# 時空間の分割とビデオ画像のパイプライン処理による高速三次元再構成

亀田能成[†] 太尾田健男[††,☆]
角所考[†] 美濃導彦[†]

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 (一秒あたり約 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．

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

Yoshinari Kameda,[†] Takeo Taoda,[††,☆] Koh Kakusho[†]
and Michihiko Minoh [†]

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 (about 700,000 voxels) per second, and the latency was 0.56 seconds.

## 1. はじめに

実空間をコンピュータ上に三次元世界として再構成すると，これに対してコンピュータグラフィックスの技術を用いて様々な加工を行うことができる．たとえば，実空間内の物体を仮想空間内で別の物体に置き換えて表示したり，現実のカメラでは撮影不可能な位置から撮影したかのような画像を作成したりすることが可能となる[5)8)]．このような技術を仮想空間におけるマンマシンインタラクションに応用していくためには，実時間で実空間をコンピュータ上に再構成することが必要となる．

実空間をコンピュータ上に再構成するために用いられる一般的な方法は，ビデオカメラを利用する方法である．これに関しては，ステレオ計測を用いる研究[4)]や，画像中に一人の人間が写っているという仮定の下で人間のモデルを当てはめる研究[1)3)]などがあるが，これらの方法は，一方向からの観測によって得られた画像 (群) のみを用いて空間を再構成するため，隠蔽には対処できない．

† 京都大学総合情報メディアセンター
Center for Information and Multimedia Stuies, Kyoto University
†† 京都大学大学院工学研究科
Graduate School of Engineering, Kyoto University
☆ 現在，NEC ソフトウェア中国
Presently with NEC Software Chugoku

この問題に対処できる方法として，複数の視点から撮影した画像を補間することによって新しい視点での画像を合成する方法[2]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[7]，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[5]などがある．しかし，このうち Hirose の手法[2]や Tomasi の手法[7]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[5]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[6][9]を利用し，対象となる時空間をボクセルで表現する．その際，再構成の対象となる空間が静物体と動物体から構成されているという前提のもとに，静物体のモデルを利用して空間の再構成における計算量を削減する．この処理の実現に際しては，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．

以降，2 章で視体積交差法とボクセルによる三次元再構成法の概要を説明する．3 章では，空間の再構成に視体積交差法を用いたときに可能となるビデオ画像処理の分割方法と，パイプラインの構成による高速三次元再構成方法について述べる．4 章では，時空間の分割によりさらに高速に空間が再構成できることを示す．5 章では，3 章と 4 章とで述べたシステムを実際に構築し，その能力を検証する．6 章では，本研究で明らかになったことをまとめ，今後の課題について述べる．

## 2. 視体積交差法による三次元再構成

本章では，視体積交差法による三次元再構成法について説明する．視体積交差法では，同一時刻に撮影された複数枚の画像を利用して物体の形状を求める．以下に，空間をボクセルで表現した場合に視体積交差法を適用する方法について述べる．

### 2.1 視体積交差法による三次元再構成

本節では，透視投影モデルを利用した空間のモデリングについて説明し，静物体のモデルを用いた視体積交差法について述べる．

#### 2.1.1 透視投影モデル

全てのカメラパラメータ，すなわちパン ・チルト ・ツイスト ・カメラの焦点位置及び焦点距離等の内部パラメータは固定で既知とする．

撮像系に透視投影を仮定すると，画像平面上の点 $p$ に投影され得る空間内の点 $\mathbf{x}$ は図 1 に示されるように (1) 式で表現される半直線をなす．

$$\mathbf{x} = \tau \mathbf{l}(p) + \mathbf{c}\ , \quad \tau > 0 \tag{1}$$

ここで $\tau$ は点 $\mathbf{x}$ への 焦点 $\mathbf{c}$ からの距離を示す．また，$\mathbf{l}(p)$ は焦点 $\mathbf{c}$ を通り画像平面上の点 $p$ へ向かう直線の方向ベクトルである．

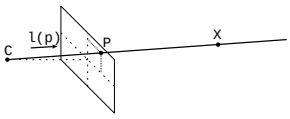



図 1 透視投影モデル
Fig. 1 Perspective Projection Model.

