

Block Assembly for Global Registration of Building Scans

Feilong Yan, Liangliang Nan, Peter Wonka
King Abdullah University of Science and Technology (KAUST), KSA

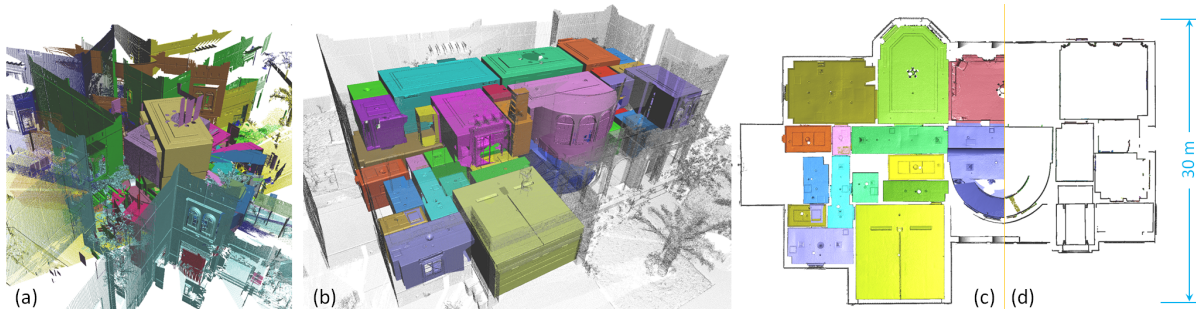


Figure 1: Our method can register individual scans of isolated rooms (e.g., bedrooms, bathrooms, etc.) of a building along with scans of the building's exterior into a single point cloud. (a) The input scans rendered with different colors. (b) Our registration result. The rooms are rendered with distinct colors and the exterior of the building is down-sampled and rendered in a uniform gray color. (c) A top view of the registered rooms in the first floor. (d) A cross section of the registered data in (b) at 2.0m from the ground.

Abstract

We propose a framework for global registration of building scans. The first contribution of our work is to detect and use portals (e.g., doors and windows) to improve the local registration between two scans. Our second contribution is an optimization based on a linear integer programming formulation. We abstract each scan as a block and model the blocks registration as an optimization problem that aims at maximizing the overall matching score of the entire scene. We propose an efficient solution to this optimization problem by iteratively detecting and adding local constraints. We demonstrate the effectiveness of the proposed method on buildings of various styles and that our approach is superior to the current state of the art.

Keywords: registration, reconstruction, assembly, building, indoor, laser scanning, point cloud

Concepts: •Computing methodologies → Matching; Reconstruction;

1 Introduction

Acquiring 3D buildings is an important prerequisite for applications such as simulation, navigation, virtual reality, and entertainment. In this work, we are interested in scanning architectural structures at a larger scale, e.g., office buildings, sports centers, classrooms, or complete residential buildings, including their exteriors. One possible approach to scanning a building's interior is to use a hand-held scanner [Chen et al. 2013; Zhou and Koltun 2013; Choi

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2016 ACM. SA'16 Technical Papers, December 05-08, 2016, Macao ISBN: 978-1-4503-4514-9/16/12 DOI: <http://dx.doi.org/10.1145/2980179.2980241>

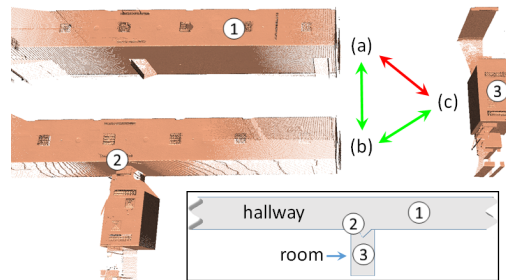


Figure 2: Registration is sensitive to scanner positions. The inset shows the 2D layout of the scene consisting of a long hallway and a room. Scans (a), (b), and (c) are obtained at positions 1, 2, and 3, respectively. Traditional methods can register (a) to (b) and (b) to (c), but they cannot register (a) to (c) due to large flat regions and insufficient features.

et al. 2015; Kahler et al.] that can provide good results for scenes consisting of a small set of rooms. However, since existing hand-held scanners have narrow fields of view, many frames have to be captured to ensure reasonable coverage even for a small scene (e.g., an office). This often leads to severe drift or failure in registration of these frames. In addition, hand-held scanners have restricted range limits. For example, the sensor range of Microsoft Kinect is $[0.6m, 4m]$, which is not appropriate for scanning large indoor scenes or the exteriors of buildings. We thus use a static scanner to scan large buildings and we optionally use hand-held scanners to capture individual rooms. The technical problem we address in this paper is to register a set of individual scans obtained by one or multiple scanners.

Registration of building scans is challenging for several reasons. Firstly, in contrast to hand-held scans in which adjacent frames are spatially close, scans obtained by a static scanner are far apart. Therefore, registration of these scans becomes a global matching problem. Secondly, unlike general objects (e.g., statues, mechanical parts) for which rich geometric features can be extracted, buildings mainly consist of piecewise planar regions, which means that fewer unique features can be extracted for registration.

For example, in Figure 2, scan (a) cannot be registered to (c) because the two scans can slide due to insufficient constraints. Thirdly, sufficient overlaps required for registration [Matterport 2016] cannot always be guaranteed during scanning. It is easy to introduce extra scanning positions to create overlaps for a building’s interior or exterior, but extra scans may not help when registering interior scans to exterior scans of a building (see Figure 4 (a) for an example). Note that introducing extra scans leads to a larger number of scans, but the extra scans usually do not provide additional information describing the scene’s geometry. Lastly, multiple users could be involved in performing uncoordinated scanning in parallel of large scenes, which makes it harder to guarantee sufficient overlap between neighboring scans.

Ideally, one option to simplify the registration problem is to exploit high-precision positioning techniques. However, existing positioning techniques, such as GPS, can only roughly position individual scans of outdoor environments (with positioning errors larger than 3.5m [GPS.gov 2015]). For indoor scanning, accuracy becomes even more problematic due to attenuation and occlusion of satellite signals. Other well-developed indoor positioning techniques, mainly WiFi positioning systems, require a lot of additional work to create the setup. Their accuracy depends on the number of access points, and even the state-of-the-art accuracy is still too low (1–5m) for registration purposes [Dardari et al. 2015]. In engineering scenarios, markers are usually required and are carefully placed in the scene before scanning, and manual work is required to obtain the registration. Our work aims to lower such requirements in the scanning process. We thus assume that scans are given without additional positioning or marker information.

We propose two novel methods to tackle the problem of global registration of building scans. The first method improves local registration, and the second one improves global registration. Firstly, we make use of common building characteristics and observe that buildings usually contain quite a few portals (e.g., doors, windows) between neighboring blocks. These portals are natural connectors that can provide promising hints for connecting isolated scans. This observation inspires us to use portals as additional information for local registration, complementing traditional registration methods based on features in overlapping regions. Experiments on large scenes indicate that portals are valid and effective features for registering scans without overlapping areas and scans that capture both the exteriors and interiors of buildings.

Secondly, we observe that the registration of isolated scans is similar to solving a jigsaw puzzle. This observation inspires us to formulate global registration as a combinatorial optimization problem. Unfortunately, related work on solving similar problems indicates that exploration of the entire search space is actually impractical [Huber 2002; Huang et al. 2006] because the problem is NP hard. For scenes consisting of a few (e.g., 5) isolated scans, it is possible to manually align them with reasonable accuracy, but this strategy becomes very time-consuming if the number of isolated scans becomes larger (e.g., 10). To tackle the global registration problem, we propose a novel formulation based on linear integer programming, which enables us to make use of various strategies available in state-of-the-art solvers to prune the search space. However, this approach is still too slow for large problems. As a solution, we propose an acceleration strategy based on iteratively adding constraints instead of exhaustively enumerating them at the beginning of the optimization.

In the results, we compare our method with four competing algorithms to demonstrate the effectiveness of our approach. Figure 1 shows a registration result of a two-floor residential building consisting of 29 rooms captured by 37 scans.

