Active Learning for Piecewise Planar 3D Reconstruction

Adarsh Kowdle
Cornell University
apk64@cornell.edu

Yao-Jen Chang
Cornell University
ychang@cornell.edu

Andrew Gallagher
Eastman Kodak Company
andrew.gallagher@kodak.com

Tsuhan Chen
Cornell University
tsuhan@ece.cornell.edu

Abstract

This paper presents an active-learning algorithm for piecewise planar 3D reconstruction of a scene. While previous interactive algorithms require the user to provide tedious interactions to identify all the planes in the scene, we build on successful ideas from the automatic algorithms and introduce the idea of active learning, thereby improving the reconstructions while considerably reducing the effort. Our algorithm first attempts to obtain a piecewise planar reconstruction of the scene automatically through an energy minimization framework. The proposed active-learning algorithm then uses intuitive cues to quantify the uncertainty of the algorithm and suggest regions, querying the user to provide support for the uncertain regions via simple scribbles. These interactions are used to suitably update the algorithm, leading to better reconstructions. We show through machine experiments and a user study that the proposed approach can intelligently query users for interactions on informative regions, and users can achieve better reconstructions of the scene faster, especially for scenes with textureless surfaces lacking cues like lines which automatic algorithms rely on.

1. Introduction

There has been significant progress in recovering 3D structure of a scene given a few images of the scene from multiple poses. While a number of automatic algorithms [13–15, 24–26, 30, 32] have been shown to generate good models, the cues in a number of scenes are not sufficient to hypothesize a good structure of the scene. In particular, textureless surfaces, specular surfaces, and a lack of geometric cues such as lines, hinders their performance.

On the other hand, when we humans look at a scene we can much better discern the geometric structure that underlies the pixel data we view. Interactive 3D reconstruction algorithms try to exploit this, by requiring the user to provide interactions in the form of line drawings, 2D polygons, etc. [1, 3, 9, 10, 17, 31]. Such approaches are not only extremely cumbersome for a user, but also seem unnecessary given the performance of the automatic algorithms.

In this paper, we develop an active-learning algorithm for piecewise planar 3D reconstruction. We begin with patch based multiview stereo [14] as a good framework to reconstruct the scene. Using successful ideas from piecewise planar multiview stereo [24, 30], we cast the 3D reconstruction problem as an energy minimization problem over a graph of superpixels, with an *adaptive co-planar classifier* to model the smoothness in the graph. Inspired by the traditional definition of active learning, we quantify the uncertainty of the algorithm and propose high entropy regions for the user to provide interactions. User interactions provide support for these regions, update the co-planar classifier model and lead to improved reconstructions, and close the loop on the active-learning algorithm. Figure 1 gives an overview of the algorithm.

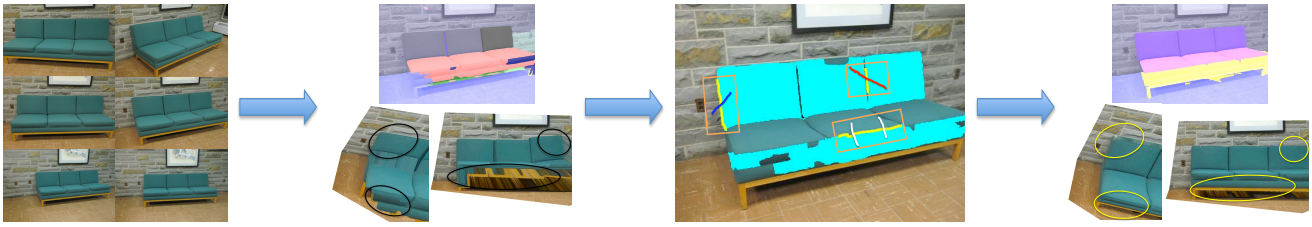
Contributions. Our primary contributions are:

- We believe we are the first to propose an active-learning framework for 3D reconstruction.
- We use very simple interactions from the user (co-planar, not-coplanar, and not-connected scribbles), which are very intuitive for any user to follow.
- We introduce a new adaptive co-planar classifier to model the smoothness in an energy-minimization framework.
- We demonstrate through user studies and machine experiments that our proposed active-learning algorithm significantly improves both the 3D reconstruction and the speed with which it is produced.

Organization. The rest of the paper is organized as follows: Section 2 discusses related work; Section 3 describes the different aspects of our active-learning algorithm in detail; Section 4 discusses the quantitative and qualitative results of our algorithm through machine experiments and user studies; Finally, Section 5 concludes the paper.

