ROBUST RANGE IMAGE REGISTRATION USING 3D LINES

Jian Yao *Mauro R. Ruggeri* *Pierluigi Taddei* *Vítor Sequeira*

European Commission - Joint Research Centre (JRC), Via E. Fermi, 2749, I-21027 Ispra (VA), Italy

ABSTRACT

We present an efficient method for accurate automatic registration of two geometrically complex 3D range scans by using 3D lines. We first detect edges from the associated 2D reflectance images and collect 3D edge contours by only taking into account valid foreground points. Then we use an efficient split-and-merge line fitting algorithm to detect 3D lines. We build a fast search codebook to efficiently match the two sets of 3D lines. This is done by computing the orientation angle and distance of pairs of 3D lines in each set, both of which are invariant under rigid transformations. Finally we recover the rigid transformation between two scans using an efficient RANSAC algorithm with robust transformation estimation that exploits two sets of corresponding 3D lines. We conclude presenting experimental results that demonstrate efficiency and accuracy of our proposed method.

Index Terms— Line matching, range image registration.

1. INTRODUCTION

In the past decade, there was a growing interest for 3D reconstruction and realistic 3D modelling of large-scale scenes such as urban structures. Applications of such models include Virtual Reality, Cultural Heritage, Urban Planning and Architecture. Commonly these applications require a combination of range sensing technology with traditional digital photography. The range sensing part is critical for the accurate geometric representation of the scenes. In recent years 3D laser scanners that are able to provide satisfying measurement accuracy for different applications appeared on the market. However, applications need a systematic and automatic way for registering 3D range scans to visualize them in a common coordinate system. This paper presents a new method that utilizes 3D lines extracted from range images to achieve automated registration.

Iterative Closest Point (ICP) [1] is one of the most popular range registration algorithms. It provides very accurate results but requires a good initial transformation. Recently researches have focused on methods to provide an initial solution which is then fed to ICP for precise registration optimization. Typically these methods rely on extracting and matching features from the input range images [2, 3, 4]. Chen and Stamos [2] proposed to extract the circular features from range images for matching. However, circular features are rare and only exist in some specific scenes. Stamos et al. [3, 4] proposed to extract planes from the range images and exploit their intersecting lines to match two scans. The extracted lines mostly correspond to real surface edges (surface normal and depth discontinuities) in 3D space. These approaches provide accurate results in scenes of urban nature with plenty of planes, but will fail in scenes that do not contain a sufficient amount of intersecting lines. Zaharescu et al. [5] developed a novel 3D invariant feature detector and descriptor for uniformly triangulated meshes, which can be used to match rigid objects and non-rigid objects. In this paper, we propose to match 3D lines extracted from range images. We detect 3D lines directly from edge contours extracted from 2D reflectance images instead of recovering the plane-plane intersecting lines as in [3, 4]. The extracted 3D lines correspond to real surface edges and also to intensity edges (which have no obvious surface normal and depth discontinuities). Thus, our method is able to register scans taken from more general scenes. The invariance of both orientation angle and distance of a line pair under rigid transformation is sufficient to filter out outliers and greatly reduce the search time. In addition, our method also introduces median adjacent distances of fitted 3D lines to reduce the matching impact of lines extracted under different resolutions.

The paper is organized as follow. Section 2 describes the method to detect 3D line from edges extracted from 2D reflectance images. The automated registration of two range images based on line matching is described in Section 3. Finally Section 4 presents the results on real datasets and concludes the paper.

2. 3D LINE EXTRACTION

Edges extracted from 2D reflectance images may fall in one of the following three types: (a) edges caused by surface normal discontinuities (crease edges), (b) edges caused by depth discontinuities (step edges), and (c) edges caused solely by intensity discontinuities (intensity edges). Most of 3D edge detectors [4] are able to detect only crease and step edges. However edges can also produce relevant intensity changes, which result in 2D edges on 2D reflectance images. In order to increase the number of detected edge contours, we apply the edge detection proposed by Wang et al. [6] to the input 2D reflectance image.

After extracting the 2D edge points we cluster them in candidate edges represented by ordered sets of connected points with enough cardinality.

