CVGIP: IMAGE UNDERSTANDING
Vol. 53, No. 1, January, pp. 1–13, 1991

Rigid Body Motion from Range Image Sequences

BERTHOLD K. P. HORN

MIT AI Laboratory, 545 Technology Square, Cambridge, Massachusetts 02139

AND

JOHN G. HARRIS

Hughes Aircraft Artificial Intelligence Center, 3011 Malibu Canyon Road, Malibu, California 90265

Received February 3, 1989; accepted March 9, 1990

An algorithm is described for recovering the six degrees of freedom of motion of a vehicle from a sequence of range images of a static environment taken by a range camera rigidly attached to the vehicle. The technique utilizes a least-squares minimization of the difference between the measured rate of change of elevation at a point and the rate predicted by the so-called *elevation rate constraint equation*. It is assumed that most of the surface is smooth enough so that local tangent planes can be constructed, and that the motion between frames is smaller than the size of most features in the range image. This method does not depend on the determination of correspondences between isolated high-level features in the range images. The algorithm has been successfully applied to data obtained from the range imager on the Autonomous Land Vehicle (ALV). Other sensors on the ALV provide an initial approximation to the motion between frames. It was found that the outputs of the vehicle sensors themselves are not suitable for accurate motion recovery because of errors in dead reckoning resulting from such problems as wheel slippage. The sensor measurements are used only to approximately register range data. The algorithm described here then recovers the difference between the true motion and that estimated from the sensor outputs. © 1991 Academic Press, Inc.

1. INTRODUCTION

Recovery of the six degrees of freedom of motion of a vehicle is an important problem in autonomous navigation. The algorithm described here will improve obstacle detection and avoidance, since the position of an obstacle can now be related to the position of the vehicle even after the obstacle leaves the field of view of the range camera. In addition, object recognition will be improved since multiple views of objects can be registered and fused. Another useful feature of this method is that it allows global maps of the terrain to be assembled from registered information extracted from many range images.

Recently the Hughes AI Center demonstrated the first cross-country map and sensor-based autonomous operation of a robotic vehicle [1]. Using data from a laser range scanner, the vehicle avoided difficult obstacles such as bushes, gullies, rock outcrops, and steep slopes. In this situation, all six degrees of freedom of motion are of importance, and one does not have the luxury of restricting oneself to planar motion, as is common in many indoor mobile robot applications. During the autonomous cross-country runs, the ALV pitched and rolled as much as 12° from the reference position while traversing rolling hills and shallow gullies. Sensors on board robot vehicles can measure such motions, but the results are subject to errors and usually not very accurate. Wheel slippage on loose surfaces, for example, contributes significantly to the dead reckoning error in cross-country experiments with the ALV. Furthermore, most vehicles do not have sensors to measure the full six degrees of freedom of motion. The ALV, for example, does not have a sensor to directly measure the vertical elevation change component.

The range sensor was developed by the Environmental Research Institute of Michigan [2]. Distance is measured by determining the phase shift between the modulation on an outgoing active laser signal and the modulation on a signal reflected by the terrain. By repeating such measurements at specified angular intervals, a range image of 64 × 256 bytes of 8 bits is built up once every half second. The field of view is 30° vertical by 80° horizontal. The maximum distance at which the sensor operates without ambiguity is 64 ft and the range resolution is 3 in. Because distance is measured using the phase shift of modulation at a single frequency, all distances are given modulo 64 ft. That is, objects at $x + 64$ ft yield the same range number as do objects at x ft. Figure 1 shows a sequence of three range scans; the vehicle moved approximately 8 ft between scans. The range images are

1049-9660/91 $3.00

presented in two forms: In Fig. 1(a) brightness encodes range, while in the shaded views of Fig. 1(b) brightness encodes surface orientation.[1]

Methods for aligning range images tend to fall into two categories. If the motion between scans is large, and there is no prior information about the movement, then symbolic feature-matching algorithms appear to be the appropriate choice [3–5]. If, on the other hand, the motion is small, then direct area-based techniques are more suitable. In the current application, the motion between range scans is large, but on-board sensors give us an estimate of the motion. This information can be used to approximately register the range maps. The motion-recovery algorithm need only deal with the differences between the vehicle's true motion and this initial estimate of the motion.

2. ALGORITHM

For a number of reasons, range imagery is not processed in its original spherical coordinate system. A *Cartesian Elevation Map* (CEM) is a more useful representation of range information. Data in the spherical, sensor-centered coordinate system of the range scanner are transformed into a Cartesian coordinate system. The Cartesian system is approximately aligned with true horizontal and vertical using readings from the vehicle's onboard pitch and roll sensors. In a CEM, the height, Z, is treated as a function of displacements X and Y in a horizontal plane. The result is a down-looking, map-view representation of terrain that is useful in autonomous navigation [6]. Depth information from other sensors, such as binocular stereo or imaging sonar, may be represented in the same form. The motion-recovery algorithm is greatly simplified when applied to CEMs rather than raw range images.

Some of these issues are explored further in the Appendix. The general derivation of the main result in the Appendix employs vector notation, which allows formulas to be written concisely. In the body of this paper, on the other hand, components of the vectors are used, which is intended to make the results easier to interpret on first reading.[2]

The first step in converting the range image from the spherical coordinate system form into a CEM is to calculate the X, Y, and Z Cartesian coordinate components of each measurement from the range $\rho(\theta, \phi)$ for each point in the image using the following transformations:

$$X = \rho(\theta, \phi) \sin \theta$$

$$Y = \rho(\theta, \phi) \cos \theta \cos \phi$$

$$Z = \rho(\theta, \phi) \cos \theta \sin \phi.$$

Here X, Y, and Z represent the cross-range, down-range, and elevation coordinates, respectively. Figure 2 shows the geometry of the ERIM range scanner ray deflection. In the range images, ϕ corresponds to the depression angle of a particular ray, and θ indicates the azimuth or lateral deflection of the ray. If vehicle pitch and roll can be estimated using on-board sensors, then the coordinates in the sensor-centered coordinate system can be further transformed into a Cartesian coordinate system aligned approximately with true horizontal and true vertical.

The way the laser light is generated and deflected by the scanning mirrors assures that rays corresponding to individual measurements in a range image all pass (approximately) through a single point.[3] In practice, the terrain is illuminated by a small but finite diameter beam, rather than an ideal ray of light. Individual beams fall on objects at different angles with respect to the local surface normal and illuminate the surface over an elliptical area that is referred to as the laser's "footprint." The measured distance to the surface is (approximately) a reflectance-weighted average over the illuminated area. Each of the 3-D points in the CEM, derived using the formulas above, denotes the approximate location of the center of a footprint.