The main contributions of our work are:

- we offer the *first* global registration framework that can handle scans of isolated rooms and scans that capture both the exteriors and interiors of buildings based on portals extracted from scans.
- we present a novel formulation of global registration based on constrained combinatorial optimization and an efficient solution by iteratively adding constraints.
- we demonstrate reliable registration results for building scans in cases where existing state-of-the-art methods fail.

2 Related Work

In this section, we review recent work on acquisition planning, pairwise registration, global registration, and block assembly.

Acquisition planning. To ensure sufficient coverage for scanning, various acquisition planning methods have been proposed to guide scanning and reconstruction of physical objects [Scott et al. 2003; Wu et al. 2014]. Recent methods [Yan et al. 2014; Xu et al. 2015] propose modification of the environment to access occluded parts of the scene. Acquisition planning aims at strategically positioning the scanner to achieve improved coverage of a physical object. By contrast, we assume multiple scans of a complex building as input, and we focus on effectively registering them into a consistent single scan.

Pairwise registration. Given two scans, pairwise registration computes a rigid or non-rigid transformation between the two input scans. Typical pairwise registration algorithms are the well-known ICP (Iterative Closest Point) [Besl and McKay 1992] and its variants [Rueckert et al. 1999; Rusinkiewicz and Levoy 2001; Li et al. 2008; Segal et al. 2009]. These registration methods rely on correspondence information of two scans (e.g., point-to-point, point-to-surface) and thus they require that two input scans have a large portion of overlap. Gelfand et al. [2005] extract key points and their descriptors from the input scans. They then establish sparse correspondences between the key points based on their descriptors. The registration is obtained using a branch-and-bound algorithm to find the optimal set of correspondences, followed by an ICP refinement step. The 4PCS registration scheme proposed by Aiger et al. [2008] is robust to noise and outliers. However, it requires reasonable overlaps to query 4-point-wide bases and it is computationally inefficient for scans of large buildings. Based on pairwise registration, the KinectFusion technique and its variants [Newcombe et al. 2011; Izadi et al. 2011; Whelan et al. 2012; Roth and Vona 2012; Nießner et al. 2013], and many successive techniques [Chen et al. 2013; Zhou and Koltun 2013; Choi et al. 2015; Kahler et al.] can incrementally fuse depth data into a single surface model. These methods are suitable for capturing relatively small scenes, but usually fail for large scenes where overlap and common features between individual scans cannot be guaranteed (see Section 1). For a more comprehensive survey of both rigid and non-rigid registration methods, please refer to [Tam et al. 2013].

Global registration. Global registration methods, which are usually built on top of pairwise registration algorithms, take as input multiple point clouds and try to minimize the accumulated matching error introduced by the pairwise registration methods. Brown and Rusinkiewicz [2007] generate correspondence hypotheses among multiple scans based on a locally weighted ICP. They then obtain the optimal locations of the feature points by minimizing the change in their relative distances computed in a neighborhood. Finally, thin-plate splines are computed based on

optimized feature correspondences to warp each of the scans. To register multiple non-rigid frames, Huang et al. [2011] focus on the alignment order of the scans to reduce the accumulated drift during incremental registration. They propose a graph structure for global alignment based on frame-to-frame surface tracking for pairwise registration. Temporally consistent registration can then be achieved by minimizing overall non-rigid transformation based on graph optimization. Zhou and Koltun [2013] estimate a rough camera trajectory using RGB-D SLAM (simultaneous localization and mapping) and detect the points of interest from a density map of the scene. To deal with the registration error accumulated along the scanning trajectory, they construct a pose graph with its edges representing the relative pose for each pair of consecutive frames. The poses of all frames are then optimized using a weighted non-linear least squares formulation.

Block assembly. Several researchers have studied the problem of assembling different pieces of an object [Cutler 1978; Huber 2002; Huang et al. 2006; Huang et al. 2012; Fu et al. 2015]. Cutler [1978] proposes computational assembly of interlocking burr puzzles, which was proven to be NP hard in later work [Huber 2002]. In Cutler’s experiments, exhaustive search is employed, but it is too slow to explore the configuration space of structures consisting of only six pieces. By assuming continuous sharp feature curves, Huang et al. [2012] compute an ambient vector field that extrapolates these curved features and this field is used to guide the alignment of two distinct parts. To register fractured object pieces, Huang et al. [2006] first classify the object faces into original and fractured faces by roughness analysis. Then using patch-based surface features, they compute potential matches to perform pairwise matching among all fractured faces. To eliminate errors in the pairwise matching step, a multi-piece registration is performed in a greedy manner, which incrementally merges the pieces based on penetration and consistency checks. This method relies heavily on the rich geometric features of the fractured surfaces, which rarely exist between scans of different building blocks. To register multiple point clouds, Huber and Hebert [2003] encode pairwise matchings using a graph structure and they obtain globally consistent registration by computing a minimum spanning tree of the graph. Using a similar graph structure, Sharp et al. [Sharp et al. 2004] obtain global registration by merging the individually optimized subgraphs of neighboring scans. This method requires that the neighborhood for each scan exists. Their optimization of subgraphs inspires us to explore more of the local registrations. The difference is that, instead of seeking out good local alignments, we integrate the most unlikely local alignments as constraints in an optimization framework, ensuring a globally optimal alignment of all scans.

All the above registration approaches require that reasonable overlaps exist among the input scans. Thus, they are not applicable to indoor scenes when isolated regions are prevalent in the input, nor to scans capturing both the interiors and exteriors of buildings. To register such scans, we rely on *portals* as the connectors and we globally register all scans based on a combinatorial optimization formulation.

3 Overview

Given a set of separately scanned point clouds representing a building, our goal is to register them into a consistent global coordinate frame. Our method consists of the following parts:

Preprocessing. In the preprocessing step, we abstract each scan as a block by detecting the largest connected component of each scan. These blocks are the inputs to our system. However, since our global registration framework is general, we also discuss possible



Figure 3: Three devices used for capturing the input data for our system. From left to right, a Leica ScanStation C10 static laser scanner, a FARO Freestyle X hand-held scanner, and a Microsoft Kinect One depth camera.

inputs of other types (see Section 4.1).

Portal detection. We automatically extract walls and portals (i.e., doors, windows) from the blocks based on heuristics. We optionally allow the user to remove and add portals to improve the automatic results (see Section 4.2).

Graph construction. The portals and blocks are encoded using a graph with its edges representing all potential pairwise alignments. We assign to each edge in the connection graph a weight representing the matching score of the two portals or two blocks connected by this edge (see Section 4.3).

Block assembly. Globally consistent registration is achieved by solving a combinatorial optimization problem. We propose a linear integer programming formulation that maximizes the overall matching score among all the blocks (see Section 4.4).

Acceleration algorithm. For efficiency, we introduce an acceleration algorithm based on the idea of iteratively adding constraints starting with the most unlikely local alignments (i.e., conflicting alignments). Instead of finding a complete set of conflicting alignments, we detect only locally conflicting matches of very few scans. Thus, the global registration turns out to be a constrained linear integer programming problem that can be efficiently solved (see Section 4.5).

4 Methodology

4.1 Input and preprocessing

We experimented with a variety of input data from a static laser scanner (Leica ScanStation C10), a hand-held scanner (FARO Freestyle X), and a consumer-level depth camera (Microsoft Kinect One) (see Figure 3) with effective operating ranges of 100m, 3m, and 1.5m, respectively. We scanned large rooms and the exteriors of the buildings with the static scanner. The remaining interiors were scanned with a combination of the three scanners. We used the Kinect One sensor to scan small rooms with KinectFusion [Izadi et al. 2011].

Due to occlusions, each point cloud from the scanners may contain a large number of disconnected parts. Several relatively small parts are not helpful in registration and some of the small parts are actually outliers. Thus, we extract the most dominant parts (we call each dominant part a *block*) from each point cloud. To this end, we build a graph where pairs of points are connected by edges if the distance between them is smaller than a threshold (20cm in our implementation). Then, the largest connected component is extracted as the representative block of the scan. The blocks (optionally down sampled) of all the scans are used as the input to our method.

