

NAIST-IS-MT9551102

修士論文

ステレオ視による奥行き推定に基づく  
実画像と仮想物体の合成

松原 重徳

1996年2月16日

奈良先端科学技術大学院大学  
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に  
修士(工学) 授与の要件として提出した修士論文である。

松原 重徳

指導教官： 横矢 直和 教授  
千原 國宏 教授  
竹村 治雄 助教授

# ステレオ視による奥行き推定に基づく 実画像と仮想物体の合成\*

松原 重徳

## 内容梗概

近年、実空間中に仮想物体を表示することによって人間の環境知覚機能を増強する技術であるオーグメンテッドリアリティ (Augmented Reality) へのニーズが高まってきている。現在、実画像に仮想物体を合成する手法としてクロマキー合成などが用いられているが、これらは仮想物体を実画像上に単に上描きする方法であるため、仮想物体が最も手前に存在する画像しか合成できないという問題がある。オーグメンテッドリアリティでは、仮想物体を3次元空間中の任意の位置に表示する必要があり、奥行きに応じて物体同士の隠蔽関係を正しく表現できなければならない。

本論文では、両眼ステレオ画像の視差から実空間の奥行きを推定し、奥行きに応じた隠蔽関係で仮想物体を合成することを目的とする。物体を2次元で表現することにより物体同士の交錯を回避する、画像合成に適した高速な奥行き推定アルゴリズムを提案する。さらに、領域単位で動きを検出する領域追跡アルゴリズムにより、高速かつ自然な動画像合成を行なう手法を示す。最後に、実際にステレオ動画像に仮想物体を合成し、本手法の有効性を検証する。

## キーワード

オーグメンテッドリアリティ、画像合成、両眼ステレオ視、オクルージョン検出、動き検出

---

\*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT9551102, 1996年2月16日.

# **Fusion of Virtual Objects and Real World Images Based on Stereo Depth Estimation\***

Shigenori Matsubara

## **Abstract**

Recently, the technology to enhance human perception by displaying virtual objects in real images, which is called Augmented Reality, is in demand. Chroma-key method is usually applied to an image composition today, which simply overlays virtual objects upon real images. However, overlays are not effective when displaying virtual objects in 3D, since occlusion between real and virtual objects is not addressed.

The purpose of this paper is to estimate the depth of real-world with a pair of stereo video images, and generate images which fuses virtual objects and real world images. We first propose a fast depth estimation algorithm optimized to image composition which can avoid collision of real and virtual objects by expressing them in 2D form. Then we propose a region tracking algorithm which is fast and natural way for moving pictures. Finally, we show experimental results to verify the proposed method.

## **Keywords:**

Augmented reality, Image composition, Binocular stereo, Occlusion detection, Motion estimation

---

\*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9551102, February 16, 1996.

# 目次

1. はじめに	1
2. オーグメンテッドリアリティと画像合成	3
2.1 画像合成の種類と応用分野	3
2.2 従来の画像合成技術	6
2.2.1 クロマキー合成	6
2.2.2 Wloka らによる画像合成手法	7
2.3 本研究の方針	7
2.3.1 動作環境と利用可能な情報	7
2.3.2 画像合成への最適化	8
3. 隠蔽関係を満たした動画像合成手法	11
3.1 奥行き推定アルゴリズム	12
3.1.1 視差候補の選出	13
3.1.2 特徴点における視差推定	14
3.1.3 視差補間	16
3.2 領域追跡アルゴリズム	20
3.2.1 同一視差領域の動き推定	20
3.2.2 領域修正	24
3.3 合成画像生成アルゴリズム	25
4. 評価実験	26
4.1 シミュレーション環境	26
4.2 実験データ	26
4.2.1 入力実画像	26
4.2.2 仮想物体画像	27
4.3 実験結果	27
4.3.1 奥行き推定の結果	27
4.3.2 物体追跡の結果	29

4.4 處理時間 . . . . .	29
<b>5. 結論</b>	<b>42</b>
謝辭	44
參考文獻	45

## 目 次

1	画像合成の種類と応用分野 . . . . .	3
2	景観シミュレーションのシステム構成 . . . . .	5
3	物体の3次元表現と2次元表現 . . . . .	9
4	物体の交錯とその回避 . . . . .	10
5	全体の処理の流れ . . . . .	11
6	標準ステレオ撮像系 . . . . .	12
7	奥行き推定アルゴリズムの流れ . . . . .	13
8	ブロックマッチングにおけるブロック形状 . . . . .	16
9	視差データの $(x_L, x_R)$ 平面へのプロット . . . . .	17
10	$(x_L, x_R)$ 平面から視差マップへの写像 . . . . .	18
11	視差の補間 . . . . .	19
12	領域追跡アルゴリズムの流れ . . . . .	20
13	輝度変化を用いた領域修正 . . . . .	24
14	合成画像生成過程 . . . . .	25
15	仮想物体画像 . . . . .	27
16	奥行き推定アルゴリズムにおけるエッジ位置の相関関数 . . . . .	28
17	入力画像 . . . . .	31
18	奥行き推定アルゴリズムによる特徴点における視差推定結果 . . . . .	32
19	奥行き推定アルゴリズムによる視差補間結果 . . . . .	33
20	奥行き推定アルゴリズムを用いた合成画像 (例1) . . . . .	34
21	奥行き推定アルゴリズムを用いた合成画像 (例2) . . . . .	35
22	領域追跡アルゴリズムによる動き推定と領域修正の結果 (1/3) . . . . .	36
23	領域追跡アルゴリズムによる動き推定と領域修正の結果 (2/3) . . . . .	37
24	領域追跡アルゴリズムによる動き推定と領域修正の結果 (3/3) . . . . .	38
25	領域追跡アルゴリズムを用いた合成画像 (1/3) . . . . .	39
26	領域追跡アルゴリズムを用いた合成画像 (2/3) . . . . .	40
27	領域追跡アルゴリズムを用いた合成画像 (3/3) . . . . .	41

## 表 目 次

1	2つの特徴点と、その間を補間するべき視差 . . . . .	19
2	シミュレーション環境 . . . . .	26
3	60 フレームの処理時間 [sec] . . . . .	30

# 1. はじめに

近年、実空間中に仮想物体を表示することによって、人間の環境知覚機能を増強する技術であるオーグメンテッドリアリティ (Augmented Reality. 以下、AR) へのニーズが高まってきている。例えば、現在実用化されているバーチャルリアリティ (Virtual Reality. 以下、VR) による景観シミュレーションでは、実空間をCGで描くために、実空間の3次元情報をコンピュータに入力するという非常に複雑な作業が伴う上、出力画像も臨場感に欠けるという問題点がある [1][2]。これは、実空間を精密にモデル化することが困難であることに起因する。そこで、実空間画像をそのまま取り込み、そこに仮想物体を合成するARが求められている。

現在、放送分野などにおいて、実画像に仮想物体を合成する手法として、クロマキー合成やマット合成などが用いられている [3]。これらは、仮想物体を実画像上に単に上描きする方法であるため、仮想物体が最も手前に存在する画像しか合成できないという問題がある。景観シミュレーションでは、仮想物体を3次元空間中の任意の位置に表示する必要があり、奥行きに応じて物体同士の隠蔽関係を正しく表現できなければならない。近年の映画やCFでは、このような要件を満たした、自然な画像を実現しているものもあるが、これらの技術のほとんどは、特化した専門家による根気強い手作業によって創造されたいわば芸術作品であり、莫大なコストと時間がかかっている [3]。

従来、ARの分野では、頭部搭載型表示装置 (Head Mounted Display. 以下、HMD) などの3次元表示装置を用い、文字や図形などの仮想物体を実空間中に表示する研究が行なわれてきた [5]。遠隔地のロボットから送られてくる実画像中に、位置を指定するポインタなどの図形を表示しながら、ロボットに作業指示を与えるという遠隔作業支援の研究も行なわれている [6][7]。また、MRI/CT<sup>1</sup> などから得られた臓器の3次元画像を、手術中の患者のビデオ画像上に重畳してHMDに表示することにより、患部の3次元位置を確認しながら手術を行なうという研究も行なわれている [8]。これらの研究はいずれも、仮想物体を3次元空間中の任意の奥行きに表示するものであるが、ここでは実物体も仮想物体も共に半

<sup>1</sup>Magnetic Resonance Imaging / Computed Tomography :磁気共鳴を利用した断層写真技術。

透明であり、同一位置に2つの物体が存在することを許しており、隠蔽関係を満足する表示方法ではない。景観シミュレーションなどでは、不透明な物体を奥行きに応じた隠蔽関係で合成する必要がある。

本研究では、両眼ステレオ画像の視差から実空間の奥行きを推定し、奥行きに応じた隠蔽関係で、3次元空間中の任意の位置に仮想物体が存在するような画像を合成することを目的とする [10]。コンピュータビジョンにおけるステレオ視では、実物体の精密な3次元形状を知ることが重要であるが、画像合成におけるステレオ視では、実物体と仮想物体のどちらが手前にあるのかという物体の相対的位置関係が重要であり、物体表面の3次元形状は重要ではない。また、動画像として自然に見える必要があるため、画像の質もさることながら処理速度も重要である。本研究ではこれらの点を考慮し、画像合成に適した奥行き推定アルゴリズムを提案する。さらに、領域単位で動きを検出する領域追跡アルゴリズムにより、高速かつ自然な動画像合成を行なう手法を提案する。

以下、2章ではARと画像合成の従来技術と本研究の方針について述べ、3章では本研究で提案する動画像合成アルゴリズムについて、4章では本手法を用いてステレオ動画像に仮想物体を合成した実験の結果を、最後に5章で結論を述べる。

## 2. オーグメンテッドリアリティと画像合成

### 2.1 画像合成の種類と応用分野

映像の世界では、実写では撮ることのできない画像へのニーズが多い。ここでは、2つの画像を合成することにより、実際には存在しない映像を創り出すことを考える。画像のソースとしては、CGにより得られる仮想物体（仮想空間）と、実写により得られる実物体（実空間）の2種類がある。仮想物体は人間が定義したものであるため、全ての3次元データは既知であるのに対し、実物体は未知であるという点で大きな違いがある。一方、画像合成時にはどちらの画像をベースにするか、すなわちどちらを背景とし、どちらを物体（前景）とするかにより、場合分けが生じる。この場合分けとそれらの合成技術の用途を図1に示す。

	<背景>	<物体>	<用途>							
(A)	仮想	仮想	仮想現実感	映画・CF製作	景観シミュレーション	組み立て作業支援	遠隔作業支援	Informed Consent	手術シミュレーション	ゲーム・アニメーション
(B)	現実	仮想								
(C)	現実	現実	バーチャルスタジオ							
(D)	仮想	現実								

図1 画像合成の種類と応用分野

(A)では、全ての3次元情報は既知であり、通常のVRはこの部類に入る。(B)では、背景（実空間）の3次元情報は未知であるため、仮想物体を実空間中の任意の場所に正しい隠蔽関係で配置するには、実空間の3次元情報を推定する必要がある。(C)では、背景だけでなく、物体の3次元情報も推定する必要がある。(D)では、物体の3次元情報の推定が必要である。(B),(C)のように、実空間中には実在しないものを表示することによって、人間の環境知覚機能を増強する技

術をオーグメンテッドリアリティ (AR) と呼ぶ。AR では実物体 (実空間) の3次元情報は未知であり、推定が必要であるという点で、VR より問題は難しいといえる。以下に、図1中の語句の説明を示す。

- 仮想現実感

実際には存在しない架空の空間をCGによって生成し、実際の空間のように疑似体験するもの。各種シミュレーションやアミューズメントなどに利用されている。リアルなCG画像を生成することが困難であり、臨場感に欠けるという問題点がある。

- バーチャルスタジオ

電子セットとも呼ばれ、テレビ番組の製作などにおいて従来背景として用いていた大道具の代わりに、仮想の画像や実写の画像を用いるもの [4]。実際の道具と違い、かさばらずに保存ができる。スタジオセット画像のライブラリ化により大道具製作やロケが不要となり、製作費の削減が可能である。特にサイエンス番組などでは、実際には目に見えない事象を説明する際に説得力のある演出が可能となる。

