



# Image-Based Modeling from a High-Resolution Route Panorama

Ryuji Shibata and Hajime Nagahara

Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531 Japan

**Abstract**—Image-based modeling methods for generating 3D models from an image sequence have been widely studied. Most of these methods, however, require huge redundant spatio-temporal images to estimate scene depth. This is not an effective use of capturing higher resolution texture. On the other hand, a route panorama, which is a continuous panoramic image along a path, is an efficient way of consolidating information from multiple viewpoints into a single image. A route panorama captured by a line camera also has the advantage of capturing higher resolution easily. In this paper, we propose a method for estimating the depth of an image from a route panorama using color drifts. The proposed method detects color drift by deformable window matching of the color channels. It also uses a hierarchical belief propagation to estimate the depth stably and decrease the computation cost thereof.

**Index Terms**—Image-Based Modeling, Route Panorama, Deformable Window Matching, Belief-Propagation

## I. INTRODUCTION

Recently, 3D models of real objects have been utilized in various applications such as movies, digital archives, city modeling, and a digital globe on the Internet. For these applications, photorealistic models with higher resolution textures are required to increase the feeling of reality and to observe the smaller details. However, it is both extremely costly and time consuming to build such models since they are currently built by hand.

Much research [1-4] has focused on investigating modeling methods that create models from images automatically, such as image-based modeling. Although these methods have been used for 3D modeling from an image sequence, most of them require a huge spatio-temporal image volume including dense and smooth viewpoint changes for stable accurate estimations. Such dense image data are helpful in estimating 3D geometry; however, it is mostly redundant with respect to obtaining texture images, because the images from neighboring viewpoints are similar. Of course, sensor resolution is ultimately the limiting factor in obtaining high resolution textures, but this redundancy has also prevented an increase in resolution owing to the huge data storage required to implement a larger field or higher resolution for imaging and modeling. Some research [5, 6] has focused on improving the

texture resolution from the redundancy of images by using a super resolution technique. However, the resolution improvement of super resolution is limited and not at all substantial.

The route panorama [7, 8] has been proposed to suppress image data with multiple views. It can be described as a long belt-like stitched panorama image of one vertical line on every continuous frame as the camera moves along a path, and consolidates information from multiple views into a single image. Many imaging applications can capture a route panorama, such as a commercial flatbed scanner. Some 3D scanners [9] capture the panorama image using a rotating sensor motion to get the entire 360 degree texture of the target object. Google street view [10] uses multiple sequential camera images captured while a car moves through a city street. We can create a route panorama from this kind of captured images to obtain large area visual information of the city [7].

Previously we proposed capturing a route panorama using a line camera [11]. This method is beneficial in that it captures higher resolution without any redundancy in the capture or storage of the image data. For instance, a line camera has a 1D line CCD with higher resolution, several thousands of pixels, than the vertical resolution of common 2D imaging devices. Using the camera to capture a route panorama is efficient because only one vertical line is used for stitching the panorama. The route panoramas obtained from CCD of 3D scanners could typically only be used to capture texture information for a 3D model, because the 3D geometry is obtained from a laser range finder equipped with the scanners. Nevertheless, such a route panorama still has a depth key as color drifts, although the image has no overlapping scene information. Previously we proposed a depth recovery method for a route panorama [11], although with a restriction on the linear camera motion while capturing the panorama image.

In this paper, we propose a general model for capturing a route panorama using a color line camera and a depth estimation method for the panorama using an arbitrary camera trajectory that includes rotation. Since we model the camera projection for capturing a route panorama under arbitrary motion trajectory, the proposed depth estimation method can be applied to various route panorama scanning systems that are suited to various target objects, such as rotating scanners for small relic objects as in [9], or moving cars for capturing buildings as Google [10] does. The proposed method uses deformable window matching of the color channels to detect color drifts. We also use a hierarchical belief propagation to estimate the depth stably and decrease its computation cost.

Simulations and actual experiments have been carried out using two different trajectories, namely linear and circular motion, to confirm the effectiveness of our proposed method.

## II. GEOMETRY OF A COLOR LINE CAMERA FOR CAPTURING A ROUTE PANORAMA

### 2.1 Geometry of Route Panorama

A route panorama is a long image strip, as shown in Fig. 1, obtained by continuously scanning a scene with a camera moving along an arbitrary trajectory. The panorama is usually a stitched image with the vertical line of the image center on every frame of the image sequence.

An image in the route panorama has  $x$ - $y$  coordinates, where the  $x$ -axis corresponds to the speed of motion and the frame rate of the camera. Higher resolution is obtained with slower camera motion and a higher frame rate. On the other hand, the resolution along the  $y$ -axis corresponds to the vertical physical resolution of the camera. Hence, a camera that has both high vertical resolution and a high frame rate is preferable for capturing a high-resolution route panorama.

In this study, we used a color line camera to capture a high-resolution route panorama. The line camera has a single vertical line CCD, with a larger number of vertical pixels (several thousand pixels) and faster frame rate (over 1 kHz) than the common area CCDs, even though it captures only a single 1D line image at a time. A route panorama is usually created from only a single vertical line on each frame. This means that if a camera with an area imager is used, the majority of image pixels are not used to create the panorama. Thus, a line camera is adequately suitable for capturing a high-resolution route panorama.

