# 3DNN: Viewpoint Invariant 3D Geometry Matching for Scene Understanding

Scott Satkin
Google Inc.*
satkin@google.com

Martial Hebert
Carnegie Mellon University
hebert@ri.cmu.edu

## Abstract

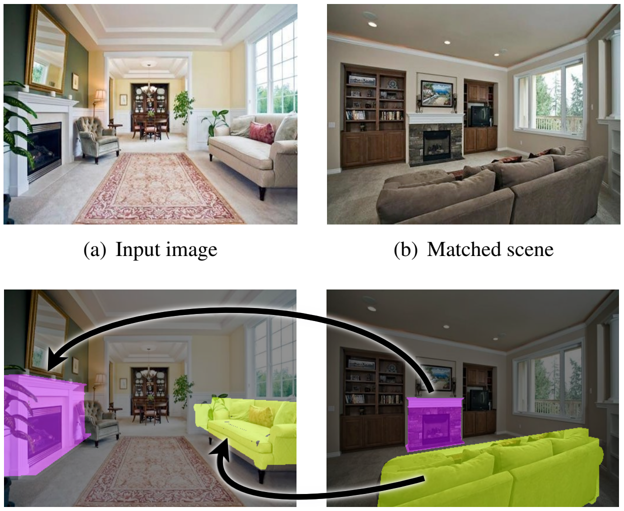
*We present a new algorithm 3DNN (3D Nearest-Neighbor), which is capable of matching an image with 3D data, independently of the viewpoint from which the image was captured. By leveraging rich annotations associated with each image, our algorithm can automatically produce precise and detailed 3D models of a scene from a single image. Moreover, we can transfer information across images to accurately label and segment objects in a scene.*

*The true benefit of 3DNN compared to a traditional 2D nearest-neighbor approach is that by generalizing across viewpoints, we free ourselves from the need to have training examples captured from all possible viewpoints. Thus, we are able to achieve comparable results using orders of magnitude less data, and recognize objects from never-before-seen viewpoints. In this work, we describe the 3DNN algorithm and rigorously evaluate its performance for the tasks of geometry estimation and object detection/segmentation. By decoupling the viewpoint and the geometry of an image, we develop a scene matching approach which is truly 100% viewpoint invariant, yielding state-of-the-art performance on challenging data.*

## 1. Introduction

Data-driven scene matching is at the forefront of the computer vision field. Researchers have demonstrated the capability of simple nearest-neighbor based approaches to match an input image (or patches of an image) with a corpus of annotated images, to "transfer" information from one image to another. These non-parametric approaches have been shown to achieve amazing performance for a wide variety of complex computer vision and graphics tasks ranging from object detection [32] and scene categorization [24] to motion synthesis [20] and even image localization [10].

Although these 2D nearest-neighbor approaches are powerful, a fundamental limitation of these techniques is the need for vast amounts of data. For a traditional image matching approach to succeed, there must be an image in the recall corpus which is very similar to the input image (i.e., captured from a similar viewpoint, lighting conditions, etc.). This has propelled the growth of datasets, which now measure in the millions of images [9, 33]. Moreover, despite these massive datasets, 2D nearest-neighbor approaches cannot generalize to never-before-scene viewpoints.



(a) Input image

(b) Matched scene

(c) Automatically transferred object labels

Figure 1. Extreme viewpoint differences. Traditional appearance based image matching approaches fail to generalize across such extreme viewpoint differences; however, our approach is able to match the geometry of these two scenes, and transfer object labels.

Recently, we demonstrated a proof-of-concept method for matching images with 3D models to estimate the geometry of a scene [28]. Building upon this work, we present a viewpoint invariant approach to match images based solely on each scene's geometry. Consider the pair of scenes in Figure 1. Note that the images were captured from drastically different viewpoints. A traditional appearance-based image matching approach such as [20, 24] would fail to generalize across such extreme viewpoint differences. Al-

*Research conducted while the author was a doctoral student at Carnegie Mellon University.

though the scenes appear quite different from the viewpoints they were captured, they have a lot in common: both scenes contain a couch facing a fireplace at approximately the same distance from each other. In this work, we show that we are able to automatically match these images by comparing the appearance of one image with the geometry of another. By decoupling the viewpoint and the geometry of an image, we develop a scene matching approach which is truly 100% viewpoint invariant.

Our algorithm, 3DNN (3D Nearest-Neighbor), is capable of producing *detailed* and *precise* 3D models of a scene from a single image. The problem of monocular 3D scene understanding has recently been gaining tremendous attention from the computer vision community (e.g.: [4, 13, 17, 18, 26, 28, 29, 30, 36, 38]). The common goal of this research is to estimate the full geometry of a scene from a single viewpoint. The ability to infer the geometry of a scene has enabled a variety of applications in both the vision and graphics fields. For example, Gupta et al. [8] use coarse geometry estimates to predict what locations in an environment afford various actions. Karsch et al. [16] use scene geometry to realistically render additional objects into a scene. Similarly, Zheng et al. [39] utilize knowledge of scene geometry to create an interactive 3D image editing tool. It is important to note, that these graphics applications require precise geometry estimates, which traditionally have involved manual annotation.

