Map Building for a Mobile Robot from Sensory Data

MINORU ASADA, MEMBER, IEEE

Abstract—**The development of an autonomous land vehicle (ALV) is a central problem in artificial intelligence and robotics, and has been extensively studied. To perform visual navigation, a robot must gather information about its environment through external sensors, interpret the output of these sensors, construct a scene map and a plan sufficient for the task at hand, and then monitor and execute the plan. As a first step, real time visual navigation systems for road following were developed in which simple methods for detecting road edges were applied in simple environments. For even slightly more complicated scenes, the difficulty of the problem increases dramatically, therefore a world model such as a map could be very important for successful navigation through such environments. A method for building a three-dimensional (3-D) world model for a mobile robot from sensory data derived from outdoor scenes is presented. The 3-D world model consists of four kinds of maps: a physical sensor map, a virtual sensor map, a local map, and a global map. First, a range image (physical sensor map) is transformed to a height map (virtual sensor map) relative to the mobile robot. Next, the height map is segmented into unexplored, occluded, traversable and obstacle regions from the height information. Moreover, obstacle regions are classified into artificial objects or natural objects according to their geometrical properties such as slope and curvature. A drawback of the height map (recovery of planes vertical to the ground plane) is overcome by using multiple height maps that include the maximum and minimum height for each point on the ground plane. Multiple height maps are useful not only for finding vertical planes but also for mapping obstacle regions into video image for segmentation. Finally, height maps are integrated into a local map by matching geometrical parameters and by updating region labels. The results obtained using landscape models and ALV simulator of the University of Maryland are shown, and constructing a global map with local maps is discussed.**

I. Introduction

THE DEVELOPMENT of an autonomous land vehicle (ALV) is a central problem in artificial intelligence and robotics, and has been extensively studied [1]–[2]. To perform visual navigation, a robot must gather information about its environment through external sensors, interpret the output of these sensors, construct a scene map and a plan sufficient for the task at hand, and then monitor and execute the plan. As a first step, real time visual navigation systems for road following were developed in which simple methods for detecting road edges were applied in simple environments [1], [4], [5], [12]. For even slightly more complicated scenes, the difficulty of the problem increases dramatically, therefore a world model such as a map could be very important for successful navigation through such environments.

Sometimes, accurate, quantitative maps may be available in advance [13], more often, maps are less descriptive and provide only global information as in a conventional geographical map [14]. In other cases, the robot may try to construct the map from sensory date in unknown environments. Hebert and Kanade [15] have analyzed ERIM range images and constructed a surface property map represented in a Cartesian coordinate system viewed from top, which yields surface type of each point and its geometric parameters for segmentation of scene map into traversable and obstacle regions. Tsuji and Zheng [16] discussed the differences between the two-dimensional (2-D) maps in [15] and perspective maps proposed in their stereo-vision-based mobile robot system. Their point is that 2-D maps are easy to understand but do not naturally capture sensor resolution and accuracy. They used perspective maps for navigation in which three-dimensional (3-D) information obtained by stereo vision is represented in the image coordinate system. However, integration of perspective maps obtained at different locations on a single perspective map seems difficult. Having both types of maps in a hierarchical representation and referring to each other when necessary is one solution for the above problem.

Elfes [17] has developed a sonar-based mapping and navigation system which constructs sonar maps of the environments viewed from the top and updates them with recently acquired sonar information. He proposed a hierarchical representation of sonar map which includes three kinds of axes: an abstract axis, a resolution axis, and a geographic axis. In his system, the outputs of sonar sensors are directly mapped to a 2-D map, therefore, the difference between sensor maps and the 2-D maps is implicit, and other sensory data such as video images and range images seem difficult to be represented in this hierarchy.

In this paper, we propose a method for building a hierarchical representation of a 3-D world model for a

Manuscript received April 1, 1989; revised January 15, 1990. This work was supported in part by the Defense Advanced Research Projects Agency and the U.S. Army Engineer Topographic Laboratory under contract DACA76-84-C-0004, and in part by the Grant-In-Aid for Scientific Research from the Ministry of Education, Science, and Culture, Japanese Government. This work was partially presented at the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA, 1988.

The author was with Center for Automation Research, University of Maryland, College Park, MD 20742. He is now with the Department of Mechanical Engineering for Computer-Controlled Machinery, Osaka University, Suita, Osaka 565, Japan.

IEEE Log Number 9037596.

0018-9472/90/1100-1326$01.00

mobile robot, making the relation between coordinate systems at different levels explicit. With this model, the mobile robot can derive useful information from a map at adequate level to accomplish various kinds of tasks such as visual navigation, obstacle avoidance, and landmarks and/or objects recognition. The 3-D world model consists of four kinds of maps: a physical sensor map, a virtual sensor map, a local map, and a global map. A physical sensor map usually represents sensory data or analyzed data in the sensor-based coordinate system from which the sensory data is taken (e.g., perspective map in [16], or ERIM range image in [15]).

A virtual sensor map represents the sensory data (in the physical sensor map) in the vehicle-centered Cartesian coordinate system. Any other type of coordinate system such as a cylindrical one can be applicable to the virtual sensor map representation in the context of representing sensory data in the vehicle-centered coordinate system. However, the Cartesian mapping seems more suitable for the virtual sensor map representation because the size of the cell on the map (it corresponds to the resolution of the map) is constant everywhere, therefore, fusing the data at the same point but observed from different view points is much easier than other type of mapping such as a cylindrical mapping or Delaunay triangulation which requires a complex algorithm to access data points and their neighborhoods [18]. As one example of a particular instance of the virtual sensor map, we introduce a height map that represents the height information transformed from a range image in the vehicle-centered Cartesian coordinate system. The 2-D map in [15] and Cartesian elevation map (CEM) in [19] are also categorized into the virtual sensor map representation. Virtual sensor maps are integrated into a local map that is represented in the object-centered coordinate system. The local map has its own reference (object) on which the integration of the virtual sensor maps is based, therefore, a new local map with a new reference is generated when the current reference cannot be observed as the robot moves. Thus, a number of local maps are generated along with the robot navigation. A global map consists of these local maps and the geometrical relationship between them. How to build a global map with relational local maps is proposed by Asada et al. [26]. In this paper, we focus on the building the physical sensor maps, the virtual sensor maps, and the local map from the video and range data, and have not dealt with how to build the global map.