### 2.2 Origin of Color Drifts

A color line camera uses three parallel line CCDs, one for each RGB color channel, to create a color line image. These line CCDs are aligned, but at slightly shifted positions. Therefore, the captured image has color drift at its edge as shown in Fig. 2. Fig. 3 shows a model of a color line camera. The line image of the green channel is centered on the image plane. The red and blue channels are shifted  $\Delta l$  from the green channel. The focal length of the camera, that is, the distance between the principal point of the lens and the line CCD, is defined as  $f$ . The R-G and G-B line gaps cause differences in the incident angle  $\theta$ , where  $\theta$  is expressed as:

$$\theta = \arctan(\Delta l / f). \quad (1)$$

$\theta$  is an intrinsic parameter and is not dependent on the captured object since the focal length  $f$  and the gaps  $\Delta l$  are defined by the selection of line camera or lens.

### 2.3 Geometry of 3D Reconstruction

Fig. 1 shows the projective relationship of a route panorama. We define  $X$ ,  $Y$  and  $Z$  as world coordinates, as shown in Fig. 1. The camera captures an image sequence while moving along an arbitrary camera trajectory, which is generally described as  $\mathbf{S}(t)$ , where  $\mathbf{S}(t)$  is a series or function of positions and orientations in 3D space at time  $t$ . We assume

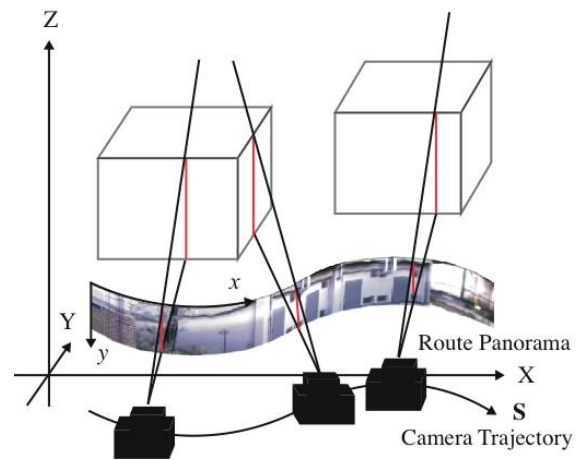


Fig. 1 Capturing a route panorama



Fig. 2 Color drift of a captured image

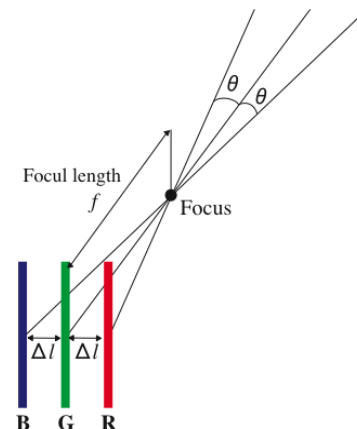


Fig. 3 Model of a color line camera

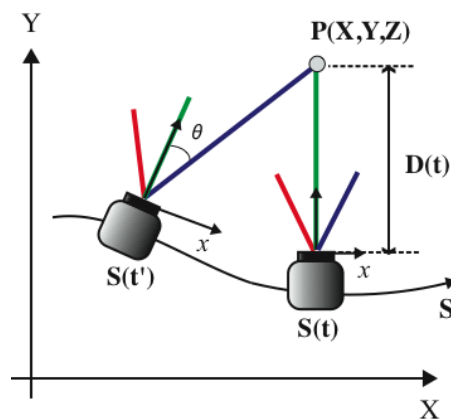


Fig. 4 Geometry of a route panorama

that  $\mathbf{S}(t)$  is known by controlling the camera using a motion stage or by using position sensors such as a differential GPS and compass. By capturing images while moving the camera, we obtain three route panorama images  $I_R(x,y)$ ,  $I_G(x,y)$ , and  $I_B(x,y)$ , one for each RGB color channel. The  $y$ -axis indicates the 1D line image coordinate of the camera, while the  $x$ -axis denotes the time at which the line was captured. This is described as:

$$t = r \cdot x + t_0, \quad (2)$$

where  $r$  is the frame rate of the camera and  $t_0$  is the time of origin. The frame rate  $r$  decides the horizontal resolution of the route panorama image.

Each RGB channel image is projected on a different viewpoint because of the angular difference of  $\theta$ , as shown in Fig. 4. Note that although we only refer to the G-B channels in the explanation below, it is equally applicable to the R-G channels since the relation between the R and G channels is the same.

If an arbitrary point  $P$  on a scene is distant from the camera trajectory, the point  $P$  is captured from different viewpoints,  $\mathbf{S}(t)$  and  $\mathbf{S}(t')$ . Here, we assume that the green channel is the reference for the image coordinate  $(x,y)$ . Hence, the scene point  $P$  is projected onto the image points  $p(x,y)$  and  $p(x+d,y)$  for the green and blue color channels, respectively. This projective difference causes the color drift  $d$  of the route panorama.

If we find the color drift  $d$  at image point  $p(x,y)$ , which is the point on the green channel at time  $t$ , the time  $t'$  when the B channel captures the same scene point is expressed as:

$$t' = t - \frac{d}{r}. \quad (3)$$

Therefore, the scene depth  $D_{x,y}$  at image point  $p(x,y)$  is generally described as a function of color drift  $d$ , camera trajectory  $\mathbf{S}$ , and time  $t$ :

$$D_{x,y} = f(d, \mathbf{S}, t). \quad (4)$$

If we consider a specific condition, whereby the camera moves along the  $X-Y$  plane and rotates about the  $Z$ -axis, the trajectory is expressed as:

$$\mathbf{S}(t) = [X_s(t), Y_s(t), \phi(t)]^T, \quad (5)$$

where  $\phi(t)$  is the angle between the principal axis of the camera and the  $Y$ -axis.

