Reconstruction of 3-D Road Geometry from Images for Autonomous Land Vehicles

KENICHI KANATANI AND KAZUNARI WATANABE

Abstract—**A new algorithm for reconstructing 3-D road geometry from images is presented for the purpose of autonomously navigating land vehicles. The reconstruction is based on an idealized road model: A road is assumed to be generated by a horizontal line segment of a fixed**

Manuscript received June 10, 1988; revised June 2, 1989. This work was supported in part by the Yazaki Memorial Foundation for Science and Technology and the Inamori Foundation. Part of the material in this communication was presented at the IEEE International Workshop on Intelligent Robots and Systems, Tokyo, Japan, October-November 1988.

K. Kanatani is with the Department of Computer Science, Gunma University, Kiryu, Gunma 376, Japan.

K. Watanabe is with NTT Corporation, Uchisaiwai-cho, Chiyoda-ku, Tokyo 100, Japan.

IEEE Log Number 8929805.

length sweeping in the scene. The constraints that ideal road images must satisfy are expressed as a set of *differential equations*; the 3-D road geometry is reconstructed by numerically integrating these differential equations. In order to prevent numerical instability, a correction scheme is proposed for stabilizing computation: At each numerical integration step, the computed solution is modified in such a way that the required constraint is always satisfied. Some examples based on real road images are shown. We also discuss in detail the *inherent ill-posedness* of the problem and related technical issues.

I. Introduction

Recently, considerable attention has been paid to the research and development of autonomous land vehicles (ALV's) [2], [3], [10], [13], [23]-[25]. The ultimate aim of the research is to build vehicles which navigate autonomously by taking video images of the scene ahead, identifying the road, computing the 3-D geometry, and determining the course of navigation. The use of a guidance system, e.g., guiding lines painted on the road surface, makes the vehicle control easier [7], [11], [16], but if the vehicle is to move along an arbitrary road in an uncontrolled environment, we need several sophisticated modules:

- First, an *image analysis module* is necessary for segmenting input images, identifying roads and obstacles, and detecting road boundaries.
- Second, a *geometric reasoning module* is necessary for interpreting the 3-D road geometry.
- Then, a *path planning module* is needed for computing the course of the vehicle with desirable properties (shortest distance, obstacle avoidance, etc.).
- Finally, a *navigation control module* is required for driving the vehicle as planned.

In the past, effort was concentrated on the image analysis module, road boundary detection in particular. The techniques developed so far include color analysis [24], Hough transforms [4], [25], model fitting [23], and high-level reasoning [1], [14]. This communication focuses on the geometric reasoning module. One way to obtain the necessary 3-D data is direct measurement such as range sensing [8], [9], [22]. In this communication, however, we propose an algorithm to reconstruct the 3-D geometry from a *single* image by combining the imaging geometry of perspective projection and an appropriate *model* that idealizes real roads. This approach has the advantage of not requiring any additional devices.

Many of previously proposed methods are based on finding pairs of road boundary segments that are *supposedly parallel* in the scene and then computing the *vanishing points* they define [15], [25]. The computation is very easy if the road is assumed to be either horizontal (but curved) or straight (but not necessarily horizontal). In general, however, it is not easy to find such locally parallel pairs. DeMenthon [5] proposed a discrete numerical method based on the road model proposed by Ozawa and Rosenfeld [18]. His algorithm iteratively computes the 3-D road geometry from a pair of starting points whose 3-D positions are assumed to be known. However, a cubic equation must be solved at each step, and the correct root must be chosen from among the computed three roots. Once a wrong root is chosen, all the remaining steps become meaningless. Sakurai *et al.* [20] proposed a parametric fitting approach by preparing several prototypes of the 3-D road shape. Recently, DeMenthon [6], [17] proposed a new approach based on the assumption that the road is "locally flat," showing that the solution can be computed locally.

The approach presented in this communication is *differential* as opposed to the *discrete* approach of DeMenthon [5] and *nonparametric* as opposed to the *parametric* approach of Sakurai *et al.* [21]. Our approach is as follows:

- We first derive a set of *differential equations* which relate the 3-D road geometry with the projection image.

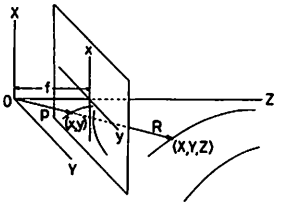

Fig. 1. Point (X, Y, Z) in the scene is projected onto point (x, y) on the image plane $Z = f$ by perspective projection from the viewpoint located at the coordinate origin O.

- Then, we compute the 3-D road shape by *numerically integrating* these differential equations from given initial data.

Thus the solution is unique, and branching never appears.

