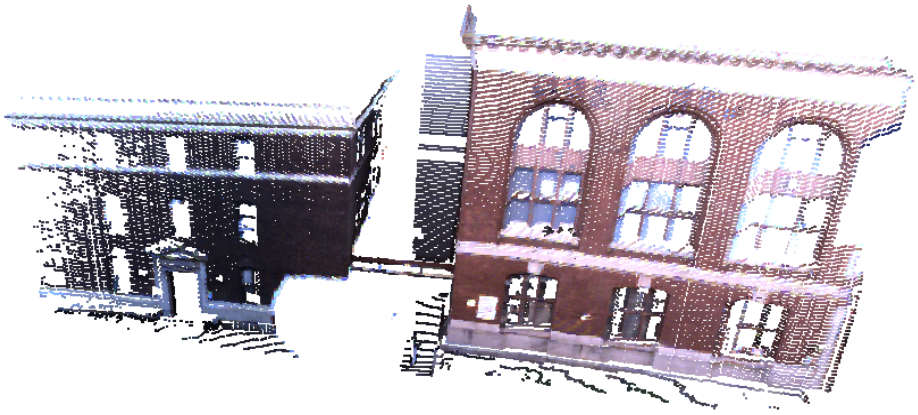


Figure 24 Registered data and model point clouds

5.6 SEQUENTIAL REGISTRATION OF MULTIPLE POINT CLOUDS

3D range ICP and H ICP algorithms have been applied on several outdoor map segments. Color point clouds taken from eight different vantage points have been registered together to construct a large scale color range map. Figure 25 shows the top view of outdoor mapping area in aerial image. This scene includes trees, road, electrical poles and buildings. Figure 26 shows the registered map and the vantage points from which map segments are obtained. Pair-wise registration is applied to construct a single map about the reference coordinate of the first map segment. 3D search radius in k-d tree was set as 1.5 and the 3D range data was normalized based on this radius. Hue value was normalized to a 0-1 range, hue search radius was set to be 0.15,

and hue weight was set to 5.0. The final error and the number of iterations required to register the point clouds is shown in Table 2. HICP requires less number of iterations than 3D ICP.

Figure25 Aerial image of outdoor mapping area and vantage positions.

