

AUTOMATED RECOGNITION OF 3D CAD OBJECTS IN SITE LASER SCANS FOR PROJECT 3D STATUS VISUALIZATION AND PERFORMANCE CONTROL

Frederic Bosche¹, Member, ASCE,
Carl T. Haas², Member, ASCE,
and Burcu Akinci³, Member, ASCE

ABSTRACT

This paper presents a new approach that allows automated recognition of three-dimensional (3D) Computer-Aided Design (CAD) objects from 3D site laser scans. This approach provides a robust and efficient means to recognize objects in a scene by integrating planning technologies, such as multi-dimensional CAD modeling, and field technologies, such as 3D laser scanning. Using such an approach, it would be possible to visualize the 3D status of a project and automate some tasks related to project control. These tasks include: 3D progress tracking, productivity tracking, and construction dimensional quality assessment and quality control (QA/QC). This paper provides an overview of the developed approach, and demonstrates its performance in object recognition and project 3D status visualization, with data collected from a construction job site.

Keywords: 3D, object recognition, automation, performance, visualization, project control

INTRODUCTION

In the last decades, the exponential increase in computational capacities has allowed the Architectural Engineering Construction & Facility Management (AEC&FM) industry develop and implement more powerful office and field technologies. These include multi-dimensional Computer-Aided Design (CAD) modeling on the office side, and three-dimensional (3D) sensing technologies, such as total stations, the Global Positioning System (GPS), Radio Frequency Identification (RFID) and Ultra Wide Band (UWB) tracking systems and 3D laser scanning (also referred to as LADAR and LIDAR scanning) on the field side. The most recent and a promising technology is 3D laser scanning. It is already used in several applications, but the authors show below that it has a major limitation that limits the industry practitioners' abilities to fully take advantage of it.

¹Department of Civil and Environmental Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1. E-mail: fbosche@engmail.uwaterloo.ca

²Department of Civil and Environmental Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1. E-mail: chaas@civmail.uwaterloo.ca

³Department of Civil and Environmental Engineering, Carnegie Mellon University, Porter Hall 119, Pittsburgh, PA, 15213. E-mail: bakinci@cmu.edu

Indeed, many project performance control tasks require 3D as-designed and as-built information organized at the object level (*e.g.* beam, column, floor, wall and pipe). These tasks include: (1) construction progress tracking, (2) productivity tracking, (3) construction quality assessment and quality control (QA/QC), and (4) life-cycle 3D health monitoring. On one side, multi-dimensional CAD software, and more recently building, infrastructure and industrial facility information models (*e.g.* BIM, BrIM, ISO 15296) are being developed for project and facility life-cycle management. They are typically built upon a project's 3D model, which is a 3D representation of the as-designed project dimensional specifications, and which organizes 3D as-designed information at the object level. On the other side, laser scans capture comprehensive and detailed 3D as-built information. It, thus, provides an opportunity to correspond 3D as-built and as-designed spatial models of a project and support the project performance control tasks stated above. However, it is currently too complex to organize (or segment) laser scanned data at the object level. Approaches, which are currently available in point cloud processing software, can be considered as computer-aided manual data segmentation tools. More generally, limited progress has been made in the robust automated recognition of 3D CAD objects from range data, in particular in the AEC&FM context. The work presented herein makes this automated object recognition possible.

A first version of the approach presented here has been published in (Bosche and Haas 2008). This paper focuses on several significant improvements since that previous publication, and then, using data collected from a construction project, demonstrates the performance of the developed approach in terms of: (1) object recognition quality, robustness and time; and consequently (2) project 3D status visualization.

AN APPROACH FOR AUTOMATED RECOGNITION OF 3D CAD MODEL OBJECTS IN 3D LASER SCANS

Previous Work And Limitations

Three-dimensional object recognition in 3D laser scans is a model matching problem that has been investigated in the past. A review of common strategies for model matching object recognition can be found in (Arman and Aggarwal 1993), and some examples of systems for automated 3D object recognition in range images can be found in (Arman and Aggarwal 1990; Reid and Brady 1992; Johnson and Hebert 1999). One characteristic of these approaches, which were mainly developed for robotics applications, is that the pose of the objects in the scanned data is assumed to be unknown *a priori*. While this constraint reflects the most general situation for developing very robust approaches, in the context of the problem investigated here, this constraint can be removed.

In the AEC&FM industry, it is possible to register both 3D CAD models and 3D laser scans in a common coordinate frame using project 3D reference points, often referred to as *facility tie points* or *benchmarks*, that

are already used in surveying activities. As a result, in the investigated problem, tie point -based registration can be used, so that the pose of the search model objects in the scanned data can be assumed known *a priori*.

While the approaches referenced above could still be applied, they would remain limited, particularly because they are generally not robust with cluttered scenes with high levels of occlusion, such as construction site scanned scenes. This is particularly due to the fact that they search objects one at a time.

Overview of the Object Recognition Approach

The approach proposed here searches the entire 3D CAD model of a project at once in order to recognize each of its 3D objects, so that occlusions of model objects due to other model objects are taken into account. It consists of a series of five consecutive steps: (1) Convert the 3D CAD model into STL format; (2) Register the 3D model in the scan's spherical coordinate frame; (3) Calculate the as-planned scan; (4) Recognize the as-planned points; and (5) recognize the model objects.

