Segmentation of Salient Regions in Outdoor Scenes Using Imagery and 3-D Data

Gunhee Kim Daniel Huber Martial Hebert *
The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA
{gunhee,dhuber,hebert}@cs.cmu.edu

Abstract

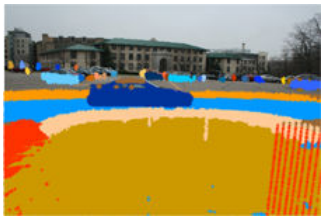
This paper describes a segmentation method for extracting salient regions in outdoor scenes using both 3-D laser scans and imagery information. Our approach is a bottom-up attentive process without any high-level priors, models, or learning. As a mid-level vision task, it is not only robust against noise and outliers but it also provides valuable information for other high-level tasks in the form of optimal segments and their ranked saliency. In this paper, we propose a new saliency definition for 3-D point clouds and we incorporate it with saliency features from color information.

1. INTRODUCTION

This paper discusses an algorithm for segmenting the salient regions of a scene using both 3-D laser scan and imagery information. Our problem statement is shown in Fig.1. Given an image and its corresponding 3-D scan data, our algorithm delineates dominant salient regions in a bottom-up manner *without any high-level priors, models, or learning*. The proposed system does not make any assumption on neither the number of salient regions nor the explicit appearance model of the salient regions. Only low-level information from an image and 3D scan data such as colors at pixels and the (x, y, z)-coordinates of a point cloud are used for saliency detection.

The motivation for this research stems from the development of a perception system for unmanned ground vehicles, which require recognizing stationary objects, such as parked vehicles, in the environment. In the absence of any other information, this type of recognition task is difficult because of the potential for high FAR and because of the high computational cost involved in scanning an entire image for objects. The recognition is greatly simplified if we can fixate automatically on a small number of candidate regions in the image. Our objective in this paper is to propose a simple and effective way of extracting these regions, thus providing a fast pre-processing step that can reduce dramatically the complexity of later recognition processes. Of course, one could argue that this new task is at least as challenging as the recognition task itself, in which case we would not gain anything at all by developing this segmentation step! We address this objection by using co-registered 3-D data in addition to the usual color images. This is a reasonable assumption in the context of unmanned ground vehicles which often have means to collect 3-D data. The key observation is that it is relatively easy to extract initial clusters from 3-D data, and that the image data can be brought in to refine the initial segmentation of the scene and to evaluate the relative importance (saliency) of each of the regions, thus completely bypassing the much more difficult

(a) Input: an image and its corresponding 3-D scan data



(b) Output: saliency clustering and segmentation

Figure 1. Problem definition; Given a pair of image and 3-D scan data, *k* most salient regions are segmented using only low-level features. (These figures are best viewed in color.)

*This work was prepared through participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0012.

problem of segmenting images directly. This approach to the problem is in fact quite intuitive. Consider the region formed by grouping the data seen on a car in the scene. In the image, the car may be many different colors, making any color segmentation scheme operating directly on the color image doomed to failure, while the corresponding set of 3-D points does form a natural cluster which is well-separated from the rest of the data.

The main contribution of this paper is an approach to bottom-up detection of salient regions using two sensor modalities - imagery and 3-D data. The detection proceeds in two steps: 1) Grouping of the 3-D points into 3-D clusters to form initial candidate regions; and 2) Refinement of the regions and computation of their relative saliency using the image data. The criterion used for grouping measures how well a cluster fits a family of probability density functions (pdfs). Based on this definition, we perform clustering in which a 3-D point cloud is partitioned so that each patch is optimal in the sense that the overall representation cost of the point set by given pdfs is minimized. A detailed discussion of this technique is provided in section 2.2. Then, the saliency of every cluster is computed based on low-level information from both imagery and 3-D data. Finally, the dominant regions (highest saliency) are segmented out from the image for high-level tasks such as object recognition.

The proposed method has two advantages as a mid-level vision task. First, the approach automatically removes non-important or erroneous information. The left image of Fig.1.(b) shows a typical example. Some errors of laser measurement, a bunch of parallel lines in the lower right of the image, are clearly rejected as a noise cluster. Second, since the final output of our algorithm includes the relative saliency of the regions, it provides ordering information to the other perception algorithms as to which region should be evaluated first. For example, object detection can be performed in the order of relative importance of the regions.

