A 3D Segments Based Algorithm for Heterogeneous Data Registration

Rahima Djahel, Pascal Monasse, Bruno Vallet

A 3D SEGMENTS BASED ALGORITHM FOR HETEROGENEOUS DATA REGISTRATION

Rahima Djahel[1]*, Pascal Monasse[1], Bruno Vallet[2]

[1] LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS , F-77454 Marne-la-Vallée, France -firstname.lastname@enpc.fr
[2] LASTIG, Univ Gustave Eiffel, IGN-ENSG, F-94160 Saint-Mandé, France - firstname.lastname@ign.fr

KEY WORDS: 3D segments, 3D reconstruction, Clustering, RANSAC, Simulated Annealing, Registration. Heterogeneous data.

ABSTRACT:

Combining image and LiDAR draws increasing interest in surface reconstruction, city and building modeling for constructing 3D virtual reality models because of their complementary nature. However, to gain from this complementarity, these data sources must be precisely registered. In this paper, we propose a new primitive based registration algorithm that takes 3D segments as features. The objective of the proposed algorithm is to register heterogeneous data. The heterogeneity is both in data type (image and LiDAR) and acquisition platform (terrestrial and aerial). Our algorithm starts by extracting 3D segments from LiDAR and image data with state of the art algorithms. Then it clusters the 3D segments of each data according to their directions. The obtained clusters are associated to find possible rotations, then 3D segments from associated clusters are matched in order to find the translation and scale factor minimizing a distance criteria between the two sets of 3D segments. Two optimizers (simulated annealing and RANSAC) are tested to minimize this distance criterion, first on synthetic data, then on real data. The experiments carried out demonstrate the robustness and speed of RANSAC compared to simulated annealing.

1. INTRODUCTION

1.1 Context

The volume of datasets acquired by optical and LiDAR systems is rapidly increasing due to demand from a variety of applications including remote sensing, 3D mapping and autonomous driving. 3D scene analysis and reconstruction from image and LiDAR is an active research area in computer vision. On the one hand, the LiDAR data provides highly accurate and robust surface information. On the other hand, the image provides high resolution details and rich spectral information Kim et al. (2006) but the 3D geometry estimated from dense marching is less robust and accurate, in particular in homogeneous, specular or repetitive regions and near depth discontinuities. Hence, integrating data from these two sources can lead to a more robust and complete semantic segmentation and reconstruction of 3D scenes. Following a classical methodology Kumar Mishra and Zhang (2012), the problem of registration is decomposed in three main steps: (i) Feature extraction, (ii) Feature matching, (iii) Transformation model estimation.

1.2 State of the art

There is a considerable amount of prior work on image/LiDAR registration. We start by classifying some existing methods according to their types and attributes, citing the advantages and limitations of each category.

1.2.1 Keypoint based method Corners are widely used keypoints for image/image registration, which can be extended to image/LiDAR registration Ding et al. (2008). The uniqueness and high precision in the localization are the strongest properties of corners Kumar Mishra and Zhang (2012). However, corners are not always easily matched between LiDAR and image.

1.2.2 Linear feature based method Straight lines are the most used feature for image/LiDAR registration because they can be automatically, accurately and efficiently extracted form the LiDAR and image data Habib et al. (2006) especially if we have a set overlapping images Deng et al. (2008).

1.2.3 Structural feature based method High level structural features such as rectangles can help increasing the robustness of both detection and matching steps for image/LiDAR registration. These structural features can be extracted from both data sets as connected segments Liu and Stamos (2012). This kind of features can be efficiently used to estimate camera translation. Camera rotation can be estimated using at least two vanishing points. The major limitation of these methods consists in the dependence on either the strong presence of parallels lines to infer vanishing points or availability of feature pair correspondences Wang and Ferrie (2015). Moreover, these methods are not efficective when applied to aerial data.

1.2.4 Mutual information based method Statistical and information theoretic methods have demonstrated excellent performance for a wide variety of 2D-2D and 2D-3D registration applications. Mutual information, a central concept of information theory, consists in measuring the statistical correlation of two random variables, which is a measure of the amount of information one random variable contains about the other Cover (1999). Recent methods use mutual information(MI) as a statistical metric to register image and LiDAR data Lu et al. (2019). A mutual information (MI) approach was proposed in Mastin et al. (2009). This method performs the registration by seeking the camera matrix that maximizes the MI between the distribution of image features and projected LiDAR features. The authors of Wang and Ferrie (2015) search for the optimal camera pose through maximizing the MI between camera image and LiDAR attribues (LiDAR intensity image, LiDAR elevation image) Wang and Ferrie (2015). The major limitation of

* Corresponding author

these methods lies in the decreasing registration accuracy when only LiDAR elevation image is used.

1.2.5 Frequency Based Registration The most popular frequency based method is phase correlation. Due to differences in data characteristics, these frequency based methods cannot be applied directly to register images and LiDAR data. Zhu et al. (2020) has proposed a new registration method to deal with the problem of aerial image and LiDAR registration. This method is based on structural features and 3D phase correlation, in order to address significant geometric distortions and nonlinear intensity differences between the aerial and LiDAR intensity images. The 3D phase correlation is used to detect control points (CPs) between aerial images and LiDAR data in the frequency domain, that will be used to correct the exterior orientation elements. The main limitation of this method lies in the use of the intensity images of LiDAR only, which contains less information about the topography.
After this simple comparison, we advocate that the best methods to apply for image/LiDAR registration in urban environments are mutual information and straight lines based method. We have chosen to develop a straight lines based algorithm for the following reasons:

- In building environment, most objects are composed of planar surfaces delimited by straight lines.
- On our scenes of interest, we have both overlapping images and LiDAR scans
- There are efficient algorithms in the literature to automatically and precisely extract these features from both images and LiDAR data.

