Optical Motion Capture System with Pan-Tilt Camera Tracking and Realtime Data Processing

Kazutaka KURIHARA^{*1}, Shin'ichiro HOSHINO^{*1}, Katsu YAMANE^{*1}, and Yoshihiko NAKAMURA^{*1*2} (E-mail: kurihara@ynl.t.u-tokyo.ac.jp)

^{*1}Dept. of Mechano-Informatics, Univ. of Tokyo 7-3-1 Hongo Bunkyo-ku Tokyo, 113-0033 JAPAN

*²CREST, JST

Abstract

This paper presents the realtime processing of optical motion capture with pan-tilt camera tracking. Pantilt camera tracking expands the range of capturing field dynamically. Asymmetrical marker distribution and polyhedra search algorithm realize robust labeling against marker missing. The algorithm is developed for parallel cluster computation and enables realtime data processing. Experimental results demonstrate effectiveness of the system.

Key Words: Optical motion capture, Pan-tilt camera, Realtime labeling, Pallarel computing, Behavior Capture.

1 Introduction

Analysis of human motion has increased its importance and demand as more computationally intensive approaches become available in controlling humanoid robots, creating CG human motions, and planning medical procedures [1][2][3][4][5][6]. To investigate the control algorithms of human behaviors and the information processing behind them, it would be useful to measure multimodal data such as the foot forces, the surface EMG signals, the sight directions, the diameters of the pupils, and the brain waves as well as the geometric motion patterns. We have developed the Behavior Capture Studio, where we can capture such multimodal human behavior data synchronously and in realtime. Realtime applications involve, but are not limited to, the whole body motion cockpit for humanoid robots.

This paper presents the development of the optical motion capture system developed for the Behavior Capture Studio. An advantage of the optical motion capture is the absence of constraints. A subject is requested only to wear a suit with many small reflective markers to be observed by several cameras to reconstruct 3D motion data. The drawbacks of the optical motion capture, on the other hands, are:

- 1. The subject's motion is limited within the common sight of the cameras, and therefore it requires sizable environments.
- 2. The 3D reconstruction of motion data commonly requires human intervention to compensate mislabeling due to occluded markers.

This paper proposes the pan-tilt tracking camera to overcome the first problem. A system of six pan-tilt cameras is developed as a prototype. Also discussed in this paper is the computational issue of 3D reconstruction. The 3D reconstruction of captured data involves identification of corresponding marker points in multiple camera images, which we call "labeling." Automatic reliable labeling without human assistance is a difficult problem to solve in a realtime level and requires a large amount of computation. This paper proposes an automatic labeling algorithm and its parallel computation using PC cluster.

2 Behavior Capture

2.1 The Behavior Capture Studio

We constructed the Behavior Capture Studio shown in Fig.1. It consists of the optical motion capture system, an eye-mark recorder(NAC EMR-8), floor reaction force sensor plates(KISLER), ultra sonic position trackers(Intersense IS-600), a brain wave/EMG



Fig.1: Behavior Capture Studio

recorder(NEC SYNA ACT MT11), and a 3D image projection system. The equipments are networked and synchronized, and allow temporal multimodal measurements of human behaviors.

2.2 Optical Motion Capture

Motion capture system has two elements. One is the equipment that is attached to the subject and transmits the motion information, and the other is the equipment that is set around the subject and receives the information. In the case of optical motion capture systems, passive optical markers correspond to the former, and multiple cameras play a role of the latter. The subject wears a suit with many small retroreflective markers, which reflect a light in the direction that the light came from as seen in Fig.2. A camera unit consists of a $262[\text{fps}](512 \times 512 \text{ pixels})$, 8bit) monochrome CCD camera, LED lights and two direct-drive motors for pan and tilt axes of Fig.3, and six camera units are placed on the ceiling like Fig.4 . Note that the detailed camera position and orientation are determined by calibration as well as the other geometric and optical parameters. The pan-tilt camera tracking system achieves 2.364 times wider capture area than the fixed camera setting of the same cameras.