A first version of this approach has been published in (Bosche and Haas 2008). However, several significant improvements have been made to the steps 4 and 5 of the approach, so that the approach is described again here, with an emphasis on those improvements.

1 - Convert the 3D CAD model

In order to use the 3D information contained in the 3D CAD model, full access to the model description is required. However, 3D CAD models are generally stored in protected proprietary 3D CAD engine formats (*e.g.* DXF, DWG, DGN, etc.). The authors have thus chosen to convert the 3D CAD models into the open-source STereoLithography (STL) format, which approximates the surface of 3D objects with tessellations of triangular facets. There are two main reasons why this format was chosen: (1) Conversion of 3D CAD models into STL format is faithful because any surface can be accurately approximated with a tessellation of triangular facets. Flat surfaces are exactly represented, but curved surfaces are approximated using a user-defined tolerance, the maximum chord height that is typically set to very low values to ensure faithful conversion (3D Systems Inc. 1989); and (2) This format is particularly adapted to the developed approach. In particular, it enables an efficient implementation of the calculation of the as-planned scan (see section 3 - *Calculate the as-planned range point cloud*).

2 - Scan-reference the 3D model

Scan-model registration information, that can be obtained in practice using *facility tie points* (also referred to as *benchmarks*), is used to reference the STL-formatted project 3D model in the scanner's spherical coordinate frame. In this coordinate frame, the world is viewed from the point of view of the scanner, in a similar manner to virtual or augmented reality.

3 - Calculate the as-planned range point cloud

For each original scanned range point, referred here to *as as-built range point*, a corresponding *as-planned range point* is calculated. It is first assigned the same pan and tilt angles as the ones of the as-built range point. Then, its range is calculated by performing the virtual single point scan defined by this direction and the 3D model as the virtually scanned world. The closest intersection point of this scanning direction with an object STL facet is the as-planned range point and thus defines its range. The as-planned point is additionally assigned, as an *IDobject* feature, the name or ID of the object to which the intersected facet belongs. Once the as-planned points corresponding to all the as-built points have been calculated, they can be sorted by their *IDobject* feature, so that each object is assigned an as-planned range point cloud.

Since the number of scanned range points and the number of objects (and consequently facets) in a 3D model can be very large, some means to reduce the complexity of the calculation of the as-planned range point cloud must be identified. It is observed that the problem of calculating each as-planned range point is similar to a problem faced in first-person shooter video games (*e.g.* “Doom” by Id Software). In such games, the surfaces of 3D objects constituting the environments (including the characters) are approximated with tessellations of triangles, and, at the moment when the player “shoots”, it must be identified which object is hit by the ray defined by the shot direction. A typical approach used to solve this video game problem is to pre-calculate the minimum bounding sphere of each object’s facet, and similarly of each object. These spheres present three advantages: (1) Their referencing with respect to a given referencing frame (in the case of the video game, the coordinate frame of the user who is moving in the environment) is very simple since only the center point needs to be referenced; (2) If a ray does not intersect the sphere, it cannot possibly intersect the facet (or object); and (3) It can be quickly calculated whether a ray intersects a sphere. As a result, the facets of which the ray intersects the bounding spheres can be quickly identified, and only those ones are further investigated for actual intersection with the ray — the intersection with the bounding sphere is a necessary, but not sufficient condition for the intersection with the facet (or object).

Such a pruning technique could be used here to reduce the complexity of the calculation of the as-planned range point cloud, particularly since this technique enables the video game problem be solved in real-time. However, it must be noted that it is optimized for the calculation of a single point (“shot”) from a given location, while the problem investigated here aims at finding the closest intersection point for a potentially very large number of scanning directions. As a result, a different pruning technique was developed. This technique works as follows: (1) The bounding pan and tilt angles of each object’s facet (and object) are calculated in the scan’s spherical coordinate frame; and (2) For each as-planned point scanning direction, its intersection is only calculated with the facets, whose bounding pan and tilt angles surround it. It is

demonstrated, in Section *Computational Performance*, that this technique performs here better than the sphere-based one.

4 - Recognize the as-planned points

For each object, each as-planned point can be matched to its corresponding as-built point. This requires a *point recognition metric*. Since they share the same pan and tilt angles, only their ranges need to be compared. The chosen point recognition metric is thus the comparison of the difference between the as-built and as-planned point ranges, $\Delta\rho$, with a pre-defined maximum threshold, $\Delta\rho_{max}$. If $|\Delta\rho|$ is smaller than or equal to $\Delta\rho_{max}$, then the point is recognized. The problem is then to effectively and automatically estimate a value of $\Delta\rho_{max}$ leading to good recognition results. It is suggested to calculate $\Delta\rho_{max}$ as a function of the mean registration error, $\overline{\epsilon_{Reg}}$, and a bounding value of the maximum expected project construction error, ϵ_{Const} , such that:

$$\Delta\rho_{max} = \overline{\epsilon_{Reg}} + \epsilon_{Const} \quad (1)$$

By taking into account both the error resulting from the construction process and the error resulting from the registration, $\Delta\rho_{max}$ values, which are estimated with this formula, enable robust point recognition results. A value of ϵ_{Const} must, however, be defined *a priori*. The authors have chosen a value of 50 mm, which they think is an acceptable bounding value of typical construction errors. The performance of this automated estimation of $\Delta\rho_{max}$ is demonstrated in Section *Performance Analysis*. Here, it is necessary to highlight that this automated estimation of $\Delta\rho_{max}$ is an improvement from the approach presented in (Bosche and Haas 2008) that used a manual estimation.