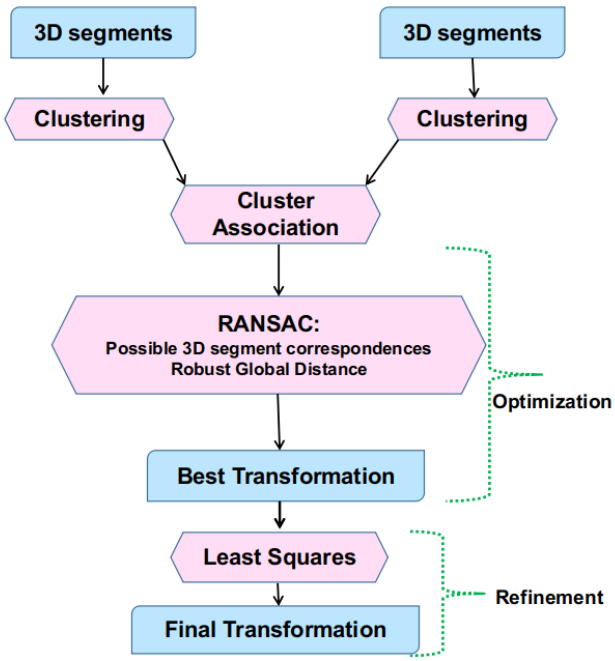


Figure 1. Pipeline details.

1.3 Overview and contributions

In a man made environment, we advocate that the best feature to use for image/LiDAR registration is the straight line segment. Straight line segments can be reliably, accurately and automatically extracted from both LiDAR and image data under sufficient image overlap. They aggregate more information than points, therefore they are less sensitive to dense noise and are more frequent than more complex primitives (rectangles).

In this paper, we propose a new method for image/LiDAR registration, that consists in:

- Feature extraction:
  - 3D line segments reconstruction from a set of overlapping 2D images.
  - 3D line segments detection in LiDAR data.
- Feature matching: in the absence of multi-modal descriptors for lines extracted from imagery and LiDAR, all pairs of lines may match a priori. We thus rely on randomized matching of line segments (RANSAC and simulated annealing), which is made possible by the relatively limited number of lines to be matched.
- Estimating the geometric transformation (scaling-rotation-translation) that minimizes the distance between the two segments sets . At each step of our optimization, we use a quality criterion to evaluate the corresponding transformation and select the best one in order to be fully robust to initialization as shown in 1

Our contributions are the following:

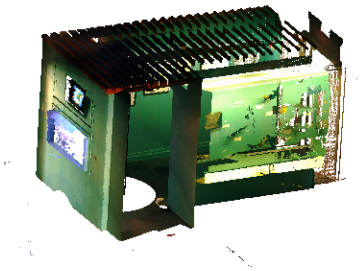
1. We define a new distance between 3D line segments that combines both their overlap, their length and their 3D distance, while being robust to long line fragmentation.
2. We show that clustering the segments according to their direction helps generating plausible hypotheses for candidate motion registration.
3. We show that a RANSAC procedure can efficiently minimize the error criterion, better than simulated annealing, and provide an accurate registration between images and LiDAR.
4. We show that our algorithm is robust to initialization.
5. We propose an optional way to exploit the vertical direction (if it is known for both data sources) to accelerate the optimization by limiting the search space.

2. 3D SEGMENT EXTRACTION

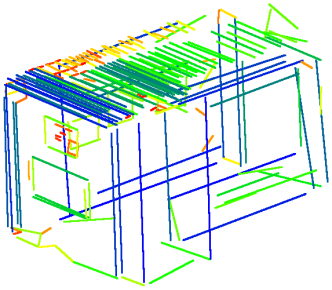
2.1 3D Line Segment Extraction from LiDAR Data

3D line segment detection is a crucial step in our registration procedure. According to Lu et al. (2019), this problem can be classified into three categories:

- Point based methods use least square fitting of 3D line segments after detecting the boundary points. The main problem of this kind of method is the non-robustness to the noise.
- Plane based methods: a 3D line segment can be generated by the intersection of two 3D planes. The drawback is that the endpoints of the intersection line are generally difficult to determine.



(a) Indoor point cloud



(b) Extracted 3D lines

Figure 2. 3D line segments detection from an indoor scan

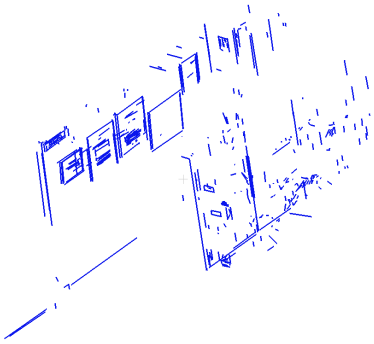
- Image based methods start by converting the 3D point cloud into images. They then extract 2D line segments for each image. Finally, the 2D line segments are reprojected to the point cloud to get the final 3D line segments. The difficulty of this kind of method is that the adequate resolution is hard to determine.