There arise, however, two issues to be settled. First, we must deal with *computational instability*. The computed solution always satisfies the required constraints if the integration is exact. However, as the numerical integration proceeds, the accumulated error can grow very rapidly, and the computed road shape may no longer satisfy the constrains. In order to avoid this computational instability, we proposed a scheme of *stabilizing correction*: The computed solution is modified at each numerical integration step in such a way that it always satisfies the required constraints. Some examples based on real road images will be shown.

The second issue is more fundamental; it is the *inherent ill-posedness* of the problem. Since roads stretch away from the viewer, the solution is more and more sensitively affected by the image data as the distance from the viewer increases; even a perturbation of one pixel can affect the solution by several kilometers in the distant part of the road. This instability is *inherent* to the problem itself and cannot be avoided even though the computation is exact. Thus the 3-D road reconstruction is an extremely *ill-posed* problem [19], [20]. This point does not seem to have been given full attention in the past [5], [6], [25]. We will discuss this issue in detail, and show that our stabilizing correction scheme works as *regularization* [19], [20].

II. Perspective Projection of the Road Boundary

We use a Cartesian XYZ-coordinate system fixed to the camera. The camera imaging geometry is modeled by Fig. 1: The coordinate origin O corresponds to the center of the camera lens, the Z axis corresponds to the camera optical axis, and the plane $Z = f$ is identified as the image plane. A point in the scene is projected onto the intersection of the image plane with the ray starting from the origin O, which we call the *viewpoint*, and passing through that point. The constant f, which we call the *focal length*, corresponds to the distance between the center of the lens and the surface of the film. We assume that its value is known.

Take an image xy-coordinate system on the image plane so that $(0, 0, f)$ is the image origin, and the x and y axes are, respectively, parallel to the X and Y axes. Then, a point (X, Y, Z) in the scene is projected onto the point (x, y) on the image plane as follows:

$$x = fX/Z \qquad y = fY/Z. \tag{1}$$

Consider two space curves $C_l: (X_l(l), Y_l(l), Z_l(l))$ and $C_r: (X_r(r), Y_r(r), Z_r(r))$ in the scene. Here, l and r are arbitrary parameterizations (not necessarily their arc lengths). Let $c_l: (x_l(l), y_l(l))$ and $c_r: (x_r(r), y_r(r))$ be the plane curves on the image plane resulting from the perspective projection of the space curves C_l and C_r. In the following, we use an overdot to denote differentiation with respect to the parameter involved in the expression. For instance, $\dot{x}_l$ and $\dot{Y}_l$, respectively, mean dx_l/dl and dY_l/dl, while $\dot{y}_r$ and $\dot{Z}_r$, respectively, mean dy_r/dr and dZ_r/dr (so, the same dot means either d/dl or d/dr depending on the parameter involved in the expression). We also use letter s for expressions valid for both l and r. With this convention, we have from (1)

$$X_s = x_s Z_s/f \qquad Y_s = y_s Z_s/f, \qquad s = l, r. \tag{2}$$

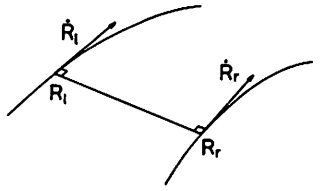

Fig. 2. A road cross segment has a fixed length and intersects with the road boundaries perpendicularly.

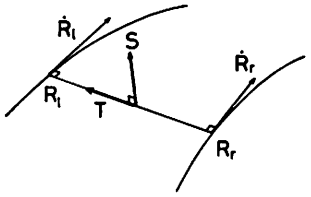

Fig. 3. The vectors $\dot{R}_l$ and $\dot{R}_r$ are tangent to the road boundaries and also orthogonal to both the cross-segment orientation T and the road surface normal S.

By differentiation, we have

$$\dot{X}_s = \frac{1}{f}(Z_s\dot{x}_s + x_s\dot{Z}_s) \qquad \dot{Y}_s = \frac{1}{f}(Z_s\dot{y}_s + y_s\dot{Z}_s), \qquad s = l, r. \tag{3}$$

Let the two space curves C_l and C_r be the two boundaries of a road of constant width, always keeping the same distance between them. In mathematical terms, this constraint is described as follows (Fig. 2):

- There exists a (yet unknown) one-to-one correspondence between the space curves C_l and C_r (i.e., a one-to-one correspondence between the two parameters l and r).
- The two corresponding points keep the same distance.
- The line segment connecting them, which we call the road *cross segment*, meets both space curves perpendicularly.