Current approaches for monocular geometry estimation typically produce coarse results, modeling each object with bounding cuboids (e.g.: [4, 13, 18, 25]). More recently, we proposed matching images with 3D models harvested from the Internet, to generate results with greater detail [28]. However, because this approach aims to match the exact configuration of objects in an image, with an identical furniture configuration from a library of 3D models, the algorithm does not have the flexibility required to precisely reconstruct the diversity of object configurations found in natural scenes. Additionally, many scene understanding approaches such as [7, 8, 28] make limiting assumptions regarding the robustness of existing monocular autocalibration algorithms. When this preliminary stage of their pipelines fail, the algorithms are unable to recover and produce a reasonable result.

Our work addresses these fundamental limitations by simultaneously searching over both the camera parameters used to capture an image, as well as the underlying scene geometry. In addition, we present a refinement algorithm which takes a rough initial estimate of the structure of a scene, and adjusts the locations of objects in 3D, such that their projections align with predicted object locations in the image plane.

In this paper, we describe the 3DNN algorithm and evaluate its performance for the tasks of object detection/segmentation, as well as monocular geometry reconstruction. We show that 3DNN is capable of not only producing state-of-the-art geometry estimation results, but is also capable of precisely localizing and segmenting objects in an image. Our experiments compare 3DNN with traditional 2D nearest-neighbor approaches to demonstrate the benefits of viewpoint invariant scene matching.

## 2. Approach

We estimate the viewpoint from which an image was captured, and search for a 3D model that best matches the input image when rendered from this viewpoint. This builds upon the analysis via synthesis approach introduced in [28]. Rather than limiting ourselves to a fixed set of object configurations, our approach begins with a 3D model which closely matches an image, and then undergoes a *geometry refinement* stage, which adjusts the locations of each object in the hypothesized geometry to produce a result which more precisely matches the input image.

This type of fine-grained geometry refinement is challenging, and requires a set of features which are sufficiently discriminative to identify when rendered objects are precisely aligned in the image plane. Thus, we present a *new set of features* which improve the overall accuracy of our scene matching algorithm, enabling this geometry refinement stage.

In addition, we introduce a *viewpoint selection* process which does not commit to a single viewpoint estimate. We consider many camera pose hypotheses and use a learned cost function to select the camera parameters which enable the best scene geometry match.

At the core of our approach is the use of datasets of images with corresponding 3D descriptions. This immediately raises natural questions as to where this data is coming from, and whether or not there exists sufficient quantities of images with corresponding 3D models. In fact, the development of such 3D content is exploding, in large part due to the availability of low-cost RGBD cameras (Microsoft Kinect [2]), which has been a catalyst for the rapid increase in 2.5D data. Researchers are now working on automated methods for inferring the full 3D geometry of a scene given a 2.5D projection [31]. As these approaches become more effective, there will be massive amounts of images with associated 3D models, allowing for the first time the exciting possibilities afforded by using the full power of geometric information in conjunction with conventional appearance-based techniques. Our work shows how these emerging new sources of data can be used by quantifying their effectiveness in terms of matching efficiency (dataset size), generalization to unseen viewpoints, geometry estimation, and object segmentation.

## 2.1. Similarity Features

Many vision researchers have emphasized and demonstrated the importance of high-quality features for various image matching tasks. For example, the SIFT descriptor of Lowe [21] and Oliva et al.'s GIST descriptor [24] have been shown to outperform many other descriptors for common vision tasks [6, 22]. Given the novelty of 3D scene matching approaches, there still remains substantial room for improvement via feature engineering.

Therefore, we develop a series of features which are specifically designed to achieve our goal of precisely detecting objects and delineating their boundaries. To accurately predict the locations of objects in an image, we train a probabilistic classifier using the algorithm of Munoz et al. [23].[1] For each pixel, we estimate the likelihood of an object being present. This $p(\text{object})$ descriptor is compared to hypothesized object locations via rendering to compute a similarity feature indicating how well hypothesized objects align with predicted object locations. This similarity feature is akin to [28]'s use of geometric context [11, 15]; however, it is more robust to the diversity of object colors, textures and illumination conditions seen in the SUN database [37]. Figure 2 shows an example of a relatively simple scene for which [11] is unable to accurately estimate the locations of objects; however, our approach succeeds. We incorporate this new $p(\text{object})$ feature into the existing set of similarity features from [28].

