

# 3-D Segmentation and Motion Estimation of Range Data for Crash Prevention

Hamid Gharavi, *Fellow, IEEE*, and Shaoshuai Gao

**Abstract**—In this paper we present a 3-D object segmentation and motion estimation scheme of range video data for crash prevention applications. The segmentation is based on slope values of every point in the scene with which atomic regions are constructed by region growing. Atomic regions are merged with the flat surfaces (non-ground points) according to their mean height and the range difference of neighboring points. Following segmentation we apply a 3-D recursive motion estimation algorithm to each moving object. Simulation results show that the segmentation scheme together with the recursive motion estimation algorithm can be highly effective in estimating 3-D motion parameters of multiple moving objects in range video data.

## I. INTRODUCTION

Advanced vehicle-based safety and warning systems use 3D sensors (radar, laser scanners, stereo) and cameras to measure road geometry (position, curvature), and range to obstacles in order to warn a driver of an impending crash and/or to activate safety devices (air bags, brakes, steering). For such sensitive operations high-speed range cameras that are capable of acquiring raster depth measurements of an object with relatively high frame rates, would be essential. These images can provide precise measurements of the geometry of the 3-D environment, including all three Cartesian coordinates of the points on an object. This can make motion estimation and object tracking much easier and more reliable compared with using only video intensity images. In our earlier investigation we have shown that displacements of objects with complex 3-D motion in range moving images can be accurately estimated for a single object [13]. However, a typical road scene may contain multiple moving objects, which need to be segmented before 3-D motion estimation can be applied.

There are many schemes for range image segmentation [1]-[10]. These segmentation schemes can be broadly classified into two basic approaches: region-based [1], [4], [5], [16], and edge-based [3], [6]-[8]. The region-based segmentation algorithms are based on detecting continuous surfaces that have similar geometrical properties. The main drawback of these techniques is that they are not very accurate and can produce distorted surface boundaries. Edge-based schemes rely on detecting discontinuities of surfaces (i.e. edges) and thus, can localize surface boundaries more

precisely compared with region-based algorithms. On the other hand, detected edges may not be complete due to the noise, sampling, and the limited capability of the edge detection algorithm. To connect an incomplete detected edge point to a solid edge, some algorithms such as Hough transform [15], can be applied. In both cases, the image is normally segmented into surfaces including planar surfaces and/or curved surfaces, which can be very useful for object recognition and classification. In particular, for applications such as crash prevention, it would be crucial to accurately track vehicles and other obstacles on the road.

In this paper, we examine a region growing segmentation scheme, which has been applied prior to 3-D motion estimation. The segmentation scheme is described in Section II. Section III presents the simulation results using a sequence of synthetically generated range video images.

## II. OBJECT SEGMENTATION

In a realistic traffic scene there are roads, cars, buses, pedestrians, and so forth. In order to estimate motion parameters of moving objects, it would be necessary to segment the scene into multiple objects. As an example, Fig. 1 shows one frame of a range image sequence that consists of two cars on the road. Both cars in the range image overlap, but each moves at a different speed.

### A. Slope Calculation and Thresholding

In the segmentation process, and before separating each vehicle, it would be more convenient to remove the road (or the ground) first. This can be achieved by computing the radial slope [11] of every pixel (indicated by their Cartesian coordinates  $(X, Y, Z)$ ) using two vertical neighboring pixel's Cartesian coordinates. That is,

$$\text{slope}(x, y) = \arctg \left| \frac{Y(x, y+1) - Y(x, y)}{Z(x, y+1) - Z(x, y)} \right|, \quad (1)$$

where  $x, y$  are the sensor grid (range image) index.

In [11], the slope values are computed by a more complex process. That is, based on sensor geometry, the slope values are computed by the depth and the enclosed angle values of each pair of vertically adjacent beams. Compared with (1) however, this would be computationally more extensive. Bear in mind that the image data has to be transformed to the Cartesian coordinates for the motion estimation process, and such a transformation can be done prior to the segmentation process.

Since the real range images will have some noise, some of

Manuscript received January 15, 2007.