5 - Recognize the CAD objects

For each object, once all of its as-planned cloud points have been matched to their corresponding as-built points, it is possible to infer whether the object is recognized or not. This requires an *object recognition metric*. A basic metric might consider the number of recognized points. However, such a metric, which was originally proposed in (Bosche and Haas 2008), is not robust with different angular resolutions of scans and scanner-object distances.

Another metric, based on the object's *recognized surface*, is preferred. For each object, its recognized surface, $Surf_R$, is calculated as the weighted sum of its recognized points, where each point's weight is its *covered surface*. The covered surface of a point is roughly defined as the area delimited by the equidistant boundaries between it and its immediate neighboring points. It is calculated as a function of the scan's angular resolution, the as-planned point range and the as-planned point reflection angle — the angle between the point scanning direction and the normal to the surface from which it is obtained. The object's recognized facet is thus essentially invariant with these scan and point parameters.

Then, $Surf_R$ is compared to a threshold $Surf_{min}$. If $Surf_R$ is larger than or equal to $Surf_{min}$, the object is considered recognized. The problem is then to find an efficient approach to automatically estimate a value of $Surf_{min}$ leading to good recognition results. Since $Surf_R$ is invariant with the scan angular resolution, in order for the object recognition metric to remain invariant with this factor, $Surf_{min}$ must also be automatically adjusted with it. It is suggested that $Surf_{min}$ be calculated as a function of the maximum range between the scanner and the 3D model ($Model.\rho_{max}$), the scan's angular resolution (Res_φ and Res_θ), and a pre-defined minimum number of points (n), using the following formula:

$$Surf_{min} = n \tan(Res_\varphi) \tan(Res_\theta) (Model.\rho_{max})^2 \quad (2)$$

In Equation 2, n can be interpreted as the minimum number of points that must be recognized so that, at the range $Model.\rho_{max}$, their total covered surface is larger than $Surf_{min}$. Since all the objects in the model are located at ranges inferior or equal to $Model.\rho_{max}$, this ensures that, for each of them, at least n of its as-planned points will have to be recognized so that its recognized surface, $Surf_R$, is larger than $Surf_{min}$. The value of n must however be defined *a priori*. The authors have chosen for their experiments a value of $n = 5$ points. It is expected that this value be: (1) high enough to avoid Type I recognition errors that may result from the recognition of too few range points ($n < 5$) and (2) low enough to avoid Type II recognition errors that may result from the recognition of not enough range points ($n > 5$). While the choice of a larger value of n could be argued, the performance of this automated estimation of $Surf_{min}$ is demonstrated in section *Performance Analysis*.

PERFORMANCE ANALYSIS

Experimental Data

Experiments with real-life data are conducted to investigate the performances of the proposed approach in terms of: (1) object recognition quality and computational complexity; and consequently (2) project 3D status visualization. The data used here was obtained from the construction of a building that is part of a power plant project in downtown Toronto in Ontario, Canada (see section *Acknowledgements*). The building is 60m long by 15m wide by 9.5m high. It has a steel structure, the construction of which was the focus of the conducted experiments. Figure 1 presents the 3D CAD model and one scan (colored) of the building steel structure. The 3D model contains 612 objects with a total of 19,478 facets. The as-built data used in the experiments presented here consists of three scans, acquired with the TrimbleTM GX 3D laser scanner, that uses time-of-flight technology. Table 1 provides relevant information about each of the three scans.

Object Recognition Performance

In this section, we investigate the object recognition performance of the developed approach, and more particularly of the automated estimations of the thresholds $\Delta\rho_{max}$ and $Surf_{min}$.

Recognition Performance Metrics

First of all, the problem investigated here is an object recognition problem. The performance of the developed approach can thus be analyzed by using the common object recognition performance measures that are: the recall rate (or true positive or sensitivity rate), the specificity rate (or true negative rate), and the precision rate. In the investigated problem, these are defined as follows:

Recall: The number of properly recognized model objects divided by the total number of search model objects that are in the investigated scan.

Specificity: The number of properly not recognized model objects divided by the total number of model objects that are not in the investigated scan.

Precision: The number of properly recognized model objects divided by the total number of recognized model objects.

It must be noted that the calculations of these performance metrics require the manual estimation by visual inspection of which model objects are actually present in each scan. This estimation has been performed conservatively, so that the results are expected to be generally biased toward lower performances.

Experimental Results

It has been suggested that the value of $\Delta\rho_{max}$ be automatically estimated using Equation 1. Figure 2 shows the scan mean registration error $\overline{\epsilon_{Reg}}$, the automatically estimated value of $\Delta\rho_{max}$ (here, $\Delta\rho_{max} = 29.6 + 50 = 79.6 \text{ mm}$) and the recognition performances for different $\Delta\rho_{max}$ values, for *Scan 1* (presented in Table 1). In these experiments, $Surf_{min}$ is set to its automatically estimated value (here $Surf_{min} = 0.0109 \text{ m}^2$), which will be shown later in this section to be an appropriate value.

The results in Figure 2 first show that, overall, the developed approach, with automatically estimated $\Delta\rho_{max}$ and $Surf_{min}$ thresholds, achieves high recall, specificity and precision rates. This demonstrates that the overall approach performs well in general.