III. Differential Constraints on the Road Boundary

Let us assume that a spline curve is fitted to each road boundary on the image plane so that the image data x_s, y_s, $\dot{x}_s$, $\dot{y}_s$, $s = l, r$, are always available for arbitrary values of s. Put $R_s = (X_s, Y_s, Z_s)$, $\dot{R}_s = (\dot{X}_s, \dot{Y}_s, \dot{Z}_s)$, $s = l, r$. We want to determine the tangent vectors $\dot{R}_s$, $s = l, r$, assuming that the 3-D shape has already been reconstructed up to points R_s, $s = l, r$. Since (3) give two constraints on $\dot{X}_s$, $\dot{Y}_s$, $\dot{Z}_s$, $s = l, r$, the tangent vectors $\dot{R}_s$, $s = l, r$, are determined if one additional constraint is found. There exist two alternatives:

i) The tangent vectors $\dot{R}_s$, $s = l, r$, must be orthogonal to the cross-segment vector $R_l - R_r$ (Fig. 3). Hence, if we put

$$T \equiv \frac{R_l - R_r}{\|R_l - R_r\|} = \frac{(X_l - X_r, Y_l - Y_r, Z_l - Z_r)}{\sqrt{(X_l - X_r)^2 + (Y_l - Y_r)^2 + (Z_l - Z_r)^2}} \tag{4}$$

we have a constraint $(\dot{R}_s, T) = 0$, $s = l, r$, where $(\cdot, \cdot)$ denotes inner product. If we put $T = (T_1, T_2, T_3)$, this constraint is written as

$$T_1\dot{X}_s + T_2\dot{Y}_s + T_3\dot{Z}_s = 0, \qquad s = l, r. \tag{5}$$

ii) The tangent vectors $\dot{R}_s$, $s = l, r$, must be orthogonal to the *surface normal* to the road (Fig. 3). Let $S = (S_1, S_2, S_3)$ be the unit vector normal to the road surface. Since the 3-D road shape is assumed to have already been reconstructed up to the present position, the road surface normal S at the present position is also known. The resulting constraint is $(\dot{R}_s, S) = 0$, $s = l, r$, or

$$S_1\dot{X}_s + S_2\dot{Y}_s + S_3\dot{Z}_s = 0, \qquad s = l, r. \tag{6}$$

Thus the tangent vectors $\dot{R}_s$, $s = l, r$, must be orthogonal to both T and S. Since (3) are already available, requiring both (5) and (6) is *overspecification*. It seems that one of these can be chosen arbitrarily, but this is not desirable as we can see immediately if we substitute (3) into (5) and (6)

$$(T_1x_s + T_2y_s + T_3f)\dot{Z}_s = -(T_1\dot{x}_s + T_2\dot{y}_s)Z_s, \qquad s = l, r \tag{7}$$

$$(S_1x_s + S_2y_s + S_3f)\dot{Z}_s = -(S_1\dot{x}_s + S_2\dot{y}_s)Z_s, \qquad s = l, r. \tag{8}$$

Note that $p_s = (x_s, y_s, f)$ is the vector indicating the 3-D orientation of the ray starting from the viewpoint O and passing through the point R_s on the road boundary (Fig. 1). Since the left-hand sides of (7) and (8) are, respectively, written as $(p_s, T)\dot{Z}_s$ and $(p_s, S)\dot{Z}_s$, we see that

- (7) is unable to determine $\dot{Z}_s$ at points at which T happens to be orthogonal to p_s;
- (8) is unable to determine $\dot{Z}_s$ at points at which S happens to be orthogonal to p_s.

Moreover, when numerical computation is performed, computational instability occurs at points at which $(p_s, T) \approx 0$ or $(p_s, S) \approx 0$.

Here, we determine $\dot{Z}_s$, $s = l, r$, in such a way that both (7) and (8) are satisfied *on the average* in the sense of least squares. We choose the value of $\dot{Z}_s$ by

$$[(T_1x_s + T_2y_s + T_3f)\dot{Z}_s + (T_1\dot{x}_s + T_2\dot{y}_s)Z_s]^2 + [(S_1x_s + S_2y_s + S_3f)\dot{Z}_s + (S_1\dot{x}_s + S_2\dot{y}_s)Z_s]^2 \to \min. \tag{9}$$

The solution is

$$\dot{Z}_s = -\frac{(T_1x_s + T_2y_s + T_3f)(T_1\dot{x}_s + T_2\dot{y}_s) + (S_1x_s + S_2y_s + S_3f)(S_1\dot{x}_s + S_2\dot{y}_s)}{(S_1x_s + S_2y_s + S_3f)^2 + (T_1x_s + T_2y_s + T_3f)^2}Z_s. \tag{10}$$

The other components $\dot{X}_s$, $\dot{Y}_s$, $s = l, r$, are computed from (3). Thus we have obtained a set of differential equations which expresses $\dot{X}_s$, $\dot{Y}_s$, $\dot{Z}_s$, $s = l, r$, in terms of X_s, Y_s, Z_s, $s = l, r$, and other known quantities.

