Interactive 3D Modeling of Indoor Environments with a Consumer Depth Camera

Hao Du[1], **Peter Henry**[1], **Xiaofeng Ren**[2], **Marvin Cheng**[1], **Dan B Goldman**[3], **Steven M. Seitz**[1], **Dieter Fox**[1,2]

[1]Department of Computer Science and Engineering, University of Washington, Seattle, WA, USA
{duhao,peter,kaiwen,fox,seitz}@cs.washington.edu

[2]Intel Labs Seattle, Seattle, WA, USA
xiaofeng.ren@intel.com

[3]Adobe Systems, Seattle, WA, USA
dgoldman@adobe.com

ABSTRACT

Detailed 3D visual models of indoor spaces, from walls and floors to objects and their configurations, can provide extensive knowledge about the environments as well as rich contextual information of people living therein. Vision-based 3D modeling has only seen limited success in applications, as it faces many technical challenges that only a few experts understand, let alone solve. In this work we utilize (Kinect style) consumer depth cameras to enable non-expert users to scan their personal spaces into 3D models. We build a prototype mobile system for 3D modeling that runs in real-time on a laptop, assisting and interacting with the user on-the-fly. Color and depth are jointly used to achieve robust 3D registration. The system offers online feedback and hints, tolerates human errors and alignment failures, and helps to obtain complete scene coverage. We show that our prototype system can both scan large environments (50 meters across) and at the same time preserve fine details (centimeter accuracy). The capability of detailed 3D modeling leads to many promising applications such as accurate 3D localization, measuring dimensions, and interactive visualization.

Author Keywords

3D mapping, localization, depth camera, user interaction.

ACM Classification Keywords

H.5.m Information interfaces and presentation: Miscellaneous.

General Terms

Algorithms, Design, Experimentation, Measurement.

INTRODUCTION

Detailed 3D visual models of indoor spaces, from walls and floors to objects and their configurations, can provide extensive knowledge about the environments as well as rich contextual information of people living therein and how they interact with the environment. As cameras become ubiquitous and see fast growing usage, accurate 3D visual models will be a cornerstone of future context-aware applications.

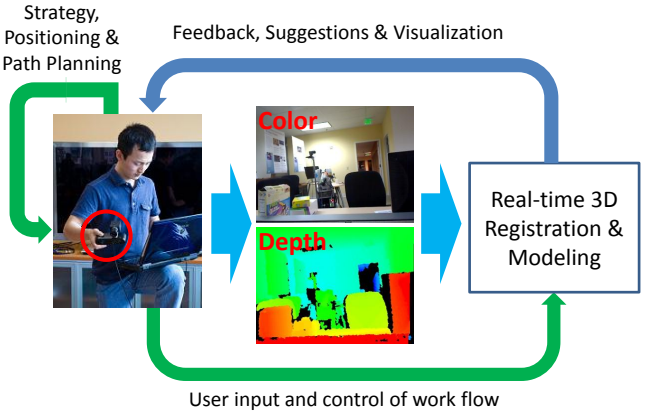

Figure 1. Interactive 3D mapping: color and depth frames from a consumer depth camera are aligned and globally registered in real time. The system alerts the user if the current data cannot be aligned, and provides guidance on where more data needs to be collected. The user can "rewind" and resume the process, and also check the quality and completeness of the 3D model on-the-fly.

Accurate 3D models are difficult to build and thus have rarely been used in applications. Most existing 3D scanning products work only on small objects. Scanning an indoor environment is possible but costly, usually requiring heavy-duty laser scanners on special platforms such as robots [30]. These systems are far from accessible to average consumers.

Recently, image-based 3D modeling has become feasible, with Photo Tourism [28] being a prominent example how 3D structures can be recovered by analyzing and matching photos. There has been research work showing promising results on city-scale outdoor scenes [24, 1]. On the other hand, indoor personal spaces remain difficult due to many technical challenges such as low lighting and textureless surfaces. Limited successes have been reported, including the use of the Manhattan world assumption [8], trading 3D geometric accuracy for robustness in a constrained setting.

UbiComp'11, September 17–21, 2011, Beijing, China.

In this work, we present a prototype mobile system that enables a non-expert user to build dense and complete models for his/her personal environments, running on a laptop in real-time and interacting with the user on-the-fly. One technology that makes this feasible is the wide availability of affordable depth cameras, such as those deployed in the Microsoft Kinect system [19, 25]. These cameras directly provide dense color and depth information. However, their field of view is limited (about 60°) and the data is rather noisy and low resolution (640×480). Henry et al [12] showed that such cameras are suitable for dense 3D modeling, but much was left to be desired, such as robustness for use by non-experts, or complete coverage of the environment including featureless or low-light areas.

The key idea behind our work is that, by running 3D modeling in real-time on a mobile device, the system can explore the space together with the user and take advantage of online user interaction and guidance (see Figure 1). We design an interactive 3D modeling system so that the user holds a depth camera in-hand to freely scan an environment and get feedback on-the-spot. Online user interaction, with a freely moving depth camera, solves many of the challenging issues in 3D environment modeling:

Robustness: We compute 3D alignments of depth frames on-the-fly, so that the system can detect failures (due to many reasons such as fast motions or featureless areas) and prompt the user to "rewind" and resume scanning. The success of 3D registration of consecutive frames is thus "guaranteed".

Completeness: A 3D environment model is constructed on-the-fly. The user can check the model in 3D at any time for coverage and quality. The system also automatically provides suggestions where the map may yet be incomplete.

