

社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
PRMU98-248 (1999-03)

多視点カメラを用いた実時間全身モーションキャプチャシステム - 視覚に基づく仮想世界との実時間インタラクション -

米元 聡 松本 明日香 有田 大作 谷口 倫一郎

九州大学大学院システム情報科学研究科

〒 816-8580 福岡県春日市春日公園 6-1

TEL: 092-583-7618

{yonemoto,asuka,arita,rin}@limu.is.kyushu-u.ac.jp

あらまし

我々は実空間の動物の動作を、多視点カメラを用いて実時間で計測し、その情報をもとにリアリティの高い仮想空間を構成することを目指している。ビデオカメラを用いた画像ベースの実空間センシングは、センサ接触型の方法では計測できない情報（形状特性や表面など）を得ることができる利点がある。そこで、我々は画像ベースのセンシングの実用化を目指す。まず実時間性を重視するため、人間の全身につけたカラーマーカの追跡を行なうプロトタイプシステム、つまり多視点カメラを用いた実時間モーションキャプチャシステムの開発を行なった。本システムを高速通信系 Myrinet で接続した PC クラスタ上へ実装し、実時間かつオンラインで動作可能な仮想現実アプリケーションに適用した。

キーワード 実時間多視点動画処理 多視点融合 多関節物体 モーションキャプチャ

A Real-time Full-body Motion Capture System with Multiple Cameras

Satoshi YONEMOTO, Asuka MATSUMOTO, Daisaku ARITA and Rin-ichiro TANIGUCHI

Graduate School of Information Science and Electrical Engineering, Kyushu University

6-1, Kasuga-koen, Kasuga, Fukuoka 816-8580 JAPAN

+81-92-583-7618

{yonemoto,asuka,arita,rin}@limu.is.kyushu-u.ac.jp

Abstract *This paper presents a real-time motion capture system of 3D multi-part objects, whose purpose is to do seamless mapping objects in the real world into virtual environments easily. To achieve this requirement, we have been developing a vision-based motion capture system which reconstructs time-varying motion parameters of 3-D multi-part objects. In this paper, as our first vision-based system, we have implemented a color-marker based motion capture system and have demonstrated our motion capture and reconstruction system works real-time for multiple-part objects on PC-cluster.*

key words real-time processing, multiview fusion, multi-part objects, motion capture

1 はじめに

1.1 研究の目的

人間の動作を実時間で計測し、認識・再生することができれば、3-D CG アニメーションやビデオゲームにおけるキャラクタの動作生成、人間と機械との仮想空間上でのインタラクション、ロボットの遠隔操作など、多くのアプリケーションでの適用が期待できる。そのような目的を達成するため、我々は多視点カメラセンサを用いた実時間モーションキャプチャシステムの開発を行なっている。従来より、人間の動作を実時間でセンシングする技術として、光学式、磁気式などセンサ接触型の方法が用いられてきた。一般に、それらの手法によるものは、システムが大規模になり高価になること、測定対象の動きに制限が生じること、必要以上に身体的な拘束を課してしまうことなどの問題がある。一方、近年になって、非接触型な方法としてビデオカメラで撮影した映像から3次元動作を計測する方法が、コンピュータビジョン手法の応用により試みられるようになってきたが [1]、実時間¹で計測およびオンラインで再生ができないことや、センサの配置などによる物理的な制約以上に、多くのシーンに関する制約が必要であったり、一般に必要とされるようなモーションキャプチャのデータとしては、精度の面で課題があるなど、実用レベルにまで達していないのが実情である。しかし、非接触あるいは接触型の方法で、ビデオカメラによりセンシングする技術は、対象に関して、またその他背景に関して、他の接触型の装置で計測不可能な情報（例えば表面情報や形状など）を同時に得ることができるという可能性を有しているため、我々は実時間でビデオカメラを用いてセンシング・および認識・再生する技術の実用化を進めている。

1.2 関連研究

前述した通り、ビデオカメラで撮影した映像より人間の全身動作を推定し、仮想空間上でインタラクションする方法としては、Pfunder システム [1] が既に提案されている。このシステムは、2次元の Blob 特徴を抽出し、完全非接触な方法で3次元推定を行なう。また、比較的安価なシステム構成でありながら、実時間に近い速さで動作可能である。しかしながら、仮想空間上のキャラクタに全身動作を模擬させるような場合、このような非接触な方法で十分な精度を保ちつつ全身動作を推定するには、形状や動きの精度など、課題が残る。より詳細な形状・動きパラメータを推定する方法としては、オフラインで獲得された画像系列に対し、モデルベースで解析する方法 [3][4] などもある。しかし処理時間が比較的膨大であることや、モデル像と画像のマッチングを行なうに必要な精度である初期情報はオフラインでしか得ることができないという制約のため、実時間での適用には課題が残っている。

