# Target-Free Automatic Registration of Point Clouds

**Pileun Kim[a], Yong Kwon Cho[a], and Jingdao Chen[b]**

[a]School of Civil and Environmental Engineering, Georgia Institute of Technology
[b]School of Electrical and Computer Engineering, Robotics Institute, Georgia Institute of Technology
E-mail: pkim45@gatech.edu,yong.cho@ce.gatech.edu, jchen490@gatech.edu

**Abstract –**

**The aim of this paper is to introduce a novel method that automatically registers colored 3D point cloud sets without using targets or any other manual alignment processes. For fully automated point cloud registration without targets or landmarks, our approach utilizes feature detection algorithms used in computer vision. A digital camera and a laser scanner is utilized and the sensor data is merged based on a kinematic solution. The proposed approach is to detect and extract common features not directly from a 3D point cloud but from digital images corresponding to the point clouds. The initial alignment is achieved by matching common SURF features from corresponding digital images. Further alignment is obtained using plane segmentation and matching from the 3D point clouds. The test outcomes show promising results in terms of registration accuracy and processing time.**

**Keywords –**

**3D LIDAR; Point cloud; Registration; Feature point; RGB image; Segmentation**

## 1 Introduction

Virtual 3D scenario modelling and mapping is critical for construction applications for understanding the scene of interest, monitoring construction progress and detecting safety hazards. The abundant amount of point cloud data can be used to efficiently model a construction site. In most situations, the construction site has to be scanned from many different viewpoints in order to obtain a complete reconstruction of the site. This is due to the fact that some scans may be affected by occlusion and each scan position has its own local coordinate system. Therefore, all the individual point clouds collected in the local coordinate frame must be transformed to a global coordinate system in a procedure known as point cloud registration. However, most of the current methods used for point cloud registration work properly only if there is a good initial alignment or manually marked correspondences and targets [1]. Therefore, an effective point cloud registration method is an indispensable tool to merge several acquired point cloud data from different scanning points in the as-built construction site modelling pipeline. In this study, a target- free automatic point cloud registration framework is introduced.

## 2 Literature Review

Target-free automatic point cloud registration methods have been widely studied in the literature to overcome the limitations of target-based registration methods. There are three types of point cloud registration methods in existence: 1) ICP based, 2) feature-based, and 3) geo referencing based.

### 2.1 Iterative Closest Point (ICP) based

The most popular method for point cloud registration is the iterative closest point (ICP) algorithm developed by Besl and McKay [2], Chen and Medioni [3]. In the ICP algorithm, the closest points in two different scans are used as relative control points. Furthermore, the iterative closest compatible point (ICCP) algorithm has been presented to reduce the search space of the ICP algorithm. In the ICCP algorithm, the minimization of distance is accomplished only between the pairs of points considered compatible on their viewpoint invariant properties[4].In addition, Men and Pochiraju [5] integrated Hue values in the registration process to achieve a 4D ICP algorithm. With the Hue values, the ICP algorithm is able to achieve higher accuracy and faster convergence. However, ICP-based registration methods still incur problems with calculation time due to the heavy computation load involved in the ICP algorithm. Also, the performance may not be reliable depending on the overlapping area and the initial starting points [6].

### 2.2 Feature based

Feature-based registration can be realized without knowing initial starting points since 2D images are used to assist the recognition of feature points. The feature point extraction based method utilized 2D intensity images with Scale-invariant feature transform (SIFT) [7]. However, this method is very sensitive to the overlapping area size. In addition, a large number of scans are necessary to achieve good performance, and the image feature extraction process can be heavily affected by the environment due to illumination changes. Another significant disadvantage of feature-based registration is that a heavy amount of calculation is involved [8]. Even though thousands of feature points can be extracted from each scan based on geometry or image information, most of them are filtered out because of incorrect matches.

### 2.3 Geo referencing based

Olsen and Johnstone [9] proposed a registration method where the position of each scan spot is derived from GPS. This method is widely used in outdoor studies, but suffers from a lack of accuracy in the registration process because of the low accuracy of GPS. For indoor registration, an automatic construction of 3D basic-semantic models of inhabited interiors was developed using laser scanners with RFID [10]. This method is applicable only for indoor spaces, and the laser scanner is required to be set up in close proximity to the objects in order to recognize the RFID tags. Thus, geo-referencing based registration using sensors such as GPS and RFID is not suitable for all situations because of the limitations in sensor performance [11].