2. Related Work

Automatic algorithms. 3D reconstruction from multiple images is an active research topic in the computer vision community. There has been significant success with automatic algorithms [13–15, 24–26, 30, 32]. While some of



(a) (b) (c) (d)

Figure 1: (a) shows a set of multiview images of a scene; (b) shows the result of the automatic algorithm, the plane labeling shown on the top indicates the inaccurate labeling, the novel views of the 3D model are shown at the bottom with black circles showing the errors. (c) the proposed active-learning algorithm quantifies the uncertainty of the algorithm and detects the uncertain regions (in cyan), the uncertainty boxes (in orange) with the highlighted edges (in yellow) are used to query the user for support, the user provides any of three types of interactions within each box via simple scribbles across the highlighted edge, coplanar scribbles (red), not-coplanar scribbles (white) or not-connected scribbles (blue) as shown; (d) shows the result of the algorithm after incorporating the information provided by the user, plane labeling on top shows the improved labeling, the improved reconstruction is shown below through novel viewpoints with yellow circles illustrating the corrected geometry. (Best viewed in color).

these works [25,26] are geared towards video, some [15,32] are geared towards unordered photo collections on the internet. Most of these works require a large photo collection.

When the number of input images is restricted, these automatic algorithms fail to produce a dense reconstruction. A number of multiview stereo algorithms try to obtain a dense depth map for the scene from a set of images. A survey of these methods has been provided by Seitz *et al*. [28]. With a small set of images the reconstruction is incomplete, leaving holes on textureless surfaces and specular reflections. Planar approximations to the scene [12, 24, 30] help obtain more visually pleasing reconstructions. However, these algorithms use image features such as strong edges and lines, which may be absent in textureless surfaces. This has led to interactive algorithms.

Interactive algorithms. There have been many interactive 3D reconstruction algorithms [1,3,9,10,17,23,31,34]. The user interactions required range from providing feature correspondence, to providing plane boundaries and line models of the scene. Debevec *et al*. proposed an algorithm to reconstruct man-made architectures by marking the edges in the structure and by exploiting symmetry in man-made structures [10]. Hengel *et al*. [17] and Sinha *et al*. [31] require the user to provide a detailed line model of the object or marking all the 2D plane polygons in the scene, respectively; and reconstruct the scene by incorporating geometric information from structure-from-motion. Kowdle *et al*. [23] used user interactions in an interactive co-segmentation setup for object-of-interest 3D modeling. Srivastava *et al*. also used user interactions in the form of scribbles to help improve the 3D reconstruction obtained from a single image [33]. These interactive algorithms perform better than automatic algorithms; however, they require very involved interactions from the user, which can be quite tedious.

Active-learning algorithms. Active learning is a well-established subfield of machine learning [29], which has been shown to benefit a number of computer vision applications such as object categorization [20], image retrieval [16, 37], video classification [36], dataset annotation [8], and interactive co-segementation [4]; maximizing the knowledge gain while valuing the user effort [35].

Batra *et al*. [4] proposed an approach for interactive co-segmentation where, starting from the user interactions (scribbles) to identify the object-of-interest (OOI), the algorithm exploits a number of cues using the scribbles, and identifies informative regions to request the user for more interactions. Interactive 3D reconstruction, however, is not a trivial extension of this binary problem to multi-class segmentation. Rich information is already embedded in multiple images of a scene, which an automatic algorithm can fully utilize. However, the automatic algorithms fall short where texture or geometry cues cannot be easily identified from the images. Therefore, we formulate interactive 3D reconstruction as an error-correction and learning problem, where active-learning identifies uncertain regions, requests the user to provide geometric cues, and adapts the algorithm for the specific scene based on the user interactions.

3. Active-learning Algorithm

We refer to Figure 2 and consider the ingredients needed for an active-learning algorithm in the context of 3D reconstruction. The integral components are: an automatic 3D reconstruction algorithm with the ability to incorporate user feedback; an approach to quantify the uncertainty of the algorithm and sample the most informative queries for user feedback; the human oracle who provides suitable interactions in response to the query; and lastly, an approach to seamlessly incorporate the feedback from the user into the algorithm. We describe each of the above aspects with respect to our algorithm in detail in the following sections.

3.1. Automatic 3D reconstruction algorithm

We develop a piecewise planar 3D reconstruction algorithm using successful ideas from recent works [24, 30].

3.1.1 Dense plane hypothesis generation

We use patch-based multiview stereo (PMVS) by Furukawa *et al*. [14] as a preprocessing step, which although not as accurate as the sparse point cloud from SFM [32], provides

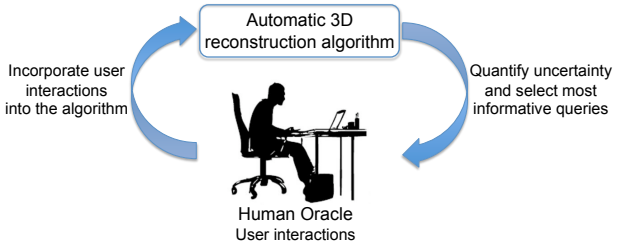


Figure 2: Active learning for 3D reconstruction.

a much denser set of points that span the scene. Similar to [30], we hypothesize dominant planes by analyzing the distribution of depths of the 3D points along each hypothesized normal (hypothesized using the vanishing directions). We break down an image into superpixels [1] and use the assumption that every superpixel would lie on a planar surface [24, 27]. Using these superpixels, we hypothesize additional planes by fitting planes to 3D points that project onto the same superpixel. In practice, we observe that this allows us to add new planes not hypothesized before as their normals are different from the dominant normal directions.