In our system, a range image, one example of the physical sensor map, is transformed to a height map, one example of the virtual sensor map, in the mobile robot centered Cartesian coordinate system. The height is estimated from the assumed ground plane on which the vehicle exists. First, we segment the height map into unexplored, occluded, traversable and obstacle regions from the height information, and then classify obstacle regions into artificial objects or natural objects according to their geometrical properties such as slope and curvature. A drawback of the height map—recovery of planes vertical to the ground plane—is overcome by using a multiple height map that includes the maximum and minimum heights for each point on the ground plane. The multiple height map is useful not only for finding vertical planes but also for mapping obstacle regions into video image (another sensor map) for segmentation. Finally, the system integrates height maps observed at different locations into a local map, matching geometrical parameters of obstacle and traversable regions and updating region labels. We show the results applied to landscape models using ALV simulator of the University of Maryland [4], and discuss about constructing a global map with local maps.

II. System Configuration

A. Physical Simulation System of ALV and Its Environments

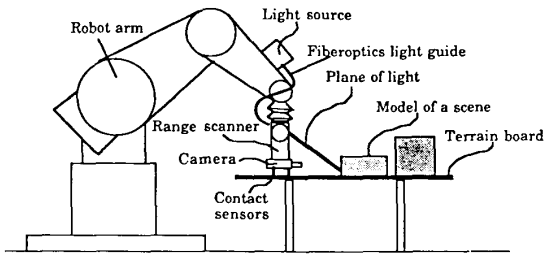
Our ALV physical simulation system was developed in our laboratory [4] for providing a low cost experimental environment for navigation (as opposed to an outdoor vehicle [7], [15]). A range finder based on structured light was recently added to this system. Planes of light are projected from a rotating mirror controlled by a stepping motor (see [20] for more detail). More recently, we extended the system in two ways. First, we developed a drive simulator program which controls the speed and steering angle of the vehicle (robot arm) during the motion. The camera height and camera tilt to the ground plane are kept constant during the motion through the position feedback of three leg sensors attached to the camera.

Previously, a wooden terrain board on which a road network is painted was set vertically to increase the flexibility of camera motion simulated by the robot arm. Due to its vertical setting, it was very difficult to put landscape model such as trees, bushes, buildings and other vehicles on the board. Thus, we set the terrain board horizontally so that we can place any landscape model without permanently fixing their positions. Figs. 1(a) and (b) show our experimental setup and a picture of the new simulation board with many landscape models such as trees, bushes, cabins, mail box and cars. The robot arm attached with a TV camera and a light-stripe range scanner is set on the board to input a picture and a range image.

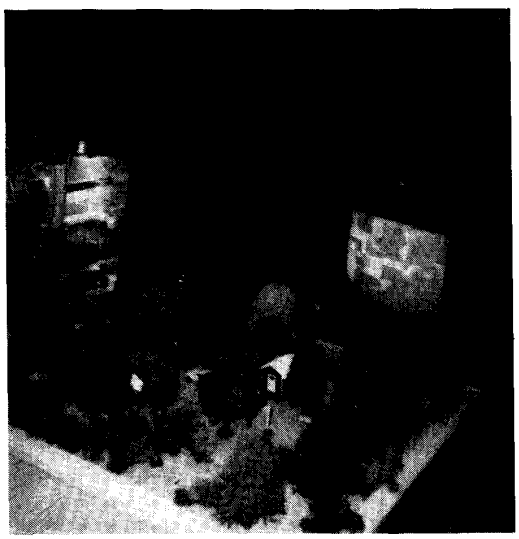
B. Overall of Map Building System

Fig. 2 shows the architecture of our system. In this figure, we omit other modules such as path planner, navigator, pilot and supervisor in [8] in order to concentrate on the map building system.

The 3-D world model for a mobile robot consists of four kinds of maps: a physical sensor map, a virtual sensor map, a local map, and a global map. Elfes [17] proposed multiple axes of representation of a sensor map in his sonar mapping and navigation system (resolution axis, geographical axis and abstraction axis) and adopted three



(a)



(b)

Fig. 1. Simulation board. (a) Experimental set up (from [20]). (b) Photo of new simulation board.

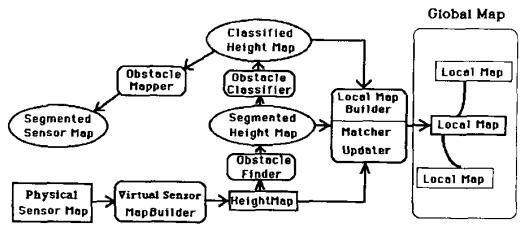


Fig. 2. Overview of map building system.

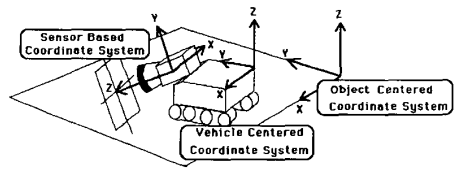


Fig. 3. Geometrical relation of coordinate systems between three maps.

levels (view, local map, global map) in the geographical axis. We extend and generalize these levels so that other sensory data such as range and video data can be represented. Fig. 3 shows the geometrical relation between the three coordinate systems of the physical sensor map (sensor-based), the virtual sensor map (vehicle-centered), and the local map (object-centered). The global map is a set of the local maps and discussed in Section IV. Each sensor has its own coordinate system; for example, an intensity image is represented in the camera-centered coordinate system and a range image in the range-finder-centered coordinate system, both of which are fixed to the robot (vehicle). Here, we assume that the relation between sensor coordinate systems and vehicle coordinate system is known, and that the motion information is available, but not always accurate. A virtual sensor map builder (VSMB) builds the virtual sensor maps from the physical sensor map. Stereo matching, which we do not consider in this paper, is one possible strategy of virtual sensor map building for obtaining the depth map (virtual sensor map). Here, we introduce a height map obtained from the range image as a virtual sensor map. The height map is analyzed by the obstacle finder (OF) and the obstacle classifier (OC) to segment it into unexplored, occluded, traversable and obstacle regions and then to classify obstacle regions into artificial or natural objects. The result of the height map analysis is mapped onto the intensity image by the obstacle mapper (OM) in order to segment the intensity image. The local map builder (LMB) builds a local map, matching and updating virtual sensor maps at different observing stations in the world coordinate system. In the following, each module is described along with some experimental results.

