3DMatch: Learning the Matching of Local 3D Geometry in Range Scans

Andy Zeng[1] Shuran Song[1] Matthias Nießner[2] Matthew Fisher[2] Jianxiong Xiao[1]
[1]Princeton University [2]Stanford University

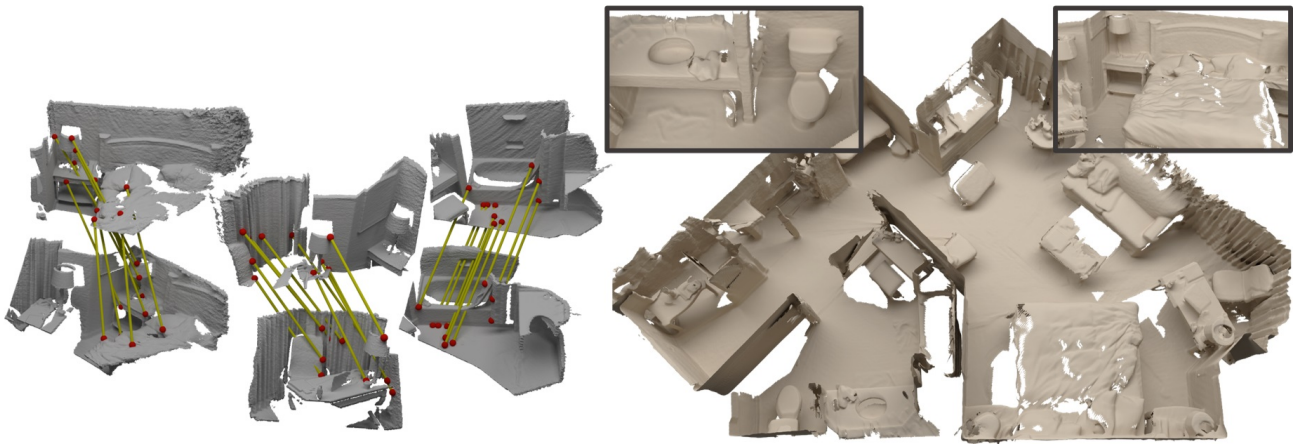

Figure 1: We learn an efficient and robust 3D descriptor for matching local geometry. Specifically, we focus on partial, noisy 3D data obtained from commodity range sensors. On the left, we show three sample pairs of geometric fragments from a scan whose features are matched with our method. We can use these matches of sparse geometric features in order to obtain a 3D reconstruction using a simple (geometric-only) sparse bundle adjustment formulation.

Abstract

Establishing correspondences between 3D geometries is essential to a large variety of graphics and vision applications, including 3D reconstruction, localization, and shape matching. Despite significant progress, geometric matching on real-world 3D data is still a challenging task due to the noisy, low-resolution, and incomplete nature of scanning data. These difficulties limit the performance of current state-of-art methods which are typically based on histograms over geometric properties. In this paper, we introduce 3DMatch[1], a data-driven local feature learner that jointly learns a geometric feature representation and an associated metric function from a large collection of real-world scanning data. We represent 3D geometry using accumulated distance fields around key-point locations. This representation is suited to handle noisy and partial scanning data, and concurrently supports deep learning with convolutional neural networks directly in 3D. To train the networks, we propose a way to automatically generate correspondence labels for deep learning by leveraging existing RGB-D reconstruction algorithms. In our results, we demonstrate that we are able to outperform state-of-the-art approaches by a significant margin. In addition, we show the robustness of our descriptor in a purely geometric sparse bundle adjustment pipeline for 3D reconstruction.

[1]All of our code, datasets, and trained models are publicly available: http://3dmatch.cs.princeton.edu/

1. Introduction

Matching 3D geometry has a long history starting in the early days of computer graphics and vision. With the rise of commodity range sensing technologies (e.g., the Microsoft Kinect), this research has become paramount to many applications including 3D scene reconstruction, localization, and object retrieval. However, establishing correspondences between local geometric features in low-resolution, noisy, and partial 3D data is still a challenging task. While there are a wide range of low-level hand-crafted geometric feature descriptors that can be used for this task, they are mostly based on signatures derived from histograms over static geometric properties [18, 19, 25], which are often unstable or inconsistent in real-world partial scans. For instance, state-of-the-art 3D reconstruction methods [6] note the unsatisfactory performance of current geometric matching algorithms for the registration of fused fragments, thus requiring significant algorithmic effort to deal with outliers and to establish global correspondences.

In response to these difficulties, we introduce 3DMatch, a data-driven model that learns a robust, local geometric feature descriptor specifically for partial 3D data. We propose a unified 3D convolutional neural network (ConvNet) with an architecture particularly designed for matching local geometry. As shown in Figure 3, our model jointly

learns a geometric feature representation and an associated metric function from a large collection 3D volume correspondences gathered from real-world scanning data.

To enable the use of 3D ConvNets, we encode 3D geometry using a truncated distance function (TDF). his regular structure is amenable to 3D convolution and other kernel operations used in the ConvNet, thus allowing the model to learn geometric shape representations directly in 3D space. Moreover, this form of encoding allowed us to aggregate the information from multiple depth frames to reduce sensor noise, and is aligned with common representations for 3D reconstruction [8]. In order to capture meaningful signal within the large space of geometric shapes, we train our descriptor only on the local geometry around 3D Harris keypoints.

Training the 3D ConvNets requires labels for key point correspondences. We propose to use of existing RGB-D reconstruction algorithms to obtain good 3D alignment results in order to generate the training labels. From the reconstruction of these scans, we know the exact world-space locations for feature points from different camera views, which allows us to automatically generate a large amount of ground truth correspondences without manual annotation. As a result, we obtain globally-consistent keypoint pairs, and we are able to correlate their respective partial geometries with each other. Since each camera view yields a different occlusion pattern, this provides partial-to-partial ground truth matches.

A key insight of this process is that we can escape the feature quality used in the training process (i.e., for 3D reconstruction). For instance, a training sequence may only have a small number of matched features. But once it is reconstructed successfully, it can generate many more keypoint correspondences; even for the areas where the original feature matching fails.

Our central contribution is the data-driven 3D keypoint descriptor (3DMatch) for robustly matching local geometry. We demonstrate the performance of our method in a thorough evaluation against state-of-the-art geometry matching and fragment alignment methods, where we outperform existing approaches by a significant margin. For evaluation and future study, we construct several benchmarks for matching local 3D geometry captured from real-world RGB-D scanning data. Furthermore, we demonstrate the benefits of our robust descriptor on a high-level application; even with a simple, unoptimized, and geometry-only sparse bundle adjustment formulation, we are able to obtain globally-consistent 3D reconstructions in challenging 3D environments (see Figure 1).

2. Related Work

Handcrafted Geometric Descriptors Many handcrafted geometric descriptors have been proposed in the last decade. Examples include Spin Images [18], Geometry Histograms [12], and Signatures of Histograms [29]. Many of these descriptors are now available in a unified framework, the Point Cloud Library [3]. A popular state-of-the-art method in PCL is point feature histograms (PFH), which are based on the relationship of all surface normals and curvature estimates [26]. Aiger et al. [2] show that pairs of fragments can be aligned by finding sets of four congruent points (4PCS); Mellado et al. [22] extended this work to Super 4PCS.

