

細粒度データ転送機構による実時間並列画像処理環境RPVの低レイテンシ化

有田, 大作
九州大学システム情報科学研究院知能システム学部門

谷口, 倫一郎
九州大学システム情報科学研究院知能システム学部門

<https://hdl.handle.net/2324/5674>

出版情報：電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解. 100 (702), pp.79-86, 2001-03-16. 電子情報通信学会

バージョン：

権利関係：

細粒度データ転送機構による 実時間並列画像処理環境RPVの低レイテンシ化

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院知能システム学部門

福岡県春日市春日公園 6-1

092-583-7618

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

あらまし我々はこれまでの研究により, 分散並列計算機を利用した実時間並列画像処理環境 RPV を実現した. しかし RPV の問題点として, 計算機をパイプライン型に並べることにより計算能力 (スループット) を向上させることが可能にもかかわらず, 画像が入力されてから結果が出力されるまでの時間 (レイテンシ) が増大してしまうことが挙げられた. 今回の発表では, この問題を解決するために細粒度データ転送機構を導入した RPV-II について説明し, また, これを利用した視体積交差法による対象物の 3 次元形状復元システムについて述べる.

キーワード 動画画像処理, 実時間処理, 3 次元形状復元, 細粒度データ転送, PC クラスタ

Reduction of latency using fine grained data transfer on real-time parallel image processing environment RPV

Daisaku Arita, Rin-ichiro Taniguchi

Department of Intelligent Systems, Kyushu University

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

+81-92-583-7618

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

Abstract We have developed RPV, a real-time parallel image processing environment. However it has a problem that pipeline-based parallel processing scheme, which makes the computation power (throughput) higher makes latency, or the time interval between capturing images and outputting results larger. In this report, we describe RPV-II supporting fine grained data transfer mechanism which solves the latency problem. Also we present volume reconstruction system using visual cone intersection method on RPV-II.

key words video processing, real-time processing, volume reconstruction, fine grained data transfer, PC-cluster

1 はじめに

近年、コンピュータビジョンの応用として、広域監視、モーションキャプチャ、遠隔講義などの実時間画像処理研究が行われるようになってきた。これらを実現する計算機システムには、複数カメラを接続できるだけのI/O能力と獲得された画像データを実時間で処理するだけの計算能力が要求される。このため、このようなシステムを1台の計算機で実現することは難しく、複数の計算機を高速ネットワークで接続した分散並列計算機を利用することが考えられる[1, 2, 3, 4, 5]。しかし、分散並列計算機上に実時間画像処理システムを構築する場合、実時間データ転送機構、同期機構、エラー処理機構といった機能を実現しなければならず、このためには、ハードウェアやOSなどの知識が必要であり、プログラミングは容易ではない。そこで我々は、分散並列計算機上で実時間画像処理システムを構築するためのプログラミング環境RPVを提案してきた[1, 2, 6]。これにより、アプリケーションプログラマは、RPVが提供する上記の機能を利用し、計算機間のデータフローと各計算機上のデータ処理タスクを記述するだけで、容易に実時間画像処理システムを構築することが可能となる。

しかし、計算能力(スループット)を高めるためにRPVは計算機をパイプライン型に並べる並列方式を採用しており、その副作用として画像が入力されてから結果が出力されるまでの時間(レイテンシ)が増大してしまうという問題点があった。これは出力結果を保存していくだけのアプリケーションであれば問題とはならないが、マンマシンインタフェイスのような対話型のアプリケーションでは大きな問題となってしまう。そこで本報告ではこの問題を解決するために、RPVに細粒度データ転送機能を加えたRPV-IIを提案する。これにより多くのアプリケーションにおいてレイテンシを大幅に削減することが可能となる。さらに本報告では、RPV-IIを用いて視体積交差法[7]による実時間3次元形状復元システムを構築し、レイテンシが削減されていることを確かめる。

2 実時間並列画像処理環境 RPV

2.1 概要

我々の研究では分散並列計算機としてPCクラスタを利用している。これは24台のPC(PentiumIII SMP, Linux)をギガビットネットワークの一種であるミリネット¹によって接続したものである(図1と図2参照)。これに6台のIEEE1394ベースデジタルカメラ[8]が接続されており、非圧縮画像を実時間で獲得することが可能である。