3.1.2 Energy minimization

The dense plane hypothesis stage results in about sixty planes. This dense set of discrete labels changes the piecewise planar reconstruction problem to a multi-label segmentation problem, formulated as an energy minimization problem over the superpixels and solved via graph cuts.

We build a graph, $G = (V, E)$, over the superpixels, with edges connecting adjacent superpixels. The image X is represented as a collection of n nodes (superpixels) to be labeled, $X = \{X_1, X_2, \ldots, X_n\}$. We define an energy function over the image as follows:

$$E(X) = \sum_{i \in V} E_i(X_i) + \lambda \sum_{(i,j) \in E} E_{ij}\left(X_i, X_j\right) \qquad (1)$$

where, $E_i(X_i)$ is the data term indicating the cost of assigning a superpixel to one of the labeled classes, and $E_{ij}(X_i, X_j)$ is the smoothness term used for penalizing label disagreement between neighbors.

Data term. For a particular view, we compute homographies for each plane to warp the other images to that view. We use normalized cross-correlation (NCC) to quantify the warp error. We refer the reader to [30] for more details. We compute the NCC using the superpixel as support at each pixel as opposed to a constant window. We also compute a color term that measures the mean color difference of each superpixel between the original and the warped image. We use a weighted combination of the two normalized terms as the data term with the weights tuned by observing the performance on one of the datasets.

Smoothness term: Co-planar classifier. We introduce an *adaptive* co-planar classifier to model the smoothness term. We learn a classifier that given a pair of adjacent superpixels returns a score representing the co-planarity of the superpixels. We use the geometric context dataset by Hoiem *et al*. [18] (with seven ground truth geometric labels). Adjacent superpixels with the same geometric label are used as positive data points while pairs with different labels, are used as negative data points. We note that adjacent superpixels lying on occluding 'parallel' planes would be bad data points, but, in practice this does not hinder the performance. We use *relative* features (difference features) such as color, texture, and shape features (more details about features in [18]) for each pair of superpixels as the feature vector for each data point and learn a logistic regression model. This model is continuously updated by the active-learning algorithm. We note that one can also use laser image data to learn a co-planar classifier by fitting planes to the laser data to obtain the samples needed [27].

We use a Contrast Sensitive Pott's Model to model the smoothness term.

$$E_{ij}\left(X_i, X_j\right) = I(X_i \neq X_j)\, exp\left(-\beta d_{ij}\right) \qquad (2)$$

The smoothness penalty when adjacent superpixels take different labels should be high when the contrast d_{ij} is low or when the superpixels are likely to be co-planar and high otherwise. Thus, given a pair of adjacent superpixels, using the learnt co-planar classifier, we obtain a score that represents the likelihood of this pair being co-planar. This score is used to model the contrast d_{ij} (1 - similarity score) in the Contrast Sensitive Pott's Model.

Finally, we use graph-cuts (with α-expansion) to compute the MAP labels for all superpixels, using the implementation by Bagon [2] and Boykov *et al*. [5, 6, 22].

3.2. What is the uncertainty?

An important aspect of an active-learning algorithm is to identify the uncertainty of the algorithm. Intuitively, since our algorithm follows an energy minimization framework to solve the multilabel problem over the graph of superpixels, we quantify the uncertainty of the algorithm with respect to the uncertainty in labeling the superpixels. At a high level, we evaluate the uncertainty of a superpixel in terms of *confidence* and *ambiguity*, described in detail below.

3.2.1 Confidence

Confidence quantifies how confident the algorithm is to assign a particular plane hypothesis to the superpixel. Low confidence superpixels represent high uncertainty regions, for example, occlusions. We obtain these regions via the energy minimization framework.

Motivated by the multi-view stereo work by Campbell *et al*. [7], we add an additional label to our set of discrete labels and refer to it as the *unknown* label. For every superpixel, X_i where $i \in V(all\ superpixels)$, the data term

[1] We use Felzenswalb and Huttenlocher's segmentation algorithm [11] to break each image down to about 400 superpixels.

$E_i(X_i)$ for the *unknown* label is set at a constant penalty. Intuitively, this penalty is large enough so it does not affect the data terms of the more confident superpixels while low enough so that low confidence superpixels are separated out. We use the median of all the data terms, which serves as a safe data term value in practice for the *unknown* label. As opposed to using a simple threshold on the data terms to determine low confidence regions, this approach gives the smoothness term an opportunity to try to derive support,when possible, for the low confidence superpixels from their neighbors. The superpixels that take the *unknown* label after the minimization are called *uncertain* superpixels.