While these methods have made significant progress, they still struggle to handle noisy, low-resolution, and specifically incomplete data from commodity range sensors. Since handcrafted geometric descriptors are mostly based on signatures derived from histograms over static geometric properties, they are often unstable or inconsistent in partial scans. To handle the low precision of state-of-the-art geometric matching methods, Choi et al. [6] designed a method for 3D reconstruction to specifically handle mismatched geometry in a robust optimization procedure. The goal of our work is to provide a new type of local geometric feature that can provide much more robust and accurate matching results.

2D Feature Learning for Images The recent availability of large-scale labeled image data has opened up new opportunities to use data-driven approaches for designing 2D image descriptors. For instance, Trzcinski et al. [31] and Simonyan et al. [28] learn a non-linearity mapping from intensity patches to image feature descriptors under a pre-defined metric such as the Euclidean or Mahalanobis distance. In addition, Jia and Darrell [17], and Jain et al. [16] demonstrate that feature learning can be extended further by learning a feature comparison metric in addition to the feature descriptor. More recently, it has been proposed to use a deep convolutional neutral network to jointly learn the descriptor and the comparison metric for local 2D RGB patches [38, 13]. Inspired by the success of these 2D preprints, we design a similar network structure with a unified feature and metric learning architecture that operates over 3D data to learn 3D keypoint correspondences for matching local geometry. The main objective of the network is to regress an end-to-end, keypoint-to-feature mapping and feature metric that can robustly establish geometric correspondences between viewpoint variant, noisy, and partial 3D scanning data.

3D Feature Learning for CAD Models Aside from learning 2D features, there has also been rapid progress in the use of deep convolutional neural networks on three dimensional data. For example, 3D ShapeNets [34] introduced 3D deep learning for modeling 3D shapes, and demonstrated that powerful 3D features can be learned from

a large collection of 3D CAD models. Additionally, several recent works [21, 11] also extract deep learning features from 3D data for the task of object retrieval and classification for CAD models. While these works are inspiring, their focus is centered on extracting global features from complete 3D CAD models instead of local geometric features from real-world RGB-D scanning data, the latter of which is the goal of our 3DMatch descriptor.

3. Geometric Representation

The goal of geometric matching is to establish robust correspondences between 'fragments' of 3D geometry. While there are many possible ways to obtain and represent geometry, we focus on a realistic case where our primary source of input data is made up of depth frames collected from (but not limited to) commodity range sensors, such as the Microsoft Kinect or Asus Xtion Pro. By default, these depth maps are not aligned to any global coordinate system, and for our purposes, we do not use any color information – focusing only on the information available from the depth channels.

Our 3DMatch descriptor performs geometric keypoint matching by comparing uniformly-sampled distance fields surrounding points of interest. In Section 3.1, we describe how we convert scanning data into geometric fragments and how we represent these fragments as distance fields. In Section 3.2, we describe how we extract points of interest from these distance fields.

3.1. Fragment Volume Representation

Our method operates on geometric fragments, each composed of N consecutive depth frames fused together into a distance field. When $N = 1$, each fragment contains only a single frame. As N becomes larger, each fragment is able to integrate more information from multiple depth frames to smooth sensor noise and to increase the fragment's field of view so that it contains more geometric information. However, we also keep N small enough so that a standard local alignment method can easily provide high-quality fragments without accumulating too much drift error. In our case, we align all frames in a fragment using a dense point-to-point and point-to-plane iterative closest-point method (ICP) [4] with $N = (30, 50)$. If we cannot find a valid intra-fragment alignment, we discard the fragment due to a lack of geometric features (e.g., a planar wall). With the relative transformations between the frames of a fragment obtained from local alignment, we use volumetric fusion [8] to fuse the frames' depth data into a shared voxel volume that is anchored to the first camera frame of the fragment. This provides us with a truncated signed distance field (TSDF), which is the same representation that is generated by modern real-time reconstruction pipelines.

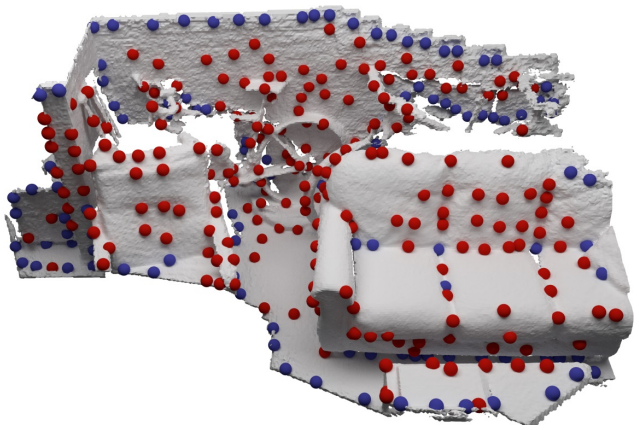

Figure 2: We use Harris corner responses to detect keypoints. We discard keypoints whose local regions are not observed by enough frames (blue). Only well-observed keypoints (red) and their local regions are used to train 3DMatch.

Using a uniformly-sampled distance field representation for encoding 3D data has significant advantages: the discrete but regular structure allows for kernel operations, such as 3D convolutions, to be performed over the data, which is paving the way for gradient-capturing kernels that can be regressed by deep learning architectures. Moreover, compared to a 2D encoding of shape using depth patches, a voxelized 3D representation preserves real-world spatial scale information, is invariant to projective scaling, and naturally provides grounds for learning the invariance of spatial rotation.

For our purposes, we ignore the sign of the TSDF, and detect and compute features on the truncated distance field (TDF). Since we are training a gradient-sensitive convolutional neural network architecture (described in Section 4), we want to minimize the confusion of the lower level kernels by constraining the heaviest gradients to be exclusively located over the detected surface areas. By removing the sign of the TSDF, we no longer distinguish between unobserved and observed free space. Consequently, the highest distance field gradients are now concentrated around surfaces rather than in the shadow boundaries of the camera viewing frustum.

3.2. Keypoint Volume Representation

Once we have generated a fragment and its TDF, we detect points of interest within the fragment and extract their local TDF regions. Our goal is to focus the descriptor on geometrically discriminative regions of a the environment. We find that the granularity of regions around keypoints (radius of $\approx$ 15cm) is a reasonable choice for typical indoor environments: it is local enough to guarantee a certain amount of contextual coverage while also being discrimina-

tive enough for capturing sufficient geometric detail.