- 映画・CF 製作

人目を引き付ける映像を製作するためには、あらゆるアイデアを映像化する必要がある。実写では撮れない映像を創るニーズが多い。

- 景観シミュレーション

空間中に建築予定の建築物を表示することにより、建築前に建築後の景観を評価するのに役立つ。

- 組み立て作業支援

機械の組み立て現場などにおいて、HMDを通して実空間中に作業指示などを提示することにより、作業ミスの減少や効率化を図る。

- 遠隔作業支援

実際に人間が行けない空間 (深海、原子炉、宇宙空間など) にロボットを送

り、そこから送られてくる画像を用いて作業指示を与える [6][7]。ステレオ画像中にマーカーなどを表示し、3次元位置を指定することが可能である。

- Informed Consent

医療において、医者が患者に病状をわかりやすく説明するのに用いる。治療方法に関する患者自身の意志を尊重する上で、患者に正しい判断材料となる情報を与えるのに役立つ。

- 手術シミュレーション

MRI/CT などから得られた臓器の3次元画像を、手術中の患者のビデオ画像上に重畳してHMDに表示することにより、患部の3次元位置を確認しながら手術を行なう [8]。手術の仮想的なトレーニングをすることにも役立つ。

- ゲーム・アミューズメント

HMDを用いた、様々な没入型アトラクションに応用できる。

本研究では、図1-(B),(C)の分野、特に景観シミュレーションやバーチャルスタジオへの応用を想定している。景観シミュレーションを例にあげた場合のシステム構成を図2に示す。ここでは、カメラも実空間中の物体も移動可能である。また、仮想物体も別に動きを定義することにより、移動可能である。このようなシステムにより、ステレオカメラ搭載型HMDを装着した人間は、実空間中を自由に移動しながら、必要な3次元位置に仮想物体を見ることが可能となる。

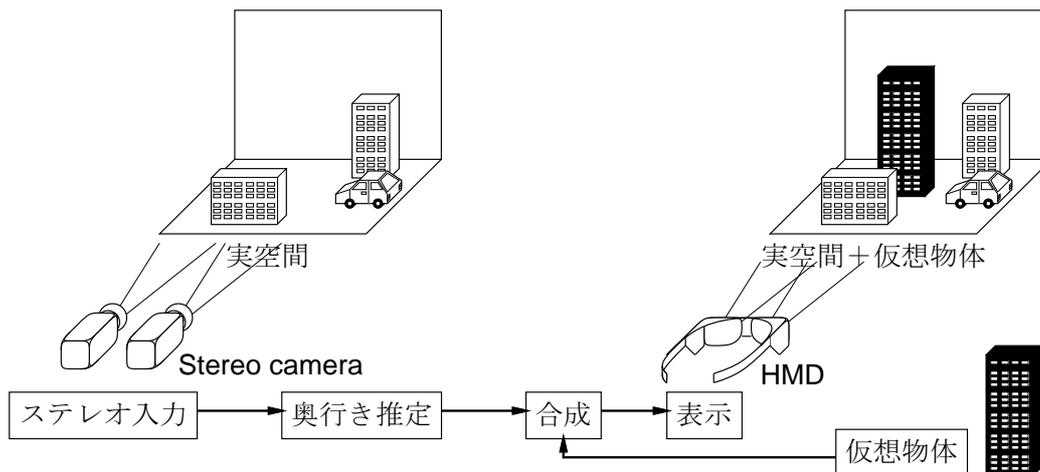


図2 景観シミュレーションのシステム構成

## 2.2 従来の画像合成技術

### 2.2.1 クロマキー合成

現在用いられている画像合成技術の1つに、クロマキー合成がある。クロマキー合成とは、クロマキーブルーと呼ばれる特殊な背景色をバックに物体を撮影し、得られた画像中のクロマキーブルーの部分だけを別画像に置き換えるという画像合成技術である。これにより、背景として任意のシーンを合成することが可能となり、スタジオにいる出演者が、あたかも別の場所にいるかのような画像を生成することができる。この手法は、現状では数少ないリアルタイムの画像合成技術であり、電子セットやバーチャルスタジオへの応用が考えられている。しかし、以下のような制約条件から、クロマキー合成だけではバーチャルスタジオとしての十分な機能を期待できないのが現状である。

#### (A) 前景と背景の位置関係が一意

物体が最も手前に存在するような画像しか生成できず、例えば、出演者が背景中を行き交う車の向う側に立っているというような画像は合成できない。

#### (B) カメラワークの禁止

被写体と背景が合理的な位置関係を保つためには、パン、チルト、ズームなどのカメラワークは許されない。

#### (C) 撮影環境が限定

クロマキーブルーの特殊な背景が必要であり、照明も一様でなければならないなど、撮影上の制約が多い。また、被写体内にクロマキーブルーと同色の物体があってはいけない。

動的環境において、空間中の任意の位置に仮想物体を表示する必要のあるARでは、特に(A), (B)は大きな問題となる。そこで、物体の位置に応じた隠蔽関係で仮想物体を合成する技術が期待されている。

## 2.2.2 Wloka らによる画像合成手法

動的環境下で、空間中の任意の位置に仮想物体を表示するためには、実空間の奥行きを推定して実物体と仮想物体の前後関係を判定し、それに応じた隠蔽関係で画像を生成する必要がある。Wloka らは、両眼ステレオ画像の視差から実空間の奥行きを推定するアルゴリズムを高速化し、仮想物体の合成に適用した [9]。しかしこのアルゴリズムは、一般的にコンピュータビジョンで使われているステレオアルゴリズムを高速化したものであり、いわゆる画像合成用に最適化されたものではない。すなわち、本アルゴリズムでは物体を 3次元で扱うため、仮想物体が実物体の近傍に存在する場合は、物体同士の衝突が起こることがある。したがって、本手法による合成では、実物体と仮想物体が交錯した画像が生成されるなど、合成画像が不自然に見えるという問題点があった。

## 2.3 本研究の方針

### 2.3.1 動作環境と利用可能な情報

AR では、カメラも実空間中の物体も、共に動的であるという条件が重要である。この条件により、カメラ搭載型 HMD を装着した人間は、実空間中を自由に移動しながら、刻々と変わる外界の状態にマッチした仮想物体を HMD を通して見ることができる。

本節では、このような環境下で実際に得ることができる情報にはどのようなものがあるかを考える。人間の奥行き知覚は、両眼視差、オクルージョン、運動視差、ピント調節、輻輳、大きさ、テクスチャ勾配、陰影などの情報をもとに推定されている [11][12]。この中で、本研究ではどの情報が利用可能であるかを考える。

- 陰影

動作環境としては、一般の生活空間を考えているため、レーザーレンジファインダなど、環境に影響を与える 能動型ステレオ [13] はここでは適さない。あくまでも環境に影響を与えず、受動的に得られる情報のみを用いて奥行きを推定する 受動型ステレオ [13] が理想的である。これにより、陰影の情報は利用できない。

- 運動視差

ハードウェア簡略化のため、カメラの位置や方向を調べる装置はここでは用いない。これにより、カメラの動きと画像の動きから奥行きを推定する手法である運動視差の情報は利用できない。

- ピント調節、輻輳

ハードウェア簡略化のため、カメラのピントや輻輳角度を調節する装置はここでは用いない。これにより、ピント調節、輻輳の情報は利用できない。

- 大きさ、テクスチャ勾配

物体の大きさなどに関する先見的知識は、ここでは用いない。先見的知識の内容によっては、一般性が失われる可能性があるからである。これを正しく用いるためには、人間と同レベルの認識システムと知識が必要となり、現在の技術では不可能である。これにより、先見的知識が必要である大きさ、テクスチャ勾配の情報は利用できない。

最終的に、本研究で利用できる情報は、両眼視差、オクルージョンのみということになる。本システムでは両眼ステレオ画像を入力とし、左右画像間の視差から奥行きを推定することを考える。

### 2.3.2 画像合成への最適化

仮想物体と実物体の衝突検出も行なえるような、コンピュータビジョン的なアプローチによる AR システムを構築するためには、実空間の精密な 3 次元構造を調べ、コンピュータ内部に実空間と同じ空間を仮想的に持つ必要がある。しかし、2 次元の濃淡画像から 3 次元空間を推定するという問題は不良設定問題であり、何らかの拘束条件がないと解くことはできない [14]。ここで導入する拘束条件によっては、一般性が失われるという危険もある。また、受動型ステレオから得られる情報は比較的少ないことから、精密な 3 次元情報を推定することは事実上困難である。さらに、処理速度の面でも大きな問題がある。衝突検出などを行なわない簡易型の AR システムなら、すなわち仮想物体の画像を合成することだけが目的ならば、3 次元の仮想空間をコンピュータ内部に持たずに、2 次元的な処理

でも実現できる。扱うデータの次元は、計算コストに大きく影響するため、高速性を考えるならば次元は低い方が有利である。

コンピュータビジョンにおけるステレオ視では、物体表面の絶対距離、すなわち物体の3次元形状を高精度に知ることが重要である。一方、2つの物体の隠蔽関係は、どちらの物体が手前にあるのかという相対的位置関係によって決定されるため、画像合成におけるステレオ視では、物体の相対的位置関係が重要である。すなわち、物体表面の3次元構造などは重要ではなく、画像上での遮蔽輪郭に意味があり、遮蔽輪郭で囲まれた領域の前後関係がわかることが重要である。画像平面は2次元であることから、本研究では、画像中の各物体を2次元で表現する。各物体は、視点から見て垂直な平面構造をもち、その平面の $z$ 座標で物体の奥行きを記述する(図3)。すなわち、同一物体内の全ての点は同じ奥行きを持つと仮定する。

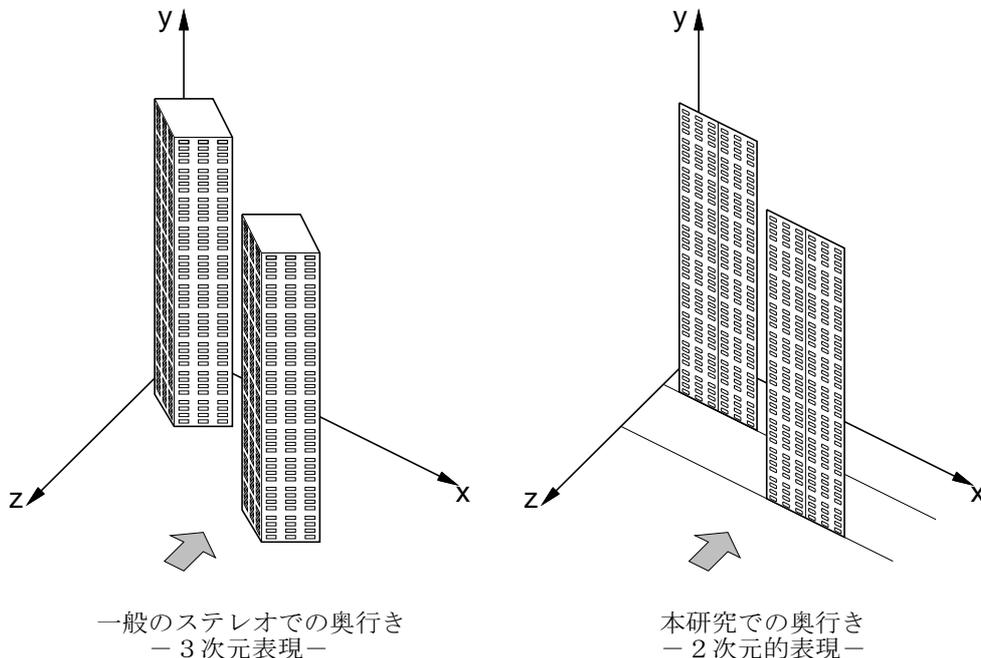


図3 物体の3次元表現と2次元表現

物体を2次元表現することの長所は以下のとおりである。

- 処理の高速化

奥行きが離散的であるため、奥行き推定処理では、限られた数の奥行きの中から最適なものを選べば良いため、処理を高速化できる。

- 実物体と仮想物体の交錯チェックの回避

3次元空間中に3次元物体を配置する場合は、物体同士の衝突の問題が起こる可能性がある(図4-(c))。しかし、3次元空間中に2次元物体を配置するのであれば、衝突は簡単に回避できる。2次元表現により物体の奥行き方向の精度は落ちるが、実物体と仮想物体が交錯するという物理的に起こり得ない画像を生成することは避けられる(図4-(d))。