#### 2.1.2 動物体被視空間

本節では，空間のモデルについて説明し，次に，1 台のカメラによって撮影され得る空間 (視体積) と，動物体が撮影され得る部分空間 (動物体被視空間) とについて説明する．

再構成を行う空間内に存在する物体のうち，時間が経過しても位置や形状の変化しない物体 (静物体) は既知であるとし，あらかじめ計算機にそのモデルを持たせておく．すなわち，静物体として定義されていない物体 (動物体) の位置や形状を各時刻で求めることにより，空間が再構成されることになる．本稿では，このようにして再構成する空間のことを対象空間と呼び $\mathcal{W}$ で表す．また，対象空間に時間の概念を加えたものを 対象時空間 と呼び $\widehat{\mathcal{W}}$ で表す．

対象空間 $\mathcal{W}$ の部分空間で，カメラ $C$ の撮像範囲のことを視体積と呼び $\mathcal{V}$ で表す．視体積 $\mathcal{V}$ はカメラ $C$ のカメラパラメータにより決定される．カメラ $C$ により撮影された画像を $I$ とすると，(1) 式より，カメラ $C$ の視体積は

$$\mathcal{V} = \mathcal{W} \cap \mathcal{K} \tag{2}$$

ただし，

$$\mathcal{K} = \left\{ \mathbf{x} \mid \mathbf{x} = \tau\, \mathbf{l}(p) + \mathbf{c} \right. \\ \left. for \quad \forall p \in I\ ,\ \forall \tau > 0 \right\} \tag{3}$$

と表される．カメラ $C$ により撮影される動物体はカメラ $C$ の視体積 $\mathcal{V}$ の中に存在する．

静物体のモデルを利用することにより，カメラ $C$ に撮影される動物体が存在できる空間はさらに限定される．

動物体被視空間 $\mathcal{U}$ は視体積 $\mathcal{V}$ から静物体が占有する空間の集合 $\mathcal{S}$ と，静物体により隠蔽される空間の集合 $\mathcal{T}$ を除くことにより得られる．静物体の集合 $\mathcal{S}$ を視体積 $\mathcal{V}$ から除くことができるのは，動物体は静物体の中に入ることができないためである．静物体により隠蔽される空間の集合 $\mathcal{T}$ を除くのは，静物体により隠蔽された空間に存在する動物体は撮影できないためである．つまり，

$$\mathcal{U} = \mathcal{V} \cap \overline{\mathcal{S}} \cap \overline{\mathcal{T}} \tag{4}$$

である．

ここで，静物体内部の全ての点 $\mathbf{s} \in \mathcal{S}$ に対して，画像平面上に投影される点 $p$ とカメラ $C$ の焦点 $\mathbf{c}$ からの距離 $\tau$ の組が $(1)$ 式よりそれぞれただ一つ定まる．画像平面上の点 $p$ に投影される点 $\mathbf{s} \in \mathcal{S}$ のうち，焦点 $\mathbf{c}$ に最も近い点 $\mathbf{s}_{near}$ は，点 $p$ に投影される他の点 $\mathbf{s} \in \mathcal{S}$ と，これらの点により隠蔽される点 $\mathbf{t} \in \mathcal{T}$ を全て隠蔽する．この $\mathbf{s}_{near}$ と焦点 $\mathbf{c}$ との距離を $\tau_s(p)$ とおき，

$$\mathcal{L} = \mathcal{S} \cup \mathcal{T} \tag{5}$$

とすると，

$$\mathcal{L} = \left\{ \mathbf{x} \quad | \mathbf{x} = \tau_{\mathcal{L}}(p)\ \mathbf{l}(p) + \mathbf{c} \right. \tag{6}$$
$$\left. for \quad \forall \tau_{\mathcal{L}}(p) \geq \tau_s(p) \right\}$$

である．動物体被視空間 $\mathcal{U}$ は $(2)$ $(4)$ $(5)$ 式より

$$\mathcal{U} = \mathcal{W} \cap \mathcal{K} \cap \overline{\mathcal{L}} \tag{7}$$

である．これを図示すると図 2のようになる．ただし，図 2では簡単のため空間を二次元で表現している．全てのカメラパラメータと静物体のモデルは時間によって変化しないため，$\mathcal{U}$ も時間に対して不変である．

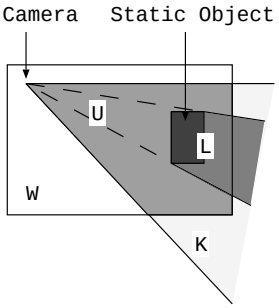



図 2 動物体被視空間
Fig. 2 Dynamic Object Observed Space.

