

PC クラスタを用いたリアルタイム手形状推定

藪上 勝宏[†] 上田 悦子[†] 松本 吉央[†] 小笠原 司[†]

奈良先端科学技術大学院大学 (katsu-ya,etsuko-u,yoshio,ogasawar@is.aist-nara.ac.jp)

1 はじめに

人間の手動作は、言葉、感情の表現などといったコミュニケーションの手段として、あるいは道具や形状の操作手段として多くの機能を有する。この人間の手動作をコンピュータとのインタラクションとして用いることは有効なアプローチであるといえる。そのためには、手の3次元形状をリアルタイムで推定することが必要である。

我々は操作的な手動作入力インターフェースに応用可能な、多視点画像に基づく手の形状推定法を提案してきた [1]。しかしながら、処理に多大な計算資源を必要とするため、処理速度十分であるとは言えなかった。

処理速度を得るための方法の一つとして PC クラスタがある。画像処理や動作計測のために PC クラスタを用いたリアルタイムシステムには、画像データベース探索の並列化による手形状推定 [2] や、人体動作の3次元表現 [3]、モーションキャプチャ [4] などがあげられる。本稿では提案してきた手形状推定の処理速度を向上させるための PC クラスタへの実装とその結果について述べる。

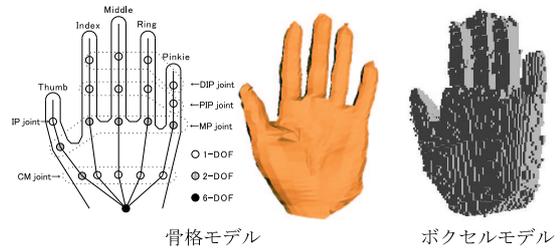


図 1: 手の形状表現モデル

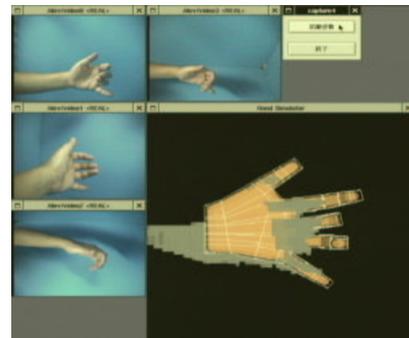


図 2: 手形状推定の様子

2 手形状推定アルゴリズム

これまでに提案してきた手形状推定手法の概要について述べる。この手法は予めシステム内に構築した図 1 に示す骨格モデルと呼ばれる手のモデルと多視点画像から得られた3次元化したボクセルモデルと呼ばれる観測データとの3次元フィッティングをもとにしている。図 2 は実際の手形状推定の様子である。処理の流れを以下に示す。

- Step1: 4台のカメラから手画像を得る。
- Step2: 得られた画像から手のシルエットを抽出し、片側距離変換マップ [5] を作成する。
- Step3: 片側距離変換マップを用いて3次元空間中の手領域を表すボクセルモデルを作成する。
- Step4: ボクセルモデルと骨格モデルとのフィッティングを行い手の形状を推定する。

3 PC クラスタを用いた手形状推定

本研究で用いる PC クラスタは、Score5.4 を用いて 9 台の PC (Dual Xeon 2.4GHz, Memory 1GB, Gigabit Ethernet) より構成されている。そのうち 8 台が計算用、残り 1 台が処理用である。なお、Xeon プロセッサに搭載された Hyper Threading 機能により、1 台のホストは 4 つの CPU を仮想的に持つ。

Realtime Hand Pose Estimation using PC Cluster System
Katsuhiko Yabugami, Etsuko Ueda, Yoshio Matsumoto, and Tsukasa Ogasawara

[†]Nara Institute of Science and Technology.