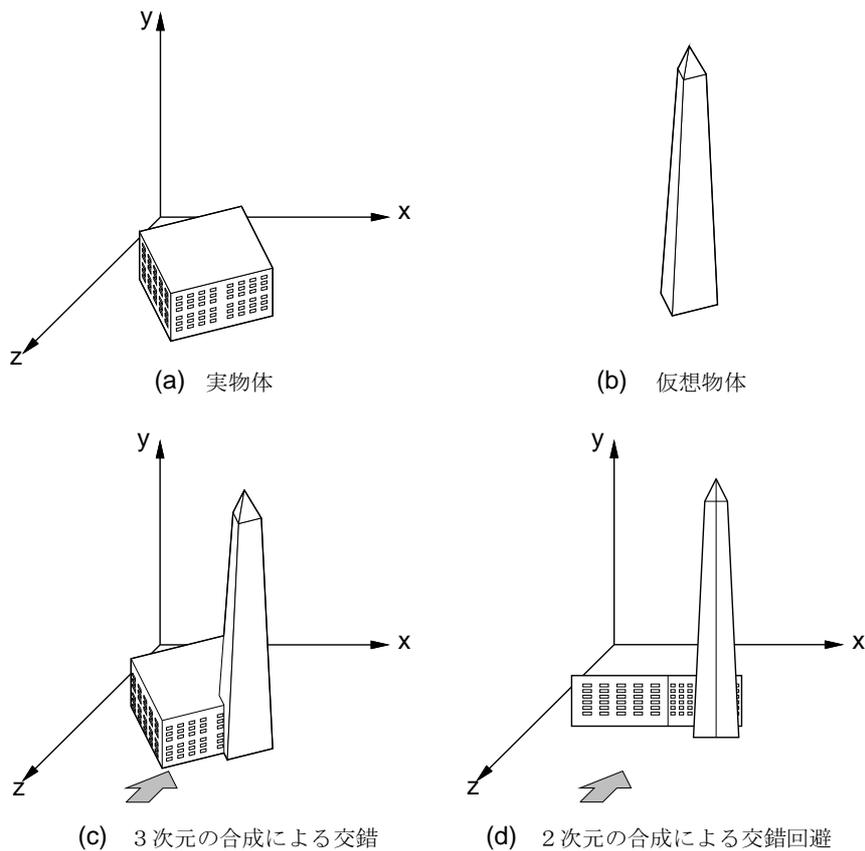


図4 物体の交錯とその回避

### 3. 隠蔽関係を満たした動画像合成手法

本研究では、カメラもシーン内の物体も共に移動する動画像を対象としている。奥行きに応じた隠蔽関係で動画像合成を行なうには、各フレームにおいて視差マップ（画像中の全点における視差）を求め、それに応じて画像を合成する必要がある。受動型ステレオでは、ステレオ画像の視差から奥行きを求めるが、動画像シーケンスでは時間軸方向に強い相関があるため、時間軸方向につながりのある処理をするのが効果的と考えられる。ここでは図5に示すように、最初に視差を求めて奥行き推定を行ない、以後は同一視差領域を追跡することにより、距離マップの更新を行なう。この処理をある周期（フレーム）で行なう。領域追跡の処理は奥行き推定の処理よりも高速であり、時間軸相関も利用できるため、高速かつ自然な動画像の合成が可能となる。次節以降では、奥行き推定アルゴリズム、領域追跡アルゴリズムのそれぞれの処理について述べる。

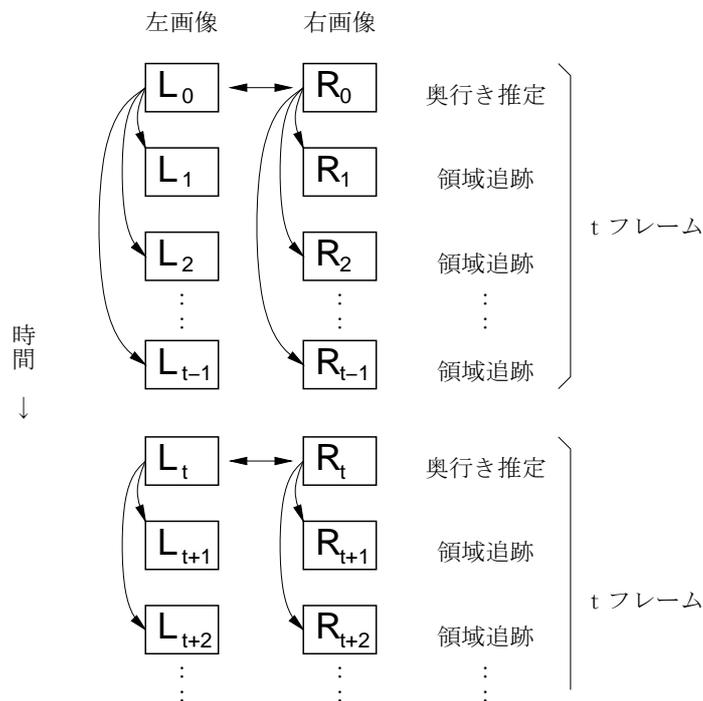


図5 全体の処理の流れ

### 3.1 奥行き推定アルゴリズム

まず、本研究で用いるステレオ撮像系を図6に示す。左右等価なカメラを、光軸平行かつ水平に設置する。カメラの焦点距離を  $f$ 、カメラの基線長を  $b$  とする。空間中の点  $P(X, Y, Z)$  が左右画像上の座標  $(x_L, y_L), (y_R, y_R)$  にそれぞれ現れたとすると、エピポーラ拘束により

$$y_L = y_R \quad (1)$$

が成立し、点  $P(X, Y, Z)$  の座標はそれぞれ次のように表すことができる [15]。

$$X = \frac{b \cdot (x_L + x_R) / 2}{x_L - x_R} \quad (2)$$

$$Y = \frac{b \cdot y_L}{x_L - x_R} \quad (3)$$

$$Z = \frac{b \cdot f}{x_L - x_R} \quad (4)$$

式(4)より、 $b, f$  が固定であれば奥行きは視差  $d \equiv x_L - x_R$  のみに依存することがわかる。すなわち、奥行き推定は視差推定と等価である。

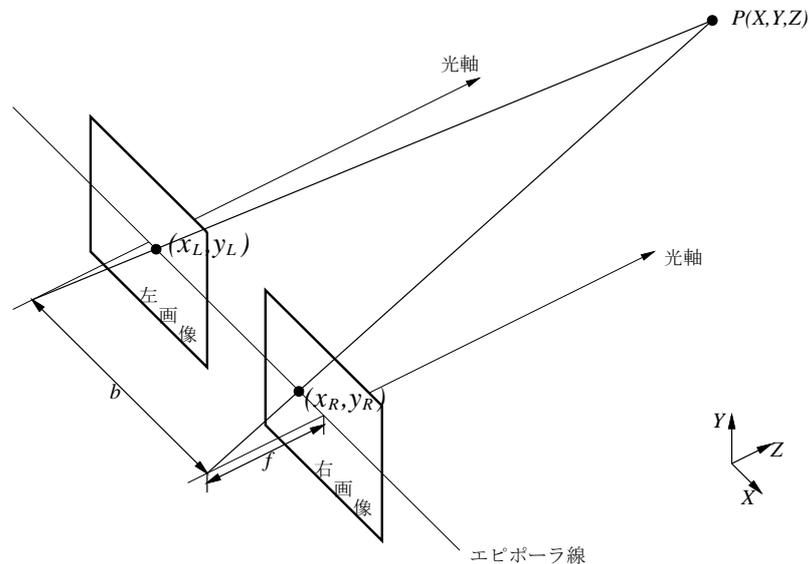


図6 標準ステレオ撮像系

視差推定では、左右画像間の対応点をいかに正しく求めるかが重要である。視差は横方向のずれであるので、横方向に輝度変化がある点（特徴点）でなければ、正しく対応点を求めることはできない。本節では、このことを考慮しながら、視差マップを求める手法を示す。

奥行き推定アルゴリズムの流れを図7に示す。まず始めに、画像内の支配的な視差の候補  $\{d_n\}$  を選出する。これは、後の処理で  $\{d_n\}$  についてのみ注目することにより処理を削減し、全体としての処理を高速化するためのものである。次に、特徴点についてのみ  $\{d_n\}$  の中から最適な視差  $d$  を推定する。この段階で、正しく視差が求まると期待できる点については視差が求まる。最後に、視差が求まっていない領域を補間することによって視差マップを得る。以下に、各ステップごとの処理を詳細に説明する。

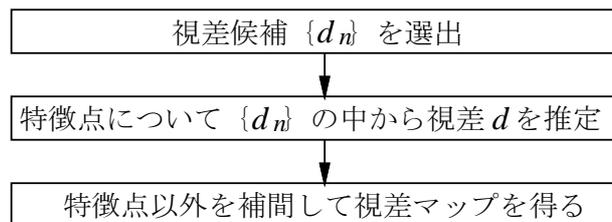


図7 奥行き推定アルゴリズムの流れ

### 3.1.1 視差候補の選出

#### 【処理内容】

後の処理を軽減するため、画像内の支配的な視差の候補  $\{d_n\}$  を選出する。

#### 【エッジ位置の相関関数】

支配的な視差を検出するために、エッジ位置の相関関数を求める。ここでのエッジとは、輝度レベル  $I(x, y)$  の横方向の変化  $\frac{\partial I(x, y)}{\partial x}$  とする<sup>2</sup>。視差を  $d$  としたとき、左画像上の点  $I_L(x, y)$  に対応する右画像上の点は  $I_R(x - d, y)$

<sup>2</sup>ここでは対称性を考慮し、 $\frac{\partial I_L(x, y)}{\partial x} \equiv I_L(x + 1, y) - I_L(x - 1, y)$  として離散化した。

であることから、左右の画像間のエッジの相関関数は次のように表せる。

$$\sum_x^{all} \sum_y^{all} \left\{ \frac{\partial I_L(x, y)}{\partial x} - \frac{\partial I_R(x - d, y)}{\partial x} \right\}^2 \quad (5)$$

$$= \sum_x^{all} \sum_y^{all} \left\{ \left( \frac{\partial I_L(x, y)}{\partial x} \right)^2 + \left( \frac{\partial I_R(x - d, y)}{\partial x} \right)^2 \right\} - 2 \sum_x^{all} \sum_y^{all} \frac{\partial I_L(x, y)}{\partial x} \cdot \frac{\partial I_R(x - d, y)}{\partial x} \quad (6)$$

この関数は、右画像が左画像に対して  $d$  だけずれた時のエッジの相違度を表す。ここで、式 (6) の第 1 項は、加算領域が十分大きい場合は  $d$  に関して不変であると考えてよいので、第 2 項のみが意味を持つ。よって第 2 項を  $-2$  で割った関数を  $corr(d)$  として定義する。

$$corr(d) = \sum_x^{all} \sum_y^{all} \frac{\partial I_L(x, y)}{\partial x} \cdot \frac{\partial I_R(x - d, y)}{\partial x} \quad (7)$$

$corr(d)$  は、右画像が左画像に対して  $d$  だけずれた時のエッジの類似度を表す。 $corr(d)$  の値が大きいところは、画像内に  $d$  の視差がある可能性が高いことを示しているため、それらを列挙して視差候補  $\{d_n\}$  とする。

