LEARNING BOUNDARIES WITH COLOR AND DEPTH

Zhaoyin Jia, Andrew Gallagher, Tsuhan Chen

School of Electrical and Computer Engineering, Cornell University

ABSTRACT

To enable high-level understanding of a scene, it is important to understand the occlusion and connected boundaries of objects in the image. In this paper, we propose a new framework for inferring boundaries from color and depth information.

Even with depth information, it is not a trivial task to find and classify boundaries. Real-world depth images are noisy, especially at object boundaries, where our task is focused. Our approach uses features from both the color (which are sharp at object boundaries) and depth images (for providing geometric cues) to detect boundaries and classify them as occlusion or connected boundaries. We propose depth features based on surface fitting from sparse point clouds, and perform inference with a Conditional Random Field. One advantage of our approach is that occlusion and connected boundaries are identified with a single, common model.

Experiments show that our mid-level color and depth features outperform using either depth or color alone, and our method surpasses the performance of baseline boundary detection methods.

***Index Terms*—** Image edge detection, Image segmentation, Markov random fields.

1. INTRODUCTION

Object boundaries in images are important clues towards the high level interpretation of the scene [1] [2]. In general, three types of boundaries exist: (a) occlusion boundaries, which are the edges produced by one object occluding the other; (b) connected boundaries, which refer to the touching edges of two connecting objects; (c) homogenous boundaries, which are produced by the texture from the object. One example is shown in Fig. 1. In this paper, we learn to detect boundaries on color and depth image pairs.

Occlusion and connected boundaries are important edges for understanding the geometry of a scene as well as the layout of objects within the scene, shown in [3], [4], [5], [1], and [6]. However, identifying them in a robust manner is not an easy task. In some cases, prior semantic knowledge of the scene (e.g. "ground", "sky" or geometric context) has to be introduced for occlusion boundary recovery ([1] [7]). This additional knowledge may not be applicable for generic and complex scene images, as in [8] [9], or images of objects at a macro view, as shown in Fig. 1. This is where the 3D depth can play an important role and help most [10]. Specifically, in this paper we focus on the depth data from Kinect sensors.

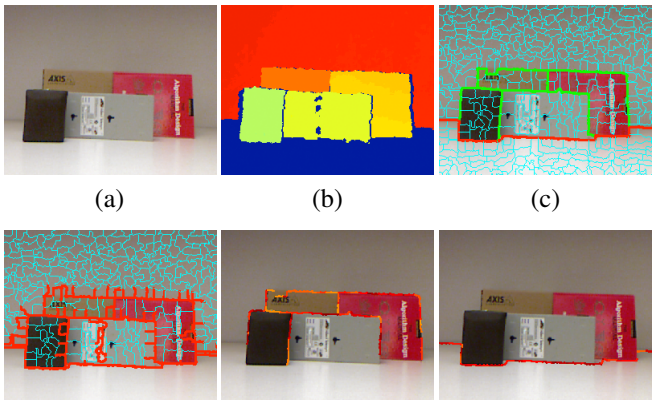


Fig. 1. Boundary examples: (a) a color and (b) a depth image from the Kinect sensor. (c) We extract all the possible edges by densely segmenting the color image, and label the following three types of boundaries: **homogeneous boundary** (cyan), **occlusion boundary** (green), and **connected boundary** (red). (d) presents the result when naively using the depth edge detection result (i.e. Canny edge detector on depth image) to label the occlusion boundary. Our learning based framework better detects the occlusion boundary (e), and the connected boundary (f). The color in (e) and (f) indicates the classification beliefs (redder indicates a higher belief).

However, to identify the occlusion and connected boundaries, simply "adding" the Kinect depth data may not solve the problem, because the depth information is quite noisy, especially in the region of the object boundaries [11] [12]. Fig. 1 (b) and (d) provide exemplar images. In general, depth images fail to produce the sharp edges common in color images, which are the regions that are most vital to our problem of reasoning about occlusion and connected boundaries. We propose our learning-based framework and develop novel 3D features to address this problem. We use a 3D surface-based segmentation to overcome the noisiness of the depth data. This segmentation step can avoid local decision pitfalls, and forms a better joint interpretation of the surfaces.