4.2 Portal detection

Given that blocks may not have overlaps, we rely on portals to build the connections between them. We consider two types of portals: door portals and window portals. Figure 4 (a) shows an example of door portals that we use to connect interior and exterior scans of a building. One encouraging characteristic of the portals is that, unlike other objects that suffer from occlusions (such as chairs), portals reside in walls so that their shapes are usually clearly captured by the scanner (see Figure 4 (b)), providing promising hints for registering the interior scan to the exterior scan of a room (see Figure 4 (c)). Given these properties of portals, we propose a portal detection algorithm based on heuristics.

Our portal detection starts with extracting walls from the blocks. We first extract planes from each block using RANSAC [Schnabel et al. 2007]. Then, large vertical planes are labeled as walls. To identify the ground, we require upright orientation of the scans, which in principle could be automatically computed by analyzing the data [Fu et al. 2008; Ghanem et al. 2015]. However, we mainly use the upright orientation information provided by the scanners, or we let the user select the ground plane to define the upright orientation.

Next, we extract the clearly captured boundaries of the portals that will then be used to match one block with others in the assembly step. Without ambiguity, we optionally call these boundary points a *portal* in the remainder of the paper. There are multiple ways for extracting the boundaries from a point cloud. One idea is to run a line-sweeping algorithm along both horizontal and vertical directions. Then, the gradient extrema in the point distribution along the sweeps give extracted horizontal and vertical lines, as in [Nan et al. 2010]. This is simple and effective for detecting rectangular-shaped boundaries from a planar point cloud. To ensure robust matching between portals, more general shapes of portals (e.g., a door with an arc on the top) should also be considered. Another idea is to analyze the surface variations in the point cloud as proposed by Pauly et al. [2003]. This works very well for scans of free-from surfaces with a moderate level of noise, but it is not suitable for our data as most surfaces are planar and higher-level noise exists. Considering that portals are 2D shapes lying on the walls, we propose to detect the boundaries by projecting the points onto the supporting planes of the walls, followed by sequentially applying two morphological operations for images: dilation and erosion. After that, portals are defined as the 3D points whose projections are within a threshold to the images boundaries (10 pixels in our implementation). Figure 4 (b) shows the portal detection result of the two doors.

In the unlikely cases where automatic detection fails, we allow the user to remove and add portals to improve the automatic results. To do a manual correction, the user needs to roughly mark only the region containing a false positive (a non-portal detected as a portal) or a false negative (a portal not detected). Note that we do not require all portals to be correctly detected because the subsequent block assembly step can tolerate some errors.

4.3 Graph construction

Once the portals are extracted from the blocks, we construct a connection graph that encodes the possible pairwise matching of the portals of all the blocks. Specifically, each node of the graph represents a portal of a block, and each edge in the graph connects a pair of portals from two different blocks. To cope with scans that demonstrate overlap and sufficient correspondence, we also encode all the blocks as nodes in the graph, but we connect pairs of blocks by an edge only if they have sufficient common key points to be

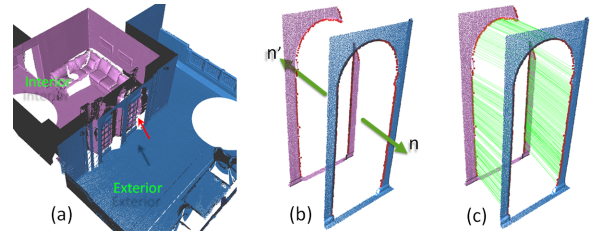


Figure 4: Portal detection and matching. (a) Two individual scans capturing the interior (purple) and exterior (blue) of a room. The red arrow indicates two door portals, one on each side of the door. (b) The extracted portals (red points). (c) The matching of the two portals. Green lines indicate the correspondences of the points.

registered with each other. More details on key point extraction and matching are described by Gelfand et al. [2005]. Note that when constructing the graph, we connect scan pairs only if they do not penetrate into each other (see Section 4.5). This penetration test will prune a large number of false local registrations. After the connection graph is constructed, we assign to each edge a matching score (encoding a distance measure) and a transformation (encoding the relative position and orientation) between the two blocks. The computed transformation will then be used for aligning the two blocks in the subsequent block assembly step.

The matching score and transformation between pairs of portals or pairs of blocks are computed as follows:

- **Pairwise portal matching.** We first roughly align the points of the two portals using PCA (Principal Component Analysis) [Liu and Ramani 2009]. Then, we represent the data of the two portals with Gaussian Mixture Models (GMMs) and compute a non-linear registration by finding the alignment between the two GMMs [Jian and Vemuri 2011]. The L_2 distance between the registered GMMs gives us a distance measure of the portal pair. Note that by computing a non-rigid transformation, our portal matching algorithm is robust to imperfections (i.e., noise, outliers, possible occluded boundaries, and non-linear distortions) in the detected portals. As can be seen in Figure 4 (c), although the upper part of the portal in the interior scan is occluded by a curtain during the scanning, our algorithm based on non-linear registration successfully extracts dense correspondences between the two portals. From these correspondences, we can compute a unique rigid transformation (rotation and translation) as an initial alignment of the two blocks. Since the supporting planes (e.g., the walls where the portals reside) of the two blocks are parallel, we compute the transformation such that it conforms to this parallelism constraint. Specifically, the constrained transformation can be computed by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{R}, \mathbf{t}} \quad & \sum_{i=1}^n \|\mathbf{q}'_i - \mathbf{R}\mathbf{q}_i - (\mathbf{t} + t_{wall}\mathbf{n})\|^2 \\ \text{s.t.} \quad & \mathbf{n}' + \mathbf{R}\mathbf{n} = 0 \end{aligned} \quad (1)$$

where $\mathbf{Q} = \{\mathbf{q}_i\}$ and $\mathbf{Q}' = \{\mathbf{q}'_i\}$ are the n pairs of corresponding 3D points, \mathbf{n} and \mathbf{n}' denote the normals of the two supporting planes (see Figure 4 (b)), and \mathbf{R} , \mathbf{t} are the rotation and translation to be computed, respectively. Here, we assume that the walls of a building have a uniform thickness, t_{wall} , that is specified by the user for both interior and exterior walls. Our ICP-based refinement step after block assembly will correct minor deviations from this assumption. This

is a quadratic programming problem under linear equality constraints. We solve it using an iterative active set method (i.e., the QP-BLEIC solver provided by [Bochkanov 2015]).

- **Pairwise block matching.** For a pair of blocks, we first extract a set of key points using a variant of the Harris key point detector [Harris and Stephens 1988] and compute their Point Feature Histogram (PFH) descriptors at each key point [Rusu and Cousins 2011]. Since the number of the key points is significantly smaller than the total number of points in a block, we can perform efficient matching between the two blocks. To do so, we exploit a RANSAC-based alignment of the two sets of key points followed by an ICP refinement step. Since the scans we are dealing with contain only rigid transformations, we randomly choose three pairs of key points that have the same relative distances. Then, the distance between the two blocks is defined by the average pointwise distance between the two blocks after ICP refinement. We then compute the transformation between the two blocks from the corresponding key points using the method described in [Horn 1987].

We normalize the two types of distances separately into the range of $[0, 1]$. Then we assign the score s_i to the i_{th} edge in the connection graph, which is defined as $s_i = 1 - d_i$, where d_i is the aforementioned normalized distance.

4.4 Block assembly

Now we detail our framework for the global assembly of the blocks based on the connection graph. We first introduce the energy terms. The objective function is a weighted sum of these terms. In the next section, we introduce an acceleration algorithm to efficiently solve the optimization problem.

Our block assembly algorithm is designed to choose an optimal set of pairwise matchings representing a globally consistent alignment of all the input blocks. As each pairwise matching is encoded as an edge with a matching score in the connection graph, the solution to our block assembly problem is equivalent to selecting a subset of the edges of the graph such that the overall confidence in the matching is as high as possible, while all the nodes in the graph remain connected and a unique transformation of each block can be computed with respect to an arbitrarily chosen reference block. To this end, we formulate the block assembly as a combinatorial optimization problem based on linear integer programming.

