*Regular Paper*

# A Method for Reconstructing Structure from Omnidirectional View Sequence without Feature Matching

Daniel Moldovan,† Takahiro Miyashita†† and Hiroshi Ishiguro†††

This paper describes a new method for reconstructing the 2D structure of an environment using an omnidirectional image sequence. The process starts by capturing the images with an omnidirectional camera mounted on a mobile platform that is moving on a straight path. By exploiting the characteristics of virtual omnidirectional images generated at arbitrary viewpoints in the environment, we are able to synthesize a 3-D visual representation of the environment. The 2D structure, parallel to the ground plane, will emerge by analyzing the 3-D visual representation. This method directly reconstructs the 2D structure from an omnidirectional visual sequence without using the feature matching process needed in multiple camera stereo. In the experimentation, we have applied this method to indoor and outdoor environments obtaining promising results.

## 1. Introduction

In previous works, various researchers have explored the use of OD (Omni-Directional) camera systems, in the context of robotic applications, for reconstructing environments from video imagery. By combining the measurements obtained from the video imagery with odometry measurements from the robot, Yagi and Kawato [15], Tsuji and Ishiguro [4),5)] constructed maps of the robot environment.

In order to retrieve the 3D information from the environment, Kawasaki, et al. [8)] proposed a spatio-temporal analysis of omni images. They proposed a hybrid method using the epipolar-plane image and the model-based analysis, performing a matching between video data and models. 3D information was retrieved from video data by using the matching results.

In contrast with their approach, this paper describes a new method for reconstructing the 2D structure of an environment from a sequence of OD images recorded on a rectilinear path. Our method does not require the matching process but it rather needs to generate many OD images (that will be called *virtual OD images* throughout this paper).

Similar approaches to ours are methods for approximately realizing *plenoptic functions* [1),9)], such as *lumigraph* [3)] and *light field rendering* [10)]. Recently, Taylor [14)] presented an approach for capturing the appearance of immersive scenes by combining techniques from structure from motion with ideas from image-based rendering. The limitation of this approach in the context of robot navigation is that it actually doesn't offer information about the structure of objects surrounding the robot.

By being able to create any view from any position to any direction on the ground, Takahashi's work [13)] is the most closely related with ours. However, their work is to reconstruct normal views with a limited visual field. In our work, we have improved this idea in order to find directly the structure of the environment from many virtual OD views.

Another related work is of J.P Mellor, et al. [11)]. They built and then analyzed an epipolar image in order to accumulate evidence about the depth at each image pixel. Comparing with our method there are three distinct differences: (1) they used a 3D arrangement of the cameras while we are using an epipolar-plane arrangement; (2) they inspected the surface of the objects while we thoroughly analyze the environment in an immersive way; (3) they used GPS in order to find the relative position of the cameras while we precisely approximated the local areas with straight lines (T-Net) [5)], memorized with a pair of feature points located at the end of the path. The details are described in Section 3.

This paper is organized as follows. In Section 2 we review two of the main epipolar con-

† Department of Computer and Communication Science, Wakayama University
†† ATR Intelligent Robotics and Communication Laboratories
††† Department of Adaptive Machine Systems, Graduate School of Engineering, Osaka University

straints used in stereo methods and in Section 3 we present the originality brought by our method. In Section 4 we describe the method for reconstructing the 2D structure from virtual OD images. Section 5 presents the results of the experiments using both indoor and outdoor scenes. Section 6 includes conclusions regarding this method, along with discussions of future work.

## 2. Epipolar Constraint in Stereo Methods

From the beginning of computer vision research, there have been many works aimed at recovering the three-dimensional information (depth) from two-dimensional images. The problem of recovering depth from a set of images is essentially the correspondence problem.

Finding potential corresponding points in each of the other images involves matching some image property in two or more images. Once a correspondence is known, solving for depth is simple a matter of geometry.

Feature matching method proposed so far, such as template matching is not stable especially for long base lines between cameras. On the other hand, longer baselines result in more precise depths. This leads to a conflict: short baselines simplify the matching process but produce imprecise results; long baselines produce precise results but complicate the matching process.

In order to solve the correspondence problem for several special cases of camera motion, Bolles, et al. [2] used an epipolar constraint for building a special image, which they called it *epipolar-plane image*. But even if the computational costs have been reduced, the feature matching problem could not be completely eliminated.