¹本研究では実時間はビデオレート (30fps) の時間と定義する。

本報告では、その第一段階として、非接触なアルゴリズムの実現の前に実時間性を重要視した、カラーマーカを用いたプロトタイプシステム、すなわち、多視点カメラセンサ系により実時間で比較的精度の高い全身動作の計測および再生を行なうシステムについて提案する。以下では、カラーマーカを利用した実時間モーションキャプチャシステムの概要について説明し、最後に実験結果について述べ、全身動作の場合でも、仮想空間での再現系まで含めて実時間 (30fps) で動作することを示す。

2 システムの概要

我々は、本プロトタイプシステムを、高速ネットワーク通信機構により接続された複数の分散 PC から成る PC クラスタ上に実装した²。

2.1 システムの流れ

以下にシステムの流れを示す。

1. 初期設定
2. 各視点におけるカラーマーカ領域重心追跡
3. 多関節物体の3次元動きパラメータ推定
4. 仮想空間の実時間レンダリング
5. 2-4. を実時間で繰り返し処理する。

2.2 実時間多視点動画画像処理のための PC クラスタの必要性

実世界において、知能視覚システムが有効に機能するために必要な基本特性として以下が挙げられる [2]。

- 実時間性、オンライン性、ロバスト性、柔軟性

上記の基本特性を満たす分散協調視覚システムを実現するにあたり、すなわち、まず実時間で動作可能にするためには、高性能なハードウェアの整備が必要で、分散 PC 間の同期した時間の概念も必要である。また、システムがオンラインで安定に動作するには、処理遅れなどによるエラー処理をシステムに付加する必要もある。さらに、マルチセンサフュージョンのように多様なセンサ情報融合を実現するためのフレームワークを用いることで、ロバストで柔軟なシステムが実現できる。上記のロバスト性と柔軟性を実現するため、多視点動画画像処理が期待されており、マルチセンサにより獲得された冗長な証拠情報を互いに参照することで、単一なセンサで起こりうるオクルージョンなど様々なエラー処理に対処することが可能となることが知られている。しかし、この場合、実時間性やオンライン性の基本特

²本研究のように多視点動画画像の実時間処理を考えた場合、入力センサから発生するデータのバンド幅を考えると、高性能の分散処理システムが必要不可欠になる。PC クラスタはそのためのものであり、PC クラスタ上のソフトウェアシステムを考えることも我々の重要な研究テーマである (CDV プロジェクト [2] の一環として実施している)。

性の実現が困難になっていることが多い。そこで、実時間多視点動画画像処理を実現するにあたり、高性能な分散システムが必要となり、システム内の各分散コンポーネント間の時間の同期機構の実現や、システムがオンラインで動作するためのエラー処理が最大の問題となる。

我々は、そのような目的のために PC クラスタを構成し、NTP(Network Time Protocol) に基づいた同期機構のメカニズムを開発した。分散システム内の各処理モジュールは、PC 内のプロセスに割り当てられ、同期機構により並列に動作する。PC クラスタの詳細および同期機構の詳細については文献 [6][7] 参照のこと。

2.3 処理モジュールの通信・設計

図 1 は、PC クラスタの各 PC 上に本システムの処理モジュールを並列パイプライン配置したものである。それら各処理モジュールを以下のように設計している。

画像獲得モジュール (ICM)

これらの処理モジュール ICM は、画像 (系列) の取り込みとリサイズ (320 × 240) を行なう。各 $ICM_v (v = 1, \dots, N; N$ は視点数) はそれぞれ、2 次元画像特徴抽出モジュール (2DPM_v) に画像を送信する。

2 次元画像特徴抽出モジュール (2DPM)

これらの処理モジュール 2DPM は、2 次元画像特徴の抽出処理 (マーカ領域重心追跡) を行なう。各 2DPM はそれぞれ ICM より画像を受信し、処理結果 (マーカ重心) を 3 次元復元モジュール (3DPM) へ送信する。また、PC の台数制限がある場合は、ICM と 2DPM をまとめて 1 つの処理モジュールとする。

3 次元復元モジュール (3DPM)

処理モジュール 3DPM は、2DPM_v ($v = 1, \dots, N$) で得られた処理結果を融合し、3 次元情報 (関節位置) の推定 (復元) を行なった後、結果を実時間レンダリングモジュール (RRM) へ送信する。

実時間レンダリングモジュール (RRM)

処理モジュール RRM は、仮想空間の実時間フルスクリーンレンダリング (表示) を行なう。また受信した 3 次元情報より、対象物モデルの位置姿勢計算もこの処理モジュールにおいて行なう。以下、各処理モジュールの通信内容について述べる。

ICM:

-送信メッセージ (to 2DPM_v)

リサイズした入力画像

2DPM:

-送信メッセージ (to 3DPM)

2 次元画像特徴抽出結果 (2 次元重心)
状態 (可観測か不可観測か)
マーカ重心追跡の信頼性

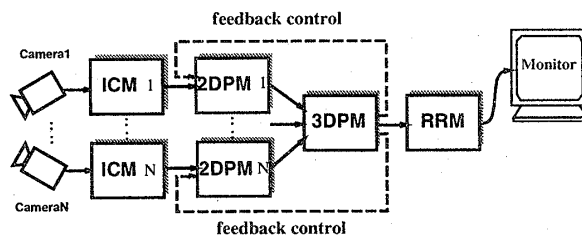


図 1: PC クラスタにおける処理モジュールの配置

-受信メッセージ (from ICM_v)

リサイズした入力画像

-受信メッセージ (from 3DPM)

フィードバック情報
時刻 t におけるマーカ領域重心の予測位置
3DPM からの不可観測状態遷移の通知

3DPM:

-送信メッセージ (to 3DPM)

3 次元推定結果 (関節位置)

-送信メッセージ (to 2DPM_v ($v = 1, \dots, N$))

フィードバック情報

-受信メッセージ (from 2DPM_v ($v = 1, \dots, N$))

2 次元画像特徴抽出結果 (2 次元重心)
状態 (可観測か不可観測か)
マーカ重心追跡の信頼性

RRM:

-受信メッセージ (from 3DPM)

3 次元推定結果 (関節位置)

2.3.1 2-D 画像特徴抽出モジュールの状態遷移

2DPM_v の状態を、可観測 (マーカの 2 次元重心が追跡可能) か (observable), 不可観測か (unobservable) で定義する。各 v 番目のモジュール自身の信頼性評価と 3DPM からのフィードバック情報により、可能な状態遷移およびアクションを以下のように定義する。

(i) $observable(t-1) \Rightarrow observable(t)$:

現在の重心位置を送信する。

(ii) $observable(t-1) \Rightarrow unobservable(t)$:

マーカ領域の画素が全くカウントされない時、あるいは 3DPM からの不可観測強制的通知が届いた時にこの状態に遷移する。このとき、現在の推定位置として予測位置を送信する。

(iii) $unobservable(t-1) \Rightarrow observable(t)$:

再びマーカ画素がカウントされたならばこの状態が起る。マーカ領域重心位置が再度送信される。

(iv) *unobservable(t-1)* \Rightarrow *unobservable(t)*:

マーカ領域重心の代わりに、現在の推定位置としてフィードバックにより得たマーカ重心の予測位置を保持しておく。これは、出現により不可観測から可観測へ移行する際に備えて、探索範囲をとらえておくためであり、オクルージョンによるエラー処理からの回復において効果がある。

3 初期設定

以下では、本システムを動作させるにあたり必要となる初期設定について述べる。

3.1 前提

事前準備：色マーカ(球)を各関節位置に設定する。システムが安定に動作するようマーカの色はなるべく異なるものを用意する。

システムの動作条件：

- 撮影環境は、なるべく照明が一定になるよう、固定光源下の室内の方がよい。また、マーカ色に類似した物体は撮影シーンから予め除いておくほうがよい。
- 初期設定の間は、対象物はほぼ静止している。

多関節物体モデルの幾何：対象物(多関節物体)は木構造で表現する。また、実寸に対応させるため、ワールド座標はキャリブレーション時のワールド座標と同一に設定する。

3.2 多視点カメラのキャリブレーション

本システムは予め、キャリブレーションしておく必要がある。キャリブレーション物体を用い、統一ワールド座標のもとで全視点についてキャリブレーションを行なう。キャリブレーションとしてレンズ歪みを考慮した Tsai の方法 [5] を利用している。

背景物体のマーカ類似色による影響をなるべく取り除くため、本システムでは、マーカ領域抽出時に背景差分処理を行なう。背景画像は、各視点ごとに撮影しておくが、動作計測時と同一の照明環境で撮影しておく必要がある。

4 カラーマーカ重心位置の推定

4.1 マーカの色パラメータ学習

基本的な色パラメータ学習の原理は、[8] で提案された方法による。画素の r, g, b を、以下の明度 i を媒介変数とした 2 次関数で表現する。

$$\begin{aligned} i &= (r + g + b) / 3 \\ \hat{r}(i) &= R_2 i^2 + R_1 i \\ \hat{g}(i) &= G_2 i^2 + G_1 i \\ \hat{b}(i) &= B_2 i^2 + B_1 i. \end{aligned} \quad (1)$$