Let $\mathbf{B} = \{b_1, \dots, b_M\}$ denote the M input blocks and let $\mathbf{X} = \{x_1, \dots, x_N\}$ denote the binary labels for both pairwise portal matchings and pairwise block matchings encoded in the connection graph. Our objective function is then defined as the sum of two energy terms: matching confidence and connectivity.

- **Matching confidence.** The matching confidence term, $E_m(\mathbf{X})$, measures the overall alignment quality of the final registration. It is defined as the average matching score of all the pairwise matchings:

$$E_m(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N (s_i \cdot x_i), \quad (2)$$

where s_i is the matching score associated with the i_{th} edge. The variable x_i encodes if an edge is chosen (1) or not chosen (0). Thus, this term encourages the final alignment of all the blocks to be tight.

- **Connectivity.** To favor as many as possible blocks to be registered, we prefer that more of the edges (pairwise matchings) remain in the final registration. Thus, we introduce a

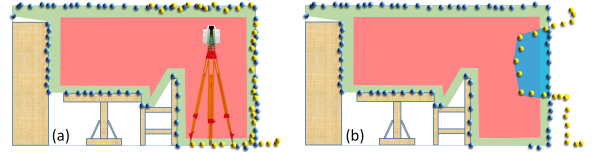


Figure 5: Non-conflicting local registration (a) and conflicting local registration (b) of two scans. The blue and yellow dots represent two scans to be registered. The red region indicates the free space of the blue scan. In the corresponding octree structure, the free space is shrunk by 10cm (the green region) to tolerate noise. The magnitude of the conflict in the right example is measured by the volume of the blue region.

connectivity term, $E_c(\mathbf{X})$, that is defined as the portion of the edges remaining for aligning pairs of portals or blocks:

$$E_c(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N x_i. \quad (3)$$

To ensure global consistency, we add two types of exclusiveness constraints to the optimization problem. The first type is that one portal in a block cannot be aligned with multiple portals. The second type forbids points from penetrating into the free space of other blocks, which is defined as the 3D volume visible to the scanner. Figure 5 (a) illustrates the free space. In our implementation, we use an octree constructed on each block to approximate the space and to query if points are transformed into the empty space of other blocks. To distinguish the two types of constraints, we call the latter a *conflict* constraint in the remainder of the paper. Our global registration optimization can be formulated as

$$\begin{aligned} \max_{\mathbf{X}} \quad & E = \lambda_m E_m + \lambda_c E_c \\ \text{s.t.} \quad & \begin{cases} \sum_{i \in \mathcal{N}(p_j)} x_i \leq 1, & p_j \in \mathbf{P} \text{ and } 1 \leq i \leq N \\ C_k = \emptyset, & C_k \in \mathbb{P} \\ x_i \in \{0, 1\}, & 1 \leq i \leq N, \end{cases} \end{aligned} \quad (4)$$

where \mathbf{P} denotes all the portals and $i \in \mathcal{N}(p_j)$ denotes the edges connecting portal p_j with its neighboring portals in the connection graph. Thus, $\sum_{i \in \mathcal{N}(p_j)} x_i$ counts the number of portals connected with p_j . This constraint ensures that each portal is connected only to one or zero other portals. The set \mathbb{P} denotes the power set of all possible alignments, which can be obtained by traversing all the sub-graphs of the connection graph and checking if points transformed into the free space of other blocks exist in the sub-graph. Thus, the conflict constraint, $C_k = \emptyset$, ensures that no point penetrates into the free space of other blocks. λ_m and λ_c are two weights that balance between the two competing terms.

4.5 Acceleration algorithm

Theoretically, the optimal alignment of all the blocks could be obtained by solving the constrained optimization problem defined in Equation 4. This is a typical linear integer programming problem that can be efficiently solved given a moderate number of constraints. In our problem, however, enumerating the constraints requires exploring the power set of all possible alignments of the blocks, which is impractical for a larger number of scans [Huber 2002]. To accelerate our algorithm, we modify two aspects of our optimization formulation. First, instead of enumerating the

extremely large number of conflicts in the full power set, we detect local conflicts in only a small number (i.e., 4) of blocks. Second, we introduce another energy term ‘conflict tolerance’ that encodes the locally conflicting alignments as soft constraints. By making these modifications to the formulation, we achieve the optimal assembly of the blocks by iteratively adding constraints and solving the modified optimization problem.

Constraint set. Given that the number of locally conflicting alignments is much smaller than that of the power set, our acceleration relies on the fact that forbidding locally conflicting alignments can contribute to the consistency of the final global registration. These locally conflicting alignments will be encoded as exclusiveness constraints to discourage these local alignments in the subsequent optimization step.

Conflict tolerance. An exact conflict detection algorithm tends to return erroneous results when a high-level of noise or a large amount of outliers exists in the scans. There are two types of errors in conflict detection. First, conflicting local alignments of a few scans may be detected as non-conflicting. This does not affect much of the final block assembly because these conflicting local registrations usually have very low matching confidence. Since we are performing a global optimization, the matching confidence term will ensure good matchings are chosen in the final registration. Second, good local alignments may be detected as conflicting. In such a case, these false positive conflicting local alignments will be forbidden in the final registration if they are encoded as hard constraints in the optimization. To prevent good matchings from being pruned in the final registration, we modify our objective function to tolerate some level of conflicts caused by noise and outliers, and we encode these conflicts as soft constraints in the objective function. For every pairwise alignment represented by an edge in the connection graph, we quantify its conflict as the volume of the conflicting regions. Thus, the conflict tolerance term can be defined as

$$E_t(\mathbf{X}) = \frac{\frac{1}{N} \sum_{i=1}^N (v_i \cdot x_i)}{\frac{1}{M} \sum_{j=1}^M \text{vol}(b_j)}, \quad (5)$$

where v_i is the volume of the conflicting region when aligning two blocks connected by the i_{th} edge in the connection graph, and $\text{vol}(b_j)$ measures the volume of block b_j . The numerator defines the average conflicting volume, and the denominator represents the average volume of all the input blocks. In our implementation, the volume of a block is computed from its convex hull constructed using a down-sampled version of the block. Figure 5 (b) illustrates the conflicting volume.

Now the final constrained optimization formulation has the following form:

$$\begin{aligned} \min_{\mathbf{X}} \quad & E = \lambda_m(1 - E_m) + \lambda_c(1 - E_c) + \lambda_t E_t \\ \text{s.t.} \quad & \begin{cases} \sum_{i \in \mathcal{N}(p_j)} x_i \leq 1, & p_j \in \mathbf{P} \text{ and } 1 \leq j \leq N \\ x_i \in \{0, 1\}, & 1 \leq i \leq N, \end{cases} \end{aligned} \quad (6)$$

where λ_t is the weight of the newly introduced conflict tolerance term that penalizes the overall volume of the conflicting alignments. Other symbols are the same as in Equation 4. Note that since we want to favor both matching confidence (E_m) and connectivity (E_c), we subtract them from 1 so that the problem is equivalent to the minimization of the new energy function. In our implementation, we set λ_m , λ_c , and λ_t to be 1, 10, and 1, respectively.

With a considerably smaller number of exclusiveness constraints, our optimization problem defined in Equation 6 can now be solved very efficiently (we use a solver provided by [Gurobi 2015]), which

yields a consistent alignment of most of the blocks. In the very unlikely case where very few local alignments remain conflicting after each optimization step, we add them back to the constraint set and rerun the optimization. We iterate this process until a conflict-free registration is achieved. Finally, we refine the registration using an ICP step [Rusinkiewicz and Levoy 2001].

5 Results and Discussion

We have applied our block assembly algorithm to a variety of large indoor scenes and entire buildings. The rooms in all the example scenes are scanned with their doors in arbitrary positions, unless explicitly stated otherwise. Experiments demonstrate that portals are valid and effective features for augmenting local registrations. By using the acceleration strategy, our method is capable of registering a large number of scans.

