Fine-To-Coarse Global Registration of RGB-D Scans

Maciej Halber
Princeton University
mhalber@cs.princeton.edu

Thomas Funkhouser
Princeton University
funk@cs.princeton.edu

Abstract

RGB-D scanning of indoor environments is important for many applications, including real estate, interior design, and virtual reality. However, it is still challenging to register RGB-D images from a hand-held camera over a long video sequence into a globally consistent 3D model. Current methods often can lose tracking or drift and thus fail to reconstruct salient structures in large environments (e.g., parallel walls in different rooms). To address this problem, we propose a "fine-to-coarse" global registration algorithm that leverages robust registrations at finer scales to seed detection and enforcement of new correspondence and structural constraints at coarser scales. To test global registration algorithms, we provide a benchmark with 10,401 manually-clicked point correspondences in 25 scenes from the SUN3D dataset. During experiments with this benchmark, we find that our fine-to-coarse algorithm registers long RGB-D sequences better than previous methods.

1. Introduction

The proliferation of inexpensive RGB-D video cameras allows for easy scanning of static indoor environments, enabling applications in many domains, including cultural heritage, real estate and virtual reality. Motivated by these, our goal is to create a method that takes a sequence of RGB-D frames captured with a hand-held camera as input and produces a globally consistent 3D model as output. We would like the algorithm to register frames robustly in a wide range of indoor environments (offices, homes, museums, etc.), execute off-line within practical computational limits, and work with data acquired by inexpensive commodity cameras, so that it can be used by non-experts.

Despite much prior work, it is still difficult to register RGB-D data acquired with a hand-held camera. Although camera poses can usually be tracked over short distances [28], local tracking often fails in textureless regions and/or drifts over long ranges [9, 30] (left side of Figure 1). These errors can be fixed with asynchronous or global optimizations based on detected loop closures [8, 19, 46]. However, finding loop closures is difficult without prior constraints in large real-world scans with multiple rooms and/or repeated structures. In our experience, even state-of-the-art global registration methods produce warped surfaces and improbable structures in these cases [8].

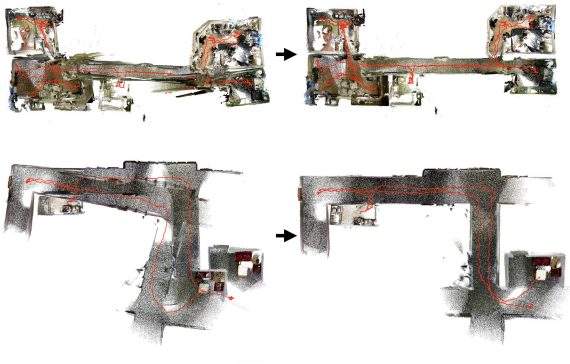

Figure 1: We present a fine-to-coarse optimization strategy for globally registering RGB-D scans in indoor environments. Given an initial registration (left), our algorithm iteratively detects and enforces planar structures and feature correspondences at increasing scales. This way, it discovers long-range constraints important for a globally consistent registration – e.g., note how opposing walls are parallel even across different rooms in our results on the right.

To address this issue, global refinement methods have been proposed based on fitting structural models [24, 25, 27] and/or aligning closest point correspondences [5, 14]. However, these methods only succeed when the alignments provided as input are nearly correct. Otherwise, they may detect and amplify erroneous constraints found in the misaligned inputs.

We introduce a new "fine-to-coarse" global registration algorithm that refines an initial set of camera poses by iteratively detecting and enforcing geometric constraints within gradually growing subsets of the trajectory. During each iteration, closest point and geo-

metric constraints (parallelism, perpendicularity, etc.) are detected and enforced only within "windows" of neighboring RGB-D frames. Windows start small, such that relative initial alignments are likely to be correct. As the algorithm proceeds, windows gradually increase in size, enabling detection of longer-range correspondences and large-scale geometric structures, leveraging the improved trajectory provided by previous iterations. This process continues until a single window includes the entire scan and a global refinement can be done robustly.

The advantage of this "fine-to-coarse" approach is that the closest point correspondences and planar structures are detected in each iteration only at the scales at which previous iterations have already aligned the scans. Enforcing these constraints in one iteration improves the registration for the next. For example in Figure 2, note how geometric constraints between walls become easier to detect in each iteration (left to right), and enforcement of those constraints gradually rectifies the reconstruction. In late iterations, the alignment is almost perfect, making it trivial to detect very large-scale structures and long-range constraints (e.g., parallel walls in different rooms), which are crucial for correct global registration.

To evaluate this algorithm and enable comparisons between future work, we have created a new registration benchmark based on the SUN3D dataset [47]. It contains 10,401 manually-clicked point correspondences in RGB-D scans containing 149,011 frames in 25 scenes, many of which span multiple rooms. During experiments with this new benchmark, we find that our fine-to-coarse algorithm produces more accurate global registrations and handles more difficult inputs than previous approaches.

Overall, the research contributions of this paper are three-fold. First, we propose a new fine-to-coarse, iterative refinement strategy for global registration of large-scale RGB-D scans. Second, we introduce a new benchmark dataset for evaluating global registration algorithms quantitatively on real RGB-D scans. Finally, we provide results of ablation studies revealing trade-offs for different components of our global registration algorithm.

2. Related Work

There has been a long history of research on registration of RGB-D images in both computer graphics and computer vision, as well as in augmented reality, robotics, and other fields [37]. The following paragraphs describe the work most closely related to ours.