2. DETECTION OF SALIENT REGIONS

2.1. Related work

Saliency may be an elusive term, but it generally means the *relative importance* of a region in a scene. Recently, the computational modeling of visual attention has been actively studied in computer vision [3, 8, 9, 10, 12], which differ from each other in the details of the mathematical models used for the definition of the saliency. For example, saliency is computed by center-surround contrast [9], entropy [10], self-information [3], and graph representations using multiple features [8]. Although the computational models of saliency vary, the underlying definition is based on a certain degree of local information complexity and rarity.

It is known that there are two types of processes which contribute to visual attention: bottom-up and top-town processes [9]. The bottom-up factor is motivated by the stimulus in a rapid and involuntary manner. Thus, this framework is driven by low-level features in the scene such as intensity, contrast, color, orientation, and motion. On the other hand, the top-down factor is controlled by high-level concepts like memories and experiences. For example, people naturally pay more attention to some familiar objects like a person, a car, or an animal, rather than the cluttered background.

Several fixation systems based on 3-D data have been proposed such as [4] and [7]. However, they are direct extensions to new sensor modality and largely rely on the image-based attention models. The BILAS framework [7] is built on the basis of Itti et al's work [9] , and Cole et al's system [4] adopts Kadir and Brady's salient region detection algorithm [10].

2.2. Saliency of 3-D data

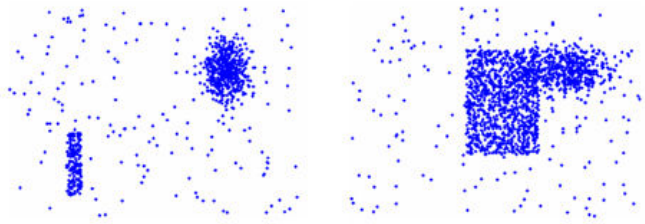


Figure 2. Two examples of saliency in a 2-D point set.

In order to explain the concept of saliency in 3-D data, we show two simple examples in Fig.2. Although each of the pictures consists of a large number of points, we can easily identify two salient regions in each picture. In the left figure, the upper right circular cluster and lower-left rectangular one are clearly the dominant clusters. In the right figure, a rectangle and a connected circular blob are dominant but, in this case, they are not obviously disconnected clusters. Instead, more complex laws of perceptual grouping [13] justify the emergence of those two clusters. According to the *proximity law*, the points which are close to each other are grouped together, so the two blobs naturally pop out in the left figure. However, to balance the *proximity law*, the *continuity law* and *simplicity law* segment the point set in the second example into two distinct clusters. Although only a single blob is in the image, it is more natural to think of it as two blobs which are connected to each other.

It is also interesting to note that on the right example, the two parts have *different continuity properties*. One is composed of four straight lines with right angles while the other one is delineated by a smooth shape. Moreover, according to *simplicity law*, splitting into two gives *easier description* of the shape (i.e., a rectangle and an elongated circle) rather

than a complicated single shape. Informally, this suggests that the two clusters correspond to two different natural distributions of points and that this information is important in splitting the single connected region into two parts.

Based on this observation, we formulate the detection of candidate regions in 3-D data as finding the set of clusters that best fit 3-D distributions from a pre-defined family of distributions. To be consistent with the literature on this topic, it is convenient to think of this family of distributions as a family of probability density functions (pdfs). Here, we consider a family that consists of two types of pdfs: Gaussians and uniform distributions. In the example of Fig.2 (right), the rectangular blob is an instance of the uniform distribution, while the ellipsoidal blob is an instance of the Gaussian distribution.

Since our objective is to develop a salient detection method as a mid-level vision task, the two pdfs can be considered reasonable choices. Although in this paper we focus on a small family of pdfs, this approach is flexible in that, if a more detailed prior is available, then we may use more specific pdfs to constrain the segmentation. For example, if we have a model of a car-like shape, the most salient regions would be the sets of points which best fit the car pdfs. We may model a car in the form of Gaussian mixture models, and directly find the salient regions (e.g., most probable region where a car exists in a scene in this context).

The fundamental tool that we need for this approach is a way to evaluate how well a set of 3-D regions fit a family of pdfs. For this purpose, we propose to use a recently introduced algorithm termed *Robust Information-theoretic Clustering* (RIC) [1]. Based on the minimum description length (MDL) principle, the RIC algorithm finds the most optimal clustering of the data with a given family of pdfs. We will discuss the details of this algorithm in the next section.