Kanade-Okutomi's *multi-baseline stereo* [12] gave a better solution to the matching problem. They have used multiple stereo pairs with different baselines generated by a lateral displacement of cameras and performed a simple matching by computing the sum of squared-difference (SSD) values.

By representing the SSD functions with respect to the inverse depth (1/z) and then by simply adding, to produce the sum of SSDs, the false matches were cancelling each other out. The resulting function exhibited a unique minimum at the correct matching position even when the intensity patterns of the scenes included ambiguities or repetitive patterns.

Recently, the algorithm of Mellor, et al. [11] succeeded in detecting the depths of image pixels without employing a feature matching process. For this analysis they defined an *epipolar image* similar to an epipolar-plane image but with one critical difference that ensured it can be constructed for every pixel in an arbitrary set of images.

Instead of using projections of a single epipolar plane, they built the epipolar image from the pencil of epipolar planes defined by the line through one of the camera centers and one of the pixels in the reference image.

The epipolar image was constructed by organizing a two-dimensional array with the epipolar lines from different images as rows. The columns of this matrix represented possible sets of correspondences ordered by depth.

One of the limitations of this method in the context of robotic applications is that its main focus is represented by the detection of the structure of the isolated objects than of the surrounding environment. Moreover, a 3D arrangement of cameras and their precise locations have to be known in order to detect the structure of objects. The method was intended for recovering the depth maps of built geometry (architectural facades) employing thus only an inspection of the objects' surface.

## 3. Epipolar Constraint in Our Method

Comparing with the methods described above, our method can be considered as an extension of their best merits. In order to increase the number of cameras we are sampling a video sequence from an OD camera that is moving along a straight path while keeping its image plane parallel with the ground plane and at a constant height from it (**Fig. 1**).

By employing an epipolar constraint similar with the one used by the epipolar-plane image we intend to reduce the computational cost re-

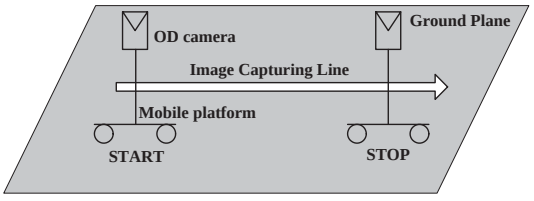


**Fig. 1** Mounted on top of a mobile platform, the OD camera is moving along a straight line capturing an OD video sequence.

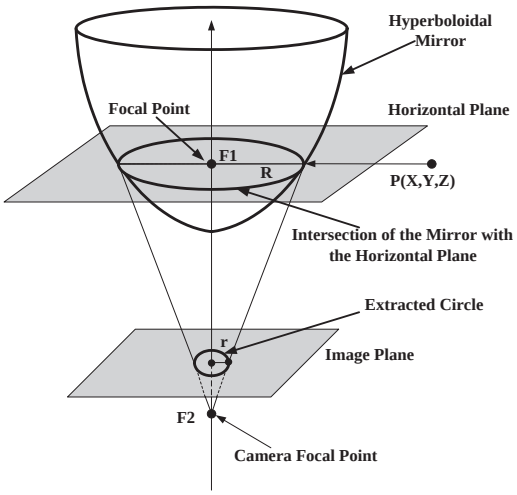


**Fig. 2** The extracted circle is corresponding to the intersection of the hyperboloidal mirror and the horizontal plane that is crossing the mirror's focal point.

lated with the correspondence problem. Therefore, from each of the sampled OD images we extract the circle that is corresponding to the intersection of the horizontal plane crossing the focal point of the hyperboloidal mirror, and the mirror itself (**Fig. 2**).

By using this constraint, the pixels from the extracted circles will correspond to the objects located at the same height from the ground as the OD camera's focal point. In this way we obtain a direct access to the height of the structures we are processing.

## 4. Structure from Virtual OD Images

### 4.1 Synthesis of Virtual OD Images

The omnidirectional camera that we used for image acquisition is composed of a CCD camera upward looking at a hyperboloidal mirror. For the outdoor experiments, the camera is mounted on a support on the roof of a vehicle and for the indoor experiments the camera is mounted on a mobile platform.

