Dense Scene Reconstruction with Points of Interest

Qian-Yi Zhou Vladlen Koltun
Stanford University

Figure 1: *Rodin's "The Burghers of Calais," reconstructed from a stream of images produced by a handheld commodity range camera. The statues are 2 meters tall.*

Abstract

We present an approach to detailed reconstruction of complex real-world scenes with a handheld commodity range sensor. The user moves the sensor freely through the environment and images the scene. An offline registration and integration pipeline produces a detailed scene model. To deal with the complex sensor trajectories required to produce detailed reconstructions with a consumer-grade sensor, our pipeline detects points of interest in the scene and preserves detailed geometry around them while a global optimization distributes residual registration errors through the environment. Our results demonstrate that detailed reconstructions of complex scenes can be obtained with a consumer-grade camera.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

Keywords: scene reconstruction, range imaging

Links: DL PDF

1 Introduction

Acquisition of high-quality digital representations of real-world scenes is one of the key research goals in computer graphics. The ability to easily create detailed three-dimensional models of physical environments can accelerate the production of computer games and special effects, support retail and travel, and provide valuable data for training computer vision systems. Consumer-grade range cameras are a promising source of input for the creation of such three-dimensional models. These cameras stream range images at high frame rates [Microsoft 2010]. They are easily portable, low cost, and widely available.

There are two difficulties in using range videos streamed by these cameras to acquire detailed scene models that can be used for computer graphics applications. The first is the fidelity of the data. Consumer-grade range sensors have a narrow field of view and errors of 2-3 centimeters at typical operating ranges [Khoshelham and Elberink 2012]. The second difficulty is the complexity of the camera trajectory that is necessary for a detailed reconstruction. In a complex scene, the user must move the sensor along a trajectory that weaves around objects to image them from many points of view. Sufficient data must be acquired to minimize disocclusion gaps and to redundantly image detailed objects so as to average out errors in individual frames. In practice, medium-scale scenes require minutes of input data in order to satisfactorily cover the surfaces of all objects in the scene. For example, the scene in Figure 1, which spans an area of 50 square meters, was reconstructed from a 6 minute long input stream that contains over 11 thousand frames.

Continuous sensor trajectories are key to counteracting the imaging errors, because they enable registering incoming frames to a growing local model of the scene, which stabilizes the estimation of camera pose and averages out input noise [Newcombe et al. 2011]. Yet detailed acquisition of complex scenes leads to long camera paths with complex spatial structure. This necessitates the use of global optimization to deal with the accumulated registration errors [Henry et al. 2012]. Unfortunately, such global optimization distributes the residual error throughout the path and can corrupt the detailed surface shape of objects in the scene.

In this paper, we present an approach to detailed reconstruction of complex scenes with handheld commodity range sensors. The

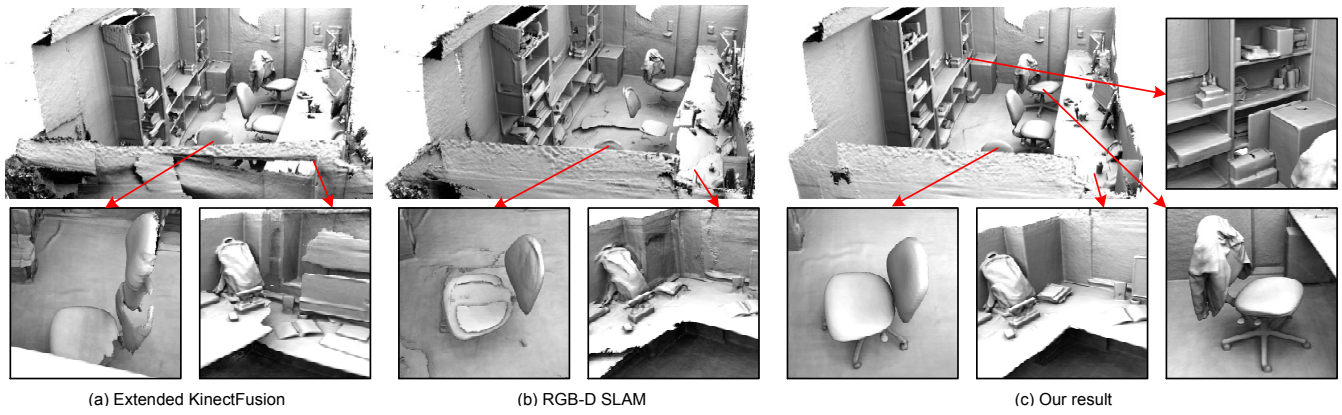


Figure 2: *Reconstruction of an indoor environment that was imaged in detail. (a) Extended KinectFusion accumulates registration drift over the long and spatially extended camera trajectory and does not produce a globally consistent reconstruction. (b) A state of the art RGB-D SLAM system produces a globally consistent reconstruction, but does not preserve local geometric detail. (c) Our approach combines frame-to-model registration with global optimization and protects densely scanned parts of the scene.*

key idea is to combine frame-to-model registration with an offline optimization framework that handles loop closures and produces a globally consistent reconstruction. We build locally fused models for overlapping parts of the scene and use them to initialize a global graph-based optimization that distributes residual error. To protect detailed object shapes, we detect densely scanned points of interest in the scene and preserve geometric detail in the surrounding regions during the optimization.

We demonstrate that our approach produces globally consistent and locally detailed reconstructions of indoor and outdoor scenes despite the limited fidelity of the input data.

2 Related Work

Key early work on object reconstruction from range images was conducted by Chen and Medioni [1992] and Turk and Levoy [1994], who identified two stages in the reconstruction process: registration, which brings the range images into alignment, and integration, which uses the aligned images to compute a single surface representation. Both works used variants of the ICP algorithm for registration, initialized by manual alignment. Turk and Levoy described an integration scheme that merged triangle meshes created from individual range images, producing a unified mesh for the reconstructed object. Curless and Levoy [1996] further focused on the integration stage and developed a volumetric technique that fuses range images into a voxel grid. The grid stores a signed distance function (SDF) that represents a linear combination of distances to range measurements. This technique has been widely adopted due to its robustness and generality. (See [Fuhrmann and Goesele 2011] for a recent extension.)