Therefore, there are significant challenges involved in order to achieve a target-free automation point cloud registration method in complex data collection environments in a rapid and accurate manner.

## 3 Hybrid 3D LIDAR system

In this study, a robotic hybrid Light Detection And Ranging (LiDAR) system was used, consisting of four SICK LMS511 2D line laser scanners (65 meter working ranges at 25Hz scan speed, 200 sec / 360º scan, 190º for vertical line), and a regular DSLR 2D camera, as shown in Figure 1. The resolution of each line laser is 0.1667 degrees in a vertical direction and 0.072 degrees in a horizontal direction. The customized 3D LiDAR system provides more flexibility in hardware control and software programming compared to a commercial LiDAR scanner.

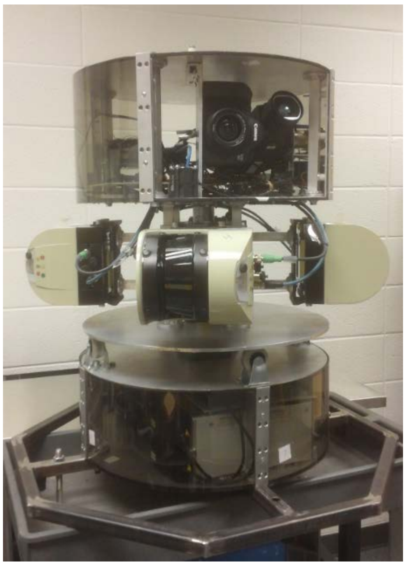

**Figure 1 Hybrid 3D LIDAR system**

### 3.1 Data fusion of point clouds and digital images

A digital camera captures RGB texture data from the surroundings, which can be mapped on the 3D point clouds. In the texture mapping process, a camera calibration step is necessary for the digital camera. There are two kinds of matrices for camera calibration. The camera calibration process is finding internal and external parametric matrix for the camera which affect the image processing process. The internal parametric matrix, also known as intrinsic parameters, consist of focal length, image sensor format, and principal point that could be estimated by the pinhole camera model. The intrinsic parameters can be denoted by Equation (1).

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

The parameters $f_x$ and $f_y$ are associated with the focal length whereas the parameters $c_x$ and $c_y$ representthe principal point [12].

The external parametric matrix, also known as extrinsic parameters, denote the coordinate transformation from 3D world coordinates to 3D camera coordinates. This transformation is necessary since the laser scanned 3D point cloud data is obtained in 3D world coordinates. The extrinsic parameters can be obtained through a kinematic relationship based on the mounting configuration.

$$K_{ext} = [R \quad T] = \begin{bmatrix} R_{11} R_{12} R_{13} T_x \\ R_{21} R_{22} R_{23} T_y \\ R_{31} R_{23} R_{33} T_z \end{bmatrix} \qquad (2)$$

Using these intrinsic and extrinsic parameters, the laser scanned 3D point cloud can be transformed to 3D camera coordinates according to Equation (3).

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = K_{int} K_{ext} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} R_{12} R_{13} T_x \\ R_{21} R_{22} R_{23} T_y \\ R_{31} R_{23} R_{33} T_z \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \qquad (3)$$

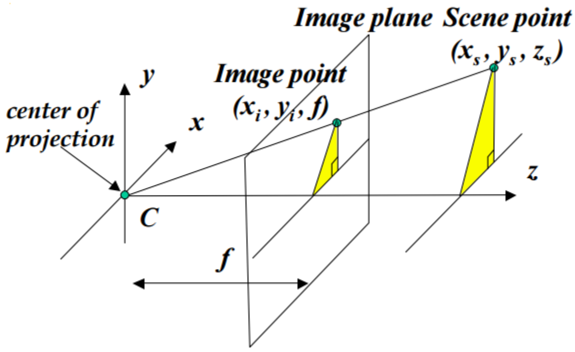


**Figure 2 Projection from world coordinate to image plane**

Then, the 3D camera coordinates can be transformed to the 2D digital image plane by the Equation (4).

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \frac{u}{w} \\ \frac{v}{w} \end{bmatrix} \qquad (4)$$

Thus, the coordinate systems of 3D point cloud data and RGB image data are aligned using the concept of perspective projection [13]. Figure 2 shows how the point cloud of scene transforms to the image plane. This enables a correct texture mapping between a point cloud and digital camera images.