These differential equations cannot be integrated at this stage, because we do not know the *correspondence between the two parameters* l and r; these parameters were arbitrarily assigned to the road boundary image. Here, we simply relate l with r so that both points R_l and R_r proceed by *the same distance* along the road boundaries when l and r are, respectively, incremented by dl and dr. In other words, we require $\|\dot{R}_l\|dl = \|\dot{R}_r\|dr$, or

$$dr/dl = \|\dot{R}_l\|/\|\dot{R}_r\|. \tag{11}$$

If the differential relations for $\dot{R}_l$ and $\dot{R}_r$, which were obtained in the previous section, are substituted, the right-hand side is expressed in terms of X_s, Y_s, Z_s, $s = l, r$, and other known quantities. Thus the above equation enables us to keep track of the correspondence between l and r. Since X_l, Y_l, X_r, Y_r are expressed in terms of Z_l and Z_r by (2), we obtain a set of three differential equations in the form

$$\frac{dr}{dl} = F_0(l, r, Z_l, Z_r)$$
$$\frac{dZ_l}{dl} = F_l(l, r, Z_l, Z_r)$$
$$\frac{dZ_r}{dl} = F_r(l, r, Z_l, Z_r). \tag{12}$$

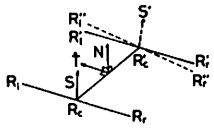

Fig. 4. Correction of the computed road cross segment.

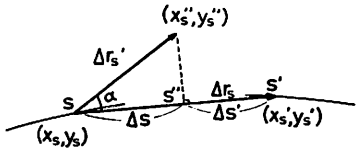

Fig. 5. Correction of the road boundary parameter.

(The right-hand side of the third equation is $\dot{Z}_r dr/dl$.) The 3-D road shape is determined by numerically integrating these equations with respect to parameter l from known initial values $Z_l(l_0)$, $Z_r(r_0)$, r_0 at $l = l_0$.

IV. Correction of the Solution

The constraint (11) holds only approximately; it is exact only when the road is not curved horizontally (but can be bent up and down). If the road is curved, the point on the outer boundary must proceed by a longer distance than the corresponding point on the inner boundary. The reason why the approximation (11) is used is that the resulting solution is only approximate even if the exact constraint is used. Since the differential equations must be solved numerically, the growth of accumulated error is inevitable. As a result, the constraints that the exact solution should satisfy may not hold for the computed solution.

Our strategy here is as follows: At each step of the numerical integration, we *correct* the solution in such a way that it necessarily satisfies the required constraints. Let $\boldsymbol{R}_l$ and $\boldsymbol{R}_r$ be the two corresponding positions on the road boundary whose 3-D positions have already been established by the preceding step (or given as initial data). By the inductive hypothesis, the cross-segment vector $\boldsymbol{R}_l - \boldsymbol{R}_r$ is assumed to be horizontal. Apply one step of numerical integration, and let $\boldsymbol{R}_l', \boldsymbol{R}_r'$ be the computed positions. Compute the *center points*

$$\boldsymbol{R}_c = \frac{1}{2}(\boldsymbol{R}_l + \boldsymbol{R}_r) \qquad \boldsymbol{R}_c' = \frac{1}{2}(\boldsymbol{R}_l' + \boldsymbol{R}_r'). \tag{13}$$

We correct the computed positions $\boldsymbol{R}_l'$ and $\boldsymbol{R}_r'$ into new positions $\boldsymbol{R}_l''$ and $\boldsymbol{R}_r''$ in such a way that the following four conditions are all satisfied (Fig. 4).

- The midpoint of $\boldsymbol{R}_l''$ and $\boldsymbol{R}_r''$ is $\boldsymbol{R}_c'$

$$\frac{1}{2}(\boldsymbol{R}_l'' + \boldsymbol{R}_r'') = \boldsymbol{R}_c'. \tag{14}$$

- The cross-segment vector $\boldsymbol{R}_l'' - \boldsymbol{R}_r''$ is horizontal

$$(\boldsymbol{R}_l'' - \boldsymbol{R}_r'', \boldsymbol{N}) = 0. \tag{15}$$

Here, $\boldsymbol{N}$ is the unit vector indicating the vertical direction.