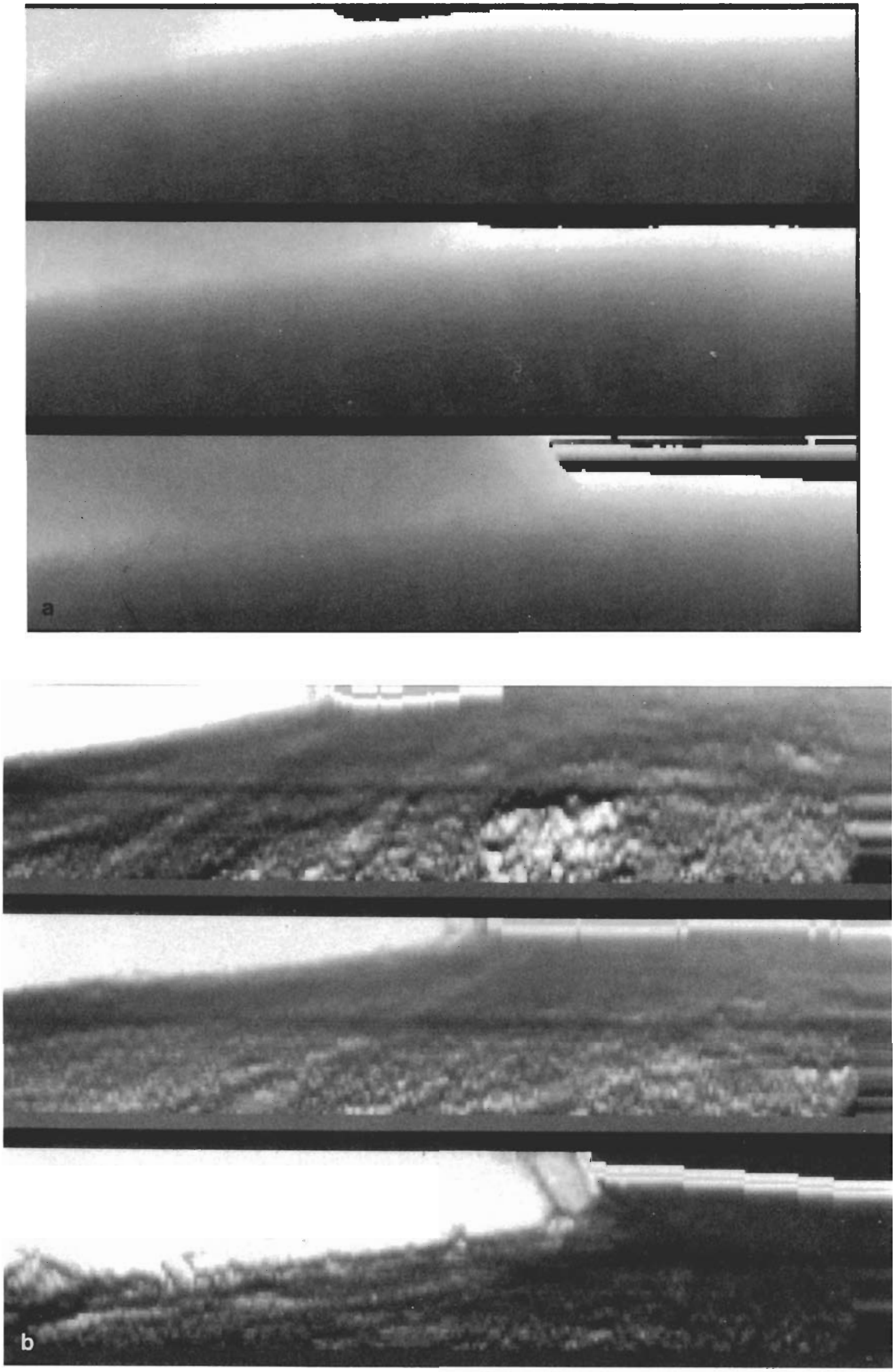
It should be clear that these points are in general not regularly placed on the terrain surface. Figure 3 shows the actual (X, Y) positions of each of the scanned points for the first range image in Fig. 1, within an 80 ft × 80 ft region in front of the scanner. Elevation data (Z values) are known only at the discrete points indicated. As one would expect, the sparsity of scanned points increases

[1] For more information on shaded display of terrain, see, e.g., [17].

[2] The two derivations also differ in other subtle ways that are discussed briefly in the Appendix, but that are not important here.

[3] Thus the geometry of range image formation here is similar to the geometry of perspective projection in ordinary optical image formation.

FIG. 1. (a) Sequence of three laser range images taken from the moving ALV. Each scan consists of 64 × 256 bytes of 8 bits that encode distance. Brightness here corresponds to distance, so brigher points are farther from the range camera than darker ones. The terrain is gently rolling, with a large outcrop in the upper left-hand side of the field of view. (There is a range ambiguity, since distances are measured modulo 64 ft. This explains the apparent discontinuity in range in the top right hand corner of the third image.) (b) Shaded views of sequence of three laser range images. Brightness here corresponds to the slope of the surface relative to the camera. Surfaces facing toward the right side appear brighter than surfaces turned towards the left. (So the large outcrop in the top left corner appears bright because it faces predominantly towards the right.) This mode of presentation makes apparent more of the detailed surface undulations, such as the tire tracks in the grass and the rock in the foreground of the first range image.


a
b

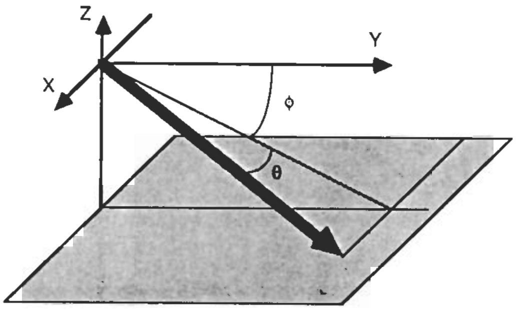


FIG. 2. ERIM range scanner coordinate system. Here ϕ corresponds to the depression angle of a particular ray, and θ indicates the azimuth or lateral deflection of the ray. (For rays aimed downward, such as the one shown in this figure, the angle ϕ is considered to be negative.)

with distance from the scanner, as well as with the inclination of the average local surface normal with respect to the incident rays. The complex laser scanning procedure with its forshortening effects and averaging over footprints can be approximated by a smoothing operation followed by a sampling process. Theoretically, if the smoothed terrain were actually bandlimited and and sampled frequently enough (that is, within the Nyquist rate), then the original terrain could be accurately reconstructed by interpolating a smooth surface between the scanned points. This is not an easy task, however, since the known depth values do not fall on a regular grid in the XY-plane.

There will usually be some regions in which sampling is not dense enough to properly reconstruct a smooth surface. For example, any region outside the field of view of the scanner, or in the shadow of some tall feature in the terrain, will be unknown. Areas where a local weighted average density of samples is below some threshold are located and excluded from the interpolation process. An iterative interpolation algorithm is used to fill in a continuous surface in all regions where the sampling is dense enough. In other respects the interpolation algorithm is similar to those used for recovering digital terrain models (DEMs) from contour maps [7] and those used to interpolate smooth surfaces from sparse binocular stereo data [8, 9]. It was found to be advantageous to use the elastic membrane model for interpolation rather than the more elaborate thin-plate model. This makes the iterative solution much simpler and faster. Figure 4 shows the final interpolated CEM.

A time-varying CEM can be viewed as a function of the form $Z(X, Y, t)$. Taking a full time derivative of Z via the

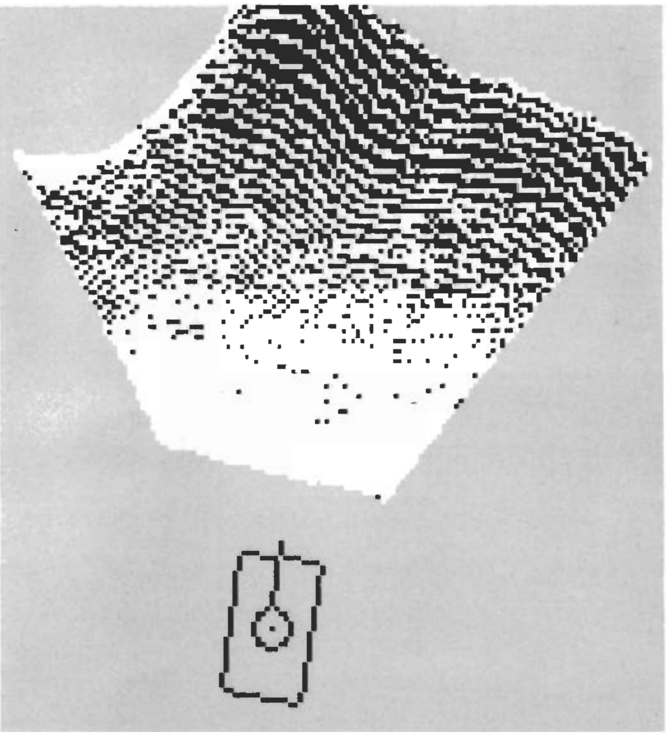


FIG. 3. Constraint-point arrays generated from the first laser range scan shown in Fig. 1. Each array covers an 80 ft × 80 ft patch of terrain in front of the vehicle. A pixel here represents a 6 in. × 6 in. square area of terrain. Bright pixels indicate points on the ground where some scan ray hits and where, as a result, elevation data is known. Elevations at points corresponding to dark pixels have to be interpolated from the known elevation data.