Real-time reconstruction. Most prior work has focused on real-time registration motivated by SLAM applications in robotics and augmented reality [37]. Early systems use ICP to estimate pairwise alignments of adjacent video frames [4] and feature matching techniques to detect and align loop closures [2]. More recent methods align frames to a scene model, represented as a point cloud [19, 21, 34, 46] or an implicit function [6, 9, 20, 28, 42, 44, 45]. With these methods, small local alignment errors can accumulate to form gross inconsistencies at large scales [22, 30].

Off-line global registration. To rectify misalignments in on-line camera pose estimates, it is common to use off-line or asynchronously executed global registration procedures. A common formulation is to compute a pose graph with edges representing pairwise transformations between frames and then optimize an objective function penalizing deviations from these pairwise alignments [16, 19, 50, 51]. A major challenge in these approaches is to identify which pairs should be considered as loop closures. Previous methods have searched for similar images with Bag-of-Words models [2], randomized fern encodings [46], convolutional neural networks [7], and other methods. Choi et al. [8] recently proposed a method that uses indicator variables to identify true loop closures during global optimization using a least-squares formulation. In our experiments, their algorithm is successful on scans of small environments, but not for ones with multiple rooms, large-scale structures, and/or many repeated elements.

Hierarchical graph optimization. Some methods fuse subgraphs of a pose graph hierarchically to improve optimization robustness and efficiency [8, 13, 15, 33, 40]. Some of the ideas motivating these methods are related to ours. However, they detect all potential loop closures before the optimization starts. In contrast, we detect new constraints (planar relationships and feature correspondences) in the inner loop of an iterative refinement, which enables gradual discovery of large-scale structures and long-range constraints as the registration gets better.

Iterative refinement. Other methods have used Iterative Closest Point [4] to compute global registrations [5, 14, 32]. The advantage of this approach is that dense correspondences (including loop closures) are found only with local searches for closest points based on a prior alignments, rather than with global searches that consider all pairs of frames. However, ICP generally requires a good initial alignment and thus is rarely used for global RGB-D registration except as fine-scale refinement in the last step [8]. Our "fine-to-coarse" strategy addresses that specific limitation.

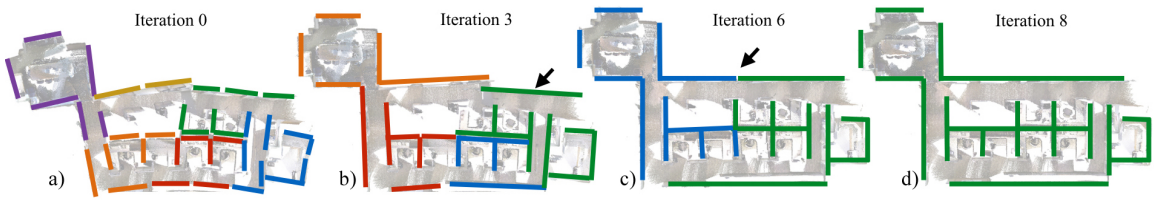

Figure 2: Schematic view of fine-to-coarse registration. Starting with initial alignment T_0 shown on the left, our algorithm detects and enforces structures in local regions (color-coded) in the first few iterations. As the algorithm progresses, the trajectory is refined, allowing for detection of larger geometrical structures. By iteration 6, we have properly aligned the wall marked by the arrow, without using explicit loop closures.

3. Approach

In this paper, we describe a global registration algorithm that leverages detection and enforcement of nearly-satisfied constraints in the inner loop of an iterative refinement procedure. The algorithm starts with an initial, imperfect registration and then follows the general E-M strategy of alternating between a discrete E-step (detecting a set of viable constraints) and a continuous M-step (solving for the camera poses that best satisfy the constraints).

Though the method is general, we consider two major types of constraints in this work: feature correspondences and planar structure relationships. During each iteration of the algorithm, constraints are created based on correspondences between closest compatible features (like in ICP) and based on geometric relationships between detected planar structures (parallelism, orthogonality, etc.). The constraints are integrated into a global optimization that refines camera poses before proceeding to the next iteration.

The key new idea is that the detection of constraints occurs in every iteration within sliding windows that grow gradually as the algorithm proceeds. In the early iterations, a small number of neighboring RGB-D frames are within each window. Since the relative camera poses of the initial alignment should be nearly correct for neighboring frames, it is possible to detect structural constraints and closest point correspondences robustly within these small windows, even if the global reconstruction is grossly inaccurate (Figure 2a). As the iterations proceed, the window size increases, enabling detection and enforcement of larger-scale and longer-range planar structures and correspondence constraints (Figure 2c). Since previous iterations have optimized the camera trajectory based on constraints discovered within smaller windows, we can expect the current trajectory estimate to be nearly correct within each window and use it to discover planar structures and feature correspondences. Ultimately, in the last iteration, the final window contains all the input data and the algorithm performs a global optimization of large-scale structures and correspondences spanning the entire trajectory in one large joint optimization (Figure 2d).

This approach has two important differences from previous work. First, it avoids a global search for pairwise loop closures – they are instead found incrementally as the registration becomes nearly aligned. Second, it discovers and enforces large-scale geometric constraints (like planar structure relationships) even though they might not be evident in the initial alignment (e.g., the parallel relationship between the leftmost and rightmost walls in the example of Figure 2 would be difficult to infer in iteration 0, but is simple to detect in Iteration 6). As a result, our method achieves significantly better registration results for large-scale scans compared to previous methods (Section 5).

4. Algorithm

The input to our system is a set of n RGB-D images I acquired with a consumer level RGB-D camera. The output is a set of camera poses T, where $T[k]$ represents position and orientation of the camera for $I[k]$.