The computation of the optical motion capture proceeds according to the following flow:

- Capturing markers by multiple cameras.
- Marker extraction on the camera image.
- Reconstruction of marker position.
- Marker Labeling.
- Computing joint angles.



Fig.2: Capture Suit with Markers



Fig.3: Outline of Pan-Tilt Camera

In this paper, we deal with the phases up to marker labeling.

2.3 Computer System

Five kind of computers are installed, and they are networked by MPI(Message Passing Interface) shown in Fig.5.

• Camera image processing PC

Each pan-tilt camera is connected to a PC via CORECO VIPER DIGITAL image processing board. The sequence of image capture, marker extraction, and calculation of the marker vector in the local coordinate is processed on this PC. The iterative calculation of calibration is also done on this PC.

• Motor control PC

A motor control PC controls three pan-tilt cameras. The pan and tilt direct drive motors are PID-controlled by each motor driver.

• Marker reconstruction PC

The marker vector data from Camera image processing PC and the camera parameter data from Motor control PC are gathered and three dimensional marker position is reconstructed.



Fig.5: System of "Tracking Motion Capture"

• Server PC

All above described computers are controlled by this PC via MPI.

• Labeling PC

All 3D-reconstructed markers and estimated missing markers are identified as described in the following chapter.

3 Realtime Labeling

3.1 Polyhedra Search Algorithm

After the three dimensional marker position data are obtained, there is a task to identify each marker.

It is called labeling. Usually, labeling is executed offline, meaning that is executed after the capture. In offline labeling, we can optimize the labeling output to be consistent for each moment [7]. But in the realtime labeling we propose, such optimization is impossible. We should compensate for missing markers with some appropriate estimation, and should make effort to do labeling only with the obtained data of each moment. Because in realtime processing, we cannot obtain future data, cannot always rely on the past output. We assume the marker position data have some errors. If past output is considered as completely true, the algorithm is weak against a single error. So labeling algorithm should be designed not to inherit past possibility of errors. And to obtain fine motion data, the algorithm should be able to determine the position and orientation of each body part.

To meet these demands, we established Polyhedra Search Algorithm and Asymmetrical Marker Distribution. The first step is to set the unique polyhedra on each body part of the subject. At least three points are needed for each rigid body to be determined its position and orientation. Fig.6 shows an example of marker distribution that each body part has at least three markers. In the marker position data, the targeted group is searched as the set of markers which are thought of the vertices of set polyhedron. The procedure is as follows:

- 1. To make a distance table which all the combinations of existing markers are described.
- 2. To search the triangles that the targeted polyhedron consists of.
- 3. To confirm the existence of the other targeted markers based on the triangle.
- 4. To check the whole body consistency using labeling element model.

The search of triangles in the distance table is illustrated in Fig.7. If some triangles are found, the rest of markers are searched by the local coordinate spanned by the triangle. For example, When searching tetrahedron ABCD(Fig.8), at first the triangle ABC is searched, we obtain the local coordinate spanned by $p, q, p \times q$. Then the vector toward D, denoted by r, is calculated by initially known parameters s, t, u as follows:

$$\boldsymbol{r} = s \boldsymbol{p} + t \boldsymbol{q} + u \boldsymbol{p} \times \boldsymbol{q} \tag{1}$$

s, t, u are calculated at the initial moment when all



Fig.6: 48 Markers Placed on 34 DOF Model



Fig.7: Searching Triangle

four positions of ABCD are known:

$$\begin{pmatrix} s \\ t \\ u \end{pmatrix} = \begin{pmatrix} \boldsymbol{p} & \boldsymbol{q} & \boldsymbol{p} \times \boldsymbol{q} \end{pmatrix}^{-1} \boldsymbol{r}$$
(2)

From the fifth marker onward, the same method is available. Using this algorithm, by searching at least three markers of each polyhedron we confirm the existence of the rest of markers and can estimate the position if some markers are missing.

