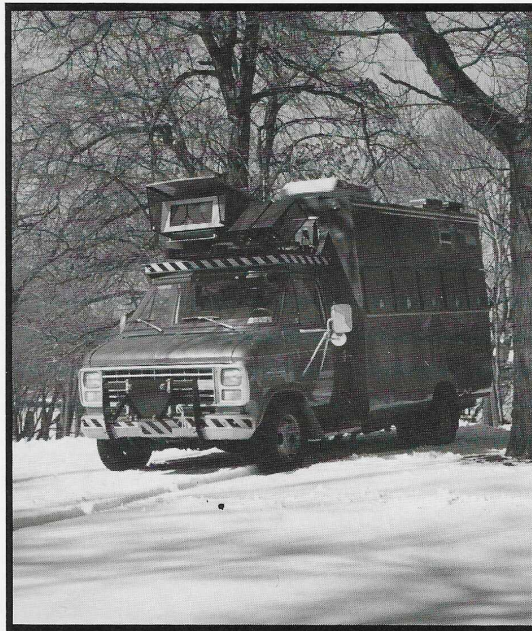

Vision and Navigation

The Carnegie Mellon Navlab

**edited by
Charles E. Thorpe**

foreword by Takeo Kanade



Kluwer Academic Publishers

VISION AND NAVIGATION

The Carnegie Mellon Navlab

edited by

Charles E. Thorpe
Carnegie Mellon University

with a foreword by

Takeo Kanade



KLUWER ACADEMIC PUBLISHERS
Boston/Dordrecht/London

Chapter 8: 3-D Vision Techniques for Autonomous Vehicles

Martial Hebert, InSo Kweon, and Takeo Kanade

8.1 Introduction

There are many sensing strategies for perception for mobile robots, including single camera systems, sonars, passive stereo, and laser range finders. Here, we focus on perception algorithms for range sensors that provide 3-D data directly by active sensing. Using such sensors has the advantage of eliminating the calibration problems and computational costs inherent in passive techniques such as stereo. This Chapter describes some of the techniques we have developed in this area of research. The aim of our work is to produce models of the environment, which we call the *terrain*, for path planning and object recognition.

¹

We first describe the range sensor that we used in this work in Section 8.2. Even though we tested our algorithm on one specific range sensor, we believe that the sensor characteristics of Section 8.2 are fairly typical of a wide range of sensors [4]. The algorithms for building a terrain representation from a single sensor frame are discussed in Section 8.3 in which we introduce the concept of dividing the terrain representation algorithms into three levels depending on the sophistication of the path planner that would use the representation, and on the anticipated difficulty of the terrain. Since a mobile robot is by definition a dynamic system, it must process not one, but many observations along the course of its trajectory. The 3-D vision algorithms must therefore be able to reason about representations that are built from sensory data taken from different locations. We investigate this type of algorithm in Section 8.4 in which we propose algorithms for matching and merging multiple terrain representations. Finally, the 3-D vision algorithms that we propose are not meant to be used in isolation, but have to be eventually integrated in a system that includes other sensors. A typical example is the case of road following in which color cameras can track the road, while a range sensor can detect unexpected obstacles. Another example is a mission in which a scene must be interpreted in order to identify specific objects, in which case all the available sensors must contribute to the final scene analysis. We propose some algorithms for fusing 3-D representations with representations obtained from a color camera in Section 8.5. We also

¹An earlier version of this paper appeared as part of CMU Robotics Institute report CMU-RI-TR-89-5. Reprinted by permission.

describe the application of this sensor fusion to a simple natural scene analysis program. Perception techniques for mobile robots have to be eventually validated by using real robots in real environments. We have implemented the 3-D vision techniques presented in this report on both the Terregator and the Navlab. The results presented in Sections 8.3.3 and 8.3.4 come from the 3-D vision module that we integrated in the Navlab system [40]. In addition, some of the techniques were used for the Ambler [2], a hexapod designed for planetary exploration. This vehicle is designed for navigation on very rugged terrain including high slopes, rocks, and wide gullies.

8.2 Active range and reflectance sensing

The basic principle of active sensing techniques is to observe the reflection of a reference signal (sonar, laser, radar, etc.) produced by an object in the environment in order to compute the distance between the sensor and that object. In addition to the distance, the sensor may report the intensity of the reflected signal which is related to physical surface properties of the object. In accordance with tradition, we will refer to this type of intensity data as "reflectance" data even though the quantity measured is not the actual reflectance coefficient of the surface.

Active sensors are attractive to mobile robot researchers for two main reasons: first, they provide range data without the computation overhead associated with conventional passive techniques such as stereo vision, which is important in time critical applications such as obstacle detection. Second, it is largely insensitive to outside illumination conditions, considerably simplifying the image analysis problem. This is especially important for images of outdoor scenes in which illumination cannot be controlled or predicted. For example, the active reflectance images of outside scenes do not contain any shadows from the sun. In addition, active range finding technology has developed to the extent that makes it realistic to consider it as part of practical mobile robot implementations in the short term [4].

