A High-speed Human Motion Recovery Based on Back Projection

M. Uchinoumi*, J. K. Tan*, S. Ishikawa*, T. Naito**, M. Yokota**

*Department of Control Engineering, Kyushu Institute of Technology
Sensuicho 1-1, Tobata, Kitakyushu 804-8550, Japan

Email: {etheltan, ishikawa}@cntl.kyutech.ac.jp

**Department of Periodontology and Endodontology, Kyushu Dental College
Manazuru 2-6-1, Kokurakita, Kitakyushu 803-8580, Japan

Abstract

This paper describes a novel technique for high-speed human motion recovery. For a 3-D human motion recovery, stereoscopic vision is a normal technique as well as the employment of the magnetic field. But they cannot escape from the difficulty of marker tracking. In this paper, a novel motion recovery technique is proposed based on back projection of silhouette images. This technique has some advantages over others that it does not employ markers and that it has a simple architecture. In the performed experiment, the proposed motion recovery technique is implemented on a system containing a LAN, a host computer and four pairs of a camera and a computer and it achieves high-speed human motion recovery.

Keywords: Motion recovery, human motion, back projection, silhouette, real-time.

1. Introduction

Three-dimensional human motion recovery is an attractive field of study and recently it has gained much attention among various communities related to digital image processing. It is, for example, employed for producing 3-D characters in video games. Evaluating 3-D motions in sports or in rehabilitation is as well an important application field.

An optical motion recovery is a technique that captures images by cameras and recovers 3-D motion sequences of a human by a computer. Existent major motion recovery techniques are divided into two categories; one with marker tracking [1,2] and the other without markers [3]. In the former, 3-D positions of the markers attached on a human body recover by geometric computation. But automatic marker tracking remains a bottleneck of the technique. On the other hand, motion recovery without markers has an advantage over the former in its high-speed nature in processing. But it still suffers from heavy computational load, resulting in expensive products. These drawbacks prevent the existent motion recovery techniques from being popular among various users who need motion recovery for their familiar subjects such as evaluation of the effect of exercises in rehabilitation.

In this paper, a novel technique is presented for 3-D human motion recovery based on back projection of silhouette images. It contains some accelerated computational techniques that contribute to high-speed motion recovery. They include (1) simple architecture by the employment of a server-client system for computation of back projected images, (2) use of a look up table to realize quick search for the voxels within the object concerned, and (3) use of difference images for volume recovery. Experiments are performed employing the proposed technique. The implemented system contains a host computer as a server and four camera-computer units as clients mutually connected via a LAN. Some results are shown on the recovery. The achievement of this study and further issues to be challenged are finally discussed.

2. Shape recovery by back projection

The idea of back projection is simple. Suppose one takes an image of an object by a camera. Subtraction of a background image from the acquired image yields a mask image of the object after some steps of image processing. The mask image is equivalent to a silhouette of the object from the camera orientation. Then one is able to imagine a cone defined by connecting the center of the camera lens and every point on the contour of the silhouette on the image plane, i.e., an image hull. Clearly the object exists within the cone. If one takes many images of the object from different orientations, each image defines a cone and the original object is obtained from the intersection of all the cones. This is the idea of back projection. See **Fig.1** for intuitive explanation of back projection. It is obvious that concave portions of the object concerned are lost in this recovery technique, since they cannot be observed on the silhouette from any orientations.

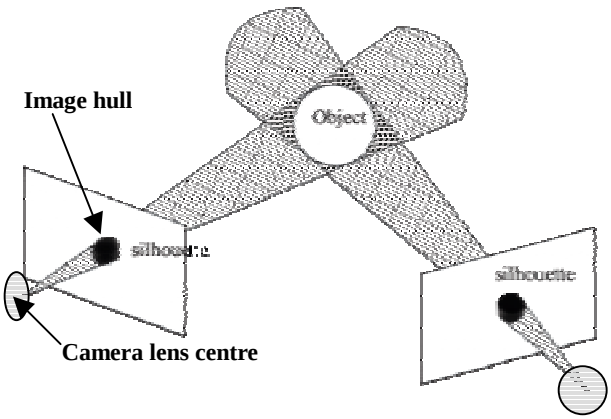

Fig. 1. Explanation of back projection.

