Shape Anchors for Data-driven Multi-view Reconstruction

Andrew Owens
MIT CSAIL
andrewo@mit.edu

Jianxiong Xiao
Princeton University
xj@princeton.edu

Antonio Torralba
MIT CSAIL
torralba@csail.mit.edu

William Freeman
MIT CSAIL
billf@mit.edu

Abstract

We present a data-driven method for building dense 3D reconstructions using a combination of recognition and multi-view cues. Our approach is based on the idea that there are image patches that are so distinctive that we can accurately estimate their latent 3D shapes solely using recognition. We call these patches shape anchors*, and we use them as the basis of a multi-view reconstruction system that transfers dense, complex geometry between scenes. We "anchor" our 3D interpretation from these patches, using them to predict geometry for parts of the scene that are relatively ambiguous. The resulting algorithm produces dense reconstructions from stereo point clouds that are sparse and noisy, and we demonstrate it on a challenging dataset of real-world, indoor scenes.*

1. Introduction

While there are many cues that could be used to estimate depth from a video, the most successful approaches rely almost exclusively on cues based on multiple-view geometry. These *multi-view* cues, such as parallax and occlusion ordering, are highly reliable, but they are not always available, and the resulting reconstructions are often incomplete – containing structure, for example, only where stable image correspondences can be found. What's often missing in these reconstructions is surface information: for example it is often difficult to tell from just a stereo point cloud whether the floor and wall intersect in a clean right angle or in a more rounded way.

Single-image *recognition* cues, on the other hand, are highly informative about surfaces, but they are comparatively unreliable, since for any given image patch, there usually are several possible 3D interpretations. Recent single-image reconstruction work has dealt with this problem by imposing strong regularization on the result e.g. with a Markov Random Field [21] or by transferring depth from a small number of matching images [16]; however, it is not clear how to use these heavily regularized reconstructions when high-accuracy multi-view cues are available as well.

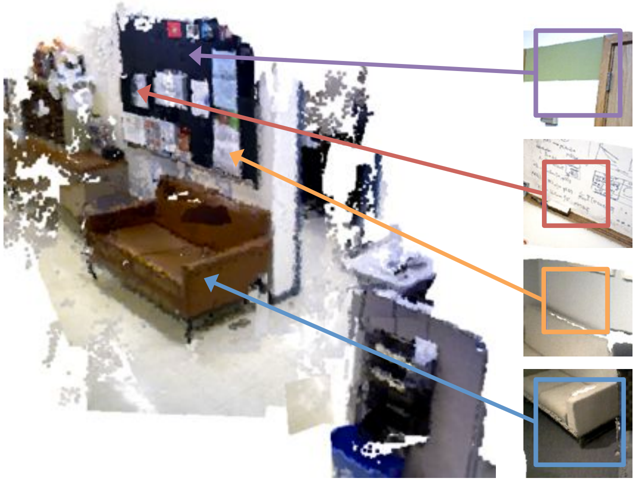


Figure 1. We transfer dense point clouds from training set, combining recognition and multi-view cues.

Despite the ambiguity of image patches in general, we hypothesize that many patches are so distinctive that their latent 3D shapes can be estimated using recognition cues alone. We call these distinctive patches and their associated reconstructions *shape anchors* (Figure 2), and in this paper we describe how to use them in conjunction with multi-view cues to produce dense 3D reconstructions (Figure 1).

We start with a sparse point cloud produced by multi-view stereo [11] and apply recognition cues cautiously, estimating dense geometry only in places where the combination of image and multi-view evidence tells us that our predictions are likely to be accurate. We then use these confident predictions to *anchor* additional reconstruction, predicting 3D shape in places where the solution is more ambiguous. Since our approach is based on transferring depth from an RGB-D database, it can be used to estimate the geometry for a wide variety of 3D structures, and it is well suited for reconstructing scenes that share common objects and architectural styles with the training data.

Our goal in this work is to build dense 3D reconstructions of real-world scenes, and to do so with accuracy at the level of a few centimeters. We use videos extracted from SUN3D [28] – cluttered indoor scenes with uncontrolled camera motion. These factors make it harder to use multi-

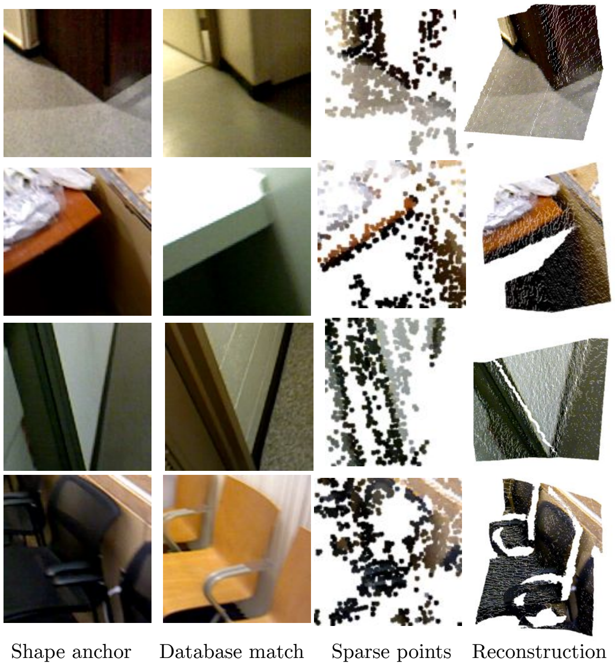


Figure 2. Shape anchors (left) are distinctive image patches whose 3D shapes can be predicted from their appearance alone. We transfer the geometry from another patch (second column) to the scene after measuring its similarity to a sparse stereo point cloud (third column), resulting in a dense 3D reconstruction (right). [1]

