Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction

Bosche, Frederic Nicolas

Published in:
Advanced Engineering Informatics

DOI:
10.1016/j.aei.2009.08.006

Publication date:
2010

Automated Recognition of 3D CAD Model Objects and Calculation of As-built Dimensions for Dimensional Compliance Control in Construction

Frédéric Bosché[1]

Computer Vision Laboratory
ETH Zürich

Abstract

The construction industry lacks solutions for accurately, comprehensively and efficiently tracking the three dimensional (3D) status of buildings under construction. Such information is however critical to the successful management of construction projects: it supports fundamental activities such as progress tracking and construction dimensional quality control. In this paper, a new approach for automated recognition of project 3D Computer-Aided Design (CAD) model objects in large laser scans is presented, with significant improvements compared to the one originally proposed in [11]. A more robust point matching method is used and registration quality is improved with the implementation of an Iterative Closest Point (ICP) based fine registration step.

Once the optimal registration of the project's CAD model with a site scan is obtained, a similar ICP-based registration algorithm is proposed to calculate the as-built poses of the CAD model objects. These as-built poses are then used for automatically controlling the compliance of the project with respect to corresponding dimensional tolerances.

Experimental results are presented with data obtained from the erection of an industrial building's steel structure. They demonstrate the performance in real field conditions of the model registration and object recognition algorithms, and show the potential of the proposed approach for as-built dimension calculation and control.

Key words: Automation, 3D, Object Recognition, Dimensional Compliance Control, Laser Scanning, CAD Model, Registration

1. Introduction

It has been repeatedly reported that construction performance is correlated to the availability and accuracy of information on site status, in particular three-dimensional (3D) status [29, 2, 37]. Current techniques for project 3D status tracking are based on manual measurements that are time- and labor- demanding, and therefore too expensive and often too unreliable to be comprehensively applied on sites [14, 29]. There is therefore a need for effective and efficient project 3D status tracking.

1.1. Existing Research

Many research initiatives are investigating the use of remote sensing technologies, in particular digital imaging and terrestrial laser scanning, to improve the efficiency and effectiveness of site 3D data collection for project control activities, in particular progress tracking and dimensional compliance control.

1.1.1. Vision based Systems

Several vision-based systems have been proposed for tracking construction progress [26, 42, 14, 25, 19]. Their general strategy is to first register site digital pictures and the project 3D Computer-Aided Design (CAD) model in a common coordinate system using camera pose estimation techniques. The next step is to compare the site digital pictures to the project model. To do this comparison, different alternative approaches have been proposed. In [47, 42, 14], each 2D image

[1]Computer Vision Laboratory - DDM Group, ETH Zürich, HIT H 11.1, Wolfgang-Pauli-strasse 27, 8093 Zürich, Switzerland; +41 44 633 70 12; bosche@vision.ee.ethz.ch; http://people.ee.ethz.ch/~bosche

is compared with a corresponding virtual image generated using the 3D model and the estimated camera pose; In [26] a 3D model is reconstructed from the site pictures and then compared to the CAD model. Finally, in [25, 19], Lukins et al. propose a similar approach to [42, 14] but with an additional first step comparing each picture with pictures previously acquired with the same camera pose. The presence of newly built elements is then only searched in *regions of interest* identified as regions of the images presenting significant changes since the previous shot. The work of Lukins et al. [25, 19] appears to be the most promising by its level of automation (human intervention is only required for the installation of the camera and its initial registration with the project CAD model), its robustness to varying environmental conditions, and its capacity to distinguish regions of interests for progress detection. Nonetheless, several limitations remain such as: (1) the size and type of detectable changes; (2) the sensitivity of the region-of-interest and object detectors to changing lighting conditions, particularly in the presence of shadows; (3) the robustness to *unexpected occlusions*[2] — unexpected occlusions are very common on construction sites and may result in false detection of regions of interest, which may then result in false object detection; and (4) the impossibility to resolve depth ambiguities.

In the case of dimensional compliance control, Ordóñez et al. [30] proposed two different image-based approaches for controlling dimensions of flat elements. These approaches, however, require significant human input as points to be measured must be selected manually. Shih and Dunston [40] also recently presented results on the evaluation of Augmented Reality (AR) for steel column inspection (anchor bolt positions and plumbness). These results demonstrate the feasibility of accurate AR-based steel column inspection. However, the system requires not only significant human interaction, but also a user with strong skills in 3D visualization and 2D projection necessary for the registration of the models in the images. As a result, despite the accuracy of the results reported by the authors, inspections can be time-consuming and expensive.

Overall, as already noted by Rabbani and van den Heuvel [35], despite the major advances that have been made in image processing in the last couple of decades, image-based approaches, particularly when not implementing dense stereo-vision, will always remain limited because they aim at extracting dense 3D information from 2D images. This is an ill-conditioned problem that is very difficult to solve, particularly in the case of complex scenes such as construction sites. Goldparvar-Fard et al. [14] do propose a method enabling the extraction of sparse 3D data from series of images (using the principle of structure-from-motion). However, no results have yet been published demonstrating the feasibility and reliably of automatically analyzing this sparse 3D data to infer such information as progress or dimensional quality.

1.1.2. Laser Scanning based Systems

Contrary to digital imaging, laser scanning actually acquires 3D data, and with good accuracy (*e.g.* ~ 12 *mm* at 100 *m* with the scanner used in this research [44]) and high point density (*e.g.* maximum horizontal/vertical resolutions of ~ 60 μrad with the scanner used in this research, which results in about one point per 3 *mm* at 50 *m* [44]). Thus, despite their initial cost (between $\$50,000$ and $\$100,000$ for a standard scanner), laser scanners present characteristics that are well adapted to project 3D status tracking, and thus progress tracking and dimensional quality control [13, 16, 15, 3]. For example, Gordon et al. [16] present a system that uses a 3D free-form shape recognition algorithm for automatically recognizing CAD objects in laser point clouds. Shapes are recognized by representing, and subsequently matching, model and sensed data using local surface descriptors called *spin-images* [21]. This approach is very general and can be very robust to occlusions because it is based on local features. However, its performance significantly decreases in complex situations, such as in construction sites where search objects do not necessarily have very distinctive features. Indeed, in object recognition, it is important that the data representation be *unambiguous*: no two objects should have the same representation [4]. The problem with building 3D models is that the objects that constitute them (*e.g.* columns, beams) present strong similarities resulting in strong ambiguities.

Shih and Wang [39] reported a laser scanning- based system for controlling the dimensional compliance of finished walls. Biddiscombe [8] reported the use of laser scanning on an actual tunneling project for controlling as-built dimensions. Similarly, [17] and [31] reported results on the use of laser scanning for structural health monitoring. Despite these many works and the industry-wide acknowledgement of the accuracy and versatility of laser scanners [20], the use of laser scanning on construction sites remains very limited. We suppose that the

[2] *Expected occlusions* refer to occlusions due to objects that are part of the building of which we use the 3D CAD model (*e.g.* beam, column, floor, door). On the contrary, *unexpected occlusions* refer to occlusions due to objects that are not part of that building (*e.g.* temporary structures, workers, and/or other buildings).

probable reason is that the currently proposed systems (such as those above) present low levels of automation and/or poor efficiency (*e.g.* in terms of the number of elements that can be investigated in a day), resulting in benefit-cost ratios which are not favorable compared to those achieved with traditional manual inspections.