In this work, we have chosen the algorithm proposed in Lu et al. (2019) to extract 3D line segments from the LiDAR data. It is a simple and efficient algorithm that starts by segmenting the point cloud into planar 3D regions via region growing and merging. All the points belonging to the same planar region are projected into the supporting plane of this region to form a 2D image. Then, 2D contour extraction and least square fitting are performed to detect 2D line segments. Finally, these 2D line segments are transformed back into the 3D frame to get the 3D segments.

2.2 3D line segment extraction from image data

3D line segments reconstruction from 2D images has received continued attention in the photogrammetry and computer vision literature. The first proposed methods used pairwise reconstruction where only two overlapping images are required to perform the reconstruction. Other methods have recently been proposed which use multi-view reconstruction where various overlapping images are used and the camera pose can be obtained using an SfM (Structure from Motion) pipeline. In this work, we chose the approach developed in Hofer et al. (2017), which uses an oriented image sequence as input. The camera pose can be obtained by any conventional SfM pipeline. 2D line segments are detected using the LSD algorithm Von Gioi et al. (2012). Potential matches between the 2D line segments from different images are evaluated using a scoring formulation based on mutual support to separate correct form incorrect segment matches. 2D segments from different views are clustered using an efficient graph clustering formulation to obtain clusters of corresponding 2D segments. For each cluster,

(a) An image from the 2D image sequence



(b) The reconstructed lines

Figure 3. Reconstructed 3D line cloud

a 3D segment minimizing reprojection error with the 2D segments of the cluster is reconstructed. We used the code available on `https://github.com/manhofer/Line3Dpp` to generate the 3D lines. To improve the performance of this algorithm we have replaced LSD by its more robust variant MLSD Salaün et al. (2016).

3. 3D SEGMENTS BASED REGISTRATION

Registration is usually performed either based on a global distance metric between the two data sets to register, or by matching features. In our case, the line segments are not characteristic enough to match them robustly independently. This is why we preferred to define a global robust distance between two segment sets and propose a robust approach to minimize this distance.

3.1 Distance between segment sets

We propose to define a distance between 3D segment sets in a way that minimizing this distance will favor important overlaps between segments and small distances over these overlaps while being robust to outliers, as many lines extracted form one data set will have no counterpart in the other. This will implicitly also minimize angles between the segments as an important angle with an important overlap implies a large distance. Given two sets of 3D segments $[A_i^1 B_i^1]$ and $[A_j^2 B_j^2]$ with center points $G_i^k = (A_i^1 + B_i^1)/2$ and direction vectors $\mathbf{d}_i^k = \overrightarrow{A_i^k B_i^k}/||\overrightarrow{A_i^k B_i^k}||$, we start by defining a relative overlap between two segments over the projection p on the bisector of the two segments:

$$overlap([A_i^1 B_i^1], [A_j^2 B_j^2]) = \frac{|p([A_i^1 B_i^1]) \cap p([A_j^2 B_j^2])|}{min(|p([A_i^1 B_i^1])|, |p([A_j^2 B_j^2])|)}$$

We define this bisector as the line passing through the closest point $P_{i,j}$ to $(A_i^1 B_i^1)$ and $(A_j^2 B_j^2)$ and oriented by the bisector

$\mathbf{v}_{i,j}$ of $\mathbf{d}_i^1$ and $\mathbf{d}_j^2$. Calling

$$[\mathbf{v}]_\times = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix} \text{ such that } [\mathbf{v}]_\times \mathbf{u} = \mathbf{v} \times \mathbf{u}, \tag{1}$$

$\times$ denoting the 3D vector product, we have

$$P_{i,j} = \left([\mathbf{d}_i^1]_\times^2 + [\mathbf{d}_j^2]_\times^2\right)^{-1} \left([\mathbf{d}_i^1]_\times^2 A_i^1 + [\mathbf{d}_j^2]_\times^2 A_j^2\right) \tag{2}$$

The projection on the bisector is

$$p(P) = P_{i,j} + c(P)\mathbf{v}_{i,j}, \tag{3}$$

with the projected point's linear coordinate

$$c(P) = \overrightarrow{P_{i,j}P}.\mathbf{v}_{i,j}. \tag{4}$$

The length of the projected segment is then defined to be oriented as

$$|p([A_i^1 B_i^1])| = max(c(A_i^1), c(B_i^1)) - min(c(A_i^1), c(B_i^1)) \tag{5}$$

and the overlap length of the projected segments as:

$$\begin{gathered} |p([A_i^1 B_i^1]) \cap p([A_j^2 B_j^2])| = \\ min(max(c(A_i^1), c(B_i^1)), max(c(A_j^2), c(B_j^2))) \\ -max(min(c(A_i^1), c(B_i^1)), min(c(A_j^2), c(B_j^2))) \end{gathered} \tag{6}$$