Processing proceeds as shown in Algorithm 1. During a preprocessing phase, we first extract features F and base planar regions B from all images in I, estimate a set of local, pairwise alignment transformations L, and concatenate these local transformations to form an initial guess for global transformations T_0. Then, in each iteration i, we refine the transformations T_i by first detecting both feature correspondence constraints C_i and structural model constraints S_i based on detected clusters of coplanar base planar regions P_i and geometrical constraints (H_i and G_i) between them. We then optimize the global transformations for the next iteration T_{i+1} by minimizing an error function encoding penalties for the detected constraints. We finish by doubling the size of the window for next iteration, $l_{i+1} = 2l_i$. The following subsections describe the core ideas for each of these steps. The full implementation details appear in the supplemental material.

Input: Images I, window length l_0, n_iter;
Output: Camera transformations T;
F = ExtractFeatures(I);
B = CreateBaseProxies(I);
L = AlignAdjacentImages(I);
T_0 = ConcatenateTransformations(L);
for $i \leftarrow 0$ **to** n_iter **do**
  $\{P_i, H_i\}$ = ClusterCoplanarProxies(B_i, l_i);
  G_i = DetectGeometricConstraints(P_i);
  C_i = CreateCorrespConstraints(F_i, l_i);
  $S_i = \{P_i, G_i, H_i\}$;
  T_{i+1}=Solve $argmin_T E(T_i, S_i, C_i)$;
  $l_{i+1} = 2l_i$;
end

Algorithm 1: Fine-to-coarse refinement

4.1. Preprocessing

Extracting Features. The first step of preprocessing is to extract a dense set of features F from input RGB-D images I. Our goal in this step is to construct a set of well-spaced and repeatable features that can be matched robustly later, when searching for correspondences. We have experimented with a number of feature types, including SIFT and Harris corners in both color and depth images. However, we have ultimately found planar patches [3, 10, 11, 12, 25, 29, 31, 35, 39, 41, 43] and linear edges along creases and contours in depth images [52] to be most robust. Features are detected per pixel, for every 5^{th} frame, and then subsampled using the Poisson Dart Algorithm, with a minimum spacing between features equal to $0.05m$. Once F is created, we define a feature from image $I[k]$ at iteration i as $F_i[k][j] = \{T_i[k](p_j), T_i[k](\vec{n_j}), T_i[k](\vec{d_j})\}$ where p_j, $\vec{n_j}$ and $\vec{d_j}$ respectively denote the feature's position, normal (for planar patches) and direction (for linear edges) in the camera space.

Creating Base Planar Proxies. The next step is to extract base planar regions (which we will refer to as proxies) B from input images I. Our goal is to create base proxies that can form the basis of geometrical constraints introduced later during the fine-to-coarse refinement. To do this, we use a method based on agglomerative hierarchical clustering, where clusters of nearly co-planar features are repeatedly merged based on the compatibilities of their positions and normals (for details, see supplemental material). Once B is created, we define a proxy from image $I[k]$ at iteration i as $B_i[k][j] = \{T_i[k](p_j), T_i[k](\vec{n_j})\}$ where p_j, is the centroid of inlier features, and $\vec{n}_j$ is the fitted normal.

Aligning Adjacent Images. The final step of preprocessing is to estimate a set of local alignment transformations L for input images I. Our goal in this step is to create local alignment transformations that can be used later in the optimization to preserve the local shape of the estimated camera trajectory. To accomplish this goal, we use a pairwise image alignment approach based on Xiao et al. [47]: we detect SIFT features in images $\{I[k-1], I[k]\}$, prune out ones without valid (missing or high) depth values, and then use RANSAC on backprojected SIFT keypoints to search for the rigid transformation $L[k]$ aligning as many of these keypoints as possible. We form the initial camera-to-world transformations T_0 by simply concatenating the estimated local transformations L ($T_0[0] = I_{4\times4}$; $T_0[k] = L[k-1]T_0[k-1]$; $k \in [1, n]$). This process gives us an initial set of transformations that are locally accurate, but not globally consistent.

4.2. Fine-to-Coarse Refinement

After preprocessing the images I, the algorithm iteratively detects constraints within windows of increasing sizes and solves for all camera transformations T based on those constraints. The input to each iteration i is a window size l_i ($l_0 = 3m$) and a set of transformations T_i from the previous iteration. The output is a set of new camera transformations T_{i+1}.

Creating Co-planarity Constraints. We model co-planarity constraints by clustering the transformed base proxies B_i into representative cluster proxies $P_i[j] = \{p_j, \vec{n_j}\}$. Clustering is achieved using the same agglomerative hierarchical clustering algorithm used for base proxies extraction. However, in this step, rather than clustering features within each individual image, we cluster base proxies from different images whose distance along the estimated trajectory is less than the current window size l_i. We then insert two types of constraints into H_i, a set of feature-to-proxy constraints joining the frame features F_i to B_i, and a set of proxy-to-proxy constraints joining members of B_i to their representative cluster proxy in P_i. The constraint hierarchy implied by H_i is depicted for a single-room example in Figure 3. Note that the structure is shown for a late iteration, and thus the planar structures in green span entire walls. In contrast to previous methods based on alignment to planes [25, 35, 49], it is possible for us to detect these large planar structures because previous iterations already aligned overlapping subsets of the walls.