¹Myricom, USA

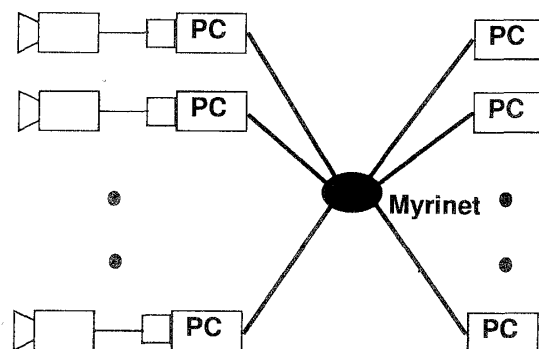


図 1: PC クラスタ

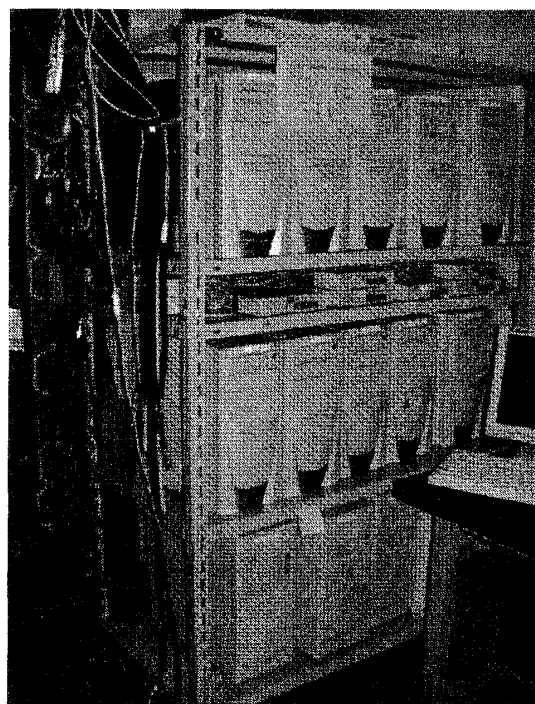


図 2: PC クラスタの外観

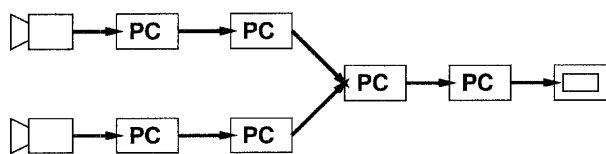
このようなPCクラスタ上で、以下のような並列処理方式とその組み合わせを実現している。プログラミング構造の視点から見ると、それぞれのPCは画像処理の構造化プログラムの一つの要素に対応している。

データ統合 複数のカメラによって獲得されたデータを後続のPCで統合する(図3(a)参照)。

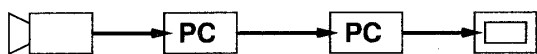
パイプライン並列処理 処理全体を部分関数に分割し、それぞれを異なるPCに割り当てる(図3(b)参照)。

データ並列処理 データ全体を複数の部分データに分割し、それぞれを異なるPCに割り当てる(図3(c)参照)。

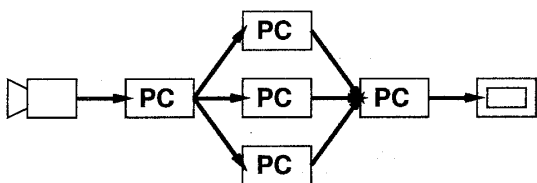
機能並列処理 データをマルチキャストし、複数のPCで



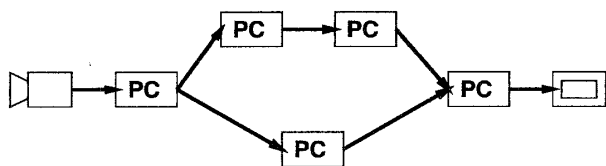
(a) データ統合



(b) パイプライン並列処理



(c) データ並列処理



(d) 機能並列処理

図 3: 並列処理方式

異なる処理を行い、後続の PC で結果を統合する(図 3(d) 参照)。

2.2 実時間並列画像処理のための機能

前節で述べたような並列処理方式を実現するためには、以下の機能がシステムには要求される。

実時間データ転送機構 ミリネット上の通信ライブラリとしては RWCP が開発した PM ライブラリ [9] を利用しており、これは実時間動画転送 (640 × 480 のカラー画像を 30fps で転送) を行うに十分大きな実効帯域を持っている。また、効率よくデータ転送を行うために、各 PC 上で複数のモジュール (実体は UNIX のプロセス) を起動し、データの受信、処理、送信を並行して行えるようにしている。

