

第3章 遅延補償アルゴリズムを統合した奥行き獲得手法

本章では、時系列の相関を利用した遅延補償アルゴリズムとステレオアルゴリズムを統合することで、ステレオ処理で生じる遅延問題を解決する手法について述べる。

3.1 遅延補償アルゴリズムを統合した奥行き獲得手法

前述のとおり、ステレオ法を用いた奥行き推定には、パイプライン処理に伴う出力の遅延を生じる。このパイプライン処理で生じる遅延は、アルゴリズムの簡単化やアルゴリズム中の個々の処理を高速化するだけでは完全に解決することはできない。そこで、単純にステレオ処理を高速化するのではなく、過去のシーンの奥行きマップを利用して現在のシーンの奥行きマップを推定する遅延補償アルゴリズムと従来のステレオ法を統合することで遅延問題に対応する手法を提案する。

図3.1に遅延補償アルゴリズムを統合した奥行き獲得アルゴリズムの概要を示す。処理はステレオ部と遅延補償部から構成されており、ステレオ部で従来のアルゴリズムを用いてシーンの奥行き推定を行うのと並行して、遅延補償部で時系列の相関関係からステレオ部で推定された過去の奥行きマップを参照し、現在のシーンの奥行きマップを推定する。そのため、ステレオ部では出力に多少の遅延が生じるもののが高精度な奥行き推定を行い、遅延補償部では精度は劣るが遅延の少ない奥行き推定を行う。

すなわち、現在のシーンの奥行き推定に過去の奥行きマップを利用することで、ステレオの出力の遅延に影響なく高速に現在の奥行きマップを出力できる。しかし、ステレオの遅延が大きいと、比較する映像間の相関が低くなり、推定される奥行きマップの精度が低下する可能性が高くなる。そのため、遅延補償アルゴリズムで高い奥行き推定精度を確保するためには、時系列の相関による映像間の対応精度を向上させるだけではなく、ステレオ法による高精度な奥行き推定と、ステレオ処理の遅延短縮も必要である。

以上の理由からステレオ部では、多眼ステレオ法を用いてシーンの物体輪郭が鮮明でかつ精度の高い奥行き情報の推定を行う。遅延補償部では、時系列の時間的空間的相関を利用して、現在の入力映像と過去の入力映像との対応を求め、得られた対応関係からステレオ法によって獲得した過去の奥行きマップを参照することで遅延補償を行う。

3.1 奥行きの遅延補償アルゴリズム

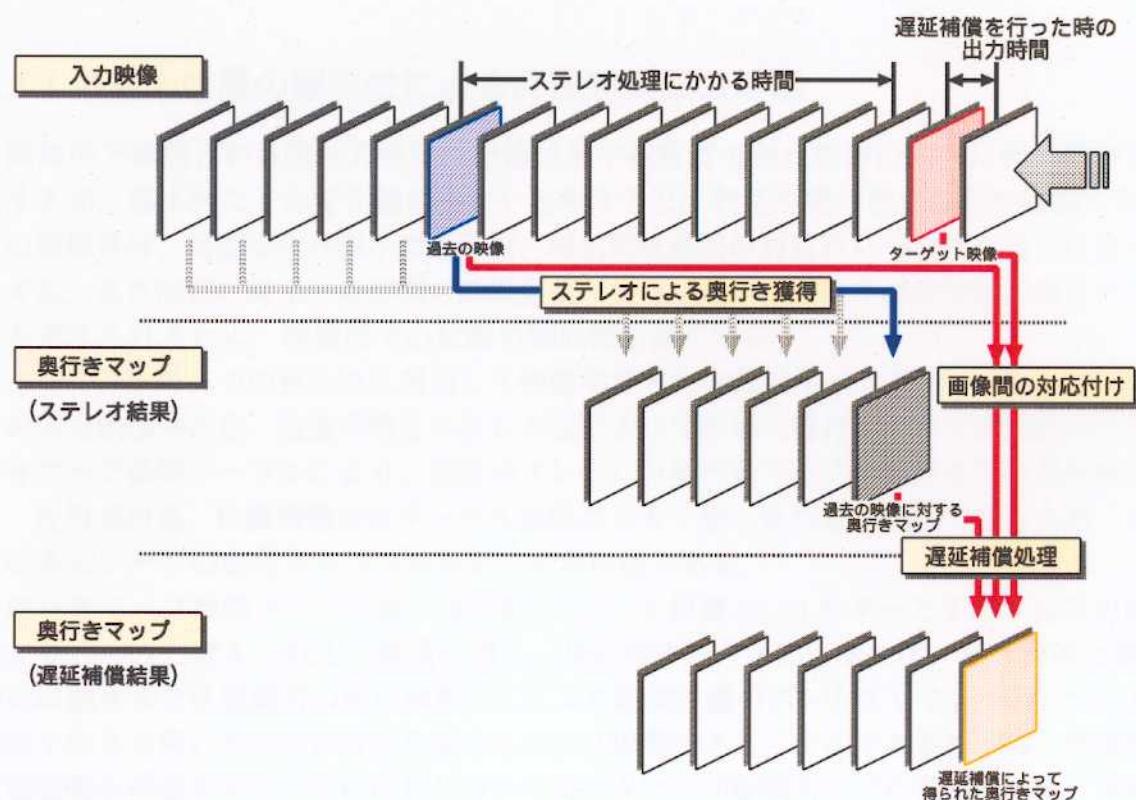


図 3.1: 遅延補償アルゴリズムを統合した奥行き獲得手法

3.2 時系列の相関を用いた遅延補償アルゴリズム

本節では、時系列の相関を利用して映像間の対応を求める手法について説明する。本研究では時系列映像中の対応点には、ステレオによる対応点探索での仮定と同様に特徴の類似性が成立するものとし、この類似性のもとづいて時系列映像間の対応点を高速に精度良く獲得する手法を提案する。遅延補償による奥行き推定精度のロバスト性を高めるため、時系列の対応を求める手法はシーンの状況によって最適なものを選択的に使用できるように複数の手法を用意する。これらの手法は高速であることが必須であるため、簡潔なアルゴリズムを用いて実現する。

3.2.1 色と位置の類似性による時系列映像の対応

映像中で観測される物体の色は、光源位置や観察者の視点位置の変化に伴い微妙に変化するが、基本的にその変化量は小さいと考えると、対応点間の色の類似性は高くなる。この類似性は、時間変化が微小であれば、時系列映像間の対応点についても当てはまるといえる。また同様に時系列映像間の時間変化が少ない場合、映像中の動物体の動きが小さいと考えられるため、映像間での位置の類似性も高くなる。

