

Reconstruction of Outer Surfaces by Bi-directional Rays

Hisayoshi Zaima¹

Tsuyoshi Yamamoto²

ABSTRACT

We propose a method for reconstructing the shape and color of objects from ordinary video frames. On the assumption that the objects are located in voxel space, we reconstruct the objects as voxel models by identifying individual voxel properties. Since this method allows a high degree of freedom when taking pictures, this method offers a wider range of application than conventional methods. Since our method does not involve extracting and tracking feature points, the algorithm does not suffer from the problems of corresponding feature points and human intervention. In addition, as there is no need to pay attention to correlation between frames, our approach is easy to deal with. Our method assumes that the objects exist in voxel space. This method identifies the voxel properties so as to keep the appearance from frame to frame consistent. The voxel properties are state variables that represent shape and color, and they are calculated by the backward and forward shooting of rays. During the calculation each voxel has several candidate colors with the correct one only determined at the end. The reconstructed voxel models are rendered from new viewpoints to generate new views.

Keywords: computer graphics, 3 dimensional modeling, voxel space, ray tracing, rendering, reconstruction

1. INTRODUCTION

Approaches to 3 dimensional computer graphics are classified into *model based approaches* that identify geometric information, and *image based approaches* that don't. Recently some hybrid methods have also been proposed. [Debev96b] Model based approaches can be further classified into *creation approaches*, where the user creates a model, and *reproduction approaches* where the computer tries to reconstruct a model of real objects. We concern ourselves with reproduction approaches. Model based reproduction can be further classified into *active approaches*, such as using range finders, and *passive approaches*, such as SFM (Structure From Motion), voting methods and so on. Our approach reconstructs objects' shape and color from ordinary video frames, so it can be regarded as a *passive approach*. For use in application, methods that provide models of objects taken from ordinary video frames in

general environments are desired. Especially, because of the development of network technology and equipment for receiving pictures, the demand is growing in various fields. Reconstruction from feature points is useful but so is the ability to generate outer surfaces of constant data size.

In our approach, the reconstructed models are expressed as voxel-expressions. Since conversion from voxels to polygons is possible using the marching cubes method, our present objective is to generate high quality voxel data.[Loren87e]

2. FUNDAMENTAL PARADIGM

2.1 Voxel Property

Our method assumes that objects exist in voxel space and reconstructs the possibility of existence and color

¹ Doctoral Program in Engineering, Hokkaido University
e-mail:zaima@cc.hokudai.ac.jp

² Computing Center, Hokkaido University
e-mail:yamamoto@cc.hokudai.ac.jp

stat	candidate
on the surface	{R _j ,G _j ,B _j j > 0}
off the surface	{ -, -, - }

Table. 1. Voxel properties

for each voxel. The voxel properties are given by the state variables shown in Table 1. Voxels that may possibly express an objects' surface, are given *on the surface* as state variables and have more than one candidate color for each. If they lose that possibility during the reconstruction processes, the state variable is changed into *off the surface* and the candidate colors are removed. This state changing is irreversible. voxels which maintain the state *on the surface* after executing all processes should express the final surfaces of the objects.

2.2 Bi-directional Ray Shooting

Fig.1 shows the basic idea. Since this method assumes that the objects are from a natural scene, different camera angles should lead to different backgrounds for each frame. In other word, the voxels on the surface, such as V1, should have the same appearance from different viewpoints as shown in Fig. 1. Meanwhile the voxels off the surface, such as V2, are not ensured to have an identical appearance from different viewpoints. So by making use of a number of frames, taken from different positions and verifying the appearance for all voxels from all frames, we can determine if each state variable is *on the surface* or not. This verification is carried out through a backward and forward process.

2.3 The Flow of Reconstruction

We show the whole flow for the calculation in Fig. 2. The backward and forward processes are executed sequentially. These processes consist of loop sub-processes depending on voxel or pixel. Extracting candidate colors for each voxel is executed as a backward process, and verifying the colors in order to select only one is executed as a forward process. Following that, forward rays are shot again in order to fix the outer surfaces and colors.