3.2.2 Ambiguity

Ambiguity quantifies the uncertainty of the algorithm between different plane hypotheses. Superpixels that are ambiguous about multiple plane hypotheses represent high uncertainty regions, for example, textureless surfaces, specular surfaces, inaccurate plane hypotheses, etc. One approach to determine ambiguous data points in a multiclass labeling problem would be to analyze the data terms, using the idea that the entropy of the data terms of ambiguous data points would be high [19]. However, the entropy in the data terms is not sufficient to capture *all* the ambiguity because the effects of the smoothness term are ignored. We thus evaluate the ambiguity by determining the ambiguity of resulting MAP labeling after incorporating the effect of the smoothness. We do so by using the *Graph-cut uncertainty* similar to Batra *et al*. [4], as explained below.

Let the minimum energy $E(X)$ for the graph $G = (V, E)$ be E_{min}. Given the complete set of plane hypotheses (L labels), suppose that for a superpixel X_i the minimum energy label is l_i. We flip the label of superpixel X_i from l_i to one of the the other labels l_j in L and recompute the energy, $E_{i\to j}$ of the labeling. At each such flip stage, we compute the absolute difference between the minimum energy (E_{min}) and flip energy ($E_{i\to j}$),

$$E(X_i)_{(\Delta[i\to j])} = |(E_{min} - E_{i\to j})| \tag{3}$$

The ambiguity for every superpixel is computed by measuring the minimum of all such flip energy differences,

$$E(X_i)_{ambig} = \min_{j\in L\setminus i} E(X_i)_{(\Delta[i\to j])} \tag{4}$$

The intuition behind this is simple. If the algorithm does not have high ambiguity about assigning a particular plane hypothesis to a superpixel, the ambiguity energy difference, $E(X_i)_{ambig}$ should be high. However, if this value is low, it amounts to ambiguity between different plane hypotheses and hence uncertainty. We normalize the ambiguity energy differences and threshold that at 95% to obtain the top 5% of ambiguous superpixels. These are again called *uncertain* superpixels. We note that min-marginals by Kohli *et al*. [21] could also be used to capture ambiguity.

3.2.3 Region level uncertainty

In addition to the superpixel level uncertainty, we determine *region level* uncertainty. We determine regions (groups of superpixels) that take a particular independent plane label but have no support from the 3D point cloud, i.e. none of the 3D points project onto the region, and label them as uncertain. The intuition here is that, a set of superpixels with no support from the 3D points, taking their own independent plane label amounts to uncertainty.

3.2.4 Quantifying uncertainty

Grouping the uncertain superpixels to uncertain regions, we first identify and highlight all the boundary or support edges where user interaction might be needed. To ease the interactive process, we draw a box (uncertainty box) centered at this edge, scaled to be the larger of a minimum predefined size or two standard deviations of the edge size. Our active-learning algorithm then queries the user with the regions with the highest uncertainty or information gain. We thus need a metric to quantify the uncertainty of each box.

Consider *n* normals that span all the planes in the scene (from the initial plane hypothesis step). Superpixels are organized in increasing order of cost, based on the lowest cost the superpixel pays for adopting a particular normal (e.g. $C1_s, C2_s, \ldots, Cn_s$). This gives an indication about how certain it is about taking a particular normal. For a low uncertainty region, the value $C1_s$ would be considerably lower than the next best normal, i.e. $C2_s$.

Let R_i indicate the region under a box i that represents the set of all superpixels part of the uncertain region under the box. Let $R_{i,support}$ indicate the region under box i *not* part of the uncertain region under the box. Let $coplanarity(e)$ represent the score of the co-planar classifier for an edge e between two superpixels, and E_i indicate the set of all edges under a box i. The uncertainty is quantified through four terms: Ambiguity of the region in the box (A), Confidence of the support region in the box (F), Graph-cut uncertainty (GCU), and Co-planar classifier uncertainty (CoP).

$$A_i = \max_{s\in R_i} \frac{C1_s}{C2_s} \tag{5}$$

$$F_i = \max_{s\in R_{i,support}} (1 - C1_s) \tag{6}$$

$$GCU_i = \max_{s\in R_i} E(X_s)_{ambig} \tag{7}$$

$$CoP_i = \max_{e\in E_i} coplanarity(e) \tag{8}$$

Our final uncertainly score for box i, $Uncertainty_i$, is the sum of each of the component uncertainties defined in Eqn (5)-(8), using an equal weighting for each term as a fair setting. In practice, equal weights work well, as we show in Section 4. We rank the boxes according to this score and query the user with the top three uncertainty boxes for some

support. We note that we can achieve a steady improvement by querying the user with only *one* most uncertain box instead of the top three, however, this would need additional iterations of the algorithm, requiring additional user interactions and incurring processing overhead.