2.3. Proposed approach

In this section, we explain the proposed saliency detection in detail. First, we apply saliency clustering to the 3-D data. The clusters are projected on the corresponding image, and the saliency values of every region are computed by using both imagery and 3-D information. Finally, we perform image segmentation of the several top-ranked saliency regions.

2.3.1 Input

An image and its corresponding 3-D point cloud are given as input as shown in Fig.3.(a)-(b). We assume that a camera and 3-D scanner are calibrated so that points can be projected on an image. However, our algorithm does not require accurate calibration. Since a given data is iteratively clustered into the stable parts and noisy parts, the irregular measurements near the depth boundaries are easily rejected.

2.3.2 Preliminary clustering

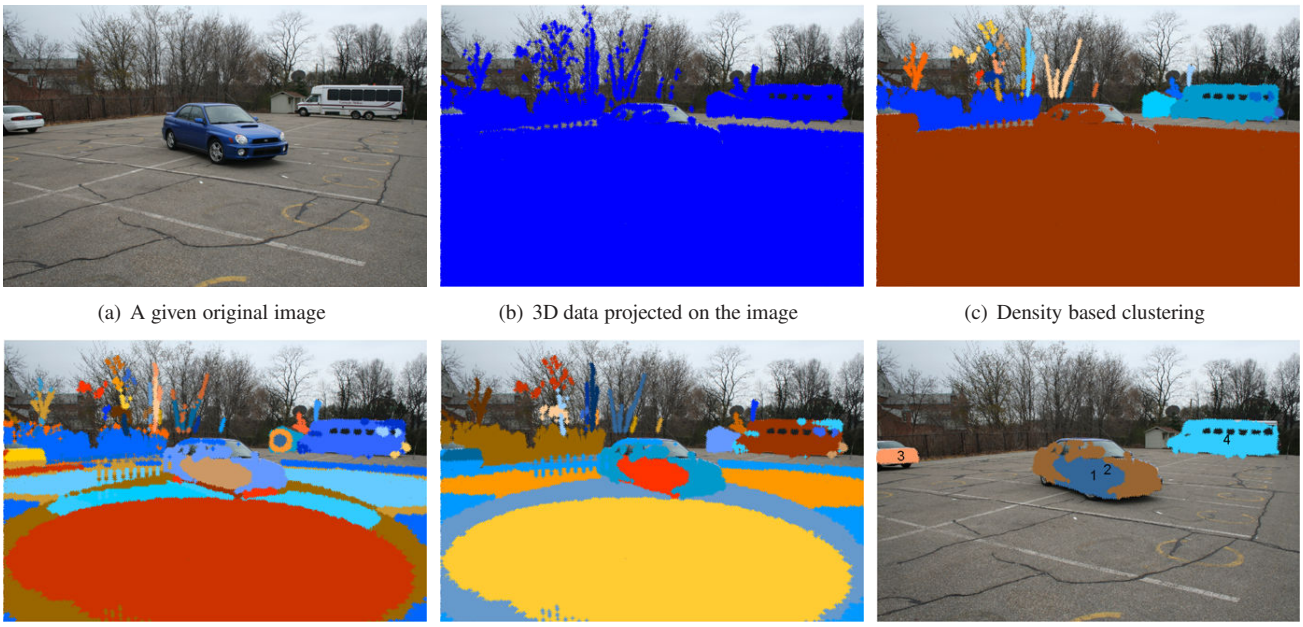
The RIC requires an initial clustering as input. Based on this, the RIC discovers the information-theoretically optimal clustering solution. The RIC algorithm is independent of the initial clustering algorithm. In particular, it can be combined with any clustering schemes such as K-means, K-medoids, or spectral clustering. In this paper, we use a well-known density based clustering technique DBSCAN [5]. It is based on local connectivity and density function such as densely connected points. In practice, DBSCAN is able to discover clusters of arbitrary shape while ignoring the small variations due to noise. However, the limitation of this clustering is that it may lose important details. For example, in Fig.3.(c), the leftmost white car and centered blue car are connected to the ground as one cluster.

2.3.3 Running the RIC

The RIC algorithm consists of the following two parts: 1) robust fitting to purify the clusters from noise and 2) cluster merging to find the best clustering result by combining unnecessarily over-split clusters. The robust fitting process splits the data into the most stable *core* part and irregular *noise* part. In other words, we first find the region which best matches to given pdfs, and the other region is classified into the *noise* part. Therefore, the selected core regions are highly likely to be consistently uniform or Gaussian shaped. Also, in the merging process, two blobs are merged only when the merged region gives us a much better description to given pdfs.

