MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

Structure from Motion using Coarse to Fine 3D Voting

Weiming YAO Tsutomu HORIKOSHI
Takayuki YASUNO Satoshi SUZUKI

NTT Human Interface Laboratories
Nippon Telegraph and Telephone Corporation
1-2356, Take, Yokosuka-shi, Kanagawa 238-03, Japan
e-mail: yao@nttcvg.NTT.jp

Abstract

The proposed algorithm divides the scene space into voxels and votes for the voxels using the 3D sight lines of the camera. Thus voxels that correspond to feature points on 3D objects have high voting scores. The algorithm repeats the voting step while changing the voxel size from coarse to fine until it achieves a given scene space resolution. In this process, only the voxels having high votes are divided into smaller voxels and revoted. The algorithm is tested with real images. Three dimensional objects are successfully reconstructed from complex scenes.

1 Introduction

Many new services could be created if three dimensional objects could be accurately reconstructed from image sequences. Previous algorithms using image sequences consist of two steps[1, 3, 6].

The first step is to form the "Spatio-temporal image"(images stacked along the time axis). The camera is assumed to move with linear motion such that the paths of feature points appear on the epipolar plane as straight lines. The second step is to track these lines and then decide the depth of feature points from their slopes.

Existing algorithms have three problems.

1. Camera movement must be linear to constrain the feature point paths on the epipolar plane. This restricts the degree of freedom for autonomous robots and vehicles.

2. Since the paths of feature points are interrupted by occlusions, the extraction of paths is very difficult if there are many occlusions in the scene.

3. The "Spatio-temporal image" must be taken at small time intervals to keep the paths of feature points continuous and also make their extraction easy. Unfortunately, the volume of image data is large and the computation cost is very high.

To overcome problems 1 and 2, Hamano et al. proposed a 3D voting algorithm for structure from motion[5]. This algorithm divides the scene space into voxels and votes for the voxels using the 3D sight lines of the camera which are defined as the lines passing through both the optical center of camera and feature points on the image frames. This voting process adds a weighted value to the voxels containing sight lines. Thus voxels that correspond to feature points on 3D objects will have high voting scores because the sight lines often intersect at these voxels.

This paper extends Hamano's algorithm to overcome the problem 3 described above. The proposed algorithm repeats the voting step while changing the voxel size from coarse to fine until it achieves a given scene space resolution. The size of pixels on the images and the time interval of frames are also changed to match the voxel size. In this process, only the voxels having high votes are divided into smaller voxels and revoted. This algorithm has four advantages:

- It efficiently develops multi-resolution three dimensional descriptions of objects.
- The coarse to fine voting process reduces the computation costs.
- There are no restrictions on the degrees of freedom if the path of the camera is known.
- The influence of occlusions is minimized because it is unnecessary to track the paths of feature points.

The algorithm was tested with real images captured at 30 frames/second by a camera moving with non-linear motion. It successfully reconstructed three dimensional objects from complex scenes and its performance was also evaluated quantitatively.

2 Principle

2.1 Projection

The relationship between the optical center of the camera, feature points on the image plane and voxels in the scene space is shown in Fig.1(a).