We then detect step edges caused by one surface occluding another (occlusion edges) which are not really caused by 3D surface boundaries and that, in the general case, are not visible from other viewpoints. Let P be an edge point and

B_1 and B_2 the two nearest neighbours of the original image along the gradient direction in P. We say that the point P is an occlusion edge point and B_1 is a foreground point on a 3D surface boundary if $\|P - B_1\| > r_d\|P - B_2\|$ and $depth(B_1) < depth(P)$ where $\|\cdot\|$ denotes the L2-norm, $depth(X)$ denotes the depth value of the point X, and r_d is a constant factor ($r_d = 10$ in our experiments). Notice that occlusion edge points might represent false edges whereas foreground points represents a 3D surface boundary. We thus replace the coordinates of all occlusion edge points with the coordinates of their respective foreground points.

Secondly we remove points affected by measurement noise and outliers of the edge detection step. Let $C = \{X_i\}_{i=1}^N$ be an edge contour consisting of N points. For this contour, we compute a *median adjacent distance* $\bar{d}_c$, which is the median of distances $\{\|X_{i-1} - X_i\|\}_{i=2}^N$ between two adjacent 3D points on the contour. The point X_i is removed from the contour C if $\min(\|X_{i-1} - X_i\|, \|X_i - X_{i+1}\|) > \alpha_c \bar{d}_c$ where α_c is a predefined value ($\alpha_c = 5$ in this paper).

Finally we fit 3D lines to an edge contour by utilizing a modified version of the popular standard Split-and-Merge algorithm [7]. Initially we fit a 3D line to the whole contour segment using least-squares computing the distances of the points on the contour segment to the fitted line. The contour segment would be regarded as a successfully fitted line if the maximal point-to-line distance at the point P is less than a threshold $t_m = \alpha_m \bar{d}_{fc}$ and most (90% in this paper) of distances are less than a threshold $t_p = \alpha_p \bar{d}_{fc}$ where $\alpha_m > \alpha_p$ and $\bar{d}_{fc}$ is the median adjacent distance of this contour segment. Otherwise, we split the contour segment at P into two segments. Finally if a segment is not long enough it is discarded. In this way, the algorithm recursively splits the contour until all splitted segments have been checked. The resulting lines are fitted again using their inlier points. After the split step the algorithm replaces collinear lines into new lines representing their union. After we apply the Split-and-Merge algorithm on all the edge contours, we further merge collinear lines. Note that here we keep the lines to be merged, which are useful for line matching. The set of L 3D lines extracted from a range image is described using the following notation:

$$\mathcal{L} = \{\mathbf{l}_i | \mathbf{l}_i = (P_i, V_i, \mathcal{X}_i, \bar{d}_i)\}_{i=1}^L$$

where P_i is the midpoint of the line $\mathbf{l}_i$, V_i is a normalized line orientation (i.e. $\|V_i\| = 1$), $\mathcal{X}_i$ is a set of 3D inlier points belonging to the line $\mathbf{l}_i$, and $\bar{d}_i$ is the median adjacent distance. Fig. 1 shows the extracted 3D lines of some scans.

3. LINE MATCHING

In order to align two overlapping range scans S and S' taken from different viewpoints we seek to recover the transformation $\mathcal{T} = (\mathbf{R}, \mathbf{T})$ that brings points in S to points in S' where $\mathbf{R}$ is the 3×3 rotation matrix and $\mathbf{T} = [T_x, T_y, T_z]^\top$ is the translation vector.

3.1. Transformation Estimation

Due to the fact that the endpoints of extracted line segments can never be exactly localized, and thus matched, we must rely only on the orientations and positions of the underlining lines to estimate the transformation.

Let $\mathcal{L} = \{\mathbf{l}_i\}_{i=1}^L$ and $\mathcal{L}' = \{\mathbf{l}'_m\}_{m=1}^{L'}$ be two sets of 3D lines extracted from S and S' respectively. The transformation recovery is a well posed problem if $\mathcal{L}$ and $\mathcal{L}'$ contain two or more matching pairs of lines that are not all parallel. By using $M (M \geq 2)$ pairs of such matched lines, in fact, the method proposed in [8] provides a closed-form solution for the transformation $\mathcal{T}$. This method cannot be applied directly due to the possible lack of accuracy of the line localization. This is due to the presence of noise on the extracted 2D edges. In addition range images contain more points related to regions close to the scanner sensor and more points can be also seen with a low incidence angle. Such regions are said to have a high resolution with respect to other regions of the same range image. In the general case localized line pairs related to points of regions with different resolution will not precisely match even in noise free conditions.

To overcome this problem we consider M pairs of matched lines that are not all parallel and compute the transformation $\mathcal{T}$. We then verify $\mathcal{T}$ by transforming all lines in $\mathcal{L}$ and check for their corresponding lines in $\mathcal{L}'$. All newly matched lines are used to refine the transformation.