### 3.1.2 特徴点における視差推定

#### 【処理内容】

画像中の特徴点について、ブロックマッチングにより視差候補  $\{d_n\}$  の中から最適な視差  $d$  を求める。

#### 【ブロックマッチング】

ある大きさのブロックを単位としたテンプレートマッチングのことを、ブロックマッチングと呼ぶ。一方の画像を固定して他方をずらしながら、ブロック内の絶対値誤差和が最小になる場所を探すことにより、画像のずれを推定する。この処理は、ブロック内に特徴がなければずれを正しく推定することができない。ここでは、 $|\frac{\partial I(x, y)}{\partial x}| \geq ft$  を満たす点<sup>3</sup> についてのみ

<sup>3</sup>ここでの閾値は  $ft = 30$  とし、経験的に決定した。

ブロックマッチングを行ない、式(8)の  $err(d_n)$  を最小にする  $d$  を  $\{d_n\}$  の中から決定する。この段階で、各特徴点において視差が求まる。

$$err(d_n) = \sum_{\Delta x=-k_x}^{+k_x} \sum_{\Delta y=-k_y}^{+k_y} |I_L(x+\Delta x, y+\Delta y) - I_R(x-d_n+\Delta x, y+\Delta y)| \quad (8)$$

ブロックマッチング自体の計算量は比較的多いが、ここではブロックマッチングを行なうべき点は特徴点のみであり、評価すべき視差も視差候補  $\{d_n\}$  のみであり共に少ないため、高速性は十分保てる。

なお、ここでは左画像の奥行きを求める処理を示したが、同様に右画像の奥行きも求める。

#### 【ブロック形状】

ブロックマッチングでは  $3 \times 3$ 、 $5 \times 5$ 、 $7 \times 7$ 、… などの正方ブロックを用いるのが一般的である。ブロックサイズは小さい方が処理は高速であるが、ブロック内部にマッチングに有効な特徴が含まれる可能性は低くなるため、誤った結果を出す危険性がある。特にぼけた画像では、明確な特徴を捕らえにくくなるため不利である。一方、ブロックサイズを大きくしすぎると、ブロック内部にマッチングには余分な特徴まで含んでしまう可能性が高くなり、これも誤った結果を出すことにつながる。また、ブロックサイズは計算コストに大きく影響するため、必要最小限の大きさのブロックサイズを選ぶことが重要である。

本研究では特に高速性が要求されるため、ブロックサイズは小さい方が適している。しかし、 $3 \times 3$  のブロックで常に正しいマッチングをとることは困難である。 $\pm 0.5[\text{pixel}]$  のサンプリング誤差は不可避であり、 $3 \times 3$  のブロック内の多くの点は誤差範囲となるため、ロバスト性に欠ける。より小さいブロックでロバスト性を向上させるには、マッチングに対してより貢献度の高い点だけを使ってブロックを形成するべきである。本研究では、視差によるずれは横方向に限られるため、横方向の点の並びがマッチングに対する貢献度はより高いと考え、 $5 \times 1$  の横長の非正方ブロックを用い

た。これにより、 $3 \times 3$  の正方ブロックを用いるよりも多くの場合でサンプリング誤差やノイズに強くなり、ロバストかつ高速であることを確認した。

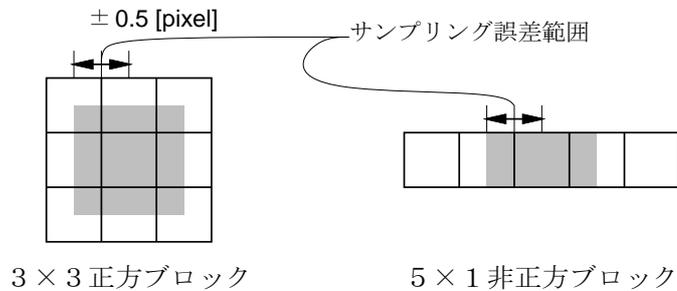


図 8 ブロックマッチングにおけるブロック形状

### 3.1.3 視差補間

#### 【処理内容】

視差が求まっていない領域を補間し、密な視差マップを作成する。

#### 【左右の視差データの一致と補間】

ここまでの段階で、左右ともに特徴点について視差は求まったが、それ以外の点では求まっていないため、その領域を補間する必要がある。ここで、左右の画像は互いに独立なものではないため、左右の視差マップ間で矛盾がないように補間しなければならない。左右で矛盾のあるステレオペアで立体視をすると視野闘争<sup>4</sup> が起こり [16]、目の疲労を起こす原因になるため、改善が望ましい。また、左右それぞれで閉じた処理を行なうより、両方に共通した特徴に注目した方が、フォールスポジティブ<sup>5</sup> は減少でき、ロバスト性が向上すると考えられる。このような理由から、左右の視差データの一致をとりながら補間する。

<sup>4</sup>左右眼に与えられる視覚刺激の形や動きなどが互いに極端に異なり 1 つに融合できない場合に、各々の視覚刺激が交互に知覚される現象。

<sup>5</sup>False Positive : 陰性であるものを、誤って陽性と判断すること。

以下に、具体的な処理を示す。現在求まっている視差データの横方向1ラインについて注目し、左右画像それぞれのx座標  $x_L, x_R$  を軸にとった  $(x_L, x_R)$  平面上に、視差データをプロットする。左画像中の点  $x_L$  における視差を  $d$  とすると、点  $x_L$  に対応する右画像中の点は  $x_R = x_L - d$  となる。視差候補は  $\{d_n\}$  に限られているので、全ての点は  $x_R = x_L - \{d_n\}$  のいずれかの直線上にプロットされる 図9-(a))。同様に、右画像中の点  $x_R$  に対応する左画像中の点は  $x_L = x_R + d$  となるため、(a) と同じ直線上にプロットされる 図9-(b))。ここで、(a),(b) どちらか一方でしかプロットされていない点はブロックマッチングに失敗した可能性が高く、信頼性が低いため、これらのデータは捨てる。両方でプロットされている点は左右の一致がとれている点であり、信頼性が高いため、以降はこれらのデータのみを用いる 図9-(c))。

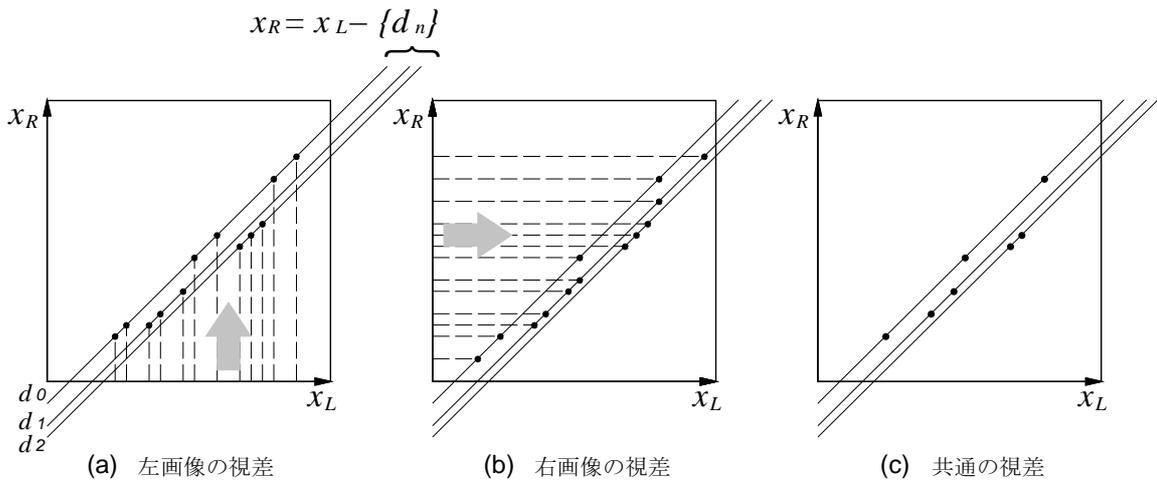


図9 視差データの  $(x_L, x_R)$  平面へのプロット

次に、離散的に得られたこれらの視差データを補間して、左右それぞれの視差マップを得る。左の視差マップは、 $x_R = x_L - \{d_n\}$  上の各点を  $x_L$  軸上に投影し、2つの隣接する特徴点間を小さい方の視差データで補間する 図10-(a))。右の視差マップは  $x_R$  軸上に投影し、同様に補間する 図10-(b))。

全点の視差をブロックマッチングで求めた視差マップでは、オクルージョン領域の視差は不定になるという問題があるが、本手法ではオクルージョン領域は背景と同じ視差を持つことになり、手前の物体の輪郭は保持されるという特徴を持っている。

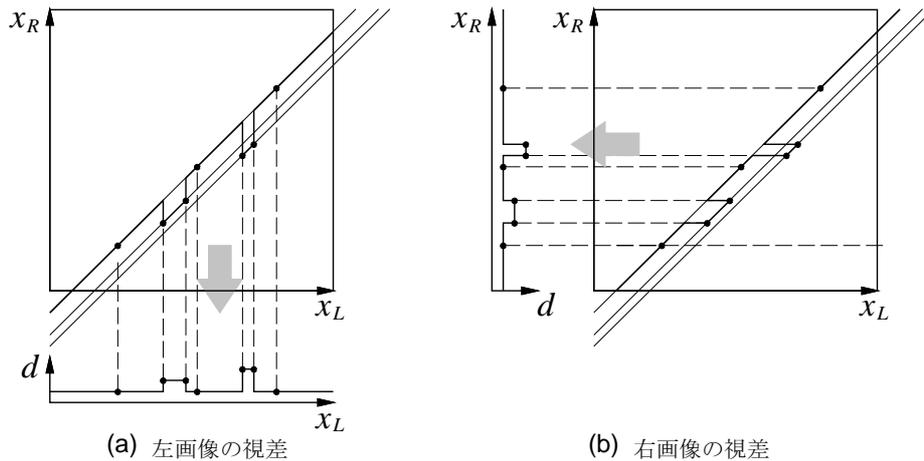


図 10  $(x_L, x_R)$  平面から視差マップへの写像

**【小さい方の視差で補間する理由】**

補間アルゴリズムにおいて、2点間を小さい方の視差で補間する理由を図 11 を用いて説明する。まず、視差は注目物体の内部／境界／外部で求まる。内部／外部では、物体表面に模様がありエッジが検出される場合は求まるが、そうでない場合は求まらないのに対し、境界はエッジであり特徴がある場合が多いので、視差は求まる可能性が高い。一方、物体の外部とは注目物体に対する背景であるため、外部の視差は相対的に小さく、内部の視差は大きい。また、境界では内部と同じ視差をもつ。

ここで、2点の視差が求まっているだけでは、その2点が内部／境界／外部のいずれであるかは判断できないが、物体の境界では必ず視差が求まると考えると、この2点の組合せは表 1 (A)～(D) のいずれかになる。

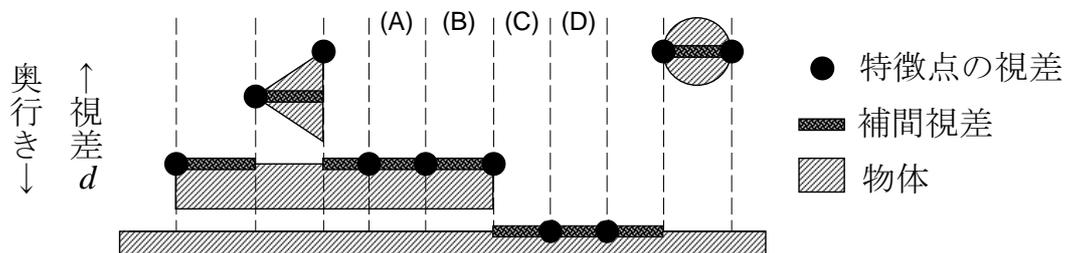


図 11 視差の補間

表 1 2つの特徴点と、その間を補間するべき視差

	2つの特徴点 視差)		補間するべき視差
(A)	内部 大)	内部 大)	内部 大)
(B)	内部 大)	境界 大)	内部 大)
(C)	境界 大)	外部 小)	外部 小)
(D)	外部 小)	外部 小)	外部 小)