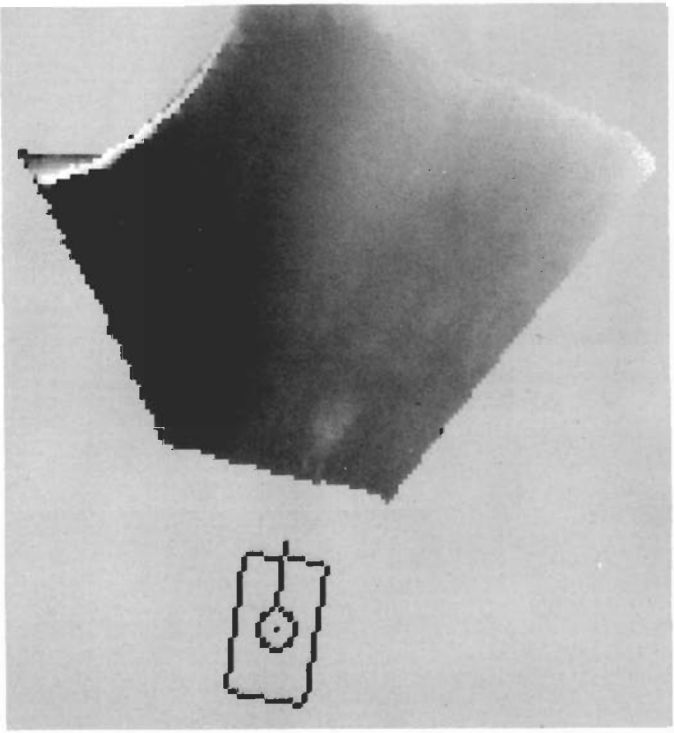


FIG. 4. Final interpolated CEM of the first range scan image shown in Fig. 1. Brightness here corresponds to elevation (that is, higher points are brighter). Note again the large outcrop near the top left (limiting the extent of the range image) and the small rock (light splotch) near the vehicle.

chain rule, the following equation is obtained

$$\frac{dZ}{dt} = \frac{\partial Z}{\partial X}\frac{dX}{dt} + \frac{\partial Z}{\partial Y}\frac{dY}{dt} + \frac{\partial Z}{\partial t}.$$

This can be written in the less intimidating form

$$\dot{Z} = p\dot{X} + q\dot{Y} + Z_t,$$

where the three partial derivatives of Z are denoted by

$$p = \frac{\partial Z}{\partial X}, \quad q = \frac{\partial Z}{\partial Y}, \quad \text{and} \quad Z_t = \frac{\partial Z}{\partial t},$$

and the components of velocity of a point in the range image are given by

$$\dot{X} = \frac{dX}{dt}, \quad \dot{Y} = \frac{dY}{dt}, \quad \text{and} \quad \dot{Z} = \frac{dZ}{dt}.$$

Note that p and q are just the slopes of the surface in the x and y direction respectively, while Z_t is the rate of change of elevation at a particular point in the CEM.

The above equation will be called the *elevation rate constraint equation*. The values of the partial derivatives p, q, and Z_t can be estimated at each pixel in the CEM, while $\dot{X}$, $\dot{Y}$, and $\dot{Z}$ are unknown. There is one such equation for every point in the CEM, so that if it contains n points, there are n equations in a total of $3n$ unknowns. The system of equations is extremely underconstrained and additional assumptions are necessary to provide a unique solution. In the above discussion no constraint on the motion of neighboring points was assumed, each point being able to move completely independently. In most real motions, even elastic deformations and fluid flows, neighboring points do, however, tend to have similar velocities. There are two basic ways of exploiting this observation in order to increase the amount of constraint.

- In analogy with a method for estimating optical flow [10], an energy function could be constructed that is a weighted sum of errors in the elevation rate constraint equation and a measure of the depature from smoothness of the velocity field. Furthermore, a penalty function for discontinuities can be added that makes it possible to allow for discontinuities along edges. This would force the recovered motion field to be smooth almost everywhere and provide a segmentation of the image into multiple moving objects.[4]
- In analogy with a so-called direct method for recovering motion from an ordinary image sequence [11], we could assume instead that the environment is a single rigid assemblage and that we have to recover the motion of the sensor relative to the environment. If a moving sensor in a rigid environment is assumed, there are only six degrees of freedom of motion to recover, so that the corresponding system of equations is now vastly overconstrained. This is the approach taken here.

Let $\mathbf{R} = (X, Y, Z)^T$ be a vector to a point on the surface (measured in a sensor-centered Cartesian coordinate system). If the sensor moves with instantaneous translational velocity $\mathbf{t}$ and instantaneous rotational velocity ω with respect to the environment, then the point $\mathbf{R}$ appears to move with a velocity

$$\frac{d\mathbf{R}}{dt} = -\mathbf{t} - \omega \times \mathbf{R}$$

with respect to the sensor [12]. The components of the velocity vectors are given by

$$\mathbf{t} = \begin{pmatrix} U \\ V \\ W \end{pmatrix} \quad \text{and} \quad \omega = \begin{pmatrix} A \\ B \\ C \end{pmatrix}.$$

Rewriting the equation for the rate of change of $\mathbf{R}$ in component form yields

$$\dot{X} = -U - BZ + CY$$
$$\dot{Y} = -V - CX + AZ$$
$$\dot{Z} = -W - AY + BX,$$

where the dots denote differentiation with respect to time. Substituting these expanded equations into the elevation rate constraint equation itself yields

$$pU + qV - W + rA + sB + tC = Z_t,$$

where

$$r = -Y - qZ, \quad s = X + pZ, \quad \text{and} \quad t = qX - pY.$$

If there are n pixels in the image, the resulting n equations can be written in matrix form as

$$\underbrace{\begin{pmatrix} p_1 & q_1 & -1 & r_1 & s_1 & t_1 \\ p_2 & q_2 & -1 & r_2 & s_2 & t_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_i & q_i & -1 & r_i & s_i & t_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_n & q_n & -1 & r_n & s_n & t_n \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} U \\ V \\ W \\ A \\ B \\ C \end{pmatrix}}_{\mathbf{x}} = \underbrace{\begin{pmatrix} (Z_t)_1 \\ (Z_t)_2 \\ \vdots \\ (Z_t)_i \\ \vdots \\ (Z_t)_n \end{pmatrix}}_{\mathbf{b}}$$

[4] We have also worked on this alternate approach and will report on its elsewhere when our work is completed.

or $A\mathbf{x} = \mathbf{b}$. The pixels are numbered from 1 to n as denoted by the subscripts. The above matrix equation corresponds to n linear equations in only six unknowns (namely U, V, W, A, B, and C).

Rather than arbitrarily choosing six of the equations and solving the resulting set of equations, a least-squares error minimization technique is employed. The least-squares solution that minimizes the norm $\|A\mathbf{x} - \mathbf{b}\|^2$ satisfies the equation

$$A^TA\mathbf{x} = A^T\mathbf{b},$$

as texts on linear algebra [13] demonstrate.