We then define a distance between a segment $L_1 \in \mathcal{L}_1$ and a segment set $\mathcal{L}_2$:

$$E^{d_{thr}}(L_1, \mathcal{L}_2) =$$

$$|L_1^2| d_{thr}^2 - \sum_{L_2 \in \mathcal{L}_2} |L_1^2 \cap L_2^1| max(0, d_{thr}^2 - Dist(L_1, L_2)^2) \tag{7}$$

where d_{thr} is a distance threshold above which a line is just considered an outlier and $Dist$ is a metric distance between segments; our choice is

$$Dist([A_i^1 B_i^1], [A_j^2 B_j^2]) = \tag{8}$$

$$\frac{dist(A_i^1, [A_j^2 B_j^2]) + dist(B_i^1, [A_j^2 B_j^2])}{4} + \tag{9}$$

$$\frac{dist(A_j^2, [A_i^1 B_i^1]) + dist(B_j^2, [A_i^1 B_i^1])}{4}. \tag{10}$$

For each line pair, this distance vanishes if L_1 is completely overlapped by segments of $\mathcal{L}_2$. Finally we can write our final symmetric robust distance between 3D segments sets as:

$$dist(\mathcal{L}_1, \mathcal{L}_2) = \sum_{L_1 \in \mathcal{L}_1} E^{d_{thr}}(L_1, \mathcal{L}_2) + \sum_{L_2 \in \mathcal{L}_2} E^{d_{thr}}(L_2, \mathcal{L}_1) \tag{11}$$

3.2 3D segments directional clustering

In order to simplify the minimization of this distance between segments sets, we cluster the segments (by direction) in order to define the rotation between the segments sets by matching clusters and not individual segments in order to be more robust and precise. This is done using a greedy algorithm described in Algorithm 1.

Algorithm 1 Greedy direction clustering

1: Input: Set of segments L, each segment $L_i = [A_i B_i]$ has a director vector $\mathbf{v}_i = \overrightarrow{A_i B_i}$, a length $len_i = ||\mathbf{v}_i||$ and a unit direction $\mathbf{d}_i = \mathbf{v}_i / len_i$.
2: Initialize an empty set of 3D segment clusters $\mathcal{C}$. We will call direction of a cluster C the weighted mean of the directions of the 3D segments:

$$\mathbf{d}(C) = \frac{\sum_{L_i \in C} sign(\mathbf{v}_i \cdot \mathbf{v}_1)\mathbf{v}_i}{||\sum_{L_i \in C} \mathbf{v}_i||}$$

3: For each segment L_i in descending order of length (from longest to shortest):

- If $\mathcal{C} = \emptyset$ or $\max_{C \in \mathcal{C}} \mathbf{d}_i \cdot \mathbf{d}(C) < \cos\epsilon$, create a new cluster $C = \{L_i\}$ and add it to $\mathcal{C}$. Note that this is equivalent to finding if L_i has an angle smaller than ϵ with an existing cluster direction but this formulation with dot products is faster to compute.
- Else add L_i to the cluster $\arg\max \mathbf{d}_i \cdot \mathbf{d}(C)$.

3.3 Valid cluster associations

To define a rotation between the two segment sets, it is sufficient to associate two pairs of segments from the two sets. However this is not very precise, so we propose to associate the segment clusters defined above. We propose two possible sets of valid associations depending on whether we have a vertical cluster for each scan.

3.3.1 LiDAR vertical cluster selection We assume that all LiDAR scans are vertically oriented (this is the case for most modern scanners) and select the cluster with the smallest angle with the Z-axis $(0, 0, 1)$ as being the vertical cluster.

3.3.2 Image vertical cluster selection We assume that the images are upright (the real world vertical projects as a nearly vertical 2D vector in all images) and select the vertical cluster of the line cloud reconstructed from the images according to Algorithm 2

Algorithm 2 Image vertical cluster

for each cluster i **do**
  for each 3D line j in cluster i **do**
    for each 2D image im associated with line j **do**
      Compute S_{im} the verticality score weighted by the length of the 2D segment $l_i = [a_i b_i]$ in the image im that has been used for the reconstruction of line j using the method proposed in section 3.3.3
    end for
    Compute av_j: the average verticality score for all images associated with line j
  end for
  Compute AV_i: the average verticality score for all 3D lines in cluster i
end for
Select the cluster which has the smallest score as the vertical cluster

3.3.3 Verticality score calculation We have an image im which contains 2D segment $l_i = [a_i b_i]$. This 2D segment l_i has a director vector $v_i = \overrightarrow{a_i b_i}$ and a unit direction $d_i = v_i / ||v_i||$. We compute S_{im} the verticality score weighted by the length of the 2D segment as the angle between d_i and y-axis $(0, 1)$ according to the equation 12:

$$im = ||v_i|| \langle d_i, y \rangle \tag{12}$$

If the two segment sets $\mathcal{S}_1$ and $\mathcal{S}_2$ (image and/or LiDAR) can be vertically oriented (the assumptions above are true), we obviously associate the vertical cluster of $\mathcal{S}_1$ to the vertical cluster of $\mathcal{S}_2$, and then associate any non vertical cluster of a $\mathcal{S}_1$ with any non vertical cluster of $\mathcal{S}_2$. In the other case (at least one scan cannot be vertically oriented), we associate any pair of clusters of $\mathcal{S}_1$ to any pair of clusters of $\mathcal{S}_2$. As for each cluster we have two possible direction vectors:$\{\mathbf{d}(C), -\mathbf{d}(C)\}$, we can:
define the variables $s_i^j : i \in [1,2], j \in [1,2], s_i^j \pm 1$

