Alignment of Point Clouds for Comparison of Infrastructures in Civil Engineering on Quality Control in Metrology

Hadi Khaksari Haddad
Computer Vision and Systems lab,
Department of Electrical and Computer
Engineering, Laval University, Quebec city,
Canada
hadi.khaksari-haddad.1@ulaval.ca

Denis Laurendeau
Computer Vision and Systems lab,
Department of Electrical and Computer
Engineering, Laval University, Quebec city,
Canada
Denis.Laurendeau@gel.ulaval.ca

ABSTRACT

For 3D point cloud registration, Go-ICP [Yang et al., 2016] has been shown to obtain the global optimal solution for a pair composed of a model point cloud and a data point cloud. Go-ICP mostly has been investigated only on standard sets of point clouds. In this paper, we demonstrate the remarkable efficacy of Go-ICP for the alignment of **very complex large-scale** point clouds to their corresponding **deformed** CAD models. In particular, given two distinct sets of point clouds taken from the exterior and the interior of a building, experiments demonstrate that Go-ICP is able to successfully align both of these sets to the point cloud of the CAD model of the whole building (both exterior and interior information included). With the experimentation presented in this paper, we demonstrate that Go-ICP can achieve excellent alignment results and that this approach can be deployed in applications aiming at comparing CAD models of a building ("as designed" model) to the point cloud of the actual building ("as-built" model). Experiments also demonstrate the efficacy of Go-ICP to align a deformed copy of a man-made object to the original object in quality control applications.

Keywords

Global 3D registration large-scale point clouds comparison of models Go-ICP.

1 INTRODUCTION

Point cloud registration is a fundamental problem in computer vision. Given two point clouds in different coordinate systems with different poses, the goal of registration is to find the transformation that best align the first point cloud to the other one. Point cloud registration plays an important role in many vision applications. An application addressed in this paper is the alignment of a large point cloud, obtained by scanning a structure such as a building, with a CAD model of this structure for the purpose of augmented reality. Comparing the "as-designed" structure (i.e. CAD) to the "as-built" structure (i.e. the actual building) is an important problem in civil engineering since it often occurs that, for various reasons, what has been built is not what was originally designed and it is desired to find where the difference occurs. Exploring this application leads to a complementary study that consists of aligning a point cloud to a deformed of copy of itself for quality control and metrology.

3D scanners, Lidars and Structure-from-Motion can be used to collect point clouds from large-scale infrastructure components such as buildings.

The Iterative Closest point (ICP) algorithm [Besl and McKay, 1992] is a well-known algorithm for registering point clouds under Euclidean transformation. ICP is also known for being subject to the problem of local minima and also requires that the relative pose between the point clouds be close to the rigid transformation that is needed to align them for the algorithm to converge.

The work presented in this paper explores a globally optimal solution to the Euclidean registration problem defined by ICP in 3D. The Go-ICP Method exploiting the well-established Branch-and-Bound (BnB) theory for global optimization [Yang et al., 2016] is adapted to the problem of aligning a point cloud collected on a very large structure with a deformed copy of the point cloud of the same structure. Another topic that is studied in the paper is to adapt Go-ICP to the alignment of a large point cloud of a structure (such as a building) with a *CAD model* of this structure. A third application to which Go-ICP is extended is to align the point cloud of man-made objects with deformed copies of themselves in the context of quality control in metrology. The paper also presents a thorough analysis of the Go-ICP hyper-parameters on the quality of registration.

The experiments presented in the paper demonstrate that Go-ICP succeeds in the alignment of a point cloud of a structure such as a building with a deformed copy

https://doi.org/10.24132/CSRN.2019.2901.1.19

ISBN 978-80-86943-37-4

of itself or with a CAD model of the structure. This functionality can be used for instance in augmented reality applications in civil engineering.

2 RELATED WORKS

Significant research has been conducted on global alignment methods and global optimal alignment methods. A class of global method is based on shape descriptors. The authors of [Johnson and Hebert, 1999] present a Greedy algorithm for related general shape representation based on spin images on using shape contexts for shape matching in [Belongie et al., 2002]. Authors in [Gelfand et al., 2005] propose a robust global registration approach by using integral volume based on local geometry and BnB.

The authors of [Rusu et al., 2009] present the Fast Point Feature Histogram that finds registration by using Sample Consensus Initial Alignment (SAC-IA). Random Sample Consensus (RANSAC) is used for matching a model to data in an application for Local Determination Problem (LPD) [Gelfand et al., 2005]. Authors in [Irani and Raghavan, 1999] propose a random sampling similar to RANSAC for alignment using 2D points while authors in [Aiger et al., 2008] exploits 4-points congruent sets without pre-filtering or noisy data removal. The authors of [Makadia et al., 2006] find the 3D rotation of two Extended Gaussian Images (EGI) based on global shape descriptors.

Another class of global methods is based on stochastic optimization. The authors of [Sandhu et al., 2010] solved point cloud registration by using particle filtering and stochastic dynamics. Range images registration is achieved by using the Genetic Algorithm (GA) proposed in [Silva et al., 2005]. The work in [Blais and Levine, 1995] presents 3D registration for range images via a cost function maximizing the quality of registration. In [Papazov and Burschka, 2011], stochastic global optimization is used for pairwise rigid point cloud registration based a cost function robust to outliers.