Following an approach used in 3D object retrieval [20], we use 3D Harris corner responses to determine keypoint locations [15]. For each voxel adjacent to the mesh surface, we determine the covariance matrix $\mathcal{C}$ of its neighborhood normals n_i, given by the gradient function of the TDF; the corner responses r_i are then given by $r_i = det(\mathcal{C}_i) - 0.04 \cdot trace(\mathcal{C}_i) \cdot trace(\mathcal{C}_i)$. On the set of all corner responses, we perform a non-maximum suppression to reduce the number of samples and apply an iterative adjustment to move the remaining samples to their local stable positions. If the information is available, we also filter out keypoints whose local regions are not observed by enough frames. We then use the remaining sparse set of keypoints and their local TDF volumes as input to our descriptor. Figure 2 visualizes the detected keypoints on a fragment. In terms of speed, the complete Harris keypoint detection and extraction process over a $512 \times 512 \times 1024$ TDF voxel volume of a geometric fragment takes only a few seconds using a single thread on an Intel Xeon Core E5-2699 CPU clocked at 2.3 GHz.

4. Correspondence Generation for Training

The most effective deep learning algorithms are supervised, which implies that enormous amounts of training data with ground truth labels is required. While it is easy to obtain data, gathering the associated labels typically requires a significant amount of manual effort (e.g., object annotations for image recognition on ImageNet [10]). Since our aim is to learn correspondences between keypoints, manual ground truth annotations would involve labeling millions of keypoint pairs between geometric fragments [37]. Fortunately, existing RGB-D reconstruction algorithms are mature enough to accurately align and fuse depth frames, even when with weak baseline features. By leveraging the reconstruction results from [9], we can automatically generate point-to-point correspondence labels on a large scale and without manual effort.

To this end, we use RGB-D scans where the same scene is recorded multiple times with different camera trajectories and varying viewpoints. In order to obtain globally-consistent reconstructions, we utilize state-of-the-art sparse and dense bundle adjustment that consider on both RGB and depth data. We then sample TDF voxel volume pairs between different views to generate labeled training data, which allows us to train our descriptor and metric network. A key component of our method is that we can escape the feature quality used in the original reconstruction process. For instance, a training sequence may be reconstructed with only a few, weak features. But once it is reconstructed successfully, it provides many more keypoint correspondences, even for the areas where the original feature matching fails.

In the context of this work, we generate training labels from the following real-world indoor RGB-D data sets:

The Microsoft 7-Scenes Dataset contains a collection of tracked RGB-D frames over 7 different scenes [27]. 'Ground truth' tracking is performed using ICP and frame-to-model alignment with respect to a dense 3D reconstruction represented by a truncated signed distance volume obtained using KinectFusion [23]. Our training split uses the sequences from six of these scenes, while the testing split contains the seventh scene.

The Analysis-by-Synthesis Dataset contains a collection of camera-tracked RGB-D frames over 12 different scenes [32]. The dataset is similar to the 7-scenes; however, the reconstruction is globally aligned with [9]. Our training split uses the sequences from 10 of these scenes, while the testing split contains the other two scenes.

These datasets were chosen because they offer various viewpoints of the same scene from a variety of different angles. This provides realistic samples of the distribution of possible scanning conditions for any given keypoint. In other words, since local keypoint volumes are oriented with respect to the camera frames, the training data from these datasets contain correspondences between keypoints whose local volumes have properties that reflect viewpoint changes and occlusions. The main idea is to train the descriptor to generalize over this kind of complexity, which is most commonly found in real-world partial 3D data.

At 30 frames per fused fragment, our collective fragment dataset features over 1,600 fragments for training and over 500 fragments for validation. A typical fragment contains 200 to 500 Harris keypoints. Using the pseudo ground truth extrinsics provided from the reconstruction systems of the training datasets, we establish ground truth correspondences between keypoints of the same scene by relating their world coordinates. Pairs of keypoints within 5cm of each other are labeled as matches, while keypoint pairs further than 5cm away from each other are labeled as non-matches. For our purposes, the 5cm search radius threshold allows the algorithm to tolerate minor alignment errors from the provided ground truth extrinsics of the datasets, while also remaining flexible to minor translation differences between Harris keypoints. Our total pool of training data (a pair of keypoint volumes per data point) contains millions of unique volume comparisons.

Although it is feasible to train a descriptor on correspondences between keypoints that can be situated at any location near the detected surface regions, we specifically chose to train on Harris keypoints (see Section 3.2) to focus the descriptor on learning over features that have an adequate amount of discriminative geometry. Despite this restriction in the training data, we show through experiments (see Section 6.1) that our model is capable of generalizing to accurately determine correspondences between randomly sampled surface points.

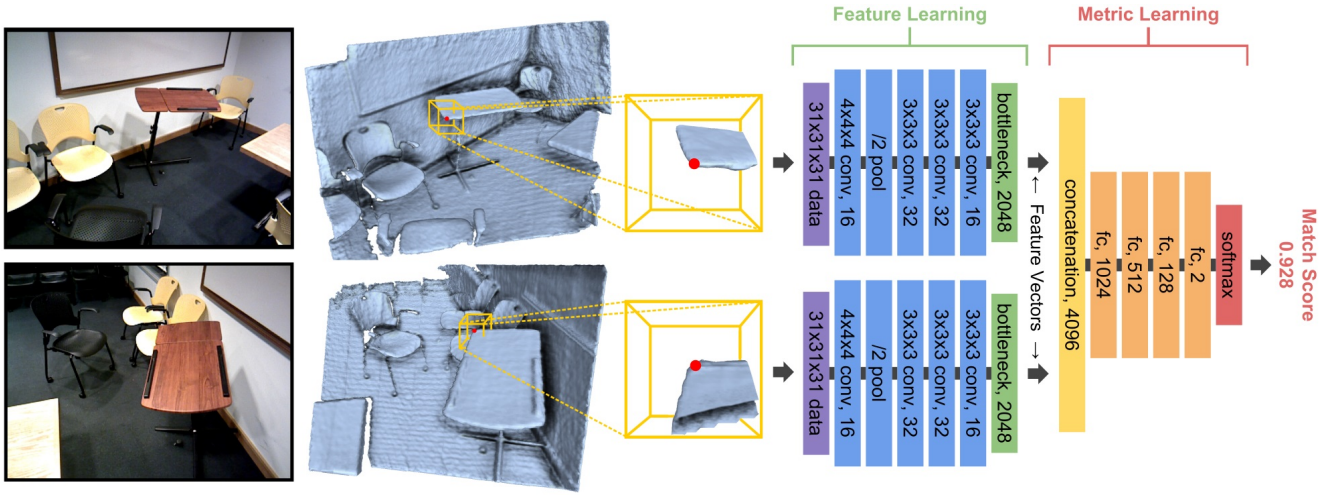

Figure 3: Matching two keypoints from different geometric fragments using 3DMatch. For each keypoint, its local 3D region is extracted from the TDF volume of its fused fragment. This 3D region is then passed through the the feature network to return a feature vector (green). To compare two keypoints to each other, their corresponding feature vectors are concatenated and fed to the metric network which returns a similarity score.

5. Geometric Matching Network

Inspired by the recent success of learning feature representations and metric functions on 2D [13], we design our 3DMatch descriptor as a unified deep neural network architecture with two core components: a *feature network* that maps a local 3D TDF volume to a high-dimensional feature representation using a 3D ConvNet, and a *metric network* that maps pairs of features to a similarity value through a set of fully connected inner product layers.

