コンピュータビジョンと
イメージメディア 115－15
（1999. 3. 19）

PCクラスタにおける実時間並列動画像処理の性能評価

有田 大作，濱田 義雄，谷口 倫一郎

九州大学大学院システム情報科学研究科知能システム学専攻
{arita,yhamada,rin}@limu.is.kyushu-u.ac.jp

我々は，複雑で高度な動画像処理や，複数のカメラを利用した動画像処理をリアルタイムで行うために，PCクラスタを利用したシステムについての研究を行っている．これは複数のPCを高速なネットワークで接続し，PC間で画像を転送しながら，並列分散動画像処理を行うシステムである．本報告では，本システムの概要について述べ，本システムの性能評価のための実験を行い，PCクラスタの有効性について述べる．

A performance evaluation of real-time parallel video processing on PC-cluster

Daisaku Arita, Yoshio Hamada, Rin-ichiro Taniguchi

Depertment of Intelligent Systems, Kyushu University

We are researching a PC-cluster system, which realizes complex and high level video processing and multi-camera video processing in real-time. The system consists of ten PCs connected by a very high speed network. In this paper, we introduce the system and show some experimental results to evaluate performance of the system.

1 はじめに

画像処理の研究を行っていく上での問題の一つに，計算量が大きいということが挙げられる．さらに，リアルタイム動画像処理では，扱わなくてはならない情報量は一層膨大になるだけでなく，時間的な制約も生じることになる．この問題を解決するために，様々な画像処理専用のハードウエアや並列計算機が開発されてきた．しかし，これらを開発するためには，時間や経費がかかりすぎるという問題がある．

また近年，複数のカメラを利用するアプリケーション，例えば広範囲な物体追跡，モーションキャプチャ，自動講義撮影，自動走行車などに関する研究が盛んに行われている．このような複数のカメラを利用する場合，物理的な制約およびI/O能力の限界により，1台の計算機に接続できるカメラの台数が制限されてしまうという問題がある．

これらの問題を解決するために，安価なPCとカメラから成る観測ステーションをネットワークによって接続した分散並列計算機環境を構築し，実世界の様々な状況を把握しようという研究が進められている [1, 2]．

最近のPCの処理能力の向上と価格の低下はめざましく，複数のPCをネットワークで結合したPCクラスタを利用することにより，性能，コストの両面ですぐれたによる並列分散システムを構築することができる [3, 4]．また，PCを利用することにより，画像キャプチャボードやビデオグラフィックスカードなどの様々な周辺機器を接続することができる．さらに，OSの選択肢も広いため，システム設計の自由度が高いという利点もある．カメラの台数を増やす場合も，PCの台数を増やすことで容易に対応できる．

ただし，これまでのPCクラスタは，並列計算機に比べて通信速度が遅く，大量のデータ通信を伴なう画像処理には向かなかった．しかし，近年の，ネットワーク技術の進歩により1Gbpsを超える高速ネットワークを低コストで利用できるようになってきた．これは非圧縮動画像のリアルタイム転送が可能な転送速度である．

そこで，我々はPC間を高速ネットワークで結合したPCクラスタ上で，容易に並列分散型の動画像処理を行うための枠組みを構築している [5, 6]．このシステムはPCクラスタ上で画像データを転送しながら並列動画像処理を実時間で行うものである．

このとき，複数のPCを利用することによる処理能力増と，通信を行うことによるコスト増の関係によっては，本システムが有効に働かない場合もありうる．そこで本稿では，本システムで実時間動画像を行うとき，一つのPCではどの程度の処理を行うことができるのか，また，どの程度の処理量であればPCクラスタで処理を行うほうが高速なのかということを，実験によって評価する．

2 PCクラスタ

2.1 PCクラスタの利点

PCクラスタを用いる利点として以下の6点が挙げられる．

複数カメラ ノードPCを増やすことで，接続可能なカメラの数を無理なく増やすことができる[1]．

コストパフォーマンス 近年のPCの高性能化および低価格化は著しい．

拡張性 ノードPCを増やすことで，容易にシステムを拡張できる．

柔軟性 汎用部品を利用しているので，最新の部品への更新が容易である．

豊富な周辺機器 画像入力やグラフィクスに関係する周辺機器が豊富にある．

OSの選択 Windows, UNIXからリアルタイムOSまで，さまざまなOSを選択できる．

2.2 ハードウエア構成

本研究で用いるPCクラスタは，10台のノードPCからなっている．各PCはPentiumIIを搭載しているAT互換機であり，OSにはLinuxを用いている．各PCは高速ネットワークMyrinetで相互に