Rusinkiewicz et al. [2002] pioneered the real-time acquisition of object shapes by range imaging. In their system, the user slowly rotates an object in front of a stationary structured light scanner that produces range images at a high frame rate. Since the system acquires many images per second, consecutive frames can be registered to each other without initial manual alignment. Further, a preview of the reconstructed shape can be shown to the user, to help identify areas that have not been adequately imaged.

These ideas were applied to shape acquisition with a handheld range sensor in the KinectFusion system [Newcombe et al. 2011]. The user moves the camera through a scene, which is represented by an SDF over a voxel grid that is maintained by graphics hardware. Each new range frame is registered to the SDF and fused into it. One of the key insights of the work is that frame-to-model registration, which aligns each incoming range image to a growing volumetric representation of the reconstructed scene, is significantly more robust than frame-to-frame registration, which aligns consecutive frames to each other. With frame-to-model registration, each registration step takes advantage of densely reconstructed scene geometry, aggregated and refined over many frames. This enables highly accurate reconstruction of individual objects and small regions (up to a few cubic meters in the original paper).

There are a number of limitations that prevent the basic KinectFusion pipeline from successfully producing detailed reconstructions of complex environments on a larger scale. The first, which has been studied extensively in follow-up work, is that a uniform voxel grid is memory-intensive and quickly exceeds the capacity of graphics hardware. This can be addressed by using a hierarchical spatial subdivision [Zeng et al. 2013] or more generally by sliding the volume through the scene to follow the camera, paging parts of the scene in and out as needed [Roth and Vona 2012; Heredia and Favier 2012; Whelan et al. 2013]. The deeper limitation is that while frame-to-model registration is substantially more accurate than frame-to-frame registration, it is not infallible, particularly given the error magnitudes in consumer-grade range cameras [Khoshelham and Elberink 2012]. Registration errors accumulate over long acquisition trajectories and can break the reconstruction. This is demonstrated in Figure 2(a), which shows the reconstruction produced by the moving-volume approach for an indoor scene that was imaged in detail. (We will refer to this approach as Extended KinectFusion.) The underlying issue is the lack of global reasoning on the camera trajectory and the scene, and specifically the lack of loop closure handling.

Global optimization has been widely employed in reconstruction from sets of range images [Pulli 1999; Huber and Hebert 2003; Brown and Rusinkiewicz 2007] and in structure from motion estimation [Triggs et al. 2000; Agarwal et al. 2010; Wu et al. 2011]. Fewer works address the challenges of detailed reconstruction from streams of highly inaccurate images produced by consumer-grade range cameras. Weise et al. [2011] describe a loop closure handling approach designed for acquisition of individual object shapes: the object is deformed as-rigidly-as-possible to accommodate detected loop closures. Cui et al. [2010] describe an object scanning pipeline with a noisy handheld camera, but do not address loop closure and other forms of global reasoning. Such issues have been considered

in depth in the context of simultaneous localization and mapping (SLAM) in robotics [Williams et al. 2009]. Two recent techniques apply global optimization to produce dense maps of indoor scenes from streams of RGB-D data [Henry et al. 2012; Endres et al. 2012]. In these systems, loop closures are detected by matching features extracted from range and color images; a graph is constructed that connects all pairs of consecutive frames and closes loops; each pair of frames connected by the graph is registered frame-to-frame; and finally an optimization is performed on the graph to globally distribute the error accumulated in all the pairwise alignments. This approach can produce dense maps for large environments, but is not designed to handle the complex trajectories we encounter when scanning scenes in detail and does not preserve detailed geometry for objects throughout the environment (Figure 2(b)). An alternative approach to dense surface mapping from range video was described by Ruhnke et al. [2012].

Scene reconstruction from collections of photographs has been studied extensively in multi-view stereo reconstruction [Seitz et al. 2006; Furukawa and Ponce 2010; Goesele et al. 2007; Furukawa et al. 2010]. Our problem is quite different in that we have a stream of range images acquired along a continuous trajectory. Pollefeys et al. [2004; 2008] reconstructed large-scale 3D scenes from video streams. In our work, the availability of range data enables detailed reconstruction of complex scenes even when reliable photometric cues are not available.

3 Overview

Our approach begins by estimating a rough camera trajectory using an RGB-D SLAM system [Endres et al. 2012]. This provides a dense map of the environment that is used as input for detecting points of interest (POI). Figure 3 illustrates the process. We perform density estimation in the map to detect parts of the environment that were scanned particularly thoroughly. Since detailed objects in the scene require prolonged imaging from many points of view to capture their detailed geometry and eliminate disocclusion gaps, we assume that pronounced density peaks correlate with visual saliency and importance. Other algorithms for estimating visual importance can be used; we found the approach described in the paper to perform well.

After we identify POI in the scene, we partition the camera trajectory into segments associated with individual points of interest as well as *connector* segments that primarily target other parts of the scene. This is done by optimizing a multi-labeled Markov random field, in which the data term estimates the degree to which an individual frame targets a point of interest. The result of the optimization is a segmentation of the trajectory.

This segmentation is used to construct locally fused models for parts of the scene. Our approach creates a large number of overlapping local models, where detailed geometry is reconstructed on a small scale, without artifacts caused by long-term accumulation of registration drift and residual error from resolution of long-term loop closures. We create a local volume around each POI and use all parts of the trajectory that are associated with this POI to create a local reconstruction. Connector segments are partitioned into overlapping fragments and locally fused reconstructions are created for each fragment.