Dense coverage: Largely due to the use of a depth sensor, our system produces visual models that are dense in 3D space, even in textureless areas, comparing favorably to most existing work in vision- or laser-based modeling. A dense model reveals fine details of the environment.

Our system is capable of scanning large-scale indoor spaces, such as office buildings of 50 meters across, with centimeter-level details accurate in both 3D geometry and color appearance. We show a variety of models built with our online mobile system covering different sizes and types, when used by both expert and non-expert users. We compare our interactive system to traditional offline approaches and demonstrate how user interaction makes the system robust enough to be accessible to everyone with a depth camera[1].

How can we use such a detailed 3D map? We demonstrate three promising directions: (1) localization in 3D space approaching decimeter accuracy; (2) measuring dimensions of spaces and objects; and (3) photorealistic visualization of 3D environments applicable to virtual remodeling and furniture shopping. We believe that a consumer-friendly 3D modeling system will open up many more opportunities for interesting applications that may benefit from a rich 3D context.

[1]Supplemental videos demonstrating our interactive system can be found at: `http://www.cs.washington.edu/robotics/projects/interactive-mapping/`

RELATED WORKS

Geometrically modeling a physical environment is a problem of general importance, and for a long time it was done manually by specialists using specialized tools. Automatically mapping an environment is challenging, and one success story is that of using a laser scanner on a mobile robot [7, 30]. Robot mapping has been shown to be robust, but is mostly limited to 2D maps and relies on expensive hardware.

In the Ubiquitous Computing community, much attention has been devoted to a closely related problem: localization. A variety of signals has been used in indoor localization [13, 17, 2], such as 802.11 [3], GSM [22], and recently powerline signals [23]. There is a limit on the localization accuracy using these low-rate signals, the state of the art being around 0.5 meters [32]. Many efforts have been made on how to simplify and avoid extensive calibration [18].

Vision-based 3D modeling techniques have been gaining popularity in recent years. Building on multi-view geometry [11] and in particular bundle adjustment algorithms [31], 3D structures can be recovered from a set of 2D views. PhotoTourism [28] is an example where sparse 3D models are constructed from web photos. There has been a lot of work on multi-view stereo techniques [26]. The patch-based framework [10], which has been most successful on object modeling, has also been applied to environment modeling. The work of Furukawa et al. [9] built on these works and obtained dense indoor models using the Manhattan world assumption.

There have been successful efforts to build real-time systems for 3D structure recovery. Davison et al. built real-time SLAM (simultaneous localization and mapping) [5] systems using monocular cameras. Klein et al. built the Parallel Tracking and Mapping (PTAM) [15] system which applies SLAM for tracking. Pollefeys et al. [24] proposed real-time solutions for street-view reconstruction. Newcombe et al. [20] recently used PTAM and optical flow techniques to compute dense depths. Many real-time systems are limited to small-scale spaces.

Due to the difficulties of indoor modeling such as low lighting and lack of texture, interactive approaches have been proposed to utilize human input. [6] was an early example showing very impressive facade models and visualizations with manual labeling. [29] used interactions to extract planes from a single image. [27] was a recent example combining user input with vanishing line analysis and multi-view stereo to recover polygonal structures. Our approach to interactive mapping is different in nature, as we enable online user interaction, utilizing user input on-the-fly for both capturing data and extracting geometric primitives.

The arrival of consumer depth cameras such as Kinect [19] and Primesense [25] is significant. We believe these affordable depth cameras will soon see extensive uses in applica-

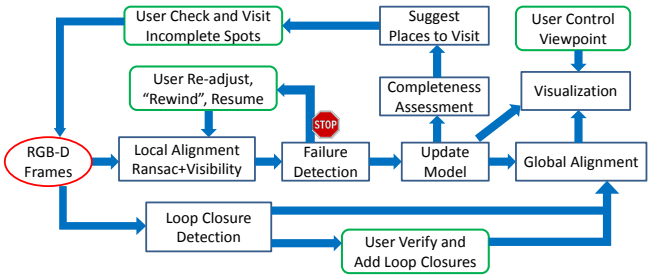

Figure 2. Detailed system overview: frame alignment, loop closure detection, and global alignment are performed in real time. Green boxes represent user interactions. The user is alerted if alignment fails, is notified of suggested places to visit, and can verify and improve the model quality via manual loop closure.

tions beyond gaming, in particular in 3D modeling, making it accessible to consumers. The work by Henry et al. [12] is the most relevant as a pioneer of 3D modeling using consumer depth cameras. They carried out experimental studies of alignment algorithms and combinations using both color and depth. Our work aims at making such a depth-camera-based modeling system work in real-time, incorporating various aspects of user interaction to make 3D modeling robust, easy to use, and capable of producing dense, complete models of personal spaces.

SYSTEM OVERVIEW

Figure 2 shows an overview of the design of our interactive mapping system. The base system follows a well-established structure of 3D mapping, which partitions the registration of RGB-D (color+depth) frames into *local alignment* plus *global alignment*. Local alignment, or *visual odometry* [21], is a frame-to-frame registration step matching the current frame to the most recent frame. The current frame is also matched to a subset of "keyframes" in the system to detect "loop closure" [16, 12, 4], i.e. whether this is a revisit to a known scene. Global alignment uses loop closure information to jointly optimize over all the RGB-D frames to produce globally consistent camera poses and maps.