Creating Geometrical Relationship Constraints. We next build a set of constraints G_i representing geometric relationships between neighboring cluster proxies from the set P_i. Our goal is to detect salient relationships between planar structures (parallel, antipar-

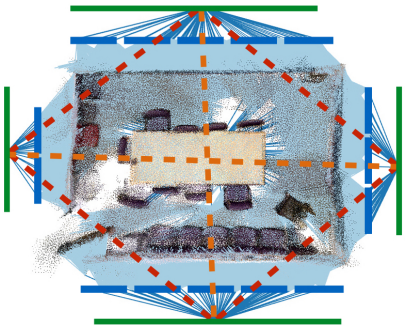

Figure 3: Exploded view of our structural model for one of the SUN3D scenes. Geometrical properties like parallelism (dashed orange) and orthogonality (dashed red) are created between parent proxies (green). Cluster proxies P_i are connected to the scan features (point-cloud) through base proxies B_i via co-planarity constraints (blue and light blue, respectively).

allel, or orthogonal) that can help guide the optimization towards the correct registration.

We create typed and weighted planar relationships for every pair of cluster proxies $\{P_i[a], P_i[b]\}$) such that the distance along a trajectory between the inlier images is less that $2l_i$. The type of the structural relationship g_{ab} and its weight w_{ab} are based on the angle between the normals, $\theta = acos(\vec{n}_a \cdot \vec{n}_b)$.

For parallel relationships the weight is defined as $w_{ab} = \exp(-\theta^2/2\sigma_\theta^2)$, for orthogonal $w_{ab} = \exp(-(\theta - \frac{\pi}{2})^2/2\sigma_\theta^2)$, and for antiparallel $w_{ab} = \exp(-(\theta - \pi)^2/2\sigma_\theta^2)$. These weights are chosen so as to guide the registration when constraints are nearly met, but have little influence when they are not. For our experiments we have chosen $\sigma = 7.5°$.

Creating Feature Correspondence Constraints. We next build a set of correspondence constraints C_i between features detected in images within the same sliding window. Following the general strategy of ICP, we construct correspondences between the closest compatible features, where compatibility is determined by a maximum distance and maximum normal angle deviation threshold, as well as a feature type check (planar features only match to planar features, etc.).

Since we expect the images within the same window to become better aligned as their poses are optimized, we set the maximum distance and angle thresholds for rejecting outliers dynamically for every pair of images based on their pairwise distance along the trajectory. The first time any two images are considered for correspondence detection (pairwise distance is $0.5l_i$) the thresholds are quite large: $0.5m$ and $30°$. Conversely, we expect close-by images to be already aligned well, thus the thresholds fall-off with the square root of decreasing pairwise distance, down to $0.2m$ and $20°$ for adjacent frames.

Finally, for performance reasons, we subsample the set of correspondences created such that the total number of them is equal to $|C_i| = 50n$.

4.3. Optimization

The final step for each iteration i is to optimize the camera transformations T_i and transformations of proxies P_i to minimize an error function encoding the detected constraints.

Our error function is a weighted sum of terms penalizing deformations of structural relationships (E_H, E_G), distances between corresponding features (E_C), misalignments of the local transformations (E_L), and large changes in transformations (E_I).

$$E(T_i, S_i, C_i) = w_H E_H(H_i) + w_g E_G(G_i) + w_C E_C(T_i, C_i) + w_L E_L(T_i) + w_I E_I(T_i, P_i)$$

Throughout the iterations weights w_H, w_G, w_C, w_L, w_I are varied linearly from an initial set of $\{1500, 1500, 1500, 1000, 1\}$ to a final one of $\{1000, 1000, 1000, 1000, 1\}$.

Structural Error. E_H and E_G are designed to enforce the constraints implied by the structural model S_i. E_H enforces coplanarity between proxies and their inlier features in depth images. Note that H_i contains both feature-to-proxy and proxy-to-proxy constraints. If we use $Q_a = \{q_a, \vec{n_a}\}$ to represent the transformed plane of a feature or proxy, we can write each error term including all these constraints as:

$$E_H(H_i) = \sum_{j=1}^{|H_i|} E_{cp}^{\rightarrow}(Q_a, Q_b) + E_{cp}^{\rightarrow}(Q_b, Q_a)$$

where $E_{cp}^{\rightarrow}(Q_a, Q_b) = \sum_{s=1}^{s_{max}} ((q_a - q_b) \cdot \vec{n_b})^2$ measures the deviation of two planar structures from coplanarity. For feature-to-proxy relationships, q_a and q_b are positions of inlier features. For proxy-to-proxy constraints, each q_s is either sampled from the boundary of a 0.5 meter radius disk around p_a, or is at the same location as p_a (in our experiments $s_{max} >= 5$).

The error in geometric relationships $E_G(G_i)$ between proxies $P_i[j]$ and $P_i[k]$ is:

$$E_G(G_i) = \sum_{j=1}^{|G|} \begin{cases} w_{jk}(\vec{n}_j - \vec{n}_k)^2 & parallel \\ w_{jk}(\vec{n}_j + \vec{n}_k)^2 & antiparallel \\ w_{jk}(\vec{n}_j \cdot \vec{n}_k)^2 & orthogonal \end{cases}$$

Feature Correspondence Error. E_C is designed to encourage alignment of detected correspondences between transformed features $F_i[s][a]$, $F_i[r][b]$:

$$E_C(T_i, C_i) = \sum_{j=1}^{|C_i|} \begin{cases} ((p'_b - p'_a) \times \vec{d'_a})^2 & edges \\ ((p'_b - p'_a) \cdot \vec{n'_a})^2 & planes \end{cases}$$

where p'_a, n'_a, d'_a and p'_b denote feature attributes transformed using respective transformations $T_i[s], T_i[r]$.