In addition, we design another similarity feature that aims to find 3D models which, when projected onto the image plane, produce edges which closely align with edges in the input image. For each hypothesized 3D model, we first analyze its surface normals to identify edges (which we define as discontinuities greater than 20°). We compare the projection of these edges onto the image plane, with edges extracted from an input image using the globalPb algorithm [3]. We use an oriented chamfer distance, which matches only edges which are within 30° of each other. This reduces the effects of spurious edges which are spatially close, but not properly oriented in the image. We use the same edge penalty truncation approach as [28], to reduce the influence of outlier edges, resulting in a four-dimensional similarity feature (corresponding to different thresholds).

These new similarity features are combined with the seven features from [28] to produce a 12-dimensional similarity feature vector. We train a support vector ranker [14] using the approach of [28] to learn weights for the augmented set of features with an added $\ell_1$ penalty term to perform feature selection and enforce sparseness in the learned weights.

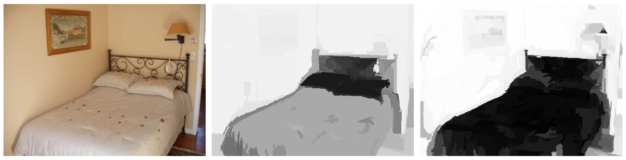


Figure 2. Comparison of $p(\text{object})$ features for the image on the left computed using (center) [11] and our approach (right).

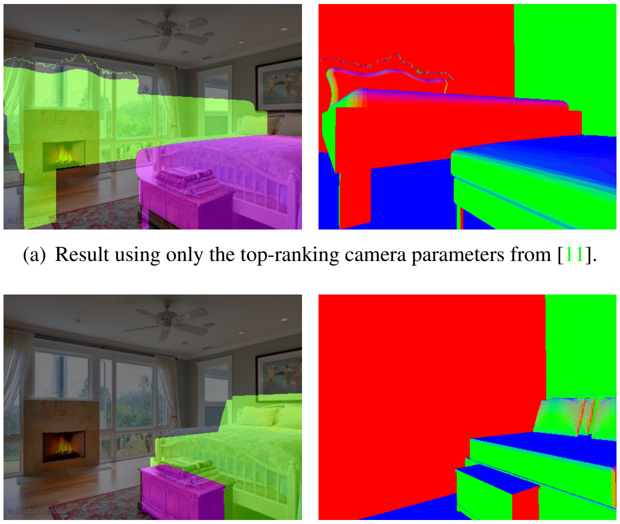
## 2.2. Viewpoint Selection

The problem of viewpoint estimation is very challenging. Estimating the layout of a room, especially in situations where objects such as furniture occlude the boundaries between the walls and the floor remains unsolved. Recently, researchers such as [12, 18, 25] proposed mechanisms for adjusting the estimated locations of walls and floors to ensure that objects (represented by cuboids) are fully contained within the boundaries of the scene. Inspired by these approaches, we aim to intelligently search over viewpoint hypotheses. Intuitively, if we can fit an object configuration using a particular viewpoint hypothesis with high confidence, then that room layout is likely correct (i.e., it allows for objects to be properly matched). By searching over possible viewpoints, we aim to alleviate the brittleness of algorithms such as [7, 8, 28], which rely on hard decisions for the estimated viewpoint of an image. These types of geometry estimation algorithms are unable to recover when the room layout estimation process fails. Thus, in this work, we do not assume any individual viewpoint hypothesis is correct. Rather, we use our learned cost function to re-rank a set of room layout hypotheses, by jointly selecting a combination of furniture and camera parameters, which together best match the image.

We search over the top $N$ room layout hypotheses, returned by the algorithm of [11]. For each individual room layout, we use the estimated camera parameters corresponding to that room layout to render every 3D model from [1]. This approach scales linearly with the number of viewpoint hypotheses explored, and is trivially parallelizable. In all our experiments, we consider the top 20 results from [11]'s room layout algorithm. However, our approach is agnostic to the source of these viewpoint hypotheses, and additional hypotheses from [19, 27, 30] or any other algorithm could easily be incorporated to improve robustness.

Figure 3 illustrates the benefit of searching over various camera parameters. The top row shows the result of 3DNN using only the top-ranking room layout from [11]. Note that the failure to accurately estimate the height of the camera causes inserted objects to be incorrectly scaled. However, by not limiting ourselves to a single camera parameter hypothesis, we can automatically select a better room layout estimate, enabling a higher-scoring geometry match to be found. Figure 3(b) uses the 10th-ranking hypothesis from [11], and has the highest matching score using our learned cost function.

[1] Training was performed using 10-fold cross validation on a subset of the SUN Database [37], for which there exist LabelMe annotations [34].



(a) Result using only the top-ranking camera parameters from [11].