そこで、この二つの類似性に着目して映像間で対応点探索することなく、過去のフレームの入力映像中の色、位置情報とステレオ法によって求めた奥行きマップを対応付けた奥行きマップ参照テーブルにより、現在のフレームの奥行きマップを獲得する手法を提案する。注目点の色、位置情報からテーブル参照により一意に奥行きを選択できるため、高速に現在のシーンの奥行きマップを得ることが可能である。

奥行きマップ参照テーブルは、点の色(r, g, b)と位置(u, v)をキーとする5変数の関数 $T(u, v, r, g, b)$ と考えられる。前述の通り、時系列映像の対応点間には、色の微妙な変化、物体の動きまたは観測者の視点移動などにより観測位置のズレが生じるため、5つのキー値をそのまま用いて正しい対応を求めることは困難である。そこで本手法では、色情報と位置情報を再量子化し、それにもとづいて奥行きマップ参照テーブルを生成する。再量子化した情報をもとに生成されたテーブルを用いることで色の変化や物体の位置変化を吸収した対応点の決定が可能になる。本手法では、カメラ位置が固定である状況を仮定し、観測者の視点移動によって起こる映像中での物体位置の大きな変化はないものと仮定する。

奥行きマップ参照テーブルの生成は以下の手順で行う。まず、式(3.1)に従って、 n bit の色情報(r, g, b)を m bit, ($m \leq n$)の色情報(r', g', b')へ再量子化する。

$$(r', g', b') = \left(\frac{r}{2^{n-m}}, \frac{g}{2^{n-m}}, \frac{b}{2^{n-m}} \right) \quad (3.1)$$

次に映像中の点の座標(u, v)は、映像を適当な大きさのブロックに分割し、同一ブロックに含まれる点を一つの集合として扱うことで再量子化を実現する。座標(u, v)から座標(u', v')への変換を式(3.2)に示す。

$$(u', v') = \left(\frac{u}{B}, \frac{v}{B} \right) \quad (3.2)$$

ここで B は量子化ブロックサイズを表す。再量子化された色 (r', g', b') と位置 (u', v') と点 (u, v) における視差 d とを対応させる(式(3.3))。これを映像中の全ての点に対して行うことで、奥行きマップ参照テーブルを生成する。

$$T(u', v', r', g', b') \leftarrow d \quad (3.3)$$

現在のシーンの注目点 (u, v) の奥行き d の決定は、テーブルを生成した手順と同様にして注目点の位置・色情報を再量子化し、その値をキーとしてテーブルを参照することによって行う((3.4))。

$$d \leftarrow T\left(\frac{u}{B}, \frac{v}{B}, \frac{r}{2^{n-m}}, \frac{g}{2^{n-m}}, \frac{b}{2^{n-m}}\right) \quad (3.4)$$

時系列における色と物体位置の類似性に注目したこの手法は、アルゴリズムが簡潔であるにも関わらず、物体の移動が少なく対応点の色の変化が小さいシーンに対して精度良く奥行きマップを獲得することが可能である。しかし、この手法で吸収できる物体の移動量は量子化ブロックサイズによって決定される。そのため、物体の動きがこの量子化ブロックサイズを越えた場合、この手法で正しい対応点の奥行きを獲得することは不可能である。

量子化ブロックサイズを大きくすることで、より大きな物体の動きへの対応が可能になるが、量子化ブロックサイズを大きくする程位置の曖昧さが増加し、映像間の対応精度が低下する。すなわち、時系列映像間での物体の動きへの対応の度合と時系列映像間の対応点の精度の間にはトレードオフが生じる。本来、時系列映像間で精度の良い対応点を獲得することが目的であるため、物体の動きへの対応については、別の手法を用いて本手法を補うことが適当だと考えられる。

3.2.2 動き情報による時系列映像の対応

時系列映像間における物体の移動量を表すパラメータとして、MPEG[22][23]で用いられる動きベクトルがある。動きベクトルは、図3.2に示すように、映像をマクロブロックと呼ばれる小ブロックに分割し、この小ブロック単位で映像間の対応探索を行うことで得られる時系列映像間の対応ブロックの移動量である。物体の大きな動きの影響により前節で提案した手法で対応不可能となった点についてこの動きベクトルを利用して時系列映像間の物体の動きを補正する。

動きベクトル推定の手順を以下に示す。まず映像を $M \times N$ のマクロブロックに分割する。各ブロックに対して、式(3.5)で定義する評価関数に従ってマクロブロック間のブロックマッチングを行い、この評価関数を最小とする (k, l) をこのブロックの動きベクトル (\hat{k}, \hat{l}) とする(式(3.6))。

$$e(k, l) = \sum_{i,j} \sqrt{(I_{i,j}^{n-1} - I_{i+k,j+l}^n)^2} \quad (3.5)$$

$$(\hat{k}, \hat{l}) = \arg \min_{k,l} e(k, l) \quad (3.6)$$

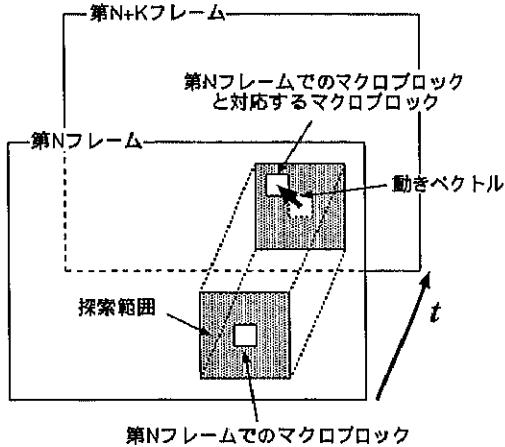


図 3.2: 動きベクトル

マクロブロックを前述の量子化ブロックサイズよりも小さいサイズに設定し、動きベクトルを求めることで、量子化したブロック内でさらに細かい動きの修正が可能になる。式(3.7)に従って修正した位置情報を再量子化(式(3.8))したものと同様に量子化した色情報をもとにテーブルを参照するようにアルゴリズムを変更する(式(3.9))。

$$(u, v) \Rightarrow (u - \hat{k}, v - \hat{l}) \quad (3.7)$$

$$(u', v') = \left(\frac{u - \hat{k}}{B}, \frac{v - \hat{l}}{B} \right) \quad (3.8)$$

$$d \leftarrow T(u', v', r', g', b') \quad (3.9)$$

3.2.3 視点位置の異なる時系列映像の対応

実時間ステレオシステムの応用として、HMDに装着されたカメラから撮影したシーンの奥行き情報を求め、実時間で観測者の見ているシーンへCG映像を合成する例が挙げられる。これまでには、カメラ位置を固定した状況でシーンを撮影することを前提として時系列映像間の対応を求める手法を提案した。本節では、上に挙げたようにカメラが観察者とともに移動するような状況への対応について考察する。

