International Journal of Computer Vision, 9:1, 55–76 (1992)

Panoramic Representation for Route Recognition by a Mobile Robot

JIANG YU ZHENG* AND SABURO TSUJI
Department of Control Engineering, Osaka University, Toyonaka, Osaka 560, Japan

Received

Abstract
Here, we explore a new theme: *Route recognition*, in robot navigation. It is faced with problems of visual sensing, spatial memory construction, and scene recognition in a global world. The strategy of this work is route description from experience, that is, a robot acquires a route description from route views taken in a trial move, and then uses it to guide the navigation along the same route. In cognition phase, a new representation of scenes along a route termed *panoramic representation* is proposed. This representation is obtained by scanning sideviews along the route, which provides rich information such as 2D projections of scenes called *Panoramic view* and *generalized panoramic view*, a path-oriented 2½D sketch, and a path description, but only contains a small amount of data. The continuous *panoramic view* (PV) and *generalized panoramic view* (GPV) are efficient in processing, compared with fusing discrete views into a complete route model. In recognition phase, the robot matches the panoramic representation memorized in the trial move and that from incoming images so that it can locate and orient itself. We employ dynamic programming and *circular dynamic programming* in coarse matching of GPVs and PVs, and employ feature matching in fine verification. The advantage of wide fields of GPV and PV brings a reliable result to the scene recognition.

1 Introduction

1.1 Objective

Much research on vision-based navigation has been focused on road-following and obstacle-avoidance, the aim of which is to drive a vehicle safely within a road in an outdoor environment [2, 3, 4, 5], as well as within a free space in a room or in a building [6, 7, 8] based on sensor data. For longer-distance navigation, however, robots will also be confronted with another problem: how to understand the route it travels. This problem includes unexplored issues of how a robot can represent and memorize its environment while moving around, and how it can recognize the scenes so as to locate and orient itself. The work described here tackles these issues.

In the outdoor environment, two kinds of representation can be considered. One is a bird's-eye view map and another is a series of route views. The robot can determine its way from either of them [9]. Although the bird's-eye view such as an aerial photograph or a city map is good for representing global relationships of routes and is straightforward in route planning, it is hard to transform into real route views for recognition. A robot using a terrain map to understand its environment has been proposed [10]. Another approach is to use route views. Landmarks specified by humans are employed in guiding the navigation [11].

We consider the robot is more intelligent if it can acquire route information autonomously by representing and memorizing scenes along a path it has wandered and looked around. The strategy of this work is route description from experience. A sequence of ground-based views are acquired when traversing routes, and they are further analyzed to obtain a map for guidance of the robot [12]. The scenario of the project is as follows: A robot moves along a certain route under the guidance of a human and autonomously memorizes the route. The robot is then commanded to pursue the same route by itself. It keeps observing the scene and locates

*Current address: ATR Communication Systems Research Laboratories, 2-2 Hikaridai, Seika, Soraku, Kyoto 619-02, Japan.

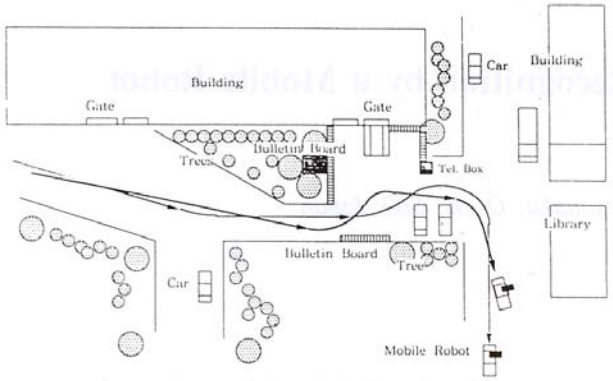

Fig. 1. A mobile robot continuously views scenes along the route by a camera. It autonomously builds the model along the route, which guides the autonomous navigation along the same route. A route in the campus of Osaka University shown in this figure is used for the experiments.

and orients itself by referring to the memorized route description so that it can instruct the road-following module where it should change its direction or stop. Figure 1 illustrates a part of such a route.

1.2 Scene Representation

The first key step to bridge local scenes along a route to a global map is, perhaps, the encoding of the numerous ground-based episodic views into a sequentially organized description that is easy to access in further operations. In this article, a representation of scenes called *panoramic representation* is introduced for memorizing and retriving the visual information acquired in the trial move [1]. It provides the major information along the route, such as 2D projections of scenes, a path-oriented 2½D sketch, and a path description. It can be used not only as an intermediate representation as Marr proposed [14], from which a more abstract and symbolic representation of the route can be built [32], but also as a description referred directly to in route recognition [13].

In order to represent a wide field of view that contains global information with less data amount, we propose a *panoramic view*, which has the following two types:

— *Panoramic view* (PV): a projection of scenes to a cylindrical retina centered at a stationary point, used as a visual memory of a place for determining location and orientation of the robot.

— *Generalized panoramic view* (GPV): a projection of scenes along a route to a surface of generalized cylinder determined by a smooth camera path along the route. It is a visual memory of the route used for locating the robot.

Both PV and GPV are obtained from integration of numerous fine slit views dynamically projected onto the image frame through a vertical slit, as the camera swirls and moves along the smooth path respectively [15, 16]. We will see that generating the panoramic views is equivalent to cutting a vertical slice in the spatiotemporal volume proposed by Bolles and Heeger [17, 18]. For linear motion, Bolles and Baker cut epipolar-plane images in the volume along optical-flow directions to determine image velocities [19]. Our slice cut in the volume is, on the contrary, nonparallel to optical flow so that it yields a projection showing 2D shapes, rather than traces of feature points.

The 2½D sketch, described by the horizontal image velocities of features in the GPV, is extracted by measuring time delays of these features moving across a pair of parallel slits in the image frame. This is done spatially on the GPVs simultaneously generated from the double slits, because the panoramic view contains an axis presenting time. Possessing front views of scenes and their 2½D descriptions, the panoramic representation maintains the major information in the spatiotemporal volume from our camera motion.

1.3 Route Recognition

In route recognition, the robot recalls route memory constructed in its experienced move to allocate its position and orientation so as to plan the moving distance and direction. The objective of this work is to quantitatively match two panoramic representations obtained at different moves along the same route, assuming the road-following process keeps the robot path within roads. We also match two panoramic views at close positions so that the robot can determine its orientation.

Utilizing the sequential and continuous characteristics of the panoramic views, we employ *dynamic programming* [20] and *circular dynamic programming* (a modified dynamic programming for periodical scenes) to match 1D color projections of GPVs and PVs, respectively, in establishing coarse correspondence. Then some distinct structure with various properties such as

shape, color, and depth in the 2D projections are verified in detail. The changes in 2D shape due to the changes in paths are adjusted by using the 2½D information acquired in establishing the representation. The implementation of matching is efficient to cope with the real-time recognition in autonomous navigation. Because of the wide field of PV and GPV, the matching can start from a fairly global level to avoid failure in changed local parts and, thus, results in a high reliability.

Since the matching of panoramic representation is still on iconic level, we have to deal with the influences from changes in path and partial changes due to dynamic objects. We discuss several approaches to modify the panoramic representations from different moves to closer states.

In route pursuit, there are two ways to guide the road-following process for the next two sitatuions. As the robot moves along a road, its heading can be restricted within a small changeable extent in the forward direction. The route-recognition task is only to find specified positions at which to stop or turn according to the memorized generalized panoramic view. This kind of route recognition is usually employed in the situation where the destination may possibly be occluded by objects intervening. On the other hand, if the robot enters a wider area such as a square, there can be more movable spaces, free headings, and good visibility. By referring to the goal object memorized in the previous PV, the robot can find its correspondent in the current panoramic view and approach it by adjusting its heading.

2 Panoramic View

2.1 Formation of Panoramic Views

A possible method in modeling a wide environment is to direct a camera toward various parts and take a lot of discrete images. However, the spatial relationship between these images is not easy to obtain, except when they share certain common fields of view. The drawbacks of this consideration are the data redundancy and the two-dimensional discontinuity in the overlapped sights, because of the difference of view points and viewing directions.

In achieving the objective that the robot can observe scenes in a larger scope and at various locations, our camera continuously scans scenes through a vertical slit. Connecting these one-dimensional slit views along the time axis results in a wide two-dimensional display which contains only a small amount of data. Figure 2 illustrate the schemes of image formation.

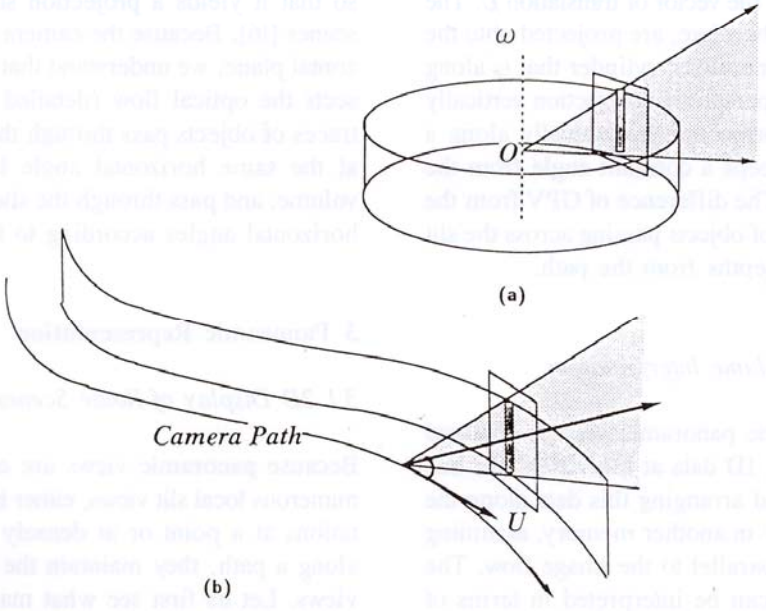

Fig. 2. Schemes for generating panoramic views. (a) A panoramic view is acquired by arranging slit imagery taken by a swiveling camera. (b) A generalized panoramic view is acquired from slit imagery taken by a camera moving along a smooth path. The optical axis is aligned with a constant angle from the tangent of the path (direction of translation U).

Suppose a camera rotates around a vertical axis at point O at a constant angular velocity ω. It takes pictures continuously through a vertical slit as figure 2(a) shows. By pasting the slit views consecutively at the angle of observation, we get a *panoramic view* which is, in fact, a cylindrical retina on which all objects visible from point O are projected. The velocity of each image point passing across the slit is the same because the rotation center coincides with the camera focus.

Let us generalize it to the situation when the camera moves along a smooth path S on the horizontal plane. Assume that the camera has a translation U changed by an angular velocity ω on the horizontal plane. In this case, the camera path becomes a circular curve. We set the camera axis I so that it keeps a constant angle with respect to the translation U. The path is defined as either concave $(-)$ or convex $(+)$, if its center of curvature is on the same side, or on the different side of the U with the camera axis. If there exists no rotation except translation, the camera will move on a straight line (the center of curvature of the path is at infinity). A smooth camera path, hence, can be divided into these three types of segments.

Under these conditions, if the camera takes continuous images of 3D scenes through a vertical slit, a *generalized panoramic view* can be constructed by connecting the 1D slit views from successive frames along the path, as depicted in figure 2(b). The slit is restricted not be be coplanar with the vector of translation U. The scenes at the sideway, therefore, are projected onto the vertical surface of a generalized cylinder that is along the path. It utilizes the perspective projection vertically and an orthographic projection horizontally along a dynamic axis which keeps a constant angle from the tangent of the path U. The difference of GPV from the PV is that the velocity of objects passing across the slit line depend on their depths from the path.

