Binary Separation on Heterogeneous Image

Fisher Yu fy@princeton.edu
Nanxi Kang nkang@princeton.edu
Siyu Liu siyuliu@princeton.edu

1 Introduction

1.1 Motivation

In computer vision, "Image Segmentation" is a process of partitioning an image into several pixel groups. It is typically used to locate objects and boundaries in an image. Particularly, "Binary Separation" - how to partition an image into two segments: "foreground" and "background" is of special interest.

Earlier techniques are either based on boundary features [Kass et al., 1988] or manual sketches [Boykov and Lea, 2006]. In this project, we propose a new approach to partition "foreground" and "background" making use of distance data of real objects in the image. To be specific, when an image is taken by camera, there exists a one-to-one mapping from pixels in the image to real objects. For each image to be processed, we have additional data telling the distances between the camera and the real objects corresponding to some pixels in the image.

It is challenging to use the distance data. Although the distance data is helpful, yet it is not complete. A typical distance data set only covers approximately 10% pixels.It is necessary to develop techniques that infers the states of other pixels based on this sparse information. Besides, the data could contain some errors. For example, some pixel may be mapped to a wrong real object resulting in a wrong distance number. Therefore, the algorithm should include procedures that help fix the inherent errors of data.

Considering the defects of distance data, we plan to only use it to generate the initial states of pixels, i.e, whether it belongs to "foreground" or not. Based on the inital states, we'll iteratively refine the segmentation following the traditional Expectation Maximization process. To do the refinement, we'll use Markov Random Fields to calculate the log-likelihood of a pixel, taking the information of the 8 pixels around it into account.

We believe our approach is more accurate than previous work. The additional distance data is a good indicator about which group a pixel belongs to, thus giving us confidence in the accuracy of segmentation, compared to earlier techniques which only use the image information alone. Moreover, our approach is flexible in that it does not need manual annotation. Finally, the segmentation decision is not likely to be misled by errors in distance data, since it will do iterative refinement based on the pixel's local neighbor information.

1.2 Data

We got our data from Google street view team and preprocessed it for use in our project. The data was collected by a car equiped with 8 cameras and 3 laser scanners. Each of the laser scanner can scan a 180 degree 2D plane at each time. Two of the laser scanners scanned vertically and the third one scanned horizontally. As the car moved, the car positions were recorded by global positioning system. Those positions were adjusted by the scans of the horizontal laser scanner using SLAM

(Simultaneous localization and mapping) techniques to get the best precision. The 3D position of each laser scan point can be estimated based on the relative positions of the car and laser scanners and therefore a 3D point cloud can be built from the scans of the vertical laser scanners. At the same time, the eight cameras were taking pictures as the car moved. Although the cameras were well calibrated, the images were taken by rolling shutters and there could exist errors in terms of image projection model if we assume each image were taken in a pinhole model.

Given the 3D point cloud and images, we project the points onto the images with the estimated point positions, camera poses and projection matrices. Then we can get images like Figure 1.

Figure 1: An image with 3D points projected into it.

In Figure 1, the black dots are the projections of the 3D points on this image. As we can see, due to all kinds of possible errors mentioned above, the 3D projections and the images are not well aligned and we can't segment the images directly based on the depth of the 3D scan points. In this project, we want to find the contour between the background (the sky area) and foreground (all the objects appears in the image, including roads and buildings). The models and computing issue will be discussed in the methods section.

2 Related Work

As a common problem in vision, segmentation has been well studied in the past. There is a large literature on segmentation and clustering among which two main lines of methods are proposed previously, feature space clustering (e.g. [Comaniciu and Meer, 1997], [Comaniciu and Meer, 1999]) and graph-based approach (e.g. [Shi and Malik, 1997], [Wu and Leahy, 1993]).

In feature sapce clustering, local features, such as low pass filter response, SIFT [Lowe, 2004], etc., are firstly extracted from the given image. For color images, color information is often included. Then those features are concatenated to form a vector as a descriptor for each pixel. In order to segment the image, we might seek a clustering of feature vectors observed in the image. A compact region of image with distinct feature would be expected to have a corresponding high density area in the feature space.