[1] もちろん，データを統合する段階の前に情報量を減らす(優位な情報のみを取り出す)処理を行わなければ，統合部への通信がボトルネックになる．

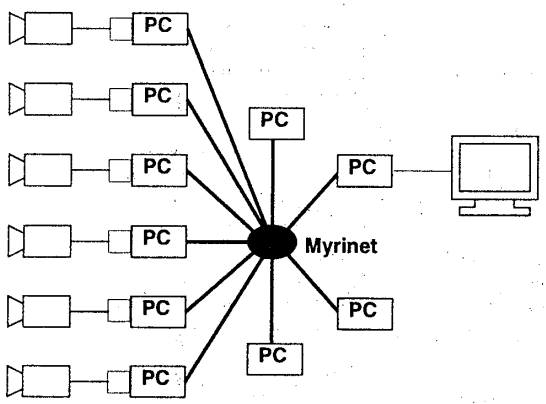


図 1: ハードウェア構成

結合されている (図 1). また, 6 台の CCD カメラがビデオキャプチャボード[2]によりノード PC に接続されている.

2.3 Myrinet と PM ライブラリ

Myrinet は, アメリカの Myricom 社が開発した, スイッチングハブを中心としたスター型の構成をとるギガビット LAN の一種である.

Myrinet 上での高速通信のために RWCP によって開発されたのが, PM 通信ライブラリである [7]. PM ライブラリの主な特徴を以下に挙げる.

- 7.5μs の低レイテンシ
- 118Mbps の高スループット
- マルチキャストサポート

3 時刻の管理

PC クラスタのような分散並列環境において動画像処理を行う場合, 時刻の管理が重要である. このために, 以下のことを行っている.

- すべてのカメラは外部同期信号を与えることによって, 完全に同期されている.
- カメラが接続された各 PC で同時刻に画像キャプチャを開始する. このために NTP を導入し, PC の内部時計を一致させている.

[2]Imaging Technology 社の IC-PCI

- カメラが接続された各 PC は, カメラからの垂直同期信号のタイミングに合わせて, フレーム同期信号 (Frame Synchronization Signal(FSS)) を発生する. FSS はデータの流れに沿って上流の PC から下流の PC へと転送される. FSS はフレーム番号を保持している. この FSS によって各 PC 間の処理の同期をとる.

4 モジュール構成

本システムで行なう動画像処理としては, 以下に挙げる 4 種類の処理方式を想定している.

パイプライン並列処理 関数全体が小さい関数に分割され, それぞれの小さい関数が各 PC で処理される (図 2 (a)).

データ並列処理 画像をいくつかの画像に分割する. それぞれの分割された画像は送り先の PC で処理される (図 2 (b)).

マルチカメラによる画像の獲得とデータの統合処理 マルチカメラによって獲得された画像は各 PC で処理された後, 途中の PC で統合される (図 2 (c)).

機能並列処理 ある PC において画像データをマルチキャストし, 各 PC で異なる関数を二つの経路により実行する (図 2 (d)).

これらの処理を効率的に実行するために, 本システムのノード PC 内では, プロセスとして (図 3) のような 4 つの基本モジュールが動いている. 各モジュール間の通信には, 割り込みと共有メモリを利用している. 以下, 各モジュールについて説明する.

データ処理モジュール Data Processing Module (DPM). 実際の画像処理を行うモジュール. データ受信モジュールからデータを受け取り, そのデータを入力として画像処理を行い, 処理結果をデータ送信モジュールに送る.