ある視点で観測された映像は、対応する3次元情報を用いて異なる視点からの見え方に変換することができる。従って、異なる視点で観測された画像間の対応を求める場合でも、一方の視点で観測された映像をもう一方の視点からの見え方に変換することで、提案した手法を修正せずに適用することが可能である。この変換を行うためには、観測されたシーンの3次元情報の他に、2視点間の相対的な位置と姿勢情報が必要となる。これらを画像処理により求めることも可能だが、全体の処理を高速化するためにカメラの位置と姿勢情報は、磁気センサやジャイロ等を用いて獲得されているものとする。

ステレオ法によって得られるシーンの3次元情報は、映像間での対応点どうしの見え方の変化を表したものであり、現実世界の3次元情報を直接に表現したものではない。そこで、映像の変換の方法として、視差情報から復元した現実の3次元情報をもとに変換を行う方法と現実の3次元情報を復元せずに直接視差情報から変換を行う方法について説明する。

3次元情報を用いた映像の変換

図3.3に示すような座標 O で観測される映像 I から、座標 $C(x_c, y_c, z_c)$ で xyz 軸まわりにそれぞれ $(\theta_x, \theta_y, \theta_z)$ 回転したカメラから観測される映像への変換を考える。説明の簡略化のために、図3.3では y 軸を省略し、2次元的に表現する。変換後の映像を I' 、 $I'(u', v')$ は点 (u', v') における物体の輝度値と定義し、以下に映像変換の手順を示す。

1. 点 (u', v') の点の視差を d と仮定すると、式(3.10)により、物体の3次元座標 (x', y', z') を復元する。

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \frac{b}{d} \begin{bmatrix} u' \\ v' \\ f \\ \alpha \end{bmatrix} \quad (3.10)$$

2. 回転行列 R (式(3.11))により回転成分を修正し、平行移動を行うことで、カメラ位置 O から見たシーンの3次元情報を得る(式(3.12))。

$$R = R(-\theta_y)R(-\theta_x)R(-\theta_z) \quad (3.11)$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \left[\begin{array}{c|cc} R & \begin{matrix} x_c \\ y_c \\ z_c \end{matrix} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \quad (3.12)$$

3. 上で求めた3次元情報 (x, y, z) を式(3.13)にしたがって映像中の座標 (u, v) に変換する。

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{\alpha z} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.13)$$

ここで、 b はステレオに用いたカメラの基線長、 f はカメラの焦点距離、 α は画像の単位[pixel]から現実の距離[mm]への変換パラメータを表す。

4. 点 (u, v) における視差が d と一致するならば、その点の輝度値 $I(u, v)$ を点 (u', v') の輝度値 $I'(u', v')$ として採用する。複数の対応点が候補に挙がった場合は、その中で最大の視差値をもつ点の輝度値を採用する。これにより、対応点の中で一番手前にあるものを選択していることになり、3次元的に矛盾のない表現を実現している。

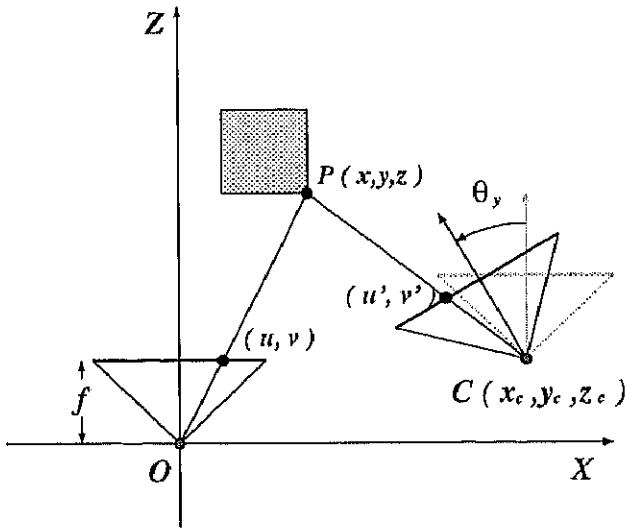


図 3.3: 映像の視点変換

視差を用いた映像の変換

カメラのレンズ中心を中心とした回転によって変化する映像の見え方の変化は、物体の奥行きに依存しない。一方カメラ位置変化に伴う映像の見え方は、物体の奥行きに依存して変化する。そこで、カメラの平行移動による見え方の変化と回転による見え方の変化を分離して求めた後に統合する。

視差はステレオ映像間の対応点の位置のズレを表現した値であるので、視差値から直接カメラの平行移動によって起こる見え方の変化を求めることが可能である。式(3.14), (3.15)に x, y 軸方向, z 軸方向に移動した時の映像中の座標の変換を表す。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{x_c d}{b} \\ 0 & 1 & \frac{y_c d}{b} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad (3.14)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{z - z_c}{z} & 0 & 0 \\ 0 & \frac{z - z_c}{z} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad (3.15)$$

$$z = \frac{bf}{\alpha d} \quad (3.16)$$

式(3.14), (3.15)をまとめると、式(3.17)のようになる。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{z - z_c}{z} & 0 & 0 \\ 0 & \frac{z - z_c}{z} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{x_c d}{b} \\ 0 & 1 & \frac{y_c d}{b} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad (3.17)$$

さらに, k , S を式(3.18), (3.19)と定義すると, 式(3.17)は式(3.20)のように表現できる。

$$k = \frac{z - z_c}{z} \quad (3.18)$$

$$S = \begin{bmatrix} 1 & 0 & \frac{x_n d}{b} \\ 0 & 1 & \frac{y_n d}{b} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = kS \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad (3.20)$$

また, x, y, z それぞれの軸まわりの回転は以下の式によって表現できる。

$$R(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{ftan\theta_x}{\alpha} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

$$R(\theta_y) = \begin{bmatrix} 1 & 0 & \frac{ftan\theta_y}{\alpha} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

$$R(\theta_z) = \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

したがって, 回転による見え方の変化は式(3.24)のように表現される。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = R(\theta_z)R(\theta_x)R(\theta_y) \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad (3.24)$$