After recording a video sequence along a straight path, we apply a sampling process and obtain a number of original OD images. From each of the recorded images we extract the color information found in the circle that is coplanar with the focal point of the hyperboloidal mirror.

Virtual OD images are generated in a dense way, immersive into the environment and coplanar with the original OD images, by collecting one pixel at a time from each of the extracted circles (**Fig. 3**).

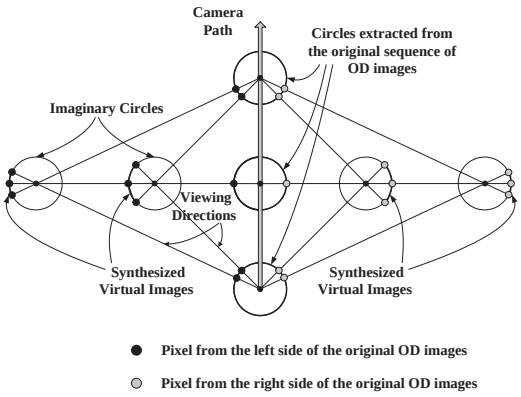


**Fig. 3** Virtual OD images are generated coplanar with the original OD images on both sides of the camera line.

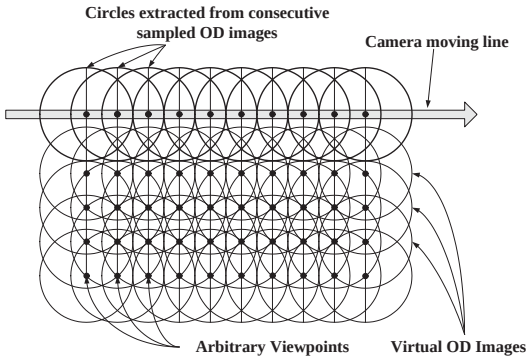


**Fig. 4** Virtual OD images are generated on the right side of the camera line, at arbitrary viewpoints and immersive into the environment.

In the process, if the rays corresponding to an original circle and an imaginary circle that has the center in an arbitrary viewpoint are collinear and if they have the same length and direction (towards the same side of the robot path), the visual information corresponding to the pixels located at the rays'extremities is the same.

In this way, the shape of a virtual OD image becomes an arc of a circle with the center in an arbitrary viewpoint.

As the location of the virtual viewpoint is moving away from the camera path, the reconstructable area in the virtual OD image becomes smaller.

From the sequence of extracted circles, we generate virtual OD images in arbitrary viewpoints, immersive into the environment (**Fig. 4**) and exploit changes that appear in each virtual image that encounters an object.

As can be noticed, our inspection is rather

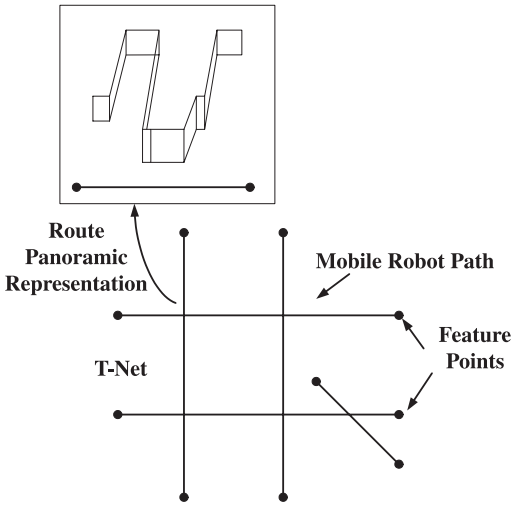


**Fig. 5** T-Net approximates local areas with straight lines, allowing a precise control of the camera motion.

thoroughly than superficially. By generating virtual images in a dense manner we can increase the resolution of our structure recovering method.

What happens if we generate a virtual view on an object? A virtual view in a free space shows an image that should be taken at the point. However, the virtual view on an object shows a monotone image filled with the object's color. As a result, the virtual image loses its texture. We have focused on this characteristic of the virtual views for detecting the 2D structure of the environment.

In order to eliminate the cases of occluded views we are taking multiple paths that are crossing each other. These paths are obtained by using the T-Net, which allows careful approximation of local areas with straight lines (**Fig. 5**) and a precise control of the camera motion.