2点間を補間するべき視差は、(A),(D)については明らかである。(B),(C)については、境界) = 内部の端)であることを考慮すると、表1のようになることがわかる。以上、(A)~(D)の全てのケースにおいて、「2点のうち小さい方の視差で補間する」という処理が共通していると言える。したがって、視差が求まっていない領域を、視差が求まっている両端の2点のうち小さい方の視差で補間すれば、横方向に関しては矛盾なく視差を補間できることになる。

## 3.2 領域追跡アルゴリズム

動画画は時間軸方向に強い相関をもつため、過去のフレームで得られた情報を利用できるものが多い。また、1組のステレオペアだけからでは正しく得られなかった情報を、複数ペア用いることにより、正しい情報に修正できる場合もある。よって、動画画においては時間軸相関を利用するのは有効な手段である。

ここでは、時間的に近接したフレーム間では、物体の位置や形の変化は十分小さいと仮定し、過去のフレームにおける同一視差領域の画像上の動きを検出することにより、視差マップの更新を行なう。全フレームで奥行き推定を行なうと、視差マップの動きが連続的でないために動画画がちらつくという問題があるが、本手法により自然な動画画の生成が可能となる。また、領域追跡処理は奥行き推定処理よりも高速であるので、処理速度の面でも適している。

領域追跡アルゴリズムの流れを 図 12 に示す。まず、同一視差領域ごとに過去フレームからの動きを検出し、その動きに応じて視差マップを更新する。次に、過去フレームと比較して、必要であれば領域修正する。以下に、各ステップごとに処理を説明する。

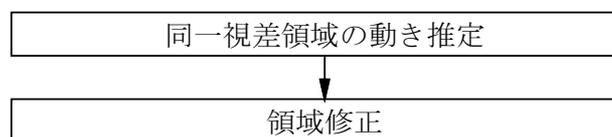


図 12 領域追跡アルゴリズムの流れ

### 3.2.1 同一視差領域の動き推定

#### 【処理内容】

同一視差領域ごとに画像上の動きを検出し、視差マップを更新する。

#### 【領域ごとの動き検出】

時間的に近接したフレーム間では、類似した視差マップが続くと考えられる。ここでは、視差マップは物体ごとの平行移動で記述できると仮定し、各

物体の動きを検出することにより視差マップを更新することを考える。しかし、実際には同一物体の識別は困難である。ここでは、同一物体は同一視差を持つという仮定のもとに、同一視差領域の動きを検出することにより、視差マップを更新する。

動きベクトルは、同一視差領域を単位としたテンプレートマッチングにより求める。視差  $d$  の領域の動きベクトルを求める式を次に示す。

$$err(v_{nx}, v_{ny}) = \sum_x^{all} \sum_y^{all} \{|I_{crnt}(x + v_{nx}, y + v_{ny}) - I_{prev}(x, y)| \cdot obj(x, y)\} \quad (9)$$

$$obj(x, y) = \begin{cases} 1; & D_{prev}(x, y) = d \text{ のとき} \\ 0; & \text{それ以外} \end{cases}$$

$I_{crnt}(x, y)$  : 現在の入力画像

$I_{prev}(x, y)$  : 過去の入力画像

$D_{prev}(x, y)$  :  $I_{prev}(x, y)$  の視差マップ

$obj(x, y)$  は注目領域内部では 1、外部では 0 の値をとる二値関数であり、 $D_{prev}(x, y)$  より求まる。 $err(v_{nx}, v_{ny})$  は注目領域内部の絶対値誤差和を表し、この値が最小になるベクトル  $(v_{nx}, v_{ny})$  がその領域の動きベクトルとなる。最後に、各領域を  $(v_{nx}, v_{ny})$  だけ平行移動することにより、視差マップを更新する。

この処理は、仮想物体より手前にある領域についてのみ行なう。これは、微小時間内ならば物体の前後関係は変わらないという仮定の下では、仮想物体より奥にある物体は、画像上でどこに移動しようと仮想物体で上描きしてかまわないからである。また、明らかに仮想物体と画像上で重ならない位置に存在する物体も追跡する必要はない。これらにより、動き推定的高速化が図れる。さらに、左右の画像中の物体の動きベクトルをそれぞれ  $(v_{Lx}, v_{Ly}), (v_{Rx}, v_{Ry})$  としたとき、必ず  $v_{Ly} = v_{Ry}$  になるという拘束条件を用い、あり得ない結果を出すことを避けることにより、動き推定のロバスト化を図る。

なお、過去フレーム  $I_{prev}$  には奥行き推定を行なったフレームを用いる。これは  $I_{prev}$  に1つ前の物体追跡結果を用いた場合、量子化誤差が伝搬するのを防ぐためである。ただし、動き探索は1つ前の物体追跡結果を中心とした範囲で行なう。

### 【動き探索範囲の考察】

動き推定では、想定している動き範囲内の全てのベクトル  $\vec{v}_n = (v_{nx}, v_{ny})$  について式 (9) の評価を行ない、 $err(v_{nx}, v_{ny})$  が最小になる  $\vec{v}_n$  を求める。当然のことながら、動き探索は真の動きベクトルより広い範囲で行なわなければならない。しかし、探索範囲の大きさは計算コストに大きく影響するため、高速化のためには探索範囲を必要最小限にする必要がある。

リアルタイム処理では、次のフレームに移るまでに各フレームでの動きベクトルの計算が終了している必要があるため、動き推定処理の計算コストが高いとフレーム間の時間間隔を長く取らなければならない。ところが、時間間隔を長くするとフレーム間の物体の移動量も大きくなるため、より広い範囲を探索する必要が生じ、さらに計算コストが高くなってしまう。ここでは、このような悪循環を避けるため、単位時間あたりの計算コストを最小にすることを考える。まず、各変数を以下のように定義する。

$p$ [pixel/sec]	: 単位時間あたりの動き最大量
$f$ [frame/sec]	: 単位時間あたりのフレーム数
$r$ [pixel/frame]	: 1 フレームの動き探索範囲 (片側)
$i$ [inst/frame]	: 1 フレームの計算コスト
$c$ [inst/sec]	: 単位時間あたりの計算コスト

$r$  は1フレームでの動きの許容最大量をカバーする必要があるため、次式の関係がある。

$$r = \frac{p}{f} \quad (10)$$

$i$  は実際のシミュレーションシステムにおいて現実に則した状態を理論的に記述することは困難であるので、実験的に求める。これは入力画像などの

諸条件によっても異なるが、次章の実験と同じ画像を用い、最も多くの処理を必要とする設定<sup>6</sup>で、 $r = \{1, 2, 3, \dots, 15\}$  について計算機のユーザー使用 CPU 時間を調べ、最小 2 乗法で 2 次式に近似して式 (11) を得た。2 次式で近似する理由は、探索範囲の面積は  $r^2$  に比例するため、計算量も  $r^2$  に比例すると考えられるからである。なお、 $(i, r)$  のグラフは  $i$  軸対称となることから  $A_1 = 0$  の拘束条件を用いた。

$$i = A_2 r^2 + A_1 r + A_0 \quad (11)$$

$$A_2 = 13.405$$

$$A_1 = 0.000$$

$$A_0 = 68.380$$

高速化の指標となる  $c$  は次式で表せる。

$$c = i \cdot f \quad (12)$$

式 (10)~(12) を用いると、 $c(r)$  は次式のようにになる。

$$c(r) = (A_2 r^2 + A_0) \cdot \frac{p}{r} \quad (13)$$

$$= \left( A_2 r + \frac{A_0}{r} \right) \cdot p \quad (14)$$

$c(r)$  を最小にする  $r$  は次の関係を満たす。

$$\frac{dc(r)}{dr} = \left( A_2 - \frac{A_0}{r^2} \right) \cdot p = 0 \quad (15)$$

$c(r)$  は下に凸の関数であるので、 $r = \sqrt{\frac{A_0}{A_2}} \approx 2.26$  のとき、計算コストは最小になる。ここでは、切り上げて  $r = 3$  とした。これは、本実験で用いたシステムにおいて、例えば  $p = 50[\text{pixel/sec}]$ <sup>7</sup> 以下の動きに対応できるとしたとき、 $f \approx 17[\text{frame/sec}]$  の処理スピードが必要であることを示している。

<sup>6</sup>具体的には、画像全域にわたる大きさの仮想物体が無限遠に存在するという設定。

<sup>7</sup>本実験で用いたシステム（表 2 データ入力部分参照）では、カメラから 3[m] の距離で約 12.5[cm/sec] の速度に相当する。

### 3.2.2 領域修正

#### 【処理内容】

領域追跡中に、注目領域内で輝度変化が起こった点は物体の外部であるとみなし、領域を修正する。

#### 【修正アルゴリズム】

1枚の画像だけでは、物体の内部／外部の区別ができない場合がある。例えば図13のように、第nフレームだけではコップの取っ手部分は穴があいているのか（外部と判断）、単に黒い模様があるのか（内部と判断）、区別ができない。どちらか明らかでない場合は、内部と考える方が一般性は保てるため、この段階では模様であると判断される。しかし、領域追跡中に背景が変わって取っ手部分の輝度変化が変化すると、これは模様ではなく穴があいているということが明らかになる。したがって、領域追跡中に、領域内で輝度変化が起こった点は物体の外部であると考え、領域から外すことにより修正する。この処理は、オクルージョンなどの原因で誤って物体内部と判断された領域の修正や、移動しながら変形する物体の領域を更新することにも効果がある。

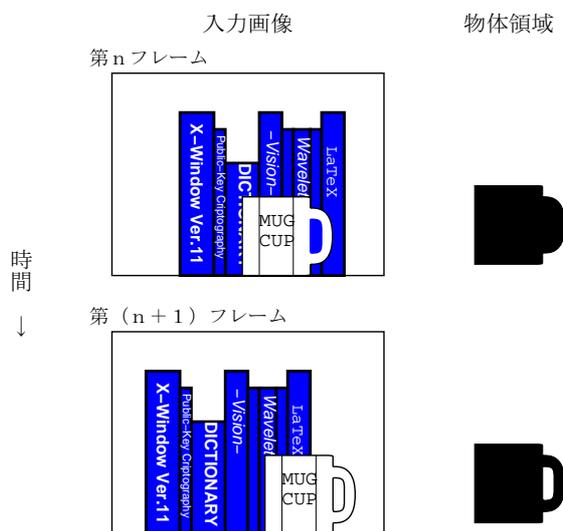


図13 輝度変化を用いた領域修正

### 3.3 合成画像生成アルゴリズム

奥行き推定や領域追跡で求めた視差マップをもとに、合成画像を生成する過程を示す。実画像 図 14-(a) の視差マップとして 図 14-(b) が得られたとする。また、仮想物体 図 14-(c) の奥行きは図 14-(d) の視差に相当する位置であるとする。なお、各視差マップは、明るいものほど手前であることを表している。ここで、図 14-(e) の各領域 A~C を以下のように処理することにより、奥行きに応じた隠蔽関係で合成画像 図 14-(f) を生成できる。

- A :仮想物体が存在しない領域 → 実画像を描く
- B :仮想物体が存在し、実物体の方が手前である領域 → 実画像を描く
- C :仮想物体が存在し、実物体の方が奥である領域 → 仮想物体を描く