1.1.3. Our Previous Work

Bosche et al. [11] recently proposed a quasi fully-automated system for recognizing project 3D CAD model objects in site laser scans. The focus is on large site scans that capture data from many objects, if possible the entire site, simultaneously. Each investigated scan and the project 3D CAD model are first registered in the same coordinate system using a standard manual *n*-point registration approach (this is the only manual step). The as-built point clouds corresponding to the CAD objects present in the scan are then automatically recognized as the projections (similarly to ray casting) of the as-built points onto the objects' models. A threshold is used for rejecting matching point pairs that are too far apart. Finally, the recognition of each CAD object is inferred by analyzing the set of scanned points with matching points on the surface of the object's model. The system is very robust to occlusions (both unexpected and expected occlusions), fairly efficient (it takes around 5 *min* to process a scan containing approximately $650,000$ points with a CAD model constituted of 612 objects) and quite accurate (in average, 80% of objects are recognized). However, this approach is unreliable enough because its accuracy depends entirely on the accuracy of the registration step, which is performed manually and relies on a few point pairs only.

In [11], the authors describe how the results obtained with this system could be used for automated project control. In the same article, they show in fact the feasibility of performing automated project progress tracking using such object recognition information. In [10], they also discuss further the feasibility of using the as-built point clouds that can be extracted from the scans for all recognized objects to automatically calculate the as-built dimensions of these objects. However, no implementation was proposed.

1.2. Contribution

The first contribution of this paper is an improved algorithm for recognition of project 3D CAD model objects in construction laser scans (Section 2). Compared to the algorithm originally proposed in [11], a more robust point matching method is used and registration quality and reliability is improved with the implementation of an ICP-based fine registration step. It is shown through multiple experiments that the resulting object recognition system has the following strengths:

- *Automation*: it is quasi fully automated, as only the initial *n*-point coarse registration must be performed manually;
- *Accuracy*: it achieves high object recognition accuracy, and this accuracy is much less sensitive to the initial manual coarse registration;
- *Robustness*: it is robust to occlusions, both expected and unexpected;
- *Efficiency*: each scan can be processed in a matter of minutes.

The second contribution of this paper is an algorithm for automatically calculating the as-built pose of the recognized CAD objects, and controlling the compliance of the project with dimensional tolerances (Section 3). The current algorithm focuses only on calculating the as-built poses of the objects (an extended version will be developed that will simultaneously calculate their as-built shapes). It is shown through multiple experiments that the proposed as-built pose calculaton and dimensional compliance control system has the following characteristics:

- *Automation*: it can be fully automated (the as-built pose calculation part is already fully automated, only the dimensional compliance control part is not);
- *Accuracy*: results for as-built dimension calculation and dimensional compliance control are promising. However, current experimental results are not conclusive, partially due to the limited accuracy of current laser scanners on field;
- *Robustness*: the calculated as-built dimensions do not seem to suffer from partial occlusions of the investigated objects;
- *Efficiency/Scalability*: the as-built poses of all installed/built objects recognized in a scan can be obtained in a matter of minutes.

1.3. System Overview

The first part of the proposed system is an algorithm that enables quasi automated recognition of project 3D CAD model objects in construction laser scans. As summarized in Figure 1, the algorithm first requires that the laser scan be coarsely registered with the project 3D CAD Model using a simple *n*-point registration approach (with $n \geq 3$). This process can be performed rapidly using commercial software, but does require some human input for the selection of at least three matching point pairs in the scan and model. As noted earlier, such coarse registration may not lead to optimal alignments of the scan and CAD model. As a result, an

automated fine registration of the scan and CAD model is implemented. This fine registration is referred to as the *model fine registration* because it uses the 3D CAD model constituted of all its objects. The recognition of each CAD object is then inferred, as done in [11], by analyzing the set of scan points that are close to, and thus match, the surface of the object's model.

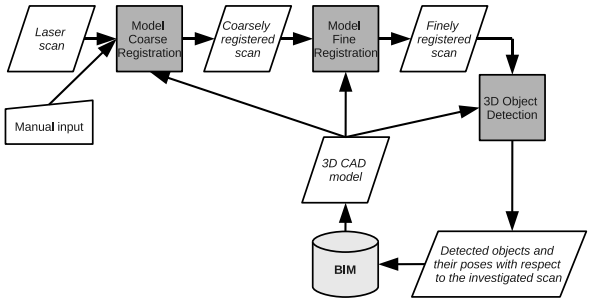


Figure 1: Flowchart of the algorithm enabling the recognition in a site laser scan of the objects constituting the project's 3D CAD model. *BIM* refers to *Building Information Model*.

The second part of the proposed system is an algorithm that calculates the as-built poses of the 3D CAD model objects recognized in a scan. These poses are then used to control the compliance of the project with respect to dimensional tolerances. As summarized in Figure 2, the algorithm first calculates the as-built poses of the recognized 3D CAD model objects using an *object fine registration* process. This registration uses the same algorithm as the one used in the *model fine registration*, but optimizes the pose of each object individually from the other objects. Once the as-built poses of all recognized objects are calculated, all dimensions that can be calculated from them, and for which compliances with tolerances must be met, are controlled.

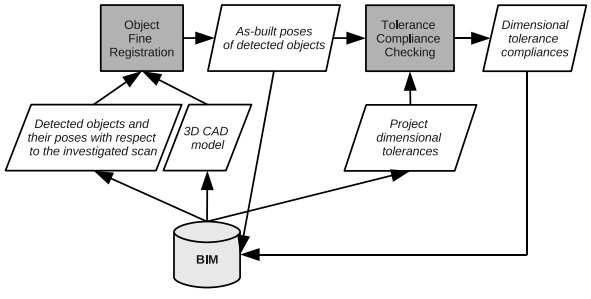


Figure 2: Flowchart of the algorithm enabling (1) the calculation of the as-built poses of all project 3D CAD model objects recognized in a site laser scan; and (2) the control of the compliance of the project with respect to dimensional tolerances.

2. Automated CAD Object Recognition

As shown in Figure 1, the CAD object recognition algorithm is divided into three steps: model coarse registration, model fine registration and object recognition. These steps are described in the following three sub-sections. Section 2.4 presents experimental results demonstrating the performance of this approach by comparing it to the approach originally proposed in [11].

2.1. Model Coarse Registration

The first step consists in obtaining a *coarse registration* of the investigated scan and the project 3D CAD model. For this, a *n*-point registration approach is used that consists in manually picking at least three pairs of matching points in the scan and model. This can be easily performed using standard commercial software packages for point cloud analysis (*e.g.* Trimble Realworks [45] or Leica CloudWorks [23]).

In [11], the authors performed the model-scan registration using this method only. However, it is known that such *coarse registration* is not reliable because it rests on only a few pairs of matched points for which the correspondences are not even ensured. In fact, one of the purposes of the registration is here to perform dimensional quality control which aims at verifying these correspondences. In order to achieve a more reliable and improved registration, an additional step should be implemented that checks, and if necessary improves, the quality of the registration. This is achieved by the *model fine registration* algorithm presented below.

2.2. Model Fine Registration

2.2.1. Previous Work

An optimization algorithm can be designed to start from a coarse registration and then locally search for a better one (in the registration space) [7]. Such a local optimization problem, generally referred to as *fine registration*, has been intensively studied in the past. A breakthrough came with the works by Arun et al. [5] and Horn [18] for least-squares estimation of registration parameters between two point clouds *with correspondence* (pairs of corresponding points in the two clouds are known *a priori*). Then, building on those prior results, Besl and McKay [7] and Chen and Medioni [12] simultaneously proposed two similar methods for the fine registration of two 3D shapes *without correspondence*. Their methods use a similar iterative algorithm, commonly referred to as the *Iterative Closest Point (ICP)* algorithm. Note that Besl and

McKay [7] refer to the registration of *3D shapes* because this algorithm can be used with 3D data sets in many different representations including point clouds, triangular sets, implicit surfaces and parametric surfaces.