Local Alignment Error. E_L is designed to encourage pairwise transformations between adjacent frames to match the ones computed during preprocessing:

$$E_L(T_i) = \sum_{j=0}^{n-1} \sum_{k=0}^{k_{max}} E_t(T_0[j+2^k]^{-1}(T_0[j]), T_i[j+2^k]^{-1}(T_i[j]))$$

where $k_{max} = 16$ and E_t measures the misalignment of transformation $T[j]$ to another $T[k]$. We compute E_t by summing the squared distances between points p_s ($s \in [1, 8]$) sampled uniformly on a 1 meter radius sphere when they are transformed by $T[j]$ versus $T[k]$:

$$E_t(T[j], T[k]) = \sum_{s=1}^{s_{max}} (T[j](p_s) - T[k](p_s))^2.$$

Inertia Error. E_I is added to provide stability for the optimization and prevent the system of equations from being under-constrained. Here we denote transformation of proxy $P_i[j]$ as $T^{P_i}[j]$.

$$E_I(T_i, P_i) = \sum_{j=1}^{|I|} (\Delta T_i[j])^2 + \sum_{j=1}^{|P_i|} (\Gamma T^{P_i}[j])^2$$

ΔA represents the sum of squared differences between Euler angle rotations and translations for A from one iteration to the next. ΓA is identical to ΔA when the previous transformation is an identity.

5. Experimental Results

We performed a series of experiments designed to test the performance of the proposed method with comparisons to previous methods and ablation studies.

New Benchmark Dataset. RGB-D scans of indoor scenes with ground truth alignments are scarce. Most contain only part of a room [1, 9, 18, 23, 26, 36, 38], have less than ten test examples [9, 26, 47], or are based on synthetic data [18, 17]. As a result, the research community has compared registration results on small, clean datasets, not representative of the large real-world scans required for most applications.

To address this issue, we introduce a new registration benchmark based on the SUN3D dataset [47]. SUN3D contains a large set RGB-D videos captured with a ASUS Xtion PRO LIVE sensor attached to a hand-held laptop in a variety of spaces (apartments, hotel rooms, classrooms, etc.). Each scan contains

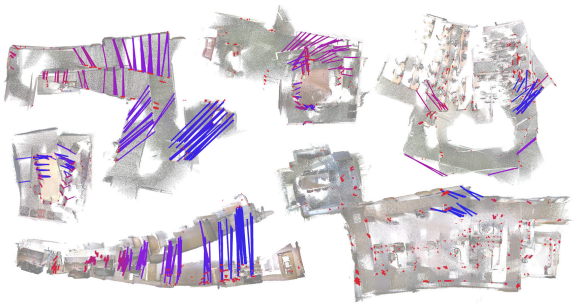

Figure 4: Ground truth correspondences for 6 out of 25 scenes in our benchmark. The visualization shows lines between manually-clicked corresponding points after alignment with T_0, the initialization for our method. Color indicates the frame distance - blue denotes loop closure pairs, while red denotes local pairs.

$10^3 - 10^4$ images, often covering multiple rooms. Previously, only eight of the scenes were released with full annotations and pose correction. Because of the lack of ground truth poses, these have not been used for quantitative evaluation of registration algorithms.

One of our contributions is to provide ground-truth point correspondences for 25 of the largest scenes in SUN3D. In all, we have manually clicked on 10,401 point correspondences with pixel-level accuracy. These ground-truth correspondences are largely in pairs of overlapping frames forming loop closures, but they also appear in pairs of nearby frames spread evenly throughout the sequence, as shown in Figure 4. The average number of correspondences per scan is 416, with a minimum of 239 and a maximum of 714.

We use these ground truth correspondences to evaluate and compare RGB-D registration algorithms by computing their root mean squared error (RMSE). To quantify a lower bound on the RMSE in this test, we have aligned the ground truth correspondences for all scenes with no other constraints and report the errors in the left column of Table 1. Note that these lower-bounds are non-zero, even though clicked correspondences are pixel-accurate. This error is due to the extreme noise in the uncalibrated SUN3D depth maps.

Comparisons to Previous Work. We evaluate our method in comparison to two prior methods for offline registration: Xiao et al.'s Sun3DSfm[48] and Choi et al.'s Robust Reconstruction of Indoor Scenes [8] (Figure 5). The first method by Xiao et al. uses the similar method for tracking, but also predicts loop closures via visual place recognition with a BoW approach and performs a global bundle adjustment to optimize for camera poses. The second method by Choi et al. fuses consecutive groups of 50 frames into fragments, aligns all pairs of fragments with a variant of RANSAC, selects pairs as potential loop closures, and then solves