Multiplying both sides of the above matrix equation by A^T yields the 6×6 system of equations

$$\sum_{i-1}^{n} \begin{pmatrix} p_i^2 & p_iq_i & -p_i & p_ir_i & p_is_i & p_it_i \\ p_iq_i & q_i^2 & -q_i & q_ir_i & q_is_i & q_it_i \\ -p_i & -q_i & 1 & -r_i & -s_i & -t_i \\ p_ir_i & q_ir_i & -r_i & r_i^2 & r_is_i & r_it_i \\ p_is_i & q_is_i & -s_i & r_is_i & s_i^2 & s_it_i \\ p_it_i & q_it_i & -t_i & r_it_i & s_it_i & t_i^2 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ A \\ B \\ C \end{pmatrix} = \sum_{i=1}^{n} (Z_t)_i \begin{pmatrix} p_i \\ q_i \\ -1 \\ r_i \\ s_i \\ t_i \end{pmatrix},$$

or more concisely,

$$\sum_{i-1}^{n} \mathbf{c}_i\mathbf{c}_i^T \begin{pmatrix} U \\ V \\ W \\ A \\ B \\ C \end{pmatrix} = \sum_{i-1}^{n} (Z_t)_i\mathbf{c}_i,$$

where

$$\mathbf{c}_i = \begin{pmatrix} p_i \\ q_i \\ -1 \\ r_i \\ s_i \\ t_i \end{pmatrix}.$$

Note that $\mathbf{c}_i$ is a vector whose components are determined locally at each point of the CEM.

3. IMPLEMENTATION

This motion-recovery algorithm has been implemented and tested on synthetic CEMs of smoothly undulating surfaces. With exact synthetic data it produces exact motion estimates. Significant amounts of random noise can be added to the synthetic data before the motion estimates are degraded noticeably, because the least squares problem is heavily overdetermined. The relationship between noise in the measurements and errors in the motion estimates is complex, as it depends on the surface shape, the type of motion, and the properties of the simulated range sensor. While detailed sensitivity analysis in the general case is hard, we did notice that performance seems to be degraded when the field of view is reduced. This is in general agreement with what is known from photogrammetry as applied to binocular stereo and motion vision [11, 14].

Experiments with synthetic range images are useful, because the result of the computation can be compared to the accurately known motion used in generating the data. Good performance on synthetic data is not unexpected, however, since there is no approximation involved in the derivation of the algorithm. It is thus more interesting to test the algorithm on CEMs derived from real range images. In this case, however, it is difficult to obtain measurements of the actual motion with sufficient accuracy for meaningful comparison with the results of the algorithm.[5] This makes it hard to directly assess the accuracy of the recovered motion. There is an indirect way of seeing how well the system performs, however: Range maps cannot be mosaicked effectively using motion information provided by the on-board sensors, but they can be put together into a global range map, with very small errors in the overlap regions, when the algorithm presented here is used to determine the actual motion between successive range camera positions.

We now describe the actual algorithm in more detail. Simple finite difference approximations are used to estimate the three different partial derivatives of Z. The terms $\mathbf{c}_i\mathbf{c}_i^T$ and $(Z_t)_i\mathbf{c}_i$ are computed at each point in the CEM and added into a total. The resulting 6×6 linear system of linear equations is easily solved. The cycle time between scans for the autonomous cross-country

[5] The algorithm can, for example, recover rotational motion components that move the range image only a fraction of a picture cell at the center of the image. This corresponds to a very small visual angle. The angular rate sensors would have to be sensitive to very low rotational speeds in order to provide a usable comparison signal. In fact, a system using something like the algorithm here may one day provide a better way of measuring attitude and angular rates than presently used systems, provided the new scheme can be implemented economically.

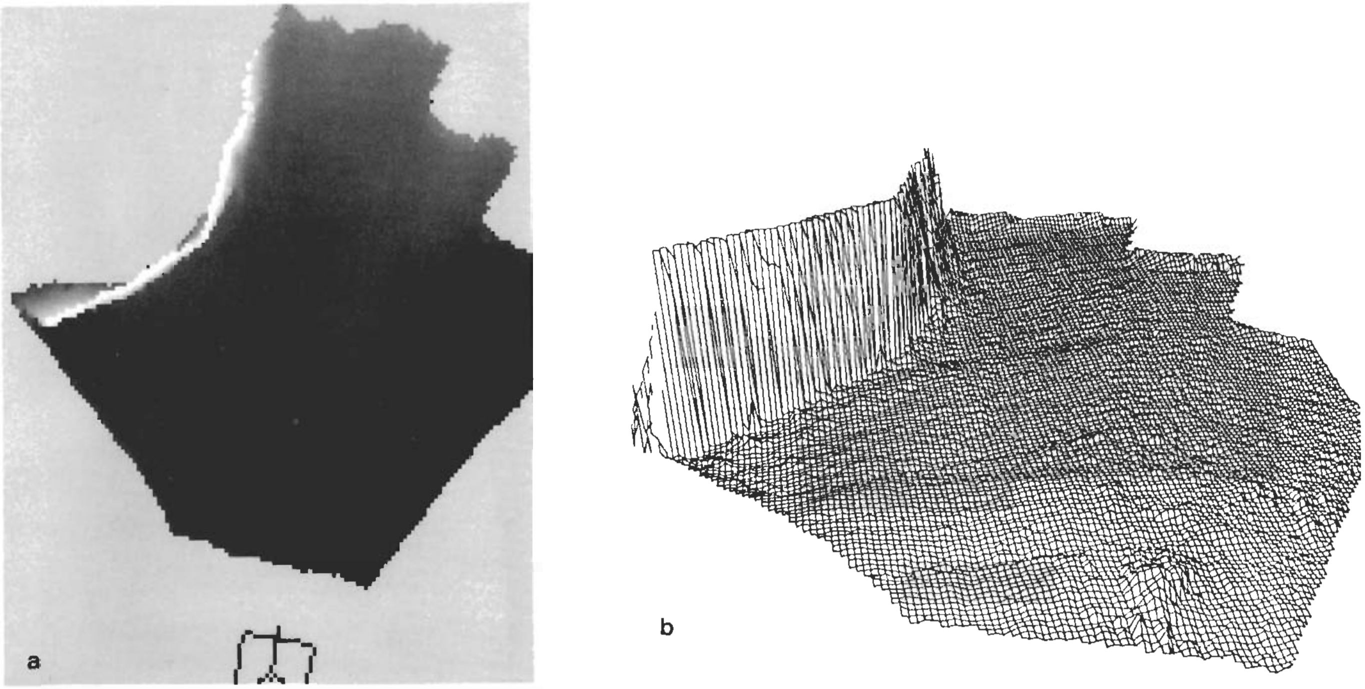


FIG. 5. (a) Fused CEM from the three range scans shown in Fig. 1. The motion recovery algorithm was used to find the exact six-degree of freedom motion between the scans. Note that this figure covers a larger area than does Fig. 4. (b) Wireframe view of the fused CEM shown in (a).

system is approximately 8 s; in this time the ALV typically moved about 10 ft.[6] Estimates from sensors on board the vehicle are used to approximately register two successive CEMs, and the motion-recovery algorithm described above is used to perform the final registration of the CEMs. Vehicle sensors provide approximate values for four of the six motion components. Unforunately, no sensors were available that could measure vertical elevation change or vehicle roll; zero values were supplied for these components during the approximate registration process. The motion-recovery algorithm typically finds an additional translation of at most a few feet and a rotation of a few degrees.

Once the six degrees of freedom of motion are known between two CEMs, it is a simple matter to transform one CEM to the coordinate system of the other. This fusion process combines information from both CEMs. Wherever points in space are adequately described by both CEMs, the most recent data are retained. In this way, large terrain maps can be built up from sequences of range images. The final fused CEM obtained from the three range scans of Fig. 1 is shown in Fig. 5. These multi-scan terrain maps have immediate applications in simplifying and improving obstacle detection performance. For example, the ground immediately in front of the vehicle cannot be seen, since the ERIM range scanner is mounted on the front of the vehicle at a height of 8 ft. The fusion algorithm described above fills in this "blind spot" in front of the vehicle, performing much better than other ad hoc solutions that have been used in the past.

