

## 非同期ステレオ動画像を用いた動的シーンの 位置とモーションの同時推定

関 晃仁 奥富 正敏

東京工業大学 大学院理工学研究科

**概要** 本論文では、非同期ステレオから得られる時系列画像を利用して、動的なシーンにおける各点の3次元位置と3次元モーションを推定する手法を提案する。通常、3次元位置とその3次元モーションを推定するためには、ステレオ対応点探索とトラッキングを別々に行なう必要があった。この方法は、ステレオカメラが同期している場合には適用可能ではあるが、非同期ステレオでは移動物体が写っていたり、カメラが移動していたりすると、各カメラで撮影されるタイミングの違いによって対象の3次元位置が異なるため、適用することは困難であった。そこで、提案手法ではステレオ対応点探索とトラッキングに相当する作業を同時に行なうことで、3次元位置と3次元モーションに加え、非同期ステレオカメラ間の撮影タイミングのずれである時間オフセットをも同時に最適化して推定する。最後に合成画像と実画像に対して提案手法を適用し、その有効性を示す。

### Simultaneous Estimation of 3D Position and Motion in Dynamic Scene Using Unsynchronized Stereo Image Sequence

Akihito Seki and Masatoshi Okutomi

Graduate School of Science and Engineering,  
Tokyo Institute of Technology

**Abstract** In this paper, we propose a simultaneous estimation method of structure and motion in dynamic scenes. Usual methods for getting structure and motion using stereo cameras need two kinds of operations: stereo correspondence and tracking. Therefore, we had to separately get correspondence between stereo images and sequential images. Our first contribution is the method of corresponding all stereo images and sequential images at once. On the other hand, most of stereo correspondence algorithms are limited under synchronized status. In the stereo rig by using unsynchronized cameras, the structure can't be obtained by stereo correspondence and triangulation because of unknown time offset between cameras. Then, our second contribution is the method of estimating structure, motion, and time offset simultaneously by using unsynchronized stereo cameras. This is done by taking advantage of the first contribution scheme. Finally, we present some experimental results.

## 1 はじめに

ロボットや車両などの移動体にステレオカメラを設置して、ステレオ計測やモーション推定を行うことで、3次元再構成、エゴモーション推定、空間情報を利用したセグメンテーション、自己位置推定や視覚誘導、3次元地図の生成、さらに障害物の検出や回避などに利用できる。

3次元位置とそのモーションの推定問題は、一般的な環境を考えると2つの場合に分けられる。1つめは、カメラに写るモーションが単一の場合である。例えば、静止しているシーン中をカメラが移動する場合がこれ

に相当し、カメラに対する全体のモーションと画像内の各点ごとの3次元位置を推定する [12, 4, 2, 3, 6]。2つめは、シーン中に複数の異なるモーションが存在する場合である。最も単純な方法は、ある時刻でステレオ対応点探索を行なうことで3次元位置を得、次にその点を次時刻の画像上でトラッキングして時系列間で対応づける。さらに、対応づけられた次時刻の点のステレオ対応点探索を行えば、各時刻の3次元位置とモーションを求めることができる。筆者らは、これに対して、画像の各点ごとに3次元位置とモーションを同時に最適化して推定する手法を提案している [7]。

これらの手法は、いずれも同期ステレオカメラを仮

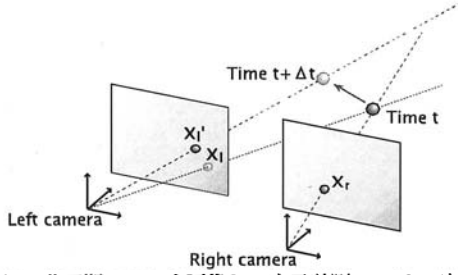


図 1: 非同期ステレオ画像と 3次元計測.  $x_r$  は、時刻  $t$  で右画像に投影された位置 ( $x_l$  は同時刻の左画像),  $x'_l$  は時刻  $t + \Delta t$  で左画像に投影された位置.  $x_r$  を通る光線と  $x'_l$  を通る光線が交差しなかったり、異なった位置に交差する.

定した手法である. 一方, 安価な市販カメラを利用した場合には, 両カメラで同一時刻に撮影ができない非同期ステレオカメラであることが多い. また, 形式の異なるカメラのように物理的に同期ができない場合も考えられる. このようなカメラを用いて, 同期ステレオカメラと同様にステレオ計測やモーション推定ができれば, 安価で同期をとるための特別な装置の不要な, カメラを選ばないシステムが実現できる. しかし, 非同期ステレオでは移動物体が画像中に写っていたり, カメラが移動していたりすると, 図 1 に示すように各カメラで撮影されるタイミングの違いによって対象の 3次元位置が異なるため, そのままでは 3次元位置やモーションを正しく推定することができない.

Svedman らは非同期カメラを使って, 特徴点の奥行きを推定する手法を提案している [13]. また, 清水らは, 非同期カメラを使って, 1 台のカメラのフレームレート以上の頻度で奥行きを推定する手法を提案している [9, 10]. しかし, これらの方法では, 非同期カメラの時間オフセットは既知としている.

Zhou らは, あらかじめ非同期画像間で対応のとれた 4 点の位置とエピポラ幾何によって, 時間オフセットを推定した後, 改めて時系列画像間でオプティカルフローを求め, 推定されたフローを時間オフセットに応じて, 同期ステレオ画像になるように画素ごとの位置と輝度をブレンディングして仮想的に同期画像を生成する. 生成された同期ステレオ画像を用いて, 従来のステレオ対応点探索を行なうことで, 奥行きを推定した [14].

一方, 提案手法では, 動的なシーンに対して非同期ステレオカメラで撮影された画像をそのまま利用して, 時間オフセットと各点の 3次元位置と 3次元モーションをすべて同時に最適化を行なう.