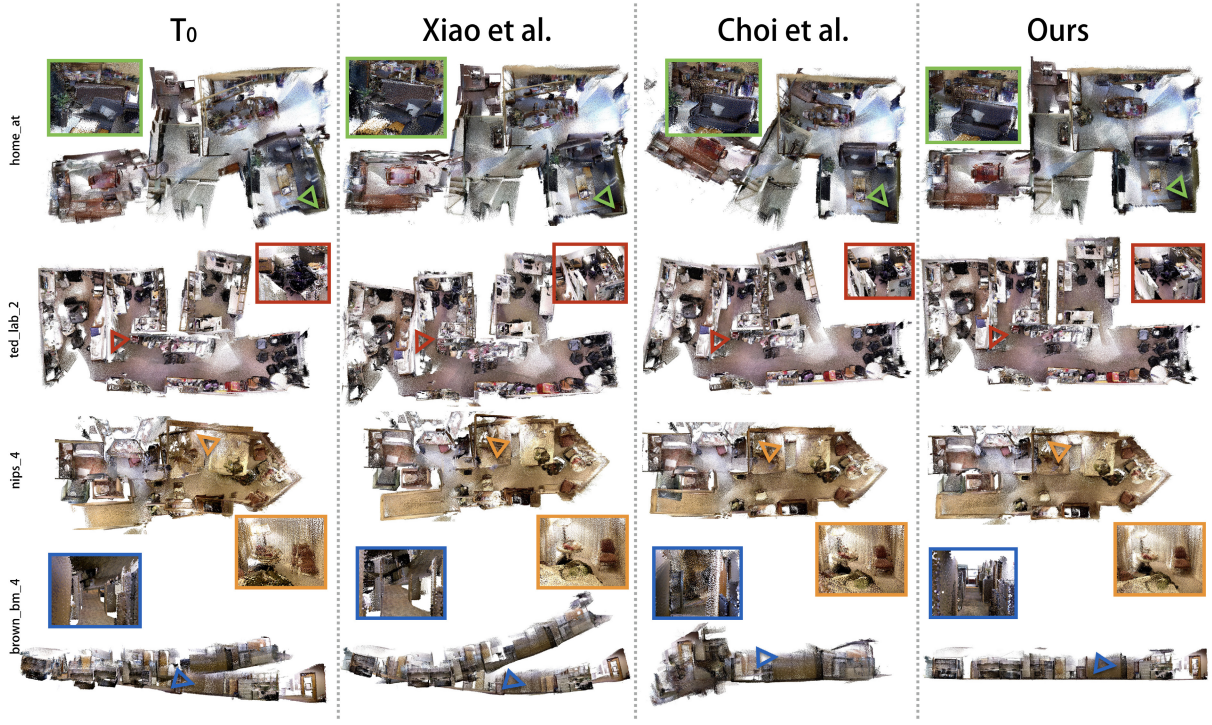

Figure 5: Qualitative comparison of global registration results for example SUN3D scenes. The rightmost column shows our results. The leftmost column shows the solution used to initialize our algorithm (T_0). The middle two columns show results produced with prior work [8, 47]. In insets, we show close-ups of particular regions. In the first two rows, our method is able to recover correct arrangement of captured multi-room environments, while previous work produces improbable structures, like intersecting rooms. The third row shows a sequence with non-Manhattan walls, which we are able to register correctly. Our method is also able to correctly align a challenging corridor sequence in the fourth row, where for Xiao et al., the visual place recognition has failed. Due to a lot of geometrical self similarities, Choi et al. is unable to recover proper geometry.

	Ground Truth	Ours	T_0	Xiao et al.	Choi et al.
Average	0.031	0.073	0.519	0.425	0.999
Standard Deviation	0.006	0.023	0.394	0.493	1.464
Median	0.031	0.065	0.410	0.214	0.247
Minimum	0.019	0.040	0.118	0.078	0.047
Maximum	0.045	0.139	1.560	2.001	5.901

Table 1: Comparison of RMSE statistics in meters with different registration methods for the 25 scenes in our SUN3D benchmark.

a least squares system of nonlinear equations that simultaneously solves for camera poses and loop closure weights. We believe this second method is the state-of-the-art for off-line global registration amongst ones with code available, even though it only uses the depth information. Comparisons are provided in the supplemental materials for several real-time reconstruction methods, which demonstrate worse performance than these off-line global methods.

Table 1 and Figure 6 show quantitative results for the comparison evaluated on our new SUN3D benchmark. Table 1 compares overall statistics of RMSEs for each algorithm, while Figure 6 shows the distributions of RMSEs. It can be seen in both of these results that our reconstruction algorithm aligns the ground truth correspondences better than either of the other

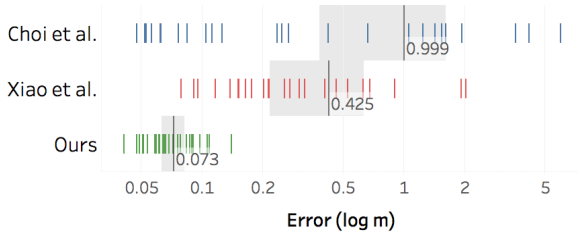

Figure 6: Quantitative comparison. Every vertical bar in each row represents the RMSE achieved for one of the 25 SUN3D scenes with the algorithm listed on the left. The vertical gray bar shows the average RMSE for each method, and the shaded gray regions represents one standard deviation.

two methods: our median error is 0.065m in comparison to 0.214m for Xiao et al. and 0.247m for Choi et al. In case-by-case comparisons, our method has the lowest error in 21 of 25 scenes.

Investigating Fine-to-Coarse Iteration. To investigate the behavior of our fine-to-coarse algorithm, we computed histograms of L_2 distances versus frame index differences between pairs of frames linked by ground-truth correspondences. Figure 7 shows a comparison of these histograms for the registrations at the

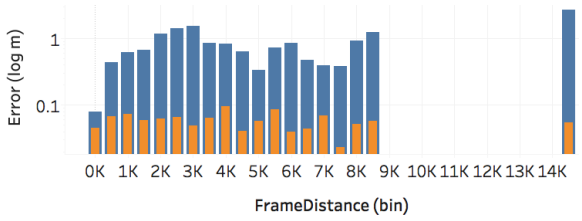

Figure 7: Investigating fine-to-coarse iteration. Each bin gathers correspondences that are specific numbers of frames away from each other in the RGB-D video. Blue bars show the correspondence errors using initial pairwise transformations (T_0), while orange bars show errors after applying our method (on a log scale). Note that errors decrease for both long-range loop closures and nearby frames.