III. Height Map Analysis

A. Virtual Sensor Map Builder (from Range Image to Height Map)

The virtual sensor map builder builds virtual sensor maps from physical sensor maps. Here, we deal with a video image taken by a single camera and a range image obtained from our range finder [20] as physical sensor maps. Even though we use the same camera to take both video and range data, our idea can be applicable to other type of range data such as a range image taken by ERIM range finder [15]. We discuss the differences between two types of the range images in Section IV. Figs. 4 show examples of these physical sensor maps. The input scene includes a straight road, T-type intersection, two cabins, one truck, two cars, a mailbox, a stop sign at the intersection, trees and bushes as shown in Figs. 4(a), (c), and (e). Figs. 4(b), (d), and (f) are the corresponding range image to Fig. 4(a), (c), and (e), respectively. All of the images are 512×512, and both the range and the intensity values are quantized into 256 levels (8 bits). The darker points are closer to the range finder and the brighter points are farther from it. In the white regions, range information is not available due to inadequate reflection or occlusion. Although actual range finders such as the ERIM range scanner [15] measure the radial distances rather than Cartesian coordinates in both axes, the range image ob-

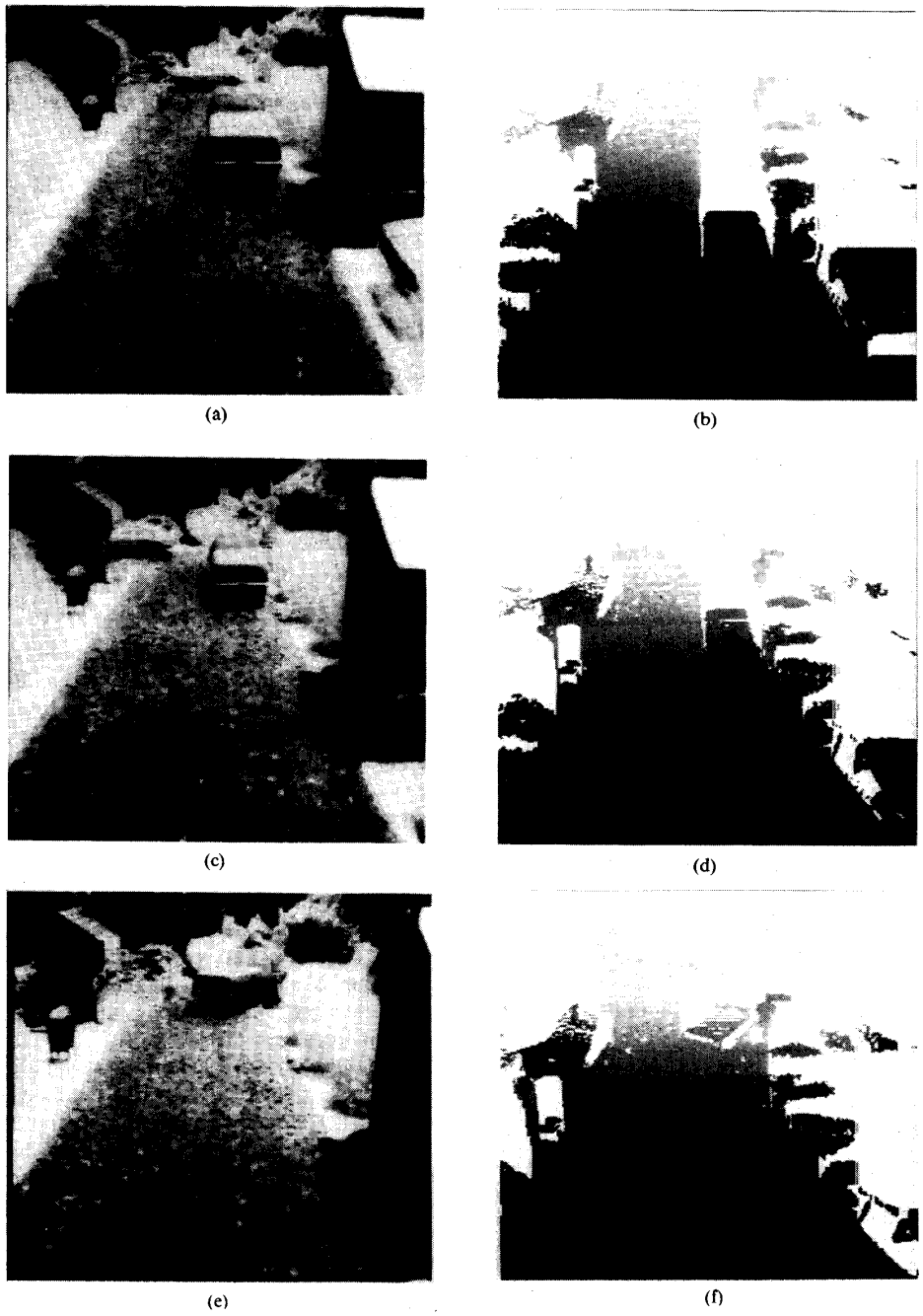


Fig. 4. Sensor maps. (a) First intensity image. (b) First range image. (c) Third intensity images. (d) Third range image. (e) Fifth intensity image. (f) Fifth range image.

tained from our range finder has irregular coordinate aces due to its special ranging geometry [20]; the vertical coordinate is radial (scanning angle of the light plane) but the horizontal coordinate is the same as that of the intensity image because range calculation is based on the triangulation with light planes and a single TV camera. Fig. 5(a) shows calibration parameters required to calculate the range.