Many approaches for global optimal registration are based on BnB. The authors of [Breuel, 2003] present an algorithm for geometric matching based on BnB for image pattern while authors in [Pfeuffer et al., 2012] propose geometric matching based on BnB for medical images. A framework based on Lipschitz global optimization theory for 3D registration using BnB is presented in [Li and Hartley, 2007]. This approach is very slow for large point-sets. Authors in [Olsson et al., 2009] find the global optimal registration via a framework matching point-to-point, point-to-line, and point-to-plane using BnB. Authors in [Parra Bustos et al., 2014] present an approach for fast search in BnB for geometric alignment. This method proposes the SO(3) spaces that are then searched by the BnB algorithms. Other global methods by Nicolas Mellado [Mellado et al., 2014] and Qian-Yi Zhou [Zhou et al., 2016] have been proposed.

The authors of [Yang et al., 2016] present the Go-ICP algorithm as a global optimal registration method based on BnB via SO(3) spaces. This approach supports 3D Euclidean registration with both translation and rotation (i.e. rigid transformation). It can work directly on dense point clouds without the need for a good initial pose or putative correspondences.

3 PROPOSED APPROACH FOR THE ALIGNMENT OF POINT CLOUDS WITH CAD MODELS AND EXPLORATION OF THE EFFECT OF HYPER-PARAMETERS OF GO-ICP ON THE ALIGNMENT

3.1 Solution of the problem of the alignment of a large point cloud with a CAD model

Registration is the process of aligning 3D data point sets that are captured from different viewpoints [1]. In this section, we focus on aligning two data sets, namely model (reference) points, and data points that are collected with a different pose. Formally, $X = \{x_i\}, i = 1,...,N$ and $Y = \{y_j\}, j = 1,...,M$ represent data points and model points, respectively. Note that it is usually assumed that the number of data points (N) is lower than the number of model points (M). Generally, the objective function for 3D registration is defined on the L_2 square error as follows:

$$E(R,t) = e_i(R,t)^2 = \sum_{i=1}^{N} \|Rx_i + t - y_{j^*})\| \qquad (1)$$

Where $R \in [\pi, \pi]^3$ is the rotation space and $t \in [-\varepsilon, \varepsilon]^3$ is the translation space, and $e_i(R,t)$ indicates the Euclidean distance between x_i transformed by (R,t) to y_{j^*} as the optimal correspondence between x_i and y_i.

I- Global 3D registration

GO-ICP [Yang et al., 2016] proposes to combine the Branch-and-Bound (BnB) optimization algorithm with ICP in order to find the global optimal transformation in Eq. 1. Unlike ICP, this algorithm is not sensitive to the initial transformation. Therefore, it can start from any initial pose and eventually end up to the global optimum solution. The key contribution of Go-ICP is the derivation of registration error bounds. In fact, identifying efficiently the upper and lower bounds of the regions of search space helps to find the global optimum more quickly (Fig. 1 (a)).

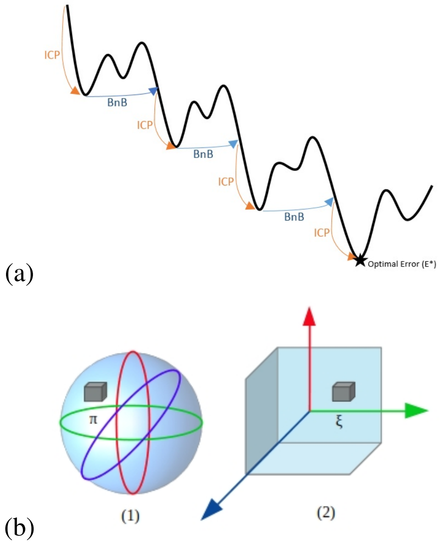


Figure 1: (a): Collaboration of BnB and ICP. (b): The search space consists of the rotation space and the translation space. The gray cubes are the sub cubes from these spaces (1) rotation space, which can be represented by a sphere with radius π. (2) the translation space which can be represented by a cube with length 2ξ. Redrawing from [Yang et al., 2016].

II- Domains for rotation and translation parameters

Generally, the search space of non-rigid registration consists of the rotation space and the translation space. The entire 3D rotation space can be modeled by a sphere with radius π. For the translation space, one can consider a bounded cube $[-\xi,\xi]^3$. The search space is depicted in Fig. 1 (b). Note that this cube encloses the sphere with radius π. The BnB algorithm is exploited to divide the search space, i.e. rotation space and translation space, into sub-cubes. To efficiently and intelligently search the rotation space and translation space, BnB should have access to the upper and lower bounds of sub-cubes. As the authors [Yang et al., 2016] point out, their main contribution is finding efficiently the upper and lower bounds of the search space. In this paper, we skip the theory that is proposed to find the error bounds and refer the reader to [Yang et al., 2016] for details.

3.2 3D Registration of deformed objects using Go-ICP

3.2.1- Data acquisition

In this section, we present the procedure for obtaining the data points and model points of an object in order to demonstrate how Go-ICP can be exploited to find the optimal registration between the point cloud of a man-made object and a modified copy of itself. This object, which was constructed by a 3D printer, consists of a cube, two convex cylinders with different heights, a convex hemisphere, a concave hemisphere, and one concave semi-cylinder (Fig. 2 (a)). We added some parts to the original object to create new objects (i.e."deformed" objects) for aligning the original object with these deformed objects (fig. 2 (b)) using Go-ICP.