A natural approach is to model the observed feature vector using a Gaussian mixture model (GMM) [Carson et al., 2002]. For a pre-determined number of clusters K, the parameters can be fitted by Maximum Likelihood using EM algorithm. In addition, K can be chosen using penalized likelihood (a.k.a. minimum description length), Baysian information criterion (BIC) and other

model selection mechanisms. In addition, other variations are proposed by replacing GMM with K-means or low dimensional representation of the covariance matrix via matrix factorization. On the other hand, to avoid extra step of determining the number of clusters for the methods mentioned above, a non-parametric model based on kernel density estimation is proposed as an alternative. Mean-shift segmentation algorithm [Comaniciu and Meer, 2002] considers the probability density of feature vectors obtained from a given image. An iterative update on the mean-shift equation is used to find the segmentation labels.

Another branch of image segmentation algorithms is graph-based method. Instead of consider each pixel independently, we form a graph with each vertex representing an individual pixel and assign weights to edges according to some similarity measure. Then, the segments are determined by graph cuts.

One naive approach would be to calcualte connected components. We could use algorithms like Kruskal or Prime to simple remove edges between dissimilar pixels. However, this method is not robust to stray links. Thus, [Felzenszwalb and Huttenlocher, 2004] introduced an improved version of Kruskal algorithm which takes in account the local variantion within each component. An alternative proposal is to formulate segmentation as a max-flow/min-cut problem. One typical example of S-T Min Cut algorithms is the normalized cut introduced by [Shi and Malik, 2000]. This method takes into account the partition size of each segment so as to avoid tiny regions. In a more sophesticated setting, Markov Random Field (MRF), which we chosed for this project, is often used to combine probabilistic modeling and graph cut in order to provide better result. The goal of learning MRFs is to minimize the objective function (a.k.a. the energy function) which describes both fitness and connectivity among neighborhood. Several training algorithms have been proposed over the years, Iterated Conditional Modes (ICM) [Besag, 1986] uses "greedy" strategy to find local minimum; Graph Cuts, two most popular ones, swap-move and expansion-move were introduced in [Boykov et al., 2001]; Max-Product Loopy Belif Propagation (LBP) [Felzenszwalb and Huttenlocher, 2004] and [Freeman et al., 2000], a variant algorithm from the original belif propagation supporting loops in the graph; Tree-reweighted Message Passing desbribed in [Kolmogorov, 2006].

3 Methods

Based on the previous work, we merge several ideas in our project. First, to classify a pixel, we can first train a Gaussian mixture model to summarize the variation of pixels in the background and foreground separately. In another way to see this, we are getting an approximate distance from a certain pixel value to the cluster it belongs to and therefore we can get a cost to assign a certain pixel to the foreground or background. On the other hand, in the Gaussian mixture model, we are assigning an latent variable to each pixel. Due to the isolation between adjacent pixels in the Gaussian model, it is easy to imagine that the output of the binary classification will be very noisy in the sense that the adjacent pixels will have different labels although they have similar colors and belong to the same object visually. To solve this problem, we use Markov random field(MRF) model to constraint the relation between the labels of adjacent pixels. Also, we use conditional random field(CRF) model to adapt this relation to the difference of the pixel colors and therefore the model becomes more flexible. Also, we explore various ways to utilize our laser data. In the following subsections, we will discuss parts of our model and elaborate how we use the scan data.

3.1 Gaussian Mixture Model

3.2 Conditional Random Field

3.3 Inference

3.4 Use of Scan Data

4 Experiment

The first experiment is done using GrabCut. As mentioned in previous section, we only use the distance transform image derived from laser point projections as an initialization for MRF in this case. The distance transform image is thresholded to form an initial labeling strategy for foreground and background. As a result, the extra information provided is not used across iterations. The result is shown in Figure 2.

Figure 2: Segmentation result with GrabCut.

Note that since we assigned equal cost to edges across the image, cars and windows sharing the similar color as the sky would be segmented as the background.

Having observed this drawback, we decided to vary the cut cost according to our initial estimation of the segments. Note that, assuming foreground objects got scanned more often than background, the natural boundaries would occur at the transition between dense and sparse areas of projected LiDAR points. Thus, in the second experiment, LaserCut, we prepross the distance transform by running simple edge detection algorithm (i.e. Sobel operator) to extract possible boundaries between foreground and background. Therefore, for regions within the potential boundaries, some color differences would be tolerated by assigning smaller weights before the corresponding term in the energy funciton. In contrast, for pixels close to potential boundaries, we intend to rely more on color differences via larger weights to refine the boundary curve. The result is shown in Figure 3.