view cues, and as a result the stereo point clouds are sparse and very noisy; by the standards of traditional multi-view benchmarks [26] [22] the reconstructions that we seek are rather coarse.

In the places where we predict depth using shape anchors, the result is dense, with accuracy close to that of multi-view stereo, and often there is qualitative information that may not be obvious from the point cloud alone (e.g. the presence of folds and corners).

2. Related work

The idea of combining single- and multi-view cues has a long history, with early work [3] [27] using stereo and shape-from-shading to infer low and high frequency shape, respectively. In a similar spirit, we use multi-view cues to provide a skeleton of a reconstruction that we then flesh out using recognition cues. Our way of combining these two cues is to use the single-image cues sparingly, hypothesizing a dense depth map for each image patch using recognition and accepting only the hypotheses that agree with the multi-view evidence.

Recent work has combined recognition and geometry as well. For example, [20] [8] build piecewise-planar or highly regularized reconstructions, [1] densely reconstructs individual objects from a particular category (also based on an anchoring idea), and [13] solves jointly for a semantic segmentation and a dense 3D model.

We are interested in using recognition to estimate structures like corners, folds, and planar regions, as well as some more complicated geometry like that of large objects (e.g. chairs and couches). In this way, our goals differ from some recent work in single-image reconstruction such as [21] [15], which model lower-level shape information, e.g. estimating per-superpixel surface orientation. Recently [16] proposed a data-driven technique that transfers depth from an RGB-D database using SIFT Flow [19]. We expect whole-image transfer to be useful for capturing coarse geometry, but getting the finer details right seemingly requires the algorithm to find a pool of nearest-neighbor images that contain all of the objects that it needs to transfer depth from. Furthermore, nearest-neighbor search performs poorly when the input is of cluttered indoor scenes, such as those in our database. Our approach avoids this problem by transferring depth at a *patch* level.

The sparsity of multi-view stereo is a well-known problem, and recent work [9] [10] has attempted to address this shortcoming for indoor scenes, producing impressive results that are well suited for visualization purposes. These techniques make strong assumptions about the geometry of the scene: [9], for example, regularizes based on the assumption that the world is Manhattan and mostly planar. Similarly, there is work [17] that estimates a dense mesh from a point cloud. The focus of our work is different and complementary: instead of using strong regularization, we attempt to get more information out of local (appearance and multi-view) evidence.

The idea of finding image patches whose appearance is informative about geometry takes inspiration from recent work in recognition, notably poselets [4] (i.e. distinctive visual patterns that are informative about human pose) and also recent work on mid-level discriminative patches [24]. We also test whether a patch is informative, but instead of defining detectors that fire when they see a particular 3D shape (which would be the analogue to a poselet in our case), we do everything using a data-driven search procedure, integrating the recognition and multi-view evidence together in our decision-making process.

3. Shape anchors

Our approach is based on reconstructing the 3D shape of individual image patches, and in its most general form this problem is impossibly hard: the shape of most image patches is highly ambiguous. We hypothesize, however, that there are image patches so distinctive that their shape can be guessed rather easily.

We call these patches and their associated reconstruc-

[1] The errors here are tolerable for the level of accuracy that we seek: e.g. we do not care about the exact position of the chair arms in the last example.

tions *shape anchors* (Figure 2), and we say that a point cloud representing a 3D-reconstructed patch is a shape anchor if it is sufficiently similar to the patch's ground-truth point cloud. Later, we will describe how to identify these correct reconstructions (Section 4) and use them to interpret the geometry for other parts of the scene (Section 5). Now we will define what it means for a patch's 3D reconstruction to be correct – in other words, for a patch and its reconstruction to be a shape anchor.

