高速三次元再構成のための 時空間の分割とビデオ画像のパイプライン処理

亀田 能成† 太尾田 健男‡ 角所 考† 美濃 導彦†

† 京都大学総合情報メディアセンター
‡ 京都大学大学院工学研究科

概要 :数台のビデオカメラと分散環境にある複数のコンピュータを用いて高速に実空間のボクセル表現を得る方法を提案する．

実験環境の静物体のモデルはあらかじめコンピュータに持たせておき，空間の再構成においては静物体が占有する空間を予め処理過程から排除するとことで，動物体の形状を実時間で求める．3 次元形状獲得方法として視体積交差法の原理を用い，ビデオ画像の取り込み，画像上の動領域の抽出，及び各ボクセルにおける動物体内外判定の 3 ステージからなるシステムを構成し，パイプライン処理化及びデータの分割によりレイテンシとスループットを向上させる．

試作した実験システムにおいて，ボクセルの大きさを一辺の長さが 5cm の立方体とし，LAN 上の 4 台のコンピュータでビデオ画像処理を行い，かつ別の 4 台のコンピュータで空間の再構成を行ったところ，一秒あたり 7.2 フレーム (70 万ボクセル) の処理速度を得た．レイテンシは 0.56 秒であった．

Pipeline Video Image Processing and Division of Spatio-Temporal Space for High Speed 3D Reconstruction

Yoshinari Kameda†, Takeo Taoda‡, Koh Kakusho†, Michihiko Minoh†

† Center for Information and Multimedia Studies, Kyoto University
‡ Graduate School of Engineering, Kyoto University

Abstract: Since 3D world reconstruction methods generally require a lot of data and calculations, it takes much time to output a 3D shape. The authors propose distributed computing to achieve real-time 3D shape reconstruction.

The authors introduce the viewing frustum method (VFM) to reconstruct a 3D space from images which are captured by several video cameras. Reconstruction process of the proposed approach is divided into three stages to improve throughput by pipeline processing, and a 3D space is also divided into some subspaces to decrease latency by reconstructing subspaces simultaneously at distributed computers.

As an experiment of this method, a part of a lecture room was reconstructed by using 4 computers for image processing, and 4 more computers for VFM. The size of each voxel which is used to describe 3D shapes was a cube of 5 centimeters on a side. The throughput of the process was 7.2 frames per second, and the latency was 0.56 seconds.

1　はじめに

画像処理によって3次元世界の形状を実時間で復元し，仮想環境やマンマシンインタラクションに用いようという期待が高まりつつある．特に，複数のカメラを分散配置する方法はある空間全体の形状を再構成するために有効な方法である．これまでに，新しい視点での画像を合成する方法 [1] や，カメラの移動を利用し複数視点の画像から合成する方法 [4]，50 台以上のカメラでドームを構成し，ステレオ計測によってドーム内の様子を再構成する方法 [2] などが提案されている．しかし，このうち Hirose の手法 [1] や Tomasi の手法 [4] は実空間中の物体は全て静止していなくてはならないなど動きを捉えることについては問題がある．また，Kanade らの手法 [2] では再構成法の実例として対象人物が運動している場合を実験しているが，処理を実時間で実現する方法については議論されていない．

実空間をコンピュータ上に実時間で再構成するための最大の問題点は，データおよび処理量が膨大なため，通常の処理では時間がかかる点である．そこで本論文では，分散処理を導入し，数台のビデオカメラと分散環境にある複数のコンピュータを用いて高速に実空間をコンピュータ群に取り込む方法を提案する．空間の再構成の手法として，視体積交差法 [3][5] を利用し，対象となる時空間をボクセルで表現する．この処理の実現に際して，スループットの向上のためにビデオ画像処理の過程を分割し，各処理過程をパイプライン化すると共に，レイテンシの削減のために，対象となる時空間を分割する方法を示す．

2　視体積交差法による三次元再構成

三次元再構成の対象空間 $\mathcal{W}$ において，そこに配置されたカメラ全てについて，パン ・チルト ・ツイスト ・カメラの焦点位置などの外部パラメータ及び焦点距離等の内部パラメータは固定で既知とする．