データ受信モジュール Data Receiving Module (DRM). データ受信モジュールは, 別の PC からのデー

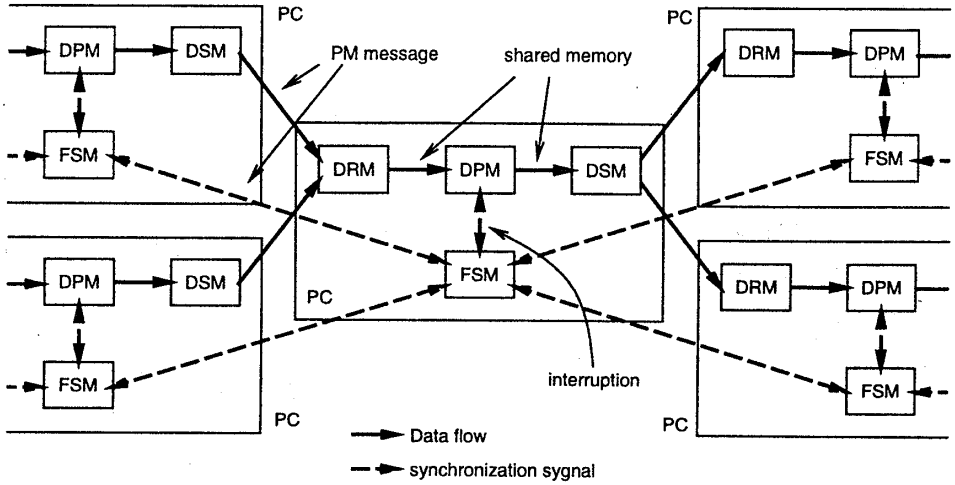
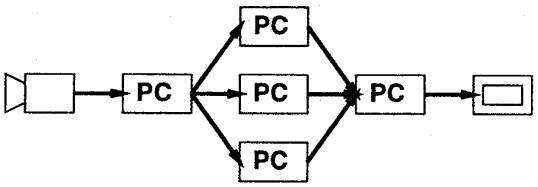
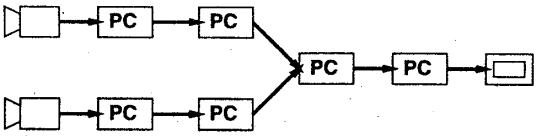


図 3: モジュール構成

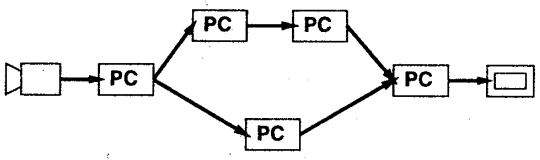
(a) パイプライン並列処理



(b) データ並列処理



(c) データ統合処理



(d) 機能並列処理

図 2: 並列処理方式

タを PM メッセージとして受信する．処理モジュールからデータを要求されると，次に処理すめきデータへのポインタを返す．また，データをキューイングするためのバッファを持っている．

データ送信モジュール Data Sending Module (DSM). データ送信モジュールは，処理モジュールからデータを受信すると，次の PC へのデータを PM メッセージとして送信する．データをキューイングするためのバッファを持っている．

フレーム同期モジュール Frame Synchronizing Module (FSM). FSM はデータ処理モジュール間の同期をとる．具体的には，FSM どうしの通信によって，FSS をデータの流れにそって上流から下流へ向けて送る．詳しくいことは後述する．

5 同期

本システムでは，実時間で画像データを受信，処理，送信することが要求される．これらのデータ転送，処理を実現するために，PC 間の同期をとることが必要である．そこで，本システムでは，以下のような 3 種類の同期機構を提供する．

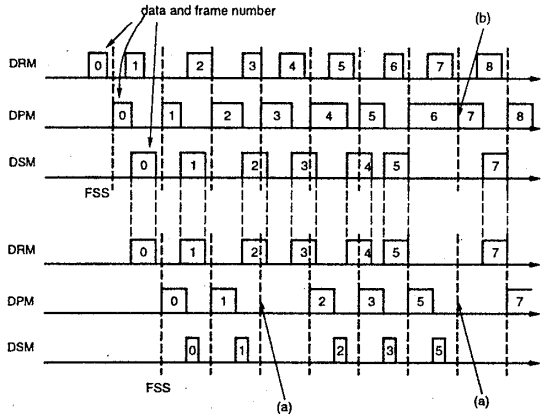


図 4: 前向き同期

5.1 前向き同期

前向き同期は，DPM に処理の開始を知らせる同期である．FSM が前段の PC の FSM から FSS を受け取ると，次段の PC の FSM に対し FSS を送信する．同時に，DPM に対しシグナルを発信する．DPM はこのシグナルを受信すると，次のフレームのデータに対する処理を開始する．DPM はシグナルを受け取ることによって，処理遅れ，データの未受信といったエラーの検出をすることができる．