The ICP algorithm is summarized in Algorithm 1 with the *Data* shape represented as a point cloud X_o. q_k is the registration obtained at the k^{th} iteration and contains 7 parameters: the scaling factor (in our case, it equals 1), the three rotation angles and the three translation distances. At each iteration, the function $C()$ calculates the points on the *Model* shape matching the given *Data* (shape) points. The function $R()$ calculates the least square optimal registration between the matched *Data* and *Model* points (*e.g.* using the method in [5] or [18]). Finally, the function $T()$ calculates whether the termination criterion is met. If the criterion is met, the algorithm stops; otherwise it goes through an additional iteration.

Input: $Data = \{P_D\}_N$, $Model$
Result: q

`Initialization:`
$X_0 = \{P_D\}_N$;
$q_0 = \{1, 0, 0, 0, 0, 0, 0\}^T$;
$k = 0$;
Continue = True;

`Iterative Search:`
while Continue = True **do**
  $X_k = q_k(X_0)$;
  $Y_k = C(X_k, Model)$;
  $q_{k+1} = R(X_0, Y_k)$;
  Continue = $T(\{X_i | i \in [1;k]\}, \{Y_i | i \in [1;k]\})$;
  $k = k + 1$;
end
$q = q_k$;

Algorithm 1: Generic *Iterative Closest Point (ICP)* algorithm for fine registering a *Data* shape with a *Model* shape. Here, the *Data* shape is assumed to be represented as a point set.

Since the works of Besl and McKay [7] and Chen and Medioni [12], many variants of the ICP algorithm have been proposed. Variations have been proposed for: the selection of *Data* points, the identification of matching *Model* points, the error metric to be minimized and the termination criterion. Since the calculation of the matching points is the most computationally expensive part of ICP algorithms, several accelerations methods have also been proposed. A good review of many of these variants published up to 2001 can be found in [38].

In the specific case of the fine registration of dense point clouds with CAD models, which is the problem we deal with here, several works have been published. These mainly have application in dimensional compliance control in manufacturing [28, 43, 34, 33]. One interesting approach has been proposed by Moron et al. [28] for fine registration of a dense point cloud with the object 3D model in STL format (triangulated mesh), and has the following characteristics: (1) all the scanned points are used; (2) matched *Model* points are the closest points on the *Model* shape (in terms of Euclidean distance); (3) no weighting strategy, in particular no outlier[3] detection and removal strategy, is considered; and (4) the calculation of matching points is accelerated using a kd-tree.

All the approaches for part inspection in the manufacturing industry, including the one by Moron et al. [28], generally assume that all the scanned data points are acquired from the surface of the object being controlled, or that *Data* points acquired from other objects (*e.g.* scan table) can be easily manually removed prior to perform the fine registration — this explains why Moron et al. [28] do not implement any method for identifying and removing outliers. However, in the case of uncontrolled environments such as construction sites, large site scans typically include many points acquired from objects that are not part of the project 3D CAD model (*e.g.* equipment, tools, temporary structures and people), and manually cleaning such scans would be far too time-consuming. This implies that any algorithm for fine registration of construction site scans with project CAD models should implement a robust point matching method capable of detecting and rejecting outliers.

2.2.2. Proposed Method

Based on the analysis above, the following ICP algorithm is proposed for performing the fine registration of a large site laser scan with a 3D model of the building under construction:

- *Selection of Data points*: All *Data* points are used (a method for robust data sampling is also proposed in Section 2.4.2 to significantly reduce the processing time without compromising the accuracy of the result).
- *Calculation of matching Model points*: Similarly to Moron et al. [28], the model is considered to be in a format in which the surfaces of the objects are all triangulated (*e.g.* STL format). Then, for each scanned *Data* point, a matching *Model* point

[3] *Outliers* are inconsistent pairs of *a priori* matched *Data* and *Model* points.

is calculated as the closest of the orthogonal projections of the *Data* point on the objects' triangulated facets. This implies that, contrary to the matching strategy used in [28], points that have no orthogonal projection on any of the objects' facets are rejected. This corresponds to rejecting points at the borders of objects[4]. This new matching strategy also differs from the one used in [11] which calculates matching *Model* points as the projections (similarly to ray casting) of the *Data* points onto the model. The proposed algorithm for calculating matching *Model* points is further detailed in Section 2.2.3 below.

- *Error metric*: The Mean Square Error (MSE) of the Euclidean distance between pairs of matched points is used as error metric. Additionally, for ensuring the robustness of the metric with respect to outliers, point pairs are rejected when:
  (1) *The Euclidean distance between two matched points is larger than a threshold* τ_D. τ_D is adjusted at each iteration k with the formula:

  $$\tau_{Dk} = \max\left\{2\sqrt{MSE_{k-1}}\ ;\ \varepsilon_{Const}\right\}$$

  where MSE_{k-1} is the MSE obtained at the $(k-1)^{\text{th}}$ iteration, and ε_{Const} is a constant distance that can be interpreted as the maximum distance at which objects with dimensional deviation should be searched for. In the experiments presented later, $\varepsilon_{Const} = 50$ *mm*. This value is chosen to be (1) large enough not to fail to recognize objects due to sensor inaccuracies (bias); (2) large enough not to fail to recognize objects that are built at a position up to 50 *mm* away from their expected position; but (3) small enough not to mismatch *Data* and *Model* points corresponding to different objects.
  (2) *The angle between the normal vectors to two matched points is larger than a threshold* τ_A. In the results presented later, $\tau_A = 45°$, but a smaller value could be preferred. As noted by Koivunen et al. [22], the proof of convergence given by Besl and McKay [7] is not valid in the case where points are rejected for criteria that are unrelated to the error metric. Nonetheless, convergence of the algorithm remains possible if the initial registration estimate is close enough to the global minimum that we aim to reach. In the experiments conducted so far, initial coarse registrations have always been sufficiently accurate so that no divergence has been observed.
- *Termination criterion*: the iterative process is stopped when the MSE improvement between the current and previous iterations is smaller than 2 mm^2. All the experiments conducted until now have shown that this is a very conservative value.

2.2.3. Point Matching Algorithm

Three main matching strategies have been proposed in the past in ICP algorithms: *point-to-point* [7], *point-to-plane* [12] and *point-to-projection* [9]. The first two, and in particular the *point-to-plane* algorithm, generally result in more accurate registrations [38, 32]. The third algorithm, however, enables faster calculations (at each iteration) because it can be significantly accelerated using techniques used in 3D scene rendering. Of the three, the *point-to-plane* approach typically converges in less iterations and the *point-to-projection* converges in the largest number of iterations.

For acceleration, kd-trees [41, 28] and/or closest-point caching [38] are commonly used. In [32], Park and Subbarao proposed another interesting acceleration technique that combines the matching speed of *point-to-projection* algorithms to the accuracy and convergence speed of *point-to-plane* algorithms.

With a similar goal as the one of Park and Subbarao [32], it is proposed here to accelerate a *point-to-point* matching algorithm by judiciously combining point rejection and acceleration techniques normally used in image rendering.

