Range data acquisition using color structured lighting and stereo vision

Chu-Song Chen[a], Yi-Ping Hung[a,*], Chiann-Chu Chiang[a], Ja-Ling Wu[b]

[a]*Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan*
[b]*Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan*

Abstract

This paper presents a new color-lighting/stereo method for 3D range data acquisition by combining color structured lighting and stereo vision. A major advantage of using stereo vision together with color stripes lighting is that there is no need to solve the problem of finding the correspondence between the color stripes projected by the light source and the color stripes observed in the images. That is, the more difficult problem of finding the correct color stripe correspondence problem between the light source and the image is replaced by an easier image-to-image stereo correspondence – which is not only easier than the above lighting-to-image correspondence problem, but also easier than the traditional stereo correspondence because a good color pattern has been projected onto the object. Another advantage of using stereo vision is that there is no need to calibrate the position and orientation for each of the projected light stripes in 3D space. In this work, a pattern of color stripes is projected onto the objects when taking images, after which edge segments are extracted from the acquired stereo image pair, and then used for finding the correct stereo correspondence. A systematic procedure is proposed in this paper for generating good color stripe patterns. To find the correct stereo correspondence, a global search method based on intra-scanline dynamic programming is adopted. A winner-take-all scheme using edge-based inter-scanline consistency is then proposed to refine the results obtained from intra-scanline search. Experimental results have shown that the proposed method can successfully generate a dense range map with only one pair of stereo images.

Keywords: Range data acquisition; Stereo vision; Structured lighting

1. Introduction

Aquisition of 3D range data has received considerable attention in recent years [1-7]. The most popular approach in structured lighting is to use a single-line stripe. The advantage of this approach is that it greatly simplifies the matching problem, but this approach has the drawback that only one single line of 3D data points can be obtained with each image shot at one time. To speed up the acquisition of 3D range data, one can adopt a multiple-line stripe pattern instead. However, the matching problem in the multiple-line case becomes much more difficult. One possibility is to use color information to simplify this difficult matching problem [1,8,9]. In this work, we have successfully developed and implemented a new method for 3D range data acquisition by combining color structured lighting and stereo vision. In the proposed range data acquisition system, we project a pattern of color stripes onto the objects because chromatic information can help a lot in solving the

* Corresponding author. Email: hung@iis.sinica.edu.tw

correspondence problem. Once the correspondence problem is solved, the 3D range data can then be computed by using triangulation. To determine stereo correspondence correctly, the projected color pattern should provide enough texture information for stereo matching. Such a good color pattern can be generated by the procedure described in Section 2.

Among the existing methods for range image acquisition, the technique proposed by Boyer and Kak [1] is most similar to our approach. However, their approach used only one single camera together with a calibrated color light source. In a single-camera approach, it is required to find the correspondence between the color stripes projected by the light source and the color stripes observed in the image. In general, due to different reflection properties (or surface albedos) of the object surface, the color of the stripes recorded by the camera is usually different from that of the stripes projected by the light source – even in the case that the objects are perfectly Lambertian. For example, a green light stripe projected by the light source may be observed as a blue one after reflection. Hence, it is difficult to solve the above

lighting-to-image correspondence problem in many practical applications. On the other hand, this does not affect our color-lighting/stereo approach if the object is Lambertian, because the color observed by the two cameras in our system will be the same, even though this observed color may not be exactly the same as the color projected by the light source. Therefore, by adding one more camera, we can replace the more difficult problem of lighting-to-image correspondence by an easier problem of image-to-image stereo correspondence. Here, stereo correspondence problem is also easier than the traditional stereo correspondence problem because a good color pattern has been projected onto the object. Notice that, for the Boyer and Kak approach, the above color correspondence problem between projection and observation becomes even more severe when the object surface has richer texture or color information. As for our color-lighting/stereo approach, the color and texture of the object surface will not hold up the stereo matching process so long as they do not blot out the texture pattern projected by the structured lighting.

Another advantage of our approach is that there is no need to calibrate the position and orientation for each of the projected light stripes in 3D space. In other words, camera is the only device needed to be calibrated. Since, in any case, we have to calibrate at least one camera for most triangulation methods, the effort of calibrating the second camera is trivial. On the other hand, accurate calibration of the structured light pattern in 3D space is not an easy task in general.

Compared to the passive-stereo approach, our color-lighting/stereo approach can produce artificial textures on the object surfaces by projecting a color stripe pattern. Hence, the stereo-matching problem can be solved in a more stable manner. Notice that for any range acquisition method using triangulation (either the single-camera approach or the stereo approach), occlusion is always a significant issue to be considered and is, in principle, difficult to solve. In our system, a global stereo matching technique (i.e. the intra-scanline search method) is used for stereo matching, and a new cost function has also been proposed for dealing with occlusion. With the proposed method, the problem of occlusion can be solved to some extent. In this work, edge-based stereo matching is used because the color structured pattern is designed to contain many sharp edges. These sharp edges are utilized to improve the accuracy of our range data acquisition system.

This paper is organized as follows. Section 2 presents a systematic method for generating a good color structured pattern. In Section 3, the intra-scanline search followed by a winner-take-all scheme using edge-based inter-scanline consistency is proposed for reliable stereo matching. Some experimental results are shown in Section 4. Section 5 gives some conclusions and discussions.

2. Generation of color structured pattern

Experimental results [8] have shown that HSI model [10] is a good color model for stereo matching. Hence, we use the HSI model for both the color pattern generation and for the matching error computation in this work. In general, we want to design a color pattern which can satisfy the following two requirements: (1) the goodness of edge detection result directly affects the quality of the obtained range data because edge-based stereo vision is used. The pattern should have strong contrast at the borders of any two adjacent stripes. (2) The correlations between any two segments of a consecutive sequence of light stripes should be as small as possible, so that false stereo matching can be avoided.

The main idea used for color pattern generations is as follows. We first generate a sequence of light stripes with brighter colors, called $C_1, C_2, \ldots, C_N$. To increase the intensity variation between adjacent light stripes, a black stripe B is inserted between each C_i and C_{i+1} ($i = 1, 2, \ldots, N - 1$). Hence, the final generated pattern is $C_1, B, C_2, B, C_3, \ldots, B, C_N$.

To ensure each C_i is brighter enough such that the border between each C_i and B can be easily extracted by the edge detection procedure, the intensity value of each C_i is set to be larger than a threshold I_{min}. On the other hand, if the intensity value of C_i is too large (i.e. C_i is nearly white), the hue of C_i in the HSI model carries little information. Therefore, we also want to generate colors with intensity values smaller than another threshold I_{max}. Hence, in general, the intensity of C_i is designated to within the range $[I_{min}, I_{max}]$. A random number generator is used to generate the intensity values in this range.