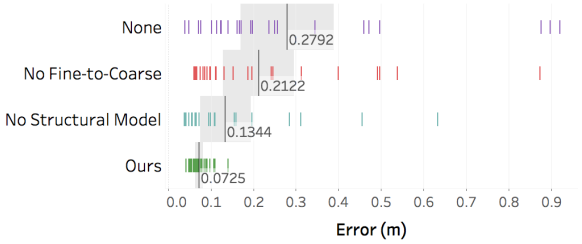

Figure 8: Ablation studies. Distributions of errors in the SUN3D benchmark for alternatives of our algorithm. Disabling coarse-to-fine iteration or structural modeling diminishes performance.

start of our algorithm (blue) and at the end (orange). It is interesting to note that our algorithm not only reduces the distances between ground-truth correspondences forming long-range loop closures (the right side of the plot), but also over short ranges. This result demonstrates that the extracted structural model helps to fix not only global alignments, but also local ones.

Ablation Studies. To investigate the value of our proposed a) fine-to-coarse iteration strategy and b) structural model, we performed comparisons of our method with all combinations of these methods enabled or disabled. The results in Figure 8 and 9 show that both provide critical improvements to the results. In particular, it is interesting to note that both the structural model and fine-to-coarse iteration strategy improve over the basic refinement. However, we are able to achieve significantly better results only when both are used. This result highlights the value of aligning local structures before searching for constraints at larger scales.

Failure Cases. Our method does not always succeed. For example, it can fail when rooms are nearly (but not exactly) rectangular (Figure 10). Failures of this type are rare – since the weight of enforcing parallelism and orthogonality constraints is low for pairs of planes at off-angles, we are able to reconstruct most scenes with non-Manhattan geometry correctly (as in the third row of Figure 6).

Timing. Our tests were run on a machine with 3.0GHz

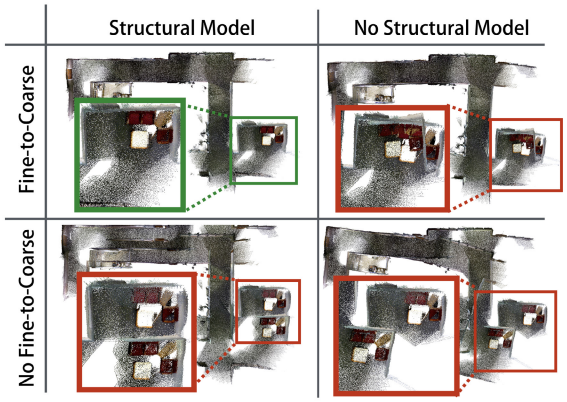

Figure 9: Qualitative examples of our ablation studies. Only our full method, using both fine-to-coarse strategy and structural model is able to align the region with red chairs correctly (see zoom-in)

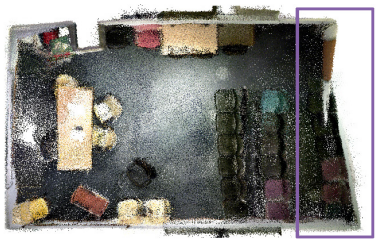

Figure 10: Failure case. In this trapezoidal room, our structural model incorrectly enforces walls on longer side of the room to be parallel, leading to intersections along shorter side (outlined).

CPU and 128Gb of RAM. Registering the shortest sequence of 875 frames took 179 seconds, while the longest with 13,401 frames took 8,147 seconds.

6. Conclusions

This paper describes a method for global registration of RGB-D scans captured with a hand-held camera in a typical indoor environment. The key idea is a fine-to-coarse scheme that detects and enforces constraints (geometric relationships and feature correspondences) within windows of gradually increasing scales in an iterative algorithm. The benefits of the proposed approach are demonstrated in experiments with a new benchmark for RGB-D registration, which contains 10,401 manually specified correspondences across 25 SUN3D scenes. This benchmark and all code are publicly available at http://scanregistration.cs.princeton.edu.

Acknowledgments

This work is supported by Intel, NVIDIA, Adobe, Pixar, and NSF (IIS-1251217 and VEC 1539014/ 1539099). It uses data provided by SUN3D [47] and code provided by Xiao et al. [47] and Choi et al [8].

References

[1] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena. Contextually guided semantic labeling and search for 3d point clouds. *IJRR*, 2012.

[2] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *Robotics, IEEE Transactions on*, 24(5):1027–1037, 2008.

[3] A. Bartoli and P. Sturm. Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *International Journal of Computer Vision*, 52(1):45–64, 2003.

[4] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. PAMI*, 14(2):239–256, 1992.

[5] B. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3D scans. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), Aug. 2007.

[6] J. Chen, D. Bautembach, and S. Izadi. Scalable real-time volumetric surface reconstruction. *ACM Trans. Graph.*, 32(4):113:1–113:16, July 2013.

[7] Z. Chen, O. Lam, A. Jacobson, and M. Milford. Convolutional neural network-based place recognition. *arXiv preprint arXiv:1411.1509*, 2014.

[8] S. Choi, Q.-Y. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *arXiv preprint arXiv:1604.01093*, 2016.

[10] M. Dou, L. Guan, J.-M. Frahm, and H. Fuchs. Exploring high-level plane primitives for indoor 3d reconstruction with a hand-held rgb-d camera. In *Asian Conference on Computer Vision*, pages 94–108, 2012.