The authors are with the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (phone: 301-975-3634; fax: 301-975-6238; e-mail: gharavi@nist.gov).

the slope values will not be correct for some noisy points. In order to alleviate this effect, we use the average coordinate values of the points in the neighboring area (includes 8 neighboring points and the current one). That is,

$$Y(x, y) = \sum_{j=-1}^1 \sum_{i=-1}^1 Y(x+i, y+j), \quad (2)$$

$$Z(x, y) = \sum_{j=-1}^1 \sum_{i=-1}^1 Z(x+i, y+j). \quad (3)$$

Subsequently, the slope values of the points, including neighboring points, are averaged as the final slope values. That is,

$$slope(x, y) = \sum_{j=-1}^1 \sum_{i=-1}^1 slope(x+i, y+j) \quad (4)$$

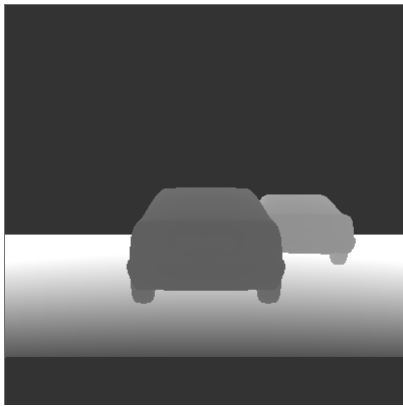


Fig. 1. Two cars on the ground.

Next, the slope values are compared with a threshold  $\Lambda$ . The pixels whose slope values are higher than  $\Lambda$  are treated as the pixel with steep slope, as illustrated in Fig. 3(a). This step can remove the road pixels, whose slope values are close to 0 and lower than  $\Lambda$ . However, most obstacles, such as vehicles may contain flat surface areas (surrounded by steep surfaces), which could also be removed in this process, thus dividing the obstacle into several parts.

### B. Region Growing

In order to regroup the separated parts into whole objects, a region growing method can be effectively utilized. Region growing is a process that put samples with the same property into one group. The region growing process can be easily deduced by looking at Fig. 2 and its operation is described in the following steps.

- 1) Find a seed point in the image grid. A simple method is using raster scan, i.e., from left to right and from top to bottom. The first scanned point will be selected as the seed.
- 2) For the seed point, its four (or eight) neighbor points (pixels) will be tested. If the neighboring pixels have the same property as the seed point (e.g., steep-slope), this point will be flagged and stored in a stack. Otherwise, this pixel will not be considered.

- 3) After all the neighbors are tested, one point in the stack will be pulled out as the new seed point.
- 4) Steps 1) - 3) should be repeated until the stack becomes empty.
- 5) Repeat steps 1) - 4) until all the points in the image have been checked.

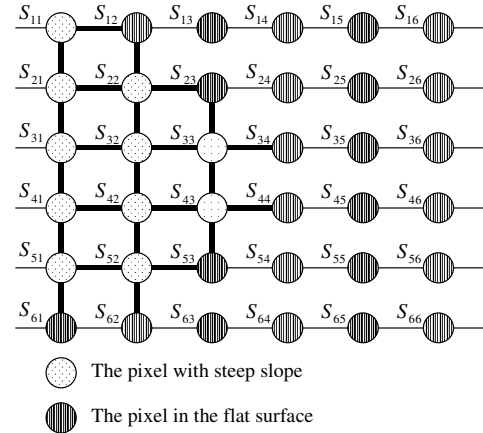


Fig. 2. Region Growing Process.

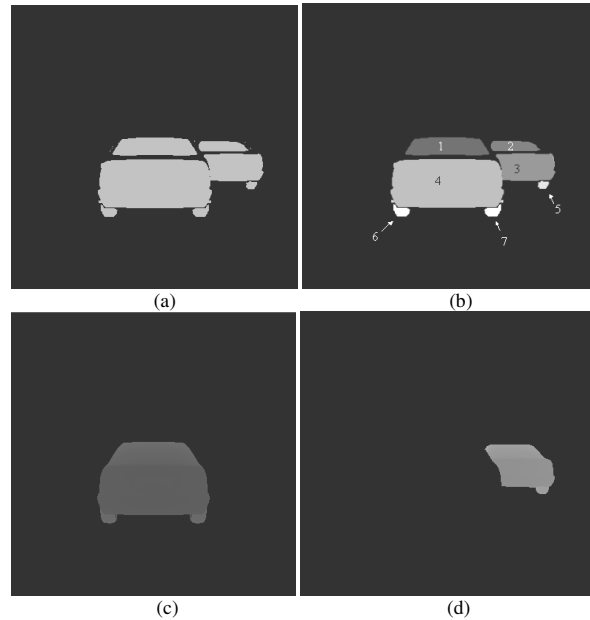


Fig. 3. Segmentation process. (a) The image after slope thresholding; (b) Region growing into atomic regions; (c) The segmented left car; (d) The segmented right car.