マーカ j に対応する学習用画像 ($j = 1, \dots, m$) おける全画素 (r_j, g_j, b_j) を用いて、6つのパラメータ、つまり、係数 $R_1, R_2, G_1, G_2, B_1, B_2$ を各色 r, g, b について、以下の式により線形最小二乗推定する (SVD 法)。

$$\begin{bmatrix} r_1 & g_1 & b_1 \\ \vdots & \vdots & \vdots \\ r_m & g_m & b_m \end{bmatrix} = \begin{bmatrix} i_1^2 & i_1 \\ \vdots & \vdots \\ i_m^2 & i_m \end{bmatrix} \begin{bmatrix} R_2 & G_2 & B_2 \\ R_1 & G_1 & B_1 \end{bmatrix} \quad (2)$$

ここで $\begin{bmatrix} r_1 & g_1 & b_1 \\ \vdots & \vdots & \vdots \\ r_m & g_m & b_m \end{bmatrix}$, $\begin{bmatrix} i_1^2 & i_1 \\ \vdots & \vdots \\ i_m^2 & i_m \end{bmatrix}$ は学習用データ。

4.2 色の識別

まず、ある (x, y) 位置の画素の明度 i および、各マーカごとの学習した色パラメータ $R_1, R_2, G_1, G_2, B_1, B_2$ を用いて、モデルとなる色 $\hat{r}(i), \hat{g}(i), \hat{b}(i)$ を学習時と同様に計算する。また、マーカ色であるかの識別基準としては以下の式を用いる。

$$error = (r - \hat{r}(i))^2 + (g - \hat{g}(i))^2 + (b - \hat{b}(i))^2 \quad (3)$$

以下の条件を満たす時、 (x, y) の画素がそのマーカ色であると判定する。

$$\begin{aligned} (i) \quad & low < i < high \\ (ii) \quad & error < error_th, \end{aligned} \quad (4)$$

ただし、 $high, low$ はそれぞれ最高明度のしきい値、最低明度のしきい値である。また、 $error_th$ は色識別のしきい値である。(i) の基準で予め不安定なデータを取り除き、(ii) の基準で色の類似度を判定する。

4.3 カラーマーカ領域重心の追跡

簡単な画像処理戦略では、実際、色領域を自動的に識別するのは容易でない。したがって、処理速度を高速に保ちつつ、より精度よくマーカ色を識別するため、次のような処理の工夫を行なっている。

限定範囲内の重心計算：画像領域全体を探索するのに対し、限定範囲を探索することで探索コストを大幅に削減し、ノイズに敏感な重心計算を比較的安定に行なうことができる。初期設定時には探索範囲を予め定義した最大の大きさに設定し、それから予め定義した最小値まで徐々に範囲を狭めていく。限定範囲のみの探索であるため、非常に高速な対象物の動作の場合など、マーカ領域を見失うことがある(すなわち重心計算に必要な画素が 0 になる場合)。その場合、再びマーカ領域を検出するため、探索範囲を逆に広げることに対処する。

背景差分の利用: 室内環境においても, 背景の物体にはマーカ色に非常によく似た色が存在する. それらは重心の計算に大きな影響を与えるため, 予め取り除く必要がある. そこで, 上記の限定範囲内において背景差分を施す.

特徴マップの導入: 類似色の競合を避けるとともに, 計算コストを抑えるため, 以下の情報を各画素単位で保持する特徴マップを導入する.

- 推定された色. すなわち式 (4) で定義される最もエラー値の低いマーカ色.
- 現在の対応色の類似度 (エラー値).
- 対応色を更新した時のタイムスタンプ. 画像系列のフレーム番号に相当する.

ここで, タイムスタンプは, 保持している情報が最新であるかどうかをチェックするために用いる. これにより, 特徴マップをフレームごとに初期化する必要がなくなり, その計算時間を省くことができる.

以上の工夫を施した時刻 t における重心計算のアルゴリズムは以下になる.

1. 各マーカに対し, 前フレームで推定したマーカの重心を中心とする限定範囲内でのみ画素の色の識別を行なう. 初期フレームの場合は, 重心は予め指定したデフォルト値 (画像中心など) とする. 前述したとおり, 初期フレームでは, 探索範囲を最大探索範囲から次第に小さくしていく. また, 重心計算に必要な画素が 0 になった場合は探索範囲を広げる.
2. 背景差分を各マーカの限定範囲内について行ない, 不必要な画素を取り除く.
3. 画素 (x, y) の r, g, b 値を用いて, 明度 i を計算した後, 既に学習したカラーパラメータに基づいて $\hat{r}(i), \hat{g}(i), \hat{b}(i)$ を計算する. 実際には, 高速化のため, $\hat{r}(i), \hat{g}(i), \hat{b}(i)$ は予め look-up-table に登録しておいて実行時に参照する.
4. 上記の式 (3) を用いて色の類似度を計算し, 識別基準 (4) を用いて対応する色かどうかを判定する. もし対応色であれば, 特徴マップに登録するとともに, 座標 x, y を重心計算のために加算する.
5. 似たマーカ色の場合, 上記の識別基準をクリアする画素はどちらもカウントされてしまう. そこで似た色どうしで二重にカウントするのを回避するため, 特徴マップを参照し, 既に登録された色より類似度が大きい色が検出されれば, 重心計算の値を修正する.
6. 上記の処理を行なった後, 各マーカの重心を計算する.