Further, we also generate features in the color domain, and concatenate all the features to supervise a Support Vector Machine (SVM). The output of the SVM is used as the unary

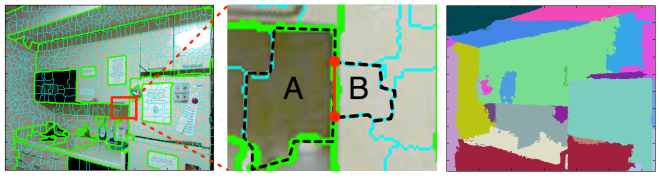
node in our graphical model. For a joint inference, we propose a Conditional Random Field (CRF) based framework, where pairwise potentials are learned by using the features computed on each junction of the boundaries. Our experiments on two different datasets prove the effectiveness of our new features, and the proposed CRF framework improves the inference accuracy compared to solely local decisions.

Related work: image-based boundary detection and segmentation has a long history. In Martin et al. [13] [14], low-level color and texture features are proposed for learning the segmentation of natural images, using a proposed human-labeled dataset [15]. Hoiem et al. [1] then extended this learning-based segmentation algorithm to the area of occlusion boundary detection and scene understanding. [1] showed that by detecting the occlusion boundary and the geometric labelings of the scene, it is easy to estimate the depth of a test image through analyzing the occlusion boundary between the object and the ground. Later [2], [6] and [16] demonstrated that this information can help other high-level interpretation of the scene, such as the object recognition. In this work, we further explore the occlusion and connected boundary detection with the help from both depth and color images.

As a mass-market depth sensor, Kinect has received wide interest from the computer vision community. Since its introduction, the color and depth information from this sensor have been applied to a wide range of computer vision tasks, such as environmental reconstruction [17], object recognition [18] [19] [20], object segmentation [8] [9], support-relation inference [9] and robotics [21]. In estimating human pose, [10] completely ignore the color information and exclusively relies on simple depth features for recognition.

2. COLOR AND DEPTH FEATURES

Initially, we densely over-segment the color image into superpixels using a watershed algorithm, shown in Fig. 1 (c). Then the task is to classify each small edge into one of the three boundary categories. We propose a set of color features x_c and depth features x_d, and train SVM based on them.

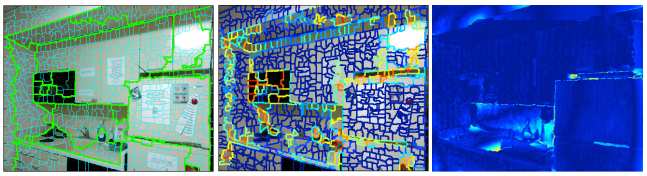


(a) (b)

Fig. 2. (a) **left:** Initial over-segmentation. The cyan edges are produced by the over-segmentation, and the green ones are the ground-truth occlusion boundaries. **right:** Each edge lies between two segments, e.g. the red edge is between segment A and B. Features are computed based on the edge and its two segments. (b) Surface segmentation results from the depth.

Edge curvature (ec): the curvature of the edge gives information for identifying a boundary. In an indoor scene, most man-made objects have structured boundaries. Homogenous boundaries are usually produced by the texture or noise, and are shaky and irregular. The actual occlusion or connected boundaries are composed of sharp straight lines. Examples are shown in Fig. 2 (a). We follow the edge histogram proposed in [5] to describe the edge curvature.

Surface segmentation and fitting: We applied the surface segmentation and fitting algorithm proposed in [22]. The intuition is to cluster the sparse point clouds by their Euclidean distance and estimated surface normals, and then apply surface fitting to refine the segmentation result. Exemplar results are shown in Fig. 2 (c). After this step, for each pixel p_i and its 3D points P_i, we have acquired its 3D surface group C_i, and the corresponding surface function $f_{C_i}(x, y, z)$.



(a) (b) (c)

Fig. 3. (a) Occlusion boundaries labeled from the surface segmentation algorithm (in section: **Surface segmentation label**). (b) Surface label distribution on each edge. (c) Surface fitting errors on each pixel.

Surface segmentation label (sl): this feature uses the result from surface segmentation algorithm [22] to predict boundaries: for each edge e and its two segments A_e, B_e, we find the most frequent surface labels of the pixels within each segment, $C(A_e)$ and $C(B_e)$. If edge e lies on two different surfaces, we mark it as positive to indicate an occlusion or connected boundary, otherwise we label it negative to indicate a homogenous boundary. Fig. 3 (a) shows the labeling result from this method.