## 3.2 RGB feature point based transformation

There are many techniques for detection of points of interests in an image such as Harris corners, SIFT, and SURF. The method of Harris features was proposed by Harris and Stephens in 1988. The notion of corner should only be taken in a general sense as it includes not only corners, but edges and keypoints. The limitation of Harris features is that they are not scale invariant and need to be recomputed for different scales. On the other hand, the Scale Invariant Feature Transform (SIFT) presented by David Lowe detects scale-invariant image feature points, which can easily be matched between images to perform tasks such as object detection and recognition, or to compute geometrical transformations between images. Additionally, the Speeded Up Robust Feature (SURF) is a robust descriptor motivated by the SIFT descriptor, both of them using local gradient histograms. The main difference between the two descriptors is the performance, where SURF decreases the computation time by using integral images for image convolutions and a Hessian matrix-based detector. The standard version of SURF is several times faster than SIFT and is claimed by the authors to be more robust against different image transformations than SIFT [14].

In this study, SURF feature points are used to obtain the initial transformation between point clouds. Once the feature points of each image are extracted, we can track points in the 3D point cloud by matching feature points in the image plane to the RGB-fused point cloud data set.

The Kabsch algorithm (root mean square distance concept) is used to estimate the transformation matrix between point clouds. The algorithm starts with two sets of paired points, P and Q, where each set of points are represented as an N×3 matrix.

$$P = \begin{bmatrix} x_1 y_1 z_1 \\ x_2 y_2 z_2 \\ \vdots \ \vdots \ \vdots \\ x_n y_n z_n \end{bmatrix} \quad Q = \begin{bmatrix} x_1{}' \ y_1{}' z_1{}' \\ x_2{}' \ y_2{}' z_2{}' \\ \vdots \ \vdots \ \vdots \\ x_n{}' \ y_n{}' z_n{}' \end{bmatrix}$$

Both sets of points should be translated, so that their centroid corresponds with the origin of the coordinate system. This could be achieved by subtracting from the point coordinates the coordinates of the respective centroid. The next step consists of calculating a covariance matrix A. The optimal rotation matrix U can be computed by singular value decomposition (SVD) of the covariance matrix A.

$$A = P^T Q = VSW^T \qquad (5)$$

Next, we decide whether we need to correct our rotation matrix in order to ensure a right-handed coordinate system using the variable *d*.

$$d = \mathrm{sign}(\det(WV^T)) \qquad (6)$$

Finally, we can calculate our optimal rotation matrix, U, as

$$\mathrm{U} = \mathrm{W}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{bmatrix} V^T \tag{7}$$

and the translation matrix D can be obtained by the difference between the centroid of each point cloud [15].

To match each point cloud set, the initial rigid transformation matrix is defined. In this case, the transformation is a perspective projection for 6 degrees of freedom, composed of a rotation matrix and a translation vector in 3 dimensions. This transformation can be written as 3x4 matrix. We can then project a point P where $\mathrm{P} = [x\ y\ z\ 1]^T$ simply by applying this transformation matrix to the point:

$$P' = TP = \begin{bmatrix} U_{11} U_{12} U_{13} D_x \\ U_{21} U_{22} U_{23} D_y \\ U_{31} U_{23} U_{33} D_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{8}$$

The summary of finding an initial transformation using RGB feature points is as follows:

1. SURF features are extracted from RGB panorama images using only 2D information
2. Identify coordinates in the image frame for each extracted feature
3. Track the feature points on the corresponding 3D point cloud with RGB-fused point cloud data
4. Compute an initial transformation matrix using Kabsch algorithm
5. Transform a point cloud to the reference point cloud