Shape similarity One of the hazards of using recognition to estimate shape is an ambiguity in absolute depth, and accordingly we use a measure of shape similarity that is invariant to the point cloud's distance from the camera (we do not model other ambiguities, e.g. rotation or scale). Specifically, if P_D is the point cloud that we estimate for a patch, and v is the camera ray passing through the patch's center, then we require P_D to satisfy the distance relationship

$$\min_{\alpha \geq 0} \phi(P_D + \alpha v, P_{GT}) \leq \tau, \tag{1}$$

where P_{GT} is the patch's ground-truth point cloud and $P_D + \alpha v$ denotes a version of the point cloud that has been shifted away from the camera by distance α, i.e. $P_D + \alpha v = \{x + \alpha v \mid x \in P_D\}$. We set τ to 10cm, so that the reconstruction is required to be accurate on the order of centimeters. Note that this value is small given that patch reconstructions are often meters in total size, and that this parameter controls the overall accuracy of the reconstruction. We define ϕ to be the average distance between points in one set to their nearest neighbors in the other, specifically

$$\phi(X, Y) = \max(\psi(X, Y), \psi(Y, X)), \tag{2}$$

where

$$\psi(X, Y) = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} ||x - y||. \tag{3}$$

In other words, for a patch's reconstruction to be considered correct (i.e. for it to be considered a shape anchor), the average distance between a reconstructed point and the nearest ground-truth point must be at most τ (and vice versa) after correcting for ambiguity in absolute depth.

We note that the two terms in ϕ, namely $\psi(P_D, P_{GT})$ and $\psi(P_{GT}, P_D)$, are analogous to the *accuracy* and *completeness* measures used in evaluating multi-view stereo algorithms [22], and are also similar to the objective functions minimized by the Iterative Closest Point method [2].

In effect, patch reconstructions are evaluated holistically: the only ones that "count" are those that are mostly right.

4. Predicting shape anchors

We start by generating multiple 3D reconstructions for every patch in the image using a data-driven search procedure – solely using single-image cues. We then introduce multi-view information and use it in combination with the image evidence to distinguish the "good" patch reconstructions (i.e. the shape anchors) from the bad. This whole process is illustrated in Figure 3.

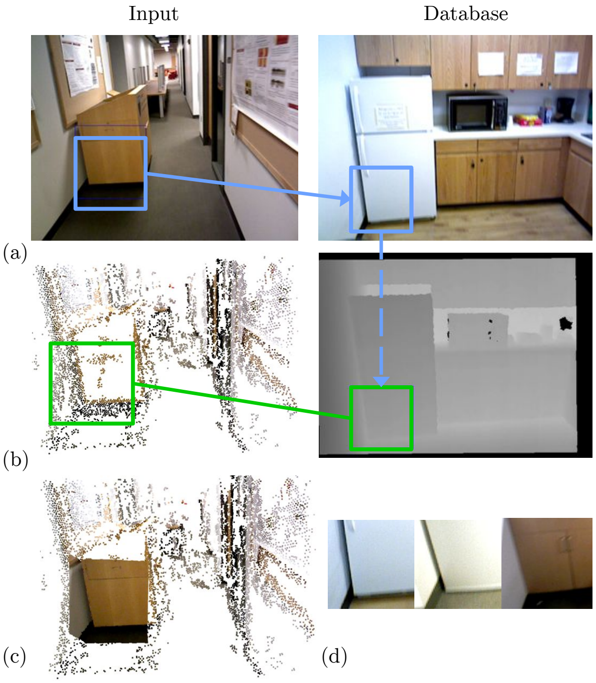


Figure 3. Finding shape anchors. (a) Given a patch, we search for the most similar patches in an RGB-D database (using only a single image as input). We then (b) compare the depth map of the best matches with the sparse stereo point cloud, transferring the dense shape if their depths agree (c). As part of this process, we test whether the top matches' shapes agree with each other (d).

4.1. Data-driven shape estimation

Under our framework, the use of recognition and multi-view cues is mostly decoupled: the goal of the "recognition system" is to produce as many good patch reconstructions (i.e. shape anchors) as possible, and the goal of the "multi-view system" is to prune the bad ones. In principle, then, there are many approaches that could be used for the recognition subcomponent – e.g. one could train detectors to recognize a list of common shapes. In this work, we choose to generate our reconstructions using a data-driven search procedure, since this allows us to represent complex geometry for a variety of scenes.

Given a set of patches from an input image, we find each one's best matches in an RGB-D database (using the "RGB" part only) and transfer the corresponding point cloud for one of the examples (using the "-D" part). Our search procedure is similar to that of [23], but instead of querying a single template, we query using all of the patches in an image.