Surface distribution (sd): for the segments A_e, B_e that edge e separates, we also retrieve the 3D surface label distribution for each segment, and include this as another feature.

For one segment, we calculate the ratio between the occurrence of the most frequent surface label C_{max} and the total number pixels. For example, if in segment A_e, 90% of its pixels belong to surface C_1, then the feature value for this segment will be $sd(A_e) = 0.9$. This feature effectively measures the confidence of the previous surface segmentation algorithm. We compute this feature on an edge basis by taking the average of the surface distribution value of each edge's two segments: $sd(e) = (sd(A_e) + sd(B_e))/2$. Fig. 3 (b) gives an example of the surface distribution value for each edge: the more red an edge is, the smaller its surface distribution value is, which indicates less confidence in the surface segmentation.

Fitting error (se, ee): for each 3D point P, we also retrieve its surface function f_C that P lies on and compute the fit error, measured in 3D space. One example of the fit error distribu-

tion is shown in Fig. 3 (c), in which the red color indicates higher fitting errors.

The surface segmentation errors usually occur at occlusion or connected boundaries where the surface function has a poor fit to the 3D points. The distribution of the fit errors gives a clue about the type of the boundary, e.g., for occlusion boundaries, the 3D points may have larger fitting errors than the points that lie on a connected boundary, because there is a large depth change from the occlusion.

We compute two types of fitting error: for each edge e (**ee**) and its surfaces A_e and B_e (**se**): the pixel-wise fit errors along the edge and within each segment. We histogram the error distribution into 40 bins with equal intervals in log space from 0 to 10 centimeters, and use this as one of the depth features.

Neighboring difference (nd): we compute two types of differences between edge e's segment A_e and B_e: (a) average depth difference, and (b) angle between the surface normals.

The average depth difference can help because occlusion boundaries may result in higher depth difference between their two sides, while connected and homogenous boundaries may expect lower values.

To compute the angle between the surface normals for segments A_e and B_e, we approximately fit a plane locally for the 3D points with each segment, and calculate the angle between their normals. The intuition is as follows: the two segments of a connected boundary may have an orientation difference around 90°. However, the occlusion and homogeneous boundaries tend to have their neighboring segments facing similar directions.

3. CONDITIONAL RANDOM FIELD

We propose a Conditional Random Field for a joint inference of boundaries. Given the initial over-segmentation, we define the unary potential, $\phi(y_i|x_i)$, and the pairwise potential $\psi(y_i, y_j|x_{i,j})$ over each edge e. y indicates the edge labels, e.g., homogenous or occlusion/connected boundaries, and x indicates the feature vector. i and j refer to the neighboring edges. Then the task is to minimize the following energy function E:

$$E = \sum_i \phi(y_i|x_i) + \sum_{i,j} \psi(y_i, y_j|x_{i,j}). \quad (1)$$

Since our color and depth features are computed on edge basis, we can concatenate them into one feature vector $x = [x_c, x_d]$, and train a Support Vector Regression f_u for the local prediction. We use linear SVM regression for fast training and testing speed. After that, we retrieve the probability $P(y|x)$ of the edge label y given the feature x, using the regression f_u, and use the negative log likelihood of this probability as the unary potential $\phi(y|x)$ in our CRF.

We learn the pairwise potential ψ for any two neighboring edge i and j that connected in the color image, meeting at a junction with position p_{jun}. First, we concatenate both color and depth features from edge i and j: $x_i = [x_{c,i}, x_{d,i}]$ and $x_j = [x_{c,j}, x_{d,j}]$. This serves as the basic feature set to learn the pairwise potential. Further, we use additional features to describe the neighboring edge relation.

Oriented SIFT: different types of boundaries will give different texture shapes at the meeting junction, and we compute a SIFT descriptor at the junction to capture such information. The underlying idea is as follows: if two edges are both occlusion/connected boundaries, then the SIFT descriptor will have a consistent large value along the boundary direction. In contrast, homogenous boundaries produce texture of random and irregular patterns, and lead to a more uniform distribution for each bin value in the SIFT descriptor. Therefore this descriptor can provide additional texture information at the junction where edges meet. Besides that, In computing the features, SIFT descriptors use a histogram approach, which can tolerate some the noise in the boundary as well as a little mis-alignment of the depth image [5].