本論文では, まず非同期ステレオ画像間の対応関係

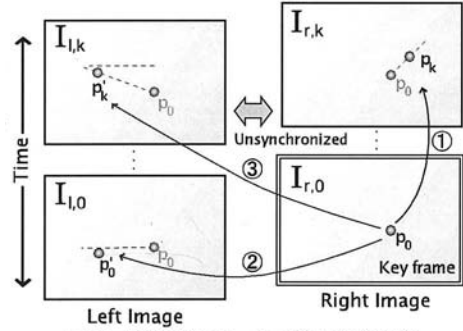


図 2: 時系列ステレオ画像の対応関係

について定式化を行ない, 次に時間オフセットと 3次元モーションと 3次元位置を同時に最適化して推定する方法について述べる. 最後に合成画像と 2 台の異なるカメラを利用した実画像を用いた実験結果について述べる.

## 2 画像間の関係

図 2 は, キーフレーム (本論文では右画像に設定)  $I_{r,0}$  と時系列ステレオ画像間の関係を表している.

本研究では, 撮影間隔の等しい非同期ステレオカメラで動的シーンを撮影しており, カメラの外部パラメータおよび内部パラメータはあらかじめ求まっていることを想定している. また, 推定するフレーム間の各点のモーションは等速直線運動と仮定する.

### 2.1 同一カメラの画像間の関係

同一カメラで異なる時刻に撮影された画像間の関係について述べる. この関係は, 図 2 の ① である. 空間中の奥行きが  $Z$  の点が画像に投影される位置を  $\mathbf{p}_0 = [x, y]^T$  とし, その点が 3D モーション  $\mathbf{T}_m = [T_{m,x}, T_{m,y}, T_{m,z}]^T$  によって変化し,  $k$  フレーム後 (前) の画像  $I_{r,k}$  中に写る点を  $\mathbf{p}_k$  とすると,

$$\mathbf{p}_k = \mathbf{p}_0 + \frac{k\mathbf{A}_x\mathbf{T}_m}{Z - kT_{m,z}} \quad (1)$$

と表される. ここで,

$$\mathbf{A}_x = \begin{bmatrix} -f & 0 & \hat{x} \\ 0 & -af & \hat{y} \end{bmatrix}, \quad \hat{x} = x - u_0, \quad \hat{y} = y - v_0 \quad (2)$$

である. また, カメラの内部パラメータは,

$$\mathbf{A} = \begin{bmatrix} f & 0 & u_0 \\ 0 & af & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

で表すことができる. 右画像には, モーションに依存するエピポラ拘束が存在する. このエピポラ拘束は,

モーションが未知であるので決まっていないが、モーションが決まると1~k番目のすべての右画像のエピポーラ拘束が一意的に決定される。

## 2.2 異なるカメラの画像間の関係

次に、異なるカメラの画像間について述べる。まず、同図②の関係について説明する。ステレオ画像間の時間オフセットを $\lambda$ と表したとき、 $\mathbf{p}_0$ が左画像 $\mathbf{I}_{l,0}$ に写る位置を $\mathbf{p}'_0$ とすると、

$$\mathbf{p}'_0 = \mathbf{p}_0 + \frac{\mathbf{A}_x\{\lambda\mathbf{T}_m + \mathbf{T}_s\}}{Z - \lambda T_{mz}} \quad (4)$$

で表される。ここで、 $\mathbf{T}_s$ はステレオカメラ間の並進ベクトルである。

次に、同図③の関係については、kフレーム後(前)の画像 $\mathbf{I}_{l,k}$ に対応する位置は、

$$\mathbf{p}'_k = \mathbf{p}_0 + \frac{\mathbf{A}_x\{(k+\lambda)\mathbf{T}_m + \mathbf{T}_s\}}{Z - (k+\lambda)T_{mz}} \quad (5)$$

として表現できる。

## 3 同時推定アルゴリズム

本節では、前節で述べた非同期画像間を利用して、まずコスト関数を定義し、次にそのコスト関数の解法について述べる。

### 3.1 コスト関数

各画像におけるキーフレームの点 $\mathbf{p}_0$ の対応位置は、点 $\mathbf{p}_0$ の回りに小領域 $W$ を考え、全画像でその小領域内の各画素が最も合う位置とする。そこで、コスト関数 $E$ を以下のように定義し、コスト関数が最小になるモーションと奥行きと時間オフセットを含むパラメータ $\mathbf{m}$ を求める。

$$E(\mathbf{m}) = \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{I}_{l,k}(\mathbf{x} + \mathbf{u}_k) - \mathbf{I}_{r,0}(\mathbf{x})]^2 + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{I}_{r,k}(\mathbf{x} + \mathbf{u}'_k) - \mathbf{I}_{r,0}(\mathbf{x})]^2 \quad (6)$$

ここで、 $\mathbf{m} = [T_{mx}, T_{my}, T_{mz}, Z, \lambda]^T$ が未知数であり、 $\mathbf{I}_*(\mathbf{x})$ は画像 $\mathbf{I}_*$ の位置 $\mathbf{x} = [x, y]^T$ における画像の輝度を表し、

$$\mathbf{u}_k = \frac{\mathbf{A}_x\{(k+\lambda)\mathbf{T}_m + \mathbf{T}_s\}}{Z - (k+\lambda)T_{mz}}, \quad \mathbf{u}'_k = \frac{k\mathbf{A}_x\mathbf{T}_m}{Z - kT_{mz}} \quad (7)$$

である。式(6)の前半は異なるカメラの時系列画像間のSSSD(sum of SSDs)を表し、後半は同一カメラの

時系列画像間のSSSDを意味している。つまり、式(6)は、マルチベースラインステレオ[5]の考え方を非同期ステレオ画像間に当てはめたものと考えられることができる。