Figure 3 visualizes our network architecture. First, the local TDF volume around each query keypoint is cropped from its geometric fragment. These volumes are then independently passed through a feature network, which maps them to a feature descriptor containing 2048 elements. Pairs of these feature vectors are then concatenated and fed through the metric network, which ends with a similarity score that classifies the two points as either matching or not matching. As we will show in Section 6, allowing the network to optimize over different intermediate feature representations produces features that are significantly more robust than methods that use a fixed representation. Our joint feature and metric network is able to automatically learn the best distance function from data without limiting itself to certain similarity measure such as $\mathcal{L}_1$ or $\mathcal{L}_2$, which is the approach taken by the Siamese network [5] [7]. Sample output of the network is visualized in Figure 8.

5.1. Network Architecture

Geometric feature network The feature network of the 3DMatchmodel constitutes a descriptor function that maps a keypoint's 3D local region to a concise feature representation. In our case, the radius of the local regions for all keypoints are set at 15cm (see Section 3.2), so the input to our feature network is structured as a $31 \times 31 \times 31$ voxel TDF volume (voxel size = 1cm with a standard truncation distance of 5cm), while the feature representation is a 2048 dimensional feature vector. Following several similar ConvNet architecture preprints for training over 2D local image patches [38, 13], the feature network consists of several convolutional layers with ReLU non-linearity and a single pooling layer. We include pooling to benefit from the response filtering properties of max pooling; but since the dimensions of the initial input volume are small, we only include only one layer of pooling to avoid a substantial loss of information. The detailed kernel sizes and number of filters are shown in Figure 3. The last fully connected layer of the feature network determines the dimensionality of the feature representation and prevents the network from overfitting [13]. Following prior work on feature learning for 2D local patches [5, 7], we ensure that the two feature network towers maintain identical parameters by sharing all updates between the networks. This maintains a globally consistent keypoint-to-feature encoding.

Metric network The metric network of 3DMatchforms a non-linear matching function that compares two feature representations and determines whether or not the two relevant keypoints correspond to each other. The input to this network is the concatenation of two feature vectors, while the output is a single confidence value between 0 and 1 that measures similarity between the keypoints, where 1 is a 'match' and 0 is a 'non-match'. Our metric network is made up of several fully-connected layers with ReLU non-linearity. The last layer uses Softmax, and its two values

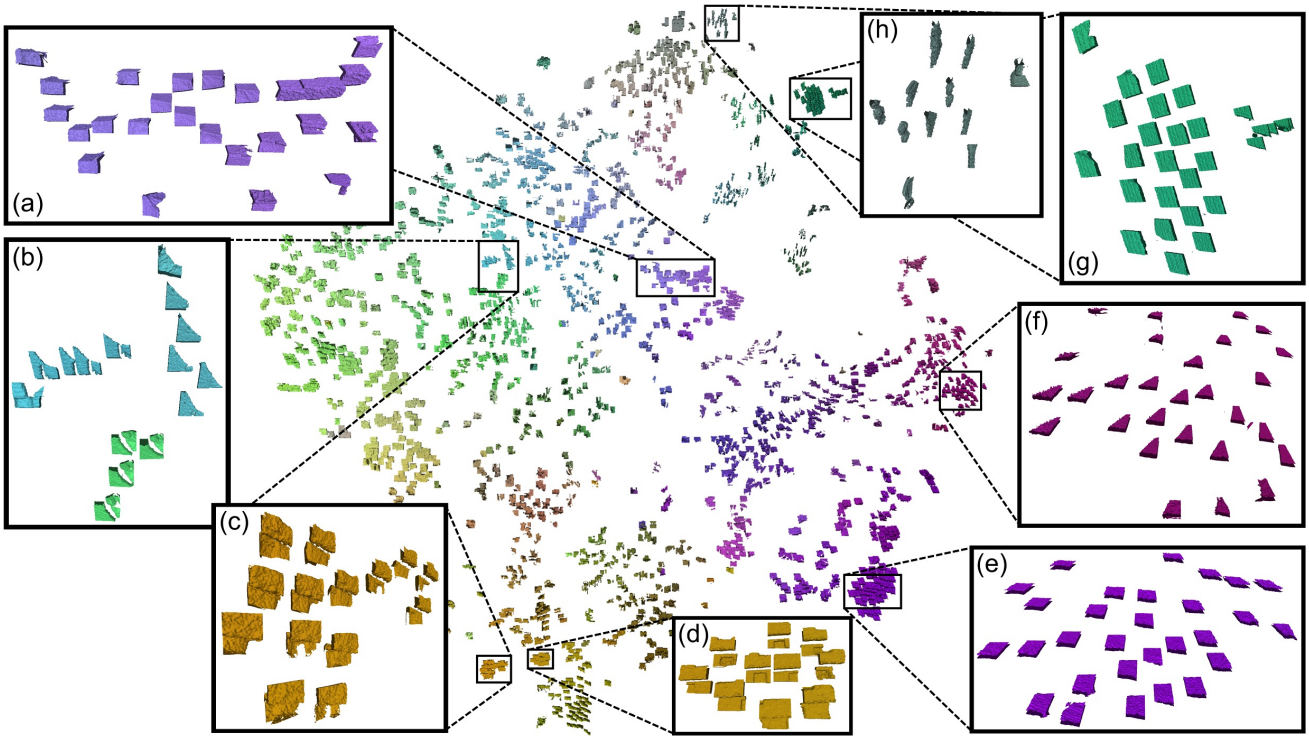

Figure 4: A feature embedding of local keypoint regions in a test scene, visualized using t-SNE. The learned features are highly predictive of geometric structure as well as local context. This embedding suggests that our 3DMatch network is able to coherently group 3D local keypoint volumes based on properties such as edges (a,f), planes (e), corners (c,d), and other local geometric structures (g, b, h) in the face of noisy and partial data.

represent the networks estimate of probability that the two features match and do not match, respectively.

Matching cost Using the training keypoint volume pairs and their associated correspondence labels acquired from the data preparation described in Section 4, we minimize the cross-entropy error

$$E = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(\hat{y}_i) + (1-y_i) log(1 - log(\hat{y}_i)] \quad (1)$$

over a training set of ground truth correspondences using stochastic gradient descent (SGD). Here, y_i is the binary label ('match' or 'non-match') for input x_i and $\hat{y}_i$ is the probability estimate from the network output by the Softmax layer.

To highlight the metric network's contribution to matching performance, we additionally experimented with an architecture where the metric network is replaced with a single contrastive loss layer that compares the bottlenecked features using Euclidean distance (L2). The absence of the metric network reduced keypoint matching performance on our benchmarks, shown in Figures 5 and 6. However, it continues to outperform the best hand-crafted feature methods. In all subsequent experiments, we used the 3DMatch's model that includes both the feature network and the metric network.

5.2. Feature Visualization

To help better understand and examine the kind of information that the neural network captures, we visualize the descriptors learned by our feature network using the t-SNE algorithm [33]. Specifically, we randomly extract 2,000 keypoint volumes from one scene of the test set and find a 2-dimensional embedding of their 2048-dimensional feature vectors. Figure 4 visualizes this embedding. For each keypoint TDF volume, we generate its mesh, and position it in its exact location in the embedding. Additionally, each keypoint volume mesh is colored with the normalized 3-dimensional embedding of its feature vector. The overall layout of the embedding suggests that the feature network is able to coherently cluster different types of local 3D geometric structures such as edges, planes, and corners.