User feedback and interaction is enabled at multiple levels in our system. First, if local alignment fails, i.e. the current frame cannot be registered into the map, the system alerts the user and pauses the mapping process. The user re-orients the camera with suggestions from the system, until local alignment succeeds and the mapping resumes. Second, the user has control over loop closure detection by verifying and adding links that are difficult decisions for an automatic algorithm. Combining both, our system is robust to algorithm failures or human errors, making it "guaranteed" to produce globally consistent maps given sufficient user supervision.

In our system, the globally aligned map is visualized on-the-fly, as shown in Figure 3. The user can see how the 3D map grows in real-time, freely change the 3D viewpoint to check for inconsistencies and completeness, and plan the camera path accordingly to ensure a complete and consistent map. To facilitate, the system continuously checks the completeness of the current map, provides visual feedback about incomplete areas, and guides the user to locations from which the view could bring added map coverage.

By closely integrating the user into the data collection phase and making him/her fully aware of the state of the process, our interactive system avoids many pitfalls in 3D mapping and achieves robustness and completeness even in challenging cases. As can be seen in Figure 1 and the supplemental video, our prototype system runs on a laptop, and is fully mobile. We envision that, in the near future, such a system can be integrated and packaged to a portable form factor with a touch interface and become accessible to everyone.

ROBUST RGB+DEPTH REGISTRATION

In this section we describe the core of our real-time 3D registration algorithm. The *3-Point* matching algorithm is used to compute full 6-D transformations between frame pairs [14]. A novel matching criterion is used to combine RANSAC inlier count with *visibility* conflict. Combining the visibility criterion, matching is much more robust in difficult cases where the inlier count is small and becomes unreliable.

RANSAC and 3-Point Matching

Following the RGB-D alignment approach in [12], we detect visual features in the color frame using the GPU implementation of the standard SIFT features [33], find matches using the SIFT ratio test, and use RANSAC to prune outliers and find the camera pose transform between two frames. It is worth noting that since we have depth information for the visual features, only 3 point pairs are needed in RANSAC search, making it much more robust and efficient than the classical 7-point solution. Moreover, the 6-D full transform can be computed without scale ambiguity.

Consider N pairs of initial feature matches between Frame F_1 and F_2, represented by 3D coordinates (X_1^i, X_2^i) in their respective reference systems. RANSAC samples the solution space of (R, T) (rotation and translation) and estimates its fitness by counting the number of inliers, f_0,

$$f_0(F_1, F_2, R, T) = \sum_i^N L(X_1^i, X_2^i, R, T), \qquad (1)$$

where,

$$L(X_1^i, X_2^i, R, T) = \begin{cases} 1, & e = \|RX_1^i + T - X_2^i\| < \epsilon \\ 0, & otherwise \end{cases}$$

and ϵ is the threshold beneath which a feature match (X_1^i, X_2^i) is determined to be an inlier. RANSAC chooses the transform consistent with the largest number of inlier matches.

Combining RANSAC with Visibility

The RANSAC framework above only accesses depth values at the SIFT feature points. To make use of dense depth information available at all pixels, we introduce a visibility confliction term and combine it with the RANSAC inlier count. Consider the 2D example shown in Figure 4 (left), showing two camera views. The circles and stars are the depth maps sampled at the camera pixels. When (R, T) is

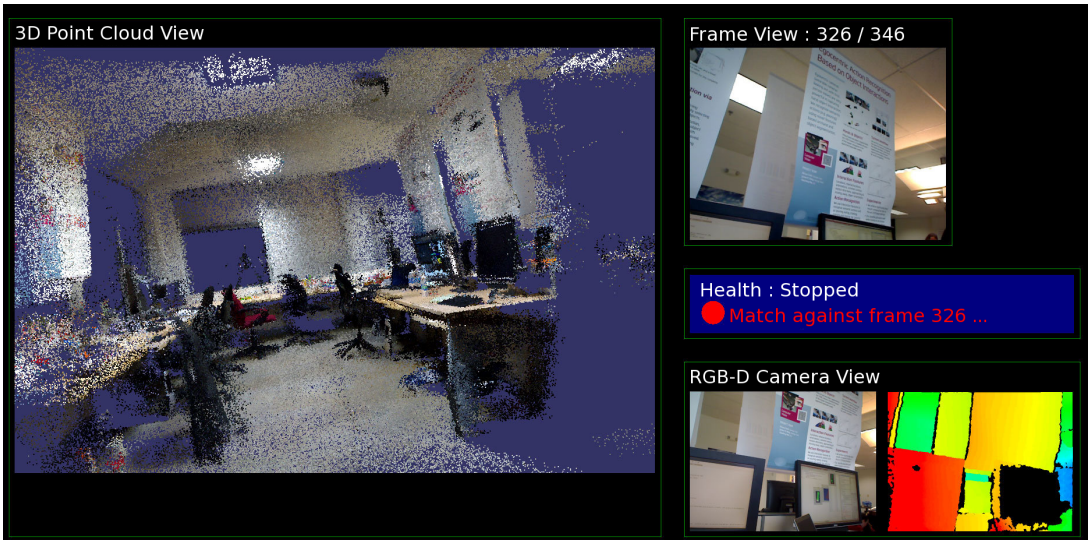

Figure 3. Real time visualization of the mapping process: The left panel provides a 3D view of the globally aligned map. The health bar in the center right panel indicates the current quality of frame alignment. In a failure case, as is shown, the user is guided to re-orient the camera with respect to a target frame registered in the map. The upper right panel shows this target frame, and lower right panel indicates the current camera view.

the genuine relative transformation, the two scenes overlap perfectly. When (R^*, T^*) is a wrong relative transformation, shown in Figure 4 (right), overlaying the point clouds from both cameras, it is possible to see visibility conflicts – when a camera captures a scene point in 3D, the space along its viewing line should be completely empty; if there exist points from the other camera in between due to a incorrect (R^*, T^*), there is a conflict.