Calling C_1^1, C_2^1 a pair of clusters of $\mathcal{S}_1$ and C_1^2, C_2^2 a pair of clusters of $\mathcal{S}_2$. In the two cases, for each cluster associations $As = \{(C_1^1, C_2^1) \leftrightarrow (C_1^2, C_2^2)\}$, we have sixteen possible forms. If one of these sixteen forms satisfies equation 13, we consider that this association is valid, we use the retained form to calculate the rotation. We reject the cluster association, if the equation 13 is not validated by any form.

$$|\angle(s_1^1\mathbf{d}(C_1^1), s_2^1\mathbf{d}(C_2^1)) - \angle(s_1^2\mathbf{d}(C_1^2), s_2^2\mathbf{d}(C_2^2))| > \epsilon \quad (13)$$

3.4 Rotation estimation

Once a the best form of the valid cluster association $As = \{(s_1^1 C_1^1, s_2^1 C_2^1) \leftrightarrow (s_1^2 C_1^2, s_2^2 C_2^2)\}$ is selected, we want to find the rotation that best aligns the corresponding two pairs of directions. After selecting the directing vectors of two clusters. We start by creating orthonormal bases $\mathcal{O}_i = (\mathbf{x}^i, \mathbf{y}^i, \mathbf{z}^i)$, where:

$$\mathbf{x}^i = s_1^i\mathbf{d}(C_1^i)$$
$$\mathbf{y}^i = \frac{s_2^i\mathbf{d}(C_2^i) - s_2^i\mathbf{d}(C_2^i)\cdot\mathbf{x}^i\mathbf{x}^i}{||s_2^i\mathbf{d}(C_2^i) - s_2^i\mathbf{d}(C_2^i)\cdot\mathbf{x}^i\mathbf{x}^i||}$$
$$\mathbf{z}^i = \mathbf{x}^i \times \mathbf{y}^i \quad (14)$$

We then compute the rotation R that aligns the associated clusters as the base change matrix between $\mathcal{O}^1$ and $\mathcal{O}^2$:

$$R = \mathcal{O}_2\mathcal{O}_1^{-1}. \quad (15)$$

3.5 Optimization

To carry out the registration of the two segment sets, we now have to find the rotation, the scale factor and the translation that minimize the distance (11) we have defined between two 3D segment sets. We want this minimization to be insensitive to initialization, so we propose two randomized optimizers: simulated annealing and RANSAC.

3.5.1 Simulated annealing optimization We use simulated annealing to explore the translation and scale parameters. In simulated annealing, a new solution is iteratively computed in the vicinity of the current solution and this new solution is accepted with a certain probability depending on its energy (the robust distance in our case). However for rotations we want to speed up the process by only exploring valid cluster associations to limit the complexity of the search, see Algorithm 3.

3.5.2 RANSAC optimization The adaptation of RANSAC to valid associations is quite straightforward. At each RANSAC iteration, we randomly select a valid cluster association, then randomly select one 3D segment in each of the associated clusters. If the distance between the supporting lines of two segments is smaller than a threshold d_{min} chosen to be the expected noise level in the corresponding scan, we reject it because matching coplanar segments will lead to degenerate scale estimation. We then compute the rotation based on cluster association using the method of Section 3.4 and estimate the scale/translation that aligns the associated 3D segments. To do so, we define the point to line distance as

$$dist(\mathbf{p}, L = \mathbf{a} + \mathbf{d}t) = \frac{||(\mathbf{a} - \mathbf{p}) \wedge \mathbf{d}||}{||\mathbf{d}||}. \quad (16)$$

Assuming that $\mathbf{d}$ is normalized, this writes

$$dist(\mathbf{p}, L = \mathbf{a} + \mathbf{d}t) = \|[\mathbf{d}]_\times(\mathbf{a}_i - (s\mathbf{p}_i + \mathbf{t}))\|. \quad (17)$$

Our goal is to find the optimal translation $\mathbf{t}$ and scale s that minimize

$$\epsilon(s, \mathbf{t}) = \sum_i ||[\mathbf{d}]_\times(\mathbf{a}_i - (s\mathbf{p}_i + \mathbf{t}))||^2. \quad (18)$$

The minimum is reached where the gradient vanishes:

$$\nabla_\mathbf{t}\epsilon(s, \mathbf{t}) = 2\sum_i [\mathbf{d}_i]_\times^2(\mathbf{a}_i - (s\mathbf{p}_i + \mathbf{t})) = 0, \quad (19)$$

$$\nabla_s\epsilon(s, \mathbf{t}) = 2\sum_i \mathbf{p}_i^t[\mathbf{d}_i]_\times^2(\mathbf{a}_i - (s\mathbf{p}_i + \mathbf{t})) = 0. \quad (20)$$

Calling:

$$\mathbf{w}_1 = \sum_i [\mathbf{d}_i]_\times^2\mathbf{a}_i \quad M_1 = \sum_i [\mathbf{d}_i]_\times^2 \quad M_2 = \sum_i [\mathbf{d}_i]_\times^2\mathbf{p}_i$$