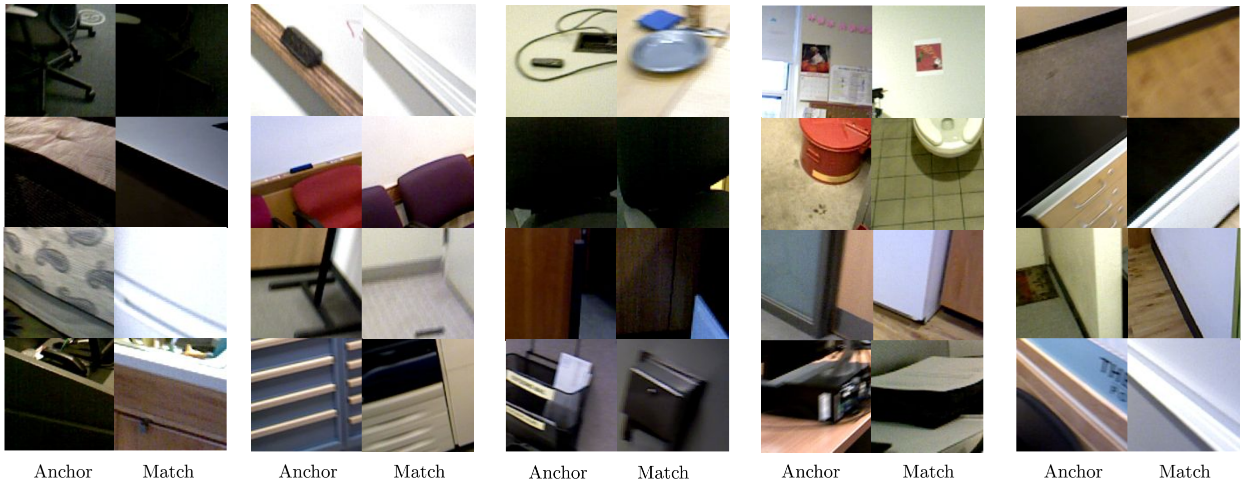


Figure 4. The highest-scoring shape anchor prediction for a sample of scenes, with their associated database matches. The corresponding stereo point clouds are not shown, but they are used as part of the scoring process.

Extracting and representing patches Following recent work in image search and object detection [12] [14], we represent each patch as a HOG template whitened by Linear Discriminant Analysis. These patches are obtained by downsampling the image, computing HOG, and extracting every overlapping 8×8-cell template. There are about 300 such patches per image, and each is 170×170 pixels (about 9% of the image area).

Searching We convolve each patch's HOG template with images in the database, searching at multiple scales and allowing detections that are 50% to 150% the width of the original patch. We keep the k highest-scoring detections for each template, resulting in k reconstructions for each patch (we use $k = 3$). We then zero-center each reconstruction by subtracting the 3D point corresponding to the patch's center. This step could likely be run faster (or at a larger scale) by using approximate convolution [6].

4.2. Distinguishing shape anchors

We now use multi-view cues to identify a subset of patch reconstructions that we are confident are shape anchors (i.e. we are confident that they satisfy Equation 1), and to resolve the absolute depth ambiguity. We start by aligning each reconstruction to a sparse point cloud (produced by multi-view stereo, see Section 6), shifting the reconstruction away from the camera so as to maximize its agreement with the sparse point cloud. More specifically, if P_D is the point cloud of the retrieved patch and v is the camera ray passing through the patch's center pixel, then we find

$$\alpha_c = \arg\min_{\alpha} \psi_N(S, P_D + \alpha v), \tag{4}$$

where S is the sparse point cloud and $\psi_N(X, Y)$ is the number of points in X that are within τ of some point in Y. We optimize this objective via grid search on α (searching ±1m from its original depth in increments of 1cm). We then align the point cloud, constructing $P'_D = P_D + \alpha_c v$.

After the alignment, we discard erroneous patch reconstructions, keeping only the ones that we are confident are shape anchors. We do this primarily by throwing out the ones that significantly disagree with the multi-view evidence; in other words, we look for reconstructions for which the recognition- and multi-view-based interpretations *coincide*. There are other sources of information that can be used as well, and we combine them using a random forest classifier [5], trained to predict which patch reconstructions are shape anchors. For each patch reconstruction, we compute three kinds of features.

Multi-view evidence Defining $H_d(X, Y)$ to be the histogram, computed over points $x \in X$, of $\min_{y \in Y} ||x - y||$, we include $H_d(S, P'_D)$ and $H_d(P'_D, S)$, where P'_D is the recentered patch reconstruction and S is the sparse point cloud. We also include the absolute difference between the patch reconstruction's depth before and after alignment. In our experience, these multi-view features are the most important ones for classification accuracy.

Image evidence We include the convolution score and the difference in pixel location between the queried and retrieved patches.

Patch informativeness These features test whether the queried patch is so distinctive that there is only one 3D shape interpretation. We measure the reconstruction's similarity to the point clouds of the other best-matching patches (Figure 3 (d)). We include $\frac{1}{k-1}\sum_{i=1}^{k-1} H_d(P'_D, C'_i)$ and $\frac{1}{k-1}\sum_{i=1}^{k-1} H_d(C'_i, P'_D)$ as features, where C'_i is the aligned patch reconstruction for one of the $k-1$ other matches. We note a similarity between this feature and the idea behind poselets [4]: we are testing whether the queried features commonly co-occur with a 3D shape.

We note that all of these features measure only the quality of the match; we do not compute any features for the point cloud itself, nor do we use any image features (e.g. HOG itself) – either of which may improve the results.

If a patch reconstruction is given a positive label by the random forest, then we consider it a *shape anchor prediction*, i.e. we are confident that it will be accurate in the sense of Equation 1. If more than one of the k matches receives a positive label, we choose the one with the highest classifier score. We show examples of shape anchors in Figure 4.

5. Interpreting geometry with shape anchors

We now describe how to "anchor" a reconstruction using the high-confidence estimates of geometry provided by shape anchors. We use them to find other patch reconstructions using contextual information (Section 5.1), and for finding planar regions (Section 5.2). Finally, we use occlusion constraints to get a shape interpretation that is coherent across views (Section 5.3).