As a measure of goodness of fit of a cluster C to a family of pdfs, the RIC defines the *volume after compression* (VAC). The mathematical definition of the VAC is described in [1]. Consequently, the optimal clustering is obtained by minimizing the overall VAC of the data. By doing so, this technique can not only detect information-theoretic optimal clusters without knowing the true number of clusters but also reject noise or outliers.

In this paper, we apply these *splitting* and *merging* processes iteratively. Suppose that we are given initial clusters $C = \{C_1, ...C_k\}$ as a result of the initial clustering. For each cluster $C_i \in C$, we search for the best split of C_i into the core cluster $C_{i,core}$ and the outlier cluster $C_{i,out}$ according to the minimal VAC value. The resultant core cluster $C_{i,core}$ is split once again to avoid missing the significant partition, and the outlier cluster $C_{i,out}$ is put into the clustering queue again and iterate the splitting process as a new cluster. In the splitting stage, the over-segmentation of the clusters is not a problem. If it is really over-split in terms of compression costs, the two clusters will be combined during the merging process.



(a) A given original image (b) 3D data projected on the image (c) Density based clustering

(d) RIC after iterative partitioning (e) RIC after merging (f) Top four most salient clusters

Figure 3. Detection of salient regions. (These figures are best viewed in color.)

The merging process is conceptually simple. Given the cluster set $C = \{C_1, ..., C_n\}$ from the previous splitting process, it just merges greedily any pair of clusters C_i and C_j if the resulting $savedCost(C_i, C_j)$ is larger than zero; $SavedCost(C_i, C_j) = VAC(C_i) + VAC(C_j) - VAC(C_i \cup C_j)$. That is, the clusters are combined if the total cost is reduced by combining them.

Fig.3.(d)-(e) show an example of the iterative RIC. We can obtain several meaningful patches from the single huge cluster in Fig.3.(c) such as the two cars and several ground regions. The ground is partitioned into several patches because of noise and because, in practice, it is not a uniformly distributed surface since the data becomes sparser farther from the laser scanner. Also, the noise filtering effect can be seen in the cluster of a white bus. Most of the points in the blob remain in the main cluster, and some in the boundaries of the bus and windows are rejected as a noise cluster. By comparing Fig.3.(d) and Fig.3.(e), we can also see the merging of unnecessarily split clusters. In Fig.3.(d), the small shed on the left side of the bus is decomposed into two circles and a boundary region, but it is fused into a single region after the merging process as shown in Fig.3.(e).

2.4. Saliency Features

Once a collection of clusters is computed from the 3-D data as described above, we need to estimate the saliency of each cluster by incorporating the appearance information from the color image. We use four different types of saliency features as shown in Eq.(1). They are computed for every cluster $\forall C_i \in C$, and the sum of the four values is assigned to each cluster as the resultant saliency measure. In Eq.(1), the first term $f_i(C_i)$ is obtained from the 3-D data and the others are calculated from the RGB color information. The saliency features are designed to capture local, regional, and global aspects of saliency, respectively, by following Liu et al.'s approach [12]. As shown in Fig.4, $f_i(C_i)$ and $f_e(C_i)$ represent the 3-D and color properties of the cluster itself, respectively. $f_h(C_i)$ indicates regional feature (i.e., the cluster itself and its neighbors), and $f_s(C_i)$ captures the global relation with the cluster and the overall image.

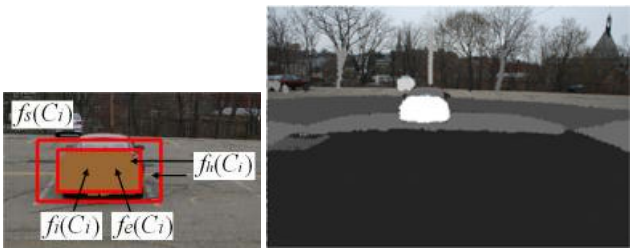


Figure 4. Saliency features. (left) The definition of saliency features. (right) The resultant saliency maps for every cluster. The brighter the region is, the higher the saliency value is.