Through the region-growing algorithm we can combine the pixels with the steep-slope property into atomic regions (Fig.

3(b)), whereby the average height of each region can be computed. Subsequently, starting at the region with the lowest mean height, pixels are merged (also via region-growing) if one of the following two conditions is met:

- 1) The pixel is spatially adjacent to the previous one and its height is more than a predefined value (e.g., 1/3 mean height of the atomic region) for the atomic region.
- 2) The pixel is spatially adjacent in depth to the previous one and has a high slope value.

The spatial adjacency of pixels in both conditions is important for merging pixels in one object and separating pixels in different objects. The first condition can make the flat surfaces between steep surface parts merge into the obstacles. The second condition ensures merging the pixels with steep-slope.

After segmentation, the road (or ground) is removed and the obstacles (two cars) are segmented (Fig. 3(c)-(d)), which makes the motion estimation of every object (car) possible.

### III. SIMULATIONS

Range images can provide precise measurements of the geometry of the 3-D environment, including all three coordinates of the objects points. In particular, taking advantage of the in depth resolution (range), it is possible to perform an accurate estimation of objects that have undergone 3-D translational and rotational movements. Estimating the complex 3-D motion displacement can best be accomplished by using time varying CEM, as proposed by Horn and Harris [12]. Their algorithm is based on the assumption that most of the surface is smooth so that local tangent planes can be constructed.

In our earlier investigation, we have shown that the estimation accuracy can be greatly enhanced by recursively estimating the 3-D motion parameters [13]. In this approach the motion parameters are iteratively estimated, where the previous estimates are used in the process at each iteration. Motion estimation is based on minimizing the mean-square difference between the current frame and the motion compensated previous frame. Thus, in this recursive process two consecutive video frames (generated at a fixed frame rate) are used to measure the rate of depth change. After each iteration, the estimated motion vector is used to reconstruct the compensated previous frame for the next iteration. A detailed description of the motion estimation technique can be found in [13]. However, in this paper we have used both the Horn and recursive techniques to evaluate the performance of the combined segmentation and motion estimation techniques used for tracking multiple objects.

In order to quantitatively analyze the segmentation and the 3-D motion estimation algorithms [13] we have developed a method to synthetically generate sequences of moving range images. In particular, these moving images are produced in such a way that a 3-D object can be displaced in accordance with the predefined motion displacement parameters. These images can allow us to evaluate the accuracy of estimated

motion vectors with reference to the actual displacement parameters.

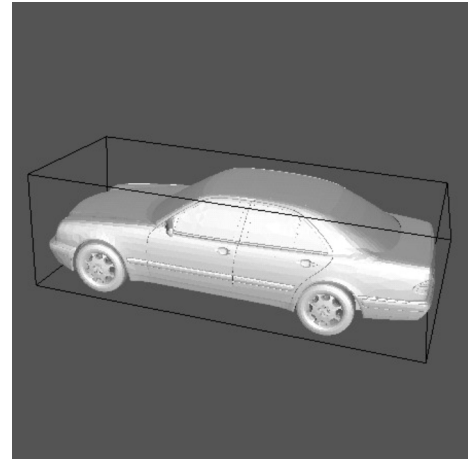


Fig 4. The OOGL file "Auto".

#### A. Generating Range Video Data

Moving range image sequences were constructed via 3-D OOGL (Object Oriented Graphics Library) files. OOGL is a 3-D object data file in which an object is defined by vertices, lines and surfaces. Fig. 4 shows a OOGL file called "Auto", which was selected here to generate range video sequences for our simulation [14]. The 3-D OOGL image was then used to generate a sequence of 2.5-D moving range image files (RIF).