- The cross-segment vector $\boldsymbol{R}_l'' - \boldsymbol{R}_r''$ has the same length as the cross-segment vector $\boldsymbol{R}_l - \boldsymbol{R}_r$

$$\|\boldsymbol{R}_l - \boldsymbol{R}_r\| = \|\boldsymbol{R}_l'' - \boldsymbol{R}_r''\|. \tag{16}$$

- The *average* cross-segment vector $[(\boldsymbol{R}_l - \boldsymbol{R}_r) + (\boldsymbol{R}_l'' - \boldsymbol{R}_r'')]/2$ is orthogonal to the *center-line vector* $\boldsymbol{R}_c' - \boldsymbol{R}_c$

$$\left(\frac{(\boldsymbol{R}_l - \boldsymbol{R}_r) + (\boldsymbol{R}_l'' - \boldsymbol{R}_r'')}{2}, \boldsymbol{R}_c' - \boldsymbol{R}_c\right) = 0. \tag{17}$$

Define

$$\hat{\boldsymbol{T}} = \frac{\boldsymbol{N} \times (\boldsymbol{R}_c' - \boldsymbol{R}_c)}{\|\boldsymbol{N} \times (\boldsymbol{R}_c' - \boldsymbol{R}_c)\|}. \tag{18}$$

This is the unit vector orthogonal to both $\boldsymbol{N}$ and $\boldsymbol{R}_c' - \boldsymbol{R}_c$. Hence, it is horizontal and orthogonal to the center-line vector. We can easily confirm that the above conditions are all satisfied if we define

$$\begin{aligned} \boldsymbol{R}_l'' &= \boldsymbol{R}_c' - (\boldsymbol{R}_l - \boldsymbol{R}_c) + 2(\boldsymbol{R}_l - \boldsymbol{R}_c, \hat{\boldsymbol{T}})\hat{\boldsymbol{T}} \\ \boldsymbol{R}_r'' &= \boldsymbol{R}_c' + (\boldsymbol{R}_l - \boldsymbol{R}_c) - 2(\boldsymbol{R}_l - \boldsymbol{R}_c, \hat{\boldsymbol{T}})\hat{\boldsymbol{T}}. \end{aligned} \tag{19}$$

The numerical integration requires the road surface normal $\boldsymbol{S}$. Strictly speaking, the surface normal $\boldsymbol{S}$ differs from position to position. For example, the surface normal at one endpoint $\boldsymbol{R}_l$ of a road cross segment may be different from that at the other endpoint $\boldsymbol{R}_r$. However, since the change of the surface normal $\boldsymbol{S}$ affects the solution only mildly in the sense of least square, we fix it at the midpoint $\boldsymbol{R}_c$ for one computation step. The road surface normal $\boldsymbol{S}'$ at $\boldsymbol{R}_c'$ to be used in the next computation step is set to

$$\boldsymbol{S}' = \frac{(\boldsymbol{R}_c' - \boldsymbol{R}_c) \times \hat{\boldsymbol{T}}}{\|(\boldsymbol{R}_c' - \boldsymbol{R}_c) \times \hat{\boldsymbol{T}}\|}. \tag{20}$$

If the predicted position $\boldsymbol{R}_s'$ is corrected into the new position $\boldsymbol{R}_s''$, the corresponding parameter value of $s = l, r$ must also be corrected. Let (x_s, y_s) and (x_s', y_s') be the image points corresponding to the original positions $\boldsymbol{R}_s$ and $\boldsymbol{R}_s'$, respectively, and let (x_s'', y_s'') be the image point corresponding to the corrected position $\boldsymbol{R}_s''$. The image point (x_s'', y_s'') may not necessarily be located on the road boundary curve c_s. We associate with this point the parameter of the closest point on the road boundary curve as follows. For $s = l, r$, put

$$\Delta \boldsymbol{r}_s \equiv (x_s' - x_s, y_s' - y_s) \qquad \Delta \boldsymbol{r}_s' \equiv (x_s'' - x_s, y_s'' - y_s) \tag{21}$$

(see Fig. 5). The length of $\Delta \boldsymbol{r}_s'$ projected onto $\Delta \boldsymbol{r}_s$ is given by

$$\Delta s = \|\boldsymbol{r}_s'\| \cos \alpha = (\Delta \boldsymbol{r}_s', \Delta \boldsymbol{r}_s)/\|\Delta \boldsymbol{r}_s\| \tag{22}$$

where α is the angle made by $\Delta \boldsymbol{r}_s$ and $\Delta \boldsymbol{r}_s'$. Let

$$\Delta s' = \|\Delta \boldsymbol{r}_s\| - \Delta s. \tag{23}$$

Let s and s' be the respective parameter values of points (x_s, y_s) and (x_s', y_s'). We associate with the points (x_s'', y_s''), $s = l, r$, the following parameter values:

$$s'' = \frac{s'\Delta s + s\Delta s'}{\Delta s + \Delta s'}, \qquad s = l, r. \tag{24}$$

(Note that this formula works correctly even if Δs and/or $\Delta s'$ are zero or negative.)

V. Examples

We took several video images of real roads near Gunma University in Kiryu, Gunma, Japan. The road boundaries were detected by tracing the white lines painted on the road, and spline curves were fitted to them. Since the techniques of road boundary detection and spline fitting are not the issues of this communication, we omit the details.