5.1. Propagating shape anchor matches

We start by repeating the search-and-classification procedure described in Section 3, restricting the search to subsequences centered on the sites of the highest-scoring shape anchor predictions (we use a subsequence of 20 frames and 200 top shape anchors). We also query smaller patches (75% of the original's width).

We also try to find good patch reconstructions for the area surrounding a shape anchor (Figure 6). We sample RGB-D patches near the matched database patch, and for each one we test whether it agrees with the corresponding patch in the query image using the method from Section 4.2 (i.e. aligning the patch's points to the stereo point cloud and then classifying it). The RGB-D patches that we attempt to transfer are non-overlapping, and we sample them from a 6×6 grid centered on the database patch.

5.2. Extrapolating planes from shape anchors

We use shape anchor predictions that are mostly planar to guide a plane-finding algorithm (Figure 5). For each prediction that contains a large planar region (75% of its points are within 5cm of a plane fit by RANSAC [7]), we fit a plane to the stereo points that are visible through the frustrum of the shape anchor's image patch using RANSAC, restricting the RANSAC hypotheses to be those that are close to the anchor's plane (their surface normals differ by less than 15°, and distance from the origin differs by no more than 10cm).

We then use the shape anchor to infer the support of the plane, possibly expanding it to be much larger than the original patch. To do this, we use the foreground-background superpixel segmentation method of [18]. For learning the foreground-background color model, superpixels containing points that are far from the plane (more than 30cm) are considered to be background observations; superpixels that intersect with the on-plane parts of the shape anchor are considered foreground observations. We consider the final, inferred foreground to be the support of the plane.

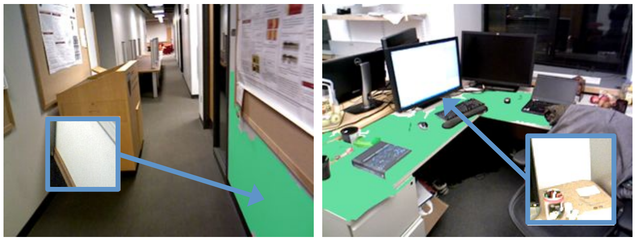


Figure 5. Plane extrapolation. If a shape anchor's reconstruction is mostly planar, we infer the support of the plane (green). We also show the database matches for the two shape anchors.

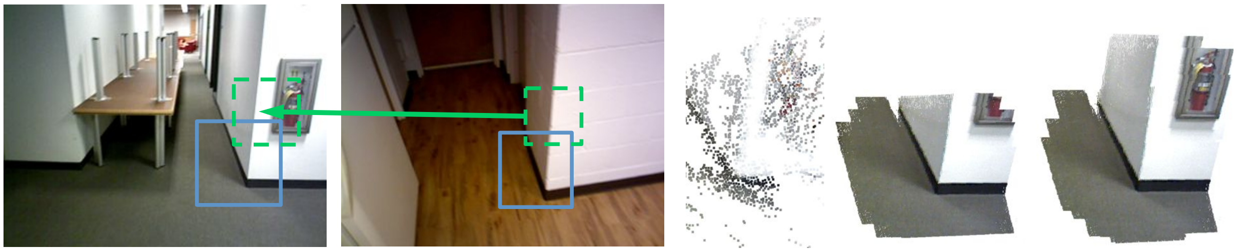
We keep the expanded plane only if it is larger than the original shape anchor and agrees with the multi-view evidence. We evaluate this by testing whether the superpixels that contain on-plane points outnumber those that contain off-plane points by a ratio of 9 to 1. Otherwise, we keep the original shape anchor.

5.3. Using occlusion constraints

Since the patch reconstructions (shape anchors and propagated patches) are predicted in isolation, they may be inconsistent with each other. To address this, we remove erroneous predictions using occlusion constraints. First, we remove points that are inconsistent with each other in a single view; we keep at each pixel only the point that comes from the patch reconstruction with the greatest classifier score. We then handle occlusions between images. For each image, we find all of the patch reconstructions from other images that are visible. If a point from one of these other images occludes a point from the image's own patch reconstructions, then this violates an occlusion constraint; we resolve this by discarding the point that comes from the patch reconstruction with the lower classifier score. Finally, we completely discard patch reconstructions for which only 10% or fewer of the points remain, since they are likely to be incorrect or redundant.

6. Results

Video dataset We derived a new dataset from SUN3D [28], an RGB-D video dataset with high-quality estimates of camera pose. Many of the scenes in this dataset share common object and architectural styles. These videos were taken with the RGB camera of a Kinect-style RGB-D sensor, so they are low resolution (640×480 pixels), and there are other factors that make multi-view reconstruction challenging, e.g. uncontrolled camera motion and poor lighting. For the point cloud visualizations in the qualitative results, we estimate the camera pose using structure from motion (SfM) instead of using the SUN3D pose estimates.