The range image is transformed to a height map in the vehicle centered coordinate system based on the known height and tilt of the range finder relative to the vehicle. The position of a point in a given coordinate system can be derived from the obtained range R and the direction to that point (it corresponds to the coordinates (x_r, y_r) on the range image). We use the Cartesian coordinate system O-XYH shown in Fig. 5(b), in which case the XY plane

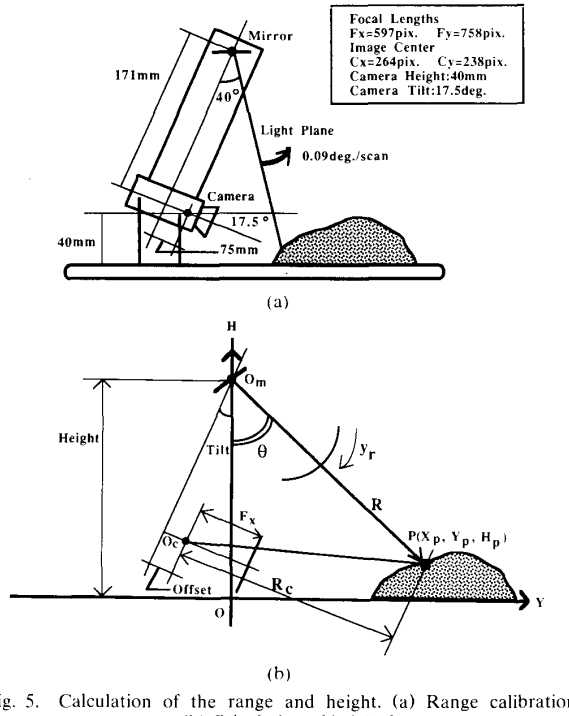


Fig. 5. Calculation of the range and height. (a) Range calibration parameters. (b) Calculation of height from range.

corresponds to the ground plane, and the origin O is just below the range finder (the reflecting mirror). The coordinates of a point $P(X_p, Y_p, H_p)$ estimated by the range finder are given by the following equations:

$$X_p = \frac{R_c x_r}{F_x}, \qquad Y_p = R\sin\theta, \qquad H_p = \text{Height} - R\cos\theta$$

where F_x, Height, θ, and R_c are the focal length in the horizontal direction of the camera image, the height of the range finder (the reflecting mirror), the vertical scanning angle, and the range in the camera-centered coordinate system which is needed to estimate X-coordinate. The last two parameters are derived from the column position y_r in the range image, the tilt angle of the camera Tilt, and the offset parameter Offset by the equations:

$$\theta = ay_r + b, \qquad R_c = R\sin(\theta + \text{Tilt}) - \text{Offset}$$

where a and b are parameters for the transformation from the column position to the vertical scanning angle. An adequate area on the ground plane in front of the vehicle is assigned for a height map and the 3-D coordinates (X_p, Y_p, H_p) are quantized into 8bits numbers. The height map is a 256×256 image, each pixel corresponds to 1 mm^2 on the simulation board. The entire map corresponds to a square of side length 256 mm; the scale of the new simulation board is 87:1 (HO scale). Gray levels encode the height from the assumed ground plane. Since the range is sparse and noisy at far points, smoothing is necessary. We applied an edge-preserving smoothing method [22] to the height map in order to avoid a mixed pixel problem of high and low points. Figs. 6 show the filtered height map of the input scene (Fig. 4(b)); Fig. 6(a) shows a gray level image and Fig. 6(b) shows its perspective view.

One drawback of the height map is that it is unable to represent vertical planes, especially these under horizontal or sloped planes because the range information corresponding to multiple points in the vertical direction is reduced to one point in the height map. This is especially undesirable since the range information on the vertical planes is more accurate than that on the horizontal planes. Thus, we compute a multiple height map for one range image that includes the maximum and minimum heights for each point on the height map, and the number of points in the range image which are mapped to one location on the height map. In Figs. 6, the maximum height is shown.

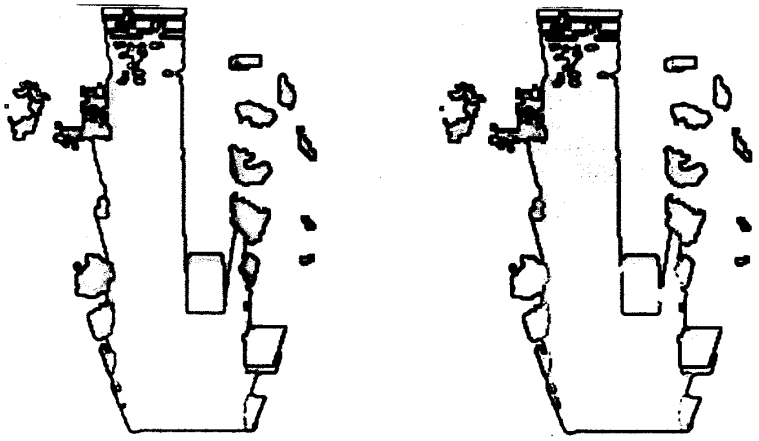
B. Obstacle Finder (Segmentation of Height Map)

The first step of the height map analysis is to segment the height map into unexplored, occluded, traversable and obstacle regions. The height map consists of two types of regions—these in which the height information is available or these it is not. The latter regions are classified into unexplored or occluded regions. Unexplored regions are outside the visual field of the range finder, and therefore are easily detected by using the calibration parameters of the range finder (height, tilt and scanning angle). The remaining regions in this category are labeled as occluded regions. Some regions which are not occluded may be classified into occluded regions if the height information is unavailable due to causes such as inadequate reflection. These regions can be often seen inside bushes or trees with many leaves.

Finding traversable regions is straightforward. First, identify these points close to the assumed ground plane and construct atomic regions for the traversable region. Next, expand the atomic regions by merging other points surrounding them which have low slope and low curvature. The desirable feature of the height map is that the outputs of the first and second derivatives of the height map correspond to the magnitudes of the slope and curvature of the surface because the locations of the points in the height map are represented with Cartesian coordinates. Strictly speaking, the outputs of the second derivative of height do not directly correspond to the curvature of the surface, but at least, it outputs zeros for the sloped plane such as a roof plane of a house. Figs. 7 show the magnitudes of the first (Sobel) and second derivatives of the height map (Fig. 6(a)) whose mask size is 3 by 3 pixels. The remaining regions are labeled as obstacle regions. Fig. 8 shows the final result of the segmentation of the height map. White, light gray, dark gray and black regions are unexplored, occluded, traversable and obstacle regions, respectively. We can see that

(a) (b)

Fig. 6. Height map. (a) Gray image. (b) Perspective view of (a).