2.2 Spatiotemporal Volume Interpretation

In the real situation, the panoramic view is obtained by continuously taking 1D data at the virtual slit line in each input image and arranging this data along the time axis consecutively in another memory, assuming the slit line is set nonparallel to the image flow. The result of this process can be interpreted in terms of spatiotemporal volume analysis [18, 19].

A *spatiotemporal volume* is composed of a dense sequence of images of dynamic scenes. Filtering out the traces of image patterns in the volume, one can find the displacements of features in a three-dimensional spatial way. The spatiotemporal volume retains all of the information observable by a moving camera, but it is too redundant for the given motions. Our analysis is designed, as described previously, to arrange views along the path into an easy-to-memorize representation, rather than a 3D Euclidean model suffering from inherent large errors in depth. As shown in figure 3(a), the result of capturing data on the slit line and arranging them along the time axis is equivalent to cutting a vertical $Y - T$ slice in the volume. The data amount of panoramic view, hence, becomes a slice in the whole volume. Figure 3(b) and (c) each show the EPI images and a GPV in linear camera motion which is the only motion with which the EPI can cope in acquiring 3D structure from given camera motion [19]. As we will see in section 3.2, the GPVs can recover most of the 3D points in our more general motion when the camera has a translation changed with a rotation.

Bolles' technique utilizes the linear motion of a camera. They cut epipolar-plane image (EPI) along the optical-flow directions in the spatiotemporal volume [17, 19], which gives both traces of feature points and spatiotemporal events such as occlusion. It is used to achieve a precise estimation of image velocity through redundant computation. The panoramic view cut in the volume is, on the contrary, nonparallel to the optical flow so that it yields a projection showing 2D shapes of scenes [16]. Because the camera notion is on the horizontal plane, we understand that the vertical slit intersects the optical flow (detailed in section 3.2). The traces of objects pass through the vertical slice of PV at the same horizontal angle in the spatiotemporal volume, and pass through the slice of GPV at different horizontal angles according to the object depth.

3 Panoramic Representation

3.1 2D Display of Route Scenes

Because panoramic views are created by connecting numerous local slit views, either in finely divided orientations at a point or at densely distributed positions along a path, they maintain the 2D continuity of the views. Let us first see what mathematical properties the panoramic views have for different camera movements. Linear and circular paths are considered, because we can approximate a smooth path by their

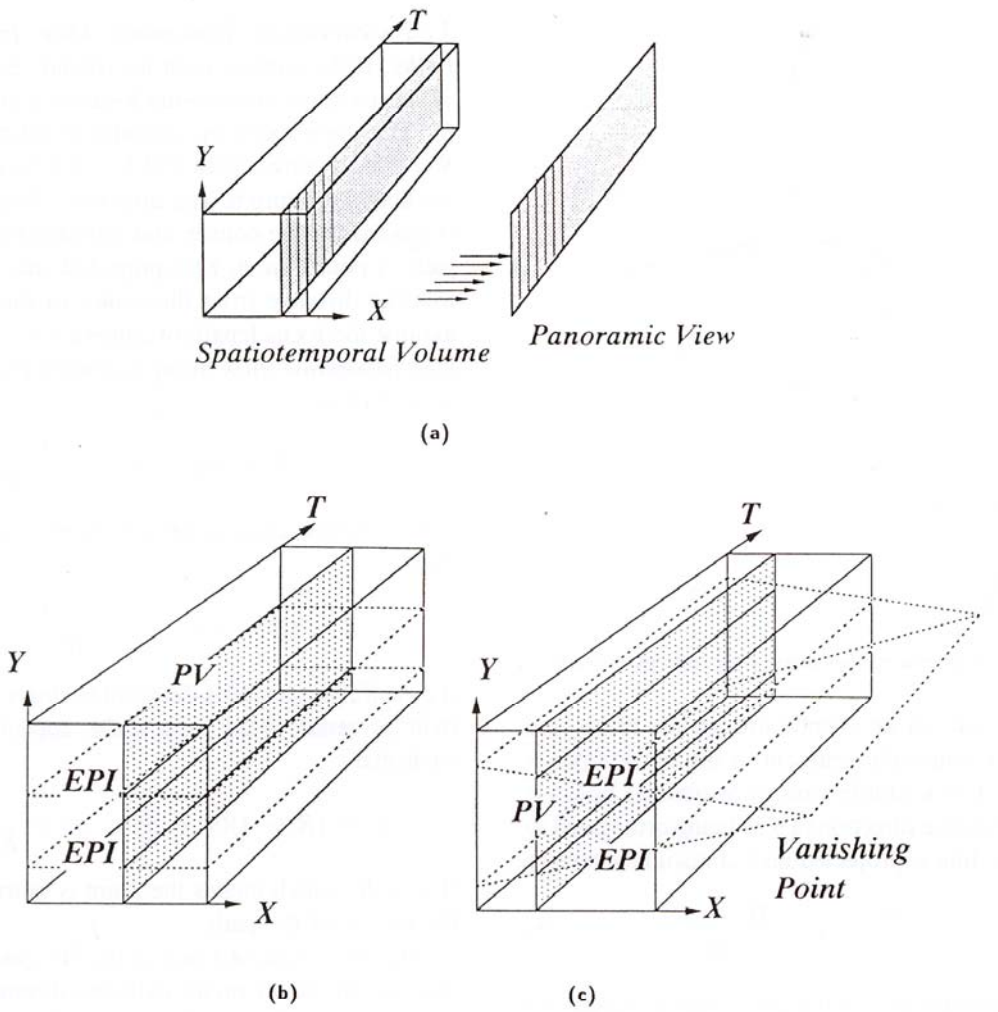

Fig. 3. Spatiotemporal volume interpretation of the panoramic views. (a) The panoramic view is corresponding to a slice in the volume nonparallel to the optical flow. (b) Panoramic views and epipolar-plane images in linear motion. The camera axis is perpendicular to the translation. $X - T$ slices are EPIs and $Y - T$ slices are panoramic views. (c) A free camera direction. Slices plassing through the vanishing point are EPIs and slices not passing through the vanishing point are generalized panoramic views in the volume.

segments. For simplicity of description and computation, the following analyses assume that the camera axis is aligned with path normals (perpendicular to translation) and its motions are ideal and known in generating GPV.

A common property of PV and GPVs for linear and circular paths is that vertical lines in the 3D world appear also as vertical lines in the panoramic views, because of the assumption that the vertical slit moves on a horizontal plane.

3.1.1 Panoramic View. For the basic scheme of panoramic viewing illustrated in figure 2(a), the camera motion is pure rotation. The image formation is modeled by a cylindrical projection shown in figure 4. Let us represent the 3D space using a cylindrical coordinate system $(O - \rho\theta Y)$, so that the origin O is at the camera focus and the Y axis is aligned with the rotation axis. The θ represents a swiveling angle from a reference direction. Projection of a point $P(\rho, \theta, Y)$ to the O yields an image point at (s, y) on the cylindrical retina whose radius is the focal length of camera f. Without losing generality, we set $f = 1$ and get

$$s = \theta, \qquad y = \frac{Y}{\rho} \tag{1}$$

The projection of a nonvertical line L onto the cylindrical retina is determined by the plane through O and

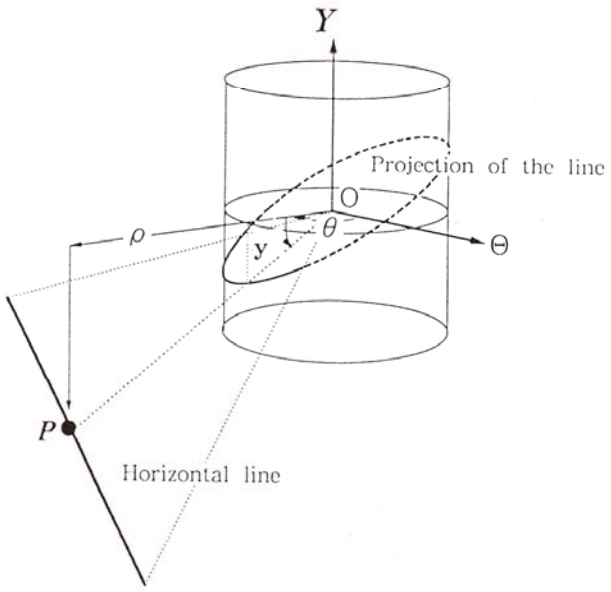

Fig. 4. Model of generating panoramic view from pure rotation.

L, which results in an elliptic arc on the retina and appears as a sinusoidal segment in the opened panoramic view. For a simple case of horizontal line, we select the reference direction of θ as being orthogonal to the line. The line is projected on a sinusoidal curve as

$$-\frac{\pi}{2} < s < \frac{\pi}{2} \qquad y = \frac{H \cos(s)}{D} \tag{2}$$

where D is the distance of the line from the Y axis and H is the height of the line.

Figure 5 displays a panoramic view of an outdoor environment taken by a camera on a robot, showing that 2D patterns appear distorted in the image. It is considered as the minimum two-dimensional data set of scenes around, because the input image taken at any angle is easily recovered from it.

3.1.2 Generalized Panoramic View from Circular Paths. If the camera path is circular, each point in a 3D space maps to a moving focus on a circle of radius R. The corresponding cylindrical retina has radius $R + f$ for a convex path and $R - f$ for a concave path, as shown in figure 6. The direction of the camera axis is inward for the convex and outward for the concave path. A point $P(\rho, \theta, Y)$ is projected onto $p(s, y)$, where ρ is the distance from the center of the paths. If we assume the focus length of camera $f = 1$, the generalized panoramic view along a convex circular path (if $\rho > R$) has,

$$s = (R + 1)\theta \qquad y = \frac{Y}{\rho - R} \tag{3}$$

For a concave circular path, P appears at $p(s, y)$ such that

$$s = (R - 1)\theta \qquad y = \frac{Y}{R - \rho} \tag{4}$$

if $\rho < R$, which means the point is closer to the camera than the center of the path. It also appears at $p'(s', y')$ such that

$$s' = (R - 1)(\theta + \pi) \qquad y' = \frac{Y}{R + \rho} \tag{5}$$

if $\rho > R$, which means the point is more distant than the center of the path.

The projection of a line in the 3D space is no more than an elliptic arc on the cylindrical retina. By selecting the same reference direction of θ as for panoramic view, a horizontal line is mapped onto a curve as follows:

For a convex circular path, the line is projected to one of the following ranges

$$\text{if } D > R \qquad -\frac{\pi}{2} < \theta < \frac{\pi}{2} \tag{6a}$$

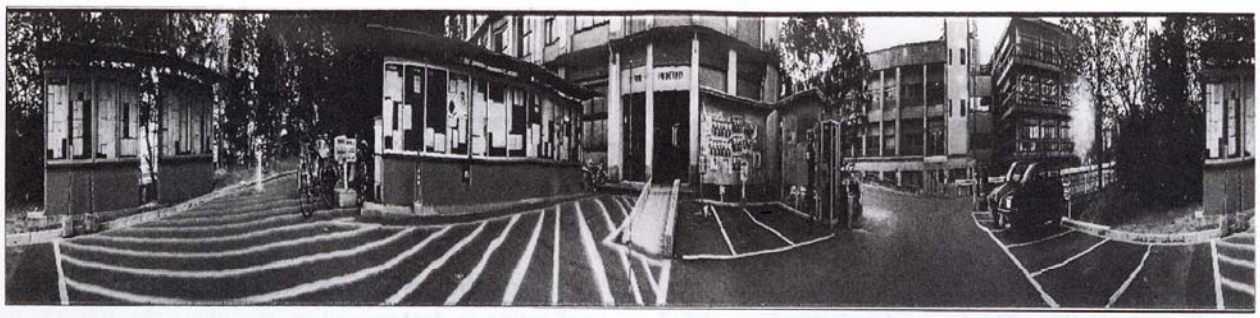

Fig. 5. Panoramic view taken by a camera on a robot turning at a corner.