Algorithm 3 Simulated Annealing

Input:
M_0: initial transformation (randomly chosen valid cluster association and randomly chosen scale/translation)
E_0: initial transformation energy
G: Current transformation
T_{max}: initial temperature
T_{min}: final temperature
$alpha$: cooling rate
$MaxIter$: maximum number of iterations
$G = M_0$, $E(G) = E_0$
while $T_{max} > T_{min}$ **do**
  for $Iter = 1 \ldots$ MaxIter **do**
    With a small probability p_{jump}, randomly select a new valid cluster association and corresponding rotation. Randomly sample a new scale/translation close to the current one, and call M_N the resulting transformation.
    Compute $\Delta_E = E(M_N) - E_0$
    if $(\Delta_E < 0)$ or $(random < e^{\frac{-\Delta_E}{T_{max}}})$ **then**
      $M_0 = M_N$
      $E_0 = E(M_N)$
    end if
    if $(E_0 < E(G))$ **then**
      $G = M_0$
      $E(G) = E_0$
    end if
  end for
  $T_{max} = \alpha * T_{max}$
end while

$$\mathbf{w}_2 = \sum_i \mathbf{p}_i^t [\mathbf{d}_i]_\times^2 \mathbf{a}_i = -\sum_i (\mathbf{d}_i \times \mathbf{p}_i) \cdot (\mathbf{d}_i \times \mathbf{a}_i)$$
$$M_3 = \sum_i \mathbf{p}_i^t [\mathbf{d}_i]_\times^2 \mathbf{p}_i = -\sum_i \|\mathbf{d}_i \times \mathbf{p}_i\|^2$$

we get

$$\begin{bmatrix} \mathbf{t} \\ s \end{bmatrix} = \begin{bmatrix} M_1 & M_2 \\ M_2^t & M_3 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}. \quad (21)$$

Finally, we keep the sampled transformation that has the minimum robust distance. Optionally, we can refine this solution by matching all pairs of segments that are close enough in the two scans (with a distance smaller than d_{thr}) and recompute the scale/translation based on all these associations to have a more precise estimation.

4. ICL IMPLEMENTATION

We have integrated the clustering and the quality criterion (the global robust distance) in the ICL paradigm in order to compare it with our algorithm in terms of performance and robustness.

4.1 Matching step

We applied a filter based on three criteria to select the correspondences between two 3D segment sets: Angle, distance and overlap

- Angle: in order to simplify the validation of this criterion, we preferred to select segments belonging to the associated clusters (to avoid choosing a threshold for this criterion)
- Distance: we have defined this distance according to equation 10
- Overlap: we have defined the overlap between two 3D segments according to equation 6

So at each iteration, we select the set of segments pairs which validate the three criteria as the matching set.

4.2 Optimization step

At each iteration:

- We have estimated the rotation matrix that aligns the direction vectors of the matched 3D segments according to Alshawa (2007).
- We have used the equations of section 3.5.2 in order to find the scale factor and the translation between the matched segments after having aligned them using the estimated rotation.
- We computed the energy(robust distance) of the estimated transformation (Rotation,scale factor, translation)

Finally, we keep the sampled transformation that has the minimum robust distance

5. EVALUATION AND DISCUSSION

The evaluation of our approach was carried out in two steps using two data types:

5.1 Evaluation on synthetic data

The first evaluation of our approach was carried out on synthetic data, where we generated two copies of a LiDAR line cloud. We then removed 25% of lines in the first copy and 33% of the lines in the second, added noise to both and applied an arbitrary known transformation (rotation, scale factor, translation) to the second. The objective of this first evaluation is:

- Compare the robustness and the speed of the two optimization algorithms (simulated annealing and RANSAC) in order to choose the best to apply.
- Evaluate the performance of our algorithm using the best optimization algorithm.
- Make a comparison between the efficiency of our algorithm and the Iterative Closest line algorithm (ICL).

The various experiments carried out have proven the robustness and speed of RANSAC compared to simulated annealing as shown in Figure 4. The obtained results have proven the performance of our algorithm to register two sets of 3D segments, whatever the initial position, unlike ICL which required a good approximation of the initial transformation to be able to converge towards the correct solution as shown in Table 1. The average running time of our algorithm for the different tests in Table 1 is 161.6 s by fixing the maximum number of iterations of RANSAC to 5000 iterations, knowing that the first 3D segments set contains 144 lines and the second contains 128 lines.

5.2 Evaluation on real data

Afterwards, we performed the registration of heterogeneous data with heterogeneous acquisition platforms:

- Terrestrial images / Terrestrial LiDAR registration as shown in Figure 5.
- Terrestrial images / Aerial image registration as shown in Figure 6.

As it is difficult to construct a ground truth for heterogeneous data registration, we evaluated the registration quality on the 3D visual results. The obtained results demonstrate our algorithm's ability to efficiently register image and LiDAR data. In addition, our algorithm has proven its robustness to small disturbances of verticality.
Achieving precise results requires fine-tuning of the algorithm's parameters. The distance threshold must be chosen reasonably so as not to consider a large number of lines as outliers. The number of iterations of RANSAC must be large enough to ensure the robustness of the algorithm.

