CUT-AND-FOLD: AUTOMATIC 3D MODELING FROM A SINGLE IMAGE

Guihang Wang[1], *Xuejin Chen*[1], *Si Chen*[2]

[1]CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System,
University of Science and Technology of China, Hefei Anhui, China.
[2]University of California San Diego, La Jolla California, USA.
wgh2012@mail.ustc.edu.cn, xjchen99@ustc.edu.cn, sic046@ucsd.edu

ABSTRACT

This paper presents a novel approach to automatically generate a plausible 3D model from a single outdoor image. Outdoor scene can be modeled as a set of planar billboards such as ground, sky and vertical trees or buildings. This Cut-and-Fold system takes the advantages of the statistical classification method for high-level geometric labeling of the image regions and the automatic image segmentation algorithm for detailed boundaries of the geometric regions to generate accurate geometric labeling and realistic rendering of novel views. Taking a single image as input, our method first roughly classifies each sub-region to the geometric categories: vertical, ground or sky. Then image segmentation algorithm is employed to refine the labeling results and generate accurate boundaries for the geometric regions. After cutting the image into several regions with their geometric labels, the system folds the regions into 3D billboards with accurate boundaries. The experimental results demonstrate that our method is capable of creating plausible 3D models automatically for a wide range of outdoor scenes.

Index Terms— scene understanding, classification, image segmentation, image-based modeling

1. INTRODUCTION

Throughout the last two decades, image-based modeling (IBM) has witnessed tremendous progress. IBM techniques are widely used in computer games, movies and virtual reality since images have the largest potential to generate realistic rendering. Lots of proposed IBM techniques are able to create realistic walkthrough environments while requiring multi-view images, tedious manual work, and special devices. It is therefore very difficult for common users to create their own virtual models easily or for industry to create 3D models quickly and automatically.

We thank the anonymous reviewers for their invaluable suggestions. This work is supported in part by grants from NSFC (No. 61103135) and the Fundamental Research Funds for the Central universities (No. WK2100230003 and No. WK2100060011).

In this paper, we present a novel approach for *automatically* creating *plausible 3D model* with *accurate region boundaries* from a single image. Our approach follows the well-known cut-and-fold illustration process to create 3D model from an outdoor scene image. An outdoor scene can be typically simplified as a set of planes, which are divided into three geometric categories: ground, sky and vertical. To cut an image into different geometric regions, many classification algorithms have been proposed using visual features, such as SIFT [1] and HoG [2]. Due to the inherent ambiguity of the local features, it is difficult for the existing algorithms to get a 100% classification rate. The *coarse* labeling for the three geometric categories will lead to many visual artifacts appear at the wrong-labeled subregions and especially the boundaries when creating the virtual walkthrough.

In contrast, image segmentation algorithms have been well-studied for several decades. While maximizing the inter-region difference and intra-region similarity, global segmentation algorithms tend to find a good boundary between different regions. Given the rough labeling of each geometric category as initial segmentation, our approach uses the GrabCut [3] to iteratively refine the label for each pixel to generate an accurate cut for each region.

The main contribution of our approach is the geometric labeling algorithm combining image segmentation and statistic learning method, which keeps the detailed boundaries of large geometric regions. As a result, our system creates a plausible 3D model from a single image by cutting accurate geometric regions first and then folding them up on the ground plane. Fig. 7 shows the accurate labeled geometric categories and 3D model rendered under novel views from a single image.

2. RELATED WORK

3D modeling from a single image has been studied for decades. Theoretically, it is impossible to accurately recover the 3D geometry from a single image without any other hints. Many approaches have been proposed to model a scene with user interactions from a single image. The Tour into the Picture system [4] provides an efficient graphical user inter-

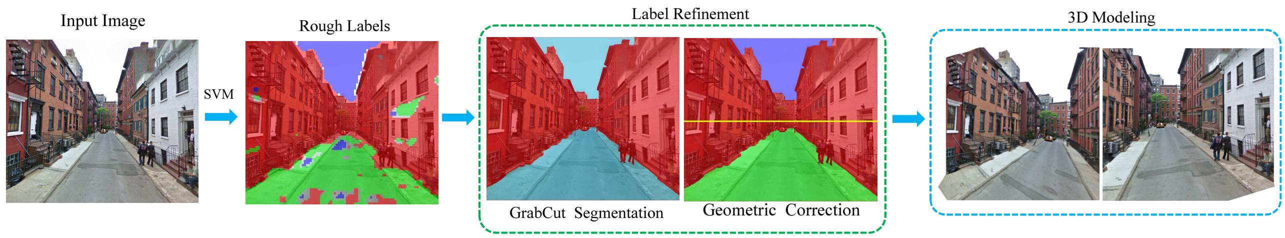

Fig. 1. Overview of our automatic modeling system.

face by employing a spidery mesh and model the scene simply as floor, ceiling, backdrop and two side planes. This method produces impressive results but works well only on one-point perspective scenes. Afterwards, more accurate approaches were proposed using more complicated user interactions and geometric constraints [5, 6, 7, 8]. By comparison, our work focuses on *fully automatic* 3D modeling.

A few systems use pre-collected data to train a system to recognize different geometric regions for automatic 3D modeling. Photo Popup system [9] proposes a fully automatic method for creating a 3D model from a single outdoor image. *Geometric labeling*, which refers to assigning each pixel to one of three geometric classes: ground, vertical and sky, as defined by Hoiem et al. [10], plays a critical role in the automatic scene reconstruction. It has been successfully addressed by a number of works. A wide variety of features (e.g., color, texture, position and perspective) are combined to a multiple segmentation framework to generate surface layout [11]. Gould et al. [12] propose a region-based model which combines appearance and scene geometry to achieve multi-class image segmentation and geometric reasoning. [13] estimates the depth information from a single image and reconstruct 3D structure without manual interactions. It requires depth images for training, which are not easily accessible for common users. Recently, some approaches [14, 15] try to combine different classifiers for related tasks of scene understanding to improve performance on some or all tasks. These methods pay their attentions to complicated features or classification framework to label the pixels as accurate as possible. By comparison, our novel framework combines statistic learning and image segmentation for geometric labeling while taking simple features and generate competitive results. Furthermore, our framework is general and any state-of-the-art classiers can be applied here as the fore end's classifier which can produce large portion of correct labeled pixels for later labeling refinement stage.

Image segmentation, which aims to divide the image into semantic regions, has been well studied for decades. Many image segmentation algorithms such as Intelligent Scissors [16], Bayes matting [17], Graph Cut [18] have been proved to be effective to generate accurate region boundaries. Carsten et al. [3] introduce GrabCut to segment objects with the user simply drags a box around the foreground. However, all these algorithms need manual interactions implying the initial foreground and background regions. Saliency Cut [19] is proposed to overcome this problem. They use saliency detection to auto-label the foreground and background. But saliency information is not robust for labeling a complex image because of the texture or shadow in the scene. Our system employs the image segmentation using roughly labeled geometric categories replacing the manual interaction to achieve a fully automatic process and more precise 3D models.

Combination of Segmentation and Classification has drawn attention to researchers recently. A classification model is presented [20] for segmentation using the complicated Gestalt cues including contour, texture, brightness and continuation. Object-specific shape information is one of the most important features used in these combination algorithms [21, 22, 23]. However, it is non-trivial to describe the shape of geometric regions in the outdoor scenes. Therefore, we take the full advantage of segmentation to generate accurate geometric regions using classification as reference.

3. OVERVIEW

To assign accurate labels to pixels and get precise boundaries, our system consists of three stages: roughly labeling, label refinement, and 3D modeling, as Fig. 1 shown. In the first stage, the pixels of the input image are labeled using statistical learning algorithm. The labeling errors occur at the boundaries and the inner part of each geometric region. In the second stage, we adopt the GrabCut segmentation algorithm to refine the precise boundaries of different geometric regions automatically. The roughly labeled sub-regions of each geometric category are set as the initialization of the iterative segmentation. In the third stage, the geometric regions are cut and folded according to the estimated camera settings. By combining classification and image segmentation, this approach does not require additional input images or manual interactions to generate far more accurate geometric labeling.

4. ROUGHLY LABELING

In the first stage, we roughly label the input image into three geometric classes: *vertical, ground* and *sky*. We use the software LIBSVM [24] and LIBLINEAR [25] to train a statistic classifier individually and apply one of them to generate rough labels in the first stage. Each training image is segmented and labeled into three geometric classes. Different from general image classification methods for the entire image category, we assign label to each pixel for accurately cutting the image. Therefore, each image is divided into small overlapping rectangular regions (patches) of the same size (30×40) with 10-pixel step. For training, we choose the patch in which all the pixels are labeled as the same category.

Features. Compared with the complicated features used in [10, 11, 14, 13, 12, 15], we use quite simple but effective features for classification. Dense SIFT features are extracted from each patch, and then clustered to 1,000 visual words using k-means. For each patch, we extract Bag of Visual Words F_{bog} [26] (1000 bins), the color histogram H_c (30bins) and height h, which are cascaded to a 1,031D descriptor.

Classification. With the descriptor of each patch as input, the classifier outputs three posteriori probabilities for belonging into the three geometric classes. Since we divide the image into overlapping patches, each pixel is contained in several patches. For 30×40 patches with step size 10, each 10×10 region must be contained in multiple patches. The pixel within the image is contained in 12 patches while the border region on the image is contained in less patches. Considering a 10×10 regions r_i in N patches, we calculate the posterior probabilities of each patch P_i for the three geometric classes: $p(g|P_i)$, $p(s|P_i)$ and $p(v|P_i)$ respectively. The small region's probability for the three classes is defined as the average from the N patches. We assign the class with the largest probability p^* to all the pixels in this small region if $p^* > 0.5$. Otherwise, the pixels are defined as unknown class.

Due to the inherent ambiguity of the local features and the statistic classifier, it leads to rough labels for the pixels. There are many labeling errors, especially at region boundary, as Fig. 2(b) shows. We can simply remove the very small regions as noise to correct some labeling errors. However, there are still a large amount of labeling errors at the region boundaries. In order to produce more realistic rendering result, more precise boundaries are desired. Image segmentation, which combines the local contrast and region consistency to generate consistent regions, is a good option to refine the rough labeling results.

5. LABEL REFINEMENT

GrabCut [3] is an efficient, interactive binary segmentation algorithm. Given the user simply drags a rectangle around of foreground, GrabCut system uses the pixels in/out the rectangle to estimate the Gaussian Mixture Model of color distribution for foreground/background. Then it iteratively updates the color distribution of the foreground and background and segments the image using graph-cut. For complex scene, more user interactions are required to get the accurate segmentation. Requiring user interaction makes this interactive segmentation algorithm difficult to use in our system. We proposed a variant of GrabCut to automatic segment the images using the rough labeling as initialization.

(a) Original image.

(b) Rough labels.

Fig. 2. Roughly labeled image. The regions in red, green, and blue color belong to vertical, ground and sky categories, respectively. The remaining pixels are unknown category.

In our system, the image should be segmented into three classes. We do a four-pass GrabCut to combine the binary segments, as shown in Fig. 3. In the first three pass, we run a binary segmentation considering each geometry category as foreground while the others are background. We combine the three segmentation into one and generate the final segmentation using geometric correction.

5.1. Initialization with Reliable Pixels

To initialize the foreground/background for each geometric category, we choose the reliable pixels labeled to the geometric regions. While saying "reliable", we choose the pixels with high posteriori probabilities for each geometric category. From the roughly labeled result described in Section 4, we get a set of pixels P for each class. The reliable pixels for each class c are selected as following:

- **step 1** Sort the pixels in P in descent order according to their probabilities to this geometric category. Remove the last $k\%$ pixels from P.
- **step 2** Generate a mask M of value of 1 for the pixels in P.
- **step 3** Compute the areas of the connected regions in M. Detect the holes whose areas are less than α in M. Fill these holes with value of 1.
- **step 4** Erode regions in M using a structuring element. β is the size of the element. We set the eroded pixels P_{lc} as a set likely to be the category c. The remaining pixels are set P_c to definitely belong to this category. We set (k, α, β) to $(0, 5000, 10)$ for sky and ground, and $(10, 5000, 20)$ for vertical in our experiments.

5.2. Four-pass Segmentation

After selecting reliable pixels, we get P_s, P_{ls}, P_g, P_{lg}, P_v and P_{lv} for the three categories. We run a four-pass segmentation to cut the image into precise geometric regions. Firstly, we run a binary segmentation considering each category as foreground and the others as background for the three categories individually. For example, we choose the sky category

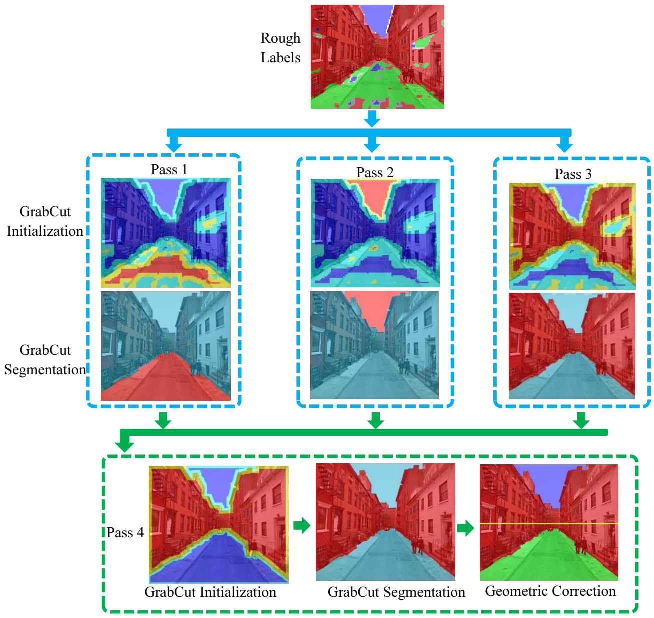

Fig. 3. Four-pass GrabCut segmentation. The regions in red, yellow, purple and sky-blue colors are set as foreground, likely to be foreground, background and likely to be background, respectively.

as foreground. We initialize the GrabCut with pixels in P_s with label 1 for foreground and pixels in the other two categories P_g, P_v with label 0 as background. The pixels in P_{ls} likely to be foreground are set to 3. All the other pixels are set to 2 as likely to be background. The initial color distribution model GMM for each category is estimated using its reliable pixels for the GrabCut segmentation.

We run GrabCut for each category and get three binary maps (Pass 1, 2, 3 in Fig. 3). Individual segmentation generally generates quite good labeling results. However, a few errors occur at the fine structure of the vertical objects, the landscape scenes and so on. Therefore, the three segmentations are combined to another GrabCut segmentation in the fourth pass. The vertical regions are set as foreground, ground and sky regions are background. The pixels belonging to sky, ground and vertical in the individual segmentation maps are defined as $\hat{P}_s$, $\hat{P}_g$ and $\hat{P}_v$, respectively. We erode $\hat{P}_s$, $\hat{P}_g$ to generate P'_s and P'_g with label 0, P'_{lg} and P'_{ls} with label 2. As well, we erode $\hat{P}_v$ to generate P'_v label 1 and P'_{lv} with label 3. With this initialization, GrabCut segmentation is employed again to generate accurate vertical regions and background.

5.3. Geometric Correction

In order to divide the background into sky and ground regions, we estimate the horizon for geometric correction. We take the concept described in [27] and simplify the method to estimate the horizon with labeled semantic geometric regions generated in the individual segmentation step. Hence we can estimate a reasonable horizon to divide the background regions after the fourth pass binary segmentation into sky and ground regions.

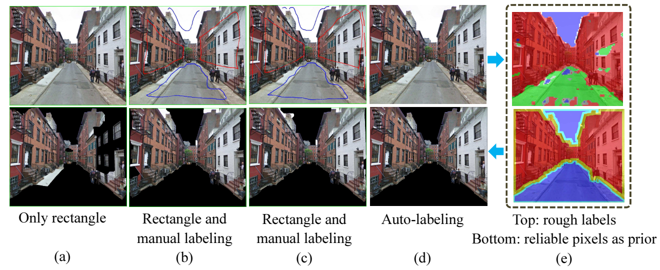

Fig. 4. Comparison of GrabCut initialized in different manual ways and our approach. (a) Segmentation with the green rectangle as prior. (b)(c) Segmentation with user drawn strokes for foreground (red) and background (blue). (d) Automatic segmentation with the roughly labeling as prior.

With our four-pass segmentation, the image is automatically segmented in to three categories. Fig. 4 demonstrates that our automatic labeling approach generates very good boundaries for each class while the GrabCut requires non-trivial user interactions to achieve a good segmentation.

6. CREATING THE 3D MODEL

So far we get precise segmentation for the three geometric classes (ground, sky and vertical) and a proper estimation of the horizon. Though it is impossible to recover its exact geometry, we can create a reasonable scaled model by the horizon and setting the remaining parameters to constants.

Camera Settings. We assume a very simple pin-hole camera with the optical center projected to the image center, zero skew and unit affine ratio. While we set the world coordinate system the same as the camera's coordinate system, we use a 1.431 radians of field of view. The scale of the reconstructed model can be determined by the ground plane's height, which we set to $y_g = -5$. Given this projection settings, the 3D coordinate (x, y, z) of each pixel (u, v) on the ground can be computed by backprojecting it with $y = -5$.

Cuts and Folds. To cut the vertical regions, we divide them into a set of vertical planes connected and orthogonal to the ground plane. From the automatic labeling results, we get detailed ground-vertical plane boundaries. These boundaries are fit to polylines using Douglas-Peucker algorithm [28]. Once the vertical-ground polyline is determined, we compute the 3D position of each vertex on the ground plane. Each line segment of the polyline is regarded as intersection line of a vertical plane with the ground. We then generate a vertical plane with its normal $n = \widehat{V_{ab}} \times (0, 1, 0)^T$ and $d = -nV_a$. The sky-vertical boundary is then back projected into this plane and get its 3D-coordinates. Finally each plane is mapped with the corresponding texture.

7. RESULTS

We test our system on two challenging datasets: Popup [9] (dataset1) and GC [10](dataset2). For each dataset, we use the same training images and test images as the state-of-the-art algorithm (for comparison). Fig. 6 shows that our method produces more accurate geometric labeling for various type-

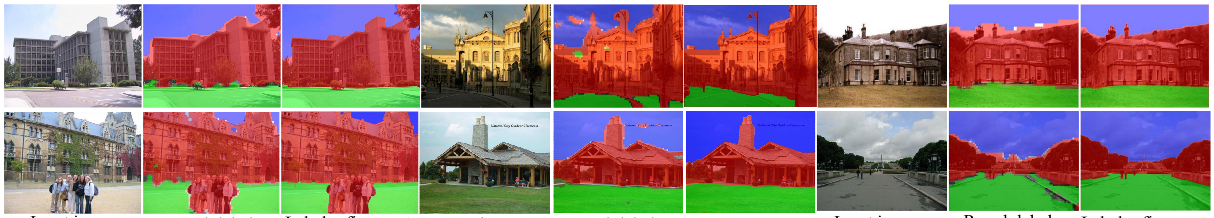

Fig. 5. Labeling results.

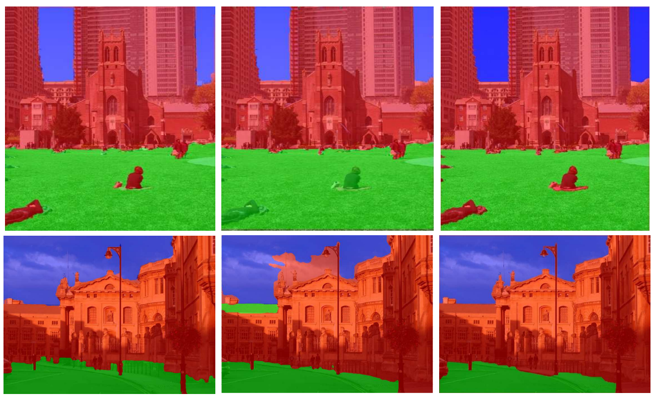

Fig. 6. Geometric labeling results generated by our method and the multiple segmentations [11].

s of objects in the outdoor scenes. Fig. 5 and Fig. 7 show more labeling results and 3D modeling results generated by our system on a number of images.

While Popup correctly labeled 87% of the pixels in the 62 test images(dataset1), 92% of the pixels are correctly labeled in our system. Table 1 gives the confusion matrix of the three geometric classes. Our label refinement method highly improves the labeling accuracy compared with Popup. The main reason is that we use the image segmentation for the optimal region consistency while the statistic classifier produces large portion of correct labeled pixels. We perform 6-fold cross validation on the dataset2 which contains 300 images. Table 2 shows that our system is effective and robust to generates competitive results, compared with the state-of-art methods. Note that the FE-CCM generates a 0.2% higher accuracy than our methods. However, we use quite simple features and a general framework, which makes the process faster. The simplicity of our method is one of its most appealing aspects.

Effect of label refinement method. Though our system uses simple features, the experimental results demonstrate that the proposed approach is really effective. The main reason is that our system uses a Four-pass GrabCut segmentation method for label refinement. As shown in shown in Table 3, the proposed label refinement method can improve the labeling accuracy by about 4% for dataset1 and 3% for dataset2 compared to the roughly labeling results. Noted that our statistic classifier uses quite simple features, Table 3 provides hope that the more reliable pixels provided by statistic classifier, the better performance our system can achieve. Since any state-of-the-art classiers can be applied here as the fore end's statistic classifier, our framework is general in this aspect.

We test our algorithm on a 3.10 GHz CPU with 4GB RAM. The total processing time for an 800x600 size image is about 30 seconds using unoptimized MATLAB code, which makes it acceptable for generating real-time virtual walkthrough.

Table 1. Confusion matrix for the geometric classes.

	Vertical	Ground	Sky
Vertical	90.56	6.18	3.27
Ground	7.97	91.68	0.34
Sky	2.49	0.21	97.31

Table 2. Comparison with other methods.

Method	Dataset2 (% of accuracy)
Baseline [10]	86.0
Multiple Segmentations [11]	88.1
CCM [14]	87.0
Region-based [12]	86.9
FE-CCM [15]	88.9
Ours with Linear SVM	88.3
Ours with Non-linear SVM	88.7

Table 3. Effect of label refinement.

		Roughly Labeling	Label Refinement
Dataset 1	Linear SVM	86.8	90.6
	SVM	87.7	92.3
Dataset 2	Linear SVM	84.9	88.3
	SVM	85.2	88.7

8. CONCLUSION

We present a novel automatic system to correctly label the geometric regions and generate a plausible 3D model for virtual walkthrough. Statistic learning and image segmentation are integrated to take the advantages of the high-level recognition and visual consistency of regions. Realistic models can be generated automatically for a wide range of outdoor scenes.

References

[1] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

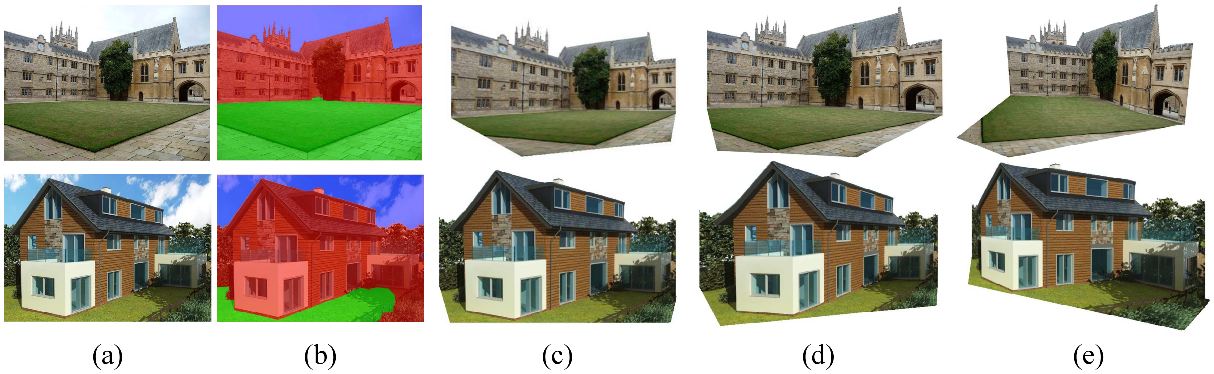

Fig. 7. Results: (a)Input image. (b)Labeling result. (c)(d)(e) Realistic rendering of novel views.

[2] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 886–893.

[3] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (TOG)*. ACM, 2004, vol. 23, pp. 309–314.

[4] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai, "Tour into the picture: using a spidery mesh interface to make animation from a single image," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 225–232.

[5] David Liebowitz, Antonio Criminisi, and Andrew Zisserman, "Creating architectural models from images," in *Computer Graphics Forum*. Wiley Online Library, 1999, vol. 18, pp. 39–50.

[6] Antonio Criminisi, Ian Reid, and Andrew Zisserman, "Single view metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, 2000.

[7] Richard Hartley and Andrew Zisserman, "Multiple view geometry in computer vision," *Robotica*, vol. 23, no. 02, pp. 271–271, 2005.

[8] Li Zhang, Guillaume Dugas-Phocion, Jean-Sebastien Samson, and Steven M Seitz, "Single-view modelling of free-form scenes," *The Journal of Visualization and Computer Animation*, vol. 13, no. 4, pp. 225–235, 2002.

[9] Derek Hoiem, Alexei A Efros, and Martial Hebert, "Automatic photo pop-up," in *ACM Transactions on Graphics (TOG)*. ACM, 2005, vol. 24, pp. 577–584.

[10] Derek Hoiem, Alexei A. Efros, and Martial Hebert, "Geometric context from a single image," in *International Conference of Computer Vision (ICCV)*. 2005, vol. 1, pp. 654 – 661, IEEE.

[11] Derek Hoiem, Alexei A. Efros, and Martial Hebert, "Recovering surface layout from an image," *Int. J. Comput. Vision*, vol. 75, no. 1, pp. 151–172, 2007.

[12] Stephen Gould, Richard Fulton, and Daphne Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1–8.

[13] Ashutosh Saxena, Min Sun, and Andrew Y Ng, "Make3d: Learning 3d scene structure from a single still image," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 824–840, 2009.

[14] Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller, "Cascaded classification models: Combining models for holistic scene understanding," in *Advances in Neural Information Processing Systems*, 2008, pp. 641–648.

[15] Congcong Li, Adarsh Kowdle, Ashutosh Saxena, and Tsuhan Chen, "Toward holistic scene understanding: Feedback enabled cascaded classification models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1394–1408, 2012.

[16] Eric N Mortensen and William A Barrett, "Intelligent scissors for image composition," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 191–198.

[17] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski, "A bayesian approach to digital matting," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. IEEE, 2001, vol. 2, pp. 264–271.

[18] Yuri Y Boykov and M-P Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. IEEE, 2001, vol. 1, pp. 105–112.

[19] Yu Fu, Jian Cheng, Zhenglong Li, and Hanqing Lu, "Saliency cuts: An automatic approach to object segmentation," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[20] Xiaofeng Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. Ninth IEEE International Conference on Computer Vision*, 2003, pp. 10–17.

[21] Eran Borenstein and Shimon Ullman, "Class-specific, top-down segmentation," in *ECCV*, 2002, pp. 109–122.

[22] M Pawan Kumar, PHS Ton, and Andrew Zisserman, "Obj cut," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 18–25.

[23] John Winn and Nebojsa Jojic, "Locus: Learning object classes with unsupervised segmentation," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. IEEE, 2005, vol. 1, pp. 756–763.

[24] Chih-Chung Chang and Chih-Jen Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27, 2011.

[25] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[26] Jun Yang, Yu-Gang Jiang, Alexander G. Hauptmann, and Chong-Wah Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, New York, NY, USA, 2007, MIR '07, pp. 197–206, ACM.

[27] Beyang Liu, Stephen Gould, and Daphne Koller, "Single image depth estimation from predicted semantic labels," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1253–1260.

[28] David H Douglas and Thomas K Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.