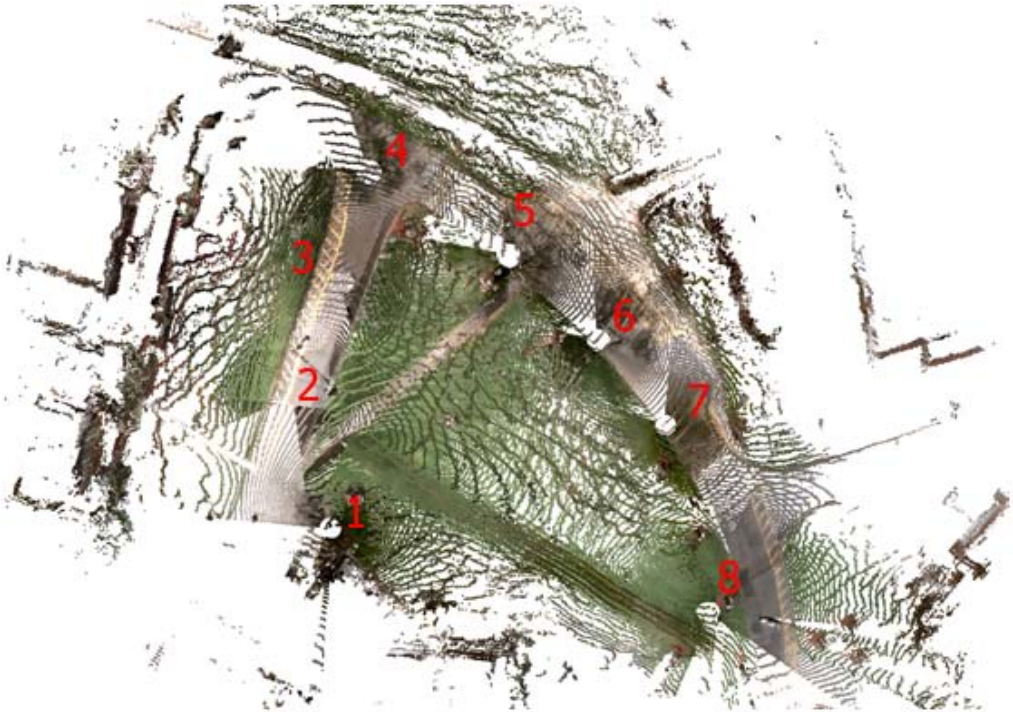
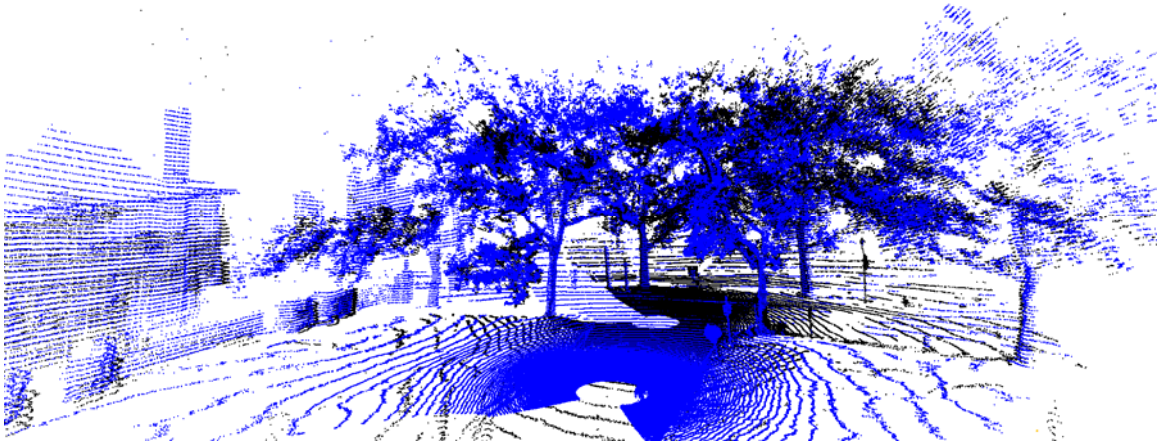


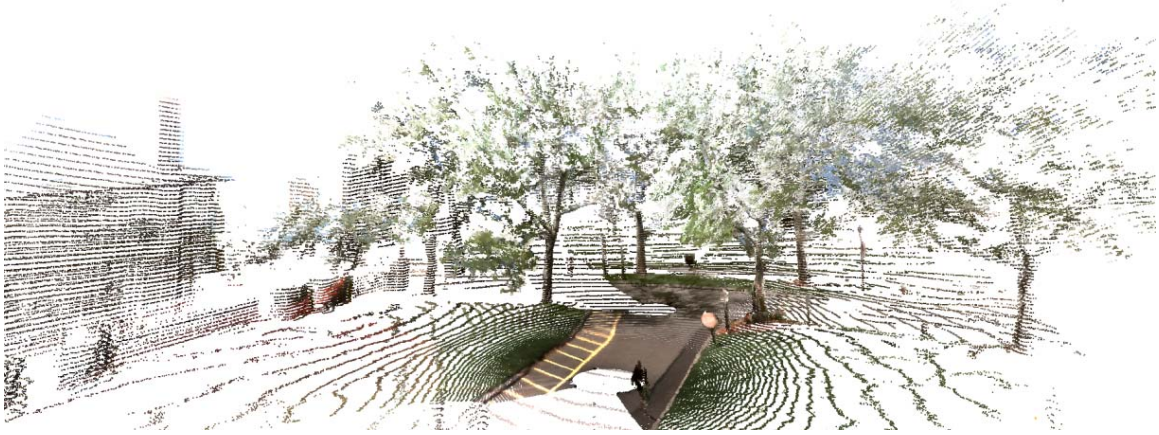
Figure26 Top view of eight sequentially registered color range maps.

Position	3D ICP Iterations	4D ICP Iterations	3D ICP Error	4D ICP Error
2	45	35	0.842	0.856
3	54	44	0.929	0.961
4	77	54	0.039	0.290
5	49	43	0.104	0.319
6	66	59	0.165	0.179
7	73	69	0.129	0.128
8	99	95	0.068	0.070

Table 2 Sequential registration of multiple point clouds.