3. ADVANTAGES OF THIS METHOD

Our approach is characterized as follows:

- (1) Feature points on the surface of objects are not required
Some methods dealing with the feature points tend to

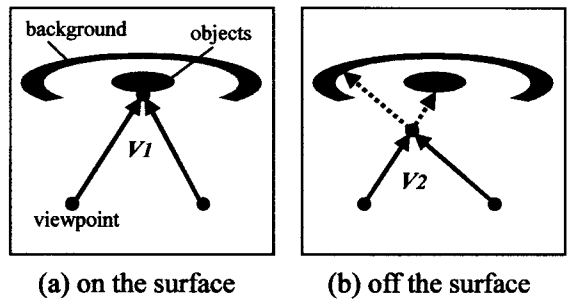


Fig. 1. Basic methodology

Main

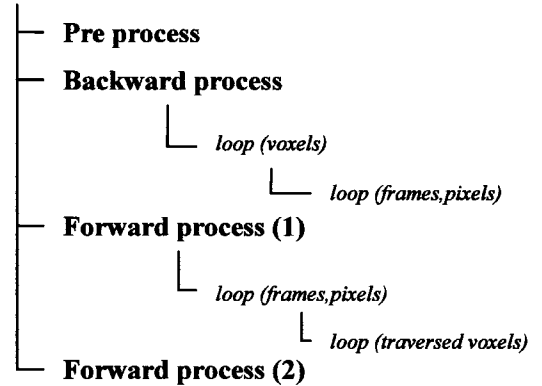


Fig. 2. Main flow of reconstruction

have problems regarding the method and the accuracy when extracting and tracking them. In some case, it may be hard to obtain a sufficient number of points on the surface, for instance with objects of a single color range (i.e. grayscale) or a curved surface with a single color. If the number of feature points is small, interpolation tends to be difficult. On the other hand, if there are a large number of points, incorrect correspondences often occur or human intervention is required, so it follows that the method becomes difficult to conduct. Although our method needs a number of control points on the background in order to perform camera calibration, it does not require any feature points on the objects. From this, even objects that are considered to be difficult to reconstruct, can be reconstructed.

(2) Outer Surfaces of whole objects

If we use feature points, it follows that the reconstructed result depends on the distributions of those points. This means that the state of the results is affected by the number and density of the feature points. Furthermore, in regions where the feature points are dense, incorrect correspondences may be easily formed as described above. Interpolation errors are easily formed in sparse regions. In our method, the amount of final information depends on the voxel space size and it is suitable for calculating outer

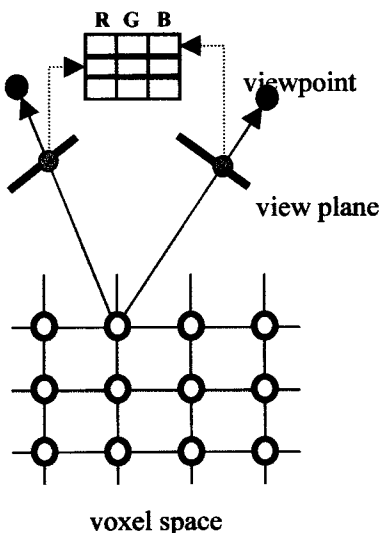


Fig. 3. The Backward Rays

surfaces with uniform density.

(3) Feasibility with a small number of frames

With some methods, the frames must be similar enough to be able to track the feature points, so that the camera angles are necessarily restricted. Furthermore, a large number of frames are required when reconstructing whole objects.[Debev96b] Our approach doesn't impose any restrictions on camera angle. Not to mention the fact that though the accuracy of the results are affected by the number of frames, we can obtain rough results from a small number of frames. Furthermore, since our method does not suffer from the order of the frames, addition and deletion of frames can be done easily.