We captured data points of the deformed objects and original points using a Creaform Go!scan 50 handheld meteorologic 3D scanner. The reflective markers installed on the the object (Fig.2 (c)) are used by the scanner for self-positioning. They are not used for the estimation of registration by Go-ICP. A total of 3345 data points were scanned on the original object and 6879 model points were scanned on the deformed object. To apply Go-ICP, we need to normalize the data sets in the interval $[-1,1]$. The search space for rotation is $[-\pi,\pi]^3$ and the one for translation is $[-0.5,0.5]^3$. To get the best result from BnB, its hyperparameters must be selected carefully. In the experiments, we considered all data points of the original object, i.e. $=$ 3345. Go-ICP uses two hyper-parameters. Parameter ρ, called the "trimming" factor, controls the robustness to outliers while parameter ε, called the registration error threshold, controls the convergence of the registration process. Parameter ρ was set to 20% and ε was set to 5. The registration of the original object and the deformed object 1 using Go-ICP is executed in 6472.8 seconds with a Root-Mean-Square (RMS) error of 7.48679 while the RMS error of using ICP alone is 14.383 (Fig. 3). Note that with lower ρ values, the registration is not as good as for $\rho = 20\%$ using the RMS error as a criterion. In addition, the algorithm running time for smaller ε values is considerably greater. Finally, the alignment results of Go-ICP are good compared to ICP alone as one can verify qualitatively in Fig. 3. The Go-ICP Method was implemented in C++ on a PC with an Intel core i7 2*4.00 GHz CPU and 32 GB RAM.

3.2.2- Analysis of the Hyperparameters of Go-ICP on the registration of a point cloud with a deformed copy of itself

The effect of trimming

To remove possible outliers, a trimming procedure can be applied on data points so that ρ percent of data points that are far from the model points are removed. Based on our observations, we found that parameter ρ (the trimming factor) has a major influence on the RMS error and running time of the algorithms (BnB and ICP). More precisely, a large value for ρ can result in erroneous registered data points since some inliers that are essential for an accurate registration are removed by the trimming procedure. On the other hand, a very small

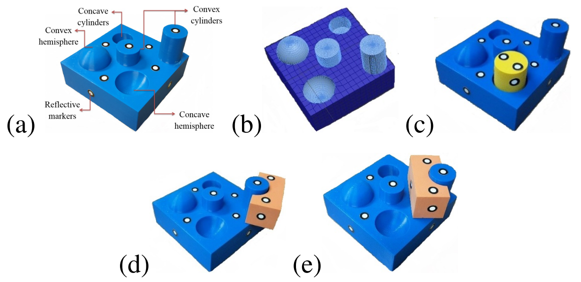


Figure 2: The data and the objects. (a) is the original object (3D print), (b) is the CAD model of the original object (data point cloud), (c) is deformed object 1 obtained by adding a convex cylinder in the concave hemisphere, (d) is deformed object 2 obtained by adding a block to the outside of the tall convex cylinder, and (e) is deformed object 3 obtained by adding a block to the inside of the tall convex cylinder.

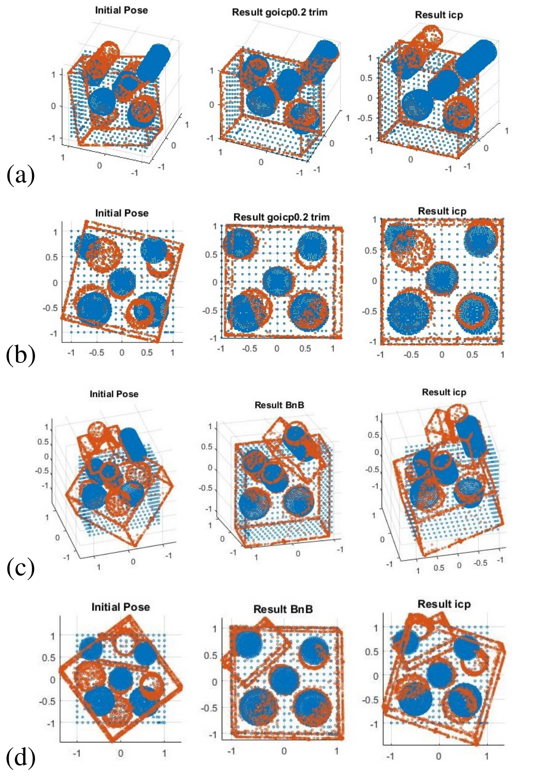


Figure 3: The results of Go-ICP for 3D registration of deformed objects. The blue point clouds are the data and red point clouds are the models (deformed objects). (a) is the 3D registration of deformed object 1 (view: axis of XYZ), (b) is the 3D registration of deformed object 1 (view: axis of xz), (c) is the 3D registration of deformed object 3 (view: axis of XYZ), and (d) is the 3D registration of deformed object 3 (view: axis of xz). Left: initial pose, center: registration with Go-ICP, right: registration with ICP alone.

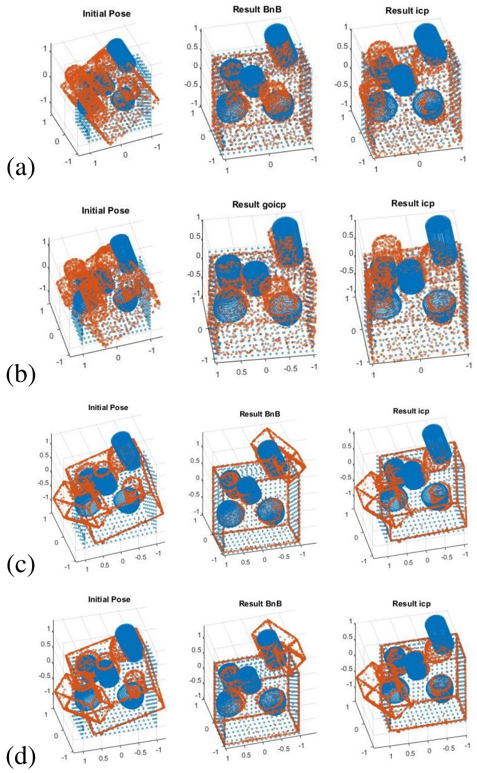


Figure 4: The effect of trimming on 3D registration of deformed objects 1 and 2. (a) is the result for $\rho = 10\%, \varepsilon = 10$ on deformed object 1, (b) is the result for $\rho = 20\%, \varepsilon = 10$ on deformed object 1, (c) is the result for $\rho = 10\%, \varepsilon = 10$ on deformed object 2, and (d) is the result for $\rho = 20\%, \varepsilon = 10$ on deformed object 2. Left: initial pose, center: registration with Go-ICP, right: registration with ICP alone.