平行移動に関する変換式(3.20)と回転に関する変換式(3.24)をまとめると式(3.25)になり, この式によって求めた (u', v') の対応点 (u, v) の輝度値 $I(u, v)$ を点 (u', v') の輝度値 $I'(u', v')$ とする。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = kR(\theta_z)R(\theta_x)R(\theta_y)S \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad (3.25)$$

映像の変換の方法として, 視差情報から復元した現実の3次元情報をもとに変換を行う方法と3次元情報を復元せずに視差情報をもとに変換を行う方法について説明した。前者の方法では, 映像中の座標と現実世界の3次元座標間の相互の変換の過程で誤差が含まれる可能性が高い。後者による変換は, 映像中の2次元座標から2次元座標への変換であるから, 前者の方法で生じるような誤差が含まれる可能性が少ない。従って, 後者の方法は本研究で使用する目的に適した方法だといえる。

3.3 汎用画像処理装置によるステレオシステムの構築

3.3.1 汎用画像処理装置の概要

本節では、システムを構成する汎用画像処理装置の概要について説明する。図 3.4 に汎用画像処理装置の構成を示す。装置はホストコンピュータ、カメラ、ビデオボード、DSP ボードによって構成される。ホストコンピュータは、DSP ボードとのデータの授受や DSP ボードで処理させるプログラムの転送を行う。ビデオボードは入力部と出力部からなり、入力部では、ビデオ信号の A/D 変換、DSP ボードへのデータ出力を行う。出力部では、データの D/A 変換および DSP ボードからのデータ入力をを行う。DSP ボードは、TI 社の C40 汎用 DSP を搭載した処理ボードであり、ビデオボードで変換された映像データもしくは、他の DSP ボードから転送されたデータをホストコンピュータから送られたプログラムに従って処理を行う。DSP ボードは、各ボードごとに CPU を持ち、他のボードと独立して動作する。システム全体の同期は、ビデオボードでビデオ信号から分離した垂直同期信号によって行う。

図 3.5 に垂直同期信号によるデータ通信の例を挙げる。ビデオボードと DSP ボード 2 枚が直列に接続されている構成を考える。ビデオ信号は奇数フィールドと偶数フィールドからなり、垂直同期信号はフレームの先頭で立上り、偶数フィールドの先頭で立ち下がる信号である。信号の立ち上がる間隔は 1/30 秒であり、以後この間隔をフレーム時間と呼ぶ。DSP ボードは 1 フレーム時間間に前段のボードから送られたデータを蓄積するが同一フレーム時間内に処理することはできない。そのため、処理は次のフレーム時間に行われる。従って図 3.5 では、ビデオ信号 A は、DSP ボード α によってビデオボードが信号を送った次のフレーム時間に処理され、同一フレーム時間内に DSP ボード β へと転送される。このフレーム時間はビデオボードにより 1/30 秒から 255/30 秒まで 1/30 秒間隔で変更することができる。

ビデオボード、DSP ボード間のデータ転送は、ISHTAR バスとブロードキャストバスの 2 種類の専用バスによって行う。これらのバスは、それぞれ仕様が異なるので、用途に合わせて使い分ける必要がある。ISHTAR バスは 32bit のバスで、ボード間で 1 対 1 の 1 方

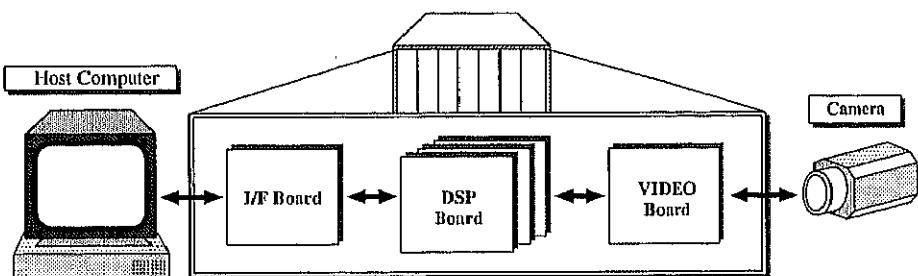


図 3.4: 汎用画像処理装置

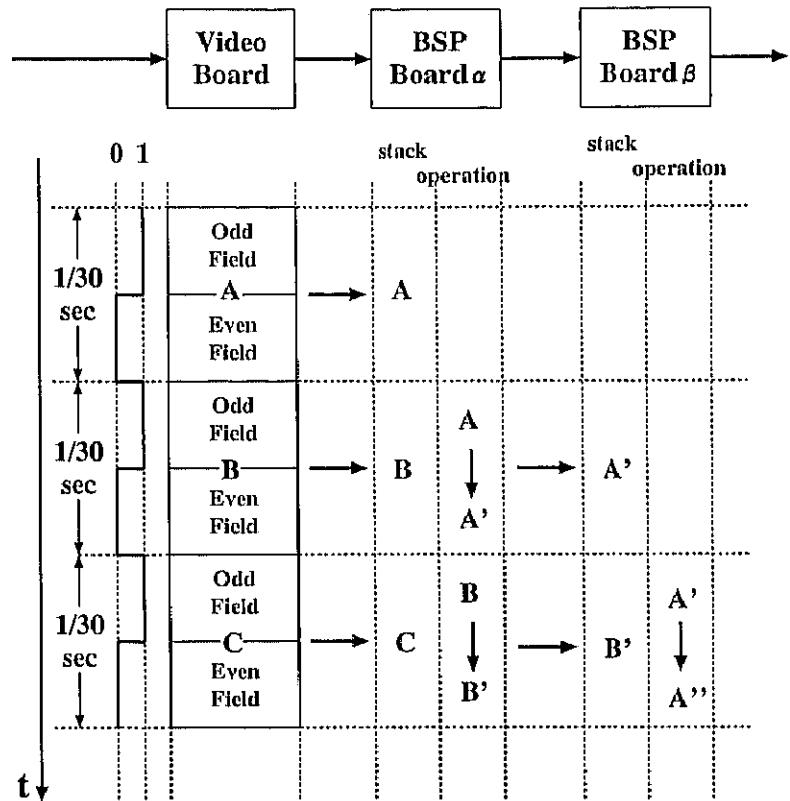


図 3.5: 垂直同期信号による制御

向の転送を行う。ブロードキャストバスは 32bit の双方向バスで、ボード間で 1 対多、多対多のデータ転送が可能である。また、32bit のバスとしてではなく 8bit 4 本のバスとしても使用できる。以後ブロードキャストバスを BC バスと省略して呼ぶ。図 3.6 にそれぞれのバスの接続例を示す。