(a) Propagation (b) Stereo (c) Shape anchors (d) Propagation result

Figure 6. Anchor propagation. The propagation step (a) starts with an anchor patch (the corner, in blue) and finds an additional match with the relatively ambiguous patch (in green). The result is a larger reconstruction (d).

We split the videos into training and test sets. The training set is used to learn the classifier (Section 3). From the test set, we sample 10-second subsequences in which the camera travels at least 2 meters. We estimate the camera pose for each sequence using Bundler [25] after sampling one in every 5 frames (from 300 frames), discarding scenes whose SfM reconstructions have significant error [2]; approximately 18% of these subsequences pass this test. We then sample 49 videos, at most one per full sequence. To get the stereo point cloud, we use PMVS [11] with default parameters. We search for shape anchors in 6 frames per video.

It takes about 2 hours to run our system on a sequence using a cluster of six 12-core computers. In the database search step, we search about 30,000 images.

Estimating absolute scale To predict shape anchors, we require the true scale of the reconstruction, which is unknown when the pose comes from SfM. We estimate this from a set of high-confidence patch reconstructions (high convolution score and low patch-location difference). Each triangulated 3D point votes for a scale – its distance to the camera divided by that of the corresponding point in the patch reconstruction – and we choose the mode. While this scale estimate is coarse, the results usually are qualitatively similar when we use the SUN3D pose instead.

Quantitative evaluation As a measure of accuracy, we estimate the distance from each reconstructed point to the nearest point in the ground-truth point cloud (Figure 7(a)). And as a rough overall measure, we also compute the median of such distances [3]. If multiple shape anchors overlap, then we take the highest-scoring point at each pixel. We find that the accuracy is close to that of PMVS, with PMVS having a larger fraction of points with near-zero error.

As an estimate of the reconstruction's completeness, we measured the fraction of ground-truth points that were within 3cm of some point in the reconstruction. When we consider only the points that fell inside the frustrums of the shape anchor prediction windows, in both the reconstruction and the ground-truth, we find that the shape anchors are more complete than the PMVS points (Figure 7(d)).

We also include statistics about the accuracy and completeness of our final patch reconstructions (Figure 7(d)), including the propagation and occlusion-testing steps (Section 5); we discard patch reconstructions below a score threshold, and we exclude the extrapolated planes so that what is being measured is the transferred geometry (though the planes are quite accurate). We find that combining our predicted geometry with the original point cloud results in a denser reconstruction with similar overall accuracy.

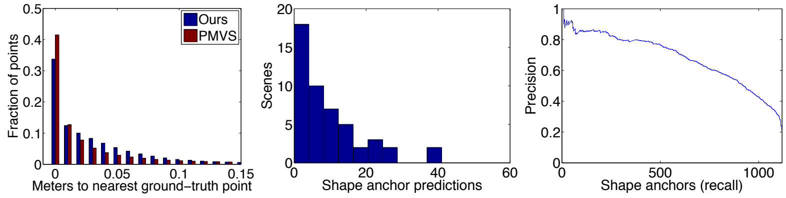
We note that the SUN3D reconstructions themselves have errors, and that our goal is simply to provide a rough comparison between our patch reconstructions and the original point cloud; a careful study of reconstruction quality would need to design error metrics more rigorously and to control for the many other sources of error. We note that there are other, purely multi-view, methods that could also perform well (e.g. an algorithm that simply estimated the ground plane would likely score well under our metrics).

Qualitative results In Figure 8, we show visualizations for some of our reconstructions (a subset of the test set). These reconstructions were created by combining the predicted patch reconstructions (i.e. shape anchor predictions plus the propagated geometry) and the extrapolated planes. We used only the highest-scoring extrapolated plane for each frame (i.e. six at most for a whole scene) so that the result mostly shows the transferred geometry. We encourage readers to consult our video fly-throughs, since it is difficult to perceive reconstruction errors in static images.

The results are dense 3D reconstructions *composed of translated point clouds from the database*, plus a small number of extrapolated planes. Our approach is well suited for transferring large pieces of distinctive architecture such as wall-floor junctions and corners (e.g. in (a)). And while some of these structures could be discovered using purely geometric approaches (e.g. by fitting planes and grouping them), we get this information automatically by transferring geometry. Our approach is also successful in transferring large, distinctive objects, e.g. a couch (b), a desk and

[2] We align the cameras to SUN3D with a rigid transformation and require at most 5cm of translation error, 15° of viewing direction error, and a median accuracy of 8cm for the SfM point cloud.

[3] [22] uses the 90th percentile, but there are many more outliers in our case (e.g. PMVS's accuracy is poor under this metric).



(a) (b) (c)

	Win. Acc.	Win. Comp.
Anchors	3.8cm	63.4%
PMVS	2.1cm	44.2%

	Full Acc.	Full Comp.
Transferred	1.8cm	21.6%
PMVS	1.6cm	27.4%
Combined	1.8cm	35.6%

(d)

Figure 7. Accuracy of transferred geometry. (a) Accuracy of points from shape anchor predictions. (b) Number of anchor predictions per scene. (c) Classifier precision and recall. (d) Accuracy and completeness measures, for both the full scene and the just the shape anchor windows. For (b) and (c), we use a non-maximum suppression procedure on the prediction windows to avoid double counting.

chair in sequence (c), and a sink in Figure 9 (whose highly reflective surface produces many erroneous stereo points).