5.3. Implementation Details

We implement our network architecture in Marvin [36], a deep learning framework that supports N-dimensional convolutional neural networks. To train the network, we randomly initialize all layers by drawing weights from a

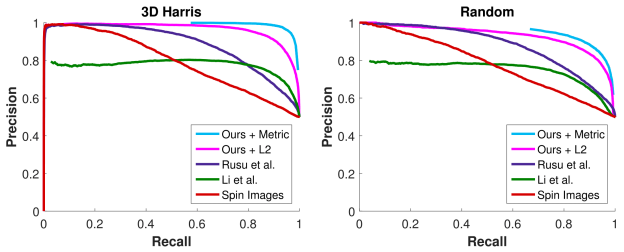

Figure 5: Precision and recall curves over match confidence thresholds (or distance thresholds) on a set of keypoint volume pairs with ground truth correspondence labels.

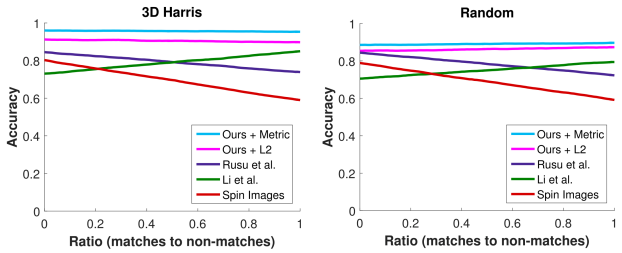

Figure 6: Accuracy of keypoint matching when varying the ratio of matches to non-matches in the testing set.

zero-mean Gaussian with standard deviation 0.01 and initialize biases to 0. During training, we set the mini-batch size to 256 keypoint volume pairs per iteration. For each of the training mini-batches, we regulate training biases by balancing the number of matching volume pairs with the number of non-matching volume pairs to a 1:1 ratio. The base learning rate is set as 0.01, and the learning rate is reduced by a factor of 0.99 every 2,000 iterations. We run SGD for 1.3 million iterations, with a momentum of 0.9 and a parameter decay of 0.0005. These learning parameters were empirically selected to optimize over validation accuracy. Due to the large variation in training data, our network requires a substantial number of training iterations for convergence.

6. Evaluation

We demonstrate the effectiveness of the 3DMatch descriptor on several real-world and synthetic tasks. In Section 6.1 and 6.2, we present several test benchmarks to evaluate the performance 3DMatch against other 3D geometric descriptors on the task of keypoint correspondence detection. In Section 6.3, we combine 3DMatch with a standard RANSAC-based alignment approach, and compare it to various other geometric registration methods on the task of fragment-to-fragment alignment and loop closure detection, as proposed for indoor scene reconstruction algorithms [6]. In both quantitative experiments, we show substantial improvements over state-of-the-art methods. And finally, to show the robustness of our descriptor in Section 6.4, we demonstrate a complete scene reconstruction pipeline that uses 3DMatch for sparse fragment alignment and loop closure.

All quantitative and qualitative evaluations performed in the subsequent sections test exclusively on data from scenes that were *not* part of training and that the 3DMatchmodel has never seen before. This includes the testing split of the datasets described in Section 4, as well as a number of different scenes from the SUN3D dataset [35]. The 3DMatchmodel has been trained with over 332 million volume comparisons.

Test condition	Spin Images	Rusu et al. (FPFH)	Li et al. (KDD)	3DMatch (ours)
3D Harris (%)	55.9	67.4	74.2	**95.6**
Random (%)	55.6	66.8	63.7	**87.5**

Table 1: Accuracy (at 95% recall) of geometric descriptors on the keypoint matching task (balanced ratio between matches and non-matches). Testing on 3D Harris is more accurate for all methods because keypoints have significantly less geometric ambiguity.

6.1. Keypoint Matching Evaluation

We directly evaluate the quality of a 3D geometric descriptor by testing its ability to classify whether two points from different scans of the same environment correspond to each other. Using environments never seen before by 3DMatch(the testing split of the datasets mentioned in Section 4), we construct an evaluation dataset by sampling a small number of keypoint volume pairs and their ground truth correspondence labels (a binary label for 'match' or 'non-match') in a way that is similar to what is done during training for the descriptor. The evaluation dataset consists of $20,000$ pairs of points and their respective local volumes, with a balanced 1:1 ratio between matches and non-matches. The keypoint pairs are formed such that they do not come from the same scans of the environment. We construct two such evaluation datasets: **Harris** and **Random**. In **Harris**, the points are chosen randomly from the set of Harris keypoints with at least one other correspondence, and in **Random** they are chosen at random from all surface points of the fragments.

We ran 3DMatch along with several other geometric descriptors on these two benchmarks. For spin images [18] and the fast point feature histograms [25], we use the implementation provided in the Point Cloud Library, tuning the algorithm's parameters specifically for the benchmark. These methods operate directly on the point cloud of the fragment, and are not able to incorporate additional information from the signed distance field. For Li et al. [20], we use an implementation provided by the authors. Their method operates on the signed distance field and uses a brute-force approach to account for varying rotations.

Figure 5 shows the precision and recall performance of the various geometric descriptors over a match confidence threshold (or distance threshold for some methods). 3DMatchis by far the most robust descriptor, and retains a precision of 82.62% at 95% recall.

In Figure 6, we fix the matching threshold for each algorithm at the optimal accuracy for a balanced 50:50 ratio between matching to non-matching volume comparisons, then vary this ratio in the evaluation dataset (via random sampling) and assess how the general accuracy of the descriptor changes with respect to this ratio. This experiment illustrates how hand-crafted descriptors are naturally weaker at establishing matching correspondences than non-matching correspondences in the face of noisy and partial data. 3DMatchretains consistent accuracy across this condition, suggesting that the probability estimate computed in the final layer of the metric network is an effective indication of absolute match strength for real data. Table 1 quantitatively shows the highest accuracy results obtained under this evaluation. In all conditions, 3DMatch has significantly higher accuracy than other geometric descriptor approaches.

6.2. Keypoint Retrieval in Scenes

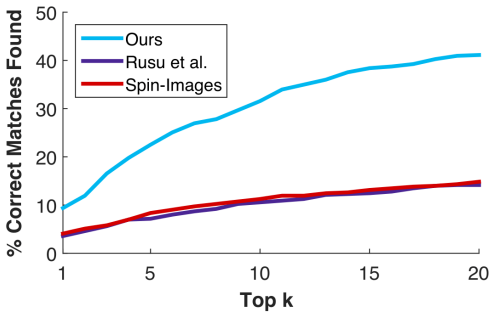

Figure 7: Keypoint retrieval in scenes. The y-axis labels the percentage of keypoints in the query sequence that can successfully find a correct correspondence from all the keypoints in the sample sequence within the top k matches under each geometric descriptor.