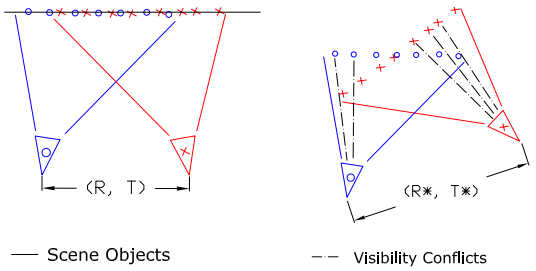

Figure 4. Visibility conflicts. Left: what a surface should look like from two depth camera views. Right: inconsistent camera poses lead to visibility conflicts, useful information for RANSAC search.

We compute visibility conflict by projecting the point cloud C_1 from frame F_1 onto the image plane of F_2, and vice versa. If the depth of a pixel of C_1 is smaller than the depth of the F_2's pixel at the corresponding location (larger than a varying threshold equal to depth uncertainty), it is counted as a visibility conflict. We compute the following quantities: number of visibility conflicts (f_1); average squared distance of points with visibility conflicts (f_2); number of visibility inliers (f_3) by counting those pixels where no visibility conflicts both ways of projections. We use a linear function to combine these with the RANSAC inlier count:

$$g(F_1, F_2, R, T) = \sum_{i=0}^{m} \alpha_i f_i \qquad (2)$$

where m is the number of quantities involved, and the weights α_i are learned through linear regression.

INTRODUCING USER INTERACTION

User interaction has been utilized in 3D modeling to overcome the incompetency of automatic algorithms in hard cases (e.g. [27]). Existing work typically views user interaction as a post-processing step: after all the data have been acquired, the user sits down at a desktop computer, and then marks up and corrects structures in the model.

Our approach utilizes interaction in a different nature – involving user interaction early in the image data acquisition stage as well as in post-processing. Decent source data is crucial for a successful reconstruction, and user interaction can significantly help data capture. We envision that a compact mobile system/device can be developed for a user to hold in his/her hand, freely move it to scan scenes, and interact with the system as the 3D map is being built. We view data acquisition and model construction as one integrated process that happens on-the-spot.

As shown in Figure 2, our prototype system incorporates three types of user interactions to achieve robustness and completeness in 3D mapping: (1) failure detection and rewind/resume; (2) scene completeness guidance; and (3) user-assisted loop closure.

Rewind & Resume and Failure Tolerance

In the case of indoor 3D mapping, each camera view or frame usually covers a small area of the entire scene, so it is crucial to align the frames together and merge them. Since consecutive frames have the most similar views, a typical approach is to align neighboring frames first.

Frame-to-frame matching, however, can fail for many reasons, especially at the hand of a non-technical user who does

not have any understanding of the mechanisms and limits of vision-based mapping. For example, if the user moves the camera too fast, motion blur will fail the matching. Even if the user is careful, matching can fail when the field of view does not contain sufficient color and geometric features. As we will see in the experiments, even with a state-of-the-art RGB-D mapping approach, failures can be frequent.

We design our system to be robust to matching failures, with what we call *Rewind and Resume*. When a frame fails to register with the previous frame, the system raises an alert, pauses the mapping process, and waits for a new frame that can successfully register (Figure 3). The system shows the user what frame it expects, so the user can move the camera to match the expected view. Moreover, it is not limited to the most recent frame: the user can either "undo" and drop the most recent frames (say to remove frames of an intruding person), or "rewind" to any previous frame in the system and "resume" at a different point. This results in a tree-structured data flow and makes the system more flexible and usable. The user can save the partial map to disk, and resume on a different day from any scene that is already captured.

Model Completeness

Capturing a complete 3D model of the scene is desired, because large missing areas in an incomplete 3D model significantly lower the visual quality. A missing area exists in the scene either because the area has never been captured or the frames that did contain the area did not get depth values, for reasons such as camera range or surface slant.

We consider the completeness in a user-defined manner. Using a passive capturing system, it can be very difficult for the user to be aware of which parts of the scene have been captured. With an online system, the user can view the current reconstruction in real time, view the up-to-date 3D model, and directly see which areas are missing.

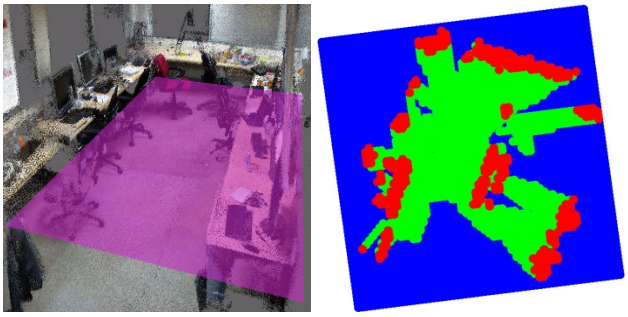

Figure 5. Completeness Guide. At the user's request, our system displays the classification of voxels from a user specified 2D slice in the 3D point cloud. Green: the voxel is guaranteed to be "empty"; Red: it is "occupied"; Blue: "unknown" area.