3.3. Putting the user in the loop

In our active-learning framework, given the uncertainty boxes, we wish to obtain user interactions in the form of support for the uncertain regions and incorporate this feedback into the algorithm to improve the reconstruction. The user provides one of three scribble based interactions described below, within each box as shown in Figure 3.

Connected and co-planar regions. When the edge highlighted in the uncertainty box is an edge between connected and co-planar regions, i.e. *same plane* (Box 1 in Figure 3), the user provides a scribble as support across the edge to indicate co-planarity, shown as the red scribble. We use this additional information to improve the support for the uncertain superpixels. This is done by adding long-range connections (non adjacent nodes) between the nodes (superpixels) scribbled on by the user to allow the algorithm to propagate the confident label to the uncertain superpixels.

Connected but not co-planar regions. In case the highlighted edge is an edge between connected but not co-planar regions, i.e. *different planes* (Box 2 in Figure 3), the algorithm would need cues about the edge shared between these two regions in order to hypothesize a good plane for the uncertain region. We do so by allowing the user to use two white scribbles across the edge to indicate the edge segment shared by the planes. Using the confident region, we can obtain the positions of these edge points in 3D. Given this information and the hypothesized normals (Section 3.1.1), we use a RANSAC based approach to find the best fit plane through the 3D edge marked by the user. We add this new plane hypothesis and estimate the corresponding data term as described in Section 3.1.2, adding hard constraints to ensure that the uncertain superpixels choose this new plane.

Not connected regions. If the highlighted uncertain edge corresponds to an edge between not connected regions, i.e. *occluding planes* (Box 3 in Figure 3), the user can indicate not-connected regions by using the blue scribble as shown. We incorporate this information into the algorithm by breaking edges between these superpixels in our graph, thereby hindering these regions from taking the same plane.

We incorporate all the interactions provided by the user and suitably reformulate the graph over superpixels. In addition to modifying the graph, we use this information as additional samples to update the co-planar classifier, which updates the smoothness term. Using the energy minimization framework (Section 3.1.2), we again obtain the MAP labels for the superpixels, which gracefully propagates the additional information given by the user. The process of obtaining uncertain regions, quantifying uncertainty, querying the user for support, and then updating the algorithm with the additional information is repeated using the new result, closing the loop on the active-learning algorithm.

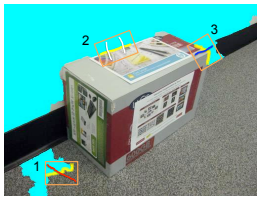


Figure 3: The user can provide three types of interactions to indicate coplanar regions (red), not-coplanar regions (white) and not-connected regions (blue) across the highlighted edge (yellow) within each uncertainty box (orange), to provide support for the uncertain regions (in cyan). (Best viewed in color).

4. Experiments and Results

In this section, we describe the datasets, the evaluation metric we use, and we discuss experiments to *quantitatively* evaluate the performance of the proposed active learning approach via machine experiments and a user study. We also discuss qualitative improvements in the reconstructions.

4.1. Datasets

We collect images spanning six scenes (each with about ten images) that lack geometric cues such as lines essential to the automatic algorithm and, include textureless surfaces or specular surfaces that hinder the performance of the automatic algorithm. We also use two standard datasets that have been used in prior automatic works [30].

4.2. Ground truth

To quantitatively evaluate the performance of the proposed active-learning algorithm, we first obtain pixel-wise ground truth segmentation of the planes for all the datasets. To capture some 3D information, we label ground truth normals for each segmented region. The ground truth pixel-wise segmentation along with their ground truth normals serves as a good quantitative indicator of the performance of the algorithm. Given the algorithm's result, we map each ground truth region to the largest label in that region in the algorithm's result, which agrees with the ground truth normal. Using these mapped labels we compute the pixel-wise labeling accuracy for each of the ground truth regions and compute the average accuracy across the datasets. We note that this metric can lead to inaccuracies in case of occluding parallel planes, however, it serves as a good metric to determine the relative performance in our experiments.

4.3. Machine experiments

In order to perform an exhaustive set of experiments to evaluate the various design choices, we develop a mechanism to generate *synthetic interactions*, which mimic the human user. For every uncertainty box queried by the algorithm, using the ground truth segmentation, normals, and

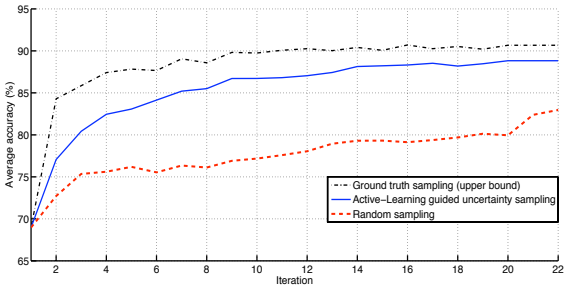


Figure 4: Machine experiments: Our proposed active-learning algorithm performs significantly better than random sampling and performs respectably compared to ground truth sampling. (Best viewed in color).