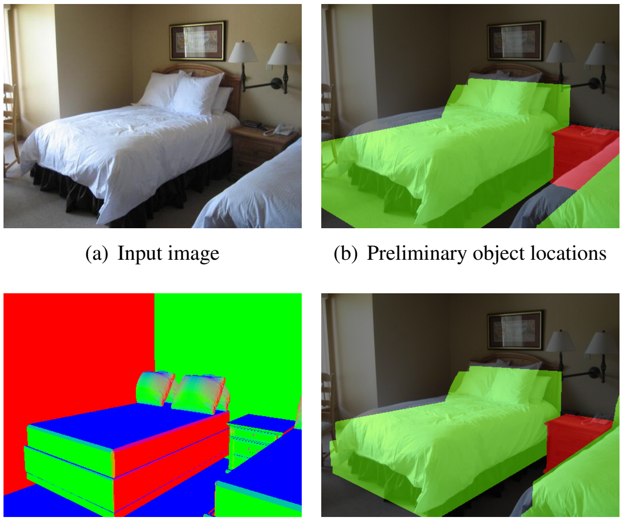
(b) Result after re-ranking the top-20 hypotheses from [11].

Figure 3. Example results highlighting the benefit of searching over viewpoint hypotheses. The top row shows the best matching scene geometry using the top-ranking room layout hypothesis of [11] (note the incorrect camera height estimate, causing objects to be rendered at the wrong scale). The bottom row show the best matching scene geometry after intelligently selecting the best room layout. For each result, matching 3D model surface normals are shown on the right next to the input image with overlayed object masks.

## 2.3. Geometry Refinement

In order to accurately segment objects in an image, and reason about their occlusions, we must precisely estimate their positions. However, a fundamental limitation of nearest-neighbor approaches is that their outputs are restricted to the space of object configurations seen in training data. This is a problem which has affected both 2D and 3D non-parametric methods. Recently, algorithms such as SIFT flow [20] have been developed to address this issue. The SIFT flow algorithm perturbs a matched image by warping the pixels to better align with the input image. However, because this approach warps pixels in the image plane, there is no longer a coherent 3D interpretation of the result. Thus, we propose a geometry refinement algorithm which is inherently 3D. Our method begins with a top-ranking 3D model for an image and searches for the best location of each object in 3D, such that the projection of these objects best align in the image plane, producing a more precise result.

We search for local refinements of the objects' locations which improve the overall geometric scene matching score, using a stochastic algorithm. In each iteration of the refinement, the locations of objects on the $x$–$y$ plane are perturbed (height off the ground remains fixed), by adding Gaussian noise ($\sigma$=1in) to the current objects' locations. If the adjusted objects' locations match the image better than the previous locations, the new locations are saved. This process repeats until convergence. In practice, a few hundred iterations are required to reach a final refined scene geometry.



(a) Input image

(b) Preliminary object locations

(c) Final refined geometry

(d) Final refined object locations

Figure 4. Effects of the geometry refinement process. Note that object boundaries are well-delineated after refinement.

Figure 4 highlights the effects of our geometry refinement process. Note the initial object locations in 4(b), when projected into the image plane do not align with the actual object boundaries. However, after refinement, in 4(d) the objects very precisely align with the image boundaries. The projected objects produce an excellent segmentation mask, and because the scene interpretation is inherently 3D, we can properly reason about occlusions and depth ordering.

## 3. Evaluation

We now evaluate the performance of our 3DNN approach. The goals of these experiments are three-fold: Firstly, we compare 3DNN with state-of-the-art appearance-based scene matching approaches. Additionally, we analyze the added benefit of each component of the 3DNN system: improved similarity features, geometry refinement and viewpoint selection. Lastly, we explore how the viewpoint invariance of 3DNN enables scene matching and the transfer of object labels using limited amounts of data.

We perform all experiments using the CMU 3D-Annotated Scene Database [1], containing 526 images of bedrooms and living rooms. All training was performed using 5-fold cross-validation to evaluate performance on all images in the dataset. Figure 5 includes example results of 3DNN on a wide variety of scenes. Note that we are able to produce accurate 3D models shown in the surface nor-