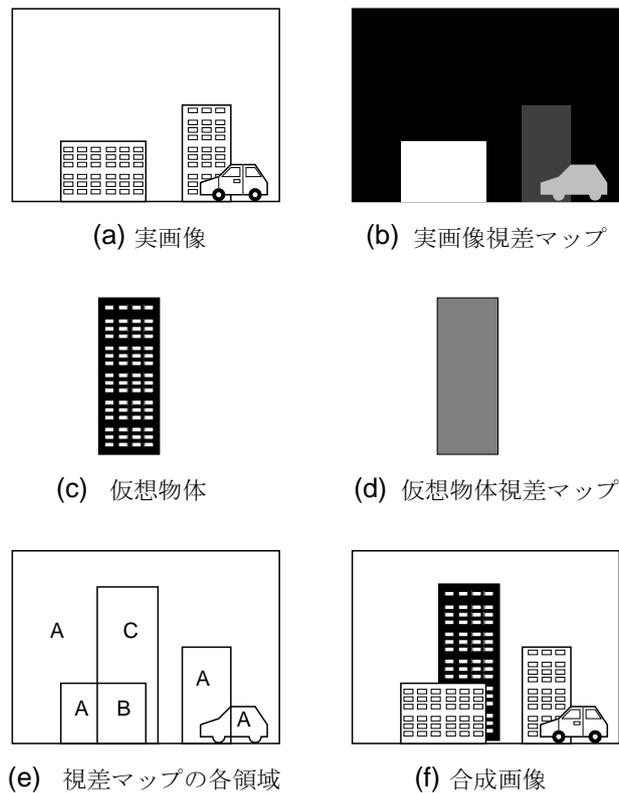


図 14 合成画像生成過程

## 4. 評価実験

本手法の有効性を検証するため、ステレオ動画像に仮想物体を合成する実験を行なった。本章では、実験データとその結果を示す。

### 4.1 シミュレーション環境

シミュレーションを行なった環境を表 2 に示す。

表 2 シミュレーション環境

データ入力部分	
カメラ	Sony 2/3 インチ 3CCD カメラ XC-007
レンズ	Canon JF25 1.4B4
ビデオ	Silicon Graphics Galileo Video
計算機	Silicon Graphics Indigo <sup>2</sup>
入力画像	320 × 242[pixel] 左右 2 枚 白黒 256 階調 60 フレーム
画像合成処理部分	
計算機	DEC alpha-3000
コンパイラ	gcc version 2.6.3 最適化オプション (-O3)

### 4.2 実験データ

#### 4.2.1 入力実画像

ステレオ入力画像の第 {0,20,40,60} フレームを図 17 に示す。シーン内には最も手前に卓上カレンダーがあり (右下)、その奥に CD があり (中央下)、背景として CRT やキャビネットなどがある。カレンダーと CD が載っているテーブルは左に平行移動しており、カメラは右にパンニングしている。

#### 4.2.2 仮想物体画像

仮想物体として合成する画像を図15に示す。スペースシャトルの領域以外には、データは存在しない。この画像を左に移動させ、飛行シーンの動画像にした。挿入する奥行きは、カレンダーの奥と、CDの奥の2通りで行なった。左右の画像は、その奥行きに相当する視差の分だけずらして合成した。

本実験では、仮想物体画像は1枚の写真をもとに作成したため、ステレオ視しても仮想物体は平面的に見えるが、3次元仮想物体を2次元に投影した左右2枚の画像を入力画像とすることにより、立体的な仮想物体を表示することが可能である。

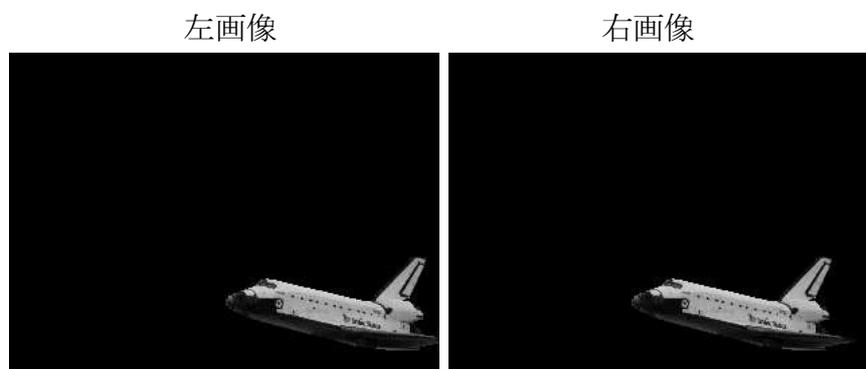


図15 仮想物体画像

### 4.3 実験結果

#### 4.3.1 奥行き推定の結果

##### 【視差候補の選出結果】

第0フレームのステレオペアにおけるエッジ位置相関関数  $corr(d)$  を図16に示す。ここでは、 $corr(d)$  が閾値  $1.5 \times 10^6$  以上でかつ極大値をとる  $d$  をピークとして選出した。各ピークについては、 $d = 22$  は背景、 $d = 36$  はCD、 $d = 42$  はカレンダーの視差を示している。

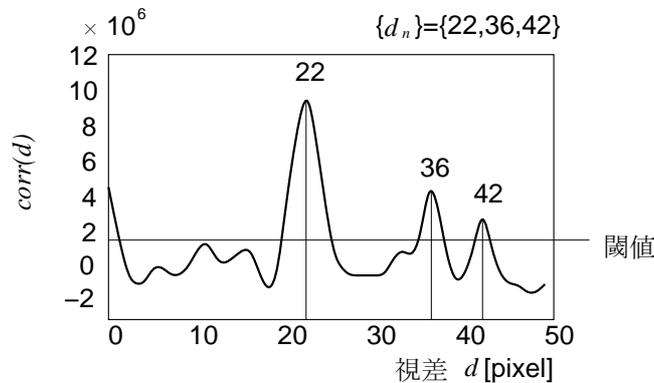


図 16 奥行き推定アルゴリズムにおけるエッジ位置の相関関数

**【特徴点における視差推定結果】**

特徴点についてのみ  $\{d_n\}$  の中から視差  $d$  を選出した結果を図 18 に示す。るい点ほど手前であることを示している。なお、視差データがない点は黒で示した。

**【視差補間結果】**

左右で矛盾がないように横方向に視差データを補間した結果を図 19 に示す。カレンダーの文字領域には同じようなパターンが並んでいるため、ブロックマッチングに失敗しているところが多い。

**【合成画像】**

視差マップ (図 19) をもとに、実画像に仮想物体を合成した例を示す。カレンダーの向うに合成した例を図 20 に、CD の向うに合成した例を図 21 に示す。これらの画像は、平行法のステレオ視により、あたかも実空間中の意図した 3 次元位置に仮想物体が存在するように感じることができる。なお、この動画像では全てのフレームで奥行き推定を行ない、各時刻で閉じた処理を行なっているため、視差マップの動きが時間軸方向に連続ではないところがある。そのため、この動画像をビデオで試聴するとちらつきを感じる。

### 4.3.2 物体追跡の結果

#### 【動き推定と領域修正の結果】

t=20 [frame] の周期で物体追跡を行なった場合の視差マップを 図 22～図 24 に示す。なお、第 {0,20,40} フレームは奥行き推定結果である。ここでは、仮想物体を CD の向うに合成したため、仮想物体より手前にあるカレンダーと CD のみを追跡した。仮想物体より奥の物体の情報は必要ないので、背景部分のデータはない。図より、各物体の動きに応じて視差マップが更新されているのがわかる。第 40 フレームでは誤って物体領域と判定されていた画像中の左上の領域が、以降のフレームでは領域修正されているのがわかる。一方、第 20 フレームの CD 右上の領域の奥行き推定の誤りは、以降のフレームでも継続されている。奥行き推定で誤ると、その周期内のフレームは全て誤るため、奥行き推定は正確に行なう必要がある。

#### 【合成画像】

視差マップ (図 22～図 24) をもとに奥行きを判定し、仮想物体を CD の向うに合成した例を 図 25～図 27 に示す。物体追跡は時間軸方向に連続的であるので、ビデオで試聴してもちらつきが少ない。しかし、第 20 フレームで失敗した奥行き推定の影響はその周期内で続くため 領域修正アルゴリズムはフォールスポジティブは減少できるが、フォールスネガティブ<sup>8</sup>は減少できない)、奥行き推定で誤りが存在すると目立つという問題がある。

## 4.4 処理時間

60 フレームの処理に要したユーザー使用 CPU 時間を 表 3 に示す。周期 t [frame] が大きいほど、すなわち領域追跡処理を多くするほど、全体の処理は高速になる。処理のほとんどを領域追跡処理で済ませるためには、最初の奥行き推定で誤らないことが重要である。

現状のプログラムはシミュレーション用であり、画像合成にはかならずしも必要でない機能も付いてあるため、不要部分を削除することにより、さらに高速化

<sup>8</sup>False Negative :陽性であるものを、誤って陰性と判断すること。

は可能である。また、ハードウェア化により、準リアルタイム処理も期待できる。

表 3 60 フレームの処理時間 [sec]

		仮想物体の合成位置	
		カレンダーの奥	CD の奥
周 期 t	1	87.2	87.2
	2	55.9	61.1
	5	37.0	45.7
	10	31.7	40.3
	20	29.9	37.7
	60	31.4	34.1