同期機構 実時間処理のためには各 PC がデータ処理やデータ転送の遅れが起きていないかどうか知る必要

がある。このために、1 フレーム時間ごとフレーム同期信号 (FSS) をすべての PC に送信し、各 PC は FSS を受け取ると次フレームのデータに対する処理を開始するようにする。FSS を受け取ったときに、前フレームのデータに対する処理が終了していなかった場合や、次フレームのデータが受信されていなかった場合は、遅れが発生しているということになる。

エラー処理機構 同期処理によって遅れが判明した場合、エラー状態から正常状態に復帰する必要がある。これを行うのがエラー処理機構である。データ処理タスクに合わせて、アプリケーションプログラマはエラー処理の方法を選択することができる。

3 ストリームデータ転送方式

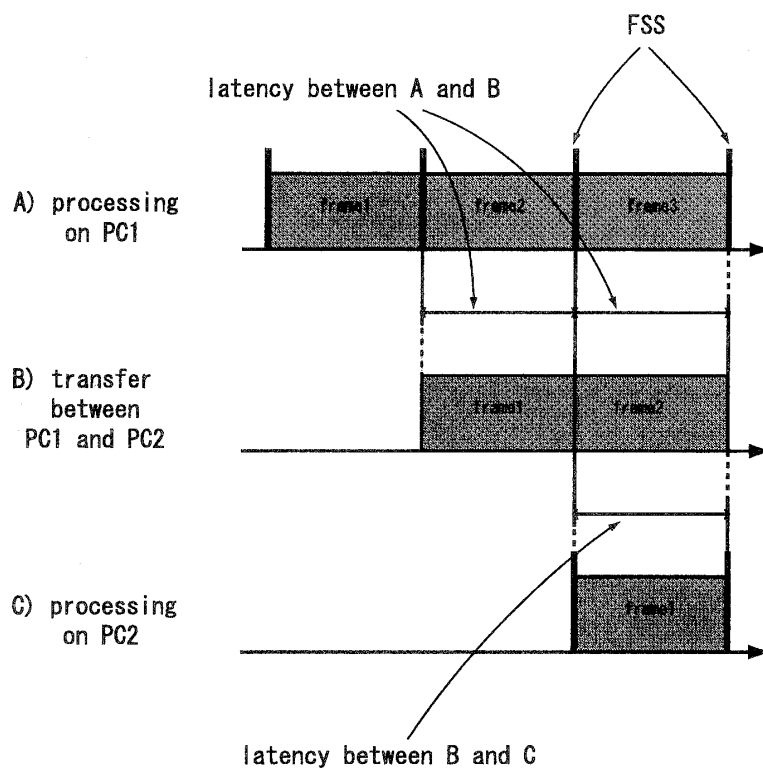
複数の PC を利用することにより総計算能力が増すので、システム全体のスループットを向上させることができる。しかし、このときパイプラインの段数に比例してレイテンシが増大してしまう。これは、RPV における同期データ転送を利用すると、すべての PC が 1 フレーム時間 τ (NTSC カメラを利用した場合 33 ミリ秒) おきに到着する FSS に同期して動作していることにより、データ転送 1 段につき 1 フレーム時間、データ処理 1 段につき 1 フレーム時間のレイテンシが加えられるからである (図 4(a) 参照)。したがって、 N 台の PC によるパイプライン並列処理を行うとそのレイテンシは $2N\tau$ になる。例えば 3 段のパイプライン並列処理によるレイテンシは約 0.2 秒になり、インタラクティブなアプリケーションにとっては大きな問題となってしまう。