カメラモデルに透視投影を仮定すると，各カメラにおける動領域 D_i から，動物体が存在し得る空間 $\mathcal{D}_i$ が規定できる．これはカメラ焦点を頂点とする錐体をなす．動物体が存在しうる空間 $\mathcal{C}$ は，

$$\mathcal{C} = \bigcap_{i=1}^{n_{cam}} \mathcal{D}_i \qquad (1)$$

で定義できる．$\mathcal{C}$ を CESS (Cumulative Existence Shadow Space) と呼ぶ．ボクセルに対する視体積交差法では，各ボクセルごとに $\mathcal{C}$ に属するかどうかを (1) 式を用いて判定する．

なお，実空間内の三次元形状を実時間で再構成する場合，静物体については時間的に変化しないので事前にその形状を取得しておくことで再構成にかかる計算量を削減する．また，対象空間のうち，n_{cam} 台によって観察できる空間のことを観測空間 $\mathcal{X}$ と呼ぶ．$\mathcal{C}$ は $\mathcal{X}$ 内部でのみ計算可能である．

3　分散処理とパイプライン構成

3.1　処理の分割

視体積交差法により対象空間 $\mathcal{W}$ を再構成するために，処理をビデオ画像の取り込み ・ビデオ画像から動領域の抽出 ・視体積交差法による $\mathcal{W}$ の再構成の 3 ステージに分割する．本研究では，これらのステージをそれぞれビデオサーバ ・エクストラクタ ・**3D** コンポーザの 3 種のプロセスが行う．また，これら 3 つのプロセスによる空間の再構成結果を利用するプロセスを **CESS** クライアントと呼ぶ．

ビデオサーバ ・エクストラクタ ・3D コンポーザで行われるデータ処理は独立しているため，プロセス間通信と，各プロセスで行われるデータ処理を制御し，各プロセス間の同期をとることにより，パイプラインを構成することができる．各プロセスを制御し，同期をとるプロセスを用意し，これをスケジューラと呼ぶ．

3.2　ビデオサーバ

ビデオサーバは，ビデオ画像の取り込みとビデオ画像データの送出という 2 種の処理を，他のプロセスからの命令を受けて実行する．

ビデオカメラを n_{cam} 台用意する場合，ビデオサーバは n_{cam} 個必要となる．

なお，時刻 t に撮影されたビデオ画像 $I_i(t)$ の集合をフレーム t と呼ぶ．

3.3　エクストラクタ

視体積交差法では，ビデオ画像 $I_i(t)$ 上で動領域 $D_i(t)$ の抽出を行う必要がある．

この動領域は，背景画像との差分により求める．差分画像中に多数の小領域が発生すると，視体積交差法の適用の際に計算量が増大するので，差分画像中の小領域はノイズとして除去する．

エクストラクタは上記の処理を行うプロセスであり，外部からの命令に基づき，ビデオ画像データの読み込み，動領域の抽出，動領域データの送出の 3 種の処理を実行する．

エクストラクタはカメラの台数だけ用意する．

3.4 3D コンポーザ

フレーム t の $D_i(t)$ を抽出した n_{cam} 個のエクストラクタから $D_i(t)$ のデータを受け取り，そのデータを元にボクセルデータの生成を行う．3D コンポーザのプロセスは，外部からの命令に応じて，$D_i(t)$ の読み込み，$\mathcal{W}(t)$ の再構成，空間データの送出，の 3 種の処理を行う．

なお，ボクセルが CESS に含まれるかどうかの判定は，(1) 式に示すようにそのボクセルがフレーム t における各画像平面上の動領域のいずれにも含まれているかどうかによって行われる．このため，3D コンポーザで必要な計算量は，ボクセルの画像平面への射影回数に依存する．

本節では 3D コンポーザを 1 つの処理単位とみなす．$\mathcal{W}$ の再構成は複数の 3D コンポーザを利用することによっても実現できるが，これについては対象時空間 $\widehat{\mathcal{W}}$ の分割と併せて 4節で述べる．