DSP ボードは、 1024×256 画素のフレームメモリを持ち、1 画素当たりのデータ量は 32bit である。また、ダブルバッファリングが可能であり、高速な処理が行える。DSP からの出力は自動的に行われ、他のボードとの同期をハードウェアで行っているため、ソフトウェアで同期の考慮をする必要がなく、各 DSP が独立に動作できるためデータ転送のオーバーヘッドが軽減され、高速処理が可能である。

3.3.2 ステレオシステムの構築

図 3.7 に構築したステレオシステムを示す。シーンの撮影は、格子状に配置した 4 台のカメラユニットによって行う。カメラユニットで撮影した画像を各 DSP ボードで処理し、視差推定結果をモニターに出力する。図 3.8 に汎用画像処理装置による多眼ステレオ法と遅延補償アルゴリズムを組み合わせた実時間ステレオシステムの構成を示す。システムはステレオ部と遅延補償部から構成され、ステレオ部によって推定された過去のシーンの奥

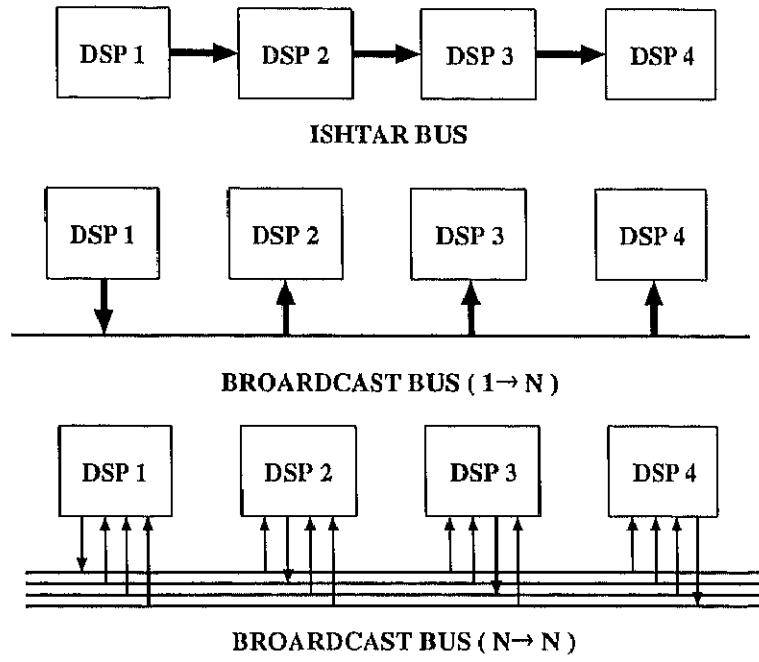


図 3.6: バスの接続例

行き情報と、遅延補償部によって求められた過去のシーンと現在のシーンの対応関係とともに現在のシーンの奥行き情報を推定する。

入力系は、4台のカメラを格子上に配置したカメラユニットを用いる。各カメラからの出力は、4分割ユニットを通じて1つの映像信号に合成され、ステレオ部と遅延補償部のそれぞれのユニットに入力される。4台のカメラからの映像を4分割ユニットによって1つの映像信号に合成することで、4台のカメラからの映像を1枚のビデオボードに入力することが可能になる。それぞれのカメラからの出力は、 $640 \times 480[\text{pixel}]$ であるが、4分割ユニット内で各映像は $1/2(320 \times 240[\text{pixel}])$ に縮小される。さらにDSPボード内では、一方のフィールドのみがバッファに蓄積される。これに合わせて水平方向のサイズも $1/2$ に縮小する。最終的に処理される画像の大きさは $160 \times 120[\text{pixel}]$ である。ステレオ部に入力された映像は、空間的に分割されステレオ処理される。遅延補償部では、ステレオ部からの出力である視差情報とステレオ画像から視差参照テーブルを作成し、4分割ユニットからの出力映像と視差参照テーブルとの対応から現在のシーンの視差情報を推定する。各ユニット内の処理の詳細は以降の節で紹介する。

3.3.3 システムの詳細

ステレオ部(図3.8左上のブロック)では、格子状に配置されたカメラによって撮影された4枚の画像を用いて4眼ステレオを行う。ステレオ部は、各カメラで撮影された画像間の位置補正を行うDSPボードAとステレオ演算を行うDSPボードB,C,Dによって構成

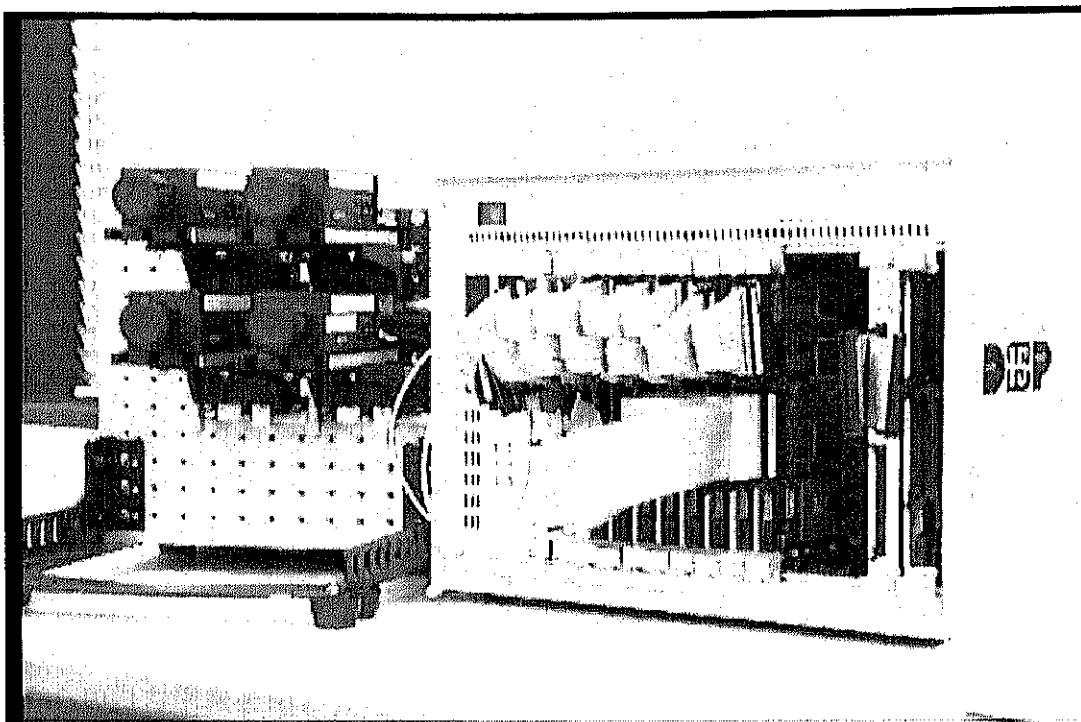


図 3.7: 実時間ステレオシステム(外観)