Figures 1 and 6 show residential buildings with two floors. Both the interior and exterior were scanned. The initial scans are captured using a combination of the static Leica scanner and the FARO hand-held scanner. The interiors of the two buildings consist of multiple isolated rooms. We scanned these rooms with their doors closed and thus each room is represented by a single scan. In contrast, the scans of the exteriors of the buildings have significant overlapping regions. Our method successfully registers the non-overlapping scans of the rooms along with the building exterior into single consistent point clouds covering entire buildings.

In Figure 7, 24 classrooms on the same floor are scanned using only the static Leica scanner. In this scene, the largest classroom was captured by two scans and the smaller classrooms were captured by one scan each. Note that the long hallways are captured by multiple scans with significant overlapping regions. As can be seen from the top view, our block assembly method can obtain a globally consistent registration of the entire floor.

Figures 8 and 9 show the global registration results of two large sports centers. Note that the rooms in Figure 8 are positioned along a curved hallway and some rooms in these two buildings have curved walls. Since the curved walls are locally flat and clean, our automatic portal detection algorithm successfully detects most of their door portals (see Table 1). Using these portals along with the overlapping regions in the scans of the hallways, our method registers all the scans into a single consistent point cloud.

Figures 10 and 11 show the registration results of two relatively small scenes. In Figure 10, the isolated rooms are scanned using the FARO hand-held scanner and the hallways using the static Leica scanner. Although there are no actual doors between individual rooms, the openings connecting rooms to the hallways turn out to be effective portals connecting the isolated rooms. The scene consisting of 4 rooms in Figure 11 is scanned using only the Kinect One scanner. We can see that although the initial scans are more noisy than the data from the other two scanners, our method still successfully extracts the portals to register the isolated rooms into a single point cloud. Since we are seeking a rigid global registration, the result is less accurate due to the noise and distortion (e.g., in the walls) in the input.

Portal detection. As can be seen from Table 1, our portal detection can automatically detect most of the portals requiring very few manual corrections. This is because portals usually do not suffer from occlusions and their boundaries can be clearly captured by the scanners. The performance of our portal detection algorithm can also be evaluated by computing precision and recall for each example, which is plotted in Figure 12.

Effect of portals. From the percentage of edges selected by our

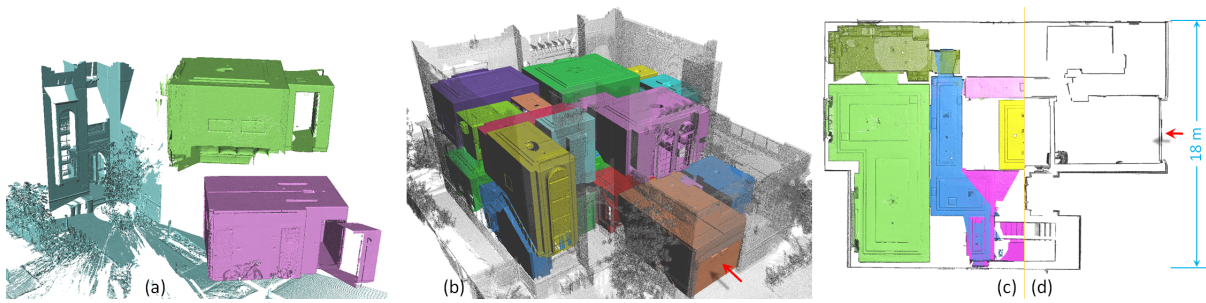


Figure 6: Global registration of a two-floor residential building consisting of 26 isolated rooms. (a) 3 out of the 31 input scans. (b) The registration result (isolated rooms are in different colors and the exterior of the building is down-sampled and rendered in dark gray). (c) A top view of the registered rooms in the first floor. (d) A cross section of the first floor at a height of 2.0m.

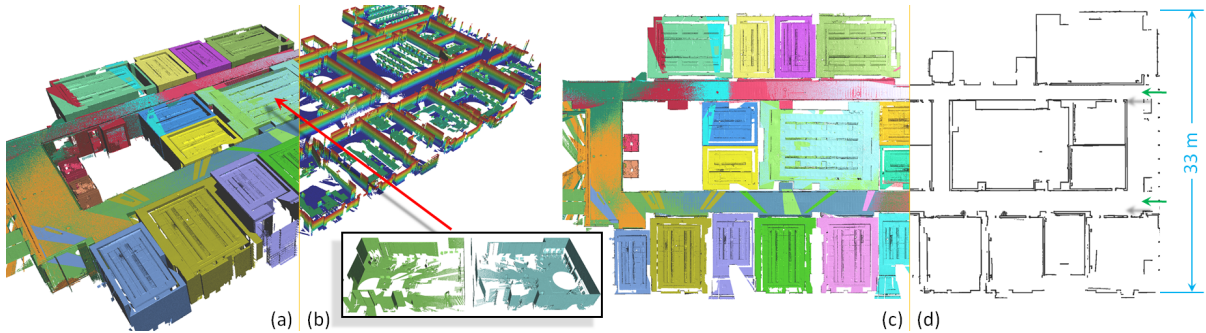


Figure 7: Registration of 24 classrooms and 2 long hallways (see the green arrows) lying on the same floor. (a) The registered scans rendered with roofs. (b) The registered scene without roofs (color coded by Z-values). The inset shows the largest classroom captured by two scans. (c) A top view of this floor with roofs. (d) A cross section of the registered scans.

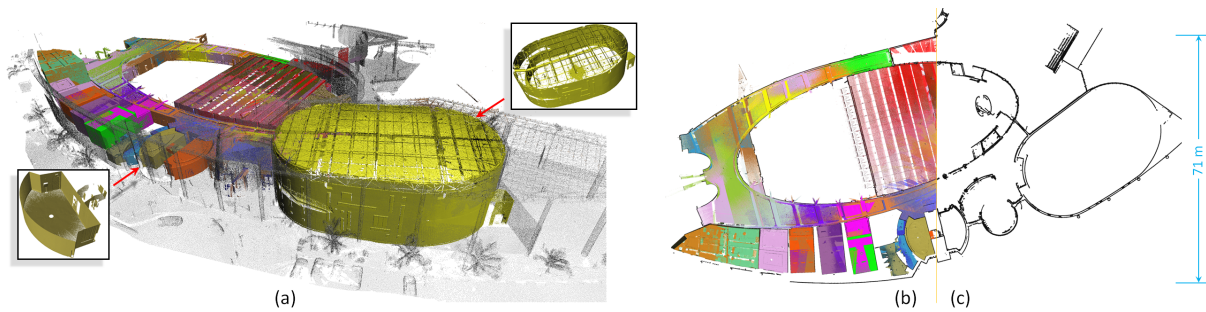


Figure 8: Registration result of a sports center consisting of 8 isolated rooms positioned along a curved hallway. (a) The registered scans. (b) A top view of the registration result. (c) A cross section of the registered scans. Individual scans are coded in different colors.

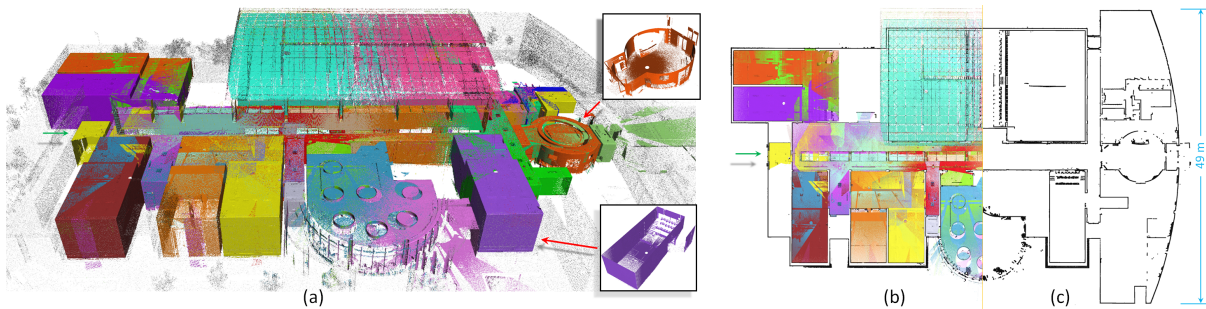


Figure 9: Registration result of another sports center consisting of 26 individual rooms. (a) The registered scans. (b) A top view of the registration result. (c) A cross section of the registered scans. Individual scans are coded in different colors.