3.1 アルゴリズムの PC クラスタへの実装

手形状推定アルゴリズムは 2 章で述べたような処理のため、Step1, Step2 のビデオキャプチャ・画像処理、Step3 のボクセルモデル作成、Step4 の手形状推定、の 3 つの処理に分割し、パイプライン化を行う。また、画像処理およびボクセルモデル作成処理においてはそれぞれ担当するホスト内で処理を分割して並列化を行う。

図 3 に PC クラスタを用いた手形状推定システムについて示す。今回の実装では計算用ホスト 8 台のうち 5 台を使用している。Host1 および Host2 はそれぞれ 2 台の IEEE1394 カメラと接続されており、画像のキャプチャ・画像処理を行い、片側距離変換マップを作成する。画像処理は各 CPU に各カメラの画像を 1 枚ずつ割り当てることで並列化している。片側距離変換マップはネットワークを通じて Host3 および Host4 で共有される。Host3 および Host4 では、それぞれ 4 つの CPU を用いて手のボ

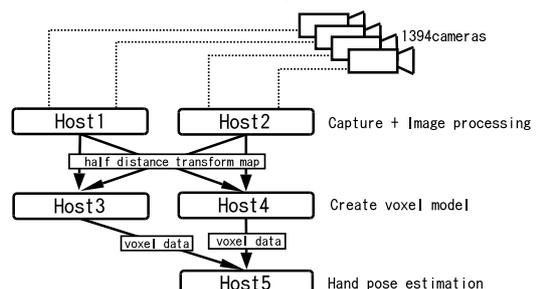


図 3: PC クラスタを用いた手形状推定システム

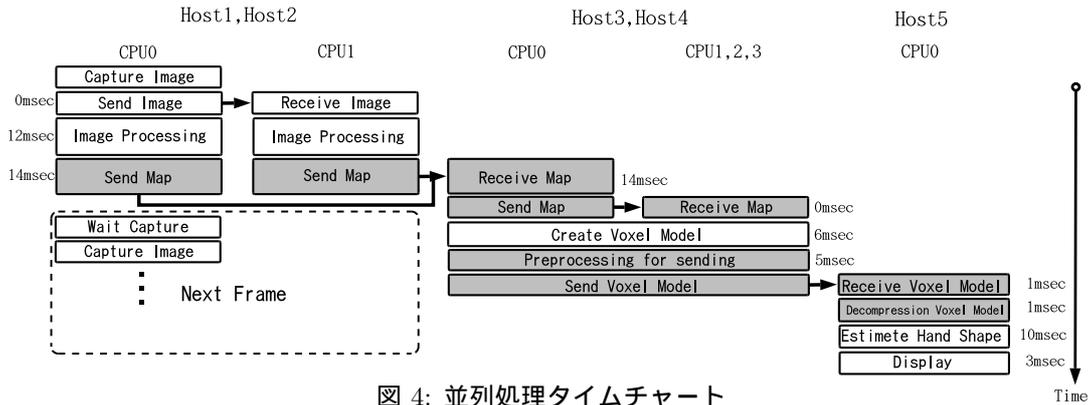


図 4: 並列処理タイムチャート

表 1: 処理時間の比較

	従来手法	提案手法
キャプチャ・画像処理	130 ms	26 ms
ボクセル作成	50 ms	26 ms
手形状推定	160 ms	15 ms
処理遅延	340 ms	52 ms
処理サイクル	340 ms	33 ms

クセルモデル作成を並列に行う．立方体のボクセル空間を 8 つに等分割し，それぞれを 8 つの CPU で分担し，ボクセルを作成することで処理を並列に行う．Host5 において，ボクセルモデルと手の骨格モデルとのフィッティングを行い，手形状の推定を行う．ボクセルモデルの生データ (16MB) を毎回転送すると約 125ms の通信時間が必要となるが，1 フレーム前のボクセルモデルとの差分値を転送することで差分の処理時間約 5ms，通信時間約 1ms の計 6ms 程度に抑えている．

3.2 PC クラスタへの実装の評価

並列処理のタイムチャートを図 4 に示す．キャプチャから推定結果の表示まで約 52ms かかるが，各ホストでの処理が最大 26ms で終わっていることから，ビデオレート (33ms 周期) で手形状推定が行うことができている．