### 2.1.3 視体積交差法

この節では，実空間中の動物体 $\mathcal{O}$ と，複数カメラで同一時刻に撮影された画像上において動物体 $\mathcal{O}$ が撮影された領域（動領域 $D$ ）との関係を示す．

カメラ $C_i$ で撮影された一枚の画像 $I_i$ 上の動領域を $D_i$ とすると，$D_i$ 内の点 $p_i$ は $(1)$ 式で表される直線上の点の像である．よって，カメラ $C_i$ の動物体被視空間が $\mathcal{U}_i$ であれば，動領域 $D_i$ は

$$\mathcal{D}_i = \mathcal{U}_i \ \cap \ \mathcal{E}_i \tag{8}$$

ただし

$$\mathcal{E}_i = \left\{ \mathbf{x} \quad | \mathbf{x} = \tau_i\ \mathbf{l}_i(p_i) + \mathbf{c}_i \right.$$
$$\left. for \quad \forall p_i \in D_i\ ,\ \forall \tau_i > 0 \right\} \tag{9}$$

で表される空間 $\mathcal{D}_i$ の写像である．ここで，$\mathbf{c}_i$ はカメラ $C_i$ の焦点の位置，$f_i$ は $C_i$ の焦点距離，$\mathbf{l}_i(p_i)$ は $C_i$ を通り点 $p_i$ へ向かう直線の方向ベクトル，$\tau_i$ は $C_i$ からの距離を示すパラメータ変数である．$\mathcal{D}_i$ は，カメラの焦点を頂点とする錐体となり，この錐体 $\mathcal{D}_i$ のことをカメラ $C_i$ の動物体存在可能空間と呼ぶ．この定義を図 3に表す．簡単のため，図 2と同様に，空間は二次元で表現されている．画像 $I_i$ 上で一つの動物体 $\mathcal{O}$ が撮影された動領域が $D_i$ であれば，動物体 $\mathcal{O}$ は錐体 $\mathcal{D}_i$ に内接している．

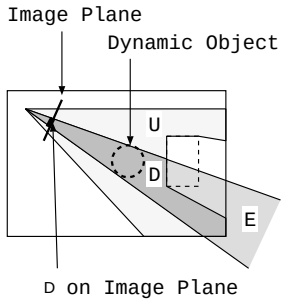



図 3 1つのカメラの動物体存在可能空間
Fig. 3 Exsistence Shadow Space of One Camera.

次に，$n_{cam}$ 台のカメラで同時に動物体 $\mathcal{O}$ を撮影した場合について考える．以下では同一時刻に撮影した画像群のことをフレームと呼び，フレームの番号を $t$ で表す．また，フレーム $t$ に含まれる画像のうち，カメラ $C_i$ $(i = 1, 2, \cdots, n_{cam})$ で撮影した画像を $I_i(t)$ で表す．

上の議論より画像 $I_i(t)$ 上で動物体 $\mathcal{O}$ が撮影された動領域 $D_i$ が得られれば，$\mathcal{O}$ は $(8)$ 式で表される錐体 $\mathcal{D}_i(t)$ に内接することが分かる．このことは $n_{cam}$ 台のカメラ全てについて同じことが言える．すなわち，動物体 $\mathcal{O}$ は $n_{cam}$ 個の動物体存在可能空間の積に内

接している．以下では全てのカメラの動物体存在可能空間の積のことを，単に動物体存在可能空間（CESS: Cumulative Existence Shadow Space）と呼び，$\mathcal{C}$ で表す．

$$\mathcal{C}(t) = \bigcap_{i=1}^{n_{cam}} \mathcal{D}_i(t) \qquad (10)$$

3 台のカメラの場合を摸式的に表すと図 4のようになる．

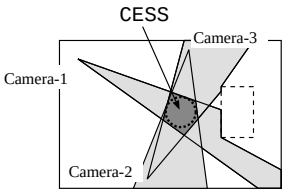



図 4　動物体存在可能空間
Fig. 4　Cumulative Exsistence Shadow Space.