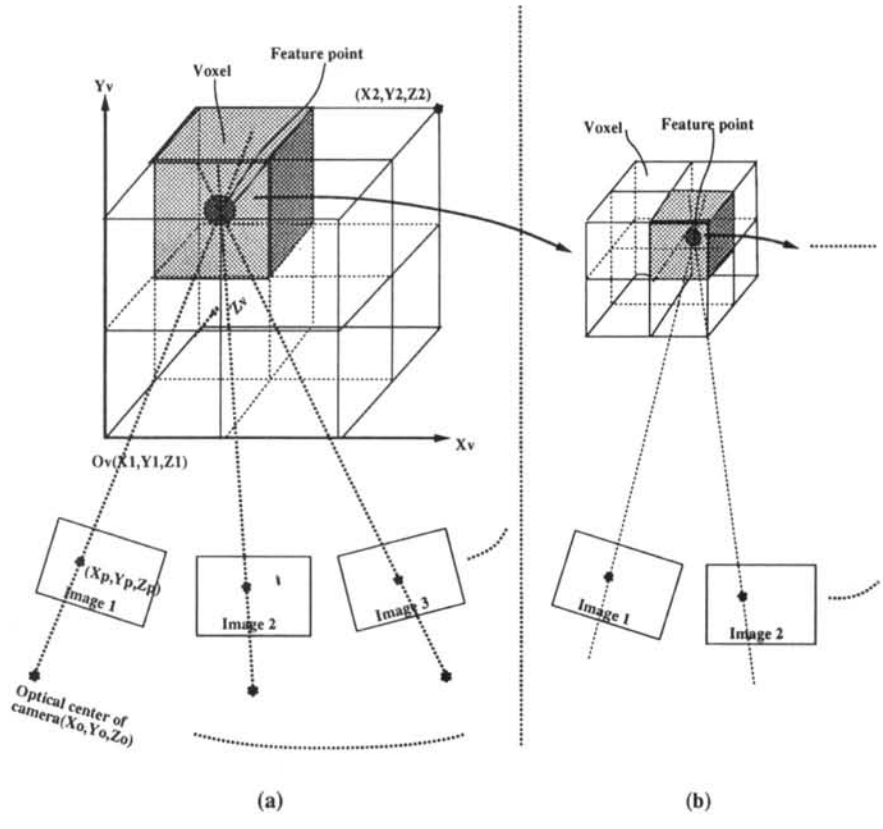


Figure 1: Projection of sight line of camera

At each camera location, sight lines passing through both the optical center of the camera and feature points on the image plane are projected into the scene space. The sight lines (x, y, z) can be decided by:

$$\frac{x - x_o}{x_p - x_o} = \frac{y - y_o}{y_p - y_o} = \frac{z - z_o}{z_p - z_o} \tag{1}$$

Here (x_p, y_p, z_p) and (x_o, y_o, z_o) are the world coordinates of the pixel and the optical center of the camera.

The scene space coordinate (x_v, y_v, z_v) is determined by giving the world coordinates (x_1, y_1, z_1) and (x_2, y_2, z_2) of the origin and the farthest point from the origin of the scene space coordinate system.

$$\begin{cases} x_v = \dfrac{(z - z_o)}{z_p - z_o}(x_p - x_o) + x_o - x_1 \\ y_v = \dfrac{(z - z_o)}{z_p - z_o}(y_p - y_o) + y_o - y_1 \\ z_1 < z_v < z_2 \end{cases} \tag{2}$$

Since the scene space is divided into voxels, the scene space coordinate (x_v, y_v, z_v) determines the voxels through the sight lines pass.

2.2 Voting Process

First, the values of voxels are set to zero. Next, the sight lines are projected into the scene space. In this projection, the intensity of the voxel through which a sight line passes is incremented by the intensity $i(r, c)$ of the feature point on the image plane through which the sight line also passes.

After the first voting round, voxels having high values are detected. Such voxels are divided into eight subvoxels as shown in Fig.1(b). The voting process is performed again. This process is repeated until the voxel scale satisfies a given space resolution.

The given space resolution is changed from coarse to fine to obtain a multi-scale structures of the objects.

2.3 False Feature Points

The voting process is very simple, however it produces false feature points because sight lines cross at feature points as well as at non-feature points. This problem has been solved by a weighted voting function of Hamano's method[5].

3 Image Resolution and Space Resolution

Let $(\Delta u, \Delta v)$ and $(\Delta x, \Delta y)$ denote the horizontal and vertical sizes of pixels and voxels in the camera coordinate system respectively as shown in Fig.2.