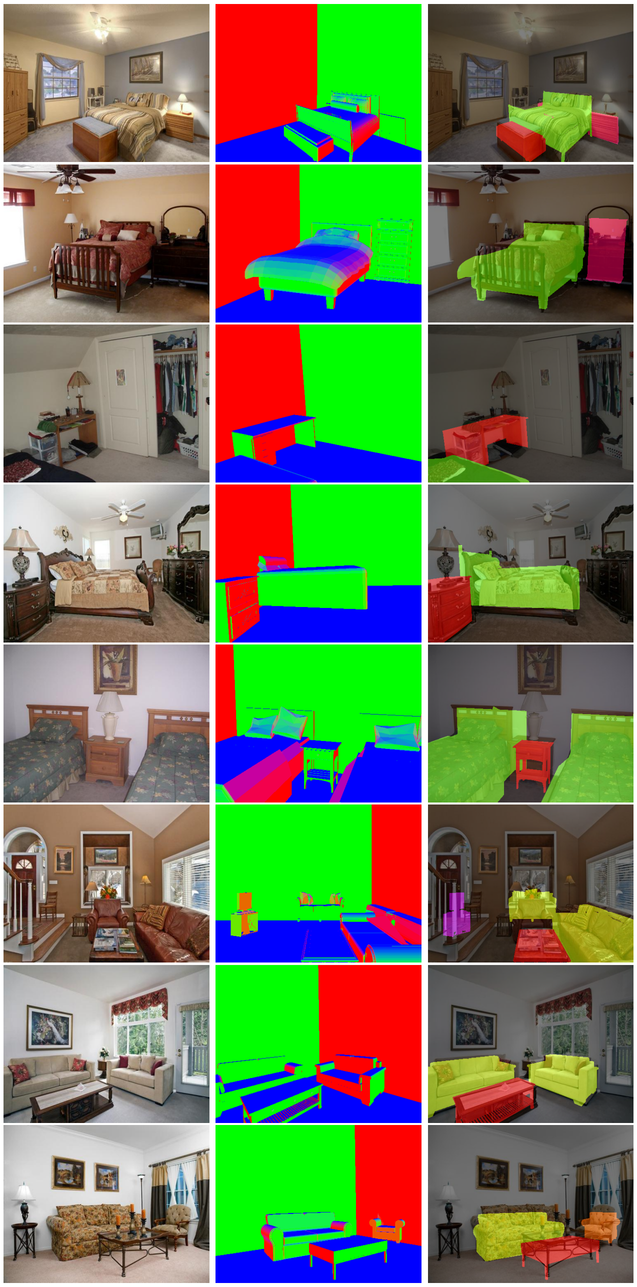


Figure 5. Qualitative results. From left to right: input images, surface normal renderings and overlaid object segmentation masks.

mal renderings beside each input image. In addition, each object's boundaries are well-delineated due to our geometry refinement stage, as indicated in the overlaid object segmentation masks.

### 3.1. Geometry Estimation

We now quantify the performance of 3DNN with a variety of baseline scene matching approaches, including state-of-the-art 2D nearest-neighbor approaches. We report performance using the two "Pixelwise Surface Normal Accuracy" metrics from [28], one measuring how accurately the surface normals of all pixels are predicted, the second evaluating only those pixels which correspond to objects in the ground-truth annotations.

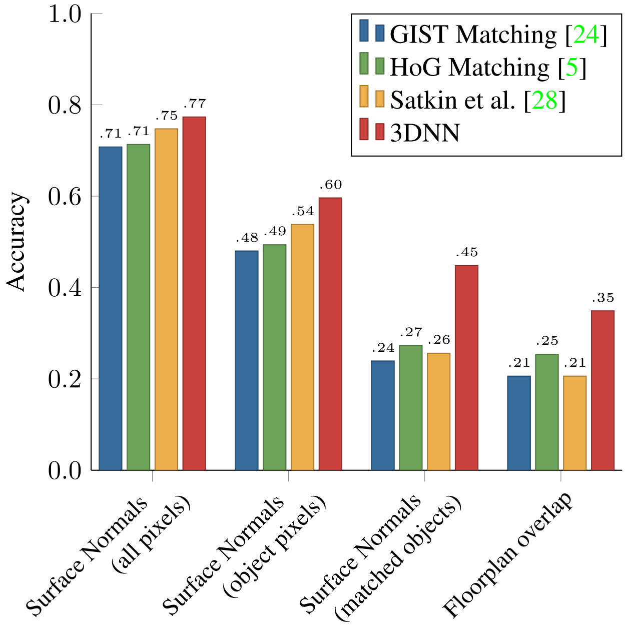


Figure 6. Comparison of 3DNN with state-of-the-art 2D nearest-neighbor approaches and the geometry matching algorithm of [28].

Although these metrics are informative for the task of surface normal prediction, they are unable to capture how accurately objects in an image are localized. For example, a horizontal surface corresponding to a bed in an image may be scored as "correct" even if the predicted scene contains no objects. This is because the horizontal floor has the same orientation as the bed's surface. Thus, we present results computed using a new metric, "Matched Objects Surface Normal Accuracy." This is a strict metric which requires two criteria to be met: For each pixel corresponding to objects in the ground-truth annotation, we must first correctly predict that there is an object at that location. We compute the dot product between ground-truth and predicted surface normals only at those pixels for which we "match" an object. Unmatched object pixels receive a score of 0. This metric is more sensitive to correctly predicting the exact locations and geometries of objects in a scene.

In [13] and [28], the authors present various metrics for how accurately their algorithms can predict the 3D freespace of a scene. These metrics require rectifying the predicted scene geometry, and are ill-posed when the estimated viewpoint deviates substantially from the ground-truth camera parameters. Thus, we develop another new metric to measure freespace prediction in the image plane: "Floorplan Overlap Score." For each object in the scene, we render its "footprint" by setting the height of each polygon to 0. A simple pixel-wise overlap score (intersection/union) of the object footprints can now be used to compare the ground-truth floorplan of a scene with our estimated scene geometry.