これを解決するために、細粒度データ転送を行うストリームデータ転送方式を導入した RPV-II を提案する。ストリームデータ転送方式は、1 フレーム分のデータを小さなパケット (例えば画素の集合、ボクセルの集合など) に分割し、パケット単位でデータ転送やデータ処理を行う方式である。これにより各 PC において、到着したパケットから順次処理を行い、処理が終了したパケットから順次送信を行うことで、レイテンシを大幅に減少させることが可能となる (図 4(b) 参照)。

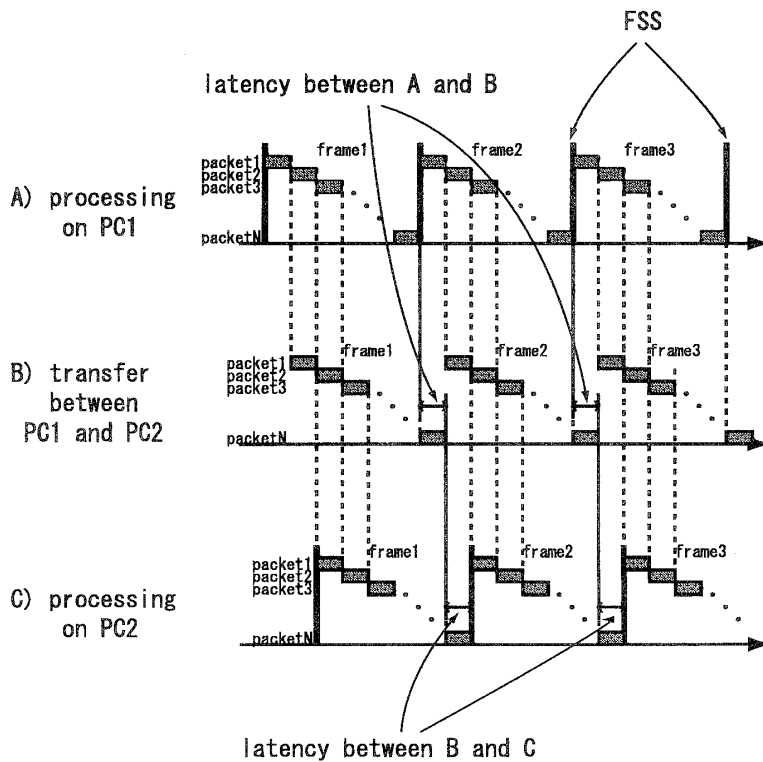
これまでの同期データ転送方式に比べて、ストリームデータ転送方式の特徴としては以下の点が挙げられる。

- スループットを保ったままレイテンシを減少させることができること 各 PC でのデータ処理量は同じなので、スループットは変わらずに²、レイテンシは $1/M$ になる (M は 1 フレームあたりのパケット数)。

²ただし、パケットの数が多くなるとデータ転送のオーバーヘッドが無視できなくなるので、スループットは低下してしまう。



(a) 同期データ転送



(b) ストリームデータ転送

図 4: データ転送方式の比較

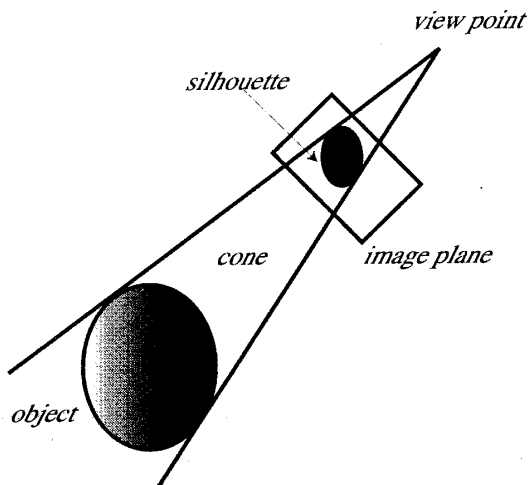


図 5: 視体積

つまり、全体のレイテンシは $2N\tau$ から $2N\tau/M$ となる。

- 前後の計算機において、同じパケットの順番でデータ処理を行う必要があること 前段の PC での処理によって出力された順番で、後段の PC はパケットを受信することになるので、後段の PC において同じ順番で処理を行うことができない場合は、処理が可能になるまでの時間 (通常は 1 フレーム分のパケットがすべて届くまでの時間) がレイテンシに加えられてしまう。