Then, Figure 2 shows that, for values of $\Delta\rho_{max}$ lower than $\overline{\epsilon_{Reg}}$, the recall rate is very low, although the precision and specificity rates are very high. But, for values of $\Delta\rho_{max}$ higher than $\overline{\epsilon_{Reg}}$, the recall rate is very much higher with not significantly lower precision and specificity rates. Therefore, using $\overline{\epsilon_{Reg}}$ as a minimum for $\Delta\rho_{max}$ is appropriate. The value of ϵ_{Const} of 50 mm also appears to be generally adequate. Similar results were obtained with other scans, and, overall, this automated estimation of $\Delta\rho_{max}$ appears to

lead to a good compromise between high recall rates on one side, and high specificity and precision rates on the other.

Then, it has been suggested to automatically set $Surf_{min}$ with Equation 2. Figure 3 shows, for *Scan 1* too, the automatically estimated $Surf_{min}$ value (here $Surf_{min} = 0.0109 m^2$) and the object recognition performances for different values of $Surf_{min}$ (note the logarithmic scale of the x axis). In these experiments, $\Delta\rho_{max}$ is set to its automatically estimated value (here $\Delta\rho_{max} = 79.6 mm$), which has already been shown to be appropriate.

The results in Figure 3 show that, for values of $Surf_{min}$ higher than the automatically calculated one, the recall rate is very low, although the specificity and precision rates are very high. But, for values of $Surf_{min}$ lower than the automatically calculated one, the recall rate is very much higher with not significantly lower specificity and precision rates. Therefore, the method for the automated estimation of $Surf_{min}$ appears to also lead to a good compromise between high recall rates on one side, and high specificity and precision rates on the other.

Computational Performance

Overall Computational Performance

First of all, note that the conversion of the 3D model into STL format (Step 1) only needs to be performed once whatever the number and locations of investigated scans. The complexity of this step is thus not important, so that it is discarded from the rest of this analysis.

Then, a set of three experiments is conducted to investigate the impact of the different approach process steps on the overall computational complexity. These experiments, *Experiment 1*, *Experiment 2* and *Experiment 3*, are conducted with the three scans, *Scan 1*, *Scan 2* and *Scan 3* respectively, and by considering the search of the entire 3D model. The computational times obtained for each of them for the steps 2, 3 and 4 and 5 combined are presented in Table 2. Note that these were obtained by running a VB.NET developed algorithm on a computer having a 2.41 GHz processor and 2 GB RAM memory.

It first appears in Table 2 that it takes, for instance in *Experiment 1*, overall only about three minutes to recognize the as-built point clouds of all the objects constituting the 3D model of the project (here 612 objects) from the original scan containing 810,399 points. Considering the size of the model and scanned data sets, as well as the object recognition performances presented in the previous section, these computational times can be argued to be relatively short.

Table 2 also shows that Steps 2 and 3 are the most critical ones in terms of computational times. Their relative impact must however be further discussed. First of all, the computational time of Step 2 is strongly correlated with the value of a parameter, *Incr*. The parameter *Incr* is used to approximate each facet's

edge by a series of evenly spaced points (*Incr* specifies the size of the space) so that the bounding tilt angles of each facet are calculated as the bounding tilt angles of the sets of points approximating its edges. In the experiments conducted here, *Incr* is set to 10 *mm*. The same experiment conducted with a value of 100 *mm* leads to the same object recognition results, but with a computational time of Step 2 being an order of magnitude smaller. This indicates that the value of *Incr* could be adjusted to different situations. In fact, *Incr* could be adjusted automatically for each facet as a function of the facet’s bounding pan angles, the distance of the facet to the scanner, and the scan angular resolution. This automated estimation of *Incr* has however not been investigated at this point, and the small value of 10 *mm* is used to ensure a good estimation of the bounding tilt angles of any facet, despite its negative impact on the computational time of Step 2.

Besides, it is expected that scanned range point clouds investigated in real life applications contain far more points than the scans used here. In fact, laser scanners of new generation already enable the acquisition of scans with angular resolutions down to about 150 μ *rad*, which is four to ten times denser than the scans used here. As a result, it is expected that, in practice, the computational time of the Step 3 becomes much longer than the one of Step 2. Furthermore, if it is decided to place a laser scanner for long periods or even the entire duration of a project at a fixed location and conduct many scans from that location, then the Step 2 only has to be conducted once for all those scans, reducing even more its impact on the overall computational complexity.

Performance of the Pruning Technique

It is then of interest to compare the combined computational times of Steps 2 and 3 for the proposed method, with those that would be obtained using other pruning techniques, such as the sphere-based technique that is used in first-person shooter computer games, as described earlier.

A new experiment, *Experiment 3'*, is thus conducted with *Scan 3* and the developed object recognition approach, but using a sphere-based pruning technique implemented as follows:

1. The center of the minimum bounding sphere of each STL facet and each STL object is calculated off-line (prior to conduct scans). The calculation of the bounding spheres of a set of n 3D points has been a very intensively investigated problem. In the experiment conducted here, the approach presented in (Ritter 1990) is used for the calculation of the bounding sphere of each STL object. Note that this approach does not calculate the exact minimum bounding sphere, but is a computationally efficient approach to accurately estimate it. The calculation of the minimum bounding sphere of a STL facet is a special case of the problem solved in (Ritter 1990), and for which the solution can be obtained using a simple deterministic method.