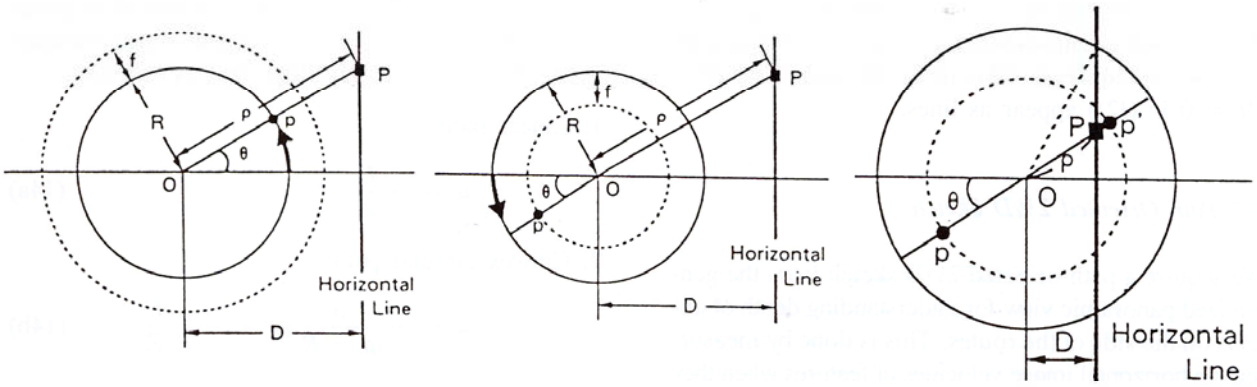

Fig. 6. Panoramic viewing from a convex circular path (left) and a concave circular path when the object is farther from (middle) and nearer to (right) the camera than the center of the circular path.

$$\text{if } D < R, \quad -\frac{\pi}{2} < \theta < -\cos^{-1}\frac{D}{R},$$

$$\text{and } \cos^{-1}\frac{D}{R} < \theta < \frac{\pi}{2} \tag{6b}$$

as

$$y = \frac{H\cos\theta}{D - R\cos\theta} \tag{6c}$$

where $\theta = s/(R + 1)$.

For a concave circular path, the line appears in the range

$$-\frac{\pi}{2} < \theta < \frac{\pi}{2} \tag{7a}$$

as

$$y = \frac{H\cos\theta}{D - R\cos\theta} \tag{7b}$$

and if $D < R$, the line is projected also to another range,

$$\pi - \cos^{-1}\frac{D}{R} < \theta < \pi + \cos^{-1}\frac{D}{R} \tag{8a}$$

as

$$y = \frac{H\cos\theta}{D + R\cos\theta} \tag{8b}$$

3.1.3 Generalized Panoramic View from a Linear Path. Suppose the camera moves along a linear path as figure 7 shows. We set a Cartesian coordinate system XYZ, with its origin O at the camera focus and the X and Z axes pointed along the motion direction and the camera axis, respectively. The scenes, therefore, are

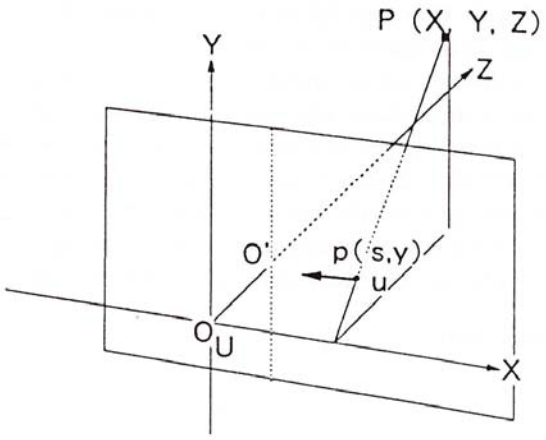

Fig. 7. Panoramic viewing from a linear path.

projected onto the GPV by an orthogonal projection horizontally and a perspective projection vertically. A point $P(X, Y, Z)$ is projected onto $p(s, y)$ in the generalized panoramic view such that

$$s = X \qquad y = \frac{Y}{Z} \tag{9}$$

assuming $f = 1$. Then a 3D line specified by the following equations

$$\frac{X - X_0}{A} = \frac{Y - Y_0}{B} = \frac{Z - Z_0}{C} \tag{10}$$

appears in the GPV as

$$y = \frac{AY_0 + B(s - X_0)}{AZ_0 + C(s - X_0)} \tag{11}$$

If the line is horizontal ($B = 0$)

$$y = \frac{H}{AX_0 + C(s - X_0)} \tag{12}$$

where $H = AY_0$. Most horizontal lines in the space are thus mapped as curves while those parallel to the path ($C = 0$ in equation (12)) or at the same elevation ($H = 0$ in (12)) appear as lines.

3.2 Path-Oriented 2½D Sketch

We acquire a path-oriented 2½D sketch from the generalized panoramic view for understanding depth of objects on the side of the routes. This is done by measuring the horizontal image velocities of features when they pass through the vertical slit, and attaching them to the corresponding parts in the 2D projection.

3.2.1 Image Velocities Along Different Paths. We explore the relationship between the image velocities of features through the slit and different paths. Let U and ω be linear and angular velocities of the camera, respectively, with the camera axis perpendicular to the translation. Let $P(X, Y, Z)$ denote a point in a camera-centered coordinates system. The horizontal and vertical image velocities u and v of its projection $p(x', y')$ along different paths can be easily deduced as follows [21]:

1. Linear path

$$u = -\frac{U}{Z}, \qquad v = 0 \tag{13a}$$

2. Convex circular path

$$u = -\frac{U}{Z} - (1 + x'^2)\omega = -\frac{U}{Z} - (1 + x'^2)\frac{U}{R}$$

$$v = -x'y'\omega = -x'y'\frac{U}{R} \tag{13b}$$

3. Concave circular path

$$u = -\frac{U}{Z} + (1 + x'^2)\omega = -\frac{U}{Z} + (1 + x'^2)\frac{U}{R}$$

$$v = x'y'\omega = x'y'\frac{U}{R} \tag{13c}$$

assuming the focus length of camera $f = 1$.

For any vertical slit at coordinate x' on the frame, the horizontal image velocity u will not be zero on it (13), except for the feature on a 3D vertical cylinder described by $Z = (1 + x'^2)R$ when the camera moves along a concave path (13c). This means the vertical slit intersects the optical flow in the frame.

If the vertical slit line is set at the center of the image ($x' = 0$ in (13)), the nonzero horizontal image velocities at the slit can be simply described as follows:

1. Linear path:

$$u = -\frac{U}{Z} \tag{14a}$$

2. Convex circular path:

$$u = -\frac{\rho}{\rho - R}\omega = -\frac{\rho}{R}\frac{U}{Z} \tag{14b}$$

where $\rho = R + Z$.

3. Concave circular path: If the object point is nearer from the camera than the center of the circular path,

$$u = -\frac{\rho}{R - \rho}\omega = -\frac{\rho}{R}\frac{U}{Z} \tag{14c}$$

where $\rho = R - Z$.

If the object point is farther than the center to the path,

$$u = \frac{\rho}{R + \rho}\omega = \frac{\rho}{R}\frac{U}{Z} \tag{14d}$$

where $\rho = Z - R$.

Object velocities in the image are inversely proportional to their depths, as shown by (14). The image velocity increases by a factor of ρ/R more than that in the linear camera motion, if the path is convex. For a concave path, the velocity decreases by the same factor. As illustrated intuitively in figure 8, the points nearer to the center $O1$ have different moving direction from the camera motion, while the points more distant from the center $O1$ have the same direction as the camera motion.

3.2.2 Acquiring Image Velocities at the Center Line. Now that we have generated a nonredundant representation of scenes, it is beneficial to estimate the image velocity from the generalized panoramic views. Let us assume a stable robot motion such that both motion class and velocity are invariant, at least for a short period. The horizontal image velocity of a feature on the slit line is estimated as its average velocity between two vertical slits placed symmetrically to the slit line. If we generate two GPVs from the double slits simultaneously, we can find two projections of the feature point on them with different t coordinates along the time axes in the GPVs. The time for the point to pass through the given distance between the double slits can be

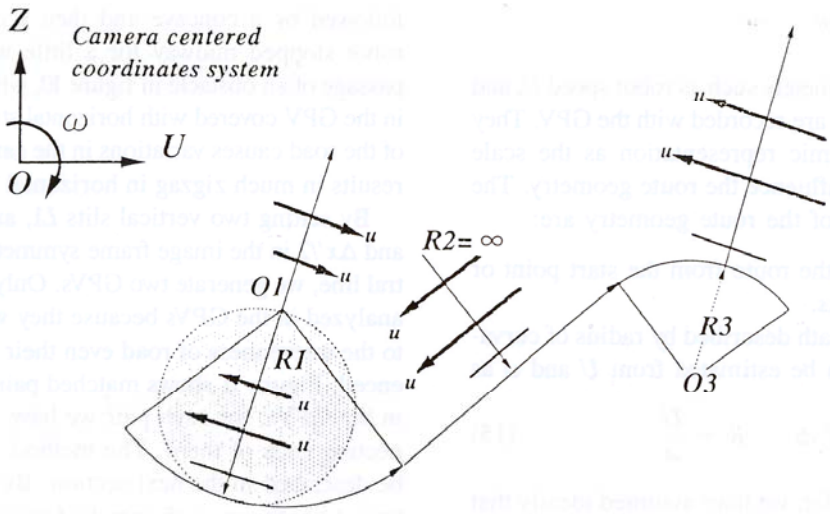

Fig. 8. Image velocity changes when camera moves on different paths. The features in the shaded circle have different horizontal image velocity direction from that out of the circle.

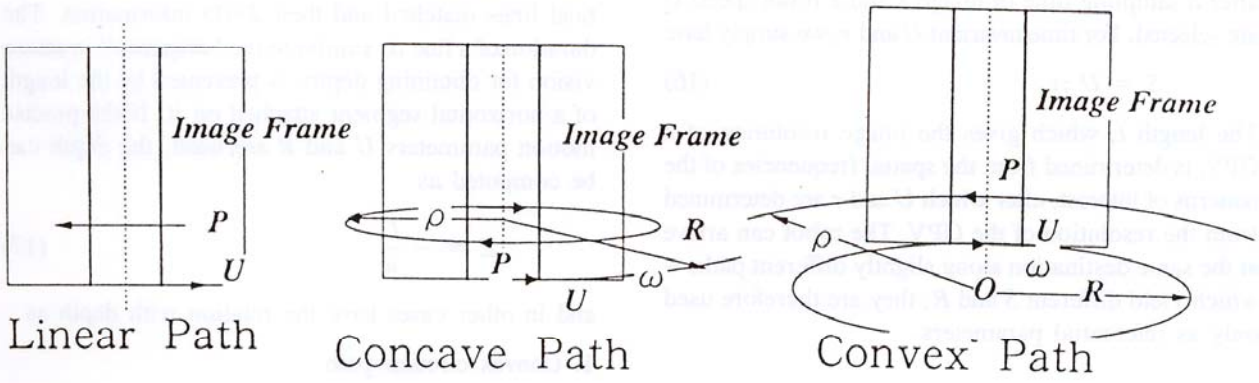

Fig. 9. Trace of a point in the field of view when GPVs from linear, concave, and convex paths are generated.

obtained in a spatial way by looking at the time delay between two projections of the point in the generated GPVs. The problem becomes one of matching feature points in two GPVs generated from the double slits. Fortunately, we find that setting two sampling lines symmetric to the central line provides an important constraint for matching feature points in two generated GPVs in order to compute the image velocity at the central line.