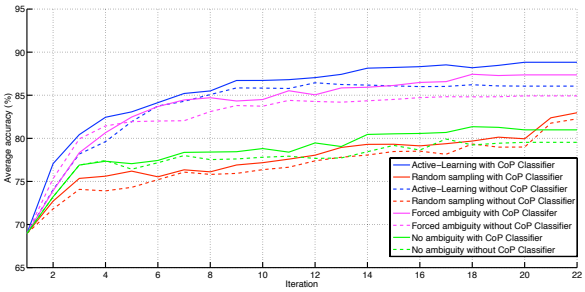


Figure 5: Machine experiments: Our proposed active-learning algorithm produces the most accurate reconstructions, validating our design choices. (Section 4.3). (Best viewed in color).

the occlusion boundaries (manually labeled), we provide one of three interactions described in Section 3.3. We note that an iteration in our experiments refers to providing the interactions in any three distinct locations (e.g. within the three uncertainty boxes in the active-learning experiment).

Performance of active learning. We evaluate the performance of the proposed active-learning algorithm against ground truth sampling (an upper bound) and a random sampling experiment as shown in Figure 4.

In the ground truth sampling experiment (black curve), at each iteration, we compute a 2D error map using the algorithm's output and the 'ground truth'. The machine interactions are then aimed to provide support to these error regions, beginning from the largest error region, in the order of decreasing size. This is a good upper bound since at each iteration we aim to achieve the best improvement by directly correcting the errors. The active-learning experiment (blue curve) evaluates the performance of the proposed algorithm in which, the machine interactions are guided by the uncertainty boxes indicated by the active-learning algorithm. In the random sampling experiment (red), we do not use the proposed active-learning algorithm to choose the uncertain regions, but instead randomly sample the uncertainty boxes along the segmentation boundaries.

We see from Figure 4 that the proposed active-learning algorithm performs much better than random sampling and, in addition, performs respectably when compared to the upper bound, given that it does not have the luxury to access ground truth while querying interactions. We also note that it can achieve the peak performance achieved

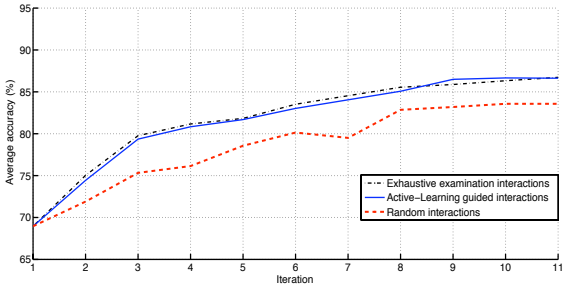


Figure 6: User study: The proposed active-learning algorithm not only out performs random interactions, but performs at par with exhaustive examination in significantly lower time (Section 4.4). (Best viewed in color).

by the random sampling at the end of more than twenty iterations in as few as four iterations.

Evaluating algorithm design choices. We evaluate the design choices we incorporated into the proposed active-learning algorithm. We first evaluate the effectiveness of incorporating 'ambiguity' to describe uncertainty. The solid green curve in Figure 5 shows the performance of the algorithm when we ignore ambiguity and only rely on the confidence measure. In comparison with our active-learning curve (solid blue), we see that when the algorithm quantifies only the low confidence regions as uncertain, it fails to capture several critical uncertain regions, leading to a very slow and minimal improvement in performance.

In our algorithm, we use graph-cut uncertainty to capture ambiguity. We evaluate this choice observing the performance when we use the entropy of the data terms to directly detect the ambiguous regions, forced ambiguity curve (solid magenta) in Figure 5. This firstly strengthens the importance of ambiguity on comparing with the no ambiguity green curve and in comparison with the active-learning curve (solid blue) shows that graph-cut uncertainty captures relevant regions which are missed by the forced ambiguity.

Lastly, we evaluate the adaptive co-planar classifier. In Figure 5, comparing the solid curves with the corresponding dashed curves shows that using the adaptive co-planar classifier (CoP) gives steady improvement in performance in all the experiments.

4.4. User study

We perform a user study with ten users and three experiments to evaluate the performance of the algorithm. Figure 6 shows the performance of the users. We restrict the number of iterations to reduce the effort of the users.

The first experiment is the random interactions experiment, in which we show the user the segmentation boundaries from the algorithm, however, with no indication about which regions are erroneous, as shown in Figure 7a. The user was instructed to provide three distinct interactions across any edge by observing the segmentations, with the only cue that each segmented region corresponds to a planar surface according to the algorithm. The *red* curve in Figure 6 shows the performance of the users.

The second experiment is the exhaustive examination experiment. Here, in addition to the segmentation boundaries, we color code the normals of each segment as shown in Figure 7b. The user was again instructed to provide three distinct interactions across any edge by observing the errors in the segmentations, with the normals guiding them towards erroneous regions. This leads to much better performance as seen by the *black* curve in Figure 6.