To evaluate the performance of 3DMatchagainst other geometric descriptors on a more realistic distribution of correspondences, we survey the descriptors' ability to match a 3D keypoint to its ground truth correspondence from different scans of the same large environment. Unlike the evaluation in Section 6.1, this query has only several positive correspondences among thousands of negative correspondences. Specifically, we fuse fragments from two RGB-D frame sequences of the same scene from our test set of the 7-scenes dataset. We call these two sequences the *query sequence* and the *sample sequence*, respectively. Using the ground-truth extrinsics from the dataset, both sequences are related to each other in world space (this information is used only for evaluation). Within each fragment of both sequences, we compute a set of 3D Harris keypoints and extract their local TDF volumes. All Harris keypoints within 5cm of each other in world space are considered to be ground truth correspondences. Since each keypoint and its local TDF volume are anchored with respect to the camera coordinate frame of the scans, the ground truth matching keypoint volumes are oriented according to a variety of different camera viewing angles.

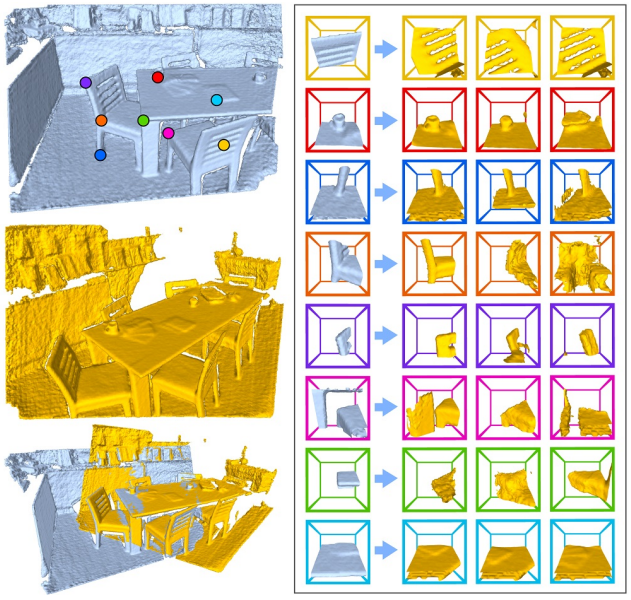

Figure 8: Sample Harris keypoints (middle column) detected in the blue fragment, and their top-3 matching Harris keypoints from the yellow fragment. Top-k is based on the local descriptor similarity scores computed using 3DMatch.

For each keypoint in the query sequence, we examine how well the descriptor can find a ground truth correspondence from the sample sequence within the top-k of similarity scores (or closest in feature distances). For example, when $k = 1$, we measure the % of keypoints in the query sequence whose comparison to a ground truth correspondence (vs. all other sample keypoints) returns the highest similarity score. Figure 7 shows the percentage of query keypoints that can successfully find a correct correspondence from the sample sequence within the top k matches, where k varies from 1 to 20. At $k = 5$, with 3DMatchover 20% of keypoints are able to find the correct match compared to only 8% with the FPFH descriptor. As we will show in Section 6.3, this improved recall is important for algorithms such as RANSAC-based alignment [25]. Figure 8 visualizes several examples of top-k keypoint volumes matches found in this way in between two fragments from different scans.

6.3. Fragment Alignment Evaluation

One application of geometric descriptors is the registration of two fragments from a 3D scan, a subcomponent of some 3D reconstruction approaches [6]. The goal of surface registration is to determine if two fragments (P_i, P_j) overlap, and if they do provide an estimate of their relative geometric transformation T_{ij}. To focus our evaluation on a registration method's ability to detect and align loop closures, we only consider pairs of fragments that are not time-based adjacent to each other. We start by showing the results of registration over an existing synthetic dataset designed for indoor scene reconstruction, and then demonstrate the results on fragments scanned from real-world environments. We show that a standard RANSAC-based registration approach using 3DMatch significantly outperforms other state-of-the-art geometric registration methods for loop closure detection and fragment-to-fragment alignment.

Following the evaluation scheme introduced by Choi et al. [6], we evaluate the accuracy of alignment for both the synthetic and real datasets by measuring the effect of T_{ij} on the ground-truth correspondences K_{ij}^* between P_i and P_j. The transformation is accepted if it brings the ground-truth correspondence pairs into alignment. In other words, T_{ij} is considered a true positive if the RMSE of the ground-truth correspondences is below a threshold τ

$$\frac{1}{|K_{ij}^*|} \sum_{(p^*,q^*)\in K_{ij}^*} ||T_{ij}p^* - q^*||^2 < \tau^2 \qquad (2)$$

where p^* and q^* are the ground-truth correspondence points in P_i and P_j, respectively. Choi et al. [6] uses a fairly liberal threshold at $\tau = 0.2$. Lower values of τ emphasize alignment quality in a more fine-grain manner.

We perform registration between fused fragments using 3DMatch as the descriptor in a RANSAC-based approach, following the PCL implementation of Rusu et al. [25]. However, instead of computing features over a subsampled point cloud, we compute features over the set of 3D Harris keypoints. Specifically, we first map each 3D Harris keypoint to its top k strongest correspondences with respect to 3DMatch's matching scores. Then for each RANSAC iteration, we randomly choose 3 correspondences to estimate a rigid transformation. The final transformation is the one with the highest number of inlier correspondences — those whose keypoint pairs are within 5cm of each other after alignment and has a 3DMatch matching score of at least 0.5. We reject the alignment if the final RANSAC transformation has less than 15 inlier correspondences.

Synthetic dataset. The work of Choi et al. [6] evaluates geometric registration using the synthetic ICL-NUIM dataset [14]. Figure 9 shows the results of this evaluation

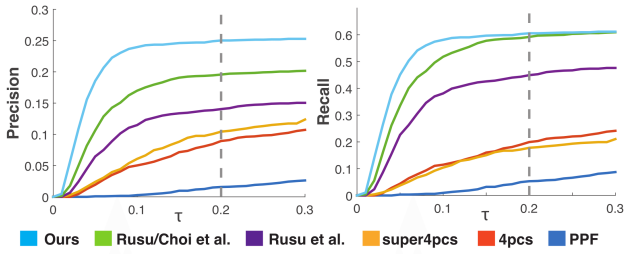

Figure 9: Precision and recall vs. matching tolerance τ between different methods on the fragment-to-fragment geometric registration benchmark from Choi et al. (the dotted line marks the RMSE τ set to compute their reported PR numbers).

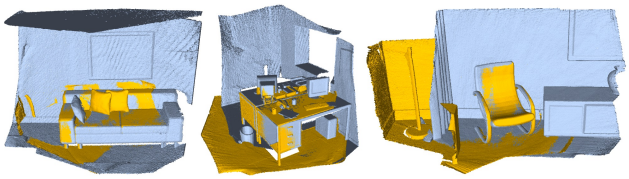

Figure 10: Sample false positive alignments between synthetic fragments from the augmented ICL-NUIM dataset. This highlights the synthetic nature of the fragments from the augmented ICL-NUIM dataset.