### 3.2 コスト関数の最小化

コスト関数を最小化するために、式(6)の近似をする。

まず、微小な要素を持つ行列 $\mathbf{D}$ を用い、座標変換 $\tilde{\mathbf{x}}'' \sim (\mathbf{I} + \mathbf{D})\tilde{\mathbf{x}}$ を考える。ここで、 $\tilde{\mathbf{x}}$ は $\mathbf{x}$ の同次座標表現とし、 $\mathbf{I}$ は3次の単位ベクトルである。 $\tilde{\mathbf{I}}_{l,k}(\mathbf{x}'')$ \*について座標 $\mathbf{x}$ の周りで1次のテイラー展開をすると、

$$\begin{aligned} \mathbf{I}_{l,k}(\mathbf{x} + \mathbf{u}_k) - \mathbf{I}_{r,0}(\mathbf{x}) \\ = \tilde{\mathbf{I}}_{l,k}(\mathbf{x}'') - \mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k} \end{aligned} \quad (8)$$

が得られる。ここで、 $\mathbf{g}_{l,k}^T$ は $\tilde{\mathbf{I}}_{l,k}$ の画像勾配、 $\mathbf{J}_{l,k}^T$ はJacobian、 $e_{lr,k}$ は $\tilde{\mathbf{I}}_{l,k}$ と $\mathbf{I}_{r,0}$ の輝度差分である。なお、詳しい式展開は付録Aにあり、Jacobianは式(36)として求められる。

前述した手順と同様に計算すると、

$$\begin{aligned} \mathbf{I}_{r,k}(\mathbf{x} + \mathbf{u}'_k) - \mathbf{I}_{r,0}(\mathbf{x}) \\ = \tilde{\mathbf{I}}_{r,k}(\mathbf{x}'') - \mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \mathbf{T}^T \Delta \mathbf{m} + e_{rr,k} \end{aligned} \quad (9)$$

が得られる。ここで、 $\tilde{\mathbf{I}}_{r,k}(\mathbf{x})$ 、 $\mathbf{g}_{r,k}^T$ 、 $e_{rr,k}$ は式(8)の画像 $\mathbf{I}_{l,k}$ が画像 $\mathbf{I}_{r,k}$ に置き換わったものである。なお、詳しい式展開は付録Bにあり、Jacobianである $\mathbf{J}_{r,k}^T$ は式(37)、 $\mathbf{T}$ は式(41)としてそれぞれ求めることができる。

非同期ステレオ画像から奥行きとモーションと時間オフセットを求めるには、最低2ペア計4枚の時系列ステレオ画像と推定したい点が1つ必要である。以下では、まず1点を利用したコスト関数の最小化方法について述べ、続いてより一般的な場合である、複数点を同時に使って、コスト関数を最小化する方法を述べる。

#### 3.2.1 1点についての解法

点 $\mathbf{p}_0$ の1つだけを考えると、式(8),(9)を考慮して、コスト関数 $E$ は以下ようになる。

$$\begin{aligned} E(\Delta \mathbf{m}) = \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k}]^2 \\ + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \mathbf{T}^T \Delta \mathbf{m} + e_{rr,k}]^2 \end{aligned} \quad (10)$$

\* $\tilde{\mathbf{I}}_{l,k}(\mathbf{x})$ は $\mathbf{I}_{l,k}$ を座標 $(\mathbf{x} + \mathbf{u}_k)$ でサブサンプリングして生成した画像である。

さて、上式を最小化するために、コスト関数  $E$  を  $\Delta \mathbf{m}$  で偏微分して、

$$\frac{\partial E}{\partial \Delta \mathbf{m}} = \sum_{k=0}^n \sum_{\mathbf{x} \in W} \left[ \mathbf{J}_{l,k} \mathbf{g}_{l,k} \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k} \mathbf{J}_{l,k} \mathbf{g}_{l,k} \right] + \sum_{k=1}^n \sum_{\mathbf{x} \in W} \left[ \mathbf{T} \mathbf{J}_{r,k} \mathbf{g}_{r,k} \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \mathbf{T}^T \Delta \mathbf{m} + e_{rr,k} \mathbf{T} \mathbf{J}_{r,k} \mathbf{g}_{r,k} \right] = 0 \quad (11)$$

この式を整理すると、

$$\mathbf{A} \Delta \mathbf{m} = -\mathbf{b} \quad (12)$$

となる。ここで、

$$\begin{aligned} \mathbf{A} &= \sum_{k=0}^n \sum_{\mathbf{x} \in W} \mathbf{J}_{l,k} \mathbf{g}_{l,k} \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \\ &\quad + \sum_{k=1}^n \sum_{\mathbf{x} \in W} \mathbf{T} \mathbf{J}_{r,k} \mathbf{g}_{r,k} \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \mathbf{T}^T \\ \mathbf{b} &= \sum_{k=0}^n \sum_{\mathbf{x} \in W} e_{lr,k} \mathbf{J}_{l,k} \mathbf{g}_{l,k} \\ &\quad + \sum_{k=1}^n \sum_{\mathbf{x} \in W} e_{rr,k} \mathbf{T} \mathbf{J}_{r,k} \mathbf{g}_{r,k} \end{aligned} \quad (13)$$

と表される。式 (12) より  $\Delta \mathbf{m}$  を求め、さらにコスト関数を最小にするために  $\Delta \mathbf{m}$  を繰り返し更新する。その際のパラメータの更新は

$$\mathbf{m} \leftarrow \mathbf{m} + \Delta \mathbf{m} \quad (14)$$

である。これによって、奥行きとモーシオンと時間オフセットを同時に最適化しており、複数枚の画像間の対応を同時に取ることに相当している。

### 3.2.2 複数点についての解法

前節では、1点に対してコスト関数を最小化する方法を説明した。画像内の複数の点のモーシオンと奥行きを求めたい場合、前節の方法で1点ごとに求めて行くのでは、本来画像内で一定であるはずの時間オフセットを別々に推定してしまう。そこで、複数の点ごとに異なるモーシオンと奥行き、さらにそれらの点すべてが同一な時間オフセットとして、コスト関数を最適化する。