3.5 パイプライン構成

ビデオサーバ，エクストラクタ，3D コンポーザの 3 種のプロセス及び CESS クライアントは，データ読み込み ･データ処理 ･データ送出の 3 種の処理を繰り返し行っている．コンピュータを LAN で接続した分散処理環境では，データの処理速度に比べデータ送受信のコストが高い．このため，ネットワークの通信容量に対し転送するデータの量が少ない場合には，データの読み込みと送出は同時に行った方が効率がよいので，各プロセスの入出力待ちを多重化させる．

各プロセスについてデータの読み込みと送出の少なくともどちらか一方を行っている状態のことをこのプロセスの通信フェーズと呼ぶ．また，データ処理を行っている状態のことをプロセスの実行フェーズと呼ぶ．

ビデオカメラで撮影したビデオ画像が，ビデオサーバ ･エクストラクタ ･3D コンポーザで順番に処理されると，CESS クライアントは再構成された $\mathcal{W}(t)$ のデータを利用できる．これら 4 種のプロセス間では，通信フェーズを除いて通信は行われず，それぞれのプロセスが独立してデータ処理を行っている．このため，全てのプロセスで実行フェーズを同時に実行することができる．また，入出力を多重化することにより，通信フェーズも全てのプロセスで同時に行うことができる．

ビデオサーバ ,エクストラクタ ,3D コンポーザの実行フェーズに要する時間を各々 $T_{cam}, T_{ext}, T_{cmp}$ とすると，これらの長さがほぼ同じでかつ通信時間もほぼ同じ長さである場合 ,全てのプロセスで実行フェーズを同時に実行し ,全てのプロセスの実行フェーズ終了後に ,全てのプロセスで通信フェーズを同時に実行することにより ,パイプラインを構成することができる .全てのプロセスで実行フェーズを同時に実行している状態をシステムの実行フェーズと呼び，全てのプロセスで通信フェーズを同時に実行している状態をシステムの通信フェーズと呼ぶ．

以上のようなパイプライン構成をもとに ,3D コンポーザを 1 つ用意した場合のデータの流れを図 1に示す．

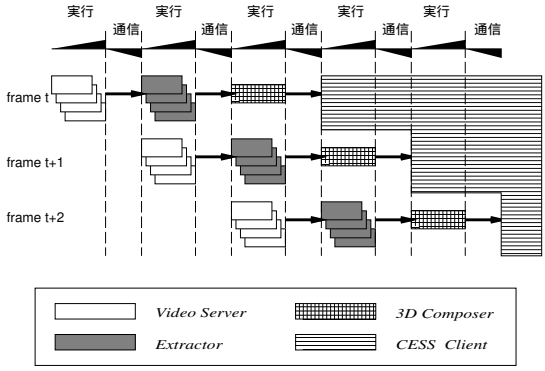

図 1: パイプラインの構成

4 観察時空間の分割と高速化

2節で述べた空間再構成法では，$\mathcal{C}(t)$ を表現する各ボクセル が動物体 $\mathcal{O}$ 内に対応し得るかどうかを，他のフレームの動物体存在可能空間 $\mathcal{C}(t')$ や，フレーム t の他のボクセルの値によらず求めることができる．すなわちフレーム t において，対象時空間 $\widehat{\mathcal{W}}$ を部分空間 $\widehat{\mathcal{W}}_j$ に分割し，分散環

境で各々独立に動物体存在可能空間を求めることで，??節で一つの処理単位として捉えていた 3D コンポーザを複数の 3D コンポーザプロセスで構成することが可能となる．

本節では，$\widehat{\mathcal{W}}$ の分割方法をいくつか挙げ，それぞれの場合の性能や特質について論じ，複数の 3D コンポーザプロセスを用意できる場合にどのような構成をとればスループットとレイテンシが向上するかについて述べる．