At sample time t (t=1,2,…,T), a silhouette image $I_i(t)$ (i=1,2,…,n) of a human in motion is made from the image frame $\hat{I}_i(t)$ by subtracting the background image I_i^{B}, i.e.,

$$I_i(t) = \hat{I}_i(t) - I_i^{\mathrm{B}} \tag{1}$$

Image $I_i(t)$ contains a figure part and a ground. It then receives binarization. The resultant image is again denoted by $I_i(t)$ in which a human region denoted by $R_i(t)$ remains as a figure with the value 1. The ground is given the value 0. The region $R_i(t)$ is then projected back into the 3-D space to yield a volumetric region (or a cone) $V_i(t)$. These cones $V_i(t)$ (i=1,2, …,n) are collected and intersected to obtain a common region by

$$V(t) = \bigcap_{i=1,\ldots,n} V_i(t) \tag{2}$$

The idea of the intersection is explained in **Fig.2** as a 2-D case, in which (a-c) three camera-client pairs produce cones and (d) their common part is extracted.

This procedure is repeated for t=1,2,…,T to obtain the set $\{V(t) \mid t = 1,2,...,T\}$, which gives the recovered 3-D motion sequence.

In order to use this technique, all the observing cameras must be calibrated in advance, since the back projection is performed employing direct linear transformation [4].

3. Strategies for high-speed computation

In order to realize high-speed shape recovery, some ideas are employed in the computation of back projection:

(1) Use of a server-client system to distribute computational load to client computers;

(2) Use of a look up table (LUT) to realize quick search for those voxels in the 3-D space whose projection onto a camera image plane is a single pixel within the silhouette in the image plane;

(3) Use of difference images for successive volume recovery, *i.e.*, the 3-D area at sample time t is produced by the back projection of those pixels in the region that received displacement between the image frame at t-1 and the image frame at t.

3.1 Distributing computational load by parallel architecture

In the back projection technique, a larger number of image taking cameras can realize higher precision of recovered motion. To speed up the computation, a server-client system is employed, when implementing the technique. Cones $V_i(t)$ (i=1,2,…,n) are produced at client computers i (i=1,2,…,n), respectively, and they are transferred through a LAN to a server, where the intersection by Eq.(2) is performed to yield $V(t)$.

One of the advantages of a server-client system is that it is a distributed computation system. If one needs to get more precise shape of the object concerned, one only has to connect more number of camera and client pairs to the system. This doesn't increase the overall computational load much, as the computation is done in parallel. Too many clients, however, cause the increase in data transfer time in the LAN, which may give negative influence on high-speed computation.

3.2 Listing correspondence between pixels and voxels

In the back projection method, the camera system employed must be done calibration in advance. Suppose that a voxel $\boldsymbol{X}_k \equiv (X_k, Y_k, Z_k)$ in the *XYZ* world coordinate system be projected onto the pixel $\boldsymbol{x}_{ij} \equiv (x_{ij}, y_{ij})$ of the ith camera's image plane. Then, as their relation, we have $x_{ij} \equiv x_{ij}(X_k, Y_k, Z_k)$

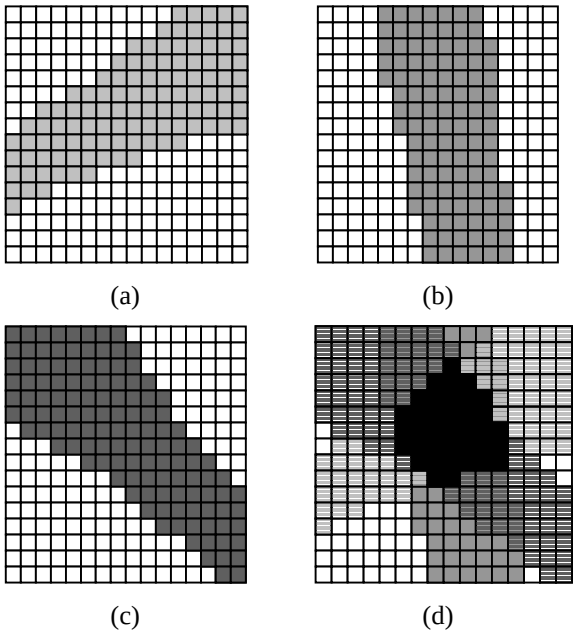

Fig. 2. The idea of back projection and intersection.

and $y_{ij} \equiv y_{ij}(X_k, Y_k, Z_k)$ by direct linear transformation method [4].