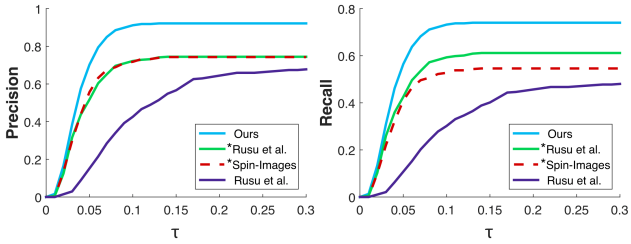

Figure 11: Precision and recall vs. matching tolerance τ between different methods on a real–world dataset. Methods with an asterisk are performed using RANSAC on Harris keypoints.

on 3DMatch and the geometric registration methods presented in Choi et al. [6]. 3DMatch provides consistently stronger alignment results, especially when τ is small, indicating that 3DMatch is more accurate at a fine-grained level than the other methods. Using Choi et al.'s threshold at $\tau = 0.2$, 3DMatch surpasses other registration methods with an average precision of 25.0% and average recall of 60.6%. The precision of all methods is low due to the minimal geometric complexity of and large set of duplicated geometry found within the scenes of the synthetic dataset; see Figure 10 for examples.

Real-world dataset. We further evaluate the performance of 3DMatchagainst other methods for registering fragments from real-world scans, which have greater geometric com-

plexities compared to the synthetic scenes of the ICL-NUIM dataset. We form fragments from the test datasets (Section 4) and attempt to align non-consecutive pairs of fragments in each scene. Figure 11 shows the precision and recall results against the matching tolerance threshold τ from Equation 2. For Rusu et al., we tuned an implementation in the Point Cloud Library (PCL), where the run-time of each fragment comparison was restricted to be approximately 10s on a single core of an Intel i7-4820K. We also evaluate over the registration results using Rusu et al. and Spin-Images as local descriptors for the same keypoint-based pipeline used by 3DMatchregistration method. Overall, 3DMatch performs the best on real-world data; at $\tau = 0.2$, 3DMatch has a precision of 92.2% and a recall of 74.0%.

6.4. Scene Reconstruction using 3DMatch

A core challenge in scene reconstruction is loop closure, where a correspondence needs to be formed between the same location when viewed from significantly different perspectives. Both color and depth information provide different channels of information that can be used to detect these long-range correspondences. However, color-based descriptors, such as SIFT, often fail to find correct correspondences when there are wide-baseline viewpoint changes or drastic lighting differences. Figure 12 shows some challenging loop closure cases that are known to be difficult for color-based descriptors. We show that our 3D descriptor is able to correctly align the fragments of these loop closure instances using geometric information.

Our scene reconstruction procedure is based on a standard sparse bundle adjustment pipeline [30, 1]. This pipeline is widely used to obtain global alignment from a set of RGB-D frames, generating globally-consistent 3D reconstructions. The key idea is to minimize the distance between a sparse set of matched feature points, whose positions are given in camera space of the corresponding RGB-D frames. Traditionally, sparse RGB features, such as SIFT or SURF, are used to establish feature matches between frames. With our 3DMatch descriptor, we are able to establish feature matches and formulate the bundle adjustment problem purely on geometric feature pairs:

$$\text{argmin}_{\mathbf{T}} \sum_{i,j}^{\text{frags.}} \sum_{k}^{\text{corresp.}} ||T_i p_{ik} - T_j p_{jk}||_2^2$$

where T_i, and T_j are poses for fragments i and j while p_{ik} and p_{jk} represent the corresponding feature pairs between the two fragments. These correspondences are obtained from aligning all pairs of fragments in the sequence using the algorithm described in Section 6.3.

In order to demonstrate the robustness of our descriptor, we use this very simple formulation for reconstruction, which results in a trivial non-linear least squares solve. Although our reconstruction pipeline is less powerful than the robust optimization presented by Choi et al. [6], we are nevertheless able to generate globally-consistent alignments in challenging scenes using geometric information. For the final reconstruction, we fuse all depth frames using a volumetric fusion implementation [24] to generate a dense 3D reconstruction as shown in Figure 1.

In addition, we show that our descriptor can be combined with sparse RGB features, which provide additional sparse correspondences to support the reconstruction process, especially when there is insufficient geometric information in the fragments. For instance, as shown in the reconstruction results in Figure 13, combining correspondences from both SIFT and 3DMatch significantly improves the alignment quality.

7. Conclusion and Future Work

We have presented 3DMatch, a geometric descriptor trained from real-world data, and demonstrated its effectiveness at keypoint matching, fragment alignment, and scene reconstruction. We plan to make all source code, pre-trained models, and evaluation benchmarks publicly available.

Paralleling research in RGB image descriptors, we believe that the trend away from hand-tailored, histogram-based descriptors and towards data-driven representations will continue. The growth of both 2D and 3D mapping systems will create larger databases of scenes annotated with rich correspondences. These correspondences link the same spatial location across different sensors, times, and lighting conditions. Data-driven architectures such as 3DMatch will continue to benefit from this data, enabling the construction of powerful descriptors that are robust to these variations.

Acknowledgement This work is supported by the NSF/Intel VEC program and Google Faculty Award (Jianxiong Xiao). Shuran Song is supported by a Facebook fellowship and Matthias Nießner is a member of the Max Planck Center for Visual Computing and Communications (MPC-VCC). We also gratefully acknowledge the support from NVIDIA and Intel for hardware donations.

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM Transactions on Graphics (TOG)*, volume 27, page 85. ACM, 2008.

[3] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze.