The range sensor we used is a time-of-flight laser range finder developed by the Environmental Research Institute of Michigan (ERIM). The basic principle of the sensor is to measure the difference of phase between a laser beam and its reflection from the scene [44]. A two-mirror scanning system allows the beam to be directed anywhere within a $30^\circ \times 80^\circ$ field of view. The data produced by the ERIM sensor is a 64×256 range image. Range is coded in eight bits from zero to 64 feet, which corresponds to a range resolution of three inches. All measurements are relative since the sensor measures differences of

phase. That is, a range value is known *modulo* 64 feet. We have adjusted the sensor so that the range value 0 corresponds to the mirrors. In addition to range images, the sensor also produces active reflectance images of the same format ($64 \times 256 \times 8$ bits). The reflectance at each pixel encodes the energy of the reflected laser beam at each point. Figure 8.2 shows a pair of range and reflectance images of an outdoor scene. The next two Sections describe the range and reflectance data in more details.

8.2.1 From range pixels to points in space

The position of a point in a given coordinate system can be derived from the measured range and the direction of the beam at that point. We usually use the Cartesian coordinate system shown in Figure 8.1, in which case the coordinates of a point measured by the range sensor are given by the equations:

$$\begin{aligned} x &= D \sin \theta \\ y &= D \cos \phi \cos \theta \\ z &= D \sin \phi \cos \theta \end{aligned} \quad (8.1)$$

where ϕ and θ are the vertical and horizontal scanning angles of the beam direction. Note that the reference coordinate system is not the same as in [19] for consistency reasons. The two angles are derived from the row and column position in the range image, (r, c) , by the equations:

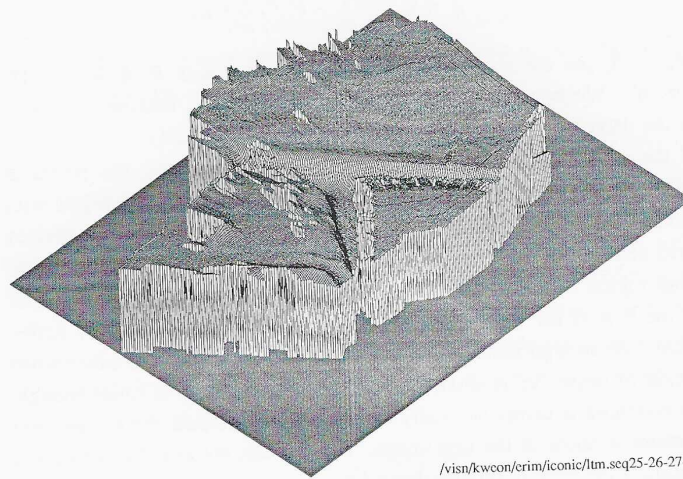
$$\begin{aligned} \theta &= \theta_0 + c \times \Delta\theta \\ \phi &= \phi_0 + r \times \Delta\phi \end{aligned} \quad (8.2)$$

where θ_0 (respectively ϕ_0) is the starting horizontal (respectively vertical) scanning angle, and $\Delta\theta$ (respectively $\Delta\phi$) is the angular step between two consecutive columns (respectively rows). Figure 8.3 shows an overhead view of the scene of Figure 8.2, in which the coordinates of the points are computed using Eq. (8.3).

8.2.2 Reflectance images

A reflectance image from the ERIM sensor is an image of the energy reflected by a laser beam. Unlike conventional intensity images, this data provides us with information which is to a large extent independent of the environmental illumination. In particular, the reflectance images contain no shadows from outside illumination. The measured energy does depend, however, on the shape

therefore important to show that T_0 is close to the minimum. Figure 8.29 plots the value of the ν_i 's with respect to the number of iterations. These curves show that E converges in a smooth fashion. The coefficient k that controls the rate of convergence is very conservative in this case in order to avoid oscillations about the minimum.



/visn/kwcon/crim/iconic/ltm.seq25-26-27-28.gif(view1)

Figure 8.28: Combining four maps by the iconic matching algorithm

8.5 Combining range and intensity data

In the previous Section we have concentrated on the use of 3-D vision as it relates solely to the navigation capabilities of mobile robots. Geometric accuracy was