The proposed *point-to-point* matching algorithm consists in calculating for each scanned *Data* point, P_D, a matching *Model* point, P_M, that is the closest of the orthogonal projections of P_D on the CAD model's triangular facets. This problem resembles somewhat the ray casting problem faced in computer graphics for rendering 3D scenes. Ray casting can be accelerated by implementing facet culling techniques that quickly narrow down the set of facets among which the closest projection lies. These techniques include: frustum culling, back-facing/visibility culling, as well as using special model data structures such as BSP trees, kd trees, octrees or bounding volume hierarchies (BVH). Due to its high potential for parallelization, rendering by ray casting can be further accelerated by implementation on Graphics Processing Units (GPUs). The difference be-

[4] This rejection criterion is different but related to the rejection of "border points" suggested by Turk and Levoy [46] in the case of the registration of two meshes.

tween our problem and the ray casting rendering problem is that in our problem the direction of projection is unknown (it is not necessarily along the ray coming from the scan's origin). This means that none of the data structures above seems applicable. Only back-facing/visibility culling and frustum culling remain possible.

Distance-based outlier rejection is commonly implemented in ICP algorithms, and is implemented here with the threshold τ_D. Therefore, as illustrated in Figure 3, a *frustum* (also referred to as a *quad-cone*) can be constructed for each *Data* point, centered on the point's scanning direction (ray), and with opening spherical angles equal to:

$$\alpha_\varphi = \alpha_\theta = 2 \arctan\left(\tau_D / P_D.\rho\right)$$

where $P_D.\rho$ is the range of the given data point P_D. This point's frustum has the following important characteristic: if the distance between the point and its orthogonal projection on a facet is lower than τ_D, then the facet must intersect the point's frustum.

We make the following three observations: (1) a facet intersects a point's frustum if its own frustum intersects the point's frustum; (2) the frustum of a set of facets contains the frustums of all these facets; and (3) as illustrated in Figure 4, the facets of a construction project 3D CAD models are naturally grouped into at least three hierarchical groups – single facet, object and model.

As a result of these observations, we propose the following method for accelerating the proposed *point-to-point* matching algorithm. First, a *Bounding Volume Hierarchy (BVH)* is calculated for the project 3D CAD model, in which each bounding volume is the frustum of a facet hierarchical group, as identified in Figure 4. Then, back-facing culling and frustum culling are performed to remove all the facets from the BVH on which no matching point can possibly be found. Finally, for each scanned *Data* point, its frustum is calculated as described above. The facets on which the matching *Model* point may be found (*i.e.* on which the orthogonal projection should be calculated) are identified by going through the model's BVH. They are the facets whose frustums intesect the *Data* point's frustum. The complete point matching algorithm is summarized in Algorithm 2.

The BVH, back-facing culling and frustum culling depend on the location of the scanner, *i.e.* on the registration, and thus must be recalculated at each iteration of the fine registration algorithm. But, despite these necessary recalculations, they enable a significant acceleration of each iteration of the algorithm.

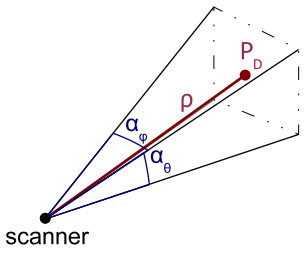


Figure 3: The frustum $\left(\alpha_\varphi, \alpha_\theta\right)$ of each scanned *Data* point P_D is calculated based on the point's range ρ and the distance threshold τ_D that is used for outlier detection and removal in the ICP fine registration algorithm.

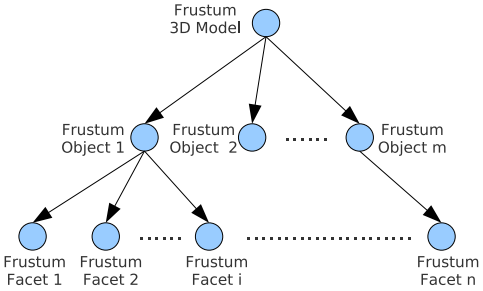


Figure 4: In a construction project 3D CAD model, the facets constituting the triangulated model are naturally grouped into (at least) three hierachical groups: single facet, object, and model.

2.3. Object Recognition

At the end of the registration process above, the project 3D model and the investigated scan are optimally registered. Further, it is known from which CAD model object the *Model* points which were matched at the last iteration come from. As a result, each CAD object can be assigned *temporary*[5] *as-designed* (*Model*) and corresponding *as-built* (*Data*) point clouds. The analysis of the as-built point cloud can then lead to the recognition of the object itself using the recognition metric defined in [11]: an object is recognized when the surface covered by its recognized as-built point cloud is larger than a threshold $Surf_{min}$. $Surf_{min}$ is calculated quasi automatically. Only a minimum number of recognized as-built points n must be manually defined. We refer the reader to [11] for details on: (1) the automated calculation of the surface *covered* by the recognized as-built point cloud; and (2) the calculation of the threshold $Surf_{min}$.

[5]It is explained in Section 3.1 why these point clouds are considered temporary.

```
Input: Scan, Model
Result: {P_M}

Model_BVH ← CalculateModelBVH(Model);
Model_BVH ← FrustumCulling(Model_BVH, Scan.Frustum);
Model_BVH ← BackFacingCulling(Model_BVH, Scan.Origin);
for each Scan.P_D do
    Dist ← ∞;
    for each Model_BVH.Object do
        if Intersect(P_D.Frustum,
        Model_BVH.Object.Frustum) = True then
            for each Model_BVH.Object.Facet do
                if Intersect(P_D.Frustum,
                Model_BVH.Object.Facet.Frustum) = True
                then
                    P'_M ←
                    Project(Model_BVH.Object.Facet,
                    P_D);
                    if Exist(P'_M) = True and ‖ P'_M - P_D ‖
                    < Dist then
                        P_M ← P'_M;
                        Dist ← ‖ P_M - P_D ‖;
                    end
                end
            end
        end
    end
end
```

Algorithm 2: Algorithm calculating a matching *Model* point P_M to each scanned *Data* point P_D. P_M is calculated as the closest of the orthogonal projections of P_D on the facets constituting the matched 3D CAD model.

2.4. Experimental Results

Experiments have been conducted to compare the performances of the proposed object recognition approach (*New*) and the one originally proposed in [11] (*Old*). Since the actual object recognition metric (Section 2.3) is the same in both approaches, these experiments demonstrate more precisely the impact of: (1) the additional model fine registration step; and (2) the new point matching metric used by the fine registration and object recognition algorithms.

The same data set as in [11] has been used. It consists of:

- *Five laser scans* acquired at different stages of the construction of the steel structure of one of the buildings of the Portland Energy Center (PEC) powerplant project in Toronto, Canada. Table 1 summarizes the main characteristics of the five scans. One of the scans, Scan 4, can be seen in Figure 5(a).
- *The 3D CAD model* of the building's steel structure containing 612 objects with a total of 19,478 facets. The objects have various sizes, ranging from large beams to small tie bars. The CAD model of the building's structure can be seen in Figure 5(b).

Table 1: Characteristics (number of scanned points and resolution) of the five scans used in the experiments.

Scan ID	N° of points	Resolution (μrad) Hor.	Vert.
1	691,906	582	582
2	723,523	582	582
3	810,399	582	582
4	650,941	582	582
5	134,263	300	300

2.4.1. Registration Accuracy

Table 2 shows the MSE and the number of matched points obtained with both approaches at the end of the registration process. As expected, the new approach with model fine registration systematically achieves lower final MSEs. The increase in the number of matched point pairs under the new approach offers futher evidence of the improvement in the alignment of the scans with the CAD model. Therefore, the proposed algorithm for fine registration succeeds in significantly improving the alignment of the coarsely registered scans with the CAD model, despite the presence of many outliers.

Table 2: Mean Square Error (MSE) and number of matched point pairs (N° of matches) after registration with the newly proposed (*New*) and originally proposed (*Old*) approaches.