Comparing to GrabCut model, we successfully eliminated the white cars in the front from the background.

To further evaluate our model, we did another experiment on the same image with a widely used MRF model, Ising model. In this setting, color difference between neighboring pixels is completely ignored by assigning the same cost for connectivity to all edges in the MRF graph. The result is shown in Figure 4.

Figure 3: Segmentation result with LaserCut.

(a) Large threshold on initialization (b) Small threshold on initialization

Figure 4: Segmentation result with Ising Model.

Taking a close look at the boundary Ising model found, we observed that the foreground objects are outlined by a "margin" of the sky. This is probably due to we only used Gaussian mixture model to measure the clustering in color. Since there are white cars on the road, the GMM would intend to label part of the sky as the foreground due to larger threshold 4(a) on distance transform image. However, smaller threshold 4(b) values would result in labeling the white cars in the front to background initial, which would be generally hard to correct in later iteration under this model.

5 Evaluation

In general, evaluating segmentation result is hard due to the absence of precise quantitative measurement without ground truth. In addition, there is another important issue centers around the use of energy to compare energy minimization algorithms. The goal in computer vision is not to compute

the lowest energy but the most accurate one, besides computing the global minimum was shown to be NP-complete in general [Boykov et al., 2001]. In the paper by [Szeliski et al., 2008], a quantitative comparison between lowest energy achieved by different energy minimization algorithms and the energy calculated from ground truth provides experimental proof for this argument. Although one can argue that we could manually draw the boundaries for each image we used in this project, due to high resolution (1936 x 2592) and limited size of sample labling from different users we can account for given the time, the accuracy and unbiasness of the ground truth cannot be guaranteed.

As an alternative, we decided to conduct a comparison study among the three experiments we did mentioned in previous section. The best model as well as the final model we have is LaserCut which has the advantage of adaptiveness of initial guess through out all iterations. As we discussed before, comparing to GrabCut model, LaserCut successfully classified the cars with similar color as the background as part of the foreground. This observation suggests the effectiveness of adaptively using extra information acquired from LiDAR scan. On the other hand, comparing to Ising model, LaserCut could provide more complicated and accurate boundary between foreground and background, which indicates the effectiveness of separation on color differences when close to the boundary.

Granted, there are still areas near the boundary being mislabelled, for instance, some leaves of the tree in Figure 3. It is generally hard to determine the exact bounday of such complicated foreground object. Even when we have captured images with such high resolution, it is still far from certain that all the details of objects far back to the scene can be accurately preserved. Moreover, even though we can somehow manage to obtain all the details in the scene, the correctness of boundaries around complicated objects like trees would be rather subjective to different people.

Therefore, in spite of the fact that most of the model evaluation is based on visual check in this project, we hope our observations and analysis could make a plausible case for the advantage of LaserCut model over the others.