In order to further assist users in finding uncaptured areas, our system is able to estimate completeness. Consider a bounding box that contains the currently reconstructed point cloud. The inside of the bounding box can be represented by 3D grid voxels. Each grid voxel is classified into one of the three categories: (1) "occupied": there is at least a scene point in that voxel; (2) "empty": there must be no scene point in the voxel; (3) "unknown": none of the above. All voxels are initialized in Category (3). A voxel is set as Category (1) when there exists a scene point. A voxel is set as Category (2) when it is not (1) and the voxel has been seen through by any of the existing camera viewing line.

Figure 5 shows the classification of voxels from a user specified 2D slice in the 3D point cloud. Green: the voxel is guaranteed to be "empty"; Red: it is "occupied"; Blue: "unknown" area. The user's goal is then to paint all areas in either green or red by exploring the 3D space.

Interactive Loop Closure

Loop closure (global pose optimization based on frame-pair matches captured at different times) helps to fix the accumulated errors originating from sequential frame-to-frame matching. Automatic loop closure is hard as there may be large differences between scene frames over a loop [12]. A single matching outlier due to low lighting or few features could cause the whole loop to be inconsistent.

A decent loop closure does not require matches between many pairs of frames. When combined with frame-to-frame alignments, a few matches over key frames can well constrain the entire loop and facilitate global consistency. This provides us an opportunity to involve user supervision. Our system runs an automatic matching algorithm to select candidate frame pairs, ranks them using the proposed visibility criterion, and then suggests frame pairs with large visibility conflicts. The user can select any frame pair to perform a RANSAC or ICP based alignment to add a loop closure constraint, inspect the resulting map, and then decide to accept or reject the newly added constraint.

3D MAPPING EVALUATIONS

In this section, we present experimental results of our 3D mapping system. We show three types of evaluations: (1) how our RGB-D RANSAC utilizes dense depth information and outperforms classical computer vision solutions; (2) how user interaction plays an essential role in making mapping successful; and (3) how our results compare to the state of the art in vision-based 3D modeling. We also show examples of the 3D models that we have scanned with the system.

We use a PrimeSense camera [25] with resolution 640-by-480 and an effective range of $0.5m$ to $5m$. The depth uncertainty, needed for visibility conflict, is calculated from camera specifications ($7.5cm$ baseline, 570 pixel focal length), approximately $0.03cm$ at a distance of $0.3m$ and $7cm$ at $5m$. All the computations and visualizations are done on an Intel i7-720qm laptop at 3 to 4 frames per second.

RGB-D RANSAC for Visual Odometry

We compare the accuracy of the 3-point RANSAC solution on RGB-D frames to that of the 7-point algorithm [11]. We generate feature pairs via bidirectional matching, considering a pair matched if they are both the best matching feature to the other, and the second best match has significantly higher distance in the feature descriptor space (ratio > 1.25).

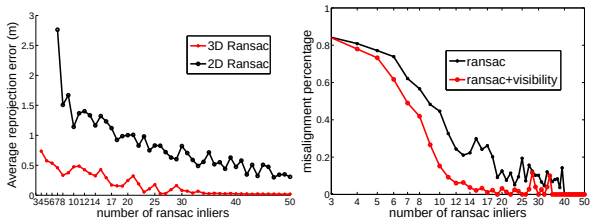

Figure 6. (Left) Alignment errors of 7 point 2D RANSAC and 3 point 3D RANSAC versus number of inliers. (Right) Percentage of misaligned frames of 3D RANSAC vs 3D RANSAC+Visibility.

2D versus 3D RANSAC. We collect a long RGB-D sequence in the lab through slow motion and use our system to generate a globally optimized map, shown in the bottom left panel of Figure 12. Visual inspection shows that the resulting map is highly consistent, and we use the globally optimized poses as groundtruth. We randomly sample frame pairs from this dataset and compare the 7-point 2D RANSAC (without further optimization after obtaining 7-point algorithm) solution and the 3-point 3D RANSAC solution. 2D RANSAC finds $15,149$ valid pairs (≥ 7 inliers), with an average reprojection error of 0.69m. 3D RANSAC finds $15,693$ valid pairs (≥ 3 inliers) with an error of 0.31m. Figure 6(left) shows the reprojection error versus the number of inliers in the solution. Using depth information within RANSAC (3D RANSAC) clearly reduces reprojection errors and generates good results even for a small number of inliers.

3D RANSAC with Visibility Features. We also compare the accuracy of 3-Point RANSAC with and without incorporating the visibility conflict criterion. We collect a dataset by placing the depth camera at 12 different locations, measure their ground truth distances, and rotate the camera at each location to take 100 depth frames. We randomly pick pairs of camera frames, and randomly split the set into training (linear regression for weights) and testing. Figure 6 (right) shows the results using a threshold of $0.1m$ on camera translation for determining a misalignment. "RANSAC + Visibility" produces more accurate camera poses and is capable of working with a lower threshold on RANSAC inliers.

The Importance of User Interaction

To evaluate the capability of our interactive system to generate improved visual odometry data, we performed a small study in which four persons were tasked to collect data for a map of a small meeting room. A 3D map generated with our system is shown in Figure 7. For each of the four people, we determined if (s)he was able to collect data that can be consecutively aligned for visual odometry. Two of the people were "expert users" who had substantial experience in using the depth camera for mapping purposes. Two persons were "novice users" who had not studied mapping. The different mapping runs collected between 357 and 781 frames. In all cases, the users were able to complete the mapping process using the interactive system.