Figure 9 intuitively shows the basic traces of a point P in the image frames for different paths. Since the camera axis is directed or inversely directed to the center of the circular or linear paths (aligned with the normal of the path), the point P moves on circles or a line relative to camera-centered coordinate system. It has traces as ellipses or a line symmetric to the central line in the image. Therefore, it passes across the double slits at the same height. We have the following constraint for matching feature points in GPVs from linear or circular paths.

C1: *If we generate two GPVs from a pair of parallel slits symmetric to the central vertical line, the projections of a point P in the two GPVs have the same y coordinates.*

This also can be noticed from (13), since the absolute value of v and the value of u are symmetric to the center line ($x' = 0$), and v is zero on it.

In principle, we can recover depth of a feature point from two view points. The GPVs taken with the camera slits pointed in different directions really are orthographic stereo.

3.3 Route Description

Image formation parameters such as robot speed U, and the angular velocity ω are recorded with the GPV. They influence the panoramic representation as the scale changes but do not influence the route geometry. The intrinsic parameters of the route geometry are:

— Distance S along the route from the start point or selected landmarks.
— Curvature of the path described by radius of curvature R. These can be estimated from U and ω as

$$S = \int U\,dt \qquad R = \frac{U}{\omega} \tag{15}$$

In the analyses so far, we have assumed ideally that the GPV has the same length with S. However, in a real situation, the panoramic view generalized from dynamic sampling on a slit line has a length t scaled, after a sampling time of image τ and a robot speed U are selected. For time invariant U and τ, we simply have

$$S = U\,\tau t \tag{16}$$

The length t, which gives the image resolution of a GPV, is determined from the spatial frequencies of the patterns of interest, after which U and τ are determined from the resolution of the GPV. The robot can arrive at the same destination along slightly different paths—which yield different S and R; they are therefore used only as referential parameters.

4 Acquiring Stable Panoramic Representations

4.1 Acquiring Panoramic Representation from the Real World

Experiments were made on the route shown in figure 1, using a mobile robot (an automatic guided vehicle for factory automation). A color TV camera on the robot can be swiveled on a rotating table for the panoramic view. An on-board microcomputer, which controls the robot and camera motion, communicates with a Sun4 workstation with an image processor.

Figure 10 (see color section, page 65) shows an example of a generalized panoramic view of size 2048× 128 pixels acquired while the robot moved about 100 m. The camera axis is set simply perpendicular to the forward direction of the robot. The sampling rate was constant and the robot consists of an almost linear path followed by a concave and then a convex path. The robot stopped midway for a little while, waiting for passage of an obstacle in figure 10, which yields an area in the GPV covered with horizontal stripes. Unevenness of the road causes variations in the camera pitch, which results in much zigzag in horizontal lines in the GPV.

By setting two vertical slits $L1$, and $L2$ at $-\Delta x'/2$, and $\Delta x'/2$ in the image frame symmetrically to the central line, we generate two GPVs. Only vertical lines are analyzed in the GPVs because they will not break due to the unevenness of road even their heights are influenced. Figure 11 shows matched pairs of vertical lines in the GPVs; for each pair we have another line connecting ends of them. The method for matching will be described in the next section. By finding the duration Δt between each matched pair in the GPVs, we obtain the time interval needed for the line in penetrating $L1$, $L2$. The horizontal image velocity u at the center line is computed from $\Delta x'/\Delta t$. Figure 12 shows the vertical lines matched and their 2½D information. The duration of a line Δt similar to the "disparity" in stereo vision for obtaining depth, is presented by the length of a horizontal segment attached on it. If the precise motion parameters U and R are used, the depth can be computed as

$$Z = -\frac{U}{u} \tag{17}$$

and in other cases have the relation with depth as

1. Convex circular path

$$z = -\frac{RU}{uR + U} \tag{18a}$$

2. Point nearer than the center of concave circular path ($u < 0$)

$$z = -\frac{RU}{uR - U} \tag{18b}$$

3. Point farther than the center of concave circular path ($u > 0$)

$$z = \frac{RU}{uR - U} \tag{18c}$$

We do not intend to use the robot speed U and ω to estimate the position of arrival because of the error accumulation. Our attempt is, practically, to build a more flexible route model in which approximate geometry is described. The robot locates itself by referring to the scenes on the route, assuming it can move along

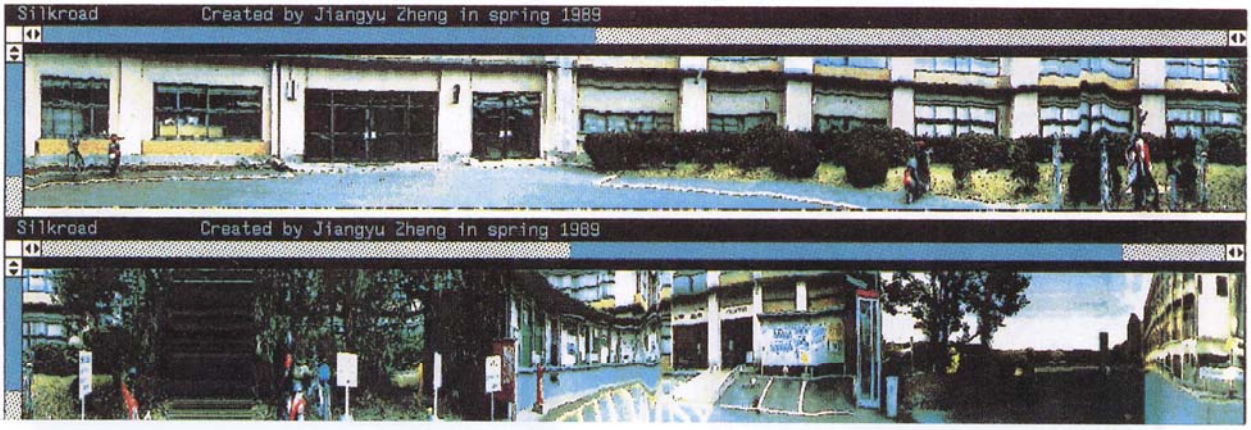

Fig. 10. Panoramic view obtained along the route shown in figure 1.

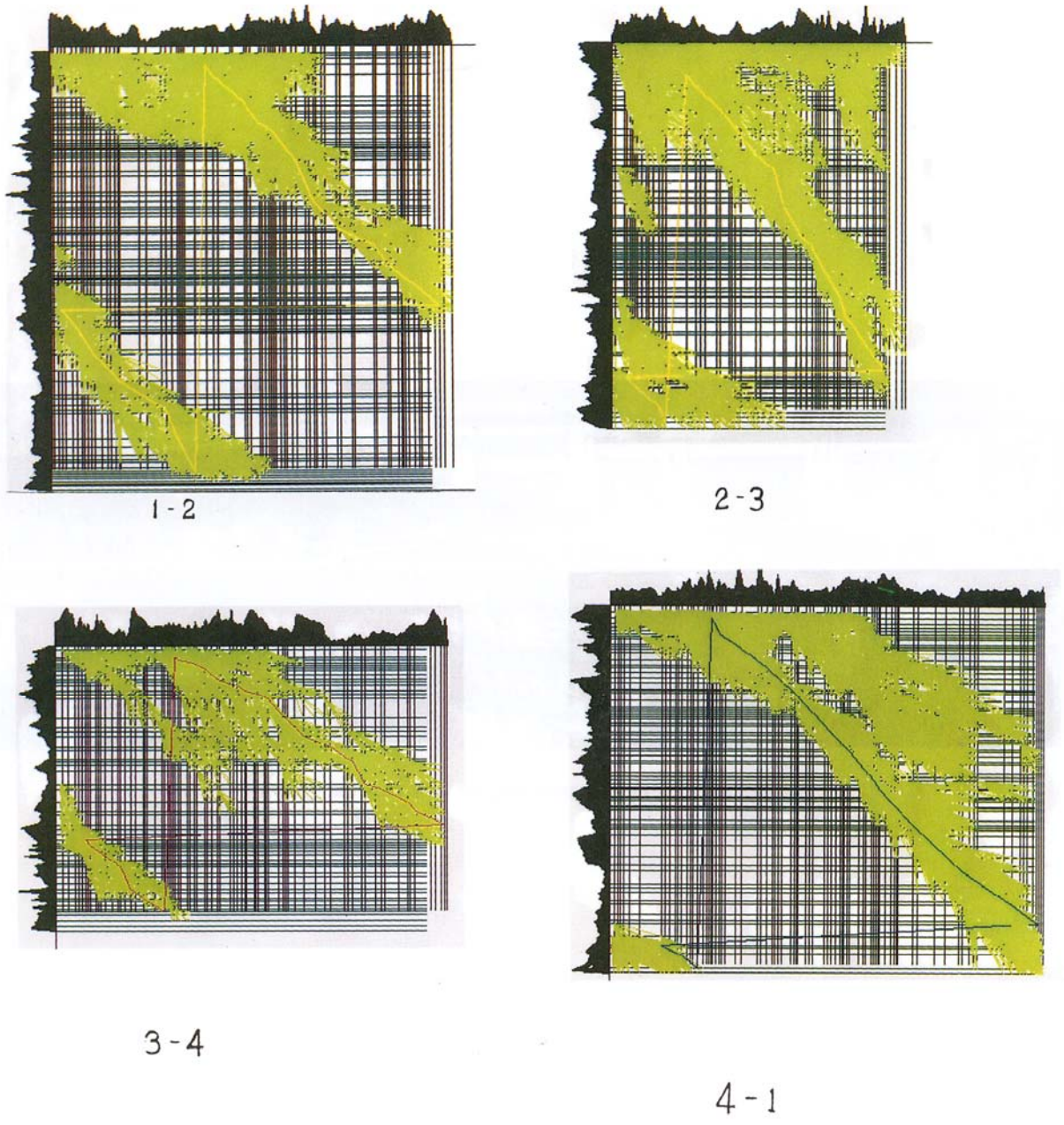

Fig. 19. Matching spaces of the PVs 1–2, 2–3, 3–4, and 4–1 in which the closed curves represent the optimal correspondences.