ρ (or $\rho = 0$) might lead to a very large running time since the outliers prevent BnB to converge to a reasonable RMS error. A very small ρ (or ρ close to 0) does not remove any outliers that may be at a large distance from the model and all points must be considered in the optimization. In other words, due to the presence of outliers, the RMS error cannot converge to the selected threshold (ε). For example, in Fig. 4, considering $\rho = 10\%$, $\rho = 20\%$, and $\varepsilon = 10$ for Deformed object 1, 2 results in inaccurate registration, while using $\rho = 10\%$ and $\varepsilon = 10$ registers Deformed objects 1, 2 more accurately. Therefore, to get the best results from BnB and ICP, one should tune ρ for the specific data point set in order to remove outliers properly. In all the experiments, we find the best ρ factor that produces the best registration in a reasonable running time e.g. less than 10 minutes for most experiments except the ones with very large point clouds (with millions of points). The curves of the effect of the trimming factor on RMS error and computation time are shown in Fig. 5.

The effect of registration error threshold ε

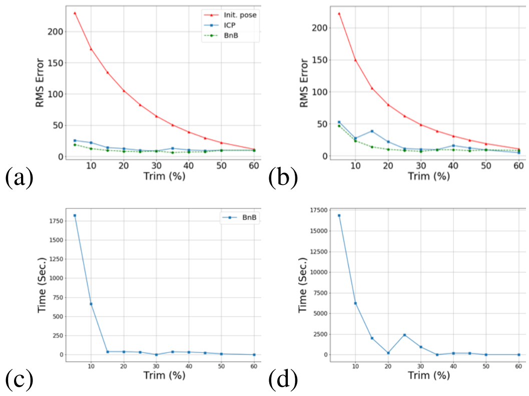


Figure 5: The curve of the impact of the trimming factor ρ on registration error and running time. The red curves are the initial RMS errors, the blue curves are the RMS errors of ICP, and the green curves are the RMS errors of Go-ICP. (a) is the curve of the effect of trimming on the 3D registration error of deformed object 1, (b) is the result for deformed object 2, (c) is the curve of the effect of trimming on execution time (seconds) of the 3D registration of deformed object 1, and (d) is execution time (seconds) for deformed object 2.

The accuracy (RMS error) and running time of BnB highly depend on the threshold (ε) as it defines the stopping criterion. In other words, BnB terminates its registration process when the optimal RMS error is close enough to the lower bound RMS error, i.e. $E^* - E_r < \varepsilon$ (E^* is the optimal RMS error and E_r is the lower bound RMS error of the domain of rotation space). Setting the threshold to a small value can register data points to model points more accurately but the running time is very high. On the other hand, a greater threshold can lead to a fast execution time, but the accuracy of registration decreases. It is worthwhile to note that the initial optimal RMS error of Go-ICP is computed by ICP. So, for large threshold values, Go-ICP can terminate very fast since its initial optimal RMS error can become lower than the specified threshold quickly. For example, the running time for Deformed objects 1, 2 with $\varepsilon = 7$ is only 1839.64 seconds for deformed object 1 and 4258.06 seconds for deformed object 2 since ICP as the first step of BnB achieves a RMS error of 0.685 which is lower than $\varepsilon = 5$ (Note that in this case the lower bound RMS error was zero) (Fig. 6). However, this RMS error achieved by ICP is not the global minimum. Therefore, we highlight that for reaching the global minimum of the objective function, one should set the threshold appropriately. The curve of the effect of the threshold ε on the RMS error and on the running time is shown in Fig. 7.

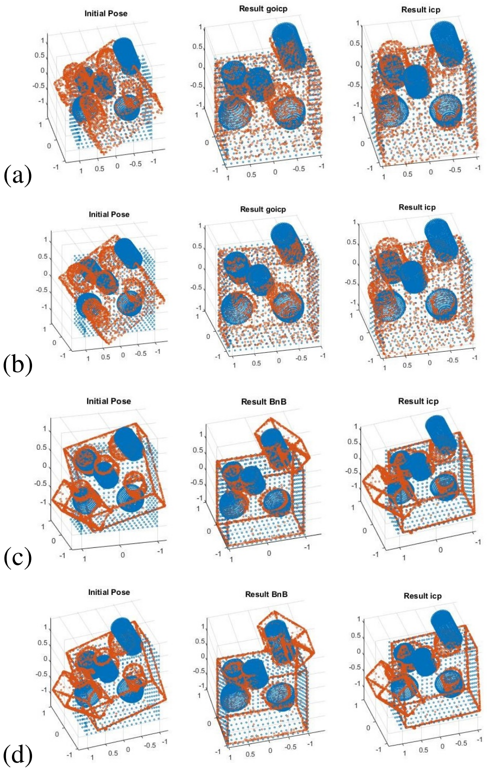


Figure 6: The effect of threshold ε on 3D registration for deformed objects 1 and 2. (a) is the result of $\varepsilon = 5, \rho = 10\%$ on deformed object 1, (b) is the result of $\varepsilon = 15, \rho = 10\%$ on deformed object 1, (c) is the result of $\varepsilon = 5, \rho = 10\%$ on deformed object 2, and (d) is the result of $\varepsilon = 15, \rho = 10\%$ on deformed object 2. Left: initial pose, center: registration with Go-ICP, right: registration with ICP alone.