Under this condition of motion, (4) is written as:

$$D_x = Y_s(t) - Y_s(t') + \frac{X_s(t) - X_s(t')}{\tan(\theta + \phi(t) - \phi(t'))}. \quad (6)$$

Thus, the 3D position of  $P(X,Y,Z)$  is obtained as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} D_x \sin \phi(t) + X_s(t) \\ D_x \cos \phi(t) + Y_s(t) \\ D_x y / f \end{bmatrix}. \quad (7)$$

We now consider two cases of camera motion, namely, linear and circular motion. If the camera moves linearly with constant velocity, (5) becomes:

$$\mathbf{S}(t) = [Vt, 0, 0]^T, \quad (8)$$

where  $V$  is the constant translative velocity. Substituting (3) and (8) into (6), (4) is simply expressed as:

$$D = \frac{V}{r \tan \theta} \cdot d. \quad (9)$$

If the camera moves around an object with constant angular velocity, (5) becomes:

$$\mathbf{S}(t) = [L \sin \omega t, L(1 - \cos \omega t), \omega t]^T, \quad (10)$$

where  $\omega$  is the constant angular velocity and  $L$  is the distance from the rotation center. Substituting (3) and (10) into (6), (4) is simply expressed as:

$$D = \frac{L \omega d}{r \theta + \omega d}. \quad (11)$$

Note that both (9) and (11) are independent of image point  $p(x,y)$  and that the depth  $D$  is only related to the color drift  $d$  if we assume constant velocity.

### III. ALGORITHM FOR ESTIMATING DEPTH

#### 3.1 Overview of the Proposed Algorithm

We propose a method that incorporates deformable window matching and hierarchical belief propagation to detect color drift. Fig. 5 gives a flowchart of the method.

A standard window matching method can detect the size of a color drift [11]. However, the matching lacks accuracy because it assumes that the captured object's surfaces are locally shaped as a frontal plane parallel to the camera trajectory. This "front-parallel" assumption causes a discontinuous surface on a reconstructed 3D object (Fig. 6 (a)), because scene structure is usually not parallel to the camera trajectory under arbitrary motion, particularly when this includes rotation. Devernay and Faugeras proposed deformable window matching [12] using a "slanted-plane" assumption. Their method detects not only the disparity but also the slant of the captured plane by deformable window matching. Using this "slanted-plane" assumption, a discontinuous surface can be avoided (Fig. 6 (b)).

The matching method is, however, unstable or unable to detect color drift in a texture-less area. Previous research used a Bayesian approach to detect stereo disparity [13-15] to solve this problem. Li [16, 17] proposed a belief propagation method applied in conjunction with deformable window matching. We use this method for estimating depth.

Estimation by belief propagation (BP) has high computation cost, because it makes use of recursion and our target image has a particularly high resolution. Consequently, we use a hierarchical structure for BP estimation, starting at the lower layers and proceeding to the higher layers as shown in Fig. 8. This recursive processing of the hierarchical belief propagation realizes a more stable and faster estimation even with a higher resolution. Once the estimations at the highest layer (original size of image) have been completed, we calculate the depths from the final estimated color drifts  $d$  on each pixel.

In the following sections, we explain each process in detail.

#### 3.2 Detecting Color Drifts by Deformable Window Matching

The input color route panorama is first divided into RGB channel images and then the method searches for a corresponding window texture among the RGB channels. We assume that most parts of a scene are gray and therefore there is a similarity between the textures of the color channels.

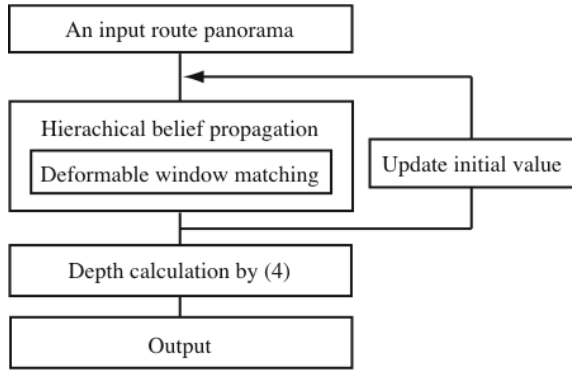


Fig. 5 Color drift of a captured image

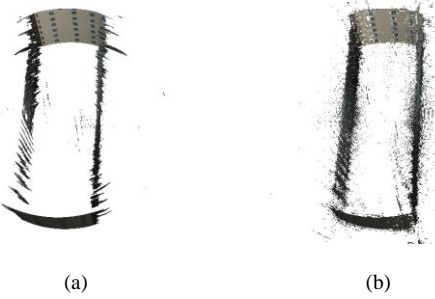


Fig. 6 Comparison between the “font-parallel” assumption and “slanted-plane” assumption: (a) result of “font-parallel” assumption using standard BP [14], (b) result of “slanted-plane” assumption using our method

A color drift  $d$  appears on the  $x$ -axis of the route panorama when the camera moves horizontally. We set the reference window on the G channel image and search for a similar window with a  $-d$  shift on the R channel and a  $+d$  shift on the B channel to detect the size of the color drift. In this process, the search windows on both the R and B channels are deformed with respect to the slant of the object surface [12]. In other words, disparity  $d$  is combined with slant parameters  $\partial d / \partial u$  and  $\partial d / \partial v$ , causing the disparity to have different values at each point in the window. This is expressed as:

$$d(u, v) = d + \frac{\partial d}{\partial u}(u - u_c) + \frac{\partial d}{\partial v}(v - v_c), \quad (12)$$

where  $(u, v)$  are the coordinates of the window field and  $(u_c, v_c)$  the center point of the window.