2. During the scan-referencing (Step 2), the bounding spheres are also scan-referenced, which only requires the scan-referencing (rotation and translation) of their centers.
3. For the calculation of each as-planned range point (Step 3), the pruning consists in investigating the intersection of its scanning direction with only those facets (objects) for which the minimum bounding spheres intersect the scanning ray.

The computational times of Steps 2 to 5 obtained for *Experiment 3'* are presented in the last column of Table 2, and can be compared with the times obtained for *Experiment 3*. It clearly appears that the pruning technique based on the bounding angles is significantly more computationally efficient here. The reason is that we deal with many rays (millions), while the sphere-based technique is specifically developed for dealing with one ray. While the proposed method requires more time for Step 2, it has a significantly lower computational time for Step 3.

Note that, the computational time of Step 2 being far smaller for the sphere-based technique (and it could actually be reduced even more), this technique remains significantly more efficient for solving the first-person shooter video game problem where the intersection of only one ray has to be calculated.

3D Project Status Visualization

3D Project Progress Status

The recognition results obtained for any given scan with the developed approach can be used to display accurate information about the current 3D status, and consequently the 3D progress, of the project at the time of the scan, to the user, who is typically the project management team. For instance, Figure 4 displays the object recognition results obtained in *Experiment 1*. In this figure, the scanner is represented at the location from which the investigated scan was conducted for a proper interpretation of the results. Also, each of the 3D model objects is colored in one of three colors with the following meanings:

Gray: The object is not expected (planned) to be recognized in this scan.

Green: The object is expected to be recognized and is recognized in the scan.

Red: The object is expected to be recognized, but is not recognized in the scan. This must, however, not lead to the conclusion that the object is not built. Several situations must in fact be distinguished.

The object is in the scan. It is then colored in red because it is built, but at the wrong location.

The object is not in the scan. This may occur in three different situations:

- The construction is behind schedule.

- The search 3D model does not adequately represent the project in the state it is expected to be found.
- The object is occluded by another object that is not part of the 3D model (*e.g.* piece of equipment).

Since, an object colored in red may mean different things, it must be interpreted as a *warning*, or *flag*, implying that this particular object requires further analysis. Note that, two of the four situations identified above can be somewhat avoided using good practice. First, a more realistic 3D model can be used for the object recognition by using a project 4D model instead of the 3D model. Since a 4D model is constructed with the project schedule, an up-to-date schedule would enable the extraction of a 3D model that would better reflect the true state of the project at the time of the scan. Then, external occlusions (occlusions to non-model objects) can be avoided by cleaning the scanned site prior to conduct any scan and locating the scanner so that external occlusions due to objects that cannot be removed are minimized. If these best practices are implemented, an object colored in red will then indicate either that it is built at the wrong location, or that construction is behind schedule, the first case being easily identifiable by investigating the scan manually.

In the example in Figure 4, it can first be seen that most of the 3D model objects (exactly 466 objects) are expected to be recognized in the investigated scan. Out of these, a majority of them (exactly 280 objects) is actually recognized in the scan. While it can be noted that the main structural elements are well recognized, 186 elements still are not recognized (colored in red). But, as mentioned above, these objects may not be recognized for several reasons. For instance, the 131 objects constituting the small inner structure at the back of the building, and the six purlins colored in red in the ceiling of the structure, were not expected to be built at the time of the scan. Therefore, they are not recognized because the entire project 3D model was searched in the scan, instead of a more realistic 3D model that would be extracted from the project's 4D model. Then, the small elements around two of the door frames in the front long wall are not recognized because they were occluded by other non-model objects, such as a set of portable toilets.

Next, the three door frames colored in red in the front long side (6 objects) are in the scan, but not recognized. It was thus concluded that they were built at the wrong location, which has been confirmed visually.

Finally, many of the furthest objects from the scanner are missed, including 5 fairly large joists and the column in the middle of the back side of the building. An important reason why they are not recognized is that, from the scanner's location, they are largely occluded by other model objects, so that only small parts of their total surfaces were actually expected to be recognized in the scan, and their recognized surfaces

often simply fell short of the $Surf_{min}$ threshold. Another important reason for failing to recognize objects is registration error. The impact of registration error on the proposed approach is discussed in more detail in the section *Impact of Registration Error*, below.

Element 3D Quality Status

Further than the simple binary recognition of objects, the results provided by the developed approach enable a more detailed analysis of the matching between the as-built and corresponding as-planned object. First of all, the differences between the as-planned and as-built ranges of each object's point can be mapped to display some potential small location and orientation issues for each object to the user. Figure 5, for instance, displays (1) the designed 3D representation of a column part of the 3D model used in the experiments presented in this paper, as well as (2) the recognized point cloud of that column automatically recognized in *Scan 1 (Experiment 1)*. Each recognized range point is colored with respect to the difference between its as-planned and corresponding as-built ranges, $\Delta\rho$. This mapping of the $\Delta\rho$ values obtained for all the recognized points enables the user visualizing whether the object is generally correctly positioned. For instance, if all the points have colors in the yellow-red ranges, such as in the example in Figure 5, then it may be concluded that the object generally has a proper orientation, but is built slightly too close to the scanner. On the contrary, if all the points have colors in the yellow-green ranges, then it may be concluded that the object generally has a proper orientation, but is built slightly too far from the scanner. Finally, if the points have colors that significantly vary from one end of the object to the other end, then it may be concluded that the object's orientation (*e.g.* plumbness) is not correct. Overall, note that the developed approach being rapid, it could be used with this color coding for real time steel frame alignment during installation operations.