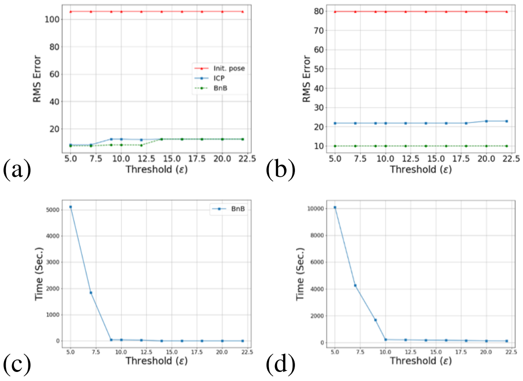


Figure 7: The curve of the effect of the threshold ε on registration error ((a) and (b)) and execution time in seconds ((c) and (d)). The red curves are the initial errors, the blue curves are the errors of ICP, and the green curves are the errors of Go-ICP. Effect of ε on the registration for deformed object 1 (a) and deformed object 2 (b). Effect of ε on execution time for deformed object 1 (c) and deformed object 2 (d).

The effect of reducing the number of points (Nd) on the execution time of Go-ICP.

We also studied how down-sampling the point clouds affect the registration achieved by go-ICP. We downsampled the original and model point clouds and repeated the registration experiments for deformed object 1, 2, and 3. Nd is the factor by which the number of points were reduced (i.e. $N_{down} = N_{total/Nd}$). As shown in fig. 8, the registration values (R, T) are the same for all values of Nd but the execution time varies. It is surprising to see that decreasing the number of points does not necessarily reduce the computation time specially for deformed object 3 in Fig. 8 (c).

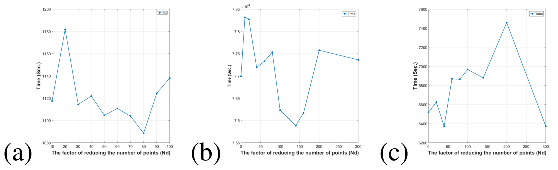


Figure 8: The effect of the down-sampling factor Nd on Go-ICP execution time. (a): deformed object 1. (b): deformed object 2. (c): deformed object 3.

3.2.3- Alignment of large-size point clouds using Go-ICP

In this section, we apply the Go-ICP approach to the alignment of large-scale point clouds and show that it can achieve good alignment.

We use the CAD model of a building with 387,564 data points (Fig. 9 (a)) and two real large-scale point clouds (inside point cloud with 8,497,325 data points and outside point cloud with 26,574,097 data points) captured by a Lidar (Fig. 9 (e, g)). As can be seen in Fig. 9 (b), the data points on the CAD model are not adequate for alignment because they are not distributed evenly which prevents Go-ICP to find the alignment of this CAD model with large-scale point clouds. Therefore, the CAD model is resampled using Blender to yield 330,327 data points evenly distributed on the inside and outside (Fig. 9 (c) and (d)). For the alignment, the large-scale point clouds were cropped to keep only the points belonging to interior or exterior of a building (Fig. 9 (f, h)). Consequently, the points on the ground were discarded.

Go-ICP was able to find the alignment between the resampled CAD model and the large-scale point cloud of the inside and outside of the building. For these 3D registration experiments, we considered $\rho = 20\%$ for trimming and $\varepsilon = 5$ as the registration error. The number of points of the cropped point cloud of the inside of the building is 6,297,579. The execution time of this registration is 7742.02 seconds for a RMS error of 6.7934. The ICP alone RMS error was 12.7098 (Fig. 10 (a)). Go-ICP estimates the rotation and translation parameters. The parameters are then used to align the original CAD model of the building with the cropped point cloud of the inside of the building. (Fig. 10 (c)).

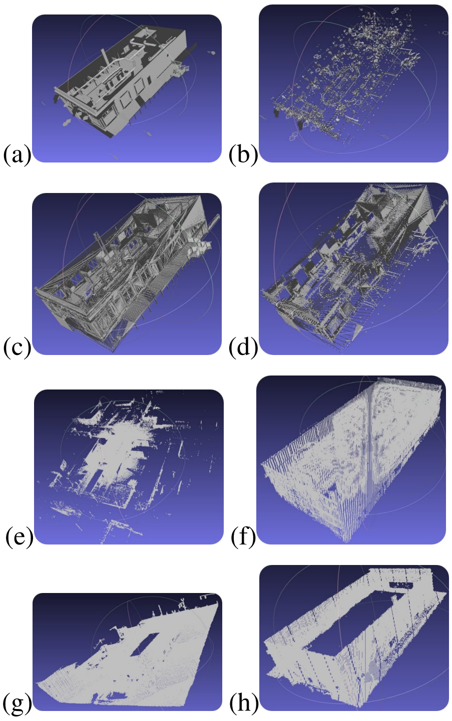


Figure 9: The point clouds of a building and its large-scale point clouds. (a, b) are the CAD model of the building (original CAD model) and its point cloud, (c,d) resampled of data points of the original CAD model with a more even spatial distribution of the points, (e, f) is the point cloud of the inside of the building and its cropped point cloud, (g, h) is the point cloud of the outside of the building and its cropped point cloud.