4.1 空間分割による高速化

3D コンポーザを r個用意できるとすると，フレーム t の対象空間 $\mathcal{W}(t)$ を r個に分割して，各 3D コンポーザで同時に部分空間 $\mathcal{W}_j(t)$ を再構成することができる．このとき，3D コンポーザ間のプロセス間通信は必要ない．

ところで，動物体が移動すると CESS の分布も変化する．このため，対象空間 $\mathcal{W}(t)$ の分割位置を一定にしたままでは，各 3D コンポーザに割り付けられる計算量が最適とならず，一部の 3D コンポーザがアイドル状態になって実行フェーズにかかる時間 $T_{cmp}(t)$ が長くなることがある．そこで，対象空間 $\mathcal{W}(t)$ の分割方法をフレームごとに変更し，各 3D コンポーザのアイドル時間の削減を図る．具体的には，各 3D コンポーザを実行するプロセッサの能力が均一であると仮定した場合，3D コンポーザ j における $T^j_{cmp}(t)$ がボクセルの画像平面への射影回数に依存することから，フレーム t での各 3D コンポーザでの射影回数をもとに，フレーム $t+1$ での各 3D コンポーザでの射影回数が同数になるように $\mathcal{W}(t+1)$ を再分割する．

4.2 時間分割による高速化

データの時間方向の局所性により，異なるフレームの対象空間 $\mathcal{W}(t_a)$ と $\mathcal{W}(t_b)$ を同時に再構成することができる．たとえばある 3D コンポーザ P_1^{cmp} が対象空間 $\mathcal{W}(t)$ を再構成している間に，別の P_2^{cmp} は $\mathcal{W}(t+1)$ を，また別の P_3^{cmp} は $\mathcal{W}(t+2)$ を再構成することができる．この場合に達成されるスループットの向上のことを時間分割による高速化と呼ぶ．

4.3 時空間分割による高速化

n_{cmp} 個の 3D コンポーザが用意できるとき，s 個ずつを 1 組として使用して対象空間 $\mathcal{W}(t)$ を再構成することが考えられる．この場合，対象時空間 $\widehat{\mathcal{W}}$ を空間方向に s 個，時間方向に r 個に分割することになる．ただし，$n_{cmp} = rs$ である．この場合に達成されるスループットとレイテンシの向上のことを，時空間分割による高速化と呼ぶ．4.1 節の空間分割は時間の分割数が 1 $(r = 1)$ の場合に，4.2節の時間分割は空間の分割数が 1 $(s = 1)$ の場合に相当する．

5 実験及び考察

本稿で提案した手法により，講義室内の動物体を再構成し CESS データを出力する実験を行った．対象空間 $\mathcal{W}$ は京都大学情報工学教室第二講義室である．空間座標系と講義室の大きさ，実験に使用した 4 台のカメラの焦点の座標を図 2 に示す．

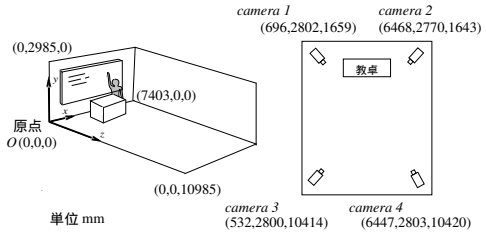

図 2: 対象空間とカメラの配置 (単位 mm)

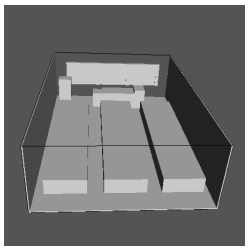

図 3: 講義室内の静物体のモデル

ビデオ画像をコンピュータに取り込む 4 個のビデオサーバは 4 台のコンピュータ上に配置する．この 4 台のコンピュータには SUN Ultra2 Model 1200 を使用した．エクストラクタもビデオサーバと同じコンピュータ上に配置した．3D コンポーザには，4 台の SUN Ultra1 Model 170 を使用した．これら 8 台のコンピュータは，データ通信用に ATM ネットワーク，同期制御通信用に 100BaseT イーサネットで接続されている．