In the body part candidates obtained by the polyhedra search, there are many combinations that the spatial relationship is inappropriate as a human body. To output appropriate labeling result, we should choose the best combination that the relationship between adjacent parts is valid. We now introduce Labeling Element Model illustrated in Fig.9. Each part is connected together (Fig.10). The element consists of



Fig.8: Searching Tetrahedron : Notation



Fig.9: Labeling Element Model

two kinds of markers. One is a set of markers whose distances between each other do not change, like five markers on the head. The other is an adjacent external marker whose distance from the targeted part does not change so much, like a marker on the neck. By checking the distance between adjacent parts, the inappropriate connection such as head-to-foot connection is avoided.

3.2 Asymmetrical Marker Distribution

In this search method, if there are many congruent figures, many candidates are hit in a search and that causes the necessity to select. It takes much time. To avoid it, we suggest asymmetrical marker distribution as following.

- To be careful not to make congruent or mirror symmetrical polyhedra.
- To be careful not to make polyhedra that have ambiguity over the orientation like isosceles triangles.

We experimented to do labeling some captured data of subject's motion whom 20 markers were attached as asymmetrically as possible. If all five tetrahedra are congruent equilateral tetrahedra, the number of candidates of single polyhedron search amounts to 120. The result shows mean number of candidates of single polyhedron search to be 3.3. We confirmed that asymmetrical marker distribution can reduce the number candidates of polyhedra search.



Fig.10: Labeling Element Model : Links



Fig.11: Process Assignment

3.3 Parallel Computing

When considering realtime processing of optical motion capture system, it is important to reduce the labeling computation time. Since we use DALSA CA-D6 camera that can capture in the rate of 262fps, in the ultimate sense we hope to label also in the rate or faster. As an approach toward it, we modified the polyhedra search algorithm to the parallel computing version, and applied it on the cluster computer.

In the polyhedra search algorithm, the body-partsearch phase can be executed independently for each body part, simultaneously. So we spread the computational load by distributing the searches for each machine as process. Fig.11 shows the parallel computation scheme and process assignment. Implementation is done by Message Passing Interface. For each moment, each machine receives marker position data from the server, and searches only one polyhedron that is in charge of, and sends the candidates to the server.

To demonstrate the efficiency of parallelization, we compared computation time between serial and parallel implementation as following.

1. We made 6 congruent polyhedra that each of

Table 1: Serial Computation Time [ms]

Polyhedra	24 Mkrs.	30 Mkrs.	36 Mkrs.	42 Mkrs.
4 Mkrs. \times 6	1.54	2.29	3.08	4.00
5 Mkrs. \times 6	*	7.81	7.85	10.2
6 Mkrs. \times 6	*	*	16.6	21.2
7 Mkrs. \times 6	*	*	*	45.7

Table 2: Parallel Computation Time [ms]

Polyhedra	24 Mkrs.	30 Mkrs.	36 Mkrs.	42 Mkrs.
4 Mkrs. \times 6	0.340	0.460	0.600	0.760
5 Mkrs. \times 6	*	1.41	1.41	1.81
6 Mkrs. \times 6	*	*	2.89	3.69
7 Mkrs. \times 6	*	*	*	7.70

them consists of 4 to 7 markers.

- We made imaginary captured data sequence of 24, 30, 36, 42 markers. All of them includes above described polyhedra.
- 3. We did labeling the data with serial and parallel implementation. All used processors are PentiumIII 1GHz.

Table.1 and Table.2 shows the result. The parallel computation time was 5.93 times faster than the serial one, at a certain case (theoretical maximum is 6 times because we used 6 parallel processors). Note that this is an evaluation experiment with overloaded condition, so the computation time for typical motion capture is much faster than the result.

4 Experiments

We placed 20 markers as asymmetrically as possible on the subject. The captured subject's motion was labeled and an example CG character's motion was generated with a certain inverse kinematics algorithm. Fig.12 shows the captured marker data, labeled marker data(connected with lines), and CG character's motion. Table.3 shows the parallel computation time. 310 [μ s] high speed labeling was achieved. Note that missing markers in Fig.?? are correctly estimated in Fig.?? . By this practical experiment, the effective-ness of the labeling algorithm for realtime processing is confirmed.