We compute this feature as follows: the SIFT descriptor is centered at the meeting junction position p_{jun}, and aligned with the direction of each edge. Then we compute a fixed size (5 pixels per bin) SIFT descriptor for each edge on both the color (converted into gray scale to follow the convention of SIFT) and depth image. After that, we concatenate the descriptors on different image domains. This forms the oriented SIFT feature x_s to learn pairwise potentials.

4. EXPERIMENTS

We experiments on two datasets: depth-order dataset [5], and the public NYU Knect dataset of indoor scenes [8]. We compare our final approach (**crf**) with the following algorithms:

- **base:** uses the color and texture features proposed in [1]. This serves as the basic feature set for color image boundary detection (no depth). For the following algorithms, we add different feature sets to this **base** approach, e.g. the following approaches are feature set in addition to **base**.
- **ec** to **nd** We add each feature **(ec, sd, se, ee, nd)** individually in addition to **base**.
- **all:** we combine all the feature sets.
- **crf:** our final CRF model.

Depth order dataset: We manually label the occlusion and connected boundaries for 200 images in this dataset, and split the dataset into two halves for separate training and testing.

We evaluate different algorithm by comparing the average precision of detecting boundaries, and present the results in Table.1, top two row. Overall, it proves that our proposed framework works for both occlusion and connected boundary detections. Without depth information, using the base features from [1] provides a lower bound on performance, and our edge curvature feature still improves by around 3% performance in average precision.

	base	ec	sl	sd	se	ee	nd	all	crf
d-conn	46.0	48.3	51.9	63.2	79.6	78.7	68.5	88.0	**90.3**
d-occ	59.1	60.2	66.1	78.2	76.7	67.6	78.1	86.9	**89.1**
n-occ	50.9	51.1	53.5	53.6	54.5	53.3	55.0	58.1	**60.1**

Table 1. Average precision for different approaches on our kinect depth order dataset: connected boundary (**d-conn**), occlusion boundary (**d-occ**), and occlusion boundary detection result on NYU depth dataset (**n-occ**).

Adding depth features definitely helps the tasks. Directly using the surface segmentation in [22] **sl** gives 6% boost for classifying connected boundaries, and 8% for occlusion boundaries. In addition, our proposed depth feature sets (**sd**,**se**,**ee**,**nd**) also produce better results than **base**, giving around 70% to 80% average precisions. When combining all the feature sets (**all**), it outperforms the individual feature set by a large margin, leading to an average precision of nearly 90% for both occlusion and connected boundary detection. Compared to the individual depth features (columns from **sl** to **nd**), the combined one (**all**) achieves at least a 10% improvement.

Finally, our proposed CRF model still improves the performance by 2% compared with **all**, and gives the best result of all the approaches, because it encourages continuity between boundaries. Some example images of our boundary detection results using **crf** are shown in Fig. 4. It shows that our learning framework reliably identify both occlusion and connected boundaries in different scenarios.

Fig. 4. Boundary detection result using the proposed algorithm. It reliably detects the connected (top two) and occlusion (bottom two) boundaries in different scenarios.The color indicates the confidence in classification. The more red it is, the larger the belief.

4.1. NYU dataset

We also experiment on the public NYU depth dataset [8]. This dataset only provides the object segmentation, and we approximately use it as the occlusion boundary to fit our task. This dataset contains 2284 frames of Kinect image pairs. However, many of them are of the same scene and near consecutive frames in a video. Therefore, we sample the dataset into 600 images, ensuring the remaining images are not too similar to each other. After that, we follow the same settings as the previous experiments for training and testing. The experiment results are shown in Table.1, bottom row.

Our proposed edge curvature feature improves the performance over the baseline color feature. The proposed depth feature sets (**ec** to **nd**) show the benefit of bringing the depth information. They achieve around 55% in average precision, and all outperform the color-only scheme by 2% to 6%. The final combined CRF model gives the best performance, achieves near 10% absolute boost from 51% to 61% comparing to **base**, and has 5% improvements in average precision to the individual depth feature sets. Some results are shown in Fig. 5.