7. 1.~6. の処理を各フレームで繰り返す行なうことにより, マーカの重心位置が検出される.

自己評価 各視点において推定された結果がどの程度信頼できるものか, まず, 抽出アルゴリズム自身で評価する必要がある. その信頼性の目安として, 対応画素の分散や評価値が考えられる. 本報告では, 推定した重心の精度を上げつつ, その評価を行なうために, 各マーカに対し, 信頼度 R を以下のように定義する.

$$R = \frac{N_f}{N_s} (\leq 1) \quad (5)$$

N_s : 探索範囲内の対応画素カウント数

N_f : 特徴マップ中の推定重心の局所領域 w 内の
カウント数

この信頼度によりデータの凝縮性を評価する. この評価値を, 3-D 復元モジュール 3DPM に送信し, 多視点の視点選択・融合の基準に用いる. これらの機能は視線選択・融合アルゴリズムにおいて実現している.

4.4 3次元マーカ位置の追跡

少なくとも 2 視点からの視線が求められれば, 3次元位置をステレオの原理によって一意に復元できる. しかし, (特に, 静止カメラの) 2 視点では部分的な情報のみしか観測できないため, 2 視線では 3次元位置を連続的に推定することは難しい. 従って以下の対処が必要となる.

4.4.1 3次元マーカ位置推定の詳細

マーカ (関節点) 位置計算の詳細は以下の通りである.

1. 各視点ごとのキャリブレーション情報を用いて, カメラ座標原点, 重心を通る直線 (視線) を求める.
2. 統合する視線 $T_j (j = 1, \dots, J; \text{信頼性の高い順にならべた } J \text{ 個の視線})$ を選択する (視線選択のアルゴリズムについては後述).
3. それら視線の交点を求める. 実際はノイズなどの影響でこれらの直線はねじれの位置にあることが多い. そこで, 最もこれらの直線からその最小距離の点を求めることで近似する³.

最も信頼性の高い視線が, $\mathbf{T}_{top} = \mathbf{o}_{top} + t_{top} \mathbf{d}_{top} (t_{top}$ は媒介変数) と表現される時, また, その他の選択された信頼性の高い視線が, $\mathbf{T}_j = \mathbf{o}_j + t_j \mathbf{d}_j (t_j$ はグミーパラメータ) と表現される時, 最も信頼性の高い視線上にあり, 後者の視線群から最小距離である点 \mathbf{T} は, t_{top} をパラメータとして求めると,

³ 実際この方法では 3次元空間での誤差は比較的大きいとされるが, 仮想空間にマッピングする対象物情報の精度としては十分である.

$$\mathbf{T} = \mathbf{o}_{top} - \frac{\sum_j^J (\mathbf{d}_{top} \times \mathbf{m}_j, \mathbf{o}_{top} \times \mathbf{m}_j - \mathbf{n}_j)}{\sum_j^J \|\mathbf{d}_{top} \times \mathbf{m}_j\|^2} \mathbf{d}_{top}, (6)$$

ただし

$$\mathbf{m}_j = \frac{\mathbf{d}_j}{\sqrt{1 + \|\mathbf{o}_j \times \mathbf{d}_j\|^2}},$$

$$\mathbf{n}_j = \frac{\mathbf{o}_j \times \mathbf{d}_j}{\sqrt{1 + \|\mathbf{o}_j \times \mathbf{d}_j\|^2}}.$$

この点 \mathbf{T} が関節点位置 $(T_x^i, T_y^i, T_z^i)^T$ である。

4.4.2 多視線選択・融合アルゴリズム

各関節位置について独立に視線（群）を選択するアルゴリズムは以下の通りである。

1. 最も信頼性の高い視線，すなわち，信頼性 R の値が最大値の視線を選択する。
2. 残りの信頼性の高い視線を，まず信頼性 R のしきい値基準，

$$R > vl_{2D}, \text{ただし } vl_{2D} \text{ はしきい値.} \quad (7)$$

を用いて選択する。

3. 次に，より精度よく3次元復元計算を行なうため，2. の基準で選択された視線のうち，1. で求めた最も信頼性の高い視線との間に成立するエピポーラ制約をよく満たすもののみをさらに選択する。