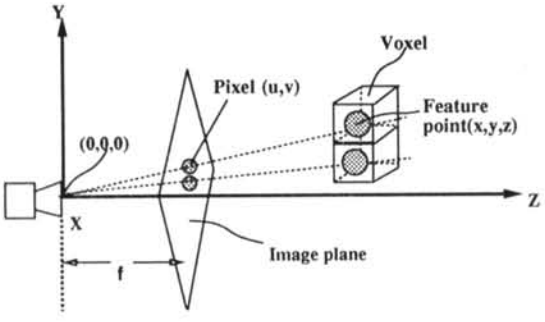


Figure 2: Image resolution and space resolution

The coordinate of voxel (x, y) is calculated from the coordinate of pixel (u, v) by the following equation[2]:

$$(x, y) = (\frac{u \cdot z}{f}, \frac{v \cdot z)}{f}) \quad (3)$$

Here f is the focal length of camera and z is the distance between the optical center of the camera and the voxel in the camera coordinate system.

If Δu and Δv are much smaller than Δx and Δy respectively, the sight lines passing through neighboring two pixels are projected into one voxel. The weighted voting function for deleting false feature points[5] decreases the voting score of this voxel.

If the distance from the optical center of the camera to the scene space is d, this problem can be avoided by setting the size of voxel $(\Delta x, \Delta y)$ to satisfy the following constraint.

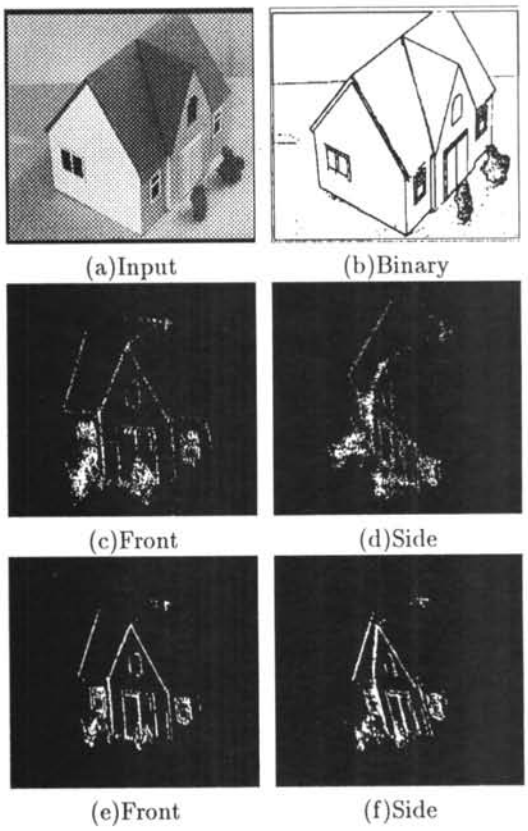
$$\Delta x \leq (\frac{\Delta u \cdot d}{f}, \Delta y \leq \frac{\Delta v \cdot d}{f}) \quad (4)$$

4 Experiment

A sequence of 90 image frames was taken by a non-linearly moving camera at 30 frames/second video rate. The image size is 360 × 252 pixels. First, we detect feature points on the image sequence with a Canny filter[4], and represent them as a binary image sequence. Next, they are transformed into a pyramid image sequence by repeating under operation:

$$p_n(\frac{x+1}{2}, \frac{y+1}{2}) = \bigvee_{i=-2}^{2} \bigvee_{j=-2}^{2} p_{n-1}(x+i, y+j) \quad (5)$$

where p_n is pixel value at level n of the pyramid and $x = 0, 1, 2, \cdots, w-1$; $y = 0, 1, 2, \cdots, h-1$. '$\bigvee$' denotes the logical sum. The coarsest image is 90 × 90 pixels. From constraint (4), a scene space of size 600 × 600 × 600 mm is at first divided into 60 × 60 × 60 voxels. The half of frames(images) are deleted to make the sparse image sequence while each increase in the level of the pyramid. Our method produces 3D representations with 240 × 240 × 240 voxels and 480 × 480 × 480 voxels as shown in Fig.3.