There are many proposed methods for reconstructing 3 dimensional geometry from 2 dimensional pictures.[Muras95f],[Balla82a] For examples motion stereo, SFM or the voting method. As described above, the most fundamental difference between our method and SFM is the use of the frames and feature points. Since there is no need to track feature points, the voting method is also promising. However extracting, at least, is indispensable. Furthermore, there are some problems for example voting for non-existent voxels is considered to be a problem that causes artifacts. [Kawat94d],[Takah95g] Meanwhile, our approach does not suffer from this problem.

4. BI-DIRECTIONAL RAYS

4.1 The Backward Process

4.1.1 Shooting Rays

Backward rays are shot from each voxel to each

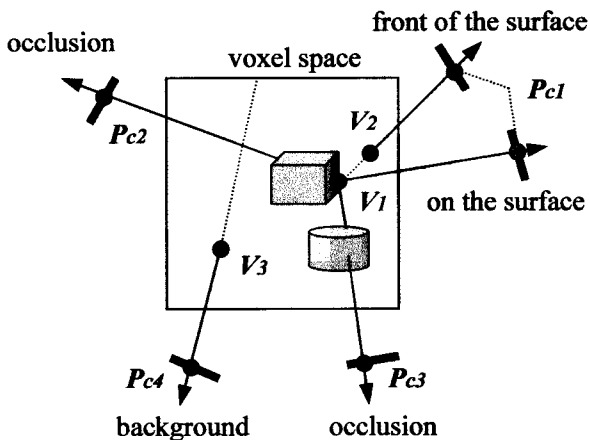


Fig. 4. Variations of candidate colors

viewpoint as follows.

$$\frac{X_{vp} - X}{X_{vp} - X_{voxel}} = \dots = t \quad (1)$$

If the ray intersects with the view plane, it takes the pixel color at the intersection point. This pixel color might be the true color of the voxel. It can also be regarded as a candidate color. By shooting the backward rays from one voxel to all viewpoints, we get an array of candidate colors. (see Fig. 3) Since it is impossible to identify which rays are shot from the visible area, all rays must be dealt with using the same conditions. Fig 4 shows some of the factors that affect the color.

Even though the voxel {V1} is located on the surface, the list of colors generated by backward rays includes some incorrect colors due to occlusion, such as {C2, C3}. The voxels {V2, V3} located off the surface also include various colors in their color list, for instance the original color {C1} and the background color {C4}. Therefore, even though the voxel is located on the surface of the object, it may have incorrect colors caused by rear projections or occlusions. So we must be careful since the most frequently occurring colors in the list are not always the true voxel colors. In other words, it is just a possibility that the true color is in the list {C1, C3...Cn} for {V1}.

4.1.2 Extraction of Candidate Colors

This process calculates the list and narrows down the candidate colors to under six colors. A convex surface can be seen from 180 degrees. Since the inner product of two rays may be negative, zero or positive, our method classifies backward rays depending on their directional vectors. We define six basis vectors {(1,0,0),(-1,0,0)...(0,0,-1)} and group all the rays by

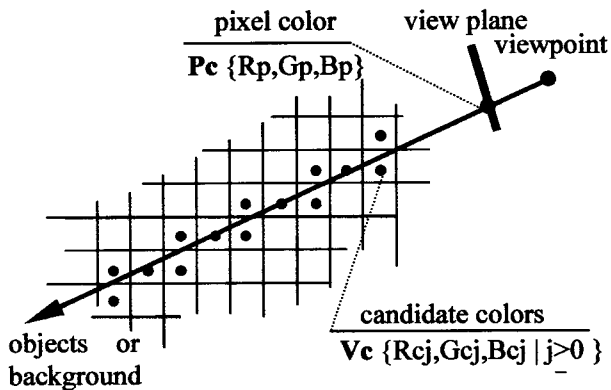


Fig. 5. The Forward Rays

the coefficient of the inner product of the directional vector and basis vector. The most frequently occurring color is regarded as the candidate color for its group. Except when there is a very small visible area due to occlusion, the most frequently occurring color from a visible area has a very high probability of being the surface color. In some cases, the same color is selected from different groups while in other cases no color is selected from a certain group since the required conditions were not met. When the voxel has more than one (and less than six) candidate colors, the state variable is set to be *on the surface*, otherwise, it is set to be *off the surface*.