The geometry obtained from these local models is not used directly in the final reconstruction. Instead, these models are used to obtain refined estimates for the camera pose. We use a two-pass frame-to-model registration procedure that produces considerably more stable and accurate estimates for relative camera pose. The first pass, in which local volumetric models for parts of the scene are constructed, is described above. In the second pass, we go over

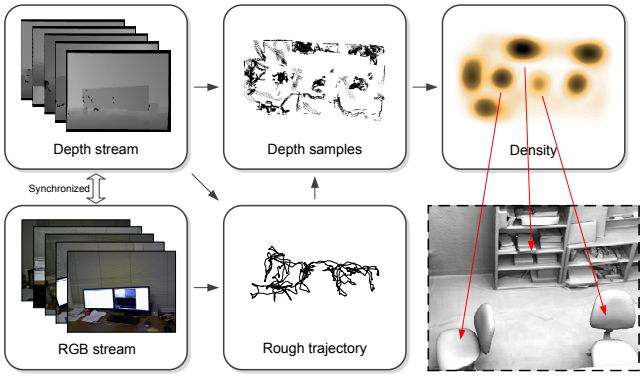


Figure 3: *Scene analysis. Points of interest are identified as strong modes in a density function induced by localized samples from the input range images.*

each trajectory segment again and register it to an already completed local model. We then take the transforms between pairs of frames along the trajectory and associate them with edges in a pose graph. These pairwise transforms are considerably more accurate than what is obtained by progressive frame-to-model registration, and this two-pass local registration is necessary for obtaining high-quality results on a large scale. Two-pass frame-to-model registration was used in Figures 2(a) and 2(b): without it these reconstructions would look considerably worse.

After refined pairwise registration estimates are obtained as described above, we optimize all camera poses along the trajectory globally, using least-squares optimization over a pose graph. The graph connects each consecutive pair of frames and includes edges that close loops. Relative pose estimates for pairs of POI frames are treated as hard constraints to preserve the detailed geometry around points of interest and relative pose estimates that involve connector frames are treated as soft constraints. Connector segments thus function as flexible buffers that absorb the residual error during global optimization of the pose graph.

Finally, we integrate all the range images within a large volume that encompasses the entire scene. We use a weighting scheme that further protects POI geometry.

4 Points of Interest

4.1 Scene Analysis

The point of interest detection pipeline is summarized in Figure 3. As described in Section 3, our pipeline begins by performing simultaneous localization and mapping on the input stream. We use the RGB-D SLAM system of Endres et al. [2012]. The system produces a localized camera trajectory. Let $\widetilde{\mathbf{T}}_k$ denote the rigid transform that maps the k-th range image D_k from its local coordinate frame to the global scene coordinate frame. We uniformly sample a set of points $\{\mathbf{p}_i^k\}$ from D_k, for all k, to obtain a set of localized points $P = \{\widetilde{\mathbf{T}}_k \mathbf{p}_i^k\}$ that roughly map out the distribution of input data over the scene. Our current implementation uses 2D density estimation to estimate the locations of points of interest. We do not see significant obstacles to estimating POI locations in 3D, but had no need for it. We detect the dominant plane in the scene, which is usually the ground plane, using RANSAC. Let $\mathbb{P}_g$ denote the projection operator onto this plane g. Consider the planar point set $\bar{P} = \{\bar{\mathbf{p}}_i^k\} = \{\mathbb{P}_g(\widetilde{\mathbf{T}}_k \mathbf{p}_i^k)\}$. We will identify POI locations by finding modes in a density function estimated from $\bar{P}$.

We weight the points to prioritize sample points based on their distance from the principal axis of the camera in their frame of origin, since error magnitudes increase with distance from the principal axis [Khoshelham and Elberink 2012] and users also tend to naturally target objects of interest. The weight associated with point $\mathbf{p}_i^k$ is

$$w_i^k = \tau \exp(-(\mathbf{d}_i^k)^2/2\sigma^2), \tag{1}$$

where $\mathbf{d}_i^k$ is the distance between $\mathbf{p}_i^k$ and the principal axis of the camera at frame k, and τ is a normalization constant.

We now estimate POI locations by finding modes in a density function ρ induced by the points $\bar{P}$, weighted as described above. We use mean shift [Comaniciu and Meer 2002]. The density is defined as

$$\rho(\mathbf{x}) = \textstyle\sum_i w_i^k K(\|\mathbf{x} - \bar{\mathbf{p}}_i^k\|/h), \tag{2}$$

where K is the Epanechnikov kernel. We set the bandwidth h empirically to 0.5m. We merge modes when their distance is below h. The final set $\{\mathbf{s}_j\}$ of identified POI is denoted by S.

4.2 Trajectory Segmentation

After POI locations have been determined, we partition the camera trajectory into POI segments and connectors. We formulate this as a labeling problem over the set of frames. For each depth image D_k, the goal is to either associate it with a point of interest $\mathbf{s}_j$, which corresponds to assigning it the corresponding label ($D_k \rightsquigarrow \mathbf{s}_j$) or to make it a connector frame, which corresponds to assigning it a special label $\mathbf{c}$ ($D_k \rightsquigarrow \mathbf{c}$). Since the frames form a path, the labeling problem can be naturally expressed in terms of a pairwise multi-label Markov random field (MRF). We define the MRF energy as

$$E(L) = -\sum_k E_d(l_k; D_k) - \lambda \sum_k E_s(l_k, l_{k+1}), \tag{3}$$

where L is the labeling over all frames.

The data term $E_d(l_k; D_k)$ expresses the likelihood that the depth image D_k points to a particular POI or doesn't specifically point to any POI and can be labeled a connector. Specifically, for depth image D_k and POI $\mathbf{s}$ we define the data term in terms of proximities of samples from D_k to $\mathbf{s}$:

$$E_d(\mathbf{s}; D_k) = \textstyle\sum_i w_i^k \exp(\|\bar{\mathbf{p}}_i^k - \mathbf{s}\|^2/2\delta^2), \tag{4}$$

where w_i^k is the weight defined in (1) and δ controls the range of influence of points in the scene and is set to $\delta = h/2$. The data term for $l_k = \mathbf{c}$ is less direct, since there is no direct visual evidence for a depth image being a connector. We set the energy of $D_k \rightsquigarrow \mathbf{c}$ to be inversely correlated with the confidence of the POI labels:

$$E_d(\mathbf{c}; D_k) = \xi - \max_j E_d(\mathbf{s}_j; D_k), \tag{5}$$

where $\xi = 0.4$ in our implementation.