並列化を行わずにクラスタ上の 1 つの CPU に実装した場合，通信時間，通信処理の時間 (図 4 の灰色部分) がなくなるが，その他の処理をシーケンシャルに行うため，約 99ms 必要となる (なお，Host3,4 のボクセルモデル作成は Hyper Threading による並列化のロスがあるため，1 つの CPU で実際にボクセルモデル作成を行うと処理時間は約 34ms になる)．従って，本システムでは，Hyper Threading による仮想的な CPU を 13 個 (実 CPU としては 9 個) 用いることで 3.8 倍の高速化が実現できたといえる．なお，このシステムの処理時間と通信時間を合わせた CPU 利用率は，56.7% である．

表 1 に従来システム (CPU:PentiumIII 1GHz Dual) との処理時間の比較を示す．従来手法では 1 台の PC を用いていたため，処理遅延，処理サイクルともに約 340ms であったが，PC クラスタを用いることで遅延時間が 1/7，処理の 1 サイクルにかかる時間が 1/10 となった．

3.3 手形状推定の応用

提案した手形状推定システムを用いて図 5 に示すようなバーチャルろくろシステムへと応用した．このシステ

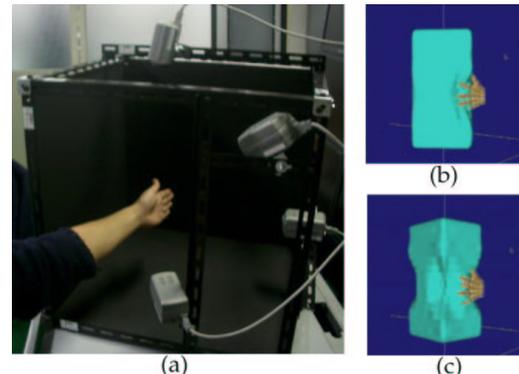


図 5: バーチャルろくろシステム

ムは，図 5(a) の観測システム中の手動作を用いて図 5(b) のようにバーチャル空間中の回転する粘土をろくろの上で陶芸を行うように変形させることができる．図 5(c) は変形後の様子である．

4 おわりに

本報告では従来提案してきた手形状推定アルゴリズムを PC クラスタ上へ実装を行い，ビデオレートでの手形状推定が可能となった．また，操作的な手動作を高速に推定できることから，3 次元モデリングシステムインタフェースとしてバーチャルクレイを变形するシステムへの応用を行った．今後は，推定精度の向上をはかるとともに，実環境で使用できる自然で直感的な 3 次元モデリングシステムのインタフェースを構築することを目指す．

なお，本研究の一部は文部科学省の科研費 (萌芽研究 14658115) および 21 世紀 COE プログラム「ユビキタス統合メディアコンピューティング」の助成を得た．

参考文献

- [1] Etsuko Ueda, Yoshio Matsumoto, Masakazu Imai, Tsukasa Ogasawara: "A Hand-Pose Estimation for Vision-Based Human Interfaces", IEEE Transactions on Industrial Electronics, Vol.50, No.4, pp.676-684, 2003.
- [2] N. Shimada, K. Kimura and Y. Shirai: "Real-time 3-D Hand Posture Estimation based on 2-D Appearance Retrieval Using Monocular Camera", ICCV2001, pp.23-30, 2001.
- [3] ウ 小軍, 東海彰吾, 和田俊和, 松山隆司: "PC クラスタを用いた身体動作の実時間 3 次元映像化", 情報処理学会研究報告, Vol.2000-CVIM-121, p65-72, 2000.
- [4] 有田 大作, 濱田 義雄, 米元 聡, 谷口 倫一郎: "PC クラスタを利用した実時間並列画像処理環境 RPV", 電子情報通信学会論文誌, Vol.J84-D-II, No.6 ,pp965-975, 2001.
- [5] Richard Szeliski: "Rapid Octree Construction from Image Sequences", CVGIP:Image Understanding, Vol.58, No.1, pp.23-32, 1993.