Our method is less successful in modeling fine-scale geometry, partly due to the large patch size and the distance threshold of 10cm that we require for shape anchors (Equation 1). For example, in (d), we model a chair arm using a patch from a bathroom. We also sometimes transfer patches containing extra geometry: in (a) we hallucinate a chair while transferring a wall. We make no attempt to align shape anchors beyond translating them, so walls may be at the wrong angles, e.g. in (a) and Figure 9.

We note that the magnitude of the errors is usually not too large, since the classifier is unlikely to introduce a shape anchor that strays too far from the sparse point cloud. Erroneous transfers often resemble the result of fitting a plane to a small neighborhood of points. The number of shape anchor predictions can also vary a great deal between scenes (Figure 7(b)), meaning that for many scenes the results are sparser than the ones presented here (please see our video for examples). This is partly due to the data-driven nature of our algorithm: for some scenes it is hard to find matches even when the search is conducted at the patch level.

On the other hand, our method produces very dense reconstructions when the training set does contain relevant scenes. In Figure 9, we show an example where geometry is transferred between apartment units in the same building.

7. Conclusion

In this work, we introduced *shape anchors*, image patches whose shape can easily be recognized from the patch itself, and which can be used to "anchor" a reconstruction. We gave several examples of how they can be used to build dense 3D reconstructions, and we hope that this representation will find other uses as well. We also believe that the recognition task presented in this work – namely that of generating accurate 3D reconstructions from image patches – is an interesting problem with many solutions beyond the data-driven search method described here.

Acknowledgments. We thank Phillip Isola for the helpful discussions. This work was supported by an NDSEG Fellowship to A.O., ONR MURI N000141010933 to A.T., and NSF grant CGV-1212928 to W.F.

References

[1] Y. Bao, M. Chandraker, Y. Lin, and S. Savarese. Dense object reconstruction with semantic priors. In *CVPR*, 2013. 2

[2] P. Besl and N. McKay. A method for registration of 3-d shapes. *Trans. PAMI*, 1992. 3

[3] A. Blake, A. Zisserman, and G. Knowles. Surface descriptions from stereo and shading. *Image and Vision Computing*, 1985. 2

[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 2, 4

[5] L. Breiman. Random forests. *Mach. learning*, 45(1):5–32, 2001. 4

[6] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*, 2013. 4

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 5

[8] A. Flint, D. Murray, and I. Reid. Manhattan scene understanding using monocular, stereo, and 3d features. In *ICCV*, 2011. 2

[9] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Manhattan-world stereo. In *CVPR*, 2009. 2

[10] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *ICCV*, 2009. 2

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *Trans. PAMI*, 2010. 1, 6

[12] M. Gharbi, T. Malisiewicz, S. Paris, and F. Durand. A gaussian approximation of feature space for fast image similarity. *Technical Report 2012-032, MIT CSAIL*, 2012. 4

[13] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3d scene reconstruction and class segmentation. In *CVPR*, 2013. 2

[14] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. *ECCV*, 2012. 4

[15] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007. 2

[16] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *ECCV*, 2012. 1, 2

[17] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Eurographics*, 2006. 2

[18] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM ToG (SIGGRAPH)*, 2004. 5

[19] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, 2008. 2

[20] A. Saxena, M. Sun, and A. Ng. 3-d reconstruction from sparse views using monocular vision. In *ICCV workshop on Virtual Representations and Modeling of Large-scale Environments (VRML)*, 2007. 2

[21] A. Saxena, M. Sun, and A. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 2009. 1, 2