The smoothness term is defined as follows:

$$E_s(l_k, l_{k+1}) = -\begin{cases} 0 & \text{if } l_k = l_{k+1} \\ \infty & \text{if } l_k = \mathbf{s}_u \text{ and } l_{k+1} = \mathbf{s}_v \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

This ensures that POI segments are buffered by connectors that can absorb residual registration errors during the global optimization. We optimize $E(L)$ using graph cuts [Boykov et al. 2001]. Let $\{\mathcal{M}_t\}$ be the set of contiguously-labeled segments along the trajectory. The label associated with a segment $\mathcal{M}_t$ is denoted by $L(\mathcal{M}_t) \in S \cup \{\mathbf{c}\}$.

5 Scene Reconstruction

5.1 Two-Pass Registration

We use a two-pass procedure for estimating the relative poses of pairs of frames. We first describe the procedure for POI segments. The handling of connectors is described subsequently.

For each POI $\mathbf{s}_j$, we create a local volumetric model centered at $\mathbf{s}_j$. The local model is initialized as an empty signed distance function over a uniform voxel grid [Curless and Levoy 1996]. Let $\mathcal{Q}_j = \{\mathcal{M}_t | L(\mathcal{M}_t) = \mathbf{s}_j\}$ be the set of POI segments associated with $\mathbf{s}_j$. We perform two passes over $\mathcal{Q}_j$. The first pass performs registration and integration for each segment as described by Newcombe et al. [2011]. This creates a fused model (specifically, an SDF) for the local geometry around $\mathbf{s}_j$. In the second pass, we go over each frame from each segment in $\mathcal{Q}_j$ again and register it to the SDF computed in the first pass. We do not integrate the frame: we only use the already complete SDF to produce a stable estimate for the camera pose for each frame. In both passes, registration for each frame is initialized with the pose of the preceding frame [Rusinkiewicz et al. 2002; Newcombe et al. 2011], with one exception: for the first frame in each segment in $\mathcal{Q}_j$, registration is initialized by the pose estimate produced by the SLAM system during the initial mapping pass. The second pass results in a refined estimate $\mathbf{T}_u^j$ for the pose of any frame D_u from any segment in $\mathcal{Q}_j$. For the graph-based global optimization described in Section 5.2, we will need estimates for the relative transforms between pairs of frames associated with $\mathbf{s}_j$. For a pair of frames D_u and D_v, this estimate is set to $(\mathbf{T}_v^j)^{-1}\mathbf{T}_u^j$.

We now describe the two-pass registration procedure for connector segments. The handling of connectors is slightly different because they are not guaranteed to be spatially compact: a connector segment can be very long, both spatially and temporally. To produce stable pose estimates for connector frames, we cover each connector segment $\mathcal{M}_t$ by overlapping fragments of length κ. The precise value of κ is not particularly important, as long as it is large enough to allow for a fused local model to be created and not so large that substantial registration drift begins to accumulate. In our implementation, κ is set to 50. For each such fragment, we perform two-pass registration as described above. We also extend connector segments by a few frames on each end into the neighboring POI segments, to obtain stable estimates for the relative pose of the terminal frame on each end of the connector segment and the adjacent POI frame.

To further stabilize relative pose estimation, we cover $\mathcal{M}_t$ with fragments so that each pair of consecutive frames is covered by three distinct fragments. (Thus fragments begin at frames $n\kappa$, $(n + 1/3)\kappa$, and $(n + 2/3)\kappa$.) This yields three estimates for the relative pose of two consecutive frames. We use these to identify the most stable estimate as follows. Let the translational components of the three transforms be denoted by $\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{t}_3$. We consider their distances $\|\mathbf{t}_1 - \mathbf{t}_2\|$, $\|\mathbf{t}_2 - \mathbf{t}_3\|$, and $\|\mathbf{t}_3 - \mathbf{t}_1\|$, and discard the two estimates that span the longest distance.

Finally, we note that even highly stable frame-to-model registration can drift when dealing with featureless shapes, such as walls. Therefore, when the linear system for minimizing the ICP energy for a frame D_k is ill-conditioned, or when the camera translation between consecutive frames is unreasonably large (> 0.1m), we discard the ICP registration and use the initial pose estimate $\widetilde{\mathbf{T}}_k$ instead.

5.2 Global Optimization

After obtaining fine-grained relative pose estimates for pairs of consecutive frames, we need to obtain globally consistent pose estimates for all frames in the scene. Given the large error magnitudes in the input data, registration errors invariably accumulate over long trajectories. To minimize the aggregate error and distribute its residual, we use graph-based global optimization. To set up the pose graph G, we first connect every pair of consecutive frames by an edge. With each edge we associate a relative transform between the pair of frames, computed as described in Section 5.1. So far, the graph G is a simple path and does not reflect important loop closures in the trajectory. Next, we identify loop closures, compute refined relative pose estimates for pairs of frames that correspond to these loop closures, and add corresponding edges to G.

Our starting point is the initial localization $\widetilde{\mathbf{T}}_k$ of each frame D_k, introduced in Section 4.1. Consider the set $\{\mathcal{F}_i\}$ of trajectory fragments that include the POI segments ($\bigcup_j \mathcal{Q}_j$) and the κ-fragments that cover all connectors as described in Section 5.1. (The covering of the input trajectory by fragments is a refinement of the covering by segments, in which connector segments are further covered by fragments of length κ.) Let $\mathcal{F}_i$ be such a fragment. We use frame-to-model registration to fuse $\mathcal{F}_i$ and obtain a surface mesh. With a slight abuse of notation, we will use $\mathcal{F}_i$ to refer to both the trajectory fragment and the corresponding mesh.