Table 1: A summary of the number of scans (#Scans), number of isolated rooms (#Rooms), number of portals detected by our method and percentage of portals corrected for generating the result (#Portal / %Manual), number of edges connecting pairs of portals and percentage of them selected in the final result (#E-portal / %Selected), number of edges connecting pairs of overlapping blocks and percentage of them selected in the final result (#E-overlap / %Selected), runtime of the initial constraint detection (Constraint), runtime of the block assembly (Assembly), and the number of iterations taken in the optimization (#Iter) for each example.

Figure	#Scans	#Rooms	#Portal / %Manual	#E-portal / %Selected	#E-overlap / %Selected	Constraint	Assembly	#Iter
1	37	29	78 / 3 %	671 / 5.2 %	14 / 93 %	55 sec	73 sec	5
10	8	5	15 / 0 %	67 / 9.0 %	1 / 100 %	0.3 sec	4 sec	1
11	4	4	10 / 0 %	7 / 42.8 %	0 / -	0 sec	4 sec	1
6	31	26	73 / 5 %	538 / 4.3 %	16 / 100 %	45 sec	51 sec	19
7	49	24	74 / 0 %	1100 / 5.5 %	31 / 74 %	132 sec	8 sec	1
8	67	11	87 / 4.6 %	487 / 5.7 %	52 / 99 %	51 sec	11 sec	1
9	73	26	152 / 6 %	1472 / 2.3 %	89 / 93 %	284 sec	25 sec	2
15	156	102	376 / 2 %	10528 / 1.7 %	65 / 95 %	263 min	33 sec	9

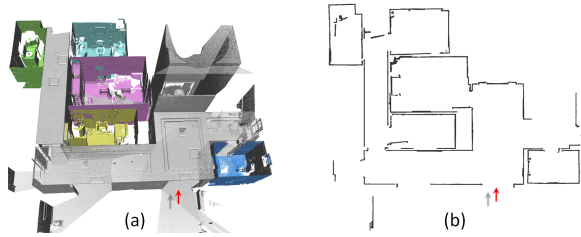


Figure 10: A scene with its rooms scanned using the FARO hand-held scanner. (a) Our registration result. The scans in gray color represent the hallway. (b) A cross section.

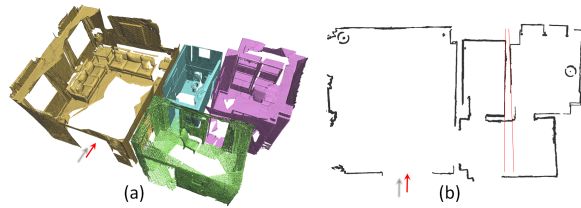


Figure 11: A scene consisting of 4 rooms captured purely by the Kinect One scanner. (a) Our registration result. (b) A cross section. Note that the Kinect One scanner introduces challenging distortions, e.g., the red lines indicate two walls that should be parallel.

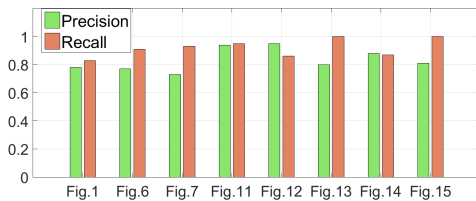


Figure 12: Precision and recall of the portal detection algorithm for each set of scans used in our experiments.

optimization (Table 1), we can see that most of the feature-based matchings are valid. This is expected as geometric features are stable. In contrast, only a small portion of portal matchings is selected. This is because real scenes usually contain doors (or windows) having the same geometry and thus many pairs have high matching scores, but most of these portal matchings cause conflicts and thus they are forbidden by our global optimization. From the percentage of edges chosen to connect portals and the percentage

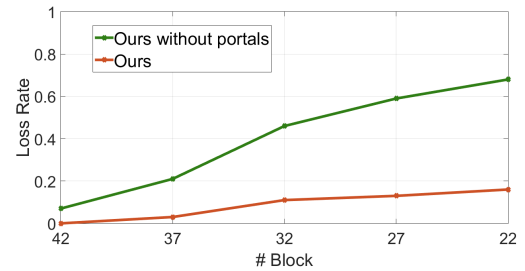


Figure 13: The effect of portals in registering the scans of the building shown in Figure 1. We first introduce new scans by positioning the scanner at the doors. Then, we gradually drop 5 scans each time and run the two methods. The loss rate is computed as the portion of scans that cannot be registered.

chosen to connect overlapping blocks, we can also conclude that geometric features control the registration when strong overlaps exist, and the portals influence the result only if common features are too weak or absent from the scans. All our tests suggest that portals are valid and important features, and they play a critical role in registering building scans.

To examine the effect of portals further, we randomly drop scans of a scene and compare the registration results with and without portal information. Before dropping scans, we first introduce more scans obtained by positioning the scanner at the doors and we leave the doors open during the scanning. Then, we gradually drop scans and run our method both with and without portal information. We record the number of the scans that cannot be registered and plot the ratio of unregistered scans. Figure 13 presents a comparison of the scene shown in Figure 1. We can see that although more overlapping regions are introduced, it is still impossible to obtain a satisfactory registration without portal information. In contrast, the use of portals ensures a more stable registration as scans are gradually dropped.

Iterative constraint adding. By using the strategy of iteratively adding local constraints instead of exhaustively enumerating them at the beginning of the optimization, we can efficiently solve our optimization problem defined in Equation 6. Experiments reveal that only a few iterations are sufficient to achieve a globally consistent registration of the scans. Table 1 presents the runtime of the block assembly step and the number of iterations for all the examples in the paper. We also compare our strategy of iteratively adding constraints with the exhaustive constraint enumeration on the scene shown in Figure 1. We record the timing for both methods by incrementally adding blocks into the scene. The run times of the

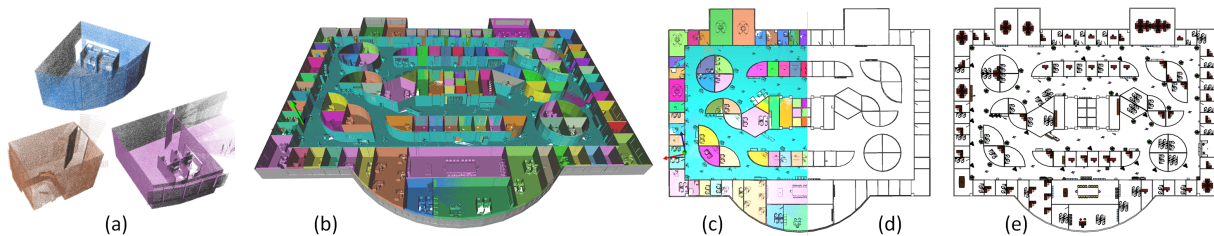


Figure 15: Registration of a large synthetic scene consisting of 102 isolated rooms. (a) Three out of 156 input scans. (b) Registered scans of the entire scene. (c) A top view of the assembled blocks. (d) A cross section of the entire floor. (e) A cross section.