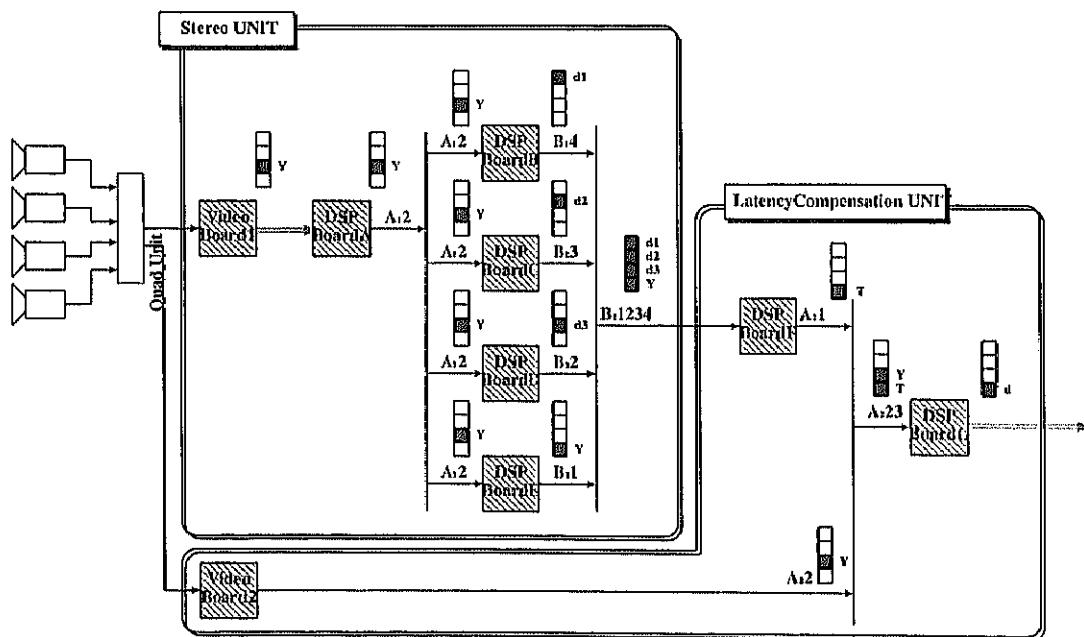


図 3.8: 実時間ステレオシステム(構成)

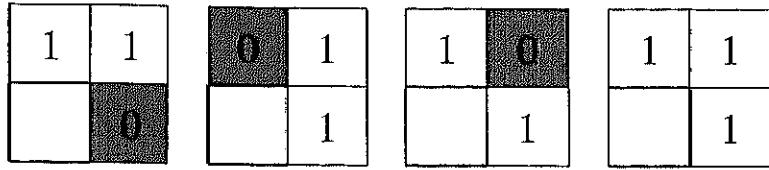


図 3.9: 4 眼ステレオ用に設定したマスクパターン

される。DSP ボード A では、各カメラの CCD のズレを修正し各画像間のエピポーラ線の整合を行う。位置の修正は、各画像中の画素を $X - Y$ 平面に対して平行移動させることで行う。DSP ボード B,C,D では、画像を空間的に 3 分割し、各々の領域に対して対応点探索を行うことで、各 DSP ボードで行う処理の分散を図っている。

対応点探索アルゴリズムは、隠れに対応可能な多眼ステレオ法を利用し、隠れを考慮した処理を行っている。本アルゴリズムで用いた隠れのマスクパターンを図 3.9 に示す。

遅延補償部(図 3.8 右下ブロック)では、視差参照テーブルの生成(DSP ボード F)とテーブル参照による視差推定(DSP ボード G)の 2 つの処理を行う。DSP ボード F では、ステレオ部からの出力情報を用いて視差参照テーブル生成を行う。ステレオ部からの出力には、ステレオ処理に用いた入力映像とそれに対応する視差情報が含まれている。入力画像の輝度値を再量子化し、その値をもとにメモリ上のテーブル座標を計算する。算出されたメモリ位置に対応する視差情報を格納することで、視差参照テーブルを生成する。

DSP ボード G では、DSP ボード F からの出力である視差参照テーブルをもとに、ビデオボードから入力される現在のシーンの視差情報を決定する。

3.3.4 計算量の見積り

N 眼ステレオおよび遅延補償アルゴリズムの計算量を見積ることで、理論上の実行処理時間の検討を行う。大きさ $X \times Y$ の画像に対して、ブロックマッチングのブロック数 W 、仮定視差数 d 、隠れマスクの数 M を用いた時のステレオ処理に必要な演算量を計算すると、

- 画像の読み込み

$$NXY \quad (3.26)$$

- 評価値の計算
 - 注目画素を中心とした矩形領域画素の読み込み

$$NW^2 \quad (3.27)$$

- 差分・絶対値計算

$$2(N-1)W^2 \quad (3.28)$$

- 差分値の総和

$$(N - 1)(W^2 - 1) \quad (3.29)$$

これらをすべての画素について、仮定視差数回繰返し、最小値決定を行う。

$$XY[\{(3N - 2)W^2 - N + 2\}d - 1] \quad (3.30)$$

- 視差値の出力

$$XY \quad (3.31)$$

式(3.26), (3.30), (3.31)よりステレオ処理に必要な演算数は次式で表される。

$$XY[\{(3N - 2)W^2 - N + 2\}d + N] \quad (3.32)$$

隅れを考慮した視差推定を行う場合、式(3.29)が式(3.33)となり、全体の演算量は式(4.21)となる。

$$(N - 1)(W^2 - 1)M \quad (3.33)$$

$$XY[\{(M + 3)(N - 1) + 1\}W^2 - M(N - 1) + 1]d + N \quad (3.34)$$

遅延補償アルゴリズムの計算量は、

- 視差参照テーブルの生成

- 画像の読み込み

$$XY \quad (3.35)$$

- テーブル生成(画素アクセス+量子化+テーブル座標計算+書き出し)

$$6XY \quad (3.36)$$

- テーブルの出力

$$XY \quad (3.37)$$

- 視差参照テーブルの参照

- 画像の読み込み

$$XY \quad (3.38)$$

- テーブル参照(画素アクセス+量子化+テーブル座標計算)

$$5XY \quad (3.39)$$

- テーブルの出力

$$XY \quad (3.40)$$

したがって、遅延補償アルゴリズムの総演算量は、次式で表される。

$$15XY \quad (3.41)$$

3.4 実験結果

3.4.1 色と位置の類似性を利用した奥行き獲得

3.2.1節で提案した色と位置の類似性を利用した時系列映像の対応による奥行き獲得手法についてステレオ動画シーケンスを用いたシミュレーション実験を行った。色の量子化ビット数、ブロックサイズを変化させて時系列映像の対応精度の評価を行った。実験に使用した動画像は、格子状に 3×3 台配置したカメラによって撮影された9眼の動画シーケンスである。各時刻におけるシーンの奥行き獲得は多眼ステレオ法SEAを用いて行った。