We then use the average of the normalized correlation coefficient (NCC) between the R-G and B-G images as the similarity function, which is defined as:

$$E(d) = \frac{C(I_R, I_G, d) + C(I_G, I_B, d)}{2}, \quad (13)$$

where

$$C(I, I', d) = \frac{\sum_{u,v}^W (I(u, v) - \bar{I}) \cdot (I'(u + d(u, v), v) - \bar{I}')}{\sqrt{\sum_{u,v}^W (I(u, v) - \bar{I})^2} \cdot \sqrt{\sum_{u,v}^W (I'(u + d(u, v), v) - \bar{I}')^2}}. \quad (14)$$

$I(u, v)$  indicates the intensity at point  $p(x, y | x = u, y = v)$  of the route panorama,  $\bar{I}$  is the average intensity, and  $W$  denotes the number of pixels in the search window. Suppose  $d(u, v) \in [d_k, d_{k+1})$ , where  $d_k$  and  $d_{k+1}$  are integer values; then  $I(u + d(u, v), v)$  has an interpolated value between  $I(u + d_k, v)$  and  $I(u + d_{k+1}, v)$ . We estimate the color drift  $d$

for every point  $p(x, y)$  of a route panorama and use a direction set method [18] to find the floating point  $\{d, \frac{\partial d}{\partial u}, \frac{\partial d}{\partial v}\}$  that gives the best score  $E(d)$ .

### 3.3 Deformable Window Matching with Belief Propagation

The deformable window matching method derives color drift and slant parameters  $\{d, \frac{\partial d}{\partial u}, \frac{\partial d}{\partial v}\}$ . We use parameters  $\partial d / \partial u$  and  $\partial d / \partial v$  to calculate the compatibilities to extend BP to include the slanted surface assumption.

Fig. 7 illustrates the Markov network model used in our problem. Each node of the Bayesian-net has hidden variables corresponding to color drifts,  $D = \{d_s\}$ . We also denote the intensities of a route panorama as observed variables,  $I = \{i_s\}$ . Then, the posterior  $P(D | I)$  can be factorized as:

$$P(D | I) \propto \prod_s \phi_s(d_s, i_s) \prod_{s, t \in N(s)} \psi_{st}(d_s, d_t), \quad (15)$$

where  $\phi_s(d_s, i_s)$  is called the local evidence for node  $d_s$ , and  $\psi_{st}(d_s, d_t)$  the compatibility matrix between nodes  $d_s$  and  $d_t$ .  $N(s)$  is a group of neighbor nodes around  $s$ . In our problem, the local evidence is formulated as:

$$\phi_s(d_s, i_s) = (1 - e_e) \exp\left\{-\frac{1 - E(d_s)}{\sigma_e}\right\} + e_e, \quad (16)$$

where  $E(d_s)$  is obtained from (13).

According to [16], for node  $s$  with label  $d_s$ , a general planar surface model at  $d_s$  is:

$$d_t(x, y) = d_s + \frac{\partial d_s}{\partial x}(x - x_s) + \frac{\partial d_s}{\partial y}(y - y_s), \quad (17)$$

where slant parameters  $\partial d_s / \partial x$  and  $\partial d_s / \partial y$  are derived from the deformable window matching as described in Section 3.2. The slant parameters represent the fact that for contextual information it is preferable for a neighboring label  $d_t$  to lie on the same planar surface (preferably not frontal parallel) with the same surface normal. The compatibility is formulated as:

$$\begin{aligned} \psi_{st}(d_s, d_t) = & (1 - e_p) \exp\left\{-\frac{\left|d_s - d_t - \frac{\partial d_t}{\partial x}(x_s - x_t) - \frac{\partial d_t}{\partial y}(y_s - y_t)\right|}{\sigma_p}\right\} \\ & + e_p \cdot \left((1 - e_N) \exp\left\{-\frac{\|\mathbf{N}_{d_s} - \mathbf{N}_{d_t}\|^2}{\sigma_N}\right\} + e_N\right) \end{aligned} \quad (18)$$

where  $\mathbf{N}$  is the surface normal in the disparity space, computed as:

$$\mathbf{N} = \frac{\left(-\frac{\partial d}{\partial x}, -\frac{\partial d}{\partial y}, 1\right)^T}{\sqrt{\left(\frac{\partial d}{\partial x}\right)^2 + \left(\frac{\partial d}{\partial y}\right)^2 + 1}}. \quad (19)$$

When the underlying model at  $d_s$  is a frontal parallel plane,  $\frac{\partial d_s}{\partial x} = \frac{\partial d_s}{\partial y} = 0$  and  $\mathbf{N}_{d_s} = \mathbf{N}_{d_t}$ , and the above compatibility can be simplified.

We obtain estimations of the color drift  $d_s$  to maximize the posterior functions at all nodes and then use a BP method [13] to solve the maximum of a posteriori estimation.

### 3.4 Hierarchical Belief Propagation

The estimation results are affected by the initial values for the BP. We use a large network for modeling the image and thus BP requires many iterations to estimate a high-resolution image because it propagates the messages simultaneously to all the neighboring nodes (that is, nodes one pixel distant). Moreover, the estimations of a texture-less area are very sensitive to image noise because the local evidence consists only of small values on a node in this area. This problem arises because BP estimates a global optimization of huge networks by propagating local messages.

We use the hierarchical BP method with a Gaussian pyramid (GP) [9]. The GP is a group of multi-resolution images that are hierarchically half the size of a low-pass filtered version of the original image. We reduce the computation cost of the BP by using lower to higher resolution images. However, since some color drift disappears on a lower resolution image as a result of low-pass filtering, there is a limit to downsizing the images to reduce computation cost.