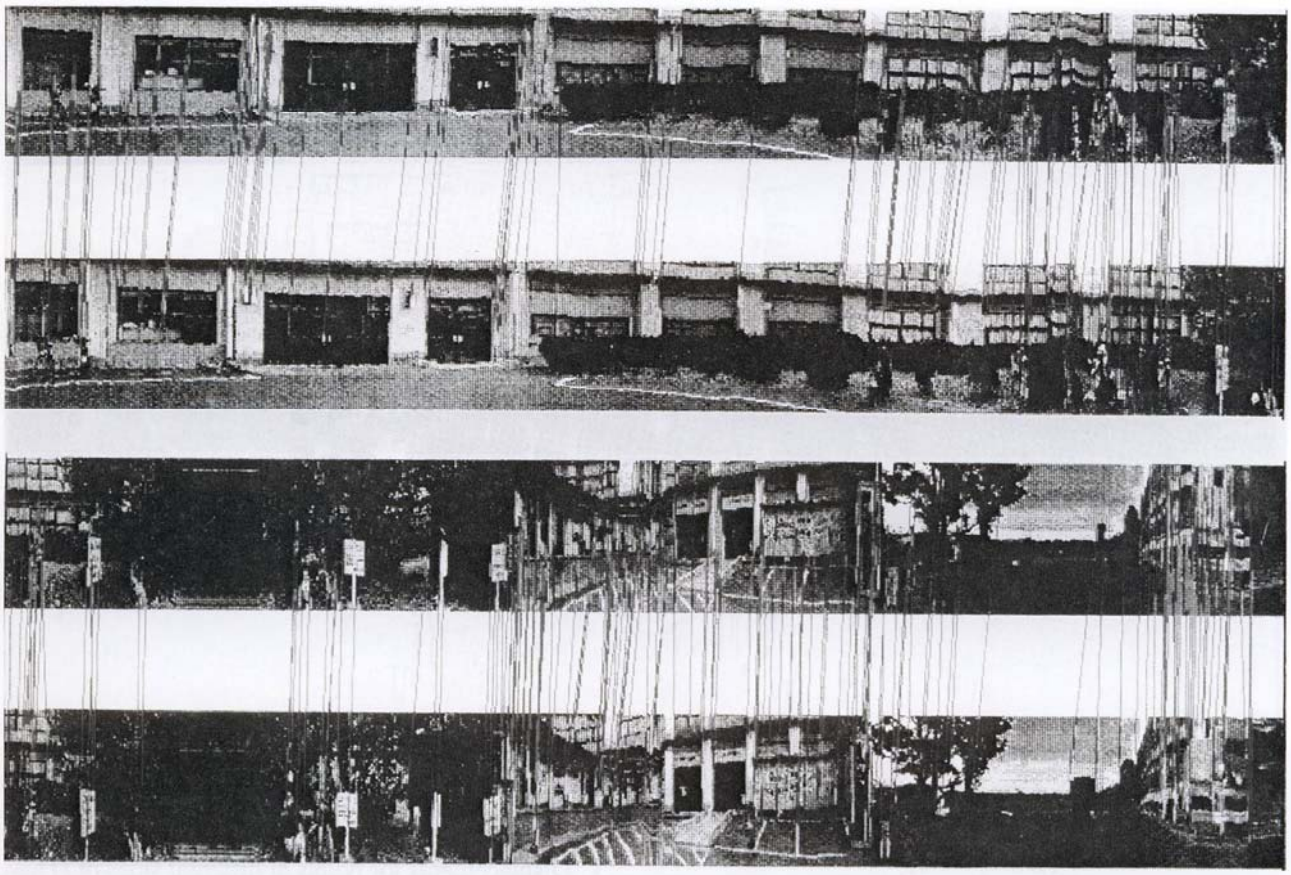

Fig. 11. Matching vertical line segments between the GPVs generated from the same images.

an almost identical path. The following are defined for instructing a road-following process to pursue the memorized route.

1. Break point, where the curvature of the path exceeds a certain threshold and is considered as a corner to turn; it is attached to the GPV. A qualitative direction such as *leftward* or *rightward* is assigned. The entire route is thus divided into subroutes connected at these break points.
2. Complex break point, where an open area or more than three branches of road are ahead, to which a panoramic view with the notation of continued branch is attached for determining direction in further move.

4.2 Shape Change in PV at Different Positions

In order to explore the matching of scenes observed in different navigations, we study the changes of panoramic views caused by small changes in the path.

As the robot autonomously arrives at some specific site where the PV was registered, it builds a new PV for selecting route. However, the positions of the two PVs are usually not coincident. Given two PVs generated at positions O and O' apart from each other by a distance D, we can analyze the displacements of 3D points in PVs. Points P_i, $i = 1, \ldots, n$ in the 3D space are denoted as $P(\rho_i, \theta_i, Y_i)$ and $P'(\rho_i', \theta_i', Y_i')$ in the two coordinate systems centered at O and O', with the line linking the centers of the systems being used as the reference axis. Their projections in both PVs are denoted as $p(\theta_i, y)$ and $p(\theta_i', y')$ as will be shown in figure 14. We have

$$Y_i = Y_i' \qquad \frac{y_i'}{y_i} = \frac{\rho_i}{\rho_i'} = \frac{\sin \theta_i}{\sin (\Delta\theta_i + \theta_i)} \tag{19}$$

Overlapping the coordinates systems O on O', we notice that the distribution of point displacements $\Delta\theta_i = \theta_i - \theta_i'$ has the following properties (see figure 13):

1. The points in the direction close to the D have smaller displacements than that perpendicular to the D (zero in the D direction), if their distances from the O are equal.

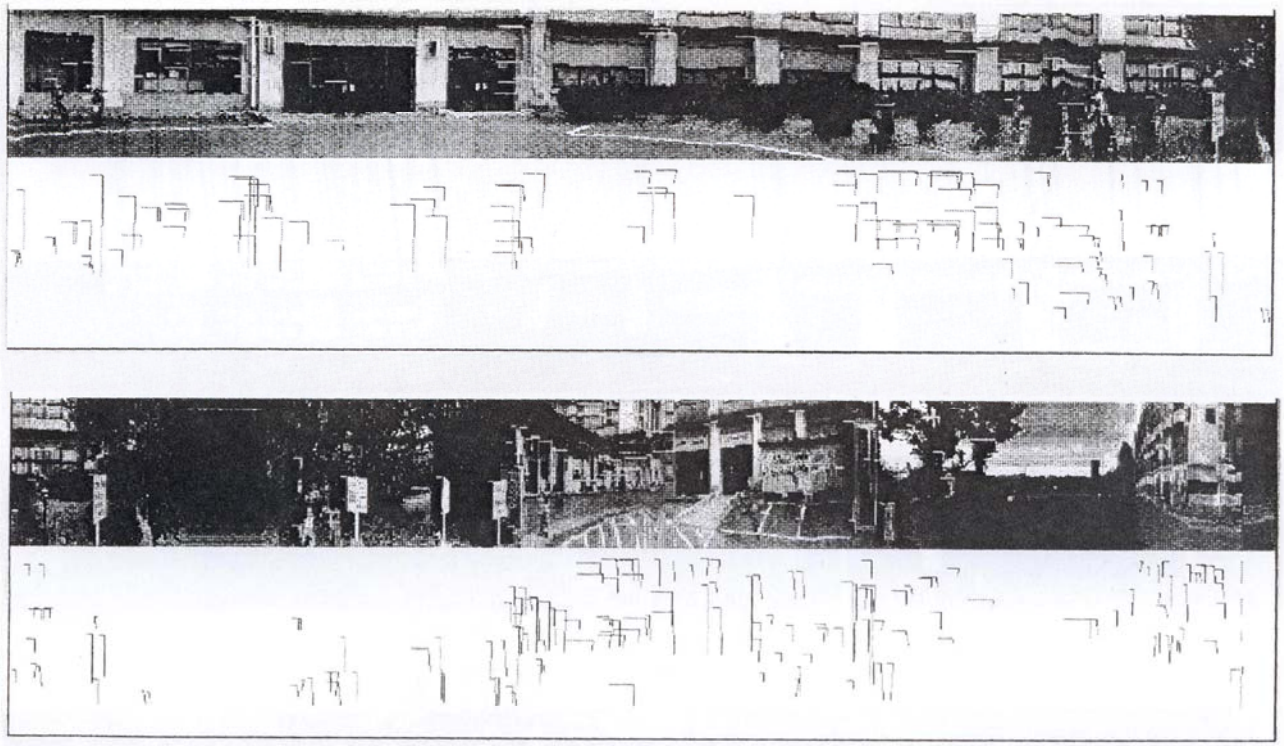

Fig. 12. Generalized panoramic view and vertical lines with depth information. The horizontal lines attached to the vertical lines in GPV indicate their delays between the two generalized panoramic views in figure 11.

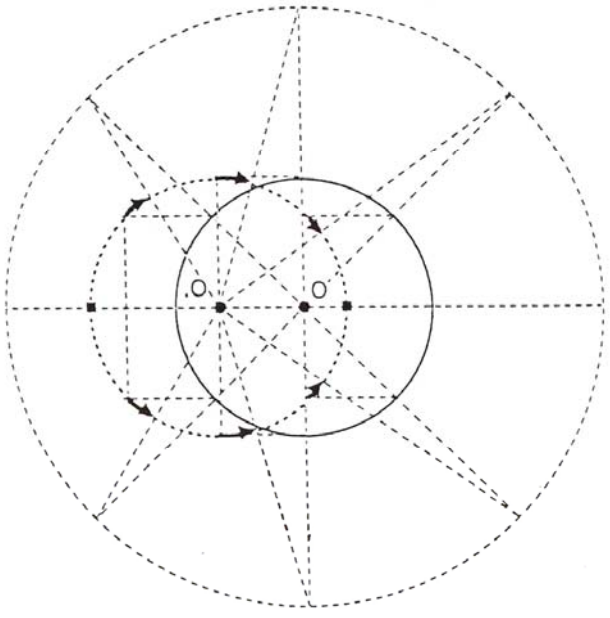

Fig. 13. Displacements of points between two PVs at different positions.

2. Distant points have smaller displacements (zero for points at infinity) than close ones in the same direction.
3. Displacements for points at different sides of D have different directions.

The y displacement of a point is on a sinusoidal curve in the PV as section 3.1.1 indicates.

In PVs, the depth ρ_i, ρ_i', the distances D between any two positions, are unknown and the robot orientations considered as the beginning positions of PVs may be different.

4.3 Shape Changes in GPV from Different Paths

4.3.1 Change of Radius of Circular Path. From (1)–(5) and (13), we can deduce that a point $p(s, y)$ in a GPV from a circular path of radius R can be mapped onto $p'(s', y')$ in a GPV created from another circular path of radius R', but with the same center. The relation between them can be given

$$s' = \frac{s(R' + 1)}{R + 1} \tag{20a}$$

$$y' = \frac{R\omega y}{R'(\omega + u) - Ru} \tag{20b}$$

for convex paths, and

$$s' = \frac{s(R' - 1)}{R - 1} \tag{21a}$$

$$y' = \frac{R\omega y}{R'(\omega - u) + Ru} \tag{21b}$$

for convex paths.

A panoramic view ($R' = 0$) can be given by the following transformation:

$$\theta' = \theta \qquad y' = -\frac{\omega y}{u} \tag{22}$$

This implies that we can normalize a 2D GPV from a circular path of radius R by transforming it into another GPV of a reference radius R' (or PV) by using the image velocities u and the motion parameters U, ω.

4.3.2 Change of Center of a Circular Path. After modifying two GPVs to PVs at their path centers, we get two sections of PV separated by D as described in section 4.2, with the difference that the depth ρ of each point is known here. Let points P_i, $i = 1, \ldots, n$ in all directions be $P(\rho_i, \theta_i, Y_i)$ and $P'(\rho_i', \theta_i', Y_i')$ in both of the the GPVs. Using elementary trigonometry, we have

$$\rho_i' = \sqrt{\rho_i^2 + D^2 - 2\rho_i D \cos\theta_i} \tag{23a}$$

$$\cot \Delta\theta_i = \frac{\rho_i}{D \sin\theta_i} - \csc\theta_i \tag{23b}$$

where $\Delta\theta_i = \theta_i' - \theta_i$.