By using only rich visual information, our approach is proposing an alternative stereo method that might offer a more robust solution for many real-world applications.

### 4.2 2D Structure from Virtual OD Images

For reconstructing the 2D structure of the environment, we apply a procedure that is consisting of 3 consecutive steps:

(1) **Generating the *Virtual Line Images***

In the beginning, corresponding to each of the extracted circles, we are rendering a number of *virtual lines* by importing one pixel at a time

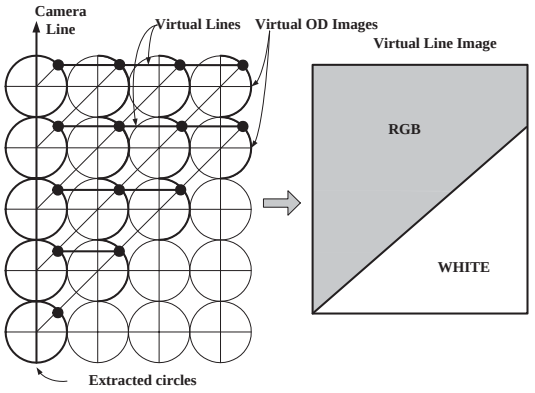


**Fig. 6** Virtual line images are built from intermediate virtual lines that are corresponding to pixels that have a certain angle with the line perpendicular to the camera path.

from each virtual OD image that is generated on a direction perpendicular to the camera path (**Fig. 6**). For each virtual line, the imported pixels must have the same angle with the line perpendicular to the camera path.

*Virtual line images* are obtained by gathering all virtual lines that correspond to a certain angle. For one side of the camera path we will generate 180 virtual line images.

Generating the entire virtual OD image in order to extract just one pixel that will be used in building the virtual lines is a time consuming process. In order to speed up the process we found that there is no need to generate the whole virtual image. That pixel can be easily recovered from the captured OD images. In other words, in order to build virtual lines we are using the principle of generating virtual OD images but without actually generating them.

For each virtual line image, the process is building a Boolean matrix that will record the data related to the presence of the color information in the newly generated virtual line images. Each of their location will be assigned a TRUE or a FALSE value (**Fig. 7**) that will be used in the 3rd step of the procedure.

(2) **Building the 3D-Visual Representation Volume (3D-VRV)**

By arranging all virtual line images one on top of the other we can build a 3-D visual representation volume (VRV) of the environment with I, X, Y axes representing the vertical section, distance from the camera path and the camera movement, respectively (**Fig. 8**).

The projection of this 3-D VRV on the X-Y plane will give us the 2D structure of the en-

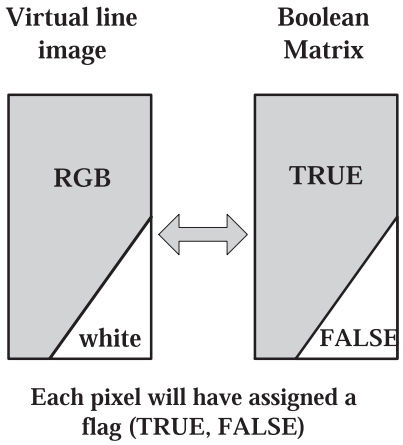


**Fig. 7** Each location in the Boolean Matrix will record the data related to the presence of the color information in the newly generated virtual line image.

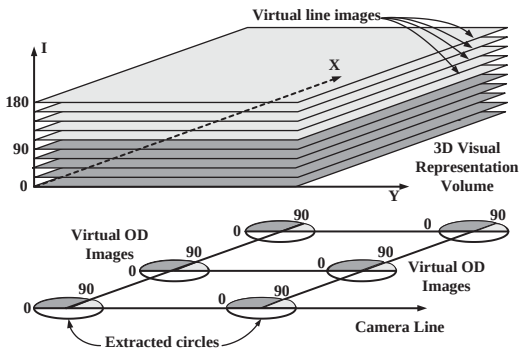


**Fig. 8** For one side of the camera line, the 3-D visual representation volume is composed of intermediate virtual line images.

vironment. The projection is seen as a process of detecting the common areas where pixels exhibit similar RGB values.