Therefore, we use a different approach with a hierarchical structure of propagation (instead of a hierarchical image), as shown in Fig. 8. In the lower layer, we just downsize the number of estimates, i.e., the nodes in the Bayesian-net, instead of creating lower resolution images. Each estimate of the nodes, however, uses an original sized image to detect the drift.

We iteratively estimate the color drift  $d$  from the lowest layer to the highest layer, as shown in Fig. 8. The optimizations using BP on the lower layers are faster and much more stable than those on the higher layers. Conversely, a higher resolution is obtained for the estimations on the higher layers. Therefore, we combine the advantages of the estimation of the lower and higher layers using the hierarchical approach. This approach efficiently optimizes a high-resolution image stably with low computation cost.

The initial value of the message  ${}^0m_{st}^0$  is initialized as a uniform probability in the first layer. The initial values  ${}^n m_{st}^0$  in layer  $n$  are initialized by the  $T$ th iterative result of the corresponding message  ${}^{n-1}m_{st}^T$  in the previous layer  $n-1$ . BP is faster and more stable in its estimations of color drifts using hierarchical belief propagation.

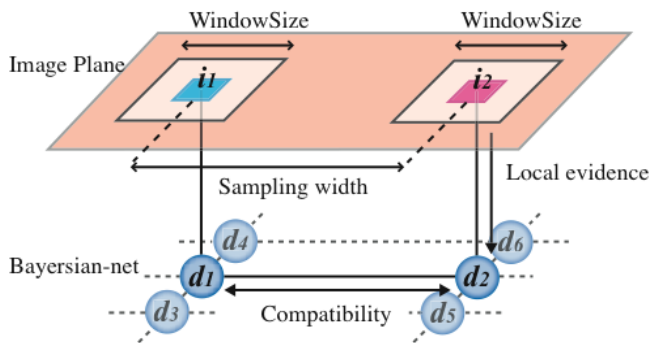


Fig. 7 Markov network model

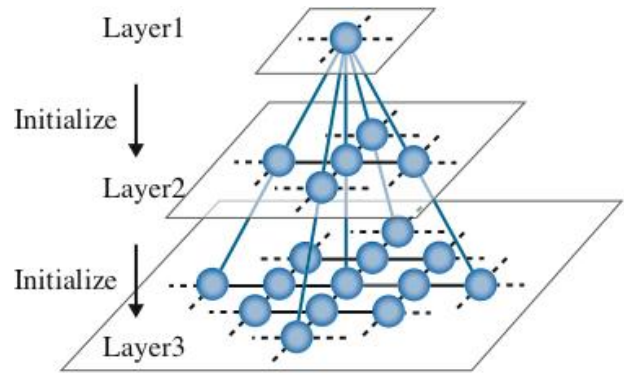


Fig. 8 Hierarchical Bayesian-net

## IV. EXPERIMENTS

### 4.1 Simulations

We evaluated our proposed method using a simulated image. The simulated input route panorama was generated by 3D CG software (Lightwave 3D: NewTeck Inc.).

We set the resolution of the first camera to  $2700 \times 1$  pixels. We also used two more cameras corresponding to the R and B sensors to emulate a color line camera. The angular difference  $\theta$  between the RGB lines was set to 0.105 degrees. We carried out the simulated experiment using both linear and circular motions as described in (8) and (10), respectively. As a result of the camera movement, we obtained multiple view line images that were stitched together to obtain the route panorama.

The simulated object, of size  $W30m \times H15m \times D5m$ , is shown in Fig. 9 (a) and (i). Fig. 9 (b) and (j) show the input route panoramas with linear translation and circular motion, respectively. Both images consist of  $2700 \times 4800$  pixels. Figs. 9 (c) and (k) show the grand truth as a depth map for each route panorama, while Figs. 9 (e) and (m) show the depth maps estimated using only the deformable window matching. Fig. 9 (f) and (n) show the depth maps estimated using our proposed method combined with BP. Fig. 9 (g), (h), (o) and (p) show the 3D reconstruction results. These depth map images give the distance from the camera as intensities; brighter pixels indicate a greater depth, while darker ones indicate a lesser depth.

The average errors between the estimated and grand truth depth maps in each case, and theoretical depth resolutions are listed in Table 1. Compared with the results of the method using the deformable matching only, the estimation accuracy of our method, which is combined with BP, is significantly better. The results also indicate that BP works well for global optimizing of estimations.

As can be seen in the depth maps of Fig. 9, there are some strange values on the window area of the building object. These were caused by specular reflections on the window. Since we use texture for estimating the depth, the problem is unavoidable as is the case in previous image based depth estimations. However, the remaining portion of the object was recovered smoothly, and we are able to visualize the outline of the object from the estimated depth map.