(a) Registered position 4(black) point cloud into position 3 (blue) point cloud.



(b) Color point cloud after registration.

Figure 27 Map registered from scans taken from two vantage points

This experiment proves that faster registration will be conducted by adding color value into registration progress. Position 3 and 4 acquired range maps have been registered together and shown in Figure 27, Figure 27(a) describes two different range map with two different color, point cloud at position 4 (black) has been registered into position 3 point cloud (blue). Combined point clouds with color are shown in Figure 27(b).

6. FUTURE RESEARCH DIRECTIONS

Point clouds are inefficient representations of geometry. Some of the future research directions can include:

(a) Efficient generation of higher order geometric representations --- lines, surfaces and solids from the point cloud data;

(b) Map completeness measures that predict the geometry missing in the occluded areas based on a knowledge-base; and

(c) Extra sensing modalities such as infrared or thermal imaging, acoustic/ultrasonic and radio frequency imaging to help determination of materials in the scene.

Architecture, surveying and engineering fields have considerable needs for automatic or semi-automatic conversion of 3D point clouds into higher order line, surface and solid models that are compatible with commercial CAD software. This enables bringing the point cloud data into existing business processes like generation of drawings for code compliance, additions and modifications to existing built areas and remodeling interior spaces.

7. CONCLUDING REMARKS

This chapter describes an algorithm to introduce color attribute into point cloud registration process and fundamental algorithms for autonomous robotic complete mapping. Normalization of range data and hue value have been applied during the registration process and quantitatively evaluate the effect of hue search range and weight for the point association process. Different hue distribution and noise effect have been discussed with specific hue rendered color point clouds. A building data set and large-scale outdoor range map has been registered using image data assisted algorithm. Use of the hue value to assist the point association and error minimization is shown to be effective during the ICP iteration schemes. Higher dimensional point association based on weighted hue and range data leads more accurate point matching result, conduct earlier convergence of ICP progress, and reduce computation time. When rigid transformation is been application in every iteration loop during the ICP period, hue value does not change in space transformation. 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.

ACKNOWLEDGEMENTS

The authors acknowledge the partial support from US/ARMY ARDEC for the development of ROAMS robot. We acknowledge the help of Mr. Biruk Gebre for generating some of the scan data sets used in this chapter.

References

Andresson,H. (2007), *Vision Aided 3D Laser Scanner Based Registration*, Paper presented at European Conference on Mobile Robots: ECMR.

Andreason,H., Lilienthal, A.J. (2010), 6D Scan Registration using Depth-interpolated Local Image Features, *Robotics andAutonomous Systems*, 59, 157-165.

Arun, K.S. , Huang, T.S. and Blostein, S.D. (1987), Least Square Fitting of Two 3D-Point Sets, *IEEE Trans on Pattern Analysis and Machine Intelligence*, 9(5), 698-700.

Basilico, N., Amigoni, F. (2009), *Exploration Strategies based on Multi-Criteria Decision Making for an Autonomous Mobile Robot,* Presented at European Conference on Mobile Robots, Mlini/Dubrovnik, Croatia.

Blais, F. (2004), Review of 20 Years of Range Sensor Development, *Journal of Electronic Imaging*, 13, 231-243.

Browell, E.V., Butler, C.F., Ismail, S. et. Al (1990), Airborne Lidar Observations in the Wintertime Arctic Stratosphere: Polar Stratospheric Clouds, *Geophysical Research Letters*, 17, 385-388.

Bsel, PJ. (1992), A Method for Registration of 3D Shapes, *IEEE Trans on Pattern Analysis and Machine Intelligence*, 14, 239-256.

Chibunichev, A.G., Velizhev, A.B. (2008), *Automatic Matching of Terrestrial Scan Data Using Orientation Histograms*, Presented at the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Beijing, China.

Chua (1997) C.J.R, Point Signatures: A New Representation for 3D Object Recognition, *International Journal on Computer Vision*, 25, 63-85.

Johnson, A., *Spin-images: A Representation for 3D Surface Matching*, Doctoral dissertation, Carnegie Mellon University, USA.