それぞれの点を  $i (= [0, q])$  とすると、コスト関数はそれらの点のコスト関数の和として、

$$E(\Delta \mathbf{M}) = \sum_{i=0}^q \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{l,k,i}^T \mathbf{J}_{l,k,i}^T \Delta \mathbf{m}_i + e_{lr,k,i}]^2 + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{r,k,i}^T \mathbf{J}_{r,k,i}^T \mathbf{T}^T \Delta \mathbf{m}_i + e_{rr,k,i}]^2 \right\} \quad (15)$$

となる。ここで、

$$\Delta \mathbf{M} = [\Delta \mathbf{T}_{m,0}, \Delta Z_0, \dots, \Delta \mathbf{T}_{m,q}, \Delta Z_q, \Delta \lambda]^T \quad (16)$$

次に、 $\Delta \mathbf{m}_i = [\Delta T_{m,x,i}, \Delta T_{m,y,i}, \Delta T_{m,z,i}, \Delta Z_i, \Delta \lambda]^T$  について以下のように変形する。

$$\Delta \mathbf{m}_i = \mathbf{K}_i^T \Delta \mathbf{M} \quad (17)$$

$$\mathbf{K}_i^T = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 & 1 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 1 \end{bmatrix}$$

これを代入すると、

$$E(\Delta \mathbf{M}) = \sum_{i=0}^q \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{l,k,i}^T \mathbf{J}_{l,k,i}^T \mathbf{K}_i^T \Delta \mathbf{M} + e_{lr,k,i}]^2 + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{r,k,i}^T \mathbf{J}_{r,k,i}^T \mathbf{T}^T \mathbf{K}_i^T \Delta \mathbf{M} + e_{rr,k,i}]^2 \right\} \quad (18)$$

この式を  $\Delta \mathbf{M}$  で偏微分して0になる位置を求めると、

$$\frac{\partial E}{\partial \Delta \mathbf{M}} = \sum_{i=0}^q \mathbf{K}_i \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{J}_{l,k,i} \mathbf{g}_{l,k,i} \mathbf{g}_{l,k,i}^T \mathbf{J}_{l,k,i}^T] + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{T} \mathbf{J}_{r,k,i} \mathbf{g}_{r,k,i} \mathbf{g}_{r,k,i}^T \mathbf{J}_{r,k,i}^T \mathbf{T}^T] \right\} \mathbf{K}_i^T \Delta \mathbf{M} + \sum_{i=0}^q \mathbf{K}_i \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [e_{lr,k,i} \mathbf{J}_{l,k,i} \mathbf{g}_{l,k,i}] + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [e_{rr,k,i} \mathbf{T} \mathbf{J}_{r,k,i} \mathbf{g}_{r,k,i}] \right\} = 0 \quad (19)$$

となる。よって、この式を整理すると、

$$\mathbf{A} \Delta \mathbf{M} = -\mathbf{b} \quad (20)$$

となる。ここで、

$$\begin{aligned} \mathbf{A} &= \sum_{i=0}^q \mathbf{K}_i \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{J}_{l,k,i} \mathbf{g}_{l,k,i} \mathbf{g}_{l,k,i}^T \mathbf{J}_{l,k,i}^T] + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{T} \mathbf{J}_{r,k,i} \mathbf{g}_{r,k,i} \mathbf{g}_{r,k,i}^T \mathbf{J}_{r,k,i}^T \mathbf{T}^T] \right\} \mathbf{K}_i^T \\ \mathbf{b} &= \sum_{i=0}^q \mathbf{K}_i \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [e_{lr,k,i} \mathbf{J}_{l,k,i} \mathbf{g}_{l,k,i}] + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [e_{rr,k,i} \mathbf{T} \mathbf{J}_{r,k,i} \mathbf{g}_{r,k,i}] \right\} \end{aligned} \quad (21)$$

である。上式により求められた  $\Delta \mathbf{M}$  を使って、パラメータを

$$\mathbf{M} \leftarrow \mathbf{M} + \Delta \mathbf{M} \quad (22)$$

として更新する。更新を繰り返すことで、 $\mathbf{M}$  を推定する。これにより、複数点のすべての奥行きとモーシオンとステレオカメラ間の時間オフセットを同時に最適化しており、各点について複数枚の画像間の対応を同時に取ることに相当している。

## 3.3 Implementation

前述した推定アルゴリズムを使って、モーシオンと奥行きと時間オフセットを推定する。まず、キーフレームから特徴点を複数選んだ後、図3に示す手順で各パラメータを同時に推定する。その際、繰り返し計算の回数を削減し、初期値にロバストにするためにガウシアンピラミッドを利用した。

また本論文では、定式化を簡単にするためにステレオカメラは平行に配置され、かつカメラの内部パラメータ

<sup>†</sup>実験では Shi らの手法 [8] を用いて特徴点を抽出した。

```

for (iter=0 to iter max or convergence condition)
  if iter is 0
    Set initial value to M.
  end if.
  for (k=0 to max image)
    for (i=0 to max point)
      Warp image  $I_{r,k,i}$  and get  $\tilde{I}_{r,k,i}$  using  $m_i$  ( $k \neq 0$ ).
      Warp image  $I_{l,k,i}$  and get  $\tilde{I}_{l,k,i}$  using  $m_i$ .
      Calculate  $g_{r,k,i}$ ,  $J_{r,k,i}$ ,  $e_{rr,k,i}$  in eq.(40) ( $k \neq 0$ ).
      Calculate  $g_{l,k,i}$ ,  $J_{l,k,i}$ ,  $e_{lr,k,i}$  in eq.(33)
    end for loop.
  end for loop.
  Calculate  $A$  and  $b$  in eq.(21).
  Calculate  $\Delta M$  using eq.(20).
  Update  $M \leftarrow M + \Delta M$  in eq.(22).
end for loop.
3D MOTION, DEPTH for each point and TIME
OFFSET parameters are given.