In order to start the numerical integration, we need initial data to start from. The most straightforward way is the direct measurement by range sensing or stereo. This is not difficult, since the measurement need be applied to only the part of the road immediately in front of the viewer. (Recently, DeMenthon [6], [17] proposed a simple procedure of using a single image based on the detection of vanishing points.) Once the road reconstruction process starts, the current location is predicted from the already reconstructed road geometry and the history of navigation.

In our experiment, we simply "back-projected" the road part near the viewer onto a hypothetical horizontal ground (assuming that the height of the viewer is known) and detected a cross segment that intersects both boundaries nearly perpendicularly. We omit the details. Then, the simultaneous differential equations (12) were numerically integrated by the Runge-Kutta method with respect to the independent variable l, and the correction scheme of Section IV was applied at each step.

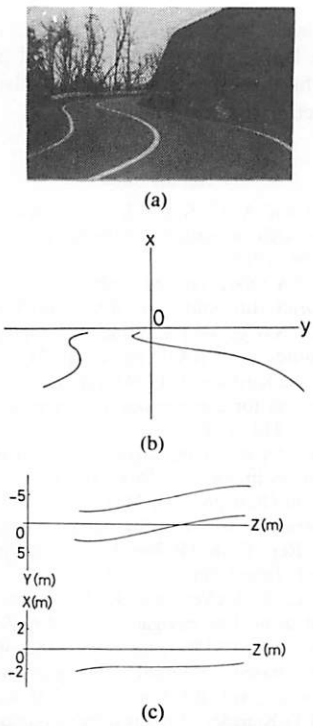

Fig. 6. (a) An image of a road. (b) Spline fitting to the road boundary. (c) The 3-D road shape reconstructed by numerical integration.

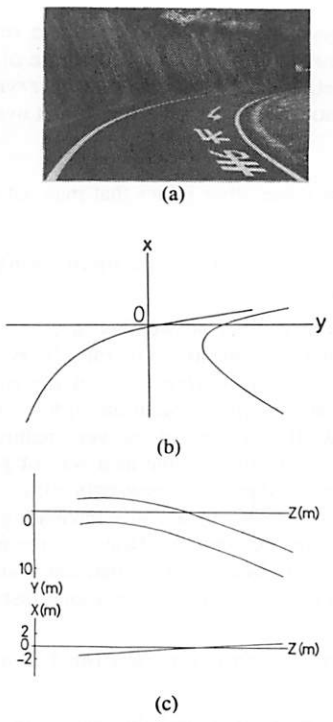

Fig. 7. (a) An image of a road. (b) Spline fitting to the road boundary. (c) The 3-D road shape reconstructed by numerical integration.

Since the integration step Δl for the Runge–Kutta method can be arbitrarily adjusted in the course of integration, it is desirable that the step Δl is chosen so that the corresponding interval ΔZ_c along the center line $\boldsymbol{R}_c$ is kept nearly the same. The relation

$$\frac{dZ_c}{dl} = \frac{d((Z_l + Z_r)/2)}{dl} = \frac{F_l(l, r, Z_l, Z_r) + F_r(l, r, Z_l, Z_r)}{2} \tag{25}$$

suggests that a reasonable choice is

$$\Delta l \approx \frac{dl}{dZ_c}\Delta Z_c = \frac{2\Delta Z_c}{F_l(l, r, Z_l, Z_r) + F_r(l, r, Z_l, Z_r)}. \tag{26}$$

Fig. 6(a) is an original road image. After the spline fitting, the road boundaries shown in Fig. 6(b) are obtained. Fig. 6(c) indicates the 3-D road shape reconstructed by numerical integration. The *top view* (orthographic projection onto the YZ plane) is shown above, and the *side view* (orthographic projection onto the ZX plane) is shown below. Fig. 7 shows another example, and the result is similarly arranged. We can see that the 3-D road shape can be reconstructed fairly well.

VI. Inherent Ill-Posedness of the Problem

A very crucial issue has been revealed by our numerical experiments. It is the *inherent ill-posedness* of the problem. This issue is at the heart of the 3-D road geometry recovery problem, but due attention does not seem to have been paid to it in the past.

In general, a straightforward way to reconstruct the 3-D geometry of an object from its projection image is to first *back-project* the image, i.e., construct a family of infinitely many candidate shapes that all yield the observed image when projected. From among these candidates, we choose one that satisfies required constraints by invoking *a priori* knowledge about the object. (Kanatani [12] called this approach the *3-D Euclidean approach* and gave a general discussion on this issue.)

The peculiarity of 3-D road reconstruction, as compared with other types of 3-D object shape reconstruction, is the fact that road extends *away* from the viewer and *covers a very long distance*. (In other reconstruction problems, the object is assumed to be located within a relatively narrow range of distance from the viewer.) If the above stated approach is taken, the reconstructed shape becomes very sensitively affected by the image data as the distance from the viewer increases; even a perturbation of one pixel will affect the reconstructed shape by many kilometers in the distance. This phenomenon is *inherent* to the problem itself, and cannot be avoided whichever scheme we use and however accurately the computation is performed. In other words, the problem is *inherently ill-posed* [19], [20].