Scan ID	*New*		*Old*	
	MSE (mm^2)	N° of matches	MSE (mm^2)	N° of matches
1	183	359,200	637	315,659
2	195	220,547	894	183,893
3	92	332,862	2109	219,970
4	111	240,327	321	218,147
5	57	80,909	143	80,170

2.4.2. Registration Efficiency

The registration improvement reported above is achieved at the computational cost of the additional model fine registration, which could be prohibitive. For instance, in the case of Scan 4, the fine registration takes about 2 minutes per iteration with a CPU implementation, with a total of 5 iterations (between 5 and 10 iterations for the other scans). The processing time for each iteration is quite fast considering that all the scanned points are used at each iteration ($\sim 650,000$ points) and

that the model contains about 20,000 facets (612 objects), indicating that the proposed point matching algorithm is well accelerated. Without the implementation of any acceleration technique, the matching point algorithm would take about two orders of magnitude more time. Furthermore, the total time of about 10 minutes for Scan 4 is fast, considering: (1) the volume of useful information that can be automatically obtained from it — it enables as-built object recognition and consequently progress control [11] and dimensional compliance control (Section 3); and (2) the time that it would take to obtain this information using traditional manual approaches (*e.g.* see discussion in Section 3).

Several techniques exist that could be implemented to further accelerate the proposed fine registration algorithm without compromising accuracy, in particular:

- *Point sampling*: A multi-resolution sampling strategy could be implemented that would use only a small percentage (*e.g.* 10%) of the scanned *Data* points in the first iteration, and increase this percentage as the registration improves.
- *GPU implementation*: The proposed algorithm has a high potential for parallelization and could be implemented on the GPU for even faster point matching calculations.

2.4.3. Object Recognition

Another important way of investigating the performance of the new approach is to compare the object recognition results it achieves with those achieved by the algorithm proposed in [11]. Table 3 details the object recognition performance statistics, *recall* and *precision* rates[6], obtained when the registration is performed using both approaches. The object recognition metric is the same for both approaches and its threshold $Surf_{min}$ is automatically calculated as $0.01\ m^2 \simeq 10\ cm \times 10\ cm$ (for $n = 5$). As expected, the new registration algorithm further improves the object recognition results: in particular, recall rates are improved for all the scans. The improvements shown here are not significant mainly because the initial coarse registrations happened to be quite accurate. If that had not been the case, the results obtained with the *Old* algorithm would have been poor while the *New* algorithm would still have been able to recover a proper registration and thus achieve good recognition results. Figure 5 further illustrates the accuracy of the object recognition algorithm and also its robustness with respect to occlusions.

[6]The *recall* rate (or *true positive* or *sensitivity* rate) is the number of properly recognized objects divided by the total number of search objects in the investigated scan. The *precision* rate is the number of properly recognized model objects divided by the total number of recognized model objects. Note that, for the calculation of these statistics, the set of objects present in each scan has been identified by manual visual inspection of the scan.

Table 3: Object recognition performance (recall $R_{\%}$ and precision $P_{\%}$) of the *New* and *Old* approaches for the 5 scans acquired of the PEC building.

Scan	*New*		*Old*	
	$R_{\%}$	$P_{\%}$	$R_{\%}$	$P_{\%}$
1	83%	93%	83%	90%
2	77%	93%	70%	92%
3	85%	93%	83%	92%
4	87%	93%	82%	94%
5	84%	82%	82%	84%
all	83%	93%	80%	91%

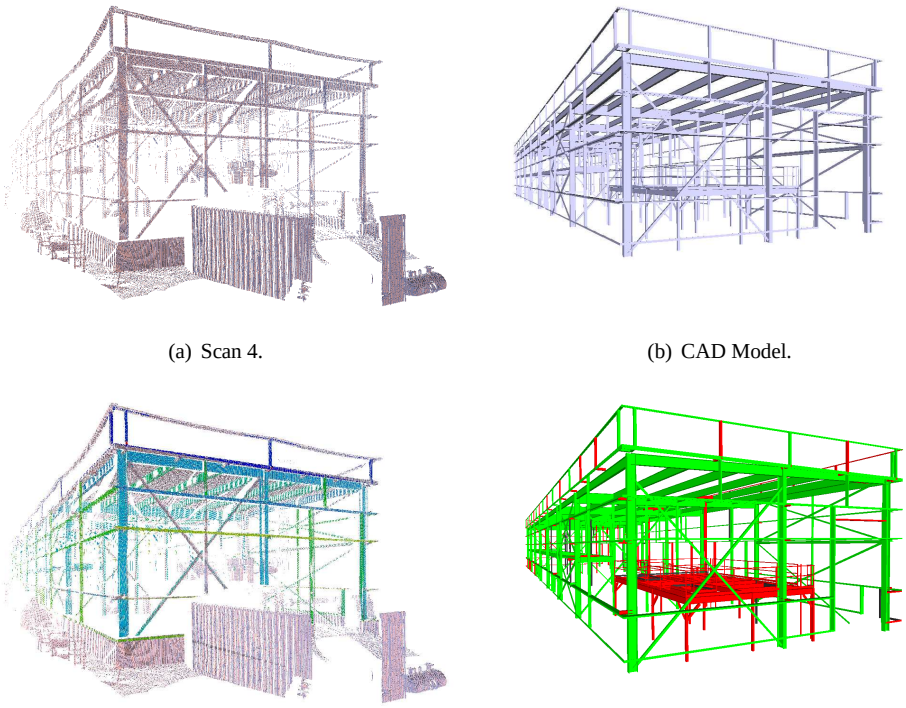
3. Calculation of Objects' As-built Dimensions and Dimensional Compliance Control

In this section, it is shown how to calculate the as-built dimensions of objects recognized in a scan using the above approach. These as-built dimensions can then be used to control the compliance of the project with respect to dimensional tolerances.

3.1. Calculation of Objects' As-built Dimensions

As-built dimensions refer to both the *pose* and the *shape* of the object. In this paper, we assume that *"each object's as-built shape dimensions already comply with the specified tolerances"*. This assumption, although not generally acceptable, is reasonable for prefabricated elements, such as steel or precast concrete elements, for which shape dimensions should comply with tolerances prior to erection [6, 27, 1]. Future research will consider the more general case for which as-built shape dimensions cannot be assumed compliant with specifications (*e.g.* cast-in-place concrete elements). A similar, although more complex, algorithm to the one presented below will then be investigated. In particular, approaches for parametric shape matching, initially investigated by Lowe [24] and later Reid and Brady [36], will be researched. The recent work by Rabani and van den Heuvel [35] on matching *Constructive Solid Geometric (CSG)* shapes to point clouds is also related to this problem.

Let's consider a single object recognized in a scan. After the registration process presented in Section 2.2, the CAD model of the object is "aligned" with a recognized as-built point cloud, and it could be concluded that its pose at that point corresponds to its as-built pose.



(a) Scan 4. (b) CAD Model.

(c) Object Recognition (in point cloud). (d) Object Recognition (in CAD model).

Figure 5: Performance of the proposed approach for automated recognition of 3D CAD model objects in large construction site laser scans: (*a*) Scan 4; (*b*) the 3D CAD model after registration with Scan 4; (*c*) object recognition results displayed in Scan 4. Each point cloud recognized as corresponding to a CAD model object is displayed with a unique color (some colors may appear similar but are in fact different). Points in gray (same color as in (a)) are those that have not been matched to any of the CAD objects; (*d*) object recognition results displayed in the CAD model, where objects colored in green are those recognized in Scan 4. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

However, this alignment is achieved globally, considering all CAD model objects combined into a single rigid model, so that the poses of the objects at the end of that registration process must in fact be considered in their *as-designed poses* (with respect to the given scan). In order to obtain their as-built poses, these must be searched for somewhat independently from one another. It is thus proposed to re-apply the ICP registration algorithm presented in Section 2.2.2 with, as input, the CAD objects in their as-designed poses. The difference here is that, at each iteration, the function $R()$, which calculates the least square optimal registration between the matching *Data* and *Model* points, is run individually for each object. This results in a refinement of the registration of the model of each object independently of the models of the other objects. For this reason, this second fine registration step is referred to as *object fine registration*. At its convergence, the CAD model of each object can be considered to be in its *as-built pose*.