4 実時間 3 次元形状復元システム

RPV-II におけるストリームデータ転送の効果を確かめるために、3 次元形状復元システムを実装した。ここでは複数視点からの動画像を利用した視体積交差法 [7] により 3 次元形状復元を行った。視体積とは視点を頂点、その視点から見た対象物のシルエットを断面とするような錐体のことである (図 5)。対象物はこの視体積の中に含まれるので、複数の視点からの視体積の交差をとることにより、対象物の 3 次元形状を求めることができる。

4.1 システム構成

本システムは以下の 4 ステージによって構成される (図 6 参照)。

1. 対象物のシルエットを抽出する (EOS) 各視点ごとに、背景差分により画像中の対象物シルエットを抽出する。今回は単純に背景画像と獲得画像の差の絶対値をしきい値処理した結果を対象物のシルエットとした。

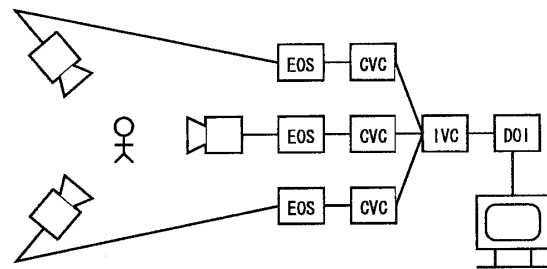


図 6: システム構成

背景差分によりシルエット画像の生成と送信はどちらも画素順で行われるの。したがって、データ処理とデータ送信の順番が同じになり、EOS から CVC へのデータ転送によるレイテンシは τ/M_p となる (M_p はシルエット画像のパケットへの分割数)。しかし、カメラからの画像はフレーム単位で獲得されるので、データ受信とデータ処理の順番の不整合が起り、EOS におけるデータ処理によるレイテンシは τ となる³。

2. シルエットから視体積を求める (CVC) 観測空間全体をボクセルによって分割し、各視点ごとにシルエット画像からボクセル表現された視体積を求める。各ボクセルごとにそのボクセルに対応する画素がシルエットに含まれるかどうかをチェックする。この処理は以下の二つの部分から成る。

シルエット画像へのボクセルの投影 事前にボクセルと画素の対応関係を計算し、ルックアップテーブルに保持しておく。実際の投影処理では、ルックアップテーブルを参照することにより処理時間を大幅に短縮することができる。

画素の包含判定 通常、ボクセルと画素は 1 対 1 には対応しないが、今回は処理を単純にするために 1 対 1 で対応させている。そのため、包含判定は容易に行うことができる。

シルエット画像は画素順に受信されるが、視体積はボクセル順に計算される。したがって、データ受信とデータ処理の順番に不整合が起り、CVC におけるデータ処理によるレイテンシは τ となる。視体積の送信はボクセル順に行われるので、データ処理とデータ送信の順番は同じなので、CVC から IVC へのデータ転送によるレイテンシは τ/M_v となる (M_v はボクセルデータのパケットへの分割数)。

3. 視体積の交差をとる (IVC) 視点ごとに作成された

³1394 カメラのインタフェースを改良し、画素単位で画像を獲得できるようにすることで、レイテンシを τ/M_p に削減することが可能である。

表 1: スループット (fps)

ボクセル数	M_v (M_p は 10 に固定)			
	125	1000	2500	5000
$40 \times 40 \times 40 = 64000$	30.00	30.00	30.00	30.00
$80 \times 80 \times 80 = 512000$	28.57	25.00	11.11	6.67
$100 \times 100 \times 100 = 1000000$	14.27	12.46	10.99	5.71

視体積を集めて、それらの交差をとり、ボクセル表現された対象物の 3 次元形状を復元する。

視体積の受信、視体積の交差処理、3 次元形状の送信はすべてボクセル順に行われる。したがって、IVC におけるデータ処理のレイテンシと IVC から DOI へのデータ転送によるレイテンシはともに τ/M_v となる。