An initial localization $\widetilde{\mathbf{T}}_i$ for $\mathcal{F}_i$ can be computed from the per-frame transforms $\widetilde{\mathbf{T}}_k$ for all $D_k \in \mathcal{F}_i$. To establish loop closure edges, we test each pair of fragments. Let $\mathcal{F}_i, \mathcal{F}_j$ be such a pair. We attempt to align $\mathcal{F}_i$ and $\mathcal{F}_j$ using ICP. If $\mathcal{F}_i$ and $\mathcal{F}_j$ are temporally overlapping, ICP is initialized using frame-to-model registration. (Specifically, consider a frame D_k that belongs to both $\mathcal{F}_i$ and $\mathcal{F}_j$. The frame-to-model registration for $\mathcal{F}_i$ and $\mathcal{F}_j$, mentioned above, yields a transform that localizes D_k within $\mathcal{F}_i$ and a transform that localizes D_k within $\mathcal{F}_j$. These transforms can be used to define an initial relative pose for $\mathcal{F}_i$ and $\mathcal{F}_j$.) If $\mathcal{F}_i$ and $\mathcal{F}_j$ are not temporally overlapping, the relative pose that is used to initialize ICP is computed using $\widetilde{\mathbf{T}}_i$ and $\widetilde{\mathbf{T}}_j$.

For some pairs of fragments, ICP will succeed in aligning large portions of the imaged surfaces. If after ICP alignment more than 20% of the points in one of the fragments are in close correspondence (distance <3cm) to the other fragment, we add a loop closure edge to G to connect this pair of fragments. Loop closure edges are established between *anchor* frames on the fragments. For a POI fragment, the anchor frame is simply the first frame of the fragment. For a connector fragment, the anchor frame is the frame that is temporally farthest from the closest POI frame. (For most fragments, this is either the first or the last frame.) Each loop closure edge is associated with the relative pose produced by ICP.

The pose graph G is optimized using a standard weighted nonlinear least squares formulation. Edges that connect frames associated with the same POI are treated as hard constraints (infinite weight). All other edges are treated as soft constraints (unit weight for edges that connect consecutive frames, weight 100 for loop closure edges). We use the $\mathbf{g^2o}$ package [Kummerle et al. 2011]. The optimization yields a rigid transform for each frame that maps it to the global scene coordinate frame.

5.3 Integration

Finally, after we obtain the definitive pose $\mathbf{T}_k$ for each frame k as described above, we perform global integration of all the range images. We initialize a signed distance function (SDF) $F_0(\mathbf{x})$ and a weight function $W_0(\mathbf{x})$ over a volume that encompasses the entire scene. For each frame k, we construct a projective SDF $f_k(\mathbf{x})$ for the localized range image $\mathbf{T}_k D_k$, with an associated weight function $w_k(\mathbf{x})$. The global distance and weight functions are updated as follows:

$$\begin{aligned} F_{k+1}(\mathbf{x}) &= \frac{W_k(\mathbf{x})F_k(\mathbf{x}) + w_{k+1}(\mathbf{x})f_{k+1}(\mathbf{x})}{W_k(\mathbf{x}) + w_{k+1}(\mathbf{x})} \\ W_{k+1}(\mathbf{x}) &= W_k(\mathbf{x}) + w_{k+1}(\mathbf{x}) \end{aligned}$$

where $F_k(\mathbf{x})$ and $W_k(\mathbf{x})$ are the cumulative distance and weight functions after integrating D_k [Curless and Levoy 1996; Newcombe et al. 2011].

During this integration process, we further protect the detailed geometry constructed around points of interest. Specifically, range images that are not associated with a point of interest $\mathbf{s}_j$ may still contain range samples from surfaces around $\mathbf{s}_j$. As shown in Figure 4, these samples can corrupt the local geometry around $\mathbf{s}_j$ since their frames were not associated with $\mathbf{s}_j$ and were not registered to the local model around $\mathbf{s}_j$. To protect such local geometry, we adopt the following weight function:

$$w_k(\mathbf{x}) = \begin{cases} \alpha + \beta K_G(\|\mathbb{P}_g(\mathbf{x}) - \bar{\mathbf{s}}_j\|/h) & \text{if } D_k \rightsquigarrow \mathbf{s}_j \\ 1 & \text{if } D_k \rightsquigarrow \mathbf{c} \end{cases}$$

where K_G is the Gaussian kernel, $\mathbb{P}_g$ is the projection operator introduced in Section 4.1, h is the bandwidth used in (2), and (α, β) are set to $(1, 10)$.

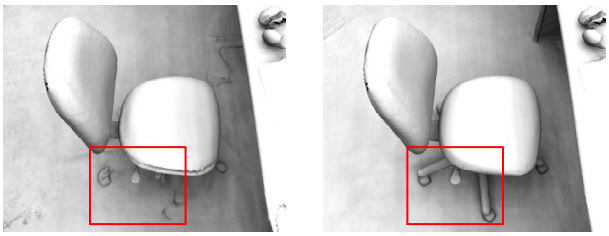


Figure 4: *We adopt a weighting function that protects local geometry around points of interest.*

6 Experiments

We use an Asus Xtion Pro Live camera, which streams VGA resolution range and color images at 30Hz. This camera uses the same PrimeSense range sensor as the Microsoft Kinect, but is somewhat smaller and lighter. It is rated for indoor use but can be comfortably operated outdoors, although not under strong sunlight. The camera is connected to a laptop that is carried in clamshell mode in a backpack. The operator holds the camera and a smartphone that is wirelessly connected to the laptop and runs a remote desktop application. The smartphone shows the color and depth input streams and a preview of the reconstruction. The preview is generated by Extended KinectFusion and helps to monitor the incoming data. When Extended KinectFusion loses track, the preview implementation resets the volume and initiates registration and integration from a clean slate.