### (3) 2D Projection of the 3D-VRV onto the Ground Plane

In order to carry-out the 2D projection of the 3D VRV, we employ an overlapping method that is iterating an XOR process between the pixels of two images (the VRV slices) using the information from the corresponding Boolean matrixes. The output of this process is an image that is keeping the pixels that have similar RGB values and is eliminating the rest of them. The comparison of two images comes down to the comparison of pixel's RGB values. Our solution for this problem was the use of a vectorial representation (**Fig. 9**) where each vector is represented as an (R, G, B) triplet.

The scalar product of two vectors corresponding to two different pixels is given by:

$$\vec{v_1}\vec{v_2} = |v_1||v_2|\cos\theta;$$

where

$$|v_1| = R_1^2 + G_1^2 + B_1^2; \quad |v_2| = R_2^2 + G_2^2 + B_2^2;$$

and

$$\vec{v_1}\vec{v_2} = R_1R_2 + G_1G_2 + B_1B_2.$$

If the angle between vectors is equal with 0 and if the vectors have the same length, they are corresponding to the same color. By varying the angle and by imposing certain lengths to the vectors we can impose accuracy thresholds in order to compare different colors.

Looking from the point of view of 2D structure recovery, if the color variance is smaller than a certain threshold, then the output is valid, meaning that there is an object. Otherwise, the result is invalid (no object).

After getting the result of comparison between two initial images, the overlapping process continues with the comparison of the result with the next slice from the STV. This process is iterative and it ends when there is no more slice to compare with.

By the simple projection, we can get a 2D image that represents environmental structure along the camera path. Note that this process for acquiring the 2D environmental map does not require a feature matching process.

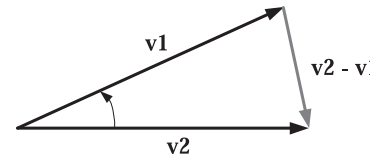


**Fig. 9** We chose a vectorial representation for each (R, G, B) triplet.

## 5. Experimental Results

The indoor experiment has been done using a static environment from our laboratory. **Figure 10** shows the sequence of the original images that were recorded with a frequency of one image/0.30 cm. The thick, gray lines from **Fig. 11** represent the walls of the cubicles. By rendering the left side of the camera path we obtain a sequence of 180 virtual line images (**Fig. 12**).

The structure on the left side of the path emerges from overlapping the corresponding 180 slices (**Fig. 13**). The meanings of the objects encircled and numbered are: (1) the cor-