4. 対象物の 3 次元形状を表示する (DOI) 復元された 3 次元データを任意の画像平面に投影し、その画像を画面に表示する。

3 次元形状の受信と画像平面への投影はどちらもボクセル順に行われる。したがって、DOI におけるデータ処理のレイテンシは τ/M_v となる。

結果として、総レイテンシは $\tau/M_p + 4\tau/M_v + 2\tau \approx 2\tau$ となる (M_p と M_v は十分に大きいと仮定)。同期データ転送方式ではレイテンシは 7τ となるので、レイテンシが大幅に削減されることが分かる。

4.2 性能評価

今回の実験では 3 台の 1394 カメラと 8 台の PC (Pentium III 700MHz SMP) によりシステムを構築した。また、観測空間は 1 辺 2m の立方体とした。

まず表 1 は、3 次元形状復元システムの最大スループット (1 秒間あたりのフレーム数) を示したものである。本システムにおける計算量はボクセル数にほぼ比例する。パケット数が少ないときは、スループットはボクセル数に反比例しており、設計通りに動作している。しかし、パケット数が多いときは、スループットがボクセル数に反比例していない。これは、データ転送にともなうオーバーヘッドが無視できないほど大きくなってしまっているからであると考えられる。なお、我々のカメラの最大フレームレートが 30fps なので、本システムの最大スループットも 30fps に制限されている。

次に表 2 はレイテンシを示したものである。実験値と設計値がほぼ一致しており、この実験からシステムが正確に動作していることが分かる⁴。

表 2: レイテンシ (ミリ秒)

ステージ	M_v (M_p is fixed on 10)			
	125	1000	2500	5000
EOS	10.154	11.163	11.140	11.070
EOS-CVC	0.309	0.337	0.350	0.329
CVC	156.260	182.816	201.501	384.612
CVI-IVC	9.904	0.143	1.644	38.518
IVC	0.187	0.073	1.107	23.135
IVC-DOI	0.315	0.155	0.501	42.939
DOI	6.407	6.326	6.390	7.195
合計 (L)	183.536	201.013	222.696	507.793
1 フレーム時間 (τ)	80.000	90.000	100.000	250.000
L/τ (実験値)	2.294	2.233	2.227	2.031
L/τ (設計値)	2.132	2.104	2.102	2.101

結果として、ボクセル数が $100 \times 100 \times 100$ 個の場合、つまりボクセルサイズが 1 辺 2cm の場合、最適なパケット数 (125 個) のときに 14.27fps のスループットと、2.294 フレーム時間のレイテンシであった。つまり、実際には $1000/14.27 \times 2.294 \approx 160$ ミリ秒のレイテンシになると思われる。

本実験は予備実験的なものであり、またスペースやキャリブレーションの都合によりカメラを 3 台しか使っていない。カメラの台数を増やす場合に問題になるのはデータを統合する PC の能力である。本システムでは IVC に入力できるボクセルデータ数は 4 視点分程度である (このとき CVC と IVC のスループットがほぼ一致する) が、IVC を多段に結合することにより 5 視点以上のシステムへと拡張することが可能である。IVC によるレイテンシはわずかであるため、IVC を多段にしてもレイテンシはほとんど増加しない。

図 7 は 3 視点からの入力画像、図 8 は復元された 3 次元形状を表示した出力画像である。出力画像は、入力画像と同じ 3 視点からの画像と仮想視点からの画像を載せている。カメラを 3 台しか使用していないため、復元された 3 次元形状の精度は決して良くはないが、今後より多くのカメラを使用することにより精度は向上するはずである。

5 おわりに

本報告では、分散並列計算機上でのストリームデータ転送による実時間並列画像処理環境 RPV-II について述べた。これによりスループットを低下させることなく、レイテンシを削減することが可能になった。さらに RPV-II を利用したアプリケーションとして、視体積交差法による 3 次元形状復元システムについて述べた。これにより、ストリームデータ転送によるレイテンシの削減が設計通