Figure 5 illustrates our results for two indoor scenes. In order to overcome occlusion, the operator has to move the camera along complicated trajectories that induce nested and coupled loops. The trajectories are shown in the figure. Figures 1 and 6 illustrate our results for two outdoor scenes. Additional results are presented in the supplementary video. Table 1 summarizes the scenes used in our experiments. The table lists the length of the camera trajectory

Model	Size	# of frames	Trajectory length	# of POI	Data collection	RGB-D SLAM	POI detection	Registration	Optimization	Integration	Total time	Triangle count
Figure 1	$50m^2$	11,230	$184m$	6	6m	3h 24m	5m	40m	10m	2h 1m	6h 26m	6,858,620
Figure 2	$17m^2$	7,198	$77m$	7	4m	1h 57m	3m	18m	52m	53m	4h 7m	5,784,009
Figure 5 left	$13m^2$	3,000	$58m$	6	2m	24m	1m	12m	16m	40m	1h 35m	3,150,436
Figure 5 right	$14m^2$	5,490	$69m$	8	3m	41m	1m	14m	52m	47m	2h 38m	5,062,748
Figure 6	$26m^2$	6,152	$78m$	3	3m	1h 54m	2m	16m	6m	48m	3h 9m	3,752,678
Figure 7	$25m^2$	1,352	$16m$	2	1m	5m	$<$1m	7m	43m	25m	1h 22m	9,075,458
Figure 8 top	$12m^2$	2,703	$36m$	0	2m	18m	1m	17m	1h 54m	21m	2h 53m	4,187,775

Table 1: *Statistics for the scenes used in our experiments. Timings are reported for a workstation with an Intel i7 3.2GHz CPU, 24GB of RAM, and an NVIDIA GeForce GTX 690 graphics card.*

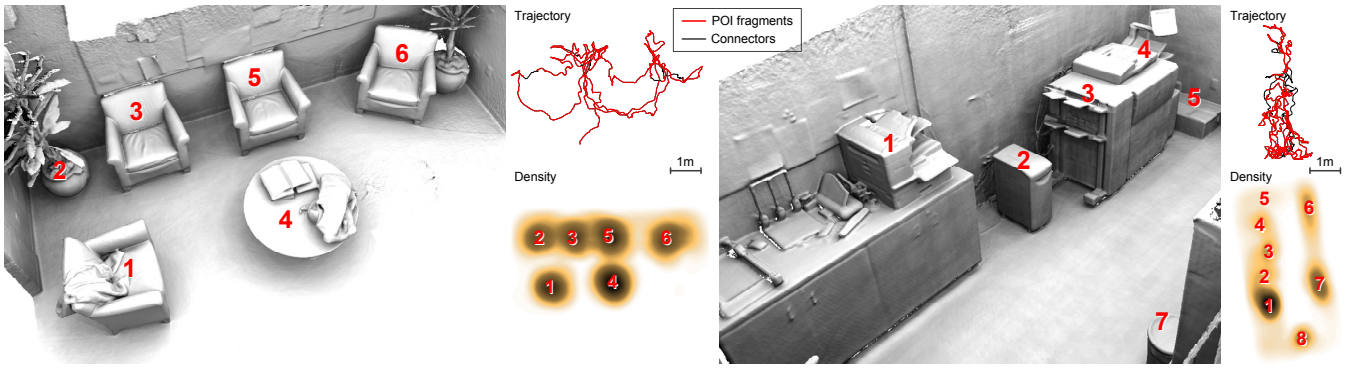


Figure 5: *Reconstruction of a lounge (left) and a copy room (right). For each scene, the figure shows the localized camera trajectory, a scale marker, the density map computed during scene analysis, and the identified points of interest.*

in each scene. This length is indicative of the complexity of the trajectories in some of the scenes. For the scene in Figure 1, for example, the camera traversed over 180 meters in a scene that is less than 10 meters in diameter.

Figure 2 compares our approach to the state of the art. We use the Extended KinectFusion implementation in the Point Cloud Library [Rusu and Cousins 2011; Heredia and Favier 2012]. For the reported experiments, we made the implementation more robust and augmented it with two-pass registration (Section 5.1). For RGB-D SLAM, we use the reference implementation of Endres et al. [2012]. All settings for RGB-D SLAM were set to maximize quality at the expense of computational efficiency. Note that Extended KinectFusion can run in real time, while RGB-D SLAM and our approach involve offline optimization and have substantially longer runtimes.

To evaluate our approach on independent benchmark data, we ran it on the five "fr1" handheld SLAM sequences from the RGB-D SLAM benchmark of Sturm et al. [2012]. The benchmark provides ground truth camera pose estimates obtained by a calibrated marker-based motion capture system. Different estimated trajectories can be evaluated against the ground truth by computing the absolute translational root mean square error (RMSE); see Sturm et al. [2012] for details. On two of the sequences, "fr1/360" and "fr1/floor", the KinectFusion frame-to-model registration (which is a component of our approach) fails due to large featureless surfaces and rapid camera movement. The results for the other three sequences are given in Table 2. Our approach produces more accurate camera trajectories than Extended KinectFusion or RGB-D SLAM for all three sequences.

For the longest and most complex sequence, "fr1/room", our approach has a RMSE of $0.09m$, a significant improvement over the $0.23m$ RMSE of Extended KinectFusion and the $0.22m$ RMSE of RGB-D SLAM. Figure 7 shows dense reconstructions of this scene obtained by our approach (left) and by using the ground truth