図 4は連続した二つの PC のタイムテーブルである．上段の PC から下段の PC へとデータが転送されている．DRM が受け取るデータは通常は，二つの FSS の間に受信される．そのため，DPM は FSS 受信のタイミングで処理を開始することができる．しかし，(a) ではデータの到着遅れ，(b) では DPM の処理遅れがおこっており，このような場合は指定されたエラー処理が行われる．

5.2 バリア同期

複数の PC からデータを受信する場合，同時刻に処理されるべき全てのデータを待たなければならない．このための同期がバリア同期である．図 5において，注目する PC は 3 台の PC(a),(b),(c) からデータを受信している．それぞれの PC からの FSS の受信時刻は，処理経路の長さの違いから異なることがある．よって，同時刻に画像キャプチャボードにより発せられた FSS が全て受信されるまで待機

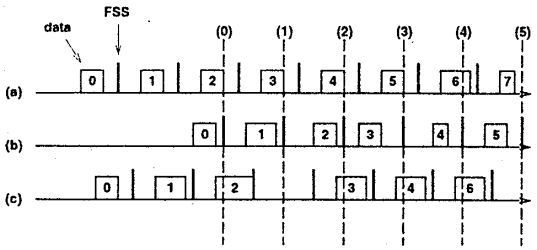


図 5: バリア同期

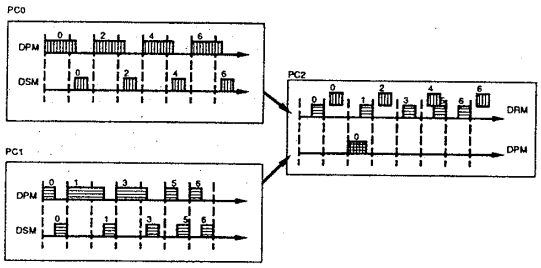
し，全て受信されると (つまり (b) からの FSS を受信すると)，FSM は次段の PC に対して FSS を送信し，処理モジュールに処理開始のシグナルを発信する．

5.3 後向き同期

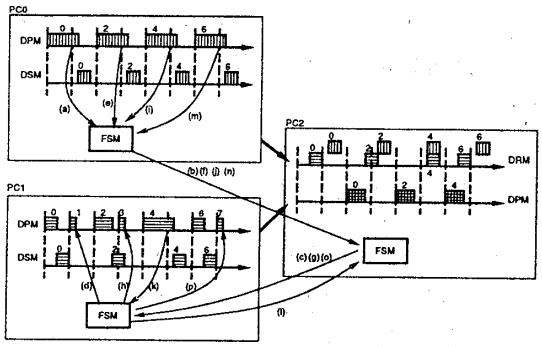
後向き同期は，無駄な処理を回避するために，他の PC に対してデータ落ちを通知するものである．データ落ちが起こるようなエラー処理を行うとき，複数の PC からデータを受信する PC において，最悪の場合ほとんど処理が実行されないことがある．これを避けるために，この後向き同期を用いる．後向き同期は，ある PC でデータ落ちが生じた場合，そのデータに対する処理のキャンセル信号を近傍の PC に送信し，そのデータに対する処理や通信を行わないようにする．

図 6(a) において，このように PC0 と PC1 で処理が行われた場合 (これは最悪の場合であるが)，PC2 はフレームデータ 0 とフレームデータ 6 の処理しか行うことができない．

それに対し図 6(a) では，後向き同期機構により，データ落ちの情報が周辺の PC に以下のように伝えられる．(a) 処理時間超過の検出 (b) フレームデータ 1 が落とされることを伝える．(c) フレームデータ 1 に対する処理が不要であることを伝える．(d) フレームデータ 1 に対する処理を中断させる．(e)(f)(g)(h) はフレームデータ 3 に対する (a)(b)(c)(d) と同様の処理である．(i)(j) も (a)(b) と同様であるが，PC1 からも同様に (k)(l) が届くので，フレームデータ 5 に対する処理は終了する．(m)(n)(o)(p) はフレームデータ 7 に対する (a)(b)(c)(d) と同様の処理である．