A RIF file is a range image format, which is based on the Cartesian coordinates ( $X, Y, Z$  components) and consists of the object points and the Mask map (indicates where there are object points). In this format, each frame is constructed by first displacing the object in the OOGL file and then transforming it to the RIF file. In this way we can create a sequence of moving range images (frames) where the object in each frame can be displaced by a predefined 3-D motion vector. Two consecutive transformed RIF images were then used as the inputs to the motion estimator.

To get the RIF files from the OOGL files, we must set some parameters such as: view plane normal, view up vector, camera position ("view reference point"), resolution of the output images, size of the camera image plane, focal length, and so on. Fig. 5 shows some of the parameters.

To move objects, one method is to change the view reference point; up vector and view plane normal when converting OOGL file to RIF file. The other method, used in our paper, is adding a transformation matrix in the OOGL file and then creating the second frame. The transformation matrix is a  $4 \times 4$  real matrix for homogeneous object transformation. It can represent all of the 3D transformations such as rotation, translation, scaling, shearing and perspective. It acts by multiplication on the right of vectors. Thus, if  $p$  is a 4-element row vector representing homogeneous coordinates of a point in the OOGL object, and  $M$  is the  $4 \times 4$  matrix, then the transformed point is  $p' = pM$ . Suppose that frame 1  $F_1 = [X_1, Y_1, Z_1]$  and transformation matrix  $M$ , frame 2

can be represented as:

$$[F_2, 1] = [F_1, 1] \times M.$$

For rigid motion objects with sensor-centered coordinates,  $M$  is in the form of

$$M = \begin{bmatrix} 1 & C & -B & 0 \\ -C & 1 & A & 0 \\ B & -A & 1 & 0 \\ U & V & W & 1 \end{bmatrix}.$$

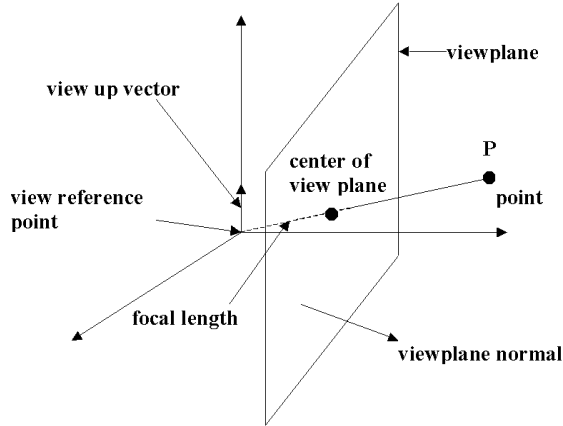


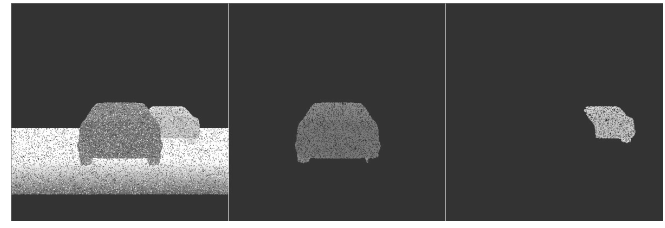
Fig. 5. Camera parameters

### B. Results

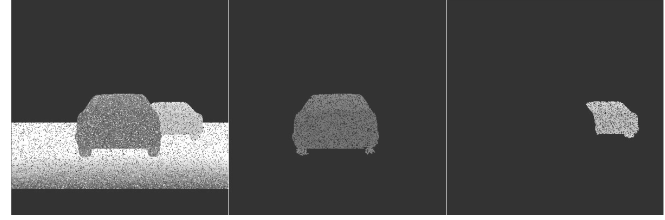
In order to simulate a realistic situation as in normal traffic, we generated range video sequences of scenes consisting of multiple objects (e.g., cars and roads). In our approach, such range image sequences are realized by first generating single-object images in RIF format. These images are then combined according to the depth value of every point. That is, for each point the sample values from the images in the same position with the closest point (the point with the least depth value) will be selected as the point in the combined range image. Fig. 6 and Fig. 7 illustrate the first frame of the two sequences used in our experiments. Both sequences are comprised of three objects: two cars and a flat surface (ground).