Fig.12: Practical Experiment

٢	Table 3: Actual	Computation time	of Label	ing [ms]
	Table Making	Polyhedra Search	Others	Sum
	0.0300	0 160	0.120	0.310

5 Conclusions

The results obtained in this research is summarized by the following four items:

- 1. We constructed the optical motion capture system that is one of the most important elements of our developing "Behavior Capture System."
- 2. The narrow range problem of optical motion capture was solved by pan-tilt camera tracking.
- 3. For the realtime labeling, Polyhedra search algorithm and asymmetrical marker distribution was established.
- 4. A parallel computing scheme for the labeling algorithm was developed, and experimental re-

sults demonstrated the robustness against marker missing and small computation time.

Acknowledgements

This research was supported by CREST program, the Japan Science and Technology Corporation.

References

- Zvi Shiller, Katz Yamane, Yoshihiko Nakamura: "Planning Motion Patterns of Human Figures Using a Multi-layered Grid and the Dynamics Filter," Proceedings of IEEE International Conference on Robotics and Automation, pp.1-8, 2001.
- [2] P.Sardain and G.Bessonnet: "Gait Analysis of a Human Walker wearing Robot Feet as Shoes," Proceedings of IEEE International Conference on Robotics and Automation, pp.2285-2292, 2001.
- [3] Satoshi Morita, Toshiyuki Kondo, Koji Ito: "Estimation of Forearm Movement from EMG Signal and Application to Prosthetic Hand Control," *Proceedings of IEEE International Conference on Robotics and Au*tomation, pp.3692-3697, 2001.

- [4] Chia-Yu E. Wang, James E. Bobrow and David J. Reinkensmeyer: "Swinging From The Hip: Use of Dynamic Motion Optimization in the Design of Robotic Gait Rehabilitation," *Proceedings of IEEE International Conference on Robotics and Automation*, pp.1433-1438, 2001.
- [5] Darrin C. Bentivegna, Christopher G. Atkeson: "Learning From Observation Using Primitives," Proceedings of IEEE International Conference on Robotics and Automation, pp.1988-1993, 2001.
- [6] Andrea Bottino, Aldo Laurentini: "Experimenting with nonintrusive motion capture in a virtual environment," *The Visual Computer*, Vol.14, pp.14-29, Springer-Verlag, 2001
- [7] Yang Song, Luis Goncalves, Enrico Di Bernardo and Pietro Perona: "Monocular Perception of Biological Motion - Detection and Labeling," *Proceedings of IEEE CVPR*, pp.805-812, 1999.
- [8] John J. Craig: Introduction to ROBOTICS, Addison-Wesley, 1989.

Appendix: Pan-Tilt Camera Calibration

The orientation of optical axis, the position of optical center, and the focal length, these are called camera parameters. Since it is hard to measure them directly, the calibration is generally used. Since we use pan-tilt camera, we also have to determine the pan and tilt rotation axes. One simple way is the sequential calibration which pan and tilt rotation axes and camera parameters are determined sequentially. But in this method, the error may be accumulated gradually. So we developed the calibration method which all the parameters can be determined simultaneously.

Each coordinates are set as shown in Fig.13, denoted by O_w, O_0, O_1, O_2, O_c . The relationships between adjacent two coordinates are described in DH parameter[8]. Note that O_0 is introduced because z axes of O_w and O_1 are nearly equal and that causes a_i, d_i to be unstable.