- 1. で求めた最も信頼性の高い視線と各視線との間で，エピポーラ制約を満たしているか，つまり，ねじれの関係にある2視線間の距離（点 \mathbf{T}_{top} とその視線間距離） D_j が以下の基準を満たすものを選択する。

$$D_j < vl_{3D}, \text{ただし } vl_{3D} \text{ はしきい値.} \quad (8)$$

- 視線ベクトル \mathbf{d}_j と $\mathbf{T}_j - \mathbf{o}_i$ の内積が以下の条件を満たす。

$$(\mathbf{d}_j \cdot (\mathbf{T}_j - \mathbf{o}_i)) > 0 \quad (9)$$

4. その最も信頼性の高い視線をパラメータ表現し，残りの信頼性の高い視線を用いて（最も信頼性の高い視線上の）3次元位置を最小2乗推定する（前述）。

選択された視線数が2よりも小さい場合は，前フレームまでに推定した現在の関節位置を保持する。

5 多関節物体の3次元動作推定

上述の方法で，各関節点のマーカ色についてその投影可視領域を観測しマーカ領域重心を抽出し追跡し，それらの重心位置からステレオの原理により，関節点 i の位置 $(T_x^i, T_y^i, T_z^i)^T$ が計算されている時，推定した3次元関節点位置と，予め定義している多関節構造情報を用いて，ルー

ト部位から順に部位モデル (*Deformable SuperQuadratics*[3] や *Cylinder*) を当てはめる⁴。

以下，モデルを関節点間に当てはめていく方法を述べる。

位置・姿勢の計算: 部位 i が，関節点 i を基準に関節点 j と接続するように当てはめられるとする。また，相対姿勢はロール・ピッチ・ヨー表現されているとする。部位 i の関節点位置 $(T_x^i, T_y^i, T_z^i)^T$ とその子部位の関節点位置 $(T_x^j, T_y^j, T_z^j)^T$ から，部位 i の部位座標系の相対姿勢 $(R_x^i, R_y^i, R_z^i)^T$ を以下のように計算する。

$$R_x^i = 0$$

$$R_y^i = \arccos\left(\frac{T_z^j - T_z^i}{|T_z^j - T_z^i|}\right) \quad (10)$$

$$R_z^i = \arctan\left(\frac{T_y^j - T_y^i}{|T_y^j - T_y^i|} \cdot \left(\frac{T_x^j - T_x^i}{|T_x^j - T_x^i|}\right)^{-1}\right)$$

現在は簡易なプリミティブモデルを用いているので，部位の長軸に対する形状の対称性を仮定すればこのように姿勢の自由度は2で良い。この式を用いれば，複雑な姿勢計算における例外処理を回避することができ，計算時間の削減，安定な姿勢計算を行なえる。また部位 i の基準位置は \mathbf{T}^i である。人体については，腕，足がこの計算を適用する部位に相当し，再帰的に全部位の部位座標系の相対姿勢を求める。

胴体・頭の位置・姿勢の計算: 我々のシステムでは，なるべくマーカ数を少なくするため，胴体・頭はマーカを用いず自動計算する。胴体の位置・姿勢は両腕の関節と両足の関節より導出し，頭は両腕の関節より導出する。

$$\mathbf{T}^s = \frac{\mathbf{T}^{ls} + \mathbf{T}^{rs}}{2} \quad (11)$$

$$\mathbf{T}^t = \frac{\mathbf{T}^{ll} + \mathbf{T}^{rl}}{2},$$

ここで，'ls'，'rs'，'ll'，'rl' は左肩，右肩，左上肢，右上肢の関節位置を表している。

計算した \mathbf{T}^s と \mathbf{T}^t を仮想的に関節点とみなし，上記の姿勢計算を行なうことで，胴体の姿勢を推定できる。また，胴体の位置は，その4つの関節点の重心位置とすればよい。頭については \mathbf{T}^s を仮想的に関節点とみなし，その位置に設定する。姿勢の自由度は本論文では0であるが，もし必要であればマーカを頭部に設定し，同様に計算することができる。

6 実験結果

以下では，実時間システムとしての性能を実証するための実験について述べる。さらに，本報告で提案した多視線選択・融合アルゴリズムの効果を6つのカメラを用いた実験により検証する。

⁴当てはめる部位モデル形状の種類は何でも良いが，本論文においては，簡単のため *Cylinder* モデルを用いている。

6.1 システム構成

本システムは以下のような装置を利用している (図 2)。

PC クラスタ: 我々の PC クラスタは, 10 個の PC (CPU は Pentium II) で構成され, 高速な PC 間通信を実現するため, Myricom 社の Myrinet ボードを用いている。また通信ライブラリとして, RWCP で開発された Pm を利用している。なお, PC クラスタは Linux 上に構築している。

センサ: PC クラスタのうち, 6 台の PC には 外部同期信号発生器により駆動された Victor 製 KY-F57 3CCD カメラ (焦点距離 4mm のレンズを利用) および Imaging Technology 社 ICPCI フレームグラバ (640 × 480, 24bit カラー画像を獲得可能。インターレースの影響除去のため 320 × 240 にリサイズして用いる) が接続されている。