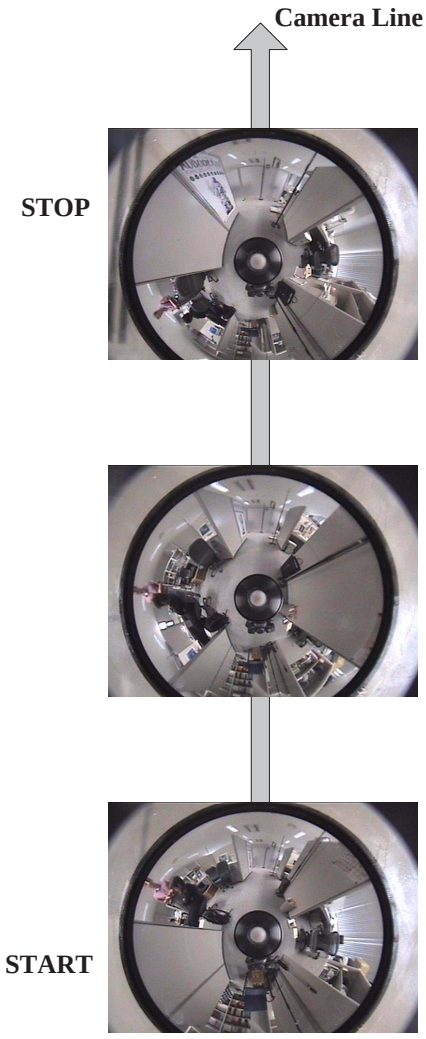


**Fig. 10** Indoor environment — omni images along the path.

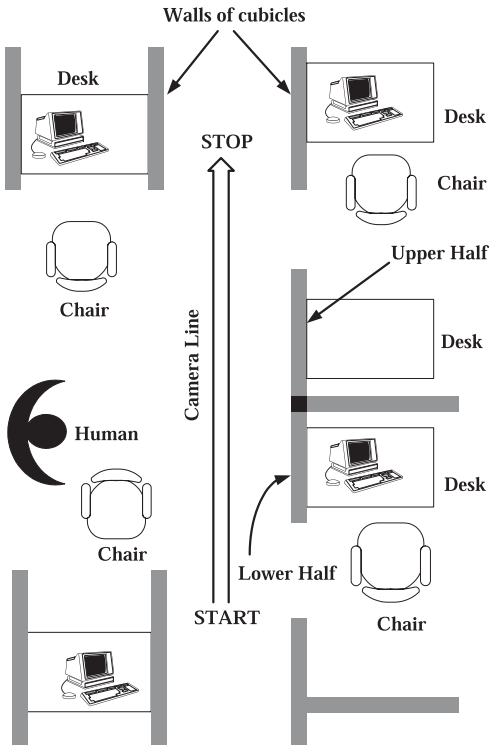


**Fig. 11** Indoor environment — simplified representation.

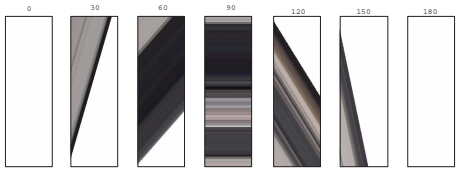


**Fig. 12** Virtual line images on the left side of the camera line.

ner of the wall; (2) the chair; (3) the standing person and the chair; (4) the corner of the wall.

By following the same procedure we determined the structure on the right side of the path (**Fig. 14**). The meanings of the encircled areas are as follows: (1) represents the upper half of the wall; (2) represents the lower part; and (3) is the edge of the desk. They resulted separated because in the original images they had an edge of a different color between them.

We have to mention that even with a low accuracy the results proved the right location of the objects in the surrounding environment.

Next are the results for an outdoor scene, located in our university campus. The camera was mounted on top of a vehicle that was moving on a public road, along a 50 meters long straight path, while recording a sequence of OD images (**Fig. 15**) with a frequency of one image per 25 cm. We drove smoothly in order to avoid tilt variations of the camera. The buildings located on each side of the road represented the structures we wanted to detect (**Fig. 16**).

The structures detected on the left side of the camera line are shown in **Fig. 17**. The two encircled objects (1 and 2) correspond to the front part of building 1 that includes two pillars of similar color separated by a balcony of a darker color.

Because of the limitation of this method in dealing with concave shapes, in the case of building 2, the inner part area (black shaded zone from Fig. 16) is assimilated into the build-

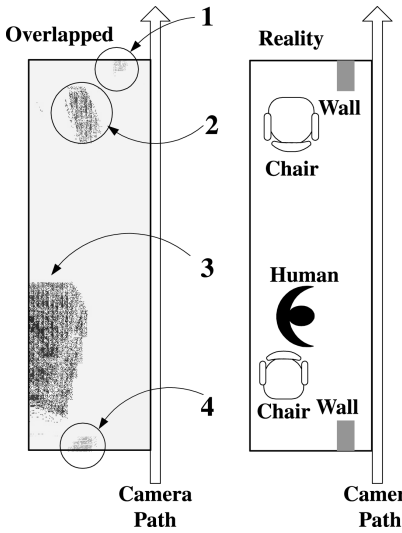


**Fig. 13** Left side structure: overlapped results and reality.

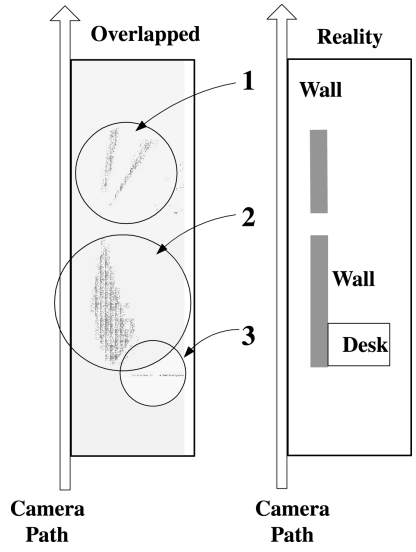


**Fig. 14** Right side structure: overlapped results and reality.

ing's shape (**Fig. 18**).