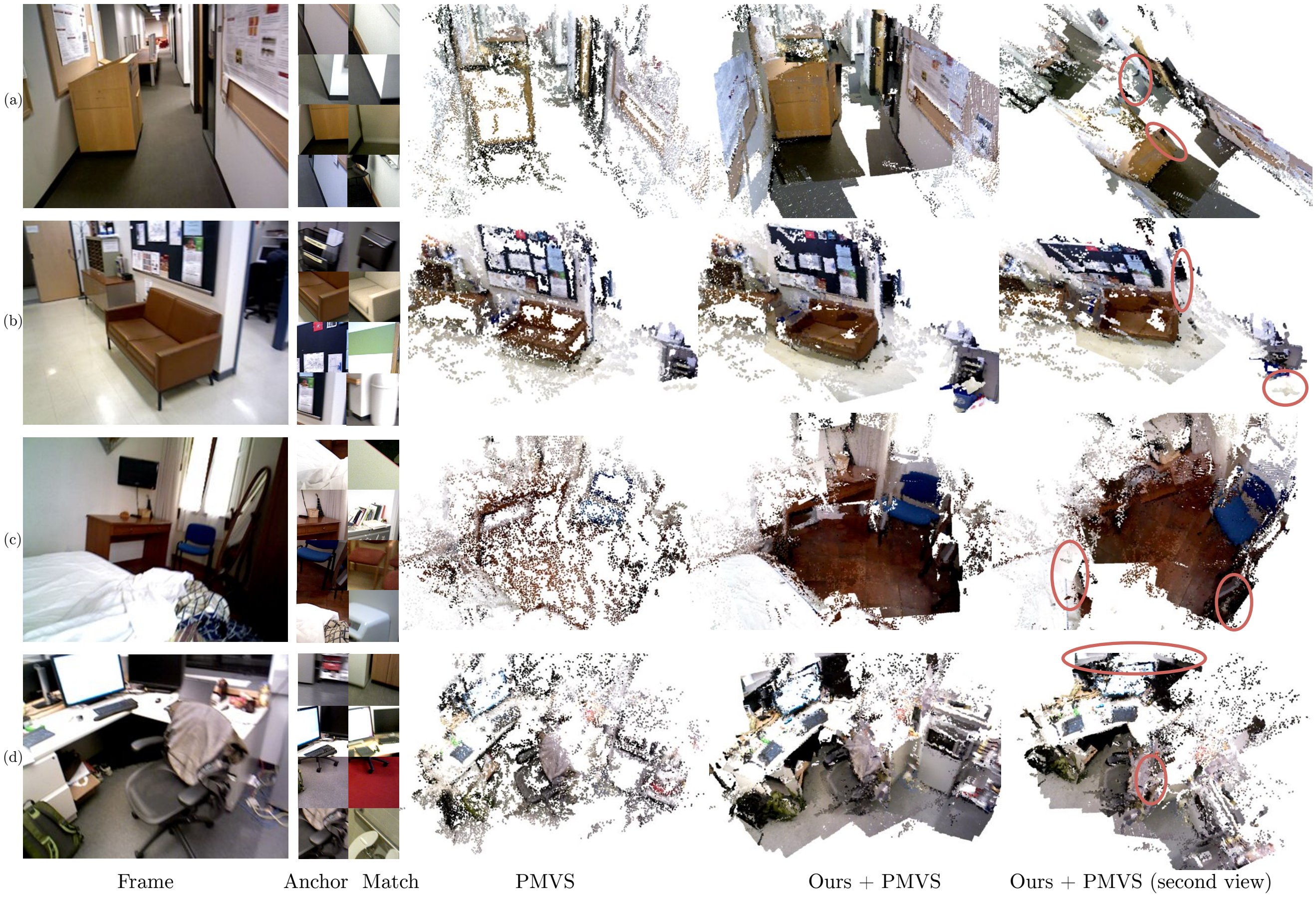


Figure 8. 3D reconstruction results for four scenes, chosen from the test set for their large number of shape anchor predictions. We show the PMVS point cloud and two views of our dense reconstruction combined with the PMVS points (our final output). For each scene, we show four shape anchor transfers, selected by hand from among the top-ten highest scoring ones (that survive occlusion testing); we show one erroneous shape anchor per scene in the last row. We mark significant errors, two per scene, with a red circle. We encourage readers to view our video fly-throughs, since it is difficult to perceive reconstruction errors in a static image.

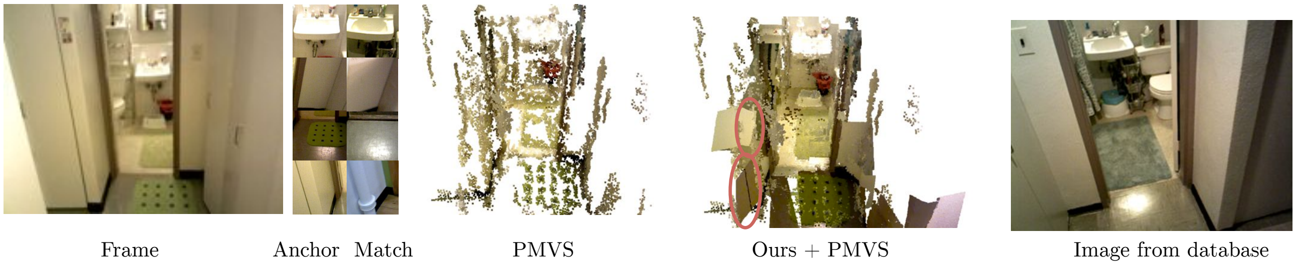


Figure 9. Training with similar scenes. When our training set contains sequences from the same apartment complex, the result is a particularly dense reconstruction.

[22] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. 2, 3, 6

[23] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. Efros. Data-driven visual similarity for cross-domain image matching. *ACM ToG (SIGGRAPH Asia)*, 2011. 3

[24] S. Singh, A. Gupta, and A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 2

[25] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM ToG (SIGGRAPH)*, 2006. 6

[26] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008. 2

[27] A. Torralba and W. Freeman. Properties and applications of shape recipes. In *CVPR*, 2003. 2

[28] J. Xiao, A. Owens, and A. Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *ICCV*, 2013. 1, 5