For the alignment of the resampled CAD model and the cropped point cloud of the outside of the building, we considered $\rho = 20\%$ for trimming and $\varepsilon = 10$ as the registration error. The number of points of the cropped point cloud of the outside of the building is 12,841,366. Go-ICP found the 3D registration shown in Fig. 11 (a) and (b). The execution time of this registration is 3559.15 seconds for a RMS error 13.9063. The ICP alone RMS error was 37.0139. The registration parameters were used to align the CAD model with the cropped point cloud of the outside of the building (Fig. 11 (c) and (d)) with Go-ICP. It is thus possible to compare the actual building ("as-built") with the CAD model of the design ("as-designed model").

In section 4, we use the Go-ICP for aligning the resampled CAD model of a building with a large-scale point cloud without any change (no cropping of the point cloud). Experiments on the alignment of man-made objects with large deformations are also presented.

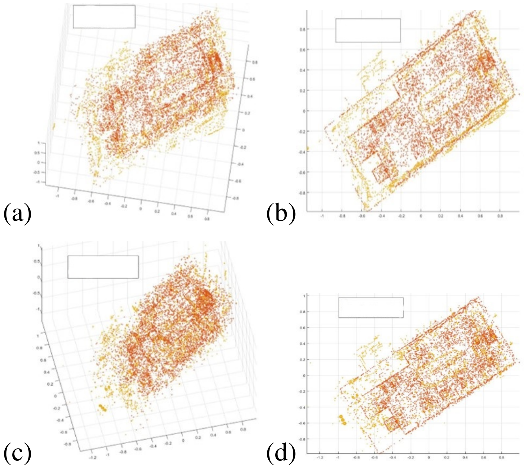


Figure 10: Result of the alignment of the resampled CAD model and the point cloud of the inside of the building: xyz view (a), xz view (b). Red: CAD model. Yellow: point cloud. The parameters of the registration are used to align the original CAD model with the point cloud: xyz view (c), xz view (d).

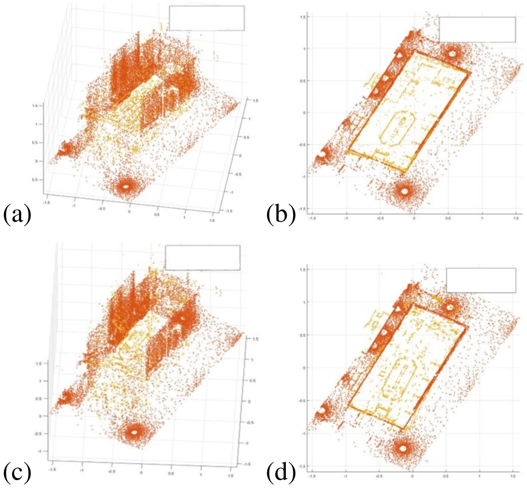


Figure 11: Result of the alignment of the resampled CAD model and the point cloud of the outside of the building: xyz view (a), xz view (b). Red: Cad model. Yellow: point cloud. The parameters of the registration are used to align the original CAD model with the point cloud: xyz view (c), xz view (d).

4 EXPERIMENTS ON THE ALIGNMENT OF THE POINT CLOUD OF DEFORMED OBJECTS WITH ITS ORIGINAL CAD MODEL

4.1 CAD model

In this section, we aim to align two deformed CAD model using Go-ICP. For this purpose, we consider the deformed CAD model 2 and the deformed CAD model that was described in Sec.3.2.1 (see Fig. 2 (d, e)). The 3D registration of these CAD models is difficult because of the large deformation. However, Go-ICP can find a good registration with $\rho = 20\%$ and $\varepsilon = 10$. Alignment results are shown in Fig. 12. In this case, the RMS error was 14.9752.

As an additional test, we used other CAD models with different shapes that represent alignment challenges (Fig. 13 (a, d)). In the first part of the test, we deformed the CAD model shown in Fig. 13 (a)(and called CAD model 1) with 3Ds Max and created a new CAD model (see Fig. 13 (b)). The challenge consists of the alignment of CAD model 1 and its deformed copy for quality control for instance. Go-ICP was able to find the 3D registration with $\rho = 10\%$ and $\varepsilon = 5$ with RMS error 4501.87 in 965.276 seconds. The result of registration is shown in Fig. 13 (d). Such an alignment can be used in metrology to compare objects with a deformed copy of themselves to assess whether the fabrication is reliable (the deformation is reflected by the large value of the RMS error).

In the second part of this test, we perform the 3D registration between CAD model 1 with the CAD model shown in Fig. 13 (e). This registration is difficult since the two objects are different but it may be interesting to compare them in a metrology application or for visualizing the objects in a virtual reality application. Go-ICP succeeded in aligning these different objects with $\rho = 10\%$, $\varepsilon = 5$, RMS error 2051.108, and an execution time of 324.502 seconds. The result is shown in Fig. 13 (g).

A final test using Go-ICP for the alignment of an object and a deformed copy of itself is shown in Fig. 14. Fig 14 (a) shows a mechanical part and a deformed copy in Fig. 14 (b). Fig. 14 (c) shows the alignment obtained with Go-ICP with $\rho = 10\%$ and $\varepsilon = 5$ with a RMS error of 227.987 in 32.119 seconds.