Because unavoidable shakes of the camera could not be avoided, the outdoor experiments were done using images that had a negligible tilt variation. In the case of stronger shakes that could be clearly noticed, the entire sequence of recorded images was discarded and the recordings were done again.

As proved by the results, one application for our method might be the recovery of the environment's coarse structure. Comparing with the indoor environments, where the flat floor

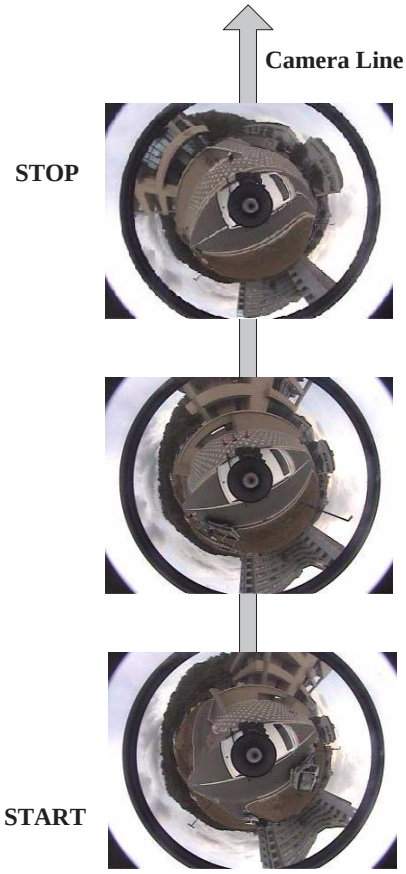


**Fig. 15** Outdoors environment: OD images along the path.

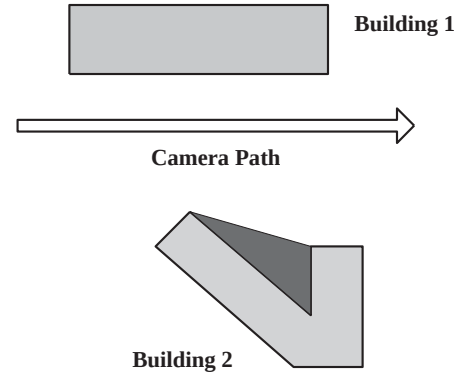


**Fig. 16** Outdoors environment: structure.

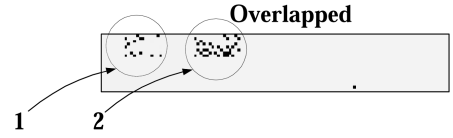


**Fig. 17** Structure on left side of the road.

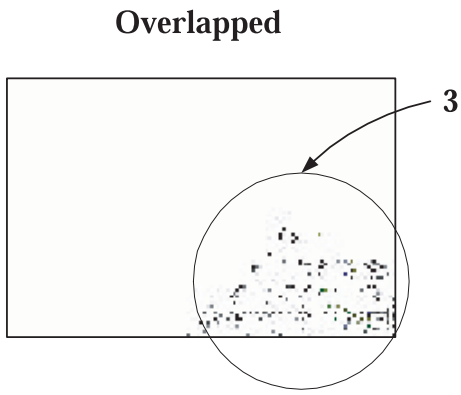


**Fig. 18** Structure on right side of the road.

constraint is satisfied for the entire camera path, the results for outdoors scenes are more liable of being affected by errors due to camera's tilt variations. However, in the case of structures of big sizes (like buildings or cars) located in the near vicinity of the camera's path, small variations of camera's tilt are not significant for the overall result.

## 6. Conclusion

This paper described a new method for reconstructing the 2D structure of the surrounding environment from image sequences taken by an OD camera. We generate virtual omnimages on both sides of the path and exploit the changes that appear in each virtual image that encounter with an object. The originality brought by this method is that it uses only the rich visual information in order to solve the correspondence problem.

The strength of our method is that it requires a single omni-camera and the processing is done in real time being well suited for real-world applications. A weak point is represented by the low accuracy in detecting the objects shape. Future work will focus on overcoming this limitation and on extending our method to 3D structure.