各プロセスは C 言語と C++ 言語を用いて実装し，プロセス間の通信は TCP/IP により行った．実験結果の一例として，CESS クライアントである CESS データ表示プログラムの様子を図 4右に示す．また $r = 2$ のときのプロセス間のデータの流れ ･制御の様子を図 5 に示す．

あるカメラからの視点　　CESS データと実写の合成

図 4: 再構成例

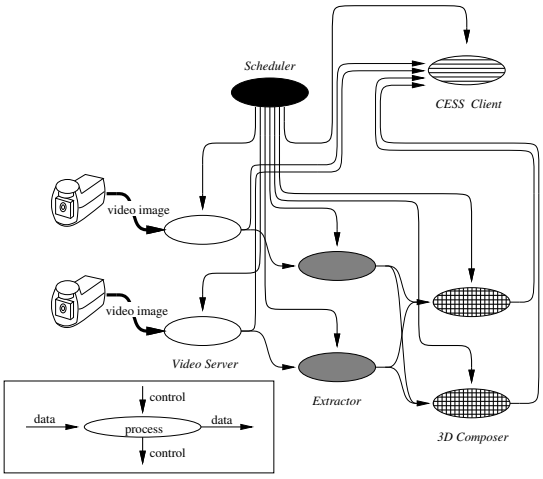

図 5: 三次元再構成システムの構成

実験では，ボクセルの大きさを一辺が 5cm の立方体とした．これは，本実験環境で講義室内を撮影した場合，1 画素が最大 5cm 平方の大きさに対応したからである．

5.1　空間分割による高速化

複数の 3D コンポーザを用いて，対象空間を空間分割する方法によるレイテンシ及びスループット向上について実験を行った．

講義室内の動物体は通常人間であるので，上下方向の運動よりも水平方向の運動が多いと考えられるため，対象空間 $\mathcal{W}$ は水平方向に分割した．

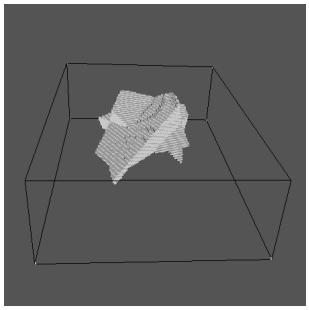

図 6: 観察空間 $\mathcal{X}$

なお，以下の対象空間 $\mathcal{W}$ を再構成する実験では，同一実験状況を保つために動物体として縦 55cm 横 55cm 奥行 25cm の箱を机の上に置いて行った．

まず，複数の 3D コンポーザを全て空間分割でデータ割り付ける方法を実験したところ，表 1に示すように，レイテンシ，スループットともに 3D コンポーザを増やすにつれて向上した．

表 1: 空間分割によるレイテンシとスループット

3D コンポーザの数	1	2	3	4
レイテンシ (msec)	1384	761	585	490
スループット (fps)	2.2	3.9	5.1	6.1

対象空間 $\mathcal{W}$ の分割位置変更の実験も同じ環境で行った．この実験では，3D コンポーザの数を 1 個から 4 個とし，1000 フレームの再構成を行った．実行フェーズにかかった時間の平均を図 7に示す．`dynamic` が動的に対象空間 $\mathcal{W}$ の再分割を行った場合である．`static` は観察空間 $\mathcal{X}$ と対象空間 $\mathcal{W}_i$ の積の体積が同じになるように対象空間 $\mathcal{W}$ を分割しておき，実験中は分割方法の変更を行わなかった場合を示す．分割方法の変更を行った方が 1 フレームあたり約 10 msec ，6% から 13% の高速化が達成されている．

5.2　時間分割による高速化

次に，時間分割の手法を用いて，複数の 3D コンポーザで異なるフレームの対象空間 $\mathcal{W}(t)$ を同時に再構成することによるスループットの向上について調べた．