4.2 The Forward Process

4.2.1 The First Process

Forward rays are shot from each viewpoint to voxel space through each pixel of the frames and expressed as follows:

$$\frac{X - X_{vp}}{X_{pixel} - X_{vp}} = \dots = t \quad (2)$$

As can be seen in Fig. 5, the ray obtains the color $\{P_c\}$ by passing through a pixel, and a list of traversed voxels is created using 3DDDA (3D Digital Different Analyzer). [Fujim86c] Each traversed voxel may have some candidate colors $\{V_{c_j} \mid j \geq 0\}$ as a result of the aforementioned process. While traversing the list of voxels, the pixel color $\{P_c\}$ and the voxel candidate colors $\{V_{c_j}\}$ are compared at each step. In this color comparison process, inconsistency makes the candidate colors invalid. If the voxel lose all candidate colors, the state variable is changed into *off the surface*. This comparison process is started with the closest voxel to the viewpoint and done sequentially until a voxel is found that has the same color as the pixel color. If it is a voxel located on the surface, the pixel color from

visible areas should meet color conditions so that the candidate colors will not be lost. If the pixel color is from an invisible area, the ray stops at the occluding object before reaching voxels located behind it. Therefore correct surface colors are never lost as a result of rays from invisible areas. After executing for all the frames, the 3D shape of the object is decided.

4.2.2 The Second Process

After shooting forward rays once, the shape is decided but there will still be several candidate colors. And the voxels inside the objects have not as yet been dealt with. Next, we again shoot forward rays for all frames in order to vote for pixel colors for voxels that are in the *on the surface* state. Following the voxel list from 3DDDA, each ray traverses the voxel space and the first voxel that maintains an *on the surface* state is elected. After voting once, the ray stops traversing. The color that has the largest number of votes in each voxel is regarded as the final color. Voxels which are not reached by the rays have no votes and are considered to be inside objects, even through they are in an *on the surface* state at that time. We can also reduce data storage requirements for rendering by ignoring inferior voxels. By shooting rays twice we have calculated the outer surfaces and colors.

4.2.3 Color Comparison

It is very important to compare the calculated colors from different viewpoints. We use a threshold value in order to do the color comparison. This value is defined as a permissible range of color intensity for the three primary colors. The basic routine is as follows:

$$V_{c_j} = \max_{\text{appear}} \{P_{c_k} \mid (\bar{e}_j, r\bar{a}y_k) \geq 0\} \quad (3)$$

where $\bar{e}_j = \{1, 0, 0\}, \dots, \{0, 0, -1\}$
 k : the number of frames

This process is executed in both a backward and forward process. When the threshold value is large, more candidates are extracted in the backward process, and it also leads to more voxels staying in the *on the surface* state.

5 EXPERIMENTS

5.1 Reconstruction from Frames of Rendering Models

Reconstructed results from animation are shown in Fig.6(a) and (b). The objects are stationary and only the camera is moved. The input data has 320×240 pixels for each frame and 50 frames. The voxel space




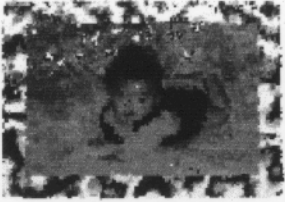
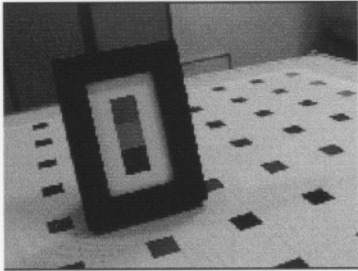

	sample of frames	generated view
(a) Animation with a painting		
(b) Animation with a picture		
(c) Hand-held 3CCD video camera		