Chung, D., Lee, Y.D.S. (1998), Registration of multiple range views using the reverse calibration technique, *Pattern Recognition*, 31(4), 457-464.

Davison, A. J.(2003), *Real-Time Simultaneous Localization and Mapping with a Single Camera*. Presented at the 9th IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin.

Druon, S. (2007), *Color Constrained ICP for Registration of Large Unconstructed 3D/Color Data Sets*, Presented at IEEE International Conference on Robotics and Automation, Roma, Italy.

Gebre, B., Men, H., Pochiraju, K. (2009), *Remotely Operated and Autonomous Mapping System(ROAMS)*, Paper presented at the 2nd Annual IEEE International Conference on Technologies for Practical Robot Applications, Woburn, MA,

Johnson, A.E., Kang, S.B. (1997), *Registration and Integration of Textured 3D Data*, Presented at International Conference on Recent Advances in 3D Didital Imaging and Modeling, Ottawa, Canada,

Joung, J.H., An, K.H., Kang, J.W. et al (2009), *3D Environment Reconstruction Using Modified Color ICP Algorithm by Fusion of a Camera and a 3D Laser Range Finger*. Presented at the 2009 IEEE International Conferrnce on Intelligent Robos and Systems, St. Louis, USA.

Kostelec, P.J., Rockmore, D.N. (2008), FFT on the Rotation Group, *Journal of Fourier Analysis and Application*,14, 145-179.

Fernandez, J.C., Singhania, A. et. Al (2007), An Overview of Lidar Point Cloud Processing Software, *GEM Center Report,* University of Florida.

Lorusso, A. (1995), *A Comparison of Four Algorithms for Estimating 3D Rigid Transformations*, Presented at British Machine Vision Conference.

Makadia, A., Iv, E.P.,Daniilidis, K. (2006), *Fully Automatic Registration of 3D Point Clouds*, Presented at the 2006 Computer Society Conference on Computer Vision and Pattern recognition, New York, NY.

Men, H., Pochiraju, K. (2010), *Hue Assisted Registration of 3D Point Clouds*, presented at ASME 2010 Interaional Design Engineering Technical Conference, Montreal, Canada.

Montemerlo, M.,Thrun, S., et al. (2003), *FastSLAM 2.0, An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges*, Presented at the International Joint Conference on Artificial Intelligence, Acapulco, Mexico.

Newman, P., Cole, D. and Ho, K. (2006), Outdoor SLAM using Visual Appearance and Laser Ranging, Presented at IEEE International Conference on Robotics and Automation, Orlando, Florida.

Nüchter, A.(2005), *6D SLAM with Approximate Data Association*. Presented at IEEE International Conference on Robotics and Automation, Barcelona, Spain.

Nuchter, A. (2007), *Catched k-d tree Search for ICP Algorithms*, Presented at 6th International Conference on 3-D Digital Imaging and Modeling, Montreal, Canada.

Oh, J. S., Choi, Y. H. et al (2004), Complete Coverage Navigation of Cleaning Robots Using Triangular-Cell-Based Map, *IEEE Transaction on Industrial Electronics*, 51, 718-727.
Olson, C. F. (2007), Probabilistic Self-Localization for Mobile Robot, *IEEE Transactions on Robotics and Automation*, 16, 55-67.
Robertson, C., (2002), Parallel Evolutionary Registration of Range Data, *Computer Vision and Image Understanding*, 87, 39-50.
Rusinkiewucz, S., (2001), *Efficient Variants of the ICP Algorithm*, Presented at 3rd International Conference on 3-D Digital Imaging and Modeling, Quebec City, Canada.
Salvi,J., Matabosch, C., and Fofi, D. et al (2007), A Review of Recent Range Image Registration Methods with Accuracy Evaluation, *Image and Vision Computing*, 25, 578-596.
Spletzer, J. R. (2003), *Sensor Fusion Techniques for Cooperative Localization in Robot Teams*, Doctoral dissertation, University of Pennsylvania.
Tarel, J., Civi, H., Cooper, D., (1998), Pose Estimation of Free-form 3D Objects without Point Matching using Algebraic Surface Models, Presented at IEEE Workshop on Model-based 3D.
Thrun, S. (1993), *Exploration and Model Building in Mobile Robot Domains*. Paper presented at IEEE International Conference on Neural Networks, San Francisco, USA.
Thrun, S., Fox, D., Burgard, W. and Dellaert, F. (2000), Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2), 99-141.
Thrun, S. (2003), Robotic Mapping: A Survey, *Exploring artificial intelligence in the new millennium* (pp.1-35), Morgan Kaufmann Publishers Inc.
Whyte, H.D., Bailey, T. (2006), Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms, *IEEE Robotics and Automation Magazine*.
Welch, G., Bishop, G. (1995), An Introduction to the Kalman Filter, *Technical Report TR 95-041*, University of North Carolina, Department of Computer Science.
Yamauchi, B. (1997), *A Frontier Based Approach for Autonomous Exploration*, Presented at IEEE International Conference on Autorotation and Robotics, Albuquerque, NM.
Zelinsky, A. (1994), Planning paths of complete coverage of an unstructured environment by a mobile robot, *Int. J. Robot. Res.*, 13(4), 315.