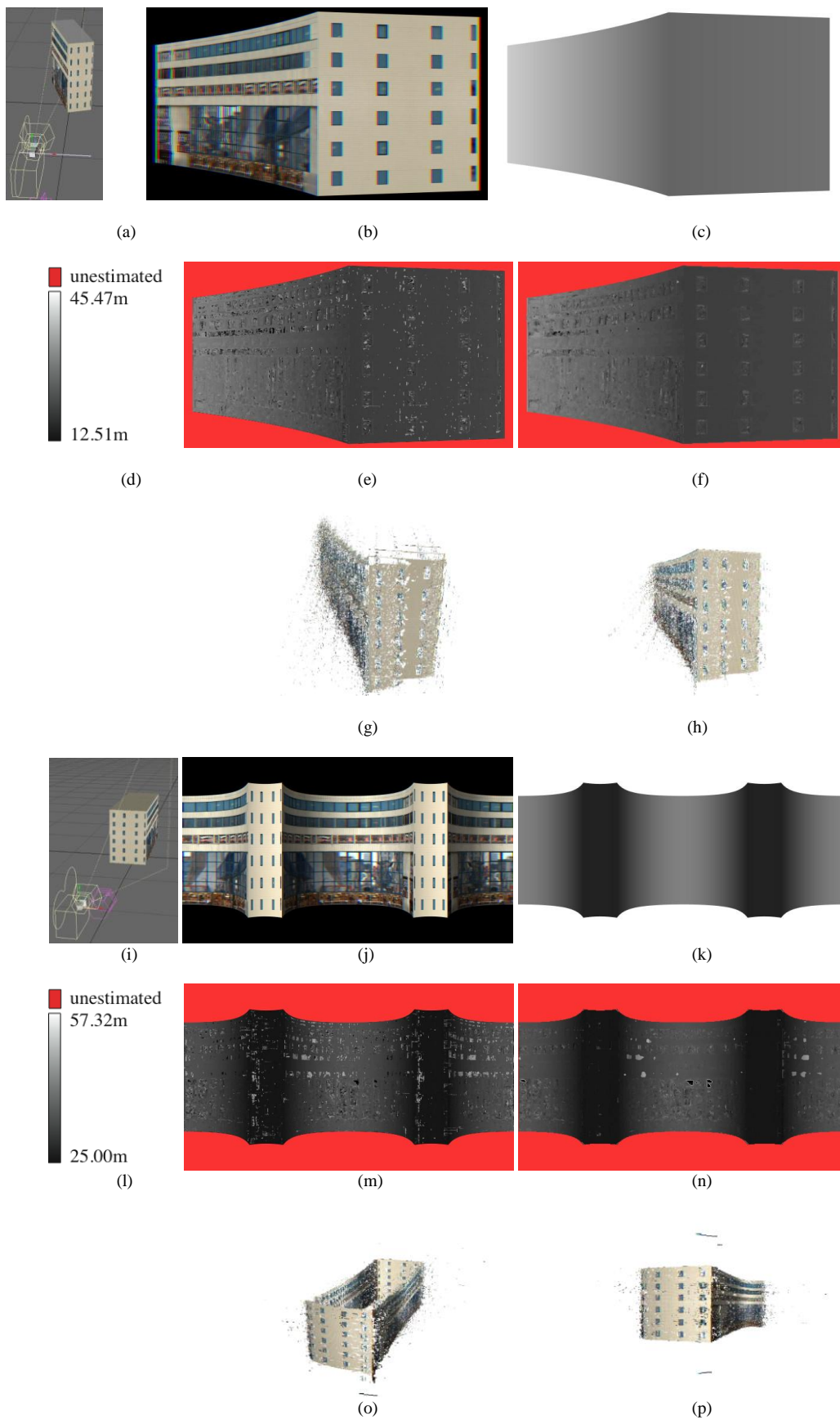


Fig. 9 Simulation results. (a)-(h) are results of linear motion. (i)-(p) are results of circular motion: (a)(i) simulated objects, (b)(j) route panoramas, (c)(k) ground truths, (d)(l) depth ranges, (e)(f)(m)(n) estimated depth maps and (g)(h)(o)(p) 3D reconstruction results (e)(m) are results by using only deformable window matching, (f)(n) are results by using our method. (Color Plate 5,6)

#### 4.2 Actual Experiments

We also carried out actual experiments using a color line camera (NSCL2700D: NED), the specifications of which are listed in Table 2. We used a lens (Nikkor50mm: Nikon) attached to the camera with the angular difference  $\theta$  set to  $3.67$

$\times 10^{-2}$  [deg] for the camera and lens combined. We also used both linear and circular motions in these experiments.

**Linear motion:** To achieve linear motion, we used an electric moving cart to translate the camera in front of a real building on our university campus. Fig. 10 (a) shows the input route panorama image. The resolution of the route panorama is