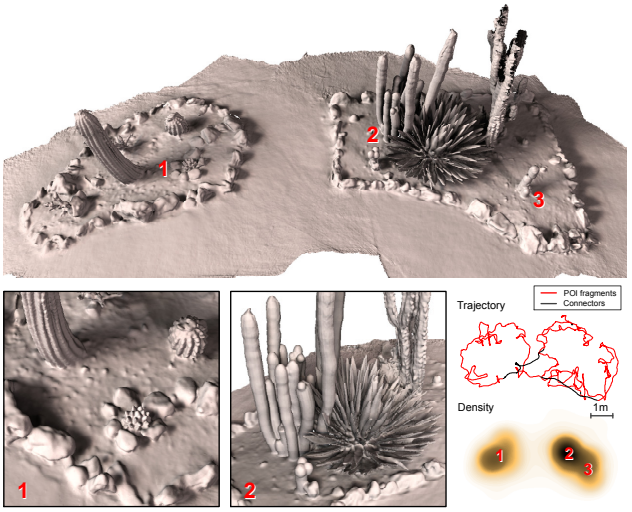


Figure 6: *Outdoor cactus garden.*

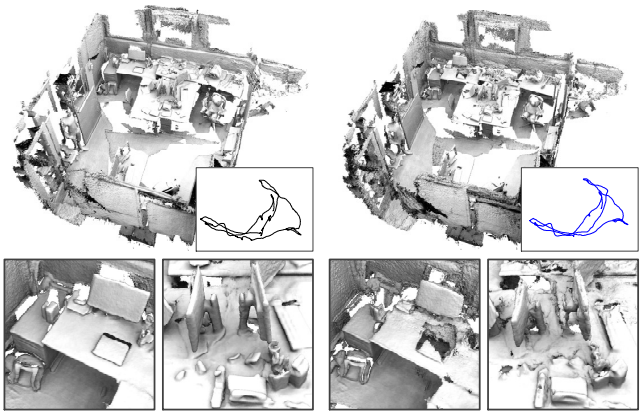
(motion capture) trajectory from the benchmark data to localize the range images, followed by standard volumetric integration (right). Our approach detects two POI in the scene and preserves geometric detail around them, as shown in the close-ups in Figure 7. Note that the ground truth camera pose estimates obtained by the motion capture system have errors of around 0.3° in the estimated camera orientation and are thus, as observed by Sturm et al., not suitable as a benchmark for detailed scene reconstruction.

7 Discussion

There are many limitations and opportunities for future work. Some of the most significant limitations are induced by the sensor, which

Sequence	# of frames	# of POI	Our method	RGB-D SLAM	EKF
fr1/desk	595	1	$0.026m$	$0.034m$	$0.059m$
fr1/desk2	639	2	$0.037m$	$0.061m$	$0.048m$
fr1/room	1,352	2	$0.087m$	$0.223m$	$0.231m$

Table 2: *Three sequences from the RGB-D SLAM benchmark, and the RMSE obtained on these sequences by our method, RGB-D SLAM, and Extended KinectFusion.*



(a) Our result (b) Mocap trajectory + Integration

Figure 7: *Results on the "fr1/room" scene from the RGB-D SLAM benchmark, with our approach on the left and the ground truth camera trajectory localized by a calibrated marker-based motion capture system on the right.*

cannot be operated under strong sunlight, does not provide reliable range data for translucent or highly specular surfaces, and has a limited effective range (roughly 0.5 to 3 meters). Furthermore, the high error magnitudes encountered even within this range limit the quality of the results that can be obtained. We expect to see more accurate sensors available to the public in the future. Nevertheless, we expect the issues encountered and the ideas introduced in our work to remain relevant for a broad class of consumer-grade sensors, including low-power and low-baseline sensors that may be integrated into future mobile devices [PrimeSense 2012]. The data streamed by such sensors will continue to suffer from high error magnitudes for many years to come and we hope that the ideas discussed in this paper can support scene reconstruction even with such low-fidelity input data.

A minor limitation of our current implementation is the largely planar scene analysis pipeline for POI detection. We expect an extension to fully three-dimensional POI mapping to be straightforward. The scenes in our current experiments already contain significant vertical structures and our trajectories contain significant vertical motions.

A more substantial limitation is at the heart of our approach. We assume that errors in the input can be dealt with by sufficiently careful estimation of the camera trajectory. However, this is not necessarily true, since the range images produced by consumer-grade sensors suffer from substantial low-frequency distortion. Even a perfect estimate for the camera trajectory may not be sufficient in itself for recovering accurate surface details. Furthermore, our approach is clearly not guaranteed to produce a perfect trajectory estimate, in part due to the hard partitioning of the trajectory into POI segments and connectors. The connectors may need to absorb substantial residual distortion, leading to visible artifacts in the reconstructed surfaces (Figure 8). We see the use of non-rigid alignment [Brown and Rusinkiewicz 2007] as a promising approach to resolving these

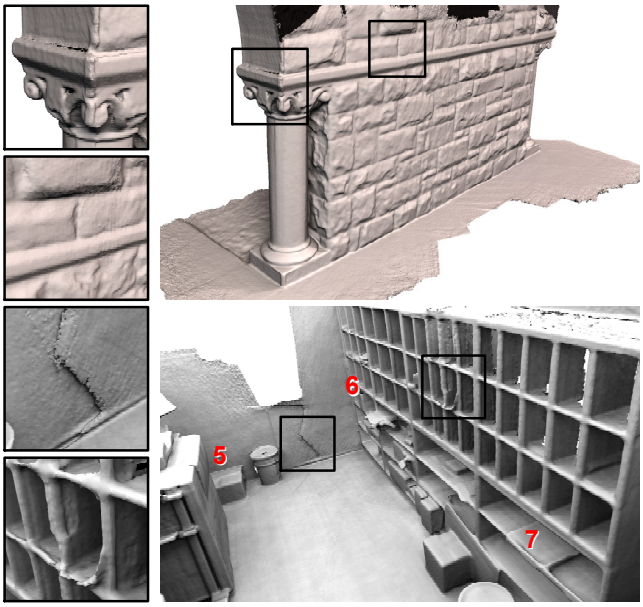


Figure 8: *Two illustrations of the limitations of our approach. Top: a* $5\text{m} \times 0.5\text{m} \times 2.6\text{m}$ *supporting stone wall. No points of interest were detected in this scene. The reconstruction is globally consistent due to loop closure detection, two-pass frame-to-model registration, and global optimization. However, some surface artifacts remain. Bottom: the back of the copy room shown in Figure 5. The connector regions that buffer points of interest 5, 6, and 7 absorb residual error, which is too large to be smoothed out by the volumetric integration and causes visible artifacts in the reconstructed surfaces.*

issues. Adaptation of ideas commonly used in bundle adjustment could be another approach to increasing the accuracy of the reconstructed geometry.

An additional avenue for future work is the integration of color data onto the reconstructed surfaces [Troccoli and Allen 2008] and the reconstruction of detailed reflectance functions [Weyrich et al. 2009]. We plan to make most of our datasets publicly available to support research on these and other aspects of high-fidelity scene reconstruction.