Employing these relations, every voxel in the *XYZ*-coordinate system is projected onto the corresponding single pixel on the image plane. It is obvious that, given a pixel $\boldsymbol{x}_{ij}$ on an image plane, the voxels on the line L_{ij} passing through the point $\boldsymbol{x}_{ij}$ and the lens center of the *i*th camera are all projected onto the pixel $\boldsymbol{x}_{ij}$. So this is normally many–to-one correspondence.

Let us denote the set of voxels on the above line L_{ij} by

$$S_{ij} = \{\boldsymbol{X}_k \equiv (X_k, Y_k, Z_k) \mid k = 1,2,..., K_{ij}\}. \quad (3)$$

Then the pairs $\{\boldsymbol{x}_{ij}, S_{ij}\}$ (*i*=1,2,…,*n*; *j*=1,2,…,*m*) are stored into a table called a LUT. Here *m* is the number of the pixels in the image plane of camera *i* (*i*=1,2,…,*n*). The elements of the set S_{ij} are calculated in advance, given the *XYZ* world coordinate system.

Once a silhouette image $I_i(t)$ is given, those pixels within the figure region $R_i(t)$ are found by scan on the image plane. Since the voxels corresponding to the pixels in $R_i(t)$ are known from the LUT, these voxels are marked 1 in the 3-D space yielding the volumetric region $V_i(t)$. The regions $V_i(t)$ (*i*=1,2,…,*n*) are all sent to the server to receive the procedure given by Eq.(2).

3.3 Employment of difference images

If the human motion concerned is not acted very fast, the difference of volumetric regions $V_i(t)$ and $V_i(t-1)$, and hence the difference of silhouette images $I_i(t)$ and $I_i(t-1)$, is not very large. Then creation of $V_i(t)$ can be done by creating only the different part between the silhouette images $I_i(t)$ and $I_i(t-1)$. If the motion is fast, this strategy may not be very effective and $V_i(t)$ had better be made directly. Selection of the strategies depends on the magnitude of the difference.

Let us denote $V_i(t)$ by $V_i(\boldsymbol{X}, t)$ and $R_i(t)$ by $R_i(\boldsymbol{x}, t)$ in order to emphasize the voxels and pixels in them, respectively. Let us also denote a difference image by $I_i^{\mathrm{d}}(t) \equiv I_i^{\mathrm{d}}(\boldsymbol{x}, t)$, i.e.,

$$I_i^{\mathrm{d}}(\boldsymbol{x}, t) = I_i(\boldsymbol{x}, t) - I_i(\boldsymbol{x}, t-1). \quad (4)$$

We further define the following two sets of pixels;

$$S_i^{\mathrm{d}}(t) = \{\boldsymbol{x} \mid I_i^{\mathrm{d}}(\boldsymbol{x}, t) = \pm 1\} \quad (5)$$

$$S_i(t) = \{\boldsymbol{x} \mid I_i(\boldsymbol{x}, t) = 1\} \quad (6)$$

Then the following three cases are applied to create successive cones. Here the number of the elements in the set *A* is denoted by n(*A*).

***Case* 0:** *t*=1:

For all $\boldsymbol{x}_{ij}$ such that $\boldsymbol{x}_{ij}$ is an element of $R_i(\boldsymbol{x}, t)$, and for all $\boldsymbol{X}$ such that $\boldsymbol{X}$ is an element of S_{ij}, let $V_i(\boldsymbol{X}, t) = 1$.

***Case* 1:** $n(S_i^{\mathrm{d}}(t)) < n(S_i(t))$ for *t*=2,3,…,*T*:

Let $V_i(\boldsymbol{X}, t) \equiv V_i(\boldsymbol{X}, t-1)$.

For all $\boldsymbol{x}_{ij}$ such that $I_i^d(\boldsymbol{x}_{ij}, t) = -1$, and for all $\boldsymbol{X}$ such that $\boldsymbol{X}$ is an element of S_{ij}, let $V_i(\boldsymbol{X}, t) = 0$.

For all $\boldsymbol{x}_{ij}$ such that $I_i^d(\boldsymbol{x}_{ij}, t) = 1$, and for all $\boldsymbol{X}$ such that $\boldsymbol{X}$ is an element of S_{ij}, let $V_i(\boldsymbol{X}, t) = 1$.

***Case* 2:** $n(S_i^{\mathrm{d}}(t)) \geq n(S_i(t))$ for *t*=2,3,…,*T*:

For all $\boldsymbol{x}_{ij}$ such that $\boldsymbol{x}_{ij}$ is an element of $R_i(\boldsymbol{x}, t)$, and for all $\boldsymbol{X}$ such that $\boldsymbol{X}$ is an element of S_{ij}, let $V_i(\boldsymbol{X}, t) = 1$.