The results for this small-scale user study are shown in Table 1. We evaluate the mapping process using several quantities: for offline mapping (similar to the setup in [12]), we can

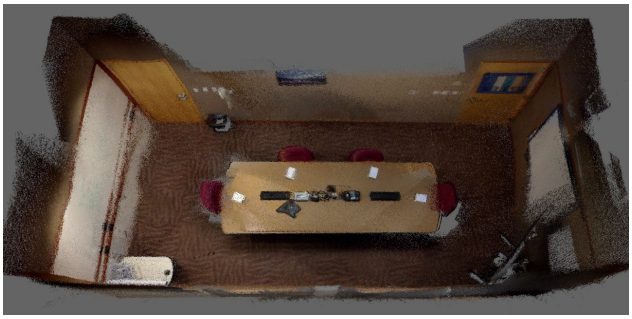

Figure 7. Top down view of 3D map of meeting room used to evaluate the benefits of interactive mapping.

detect failures in the mapping algorithm (when RANSAC fails). The first time (frame number) when failure occurs is a measure of how far a user can go into the process, or how complete a map he can expect, if there is no feedback and the user blindly continues data collection. Similarly, the average number of frames between failure is a measure of how often failure occurs and how large a map can grow in offline mapping. For online interactive mapping, the system reports failures directly and recovers from them, so we can count the number of failures when users finish. Finally, we ask the user to start and finish on a table, so the groundtruth heights of the start and end poses are the same (vertical distance zero). We compute the vertical distance of the estimated camera poses and use it as a measure of alignment error.

While the sample size may be too small to be conclusive, we do see patterns and trends in these results showing the promise of interactive mapping. We see that failures occurred often, and for a moderately difficult scenario like the conference room (low lighting and many textureless areas), the users failed early in the process using the offline system, and could only complete 3 to 10 percent of the process. The mean time between failure was only seconds. On the other hand, the online interactive system always "worked": the users all managed to recover from the failures and completed the process by going from the assigned starting point to the end. That is also reflected in the vertical alignment error – the offline errors are much larger than online errors, suggesting that offline mapping failed and interactive mapping had reasonable successes.

Comparison to Bundler and PMVS

We compared our system, using cheap depth cameras, to that of the state-of-the-art approach of using Bundler [28] and PMVS [9] which reconstruct dense 3D models from photo collections. We collected a depth camera sequence (PrimeSense) and a collection of high-res camera images (Canon 5D Mk II) of a wall and a textured white board standing in front of it. Figure 8 shows a zoom into the white board part of the reconstruction achieved by our system (left) and by Bundler+PMVS using the high-res images. Even with a lot of texture and super high-res photos, PMVS failed to produce a dense model, leaving many holes. Our system captured entire areas densely without holes.

Metrics	Offline (novice)	Interactive (novice)	Offline (expert)	Interactive (expert)
#frame at 1st failure (offline)	15.0	-	66.5	-
Mean #frame to failure (offline)	9.9	-	9.6	-
#failure (interactive)	-	9.0	-	17.0
Alignment Error (m)	1.14	0.11	1.12	0.05

Table 1. A small-scale user study of 4 users, 2 expert (who know 3D mapping) and 2 non-expert (do not know 3D mapping), comparing interactive mapping with state-of-the-art offline mapping. While the sample size may be too small to draw any conclusion, these results do strongly suggest that interactive mapping has great potential and can make the mapping process robust and error-tolerant.

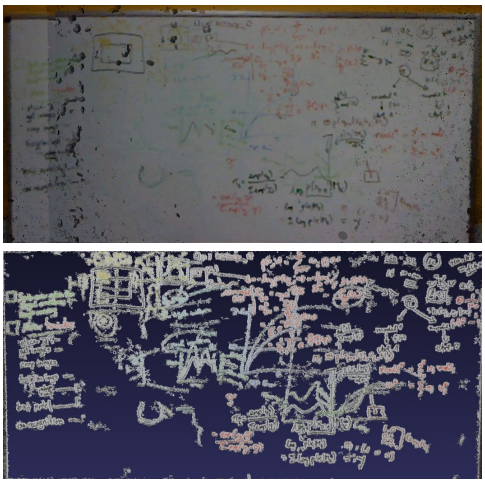

Figure 8. 3D reconstruction by our interactive system (top) and by Bundler+PMVS using high-res camera images (bottom). Blue indicates areas without any reconstruction.

Acquired Models of Large Indoor Environments

Figure 12 shows examples of maps we built with our interactive system. These results were achieved by collecting good visual odometry data using the failure detection and relocalization process. For large maps, automatic loop closure ran into troubles, and the interactive system allowed us to add loop closure constraints manually. The globally consistent map shown at the top was generated with 25 manual loop closure links merging with automatic loop closure.

3D MAPPING APPLICATIONS

What can detailed 3D models be used for if everyone can scan their personal spaces into 3D? As can be seen in the map visualizations and the supplemental video , our interactive system produces accurate 3D models that contain a large amount of detail, from large architectural elements (e.g. walls and floors) to small-sized objects situated in 3D settings. There are a multitude of possible applications, especially in context awareness and wearable computing, where the 3D map, along with objects inferred from it, provides rich context for users and their activities.

We briefly explore and demonstrate three potential applications: (1) 3D localization, locating a camera view in a precomputed map; (2) measuring dimensions, obtaining lengths and sizes in the virtual map; and (3) free exploration and photorealistic rendering of the environment, with connections to virtual furniture shopping. We leave rigorous developments and evaluations of the applications to future work.