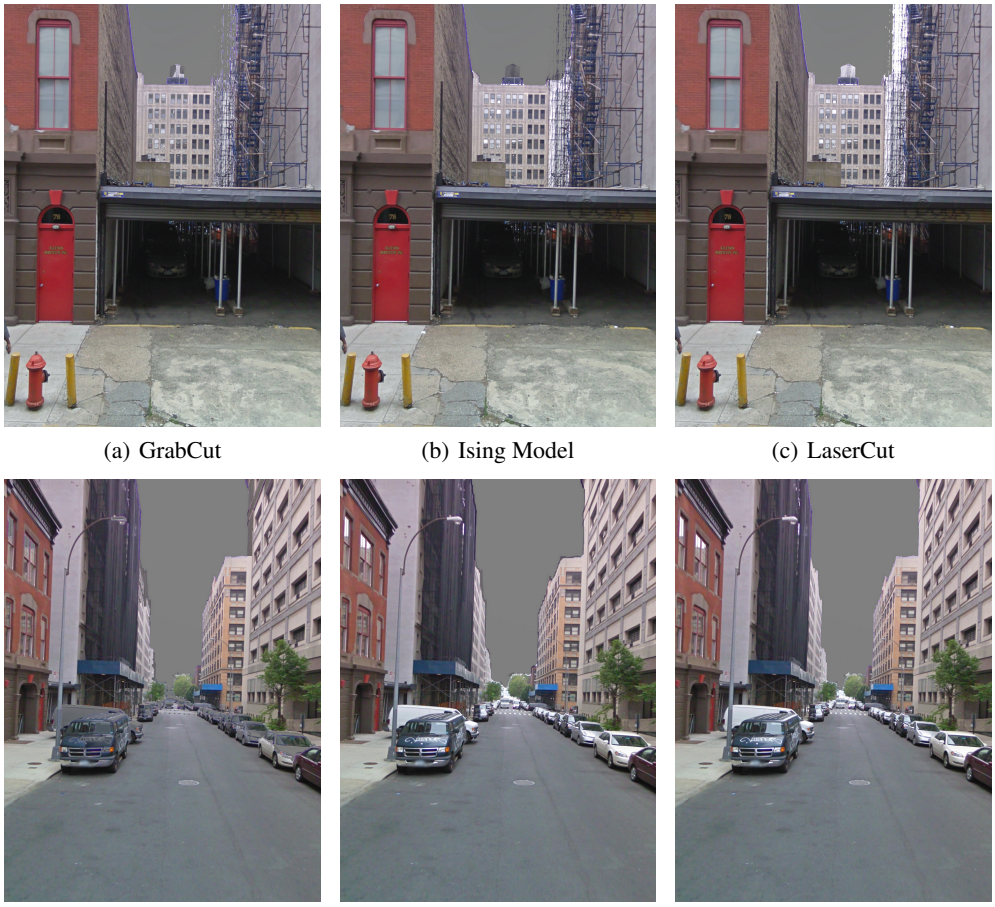
6 Conclusion and Future Work

Through careful study on the resulting segmentation of our approach, we learnt several things about the algorithm. For one thing, the algorithm produces a good approximation to the *true* boundary between foreground and background. In Figure 3, the curve along the boundary is highly curvy and very close to the truth between green leaves and sky. From the distance data, we can see there are not many "leaves" pixels with distance, therefore we conjecture that the color feature plays an important role in delimiting the boundary. For the other thing, the algorithm achieves balance between the influence of distance data, which is used to generate inital states, and the influence of color features, which is used to refine the segmentation. For example, in Figure 3, although the white cars have similar color to that of sky, they are not considered as background. Another example is that some sky pixels in the image are classified as background, depite the dataset suggests that they have finite distance to the camera. These two examples show that we are able to use color feature to fix minor errors in the distance data while still maintaining most of status generated by the distance data.

Our future work will focus on the following two aspects. First, the background pixels in our experiment images have very similar color. This fact greatly helps segmentation as we can put much weight on the color feature to infer the labels of pixels. In the future, we want to incoporate images with more colorful background. To do segmentation on such images, we may have to introduce new factors to model background and foreground. The other aspect is to do the general image segmentation. That is, segment the image into multi-layer. At present, we only use whether the pixel has a finite distance number or not. As we take the actual distance value into account, we can segement the images into several layers, with pixels in the same layers have close distance value.

7 More Results

We also did a comparison study on other images to further evaluate our algorithm.



(a) GrabCut (b) Ising Model (c) LaserCut

(d) GrabCut (e) Ising Model (f) LaserCut

Figure 5: Comparison study on other images.

As we argued before, our algorithm would usually produce a more complicated yet cleaner bounary between forground and background over ther other two.

References

J. Besag. On the statistical analysis of dirty pictures (with discussion). *Journal of Royal Statistical Society, Series B*, 48(3):259–302, 1986.

Y. Boykov and G. F. Lea. Graph Cuts and Efficient N-D Image Segmentation. *Int. J. Comput. Vision*, 70(2):109–131, Nov. 2006. ISSN 0920-5691.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, Nov. 2001. ISSN 0162-8828.

C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 750–755, 1997.

D. Comaniciu and P. Meer. Mean shift analysis and application. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1197–1203, 1999.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

P. F. Felzenszwalb and D. R. Huttenlocher. Efficient belief propagation for early vision. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–261–I–268 Vol.1. IEEE, June 2004. ISBN 0-7695-2158-4.

W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.

M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.

V. Kolmogorov. Covergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

D. G. Lowe. Distinctive image features from scale-invariant features. *International Journal of Computer Vision*, 2004.

J. Shi and J. Malik. Normalized cuts and image segmentation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug. 2000. ISSN 01628828.

R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):1068–1080, June 2008. ISSN 0162-8828.

Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1101–1113, 1993.