In the first sequence both cars are driving in a forward direction and the rear sides of the cars are captured (see Fig. 6). In the other sequence, the first car (the car on the left in Fig. 7) is moving forward, but the second car is driving in a cross section from right to left.

In Fig. 6, the parameters for the car on the left side are:  $U=1$ ,  $V=-0.5$ ,  $W=1$ ,  $A=-0.01$ ,  $B=0.02$ , and  $C=0.01$ . For the car on the right side, the parameters are set as:  $U=1$ ,  $V=0.5$ ,  $W=-1$ ,  $A=0.01$ ,  $B=-0.02$ , and  $C=0.01$ . In Fig. 7, the parameters for the car on the left side are:  $U=1$ ,  $V=-0.5$ ,  $W=-1$ ,  $A=0.01$ ,  $B=-0.02$ , and  $C=0.01$ . The parameters for the car on the right of the image (see the side of the car in Fig. 7) are:  $U=-1$ ,  $V=0.5$ ,  $W=1$ ,  $A=0.01$ ,  $B=0.02$ , and  $C=0.01$ .

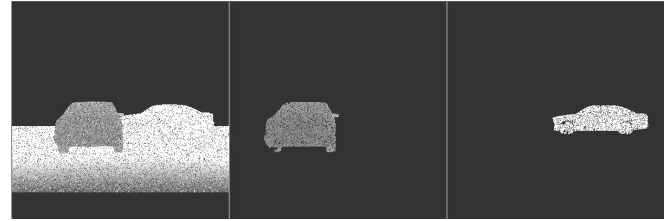


(a)

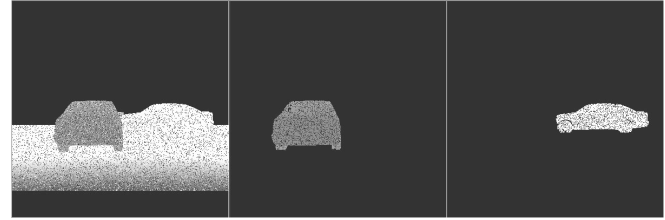


(b)

Fig. 6. Object segmentation results. (a) The first image; (b) The second image. For both the images, from left to right are: (1) The original image; (2) The segmented left car; (3): The segmented right car.



(a)



(b)

Fig. 7. Object segmentation results. (a) The first image; (b) The second image. For both the images, from left to right are: (1) The original image; (2) The segmented left car; (3): The segmented right car.

Since these images are synthetically generated, it would be necessary to deliberately corrupt them with noise. This would allow us to test the resistance of the segmentation and motion estimation schemes under more realistic conditions. Thus, in our experiments we have corrupted the range images with 20% salt and pepper impulse noise (with uniform distribution) [16].

To estimate the motion parameters in both scenarios (Fig. 6 and Fig. 7) we first applied the segmentation scheme to separate the objects. As can be seen from Fig. 6 and Fig. 7 the road is removed and both cars have been successfully separated despite the 20% impulse noise. This is mainly due to the effect of filtering shown in (2) and (3), which plays a crucial role in improving the segmentation performance.

The next step is to estimate the motion parameters for each segmented object. We then applied a recursive scheme to estimate the motion parameters for both scenarios (please note that the results for the first iteration correspond to the Horn & Harris algorithm).

We use two criteria as a measure of performance: Mean Square Error (MSE) and Motion Vector Error (MVE). The MSE between Frame 1 and Frame 2 is defined as,

$$MSE = \frac{1}{m} \sum_R [(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2],$$

where  $R$  is the region that combines both objects in two frames,  $R = MASK_1 \cup MASK_2$ , and  $m$  is the number of the points in region  $R$ .