⁴レイテンシ計測のためのオーバーヘッドにより、最大フレームレートよりも低いフレームレートで実験を行っている。



(a) 視点 1

(b) 視点 2

(c) 視点 3

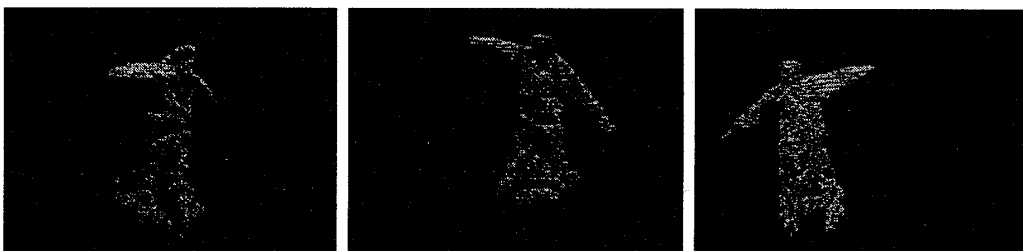
図 7: 入力画像



(a) 視点 1

(b) 視点 2

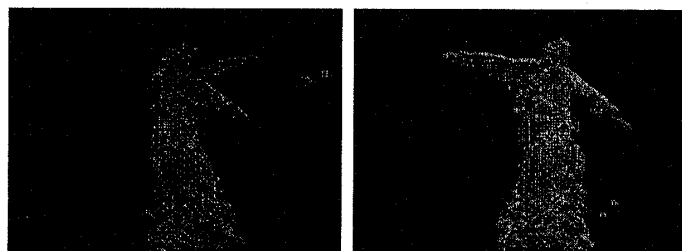
(c) 視点 3



(d) 視点 1

(e) 視点 2

(f) 視点 3



(g) 仮想視点

(h) 仮想視点

図 8: 出力画像

りに実現されていることを確認できた。

今後の課題としては以下のことが挙げられる。

- より多くの PC とカメラを使用した、本格的な 3 次元形状復元システムの構築
- より低コストなシステムを目指し、ミリネットの代わりに IEEE1394 をネットワークとして利用した PC クラスタの構築 [10].

謝辞

本研究の一部は、通信・放送機構による「次世代インテリジェント・マルチメディア情報通信網の基盤技術に関する研究」の補助を受けた。

参考文献

- [1] Daisaku Arita, Yoshio Hamada, and Rin-ichiro Taniguchi. A real-time distributed video image processing system on PC-cluster. In *Proc. of 4th International Conference of the Austrian Center for Parallel Computation*, pp. 296–305, Feb. 1999.
- [2] Daisaku Arita, Satoshi Yonemoto, and Rin-ichiro Taniguchi. Real-time computer vision on PC-cluster and its application to real-time motion capture. In *Proc. of Computer Architectures for Machine Perception*, pp. 205–214, Sep. 2000.
- [3] T. Kanade, H. Saito, and S. Vedula. The 3D room: Digitizing time-varying 3D events by synchronized multiple video streams. Technical Report CMU-RI-TR-98-34, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1998.
- [4] T. Wada, X. Wu, S. Tokai, and T. Matsuyama. Homography based parallel volume intersection: Toward real-time volume reconstruction using active cameras. In *Proc. of IEEE Workshop on Computer Architectures for Machine Perception*, pp. 331–339, Sep. 2000.
- [5] E. Borovikov and L. Davis. A distributed system for real-time volume reconstruction. In *Proc. of IEEE Workshop on Computer Architectures for Machine Perception*, pp. 183–189, Sep. 2000.
- [6] 濱田義雄, 有田大作, 谷口倫一郎. 実時間並列動画像処理ツール RPV. 情処研報コンピュータビジョンとイメージメディア (99-CVIM-119), pp. 9–16, Nov. 1999.
- [7] W. N. Martin and J. K. Aggarwal. Volumetric description of objects from multiple views. *IEEE Trans. PAMI*, Vol. 5, No. 2, pp. 150–158, 1987.
- [8] 1394 Trade Association. *1394-based Digital Camera Specification Version 1.20*.
- [9] H. Tezuka, A. Hori, Y. Ishikawa, and M. Sato. PM: An operating system coordinated high performance communication library. In P. Sloot and B. Hertzberger, editors, *High-Performance Computing and Networking*, pp. 708–717. 1225 of Lecture Notes in Computer Science, Springer-Verlag, 1997.
- [10] H. Yoshimoto, D. Arita, and R. Taniguchi. Real-time image processing on IEEE1394-based pc cluster. In *Proc. of Workshop on Parallel and Distributed Computing in Image Processing Video Processing, and Multimedia*, 2001(to be appeared).