The last experiment evaluates the proposed active-learning algorithm. We show the user the uncertain regions detected by the algorithm in cyan. We highlight the uncertain edge in yellow, and draw three orange boxes to query the user for interactions, as shown in Figure 7c. The user was instructed to follow these orange boxes and provide interactions across the edges to provide support for the cyan regions. The *blue* curve in Figure 6 shows the performance.

The active-learning algorithm performs much better than random interactions and performs at par with the exhaustive examination, indicating that the algorithm effectively guides the user towards relevant uncertain regions. In addition, the average time taken by the user for each iteration reduced from 35.4 seconds in the exhaustive examination experiment to 23.2 seconds in the active-learning experiment, indicating that we achieve performance at par with exhaustive examination in significantly lower time.

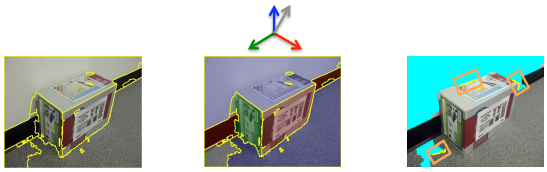
4.5. Qualitative analysis

In Figure 8, we show improvements in quality of the labeling and the 3D reconstructions as a result of incorporating the user interactions using the proposed algorithm [2].

Row 1 shows the improved reconstructions in presence of homogeneous surfaces like the wall and ground; Row 3 shows the improved result in case of an occluding object (planar approximation) and homogeneous background. Rows 4 and 5 show the output of the algorithm on public datasets used in prior work [30]. These are datasets in which the algorithm has enough cues to automatically reconstruct the scene and required minimal user interactions. These show that our automatic algorithm is not sub-optimal.

Relying on superpixels can hinder the performance in some cases. Note, for example, the error near the legs in row 3 due to a narrow superpixel leak. Row 6 demonstrates a failure case of the algorithm. In this example, there was a superpixel that leaks from the top of the tree onto the building. Since the uncertain edge we show the user always follows the superpixel boundaries, superpixel leaks can affect the performance. In this case, when queried, the user would always mark the regions as co-planar, resulting in a part of the tree labeled as part of the building behind it. However, we note that the proposed algorithm still performs significantly better than the automatic algorithm.

[2]Video: http://chenlab.ece.cornell.edu/projects/ActiveLearningFor3D/



(a) Random interactions (b) Exhaustive examination (c) Active-Learning

Figure 7: The three different user experiments conducted to evaluate the proposed algorithm (Section 4.4). (Best viewed in color).

5. Conclusions

We propose an active-learning algorithm for piecewise planar 3D reconstruction built on an energy minimization framework with a novel adaptive coplanar classifier that models the smoothness. The algorithm tries to reconstruct the scene automatically, quantifies uncertainty, and queries the user to provide support for the most uncertain regions via simple and intuitive interactions (coplanar, not-coplanar, and not-connected scribbles). The algorithm incorporates this support information and also updates the coplanar classifier model to obtain better reconstructions, thus closing the loop on the active-learning algorithm. We show through a user study and machine experiments that the proposed algorithm not only improves the reconstruction, but does so in significantly lower time than exhaustive examination.

We believe that this idea of active-learning for 3D reconstruction has a lot of potential beyond piecewise planar reconstructions. The framework of guiding the user to provide feedback to obtain better reconstructions can, not only be extended to multi-view stereo approaches (in which, other forms of interactions can aid dense *surface* reconstruction), but can also be used with works trying to obtain a 3D reconstruction from a single image, which has an inherent learning framework. Thus, the active-learning framework incorporates the positive aspects of both the automatic as well as the interactive algorithms, using the user inputs when and where needed, to render improved reconstructions.

Acknowledgments: The authors thank Anandram Sundar for the data annotation.

References

[1] Google sketchup: http://sketchup.google.com/, 2000. 1, 2

[2] S. Bagon. Matlab wrapper for graph cut, December 2006. 3

[3] A. Bartoli. A random sampling strategy for piecewise planar scene segmentation. *CVIU*, 105(1):42–59, 2007. 1, 2

[4] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. *CVPR*, 2010. 2, 4

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004. 3

[6] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *PAMI*, 20(12):1222–1239, 2001. 3