### 3.3 Point cloud registration using plane matching and corner points

Two point clouds can be further registered using the method of plane-to-plane matching. This method relies on finding three plane correspondences between the point cloud to be registered and the reference point cloud. The selected planes have to be linearly independent and intersect at a unique point in order to fully recover the transformation parameters. For example, one of the planes can be the ground plane whereas the second plane is a vertical wall in the x-axis whereas the third plane is a vertical wall in the y-axis. First, the Random Sample Consensus (RANSAC) algorithm is used to perform plane segmentation for each point cloud. The RANSAC algorithm works by iteratively sampling points from a given point cloud and estimating a set of plane parameters of the form $ax + by + cz + d = 0$. The best estimate is determined as the set of plane parameters that maximizes the number of points that are considered inliers [16]. The obtained plane parameters are used to segment the original point cloud into points belonging to the plane and the remaining points. This procedure is repeated until three suitable plane candidates are found that satisfy the linear independence criteria. Then, plane correspondences between the input point cloud and the reference point cloud are determined by finding the closest match between normal vectors. Second, the rotation component R of the transformation matrix is calculated using the plane normal vectors found in the previous step. The rotation component is determined such that the normal vectors $(n_1, n_2, n_3)$in the input point cloud will be transformedin order to match the normal vectors $(n_1, n_2, n_3)$in the reference point cloud [17]. An intermediate rotation matrix that rotates a vector $v_1$ to another vector $v_2$ is derived using Equation (9).

$$R = I + skew(v_1 \times v_2) + skew(v_1 \times v_2)^2 \frac{1 - v_1 \cdot v_2}{\|v_1 \times v_2\|} \tag{9}$$

Three intermediate rotation matrices are calculated for each plane correspondence as shown in Equation 2. The final rotation matrix is then obtained by multiplying the intermediate rotation matrices together.

$$R_1 = getRotation(n_1, n_1')$$
$$R_2 = getRotation(R_1, n_2, n_2')$$
$$R_3 = getRotation(R_2 R_1, n_3, n_3')$$
$$R = R_3 R_2 R_1$$

Third, the translation component T of the transformation matrix is calculated by comparing corner points between the point cloud to be registered and the reference point cloud. A corner point is defined as the unique intersection point between three planes. The corner point can be calculated by solving three plane equations simultaneously for the $(x, y, z)$ values. This in turn can be formulated as a matrix-vector multiplication operation using the corresponding plane parameters. Once the corner point is obtained for each point cloud, the translation vector is determined as the difference between the positions of the two corner points. The calculations involved in this step are shown in detail in Equation (10).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}^{-1} \begin{bmatrix} -d_1 \\ -d_2 \\ -d_3 \end{bmatrix}$$
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a_1' & b_1' & c_1' \\ a_2' & b_2' & c_2' \\ a_3' & b_3' & c_3' \end{bmatrix}^{-1} \begin{bmatrix} -d_1' \\ -d_2' \\ -d_3' \end{bmatrix} \tag{10}$$
$$T = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$T = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Finally, a registered version of the input point cloud is obtained by applying the rotation and translation operations to each point in the point cloud.

## 4 Test

The data acquisition process for validating the proposed framework is performed at the Bunger-Henry building in Georgia Tech.Figures3 and 4 show the two sets of RGB-fused point clouds scanned at different scan positions with the proposed texture mapping algorithm.