表示装置: 計算機で生成されたレンダリング結果を投影するプロジェクタ。また, 実時間仮想空間レンダリングのために, OpenGL アクセラレータを利用している。

その他: 12 本フリッカレス蛍光灯, マルチカメラキャリブレーションのための参照物体, 12 個のカラーマーカ (半径 35mm)。

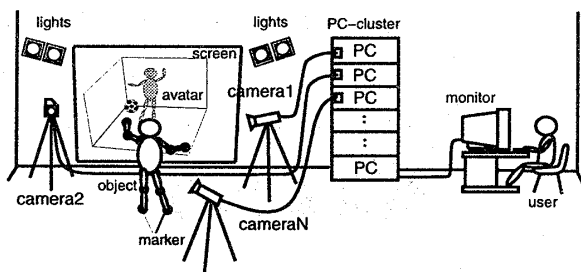


図 2: モーションキャプチャシステムの構成

6.2 視覚に基づく仮想空間と実空間の実時間インタラクション

本手法を, 視覚に基づく仮想空間と実空間の実時間インタラクションシステム (*Visually Guided 3D Animation*) として実装した (図 3, 実験 1 参照)。具体的には, マーカをつけた被計測者をモーションキャプチャし, 被計測者が, 仮想空間上の Avatar として CG のサッカーボールを実時間で操作する (蹴る, または打つ) という例題にした。そのインタラクションは簡単な衝突検出およびリアクション (跳ね返り) により実現している。

6.3 実験 1: 全身モーションキャプチャ

上記の視覚に基づく仮想空間と実空間の実時間インタラクションシステムを PC クラスタ上に実装し, オンライン

かつ実時間で (30 fps) で動作させた。図 3 は測定対象および, 計算機内の仮想空間を大型スクリーンに表示したスナップショットである。潜在的にパイプライン処理による遅れ (33msec × 3) が生じるが, それ以外の遅れは生じることなく動作した。



図 3: オンラインデモ: 計測対象 (左) と仮想空間像 (右)

6.4 実験 2: 多視線融合の効果 (6 視点の場合)

多視線融合の効果を検証するために, オフラインの 6 視点時系列 400 × 6 枚に対し, 本報告で提案した視線選択・融合アルゴリズムを適用した (図 4)。6 視点カメラの配置としてはカメラの光軸間が均等に 60 度になるよう, 対象物を中心にその円周上に配置した。本実験においては, 対象物として上部にマーカ (半径 17mm) をつけた模型の電車をを用い, レール上を等速円運動させた。また視点融合の効果をみるため, 対象物とその運動とともにある視点方向からは隠れるよう, 2 つの障害物を設置した (図 4)⁵。また, 本実験においては, 実験の都合上, 6 視点のオフラインで撮影した画像系列を用いているが, 実時間で動作可能である。

図 5 は, 推定したマーカの 3 次元位置の軌跡を示している (T_x, T_y, T_z はそれぞれ推定された x, y, z 座標)。この図において, 上のグラフ (a) は 2 視点 (視点 1, 2) のみ用いた場合の結果であり, 下のグラフ (b) は 6 視点全て用いた場合の結果である。例えば, 図グラフ (a) において, 視点 2 の 163 フレームより, オクルージョンが起り始めたため, 推定に失敗しているのがわかる (図 4 の 163 フレーム)。しかし, 6 視点全て用いたグラフ (b) の結果においては, 視線融合により安定に推定が行なわれており, その効果は明らかである。

7 おわりに

多視点カメラを利用したカラーマーカ追跡に基づいた実時間モーションキャプチャシステムを開発した。また, 仮想空間での実時間インタラクションを必要とする仮想現実アプリケーションを本システムに実際に適用し, その応用可能性を示した。多視線選択・融合アルゴリズムを提案し, 実験によりその効果を確認した。

今後の課題としては, マーカを用いている本プロトタイプシステムをもとに, 完全非接触な方法による全身動作の

⁵この実験は基本性能わかりやすくするため, 簡単な対象物のみで行なった。もちろん全身人間への適用も行なっているがその実験結果については紙面の都合上, 省略する。

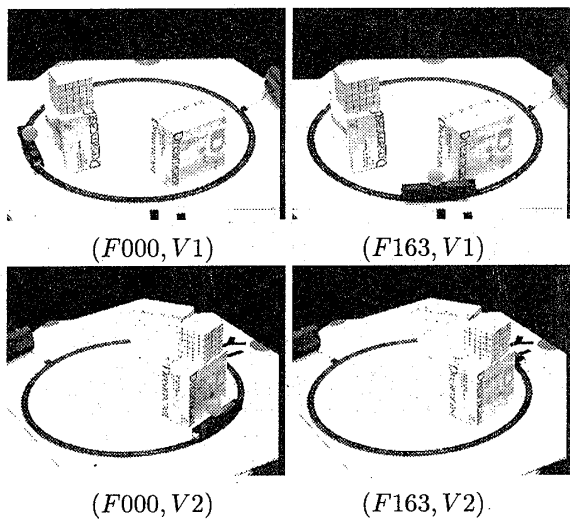


図4: 初期フレームおよびオクルージョンの発生したフレーム (163) の例 (視点1, 2のみ)

推定を行なうシステムを開発すること, および物理的な制約の導入による推定精度向上, 形状・表面情報などの獲得などが挙げられる.

謝辞

本研究の一部は, 日本学術振興会未来開拓学術研究推進事業「分散協調視覚による動的3次元状況理解」プロジェクト (JSPS-RFTF 96P00501) の補助を受けて行った.

参考文献

- [1] C.Wren, A.Azarbayejani, T.Darrell, A.Pentland, "Pfinder: Real-Time Tracking of the Human Body", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.19, No.7, pp.780-785, 1997.
- [2] 松山隆司, "分散協調視覚 - 視覚・行動・コミュニケーション機能の統合による知能の創発-", 画像の認識・理解シンポジウム MIRU'98, TP1-1, 1998.
- [3] S.Yonemoto, N.Tsuruta, and R.Taniguchi, "Tracking of 3D Multi-Part Objects Using Multiple Viewpoint Time-Varying Sequences", *Proc. ICPR98*, pp.490-494, 1998.
- [4] C.Bregler and J.Malik, "Tracking People with Twists and Exponential Maps", *Proc. CVPR98*, pp.8-15, 1998.
- [5] 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.3, No.4, pp.323-344, 1987.
- [6] D.Arita, N.Tsuruta and R.Taniguchi, "Real-time parallel video processing on PC-cluster", *SPIE-3452, Parallel and Distributed Methods for Image Processing II*, pp.23-32, 1998.

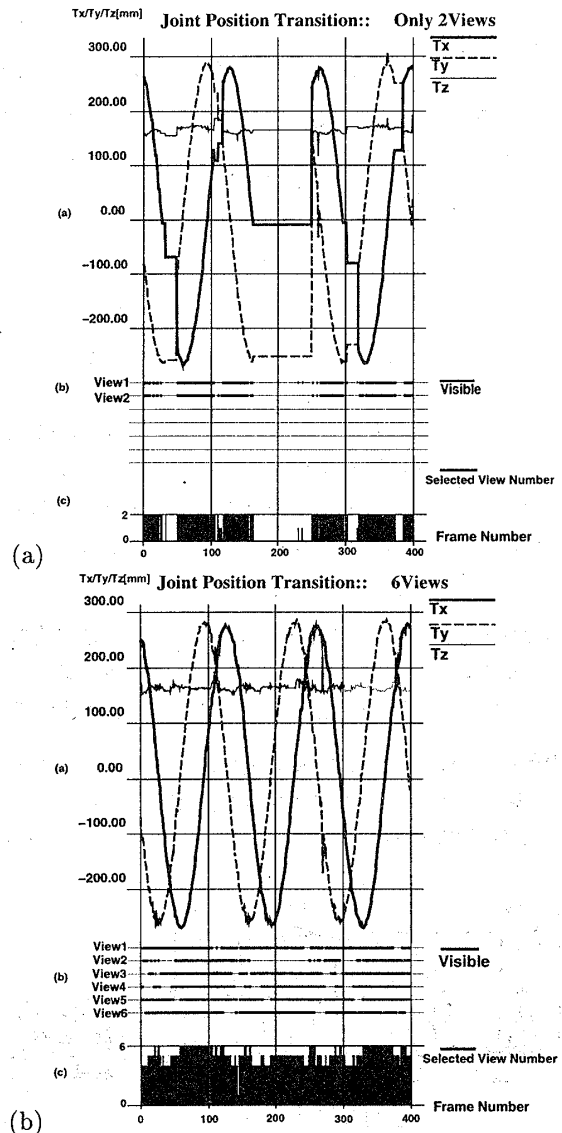


図5: 2視点のみの場合 (上), 6視点の場合 (下) のマーカーの推定位置の軌跡

- [7] R.Taniguchi and D.Arita, "A Basic Framework of Real-Time Image Processing on PC-cluster", *2th International Workshop on Cooperative Distributed Vision*, pp.119-132, 1998.
- [8] 岡本, R.Cipolla, 風間, 久野, "定性的運動認識を用いたヒューマンインターフェイス", 電子情報通信学会論文誌, Vol.J76-D-II, No.8, pp.1813-1821, 1993.