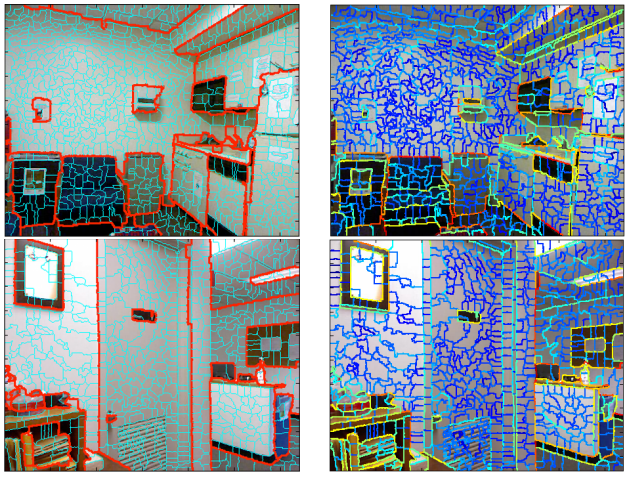


Fig. 5. Experiment results on NYU dataset. Ground-truth labels are on the left, with red indicates the occlusion boundaries, and cyan indicates the homogenous boundaries. The testing results are shown on the right. Heat map indicates the belief: the more red an edge is, the more likely it is an occlusion boundary.

5. CONCLUSION

As the types of imaging modalities increase, it will be important to combine various types of data to solve vision problems. This paper demonstrates a solution for classifying image boundaries from color and depth that is significantly improved over using one or the other type of information exclusively. We perform surface segmentation on the depth data, and generate a set of novel depth features based on the surface. We propose a CRF framework for a joint inference on boundaries. Experiments show that our proposed feature sets and the learning framework outperform the baselines.

6. REFERENCES

[1] D. Hoiem, A. N. Stein, A. A. Efros, and M. Hebert, "Recovering occlusion boundaries from a single image," in *ICCV*, 2007.

[2] D Hoiem, A A Efros, and M Hebert, "Closing the loop in scene interpretation," in *CVPR*, 2008.

[3] I. Endres and D. Hoiem, "Category independent object proposals," in *ECCV 2010*, 2010, vol. 6315.

[4] M Dimiccoli and P Salembier, "Exploiting T-junctions for depth segregation in single images," in *ICASSP*, 2009.

[5] Z. Jia, A. C. Gallagher, Y. Chang, and T. Chen, "A learning-based framework for depth ordering," in *CVPR*, 2012.

[6] B Liu, S Gould, and D Koller, "Single image depth estimation from predicted semantic labels," in *CVPR*, 2010.

[7] D Hoiem, A A Efros, and M Hebert, "Geometric context from a single image," in *ICCV*, 2005.

[8] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *ICCV-3DRR workshop*, 2011.

[9] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[10] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, 2011.

[11] B. Huhle, T. Schairer, P. Jenke, and W. Strasser, "Robust non-local denoising of colored depth data," in *Workshop of Time of Flight Camera based Computer Vision, CVPR*, 2008.

[12] I. Reisner-Kollmann and S. Maierhofer, "Consolidation of multiple depth maps," in *ICCV Workshops on Consumer Depth Cameras for Computer Vision*, 2011.

[13] D. R. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *PAMI*, vol. 26, no. 5, pp. 530–549, 2004.

[14] E. Borenstein and S. Ullman, "Learning to segment," in *ECCV*, 2004.

[15] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001.

[16] D Hoiem, A A Efros, and M Hebert, "Putting objects in perspective," *IJCV*, 2008.

[17] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox, "Interactive 3D modeling of indoor environments with a consumer depth camera," in *UbiComp*, 2011.

[18] L. Bo, K. Lai, X. Ren, and D. Fox, "Object recognition with hierarchical kernel descriptors," in *CVPR*, 2011.

[19] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3-D object dataset: Putting the kinect to work," in *ICCV Workshops on Consumer Depth Cameras for Computer Vision*, 2011.

[20] A. Anand, H. Koppula, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *NIPS*, 2011.

[21] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *IJRR*, 2012.

[22] Z. Jia, Y. Chang, T. Lin, and T. Chen, "Dense interpolation of 3d points based on surface and color," in *ICIP*, 2011.