Acknowledgements

We are grateful to the anonymous reviewers for their constructive comments, and to Radek Grzeszczuk and Microsoft Corporation for financial support of this project.

References

AGARWAL, S., SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2010. Bundle adjustment in the large. In *Proc. ECCV*.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 2001.

BROWN, B. J., AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-D scans. *ACM Transactions on Graphics 26*, 3.

CHEN, Y., AND MEDIONI, G. G. 1992. Object modelling by registration of multiple range images. *Image and Vision Computing 10*, 3.

COMANICIU, D., AND MEER, P. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5.

CUI, Y., SCHUON, S., CHAN, D., THRUN, S., AND THEOBALT, C. 2010. 3D shape scanning with a time-of-flight camera. In *Proc. CVPR*.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH*.

ENDRES, F., HESS, J., ENGELHARD, N., STURM, J., CREMERS, D., AND BURGARD, W. 2012. An evaluation of the RGB-D SLAM system. In *IEEE International Conference on Robotics and Automation (ICRA)*.

FUHRMANN, S., AND GOESELE, M. 2011. Fusion of depth maps with multiple scales. *ACM Transactions on Graphics 30*, 6.

FURUKAWA, Y., AND PONCE, J. 2010. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 8.

FURUKAWA, Y., CURLESS, B., SEITZ, S. M., AND SZELISKI, R. 2010. Towards Internet-scale multi-view stereo. In *Proc. CVPR*.

GOESELE, M., SNAVELY, N., CURLESS, B., HOPPE, H., AND SEITZ, S. M. 2007. Multi-view stereo for community photo collections. In *Proc. ICCV*.

HENRY, P., KRAININ, M., HERBST, E., REN, X., AND FOX, D. 2012. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research 31*, 5.

HEREDIA, F., AND FAVIER, R. 2012. *Kinect Fusion extensions to large scale environments*. `http://www.pointclouds.org/blog/srcs/fheredia`.

HUBER, D. F., AND HEBERT, M. 2003. Fully automatic registration of multiple 3D data sets. *Image and Vision Computing 21*, 7.

KHOSHELHAM, K., AND ELBERINK, S. O. 2012. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors 12*, 2.

KUMMERLE, R., GRISETTI, G., STRASDAT, H., KONOLIGE, K., AND BURGARD, W. 2011. g2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*.

MICROSOFT. 2010. *Kinect*. `http://www.xbox.com/en-us/kinect`.

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 *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.

POLLEFEYS, M., GOOL, L. J. V., VERGAUWEN, M., VERBIEST, F., CORNELIS, K., TOPS, J., AND KOCH, R. 2004. Visual modeling with a hand-held camera. *International Journal of Computer Vision 59*, 3.

POLLEFEYS, M., NISTÉR, D., FRAHM, J.-M., AKBARZADEH, A., MORDOHAI, P., CLIPP, B., ENGELS, C., GALLUP, D., KIM, S. J., MERRELL, P., SALMI, C., SINHA, S. N., TALTON, B., WANG, L., YANG, Q., STEWÉNIUS, H., YANG, R., WELCH, G., AND TOWLES, H. 2008. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision 78*, 2-3.

PRIMESENSE. 2012. *PrimeSense unveils Capri*. `http://www.primesense.com/news/primesense-unveils-capri/`.

PULLI, K. 1999. Multiview registration for large data sets. In *Proc. International Conference on 3D Digital Imaging and Modeling (3DIM)*.

ROTH, H., AND VONA, M. 2012. Moving volume KinectFusion. In *British Machine Vision Conference (BMVC)*.

RUHNKE, M., KÜMMERLE, R., GRISETTI, G., AND BURGARD, W. 2012. Highly accurate 3D surface models by sparse surface adjustment. In *IEEE International Conference on Robotics and Automation (ICRA)*.

RUSINKIEWICZ, S., HALL-HOLT, O., AND LEVOY, M. 2002. Real-time 3D model acquisition. *ACM Transactions on Graphics 21*, 3.

RUSU, R. B., AND COUSINS, S. 2011. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*.

SEITZ, S. M., CURLESS, B., DIEBEL, J., SCHARSTEIN, D., AND SZELISKI, R. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*.

STURM, J., ENGELHARD, N., ENDRES, F., BURGARD, W., AND CREMERS, D. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *International Conference on Intelligent Robot Systems (IROS)*.

TRIGGS, B., MCLAUCHLAN, P., HARTLEY, R., AND FITZGIBBON, A. 2000. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*.

TROCCOLI, A., AND ALLEN, P. K. 2008. Building illumination coherent 3D models of large-scale outdoor scenes. *International Journal of Computer Vision 78*, 2-3.

TURK, G., AND LEVOY, M. 1994. Zippered polygon meshes from range images. In *Proc. SIGGRAPH*.

WEISE, T., WISMER, T., LEIBE, B., AND GOOL, L. V. 2011. Online loop closure for real-time interactive 3D scanning. *Computer Vision and Image Understanding 115*, 5.

WEYRICH, T., LAWRENCE, J., LENSCH, H. P. A., RUSINKIEWICZ, S., AND ZICKLER, T. 2009. Principles of appearance acquisition and representation. *Foundations and Trends in Computer Graphics and Vision 4*, 2.

WHELAN, T., JOHANNSSON, H., KAESS, M., LEONARD, J., AND MCDONALD, J. 2013. Robust real-time visual odometry for dense RGB-D mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*.

WILLIAMS, B. P., CUMMINS, M., NEIRA, J., NEWMAN, P. M., REID, I. D., AND TARDÓS, J. D. 2009. A comparison of loop closing techniques in monocular SLAM. *Robotics and Autonomous Systems 57*, 12.

WU, C., AGARWAL, S., CURLESS, B., AND SEITZ, S. M. 2011. Multicore bundle adjustment. In *Proc. CVPR*.

ZENG, M., ZHAO, F., ZHENG, J., AND LIU, X. 2013. Octree-based fusion for realtime 3D reconstruction. *Graphical Models 75*, 3.