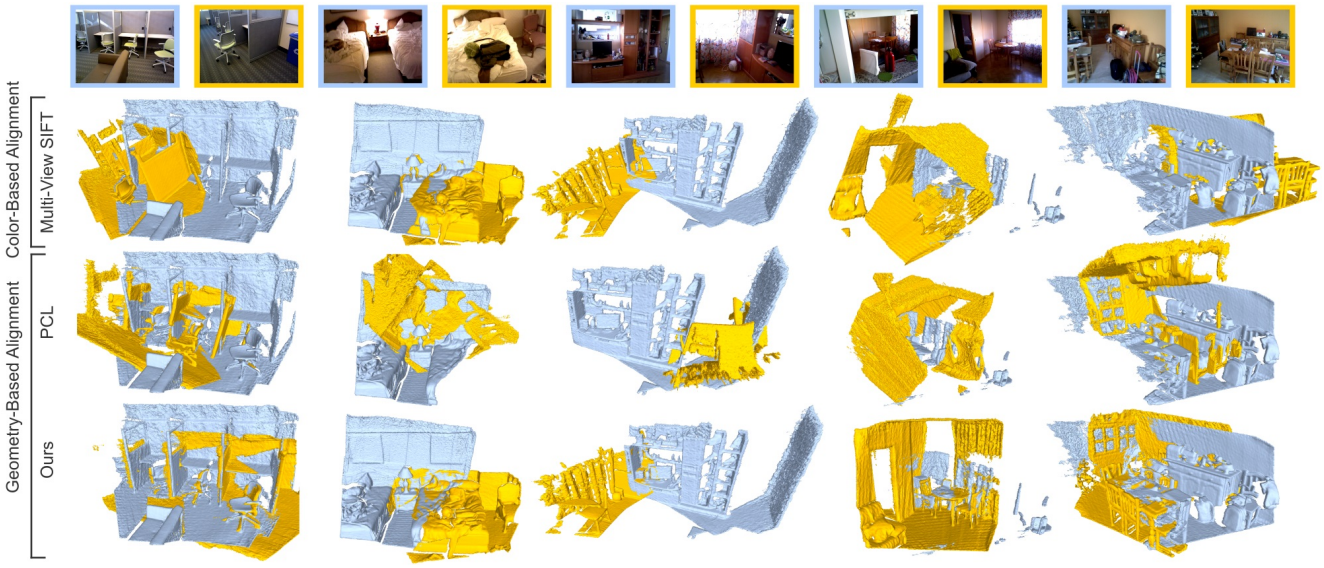

Figure 12: Examples of challenging loop closure cases from the SUN3D dataset, featuring cases with substantial noise, incomplete data, and drastic viewpoint changes. Our 3D descriptor (bottom row) with simple RANSAC described in 6.3 is able to align the geometry accurately.

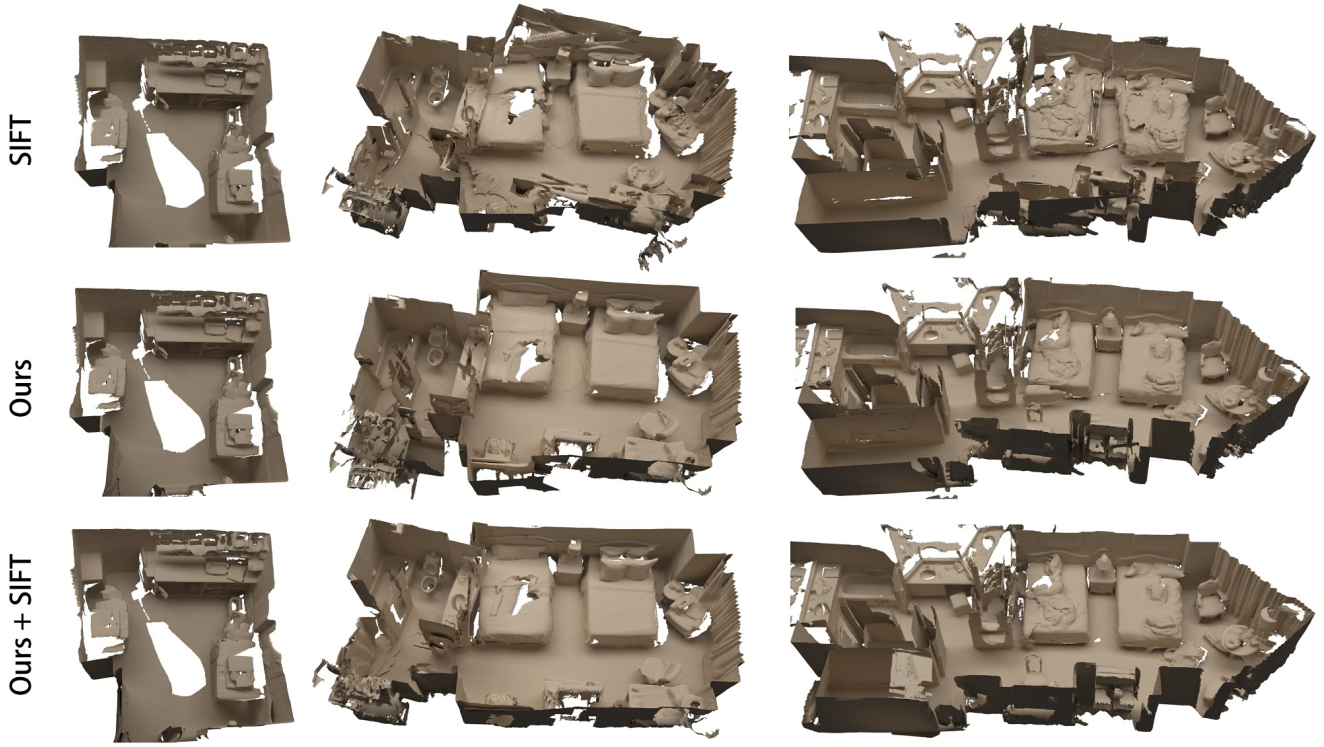

Figure 13: We show 3D reconstruction results with global pose alignment obtained with a simple sparse bundle adjustment formulation using correspondences from SIFT features, our local geometry descriptor, and both 3DMatch and SIFT.

Point cloud library. *IEEE Robotics & Automation Magazine*, 1070(9932/12), 2012.

[4] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[5] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.

[6] S. Choi, Q.-Y. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565, 2015.

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[8] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

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

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[11] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong. 3d deep shape descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2319–2328, 2015.

[12] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Computer Vision-ECCV 2004*, pages 224–237. Springer, 2004.

[13] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015.

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

[15] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[16] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon. Metric and kernel learning using a linear transformation. *The Journal of Machine Learning Research*, 13(1):519–547, 2012.

[17] Y. Jia and T. Darrell. Heavy-tailed distances for gradient based image descriptors. In *Advances in Neural Information Processing Systems*, pages 397–405, 2011.

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

[19] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *British Machine Vision Conference (BMVC'04)*, pages 779–788. The British Machine Vision Association (BMVA), 2004.

[20] Y. Li, A. Dai, L. Guibas, and M. Nießner. Database-assisted object retrieval for real-time 3d reconstruction. In *Computer Graphics Forum*, volume 34. Wiley Online Library, 2015.

[21] D. Maturana and S. Scherer. 3D Convolutional Neural Networks for Landing Zone Detection from LiDAR. In *ICRA*, 2015.

[22] N. Mellado, D. Aiger, and N. J. Mitra. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library, 2014.

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

[24] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32(6):169, 2013.

[25] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[26] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3384–3391. IEEE, 2008.

[27] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2930–2937. IEEE, 2013.

[28] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1573–1585, 2014.

[29] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *Computer Vision–ECCV 2010*, pages 356–369. Springer, 2010.

[30] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustmenta modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer, 2000.

[31] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua. Learning image descriptors with the boosting-trick. In *Advances in neural information processing systems*, pages 269–277, 2012.

[32] J. Valentin, A. Dai, M. Nießner, P. Kohli, P. Torr, S. Izadi, and C. Keskin. Learning to navigate the energy landscape, 2016.

[33] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[34] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A deep representation for volumetric shapes. 2015.

[35] J. Xiao, A. Owens, and A. Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. 2013.

[36] J. Xiao, S. Song, D. Suo, and F. Yu. Marvin: A minimalist GPU-only N-dimensional ConvNet framework. 2016. Accessed: 2015-11-10.

[37] F. Yu, J. Xiao, and T. Funkhouser. Semantic alignment of LiDAR data at city scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[38] J. Žbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. *arXiv preprint arXiv:1409.4326*, 2014.