左画像

右画像



(a) Frame=0



(b) Frame=0



(c) Frame=20



(d) Frame=20



(e) Frame=40



(f) Frame=40



(g) Frame=60



(h) Frame=60

図 17 入力画像

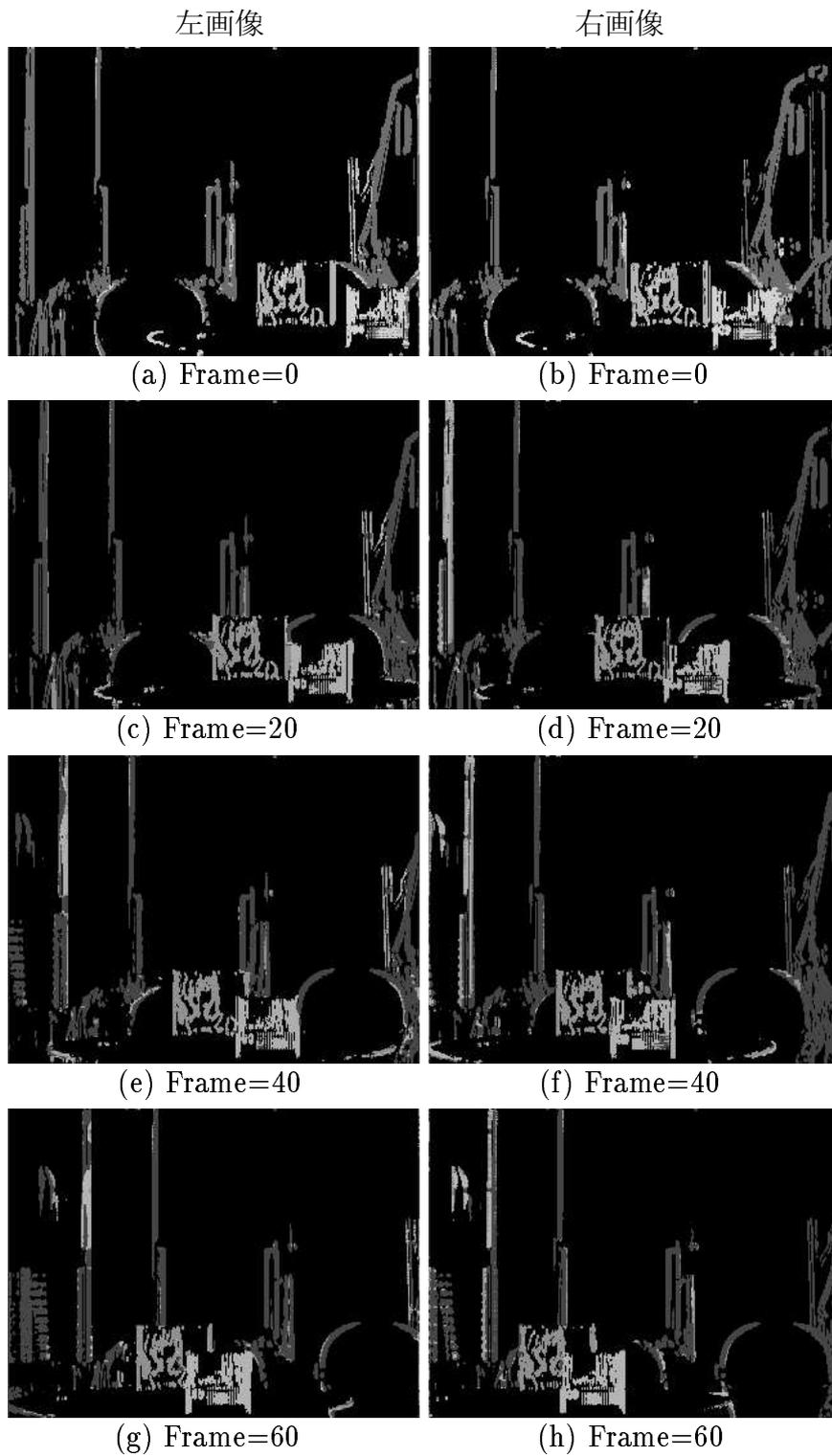


図 18 奥行き推定アルゴリズムによる特徴点における視差推定結果

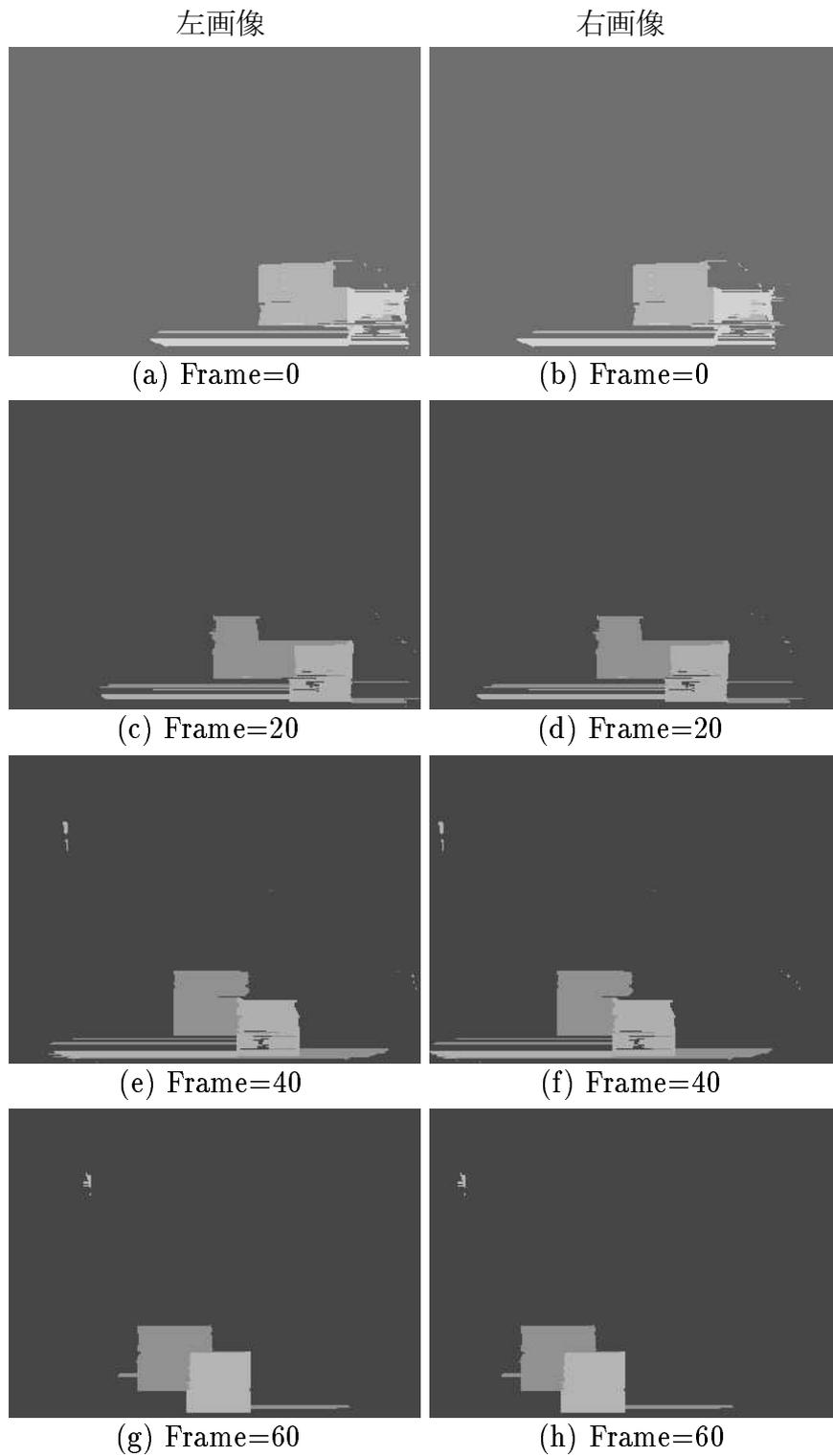


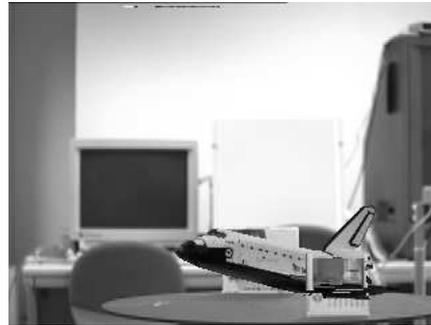
図 19 奥行き推定アルゴリズムによる視差補間結果

左画像

右画像



(a) Frame=0



(b) Frame=0



(c) Frame=20



(d) Frame=20



(e) Frame=40



(f) Frame=40



(g) Frame=60



(h) Frame=60

図 20 奥行き推定アルゴリズムを用いた合成画像 (例 1)

左画像

右画像



(a) Frame=0



(b) Frame=0



(c) Frame=20



(d) Frame=20



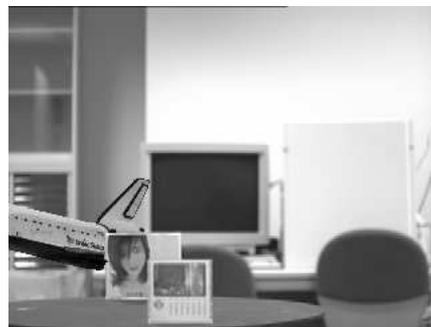
(e) Frame=40



(f) Frame=40



(g) Frame=60



(h) Frame=60

図 21 奥行き推定アルゴリズムを用いた合成画像 (例 2)

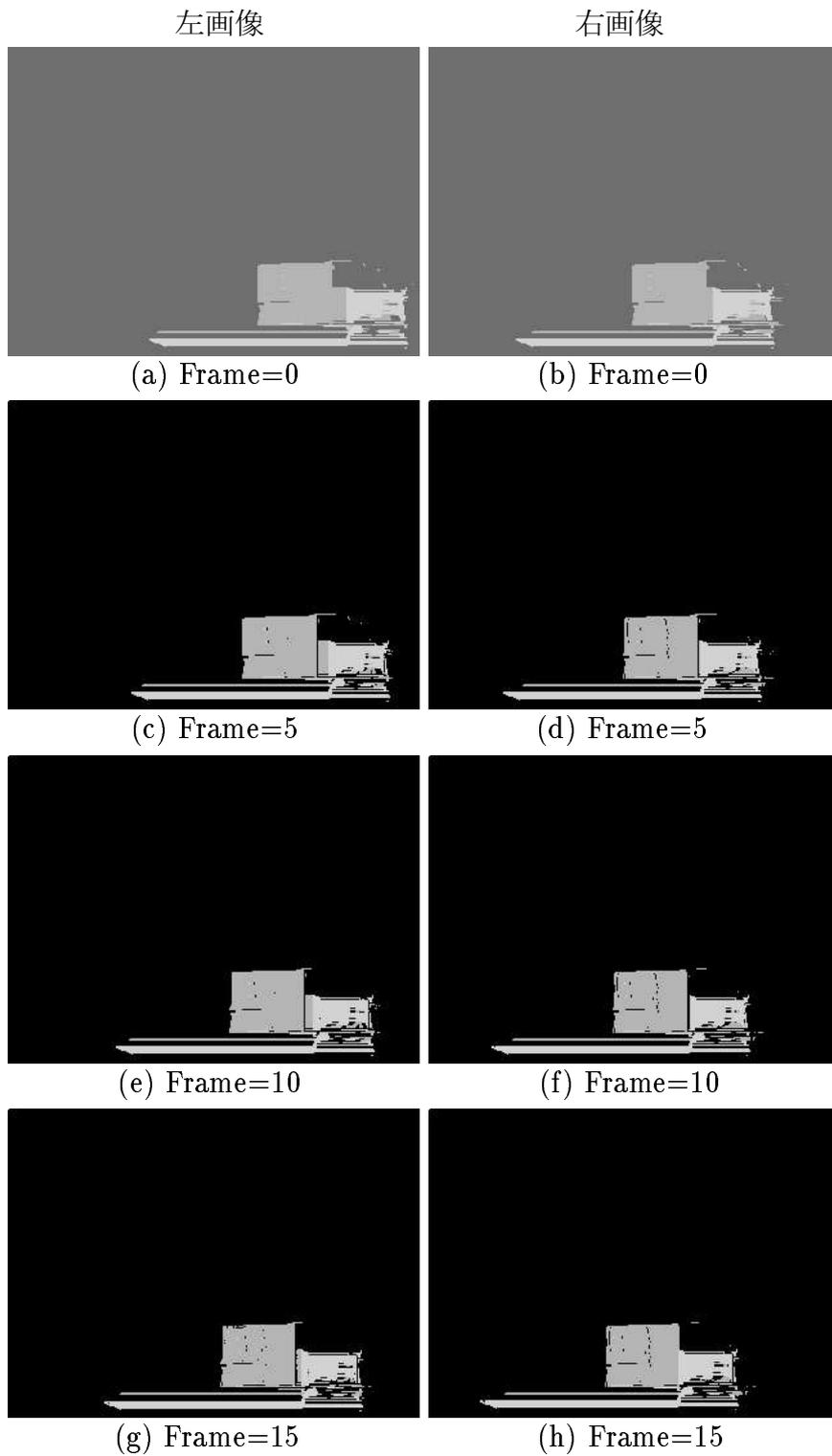
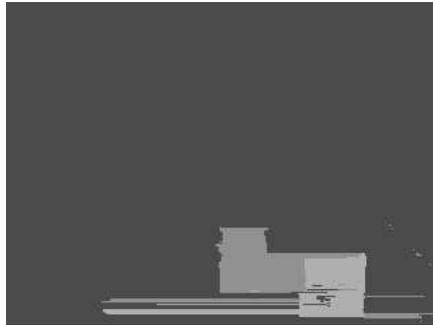


図 22 領域追跡アルゴリズムによる動き推定と領域修正の結果 (1/3)

左画像

右画像



(a) Frame=20



(b) Frame=20



(c) Frame=25



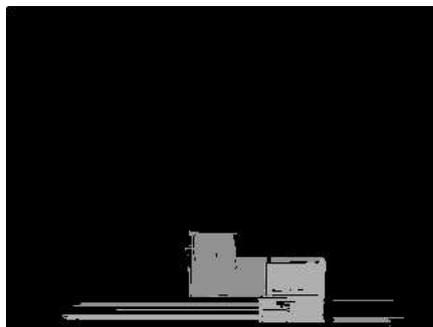
(d) Frame=25



(e) Frame=30



(f) Frame=30



(g) Frame=35



(h) Frame=35

図 23 領域追跡アルゴリズムによる動き推定と領域修正の結果 (2/3)