Fig. 6. The results of experiments

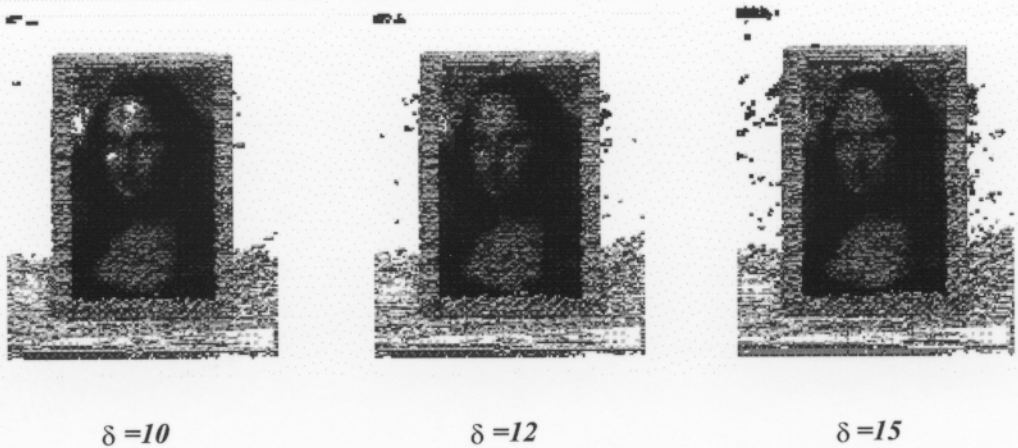


Fig. 7. Comparison of the results by the threshold value

is defined as 128^3 . We show examples of new views generated from the reconstructed models. Fig. 6(a) shows an example using a painting.

Fig. 6(b) shows the same example as Fig. 6(a) but with a photograph. In order to evaluate the effectiveness of our method in cases where there are no camera calibration errors, these examples used rendering of 3D models to obtain the 2D sequence.

Fig. 7 shows how the results vary as the threshold value changes using the example of Fig. 6(a). As can be seen from these results, setting too small a value leads to holes in the surface, while, conversely, artifacts become large when the value is too high.

5.2 Reconstruction from Video Frames

The Reconstructed results from video frames are shown in Fig. 6(c). We used a Sony DCR-VX1000 and an SGI Galileo for conversion from NTSC. The frames were taken freehand in an ordinary room.

They have 640×480 pixels and we selected 15 frames as key frames. Unlike animation, camera calibration is needed for each video frame. The main algorithm for reconstruction is the same as before, and the calibration sub-process is only a pre-process. In this example, graph paper was laid under the objects so as to obtain control points and Tsai's algorithm was used for calibration. [Tsai86j], [Tsai87k]

6 CONSIDERATION

We shall next consider the results from various angles.

6.1 Factors affecting the Accuracy of Results

(1) The accuracy of camera calibration

Even after camera calibration there will still be some errors. Large errors can cause unsuitable extraction of candidate colors in the backward process, as well as unsuitable changing of state variable by using unsuitable colors in the forward process. To avoid this problem, 8 pixel colors surrounding the current pixel at the forward ray are given as permissible colors.

(2) Uniform background

A non-uniform background is a prerequisite for this method. If there were portions of uniform background for all the frames, the colors of these regions should easily lead to artifacts. This is because it is hard for the processes to find contradictions even though the whole frames are used. Fig. 6(c) shows that the color of the graph paper for camera calibration became obvious as artifacts for that reason. Since that paper appears as a common background for all the frames, the artifacts of the paper's color have appeared.