6. CONCLUSION AND FUTURE WORK

In this paper, we have dealt with the heterogeneous data registration problem by proposing a new primitives based registration algorithm. The heterogeneity is both in data type (image and LiDAR) and acquisition platform (terrestrial and aerial). Our algorithm takes 3D segments as feature and extracts them from both LiDAR and Image data with specific state of the art algorithms. We defined a global robust distance between two segments sets and we proposed a robust approach to minimize this distance. In order to simplify this minimization, we

Transformation			Our result			ICL result		
Translation	Rotation	Scale (s_0)	Translation Error	Rotation Error	Scale (s)	Translation Error	Rotation Error	Scale (s)
0 m	0°	1	1.77e-15 m	1.06e-15°	$\frac{\lvert s-s_0\rvert}{s_0}$=0	3.6e-16 m	4.7e-15°	$\frac{\lvert s-s_0\rvert}{s_0}$=0
0.39 m	4.66°	0.85	0 m	0.04°	$\frac{\lvert s-s_0\rvert}{s_0}$=0	0.09 m	1.29°	$\frac{\lvert s-s_0\rvert}{s_0}$=0.005
1.31 m	15.66°	1.5	0 m	0.2°	$\frac{\lvert s-s_0\rvert}{s_0}$=0	0.20 m	15.39°	$\frac{\lvert s-s_0\rvert}{s_0}$=0.971
4.20 m	32.66°	2	0 m	0.2°	$\frac{\lvert s-s_0\rvert}{s_0}$=0	1.42 m	27.520	$\frac{\lvert s-s_0\rvert}{s_0}$=0.97

Table 1. Performance tests of our algorithm on synthetic data using different initial errors.

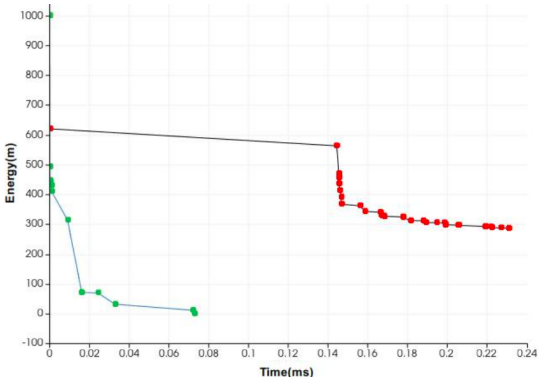


Figure 4. Comparison of the convergence speed and the robustness of RANSAC and simulated annealing: X axis represents time in milliseconds, Y axis represents the energy, green points represent the minimums estimated by RANSAC and the red points represent the minimums estimated by simulated annealing

started by clustering the 3D segments of each data set. The clusters are associated to find possible rotations, then 3D segments from associated clusters are matched in order to find the translation and scale factor minimizing the defined distance. The obtained results demonstrate the efficiency and robustness of our algorithm to register heterogeneous data (terrestrial images/terrestrial LiDAR, terrestrial images/aerial Image) Our main perspective on this work is to use planar polygons as primitives or to use combinations of more segments to have more characteristic features to match.

REFERENCES

Alshawa, M., 2007. lCL: Iterative closest line A novel point cloud registration algorithm based on linear features. *Ekscentar*, 53–59.

Cover, T. M., 1999. *Elements of information theory*. John Wiley & Sons.

Deng, F., Hu, M., Guan, H., 2008. Automatic registration between LiDAR and digital images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37, 487–490.

Ding, M., Lyngbaek, K., Zakhor, A., 2008. Automatic registration of aerial imagery with untextured 3d lidar models. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1–8.

Habib, A. F., Shin, S., Kim, C., Al-Durgham, M., 2006. Integration of photogrammetric and lidar data in a multi-primitive triangulation environment. *Innovations in 3D Geo Information Systems*, Springer, 29–45.

Hofer, M., Maurer, M., Bischof, H., 2017. Efficient 3D scene abstraction using line segments. *Computer Vision and Image Understanding*, 157, 167–178.

Kim, C., Ghanma, M., Habib, A., 2006. Integration of Photogrammetric and LIDAR data for realistic 3D model generation. *Department of Geomatics Engineering, University of Calgary, Canada*.

Kumar Mishra, R., Zhang, Y., 2012. A review of optical imagery and airborne lidar data registration methods. *The Open Remote Sensing Journal*, 5(1).

Liu, L., Stamos, I., 2012. A systematic approach for 2D-image to 3D-range registration in urban environments. *Computer Vision and Image Understanding*, 116(1), 25–37.

Lu, X., Liu, Y., Li, K., 2019. Fast 3D line segment detection from unorganized point cloud. *arXiv preprint arXiv:1901.02532*.

Mastin, A., Kepner, J., Fisher, J., 2009. Automatic registration of lidar and optical images of urban scenes. *2009 IEEE conference on computer vision and pattern recognition*, IEEE, 2639–2646.

Salaün, Y., Marlet, R., Monasse, P., 2016. Multiscale line segment detector for robust and accurate sfm. *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, 2000–2005.

Von Gioi, R. G., Jakubowicz, J., Morel, J.-M., Randall, G., 2012. LSD: a line segment detector. *Image Processing On Line*, 2, 35–55.

Wang, R., Ferrie, F. P., 2015. Automatic registration method for mobile LiDAR data. *Optical Engineering*, 54(1), 013108.