Robustness to input noise, instead, could be achieved by increasing the number M of matched line pairs. Increasing the required number of line pairs, though, reduces the probability of randomly selecting a group of inliers pairs during the RANSAC step. For example, if K lines in $\mathcal{L}$ have corresponding matched lines in $\mathcal{L}'$, the probability of successfully selecting M lines in $\mathcal{L}$, having matched lines in $\mathcal{L}'$, is $p(M) \approx \left(\frac{K}{L}\right)^M$. Thus for small $\frac{K}{L}$ and large M the successful selection probability $p(M)$ is very low.

To recover a good initial transformation using less M pairs of matched lines, we use the robust line matching algorithm proposed in [9], which minimizes the following distance measure:

$$D(\mathcal{L}'_M, \mathcal{T}\mathcal{L}_M) = \sum_{n=1}^{M} \big[L_n \|P'_n - \mathbf{T} - \mathbf{R}(P_n + s_n V_n)\|^2 + L_n^3 (1 - V_n^\top \mathbf{R} V'_n)/6\big] \quad (1)$$

where we assume that $\mathcal{L}_M = \{\mathbf{l}_n\}_{n=1}^M \in \mathcal{L}$ matches the set $\mathcal{L}'_M = \{\mathbf{l}'_n\}_{n=1}^M \in \mathcal{L}'$, $L_n = \|\mathbf{l}_n\|$ is the length of the line $\mathbf{l}_n \in \mathcal{L}_m$, s_n is the shift parameter of a point on the line $\mathbf{l}_n$. The transformation $\mathcal{T}$ that operates on $\mathcal{L}_M$ is given by

$$\mathcal{T}P_n = \mathbf{R}P_n + \mathbf{T}, \mathcal{T}V_n = \mathbf{R}V_n. \quad (2)$$

This approach is an iterative algorithm for computing the transformation $\mathcal{T}$ which is iteratively computed by updating the shift parameters $\{s_n\}_{n=1}^M$. In this paper, we modify L_n as $L_n = \sqrt{\|\mathbf{l}_n\| \cdot |\mathcal{X}_n|}$ where $|\mathcal{X}|$ denotes the size of the set $\mathcal{X}$. In addition, to avoid the impact of matched lines with different resolutions, we introduce the weights $\{w_i | w_i = (\bar{d}'_i)^{-2}\}$ that are applied to each term of the summation in (1), i.e., $D(\mathcal{L}'_M, \mathcal{T}\mathcal{L}_M) = \sum_n w_n[\cdots]$.

3.2. Building a Fast Search Codebook

Given M randomly selected lines in $\mathcal{L}$, we should test all possible $O(L'^M)$ groups of lines in $\mathcal{L}'$ to estimate the transformation. Using a blind *hypothesis-and-test* approach would require testing all possible $O(L^M L'^M)$ groups of lines, which is impractical due to the size of the search space. Given a pair of lines $(\mathbf{l}_i, \mathbf{l}_j)$ in $\mathcal{L}$, we should check $O(L'^2)$ pairs of lines in $\mathcal{L}'$. To speed up the test, we first compute the line orientation angle $\theta(\mathbf{l}_i, \mathbf{l}_j) = \text{acos}(V_i^\top V_j)$ and line-line distance $d(\mathbf{l}_i, \mathbf{l}_j)$. These two measures are invariant under rigid transformation. For a pair of near parallel lines $(\mathbf{l}_i, \mathbf{l}_j)$ with $\|\mathbf{l}_i\| \leq \|\mathbf{l}_j\|$, the line-line distance must be in the range $[0, d(\mathbf{l}_i, P_j)]$, where P_j is the midpoint of the line $\mathbf{l}_j$ and $d(\mathbf{l}, P)$ denotes the shortest distance of the point P to the line $\mathbf{l}$. To make the line-line distance a robust comparable measurement, we define the distance of two near parallel lines as $d(\mathbf{l}_i, \mathbf{l}_j) = d(\mathbf{l}_i, P_j)$. In addition, due to the fact that we do not know the line orientation, the orientation angle between two lines is converted into the range $[0, \pi/2]$.