```

図 3: 複数点の処理手順

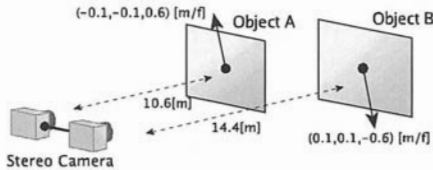


図 4: カメラとオブジェクトの配置とそのモーション

が両方のカメラで同じと仮定している。そのため、ステレオカメラが平行ステレオでない場合や内部パラメータが一致していないときは、前処理としてRectification[1]を行なうことで仮想的に平行カメラにし、かつカメラの内部パラメータをステレオカメラ間で一致させる。

## 4 実験結果

本節では、合成画像と実画像を用いた実験結果について述べる。

### 4.1 合成画像実験

合成画像実験では、図 4 に示すようにカメラに対して、奥行き異なる 2 枚の平行な板を配置した。それらの板は、それぞれ別々の運動をしている。

実験には、図 5 に示すようにステレオ画像間で 1 フレームずれた 2 フレーム計 4 枚の非同期時系列ステレオ画像を用いた。初期値としては奥行きを 12 [m] で

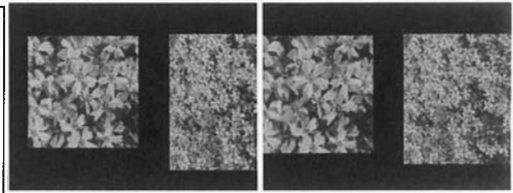


図 5: 入力した非同期ステレオ画像の一部

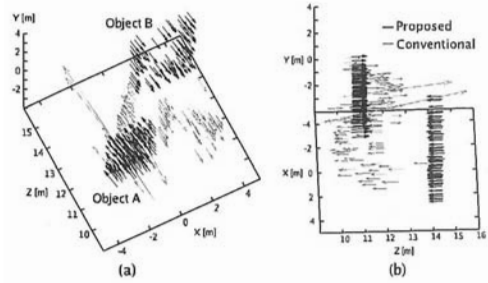


図 6: 3次元位置と3次元モーションの推定結果

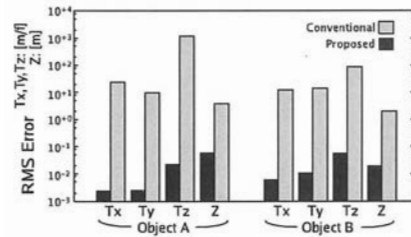


図 7: 従来手法と提案手法の推定精度比較結果

一緒に、モーションは  $T_m = [1e^{-5}, 1e^{-2}, 1e^{-1}]^T$  とし、時間オフセット  $\lambda$  は 0 とした。推定された時間オフセットは、 $-1.006$ (真値  $-1.0$ ) [frame lag] となり、正確に推定されていた。また、奥行きとモーションについては従来法と比較を行った。従来法は、非同期ステレオ画像間で対応点探索にウィンドウベースのマッチングを行い、次にその点に関して同じカメラの時系列画像間でトラッキングを行い、最後にトラッキングされた点について非同期ステレオ画像間で対応点を探索することで、各時刻の同一点に関する 3 次元位置同士の差分からモーションと奥行きを求めた。

図 6 は、提案手法と従来手法による 3 次元位置とモーションの推定結果を表している。モーションは、推定された空間位置を原点として、方向と長さはモーションの方向と大きさに対応している。同図で黒の矢印は提案手法を、グレーの矢印は従来手法をそれぞれ示している。同図 (b) を見ると、提案手法による推定結果は、矢印の開始位置が物体ごとと同じ面上にあり、か

つその長さが等しいことから、正しく推定されていることが分かる。一方、従来手法では、推定位置やモーションが散らばっており、正しく推定されていない。

図7は、提案手法と従来手法の推定結果のRMSエラーを表している。RMSエラーは、提案手法と従来手法で違いが大きかったため、縦軸に対数を取り、オブジェクトごとに計算している。従来手法では、非同期ステレオのため、対応点探索がうまくいかないことや対応がとれたとしても撮影時間ずれによる物体の位置変化から、正しく奥行きとモーションが推定できておらず、エラーが大きいたことが分かる。それに対して、提案手法では正確に推定されていることが分かる。

## 4.2 実画像実験

本実験は、Point Grey社製 Dragonfly と Sony社製 VX-2000のホームビデオカメラを用いた。これらのカメラ同士は、同期できる仕組みがないため、各カメラは別々のタイミングで撮影され、真の時間オフセットを知ることができない。入力した非同期ステレオ画像は、図8に示す2フレーム、計4枚である。シーンには、模型自動車A(同図中の記号に一致させている)と手で動かした熊のぬいぐるみCが動物体として含まれている。ステレオカメラは三脚に固定したため、背景領域や置物Bは、静止している。

初期値は、奥行きを0.7[m]、時間オフセットを0、モーションはXYZ軸ともに0.001[m/f]とした。推定された時間オフセットは、0.719[frame lag]であった。図9は、推定された3次元位置にテクスチャを張り付けてそれぞれ異なる視点から観察した結果である。Aの模型自動車や、Bの置物、Cの熊のぬいぐるみなどが正しい位置に推定されていることが分かる。