Further than this visual analysis, an *automated object pose quality control approach* can be envisioned. Such an approach would automatically fit to the recognized as-built range point cloud a similar parametric object as the one used for its design. Many parametric form fitting algorithms exist and some are already available in point cloud management software. For instance, Kwon et al. (2004) present approaches to fit parametric primitives (spheres, cylinders and boxes) to range data, and a general approach for recognizing parameterized objects in range data is described in (Chenavier et al. 1994; Reid and Brady 1995). Once the form is fitted to the point cloud, the parameters of the fitted form can be compared to the parameters of the designed form to infer location and orientation error information that is suited for comparison with tolerances typically provided in project specifications and that could be previously automatically extracted for each object (see the investigative work presented in (Boukamp and Akinici 2007)).

Impact of Registration Error

The performance results presented here are generally promising, if not good, but not necessarily as good as ultimately expected (100% recall rates, and 0% Type I and II error rates). There is, however, one particular reason for the observed lower performances: the experiments conducted here use registration data of poor quality. The mean registration error $\overline{\epsilon_{Reg}}$ for the three scans is in average equal to 21.8 mm, which is large and thus likely has a significant impact on the object recognition results — even if, to a certain degree, $\Delta\rho_{max}$ takes this error into account. The reason for these high registration error values is that facility tie points were not scanned when the scans were originally conducted. As a result, manual point matching had to be used, which typically leads to larger registration errors.

In the industry, scan registration error specifications are far more stringent with values of a couple of millimeters. With such registration errors, that can be achieved by using *facility tie points*, it is expected that object recognition results achieved by the developed approach be even better.

CONCLUSIONS

This paper presented an automated approach for the recognition of 3D CAD model objects in 3D laser scans, with a specific emphasis on validation performed using large data sets obtained from a construction site. This approach presents significant improvements to the previously published version in (Bosche and Haas 2008). In this paper, the object recognition performance of the approach is first demonstrated. In particular, the methods proposed for automatically estimating the two thresholds, $\Delta\rho_{max}$ and $Surf_{min}$, used for point and object recognition respectively, appear effective as they lead to good compromises between high recall rates on one side, and high specificity and precision rates on the other side. It is then shown that this approach demonstrates good computational efficiency, due to the use of a pruning technique that works well with the investigated problem. Finally, the object-level results provided by the developed approach can be used to display to the user the 3D status of a project, and more interestingly of all its components, for faster and better management decisions. Ultimately, applying this method to problems, such as automated progress and productivity tracking as well as automated dimensional QA/QC, provides a tremendous opportunity.

Many questions remain to be addressed. In particular, the impact of registration error must be further investigated. Fusing recognition results from different perspectives may lead to better performances. Exploitation of 3D image data not related to 3D CAD objects (as-built range points corresponding to none of the model objects) may be possible. The information produced by the as-planned point cloud generation step may be used to plan scans to achieve maximum efficiency during data collection.

ACKNOWLEDGEMENTS

This project is partially funded by a grant from the National Science Foundation grant #0409326 and

the Canada Research Chair in Construction & Management of Sustainable Infrastructure.

The authors would also like to thank SNC Lavalin and in particular Paul Murray for their support for this research, in particular for allowing Frederic Bosche to come to its project site, conduct some scanning, and publish these results.

APPENDIX I. REFERENCES

- 3D Systems Inc. (1989). “Stereolithography interface specification.” *Report No. 50065-S01-00*, 3D Systems Inc.
- Arman, F. and Aggarwal, J. (1990). “Object recognition in dense range images using a CAD system as a model base.” *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, Cincinnati, OH, USA. 1858–1863.
- Arman, F. and Aggarwal, J. (1993). “Model-based object recognition in dense-range images: a review.” *Computing Surveys (CSUR)*, 25(1), 5–43.
- Bosche, F. and Haas, C. T. (2008). “Automated retrieval of 3D CAD model objects in construction range images.” *Automation in Construction*, 17(4), 499–512.
- Boukamp, F. and Akinici, B. (2007). “Automated reasoning about construction specifications to support inspection and quality control.” *Automation in Construction*, 17(1), 90–106.
- Chenavier, F., Reid, I., and Brady, M. (1994). “Recognition of parameterized objects from 3D data: a parallel implementation.” *Image and Vision Computing*, 12(9), 573–582.
- Johnson, A. E. and Hebert, M. (1999). “Using spin images for efficient object recognition in cluttered 3D scenes.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5), 433–449.
- Kwon, S.-W., Bosche, F., Kim, C., Haas, C. T., and Liapi, K. A. (2004). “Fitting range data to primitives for rapid local 3D modeling using sparse point range clouds.” *Automation in Construction*, 13(1), 67–81.
- Reid, I. and Brady, M. (1992). “Model based recognition and range imaging for a guided vehicle.” *Image and Vision Computing*, 10(3), 197–207.
- Reid, I. D. and Brady, J. M. (1995). “Recognition of object classes from range data.” *Artificial Intelligence Journal*, 78(1-2), 289–326.
- Ritter, J. (1990). *Graphics gems*, chapter An efficient bounding sphere, 301–303. Academic Press Professional, Inc., San Diego, CA, USA.