(a) (b)

Fig. 7. (a) Slope map. (b) Curvature map.

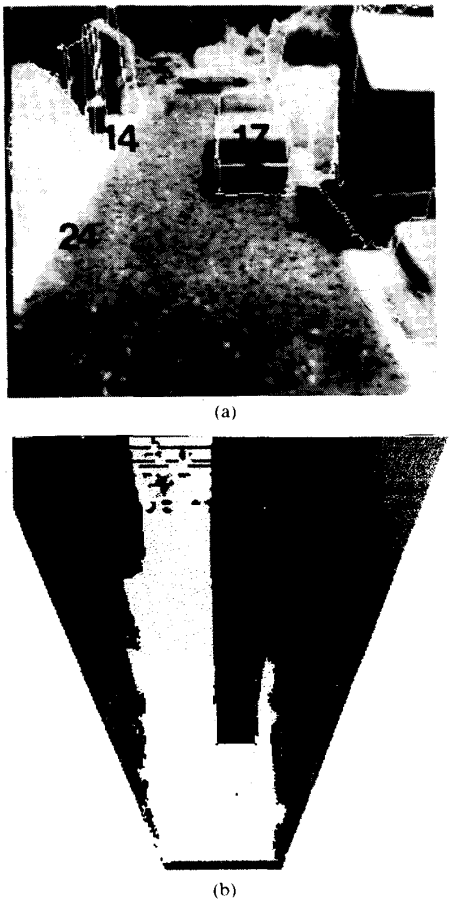
the boundary of the obstacle regions has high slope and/or high curvature (see Figs. 7).

The result of segmentation of the height map should be useful for path planning since many path planning algorithm are based on a top view of the configuration of obstacles and free space [23].

C. Obstacle Classifier

The segmented height map constructed by the OF is very useful for navigation tasks such as avoiding obstacles, but does not contain sufficient explicit information enough for higher level tasks such as landmark or object recognition. As a first step in object recognition, we try to classify obstacle regions as artificial objects or parts of natural objects. Many artificial objects such as cabins, cars, mailboxes and road signs shown in Figs. 4 have planar surfaces, which yield constant slope and low curvature in the height map and linear features in the intensity image. On the other hand, natural objects such as trees and bushes have fine structures with convex and concave surfaces, which yield various slopes and/or high curvatures in the height map and therefore large variance of brightness in the intensity image (the reverse is not always true).

Thus, utilization of not only the height map but also the intensity image is useful for obstacle classification. In order to use the brightness information in the intensity image, we map the obstacle regions to the intensity image to segment it. The mapping of obstacles to the intensity image seems at first straightforward based on the geometrical relation between the camera and the range finder. However, it is complicated by the need to correctly choose



(a)

(b)

Fig. 8. Segmentation of height map.

between the maximum and minimum heights associated with each point in the height map. Figs. 9 show a perspective view of a cube on a plane and its height map. If we map only the maximum value of the height of the cube, only a top surface of the cube is cut out from its perspective view and visible side surfaces are left undiscovered (see Fig. 9(a)). We should use the minimum height when the object is bounded by traversable regions and use the maximum height when it is occluding other objects behind it (see Fig. 9(c)). Classifying the boundary of obstacle regions in the height map and using the multiple height map, the obstacle mapper maps the obstacles to the intensity image as follows.

1) Classify each boundary point of the obstacle region in the height map according to the geometrical relation between the point, occluded region and traversable region in the segmented height map (Fig. 8).
2) Use the minimum height if that point is adjacent to a traversable region or that point is an occluded boundary (the boundary point is labeled as an occluded boundary when the occluded region is between the boundary point and the range finder).
3) Use the maximum height if that point is occluding boundary (the boundary point is labeled as an occluding boundary when the boundary point is between occluded region and the range finder).

Figs. 10 show the result of this mapping. The truck and the car in front of the cabin on the right side, the mailbox, the stop sign, and the bushes are finely segmented in the intensity image. The car at the intersection is not mapped because its location is outside the height map (far from the viewer). The roof line of the cabin on the left side is incorrect because of the bad range data. The reason that the top roof line of the cabin on the right side drops suddenly to the ground plane is that there is a lower object behind the cabin and the obstacle region in the height map includes both the cabin and the lower object.

The next step is to classify the obstacle using the properties of the height map and the brightness in the intensity image. The obstacle classifier classifies each resegmented region according to the following criteria.

1) If a region has sufficient size (larger than predetermined threshold) and constant slope (small variance of slope) and low curvature (low mean curvature and small variance of the curvature), then the region is an artificial object.
2) If a region has sufficient size and high curvature (high mean curvature and large variance of the curvature) and large variance of the brightness in the intensity image, then the region is a part of a natural object.
3) Otherwise, the region is regarded as uncertain in the current system.

In Fig. 10(b), white, hatched, and black regions are corresponding to natural, artificial, and uncertain objects, respectively. Small regions are almost always labeled as uncertain. The car in front of the cabin on the right side and the truck are correctly interpreted as artificial objects. However, the roofs of two cabins, the mailbox and the stop sign are misinterpreted as natural objects because of the high curvature due to vertical planes and/or insufficient, noisy range data. The region corresponding to the right side cabin includes a roof and the lower object behind it, therefore resegmentation into two regions is necessary [21]. Uncertain regions and some regions with vertical surfaces would require closer examination for correct interpretation.