Figure 8.29 plots
curves show
the rate
oscillations

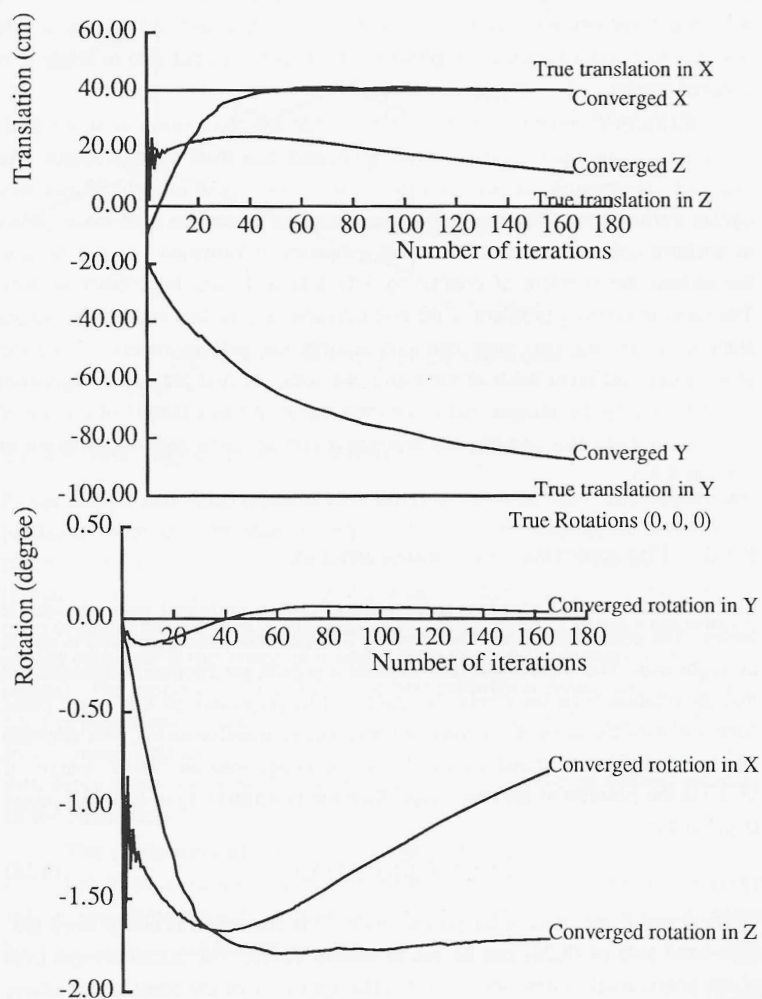


Figure 8.29: Convergence rate of the matching algorithm

the deciding factor in the choice of representations and algorithms while we gave very little attention to the extraction of semantic information. A mobile robot needs more than just navigation capabilities, however, since it also must be able to extract semantic descriptions from its sensors. For example, we will describe a landmark recognition algorithm in Section 5. In that case, the system is able not only to build a geometric representation of an object but also to relate it to a stored model.

Extracting semantic information for landmark recognition or scene analysis may require much more than just geometric data from a range sensor. For example, interpreting surface markings is the only way to unambiguously recognize traffic signs. Conversely, the recognition of a complex man-made object of uniform color is easiest when using geometric information. In this Section we address the problem of combining 3-D data with data from other sensors. The most interesting problem is the combination of 3-D data with color images since these are the two most common sensors for outdoor robots. Since the sensors have different fields of view and positions, we first present an algorithm for transforming the images into a common frame. As an example of the use of combined range/color images, we describe a simple scene analysis program in Section 8.5.3.

8.5.1 The geometry of video cameras

The video camera is a standard color vidicon camera equipped with wide-angle lenses. The color images are 480 rows by 512 columns, and each band is coded on eight bits. The wide-angle lens induces a significant geometric distortion in that the relation between a point in space and its projection on the image plane does not obey the laws of the standard perspective transformation. We alleviate this problem by first transforming the actual image into an "ideal" image: if (R, C) is the position in the real image, then the position (r, c) in the ideal image is given by:

$$r = f_r(R, C), c = f_c(R, C) \quad (8.26)$$

where f_r and f_c are third order polynomials. This correction is cheap since the right-hand side of (8.26) can be put in lookup tables. The actual computation of the polynomial is described in [30]. The geometry of the ideal image obeys the laws of the perspective projection in that if $P = [x, y, z]^t$ is a point in space, and (r, c) is its projection in the ideal image plane, then:

$$r = fx/z, c = fy/z \quad (8.27)$$

we gave
mobile robot
st be able
l describe
m is able
elate it to

ene anal-
nsor. For
ously rec-
de object
s Section
r sensors.
or images
Since the
algorithm
the use of
rogram in

wide-angle
f is coded
tortion in
age plane
alleviate
image: if
real image

(8.26)

since the
mputation
age obeys
in space,

(8.27)

where f is the focal length. In the rest of the paper, row and column positions will always refer to the positions in the ideal image, so that perspective geometry is always assumed.

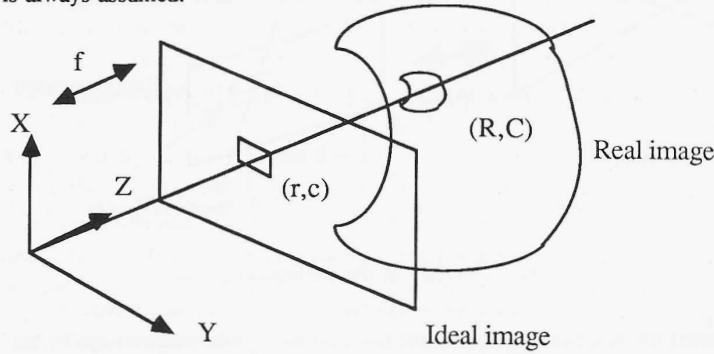


Figure 8.30: Geometry of the video camera

8.5.2 The registration problem

Range sensors and video cameras have different fields of view, orientations, and positions. In order to be able to merge data from both sensors, we first have to estimate their relative positions, known as the calibration or registration problem (Figure 8.31). We approach the problem as a minimization problem in which pairs of pixels are selected in the range and video images. The pairs are selected so that each pair is the image of a single point in space as viewed from the two sensors. The problem is then to find the best calibration parameters given these pairs of points and is further divided into two steps: we first use a simple linear least-squares approach to find a rough initial estimate of the parameters, and then apply a non-linear minimization algorithm to compute an optimal estimate of the parameters.