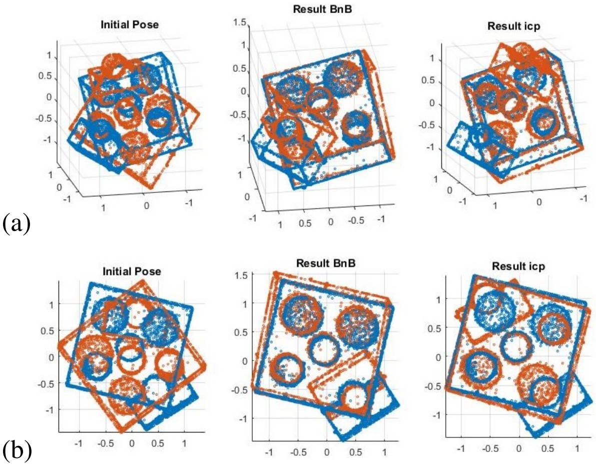


Figure 12: Result of 3D registration of deformed CAD model 2 and deformed CAD model 3. (a) XYZ axis, (b) XZ axis.

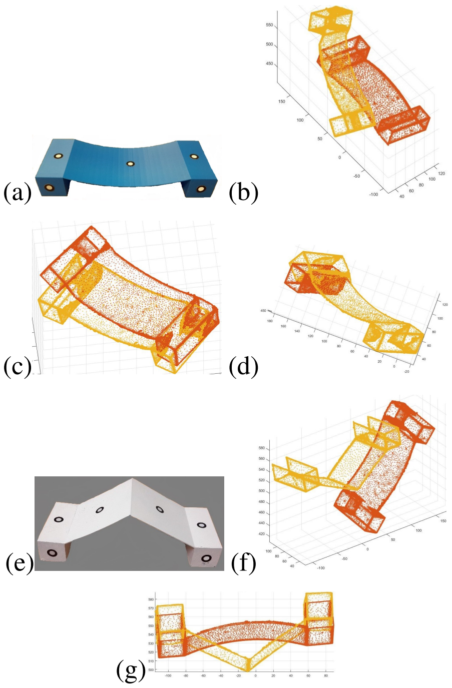


Figure 13: The results of 3D registration of the CAD models. (a) is the CAD model 1, (b) is the initial pose and point clouds of CAD model 1 and its deformed copy, (c) is the ICP alone result of the alignment of the CAD model 1 and its deformed copy, (d) is the result of the alignment of the CAD model 1 and its deformed copy, (e) is CAD model 2, (f) is the initial pose and point clouds of CAD model 1 and CAD model 2, (g) is the result of the alignment of CAD model 1 and CAD model 2.

4.2 Alignment of Large-Scale Point Clouds

The most challenging 3D registration experiment on which Go-ICP was tested is the alignment of very large-scale point clouds. For this experiment, we consider the CAD model of a building with 42,609 points and a very large-scale point cloud from the area of the original building with 1,236,922 points that was captured by a Lidar. Fig. 15 (a) shows the CAD model of the building and Fig. 15 (b, c) show the area of the building and its surroundings and the corresponding point cloud.

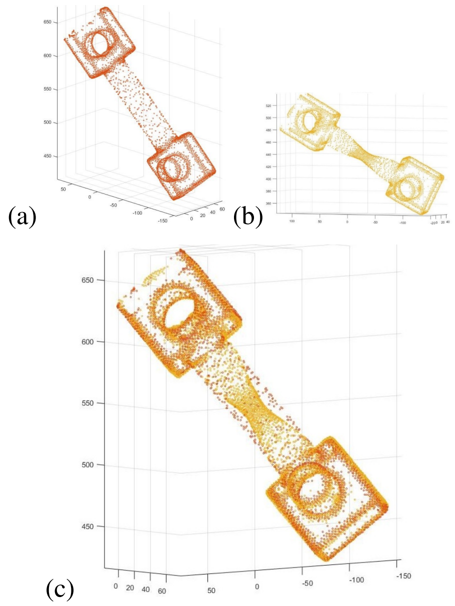


Figure 14: The results of 3D registration of mechanical parts. (a) point cloud of the part 1, (b) point cloud of deformed part 1, (c) is the result of the alignment of the part 1 and the deformed copy of itself 2.

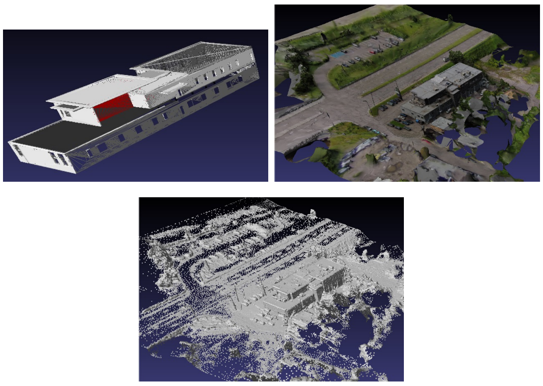


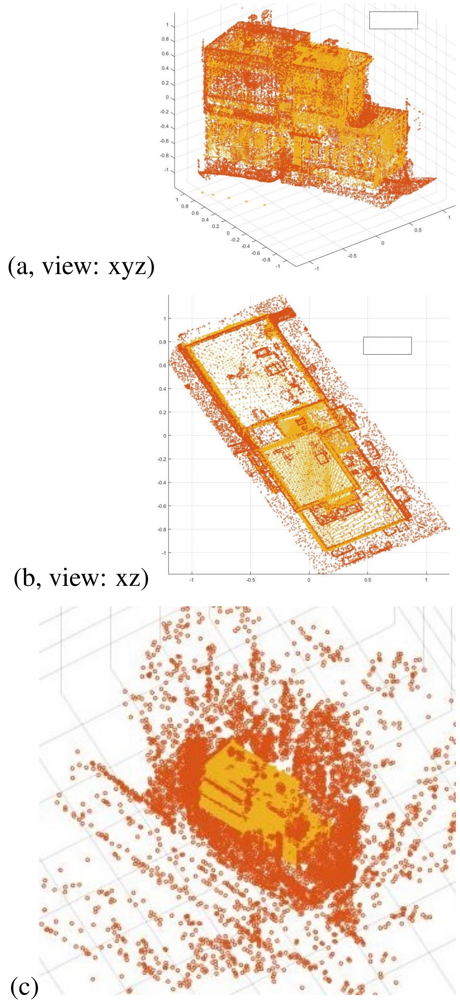
Figure 15: The CAD model of the building (a), the real area of the building (b), the large-scale point cloud of area (c).