D. Local Map Builder

During the motion of the vehicle, the system produces a sequence of virtual sensor maps constructed at different observing stations. These virtual sensor maps should be integrated into a local map in the world centered coordinate system. The local map builder consists of two parts; the first part matches a new height map and the current local map to determine the correct motion parameters of the vehicle, and the second one updates the description of region properties on the integrated local map. Matching

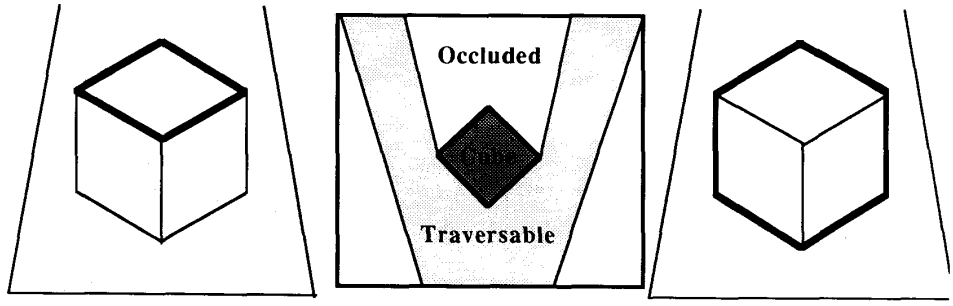


Fig. 9. Cube on a plane and its height map. (a) Only maximum height is used. (b) Height of map (a). (c) Maximum and minimum heights are used.

Fig. 10. Mapping obstacle region into the intensity image. (a) Mapped region. (b) Obstacle map.

is performed between the $(i+1)$th height map H_{i+1} and the local map L_i in which the height maps from the first to the ith observing stations are integrated. If $i=1$, then $L_i = H_i$. The point P_l in the local map L_i can be represented as the point P_h in the height map H_{i+1} by the following equation:

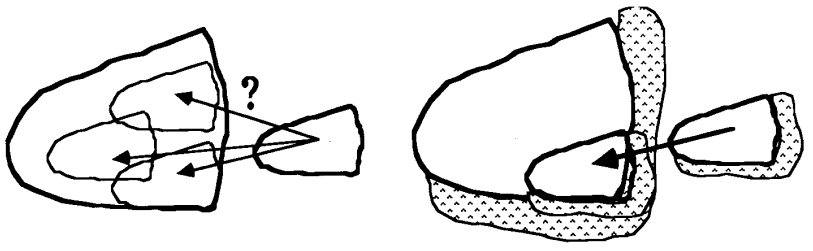
$$P_h = R_{lh}P_l + T_{lh}$$

where R_{lh} is a rotation matrix consisting of three rotational components, $\Delta\alpha$, $\Delta\beta$, and $\Delta\theta$ along each axis, and T_{lh} is a translation vector along each axis, that is, $T_{lh}^{-1} = (\Delta X, \Delta Y, \Delta H)$. The following are matching procedures.

1) Match the traversable regions between H_{i+1} and L_i. Since the traversable regions are usually the larger planar regions in the segmented map and rough estimates of the motion parameters (ΔX, ΔY, ΔH, $\Delta\alpha$, $\Delta\beta$, and $\Delta\theta$) are ordinarily available from the internal sensors of the vehicle, this matching is relatively straightforward. The traversable region in H_{i+1} and L_i are on the XY-plane in each coordinate system because we determined the orientation and location of the coordinate system so that the ground plane corresponds to the XY-plane (see Fig. 5(b)). Therefore, matching the traversable region between H_{i+1} and L_i is performed by overlaying the XY-plane in L_i onto the XY-plane in H_{i+1}. By this matching, $\Delta H, \Delta\alpha, \Delta\beta$ are determined (in our experiments, all are zeros), and the remains are ΔX, ΔY, and $\Delta\theta$; the motion parameters on the XY-plane.
2) Determine the remaining motion parameters ΔX, ΔY, and $\Delta\theta$ by matching the obstacle regions between H_{i+1} and L_i. Search for the motion parameters which take the minimum height difference for each obstacle region, starting the rough motion parameters from the vehicle navigation system as an initial value. We evaluate the following criterion function to find the minimum height difference:

$$\Delta H_i^k(\Delta x, \Delta y, \Delta\theta) = \sqrt{\frac{\Sigma\left(H_{i+1} - L_i^k(\Delta x, \Delta y, \Delta\theta)\right)^2}{N_i^k}}$$

where ΔX, ΔY, $\Delta\theta$, L_i^k, and N_i^k are estimated translation, rotation, the height of kth obstacle region in L_i, and the number of height points compared between H_i and L_{i+1} for the kth obstacle region, respectively. If we pick up only obstacle region, the matching could be ambiguous. For example, a small obstacle region with horizontal plane matches at any position of a large region with hori-



(a) (b)

Fig. 11. Expanding obstacle region to avoid the matching ambiguity.

zontal plane of the same height (see Fig. 11(a)). In order to avoid such an ambiguity, we expand obstacle regions so that they include the height information surrounding them (see Fig. 11(b)).

A special care needs to be taken for moving objects because they have different motion parameters from those of the stationary environment. In the current system, we use a heuristic for detecting moving object. The obstacle surrounded with traversable regions is a candidate moving object because the moving objects should be inside the traversable region (except for flying objects). Three regions numbered 14, 17, and 25 in Fig. 10(b) are candidate moving objects while only a region 17 (truck) is actually moving. Although search area for stationary obstacles is ± 2 mm for translation $(\Delta X, \Delta Y)$ and $\pm 2°$ for rotation $\Delta\theta$ with fine resolution of 1 mm and 1°, coarse to fine search algorithm is applied to the candidates for moving objects since moving objects might be outside the search area for the stationary objects. The algorithm is as follows: estimating motion parameters in a large search area with sparse intervals of translation (± 5 mm) and rotation ($\pm 5°$)first, and then, refining it with fine intervals same as for the stationary obstacles. In Fig. 10, motion parameters for almost all regions are correctly obtained, and as a result, only region 17 is interpreted as an object moving farther from the viewer (towards the intersection). The motion parameters ΔX, ΔY, and $\Delta\theta$ for the stationary objects are 0 mm, 15 mm, and 0°, respectively, and those for the moving object are 0 mm, 45 mm, and 0°, respectively. Fig. 12 shows the histogram of the motion parameters when $\Delta\theta = 0°$, in which the inverse number of the minimum height difference for each obstacle region are accumulated into the corresponding motion parameters $(\Delta X, \Delta Y)$. Two peaks correspond to the stationary regions (larger) and the moving region (smaller). Velocities of the moving objects are used to predict their locations and orientations in the next height map H_{i+2}.