If $\rho_i \gg D$,

$$|\Delta\theta_i| \leq \frac{D}{\rho_i} \tag{24}$$

which means that θ and θ' of a distant feature are almost the same. Any distant point matched in both GPVs gives information about the directon from one center of path to another.

4.3.3 Change of Position of a Linear Path. If the linear path of the camera is shifted to the Z direction by D, the point $p(s, y)$ in the original GPV appears in the new GPV at $p'(s, y')$ such that

$$y' = \frac{Uy}{U - Du} \tag{25}$$

This means that change of PV occurs in the y scale. If an object is distant, a small shift of path within the road will not bring a great change in the y scale because from (9), we have

$$\frac{\partial y}{\partial Z} = -(Y/Z^2) \tag{26}$$

4.3.4 Change of Direction of a Linear Path. If the direction of the linear path is changed by ϕ, a point $P(X, Y, Z)$ that projects to $p(s, y)$ in the original GPV appears at $p'(s', y')$ in the new GPV such that

$$s' = s\cos\phi - Z\sin\phi \tag{27a}$$

$$y' = \frac{y}{(s/Z)\sin\phi + \cos\phi} \tag{27b}$$

Displacements of distant objects along the s direction due to the change of the motion direction are large, as (27a) indicates.

5 Matching Two Panoramic Representations

5.1 Coarse Matching Using Dynamic Programming

In real navigation, it is difficult for a robot to pursue the exact same path in different moves. The panoramic representations are arrived at in a path-oriented fashion. The matching, as a result, must be robust for small changes in the parameters. Fortunately, we can assume that the paths will not be too far apart so that we can match the two representations. Otherwise, the robot will conclude that it has gone on a different route. This section describes matching of two generalized panoramic views obtained from two different moves.

The matching process is coarse-to-fine. Because the 2D patterns change in the t-scale due to the different robot speeds, and also because large patterns in the GPVs will keep the same order of appearance if the camera pursues close paths along the route, DP (dynamic programming) methods, which have been extensively used in speech analysis for matching words spoken at different rates, can cope with the coarse matching of two sequential GPVs. The robot first matches the color projection $h(t)$ and $h'(t)$ of two GPVs by dynamic programming, and, then, precisely matches detail structures (vertical lines $l(i)$, $i = 1, \ldots, n$ and $l'(j)$, $j = 1, \ldots, m$) by searching narrow regions around the locations determined by the color matching.

The color projection $h(t) = [R(t), G(t), B(t)]$ of a GPV,

$$R(t) = \sum_{y=0}^{128} r(t, y),$$

$$G(t) = \sum_{y=0}^{128} g(t, y), \tag{28}$$

$$B(t) = \sum_{y=0}^{128} b(t, y)$$

represents the consecutive average color in each vertical pixel line. It maintains the dominant color in a panoramic view along the routes, in which structures are ignored, for the robot to "impressively" review the route it passes. Also, a long vertical line $l(i)$ in the GPV, which is good for determining precise position of the robot, appears as an explicit "edge" (an abrupt change in the value of $h(t)$). After the projection $h(t)$ is computed, we smooth it so as to maintain the most remarkable color changes in it. The edges $E(i)$ and $E'(j)$ in the projections are used as elements of correspondents, and the color values between them are used in computing an evaluation function.

Now we can explain the DP matching of the projections at edge sequences $[E(i_k), E'(j_k)]$.

Let the t coordinates of edges $E(i)$, $E'(j)$ be t_i, t_j. The matching space of DP is a grid with the horizontal and vertical lines at t_i and t_j respectively. Suppose we have obtained a number of path candidates searched by the step $k - 1$, which represent possible matching between two sets of edges by that step. For each possible matching pair $[E(i), E'(j)]$ at the next step k, the process compares all of the paths expanded to this node in the path candidates. Among them, a path extended from a node $[E(\hat{i}), E, E'(\hat{j})]$ that minimizes an accumulative value $f(i, j)$ is registered as the optimal path reaching the current node. The accumulative evaluation $f(i, j)$ is

$$f(i, j) = \min_{\hat{i},\hat{j}}[f(\hat{i}, \hat{j}) + p(k - 1, k)] \tag{29a}$$

with the local evaluation function $p(k - 1, k)$ defined as

$$p(k - 1, k) = \sum_{t=t_{\hat{i}}}^{t_i} \|h(t) - h'(t')\| + \|E(i) - E'(j)\| \tag{29b}$$

The t' is the linear mapping from t in the small duration $(t_{\hat{i}}, t_i)$ as

$$t' = (t - t_i)\frac{(t_j - t_{\hat{j}})}{(t_i - t_{\hat{i}})} \tag{29c}$$

and

$$\begin{aligned}\|h(t) - h'(t')\| = {} & |R(t) - R'(t')| \\ & + |G(t) - G'(t')| \\ & + |B(t) - B'(t')|\end{aligned} \tag{29d}$$

and $\|E(i) - E'(j)\|$ is the difference of the strengths of two color edges $E(i)$, $E'(j)$. For all of the possible nodes expanded at step k, the optimal paths reaching them are computed. Among then, the one with the minimum evaluation value f is considered as the edge correspondent by step k. The expansion of path candidates from the searched nodes to possible matching is not restricted to only their direct neighbors, but also can jump forward several nodes so that it allows the reversed order of detailed patterns in small ranges. The process performs this incremental matching when a new GPV is being created in autonomous navigation.

Figure 14 shows the optimal matching result of both dominant regions and edges between two generalized panoramic views while finding the final destination. The value in each local range evaluated from the function $p(k - 1, k)$ of the optimal path is displayed as well. Each position where a pair of "edges" is matched is connected by a line. Figure 15 shows the searched paths in the matching space of dynamic programming, in which the profiles of the R components of $h(t)$ and $h'(t)$ are displayed on the left and top margins. The grid is drawn at the positions where edges $E(i)$ and $E'(j)$ exist. The beginning positions of the GPVs are given and the matching starts from the top-left corner. The green paths in figure 15 indicate the search of possible candidates, and the blue path indicates the optimal correspondents.

Because the GPV provides a wide field of view along the route, matching different GPVs by dynamic programming can be done with a coarse-to-fine approach from a very coarse level. The result is reliable compared with the matching using local correlation without considering any context of patterns in the GPVs.

5.2 Feature Matching by Attributes

Given two sequences of data, dynamic programming can find the optimal correspondents between them. The question of whether the results fit the real scenes in the generalized panoramic views has to be checked further. After the approximate positions of patterns are obtained, detail structures wihin a small area determined by the coarse matching are easily matched if the

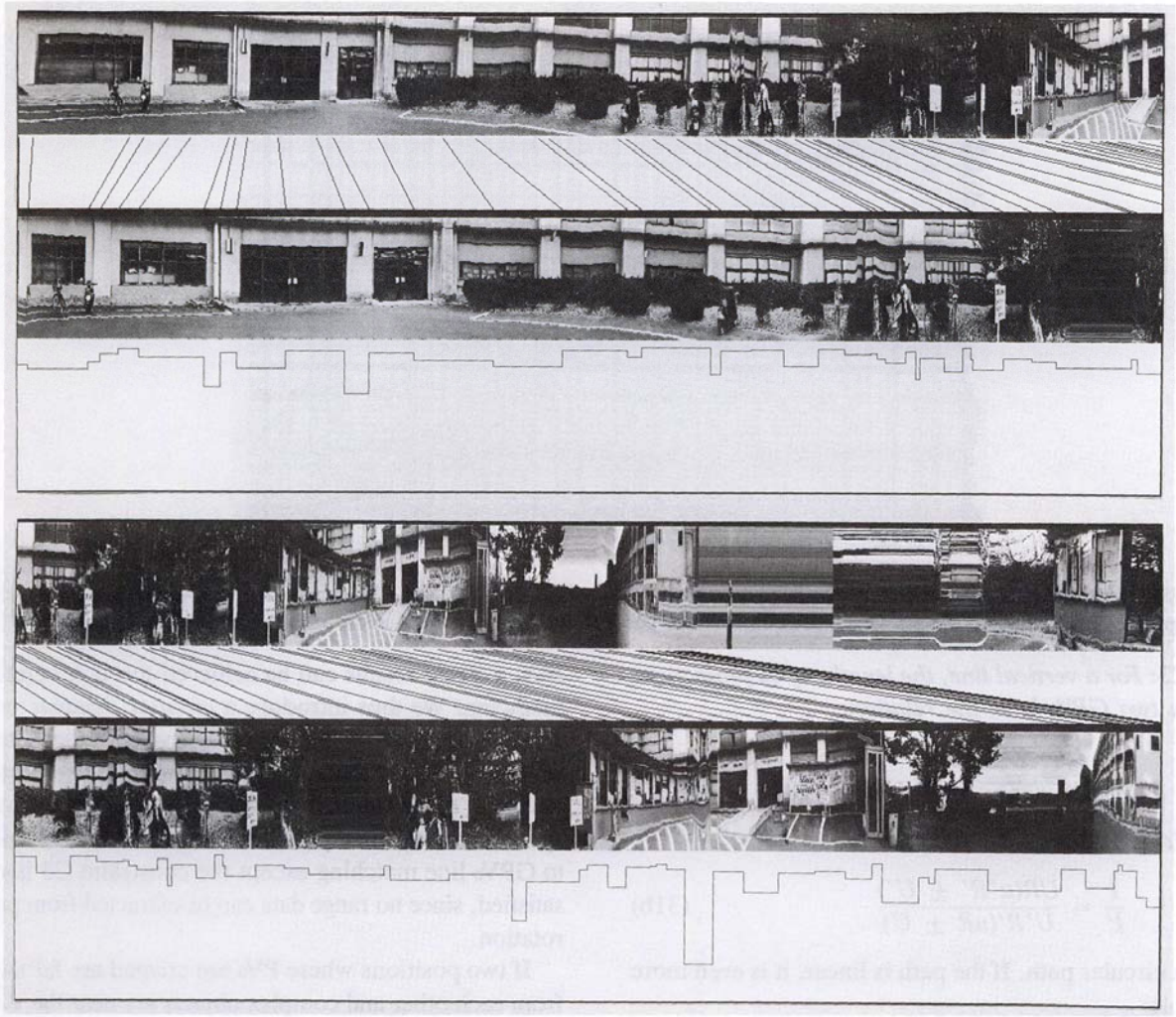

Fig. 14. Matching of color projections of two GPVs provides a coarse correspondence of GPV from different moves. The local evaluation of the optimal corresponding is displayed at the bottom.

coarse matching is correct. We match vertical lines by comparing their attributes such as length, strength, color on both sides, etc. [8].

Since the viewing distances of an object probably changes due to the separation of the paths in different moves, y-scale of 2D shapes in two GPVs may also be inconsistent. However, we can adjust feature candidates for matching to the same depth according to their acquired 2½D information, in order to check the consistency in 2D size. Besides color, average edge strength, etc. used in line base matching of GPVs, two additional constraints available are described as follows:

Let $Y1$, $Y2$ denote the Y coordinates of two end points of an almost vertical line in 3D space. Their y coordinates in a GPV are $y1$, $y2$ and the distance from the camera is Z. We have

$$\frac{y1}{Y1} = \frac{f}{Z} \qquad \frac{y2}{Y2} = \frac{f}{Z} \tag{30a}$$

and therefore,

$$\frac{y1}{y2} = \frac{Y1}{Y2} \tag{30b}$$

Since $Y1/Y2$ is a constant while the camera axis is horizontal, we have a second constraint:

C2: *The ratio of heights of two end points $y1/y2$ for an almost vertical line is invariant in GPVs of different moves.*

Let $L = Y1 - Y2$ denote the length of a vertical line observed from distances Z and Z' in different moves, and its projected lengths in the two GPVs are l and l'. The ratio l/l' gives another constraint.