The results of this new area-based CEM fusion algorithm are promising. The algorithm is robust in the presence of certain kind of erros. At times, for example, a person will walk through the field of view of the range camera while the ALV is navigating. Because of the person's relatively fast motion, the person typically appears only in one scan. Since the technique uses information from a large area of terrain, the estimated motion is not affected seriously by a small fraction of "erroneous" elevation values. Figure 6 shows an example of two range images where the small error introduced in this way does not significantly affect the fusing of successive CEMs. A symbolic feature-matching algorithm might have more trouble with range images such as these, because some of the most obvious "features" in one image do not appear in the other image. Note also that typical natural cross-country CEMs often do not have major distinctive fea-

[6] Since range imagers now take a significant amount of time to scan an image, vehicle motion during that time interval has to be taken into account. It is easy to compensate for known steady translational motion, since one can compute the time at which each range estimate is obtained. More difficult to deal with is uncontrolled rotational motion, as may occur when a vehicle is moving over uneven terrain. It is helpful in this regard to have a camera mounting that strongly attenuates rotational motion components above some frequency, as is used, for example, in the motion picture industry when filming from unsteady vibrating platforms such as helicopters.

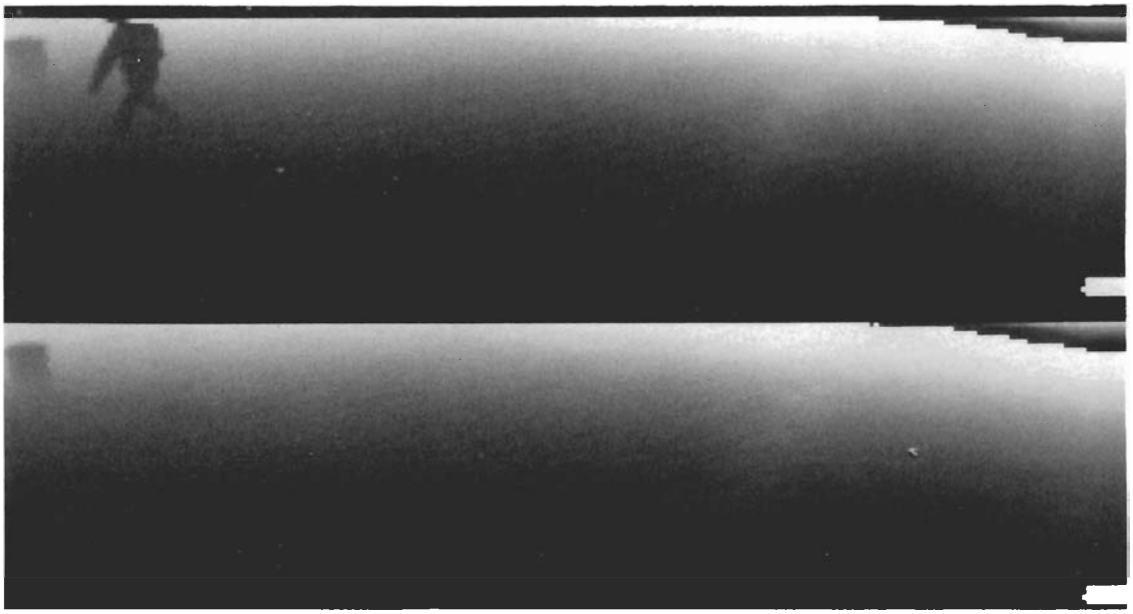


FIG. 6. Two successive range images that might give feature-based algorithms some problems because of differences in the two scenes being viewed. (Note the person walking through the upper left area of the first range image.) The new area-based algorithm had no trouble recovering the motion of the vehicle between these two CEMs.

tures that such an algorithm might use, but do contain enough gently rolling terrain for the new area-based recovery algorithm to operate effectively.

4. CONCLUSION

In the future, researchers at Hughes plan to port this algorithm to an on-board WARP systolic array processor. The core of the range processing routines have already been ported to the Connection Machine, allowing a reduction of cycle time to below the half-second cycle time of the ERIM range scanner [15]. With this fast cycle time, the vehicle motion between scans will be so small that it is expected that estimates from on-board sensors will not be required for reliable fusion of CEMs, and the new algorithm can be applied directly.

In the meantime, work is beginning on a hierarchical strategy in which a coarser level will be used to provide a motion estimate for the next finer level. Confusing detail with high spatial frequency content is suppressed in smoothed and subsampled CEMs which allows the motion-recovery algorithm to be more tolerant of poor initial motion estimates. At the same time, the motion will not be estimated very accurately at the coarser levels. Finer levels of resolution are needed to accurately recover motion. It is hoped that the coarsest level will not require any initial motion estimate.

5. APPENDIX: GENERAL FORMULATION

In the body of this paper a Cartesian coordinate system is used that is aligned with local vertical and hence is not sensor-centered. This is appropriate because presently successive range images have to be brought into approximate alignment using information from on-board sensors. This alignment is necessitated by the fact that the vehicle moves a considerable distance during the relatively long computational cycle time. When the cycle time is reduced, however, vehicle motion between scans will be so small that the algorithm can deal with it directly, without initial approximate alignment using vehicle sensor information. This makes it possible (and desirable) to work directly in a sensor-centered coordinate system. Accordingly, the method for recovering vehicle motion from time-varying range images is developed in this appendix using a sensor-centered coordinate system. Also, for conciseness, vector notation is used instead of expressions in terms of the components of vectors. To distinguish the sensor-centered Cartesian coordinate system from the external Cartesian coordinate system used earlier, three-dimensional coordinates here carry an overbar.

A.1. *Basic Approach*

Consider a time-varying range image $R(\alpha, \beta, t)$ that gives distance to points on a surface in the environment as a function of two image coordinates and time t. A number of coordinate schemes will be discussed in Section A.6; α and β may, for example, be Cartesian coordinates in a planar image or latitude and longitude in a spherical image. The task is to recover the motion of the range sensor with respect to the static environment.[7]

The instantaneous motion of the sensor can be described fully by the translational velocity, $\mathbf{t}$, of the center of projection and the rotational velocity, ω, of the sensor about an axis through the center of projection. These velocities are measured with respect to the moving sensor coordinate system, and thus are likely to be time-varying. The task is to recover the instantaneous velocities.

The procedure discussed here uses the first partial derivatives of range with respect to image coordinates, as well as time. There derivatives can be estimated from range data using first-difference methods. If the data are noisy, they may need to be smoothed first, or, equivalently, the derivatives may be estimated by fitting low-order polynomials to the measured data in a small neighborhood.

A.2. *Range Rate Constraint Equation*

The surface is assumed to be smooth enough so that local tangent planes can be constructed. Let $\hat{\mathbf{n}}$ be the (outward pointing) unit normal. (Methods for estimating the normal from the derivatives of range will be discussed in Section A.7.)

Let $\mathbf{R}$ be a vector to a point on the surface (measured in a sensor-centered coordinate system). This point appears to move with velocity

$$\dot{\mathbf{R}} = -\mathbf{t} - \omega \times \mathbf{R}$$

with respect to the sensor. The normal component of the velocity (what is left after removing the component in the tangent plane) has magnitude

$$V_n = \dot{\mathbf{R}} \cdot \hat{\mathbf{n}} = -\mathbf{t} \cdot \hat{\mathbf{n}} - [\omega \mathbf{R} \hat{\mathbf{n}}],$$

where $[\mathbf{a}\ \mathbf{b}\ \mathbf{c}]$ denotes the triple product of the vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$. Note that the velocity component in the tangent plane cannot be determined locally, since it merely moves points around in the tangent plane.[8]

What can be estimated directly from the measured data is R_t, the range rate at a particular picture cell. This range rate does not, of course, uniquely determine the velocity of a point on the surface, but it does constrain the velocity to have the form

$$\dot{\mathbf{R}} = R_t \hat{\mathbf{r}} + \mathbf{s},$$

where $\hat{\mathbf{r}}$ is a unit vector in the direction $\mathbf{R}$, while $\mathbf{s}$ is an arbitrary vector in the tangent plane. The normal component of this velocity vector is just

$$V_n = \dot{\mathbf{R}} \cdot \hat{\mathbf{n}} = R_t(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}}),$$

independent of the tangential component,[9] since $\mathbf{s} \cdot \hat{\mathbf{n}} = 0$.