Case 0 is the initial situation, i.e., *t*=1, and the region $V_i(1)$ is created directly from a silhouette image $I_i(1)$. There are two strategies for *t*=2,3,…,*T*. In case 1, the difference is smaller than the silhouette itself. The voxels are deleted from $V_i(t-1)$ that correspond to the pixels whose gray value is –1, and the voxels are added to $V_i(t-1)$ that correspond to the pixels whose gray value is 1, yielding $V_i(t)$. In case 2, the difference is larger than the silhouette itself and $V_i(t)$ is created directly from a silhouette image $I_i(t)$.

In this way, the region $V_i(t)$ is created. Once $V_i(t)$ (*i*=1,2,…,*n*) are collected at the server, the entire volumetric region $V(t)$ is computed and the entire motion is given as the set $\{V(t) : t = 1,2,...,T\}$.

4. Experimental Results

We have implemented the proposed technique in a system, which consists of four PCs and a single PC (CPU: Pentium, 1.7-3.2GHz) employed as clients and a server, respectively, and a 100Mbps LAN combining them each other. The clients are connected to respective digital cameras settled in a room. The cameras are calibrated in advance employing the DLT method. A 2m×2m×2m space in front of the cameras are digitized into 80×80×80 voxels in this particular experiment. A person acts various motions within the cubic space. The motions recover in a 3-D way by the proposed technique and the recovered motion is shown in the PC display.

In this paper, the result of a batting motion is shown. The experimental environment is shown in **Fig.3** and the recovered 3-D motion is given in **Fig.4**.

The process time for the recovery with each sample time was approximately 91 *msec* in average. It includes the time for the recovery calculation at the clients, the time for data transfer between the clients and the server, and the time for the intersection calculation and displaying the result at the server. This signifies that video images were processed every three frames. In this

Fig. 3. Experimental environment.

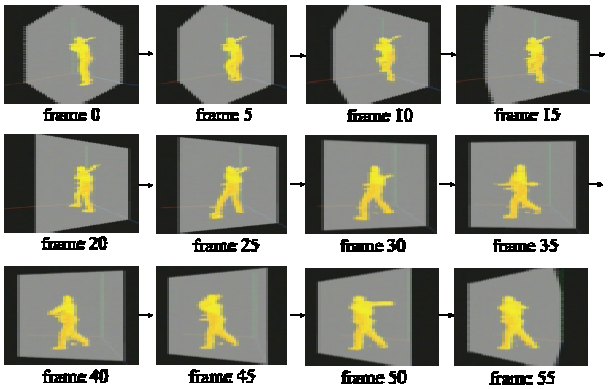

Fig. 4. Recovered batting motion.

way, high-speed human motion recovery has been realized.

5. Conclusions

A technique was proposed to perform high-speed 3-D human motion recovery based on back projection of silhouette images provided from multiple cameras. In order to speed up the computation, (i) distributed computation architecture was employed in order to build a volumetric region (a cone) in a parallel way, (ii) lists of correspondence between voxels in the world coordinate system and pixels in image planes was prepared for high-speed back projection, and (iii) difference images between successive frames were effectively used for less computational load of back projection. Performance of the proposed technique was experimentally shown.

The presented technique has an advantage over those existent techniques based on markers tracking in that it doesn't have to use markers. This makes the technique much simpler than such existent techniques, since markers tracking problem is still an issue of difficulty. Instead, the present technique cannot escape from camera calibration like other shape/motion recovery techniques, except for [2]. Ideally a technique that employs neither markers nor camera calibration for the recovery is desirable for wide spread of a motion recovery technique to various users who don't have much related knowledge. Such a technique still remains unknown.

References

[1] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, 1987.

[2] J. K. Tan, and S. Ishikawa, "Deformable shape recovery by factorization based on a spatiotemporal measurement matrix", *Computer Vision and Image Understanding*, Vol 82, No. 2, pp. 101-109, May, 2001.

[3] K. Tomiyama, et al., "A dynamic 3D object-generating algorithm from multiple viewpoints using the volume intersection and stereo matching methods", *The Journal of the Institute of Image Information and Television Engineers*, Vol 58, No. 6, pp. 797-806, 2004. (in Japanese)

[4] R. Shapiro, "Direct linear transformation for three-dimensional cinematography", Res. Quart., Vol. 49, pp. 197-205, 1978.