A trial-and-error approach is used to select a pattern such that the correlations between any two consecutive color sequences are small enough. In our algorithm, an initial value of hue (H_{init}) and a jumping value (H_{jmp}) are first selected. To increase the randomness of the hue value for each C_i, we use a random number generator to generate a jumping value $\underline{H}_{jmp}$ lying within the range of $[H_{jmp} - \varepsilon, H_{jmp} + \varepsilon]$, where ε is a selected constant. The equation used for hue generation is

$$H_{i+1} = (H_i + \underline{H}_{jmp}) \bmod 360°, H_0 = H_{init}$$

In the above, we have described the generation of intensity and hue in the HSI model for each C_i. The next step is to determine the saturation of C_i by finding the largest possible saturation value given that $H = H_i$ and $I = I_i$. Let $C = \{(I, H, S) | I = I_i, H = H_i, i = 1, 2, \ldots, N\}$. We transform all the colors in the set C into the RGB coordinate system and check whether they are valid (i.e. all the R, G, B values are within $[0, 255]$) or not. The largest possible S which satisfies that (I_i, H_i, S) is valid in the RGB space, is then selected as the saturation of C_i.

There are some parameters which need to be specified in our color pattern generation algorithm: I_{min}, I_{max}, H_{ini}, H_{jmp} and ε. Fig. 1 shows some generated patterns associated with different parameter sets. In principle, a color sequence can be modeled as a discrete-time signal. The auto-correlation function of the signal can be computed. The quality of each generated pattern is then evaluated by using the energy of the auto-correlation function of this sequence. For our purpose, it is desired that the energy of the auto-correlation function should be as small as possible. One can then minimize this energy with respect to all the above-mentioned parameters. In our experiment, we simply ran a series of random trials, and then selected the best set of parameters, which turned out to be $I_{min} = 153$, $I_{max} = 245$, $H_{init} = 0$, $H_{jmp} = 140$ and $\varepsilon = 20$. The color stripes generated by this set of parameters is shown in Fig. 2. It can be seen that this pattern has large variations between adjacent light stripes, and the correlation between any two consecutive color sequences is quite small. In our experiment, the color pattern generated was printed on a transparency and then projected onto the objects. The following section will describe the stereo matching technique used in this work.

3. Stereo-matching

The stereo-matching method used in this paper is a variation of the intra/inter-scanline method originally proposed by Ohta and Kanade [11]. The major advantage of this method is that it is a global matching technique in which the correspondences were not only determined by local similarities, but also global similarities. There are two reasons for adopting this method in this paper. First, in our approach the projected light pattern is a set of vertical light stripes; hence, vertical edge-segments can be easily extracted from the images. Since the feature-primitives used in the Ohta and Kanade method are non-horizontal edges, their method is quite suitable for our work. Second, matching errors caused by some local similarity of the projected color stripes can be avoided, because this method uses a global search, i.e. dynamic programming, to find the optimal solution of stereo correspondences.

It is well known that the epipolar constraint can considerably reduce the searching complexity. Consider Fig. 3. For a point P in the left image plane, its correspondence point must lie on line $\overline{Q_1 Q_2}$ in the right image plane. Also, the correspondence point of Q must lie on $\overline{P_1 P_2}$. Therefore, line $\overline{P_1 P_2}$ and line $\overline{Q_1 Q_2}$ can be
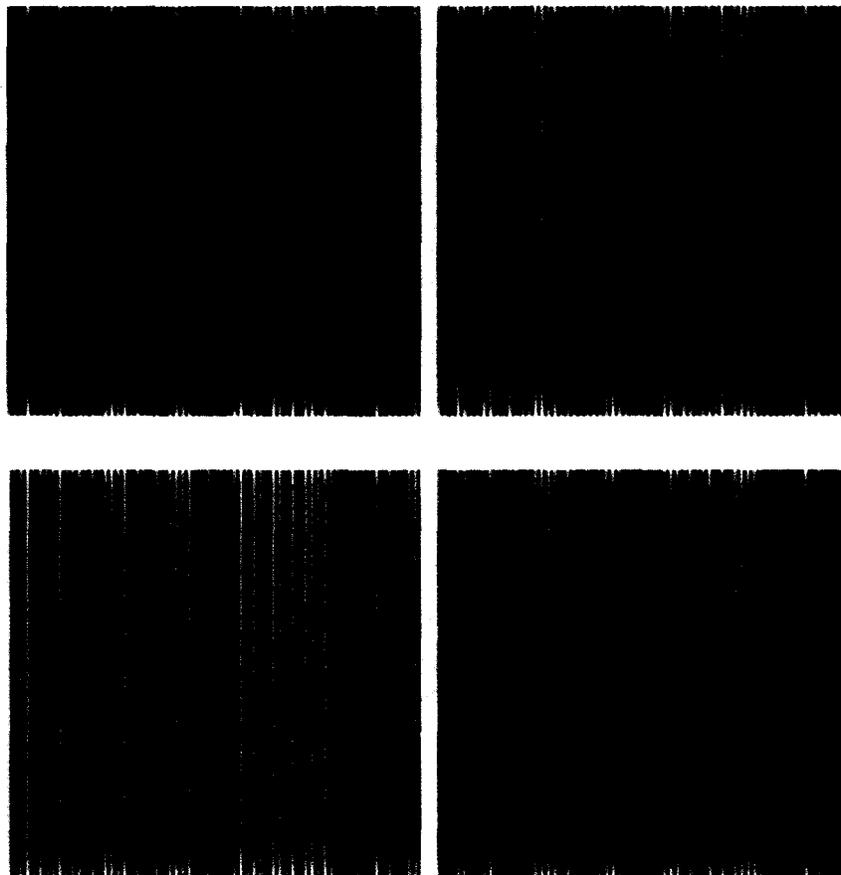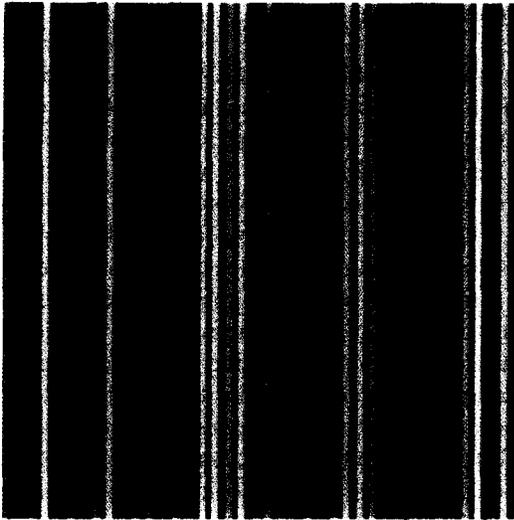

Fig. 1. Some generated color patterns.