For each pair of lines $(\mathbf{l}_i, \mathbf{l}_j)$ in $\mathcal{L}$, we can search all possibly matched pairs of lines $\{(\mathbf{l}'_{m_k}, \mathbf{l}'_{n_k})\}_{k=1}^K$ in $\mathcal{L}'$ satisfying two conditions: (1) $|\theta(\mathbf{l}_i, \mathbf{l}_j) - \theta(\mathbf{l}'_{m_k}, \mathbf{l}'_{n_k})| < th_\theta$ and (2) $|d(\mathbf{l}_i, \mathbf{l}_j) - d(\mathbf{l}'_{m_k}, \mathbf{l}'_{n_k})| < th_d$. We call these two conditions as *rough line matching constraint*. The threshold th_θ is straightforward and can be fixed for scans under different resolutions. The threshold th_d is defined as $th_d = \beta \max(\bar{d}_i, \bar{d}_j, \tilde{d}')$ where $\tilde{d}' = (\sum_{m=1}^{L'} |\mathcal{X}'_m| \bar{d}'_m) / \sum_{m=1}^{L'} |\mathcal{X}'_m|$. In this way, we can fix the parameter β to adaptively compute th_d, which is robust for matching two scans under different resolutions. If the pair $(\mathbf{l}'_m, \mathbf{l}'_n)$ in $\mathcal{L}'$ satisfies the rough line matching constraint, the pair $(\mathbf{l}'_n, \mathbf{l}'_m)$ satisfies it too. By the blind exhaustive search, we want to compare them with $\frac{L' \times (L'-1)}{2}$ pairs of lines in $\mathcal{L}'$. To reduce the search time, we build a codebook for $\mathcal{L}'$ as follows. The range $[0, \pi/2]$ of line orientation angles and the range $[d'_{\min}, d'_{\max}]$ of line-line distances are uniformly divided into B_θ and B_d bins respectively where $d'_{\min}$ and $d'_{\max}$ denote the minimal and maximal line-line distances of line pairs in $\mathcal{L}'$, respectively. Given a pair of lines $(\mathbf{l}_i, \mathbf{l}_j)$ in $\mathcal{L}$, we first compute the bin ranges $[B^s_\theta, B^e_\theta]$ and $[B^s_d, B^e_d]$ of both line orientation angles and line distances and then find desired pairs in these bin ranges.

The codebook is designed to minimize the number of pairs per bins. This reduces the search time but increases the required memory to store the codebook.

Moreover, to reduce the transformation estimation time, we only keep the best K_b pairs of lines in $\mathcal{L}'$ ($K_b = 100$ in our experiments), which have high costs: $|\theta(\mathbf{l}_i, \mathbf{l}_j) - \theta(\mathbf{l}'_{m_k}, \mathbf{l}'_{n_k})|/th_\theta + |d(\mathbf{l}_i, \mathbf{l}_j) - d(\mathbf{l}'_{m_k}, \mathbf{l}'_{n_k})|/th_d$.

Algorithm 1 describes the method of finding candidate groups of M lines in $\mathcal{L}'$ possibly matching M lines in $\mathcal{L}$.

3.3. Matching 3D Line Sets

To find the best set of matched lines between $\mathcal{L}$ and $\mathcal{L}'$, we directly use the RANSAC algorithm. In each iteration, we randomly select M lines in $\mathcal{L}$ that are not all parallel and then find all possibly matched groups of lines in $\mathcal{L}'$ using Algorithm 1. For each matched group of M lines in $\mathcal{L}'$, we estimate the transformation $\mathcal{T}$ using the closed-form solution [8].

Algorithm 1 Find groups of lines in $\mathcal{L}'$ possibly matching M lines $\mathcal{L}_M = \{\mathbf{l}_n\}_{n=1}^M$ in $\mathcal{L}$.

1. Set $i = 1, \mathcal{M} = \emptyset$;
2. Find the best K_b pairs of lines possibly matching the pair of lines $(\mathbf{l}_i, \mathbf{l}_{i+1})$ in $\mathcal{L}_M$ and put them in a set $\mathcal{S}_i$.
3. If $i = 1, \mathcal{M} = \mathcal{S}_i$.
4. Otherwise, set a new set $\mathcal{M}' = \emptyset$. Then check each group of lines in $\mathcal{M}$ and each pair of lines in $\mathcal{S}_i$. If there is a common line among them satisfying the rough line matching constraint, a new group of $(i+1)$ lines would be added into $\mathcal{M}'$.
5. Set $\mathcal{M} = \mathcal{M}'$ and $i = i + 1$, continue (go to 2) if $i < M$ and $\mathcal{M} \neq \emptyset$.
6. Otherwise, exit with the found set $\mathcal{M}$.