Given the true motion parameters  $(U, V, W, A, B, C)$  and the estimated ones  $(\hat{U}, \hat{V}, \hat{W}, \hat{A}, \hat{B}, \hat{C})$ , the MVE is defined as:

$$MVE = \frac{|U - \hat{U}| + |V - \hat{V}| + |W - \hat{W}| + |A - \hat{A}| + |B - \hat{B}| + |C - \hat{C}|}{|U| + |V| + |W| + |A| + |B| + |C|}.$$

Fig. 8 and Fig. 9 show the motion estimation results (subjective and objective) of the two cars for scenario 1 (see Fig. 6). Fig. 10 and Fig. 11 present the motion estimation results for scenario 2 (see Fig. 7). For both scenarios we can observe a substantial improvement in estimating the motion parameters using the recursive method, as compared with the Horn and Harris algorithm. The improvement in the estimation accuracy after each iteration can be seen from the objective results shown in Fig. 9 and Fig. 11 (for scenario 1 and scenario 2, respectively). Looking at MSE-based objective results in Fig 9(a) in particular, we can observe that the MSE of the right car is considerably higher than the left car. This is mainly due to the considerable overlap with the left car resulting in a variational partial covering of its rear side. Nonetheless, as far as its convergence to the actual motion displacement vector is concerned (i.e., MVE criterion), we can clearly observe the effectiveness of the motion estimation scheme from Fig. 9(b). We can also observe a similar performance in scenario 2 (see Fig. 10 and Fig. 11).

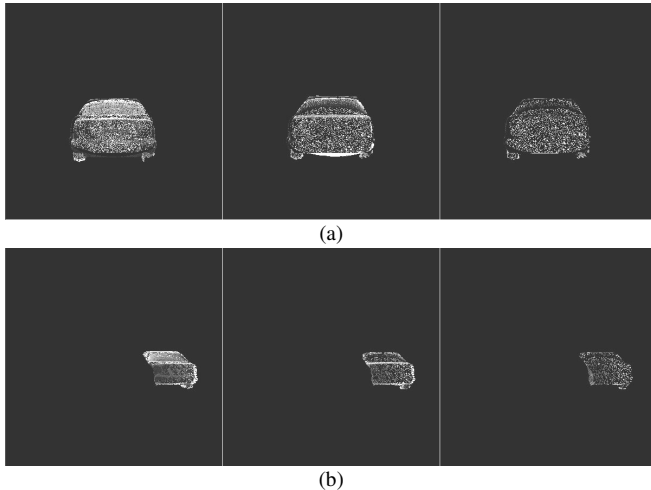


Fig. 8. Subjective comparison of different algorithms. (a) The results for the left car; (b) The results for the right car. For both cars, from left to right are: (1) The difference image between the original two images; (2) The difference image between the second image and the estimated second image using the estimated motion parameters of the first iteration (Horn's algorithm); (3) The difference image between the second image and the estimated second image using the motion parameters of the final iteration (the recursive algorithm).