The calibration problem as a minimization problem.

Let P_i be a point in space, with coordinates P_i^r with respect to the range sensor, and coordinates P_i^c with respect to the video camera. The relationship between the two coordinates is:

$$P_i^c = RP_i^r - T \quad (8.28)$$

where R is a rotation matrix, and T is a translation vector. R is a non-linear function of the orientation angles of the camera: pan (α), tilt (β), and rotation (γ). P_i^r can be computed from a pixel location in the range image. P_i^c is not

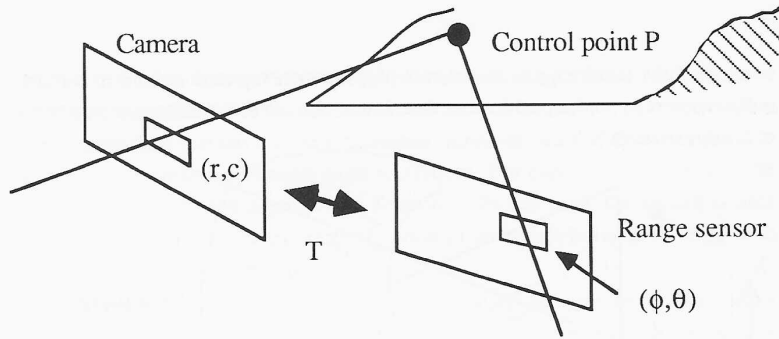


Figure 8.31: Geometry of the calibration problem

completely known, but is related to the pixel position in the video image by the perspective transformation:

$$z_i^c r_i = f x_i^c \quad (8.29)$$

$$z_i^c c_i = f y_i^c \quad (8.30)$$

where f is the focal length. Substituting (8.28) into (8.29) and (8.30) we get:

$$R_x P_i^c - T_x r_i - f R_x P_i^c + T'_x = 0 \quad (8.31)$$

$$R_z P_i^c - T_z c_i - f R_y P_i^c + T'_y = 0 \quad (8.32)$$

where R_x , R_y , and R_z are the row vectors of the rotation matrix R , and $T'_y = f T_y$, $T'_x = f T_x$.

We are now ready to reduce the calibration problem to a least-squares minimization problem. Given n points P_i , we want to find the transformation (R, T) that minimizes the left-hand sides of equations (8.31) and (8.32). We first estimate T by a linear least-squares algorithm, and then compute the optimal estimate of all the parameters.

Initial estimation of camera position.

Assuming that we have an estimate of the orientation R , we want to estimate the corresponding T . The initial value of R can be obtained by physical measurements using inclinometers. Under these conditions, the criterion to be minimized is:

$$C = \sum_{i=1}^n [(A_i - T_x B_i - f C_i + T'_x)^2 + (D_i - T_z E_i - f F_i + T'_y)^2] \quad (8.33)$$

where $A_i = R_z P_i^t r_i$, $B_i = r_i$, $C_i = R_x P_i^t$, $D_i = R_z P_i^t c_i$, $E_i = c_i$, and $F_i = R_y P_i^t$ are known and T_x, T_y, T_z, f are the unknowns.

Equation (8.33) can be put in matrix form:

$$C = \|U - AV\|^2 - \|W - BV\|^2 \quad (8.34)$$

where $V = [T_x, T_y, T_z, f]^t$, $U = [A_1, \dots, A_n]^t$, $W = [D_1, \dots, D_n]^t$,
 $A = \begin{bmatrix} B_1 & 0 & -1 & C_1 \\ \vdots & \vdots & \vdots & \vdots \\ B_n & 0 & -1 & C_n \end{bmatrix}$, and $B = \begin{bmatrix} E_1 & -1 & 0 & F_1 \\ \vdots & \vdots & \vdots & \vdots \\ E_n & -1 & 0 & F_n \end{bmatrix}$. The minimum for the criterion of Equation (8.34) is attained at the parameter vector:

$$V = (A^t A + B^t B)^{-1} (A^t U + B^t W) \quad (8.35)$$

Optimal estimation of the calibration parameters.

Once we have computed the initial estimate of V , we have to compute a more accurate estimate of (R, T) . Since R is a function of (α, β, γ) , we can transform the criterion from equation (8.33) into the form:

$$C = \sum_{i=1}^n \|I_i - H_i(S)\|^2 \quad (8.36)$$

where I_i is the 2-vector representing the pixel position in the video image, $I_i = [r_i, c_i]^t$, and S is the full vector of parameters, $S = [T_x, T_y, T_z, f, \alpha, \beta, \gamma]^t$. We cannot directly compute C_{min} since the functions H_i are non-linear; instead we linearize C by using the first order approximation of H_i [26]:

$$C \approx \sum_{i=1}^n \|I_i - H_i(S_0) - J_i \Delta S\|^2 \quad (8.37)$$

where J_i is the Jacobian of H_i with respect to S , S_0 is the current estimate of the parameter vector, and $\Delta S = S - S_0$. The right-hand side of (8.37) is minimized when its derivative with respect to ΔS vanishes, that is:

$$\sum_{i=1}^n J_i^t J_i \Delta S + J_i^t \Delta C_i = 0 \quad (8.38)$$

where $\Delta C_i = I_i - H_i(S_0)$. Therefore, the best parameter vector for the linearized criterion is:

$$\Delta S = - \sum_{i=1}^n (J_i^t J_i)^{-1} J_i^t \Delta C_i \quad (8.39)$$

Equation (8.39) is iterated until there is no change in S . At each iteration, the estimate S_0 is updated by: $S_0 \leftarrow S_0 + \Delta S$.

Implementation and performance.

The implementation of the calibration procedure follows the steps described above. Pairs of corresponding points are selected in a sequence of video and range images. We typically use twenty pairs of points carefully selected at interesting locations in the image (*e.g.* corners). An initial estimate of the camera orientation is $(0, \beta, 0)$, where β is physically measured using an inclinometer. The final estimate of S is usually obtained after less than ten iterations. This calibration procedure has to be applied only once, as long as the sensors are not displaced.

Once we have computed the calibration parameters, we can merge range and video images into a colored-range image. Instead of having one single fusion program, we implemented this as a library of fusion functions that can be divided in two categories:

1. Range $\rightarrow$ video: This set of functions takes a pixel or a set of pixels (r^e, c^e) in the range image and computes the location (r^v, c^v) in the video image. This is implemented by directly applying Equations (8.31) and (8.32).
2. Video $\rightarrow$ range: This set of functions takes a pixel or a set of pixels (r^v, c^v) in the video image and computes the location (r^e, c^e) in the range image. The computed location can be used in turn to compute the location of a intensity pixel in 3-D space by directly applying Equation (8.3). The algorithm for this second set of functions is more involved because a pixel in the video image corresponds to a line in space (Figure 8.30) so that Equations (8.31) and (8.32) cannot be applied directly. More precisely, a pixel (r^v, c^v) corresponds, after transformation by (R, T) , to a curve C in the range image. C intersects the image at locations (r^e, c^e) , where the algorithm reports the location (r^e, c^e) that is the minimum among all the range image pixels that lie on C of the distance between (r^e, c^e) and the projection of (r^v, c^v) in the video image (using the first set of functions). The algorithm is summarized in Figure 8.32.

Figure 8.33 shows the colored-range image of a scene of stairs and sidewalks. The image is obtained by mapping the intensity values from the color image onto the range image. Figure 8.34 shows a perspective view of the colored-range image. In this example [16], we first compute the location of each range pixel (r^e, c^e) in the video image, and then assign the color value to the 64×256 colored-range image. The final display is obtained by rotating the range pixels, the coordinates of which are computed using Equation (8.3).

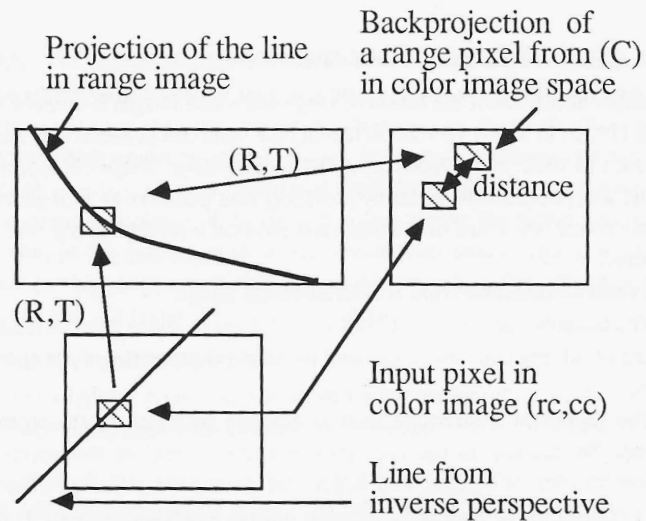
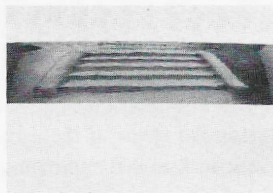
Figure 8.32: Geometry of the "video $\rightarrow$ range" transformation

Figure 8.33: Colored-range image of stairs

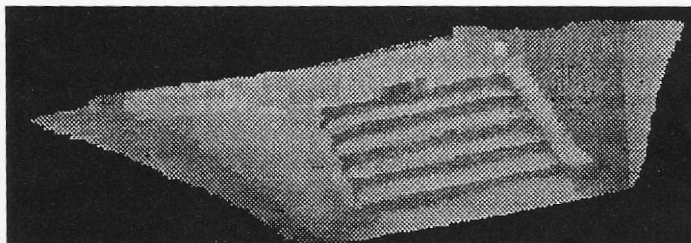


Figure 8.34: Perspective view of registered range and color images

8.5.3 Application to outdoor scene analysis

An example of the use of the fusion of range and video images is outdoor scene analysis [19,25] in which we want to identify the main components of an outdoor scene, such as trees, roads, grass, etc. The colored-range image concept makes the scene analysis problem easier by providing data pertinent to both geometric information (*e.g.* the shape of the trees) and physical information (*e.g.* the color of the road).