Fig. 2. The color stripes to be projected on the objects.

regarded as a scanline pair. Searching along a scanline pair for finding the optimal stereo correspondence is called the *intra-scanline search*. To solve this searching problem, Ohta and Kanade proposed a method based on dynamic programming [11]. This method is adopted in this work for intra-scanline search, as reviewed in Section 3.1.1. However, our method has two major differences compared to that used by Ohta and Kanade [11] for intra-scanline search. First, we have redefined the cost function so that the chromatic information can be used in stereo matching, as described in Section 3.1.2. Second, we have developed a more general definition of the cost function for dealing with stereo occlusion, as described in Section 3.1.3.

Next, consider an edge extends across several scanlines. The correspondence in one scanline should have a strong dependency (along the edge) on the correspondence in its neighboring scanlines, which is called the *inter-scanline consistency*. To utilize the inter-scanline consistency, Ohta and Kanade [11] adopted a dynamic-programming-based method which is a direct

generalization of their method used in the intra-scanline search. However, the complexity of the inter-scanline search using dynamic programming is rather high. Hence, we propose a simple winner-take-all scheme to determine the match of an edge across several scanlines based on the result of the intra-scanline search, as described in Section 3.2. According to our experience, this method can refine the matching results of the intra-scanline search in an effective way with much lower cost of computation.

3.1. Intra-scanline search

3.1.1. Review of intra-scanline search using dynamic programming

In this section, we briefly review the intra-scanline search method based on dynamic programming, which has been proposed by Ohta and Kanade [11]. The intra-scanline search can be viewed as a path finding problem on a 2D search plane. As shown in Fig. 4, the horizontal axis is the intensity profile of the left scanline, and the vertical axis is the intensity profile of the right scanline. The vertical lines show the positions of edge points on the left scanline, and the horizontal lines show the positions of edge points on the right scanline. The intersections of the vertical and horizontal lines can be regarded as the matching edge pairs on the scanlines. Each intersection is referred to as a *node*. Now, the problem of finding correspondences along a scanline pair is transformed to that of finding the optimal path on the 2D search plane.

The edge points are ordered from left to right on each scanline: from 0 to N on the left scanline and from 0 to M on the right scanline. Both ends of each scanline are included as edge points for convenience. If node $\mathbf{m} = (m, n)$ and node $\mathbf{m}' = (m', n')$ are two successive points in the path, the line segment within node $\mathbf{m}$ and node $\mathbf{m}'$ is called a *primitive path*. A primitive path is a partial path which corresponds to matching of intervals delimited by edges. The nodes at the two ends of a
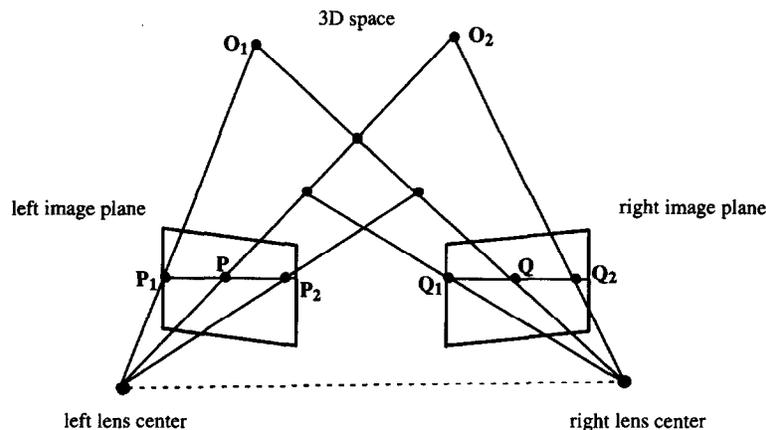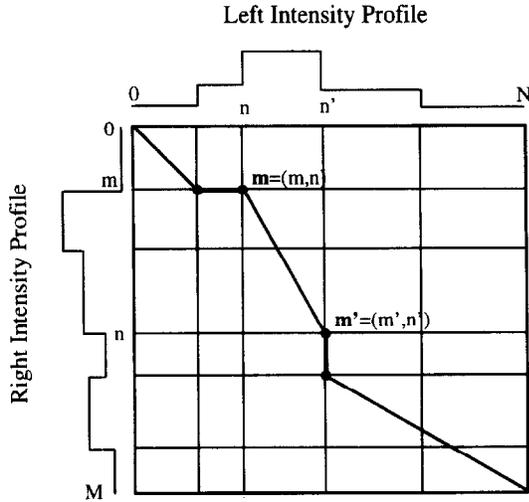

Fig. 3. The corresponding scanline pair.

Left Intensity Profile

Fig. 4. 2D search plane for intra-scanline search.

primitive path are *vertices*. A vertex at node $\mathbf{m} = (m, n)$ indicates that left edge n and right edge m are matched. The optimal matching can be viewed as the path with minimal cost. To simplify the search procedure, the ordering constraint [12] is used in the intra-scanline search. Dynamic programming can be used to compute the minimal cost of each node in the 2D-scanline search plane. The cost of a path can be recursively defined as follows: Let $C(\mathbf{m}, \mathbf{k})$ be the minimal cost of the partial path from node $\mathbf{k}$ to node $\mathbf{m}$. In particular, $C(\mathbf{m}, \mathbf{0})$ is the cost of the optimal path from the origin $\mathbf{0} = (0, 0)$ to node $\mathbf{m}$. We simplify the notation $C(\mathbf{m}, \mathbf{0})$ as $C(\mathbf{m})$ for convenience. The cost of a path is the sum of the costs of its primitive paths. Let $Cost(\mathbf{m}, \mathbf{k})$ denote the cost of the primitive path from node $\mathbf{k}$ to node $\mathbf{m}$. Then, $C(\mathbf{0}) = 0$, and $C(\mathbf{m})$ $(\mathbf{m} \neq \mathbf{0})$ can be recursively defined as

$$C(\mathbf{m}) = \min\{Cost(\mathbf{m}, \mathbf{m} - \mathbf{u}) + C(\mathbf{m} - \mathbf{u})$$

$$|\mathbf{u} \text{ is a primitive path coming to node } \mathbf{m}\},$$

where $\mathbf{u} = (u, v)$, and $0 \leq u \leq m$, $0 \leq v \leq n$.

The exact form of the cost function of the primitive path, $Cost(\mathbf{m}, \mathbf{m} - \mathbf{u})$, will be given in Section 3.1.2. At each node, the optimal primitive path $\mathbf{u}$ that gives the minimal cost is recorded. The sequence of all optimal primitive paths from node $\mathbf{0} = (0, 0)$ to node $\mathbf{m} = (m, n)$ is then the optimal path of $\mathbf{m}$.