The transformation is regarded as a good estimation if each pair of matched lines $(\mathbf{l}, \mathbf{l}')$ satisfies:

$$\begin{cases} \max(d(\mathcal{T}P_s, \mathbf{l}'), d(\mathcal{T}P_e, \mathbf{l}')) \leq \gamma_1 \bar{d}', \\ \max(d(\mathcal{T}'P'_s, \mathbf{l}'), d(\mathcal{T}'P'_e, \mathbf{l})) \leq \gamma_1 \bar{d}, \end{cases} \tag{3}$$

where (P_s, P_e) and (P'_s, P'_e) are the endpoints of the lines $\mathbf{l}$ and $\mathbf{l}'$ respectively, $\mathcal{T}'$ is the inverse transformation of $\mathcal{T}$, $\bar{d}$ and $\bar{d}'$ are the median adjacent distances of the lines $\mathbf{l}$ and $\mathbf{l}'$ respectively, and γ_1 is the parameter for the endpoint deviation measure. If a good estimation is found, we further refine the transformation using the iterative optimization algorithm as described in Section 3.1 and check each pair of matched lines using a smaller endpoint deviation parameter $\gamma_2 < \gamma_1$. We call the above conditions as *precise line matching constraint*. Notice that most of the mismatched groups will be filtered out by the first check process using the closed-form solution. After a reliable initial transformation is found if the precise matching constraint is satisfied, we verify the remaining lines in $\mathcal{L}$ and find matched lines in $\mathcal{L}'$ by first checking the rough line matching constraint with initially matched lines and then checking the precise line matching constraint. Then we recompute the transformation using the iterative optimization algorithm based on all found matches. After some iterations the transformation with most line matches is found.

4. EXPERIMENTS AND CONCLUSIONS

Our proposed automated registration method was tested on a set of real data gathered by a Z+F IMAGER 5003 laser range-scanner from different scenes. The accuracy of a point is 3mm along the laser-beam direction at a maximal distance of 79m from the scanner. In total, we tested 4 groups of scans, which consist of 4 scans, 4 scans, 3 scans and 2 scans respectively. Each group of scans was gathered from the same scene in which each pair of scans contains an overlapping region. The two threshold parameters for the rough line matching constraint were set as $th_\theta = 2.5^o$ (i.e. $\frac{2.5}{180}\pi$) and $\beta = 5$ used for adaptively computing th_d. The two parameters for the precise line matching constraint were set as $\gamma_1 = 5$ and $\gamma_2 = 2.5$. The same parameters were used in all the experiments described in this paper.

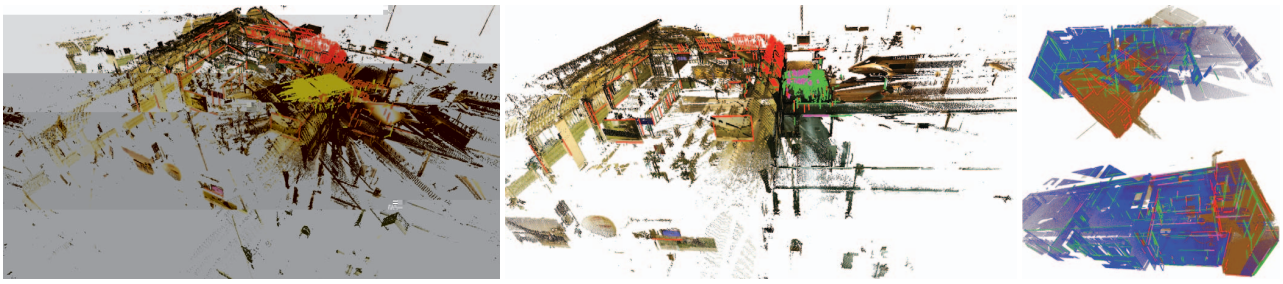

Fig. 1. Line matching results of two pairs of scans (Left: before matching, Middle: after matching, Right: top before matching and bottom after matching): *Red* lines related to first scan, *green* lines related to second scan. Matched lines are shown in *purple*.