Feature extraction from a colored-range image.

The features that we extract from a colored-range image must be related to two types of information: the shapes and the physical properties of the observed surfaces.

The geometric features are used to describe the shape of the objects in the scene. We propose to use two types of features: regions that correspond to smooth patches of surface, and edges that correspond either to transitions between regions, or to transitions between objects (occluding edges). Furthermore, we must be able to describe the features in a compact way. One common approach is to describe the regions as quadric patches, and the edges as sets of tri-dimensional line segments. More sophisticated descriptions are possible [5], such as bicubic patches or curvature descriptors. We use simpler descriptors since the range data is relatively low resolution, and we do not have the type of accurate geometric model that is suited for using higher order geometric descriptors. The descriptors attached to each geometric feature are:

- The parameters describing the shape of the surface patches. That is the parameters of the quadric surface that approximate each surface patch.
- The shape parameters of the surface patches such as center, area, and elongations.
- The 3-D polygonal description of the edges.
- The 3-D edge types: convex, concave, or occluding.

The surface patches are extracted by fitting a quadric of equation $X^TAX + B^TX + C = 0$ to the observed surfaces, where X is the Cartesian coordinate vector computed from a pixel in the range image. The fitting error,

$$E(A, B, C) = \sum_{X_i \in \text{patch}} [X_i^TAX_i + B^TX_i + C]^2 \quad (8.40)$$

is used to control the growing of regions over the observed surfaces. The parameters A, B, C are computed by minimizing $E(A, B, C)$ as in [14].

The features related to physical properties are regions of homogeneous color in the video image, that is regions within which the color values vary

smoothly. The choice of these features is motivated by the fact that an homogeneous region is presumably part of a single scene component, although the converse is not true as in the case of the shadows cast by an object on an homogeneous patch on the ground. The color homogeneity criterion we use is the distance $(X - m)^t \Sigma^{-1} (X - m)$ where m is the average mean value on the region, Σ is the covariance matrix of the color distribution over the region, and X is the color value of the current pixel in (*red, green, blue*) space. This is a standard approach to color image segmentation and pattern recognition. The descriptive parameters that are retained for each region are:

- The color statistics (m, Σ).
- The polygonal representation of the region border.
- Shape parameters such as center or moments.

The range and color features may overlap or disagree. For example, the shadow cast by an object on a flat patch of ground would divide one surface patch into two color regions. It is therefore necessary to have a cross-referencing mechanism between the two groups of features. This mechanism provides a two-way direct access to the geometric features that intersect color features. Extracting the relations between geometric and physical features is straightforward since all the features are registered in the colored-range image.

An additional piece of knowledge that is important for scene interpretation is the spatial relationships between features. For example, the fact that a vertical object is connected to a large flat plane through a concave edge may add evidence to the hypothesis that this object is a tree. As in this example, we use three types of relational data:

- The list of features connected to each geometric or color feature.
- The type of connection between two features (convex/concave/occluding) extracted from the range data.
- The length and strength of the connection. This last item is added to avoid situations in which two very close regions become accidentally connected along a small edge.

Scene interpretation from the colored-range image.

Interpreting a scene requires the recognition of the main components of the scene such as trees or roads. Since we are dealing with natural scenes, we cannot use the type of geometric matching that is used in the context of industrial parts recognition [5]. For example, we cannot assume that a given object has specific quadric parameters. Instead, we have to rely on "fuzzier" evidence such as the verticality of some objects or the flatness of others. We

therefore implemented the object models as sets of properties that translate into constraints on the surfaces, edges, and regions found in the image. For example, the description encodes four such properties:

- $P1$: The color of the trunk lies within a specific range $\Rightarrow$ constraint on the statistics (m, Σ) of a color region.
- $P2$: The shape of the trunk is roughly cylindrical $\Rightarrow$ constraint on the distribution of the principal values of the matrix A of the quadric approximation.
- $P3$: The trunk is connected to a flat region by a concave edge $\Rightarrow$ constraint on the neighbors of the surface, and the type of the connecting edge.
- $P4$: The tree has two parallel vertical occluding edges $\Rightarrow$ constraint on the 3-D edges description.

Other objects such as roads or grass areas have similar descriptions. The properties P_{ij} of the known object models M_j are evaluated on all the features F_k extracted from the colored-range image. The result of the evaluation is a score S_{ijk} for each pair (P_{ij}, F_k) . We cannot rely on individual scores since some may not be satisfied because of other objects, or because of segmentation problems. In the tree trunk example, one of the lateral occluding edges may itself be occluded by some other object, in which case the score for $P4$ would be low while the score for the other properties would still be high. In order to circumvent this problem, we first sort the possible interpretations M_j for a given feature F_k according to all the scores $(S_{ij})_i$. In doing this, we ensure that all the properties contribute to the final interpretation and that no interpretations are discarded at this stage while identifying the most plausible interpretations.