Fig.4(right) illustrates the computation of saliency features. It shows the sum of the four types of features for every cluster. The brighter the region is, the higher its saliency is. Since the regions are generated from the 3-D data, we do

not suffer from ambiguity of scaling, which is one of key issues in saliency detection. Fig.3.(f) shows the top four most salient clusters. They all are from the cars, which is consistent with human perception.

Each term in Eq.(1) is normalized in the range [0,1]. That is, each feature value is computed for all clusters, then they are normalized by $f_*(C_i) = (f_*(C_i) - min_c f_*(C_i))/(max_c f_*(C_i) - min_c f_*(C_i))$. The mathematical definition of each of $f*(C_i)$ is described in Eq.(2)-(6). All of them can be computed very efficiently from the image and the 3-D data.

$$F_s(C_i) = f_i(C_i) + f_e(C_i) + f_h(C_i) + f_s(C_i) \quad (1)$$

2.4.1 Compression costs

The first term $f_i(C_i)$ is based on $VAC(C_i)$, which is computed during the saliency clustering discussed in the last section. $f_i(C_i)$ is the value of $VAC(C_i)$ normalized by the number of points in the cluster (n_{C_i}). That means the average compression cost of the clusters, which indicates how well the cluster is fit to the reference family of pdfs.

$$f_i(C_i) = -VAC(C_i)/n_{C_i} \quad (2)$$

2.4.2 Entropy

The second term $f_e(C_i)$ is based on the Shannon entropy of color histogram of C_i. It represents local signal complexity, or unpredictability, by following the definition of saliency as suggested by Kadir and Brady [10]. We calculate the entropy of the RGB histogram of all points in a cluster as shown in Eq.(3), where $p(\theta)$ represents the histogram probability of each bin. Here, 64 bins per color channel are used.

$$f_e(C_i) = -\sum_{\theta=1}^{K} p(\theta) \log_2 p(\theta) \quad (3)$$

2.4.3 Center-surround contrast

The third term $f_h(C_i)$ is based on the center-surround contrasts proposed in the pioneering work of Itti et al [9]. The underlying argument is that larger variations are observed between a salient region and its surrounding than within the region. We use a simple variation of this concept, as proposed by Liu et al [12]. First, imagine a rectangle enclosing the cluster. Outside of this rectangle, we define a surrounding rectangle R_s with the same area of cluster C_i. (See Fig.4.) Then, we construct the RGB histogram of Ci and R_s and compute χ^2-distance between them.

$$f_h(C_i) = \chi^2(h_{C_i}, h_{R_s}) = \frac{1}{2}\sum_{m=1}^{K} \frac{[h_{C_i}(m) - h_{R_s}(m)]^2}{h_{C_i}(m) + h_{R_s}(m)} \quad (4)$$

2.4.4 Color spatial distribution

The final term $f_s(C_i)$ captures the global spatial distribution of colors in the image. Intuitively, it means that if a color occurs often in the image, the object which contains the color is less likely to be a salient region. This measure is also adopted from Liu et al.'s work [12]. A Gaussian Mixture Models (GMMs) is used to describe the distribution of colors. We first compute a saliency feature $f_s(i, I)$ for each pixel of the image as shown in Eq.(5), where $p(a|I_x)$ represents the probability of *a*-th color component assigned to each pixel. (Assuming we have A number of mixtures in the GMM model $\{w_a, \mu_a, \Sigma_a\}_{a=1}^{A}$). $V(a)$ represents the spatial variance of *a*-th color component and $D(a)$ is a weight factor that assigns less important to colors nearby image boundaries. The saliency feature of the cluster $f_s(C_i)$ is obtained by averaging $f_s(i, I)$ of all pixels in the cluster as shown in Eq.(6). Both are normalized to be in [0,1], and the detailed computation can be found in [12].

$$f_s(i, I) = \sum_a p(a|I_x)(1 - V(a))(1 - D(a)) \quad (5)$$

$$f_s(C_i) = \frac{1}{n_{C_i}} \sum_{i \in C_i} f_s(i, I) \quad (6)$$

3. Segmentation

3.1. From Sparse Points to Dense Regions

The saliency detection generates *k*-numbered clusters of 3-D data. One remaining problem is to find out what regions of an image correspond to the 3-D clusters, which are projected on pixels in the image. Because the image resolution is much higher than the point cloud density, it is clear that only a sparse subset of the pixels is actually labeled as belonging to one of the clusters. (See Fig.5.) We avoided this problem in the computation of the image-based saliency values by computing the histograms over the sparse set of pixels projected from the 3-D pixels. However, we need dense regions as a final result. We formalize the problem as a labeling problem in which a sparse set of pixels is labeled, the label of each pixel indicating to which cluster the pixel belongs, and we want to propagate the labels to all of the other pixels in the image. The Markov random field (MRF) model is a natural model for this type of problem, because the labeling of one pixel is conditioned by the labels of its neighbors. We describe the details of our MRF model in the remainder of this section.