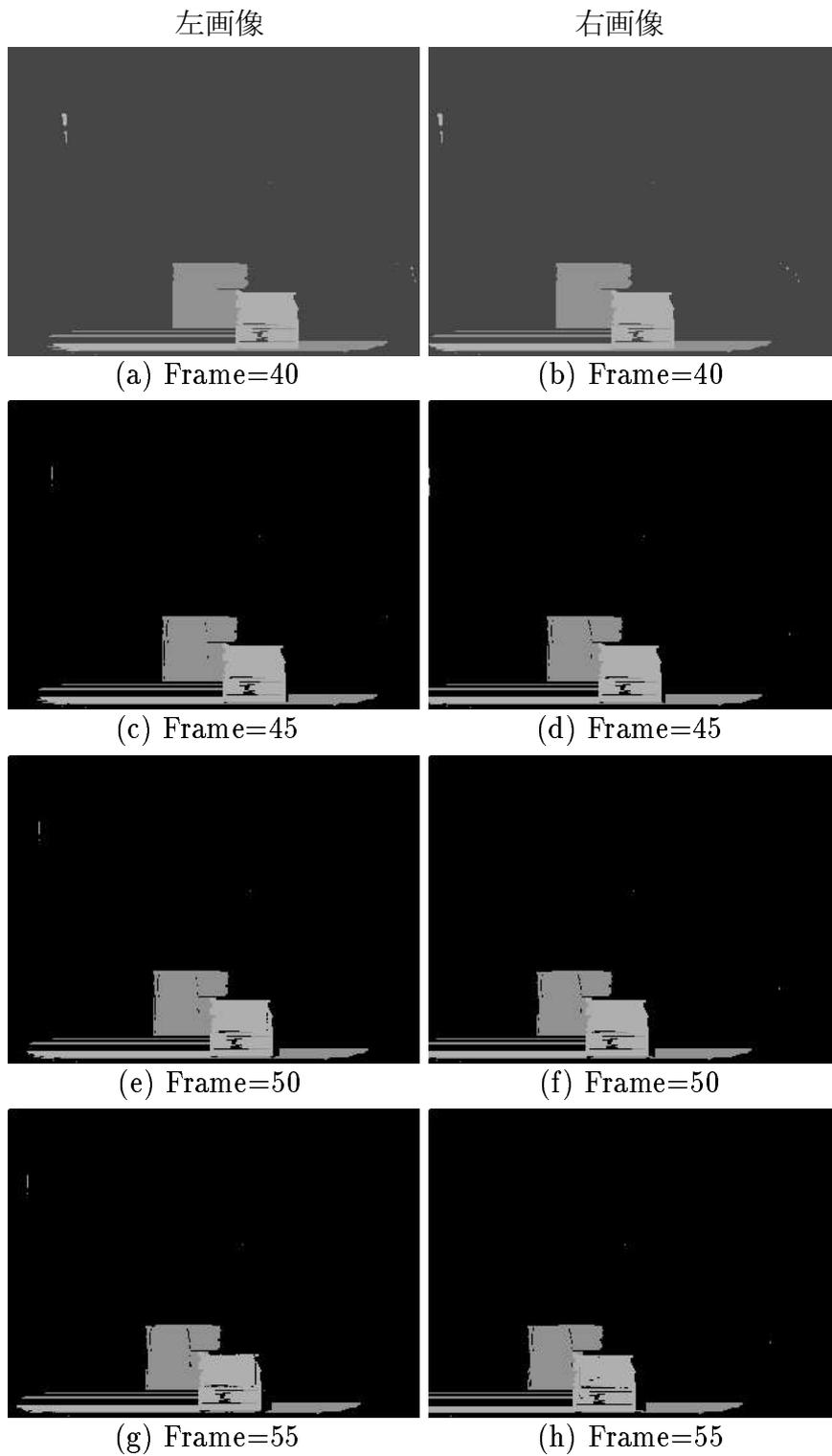


図 24 領域追跡アルゴリズムによる動き推定と領域修正の結果 (3/3)

左画像

右画像



(a) Frame=0



(b) Frame=0



(c) Frame=5



(d) Frame=5



(e) Frame=10



(f) Frame=10



(g) Frame=15



(h) Frame=15

図 25 領域追跡アルゴリズムを用いた合成画像 (1/3)

左画像

右画像



(a) Frame=20



(b) Frame=20



(c) Frame=25



(d) Frame=25



(e) Frame=30



(f) Frame=30



(g) Frame=35



(h) Frame=35

図 26 領域追跡アルゴリズムを用いた合成画像 (2/3)

左画像

右画像



(a) Frame=40



(b) Frame=40



(c) Frame=45



(d) Frame=45



(e) Frame=50



(f) Frame=50



(g) Frame=55



(h) Frame=55

図 27 領域追跡アルゴリズムを用いた合成画像 (3/3)

## 5. 結論

両眼ステレオ画像の視差から実空間の奥行きを推定し、奥行きに応じた隠蔽関係で、仮想物体を実空間の任意の3次元位置に表示する手法を示した。物体を2次元で表現することにより物体同士の交錯を回避する、画像合成に適した高速な奥行き推定アルゴリズムを提案した。また、領域単位で動きを検出する領域追跡アルゴリズムにより、高速かつ自然な動画像合成を行なう手法を示した。

本アルゴリズムの改良案として、以下のような事項が考えられる。

- 処理周期  $t$  の検討

処理周期  $t$  が小さいと、あるフレームで奥行き推定に失敗しても、次の奥行き推定ですぐ復帰できる可能性が高い反面、時間軸方向のつながりが弱くなるため、動画像としてちらついて見えるという傾向がある。また、処理速度も比較的遅いという問題もある。 $t$  を大きくすることにより、ちらつきは減少し、処理も高速化できるが、奥行き推定で誤りがあるとそれが継続され、誤りが目立ってしまうという問題がある。理想的には、奥行き推定のロバスト性を向上させ、 $t$  を大きくすることが望ましい。今後は、ロバスト性を向上させた上で、人間の視覚機能の特性も考慮し、最適な  $t$  を検討する必要がある。

- カラー化

今回の実験では白黒画像を用いたが、カラー画像を用いて適当な色空間の情報を利用することにより、白黒画像では捕らえられなかったエッジを捕らえることが可能となり、奥行き推定のロバスト性を向上できると思われる。また、合成画像もより臨場感が増すと思われる。

- 視差マップの解像度削減

本研究では、視差マップの解像度は入力画像と同じとしたが、多くの場合、物体輪郭の空間周波数は低く、細かい点状や線状の領域が混在した視差マップは考えにくい。また、動画像では少々の輪郭のずれは認知されにくい。これらの理由から、視差マップの解像度は多少低くしても良いと思われる。これにより、処理の高速化が可能である。

- カメラ位置センサの必要性

本研究では、入力データはステレオ画像のみとし、カメラ位置などの情報は用いないものとした。しかし、実空間の座標系に基づいた運動をする仮想物体を表示するためには、絶対座標系からカメラ中心座標系への座標変換が必要であり、カメラ位置を知る必要がある。したがって、景観シミュレーションなどの AR システムとして利用するためには、カメラ位置を調べるセンサーがさらに必要である。なお、カメラ位置の情報は、座標変換以外に運動視にも利用可能である。立体視+運動視により、奥行き推定のロバスト性を向上できるものと思われる。

- センサフュージョン

近年、複数のセンサ情報を相補的・冗長的に統合することにより、外界の認識・判断を正確に行なうというセンサフュージョンの考え方が注目されている [17]。本研究の応用分野については、多眼カラーカメラ、超音波レンジファインダなどの併用が考えられる。より有益な情報を選択的に利用することができれば、より高速かつロバストな奥行き推定が可能になると考えられる。

本手法は、比較的高速であり、景観シミュレーションやバーチャルスタジオ、アミューズメントなど、リアルタイムの AR への応用が可能である。今後は、カメラ搭載型 HMD などハードウェアの開発も併せて行ない、AR システムを実現したい。

## 謝辞

本研究を進めるにあたり、終始暖かい御指導を頂いた ソフトウェア基礎講座 横矢直和 教授に厚く御礼申し上げます。

副指導教官としてご助言を頂いた 像情報処理講座 千原國宏 教授，並びに ソフトウェア基礎講座 竹村治雄 助教授に深く感謝致します。

本研究への助言や研究環境の整備に協力して頂いた ソフトウェア基礎講座 岩佐英彦 助手に深く感謝いたします。

本研究を進めるにあたり，研究環境の整備に協力して頂いた ソフトウェア基礎講座 茶園篤氏、草地良規氏に深く感謝致します。また、グラフィックツールに関してお世話になりました (株) リコー 研究開発本部 原潤一氏に感謝致します。

本学における研究の機会を与えて下さいました 東亜医用電子(株) 商品開発本部 大東重則 本部長、システム開発部 河口俊夫 部長、開発企画室 渡辺充 室長、ソフト開発課 西田真一 グループリーダーに深く感謝致します。また，本学への留学のさまざまな面でお世話になりました 人事部 山本義晴氏、岩本美佐氏に感謝致します。

最後に，ソフトウェア基礎講座の諸氏には，多大なる御協力を頂きました。ここに感謝の意を表します。

## 参考文献

- [1] 笹田剛史：“建築設計・都市計画におけるコンピュータグラフィックス”，情報処理学会誌, Vol.29, No.10, pp.1177-1181 (1988).
- [2] 曾山豊, 安田孝美, 横井茂樹, 鳥脇純一郎：“景観シミュレーションのための簡易情景モデル構築と動画生成の一手法”，テレビジョン学会誌, Vol.47, No.12, pp.1647-1655 (1993).
- [3] 福井一夫：“サイエンス番組におけるCGと実写を融合する可視化技術”，電気・情報関連学会連合大会講演論文集, pp.157-166 (1995).
- [4] 佐々木和郎：“テレビにおけるバーチャルスタジオの可能性”，テレビジョン学会誌, Vol.48, No.11, pp.1383-1385 (1994).
- [5] S. Feiner, B. MacIntyre, M. Haupt, E. Solomon：“Windows on the World: 2D Windows for 3D Augmented Reality”，*Proc. UIST'93 ACM Symposium on User Interface Software and Technology*, pp.145-155 (1993).
- [6] D. Drascic：“Stereoscopic Vision and Augmented Reality”，*Scientific Computing and Automation*, pp.31-34 (1993).
- [7] P. Milgram, S. Zhai, D. Drascic：“Applications of Augmented Reality for Human-Robot Communication”，*Proc. of IROS'93 International Conference on Intelligent Robots and Systems*, pp.1467-1476 (1993).
- [8] F. Betting, J. Feldmar, N. Ayache, F. Devernay：“A New Framework for Fusing Stereo Images with Volumetric Medical Images”，*Computer Vision. Virtual Reality and Robotics in Medicine '95 Proc.*, pp.30-39 (1995).
- [9] M. Wloka and B. Anderson：“Resolving Occlusion in Augmented Reality”，*Proc. 1995 Symposium on Interactive 3D Graphics*, pp.5-12 (1995).

- [10] 松原重徳、岩佐英彦、竹村治雄、横矢直和：“ステレオ視による奥行き推定に基づく実画像と仮想物体の合成”，電子情報通信学会春季大会講演論文集，No. D-645 (1996) 発表予定) .
- [11] 魚森讓也，山田光穂：“立体・奥行き知覚”，テレビジョン学会誌，Vol.48, No.12, pp.1502-1508 (1994).
- [12] 乾敏郎：視覚情報処理の基礎，1章，サイエンス社 (1991).
- [13] 井口征士、佐藤宏介：三次元画像計測，1章，昭晃堂 (1991).
- [14] D. Marr：*VISION -A Computational Investigation into the Human Representation and Processing of Visual Information-*, W. H. Freeman & Company (1982). 乾敏郎、安藤広志訳；ビジョンー視覚の計算理論と脳内表現ー、産業図書 (1987).
- [15] B. K. P. Horn (NTT ヒューマンインターフェース研究所プロジェクト RVT 訳)；ロボットビジョンー機械は世界をどう視るかー、朝倉書店 (1990).
- [16] 加藤和夫、小林哲生、栗城眞也：“視野闘争・視野融合刺激の交替に伴う脳波と脳磁界”，生体・生理工学シンポジウム論文集，pp.49-52 (1995).
- [17] 浅田稔：“センサフュージョン”，電子情報通信学会誌，Vol.75, No.7, pp.763-769 (1992).