ADDITIONAL READING SECTION

Blais, F. (2004), Review of 20 Years of Range Sensor Development, *Journal of Electronic Imaging*, 13, 231-243.
Druon, S. (2007), *Color Constrained ICP for Registration of Large Unconstructed 3D/Color Data Sets*, Presented at IEEE International Conference on Robotics and Automation, Roma, Italy.
Gebre, B., Men, H., Pochiraju, K. (2009), *Remotely Operated and Autonomous Mapping System(ROAMS)*, Paper presented at the 2nd Annual IEEE International Conference on Technologies for Practical Robot Applications, Woburn, MA,
Fernandez, J.C., Singhania, A. et. Al (2007), An Overview of Lidar Point Cloud Processing Software, *GEM Center Report,* University of Florida.
Lorusso, A. (1995), *A Comparison of Four Algorithms for Estimating 3D Rigid Transformations*, Presented at British Machine Vision Conference.
Makadia, A., Iv, E.P.,Daniilidis, K. (2006), *Fully Automatic Registration of 3D Point Clouds*, Presented at the 2006 Computer Society Conference on Computer Vision and Pattern recognition, New York, NY.
Men, H., Pochiraju, K. (2010), *Hue Assisted Registration of 3D Point Clouds*, presented at ASME 2010 Interaional Design Engineering Technical Conference, Montreal, Canada.
Montemerlo, M.,Thrun, S., et al. (2003), *FastSLAM 2.0, An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges*, Presented at the International Joint Conference on Artificial Intelligence, Acapulco, Mexico.
Nüchter, A.(2005), *6D SLAM with Approximate Data Association*. Presented at IEEE International Conference on Robotics and Automation, Barcelona, Spain.

Nuchter, A. (2007), *Catched k-d tree Search for ICP Algorithms*, Presented at 6th International Conference on 3-D Digital Imaging and Modeling, Montreal, Canada.
Rusinkiewucz, S., (2001), *Efficient Variants of the ICP Algorithm*, Presented at 3rd International Conference on 3-D Digital Imaging and Modeling, Quebec City, Canada.
Salvi,J., Matabosch, C., and Fofi, D. et al (2007), A Review of Recent Range Image Registration Methods with Accuracy Evaluation, *Image and Vision Computing*, 25, 578-596.
Thrun, S. (2003), Robotic Mapping: A Survey, *Exploring artificial intelligence in the new millennium* (pp.1-35), Morgan Kaufmann Publishers Inc.

KEY TERMS & DEFINITIONS

3D Map: 3D map represents certain level of spatial information about the geometry features in specific area. The map is dimensionally accurate and may have a relative or absolute reference.
3D Mapping: 3D mapping is the process of applying measurement devices to construct 3D map about specified environment.
Point Cloud: Discrete points group with accurate 3D coordinates describing object surface dimensional measuremnts, usually contructed by laser ranging devices.
Color Point Cloud: Discrete points group with both dimensiaonl accurate measurement and texture property, normally generated by both ranging device and color camera.
Map registration: A process to accurately stitch pair or multiple point clouds together into single point cloud.