3.2. Approach

We add a single background cluster C_{k+1} to the collection of a *k* salient classes. The background class consists of the points randomly picked in the part of the image which is fixed distance away (e.g. 20 pixels) from the labeled points.

Figure 5. Sparse 3-D points vs. dense image regions. The number of 3-D data points projected on the image is only 193, while MRF labeling delineates the corresponding region with 2007 pixels.

Similarly to conventional Markov random field (MRF) models for computer vision problems, an image is considered as a four-connected grid graph. Let P be the set of pixels in an image and $C = \{C_1, ...C_{k+1}\}$ be a set of classes.

A label is assigned to each pixel to minimize the energy function in Eq.(7) [6]:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V(f_p - f_q) \qquad (7)$$

where *N* are the edges in the four-connected graph. $D_p(f_p)$ is the cost of assigning label f_p to pixel *p*, and is referred to as the data cost. $V(f_p - f_q)$ measures the cost of assigning label f_p and f_q to two neighboring pixels, and is normally referred to as the discontinuity cost.

In order to compute the label assigning term, $D_p(f_p)$, the color similarity between pixels is used. This approach is similar to the computation of the likelihood energy in *Lazy snapping* [11]. Since the color information of most labeled data in the same class is redundant (*e.g.*, most nodes in the class of a car body in Fig.3(a)(f) are blue), the mean color of each class $K_m^{C_i}$ is computed by k-means. The number of clusters of k-means (m) is set to 32 for the salient classes and 96 for the background class. As shown in Eq.(8), the computation of $D_p(f_p)$ is split in two different cases: 1) for the pixels already labeled by a class C_i, and 2) the unlabeled pixels $U = P \backslash \bigcup_{i=1}^{k+1} C_i$. $D_p(f_p)$ is a k+1 dimensional vector, each of which represent the assignment cost for each class C_i. For the labeled pixels $p \in C_i$, the i-th element of $D_p(f_p)$ becomes 0, and the others are assigned to extremely large value such as infinity. It prevents the labeled point from being assigned to another label. For the unlabeled pixels, we compute the minimum distance from its RGB color $I(p)$ to mean colors of every class $K_m^{C_i}$.

$$\begin{aligned} D_p^j(f_p) &= \begin{cases} 0, & \text{if } i = j \\ \infty, & \text{if } i \neq j \end{cases}, \qquad \forall p \in C_i \\ D_p^j(f_p) &= \lambda \min_m |I(p) - K_m^{C_j}|, \quad \forall p \in U \end{aligned} \qquad (8)$$

For the measure of difference between labels, $V(f_p - f_q)$ in Eq.(7), the Potts model is used in Eq.(9). It captures the fact that the cost is zero for identical labels whereas the cost is a positive constant *d* when the labels are different.

$$V(f_p, f_q) = \min(d|f_p - f_q|, d) \qquad (9)$$

Having defined the energy function used to model the interactions between pixels, the final step is infer the optimal distribution of labels by minimizing this energy. Two widely-used inference techniques for inference in MRF models are graph cuts [2] and belief propagation [16]. In this work, we use an approximate belief propagation (BP) algorithm [6], which can speed up by several orders of magnitude with a comparable performance to time-consuming standard BP or graph cuts algorithms.

4. Experiments

4.1. Data acquisition

The test data was collected using a Sick laser range scanner and a Canon Digital Rebel XT camera. For each image, we accumulate 180 degree of horizontal scans of the rangefinder to obtain a 3-D point cloud. The camera calibration is performed by using the Caltech calibration toolbox [14], and extrinsic calibration of a laser range finder to a camera is carried out by CMU rangefinder calibration toolbox [15].

4.2. Results

Fig.6 shows eight more examples of segmentation. Note that all results in this paper are obtained by using a single set of parameters.