以上のような方法で対象空間 $\mathcal{W}$ 中の動物体存在可能空間 $\mathcal{C}$ を求める方法が視体積交差法である．

なお，$n_{cam}$ 台のカメラによって同時に撮影可能な空間のことを観察空間と呼び $\mathcal{X}$ で示す．$n_{cam}$ を同時観測カメラ数と呼ぶ．CESS はこの観測空間 $\mathcal{X}$ の内部でのみ 計算可能となる．

## 3. 分散処理とパイプラインの構成による高速化

本章では，複数のビデオ画像を入力とし，対象空間 $\mathcal{W}$ を分散環境で高速に再構成する方法について述べる．

### 3.1 処理の分割

2章で説明した視体積交差法により対象空間 $\mathcal{W}$ を再構成するために，

(1) ビデオ画像の取り込み
(2) ビデオ画像から動領域の抽出
(3) 視体積交差法による対象空間 $\mathcal{W}$ の再構成

の 3 ステージに処理を分割する．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．ビデオ画像データは RGB 各 8bit で表現された画像データフォーマットで送出する．

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．

視体積交差法では，同時に撮影されたビデオ画像の組が必要になる，ビデオ画像間の同期をとる方法としては外部同期をビデオカメラに入力する方法が考えられるが，分散環境下での実時間画像取り込みでは画像取込ボードで一定時間内に必ず画像を取り込める保証が得られないので，本研究では，スケジューラによる撮影同期機構を実現する．

スケジューラは，全てのビデオサーバ $P_i^{cam}$ が取り込み可能状態（待ち状態）になると，$n_{cam}$ 台全てのビデオサーバに一斉にビデオ画像取り込み命令を発行する．ビデオ画像の取り込みが終了すると，ビデオサーバから取り込み完了がスケジューラに通知されるが，取り込み命令を発行した時刻から $t_{co}$ 以内に完了通知があったビデオサーバ $P_i^{cam}$ が撮影の同期に成功したものとする（図 5）．完了通知が $t_{co}$ 以内に届か

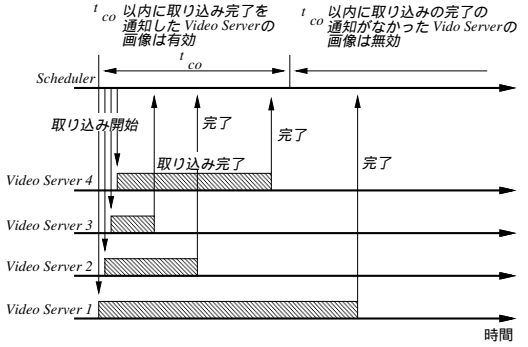



図 5　フレーム $t$ における画像の取得 ($n_{cam} = 4$)
Fig. 5　Image acquisition at frame $t$ ($n_{cam} = 4$).

なかったビデオサーバ $P_i^{cam}$ は同期に失敗したと考えるので，そのビデオ画像 $I_i(t)$ は対象空間 $\mathcal{W}(t)$ の再構成に使用されない．こうして得られた，時刻 $t$ からの短い時間 $t_{co}$ 以内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．動領域データは，二値画像のデータフォーマットで送出される．

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 $t$ の動領域 $D_i(t)$ を抽出した $n_{cam}$ 個のエクストラクタから動領域 $D_i(t)$ 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，動領域データの読み込み，対象空間 $\mathcal{W}(t)$ 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，(10) 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．

本章では 3D コンポーザを 1 つの処理単位とみなす．対象空間 $\mathcal{W}$ 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章で述べる．

### 3.5 パイプラインの構成

この節では，今まで述べてきた 3 つのプロセス，ビデオサーバ ·エクストラクタ ·3D コンポーザでパイプラインを構成する方法について述べる．

3.2節から 3.4 節で述べた 3 種のプロセス (ビデオサーバ ·エクストラクタ ·3D コンポーザ) 及び CESS クライアントは，それぞれ次の 3 種の処理を繰り返し行っている．

(1) データ読み込み
(2) データ処理
(3) データ送出

ここで，ビデオサーバ ·エクストラクタ間では RGB 各 8bit の画像データが転送され，エクストラクタ ·3D コンポーザ間では二値画像データが転送される．

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．これは，ネットワークインターフェースに対する入出力待ちが生じるためである．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．そこで，各プロセスは，通信を行うプロセスごとにデータの入力用のバッファと出力用のバッファを持つ．これにより入出力待ちを多重化する．

このようにして，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．

#### 3.5.1 パイプラインの構成による高速化

この節では，ビデオサーバ ·エクストラクタ ·3D コンポーザ ·CESS クライアントの 4 種のプロセスでパイプラインを構成する方法と，三次元再構成システムの処理能力について述べる．