(a) 後向き同期なしの場合



(b) 後向き同期ありの場合

図 6: 後向き同期

このように後向き同期を行うことによって，PC2 はフレームデータ 0, 2, 4, 6 を処理することができる．

6 性能評価実験

本システムの性能を評価するために以下の実験を行った．

実験 1 PC クラスタ内部の処理時間

実験 2 PC クラスタと単体 PC との性能比較

以下それぞれの実験について述べる．

6.1 実験 1

PC クラスタを利用したときの各内部処理にかかる時間を計測した．内部処理とは

- データ転送時間
- 画像処理時間

である．

PM ライブラリが提供する 1 対 1 の通信方式としては以下の 2 種類がある．

方式 1 プログラム上で，送信側・受信側ともに明示的にデータ転送のやりとりを記述する方式．この場合，送信側・受信側ともに都合のよいタイミングでデータ転送ができるので，PC 間の同期をとるのが容易である．

方式 2 プログラム上では，送信側のみが明示的にデータ転送のやりとりを記述する方式．この場合，受信側ではデータ到着のタイミングを制御することができないため PC 間の同期をとるのが難しいが，Myrinet インタフェース上のバッファとメインメモリとのデータの転送に DMA を利用するので，データ転送を高速に行うことができる．

この 2 種類の通信方式を用いて，3 台の PC による画像処理実験を行った．それぞれの PC では以下の処理を行った．画像は 640 × 480 の 32bit カラー画像 (8bit はダミーデータ) である．

- PC0：動画像を獲得し，順次，PC1 へ送る．
- PC1：PC0 から画像データを受け取り，平滑化処理 (平均値フィルタ) を行う．そして処理後の画像データを PC2 へ送る．
- PC2:PC1 からの画像データを受け取り，ディスプレイに動画像を表示する．

画像中での平滑化処理を行う範囲を拡大縮小することによって，PC1 における画像処理時間を調節し，指定したフレームレートでの PC1 における最大の処理時間 (ms) とそのときの処理範囲 (画素数) を求めた．

まず，平均転送時間であるが，方式 1 は 26.5ms，方式 2 は 12.1ms であった．次に，処理時間であるが，これは表 1のようになった．処理量比は最大処理時間時の処理画素数の比である．

方式 2 では方式 1 に比べ画像転送時間がほぼ半減し，最大画像処理時間は増加している．また処理量についても方式 2 のほうが方式 1 よりすぐれている．さらに 3fps の時より 10fps の時の方が方式 2 に

フレームレート	3fps	10 fps
方式 1 の最大画像処理時間 (ms)	255.8	29.2
方式 2 の最大画像処理時間 (ms)	285.5	58.9
処理量比 (方式 2/方式 1)	1.20	1.98

表 1: 実験 1 の結果

よる処理量比が向上しているのは，10fps 時ほうが画像処理量にくらべて通信量が多くなるため，方式 2 による通信コスト削減の効果が大きくなるからである．これらのことから方式 2 のほうが，同期処理にかかるコストを考慮しても，PC クラスタにおける動画像処理に有効であると言える．

6.2 実験 2

実験 2 では，単体 PC における動画像処理と，データ転送方式 2 の場合の PC クラスタによる動画像処理の比較を行った．利用した PC の性能はすべて同等である．

実験 2 で行った処理の内容は実験 1 と同じである．このとき，平滑化処理の処理量を変化させながら，そのときの最高フレームレートを調べる．また，同様の処理 (平滑化処理と画像表示) を 1 台の PC で行った場合の平滑化処理の処理量と最高フレームレートの関係も調べる．ここで，(平滑化にかかる時間)/(画像表示にかかる時間) を F-D 比と呼ぶことにする．

実験 2 の結果として図 7を得た．X 軸は F-D 比であり，Y 軸は最高フレームレートである．

F-D 比が 1 から離れるほど PC クラスタを利用する効果が小さくなる．これは負荷分散をうまく行わないと，負荷の大きな PC がボトルネックになってしまうからである．