In this case, the problem could be resolved by an improvement in the camera calibration.

(3) The density of texture and voxel space

Since this method fixes some property values for discrete voxels from the real world, objects that have denser textures than the density of voxel space tend to cause artifacts. We can see examples of this unsatisfactory result in Fig. 6(b). This case left something to be desired regarding the expression of the texture of the human face and the background in the picture. This is because the ratio of the changing of the texture in this instance was larger than that of the painting. In this case, since the reconstruction for shape is not affected, a certain improvement is expected by generating interpolated polygons. If the accuracy of the calculated shape is affected by this, other measures are necessary. An easy approach to resolve this problem is to make the threshold value larger, but this also causes other artifacts as described earlier. Another easy approach is to make the density of voxel space smaller, though it may cause an increase in calculation cost or data size. Other more efficient measures might be to use adaptive space division such as octree, or make the voxel properties fuzzy.

(4) The number of frames and their camera angles

Increasing the number of frames improves the accuracy of the results. Especially, the frames from proper camera angles should prove to be more effective. For instance, if objects have a flat surface, the frames taken from the side are important as well as those from the front.

(5) Different exposures

Depending on the location of light sources and the brightness of background, the exposure will be different for each frame. Therefore the color of the same surface point can vary from frame to frame. It means that some color manipulations are required. Since NTSC signal is accurate and stable with respect to luminosity, we fetched luminous signals and observed the transition of intensity. Fig. 8 shows the relationship between exposure and regenerated intensity. Fig. 8(a) shows the regenerated intensity of color pallet from NTSC with different exposures under the same light environment. It turns out that the manipulation of intensity has to be done so as to be around 1.6 dB per changing of 0.5 inside the practical exposure range. Fig 8(b) shows the application of this manipulation. It is an example of how the color of an identical point on the object regenerates through 10 frames taken of a natural scene. The regenerated colors from original frames are denoted with a solid line. In this case the threshold value is required to be more than 57, obtained as a maximum gap in order to pass the color comparison. However this value is not practical. We present the manipulated intensity with increasing the

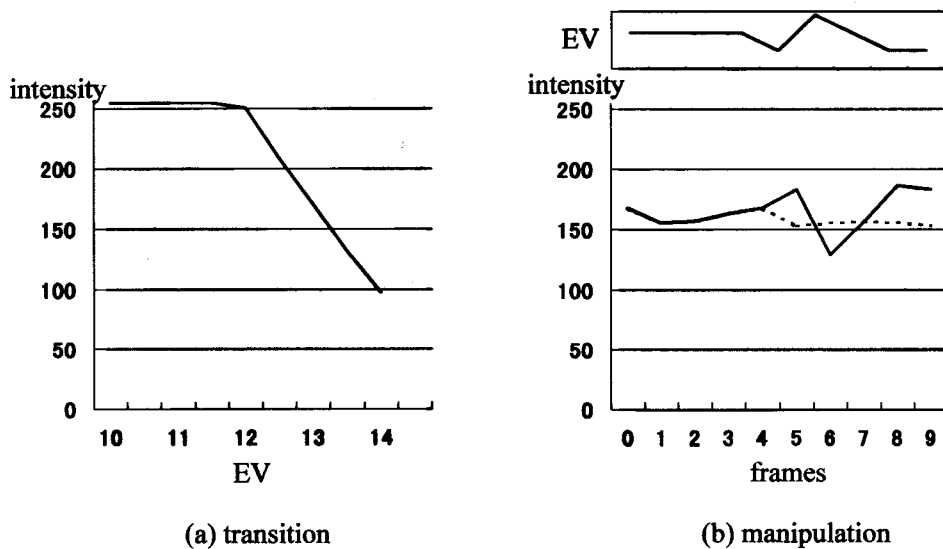


Fig. 8. The relationship between exposure and regenerated color
 (a) shows the transition of intensity with exposure(EV). (b) shows the examples of 10 frames. The bold line means actual measurement, the broken line means manipulated values with each exposure indicated in the upper graph.

value of exposure (EV) as a broken line. In manipulated frames, it turns out that we can set the threshold value to around 15.