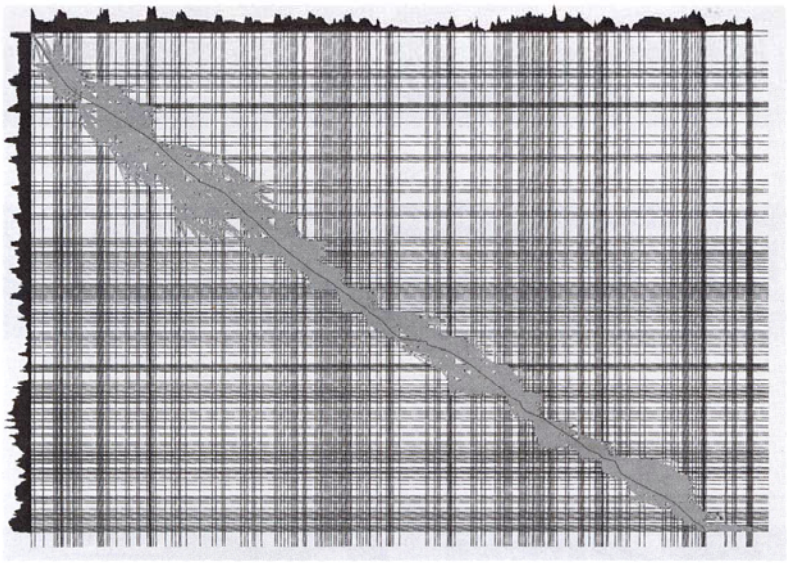

Fig. 15. Searching for the optimal path in dynamic programming space. Profiles of $R(t)$ and $R'(t')$ are displayed at the left and top margins. The grid is drawn at edge positions. Green paths incidate the possible matching and the blue one indicates the optimal correspondence.

C3: *For a vertical line, the lengths of its projections in two GPVs have the relation:*

$$\frac{l}{l'} = \frac{Z'}{Z} \tag{31a}$$

From (18) we have the relation

$$\frac{l}{l'} = \frac{UR(u'R' \pm U')}{U'R'(uR \pm U)} \tag{31b}$$

for a circular path. If the path is linear, it is even more simple:

$$\frac{l}{l'} = \frac{u'U}{uU'} \tag{31c}$$

Thus, 2½D information is used in adjusting the shape changes in y scale in matching the scenes viewed from different moves. Two lines $l(i)$, $l'(j)$ are matched if the most similar candidate of $l(i)$ is $l'(j)$, and vice versa. Figure 16 displays the matched pairs of lines.

5.3 Matching Panoramic Views

Matching of two panoramic views obtained by swiveling the camera at two locations is slightly different from the matching described in section 5.1. First, no initial position in the PVs where the dynamic programming can start is given, and we have to consider all the possible combinations of features in both PVs as initial position. Secondly, because of the circular structure of the PVs, a stable results can be achieved through iterative mathcing. We thus introduce a *circular dynamic programming* algorithm, for matching two periodic pattern sequences. After the color projections of the PVs are matched, vertical lines are checked to verify the correspondents in detailed parts. The method used is similar to GPVs line matching except the constraint C3 is not satisfied, since no range data can be extracted from pure rotation.

If two positions where PVs are created are far away from each other and complex objects are near the view points at the same time, a distinct change in shape and a serious occlusion in the PVs may occur. The failure in matching of such PVs means that the robot is too far away from the position the other PV has taken, and the next step is to approach it.

5.4 Circular Dynamic Programming

A matching approach applicable to general periodic distributions is presented here. We describe it using the panoramic view. Figure 17 depicts the basic idea of this approach termed *circular dynamic programming*. Suppose two edge sets $E = [E(i), i = 1, \ldots, n]$ and $E' = [E'(j), j = 1, \ldots, m]$ in two projections are obtained from the corresponding panoramic views. The optimal matching should draw a closed path in the searching space of dynamic programming. The circular programming works as follows.

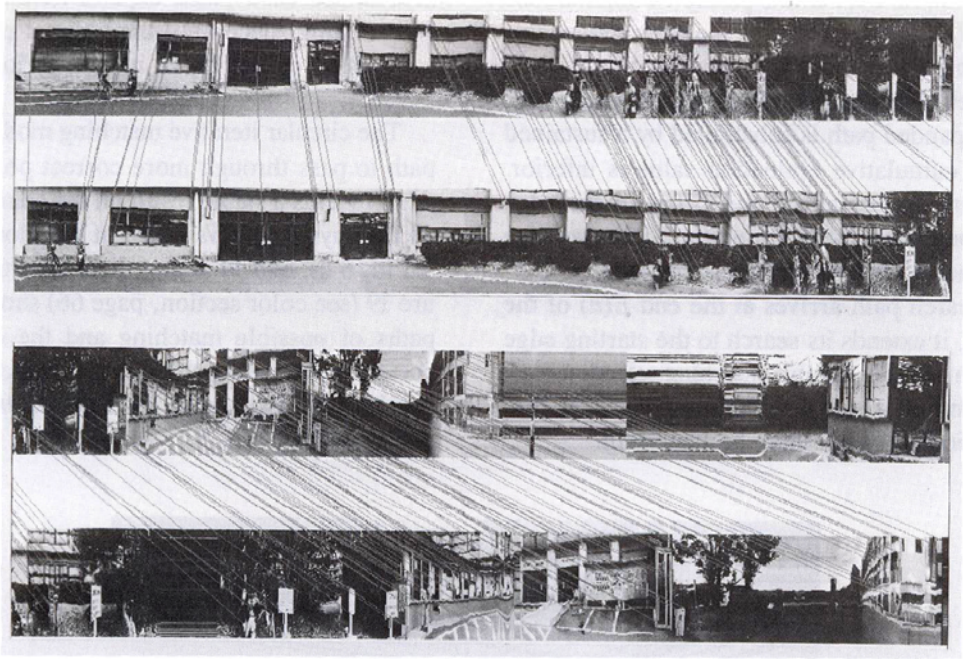

Fig. 16. Matching vertical line segments in two GPVs after coarse corresponding.

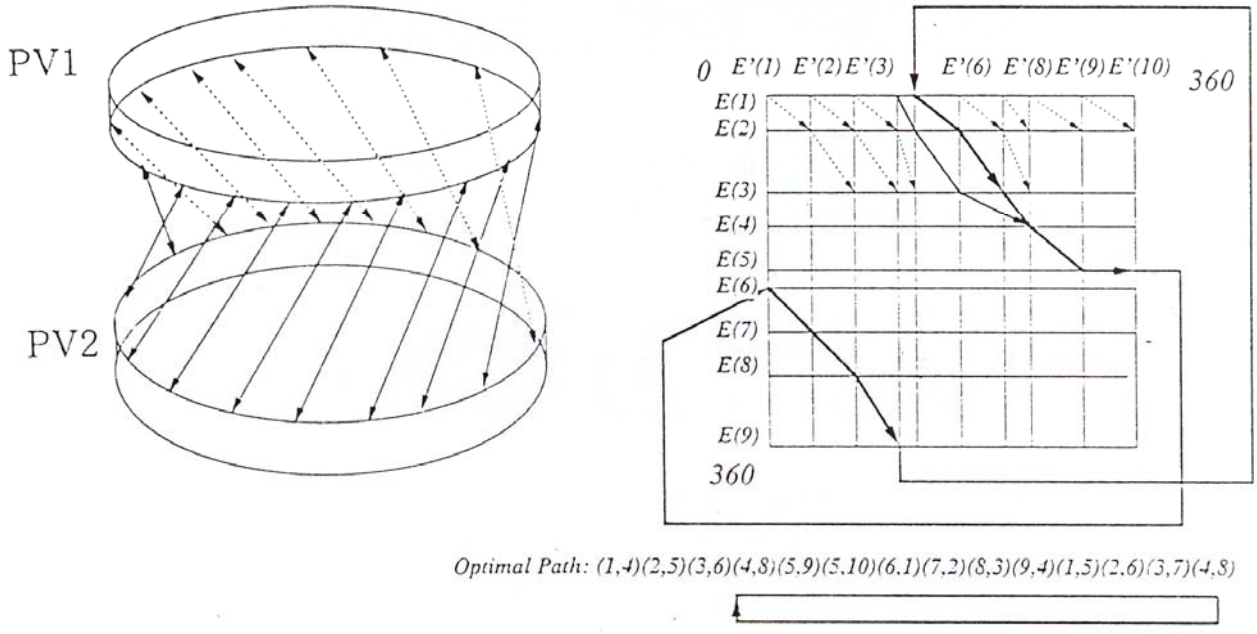

Fig. 17. Circular dynamic programming. (a) Intuitive illustration for matching two periodic PVs. (b) Searching for the optimal closed path of corresponding.

1. The search starts with all the possible combinations $[E(1), E'(j)], j = 1, \ldots, m$ in the edge sequences.
2. In order to save memory and computation, only a selected set of nodes (say 50 nodes) is expanded, and the expanded path is substituted by a sustained path if its cumulative evaluation value is inferior.
3. After a path arrives at the end $E'(m)$ of the edge set E', it continues to search from the beginning edge $E'(1)$ so that a circular search is realized.
4. When a search path arrives at the end $E(n)$ of the edge set E, it extends its search to the starting edge $E(1)$, which brings a reliable matching selection back to the beginning where the combinations taken into consideration in step 1 may be uncertain.
5. The searched paths expand iteratively across the matching space until the optimal path by that step forms a closed curve in one period.

The circular iterative matching modifies the optimal path to pass through more correct positions (step 4), which yields a stable result on the coarse level. Figure 18 displays four PVs taken at positions separated by 1.5 m, 6 m, and 10 m and their matching results. Figure 19 (see color section, page 66) shows the searched paths of possible matching and the obtained closed curves representing the optimal matchings. Matched results are also shown by lines drawn between the panoramic views.