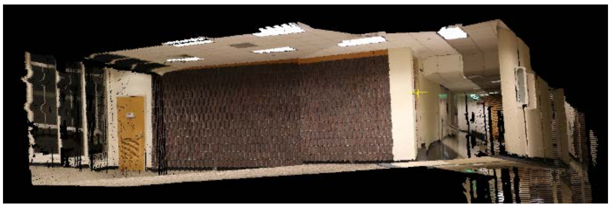

**Figure 3 3D RGB-fused point cloud at first scan position**

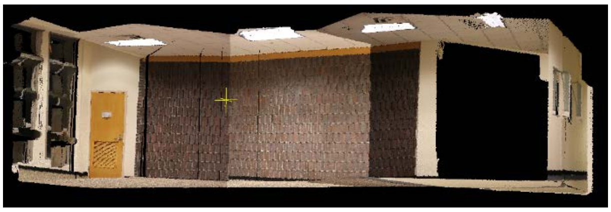

**Figure 4 3D RGB-fused point cloud at second scan position**

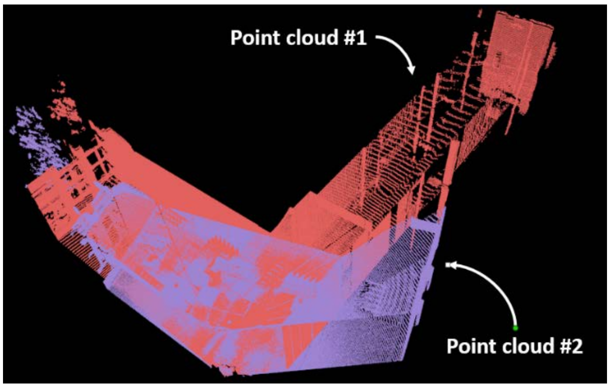


**Figure 5 Original point clouds from different scan position**

Figure 5 visualizes the two raw point cloud data set from the different scan positions. The two scan data collected as viewed are in the local coordinate system so the origin is the same. In this experiment, the first scan data set will be transformed and the second scan data set will be used as the reference data set. Each scan data collection occurred in an indoor environment where there are fewer visual features. Figure 6 shows the RGB feature extraction results using SURF features for the collected panoramic images. However, not all the feature points are correctly matched between the two panoramic views due to similar feature points in both images. For this reason, the final plane-to-plane matching step is necessary to achieve the desired accuracy for the presented framework.

**Figure 6 Feature point extraction for panoramic images**

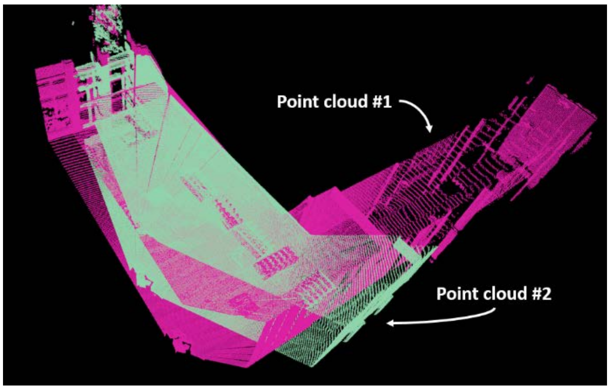


**Figure 7 Transformed and reference point cloud after feature point matching step**

Figure 7 visualizes the point cloud sets after the initial transformation using RGB feature extraction. As shown in this figure, the point cloud sets are better aligned compared to the raw data set in Figure 5. It can be verified from the reduced deviation angle in Table 1. Then, we can apply the third step which is the plane-to-plane matching algorithm. The initial alignment is sufficient that we can easily find plane correspondences in the final step. Finally, Figure 8 represents the final result for the proposed framework. To verify the result for this test, the second point cloud set is assumed as a true ground and measured the deviation angle from each reference axis at each step of the proposed framework.

**Table 1 Deviation angle**

| Dev. angle measured from | Original point clouds | After initial transform | After final transform |
|---|---|---|---|
| X axis | -17.404 | -7.188 | 0.549 |
| Y axis | -3.155 | -1.702 | 0.342 |
| Z axis | 29.899 | 9.430 | -0.811 |

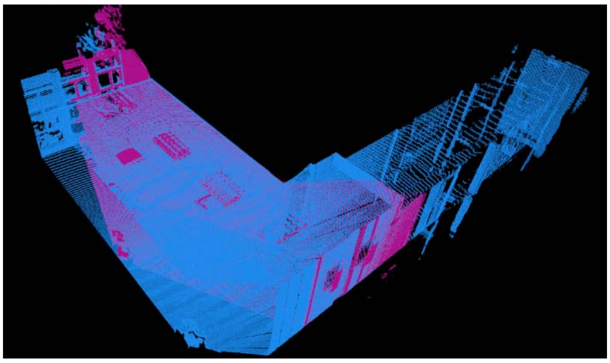

**Figure 8 Transformed and reference point cloud after plane-to-plane matching step**

# 5 Conclusion

In conclusion, a novel method for target-free automatic point cloud registration was demonstrated and validated. A laser scanning system with a digital camera was used to obtain point clouds with mapped RGB texture data. The proposed framework consists of three steps. The Figure 9 indicates the flow diagram for the presented framework. The first step involves constructing RGB-fused point clouds with collected $(x, y, z)$ position data from laser scans and RGB information from digital images. The next step involves obtaining an initial transformation matrix by extracting common SURF feature points in digital images and tracking its corresponding $(x, y, z)$ position. Lastly, plane-to-plane matching algorithm is utilized for accurate registration.