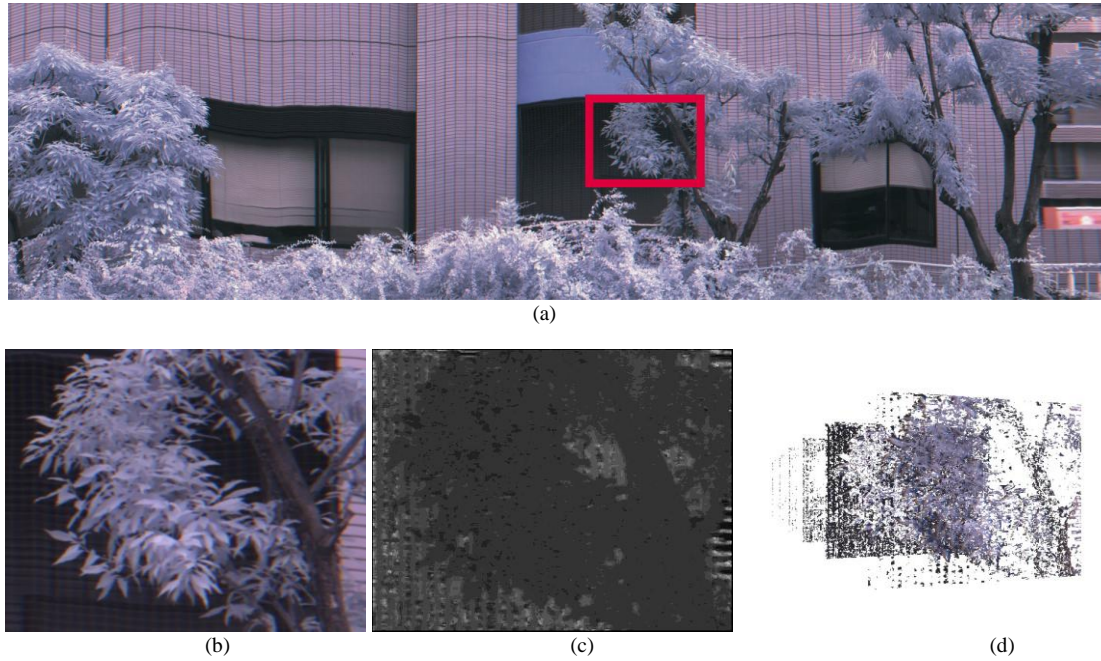


Fig. 10 Results obtained from linear motion: (a) high-resolution route panorama, (b) zoomed-in portion of (a), (c) estimated depth map and (d) 3D reconstruction result

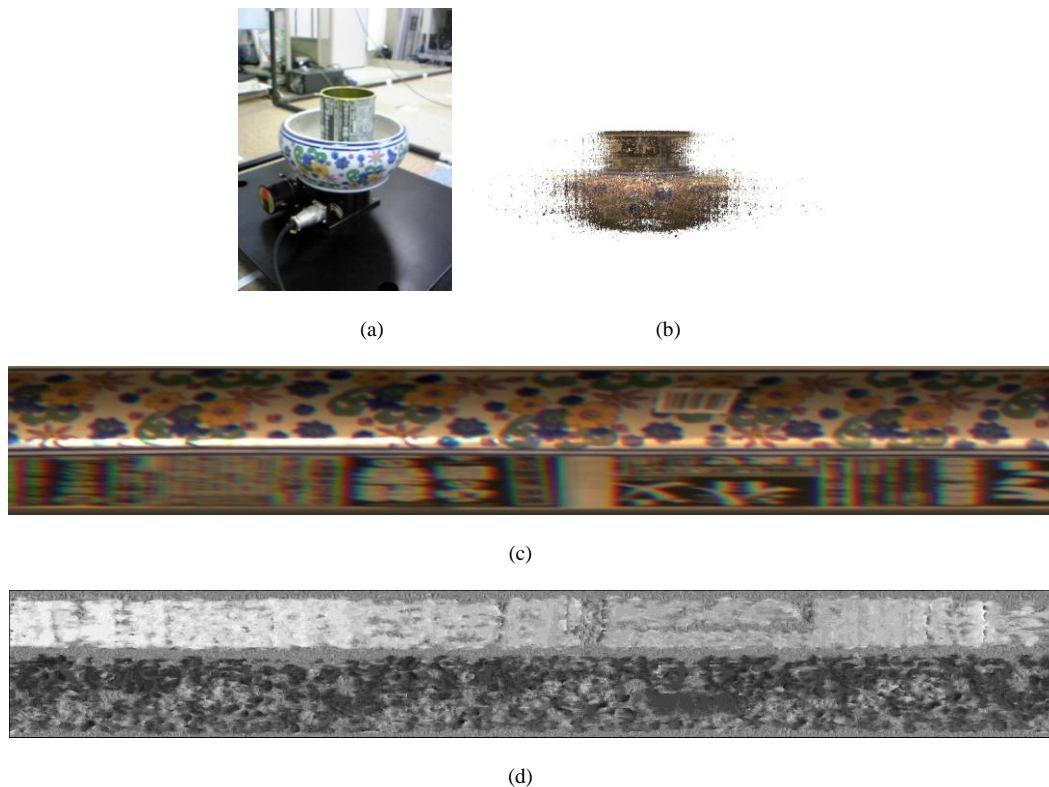


Fig. 11 Results obtained from circular motion: (a) modeling object, (b) 3D reconstruction result, (c) route panorama and (d) 3D reconstruction result

2700 × 10000 pixels. Fig. 10 (b) shows a zoomed-in portion of the image. The detail is clearly visible even though the image is zoomed-in, thereby confirming that a line camera can capture higher resolution images than common area cameras.

Fig. 10 (c) shows the results of the depth map for the zoomed-in area. The difference in depth between the tree in the foreground and the wall of the building behind is clearly visible. Fig. 10 (d) shows the result of the 3D reconstruction with a texture image. Both the high-resolution texture and depth information were generated from a single route panorama image. Distortions, due to sensor pitching, are evident in the route panorama. This problem is however, easy to solve using additional motion sensors such as a gyro. We will address this problem in future work.

**Circular motion:** To achieve circular motion, we set up a stationary camera and rotated an object, instead of moving the camera. We used an electric rotary stage to control the motion with known trajectory. This is a reasonable setting for capturing small objects such as vases, cups, dolls, etc.

To ensure that the depth differences are absolutely clear, we placed a can inside a vase. Fig. 11 (a) shows the setting used to capture the input route panorama, while Fig. 11 (c) shows the input route panorama itself. The estimated depth map image of Fig. 11 (c) is shown in Fig. 11 (d). Fig. 11 (b) depicts the result of the 3D reconstruction. As highlighted by Fig. 11 (b), the depth difference between the inside can and outside vase is clearly visible. Furthermore, the proposed method recovered the bowl-shape of the vase.

The results given in Figs. 10 and 11 confirm that the proposed method is effective with real scenes or objects using either linear or circular motion.