[11] M. Dzitsiuk, J. Sturm, R. Maier, L. Ma, and D. Cremers. De-noising, stabilizing and completing 3d reconstructions on-the-go using plane priors. *CoRR*, abs/1609.08267, 2016.

[12] H. E. Elghor, D. Roussel, F. Ababsa, and E. H. Bouyakhf. Planes detection for robust localization and mapping in rgb-d slam systems. In *3D Vision (3DV), 2015 International Conference on*, pages 452–459, Oct 2015.

[13] C. Estrada, J. Neira, and J. Tardos. Hierarchical slam: Real-time accurate mapping of large environments. *Transactions on Robotics*, 21(4):588596, 2005.

[14] J. X. Fisher Yu and T. Funkhouser. Semantic alignment of lidar data at city scale. In *28th IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[15] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):112, 2005.

[16] G. Grisetti, R. Kmmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[17] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Scenenet: Understanding real world indoor scenes with synthetic data. In *IEEE CVPR*, 2016.

[18] A. Handa, T. Whelan, J. McDonald, and A. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.

[19] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *International Symposium on Experimental Robotics (ISER)*, 2010.

[20] O. Kahler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. H. S. Torr, and D. W. Murray. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. *IEEE Transactions on Visualization and Computer Graphics (Proceedings International Symposium on Mixed and Augmented Reality 2015*, 22(11), 2015.

[21] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision – 3DV*, pages 1 – 8, DOI:10.1109/3DV.2013.9, June 2013.

[22] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao. Chisel: Real time large scale 3d reconstruction onboard a mobile device. In *Robotics Science and Systems 2015*, July 2015.

[23] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3d scene labeling. In *IEEE International Conference on Robotics and Automation*, page 30503057, 2014.

[24] Y. Li, X. Wu, Y. Chrysanthou, A. Sharf, D. Cohen-Or, and N. J. Mitra. Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics*, 30(4):52:1–52:12, 2011.

[25] L. Ma, C. Kerl, J. Stueckler, and D. Cremers. Cpa-slam: Consistent plane-model alignment for direct rgb-d slam. In *Int. Conf. on Robotics and Automation*, 2016.

[26] O. Mattausch, D. Panozzo, C. Mura, O. Sorkine-Hornung, and R. Pajarola. Object detection and classification from large-scale cluttered indoor scans. *Computer Graphics Forum*, 33(2):1121, 2014.

[27] A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra. Rapter: Rebuilding man-made scenes with regular arrangements of planes. *ACM Trans. Graph.*, 34(4):103:1–103:12, July 2015.

[28] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and . A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[29] V. Nguyen, A. Harati, and R. Siegwart. A lightweight slam algorithm using orthogonal planes for indoor mobile robotics. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 658–663. IEEE, 2007.

[30] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013.

[31] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga. Fast registration based on noisy planes with unknown correspondences for 3-D mapping. *IEEE Trans. Robotics*, 26(3):424441, June.

[32] K. Pulli. Multiview registration for large data sets. In *Proceedings of the 2Nd International Conference on 3-D Digital Imaging and Modeling*, 3DIM'99, pages 160–168, Washington, DC, USA, 1999. IEEE Computer Society.

[33] A. Ratter and C. Sammut. Local map based graph slam with hierarchical loop closure and optimisation. 2015.

[34] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 21(3):438–446, July 2002.

[35] R. Salas-Moreno, B. Glocken, P. Kelly, and A. Davison. Dense planar slam. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 157–164, Sept 2014.

[36] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. European Conf. on Comp. Vision*, 2012.

[37] P. Stotko. State of the art in real-time registration of rgb-d images. In *CESCG*, 2016.

[38] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[39] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng. Point-plane slam for hand-held 3d sensors. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5182–5189, May 2013.

[40] Y. Tang and J. Feng. Hierarchical multiview rigid registration. *Comput. Graph. Forum*, 34(5):77–87, Aug. 2015.

[41] A. Trevor, J. Rogers III, and H. Christensen. Planar surface SLAM with 3D and 2D sensors. In *ICRA*, 2012.

[42] H. Wang, J. Wang, and L. Wang. Online reconstruction of indoor scenes from rgb-d streams. In *IEEE CVPR*, 2016.

[43] J. Weingarten and R. Siegwart. 3D SLAM using planar segments. In *IROS*, page 30623067, 2006.

[44] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald. Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, Jul 2012.

[45] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. Leonard, and J. McDonald. Real-time large scale dense RGB-D SLAM with volumetric fusion. *Intl. J. of Robotics Research, IJRR*, 2014.

[46] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. ElasticFusion: Dense SLAM without a pose graph. In *Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.

[47] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. *Computer Vision, IEEE International Conference on*, 0:1625–1632, 2013.

[48] J. Xiao, A. Owens, and A. Torralba. SUN3D database: Semantic RGB-D bundle adjustment with human in the loop. In *Proc. Int. Conf. on Comp. Vision*, 2013.

[49] Y. Zhang, W. Xu, Y. Tong, and K. Zhou. Online structure analysis for real-time indoor scene reconstruction. *ACM Transactions on Graphics (TOG)*, 34(5):159, 2015.

[50] Q.-Y. Zhou and V. Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics*, 32(4), 2013.

[51] Q.-Y. Zhou and V. Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *SIGGRAPH Conf. Proc.*, 2014.

[52] Q.-Y. Zhou and V. Koltun. Depth camera tracking with contour cues. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.