Accurate 3D Localization

Once we have a 3D model constructed, we can turn our system into a localization mode, where new camera views are continuously registered but the map is no longer updated. This simple change allows us to localize a depth camera in 3D space, comparing to typical 2D localization.

Figure 9. 3D localization: using a pre-computed 3D map captured on a different day, our system localizes itself in 3D (position+orientation). Shown is the path of (3D) camera poses from two viewpoints.

Figure 9 shows an example of how 3D localization works with our approach. The map shown is built from a separate set of data captured on a different day. When we run the mapping system on a new day, new RGB-D frames are registered into the existing map through loop closure, and then continuously tracked over time in 6-D pose. The path of camera poses, of about 100 views, is shown at the center of the map from two view points. More details can be found in the supplemental video.

Although we do not have groundtruth for camera poses, it should be clear from the camera path that (1) our approach can localize in 3D, both in translation and in rotation, not

just in a flat 2D plane; (2) a vision-based localization approach, using a detailed 3D map, can achieve localization accuracy that is much higher than using other sensing modalities such as Wi-Fi. While $0.5m$ accuracy is normal for a Wi-Fi based system [32], our approach has the potential to achieve decimeter or even centimeter level accuracy. Such high-accuracy localization, possibly done with a wearable camera, can be useful for many location-based applications and also in robotics. An RGB-D approach like ours will also be robust to lighting conditions and textureless areas.

Measuring Dimensions of Spaces and Objects

A second application, directly derived from the geometric correctness of the 3D models, is to measure dimensions and sizes of spaces and objects virtually. Imagine a person goes into an environment, moves the camera and scans it into 3D. Now he has all the 3D information in the virtual model, and he can measure the dimensions of anything later. If a person builds a 3D model of his home, if there is any need for measurements, for instance when shopping for furniture, he can easily do it on a mobile device in his virtual home.

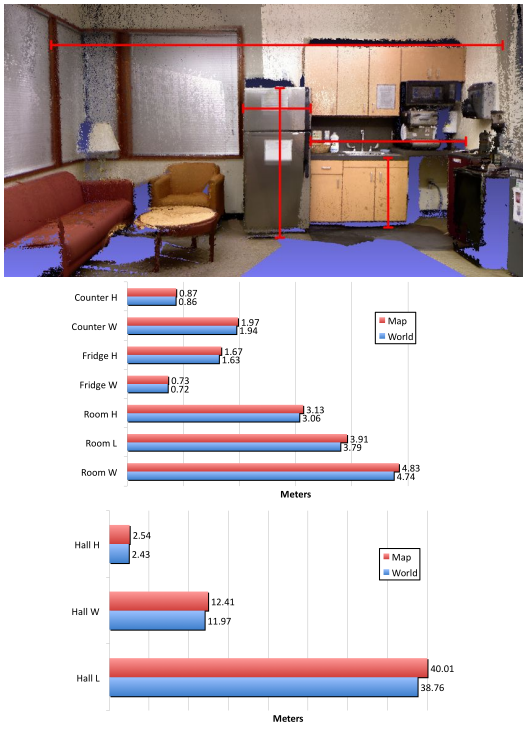

Figure 10. Measuring dimensions of objects (length, width, height) in a virtual model (top). We show quantitative evaluations of the measured virtual dimensions comparing those in the physical world (bottom).

Figure 10(top) shows an example of the dimensions that we measure from a virtual model of the Espresso Room. A single 3D model contains many details such as distances between walls and sizes of objects, which can be easily measured. In Figure 10(bottom) we compare measurements in the virtual models (espresso room and building hallway) to that done in the physical world (with a laser meter). We see that the measurements in the 3D model is accurate, with errors typically under one percent. We also notice that there is a systematic bias in these measurements, suggesting that the measurement accuracy may still improve with better calibration and/or better camera hardware.

Interactive Visualization and Furniture Shopping

One can take the furniture shopping application further: with a detailed 3D model, not only can we measure dimensions in the virtual model to see if a piece of furniture fits, we can potentially visualize, photo-realistically, how the furniture may fit into the home. The rendering can incorporate different arrangements of furniture, different colors, and changing illuminations.

Figure 11. A gesture-controlled flythrough system running on a 3D TV. The user uses hand gestures to navigate through a virtual space being rendered in real-time; he can also use gestures to place a downloaded sofa model into the constructed model.

Toward this goal, we have developed a prototype interactive visualization system, as shown in Figure 11, on a 3D TV. The system has a gesture control interface using the same type of depth camera: the user uses hand gestures to control up, down, left, right, forward and backward. The gestures are used to navigate through the virtual space while it is rendered at real-time, using level-of-detail control and hierarchical culling. The same gestures can also be used in a furniture placement mode, where a pre-downloaded furniture model (e.g. from Google 3D Warehouse) is rendered into the constructed environment model.

DISCUSSIONS AND FUTURE WORK

We have presented a prototype mobile system for 3D mapping and modeling that introduces and extensively uses online user feedback and interaction. We allow the user to freely move a camera through an indoor space, track the 3D mapping process, recover from (inevitable) registration failures, and achieve complete coverage through visual inspection and automatic guide. We have successfully scanned a variety of indoor spaces using the system, including large spaces up to 50 meters across with centimeter level details.