3.1.2. Cost function using chromatic information

Now let us describe the computation of the cost of the primitive path from node $\mathbf{m_1}$ to node $\mathbf{m_2}$, i.e. $Cost(\mathbf{m_2}, \mathbf{m_1})$, used in this work. Since chromatic information is used in our approach, a new cost function is defined in this paper for the stereo matching of color images. Assume that $\mathbf{m_1} = (m_1, n_1)$ and $\mathbf{m_2} = (m_2, n_2)$. From the camera geometry, the corresponding 3D coordinates of $\mathbf{m_1}$ and $\mathbf{m_2}$, referred to as $\mathbf{P_{m1}}$ and $\mathbf{P_{m2}}$, can be easily obtained as shown in

Fig. 5. The line segment connected by $\mathbf{P_{m1}}$ and $\mathbf{P_{m2}}$ is called L. Let E_{n1} and E_{n2} be the edge points of the n_1-th and n_2-th edges in the left scanline, and E_{m1} and E_{m2} can be the edge points of the m_1-th and m_2-th edges in the right scanline, respectively. Assume that there are k pixels $a_1, a_2, \ldots, a_k$, within the interval $[E_{n1}, E_{n2}]$ in the left scanline. Given each a_i, the corresponding point b_i can be found by projecting Q_i to the right image plane, where Q_i is the intersection point of line B and line L, and B is the back-projecting line of a_i. $Cost(\mathbf{m_2}, \mathbf{m_1})$ can then be defined as

$$Cost(\mathbf{m_2}, \mathbf{m_1}) = \sum_{i=1}^{k} \mathrm{dist}(a_i, b_i)$$

where $\mathrm{dist}(a_i, b_i)$ denotes the Euclidean distance in the HSI color space [8] and can be expressed as

$$\mathrm{dist}(a_i, b_i) = \{[I(a_i) - I(b_i)]^2 + S^2(a_i) + S^2(b_i)$$

$$- 2S(a_i)S(b_i)\cos\Theta\}^{1/2}$$

and $\Theta = \min(|H(a_i) - H(b_i)|, 360° - |H(a_i) - H(b_i)|)$.

3.1.3. Cost function for dealing with stereo occlusion

Next, let us consider the case of *stereo occlusion* when defining the cost function. Stereo occlusion occurred in the situation where some parts of objects can be seen in the left (or right) image but cannot be seen in the right (or left) image. Here, we call it *right stereo occlusion* if the object surface is occluded in the right image but not in the left image, and we call it *left stereo occlusion* if the object surface is occluded in the left image but not in the right image. In the case of stereo occlusion, the primitive path will be either *vertical* or *horizontal*, as shown in Figs. 6(a) and (b). The horizontal primitive paths are caused by right stereo occlusion, and the vertical primitive paths are caused by left stereo occlusion.

First, let us consider the cost function defined for right stereo occlusion, that is, the primitive path is horizontal. Suppose a primitive path is horizontal, and from node $(m, n - v)$ to node (m, n), as shown in Fig. 6(a). Denoting U and R to be the following sets:

$$U = \{[(m, n - v), \mathbf{p}] \mid \mathbf{p} \text{ is on the upper left side of }$$

$$(m, n - v)\}$$

where $[(m, n - v), \mathbf{p}]$ is the primitive path from node $\mathbf{p}$ to $(m, n - v)$ and

$$R = \{[\mathbf{p}, \mathbf{m}] \mid \mathbf{p} \text{ is on the lower right side of } \mathbf{m}\}$$

where $[\mathbf{p}, \mathbf{m}]$ is the primitive path from node $\mathbf{m}$ to node $\mathbf{p}$. Let a be the optimal primitive path in U, that is,

$$a = \arg\min_{q \in U} \{\overline{Cost}(q)\}$$

where $\overline{Cost}(q) = Cost(q)/r$, and r is the number of pixels of q in the left image.
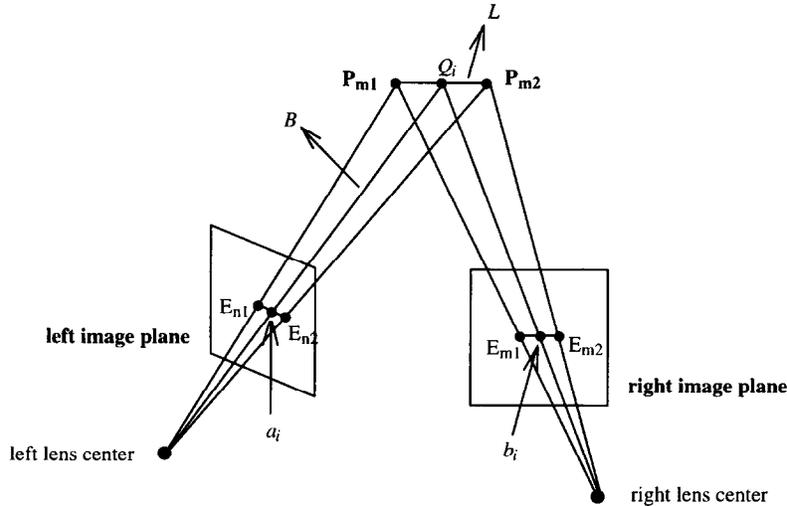
Let b be the optimal primitive path in R, that is,

Fig. 5. Determination of the correspondence point for computing the cost of a primitive path.

$$b = \arg \min_{q \in R} \{\overline{Cost}(q)\}$$

If there is an occlusion, both a and b should achieve good match, and paths drawn in dotted lines in Figs. 6(a) and (b) all have poor match. Notice that when processing node $\mathbf{m} = (m, n)$, the cost of a should have been computed, while the cost of b is not known yet. We define the cost of occlusion by the following equation:

where

$$j = \arg_i \left\{ \min_{1 \le i \le 4} \{\overline{Cost}(b_i)\} \right\}$$

$\mathbf{u} = (0, v)$ represents the primitive path coming to node $\mathbf{m}$, and th_{occ} is a threshold. Since only the paths b_1, b_2, b_3 and b_4 in R are used to get an estimate for the cost of b, a weighted sum of $\overline{Cost}(a)$ and $\overline{Cost}(b)$ is used to compute

$$Occ = \frac{\min \left\{ \min_{0 \le k < m} \{\overline{Cost}([\mathbf{m}, (k, n - v)])\}, \min_{m < l \le M} \{\overline{Cost}([(l, n), (m, n - v)])\} \right\}}{w_1 \overline{Cost}(a) + w_2 \min_{1 \le i \le 4} \{\overline{Cost}(b_i)\}}$$

where $w_1 + w_2 = 1$ and $w_1 \ge w_2$, and

$Cost(\mathbf{m}, \mathbf{m} - \mathbf{u})$

$$= \begin{cases} w_1 \overline{Cost}(a) + w_2 \overline{Cost}(b_j), & \text{if } Occ > th_{occ} \\ \infty, & \text{if } Occ \le th_{occ} \end{cases}$$

the total cost, where the weight of $\overline{Cost}(b)$ is set to be smaller than that of $\overline{Cost}(a)$. In our work, w_1 and w_2 are chosen to be 0.7 and 0.3, respectively.