3.2. Dimensional Compliance Control

Once the *as-built poses* of all recognized objects are calculated, they can be compared to their *as-designed poses* in order to infer some information on the compliance of the project with respect to dimensional tolerances. More exactly, the differences between many as-built and as-designed dimensions (within and between objects) can be calculated and compared to their corresponding tolerances defined in the project specifications, which may be specific to the project or refer to industry standards such as AISC 303-05 [1] and MNL 135-00 [27]. For example, the verticality of a column can be compared to its designed true verticality and the difference compared to the specified verticality deviation tolerance. Similarly, the distance between two adjacent columns can be easily calculated from the as-built poses of both columns and then compared to the designed distance.

3.3. Experimental Results

The experimental data used earlier is of a building's steel structure for which all the elements are prefabricated. This data can be used to assess the performance of the proposed approach under the stated assumption since all structural elements comply with shape tolerances prior to erection.

3.3.1. Accuracy

Table 4 shows the MSEs obtained after the *object fine registration* process is applied to all 5 scans. THese MSEs are lower that the corresponding MSEs obtained after the *model fine registration* process, as shown in Table 2. Furthermore, this is achieved while increasing the number of matched points. These results suggest that the object fine registration successfully improves the alignment of the model of each object with its scanned point cloud.

Table 4: Mean Square Error (MSE) and number of matched point pairs (N° of matches) after the *object fine registration* process. The results are to be compared with those shown in Table 2.

Scan ID	MSE (mm^2)	N° of matches
1	35	374,251
2	37	232,475
3	33	341,231
4	29	243,045
5	13	81,582

However, these improved alignment statistics do not necessarily demonstrate that the calculated as-built poses of the CAD objects correspond to their actual as-built poses. Indeed, compared to the problem faced for the *model fine registration*, the one faced for the *object fine registration* is very often ill-conditioned. For example, in a site scan, the top and bottom parts of a column are often both occluded, so that an optimal registration may only be achieved up to some translation along the column's main axis. Nonetheless, as shown in Figure 6 (and also Tables 5 and 6 described later) for Scan 4, no unjustified divergence is observed. This may be explained by the fact that the model coarse and fine registration steps result in good starting poses for the object fine registration step so that the algorithm tends to converge to the true as-built poses despite the possible pose ambiguities.

Tables 5 and 6 present additional results with respect to the as-built poses calculated for the 16 exterior columns of the building's structure (Figure 3.3.1). Table 5 lists the differences (distances in *mm*) between the as-designed and as-built poses of both the bottom and top center points of the 16 columns (Figure 3.3.1), as well as the plumb deviations. Table 6 lists the calculated distances between the bottom, respectively top, center points of *structurally connected* columns, *i.e.* columns that are directly connected by beams. Unfortunately, for all the values reported in these two tables, the ground truth, or at least the set of values that would have been measured manually on site, is not available. As a result, we cannot demonstrate here whether the calculated poses and distances are true. Nonetheless, the results are promising for the following reasons:

- The values in the columns ΔZ in Table 5 seem to confirm the observation in Figure 6 that the object fine registration algorithm does not diverge.
- All the pose deviations reported in Tables 5 and 6 are realistic in value.

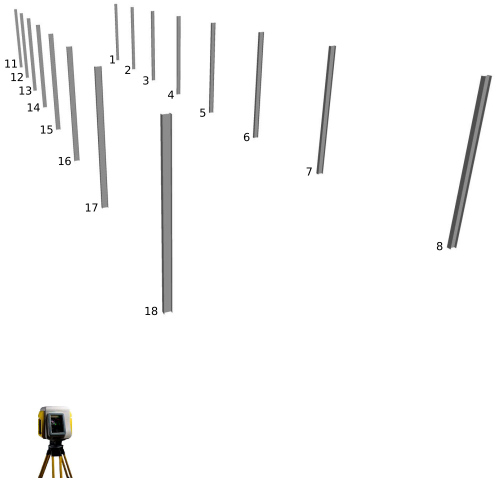
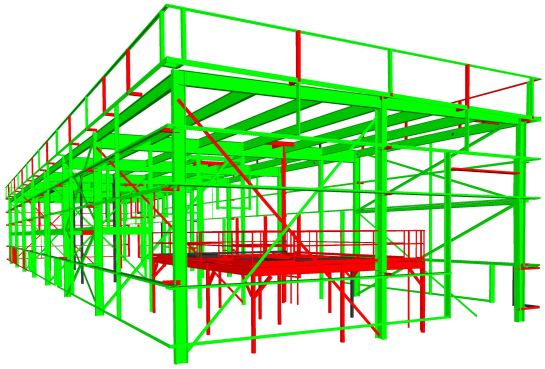


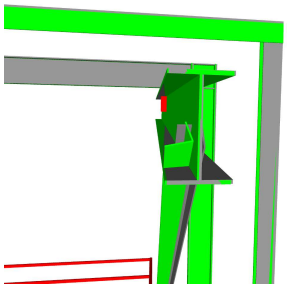
Figure 7: The 16 exterior columns of the structure of the PEC building and the location of the scanner for Scan 4.

Figure 8: The *top center point* and *bottom center point* of a column typically used as control point for dimensional control.

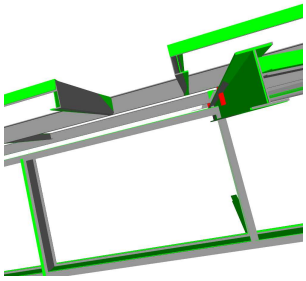
However, the results in Table 5 seem to indicate that



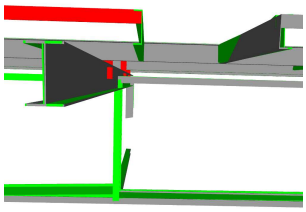
(a) Scan 4.



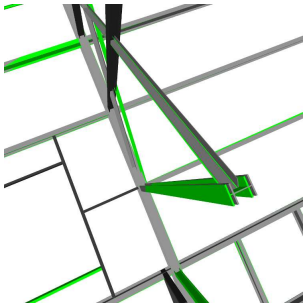
(b) Detail 1.



(c) Detail 2.



(d) Detail 3.



(e) Detail 4.

Figure 6: Building's as-built model after the *object fine registration* process: (a) each CAD object in its as-built pose. Detected objects (green) are the only ones to which this fine registration was applied; (b), (c), (d) and (e) are close-ups where the as-designed models (gray) are added to show the difference between the as-designed and as-built models of each recognized object. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

the calculated as-built pose deviations increase with the distance of the column to the scanner (range). In fact, a correlation coefficient of 0.45 is found between the calculated point pose deviations and the distances of the corresponding objects to the scanner. This correlation may have two causes:

1. *The typical decrease in accuracy of laser scanners with scanning distance*. Past a certain (yet unknown) distance, the pose calculation error resulting from the inaccuracy of the sensor may be much larger than the actual construction error, making any extracted dimension unreliable. This could be the case here, as the distances of the 16 columns to the scanner range from 20 *m* to 80 *m*, which is quite far.
2. *The number and coverage of recognized points*. The further an object is from the scanner, the lower the density of points and therefore the less points may be recognized from it. Occlusions further reduce this number. The result is potential error in the calculation of object as-built poses. Nonetheless, we observed that the least number of recognized points for any of the 16 columns is 438 (obtained for column 11), which is fairly high and thus reduces the likelihood that the number of recognized points is a significant source of error here.