3) Integrate the results of the matching into a local map and update region labels. The local map builder overlays all regions in H_{i+1} onto the local map L_i according to their motion parameters. We call the overlaid map a new local map L_{i+1}. For each point on the local map L_{i+1}, the local map builder refines the height information for each point using those in H_{i+1} and L_i. If the height information is available from both, take a mean of both heights with weights (each weight is inversely proportional to the distance from the viewer to that point). When only one height is available, adopt that height. Else, retain that point left undetermined. After integration, obstacle finder segments the integrated local map L_{i+1} into regions using the same method in Section III-C. and assigns a new label for each region. The segmented local map L_{i+1} is used for matching with next height map H_{i+2}. Thus, the local map is constructed by matching and integrating height maps observed at different locations.

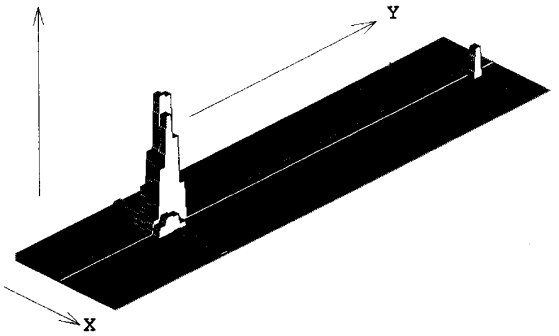


Fig. 12. Histogram of motion parameters.

Figs. 13 show the local map L_5 integrated with five height maps (from the first to the fifth height maps). Fig. 13(b) shows the roof of the cabin on the left side in the first height map H_1 (left) and the local map L_5 (right) from the different view angle from Fig. 13(a). In H_1, the roof is worm-eaten and includes noisy heights due to the shallow angle between the light plane from range scanner and the roof plane. While, in L_5, the shape of the roof is clear because the vehicle approached to the cabin, therefore, better height information was obtained. Fig. 13(c) shows the shape difference of moving object (truck) between H_5 (left) and L_5 (right). While it is worm-eaten in H_5 because of long distance from the viewer, the region corresponding to the bed of the truck is clear in L_5 where

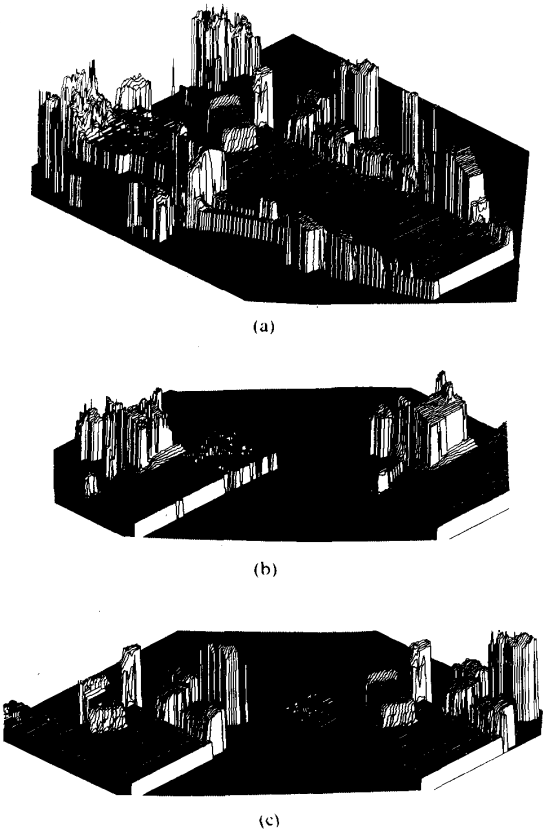


Fig. 13. Local map integration with five height maps. (a) Local map. (b) Roof of the cabin on the left side in the first height map (left) and the local map (right). (c) Shape difference of moving object (truck) between the fifth height man (left) and the local man (right).

the system successfully tracks the moving object and updates its height information.

IV. Conclusion

A map building system for a mobile robot from sensory data has been described. We introduced the virtual sensor map into the 3-D world model construction in order to represent various sensory data such as range and video data and in order to integrate them in a Cartesian coordinate system. We proposed the use of a height map, one example of virtual sensor maps, obtained from the range image for a mobile robot to support various tasks such as path planning and landmark recognition. The height map is easy to recover and calculation of geometrical properties such as slope and curvature is straightforward.

In the experiments, we used the range finder based on the structured lighting method that produces dense range images consisting of 512×512 image points. The largest difference between the range image used and actual one such as an ERIM range image [15] size of which is 256×64 is the resolution of the range image. Since the size of the height map used is smaller than that of the range image, we do not need the sophisticated technique such as the locus algorithm which can produce the dense height map from the sparse range image [18]. In our case, the edge-preserved smoothing on the height map can provide us with the sufficient height information for our task. In spite of the above difference, the proposed idea for the world model representation can be applicable to both types of the range images given the geometrical relation between the range finder, the TV camera, and the vehicle.

The current obstacle classifier uses principally the geometrical properties of height information and makes little use of brightness information in identifying the obstacles as artificial or natural objects. The experimental results include some number of errors. Errors due to vertical planes could be corrected by verifying the existence of the vertical plane in the multiple height maps and by obtaining more correct parameters for them from the range image (physical sensor map). Other errors might be corrected by the use of color images since color information is often useful for segmentation of images of outdoor scene [24].

In our experiments, we have dealt with almost flat terrains. In the case of non-flat terrain, the matching process would be more complicated because the orientation of the H-axes in a height map and a local map might be different. One alternative is to adopt the direction of the gravity as the common orientation of the H-axes in the different virtual sensor maps utilizing the output from the orientation sensor in the navigation system. This could make the matching process as simple as in the case of flat terrain.

In this paper, we have not dealt with the problem of constructing a global map with a set of lcoal maps. The method of local map builder that numerically integrates virtual sensor maps using height information could not be applicable to the construction of the global map because a traveling of long distance often results in a significant amount of positional error in the global map and because the robot cannot correct this error unless landmarks in the environment are given in advance [25]. Adada, Fukui, and Tsuji [26] developed a method of representing a global map with relational local maps where the rough description of the geometrical relation between adjacent local maps is described. We could apply their method to our scene. However, as the map scale becomes larger, the higher level descriptions are generally required to represent a world model instead of numerical representation of it. Our current system has dealt with only region labels (unexplored, occluded, traversable and obstacle) and sublabels (artificial or natural). In order to accomplish higher tasks such as landmark and/or object recognition, knowledge base of the objects expected in the scene and geometrical modeling for them are needed, and the control structure for them should be exploited. These problems are under investigation.