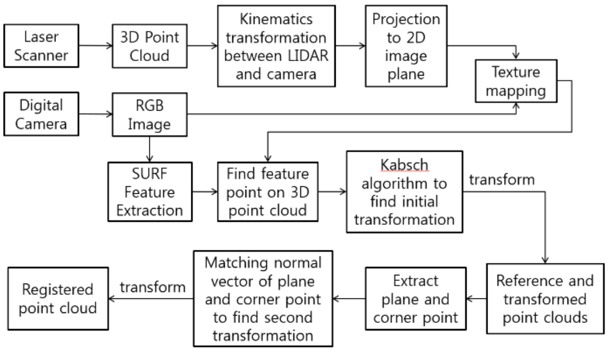


**Figure 9 Procedure of proposed framework**

Even though the proposed framework should have three plane with one corner point on overlapped area, it achieved automatic point cloud registration without the need for target references and manual adjustments. This framework consist of two sequential methods: RGB feature matching and plane-to-plane matching. The plane-to-plane matching algorithm only works fine, when the RGB feature matching method get a good results. For future work, we focus on how to get more good result for RGB feature point matching and how to apply this framework for more complex and large areas with scans from multiple viewpoints.

# 6 Acknowledgement

This material is based upon work supported by the National Science Foundation (Award #: CMMI-1358176). Any opinions, findings, and conclusions or recommendations expressed on this material are those of the authors and do not necessarily reflect the views of the NSF. This research was also supported by the Chung-Ang University Research Grants in 2015.

# References

[1] H. Xiong, S. Szedmak, and J. Piater. A study of point cloud registration with probability product kernel functions. *In Proc. 2013 Int. Conf. 3D Vision, IEEE*, pp. 207–214, Seattle, USA, 2013.

[2] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (2):239–256, 1992.

[3] Chen, Y., Medioni, G. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3): 145–155, 1992.

[4] Akca, D., Full automatic registration of laser scanner point clouds. *Optical 3-D Measurements VI*, pp. 330–337, 2003.

[5] Men, H.,Gebre, B., Pochiraju, K. Color Point Cloud

Registration with 4D ICP Algorithm. *Int. Conf. Robotics and Automation, IEEE*, pp. 1511-1516, Shanghai, China, 2011.

[6] Wang, C., Cho, Y., and Park, J. Performance Tests for Automatic Geometric Data Registration Technique for Construction Progress Monitoring. *Int. Conf. Computing in Civil and Building Engineering (ICCCBE), ASCE*, pp. 1053-106, Orlando, FL. 2014.

[7] Eo, Y. D., Pyeon, M. W., Kim, S. W., Kim, J. R., Han, D. Y. Coregistration of terrestrial lidar points by adaptive scale-invariant feature transformation with constrained geometry. *Automation in Construction*, 25, 49-58, 2012.

[8] Gai, M., Cho, Y., and Qinghua, X. Target-free automatic point clouds registration using 2D images. *Proceedings of the 2013 ASCE International Workshop on Computing in Civil Engineering*, pp.865-872, Univ. of Southern California, LA, USA. 2013.

[9] Olsen, M. J., Johnstone, E., Kuester, F., Driscoll, N., and Ashford, S. A. New Automated Point-Cloud Alignment for Ground-Based Light Detection and Ranging Data of Long Coastal Sections. *Journal of Surveying Engineering, ASCE*, 137(1), 14-25, 2011.

[10] Valero, E., Adan, A., and Cerrada, C. Automatic Construction of 3D Basic-Semantic Models of Inhabited Interiors Using Laser Scanners and RFID Sensors. *Sensors*, 12(5), 5705-5724, 2012.

[11] Mastin, A., Kepner, J. and Fisher, J. Automatic registration of lidar and optical images of urban scenes. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2639–2646, 2009

[12] R. Y. Tsai. A Versatile Camera Calibration Technique for 3D Machine Vision. *IEEE J. Robotics and Automation*.3(4), pp. 323-344, 1987.

[13] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Approach*. MIT Press, pp. 33-68, 1996

[14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, pp. 138-183, 2000.

[15] Wikipedia. https://en.wikipedia.org/wiki/Kabsch_algorithm, Accessed: 02/03/2016

[16] R. B. Rusu and S. Cousins, 3D is here: Point Cloud Library (PCL), *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011.

[17] Mathematics Stack Exchange. http://math.stackexchange.com/questions/180418/calculate-rotation-matrix-to-align-vector-a-to-vector-b-in-3d, Accessed: 05/03/2016