In summary, the current results are not sufficiently reliable to drwa any meaningful conclusions as to the accuracy of the proposed approach for dimensional compliance control. Further experiments using a more accurate laser scanner and shorter scanning ranges (up to 40-50 meters, which still corresponds to a very large volume of control) should be conducted to better assess this accuracy, and results should also be compared to some ground truth.

3.3.2. Efficiency

Assuming that the proposed approach for as-built dimensions calculation and compliance control could provide accurate information, this system is very efficient

Table 5: Results of the dimensional compliance control performed for the 16 exterior columns of the PEC building's steel structure and with Scan 4 only. Are reported for each of the columns: the distance of the object to the scanner (estimated as the distance to its bottom center point); the difference between the as-built and as-designed locations (ΔXYZ) and altitudes (ΔZ) for the bottom and top center points; and the difference between the as-built and as-designed plumbs.

Obj. ID	Estim. Range (m)	Bottom Point ΔXYZ (mm)	Bottom Point ΔZ (mm)	Top Point ΔXYZ (mm)	Top Point ΔZ (mm)	$\Delta plumb$ (%)
1	82.5	10.2	0.1	19.0	0.0	0.33
2	74.6	12.7	0.0	17.8	-0.0	0.38
3	66.7	11.3	0.0	16.9	0.0	0.34
4	58.9	8.2	0.0	12.0	0.0	0.26
5	51.2	23.3	0.0	5.2	-0.0	0.23
6	43.7	16.7	0.0	16.0	0.0	0.11
7	36.5	16.6	0.0	11.3	-0.0	0.32
8	28.3	7.7	0.0	2.2	-0.0	0.10
18	21.9	1.8	0.0	1.0	0.0	0.00
17	31.8	4.9	0.0	7.0	0.0	0.11
16	39.9	4.2	0.1	16.5	0.1	0.21
15	48.0	11.1	0.0	6.0	0.0	0.22
14	56.1	6.0	0.0	4.8	-0.0	0.09
13	64.3	11.3	0.0	13.4	0.0	0.31
12	72.4	6.8	0.0	16.4	0.0	0.30
11	80.5	5.4	0.0	19.8	0.0	0.31

Table 6: Results of the dimensional compliance control performed for the 16 exterior columns of the PEC building's steel structure and with Scan 4 only. Are reported for each pair of *structurally connected* columns: the difference between the as-built and as-designed distances (ΔXYZ) between their bottom and top center points.

Connected Components		Bottom Point ΔXYZ	Top Point ΔXYZ
Obj. ID	Obj. ID	(mm)	(mm)
2	1	-5.2	-1.3
3	2	1.2	-0.9
4	3	3.0	-4.9
5	4	30.9	-6.7
6	5	-25.2	2.6
7	6	-7.7	-8.4
8	7	3.6	1.5
17	18	3.3	-3.2
16	17	-0.8	-0.8
15	16	6.6	-2.2
14	15	-4.8	2.4
13	14	5.32	-9.81
12	13	-4.5	-3.0
11	12	-1.3	-3.4
1	11	-7.2	0.3
2	12	-3.6	1.4
3	13	-2.9	-1.4
4	14	-1.8	1.7
5	15	-1.9	-0.6
6	16	-16.2	2.1
7	17	-13.2	4.9
8	18	3.3	-1.1

compared to the time it would take to obtain similar information with traditional manual approaches. For example, it took about 35 minutes to process Scan 4, including model fine registration, object fine recognition and as-built pose calculation (out of which 30 minutes are spent on the as-built pose calculation), and it resulted in numerous calculated as-built poses and dimensions. If the scan setup, data acquisition and complete processing are combined, results could be achieved within a time frame of one to two hours. We estimate that, in order to control the same set of project dimensions, an expert would probably need at least a few hours and possibly up to one day[7]. Additionally, as shown in [11], not only would this system enable the automated calculation of as-built dimensions, but it would simultaneously enable automated progress tracking (only the *model coarse registration* has to be performed manually).

We acknowledge that the current system for dimensional compliance control considers CAD models in a mesh format. Unfortunately, such represention cannot include semantical information such as the location of control points. As a result, the as-built poses of these control points had to calculated manually based on the poses of the mesh vertices. This would not be acceptable for actual field implementation. Future work should include the replacement of the simple triangulated 3D model of the project with a *Building Information Model (BIM)* — a semantically enriched 3D CAD model — which will integrate all this information (including dimensional tolerances) so that this entire process could be implemented fully automatically.

4. Conclusion

This article presents an approach for the automated tracking of the as-built 3D status of construction sites using laser scanning and project 3D CAD modeling. The first contribution is an improved algorithm for automated recognition of project 3D CAD model objects in large site laser scans. The improvements include the development of a novel ICP algorithm for the fine registration of site laser scans with project 3D CAD models, and a more robust point matching strategy. The resulting point cloud recognition approach is (1) quasi fully automated (only the model coarse registration needs to be performed manually); (2) accurate; (3) robust to clutter and occlusions; and (4) very efficient, particularly

[7] This estimation is not based on any field measurements, but we believe that it is reasonable.

when considering the project control applications it enables, such as automated progress tracking and dimensional compliance control.

The second contribution is in fact an algorithm for calculating the as-built poses of the objects recognized by the previous algorithm, and using these poses for automated dimensional compliance control. This algorithm for as-built pose calculation uses the same ICP algorithm as above, but optimizes the pose of the models of each object independently from the poses of the models of all the other objects. Unfortunately, the experimental results are not conclusive. The results are promising but (1) they could not be verified against any ground truth; and (2) the impact of the accuracy of the laser scanner and the number and spread of recognized points on the calculated as-built poses is not yet clear.

Nonetheless, we note that, to our knowledge, this is the only reported remote sensing-based system with convincing performances for automated project 3D status tracking which is able to simultaneously support progress tracking and dimensional compliance control. In other words, despite their high initial costs, laser scanners could be efficiently and effectively used for automatically and comprehensively tracking the 3D status of construction projects, and thus for providing significant benefits to automated project control. By comparison, digital cameras are far cheaper than laser scanners, but their benefit to automated project control is yet to be demonstrated.

5. Future Work

Despite the promising results obtained with the proposed approach for as-built pose calculation and dimensional compliance control, several limitations remain and further research is required:

- The accuracy of the proposed approach for automated as-built dimension calculation is not fully verified yet. Experiments need to be conducted to: (1) compare the calculated as-built dimensions with some ground truth; and (2) thoroughly assess the impact of the accuracy of the laser scanner and the number and spread of recognized points on the calculated as-built poses.
- A method for calculating the *as-built shape* must be developed so that the system enables more comprehensive dimensional quality control.
- In the currently proposed approach, the poses of the different objects are calculated completely independently from one another, which may result in some clashes. A future algorithm should thus consider some global constraints preventing such clashes. This will require nonrigid shape parameterizations.
- The as-built pose of control points and resulting as-built dimensions is currently performed manually, and the compliance of these dimensions with specifications is also currently performed manually. Future work will thus consider using *Building Information Models (BIMs)* that integrate semantic information such as dimensional specifications to project 3D data.

Acknowledgements

We would like to thank the Competence Center for Digital Design and Modeling (DDM) for its financial support. We also thank SNC Lavalin, in particular Paul Murray, for (1) giving access to the PEC construction site, and (2), in addition to Prof. Carl T. Haas, for allowing us to publish the experimental results obtained with the data obtained on that site.