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対象空間 $\mathcal{W}(t)$ 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，スループットが向上する．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 6に示す．

一般に $T_{cam}$ と $T_{ext}$ とは共にビデオレート $(30frames/sec)$ で実現出来るのに対し，$T_{cmp}$ は $T_{cam}, T_{ext}$ よりも長くなることがある．特に，ビデオサーバ ·エクストラクタ間のビデオ画像の転送にかかる時間を $T_{trans}$ としたとき，

$$T_{cmp} > T_{cam} + T_{trans} + T_{ext} \tag{11}$$

であれば図 7 のようなパイプラインの構成にした方が図 6 の構成よりもレイテンシが小さくなる．上式が成立するかどうかはシステムを構成する機器の能力に

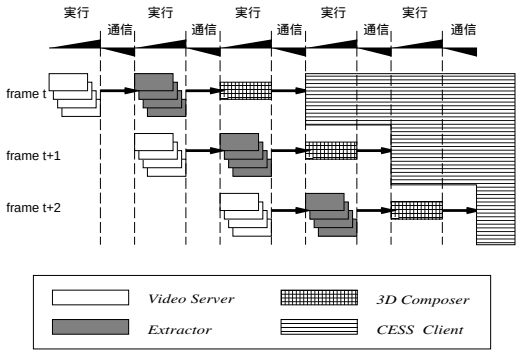



図 6 パイプラインの構成
Fig. 6 Pipeline structure.

よる．

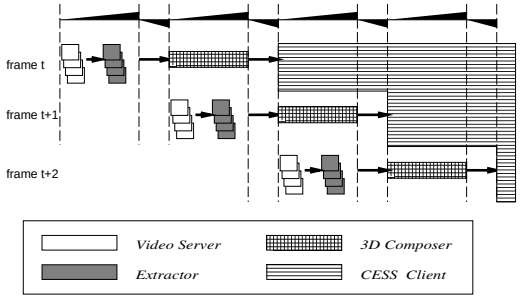



図 7 3D コンポーザ $\mathcal{W}$ の処理に時間がかかる場合のパイプラインの構成
Fig. 7 Pipeline structure in the case 3D composer takes much time.

## 4. 観察時空間の分割による高速化

2章で述べた空間再構成法では，動物体存在可能空間 $\mathcal{C}(t)$ を表す各ボクセル 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，3 章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．

本章では，対象時空間 $\widehat{\mathcal{W}}$ 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動物体存在可能空間の分布も変化する．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 $P^{cmp}_1$ が対象空間 $\mathcal{W}(t)$ を再構成している間に，別の $P^{cmp}_2$ は $\mathcal{W}(t+1)$ を，また別の $P^{cmp}_3$ 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，

$$n_{cmp} = rs \tag{12}$$

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

### 5.1 実 験 環 境

本研究で提案する手法により，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 8 に示す．

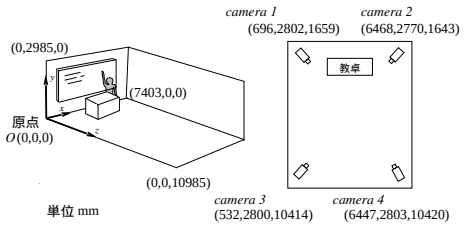



**図 8** 対象空間とカメラの配置 (単位 mm)
Fig. 8 Target space and camera position (unit:mm).

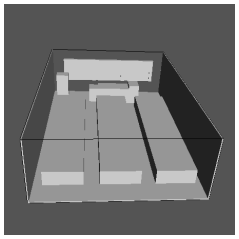


**図 9** 講義室内の静物体のモデル
Fig. 9 Static model of the lecture room.

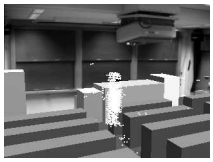
ビデオ画像をコンピュータに取り込む 4 個のビデオサーバは 4 台のコンピュータ上に配置する．この 4 台のコンピュータには SUN microsystems 社の Ultra2 Model 1200 (Solaris2.5.1) を使用した．エクストラクタもビデオサーバと同じコンピュータ上に配置した．3D コンポーザには，4 台の SUN microsystems 社の Ultra1 Model 170 (Solaris2.5) を使用した．これら 8 台のコンピュータは，データ通信用に ATM ネットワーク，同期制御通信用に 100BaseT イーサネットで接続されている．

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[10] 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 10右に示す．またこのときのプロセス間のデータの流れ・制御の様子を図 11 に示す．