Similarly, we can define the cost when the primitive path is vertical, that is, the case of left stereo occlusion. The vertical primitive path, $\mathbf{u} = (u, 0)$, coming to node $\mathbf{n} = (m, n)$ is shown in Fig. 6(b). The cost of this vertical

(a)   (b)
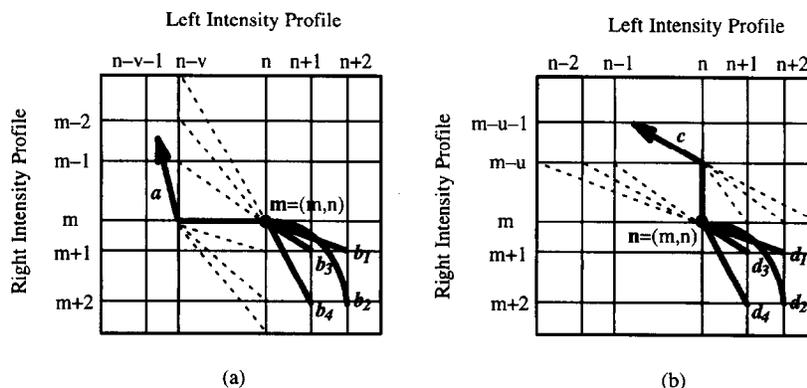
Fig. 6. The case of occlusion. (a) The primitive path is horizontal – right stereo occlusion; (b) the primitive path is vertical – left stereo occlusion.

primitive path representing occlusion is

$$
\text{Occ} = \frac{\min\left\{ \min_{0 \leq k < n} \{\overline{Cost}([\boldsymbol{n}, (m - u, k)])\}, \min_{n < l \leq N} \{\overline{Cost}([\{m, l\}, (m - u, n)])\} \right\}}{w_1 \overline{Cost}(\boldsymbol{c}) + w_2 \min_{1 \leq i \leq 4} \{\overline{Cost}(\boldsymbol{d}_i)\}}
$$

where

$Cost(\boldsymbol{n}, \boldsymbol{n} - \boldsymbol{u})$

$$
= \begin{cases} w_1 \overline{Cost}(\boldsymbol{c}) + w_2 \overline{Cost}(\boldsymbol{d}_j), & \text{if Occ} > th_{occ} \\ \infty, & \text{if Occ} \leq th_{occ} \end{cases}
$$

and

$$
j = \arg_i \left\{ \min_{1 \leq i \leq 4} \{\overline{Cost}(\boldsymbol{d}_i)\} \right\}
$$

3.1.4. Reduction of the computational complexity

In general, we can apply some geometric constraints [12] to reduce the complexity of computation. By limiting the maximal range of disparity, there are forbidden zones on the search plane, as shown by the region within the back slash lines in Fig. 7. Another useful constraint is to limit the maximum number of edges that can be skipped by a primitive path. Because we use good projected stripes pattern, the SNR is high and the detected edges are quite stable. Therefore, the possibility of having many noisy edges between two consecutive true edges is very low. Hence the search region with respect to node **m** can be further reduced to be within region A, as shown in Fig. 7. With these constraints, the time required for matching can be considerably reduced.

3.2. Inter-scanline consistency

So far, the pairs of corresponding scanlines in the left and right images have been processed independently.
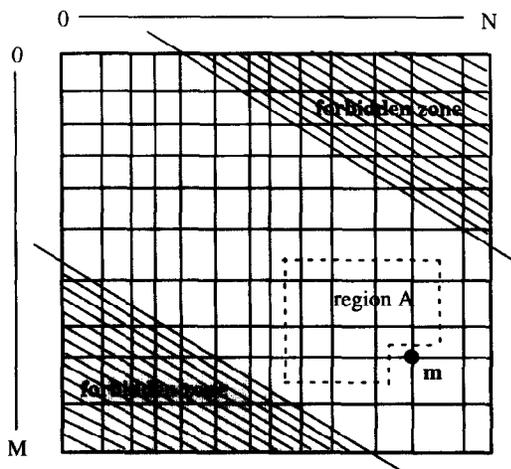

Fig. 7. Forbidden zones and the search region.

However, as mentioned above, if there is an edge extending across several scanlines, the correspondences in one scanline have a strong dependence on the correspondences in the neighboring scanlines. In the following, a new method based on a voting scheme is proposed to deal with this problem.

The inter-scanline consistency is used to refine the results of stereo correspondence obtained by the intra-scanline search. Since the inter-scanline consistency deals with connected edges, we have to label and link the isolated edge points into connected edges first. In this work, the principle used for inter-scanline consistency is a *winner-take-all* scheme. As shown in Fig. 8(a), if most points on edge *l* are corresponding to the points on edge *r* in the intra-scanline search, the edge *r* is treated as the correct correspondence for all points on edge *l*. That is, edge *r* is called the *winner* for edge *l*. Let us consider two typical cases in Fig. 8(b). In case 1, edge E_l in the left image is shorter than its winner E_r in the right image. Since most pixels on edge E_l are corresponding to edge E_r, we assert that each pixel on edge E_l should correspond to one pixel on edge E_r in the same scanline. Then, for those pixels on edge E_l whose correspondences are not E_r, their correspondence will be corrected to be E_r. In case 2, edge E_l in the left image is longer than its winner E_{r1} in the right image. Under this circumstance, we only correct those pixels of edge E_l within t_1 and t_2. After this, the other pixels of edge E_l within t_2 and t_4 need to go through the winner-take-all procedure again, until all pixels on edge E_l have been processed. The detailed algorithm for edge-based inter-scanline consistency is shown in Algorithm 1.

Algorithm 1. *Edge-based inter-scanline consistency*

Step 1. Find a connected edge which has not been processed yet in the left image, and call it edge E_l.

Step 2. Find the *winner edge* in the right image, namely, E_{r1}, of E_l. That is, most points on edge E_l correspond to the points on edge E_{r1} in the intra-scanline search.

Step 3. Assume that edge E_{r1} starts at scanline t_1 and ends at scanline t_2, where $t_1 < t_2$, and edge E_l starts at scanline t'_1 and ends at scanline t'_2, where $t'_1 < t'_2$. Then, $t_s = \max(t_1, t'_1)$ and $t_e = \min(t_2, t'_2)$.