The goal is to align the CAD model of the building to the large-scale point cloud of the building and its surrounding area without any cropping. This makes the registration problem very difficult. We divided this experiment into two steps. In the first step, the points not belonging to the building were cropped and the CAD model was aligned with the remaining points (i.e. the belonging to the building). It is still a challenge to find a good alignment. Nevertheless, Go-ICP has found the 3D registration with $\rho = 20\%$, $\varepsilon = 20$, and RMS error of 20.4129 in 48107 seconds. The result is shown in Fig. 16 (a): xyz view and (b): xz view.

In the second step, we aimed to align the CAD model to the point cloud of the building and its surrounding without any cropping. This is a significant challenge because of the number of points involved and the many local minimums of the cost function. Nevertheless Go-ICP is able to achieve the alignment with $\rho = 20\%$ and $\varepsilon = 60$ in 186,359 seconds and with a RMS error of 61.2573. Fig. 16 (c) shows the result of this 3D registration.

5 CONCLUSIONS

The Go-ICP algorithm was exploited to find the registration between a large point cloud and a deformed



(a, view: xyz)

(b, view: xz)

(c)

Figure 16: The results of the alignment between the CAD model and very large-scale point cloud. Red points: CAD model, yellow point: building and its area. (a, b) are the 3D registration of the surrounding CAD model and the original building point cloud, (c) is the result of the registration of the CAD model and the point cloud of the building and its surrounding area without cropping.

copy itself or with a CAD model. Experiments show that with a proper selection of the algorithm's hyperparameters, good alignment can be achieved. For very large point clouds, the computation time needed for achieving a good alignment is important. We are currently using Go-ICP to find the the initial alignment between the CAD models of civil engineering infrastructures(building) and point clouds collected during its construction. This initial alignment is used in an augmented reality application providing architects ways of assessing whether or not what is being built ("as-built") corresponds with what has been designed by architects ("as-designed" model). The paper also demonstrates that Go-ICP can be exploited successfully for comparing man-made objects with deformed copies of themselves for the purpose of quality control in metrology applications.

6 ACKNOWLEDGMENTS

The authors acknowledge the support of NSERC-Bentley CRD grant number 474640-14.

7 REFERENCES

[Aiger et al., 2008] Aiger, D., Mitra, N. J., and Cohen-Or, D. (2008). 4-points congruent sets for robust pairwise surface registration. In *ACM Transactions on Graphics (TOG)*, volume 27, page 85. Acm.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):509–522.

[Besl and McKay, 1992] Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics.

[Blais and Levine, 1995] Blais, G. and Levine, M. D. (1995). Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824.

[Breuel, 2003] Breuel, T. M. (2003). Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding*, 90(3):258–294.

[Gelfand et al., 2005] Gelfand, N., Mitra, N. J., Guibas, L. J., and Pottmann, H. (2005). Robust global registration. In *Symposium on geometry processing*, volume 2, page 5. Vienna, Austria.

[Irani and Raghavan, 1999] Irani, S. and Raghavan, P. (1999). Combinatorial and experimental results for randomized point matching algorithms. *Computational Geometry*, 12(1-2):17–31.

[Johnson and Hebert, 1999] Johnson, A. E. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449.

[Li and Hartley, 2007] Li, H. and Hartley, R. (2007). The 3d-3d registration problem revisited. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE.

[Makadia et al., 2006] Makadia, A., Patterson, A., and Daniilidis, K. (2006). Fully automatic registration of 3d point clouds. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 1297–1304. IEEE.

[Mellado et al., 2014] Mellado, N., Aiger, D., and Mitra, N. J. (2014). Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library.

[Olsson et al., 2009] Olsson, C., Kahl, F., and Oskarsson, M. (2009). Branch-and-bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):783–794.

[Papazov and Burschka, 2011] Papazov, C. and Burschka, D. (2011). Stochastic global optimization for robust point set registration. *Computer Vision and Image Understanding*, 115(12):1598–1609.

[Parra Bustos et al., 2014] Parra Bustos, A., Chin, T.-J., and Suter, D. (2014). Fast rotation search with stereographic projections for 3d registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3930–3937.

[Pfeuffer et al., 2012] Pfeuffer, F., Stiglmayr, M., and Klamroth, K. (2012). Discrete and geometric branch and bound algorithms for medical image registration. *Annals of Operations Research*, 196(1):737–765.

[Rusu et al., 2009] Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217. IEEE.

[Sandhu et al., 2010] Sandhu, R., Dambreville, S., and Tannenbaum, A. (2010). Point set registration via particle filtering and stochastic dynamics. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1459–1473.

[Silva et al., 2005] Silva, L., Bellon, O. R. P., and Boyer, K. L. (2005). Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE transactions on pattern analysis and machine intelligence*, 27(5):762–776.

[Yang et al., 2016] Yang, J., Li, H., Campbell, D., and Jia, Y. (2016). Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254.

[Zhou et al., 2016] Zhou, Q.-Y., Park, J., and Koltun, V. (2016). Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer.