STRUCTURAL EDGE LEARNING FOR 3-D RECONSTRUCTION FROM A SINGLE STILL IMAGE

Nan Hu

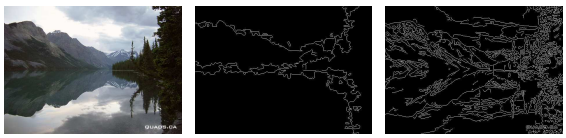
Stanford University
Electrical Engineering
nanhu@stanford.edu

ABSTRACT

Learning 3-D scene structure from a single still image has become a hot research topic recently, during which edges played a very important role as they provided critical information about the structures. While not all the intensity edges are useful, existing 3-D reconstruction methods suffered heavily when not differentiating structural edges with non-structural ones. In this report, we consider learning structural edges rather than edges from intensity values of the image. Through supervised learning, the learnt edges as shown in this report carries more stuctural information and less noise than intensity edges. The comparison of two kinds of edges are also shown in the report.

1. INTRODUCTION

Classical work on 3-D reconstruction mainly focus on using methods like stereovision [3] and structure from motion [4], which requires two (or more) images and triangulation for depth estimation. Recently, monocular vision has arouse interests of many researchers, since there are numerous monocular cues that could be utilized when estimating the depth. An good example of analogy is the human eyes. Even when using only one eye, people can always have a very good estimate the relative depth about the scene. Saxena et al. [1] proposed a method to estimate the 3-D depth information from a single still image. Their proposed approach took an image over-segmemted into a number of small planes called superpixels, and used Markov Random Field (MRF) to infer both the 3-D position and the orientation of each of these small planes. The MRF parameters were trained using supervised learning. Their algorithm was able to infer qualititively correct and visually pleasing 3-D models automatically for about 65% of the testing set. However, their methods suffered from not differentiating structural edges for non-structural ones, since not all the intensity edges carried useful information about the structured scene. For the small planes, edges are very useful in estimating the connectedness of those planes. Without the correct structural edges detected, visually far apart objects could be connected together by the MRF. Thus, it is imperative to develop an algorithm to learn structural edges to improve the performance of the their system.



(a) (b) (c)

Fig. 1. Comparison of edges. (a) Original image; (b) Edge after manually merging connected parts; (c) Edge by Canny's method.

In this report, we proposed a supervised learning algorithm to learn the structural edge from non-structural ones. For each image in the training set, Felzenszwalb's method [5] was used to group similar pixels together to get superpixels. Texture-based summary statistic features, and superpixel shape and location based features were then extracted from each superpixels in the image. By manually labeling connected objects, superpixels representing connected objects are merged together and the structural edges are at the boundary of the superpixels representing disconnected objects. Through our labeling, structural edges are differentiated from non-structural ones. Logistic regression parameters were afterwards trained using the labeled

structural and non-structural edges. As can be seen from our testing sets, our proposed method successfully inferred edges with more structural information than the intensity edges as well as less noise.

The rest of the report is organized as follows. In Section 2, our proposed models and the feature filters will be described. The experimental results are shown in Section 3. In addition, some discussion and the conclusion will be presented in Section 4

2. MODEL DESCRIPTION

2.1. 3-D Scene Reconstruction Model

Since the features extracted has a strong connection with Ashutosh's 3-D reconstruction model [1, 2], a brief introduction of his model will be first described. In [1], a polygonal mesh is used to represent the 3-D model, where the world is assumed to be composed of a set of small planes. In detail, given an image of scene, small homogeneous regines were found in the image, by using Felzenszwalb's method [5]. Those small regines are called "Superpixels". Such regions represent a coherent region in the scene with all the pixels having similiar properties, and hence is a reasonable representation. Markov Random Field (MRF) are then used to infer both the 3-d position and orientation information of the superpixels. Thus, each node in the MRF is correponding to a superpixel in the image. The MRF model is supposed to capture the following properties:

- **Image Features and Depth**: the image features of a superpixel bear some relation to the depth(and orientation) of the superpixel.
- **Connected Structure**: Except in case of occlusion, neighboring superpixels are more likely to be connected together.
- **Co-planar Stucture**: Neighboring superpixels are more likely to belong to the same plane, if they have similiar features and if there are no edges between them.
- **Co-linearity**: Long straight lines in the image are more likely to be straight lines in the 3-D model. For example, edges of buildings, sidewalk, windows, etc.

None of these properties individually can determine the 3-D structure of the scene. Thus, when combining them togeter, 'confidence factor' needs to be set up for them in the MRF. As can be imagined, edgels (edges between two neighboring superpixels) are a good way to express the confidence on the connectedness and the co-planarity of two adjacent superpixels.

2.2. Learning Model

It can be easily seen that edgels are very useful indicators of the occlusion boundaries and folds (places where two planes are connectedby no co-planar). When there is an edge, the two neighboring superpixels are more likely to be either belong to two diferent objects distant from each other, which corresponds to an occlusion, or belong to two parts of a single object, which is a possible fold. Hereafter, we will use y_{ij} to indicate the binary edge value between superpixel s_i and s_j. Then we have $y_{ij} \in \{0, 1\}$. Specifically, we set $y_{ij} = 0$ to be the edge, which doesn't conform with the conventional setting, simply because we want the larger y_{ij} values represent the higher confidence of the connectedness or co-planarity of two adjacent superpixels.

Let the features extracted from the image for each pair of neighboring superpixels be x_{ij} (the extraction of features are dicussed in Section 2.3). We can then model the response between y_{ij} and x_{ij} as a logistic function,

$$P(y_{ij}|x_{ij}; \phi) = \frac{1}{1 + \exp(-\phi^T x_{ij})}, \qquad (1)$$

To obtain the y_{ij}'s for each pair s_i and s_j of the superpixels, images were first manually labeled to connect those superpixel that are visually connected together. By this means, part of the superpixels detected automatically are merged together. Fig. 1 showed an example of the manually labeled edge image as compared with the automatically detected edge using Canny's method [6].

As can be seen, in our special problem, the edge shown as Fig. 1 (b) looks more reasonable as it conforms with the 3-D spatial structure of the scene, which is our goal for the edge learning.

2.3. Features

For each superpixel, a number of features are computed to capture the monocular cues that is useful to infer the edges. For each superpixel at location i, texture-based summary statistic features, and superpixel shape

Fig. 2. The convolution filters used for texture energies and gradients. Teh first 9 are 3×3 Law's masks. The last 6 are the oriented edge detectors at $30°$. The nine Law's masks do local , edge detection and spot detection. The 15 Law's mask were applied to the Y channel of the image. Only the first averaging filter to the color channels Cb and Cr were applied; thus 17 filter reponses were obtained. As both of energy and kurtosis were calculated, totally 34 features were obtained for each patch.

and location based features are computed. Particularly, the features are computed as the output of each of the 17 (9 Laws masks, 2 color channels in YCbCr space and 6 oriented edges, see Fig. 2) filters. As the structural edge is a characterization of the two adjacent superpixels, the 34 features for each of the two superpixels are combined together to have a totally 68 features.

3. EXPERIMENTAL RESULTS

Our experiment was done on a desktop with AMD Athlon $\times 2$ 2.0GHz CPU and 4GB Memory. In the experiment, 15 images from the database are first manually labeled by the tools provided by
http://make3d.stanford.edu/scribble/index/###,
where ### are the image number in the database. Considering the vertical difference in terms of the catogories of normally seen objects within an image, for example, normally the ground is at the bottom part of the image and the sky is at the top, we separate the image vertically into 11 rows as the features in different vertical rows are supposed to be different. Logistic regression is then applied to each row such that the parameters trained are for each row only.

In addition, to reduce the redundancy of the features, PCA with 0.99 of total variance preserved was applied to the feature vectors before the logistic regression. A comparison of the training error with PCA applied and with it not applied for each row is shown in Fig. 3.

As can be seen the error rates for both methods are roughly the same, the training time differed quite a lot as shown in Table 1. Thus, the method with PCA is prefered.

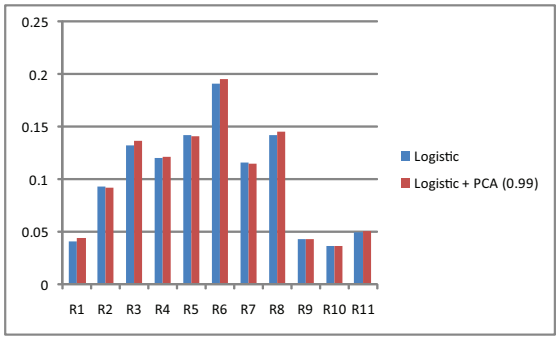


Fig. 3. Error rate of each row (R1-R11) for logistic regression with PCA (red) and without PCA (blue).

	Logistic	Logistic with PCA (0.99)
Average Training Time	4.61s	0.61s

Table 1. Total training time for logistic regression with and without PCA.

To test our proposed method on unseen images, a Leave-One-Out cross validation (LOOCV) is done on all the 15 labeled images. The averaged testing error compared with the training error for the logistic regression with PCA applied are reported in Fig. 4.

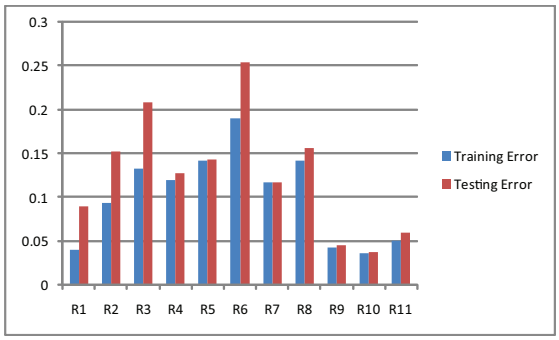


Fig. 4. Training error vs LOOCV testing error for logistic regression with PCA applied.

As expected, the testing error is a little larger than the training error, however comparable, which thus consolidates our previous assertion that the use of PCA prior to logistic regression is a reasonable choice.

Shown below in Fig. 6 are some of the learnt structural edges from the cross validation. As can be seen,

the learnt edges improved from the edges from superpixels by emphasizing on the edgels with more structural information.

4. DISCUSSION AND CONCLUSION

Logistic regression is by no means the only choice, experiments also done using Support Vector Machines (SVM) with Radial Basis Function (RBF) kernels. The comparison of the training error is shown in Fig. 5. As can be seen, SVM + RBF method improved the training error a little bit, the reason we didn't choose this method is because of the long training error. In our experiment, training of 15 images using SVM + RBF took around a whole day on the same computer.

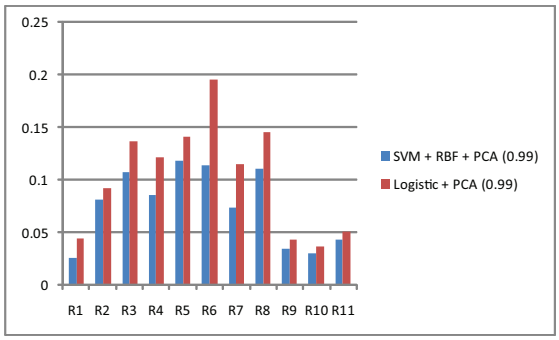


Fig. 5. Training error for SVM + RBF and logistic regression both with PCA applied.

Acknowledgement

The author would like to thank Ashutosh Saxena for helpful discussions. As edge learning is part of the project in [1], the overall structure of the 3-D reconstruction is from the paper.

5. REFERENCES

[1] Ashutosh Saxena, Min Sun, Andrew Y. Ng. Learning 3-D Scene Structure from a Single Still Image, In ICCV workshop on 3D Representation for Recognition (3dRR-07), 2007.

[2] Ashutosh Saxena, Sung H. Chung, Andrew Y. Ng. Learning Depth from Single Monocular Images, In Neural Information Processing Systems (NIPS) 18, 2005.

[3] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," Int'l Journal of Compter Vision, vol. 47, 2002.

[4] D. Forsyth and J. Ponce, Computer Vision: A Mordern Approach. Prentice Hall, 2003.

[5] F. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," Int'l Journal of Computer Vision, vol. 59, 2004.