[7] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, 2008. 3

[8] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. In *ECCV*, 2008. 2

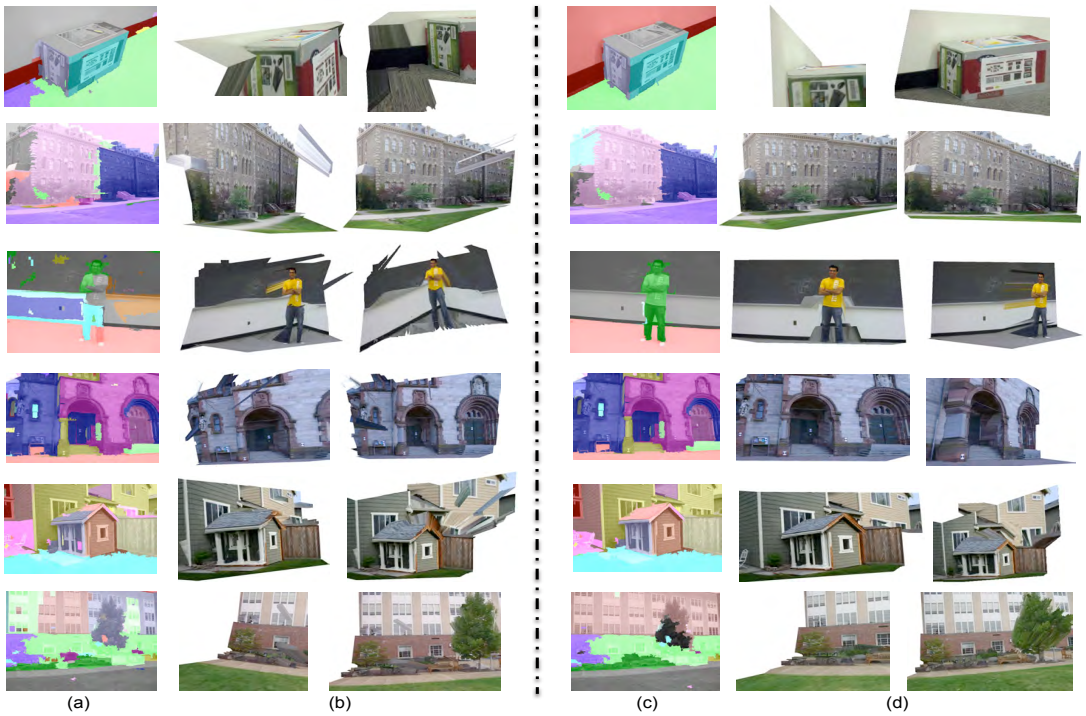


Figure 8: Qualitative results: (a) and (b) show the plane labeling and, novel views of 3D reconstruction from the automatic algorithm respectively; (c) and (d) shows the improved results using the active-learning algorithm respectively. (Best viewed in color).

[9] A. Criminisi, I. D. Reid, and A. Zisserman. Single view metrology. In *ICCV*, 1999. 1, 2

[10] P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH*, 1996. 1, 2

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. 3

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

[13] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010. 1

[14] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *PAMI*, 2009. 1, 2

[15] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007. 1, 2

[16] P. H. Gosselin and M. Cord. Active learning methods for interactive image retrieval. *IEEE Trans. on Image Processing*, 17(7):1200–1211, 2008. 2

[17] A. Hengel, A. R. Dick, T. ThormŁhlen, B. Ward, and P. H. S. Torr. Videotrace: rapid interactive scene modelling from video. *ACM Trans. Graph.*, 26(3):86, 2007. 1, 2

[18] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1), 2007. 3

[19] P. Jain and A. Kapoor. Active learning for large multi-class problems. In *CVPR*, pages 762–769, 2009. 4

[20] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, 2007. 2

[21] P. Kohli and P. H. S. Torr. Measuring uncertainty in graph cut solutions. *CVIU*, 112(1):30–38, 2008. 4

[22] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004. 3

[23] A. Kowdle, D. Batra, W. Chen, and T. Chen. iModel: Interactive co-segmentation for object of interest 3d modeling. In *ECCV - RMLE Workshop*, 2010. 2

[24] B. Micusík and J. Kosecká. Multi-view superpixel stereo in urban environments. *IJCV*, 89(1):106–119, 2010. 1, 2, 3

[25] M. Pollefeys, D. Nistr, J. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewnius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *IJCV*, 78(2-3):143–167, 2008. 1, 2

[26] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, V59(3):207–232, 2004. 1, 2

[27] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 31(5):824–840, 2009. 3

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

[29] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. 2

[30] S. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, 2009. 1, 2, 3, 5, 7

[31] S. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3d architectural modeling from unordered photo collections. *SIGGRAPH Asia*, 2008. 1, 2

[32] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH*, 2006. 1, 2

[33] S. Srivastava, A. Saxena, C. Theobalt, S. Thrun, and A. Y. Ng. i23 - rapid interactive 3d reconstruction from a single image. In *Vision, Modeling and Visualization*, 2009. 2

[34] P. F. Sturm and S. J. Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *BMVC*, 1999. 2

[35] S. Vijayanarasimhan, P. Jain, and K. Grauman. Far-sighted active learning on a budget for image and video recognition. In *CVPR*, 2010. 2

[36] R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *ICCV*, 2003. 2

[37] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, 2003. 2