Zhu, B., Ye, Y., Yang, C., Zhou, L., Liu, H., Cao, Y., 2020. Fast and Robust Registration of Aerial Images and LiDAR data Based on Structrual Features and 3D Phase Correlation. *arXiv preprint arXiv:2004.09811*.

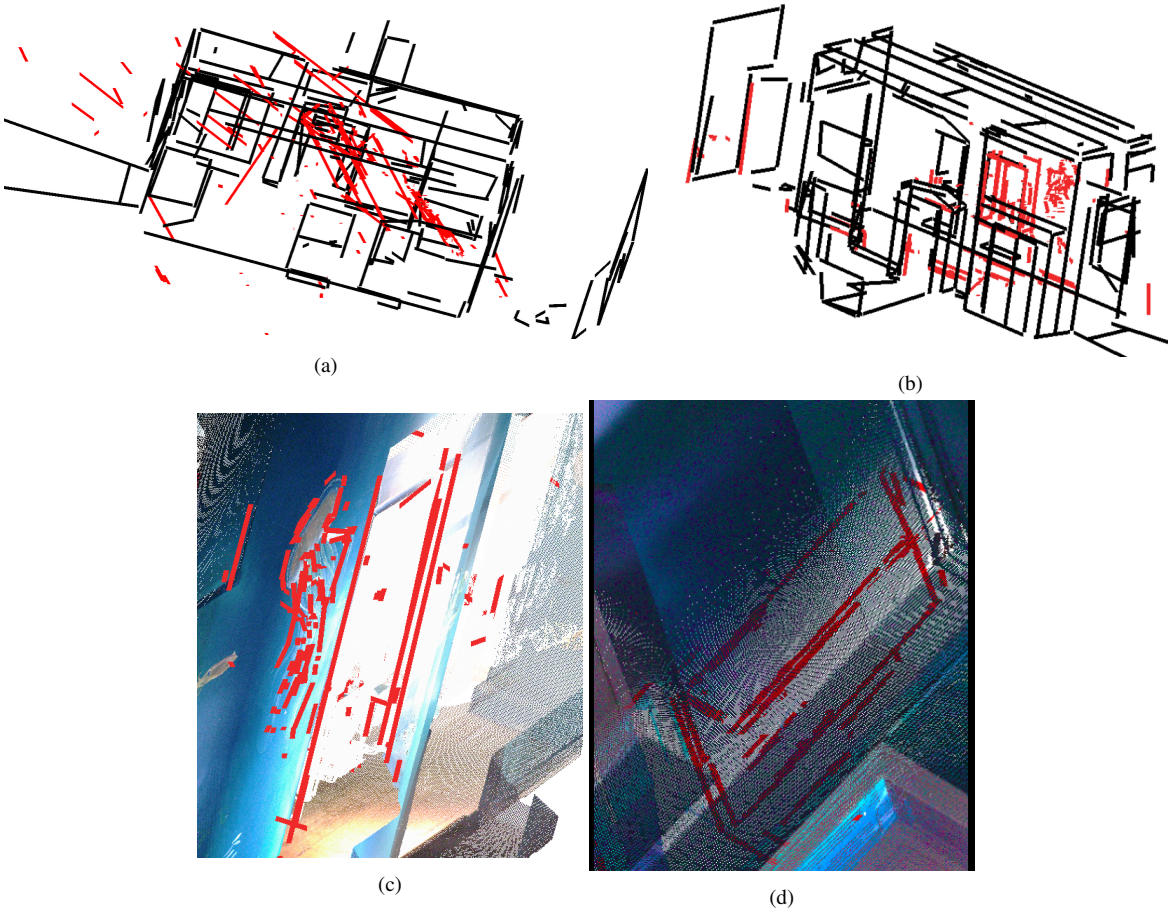


Figure 5. image / LiDAR registration results: (a) position of the two line clouds before registration(red:image lines, black LiDAR lines),(b) position of the two line clouds after registration, (c) LiDAR scan, ((d,e)registration of image lines and the LiDAR scan

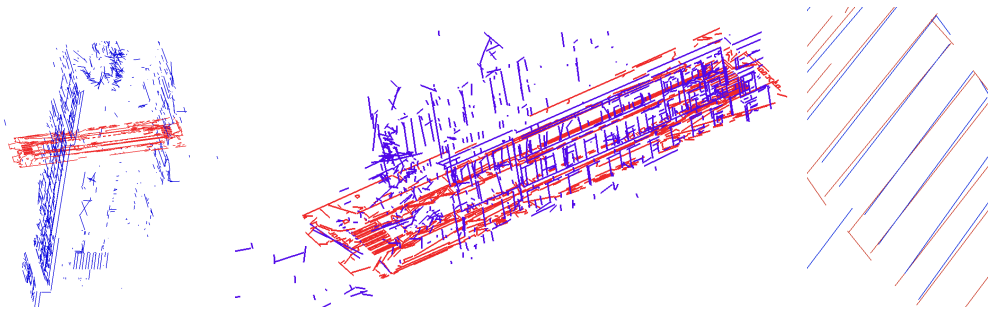


Figure 6. Aerial/Terrestrial registration: 3D lines reconstructed aerial image(red), 3D lines reconstructed from terrestrial images (blue) (a): the initial position, (b): positions after registration, (c): position of pedestrian path lines after registration