The second rows of the Fig.6 represent the results of information-theoretic optimal partitioning of 3-D data. The largest ground cluster looks like a big circle, because it happens to fit well a Gaussian pdf. Trees have complex shapes, so they are split into many irregular regions as shown from fifth to eighth examples. There are some perpendicular noisy lines in all images, which are caused by laser measurement errors. However, they are clearly rejected by the proposed method. It shows the robustness against noise of the proposed method.

The top four most salient regions are shown in the third rows. Since most of an outdoor scene are ground and sky, those regions should not be conspicuous. On the other hand, colored cars are generally salient, and they are successfully discovered. In the third example, a pole is detected as a fourth salient object since the pole has a distinctive color (i.e. brown) and matches the uniform distribution well. In the second, seventh, and eighth examples, bush plants, trees, and grounds are detected as salient regions. However, they are detected only after all the other dominant regions are discovered. As shown in the pictures, they are still ranked lower than the other actual salient objects.

One of the main limitations of the approach is that sometimes the algorithm suffers from over-segmentation or under-segmentation issues. The first salient blob in the fifth example is a typical example of under-segmentation. A red and a white cars are parked side by side, and thus they are detected as one blob. It is not easy to discriminate them

with only 3-D data since they are quite close to each other and far from the viewer. In the fourth example, we note that a car in the left of the scene is split into two, which is an example of over-segmentation. It is because in the near view a car is made of several non-parallel planes, corresponding to a decomposition of the car into a set of several uniform distributions.

The final segmentation results are quantitatively evaluated by calculating the degree of overlap of the segmentations and ground truths. In Eq.(10), the area of overlap between the predicted segmentation S_p and the hand-labeled ground truth mask S_{gt} is divided by their union. For all 41 images, the degree of overlap d_o is [0.6819$\pm$0.1093] (mean $\pm$ standard deviation).

$$d_o = area(S_p \cap S_{gt})/area(S_p \cup S_{gt}) \qquad (10)$$

This result is related to another issue, which is the difficulty in identifying a complete car with a single segment using our method. As shown in all car examples in Fig.6, most segmentation results include only car bodies but not windows. Sometimes the laser beams pass through the windows, sometimes they do not. Consequently, the irregular measurement of windows is highly likely to be detected as noise by the algorithm.

5. CONCLUSIONS

This paper has discussed an approach to the bottom-up segmentation of salient regions on the scene using imagery and 3-D LADAR data. As a mid-level vision task, our approach provides a segmentation and the ranking of each region in terms of saliency. By providing an ordering of a small set of regions on which to focus more complex object recognition processes, this algorithm is an important building block of a complete perception system.

Much remains to be done for the algorithm to be truly effective in real scenarios. First, we need to integrate it with the recognition system. The effectiveness of the algorithm will be demonstrated by showing that this bottom-up saliency detection helps object recognition (ex. car detection) in practice. Second, we can extend our approach to model-specific segmentation by incorporating more detailed prior information, for example, by refining the family of reference pdfs. Currently, our algorithm has difficulty in obtaining a complete car segmentation (for example, due to missing the window part of a car). If shape models are available, we can infer more exact boundaries of the objects.

6. ACKNOWLEDGMENTS

The authors would like to thank Claudia Plant and Jia-Yu Fan for kindly providing us with their codes of RIC with valuable comments.

References

[1] C. Böhm, C. Faloutsos, J.-Y. Pan, and C. Plant. Robust information-theoretic clustering, 2006. KDD.

[2] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images, 2001. ICCV.

[3] N. D. Bruce and J. K. Tsotsos. Saliency based on information maximization, 2005. NIPS.

[4] D. M. Cole, A. R. Harrison, and P. M. Newman. Using naturally salient regions for slam with 3d laser data, 2005. ICRA workshop on SLAM.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases, 1996. KDD.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

[7] S. Frintrop, E. Rome, A. Nüchter, and H. Surmann. A bimodal laser-based attention system. *CVIU*, 100(1-2):124–151, 2005.

[8] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency, 2006. NIPS.

[9] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259, 1998.

[10] T. Kadir and M. Brady. Saliency, scale and image description. *IJCV*, 45(2):83–105, 2001.

[11] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping, 2004. SIGGRAPH.

[12] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object, 2007. CVPR.

[13] E. Rome. Simulating perceptual clustering by gestalt principles, 2002. ÖAGM/AAPR.