We have so far extracted plausible interpretations only for individual scene features F_k . The final stage in the scene interpretation is to find the interpretations (M_{j_k}, F_k) that are globally consistent. For example, property $P3$ for the tree implies a constraint on a neighboring region, namely that this has to be a flat ground region. Formally, a set of consistency constraints C_{mn} is associated with each pair of objects (M_m, M_n) . The C_{mn} constraints are propagated through the individual interpretations (M_{j_k}, F_k) by using the connectivity information stored in the colored-range feature description. The propagation is simple considering the small number of features remaining at this stage.

The final result is a consistent set of interpretations of the scene features, and a grouping of the features into sets that correspond to the same object. The last result is a by-product of the consistency check and the use of connectivity data. Figure 8.35 shows the color and range images of a scene which contains

at translate into
ge. For example,

⇒ constraint on

constraint on the
quadratic approx-

e edge ⇒ con-
the connecting

⇒ constraint on

descriptions. The
all the features
evaluation is a
ual scores since
of segmentation
ding edges may
re for P_4 would
high. In order to
is M_j for a given
e ensure that all
interpretations are
interpretations.

individual scene
the interpretations
 P_3 for the tree
has to be a flat
s associated with
ated through the
formation stored
mple considering

e scene features,
same object. The
e of connectivity
e which contains

a road, a couple of trees, and a garbage can. Figure 8.36 shows a display of the corresponding colored-range image in which the white pixels are the points in the range image that have been mapped into the video image. This set of points is actually sparse because of the difference in resolutions between the two sensors, and some interpolation was performed to produce the dense regions of Figure 8.36.

Only a portion of the image is registered due to the difference in field of view between the two sensors (60° for the camera versus 30° in the vertical direction for the range sensor). Figure 8.37 shows a portion of the image in which the edge points from the range image are projected on the color image. The edges are interpreted as the side edges of the tree and the connection between the ground and the tree. Figure 8.38 shows the final scene interpretation. The white dots are the main edges found in the range image. The power of the colored-range image approach is demonstrated by the way the road is extracted. The road in this image is separated into many pieces by strong shadows. Even though the shadows do not satisfy the color constraint on road region, they do perform well on the shape criterion (flatness), and on the consistency criteria (both with the other road regions, and with the trees). The shadows are therefore interpreted as road regions and merge with the other regions into one road region. This type of reasoning is in general difficult to apply when only video data is used unless one uses stronger models of the objects such as an explicit model of a shadowed road region. Using the colored-range image also makes the consistency propagation a much easier task than in purely color-based scene interpretation programs [31].

8.6 Conclusion

We have described techniques for building and manipulating 3-D terrain representations from range images. We have demonstrated these techniques on real images of outdoor scenes. Some of them (Sections 8.3.3, 8.3.4, and 8.4.2) were integrated in a large mobile robot system that was successfully tested in the field. We expect that the module that manipulates and creates these terrain representations will become part of the standard core system of our outdoor mobile robots, just as a local path planner or a low-level vehicle controller are standard modules of a mobile robot system independent of its application. This work will begin by combining the polygonal terrain representation of Section 8.3.4 with the path planner of [36] in order to generate the basic capabilities for an off-road vehicle.

Many issues still remain to be investigated. First of all, we must define a uniform way of representing and combining the uncertainties in the terrain maps.