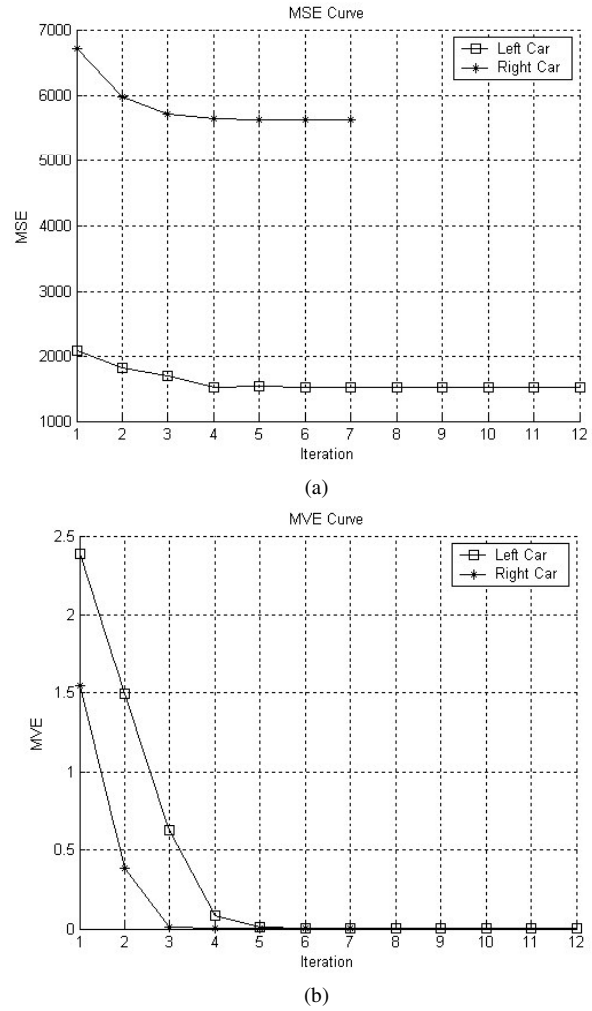


Fig. 9. Objective evaluations of the recursive motion estimation scheme for the left car ( $U=1; V=-0.5; W=1; A=-0.01; B=0.02; C=0.01$ ) and the right car ( $U=1; V=0.5; W=-1; A=0.01; B=-0.02; C=0.01$ ) segmented from Fig. 6. (a) MSE; (b) MVE.

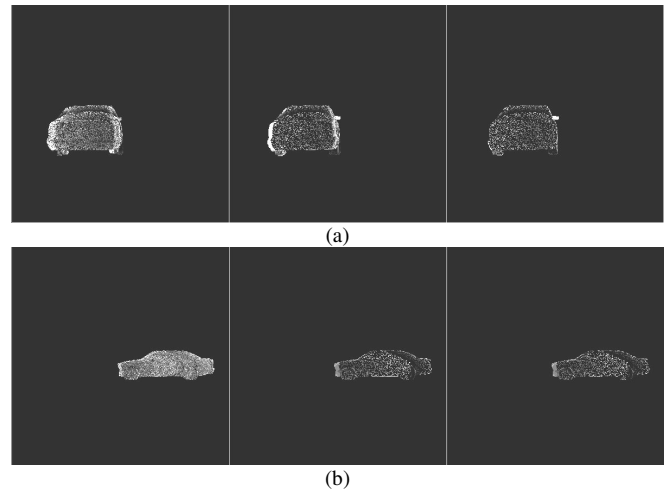


Fig. 10. Subjective comparison of different algorithms. (a) The results for the left car; (b) The results for the right car. For both cars, from left to right are: (1) The difference image between the original two images; (2) The difference image between the second image and the estimated second image using the estimated motion parameters of the first iteration (Horn's algorithm); (3) The difference image between the second image and the estimated second image using the motion parameters of the final iteration (the recursive algorithm).

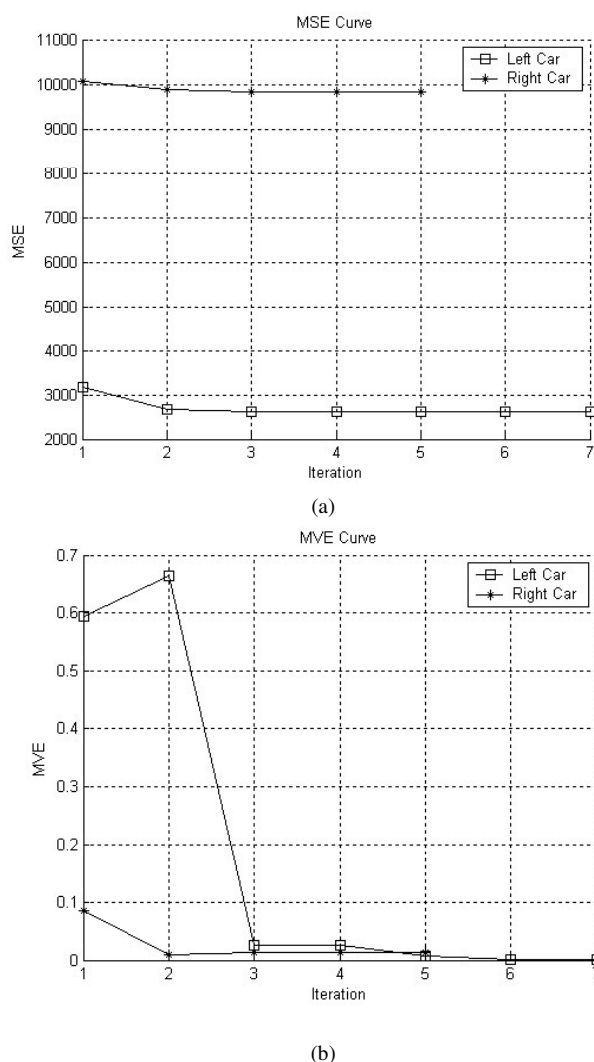


Fig. 11. Objective evaluations of the recursive motion estimation scheme for the left car ( $U=1$ ;  $V=-0.5$ ;  $W=-1$ ;  $A=0.01$ ;  $B=-0.02$ ;  $C=0.01$ ) and the right car ( $U=-1$ ;  $V=0.5$ ;  $W=1$ ;  $A=0.01$ ;  $B=0.02$ ;  $C=0.01$ ) segmented from Fig. 7. (a) MSE; (b) MVE.