(a)Input (b)Binary

(c)Front (d)Side

(e)Front (f)Side

Figure 3: Experimental results

5 Computation Cost

We compared the computation cost of our method with that of Hamano's. The scene space is divided into $r \times r \times r$ voxels at first. Each voting round divides one voxel into eight subvoxels. The highest number of voted voxels is given by equation (6):

$$\begin{aligned} V_{worst} &= \sqrt{3}\,(rP_0 + 2rP_0 + \cdots + 2^n rP_0) \\ &= (2 - \frac{1}{2^n})\,V_{Hamano} \end{aligned} \quad (6)$$

where P_0 is the number of feature points in the input image sequence. n is the level of the pyramid. $V_{Hamano} = \sqrt{3}\,2^n rP_0$ is the number of voted voxels according to Hamano's method.

Equation(6) appears to show that our computation cost is higher than Hamano's because we assume that all voxels are revoted. This assumption is not true because the voxels without feature points are deleted in the coarse to fine subdivision. We assume that the voxel deletion ratio is $\frac{1}{k}$ at each voting round. Furthermore, the number of feature points is decreased by $\frac{1}{8}$ with the fine to coarse subdivision due to the construction of the pyramid and the sparse image sequence. Therefore, the numbers of voted voxels is given by equation (7):

$$\begin{aligned} V_{proposed} &= \sqrt{3}\,(\frac{rP_0}{2^{3n}} + \frac{2rP_0}{2^{3(n-1)}k} + \cdots + \frac{2^n rP_0}{k^n}) \\ &= \frac{1 - R^{n+1}}{2^{4n}(1-R)}\,V_{Hamano} \end{aligned} \quad (7)$$

where $R = \frac{2^4}{k}$. The numbers of voted voxels with r and k are shown in Fig.4. Our computation cost is lower than that of Hamano's if the space resolution is finer than $2^8 \times 2^8 \times 2^8$ voxels. Furthermore, the increase in our computation cost is quite small.

6 Conclusion

We have proposed a structure from motion scheme that uses a coarse to fine 3D voting algorithm. This algorithm not only reduces the computation cost but also produce a coarse to fine representations of complex scenes. These representations are very useful for 3D object recognition because the data quantity of fine 3D structure is excessive and subject to corruption by noise. This algorithm still has two problems: it is sensitive to position error in the optical center of the camera caused by camera movement, and the image quantization error caused when forming digital images and pyramid images. We will address these problems and their solutions in another paper.

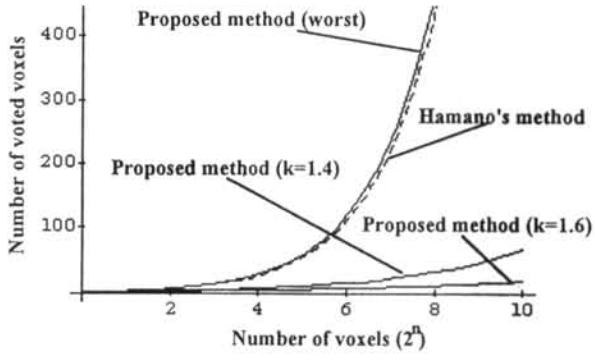


Figure 4: Computation cost

Acknowledgement

We would like to thank Drs. Takaya Endo, Kazunari Nakane and Teruo Hamano for their encouragement and advices.

References

[1] H. H Baker and R. C. Bolles. "Epipolar-plane image analysis on the spatiotemporal surface". *International Journal of Computer Vision*, 1(3):33–49, 1989.

[2] D. H. Ballard and C. M. Brown. *"Computer Vision"*. Prentice Hall, 1982.

[3] R. C. Bolles, H. H. Baker, and D. H. Marimont. "Epipolar-plane image analysis : an approach to determining structure from motion". *International Journal of Computer Vision*, 1(1):7–55, 1987.

[4] J. F. Canny. "A Computational approach to edge detection". *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[5] T. Hamano, T. Yasuno, and K. Ishii. "Direct estimation of structure from Non-linear motion by voting algorithm without tracking and matching". In *Proc. of IAPR International Conference on Pattern Recognition*, volume 1, pages 505–508, Hague, August 1992.

[6] M. Yamamoto. "Determining 3-D structure of scene from image sequences obtained by horizontal and vertical moving camera". *Lect Notes Comput Sci*, 301:458–467, 1988.