We compare 3DNN with our previous geometry matching approach [28] as well as two popular 2D nearest-neighbor approaches: GIST [24] and HoG [5] matching.[2] Figure 6, reports the results for 3DNN compared to each baseline, for the task of geometry estimation. Note that the geometry matching algorithm from [28] does not offer substantial improvements over the 2D nearest-neighbor approaches on the more challenging metrics (matched object surface normals and floorplan overlap score); however, 3DNN exhibits dramatic improvement on each of these metrics.

### 3.2. Object Detection and Segmentation

Our mechanism for inferring the structure of a scene in 3D provides us with rich information about the depth ordering and the occlusions of objects when projected onto the image plane. Thus, we should be able to not only detect the locations of objects, but also segment their spatial support in the image by precisely identifying their boundaries. To verify that using 3D cues is an attractive alternative for pixel-based object segmentation, we evaluate the per-pixel overlap score of the ground-truth and the object labels estimated by 3DNN.

Figure 7 analyzes the detection rate of 3DNN, compared to various appearance-based image matching baselines. We measure performance for the "bed" and "couch" categories, two of the most prominent objects in the CMU 3D-Annotated Scene Database. We vary the pixelwise overlap score threshold, and compute what percentage of beds are detected at each threshold. Note that at a stricter threshold of overlap score $\geq .75$, the baseline appearance-based approaches detect very few beds; however, 3DNN still performs well.

Naturally, 3DNN's ability to precisely segment objects is due in part to the geometry refinement stage. To analyze the benefits of this process, we measure the performance of 3DNN with and without the refinement stage. As anticipated, by refining the predicted locations of objects, we achieve a significant (on the order of 5%) boost in detection rate. For fair comparison, we run the SIFT flow algorithm (the state-of-the-art 2D refinement process) as a baseline. The SIFT flow algorithm of Liu et al. [20] has been shown to be a robust technique for aligning matched images. By warping each matched scene, SIFT flow refines the location of objects in the image plane, akin to our geometry refinement process. We apply the SIFT flow algorithm using code from [20]; this process takes the top-10 scene matches (using either GIST or HoG), warps each matched image, and computes the energy of each warping. We then re-rank the top-10 scene matches according to their SIFT flow energy, and score the top-ranking warped recall image. Although the SIFT flow process yields a significant boost in performance, the algorithm is still not as effective in accurately identifying and segmenting objects compared to 3DNN.

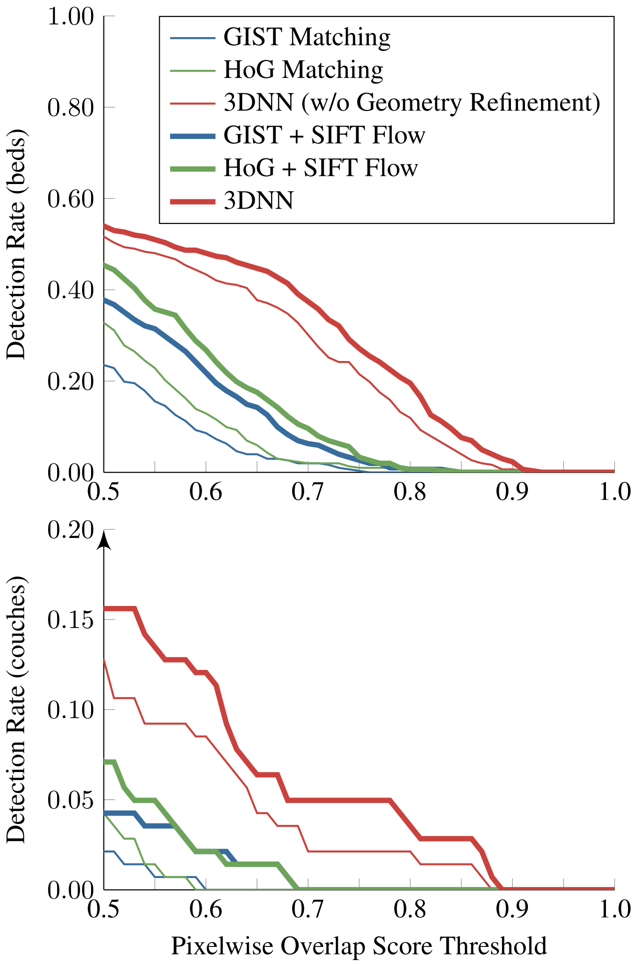


Figure 7. Object detection rate as a function of overlap score strictness for the "bed" and "couch" categories.

### 3.3. Viewpoint Selection and Geometry Refinement

In Section 2.2, we described our approach to automatically identify the viewpoint from which an image was captured, and in Section 2.3, we presented an algorithm to refine the locations of objects in 3D. We now evaluate how each of these stages affect the overall performance of the 3DNN algorithm.