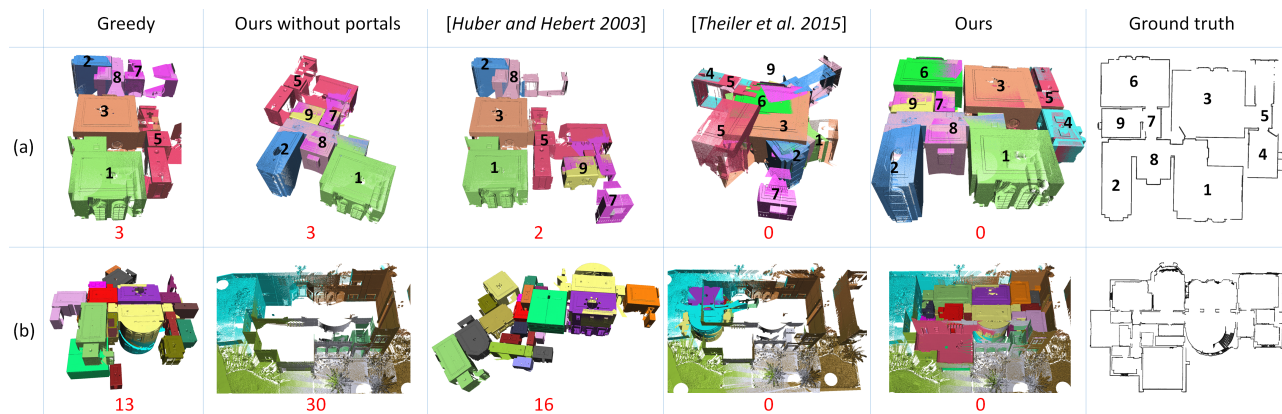


Figure 16: Comparison of our global registration method with four competing methods on two sets of scans. (a) The scans of the second floor of the building shown in Figure 6. (b) The scans of the entire building shown in Figure 1. The value at the bottom of each sub-figure indicates the number of scans that cannot be registered by the corresponding method.

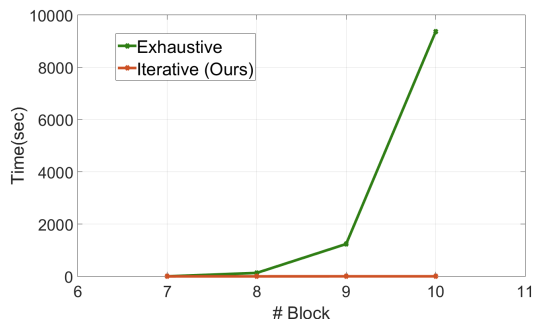


Figure 14: A comparison of our iterative constraint adding strategy with the exhaustive constraint enumeration on the scene shown in Figure 1. Since exhaustive constraint enumeration is too slow, we stopped when the number of blocks increased to 10.

two strategies are compared in the plot in Figure 14. We can see that our strategy of iteratively adding constraints can significantly speed up the optimization process and that exhaustive constraint enumeration will be too time-consuming for the larger examples presented in this paper.

Scalability. In addition to the examples of real scenes discussed above, we also tested our optimization-based block assembly algorithm on a large synthetic scene. Figure 15 shows this synthetic scene consisting of 102 isolated rooms, captured by 134 virtual scans. Our strategy of iteratively adding constraints enables us to register this large number of blocks efficiently (see Table 1 for the timing).

Timing. The run times of initial constraint detection and block assembly for each set of scans are given in Table 1. For the initial constraints, we compute local conflicts for every triplet of blocks, which is the most time-consuming step. Assuming a scene consisting of M scans, the total number of conflict tests is $C_M^3 = M(M-1)(M-2)/6$. These are not large numbers of tests and can be efficiently performed for most of the scenes described in our paper. However, for the synthetic scene consisting of 156 scans (see Figure 15), the total number of conflict tests is large (i.e., 620, 620) and it takes more than four hours.

Comparison. We also conducted comparisons of our global registration method with four other approaches. The first was a greedy approach in which we sorted the matching scores of all the edges in the connection graph. Then, starting from the edge with the highest matching score, we incrementally searched and added scans that had the highest matching score with the registered ones. To ensure consistent registration, only blocks that did not cause a conflict could be added. The second approach was a global registration method using our framework but omitting all portal matchings. The third approach was a global registration method proposed by Huber and Hebert [2003]. The fourth approach was another global registration method proposed by Theiler et al. [2015]. As can be seen from Figure 16, the results of the other four registration methods provided completely wrong room layouts due to ambiguities in the pairwise matchings. Moreover, the first three approaches managed to register only part of the scans that contain common regions. In contrast, our method successfully registered all the scans in their correct positions.

Acquisition guideline. Since portals reside in walls, their shapes are usually clearly captured. This can be seen in Figure 17 where the same room was scanned at different positions. However, a portal

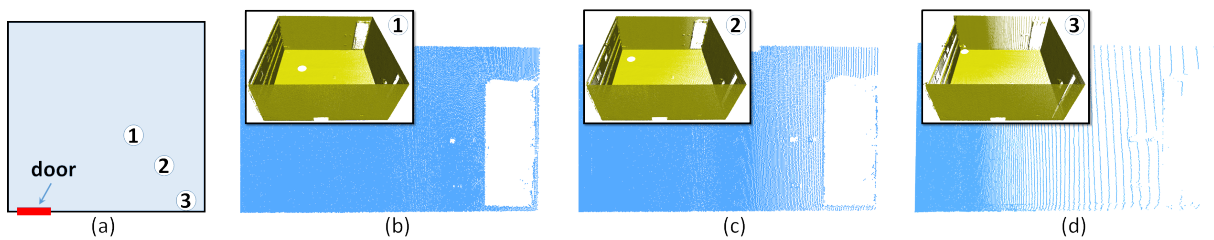


Figure 17: A room with a door scanned at three different positions with an increasing scanner angle. (a) An illustration of the door portal and scanner positions. (b), (c), and (d) show the extracted wall where the door resides for the each scanner position. The inset in each sub-figure shows the original scan of the room.

may not be properly captured in some extreme cases. For example in Figure 17 (d), the scanner was positioned at the corner of the room with a very large oblique scanning angle. This results in a point cloud with a severe decrease in density and the door portal is only partially captured. In such a case, our method cannot detect this door portal. In general, to ensure that portals can be properly captured, we advise that scanning from room corners be avoided.

Limitation. When scans do not have overlap, our method relies purely on the geometry of the portals and the non-penetrating constraints for registration. In this case, our method may not position rooms of identical geometry in their correct positions. For example, the synthetic scene shown in Figure 15 consists of many duplicated rooms. In our result, 36 rooms (i.e., 35%) are in wrong positions due to the rooms and portals having the exact same geometry. In this example, we regard such assembly as a valid interpretation of the input scene.

6 Conclusions and Future Work

We presented a global registration framework for aligning building scans. Our local registration strategy relies on portals to establish the potential connections between non-overlapping scans or scans with only limited overlap, as well as traditional local registration using feature matching. We formulate global registration as a combinatorial optimization problem. We also proposed an acceleration strategy by iteratively adding constraints, which significantly improves the overall efficiency of our method. In our results, we demonstrated the capability of our method in registering scans of indoor scenes both with and without exterior scans. Unlike existing methods that are very sensitive to the positioning of the scanner to ensure reasonable overlap between neighboring scans, we do not make any particular assumptions on data acquisition, allowing more freedom for the user to scan the scenes.

In future work, we plan to exploit a crowdsourcing strategy for scanning even larger buildings so that isolated rooms can be simultaneously scanned by many scanners and users. Scanning in such a fashion can lead to very large overlaps and missing parts in the acquisition. We believe that this will lead to very interesting and challenging registration problems that have not been tackled before.

Acknowledgements

We thank the reviewers for their valuable comments, Virginia Unkefer for proofreading, and FalconViz for lending us the Leica ScanStation C10 laser scanner. We also thank NVIDIA Corporation for the donation of the Quadro K5200 GPU used for rendering the point clouds. This work was supported by the Office of Sponsored Research (OSR) under Award No. OCRF-2014-CGR3-62140401 and the Visual Computing Center at KAUST.