[6] J.F.Canny, "A computational approach to edge detection," IEEE Trans Pattern Analysis and Machine Intelligence, 8(6): 679-698, Nov 1986.

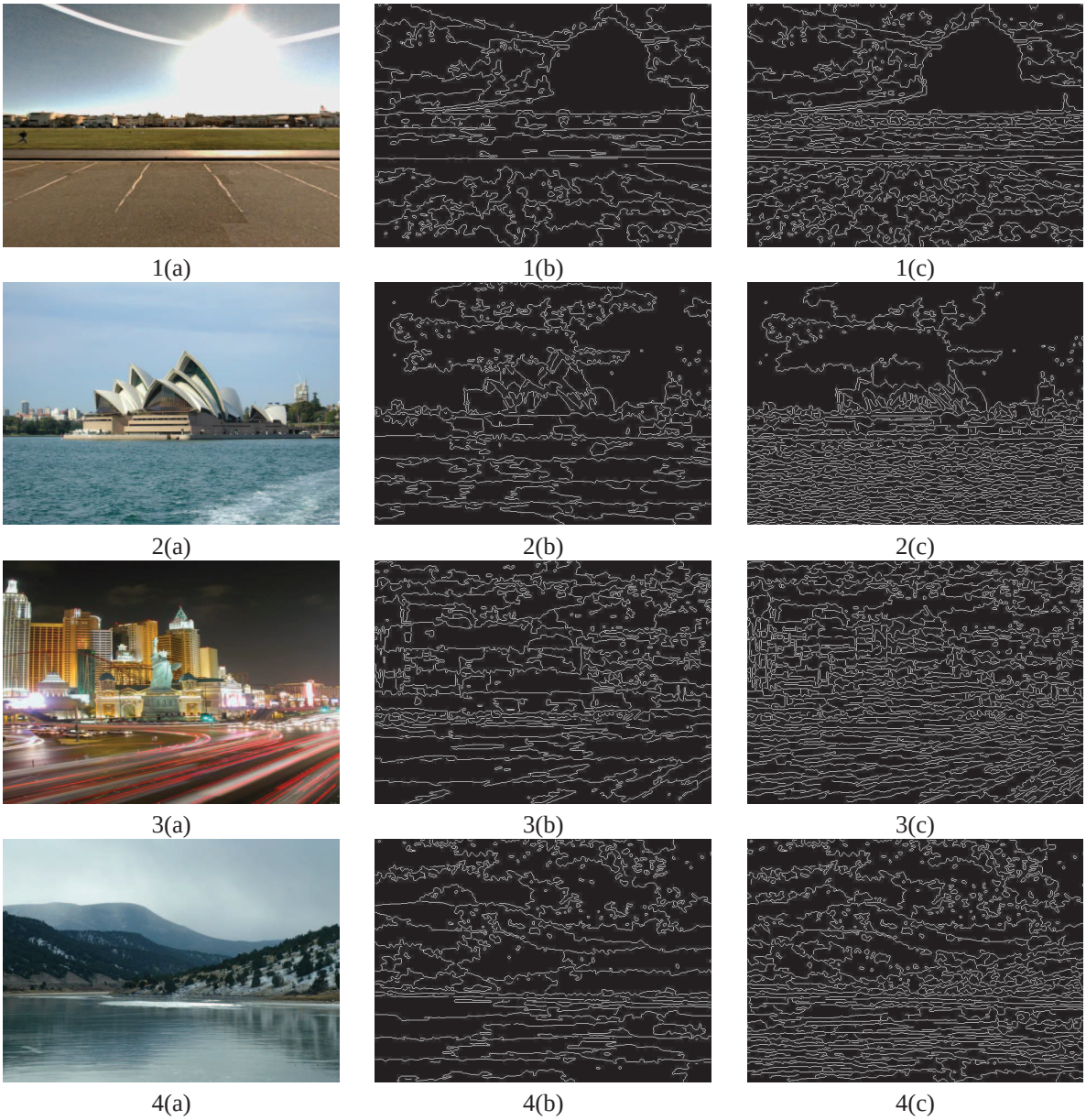


Fig. 6. Comparison of edges. (a) Original images; (b) Learnt structural edges; (c) Edges from superpixels.