6.2 The Cost of Calculation

The calculation takes about 15 minutes using an SGI Onyx (R4400, 256MB) for Fig. 6(c). The total cost depends on the number of voxels if the conditions of input are constant. The cost of the backward process is approximately proportional to the number of voxels. The cost of the forward process is proportional related to the number of traversed voxels calculated by 3DDDA. Since traversing is often stopped by color comparison, the cost is not always proportional to the number of voxels. The cost of the forward process increases by about 1.81 to 2.28 times when the numbers of voxels increase 2^3 times.

7. CONCLUSIONS

We have presented a method of reconstructing object models as voxel models using a sequence of color frames and shown some reconstructed results. With some problems such as artifacts, we have presented a novel approach and instances where humans could recognize the objects, and also confirmed the possibility of this approach.

With the advance of the technology of the Web and of the digital camera, it will become increasingly useful to be able to generate outer surfaces of objects using unrestricted frames.

However, some problems remain in this approach. In order to cope with various environments, further research is necessary with respect to the problem of exposure. With reference to expanding the application territory, objects with complicated surface properties such as high-light affection, semi-transparency or concave surfaces can not be well reconstructed now. We would like to observe and evaluate some natural scenes, and investigate optimizations. In addition, we also hope to investigate an approach without camera calibration. Both our approach and other approaches that deal with feature points, have their own advantages and disadvantages depending on the state of the objects and the end objective. Therefore we do not present a perfect method but rather suggest an algorithm that in some situations can reconstruct outer surfaces of objects through the shooting of bi-directional rays.

References

- [Balla82a] D.H.Ballard and C.M.Brown : Computer Vision, Prentice-Hall, 1982
- [Debev96b] P.E.Debevec, C.Taylor, and J.Malik : Modeling and Rendering Architecture from Photographs : A hybrid geometry and image-based approach, SIGGRAPH96, 1996
- [Fujim86c] A.Fujimoto, T.Tanaka and K.Iwata : ARTS: Accelerated Ray-Tracing System, IEEE Computer Graphics and Applications, No6(4), pp.16-26, 1986
- [Kawat94d] S.Kawato : Extraction of 3D Structure from Images of Multiple Viewpoints by Two-Step

Voting, Journal of IEICE, D-II, VOL.J77-D-II, No.12, pp.2334-2341,1994

[Loren87e] Lorenson W.E and Cline.H.E : Marching cubes : a high resolution 3D surface construction algorithm, Computer Graphics, No.21(4), pp.163-169, 1987

[Murase95f] H.Murase and S.Nayar : Visual Learning and Reconstruction of 3-D Objects from Appearance, International Journal of Computer Vision, VOL14, pp.5-24, 1995

[Seutens92g] P.Seutens, and A.J.Hanson : Computational Strategies for Object Recognition, ACM Computing Surveys, VOL.24, No.1, pp.5-61, 1992

[Takahashi95h] A.Takahashi, I.Ishii, H.Makino and M.Nakashizuka : 3D Structure from Multiple Images and an Estimation Method of Error Distribution, Journal of IEICE, D-II, VOL.J78-D-II, No.11, pp.1618-1626,1995

[Tomasi92i] C.Tomasi and T.Kanade : Shape and Motion Image Streams under Orthography : a Factorization Method, International Journal of Computer Vision, VOL 9.2, pp.137-154, 1992

[Tsai86j] R.Y.Tsai : An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision, Proc. Of CVPR'86, pp.323-344, 1986

[Tsai87k] R.Y.Tsai : A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the Shelf TV Cameras and Lenses, IEEE Journal of Robotics and Automation, VOL.RA-3, No.4, August, 1987