References

- AIGER, D., MITRA, N. J., AND COHEN-OR, D. 2008. 4-points congruent sets for robust pairwise surface registration. In *ACM Transactions on Graphics*, vol. 27, 85.
- BESL, P. J., AND MCKAY, N. D. 1992. Method for registration of 3-d shapes. In *Robotics-DL tentative*, International Society for Optics and Photonics, 586–606.
- BOCHKANOV, S., 2015. Alglib library. <http://www.alglib.net/>.
- BROWN, B., AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-D scans. *SIGGRAPH* 26, 3 (Aug.).
- CHEN, J., BAUTEMBACH, D., AND IZADI, S. 2013. Scalable real-time volumetric surface reconstruction. *ACM Transactions on Graphics* 32, 4, 113.
- CHOI, S., ZHOU, Q.-Y., AND KOLTUN, V. 2015. Robust reconstruction of indoor scenes. In *CVPR*, 5556–5565.
- CUTLER, W. 1978. The six-piece burr. *Journal of Recreational Mathematics* 10, 4, 241–250.
- DARDARI, D., CLOSAS, P., AND DJURIC, P. M. 2015. Indoor tracking: Theory, methods, and technologies. *Vehicular Technology, IEEE Transactions on* 64, 4, 1263–1278.
- FU, H., COHEN-OR, D., DROR, G., AND SHEFFER, A. 2008. Upright orientation of man-made objects. In *ACM transactions on graphics*, vol. 27, 42.
- FU, C.-W., SONG, P., YAN, X., YANG, L. W., JAYARAMAN, P. K., AND COHEN-OR, D. 2015. Computational interlocking furniture assembly. *ACM Trans. Graph.* 34, 4 (July), 91:1–91:11.
- GELFAND, N., MITRA, N. J., GUIBAS, L. J., AND POTTMANN, H. 2005. Robust global registration. In *SGP*, vol. 2, 5.
- GHANEM, B., THABET, A., CARLOS NIEBLES, J., AND CABA HEILBRON, F. 2015. Robust manhattan frame estimation from a single rgb-d image. In *CVPR*, 3772–3780.
- GPS.GOV, 2015. Gps accuracy. <http://www.gps.gov/systems/gps/performance/accuracy/>.
- GUROBI, 2015. Gurobi optimization. <http://www.gurobi.com/>.
- HARRIS, C., AND STEPHENS, M. 1988. A combined corner and edge detector. In *Alvey vision conference*, vol. 15, Citeseer, 50.
- HORN, B. K. 1987. Closed-form solution of absolute orientation using unit quaternions. *JOSA A* 4, 4, 629–642.
- HUANG, Q.-X., FLÖRY, S., GELFAND, N., HOFER, M., AND POTTMANN, H. 2006. Reassembling fractured objects by

- geometric matching. *ACM Transactions on Graphics* 25, 3, 569–578.
- HUANG, P., BUDD, C., AND HILTON, A. 2011. Global temporal registration of multiple non-rigid surface sequences. In *CVPR*, IEEE, 3473–3480.
- HUANG, H., GONG, M., COHEN-OR, D., OUYANG, Y., TAN, F., AND ZHANG, H. 2012. Field-guided registration for feature-conforming shape composition. *ACM Transactions on Graphics* 31, 6, 179.
- HUBER, D. F., AND HEBERT, M. 2003. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing* 21, 7, 637–650.
- HUBER, D. F. 2002. *Automatic three-dimensional modeling from reality*. PhD thesis, Carnegie Mellon University Pittsburgh.
- IZADI, S., KIM, D., HILLIGES, O., MOLYNEAUX, D., NEWCOMBE, R., KOHLI, P., SHOTTON, J., HODGES, S., FREEMAN, D., DAVISON, A., ET AL. 2011. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *UIST*, ACM, 559–568.
- JIAN, B., AND VEMURI, B. C. 2011. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8, 1633–1645.
- KAHLER, O., PRISACARIU, V. A., REN, C. Y., SUN, X., TORR, P. H. S., AND MURRAY, D. W. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. *IEEE Transactions on Visualization and Computer Graphics*.
- LI, H., SUMNER, R. W., AND PAULY, M. 2008. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, vol. 27, Wiley Online Library, 1421–1430.
- LIU, Y.-S., AND RAMANI, K. 2009. Robust principal axes determination for point-based shapes using least median of squares. *Computer-Aided Design* 41, 4, 293–305.
- MATTERPORT, 2016. Matterport 3d models of real interior spaces. <https://matterport.com/>.
- NAN, L., SHARF, A., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2010. Smartboxes for interactive urban reconstruction. In *SIGGRAPH*, ACM, SIGGRAPH, 93:1–93:10.
- NEWCOMBE, R. A., IZADI, S., HILLIGES, O., MOLYNEAUX, D., KIM, D., DAVISON, A. J., KOHLI, P., SHOTTON, J., HODGES, S., AND FITZGIBBON, A. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR)*, IEEE, 127–136.
- NIESSNER, M., ZOLLHÖFER, M., IZADI, S., AND STAMMINGER, M. 2013. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics* 32, 6, 169.
- PAULY, M., KEISER, R., AND GROSS, M. 2003. Multi-scale feature extraction on point-sampled surfaces. In *Computer graphics forum*, vol. 22, 281–289.
- ROTH, H., AND VONA, M. 2012. Moving volume kinectfusion. In *BMVC*, 1–11.
- RUECKERT, D., SONODA, L. I., HAYES, C., HILL, D. L., LEACH, M. O., AND HAWKES, D. J. 1999. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE Transactions on Medical Imaging* 18, 8, 712–721.
- RUSINKIEWICZ, S., AND LEVOY, M. 2001. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling*, IEEE, 145–152.
- RUSU, R. B., AND COUSINS, S. 2011. 3d is here: Point cloud library (pcl). In *International Conference on Robotics and Automation*.
- SCHENK, E., AND GUITTARD, C. 2009. Crowdsourcing: What can be outsourced to the crowd, and why. In *Workshop on Open Source Innovation*.
- SCHNABEL, R., WAHL, R., AND KLEIN, R. 2007. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, vol. 26, 214–226.
- SCOTT, W. R., ROTH, G., AND RIVEST, J.-F. 2003. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.* 35, 1 (Mar.), 64–96.
- SEGAL, A., HAEHNEL, D., AND THRUN, S. 2009. Generalized-icp. In *Robotics: Science and Systems*, vol. 2.
- SHARP, G. C., LEE, S. W., AND WEHE, D. K. 2004. Multiview registration of 3d scenes by minimizing error between coordinate frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 8, 1037–1050.
- TAM, G. K., CHENG, Z.-Q., LAI, Y.-K., LANGBEIN, F. C., LIU, Y., MARSHALL, D., MARTIN, R. R., SUN, X.-F., AND ROSIN, P. L. 2013. Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics* 19, 7, 1199–1217.
- THEILER, P. W., WEGNER, J. D., AND SCHINDLER, K. 2015. Globally consistent registration of terrestrial laser scans via graph optimization. *ISPRS Journal of Photogrammetry and Remote Sensing* 109, 126–138.
- WHELAN, T., KAESS, M., FALLON, M., JOHANSSON, H., LEONARD, J., AND McDONALD, J. 2012. Kintinuous: Spatially extended kinectfusion.
- WU, S., SUN, W., LONG, P., HUANG, H., COHEN-OR, D., GONG, M., DEUSSEN, O., AND CHEN, B. 2014. Quality-driven poisson-guided autoscanning. *ACM Trans. Graph.* 33, 6 (Nov.), 203:1–203:12.
- XU, K., HUANG, H., SHI, Y., LI, H., LONG, P., CAICHEN, J., SUN, W., AND CHEN, B. 2015. Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Trans. Graph.* 34, 6 (Oct.), 177:1–177:14.
- YAN, F., SHARF, A., LIN, W., HUANG, H., AND CHEN, B. 2014. Proactive 3d scanning of inaccessible parts. *ACM Trans. Graph.* 33, 4 (July), 157:1–157:8.
- ZHOU, Q.-Y., AND KOLTUN, V. 2013. Dense scene reconstruction with points of interest. *ACM Trans. Graph.* 32, 4 (July), 112:1–112:8.