List of Tables

1	Number of scanned points, resolution and mean registration error for the three scans used in the experiments.	17
2	Computational times (in seconds) of the Steps 2 to 5 of the recognition process for the four experiments: <i>Experiment 1</i> , <i>Experiment 2</i> , <i>Experiment 3</i> and <i>Experiment 3'</i>	18

TABLE 1. Number of scanned points, resolution and mean registration error for the three scans used in the experiments.

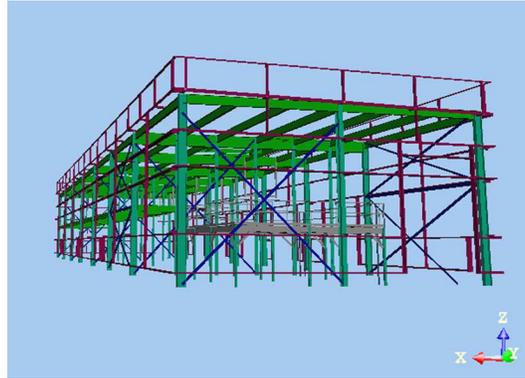
Scan	Number of range points	Resolution (μrad)		$\overline{\epsilon_{Reg}}$ (mm)
		Hor.	Vert.	
1	691,906	582	582	36.86
2	723,523	582	582	45.49
3	810,399	582	582	29.57
4	650,941	582	582	16.26
5	134,263	300	300	19.54

TABLE 2. Computational times (in seconds) of the Steps 2 to 5 of the recognition process for the four experiments: Experiment 1, Experiment 2, Experiment 3 and Experiment 3'.

Process Steps	Experiment:			
	1	2	3	3'
Step 2 - Scan-referencing:	59.0	56.5	57.2	1
Step 3 - As-planned point cloud:	141.9	109.1	16.8	450.8
Steps 4+5 - Point and object recognition:	15.5	11.2	3.5	2.9
Total (Steps 2+3+4+5):	216.4	176.8	77.5	454.7

List of Figures

1	Steel structure of the investigated PEC project building.	20
2	Object recognition performance with <i>Scan 1</i> , for different values of $\Delta\rho_{max}$, the mean registration error $\overline{\epsilon_{Reg}}$, and the automatically calculated value of $\Delta\rho_{max}$	21
3	Object recognition performance with <i>Scan 1</i> , for different values of $Surf_{min}$, and automatically estimated value of $Surf_{min}$	22
4	<i>Scan 1</i> (top) and the 3D model object recognition results obtained with the developed approach (bottom).	23
5	Model of a structural column (left) and its corresponding as-built range point cloud extracted from <i>Scan 1</i> (right).	24



(a) CAD model



(b) 3D laser scan

FIG. 1. Steel structure of the investigated PEC project building.

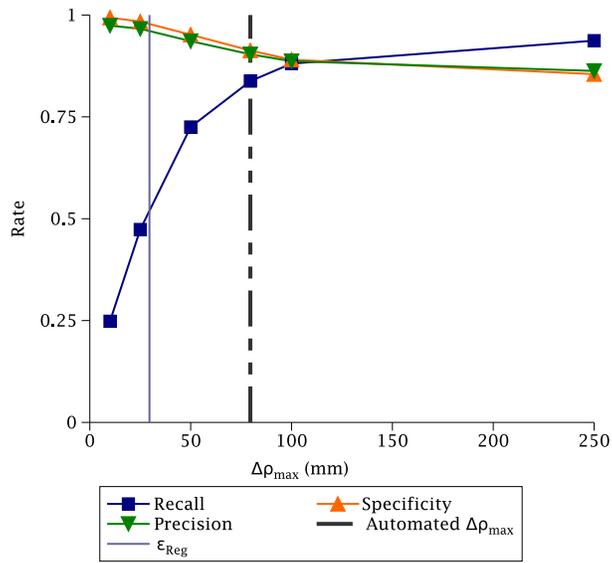


FIG. 2. Object recognition performance with Scan 1, for different values of $\Delta\rho_{max}$, the mean registration error $\overline{\epsilon_{Reg}}$, and the automatically calculated value of $\Delta\rho_{max}$.

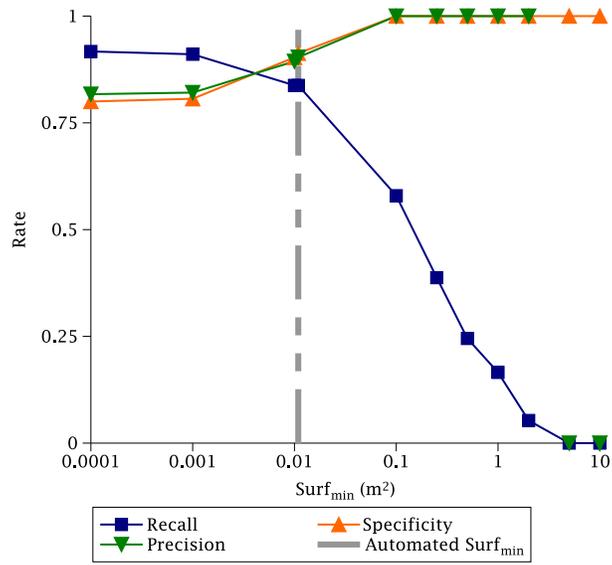


FIG. 3. Object recognition performance with Scan 1, for different values of $Surf_{min}$, and automatically estimated value of $Surf_{min}$.