Table 1 provides a detailed evaluation of the efficiency and accuracy of our algorithm on the four groups of scans. The original resolution of all these scans is 5098×2166. However, we subsampled input data with a subsampling rate of 2 for *group 1* and with a subsampling rate of 4 for *groups 2* and *3* [1]. In *group 4* we used one scan without subsampling and another scan with a subsampling rate of 4. The line matching execution time ranges from 3 to 833 seconds (less than 1 minute on average), mainly depending on the numbers of lines, the amount of matched lines, and the number M of randomly selected lines for estimating initial transformation. Time includes the computation of orientation angles and distances of all line pairs for two scans to be matched, and the building of the searching codebook (which can be computed once for each scan in practice), and the RANSAC iteration time for line matching. However, it does not include the edge detection and line extraction steps, which took around 2 to 5 seconds for each scan. The algorithms were written in Matlab without optimization and executed on a 2.67GHz Intel machine. By refining the obtained rigid transformation, the error was reduced under 3mm with a few iterations of the global point-based ICP algorithm [1]. From Table 1 we observe that $M = 3$ produced more robust transformations than $M = 2$ (i.e. similar or smaller average point distances $D_{\mathcal{T}}$ in most cases) and always took less times to perform the RANSAC step. Most of average point distances $D_{\mathcal{T}}$ are within 10mm, which was enough to provide a good initial transformation for further ICP optimization. The line matching results of two pairs of scans are shown in Fig. 1 where the first pair of scans was acquired in a Paris train station and their overlap is very small, and the second pair of scans was acquired in an indoor lab.

In conclusion, we presented an efficient line detection and line matching algorithm for the automated registration of two range scans. Our line detection algorithm considers both 3D edges and 2D intensity edges extracted from 2D reflectance images. The line matching algorithm exploits only positions and orientations of the extracted lines. Finally we provided experiments that show good registration results.

G	Lines	N_m		t (secs)		$D_{\mathcal{T}}$ (mm)		D_{ICP} (mm)	
		M=2	M=3	M=2	M=3	M=2	M=3	M=2	M=3
1	514x393	68	128	93	19	7.49	5.20	5.59	2.30
	514x424	71	68	138	41	9.25	6.13	2.95	2.99
	514x511	76	67	435	61	10.65	9.55	3.01	3.02
	393x424	125	116	51	20	3.85	4.37	2.00	2.11
	393x511	42	42	335	82	14.38	8.94	3.56	3.54
	424x511	22	22	833	72	161.6	89.91	3.31	3.64
2	179x231	31	32	59	46	4.16	6.94	2.73	2.72
	179x131	26	28	93	14	5.51	5.86	3.93	3.83
	179x139	54	55	32	6	4.46	4.50	3.30	3.21
	231x131	46	50	56	8	4.68	4.86	2.91	2.83
	231x139	27	28	58	26	6.96	7.15	2.71	2.72
	131x139	52	50	16	3	4.49	4.26	2.34	2.21
3	269x516	35	77	157	39	13.23	3.42	1.64	1.67
	269x494	47	42	93	61	4.01	5.84	2.27	2.25
	516x494	109	166	132	34	8.67	1.79	2.52	1.35
4	317x136	19	40	98	71	28.97	4.54	2.35	2.40

Table 1. Performance evaluation of our algorithm. Meaning of columns: G - scan groups number; Lines - numbers of lines extracted from the two input scans; N_m - numbers of best line matches; t - line matching times (in secs) of the Matlab implementation; $D_{\mathcal{T}}$ - average point distances (in mm) from best transformation of line matching; D_{ICP} - average point distances (in mm) after ICP optimization. Note that M denotes the number of randomly selected lines for initial transformation estimation in RANSAC.

5. REFERENCES

[1] P. J. Besl and H. D. McKay, "A method for registration of 3D shapes" in *IEEE PAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[2] C. C. Chen and I. Stamos, "Range image registration based on circular features", *3DPVT*, 2006.

[3] I. Stamos and M. Leordeanu, "Automated feature-based range registration of urban scenes of large scale", *CVPR*, 2003.

[4] C. Chao and I. Stamos, "Semi-automatic range to range registration: a feature-based method", *3DIM*, 2005.

[5] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface feature detection and description with applications to mesh matching", *CVPR*, 2009.

[6] L. Wang, S. You, and U. Neumann, "Supporting range and segment-based hysteresis thresholding in edge detection", *ICIP*, 2008.

[7] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics" in *Autonomous Robots*, vol. 23, no. 2, pp. 97–111, 2007.

[8] Z. Zhang and O. D. Faugeras, "Determining motion from 3d line segment matches: a comparative study" in *Image and Vision Computing*, vol. 9, no. 1, pp. 10–19, 1991.

[9] B. Kamgar-Parsi, "Algorithms for matching 3d line sets" in *IEEE PAMI*, vol. 26, pp. 582–593, 2004.

[1] A subsampling rate of n means that only one point of every $n \times n$ grid points will be used.