Figure 8 shows the distribution of performance gains

[2] GIST: 4 × 4 blocks, 8 orientations (code from [24]). HoG: 20 × 20 blocks, 8 orientations (code from [35]).

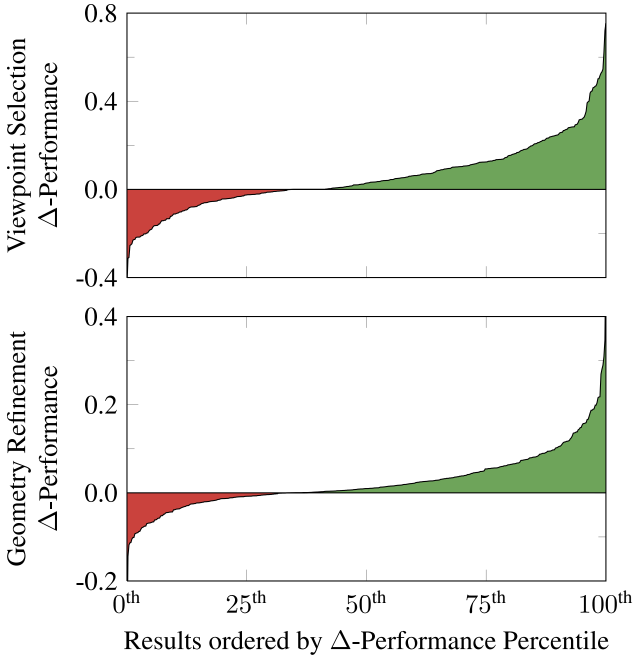


Figure 8. Distribution of improvements resulting from the Viewpoint Selection (top) and Geometry Refinement (bottom) processes. Performance is measured using the matched object surface normal scores. Green indicates a performance increase and examples in red resulted in a marginal performance decrease.

seen across all images in the CMU 3D-Annotated Scene Database as a result of the viewpoint selection and geometry refinement stages. The $y$-axis indicates how much the matched object surface normal score was affected via refinement or viewpoint selection. Note that for approximately two-thirds of the images, both the viewpoint selection and the refinement processes result in an improved scene geometry (indicated in green). Not only does viewpoint selection result in more accurate object geometries, it also improves the accuracy of room box estimation by re-ranking viewpoint hypotheses based on which room layout affords for the best 3D model matching (14.0% per-pixel error with viewpoint selection versus 16.4% error without viewpoint selection).

### 3.4. Dataset Size

It is well known that for appearance-based image matching to be effective, there must be a large recall corpus of images to match with [9, 33]. This is because the data set needs to include recall images captured from a similar viewpoint as the query image. On the contrary for 3DNN, the viewpoint and the geometry of the recall images are decoupled. Thus, each scene provides an exemplar which can be matched to images from any viewpoint.

We evaluate this by experimenting with the size of the recall corpus. Figure 9 shows how the performance of 3DNN increases as a function of dataset size, compared to GIST and HoG matching. We report results using two of the more challenging metrics: “matched object surface normal scores” (solid lines) and “floorplan overlap scores” (dashed lines). In these experiments, we consider recall dataset sizes between 1 and 500 images. For each dataset size, we select random subsets of images from the the full recall set, and report the performance of each algorithm on the smaller datasets. Due to the high variance in performance using small recall sets, we average performance across one-thousand random subsets of each size.

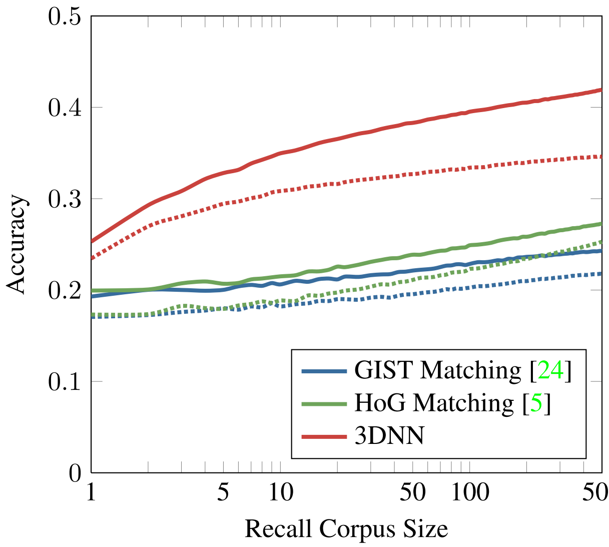


Figure 9. Accuracy as a function of dataset size. Solid lines indicate “matched objects surface normal score,” dotted lines indicate “floorplan overlap score.” Note the logarithmic $x$-axis.