図10は、推定された3次元位置にモーションを併せて表示した結果である。矢印の方向と長さは推定されたモーションの方向と大きさに相当している。ここではモーションを見易くするため、2倍に拡大して表示している。また、同図は図9の右側とほぼ同じ方向から観察した様子に相当している。Aの模型自動車領域でモーションが大きく、それらのモーションを示す矢印は平行でかつ同じ長さとなっていることから、模型自動車のモーションが正しく推定されていることが分かる。静止領域であるBには、矢印がなく、+マークだけが表示されていることから、動いていないことがわかる。Cは、AとBとも違うモーションであることがわかる。

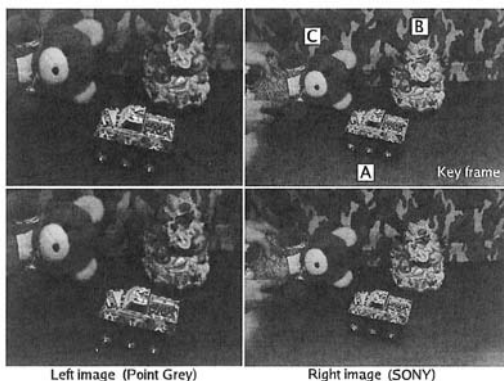


図8: 別々のカメラで撮影された非同期時系列ステレオ画像

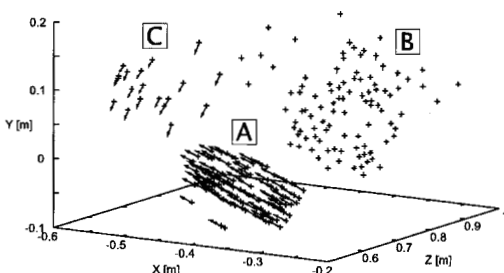


図10: 3次元推定位置とモーション推定結果. モーションを2倍に拡大して表示.

## 5 おわりに

本論文では、非同期ステレオカメラを用いた3次元位置と3次元モーションとステレオカメラ間の時間オフセットを同時に推定する手法を提案した。本手法では、各点ごとの3次元位置と3次元モーションを求めつつ、画像全体の時間オフセットを同時に推定することで、画像中に複数のモーションが含まれているような動的シーンにおいても適用することができる。最後に合成画像と実画像を用いて、提案手法の有効性を確認した。今後の課題としては、計算時間を短縮することが挙げられる。

## 参考文献

- [1] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, Vol. 12, No. 1, pp. 16–22, 2000.
- [2] K.J. Hanna and N.E. Okamoto. Combining and motion analysis for direct estimation of scene structure. In *Proc.IEEE*, pp. 357–365, 1993.

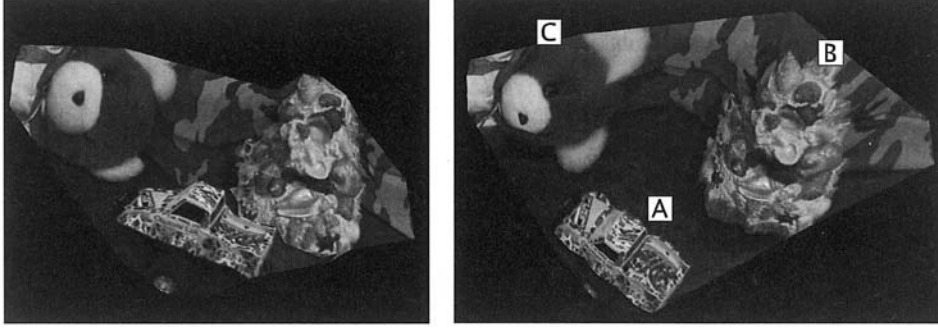


図 9: 3次元推定結果をさまざまな視点から観察した様子

- [3] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *Proc. IEEE CVPR*, pp. 454-460, June 1994.
- [4] R. Mandelbaum, G. Salgian, and H. Sawhney. Correlation-based estimation of ego-motion and structure from motion and stereo. In *Proc. ICCV*, pp. 544-550, 1998.
- [5] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 4, pp. 353-363, 1993.
- [6] S.K. Park and I.S. Kweon. Robust and direct estimation of 3-d motion and scene depth from stereo image sequences. *Pattern Recognition*, Vol. 34, No. 9, pp. 1713-1728, 2001.
- [7] 関見仁, 奥富正敏. ステレオ動画を用いた動的シーンのモーションと奥行き同時推定. 画像の認識・理解シンポジウム (MIRU), pp. 352-357, 2006.
- [8] J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE CVPR*, pp. 593-600, June 1994.
- [9] 清水彰一, 藤吉弘巨. カメラ間のシャッタータイミングのずれを利用した高速3次元位置推定. 画像の認識・理解シンポジウム (MIRU), pp. 428-433, July 2004.
- [10] S. Shimizu, H. Fujiyoshi, Y. Nagasaka, and T. Takahashi. A pseudo stereo vision method for unsynchronized cameras. In *Proc. ACCV*, Vol. Vol.1, pp. 575-580, 2004.
- [11] H.Y. Shum and R. Szeliski. Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, Vol. 36, No. 2, pp. 101-130, 2000.
- [12] G.P. Stein and A. Shashua. Direct estimation of motion and extended scene structure from a moving stereo rig. In *Proc. IEEE CVPR*, pp. 211-218, 1998.
- [13] M. Svedman, L. Goncalves, N. Karlsson, M. Munich, and P. Pirjanian. Structure from stereo vision using unsynchronized cameras for simultaneous localization mapping. In *Proc. IROS*, pp. 2-6, 2005.
- [14] C. Zhou and H. Tao. Dynamic depth recovery from unsynchronized video streams. In *Proc. IEEE CVPR*, Vol. 2, pp. 351-358, 2003.