ACKNOWLEDGMENT

The author wishes to thank Prof. Azriel Rosenfeld and Prof. Larry S. Davis for helpful comments and discussions and Mr. Daniel DeMenthon for providing range images at the University of Maryland, and Prof. Saburo Tsuji and Prof. Yoshiaki Shirai for constructive discussions at Osaka University, Japan. He also wishes to thank his son, Ryu Asada, who helped him assemble and paint the landscape models on the simulation board.

REFERENCES

[1] A. M. Waxman, J. Le Moigne, and B. Srinivasan, "Visual navigation of roadways," in *Proc. IEEE Int. Conf. Robotics Automation*, 1985, pp. 862–867.

[2] S. Tsuji, Y. Yagi, and M. Asada, "Dynamic scene analysis for a mobile robot in man-made environment," in *Proc. IEEE Int. Conf. Robotics Automation*, 1985, pp. 850–855.

[3] S. Tsuji, J. Y. Zheng, and M. Asada, "Stereo vision of a mobile robot: world constraints for image matching and interpretation," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1986, pp. 1594–1599.

[4] A. M. Waxman, J. LeMoigne, L. S. Davis, E. Liang, and T. Siddalingaiah, "A visual navigation system," in *Proc. IEEE Int. Conf. Robotics Automation*, 1986, pp. 1600–1606.

[5] R. Wallace, K. Matsuzaki, Y. Goto, J. Crisman, J. Webb, and T. Kanade, "Progress in robot road-following," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1986, pp. 1615–1621.

[6] S. A. Shafer, A. Stentz, and C. Thorpe, "An architecture for sensor fusion in a mobile robot," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1986, pp. 2002–2011.

[7] C. Thorpe, S. Shafer, T. Kanade, *et al.*, "Vision and navigation for the Carnegie Mellon Navlab," in *Proc. DARPA Image Understanding Workshop*, 1987, pp. 143–152.

[8] L. S. Davis, D. DeMenthon, R. Gajulapalli, T. R. Kushner, J. Le Moigne, and P. Veatch, "Vision-based navigation: a status report," in *Proc. DARPA Image Understanding Workshop*, 1987, pp. 153–169.

[9] Y. Goto and A. Stentz, "The CMU system for mobile robot navigation," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1987, pp. 99–105.

[10] R. A. Brooks, "A hardware retargetable distributed layered architecture for mobile robot control," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1987, pp. 106–110.

[11] R. S. Wallace, "Robot road following by adaptive color classification and shape trucking," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1987, pp. 258–263.

[12] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "Video road-following for the autonomous land vehicle," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1987, pp. 273–280.

[13] S. Tsuji, "Monitoring of a building environment by a mobile robot," in *Proc. 2nd Int. Symp. Robotics Res.*, 1985, pp. 349–365.

[14] D. Lawton, T. S. Levvit, C. C. McConnell, P. C. Nelson, and J. Glicksman, "Environmental modeling and recognition for an autonomous land vehicle," in *Proc. DARPA Image Understanding Workshop*, 1987, pp. 107–121.

[15] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-10, pp. 362–373, 1988.

[16] S. Tsuji and J. Y. Zheng, "Visual path planning by a mobile robot," in *Proc. 10th Int. Joint Conf. Artificial Intell.*, 1987, pp. 1127–1130.

[17] A. Elfes, "A sonar-based real world mapping and navigation," *IEEE J. Robotics Automat.*, vol. RA-3, 1987, pp. 249–265.

[18] M. Hebert, T. Kanada, and I. Kweon, "3-D techniques for autonomous vehicles," *Tech. Rep.* CMU-RI-TR-88-12, Robotics Institute, Carnegie-Mellon University, 1988.

[19] M. Daily *et al.*, "Autonomous cross-country navigation with ALV," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1988. pp. 718–726.

[20] D. DeMenthon, T. Siddalingaiah, and L. S. Davis, "Production of dense range images with the CVL light-stripe range scanner," *Center for Automation Res. Tech. Rep.* CAR-TR-337, University of Maryland, College Park, MD, 1987.

[21] M. Asada, "Building 3-D world model for a mobile robot from sensory data," *Center for Automation Res. Tech. Rep.* CAR-TR-332, CS-TR-1936, University of Maryland, College Park, MD, 1987.

[22] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Computer Graphics Image Processing*, vol. 9, pp. 394–407, 1979.

[23] S. Puri and L. S. Davis, "Two dimensional path planning with obstacles and shadows," *Center for Automation Res. Tech. Rep.* CAR-TR-255, CS-TR-1760, University of Maryland, College Park, MD, Jan. 1987.

[24] Y. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Computer Graphics Image Processing*, vol. 13, pp. 222–241, 1980.

[25] R. A. Brooks, "Visual map making for a mobile robot," in *Readings in Computer Vision*, M. A. Fischler and O. Firscein, Eds. Los Altos, CA: Morgan Kaufman, pp. 438–443.

[26] M. Asada, Y. Fukui, and S. Tsuji, "Representing a global world of a mobile robot with relational local maps," in *Proc. 1988 Int. Workshop Intelligent Robots and Systems*, 1988, pp. 199–204.

Minoru Asada (M'88) was born in Shiga, Japan, on October 1, 1953. He received the B.E., M.Sc., and Ph.D., degrees in control engineering from Osaka University, Osaka, Japan, in 1977, 1979, and 1982, respectively.

From 1982 to 1988, he was an Assistant Professor of Control Engineering, Osaka University, Toyonaka, Osaka, Japan. Since April 1989, he has been an Associate Professor of Mechanical Engineering for Computer-Controlled Machinery, Osaka University, Suita, Osaka, Japan. From August 1986 to October 1987, he was a visiting researcher of Center for Automation Research, University of Maryland, College Park, MD.

Dr. M. Asada is a member of the Institute of Electronics, Information and Computer Engineering (Japan), the Information Processing Society of Japan, and the Robotics Society of Japan. His current research includes Computer Vision, Artificial Intelligence, and Robotics.