空間分割の実験と同じ環境で実験を行った．1 個から 4 個の 3D コンポーザを用いたとき，実行

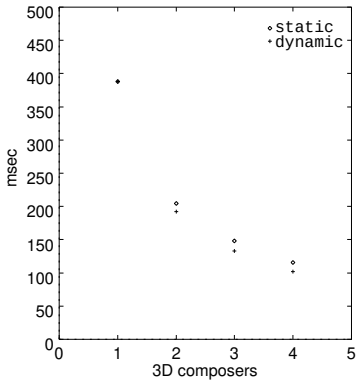

図 7: 分割方法の変更による実行フェーズにかかる時間の短縮

フェーズと通信フェーズにかかった時間を計測した．4 個の 3D コンポーザを用いた場合，レイテンシ は 730 msec であった．スループット は表 2に示す．

表 2: 3D コンポーザの数とスループット

3D コンポーザの数	1	2	3	4
スループット (fps)	2.2	4.3	6.3	7.3

5.3 時空間分割による高速化

最後に，時空間分割を用いて，複数の 3D コンポーザ群が異なるフレームの対象空間 $\mathcal{W}(t)$ を同時に再構成することによるスループットとレイテンシの向上について調べた．

実験は時間分割，空間分割による実験と同じ環境で行った．4 個の 3D コンポーザを用い，空間の分割数を 1, 2, 4 としたときのレイテンシとスループットを表 3に示す．時間分割の場合よりもレイテンシが短く，スループットは時間分割の場合とほぼ同じであった．

以上の結果から，高いスループットを要求する場合は時間分割がよく，短いレイテンシをを要求する場合は空間分割がよいことが言える．また，スループットとレイテンシの両立が要求される場合には，最高スループットと 0.1fps しか変わらないにも関わらずレイテンシが 560msec まで向上する $r = 2, s = 2$ の時空間分割がよいことがわかった．

表 3: 空間の分割数とスループット ・レイテンシ

空間の分割数 r	1	2	4
時間の分割数 s	4	2	1
レイテンシ (msec)	730	560	490
スループット (fps)	7.3	7.2	6.1

6 おわりに

本研究では，分散環境において高速に実空間をコンピュータ上に再構成する方法について提案した．実時間で実空間をコンピュータ上に再構成する上での最大の問題点は，データ量および処理量が膨大なため時間がかかることであった．そこで，ビデオ画像処理 ・空間の再構成の過程を分割し，3 種のプロセスで実行すると共に，これらのプロセスでパイプラインを構成することによりこの問題を解決し，高速な空間再構成を実現すると共に，実験によりスループットおよびレイテンシの向上を示した．

本方法では再構成の精度をあげると処理量と通信量が増大することから，今後は，通信の非同期化によるプロセス間通信の分散化などを検討する必要がある．

謝辞　本研究の一部は科学研究費基盤研究 B-09558034, 学術振興会未来開拓プロジェクト 「分散協調視覚」 (JSPS-RFTF 96P00501) の支援を受けている. ここに感謝の意を記す. また, 一部のデータ表示にアメリカ合衆国 Geometry Center の geomview パッケージを利用した.

参考文献

[1] Michitaka Hirose : Image-based virtual world generation, *IEEE MultiMedia*, Vol. 4, No. 1, pp. 27–33 (1997).

[2] Takeo Kanade, Peter Rander, and P. J. Narayanan : Virtualized reality: Constructing virtual worlds from real scenes, *IEEE MultiMedia*, Vol.4, No.1, pp. 34–47 (1997).

[3] Arun Katkere, Saied Moezzi, and Ramesh Jain : Global multi-perspective perception for autonomous mobile robots, *Workshop for Vision for Robots, 1995 IEEEE/RSJ Int. Conf. Intelligent Robots and Systems*, (1995).

[4] C. Tomasi and T. Kanade : Shape and motion from image streams under orthography: a factorization method, *Int. J. Computer Vision*, Vol.9, No.2, pp. 137–154 (1992).

[5] Saied Moezzi, Li-Cheng Tai, and Philipe Gerard : Virtual view generation for 3d digital video, *IEEE MultiMedia*, Vo.4, No.1, pp. 18–26 (1997).