Step 4. On edge E_l, find each point p whose corresponding point in the right image is not on E_{r1} within scanline t_s and t_e. Assume that point p is on scanline t. Let q be the intersection point of E_{r1} and
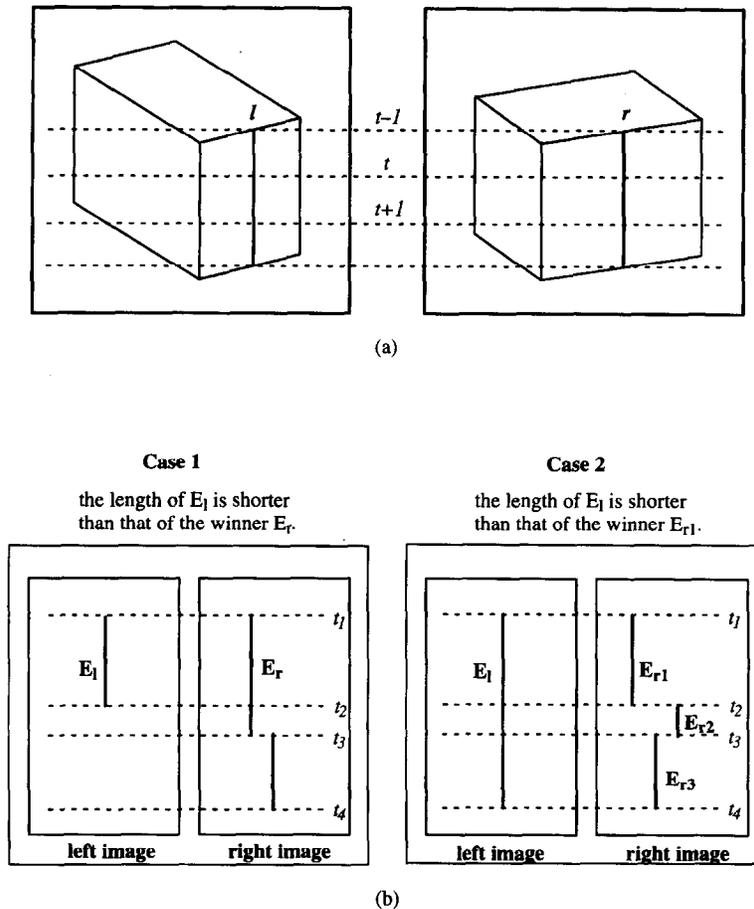
(a)

(b)

Fig. 8. (a) The edge-consistency constraint; (b) two different cases of the winner-take-all scheme using edge-based inter-scanline consistency.

scanline t, in the right image. Then point p is corrected to correspond to point q.

Step 5. Mark all points on edge E_l within scanline t_s and t_e as 'processed'.

Step 6. If there are points in the left edge image that are not 'processed', go to *Step 1*, else stop.

4. Experimental results

In our experiment, the designed color pattern is first printed on a transparency, and then the pattern is projected onto the object by an overhead projector. The camera calibration method used in this work is the one proposed by Shih *et al.* [13], where the unit used for the *object coordinate system* (OCS) is millimeters, and for the *image coordinate system* (ICS) is pixels. The grabbed stereo images are first transformed into *epipolar-corrected image pairs*, where the same row in the left and right images correspond to the same scanline. To extract the object by eliminating the background of images, an auto-thresholding method [14] is first performed on the intensity image. Then, the edge is extracted by Canny's method [15]. Edge linking is done for edge elements that are within ±45° with respect to the

vertical axis, since the projected light pattern is nearly vertical in the captured images. The 8-connected neighbors [16] are used for connected edge labeling. In the following, we show the experimental results for the following four different objects: plane, cylinder, sphere and polyhedron.

Fig. 9 shows the epipolar-corrected image pair of a planar object. Fig. 10 shows one of the optimal path on the 2D search plane at scanline 125. The obtained range data is shown in Fig. 11.

Fig. 12 shows the epipolar-corrected images of a cylinder. Fig. 13 shows the optimal path on the 2D
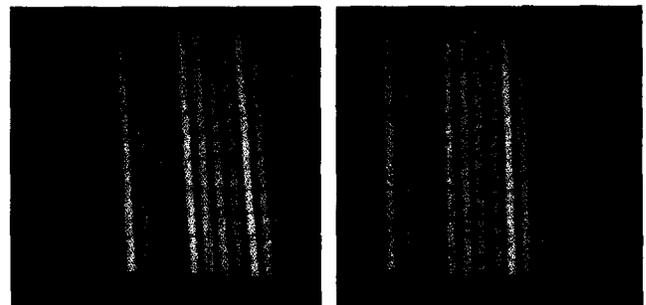

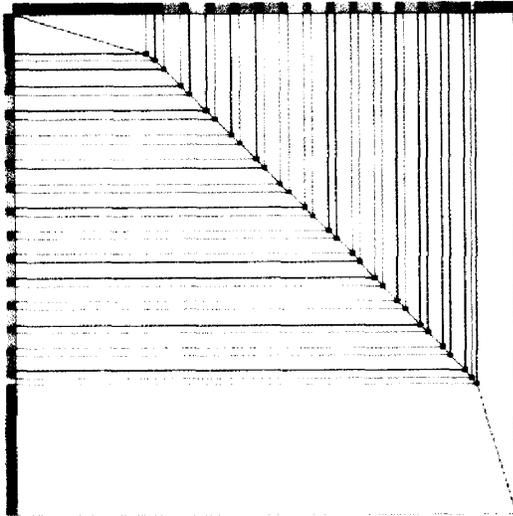Fig. 9. Epipolar corrected image pair of a plane (512 by 512).

Fig. 10. The optimal matching path of the planar object at scanline 125 of Fig. 9.

in Section 3.1.3, this occlusion has successfully been detected as shown in Fig. 13. In Fig. 13, it can be seen that the horizontal primitive path occurs on the left boundary of the cylinder, which represents the occurrence of right stereo occlusion.

Figs. 15 and 16 are the epipolar-corrected images and the obtained range data of a semi-sphere, respectively. Figs. 17 and 18 are the epipolar-corrected images and the obtained range data of a polyhedron object, respectively.