References

[1] AISC 303-05. *Code of Standard Practice for Steel Buildings and Bridges*. American Institute of Steel Construction, Inc., 2005.

[2] B. Akinci, S. Kiziltas, E. Ergen, I. Z. Karaesmen, and F. Keceli. Modeling and analyzing the impact of technology on data capture and transfer processes at construction sites: A case study. *Journal of Construction Engineering and Management*, 132(11):1148–1157, 2006.

[3] Y. Arayici. An approach for real world data modelling with the 3D terrestrial laser scanner for built environment. *Automation in Construction*, 16(6):816–829, 2007.

[4] F. Arman and J. K. Aggarwal. Model-based object recognition in dense-range images: a review. *Computing Surveys (CSUR)*, 25(1):5–43, 1993.

[5] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[6] D. K. Ballast. *Handbook of Construction Tolerances (2nd Edition)*. John Wiley & Sons, Inc., 2007.

[7] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[8] P. Biddiscombe. 3D laser scan tunnel inspections keep expressway infrastructure project on schedule. *White Paper - Trimble*, 2005.

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

[10] F. Bosché and C. T. Haas. Automated retrieval of project three-dimensional CAD objects in range point clouds to support automated dimensional QA-QC. *Information Technologies in Construction*, 13:71–85, 2008.

[11] F. Bosché, C. T. Haas, and B. Akinci. Performance of a new approach for automated 3D project performance tracking. *Journal of Computing in Civil Engineering, Special Issue on 3D Visualization*, To appear.

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

[13] G. S. Cheok and W. C. Stone. Non-intrusive scanning technology for construction management. In *Proceedings of the 16th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 645–650, Universidad Carlos III de Madrid, Madrid, Spain, September 22-24 1999.

[14] M. Goldparvar-Fard, F. Peña-Mora, and S. Savarese. D^{4}AR – a 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. *Journal of Information Technology in Construction*, 14:129–153, 2009.

[15] C. Gordon and B. Akinci. Technology and process assessment of using LADAR and embedded sensing for construction quality control. In *Proceeding of the ASCE Construction Research Congress (CRC)*, pages 557–561, Honolulu, Hawai, USA, April 5-7 2005.

[16] C. Gordon, F. Boukamp, D. Huber, E. Latimer, K. Park, and B. Akinci. Combining reality capture technologies for construction defect detection: a case study. In *Proceeding of the 9th EuropIA International Conference(EIA9): E-Activities and Intelligent Support in Design and the Built Environment*, pages 99–108, Istanbul, Turkey, October 8-10 2003.

[17] S. J. Gordon, D. D. Lichti, M. P. Stewart, and J. Franke. Modelling point clouds for precise structural deformation measurement. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXV-B5/2, 2004.

[18] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629–642, 1987.

[19] Y. M. Ibrahim, T. C. Lukins, X. Zhang, E. Trucco, and A. Kaka. Towards automated progress assessment of workpackage components in construction projects using computer vision. *Advanced Engineering Informatics*, 23:93–103, 2009.

[20] G. Jacobs. Versatility - the other "hidden instrument" inside laser scanners. *Professional Surveyor Magazine*, 24(10), 2004.

[21] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[22] V. Koivunen and J.-M. Vezien. Machine vision tools for CAGD. *International Journal of Pattern Recognition and Artificial Intelligence*, 10(2):165–182, 1996.

[23] Leica Geosystems AG. *CloudWorx 3.3 for AutoCAD*.

[24] D. G. Lowe. Fitting parameterized three-dimensional models to images. *Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991. 0162-8828.

[25] T. C. Lukins and E. Trucco. Towards automated visual assessment of progress in construction projects. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 142–151, Coventry, UK, 2007.

[26] Z. A. Memon, M. Z. Abd.Majid, and M. Mustaffar. An automatic project progress monitoring model by integrating AutoCAD and digital photos. In *Proceedings of the ASCE International Conference on Computing in Civil Engineering*, Cancun, Mexico, July 12-15 2005.

[27] MNL 135-00. *Tolerance Manual for Precast and Prestressed Concrete Construction*. Precast/Prestressed Concrete Institute, 2000.

[28] V. Moron, P. Boulanger, T. Masuda, and T. Redarce. Automatic inspection of industrial parts using 3-D optical range sensor. In *Proceedings of SPIE*, volume 2598 - Videometrics IV, pages 315–326, 1995.

[29] R. Navon. Research in automated measurement of project performance indicators. *Automation in Construction*, 16(2):176–188, 2007.

[30] C. Ordóñez, P. Arias, J. Herráez, J. Rodríguez, and M. T. Martín. Two photogrammetric methods for measuring flat elements in buildings under construction. *Automation in Construction*, 17:517–525, 2008.

[31] H. S. Park, H. M. Lee, H. Adeli, and I. Lee. A new approach for health monitoring of structures: terrestrial laser scanning. *Computer-Aided Civil and Infrastructure Engineering*, 22(1):19–30, 2007.

[32] S.-Y. Park and M. Subbarao. A fast point-to-tangent plane technique for multi-view registration. In *Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling (3DIM)*, Banff, Canada, October 06-10 2003.

[33] F. Prieto, T. Redarce, R. Lepage, and P. Boulanger. Visual system for fast and automated inspection of 3D parts. *International Journal of CAD/CAM and Computer Graphics*, 13:211–217, 1998.

[34] F. Prieto, T. Redarce, R. Lepage, and P. Boulanger. An automated inspection system. *International Journal Advanced Manufacturing Technologies*, 19:917–925, 2002.

[35] T. Rabbani and F. van den Heuvel. 3D industrial reconstruction by fitting CSG models to a combination of images and point clouds. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, volume XXXV-B5, pages 7–12, Istanbul, Turkey, July 12-13 2004.

[36] I. D. Reid and J. M. Brady. Recognition of object classes from range data. *Artificial Intelligence Journal*, 78(1-2):289–326, 1995.

[37] J. Reinhardt, B. Akinci, and J. H. J. Garrett. Navigational models for computer supported project management tasks on construction sites. *Journal of Computing in Civil Engineering*, 18(4):281–290, 2004.

[38] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 145–152, Quebec City, QC, Canada, May 2001.

[39] N.-J. Shih and P.-H. Wang. Using point cloud to inspect the construction quality of wall finish. In *Proceedings of the 22nd eCAADe Conference*, pages 573–578, Copenhagen, Denmark, September 2004.

[40] H. Shin and P. S. Dunston. Evaluation of Augmented Reality in steel column inspection. *Automation in Construction*, 18:118–129, 2009.

[41] D. A. Simon, M. Hebert, and T. Kanade. Real-time 3-D pose estimation using a high-speed range sensor. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2235–2241, San Diego, CA, USA, May 8-13 1994.

[42] L. Song. Project progress measurement using CAD-based vision system. In *Proceeding of the ASCE Construction Research Congress (CRC) "A Global Community"*, Grand Bahama Island, The Bahamas, May 6-8 2007.

[43] J.-P. Tarel and N. Boujemaa. A coarse to fine 3D registration method based on robust fuzzy clustering. *Computer Vision and Image Understanding*, 73(1):14–28, 1999.

[44] Trimble. *GX 3D Specifications*.

[45] Trimble. *RealWorks Survey 6.1 - Technical Notes*.

[46] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *21st International Conference on Computer Graphics and Interactive Techniques*, pages 311–318, Orlando, FL, USA, 1994.

[47] Vexcel Corporation. Non-contact 3D measurement for construction verification and component inspections. April 2003.