## V. CONCLUSIONS

We have proposed a general model for capturing a route panorama with a color line camera and a depth estimation method for the route panorama using an arbitrary camera trajectory. A color line camera with only one line CCD for each color channel was used to capture the route panorama. The sensor was able to obtain a high-resolution route panorama effectively. We also explained the geometry of the route panorama and the principle of recovering depth from color drift. We proposed that the depth estimation algorithm integrates hierarchical belief propagation and deformable window matching to estimate color drift corresponding to depth stably. We confirmed that the proposed method is able to recover depth using only the color drift of a panorama, and that the accuracy of the depth is sufficient for use in creating 3D models. We used two example trajectories, linear and circular camera motion, in both the simulations and real experiments. The proposed method is, however, not restricted to these simple examples and as such, we will consider trajectory analysis and optimization of an object in future work. We claimed a trade off between the redundancy of images and the FOV relation. We used a 1-line color camera as an extreme example of a narrow FOV camera to reduce redundancy. Our concept and method are however, not restricted to only the 1-line color camera. The proposed depth estimation method

can also be applied to several lines or regular cameras for a much more stable estimation and to avoid the gray world assumption.

TABLE 1: COMPARISON OF RMS ERRORS.

	Linear motion	Circular motion
Deformable window matching without BP	2.86 [m]	2.09 [m]
Deformable window matching with BP	1.64 [m]	0.82 [m]
Theoretical depth resolution	1.14 [m]	1.08 [m]

TABLE 2: SPECIFICATIONS OF THE LINE CAMERA.

Number of lines	3 lines (RGB channels)
Resolution	2700 pixels
Scan rate	1780 Hz
Line gaps	32 $\mu\text{m}$
Intensity levels	8 bit (256 levels)

## REFERENCES

- [1] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.
- [2] R. Bolles, H. Baler, and D. Marimont. Epipolar plane image analysis: *An approach to determining structure from motion*, *IJCV*, vol.1, pp.7-55, 1987.
- [3] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: *a factorization method*, *IJCV*, vol.9, no.2, pp.137-154, 1992.
- [4] A. Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding, *IEEE PAMI*, vol. 16, Issue 2, pp.150 - 162, 1994.
- [5] H. Nagahara, Y. Yagi and M. Yachida. SuperResolution Modeling Using an Omnidirectional Image Sensor, *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS; PART B: CYBERNETICS*, vol.33, no.4, pp.607-615, Aug, 2003.
- [6] H. Kawasaki, K. Ikeuchi, M. Sakauchi. Super-resolution omnidirectional camera images using spatio-temporal analysis, *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 89 Issue 6, pp. 47--59, 2006.
- [7] J. Y. Zheng. *Digital Route Panoramas*, *IEEE Multimedia*, vol.10, no.3, pp.57-68, 2003.
- [8] Z. Zhu and A. R. Hanson. LAMP: *3D layered, adaptive-resolution, and multi-perspective panorama-a new scene representation*, *CVIU*, vol. 96, pp.294-326, 2004.
- [9] Cyberware Laboratory Inc. <http://www.cyberware.com>.
- [10] Google map <http://maps.google.co.jp/>.
- [11] H. Nagahara, A. Ichikawa, M. Yachida. Depth Estimation from the Color Drift of a Route Panorama, *Proc. Int. Conf. Intelligent Robotics and Systems*, pp. 4066-4071, 2008.
- [12] F. Devernay and O. D. Faugeras. Computing differential properties of 3-d shapes from stereoscopic images without 3-D models, *In Proc. CVPR*, 1994
- [13] P. N. Belhumeur. *A Bayesian approach to Binocular Stereopsis*, *IJCV*, vol.19, no.3, pp.237-260, 1996.
- [14] J. Sun, N. Zheng, and H. Shum. *Stereo Matching Using Belief Propagation*, *ECCV, LNCS 2351*, pp. 510-524, 2002
- [15] J. Sun, N. Zheng, and H. Shum. *Stereo Matching Using Belief Propagation*, *IEEE PAMI*, vol.25, no.7, pp.787-800, 2003.
- [16] G. Li and S. W. Zucker. Stereo for Slanted Surfaces: *First Order Disparities and Normal Consistency*, *EMMCVPR, LNCS 3757*, pp. 617-632, 2005



- [17] G. Li and S. W. Zucker. *Surface Geometric Constraints for Stereo in Belief Propagation*, CVPR, 2006
- [18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C (2<sup>nd</sup> Ed.)*. Cambridge University Press, 1992.
- [19] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2<sup>nd</sup> Ed.)*, San Francisco: *Morgan Kaufmann Publishers*, 1988.



**Ryuji Shibata** is a Masters student in Systems Science and Applied Informatics at Osaka University. He received his Bachelor degree in Systems Science from Osaka University in 2008. His research interests lie in the field of computer vision. His current research focus is on depth estimation and defocusing of photographs.



**Hajime Nagahara** is an Assistant Professor at the Graduate School of Engineering Science, Osaka University, Japan. He received B. E. and M. E. degrees in Electrical and Electronic Engineering from Yamaguchi University in 1996 and 1998, respectively. He received his Ph.D. in System Engineering from Osaka University in 2001. He was a Research Associate of the Japan Society for the Promotion of Science between 2001-2003. He was a Research Associate of the Graduate School of Engineering Science, Osaka University, between 2003-2006. He was a Visiting Associate Professor at CREA University of Picardie Jules Verns, France in 2005. He was a Visiting researcher at Columbia University, USA in 2007-2008. Image processing, computer vision and virtual reality are his research fields. He received an ACM VRST2003 Honorable Mention Award in 2003.