How may such a detailed and accurate 3D model be useful? We have briefly shown and discussed three applications: 3D localization with the potential of centimeter level accuracy; measuring dimensions of spaces and objects with 99% accuracy; and photorealistic visualization of personal spaces, with possible applications in virtual furniture shopping. These are all promising directions, and we plan to explore them in depth in future work.

Figure 12. A variety of indoor spaces that we have captured using our 3D mapping system. (top) a large office space (50 meters across), with a detailed zoom-in view. (middle) another office building structure with repetitive offices and long hallways. (bottom) a (small-size) office and a (small-size) coffee room with furnitures and details.

Our work shows great promises for personal 3D mapping and opens up many possible applications. While 3D modeling is traditionally known to be hard, consumer depth cameras, real-time mapping, and user interaction have great potential to make it accessible to everyone. Our vision is that we can soon build compact mobile devices that allow a consumer to build 3D models for personal spaces. As shown in localization, such a 3D model can provide rich information for many applications that benefit from spatial contexts.

A lot more information can be extracted from such detailed 3D models other than spatial dimensions and layout. As seen from the maps in Figure 12, rich details show opportunities for semantic analysis, such as finding walls and floors, extracting objects, and learning human-object interactions.

Acknowledgments. We thank Matthai Philipose and Keith Mosher for their gesture work using PrimeSense. This work was supported by an Intel grant, and partially by NSF grants IIS-0963657 and IIS-0812671, by ONR MURI grant N00014-09-1-1052, and through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under Agreement W911NF-10-2-0016.

REFERENCES

1. S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building Rome in a Day. In *Computer Vision and Pattern Recognition (CVPR)*, pages 72–79, 2010.

2. M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: Mobile Phone Localization via Ambience Fingerprinting. In *International Conference on Mobile Computing and Networking*, pages 261–272. ACM, 2009.

3. P. Bahl and V. Padmanabhan. RADAR: An In-building RF-based User Location and Tracking System. In *INFOCOM 2000*, volume 2, pages 775–784. IEEE, 2000.

4. L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardós. Mapping Large Loops with a Single Hand-Held Camera. In *Robotics Science and Systems (RSS)*, 2007.

5. A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 1052–1067, 2007.

6. P. Debevec, C. J. Taylor, and J. Malik. Modeling and Rendering Architecture from Photographs: A Hybrid Geometryand Image-based Approach. In *SIGGRAPH*, 1996.

7. D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In *AAAI*, pages 343–349, 1999.

8. Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Manhattan-world Stereo. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009.

9. Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing Building Interiors from Images. In *Intl. Conference on Computer Vision (ICCV)*, 2009.

10. Y. Furukawa and J. Ponce. Accurate, Dense, and Robust Multi-view Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2009.

11. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

12. 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 *Intl. Symposium on Experimental Robotics (ISER)*, 2010.

13. J. Hightower and G. Borriello. Location Systems for Ubiquitous Computing. *Computer*, 34(8):57–66, 2001.

14. B. K. P. Horn. Closed-form Solution of Absolute Orientation using Unit Quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, 1987.

15. G. Klein and D. W. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Intl. Symposium on Mixed and Augmented Reality*, 2007.

16. K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-Based Maps. *Intl. Journal of Robotics Research (IJRR)*, 29(10), 2010.

17. A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place Lab: Device Positioning using Radio Beacons in the Wild. *Pervasive Computing*, pages 116–133, 2005.

18. A. LaMarca, J. Hightower, I. Smith, and S. Consolvo. Self-mapping in 802.11 Location Systems. *Ubiquitous Computing (UbiComp)*, pages 87–104, 2005.

19. Microsoft Kinect. `http://www.xbox.com/en-US/kinect`, 2010.

20. R. A. Newcombe and A. J. Davison. Live Dense Reconstruction with a Single Moving Camera. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

21. D. Nistér, O. Naroditsky, and J. Bergen. Visual Odometry. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2004.

22. V. Otsason, A. Varshavsky, A. LaMarca, and E. De Lara. Accurate GSM Indoor Localization. *Ubiquitous Computing (UbiComp)*, pages 141–158, 2005.

23. S. Patel, K. Truong, and G. Abowd. Powerline Positioning: A Practical Sub-room-level Indoor Location System for Domestic Use. *Ubiquitous Computing (UbiComp)*, pages 441–458, 2006.

24. M. Pollefeys, D. Nister, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, and H. Towles. Detailed Real-Time Urban 3D Reconstruction From Video. *Intl. Journal on Computer Vision (IJCV)*, 72(2):143–67, 2008.

25. PrimeSense. `http://www.primesense.com/`.

26. S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-view Stereo Reconstruction Algorithms. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 519–528, 2006.

27. S. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3D architectural modeling from unordered photo collections. *ACM Transactions on Graphics (TOG)*, 27(5):1–10, 2008.

28. N. Snavely, S. M. Seitz, and R. Szeliski. Photo Tourism: Exploring Photo Collections in 3D. In *SIGGRAPH*, 2006.

29. P. Sturm and S. Maybank. A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images. In *British Machine Vision Conference (BMVC)*, pages 265–274, 1999.

30. S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

31. B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle Adjustment: A Modern Synthesis. *Vision Algorithms: Theory and Practice*, pages 153–177, 2000.

32. O. Woodman and R. Harle. Pedestrian Localisation for Indoor Environments. In *Ubiquitous Computing (UbiComp)*, pages 114–123. ACM, 2008.

33. C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). `http://cs.unc.edu/~ccwu/siftgpu`, 2007.