[14] K. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. Camera calibration toolbox for matlab, 2005. `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[15] R. Unnikrishnan and M. Hebert. Fast extrinsic calibration of a laser rangefinder to a camera. Tech. report CMU-RI-TR-05-09.

[16] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Info. Theory*, 51(7):83–105, 2005.

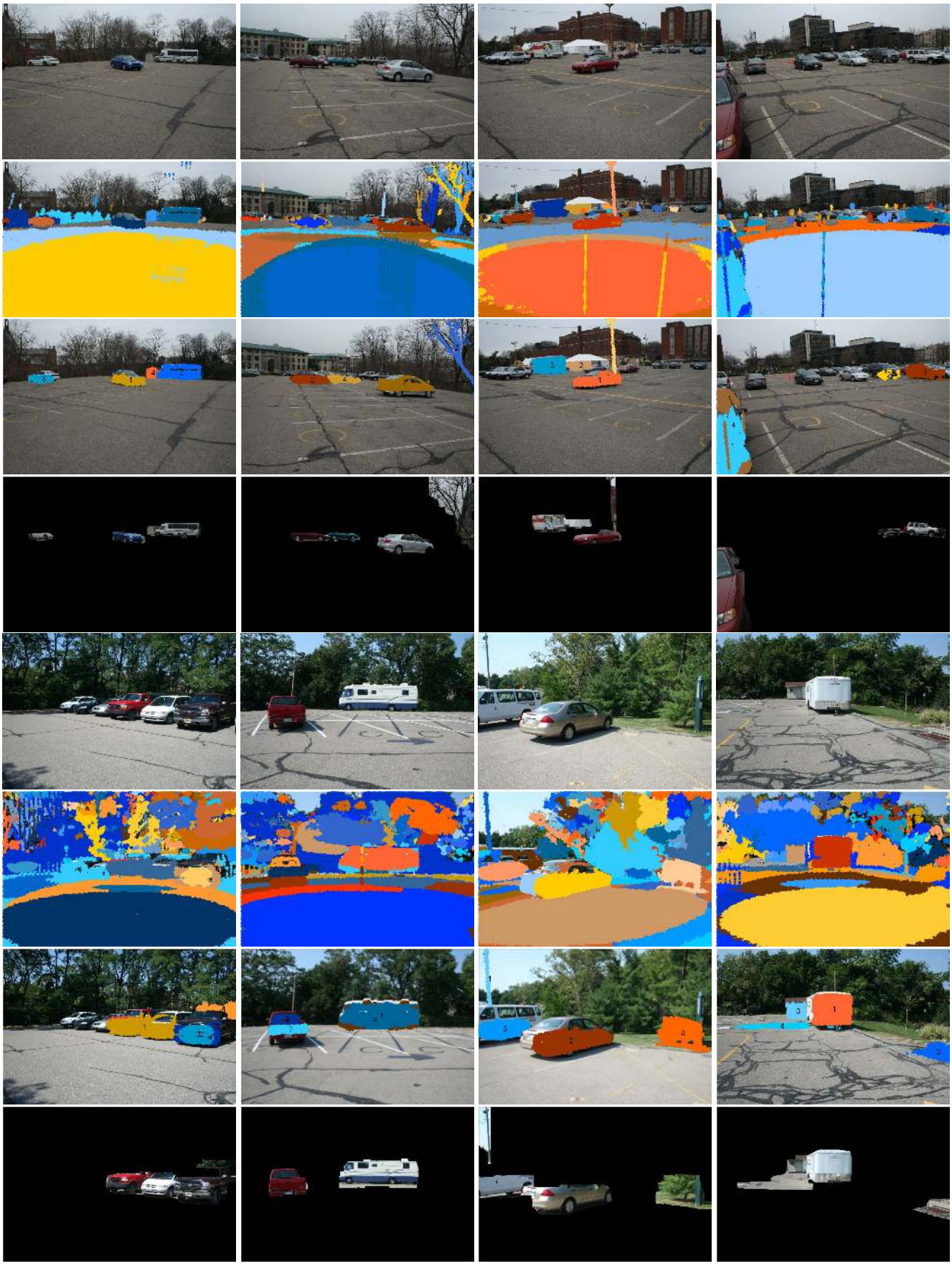


Figure 6. Eight more experiments. Each column shows a different example, and each row represents intermediate results. From top to bottom: input images, clustering, detection of the top four most salient regions, and segmentation. (These figures are best viewed in color.)