Now denoting transformation matrix from the coordinate i-1 to i by ${}^{i}\boldsymbol{T}_{i-1}(\theta_{i},\alpha_{i},d_{i},a_{i})$, the transformation matrix from the world coordinate O_{w} to the camera coordinate O_{c} is ${}^{c}\boldsymbol{T}_{w}$, denoted by:

$${}^{c}T_{w} = {}^{c}T_{2}{}^{2}T_{1}{}^{1}T_{0}{}^{0}T_{w} \tag{3}$$

Fig.14 shows the relationship between the position in the world coordinate and that in the camera image for a marker. $\boldsymbol{x}'_{\boldsymbol{c}}$ is the marker position on the camera image that the optical distortion is compensated. \boldsymbol{X} is the marker position expressed in O_w .

$$\boldsymbol{X} = (X_1 \ X_2 \ X_3)^T \tag{4}$$

$$\boldsymbol{x_c'} = (x_1 \ x_2)^T \tag{5}$$





Fig.14: Frame of Camera

Denoting the orientation of the O_w in O_c by \mathbf{R} , the origin of O_w in O_c by \mathbf{t} , we can introduce the following equation:

$$w\begin{pmatrix} \boldsymbol{x}_{c}'\\ 1 \end{pmatrix} = \begin{pmatrix} \alpha & 0 & 0\\ 0 & \alpha & 0\\ 0 & 0 & 1 \end{pmatrix} (\boldsymbol{R}|\boldsymbol{t})\begin{pmatrix} \boldsymbol{X}\\ 1 \end{pmatrix}$$
(6)

where w is arbitrary constant, and α is denoted by following equation:

$$\alpha = fk \tag{7}$$

where f is the focus length, and k is the reciprocal of the actual width of a pixel (We assume that all pixels are square.). Now we define:

$$\begin{pmatrix} q_1 & q_2 & q_3 & q_4 \\ q_5 & q_6 & q_7 & q_8 \\ q_9 & q_{10} & q_{11} & q_{12} \end{pmatrix} \equiv \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} (\boldsymbol{R}|\boldsymbol{t})$$
(8)

With this eq.8, we can transform eq.6 into following expression:

$$\boldsymbol{A}_{\boldsymbol{i}}\boldsymbol{q}=0\tag{9}$$

where

$$\boldsymbol{A_{i}} = \begin{pmatrix} X_{1} & X_{2} & X_{3} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_{1} & X_{2} & X_{3} & 1 \\ & -x_{1}X_{1} & -x_{1}X_{2} & -x_{1}X_{3} & -x_{1} \\ & -x_{2}X_{1} & -x_{2}X_{2} & -x_{2}X_{3} & -x_{2} \end{pmatrix}$$

$$(10)$$

$$\boldsymbol{q} = (q_1 \quad q_2 \quad \cdots \quad q_{12})^T \tag{11}$$

We obtain two simultaneous equations of q for a marker. For n markers, by gathering A_i that is obtained for each marker we obtain 2n simultaneous equations:

$$\begin{pmatrix} \mathbf{A_1} \\ \mathbf{A_2} \\ \vdots \\ \mathbf{A_n} \end{pmatrix} \mathbf{q} = 0 \tag{12}$$

To solve these equations, we introduce an evaluation function J as follows:

$$J = ||Aq||^2$$
(13)
$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix}$$
(14)

and minimize J by Least Gradient Method. Note that, as eq.6 shows, since q_i is determined by α and DH parameters, we use not q_i but α and DH parameters as variables of optimization.

For the calibration of pan-tilt cameras, it is necessary to capture the markers from various camera orientations. Since then \boldsymbol{q} changes, we cannot make eq.12. To avoid this problem, we introduce imaginary markers which are placed on the position that cancels the camera displacement. The imaginary marker position $\boldsymbol{X'} = (X'_1 \ X'_2 \ X'_3)^T$ that is obtained by the back rotation around the axis estimated by proceeding optimization is calculated for each joint, and substituted in $\boldsymbol{A_i}$.

The transition of J(15 parameters) is shown in Fig.15. Approximated values are set as initial values for the parameters. The computation time for 120,000 times iteration is 955s on a Pentium III 800MHz machine.

In the case of fixed camera calibration, we have to move markers at the whole field to obtain uniform



Fig.15: Translation of Criterion Value

accuracy(so called Dynamic Calibration). Since pantilt camera calibration enables markers to be fixed, their accurate position are easily obtained and we can achieve uniform accuracy in the whole field.