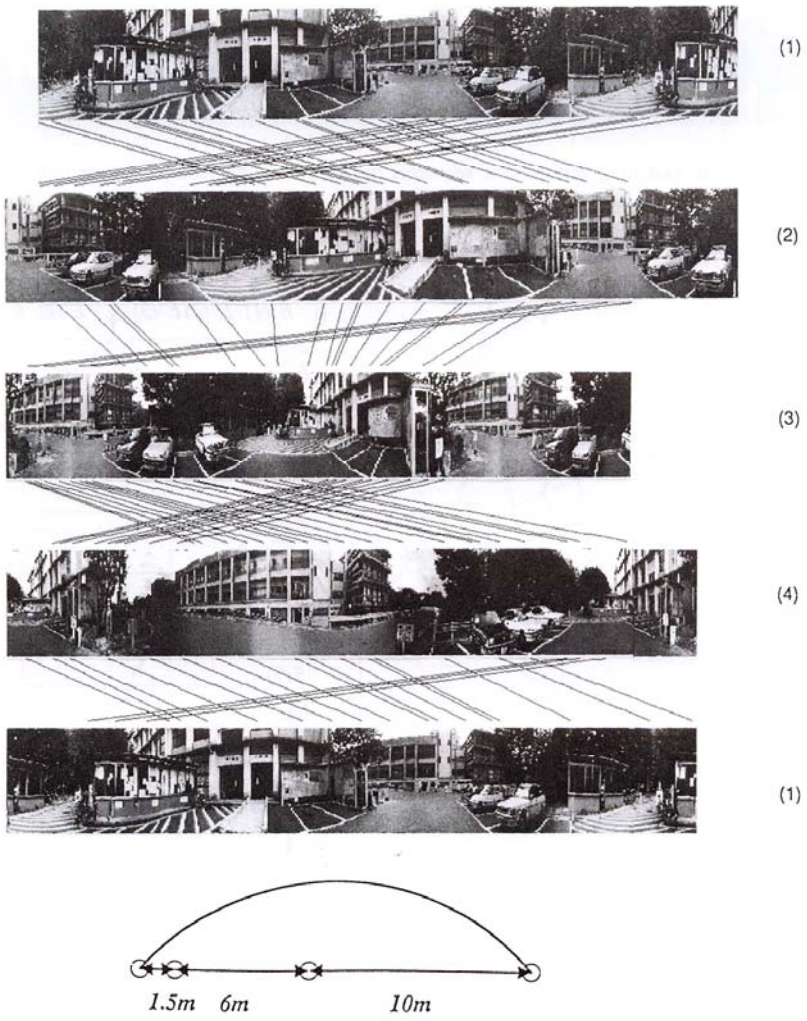

Fig. 18. Matching PVs at four positions by circular dynamic programming. Matching results 1–2, 2–3, 3–4, and 4–1 are depicted by lines connecting them.

6 Route Recognition

6.1 Route Pursuit

We have two kinds of route guidance for road following. As the robot moves along a street, the road-following process keeps the robot within the road and stops it if there is a moving object ahead. The heading of the robot is restricted within a small changeable range. The destination is possibly invisible for the reason of being occluded by objects intervening. The direction to move can be given qualitatively as *forward* and *turn left*, *right*, at a coming branch. The route-recognition task only needs to notify the road-following process at specific positions of stop or turn, by matching the current GPV with the memorized GPV.

On the other hand, if the robot enters a wide area such as a square, there are more movable spaces, free headings, and good visibility. The robot can mark the memorized destination in the new PV, by comparing with the PV generated in the experienced move. Then, the robot approaches it by adjusting its heading. Moreover, it judges its own locations from time to time, seeing whether it has arrived at the desired position by comparing with the memorized PV at that position. By matching the current PVs with the memorized PV taken in its trial move instructed by a human, the robot can figure out the spatial relation of two positions [1].

In the case of using the first approach, how accurately the estimated position corresponds to the true position depends mainly on the depth of a pattern that notifies the break point in GPV, as well as on the changeable extent of the robot heading along the road. Generating GPVs from both sides of the road may improve the accuracy of the robot position.

In the cognition phase, there is no partciular demand on processing time, since the entire model establishment can be done off-line after scenes are recorded. In automatic navigation, however, a quick response to the incoming scenes is necessary. Because of the small amount of data in a GPV, realizing the real-time recognition is promising.

Basically, the two-dimensional panoramic view is obtained in a sequential way, by which we reduce the processing of a whole frame to one line in each instance. In our system, the panoramic view is generated at the video rate. If a pipe-line processing is designed for the data sequence from a slit, it will reach real-time processing of panoramic view from gray level to line segment. Further, parallel processing is well suitable for the simultaneously generated panoramic views from double slits to estimate 2½D information, since they can be generated independently.

Dynamic programming is thought to be time consuming; it is performed only on the color projection and requires little time in each local path increment. In the example shown in figure 14, dynamic programming runs in 30 seconds (involving early processing) as does the vertical-line matching. The route-recognition task can inform the road-following task from time to time without stopping the robot.

6.2 Dealing with Temporal Objects in Panoramic Views

Since the matching of panoramic views is implemented by comparing patterns along a route, we have to consider the influences from some kinds of partial changes between the memorized panoramic view and the one being created in the autonomous move. Temporary objects such as parked cars may appear in one panoramic view but disappear in the others. Occlusion may also bring inconsistency between the two panoramic views. The robot should be able to neglect partial inconsistency so that it can continue the correct matching.

If the scale of a changed part is small, one method to detect the change can be to match the unchanged parts from a very "coarse" level to a fine level, since the wide field of panoramic view along the route allows us to smooth it to a coarse level. Local changes will not alter the correct mathcing of long GPVs on a coarse level, because the dynamic programming evaluates the optimal correspondents according to the context on the whole range, instead of the value in a specific part.

If a changed part has a large scale, both DP and the matching of line may fail. The robot will get lost in this situation. It starts multiple matching processes again at several suggested positions. If the process with the minimum accumulative evaluation among them is survived after a period of move and verification, the robot can continue its way.

7 Conclusion

We have proposed a dynamically generated *panoramic representation* for route recognition by an autonomous mobile robot. The described issues are dynamic sensing, visual memory, and route recognition in navigation.

The *panoramic representation* introduced is established from continuous input images through a slit that

is nonparallel to the optical flow in the image frame. It displays the essential information about shape and the time in a continuous 2D representation, with a minimum amount of data for a class of motion. The projection is expandable under the conditions of a moving camera. It has the attractive property that it can display a series of orientation-different or position-different views on a single 2D dynamic projection image. Multiple panoramic views can yield a 2½D sketch by computing the horizontal image velocity through the double slits. Therefore, *panoramic representation* contains the major information on the spatiotemporal volume. Since the GPVs can be created simultaneously from the image frame in real time, parallel processing can provide even higher speed in feature extraction.

In route recognition, we studied the matching of two GPVs (also PVs) generated from slightly different paths or positions such that it can guide the road-following process of a mobile robot. Since the *panoramic view* covers a wide field of view and maintains two-dimensional continuity as well, the matching can achieve a reliable result using a coarse-to-fine method, starting from a very coarse level. To solve the problem of shape change due to different paths of view, we employ the 2½D information in matching structures. The *panoramic view* can be acquired in real time; the matching allows the robot to speed up to 5 km/h in autonomous tracking of the routes, and perhaps even more for routes having simple and distant scenes.

References

1. J.Y. Zheng, Dynamic projection, panoramic representation, and route recognition, Ph.D. thesis in Osaka University, December 1989.
2. A.M. Waxman, J.J. LeMoigne, and B. Srinivasan, A visual navigation system for autonomous land vehicles, *IEEE J. Robotics Autom.* RA-3(2): 124–141, 1987.
3. C. Thorpe, M.H. Hebert, T. Kanade, and S.A. Shafer, Vision and navigation for the Carnegie-Mellon Navilab, *IEEE Trans. Patt. Anal. Mach. Intell.* 10(3): 362–373, 1988.
4. M. Turk, K.D. Morgenthaler, K.D. Gremban, and M. Marra, A vision system for autonomous land vehicle navigation, *IEEE Trans. Patt. Anal. Mach. Intell.* 10(3): 342–360.
5. Dickmanns and A. Zapp, A curvature-based scheme for improving road vehicle guidance by computer vision, *Proc. SPIE Conf. 727 on Mobile Robot*, November 1986.
6. A. Elfes, Sonar-based real-world mapping and navigation, *IEEE J. Robot. Autom.* RA-3(3): 249–265, 1987.
7. N. Ayache and O. Faugeras, Building, registering, and fusing noisy visual maps, *Proc. Intern. Conf. Comput. Vis.*, pp. 73–82, 1987.
8. S. Tsuji and J.Y. Zheng, Visual Path Planning, *Proc. 10th Intern. Joint Conf. Artif. Intell.* 2: 1127–1130, Milan, 1987.
9. G. Cohen, Memory in the real world, Lawrence Erlbaum Associates, Hove. 1989.
10. D.T. Lawton et al., Terrain models for an autonomous land vehicle, *Proc. IEEE Conf. Robot. Autom.*, pp. 2043–2051, 1986
11. B. Bhanu, H. Nasr, and S. Schaffer, Guiding an autonomous land vehicle using knowledge-based landmark recognition," *DARPA Proc. Image Understanding Workshop*, Los Angeles, pp. 432–439, 1987.
12. J.Y. Zheng, M. Asada, and S. Tsuji, Color-based panoramic representation of outdoor environment for a mobile robot, *Proc. 9th Intern. Conf. Patt. Recog.*, Rome, 2: 801–803, 1988.
13. J.Y. Zheng and S. Tsuji, Panoramic representaion of scenes for route understanding, *Proc. 10th Intern. Conf. Patt. Recog.*, Atlantic City, 1: 161–167, 1990.
14. D. Marr, Representing visual information. In A. Hanson and E.M. Riseman, eds., *Computer Vision Systems*, Academic Press: New York, 1978.
15. J.Y. Zheng and S. Tsuji, Spatial representation and analysis of temporal visual events, *Proc. IEEE Intern. Conf. Image Processing* 2: 775–779, 1989.
16. J.Y. Zheng and S. Tsuji, From anorthoscopic perception to dynamic vision, *Proc. IEEE Conf. Robot. Autom.* 2: 1154–1160, Cincinnati, 1990.
17. R.C. Bolles, H. Baker, and D.H. Marimont, Epipolar-plane image analysis: An approach to determining structure from motion, *Intern. J. Comput. Vis.* 1(1): 7–55, June 1987.
18. D.J. Heeger, Optical flow from spatiotemporal filters, *Proc. 1st Intern. Conf. Comput. Vis.* pp. 181–190, London, 1987.
19. H. Baker and R.C. Bolles, Generalizing epipolar-plane image analysis on the spatiotemporal surface, *Proc. Conf. Comput. Vis. Patt. Recog.* pp. 2–9, Ann Arbor, 1988.
20. Y. Ohta and T. Kanade, Stereo by two-level dynamic programming, *Proc. 9th Intern. Joint Conf. Artif. Intell.* 2: 1120–1126, Los Angeles, 1985.
21. A.R. Bruss and B.K.P. Horn, Passive navigation, *Comput. Vis. Graph. Image Process.* 21(1): 3–20, 1983.
22. B.K.P. Horn, Determining lightness from an image, *Comput. Graph. Image Process.* 3: 277–299, 1974.
23. E.H. Land, Recent advances in retinex theory, *Vision Research* 26: 7–21, 1986.
24. M. D'Zmura and P. Lennie, Mechanisms of color constancy, *J. Optic. Soc. Amer.* 3(10): 1662–1672, October 1986.
25. S.T. Barnand and W.B. Thompson, Disparity analysis of images, *IEEE Trans. Patt. Anal. Mach. Intell.* 2(4): 333–340, 1980.
26. J.Y. Zheng, M. Barth, and S. Tsuji, Qualitative route scene description using autonomous landmark detection, *Proc. 3rd Intern. Conf. Comput. Vis.* pp. 558–562, Osaka, 1990.
27. B.K.P. Horn and B.G. Schunk, Determining optical flow, *Artificial Intelligence*, 17: 185–203, 1981.
28. D.W. Thompson, Matching from 3-D range models into 2-D intensity scenes, *Proc. 1st Intern. Conf. Comput. Vis.*, pp. 65–72, 1987.
29. J.Y. Zheng, M. Barth, and S. Tsuji, Autonomous landmark selection for route recognition by a mobile robot, *Proc. IEEE Intern. Conf. Robot. Autom.* 3: 2004–2009, Sacramento, 1991.