To estimate the accuracy of the acquired range data, three smooth surface objects have been used to compute the fitting errors: the planar object (as shown in Fig. 11); the cylinder object (as shown in Fig. 14); and the sphere object (as shown in Fig. 16). The heights of these objects range from 150–300 mm, and the distance between these objects and the stereo cameras is approximately one meter. We fit a plane, a cylinder and a sphere to the acquired range data of the three objects, respectively.
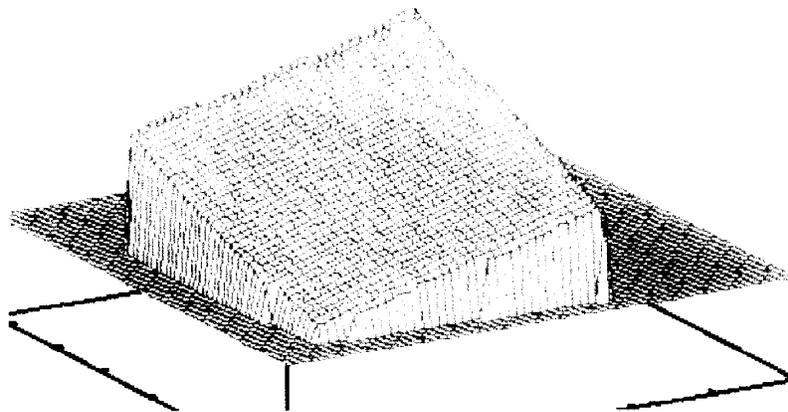

Fig. 11. 3D range data obtained for the planar object.

search plane for one scanline pair. Fig. 14 shows the obtained range data. Since the cylinder is a curved object, its left side can be seen in the left image but cannot be seen in the right image. By using the cost function defined

For the planar object, the *mean-square error* (mse) between the acquired range data and the fitted surface is 0.15 mm. For the cylinder object, the mse is 0.32 mm. For the sphere object, the mse is 0.20mm.
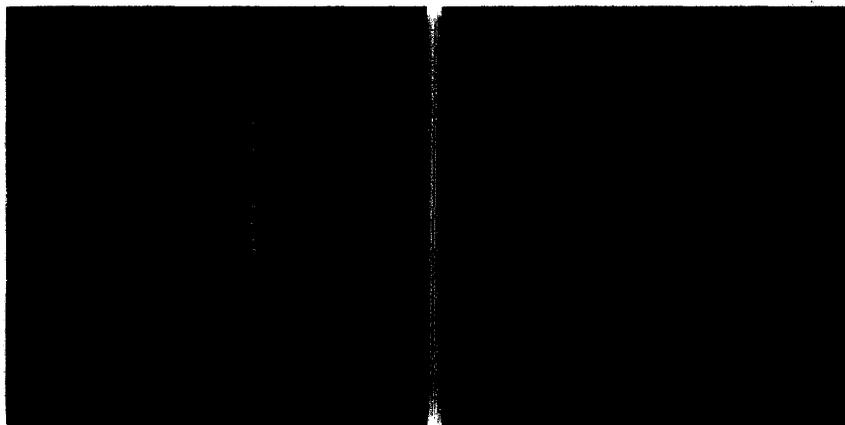

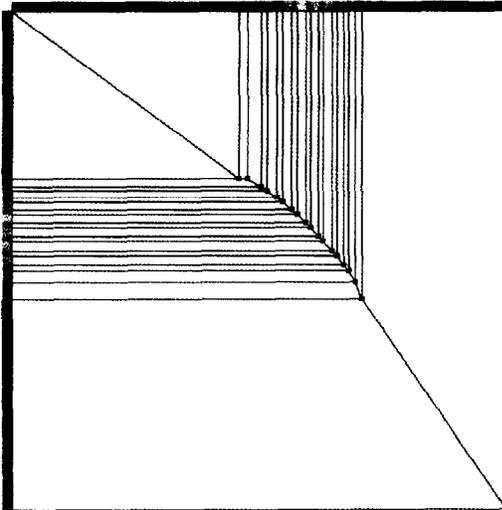Fig. 12. Epipolar corrected image pair of a cylinder.

Fig. 13. Optimal matching path of the cylinder at scanline 250 of Fig. 12.

5. Concluding remarks

In this work, we have successfully developed and implemented a new method for 3D range data acquisition by combining color structured lighting and stereo vision. The range data obtained can be used for many higher level vision tasks, such as 3D object modeling and 3D object recognition. One significant advantage of our approach is that there is no need to solve the problem of finding the correspondence between the color stripes projected by the light source and the color stripes observed in the image. In general, it is quite difficult to solve the above matching problem of finding lighting-to-image correspondence because surface albedos are usually unknown *a priori*. Instead of solving the difficult lighting-to-image correspondence problem, we solve an easier image-to-image stereo correspondence problem – which is not only easier than the lighting-to-image correspondence problem, but is also easier than the traditional stereo correspondence because a good color pattern has been projected onto the object. By projecting ample texture information
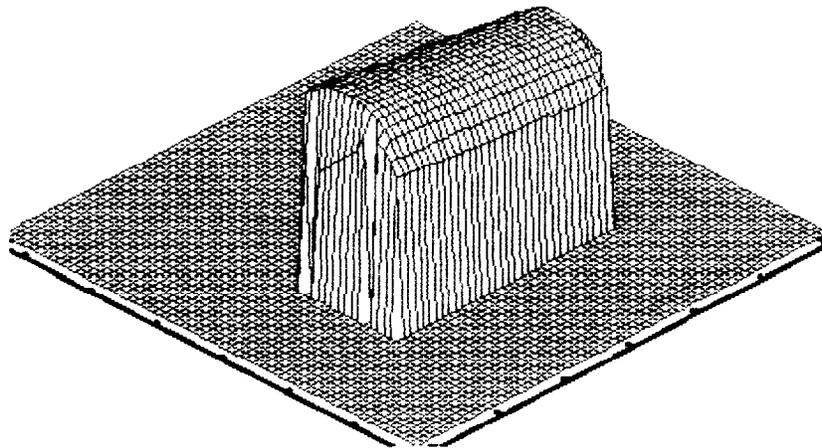

Fig. 14. 3D range data obtained for a cylinder.

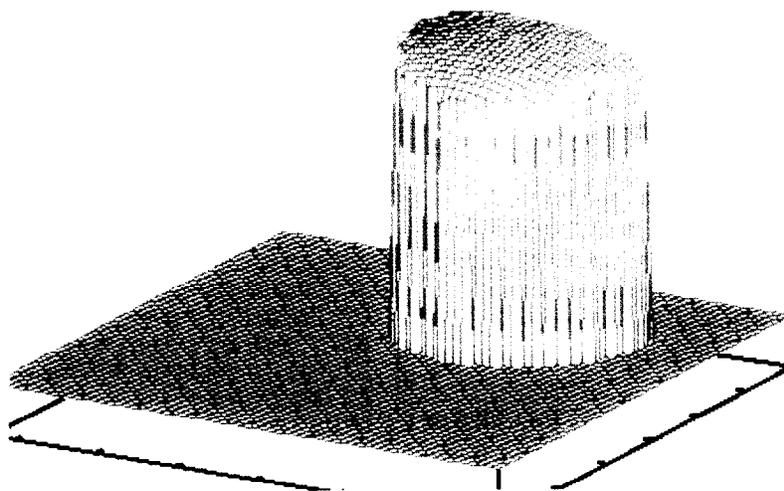Fig. 15. Epipolar corrected image pair of a semi-sphere.