対応精度の評価は、ステレオ法によって獲得した奥行きマップを正解データとし、このデータとの一致度を調べることで行った。ブロックサイズを変化させたときの時系列画像間の対応精度は、ブロックサイズが小さい程正答率が高いことが確認できた。また、ブロックサイズが大きくなる程、推定不能としてリジェクトされる点の数が減少した。これは、ブロックサイズを大きくすることで、物体の動きに対する対応精度が向上したためと考えられる。色の量子化ビット数による対応精度は、量子化ビット数が大きくなるにつれて誤り率は低下する。しかし、正答率も同様に低下し、リジェクト率が増加する。この傾向は、量子化ビット数の増加につれて対応の曖昧さが減少する一方で、対応点間の輝度値の変化を吸収できずに対応点が決定できない点が増加することによるものである。表3.1に各パラメータによる対応の評価を示す。入力画像の一部を図3.10に示す。また、多眼ステレオ法SEAによる視差推定結果および、本手法による視差推定結果とその比較を図3.11に示す。

3.4.2 動き情報による位置情報の修正

色と位置情報の類似性を用いた手法における物体の動きへの対応精度を向上させるために、動きベクトルを利用して物体の位置情報を修正する実験を行った。動きベクトルの推定は、マクロブロックの大きさを $8 \times 8[\text{pixel}]$ 、探索範囲を $16 \times 16[\text{pixel}]$ として行った。

動き情報を用いた物体位置の修正を行わない場合と比較して、動物体付近の視差推定結果が向上した。図3.12に物体位置の修正の有無による奥行きマップ推定結果の比較を示す。

表 3.1: 量子化パラメータの変化による対応精度の評価(正答率/誤り率/リジェクト率)

	量子化 bit 数 1	量子化 bit 数 2	量子化 bit 数 3	量子化 bit 数 4
ブロックサイズ 4	85.4/11.3/3.3	83.7/10.2/6.1	79.0/8.9/12.1	66.2/7.0/26.8
ブロックサイズ 8	82.5/16.0/1.5	83.0/13.9/3.1	81.2/11.7/7.0	72.8/9.2/18.0
ブロックサイズ 16	77.9/21.8/0.3	80.2/18.3/1.5	81.0/15.1/3.9	76.9/11.9/11.2
ブロックサイズ 32	70.1/29.8/0.1	75.2/24.2/0.6	78.2/20.0/1.8	78.9/15.3/5.8