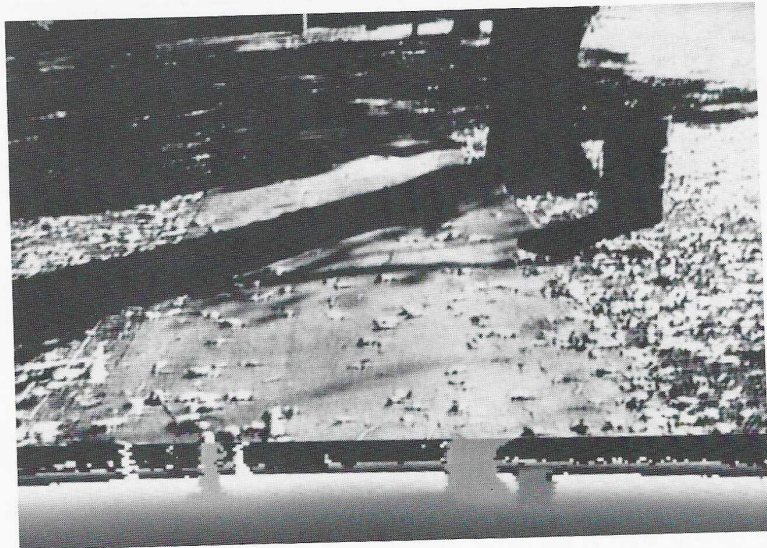


Figure 8.35: Color and range images of an outdoor scene

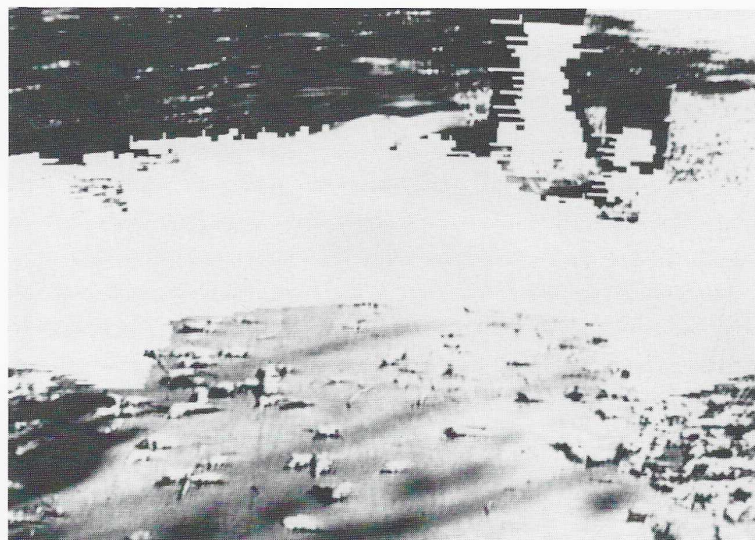


Figure 8.36: A view of the corresponding colored-range image

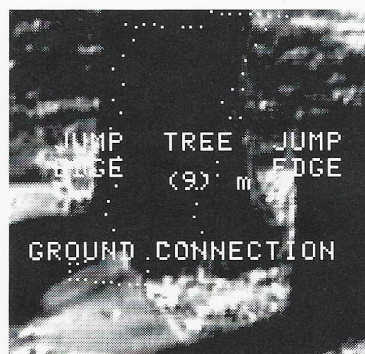


Figure 8.37: Edge features from the colored-range image

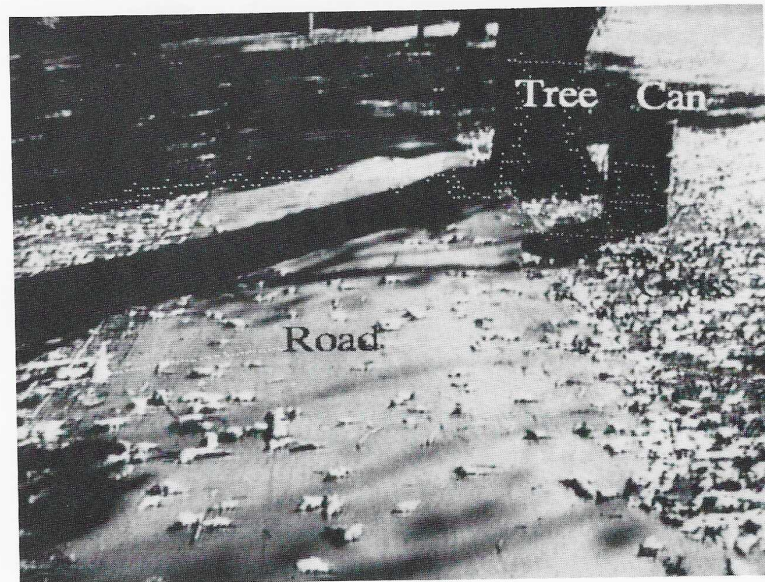


Figure 8.38: Final scene interpretation

Currently, the uncertainty models depend heavily on the type of sensor used and on the level at which the terrain is represented. Furthermore, the displacements between terrain maps are known only up to a certain level of uncertainty. This level of uncertainty must be evaluated and updated through the matching of maps, whether iconic or feature-based. Regarding the combination of the 3-D representations with representations from other sensors, we need to define an algorithm for sensor registration that is general enough for application to a variety of situations. The algorithms presented in Section 8.5 are still very dependent on the sensors that we used, and on the intended application. Registration schemes such as [17] would enable us to have a more uniform approach to the problem. An added effect of using such a registration algorithm is that we could explicitly represent errors caused by the combination of the sensors, which we did not do in Section 8.5. Another issue concerns our presentation of the three levels of terrain representation, the matching algorithms, and the sensor combination algorithms as separate problems. We should define a common perceptual architecture to integrate these algorithms in a common representation that can be part of the core system of a mobile robot. Finally, we have tackled the terrain representation problems mainly from a geometrical point of view. Except in Section 8.5, we did not attempt to extract semantic interpretations from the representations. A natural extension of this work is to use the 3-D terrain representations to identify known objects in the scene. Another application along these lines is to use the terrain maps to identify objects of interest, such as terrain regions for sampling tasks for a planetary explorer [23]. Although we have performed some preliminary experiments in that respect [18,2], extracting semantic information from terrain representations remains a major research area for outdoor mobile robots.

8.7 References

- [1] M. Asada. Building a 3-D World Model for a Mobile Robot from Sensory Data. In *Proc. IEEE Robotics and Automation*, Philadelphia, 1988.
- [2] J. Bares and W. Whittaker. Configuration of an Autonomous Robot for Mars Exploration. In *Proc. World Conference on Robotics*, 1988.
- [3] A. Bergman and C. K. Cowan. Noise-Tolerant Range Analysis for Autonomous Navigation. In *IEEE Conf. on Robotics and Automation*, San Francisco, 1986.
- [4] P. Besl. *Range Imaging Sensors*. Technical Report GMR-6090, General Motors Research Lab, Warren, MI, March 1988.