Equating this with the other equation for the normal component yields:

$$R_t(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}}) + \mathbf{t} \cdot \hat{\mathbf{n}} + [\omega \mathbf{R} \hat{\mathbf{n}}] = 0.$$

This is called the *range rate constraint equation*. The range rate constraint equation is analogous to the *brightness change constraint equation*, which is used in some methods for estimating the optical flow [10], as well as by direct methods for recovering motion from time-varying imagery [11]. The equation can be solved for R_t and used to predict the range rate at every picture cell, if the translational and rotational motion, as well as the surface orientation, are known.

A.3. *Recovering Instantaneous Velocity Components*

If the motion is not known, values for $\mathbf{t}$ and ω may be sought that make the predicted and observed range rates at every picture cell equal. In practice, of course, there will be measurement errors and so it makes more sense instead to minimize an error integral like:[10]

$$\iint (R_t(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}}) + (\mathbf{t} \cdot \hat{\mathbf{n}} + [\omega \mathbf{R} \hat{\mathbf{n}}]))^2 \, d\alpha \, d\beta.$$

The integral is over the whole range image, or over a specified image region if the image has been segmented into regions corresponding to objects moving differently.

Fortunately, the integrand is the square of a term that is linear in both the instantaneous translational and rotational velocities. This means that a closed-form solution can be obtained by differentiating the integral with respect to $\mathbf{t}$ and ω and setting the result equal to zero. The following two equations are obtained in this way:

$$\iint (R_t(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}}) + (\mathbf{t} \cdot \hat{\mathbf{n}} + [\omega \mathbf{R} \hat{\mathbf{n}}]))\hat{\mathbf{n}} \, d\alpha \, d\beta = \mathbf{0},$$

$$\iint (R_t(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}}) + (\mathbf{t} \cdot \hat{\mathbf{n}} + [\omega \mathbf{R} \hat{\mathbf{n}}]))(\mathbf{R} \times \hat{\mathbf{n}}) \, d\alpha \, d\beta = 0.$$

[7] Equivalently, the sensor can be considered fixed, with a single rigid object in motion relative to the sensor.

[8] This is the three-dimensional analog of the so-called *aperture problem* found in the estimation of optical flow [10].

[9] Note that $(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}})$ will be negative for visible surfaces.

[10] The integrand could be normalized by dividing by $(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}})$ in order to obtain the error in the range rate, but this would create problems for points near the limbs of objects, where the measurement is along a ray that is almost tangent to the surface.

Here use has been made of the identity $[\mathbf{a}\ \mathbf{b}\ \mathbf{c}] = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$. For convenience, let $\mathbf{d} = (\mathbf{R} \times \hat{\mathbf{n}})$; then

$$\left(\iint \hat{\mathbf{n}}\hat{\mathbf{n}}^T\, d\alpha\, d\beta\right) \mathbf{t} + \left(\iint \hat{\mathbf{n}}\mathbf{d}^T\, d\alpha\, d\beta\right) \omega$$

$$= -\iint R_t(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}\, d\alpha\, d\beta,$$

$$\left(\iint \mathbf{d}\hat{\mathbf{n}}^T\, d\alpha\, d\beta\right) \mathbf{t} + \left(\iint \mathbf{d}\mathbf{d}^T\, d\alpha\, d\beta\right) \omega$$

$$= -\iint R_t(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}})\mathbf{d}\, d\alpha\, d\beta,$$

where use has been made of the identities $(\mathbf{a} \cdot \mathbf{b}) = (\mathbf{b} \cdot \mathbf{a})$ and $(\mathbf{a} \cdot \mathbf{b}) = \mathbf{a}^T\mathbf{b}$. Each of the terms in parentheses is a 3 × 3 matrix obtained by integrating the indicated dyadic product, while each of the two terms appearing on the right-hand side is a vector with three components. This pair of vector equations can be viewed as a system of six linear scalar equations in the components of $\mathbf{t}$ and ω. Note that, in contrast to the situation in direct motion vision [11], the magnitude of the translational velocity vector can be recovered here, since absolute depth measurements are provided.

If range is obtained using either binocular stereo or motion vision methods, the accuracy will decrease with distance and so the error contributions should be weighted inversely with range squared. In this case the equations can be modified to read

$$\left(\iint \frac{\hat{\mathbf{n}}\hat{\mathbf{n}}^T}{R^2}\, d\alpha\, d\beta\right) \mathbf{t} + \left(\iint \frac{\hat{\mathbf{n}}\mathbf{d}^T}{R^2}\, d\alpha\, d\beta\right) \omega$$

$$= -\iint \frac{R_t}{R^2}(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}\, d\alpha\, d\beta,$$

$$\left(\iint \frac{\mathbf{d}\hat{\mathbf{n}}^T}{R^2}\, d\alpha\, d\beta\right) \mathbf{t} + \left(\iint \frac{\mathbf{d}\mathbf{d}^T}{R^2}\, d\alpha\, d\beta\right) \omega$$

$$= -\iint \frac{R_t}{R^2}(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}})\mathbf{d}\, d\alpha\, d\beta.$$

It is also possible to use the normal vector $\mathbf{n} = (p, q, -1)^T$ instead of the unit normal $\hat{\mathbf{n}}$, and $(\mathbf{R} \times \mathbf{n})$ instead of $(\mathbf{R} \times \hat{\mathbf{n}})$ for $\mathbf{d}$. The only difference is that error contributions from different areas are weighted somewhat differently in the error integral. Note that the vector $\mathbf{c}$ used in the body of this paper is the composition of $\mathbf{n}$ and $(\mathbf{R} \times \mathbf{n})$. This should help clarify the relationship between the result presented here and the version in terms of components of vectors given in the body of the paper.

A.4. *Degeneracies and Ambiguity*

The method fails when the coefficient matrix is singular. This happens, for example, when there is a single planar surface, in which case $\hat{\mathbf{n}}$ is the same everywhere. In this case the matrix obtained by integrating $\hat{\mathbf{n}}\hat{\mathbf{n}}^T$ only has rank one. The result is that one cannot determine the component of translational motion tangent to the plane, as well as the rotation about a normal to the plane. The method also fails when the sensor is at the center of a spherical surface, where $\mathbf{R} \parallel \hat{\mathbf{n}}$, because then $\mathbf{d}$ is zero everywhere. In this case the matrix obtained by integrating $\mathbf{d}\mathbf{d}^T$ is zero and so the rotational component is completely undetermined.

Actually, these are just special cases of more general ones. There is a problem, for example, when the surface is cylindrical, since the normals then all lie in a common plane. In this case the matrix obtained by integrating $\hat{\mathbf{n}}\hat{\mathbf{n}}^T$ only has rank two. The result is that the component of translation in a direction perpendicular to the common plane of the normals, that is, along the direction of the rulings of the surface, cannot be determined.

Similarly, there is a problem in determining one of the components of rotation when the surface is a solid of revolution and the sensor happens to be on the axis. In this case all of the vectors $\mathbf{d} = (\mathbf{R} \times \hat{\mathbf{n}})$ will lie in a common plane. Consequently, the matrix obtained by integrating $\mathbf{d}\mathbf{d}^T$ has only rank two, and the component of rotation perpendicular to the common plane, that is, about the axis of the solid of revolution, cannot be found.

It should be obvious that in all of these situations, no method can recover the motion unambiguously, since certain components of the motion do not affect the data. This problem is analogous to, but simpler than, the problem of *critical surfaces* that arises in photogrammetry and motion vision [14, 16].

A.5. *Implementation Notes*

In practice, the double integrals will, of course, be replaced by double sums. At each picture cell the rates of change of range with image coordinates are estimated in order to determine the local unit normal $\hat{\mathbf{n}}$ (as discussed in Section A.7). The range rate R_t is also estimated and $\mathbf{d} = (\mathbf{R} \times \hat{\mathbf{n}})$ as well as $(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}})$ computed. The symmetric 6 × 6 coefficient matrix and the right-hand-side 6-component vector are then built up by accumulating products of these intermediate results.