## References

1) Adelson, E.H. and Bergen, E.H.: *The plenoptic function and the elements of early vision*, MIT Press (1991).
2) Bolles, R.C., Baker, H.H. and Marimont, D.H.: Epipolar-plane image analysis: An approach to determining structure from motion, *International Journal of Computer Vision*, Vol.1, No.1, pp.7–55 (1987).
3) Gortler, S.J., Grzeszczuk, R., Szeliski, R. and Cohen, M.F.: The lumigraph, *Proc. SIGGRAPH*, pp.43–54 (1996).
4) Ishiguro, H., Maeda, T., Miyashita, T. and Tsuji, S.: A strategy for acquiring an environmental model with panoramic sensing by a mobile robot, *IEEE Int. Conf on Robotics and Automation*, pp.724–729 (1994).
5) Ishiguro, H., Miyashita, T. and Tsuji, S.: T-Net for navigating a vision-guided robot in a real world, *IEEE International Conference on Robotics and Automation* (1995).
6) Ishiguro, H., Ueda, K. and Tsuji, S.: Omnidirectional visual information for navigating a mobile robot, *IEEE Int. Conf. on Robotics and Automation*, pp.799–804 (1993).
7) Ishiguro, H., Yamamoto, M. and Tsuji, S.: Omni-directional stereo, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.14, No.2, pp.257–262 (Feb. 1992).
8) Kawasaki, H., Ikeuchi, K. and Sakauchi, M.: Spatio-Temporal Analysis of Omni Image, *Computer Vision and Pattern Recognition*, pp.577–584 (2000).
9) Landy, M. and Movshon, J.A. (Eds.): *Computation models of visual processing*, MIT Press (1991).
10) Levoy, M. and Hanrahan, P.: Light field rendering, *Proc. SIGGRAPH*, pp.31–42 (1996).
11) Mellor, J.P., et al.: Dense Depth Maps from Epipolar Images, *M.I.T Artificial Intelligence Laboratory* (1996).
12) Okutomi, M. and Kanade, T.A: Multiple Baseline Stereo, *Proc.1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.63–69 (1991).
13) Takahashi, T., Kawasaki, H., Ikeuchi, K. and Sakauchi, M.: Arbitrary View Position and Direction Rendering for Large-Scale Scenes, *Computer Vision and Pattern Recognition*, pp.296–303 (2000).
14) Taylor, J.C.: VideoPlus, *IEEE Workshop on Omnidirectional Vision*, pp.3–10 (2000).
15) Yagi, Y., Kawato, S. and Tsuji, S.: Real-time omnidirectional image sensor for vision guided navigation, *IEEE Journal of Robotics and Automation*, Vol.10, No.1, pp.11–21 (1994).

(Received September 4, 2002)
(Accepted March 28, 2003)

(Editor in Charge Yasushi Yagi)

**Daniel Moldovan** received his B.S. degree in Electronics from Technical University of Cluj-Napoca, Romania in 1995 and M.S. degree in Intelligent Media Systems from Wakayama University in 2002. He is currently pursuing the Ph.D. at Wakayama University with a research focus in simultaneous localisation and mapping for mobile robots navigating in non-trivial environments. His research interests are in computer vision and robotics.

**Takahiro Miyashita** received his B.S., M.S., and Ph.D. degree in engineering for computer-controlled machinery from Osaka University, Japan in 1993, 1995, and 2002 respectively. Currently he is a researcher of ATR IRC Labs. since 2002. His research interests include computer vision, vision based robots, control for multi-degrees of freedom robots, and humanoid robots in daily environment. He is a member of the RSJ and the JSAI.

**Hiroshi Ishiguro** received D. Eng. degree from Osaka University, Japan, in 1991. In 1991, he started working as a research assistant of Department of Electrical Engineering and Computer Science, Yamanashi University, Japan. Then, he moved to Department of Systems Engineering, Osaka University, Japan, as a research assistant in 1992. In 1994, he was an associate professor of Department of Information Science, Kyoto University, Japan, and started research of distributed vision using omnidirectional cameras. From 1998 to 1999, he worked in Department of Electrical and Computer Engineering, University of California, San Diego, USA, as a visiting scholar. From 1999, he is a visiting researcher in ATR Media Information Science Laboratories and he has developed interactive humanoid robots, Robovie. In 2000, he moved to Department of Computer and Communication Sciences, Wakayama University, Japan, as an associate professor and then he became a professor in 2001. Now he is a professor of Department of adaptive machine systems, Osaka University, Japan.