## A 異なるカメラの画像間の Jacobian の導出

まず, 式 (7) の  $\mathbf{u}_k$  について  $\Delta \mathbf{m}$  の変化分を考えると,

$$\mathbf{u}_k^\Delta = \frac{\mathbf{A}_x \{ (k + \lambda + \Delta\lambda)(\mathbf{T}_m + \Delta\mathbf{T}) + \mathbf{T}_s \}}{Z + \Delta Z - \{ (k + \lambda + \Delta\lambda)T_{mz} + \Delta T_z \}} \quad (23)$$

となる. さらに, 上式の分母は以下のように表せる.

$$\frac{1}{Z + \Delta Z - \{ (k + \lambda + \Delta\lambda)T_{mz} + \Delta T_z \}} = \frac{1}{Z_k} + \xi \quad (24)$$

ここで,

$$\begin{aligned} \dot{Z}_k &= Z - (k + \lambda)T_{mz} \\ \xi &= \frac{-\{\Delta Z - (k + \lambda)\Delta T_z - \Delta\lambda T_{mz}\}}{Z_k^2 + Z_k \{\Delta Z - (k + \lambda)\Delta T_z - \Delta\lambda T_{mz}\}} \end{aligned} \quad (25)$$

である. 次に, 変化量は真値に対して小さい, つまり

$$Z - (k + \lambda)T_{mz} \gg \Delta Z - (k + \lambda)\Delta T_z - \Delta\lambda T_{mz}$$

と仮定すると,

$$\xi \approx \xi' = \frac{-\Delta Z + (k + \lambda)\Delta T_z + \Delta\lambda T_{mz}}{Z_k^2} \quad (26)$$

となる.  $\xi'$  を使い, 式 (23) を書き換えると,

$$\mathbf{u}_k^\Delta = \left( \frac{1}{Z_k} + \xi' \right) \mathbf{A}_x \{ (k + \lambda + \Delta\lambda)(\mathbf{T}_m + \Delta\mathbf{T}) + \mathbf{T}_s \} \quad (27)$$

となる. また,  $\Delta$  が 2 乗の項は十分に小さい為無視し, 平行ステレオを想定しているため,  $\mathbf{T}_s = [T_{sx}, 0, 0]^T$  として, 式 (27) を変形すると,

$$\begin{aligned} \mathbf{u}_k^\Delta &\approx \left( \frac{1}{Z_k} + \frac{-\Delta Z + (k + \lambda)\Delta T_z + \Delta\lambda T_{mz}}{Z_k^2} \right) \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &+ \frac{1}{Z_k} \begin{bmatrix} -(k + \lambda)f\Delta T_x + (k + \lambda)\hat{x}_0\Delta T_z \\ + (-fT_{mx} + \hat{x}_0T_{mz})\Delta\lambda \\ -(k + \lambda)af\Delta T_y + (k + \lambda)\hat{y}_0\Delta T_z \\ + (-afT_{my} + \hat{y}_0T_{mz})\Delta\lambda \end{bmatrix} \end{aligned} \quad (28)$$

ここで,

$$\begin{aligned} \alpha &= (k + \lambda)(-fT_{mx} + \hat{x}_0T_{mz}) - fT_{sx} \\ \beta &= (k + \lambda)(-afT_{my} + \hat{y}_0T_{mz}) \end{aligned} \quad (29)$$

である。これをさらに変形すると、式 (30) を得ることができる。

$$\mathbf{x} + \mathbf{u}_k^{\Delta} = \begin{bmatrix} \gamma & 0 & \frac{\alpha'}{Z_k} \\ 0 & \gamma & \frac{\beta'}{Z_k} \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \kappa & 0 & \zeta \\ 0 & \kappa & \eta \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{I} + \mathbf{D}} \tilde{\mathbf{x}}, \quad (30)$$

ここで、

$$\begin{aligned} \tilde{\mathbf{x}} &= [x, y, 1]^T \\ \alpha' &= -(k + \lambda)(fT_{mx} + u_0T_{mz}) - fT_{sz} \\ \beta' &= -(k + \lambda)(afT_{my} + v_0T_{mz}) \\ \gamma &= 1 + \frac{(k + \lambda)T_{mz}}{Z_k} \\ \kappa &= 1 + \frac{1}{Z_k}[(k + \lambda)(T_{mz} + \Delta T_z) + T_{mz}\Delta\lambda] \\ &\quad + \xi'(k + \lambda)T_{mz} \\ \zeta &= -\frac{1}{Z_k}[(k + \lambda)(f\Delta T_x + u_0\Delta T_z) \\ &\quad + (fT_{mx} + u_0T_{mz})\Delta\lambda] + \xi'\alpha' \\ \eta &= -\frac{1}{Z_k}[(k + \lambda)(af\Delta T_y + v_0\Delta T_z) \\ &\quad + (afT_{my} + v_0T_{mz})\Delta\lambda] + \xi'\beta' \end{aligned} \quad (31)$$

で表される。

さて、次に微小な要素を持つ行列  $\mathbf{D}$  を用い、座標変換  $\tilde{\mathbf{x}}'' \sim (\mathbf{I} + \mathbf{D})\tilde{\mathbf{x}}$  を考える。ここで、 $\tilde{\mathbf{x}}$  は  $\mathbf{x}$  の同次座標表現とし、 $\mathbf{I}$  は 3 次の単位ベクトルである。 $\tilde{\mathbf{I}}_{l,k}(\mathbf{x}'')$  について座標  $\mathbf{x}$  の周りで 1 次のテイラー展開をすると、

$$\tilde{\mathbf{I}}_{l,k}(\mathbf{x}'') - \mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k}. \quad (32)$$

ここで、

$$\begin{aligned} \mathbf{g}_{l,k}^T &= \nabla \tilde{\mathbf{I}}_{l,k}(\mathbf{x}) \\ \mathbf{J}_{l,k}^T &= \frac{\partial \tilde{\mathbf{x}}''}{\partial \Delta \mathbf{m}} \\ \Delta \mathbf{m} &= [\Delta T_{mx}, \Delta T_{my}, \Delta T_{mz}, \Delta Z, \Delta \lambda]^T \\ e_{lr,k} &= \tilde{\mathbf{I}}_{l,k}(\mathbf{x}) - \mathbf{I}_{r,0}(\mathbf{x}) \end{aligned} \quad (33)$$