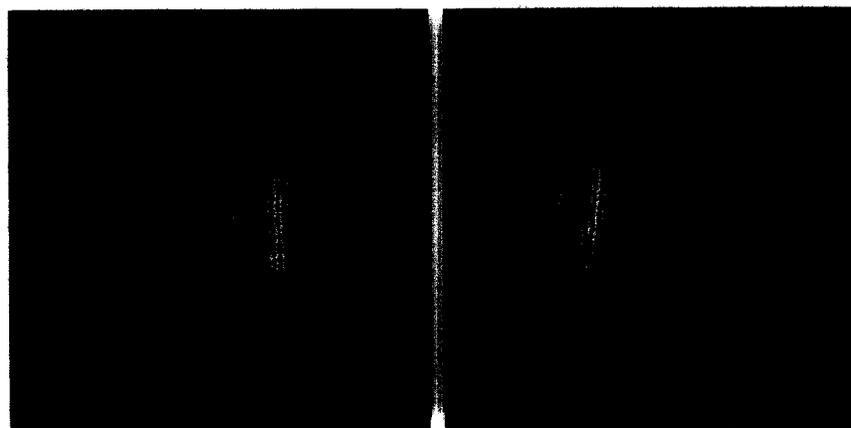Fig. 16. 3D range data obtained for a semi-sphere.

Fig. 17. Epipolar corrected image pair of a polyhedron.

onto the object surfaces, and using a global stereo matching method based on intra-scanline search and inter-scanline consistency, our approach can provide 3D range data in a more stable and accurate manner. In addition, our method does not require the calibration of the position and orientation for each of the projected light stripes in 3D space. Another advantage of our method is that it requires only one pair of stereo images
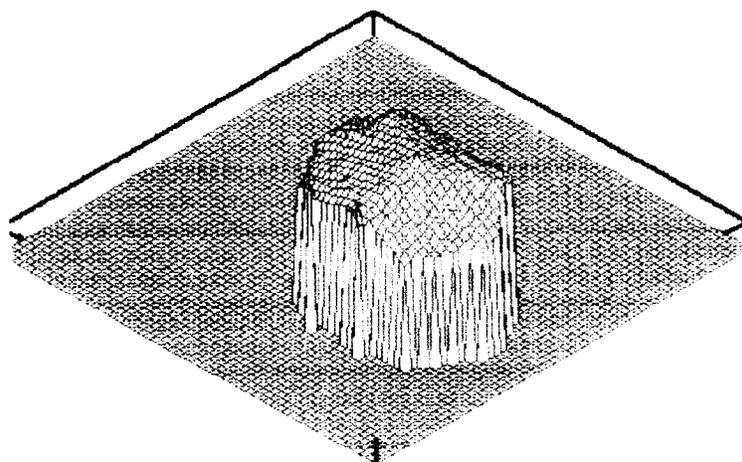

Fig. 18. 3D range data obtained for a polyhedron.

to acquire a dense range map. When using a single line method, 256 images will be needed to acquire one range image having resolution of 256 by 256. When using a binary coded method [7], $\log_2 256 = 8$ images will be needed to acquire one 256 by 256 range map. In summary, this work has the following contributions:

1. With the proposed color-lighting/stereo method, it is not necessary to solve the difficult matching problem of finding the correct color stripe correspondence between the light source and the observed image.
2. We have proposed a systematic procedure for generating good color stripe patterns for our color-lighting/ stereo method.
3. For dealing with stereo occlusion, we have formulated a new way to compute the cost function for the intra-scanline search. Our experiments have shown that stereo occlusion can be successfully detected.
4. We have also modified the cost function used in the intra-scanline search to make use of chromatic information.
5. To utilize inter-scanline consistency but with less computational complexity, we have proposed a winner-take-all scheme to further improve the matching result of the intra-scanline search.

Acknowledgement

References

[1] K.L. Boyer and A.C. Kak, Color-encoded structured light for rapid active ranging. IEEE Trans. Pattern Analysis and Machine Intelligence, (1987) 14–28.

[2] C.S. Chen, Y.P. Hung and J.L. Wu, A 3D range data acquisition system combining laser lighting and stereo vision. J. Chinese Institute of Electrical Engineering (May 1996) 157–168.

[3] R.A. Jarvis, A perspective on range finding techniques for computer vision. IEEE Trans. Pattern Analysis and Machine Intelligence, 5 (1983) 122–139.

[4] W.S. Lin and S.W. Shih, A real-time optical range finder for 3D surface scanning. Proc. First International Conference on Automation Technology, Taipei, Taiwan, 1990, 225–232.

[5] M. Maruyama and S. Abe, Range sensing by projecting multiple slits with random cuts. IEEE Trans. Pattern Analysis and Machine Intelligence, 15 (1993) 647–651.

[6] Y.F. Wang and D.I. Chang, Three-dimensional shape construction and recognition by fusing intensity and structured lighting. Pattern recognition, 25 (1992) 1411–1425.

[7] H.S. Yang, K.L. Boyer and A.C. Kak, Range data extraction and interpretation by structured light. Proc. 1st IEEE Conference on Artificial Intelligence Applications, Denver, CO, 1984, 199–205.

[8] C.H. Hsieh, C.J. Tsai, Y.P. Hung and S.C. Hsu, Use of chromatic information in region-based stereo. Proc. IPPR Conference on Computer Vision, Graphics, and Image Processing, Nantou, Taiwan, 1993, 236–243.

[9] J.R. Jodan and A.C. Bovik, Using chromatic information in dense stereo correspondence. Pattern Recognition (1992) 367–383.

[10] R.C. Gonzales and R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1992.

[11] Y. Ohta and T. Kanade, Stereo by Intra- and Inter-scanline search using dynamic programming. IEEE Trans. Pattern Analysis and Machine Intelligence, 7 (1985) 139–154.

[12] O. Faugeras, Three-dimensional Computer Vision, A Geometric Viewpoint, MIT Press, Cambridge, MA, 1993.

[13] S.W. Shih, Y.P. Hung and W.S. Lin, Accurate linear technique for camera calibration considering lens distortion by solving an Eigenvalue problem. Optical Engineering, 32 (1993) 138–149.

[14] N. Otsu, A threshold selection method from gray-level histograms. IEEE Trans. Systems, Man and Cybernetics, 9 (1979) 62–66.

[15] J. Canny, A computational approach to edge detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 8 (1986) 679–698.

[16] R.M. Haralick and L.G. Shapiro, Computer and Robot Vision, Vol. 1, Addison-Wesley, Reading, MA, 1992.