The only possible way to cope with this inherent ill-posedness is to *give priority to the knowledge about the object*, i.e., *ignore* the image data if they are not compatible with the knowledge. In the road reconstruction problem, the balance of weight between the image data and the knowledge must be changed according to the distance from the viewer: We should relatively faithfully follow the image data (i.e., the spline curves of the detected road boundary) for the part near the viewer, while we should depend on the knowledge (i.e., the road model) more and more heavily as the distance from the viewer increases. This strategy may be called *regularization* [19], [20] but is very peculiar to the road reconstruction problem.

The scheme we have presented in this communication automatically materializes this strategy. Consider (9), for instance. This greatly stabilizes computation, because the second term forces the solution to be smoothly connected to the already reconstructed part. We can say that the solution is given *inertia*. Consider the correction scheme (19). This forces the solution to satisfy the road model constraint, but at the same time we are *ignoring* the image data; the projection of the corrected road shape does not necessarily agree with the image data (see Fig. 5). In short, the image data only exert a *force*. Even if the spline curves turn abruptly, the reconstructed solution proceeds smoothly. In the extremely long distance, the reconstruction proceeds far and far ahead indefinitely by completely ignoring the image data.

After many experiments, we have observed that the reconstruction goes very smoothly even if the image data are very noisy. We have also tried various existing methods, but all failed (the reconstructed 3-D shape is extremely unrealistic, or completely wrong when the true geometry is known, even if the computation does not break down) *when the viewpoint is very low*. This vulnerability has been little reported in the past because the road shape was reconstructed either over a short distance [13], [21], [23]–[25] (assuming that that

was sufficient to navigate the vehicle) or from a very high viewpoint looking down the road [5], [6]. The root cause of this vulnerability is that all these methods *faithfully rely on observed image data*. In contrast, our method works for a low viewpoint over a long distance.

VII. Discussions

We now discuss some other issues that may naturally be raised to our scheme.

- *Is the 3-D shape reconstructed by invoking a priori knowledge accurate?*

We cannot give a general answer because accuracy of the solution depends on not only accuracy of the observed image data but also *accuracy of the knowledge* (e.g., if the road does not have a constant width, the computed solution with a constant width cannot be accurate at all). In view of the very nature of the problem, we should rather regard the scheme as a way of *guessing* (or more formally *inference* or *hypothesis generation*); its accuracy must be tested by other means (e.g., by actually navigating the vehicle). The same applies to human perception. (How "accurate" is the judgment of a man driving a car about the 3-D road shape ahead?) In general, the reconstruction is very accurate over a long distance if the road is straight.

- *Do we really need to reconstruct the 3-D road shape over a long distance?*

For actual navigation, it is probably sufficient to reconstruct the part immediately in front of the vehicle. However, prediction of the road geometry over a long distance is useful not only for actual navigation but also for many other planning purposes.

- *Can the reconstruction be carried out in real time as the vehicle navigates?*

We cannot answer this because it depends on actual implementation techniques. (Our experiments did not focus on speedup, so spline fitting was recomputed every time it became necessary.) It also depends on the interval length of numerical integration. In general, the reconstruction stage is very simple and quick, because all we need is computation of explicitly given algebraic expressions; no searches and iterations are necessary. Much more time is required for road boundary detection. If the 3-D reconstruction is intended for planning purposes other than actual navigation, the computation need not be done in real time.

VIII. Concluding Remarks

We have presented a scheme for recovering the 3-D road geometry from a single image. The basic principle is to describe the relationship between the 3-D road geometry and its 2-D image in terms of *differential equations* and then reconstruct the 3-D road shape by *numerical integration*. The process is completely *automatic*, and the solution is *unique*; no searches or iterations are necessary.

We introduced techniques to stabilize the computation—the least squares scheme to mix different types of constraints and the correction scheme to force the solution to satisfy required constraints. Examples based on real road images have also been shown.

As the purpose of this work was to establish the validity of our techniques and the experiment was performed on a small floppy-based computer, no attempt was made to measure computation times; we felt they would have little value considering the hardware used.

We have also given a detailed discussion on the very peculiar nature of the 3-D road shape reconstruction problem—its *inherent ill-posedness*. This is the core of the problem *regardless of the solution technique*, but it has not been fully understood in the past. We have pointed out that our stabilizing techniques work as *regularization*, yielding a stable solution over a long distance. We have also discussed some technical issues relevant to actual applications to ALV systems.