である。さて、Jacobian を構成する行列は

$$\frac{\partial \tilde{\mathbf{x}}''}{\partial \Delta \mathbf{m}} = \frac{\partial \mathbf{d}}{\partial \Delta \mathbf{m}} \frac{\partial \tilde{\mathbf{x}}''}{\partial \mathbf{d}} \quad (34)$$

と変形できる。 $\mathbf{d}$  は、行列  $\mathbf{D}$  を行ごとに並べたベクトルである。右辺の後半は、

$$\frac{\partial \tilde{\mathbf{x}}''}{\partial \mathbf{d}} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x^2 & -xy & -x \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2 & -y \end{bmatrix}^T \quad (35)$$

である [11]。右辺の前半は、式 (30) の  $\mathbf{D}$  に相当する部分を  $\Delta \mathbf{m}$  で偏微分することで得られる。それと式 (35) を式 (34) に代入することで Jacobian が得られ、

$$\mathbf{J}_{l,k}^T = \frac{\partial \tilde{\mathbf{x}}''}{\partial \Delta \mathbf{m}} = \frac{1}{\gamma} \begin{bmatrix} j_{11} & 0 \\ 0 & j_{22} \\ \mu & \nu \\ j_{41} & j_{42} \\ j_{51} & j_{52} \end{bmatrix} \quad (36)$$

$\tilde{\mathbf{I}}_{l,k}(\mathbf{x})$  は  $\mathbf{I}_{l,k}$  を座標  $(\mathbf{x} + \mathbf{u}_k)$  でサブサンプリングして生成した画像である。

となる。ここで、

$$\begin{aligned} j_{11} &= -\frac{(k + \lambda)f}{Z_k}, & j_{22} &= -\frac{(k + \lambda)af}{Z_k} \\ j_{41} &= -\frac{\alpha' + (k + \lambda)xT_{mz}}{Z_k}, & j_{42} &= -\frac{\beta' + (k + \lambda)yT_{mz}}{Z_k} \\ j_{51} &= -\frac{fT_{mz} - (x - u_0)T_{mz}}{Z_k} + \frac{\alpha'T_{mz} + (k + \lambda)xT_{mz}^2}{Z_k^2} \\ j_{52} &= -\frac{afT_{my} - (y - v_0)T_{mz}}{Z_k} + \frac{\beta'T_{mz} + (k + \lambda)yT_{mz}^2}{Z_k^2} \\ \mu &= \frac{1}{Z_k}(k + \lambda)(x - u_0) + \frac{1}{Z_k^2}(k + \lambda)\{\alpha' + (k + \lambda)T_{mz}x\} \\ \nu &= \frac{1}{Z_k}(k + \lambda)(y - v_0) + \frac{1}{Z_k^2}(k + \lambda)\{\beta' + (k + \lambda)T_{mz}y\} \end{aligned}$$

である。

## B 同一カメラの画像間の Jacobian の導出

式 (7) の  $\mathbf{u}'_k$  についても、付録 A の  $\mathbf{u}_k$  と同様に式展開をすることができる。最終的に、Jacobian は

$$\mathbf{J}_{r,k}^T = \frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}'} = \frac{1}{\gamma''} \begin{bmatrix} -\frac{kf}{Z_k} & 0 \\ 0 & -\frac{ka}{Z_k} \\ \mu'' & \nu'' \\ -\frac{\alpha'' + kxT_{mz}}{Z_k^2} & -\frac{\beta'' + kyT_{mz}}{Z_k^2} \end{bmatrix} \quad (37)$$

となる。ここで、

$$\begin{aligned} \Delta \mathbf{m}' &= [\Delta T_{mx}, \Delta T_{my}, \Delta T_{mz}, \Delta Z]^T \\ Z_k &= Z - kT_{mz} \\ \alpha'' &= -k(fT_{mx} + u_0T_{mz}) \\ \beta'' &= -k(afT_{my} + v_0T_{mz}) \\ \gamma'' &= 1 + \frac{kT_{mz}}{Z_k} \\ \mu'' &= \frac{1}{Z_k}k(x - u_0) + \frac{1}{Z_k^2}k(\alpha'' + kT_{mz}x) \\ \nu'' &= \frac{1}{Z_k}k(y - v_0) + \frac{1}{Z_k^2}k(\beta'' + kT_{mz}y) \end{aligned} \quad (38)$$

で表される。また、同様に

$$\tilde{\mathbf{I}}_{r,k}(\mathbf{x}'') - \mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \Delta \mathbf{m}' + e_{rr,k}. \quad (39)$$

ここで、

$$\begin{aligned} \mathbf{g}_{r,k}^T &= \nabla \tilde{\mathbf{I}}_{r,k}(\mathbf{x}) \\ \mathbf{J}_{r,k}^T &= \frac{\partial \tilde{\mathbf{x}}''}{\partial \Delta \mathbf{m}'} \\ e_{rr,k} &= \tilde{\mathbf{I}}_{r,k}(\mathbf{x}) - \mathbf{I}_{r,0}(\mathbf{x}) \end{aligned} \quad (40)$$

である。

$$\Delta \mathbf{m}' = \mathbf{T}^T \Delta \mathbf{m}, \quad \mathbf{T}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (41)$$

よって、式 (41) を用いて式 (39) の  $\Delta \mathbf{m}'$  を除去すると

$$\mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \Delta \mathbf{m}' + e_{rr,k} = \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \mathbf{T}^T \Delta \mathbf{m} + e_{rr,k} \quad (42)$$

が得られる。