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．同時観測カメラ数 $n_{cam}$ を 3 台とした場合の観察空間 $\mathcal{X}$ を図 12 に示す．このとき観察空間 $\mathcal{X}$ のボクセル数は 96,769 である．

ボクセルの大きさを小さくして空間分解能の向上を考える場合，我々の提案する手法では，3D コンポーザで行われる式 10中の演算回数はボクセル数に線形に比例する．

あるカメラからの視点



CESS データと実写の合成

**図 10** 再構成例
Fig. 10 An example of reconstruction.

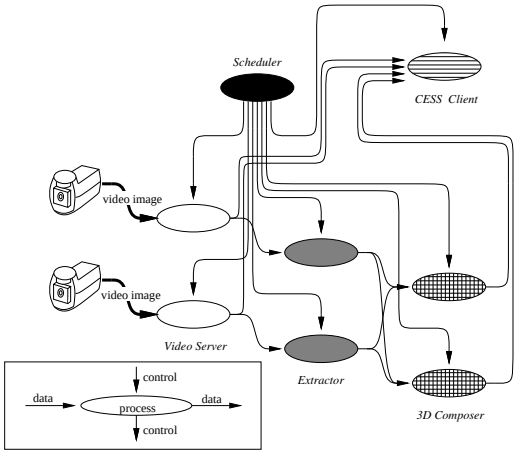



**図 11** 三次元再構成システムの構成
Fig. 11 3D reconstruction system diagram.

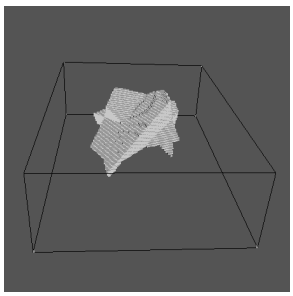


**図 12** 観察空間 $\mathcal{X}$
Fig. 12 Observed Space $\mathcal{X}$.

### 5.2 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．

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，レイテンシ (表 1)，スループット (表 2) ともに 3D コンポーザを増やすにつれて向上した．また同時に 3.5節に述べたパイプライン構成の実験をしたところ，3D コンポーザの数に関わらずレイテンシ (表 1) は 2 段のパイプライン構成のほうがよく，スループット (表 2) はほぼ同等であるという結果を得た．

表 1 パイプラインの段数とレイテンシ (単位 :msec)
Table 1 Latency with pipeline step number (unit:msec).

| 段数 \ 3D コンポーザの数 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 3 段 | 1384 | 761 | 585 | 490 |
| 2 段 | 912 | 502 | 378 | 322 |

表 2 パイプラインの段数とスループット (単位 :frames/sec)
Table 2 Throughput with pipeline step number (unit:frames/sec).

| 段数 \ 3D コンポーザの数 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 3 段 | 2.2 | 3.9 | 5.1 | 6.1 |
| 2 段 | 2.2 | 4.0 | 5.3 | 6.2 |

4.1節の対象空間 $\mathcal{W}$ 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 13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．

### 5.3 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．ただし，パイプラインの構成は 3 段である．1 個から 4 個の 3D コンポーザを用いたとき，実行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 3に示す．

### 5.4 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．

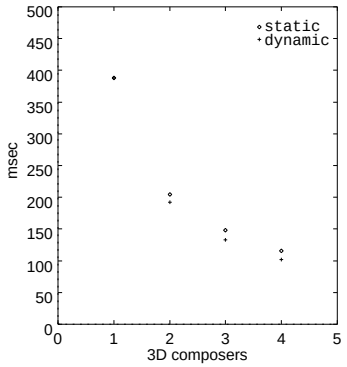



図 13 分割方法の変更による実行フェーズにかかる時間の短縮
Fig. 13 Reduction of execution phase time by changing the division method.