When applying this method to real images, a few things need to be paid special attention to:

- An attempt should be made to detect range discontinuities and to avoid using range data near them. At a range discontinuity, finite difference estimates of derivatives are likely to be based on range measurements in picture cells corresponding to patches on unrelated surfaces. Such estimates are likely not to be very accurate and will contribute to the error in the result. Also, near limbs of objects, the rates of change of range with image coordinates are very high, and finite difference estimates of surface orientations may consequently not be very accuate. In the formulation presented here, this latter effect is not much of a problem, however, since the range rate is in effect scaled by multiplication by $(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}})$, a quantity that will be small when the rays are almost tangent to the surface being scanned.

- One also needs to be careful to avoid aliasing problems due to undersampling, since the estimates of the derivatives will otherwise be corrupted. Care should be taken to low-pass filter, or at least smooth, the range data before sampling, since aliasing effects cannot be removed after sampling. If the data are unavoidably corrupted by aliasing, and most of the signal content is at lower frequencies, while the noise spectrum is more or less flat, then some improvement in performance may be attained by smoothing the sampled data, or, equivalently, by basing estimates of derivatives on computational stencils with fairly large support.
- One has to make sure that differences between quantized range measurements are large enough so that estimates of the derivatives are not corrupted too severely by quantization noise. Depending on the velocity of the sensor and distance to the scene, range values at neighboring image points or in successive frames may differ by only a few quantization steps. In this case it may be necessary to subsample, that is, use only every *n*th range value (either in space or in time), or, equivalently, to base the estimates on computational stencils with large support in both space and time. If subsampling is employed, it is important to remember to low-pass filter, or at least smooth, the range data before sub-sampling.

All of these cautionary notes apply equally well, of course, to the estimation of optical flow [10], and the direct computation of rigid body motion [11] from time-varying imagery.

A.6. *Range Image Sampling Schemes*

Two different range sampling arrangements will be considered.

- Planar sampling: the directions of sampling rays are determined by a regular rectangular grid of points on a planar surface. This is analogous to the method used most often to scan optical images. The flat surface corresponds to the image plane, the perpendicular from the center of projection onto this plane corresponds to the optical axis, and the length of the perpendicular corresponds to the "focal length." In this case, depth—distance along the "optical axis"—may be given rather than range.
- Spherical sampling: the directions of sampling rays are determined by a regular grid of longitudes and latitudes on a spherical surface. The center of projection is at the center of the sphere. This corresponds to the method now used in some range scanners. The terms *azimuth* and *elevation* are commonly used for the horizontal and vertical components of deflection, respectively. Unfortunately, the relationship between these terms and the terms latitude and longitude used here depends on the sequence of beam deflections and the arrangements of the axes with respect to local vertical.[11]

With regular planar sampling, the range image can be written in the form $\rho(x, y)$, where x and y are coordinates in the image plane.[12] The image plane is considered to be at unit distance from the center of projection.[13] With the x- and y-axes defined as above, the z-axis lies along the optical axis, and so the vector connecting the center of projection to an image point is just $\mathbf{r} = (x, y, 1)^T$. The vector to the corresponding point in the scene will be written $\mathbf{R} = (\overline{X}, \overline{Y}, \overline{Z})^T$. A planar range image $\rho(x, y)$ can be easily converted into a planar depth image $\overline{Z}(x, y)$, by noting that $\rho(x, y) = r\overline{Z}(x, y)$, where $r = \|\mathbf{r}\|$ is the length of the vector $\mathbf{r}$. Such a depth image is often more directly useful than the range image itself. Note that $\mathbf{R} = \overline{Z}\mathbf{r}$, or

$$\overline{X} = x\overline{Z}, \quad \overline{Y} = y\overline{Z}, \quad \text{and} \quad \overline{Z} = \overline{Z}.$$

Image coordinates corresponding to particular points in the scene may be computed using $x = \overline{X}/\overline{Z}$ and $y = \overline{Y}/\overline{Z}$.

With regular spherical sampling, the range image can be written instead in the form $\rho(\xi, \eta)$, where ξ is the longitude, and η is the latitude on the sphere, while ρ is the range. Let the x-axis correspond to $+\pi/2$ in latitude, while the y-axis is at zero latitude and $+\pi/2$ in longitude. Then a vector to a point on the unit sphere is given by

$$\hat{\mathbf{r}} = (\sin \eta, \cos \eta \sin \xi, \cos \eta \cos \xi)^T.$$

The vector to the corresponding point in the scene can once again be written $\mathbf{R} = (\overline{X}, \overline{Y}, \overline{Z})^T$. If $\hat{\mathbf{r}} = (1/r)\mathbf{r}$ is a unit vector in the direction $\mathbf{r}$, then clearly $\mathbf{R} = \rho\hat{\mathbf{r}}$, or

$$\overline{X} = \rho \sin \eta, \quad \overline{Y} = \rho \cos \eta \sin \xi,$$
$$\text{and} \quad \overline{Z} = \rho \cos \eta \cos \xi.$$

[11] In the case of the camera on the Viking Mars lander, for example, azimuth corresponds to longitude, while in the ERIM scanner azimuth corresponds to latitude.

[12] Measured from the principal point, which is where the perpendicular from the center of projection pierces the image plane.

[13] Equivalently, one may normalize measurements in the image plane by dividing by the focal length.

The latitude and longitude can be recovered from the coordinates of a point in the scene using[14]

$$\tan \xi = \frac{\overline{Y}}{\overline{Z}} \quad \text{and} \quad \tan \eta = \frac{\overline{X}}{\sqrt{\overline{Y}^2 + \overline{Z}^2}}.$$

A.7. *Estimating the Surface Normal*

The method for recovering motion from time-varying range images requires estimates of the surface normal. Different methods for estimating the surface normal apply to different range image sampling schemes.

With regular planar sampling of a depth image, for example, two tangent directions on the surface can be found by taking partial derivatives of the equation $\mathbf{R} = \overline{Z}\mathbf{r}$ with respect to x and y. They are

$$\mathbf{R}_x = \overline{Z}_x\mathbf{r} + \overline{Z}\hat{\mathbf{x}} \quad \text{and} \quad \mathbf{R}_y = \overline{Z}_y\mathbf{r} + \overline{Z}\hat{\mathbf{y}},$$

where use has been made of the fact that $\mathbf{r}_x$ and $\mathbf{r}_y$ are unit vectors in the x and y directions, respectively, since $\mathbf{r} = (x, y, 1)^T$. The normal is orthogonal to all tangents, so it is parallel to the cross-product of the two tangents above, that is, it is parallel to

$$\mathbf{R}_x \times \mathbf{R}_y = \overline{Z}(\overline{Z}_x(\mathbf{r} \times \hat{\mathbf{y}}) + \overline{Z}_y(\hat{\mathbf{x}} \times \mathbf{r}) + \overline{Z}\hat{\mathbf{z}}),$$

where use has been made of the fact that $\hat{\mathbf{x}} \times \hat{\mathbf{y}} = \hat{\mathbf{z}}$, a unit vector in the z direction. The expression in parentheses can be used to compute the normal direction given estimates of the partial derivatives of depth $\overline{Z}$ with respect to image coordinates x and y. In component form the above leads to

$$\mathbf{n}_d = \begin{pmatrix} -\overline{Z}_x \\ -\overline{Z}_y \\ \overline{Z} + (x\overline{Z}_x + y\overline{Z}_y) \end{pmatrix}.$$

The unit normal $\hat{\mathbf{n}}$ can be easily computed from this result.

If the field of view is very narrow, x and y will be small, and an orthographic approximation of the perspective projection can be employed. In this case, the normal may be considered to be parallel to $(-\overline{Z}_x, -\overline{Z}_y, \overline{Z})^T$.

As, an example of the general formula, consider the planar surface defined by $Z = Z_0 + pX + qY$. It is easy to show that

$$Z = \frac{Z_0}{1 - px - qy}, \quad Z_x = \frac{pZ_0}{(1 - px - qy)^2},$$
$$\text{and} \quad Z_y = \frac{qZ_0}{(1 - px - qy)^2},$$

so that the normal is parallel to

$$\frac{Z_0}{(1 - px - qy)^2} \begin{pmatrix} -p \\ -q \\ 1 \end{pmatrix}.$$

While the magnitude of this vector varies, the direction is clearly independent of image position and lies orthogonal the planar surface, as expected.