また，F-D 比が 1 のとき，処理量が半分になっているのにフレームレートは約 13/9.5 倍にしかなっていない．これから，利用可能な PC の能力のうち $(9.5\times2-13)/9.5\times2 = 6/19$ を送受信のために消費していることが分かる．同様に，F-D 比を 3 にしたとき (このとき，PC1 内での処理量は，単体 PC 内での処理量の 3/43 倍になっている) のフレームレートがほぼ同じになることから，利用可能な PC の処理能力のうち 1/4 程度をデータ送受信のために消費していることがことが分かる．このように送受信のコストが変化するのはフレームレートの違い (13fps と 3fps) のためである．ここで，フレームレートに対して通信コストが比例の関係になっていないのは，一つの PC 内で複数のモジュールを動作させることによる処理のオーバラップが，高フレームレートのときほど効果があがっているためと考えられる．

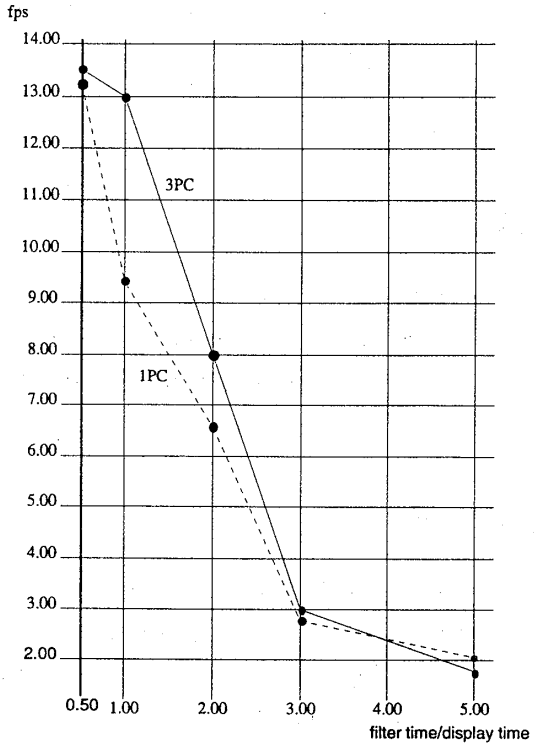


図 7: 実験 2 の結果

7 おわりに

本報告では，我々が研究している PC クラスタシステムにおける動画像処理の性能評価を行った．今後の課題としては，

- 今回はパイプライン型並列においてのみの性能評価であったので，その他の並列方式による性能評価を行わなければならないこと
- 現在よりも通信コストおよび同期コストを下げ，いわゆるビデオレート (640 × 480 画素，

フルカラー，30fps) での処理を実現できるようにすること

が挙げられる．

謝辞

本研究は，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] 松山隆司, 浅田稔, 美濃導彦, 和田俊和. “分散協調視覚プロジェクト－分散協調視覚研究システム開発の概要－”. , 情報処理学会研究会資料 , CVIM103-4, 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] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam. “PVM: Parallel Virtual Machine – A Users' Guide and Tutorial for Networked Parallel Computing”. *The MIT Press*, 1994.

[4] Message Passing Interface Forum. “MPI: A message-passing interface standard”. *International Journal of Supercomputer Applications*, Vol.8(3/4), 1994.

[5] Daisaku Arita, Naoyuki Tsuruta and Rin-ichiro Taniguchi. “Real-time parallel video image processing on PC-cluster”. *in Parallel and Distributed Methods for Image Processing II, Proceedings of SPIE*, Vol. 3452, pp.23–32, 1998.

[6] Daisaku Arita, Yoshio Hamada and Rin-ichiro Taniguchi. “A Real-time Distributed Video Image Processing System on PC-cluster”. *Proceedings of International Conference of the Austrian Center for Parallel Computation(ACPC)*, pp.296–305, 1999.

[7] H. Tezuka, A. Hori, Y. Ishikawa and M. Sato. “PM: An Operating System Coordinated High Performance Communication Library”. *High-Performance Computing and Networking* (eds. P. Sloot and B. Hertzberger), 1225 of Lecture Notes in Computer Science, Springer-Verlag, pp.708–717, 1997.