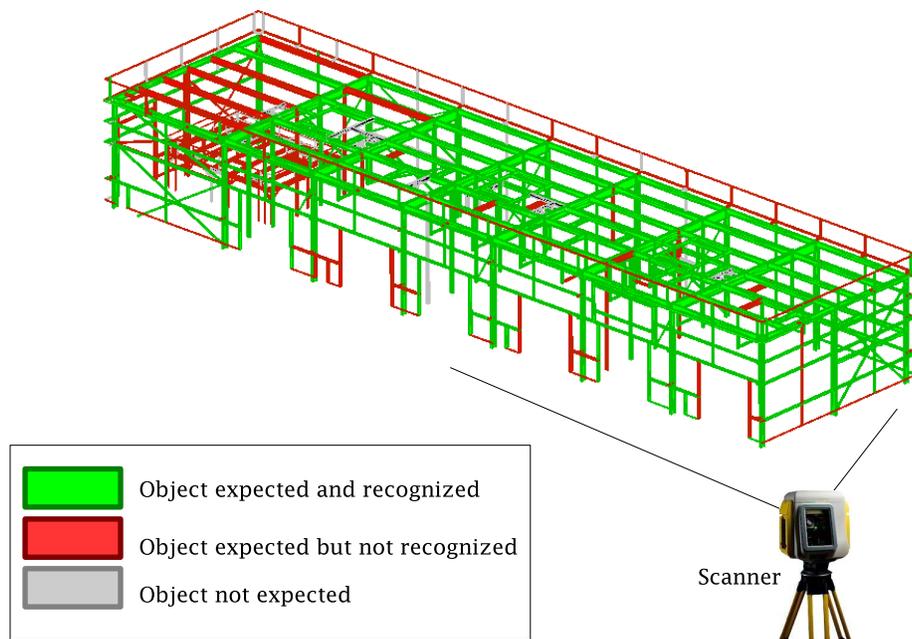
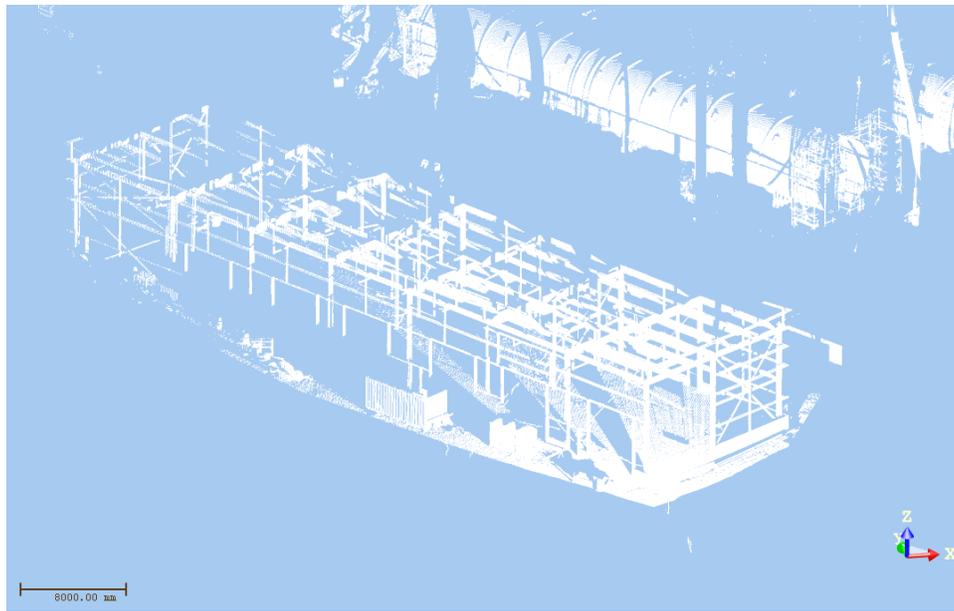


FIG. 4. Scan 1 (top) and the 3D model object recognition results obtained with the developed approach (bottom).

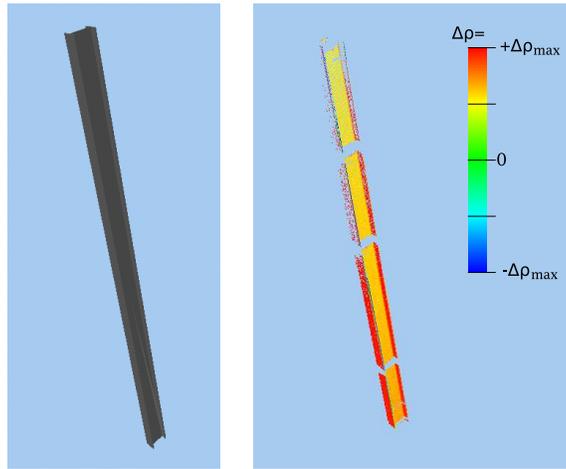


FIG. 5. Model of a structural column (left) and its corresponding as-built range point cloud extracted from Scan 1 (right).