#### IV. CONCLUSION

Humans are very good at detecting when a collision may occur, but may not be able to react quickly enough to avoid it. With the rapid progress of range camera technology, it has become possible to acquire raster depth measurements of an object, which can then be used in pre-crash awareness and prevention. In this paper we considered a region-growing scheme to separate multiple moving obstacles. A 3-D motion estimation scheme is used to estimate the velocity of the vehicles in front of the range video camera. It is shown that a combination of region growing segmentation and recursive motion estimation can enhance the tracking ability considerably.

#### REFERENCES

- [1] X. Yu, T.D. Bui, and A. Krzyzak, "Robust Estimation for Range Image Segmentation and Reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 530-538, 1994.
- [2] A. Hoover, G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher, "An experimental comparison of range image segmentation algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 673-689, July 1996.
- [3] D. Zhao, and X. Zhang, "Range-Data-Based Object Surface Segmentation via Edges and Critical Points," *IEEE Trans. Image Proc.*, vol. 6, pp. 826-830, June 1997.
- [4] K.-M. Lee, P. Meer, and R.-H. Park, "Robust Adaptive Segmentation of Range Images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 200-205, Feb. 1998.
- [5] A. Bab-Hadiashar, and D. Suter, "Robust Range Segmentation," in *Proc. ICPR 1998*, pp. 969-971, Aug. 1998.
- [6] X. Jiang, and H. Bunke, "Edge Detection in Range Images Based on Scan Line Approximation," *Computer Vision and Image Understanding*, vol. 73, pp. 183-199, Feb. 1999.
- [7] X. Jiang, "An Adaptive Contour Closure Algorithm and Its Experimental Evaluation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1252-1265, Nov. 2000.
- [8] L. Silva, O. R. P. Bellon, and P. F. U. Gotardo, "Edge-Based Image Segmentation using Curvature Sign Maps from Reflectance and Range Images," in *Proc. ICIP 2001*, pp. 730-733.
- [9] J. Min, M. Powell, and K. W. Bowyer, "Automated Performance Evaluation of Range Image Segmentation Algorithms," *IEEE Trans. Syst., Man, and Cybern. B*, vol. 34, pp. 263-271, Feb. 2004.
- [10] P. F. U. Gotardo, O. R. P. Bellon, K. L. Byer, and L. Silva, "Range Image Segmentation Into Planar and Quadric Surfaces Using an Improved Robust Estimator and Genetic Algorithm," *IEEE Trans. Syst., Man, and Cybern. B*, vol. 34, pp. 2303-2316, Dec. 2004.
- [11] B. Sabata, "Analysis of Low-Resolution Range Image Sequences," PhD thesis, University of Bern, Switzerland, 2000.
- [12] B. K. P. Horn and J. Harris, "Rigid body motion from range image sequences," *CVGIP: Image Understanding*, vol. 53, no. 1, pp. 1-13, 1991.
- [13] H. Gharavi, and S. Gao, "3-D Motion Estimation Using Range Data," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, pp. 133-143, 2007.
- [14] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3D Object Recognition from Range Images using Local Feature Histograms," *Proceedings of CVPR 2001*, vol. 2, pp. 394-399, 2001.
- [15] P.V.C. Hough, "Machine Analysis of Bubble Chamber Pictures," *International Conference on High Energy Accelerators and Instrumentation*, CERN, 1959.
- [16] K. Pulli, and M. Pietikainen, "Range Image Segmentation Based on Decomposition of Surface Normals," in *Proc. Scandinavian Conference on Image Analysis (SCIA)*, Vol. 2, pp. 893-899, 1993.