Acknowledgment

This work was motivated by the ALV project of the University of Maryland, where the first author (Kanatani) stayed during 1985–1986. He wishes to thank A. Rosenfeld, L. Davis, and D. DeMenthon of the University of Maryland and S. Ozawa of Keio University for helpful discussions. The authors also wish to thank C. Koyama for conducting the real image analysis.

References

[1] K. M. Andress and A. C. Kak, "Evidence accumulation and flow of control in a hierarchical spatial reasoning system," *AI Mag.*, vol. 9, no. 2, pp. 75–94, 1988.

[2] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robotics Automat.*, vol. RA-2, no. 1, pp. 14–23, 1986.

[3] J. L. Crowley, "Navigation for an intelligent mobile robot," *IEEE J. Robotics Automat.*, vol. RA-1, no. 1, pp. 21–41, 1985.

[4] L. S. Davis, T. R. Kushner, J. LeMoigne, and A. M. Waxman, "Road boundary detection for autonomous vehicle navigation," *Opt. Eng.*, vol. 25, pp. 409–414, 1986.

[5] D. DeMenthon, "A zero-bank algorithm for inverse perspective of a road from a single image," in *Proc. IEEE Int. Conf. on Robotics and Automation* (Releigh, NC, Mar./Apr. 1987), pp. 1444–1449.

[6] ——, "Reconstruction of a road by matching edge points in the road image," Tech. Rep. CAR-TR-368, Center for Automation Research, Univ. Maryland, June 1988.

[7] K. C. Drake, E. S. McVey, and R. M. Inigo, "Sensing error for a mobile robot using line navigation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 4, pp. 485–490, 1985.

[8] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE J. Robotics Automat.*, vol. RA-1, no. 1, pp. 31–41, 1987.

[9] M. Hebert and T. Kanade, "First results on outdoor scene analysis using range data," in *Proc. DARPA Image Understanding Workshop* (Miami Beach, FL, Dec. 1985), pp. 224–231.

[10] R. M. Inigo, E. S. McVey, B. J. Berger, and M. J. Wirts, "Machine vision applied to vehicle guidance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 6, pp. 820–826, 1984.

[11] S. Ishikawa, H. Kuwamoto, and S. Ozawa, "Visual navigation of an autonomous vehicle using white line recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 5, pp. 743–749, 1988.

[12] K. Kanatani, "3D Euclidean versus 2D non-Euclidean: Two approaches to 3D recovery from images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 3, pp. 329–333, 1989.

[13] D. Kuan, G. Phipps, and A.-C. Hsueh, "Autonomous robotic vehicle road following," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 5, pp. 648–654, 1988.

[14] J. LeMoigne, "Domain-dependent reasoning for visual navigation of roadways," *IEEE J. Robotics Automat.*, vol. 4, no. 4, pp. 419–427, 1988.

[15] S.-P. Liou and R. C. Jain, "Road following using vanishing points," *Comput. Vision Graph. Image Process.*, vol. 39, pp. 116–130, 1987.

[16] E. S. McVey, K. C. Drake, and R. M. Inigo, "Range measurements by a mobile robot using a navigation line," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 1, pp. 105–109, 1986.

[17] D. Morgenthaler, S. Hennessy, and D. DeMenthon, "Range-video fusion and comparison of inverse perspective algorithms," *IEEE Trans. Syst., Man, Cybern.* (to be published).

[18] S. Ozawa and A. Rosenfeld, "Synthesis of a road image as seen from a vehicle," *Pattern Recogn.*, vol. 19, pp. 123–145, 1986.

[19] T. Poggio and C. Koch, "Ill-posed problems in early vision: from computational theory to analogue networks," *Proc. Roy. Soc. Lond.*, Ser. B, vol. 226, pp. 303–323, 1985.

[20] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, pp. 314–319, 1985.

[21] K. Sakurai, H. Zen, H. Ohta, Y. Ushioda, and S. Ozawa, "Analysis of a road image as seen from a vehicle," in *Proc. 1st IEEE Int. Conf. on Computer Vision* London, UK, June 1987, pp. 651–656.

[22] U. K. Sharma and L. S. Davis, "Road boundary detection in range imagery for an autonomous robot," *IEEE J. Robotics Automat.*, vol. 4, no. 5, pp. 515–523, 1988.

[23] C. Thorp, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation of the Carnegie-Mellon Navlab," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 10, no. 3, pp. 362–373, 1988.

[24] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "VITS—A vision system for autonomous land vehicle navigation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 3, pp. 342–361, 1988.

[25] A. M. Waxman, J. LeMoigne, L. S. Davis, B. Srinivasan, T. Kushner, E. Liang, and T. Siddalingaiah, "A visual navigation system for autonomous land vehicles," *IEEE J. Robotics Automat.*, vol. RA-3, no. 2, pp. 124–141, 1987.