There are two important properties of 3DNN we can identify from this graph. Firstly, note that the red plots for 3DNN start out with a higher accuracy (even for a dataset size of one image). This is because our algorithm starts by estimating the room layout of each image, identifying the locations of floors and walls. On the contrary, GIST and HoG matching do not incorporate this knowledge directly, and must infer the viewpoint of the scene by finding a similar image from the recall corpus.

Secondly, note that the curves for 3DNN are steeper than for the appearance-based approaches. This is because on average, each additional training image provides more information in its geometric form, than the raw pixels used in GIST or HoG matching. This indicates that performance is increasing more quickly as a function of the dataset size, and that fewer training examples are required to achieve the same level of performance using 3DNN compared to a traditional appearance-based 2D nearest-neighbor scene matching approach. Remarkably, 3DNN is able to achieve a noticeable performance boost using a recall set size of only 10 images or fewer, due to the algorithm’s ability to generalize across never-before-seen viewpoints.

## 4. Conclusion

In this paper, we presented the 3DNN algorithm. This approach differs from traditional 2D nearest-neighbor methods by decoupling the pose of the camera capturing an image and the underlying scene geometry, enabling the transfer of information across extreme viewpoint differences. We described our robust mechanism for simultaneously searching over camera parameters and scene geometries. In addition, we presented an algorithm for refining the locations of objects in 3D to produce precise results, and the features necessary to achieve this level of fine-grained alignment.

Because our approach is inherently 3D, we can properly reason about depth ordering and occlusions to produce accurate segmentations of detected objects. Thus, 3DNN achieves dramatic improvement over state-of-the-art approaches for the tasks of object detection, segmentation and geometry estimation. In addition, we demonstrated the ability of 3DNN to generalize to never-before-seen viewpoints, enabling non-parametric scene matching to be effective using orders of magnitude less data than traditional approaches.

## Acknowledgement

This research is supported by ONR MURI Grant N000141010934.

## References

[1] CMU 3D-Annotated Scene Database. http://cmu.satkin.com/bmvc2012/. 3, 4

[2] Microsoft Corp. Redmond WA. Kinect for Xbox 360. 2

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, May 2011. 3

[4] W. Choi, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013. 2

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *ICCV*, 2005. 5, 6, 7

[6] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *CIVR*, 2009. 3

[7] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single-view geometry. In *ECCV*, 2012. 2, 3

[8] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *CVPR*, 2011. 2, 3

[9] J. Hays and A. A. Efros. Scene completion using millions of photographs. In *SIGGRAPH*, 2007. 1, 7

[10] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *CVPR*, 2008. 1

[11] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 3, 4

[12] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010. 3

[13] V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. In *CVPR*, 2012. 2, 5

[14] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *ANN*, volume 1, pages 97 –102, 1999. 3

[15] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. In *IJCV*, 2007. 3

[16] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. In *SIGGRAPH Asia*, 2011. 2

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

[18] D. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010. 2, 3

[19] D. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 3

[20] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT Flow: dense correspondence across different scenes. In *ECCV*, 2008. 1, 4, 6

[21] D. Lowe. Three-dimensional object recognition from single two-dimensional images. In *Artificial Intelligence*, volume 31, 1987. 3

[22] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005. 3

[23] D. Munoz, J. A. Bagnell, and M. Hebert. Stacked hierarchical labeling. In *ECCV*, 2010. 3

[24] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. In *Progress in Brain Research*, 2006. 1, 3, 5, 6, 7

[25] L. D. Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *CVPR*, 2012. 2, 3

[26] L. D. Pero, J. Bowdish, E. Hartley, B. Kermgard, and K. Barnard. Understanding bayesian rooms using composite 3d object models. In *CVPR*, 2013. 2

[27] L. D. Pero, J. Guan, E. Brau, J. Schlecht, and K. Barnard. Sampling bedrooms. In *CVPR*, 2011. 3

[28] S. Satkin, J. Lin, and M. Hebert. Data-driven scene understanding from 3D models. In *BMVC*, 2012. 1, 2, 3, 5, 6

[29] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D scene structure from a single still image. *PAMI*, 2009. 2

[30] A. G. Schwing and R. Urtasun. Efficient Exact Inference for 3D Indoor Scene Understanding. In *ECCV*, 2012. 2, 3

[31] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Trans. Graph.*, Nov. 2012. 2

[32] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, 2010. 1

[33] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *PAMI*, Nov. 2008. 1, 7

[34] A. Torralba, B. C. Russell, and J. Yuen. Labelme: Online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484, 2010. 3

[35] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. Inverting and visualizing features for object detection. In *MIT Technical Report*, 2013. 6

[36] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, 2010. 2

[37] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 3

[38] S. Yu, H. Zhang, and J. Malik. Inferring spatial layout from a single image via depth-ordered grouping. In *CVPR Workshop*, 2008. 2

[39] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra. Interactive images: Cuboid proxies for smart image manipulation. *ACM Transactions on Graphics*, 31(4), 2012. 2