表 3 3D コンポーザの数とスループット
Table 3 Troughput with number of 3D composers.

| 3D コンポーザの数 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| スループット (frames/sec) | 2.2 | 4.3 | 6.3 | 7.3 |

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．ただし，パイプラインの構成は 3 段である．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 4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．

表 4 空間の分割数とスループット ・レイテンシ
Table 4 Troughput and latency with number of space division.

| 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 (frames/sec) | 7.3 | 7.2 | 6.1 |

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 0.1frames/sec 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．

## 6. おわりに

本論文では，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述べた．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る．本手法では，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高速な空間再構成を実現した．

静物体のモデルをあらかじめ計算機内に持たせておくと，動物体の位置や形状を求めることで空間が再構成される．本手法では，ボクセルで表現した空間に対し，視体積交差法を適用することにより，動物体の位置や形状を求めた．この方法では，同じ時刻の他のボクセルや，他の時刻のボクセルの値によらず，各ボクセルが動物体内に対応するかどうかの判定を行うことができる．分割された小時空間を異なるプロセスで同時に再構成することにより，スループットおよびレイテンシの向上が達成された．

この手法に基づいた実験を行い，空間の再構成が高速に行われることを確認した．

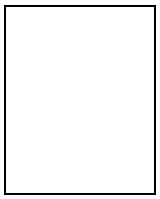
今後の課題としては，再構成の精度をあげると処理量と通信量が増大することから，非同期化によるプロセス間通信の分散化などを検討する必要がある．また，ピラミッド法[11]の応用によりさらなる高速化が実現できると共に，動物体の表面付近のみ精度を上げて再構成することが可能と考えられ，これに関する検討も課題の１つである．

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.

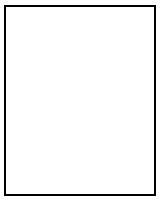
## 参 考 文 献

1) D.M. Gavrila and L.S. Davis: 3-d model-based tracking of human upper body movement : a multiview approach, *ISCV '95*, pp. 253 – 258 (1995).
2) Michitaka Hirose : Image-based virtual world generation, *IEEE MultiMedia*, Vol. 4, No. 1, pp. 27–33 (1997).
3) Yoshinari Kameda and Michihiko Minoh : A human motion estimation method using 3-successive video frames, *Proc. of Int. Conf. on Virtual System and Multimedia(VSMM) '96*, pp. 125–140 (1996).
4) Takeo Kanade, Hiroshi Kano, and Shigeru Kimura : Development of a video-rate stereo machine, *Proc. 1995 IEEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol.3, pp. 95–100 (1995).
5) 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).
6) 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).
7) 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).
8) Patrick Kelly, Ramesh Jain, etal.: An architecture for multiple perspective interactive video, *Technical report VCL-95-103*, Visual Computing Laboratory, University of California (1995).
9) 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).
10) W. Richard Stevens : UNIX ネットワークプログラミング, トッパン (1992).
11) 長尾 真 : 画像認識論, コロナ社, 1983.

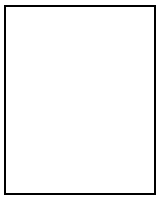
(平成 10 年 5 月 11 日受付)
(平成 10 年 11 月 9 日採録)



亀田 能成 (正会員)
平成 8 年京都大学大学院工学研究科情報工学専攻博士後期課程単位取得退学. 平成 10 年より京都大学総合情報メディアセンター助手. 画像処理，人体を対象とするモデルベースドビジョン，分散協調視覚，及びマルチメディアの研究に従事．

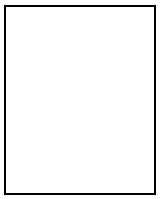


太尾田健男
平成 8 年京都大学大学院工学研究科情報工学専攻修士課程修了. 平成 10 年より NEC ソフトウェア中国. 在学中，分散協調視覚の研究に従事．



角所　考 (正会員)

昭和 63 年名古屋大学工学部電気学科卒業．平成 5 年大阪大学大学院工学研究科通信工学専攻博士課程修了．博士 (工学)．平成 4~6 年日本学術振興会特別研究員．平成 5~6 年スタンフォード大学ロボティクス研究所客員研究員．平成 6 年大阪大学産業科学研究所助手．平成 9 年より京都大学総合情報メディアセンター助教授．視覚メディア理解，マンマシンインタラクションに関する研究に従事．IEEE，ACM，電子情報通信学会，人工知能学会各会員．



美濃　導彦 (正会員)

昭和 53 年京都大学工学部情報工学科卒業．昭和 58 年同大大学院博士課程修了．同年工学部助手，昭和 62 年~63 年マサチューセッツ州立大学客員研究員，平成元年京都大学工学部附属高度情報開発実験施設助教授，平成 7 年同教授，平成 9 年京都大学総合情報メディアセンター教授．画像処理，人工知能，知的コミュニケーション関係の研究に従事．工博，ＩＥＥＥ，ＡＣＭ，情報処理学会，電子情報通信学会，画像電子学会，日本ロボット学会各会員．