If we are given a regular planar-sampled range image (as opposed to a depth image), differentiation of the equation $\mathbf{R} = \rho\hat{\mathbf{r}}$ with respect to x and y again yields two tangents,

$$\mathbf{R}_x = \rho_x\hat{\mathbf{r}} + \rho\,\frac{r\hat{\mathbf{x}} - x\hat{\mathbf{r}}}{r^2} \quad \text{and} \quad \mathbf{R}_y = \rho_y\hat{\mathbf{r}} + \frac{r\hat{\mathbf{y}} - y\hat{\mathbf{r}}}{r^2},$$

whose cross-product,

$$\mathbf{R}_x + \mathbf{R}_y = \frac{R}{r^4}((r^2\rho_x - x\rho)(\mathbf{r} \times \hat{\mathbf{y}})$$
$$+ (r^2\rho_y - y\rho)(\hat{\mathbf{x}} \times \mathbf{r}) + r^2\rho\hat{\mathbf{z}}),$$

will be parallel to the normal. In component form the above leads to

$$\mathbf{n}_p = \begin{pmatrix} -(r^2\rho_x - x\rho) \\ -(r^2\rho_y - y\rho) \\ \rho + r^2(x\rho_x + y\rho_y) \end{pmatrix}.$$

The unit normal $\hat{\mathbf{n}}$ can be easily computed from this result.

With regular spherical sampling of a range image, two tangent directions can be found by taking partial derivatives of the equation $\mathbf{R} = \rho\hat{\mathbf{r}}$ with respect to ξ and η:

$$\mathbf{R}_\xi = \rho_\xi\hat{\mathbf{r}} + \rho\hat{\mathbf{r}}_\xi \quad \text{and} \quad \mathbf{R}_\eta = \rho_\eta\hat{\mathbf{r}} + \rho\hat{\mathbf{r}}_\eta.$$

Note that the two derivatives $\hat{\mathbf{r}}_\xi$ and $\hat{\mathbf{r}}_\eta$ have to be perpendicular to $\hat{\mathbf{r}}$, since $\hat{\mathbf{r}}$ is a unit vector. It so happens that $\hat{\mathbf{r}}_\xi$ and $\hat{\mathbf{r}}_\eta$ are orthogonal to one another as well. The normal is parallel to the cross product of the two tangents, that is, parallel to

$$\mathbf{R}_\xi \times \mathbf{R}_\eta = \rho(\rho_\xi(\hat{\mathbf{r}} \times \hat{\mathbf{r}}_\eta) + \rho_\eta(\hat{\mathbf{r}}_\xi \times \hat{\mathbf{r}}) + \rho(\hat{\mathbf{r}}_\xi \times \hat{\mathbf{r}}_\eta)).$$

[14] In the choice of coordinate axes, care has to be taken to ensure that the resulting Cartesian coordinate system is right-handed [12]. This may entail, for example, having points "in front" of the camera with negative values for $\overline{Z}$, or having the "vertical" image axis be positive downward.

Now

$$(\hat{\mathbf{r}} \times \hat{\mathbf{r}}_\eta) = \sec \eta \hat{\mathbf{r}}_\xi, \quad (\hat{\mathbf{r}}_\xi \times \hat{\mathbf{r}}) = \cos \eta \hat{\mathbf{r}}_\eta,$$
$$\text{and} \quad (\hat{\mathbf{r}}_\xi \times \hat{\mathbf{r}}_\eta) = -\cos \eta \hat{\mathbf{r}},$$

so

$$\mathbf{R}_\xi \times \mathbf{R}_\eta = \rho(\rho_\xi \sec \eta \hat{\mathbf{r}}_\xi + \rho_\eta \cos \eta \hat{\mathbf{r}}_\eta - \rho \cos \eta \hat{\mathbf{r}}).$$

In component form the above leads to

$$\mathbf{n}_s = \rho_\xi \begin{pmatrix} 0 \\ \cos \xi \\ -\sin \xi \end{pmatrix} + \rho_\eta \cos \eta \begin{pmatrix} \cos \eta \\ -\sin \eta \sin \xi \\ -\sin \eta \cos \xi \end{pmatrix}$$
$$- \rho \cos \eta \begin{pmatrix} \sin \eta \\ \cos \eta \sin \xi \\ \cos \eta \cos \xi \end{pmatrix}.$$

The unit normal $\hat{\mathbf{n}}$ can be easily computed from this result.

ACKNOWLEDGMENTS

Support for research in machine vision at the MIT Artificial Intelligence Laboratory is provided in part by the Advanced Research Project Agency of the Department of Defense under Army Contract DACA76-85-C-0010. ALV work at the Hughes AI Center was sponsored by DARPA Contract DACA76-85-C-0007. Harris thanks Mike Daily for helping with the implementation of the motion recovery algorithm.

REFERENCES

1. M. Daily, J. G. Harris, D. Keirsey, K. Olin, D. Payton, K, Reiser, J. Rosenblatt, D. Tseng, and V. Wong, Autonomous cross-country navigation with the ALV, in *IEEE International Conference on Robotics and Automation, Philadelphia, PA, April 24, 1988,* pp. 718–726.
2. R. E. Sampson, 3D range sensor-phase shift detection, *Computer* **20,** 1987, 23–24.
3. M. Hebert and T. Kanada, 3D vision for outdoor navigation by an autonomous vehicle, in *DARPA Image Understanding Workshop, Stanford University, Stanford, CA, April 25, 1988,* pp. 593–601.
4. M. Asada, Building a 3-D world model for a mobile robot from sensory data, in *IEEE International Conference on Robotics and Automation, Philadelphia, PA, April 24, 1988,* pp. 918–923.
5. D., Goldgof, T. Huang, and H. Lee, Feature extraction and terrain matching, in *IEEE Computer Vision and Pattern Recognition Conference, Ann Arbor, MI, June 5, 1988,* pp. 899–904.
6. M., Daily, J. G. Harris, and K. Reiser, An operational perception system for cross-country navigation, in *IEEE Computer Vision and Pattern Recognition Conference, Ann Arbor, MI, June 5, 1988,* pp. 794–802.
7. B. K. P. Horn, Automatic hill-shading and the reflectance map, in *DARPA Image Understanding Workshop, Stanford University, Stanford, CA, April 24–25, 1979,* pp. 79–120.
8. W. E. L. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual System,* MIT Press, Cambridge, MA, 1988.
9. D. Terzopolous, Multilevel computational processes for visual surface reconstruction, *Comput. Vision Graphics Image Process.* **24,** 1983, 52–96.
10. B. K. P. Horn, and B. G. Schunck, Determining optical flow, *Artifi. Intelligence* **16,** 1981, 185–203.
11. B. K. P. Horn, and E. J. Weldon, Jr., Direct methods for recovering motion, *Int. J. Comput. Vision* **2,** 1988, 51–76.
12. B. K. P. Horn, *Robot Vision,* MIT Press, Cambridge, MA & McGraw–Hill, New York, 1986.
13. G. Strang, *Linear Algebra and its Applications,* Academic Press, New York, 1980.
14. B. K. P. Horn, Relative orientation, MIT AI Memo 994, November, 1987.
15. M. J. Daily, J. G. Harris, K. E. Olin, K. Reiser, D. Y. Tseng, and F. M. Vilnrotter, *Knowledge-Based Vision Techniques Annual Technical Report,* US Army ETL, Fort Belvoir, VA, in preparation.
16. B. K. P. Horn, Motion fields are hardly ever ambiguous, *Int. J. Comput. Vision* **1,** 1987, 259–274.
17. B. K. P. Horn, Hill shading and the reflectance map, *Proc. IEEE* **69,** 1981, 14–47.