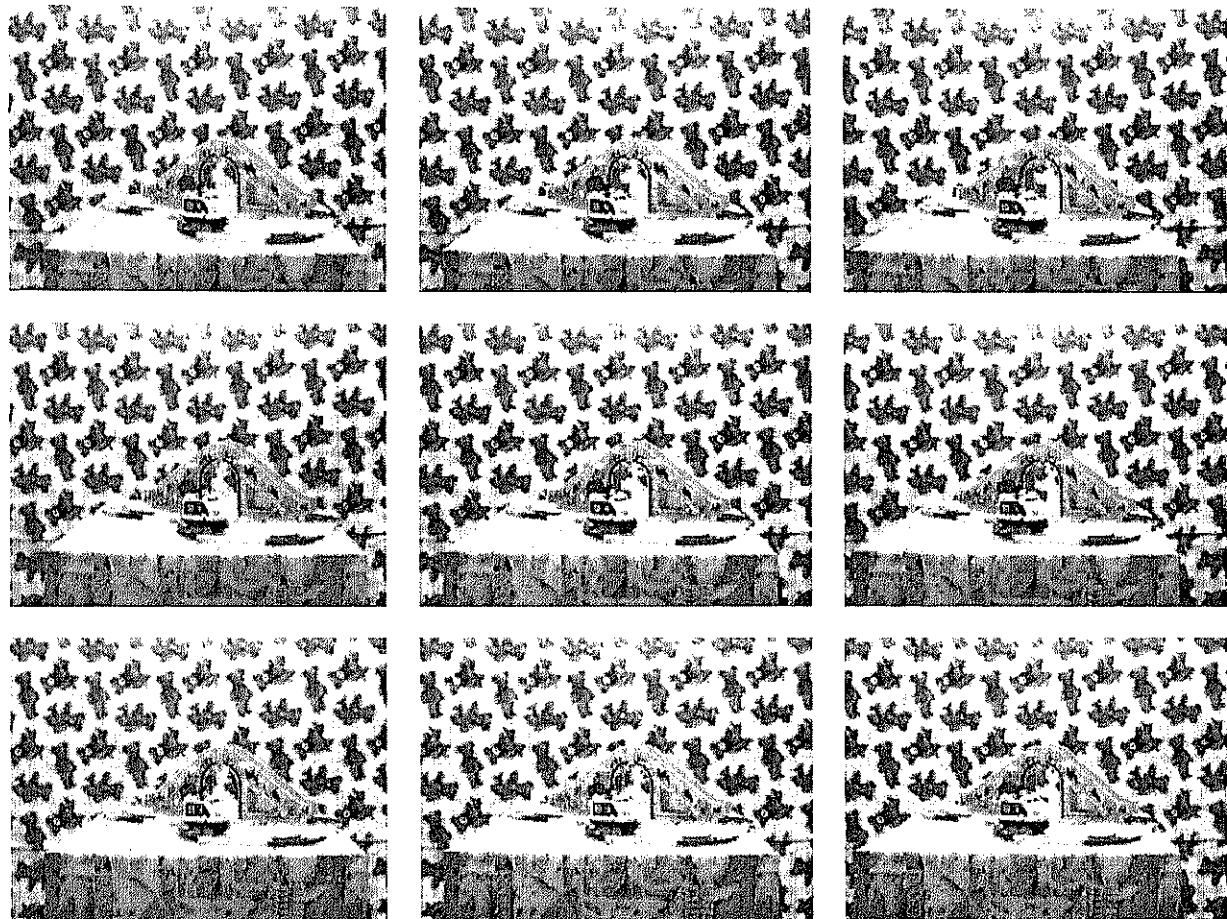


図 3.10: 実験に使用した 9 眼マトリクス配置のカメラからの映像



(a) 多眼ステレオ法 SEA による奥行きマップ

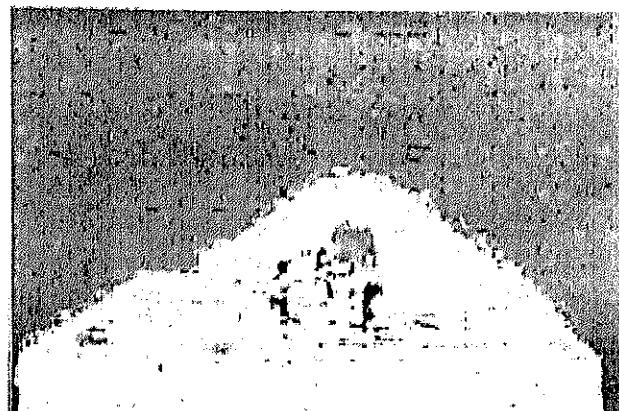


(b) 提案手法を用いて獲得した奥行きマップ

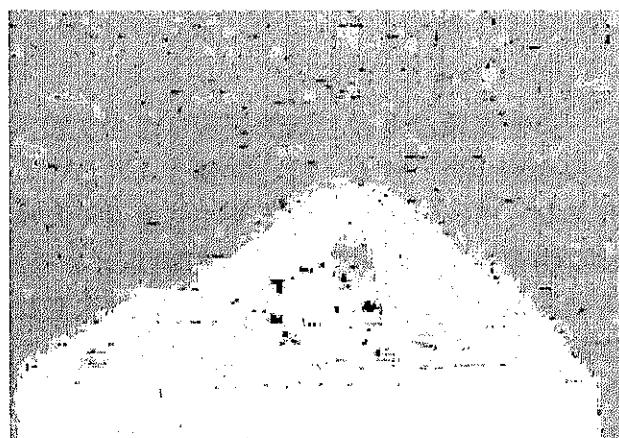


(c) (a) と (b) の差分結果

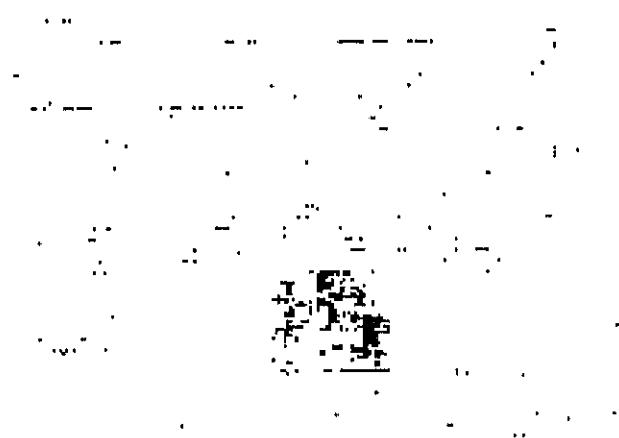
図 3.11: 遅延補償アルゴリズムによる奥行きマップ獲得結果



(a) 物体位置の修正を行わない場合の奥行きマップ



(b) 物体位置の修正を行った場合の奥行きマップ



(c) (a) と (b) の差分結果

図 3.12: 物体位置の修正を行った場合の奥行きマップ獲得結果

3.4.3 汎用画像処理装置による奥行き獲得

汎用画像処理装置を用いて構築したステレオシステムによる視差推定実験を行った。実験に使用したパラメータから得られるシステムの計算量は、ステレオ部 43,161,600 演算、遅延補償部 288,000 演算である。表 3.2 に構築したシステムの理論的な性能と実際の性能の比較を示す。ステレオ部での処理時間がおよそ 1.3 秒である。ステレオ部全体では、3 段のパイプライン処理が行われているので 4 秒近い遅延が生じる。遅延補償部では全て 1/30 秒以内で処理が行えることが確認できた。パイプライン処理は 2 段であるから、遅延は 2/30 秒である。遅延補償部での遅延がシステムの遅延であるから、本システムは、ビデオレートでかつ遅延時間 2/30 秒で動作することが確認できた。付録 A に使用したカメラヘッドの詳細、付録 B に実験に用いた各パラメータを示す。

3.5 考察

本章では、ステレオ処理で生じる遅延問題に対する解決法の一つとして、時系列の相関を利用した遅延補償アルゴリズムを提案した。また、汎用画像処理装置を用いて提案したアルゴリズムを使用したステレオシステムの構築を行った。時系列間の物体の色と位置情報の類似性を利用してシーンの対応を求めるアルゴリズムを提案し、高速な遅延補償を実現した。提案したアルゴリズムでは、動きの大きな物体の対応が正しく求められない問題があった。これに対して、動きベクトルを利用した位置修正手法を提案し、映像間でおおきな動きがある物体でも精度良く対応が求められることを確認した。更に、ステレオ法によって求めたシーンの奥行き情報を利用して異なる視点における画像を生成することにより、カメラ位置が変化する場合への対応について検討した。汎用画像処理装置を用いてステレオシステムの構築を行い、提案したアルゴリズムの有効性とシステムの性能の考察を行った。汎用画像処理装置を用いたシステムにより、ビデオレートで動作し、かつ、遅延時間が 2/30 秒である実時間ステレオを実現した。時系列の相関を利用した遅延補償アルゴリズムのロバスト性の向上、高速な画像処理装置を用いたシステムの高速化が今後の課題である。

表 3.2: システムの性能評価

	処理	演算数 (理論値)	処理時間 [s] (理論値)	処理時間 [s] (実測値)
DSP-A	位置補正	1,001,954	0.040	$0.033 < t \leq 0.066$
DSP-B,C,D	ステレオ処理	23,910,861	0.956	$1.2 < t \leq 1.233$
DSP-E	スルー	61,035	0.002	$t \leq 0.033$
DSP-F	テーブル生成	297,635	0.012	$t \leq 0.033$
DSP-G	テーブル参照	273,279	0.011	$t \leq 0.033$