社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－CVIM－157 (15)
2007／1／12

高速射影復元：徹底的な効率化を目指して

森 昭延　　ハノ・アッカーマン　　金谷 健一
岡山大学大学院自然科学研究科

未校正カメラで撮影した画像列上の特徴点の追跡から3次元形状を計算する自己校正法において，最も計算時間を要する射影復元の反復を高速化する手法を示す．まず，射影復元の基本法と双対法のアルゴリズムをまとめる．そして，射影復元の反復に含まれる固有値計算をべき乗法に置き換える．さらにべき乗法自体も加速し，部分空間の当てはめの反復もSOR法によって加速する．これらの効果をシミュレーションや実ビデオ画像によって定量的に評価し，実行時間が大幅に減少することを実証する．

Fast Projective Reconstruction: Toward Ultimate Efficiency

Akinobu Mori, Hanno Ackermann, and Kenichi Kanatani
Department of Computer Science, Okayama University, Okayama 700-8530 Japan

We accelerate the time-consuming iterations for projective reconstruction, a key component of self-calibration for computing 3-D shapes from feature point tracking over a video sequence. We first summarize the algorithms of the primal and dual methods for projective reconstruction. Then, we replace the eigenvalue computation in each step by the power method. We also accelerate the power method itself. Furthermore, we introduce the SOR method for accelerating the subspace fitting involved in the iterations. Using simulated and real video images, we evaluate the computation time in quantitative terms and demonstrate that these techniques significantly reduce the computation time.

1. まえがき

前報 [7, 10] では未校正カメラで撮影した画像列上の特徴点の追跡から3次元形状を計算する最新の「自己校正法」の計算手順を示し，シミュレーションおよび実ビデオ画像を用いてその性能評価を行った．これは，過去の研究成果の中から最も優れていると思われる手法を選択して組合せ，さらに洗練と改良を加えたものである．

この方法は，射影変換の不定性のある3次元形状を計算する「射影復元」と，それを正しい形状に変換する「ユークリッド化」の2段階から成る．前者にはMahamudら [9] およびHeydenら [5] の方法を用い，後者にはSeoら [13] の方法に精度や計算の安定性を向上させる新しい工夫を加えている．

前報 [10] のシミュレーションおよび実ビデオ画像による性能評価によると，同様の目的で用いられる「因子分解法」[6, 8, 12, 15] に比べて精度が高く，実際的な応用に対して十分良好な復元ができることが確認された．しかし，因子分解法が撮像をアフィンカメラで近似して解析的な計算を行うのに対して，自己校正法では透視投影による非線形性を反復によって処理するので，射影復元の段階で多くの実行時間を要する．

一方，Seoら [13] のユークリッド化法は，本来は解析的に求まる計算にロバスト性を高めるために数回の反復を付加するものであり，計算時間は非常に少ない．実際，実測すると射影復元の反復が全実行時間のほとんどを占める [10]．

そこで本論文では射影復元の反復の高速化を試みる．時間がかかる最大の理由は，反復の各ステップで固有値計算が必要があり，その行列の次元がフレーム数あるいは特徴点数に比例するためである．

しかし，計算される固有ベクトルは直前の反復ステップの値にほぼ等しいはずであり，毎回新たに計算する必要はなく，前回の値を補正することにすれば，計算時間が大幅に削減されると期待される．そのために，本論文では「べき乗法」を用いる．

そのべき乗法自体も反復法である．そこで，その収束の挙動を解析して反復の加速を試みる．さらに射影復元のための部分空間の当てはめの反復に対してもSOR法を適用する．そして，これらの工夫によって実行時間が大幅に削減することをシミュレーションや実ビデオ画像によって実証する．

*700-8530 岡山市津島中 3-1-1, (086)251-8173
{mori,hanno,kanatani}@suri.it.okayama-u.ac.jp

2. 射影復元の方法

射影復元の基本は前報 [7, 10] に示した次式である [4].

$$z_{\kappa\alpha}\boldsymbol{x}_{\kappa\alpha} = \boldsymbol{\Pi}_\kappa \boldsymbol{X}_\alpha, \qquad \boldsymbol{x}_\alpha = \begin{pmatrix} x_\kappa/f_0 \\ y_\kappa/f_0 \\ 1 \end{pmatrix} \tag{1}$$

ここに $(x_{\kappa\alpha}, y_{\kappa\alpha})$ は第 κ フレームの第 α 特徴点の座標であり，f_0 はある固定した定数である[1]．$z_{\kappa\alpha}$ は「射影的奥行き」と呼ばれる定数，$\boldsymbol{\Pi}_\kappa$ は第 κ フレームの 3×4 投影行列，$\boldsymbol{X}_\alpha$ は第 α 特徴点の 3 次元位置を同次座標で表す 4 次元ベクトルである．これら $z_{\kappa\alpha}$, $\boldsymbol{\Pi}_\kappa$, $\boldsymbol{X}_\alpha$ はすべては未知である．

これらを求める計算法として Mahamud ら [9] の方法（「基本法」と呼ぶ）と Heyden ら [5] の方法（「双対」と呼ぶ）が提案されている．両者は幾何学的には互いに双対な関係があり [10]，どちらがよいとは一概には言えない [10]．まず両者のアルゴリズムをまとめ，後の比較のために「原形」と呼ぶ（導出は前報 [7, 10] 参照）．以下，ベクトル $\boldsymbol{a}$, $\boldsymbol{b}$ の内積を $(\boldsymbol{a}, \boldsymbol{b})$ と書く．

2.1 基本法（原形）

入力: $\boldsymbol{x}_{\kappa\alpha}$, $\kappa = 1, ..., M$, $\alpha = 1, ..., N$
収束判定の再投影誤差 $E_{\min}$(画素)
出力: $\boldsymbol{\Pi}_\kappa$, $\kappa = 1, ..., M$, $\boldsymbol{X}_\alpha$, $\alpha = 1, ..., N$
計算:

1. 射影的奥行きを $z_{\kappa\alpha} = 1$ と初期化する．
2. $z_{1\alpha}\boldsymbol{x}_{1\alpha}, z_{2\alpha}\boldsymbol{x}_{2\alpha}, ..., z_{M\alpha}\boldsymbol{x}_{M\alpha}$ を縦に並べた $3M$ 次元ベクトル $\boldsymbol{p}_\alpha$ を計算し，単位ベクトルに正規化する．
3. 次の $3M \times 3M$ 行列 $\boldsymbol{M}$ を計算する

$$\boldsymbol{M} = \sum_{\alpha=1}^{N} \boldsymbol{p}_\alpha \boldsymbol{p}_\alpha^\top \tag{2}$$

4. 行列 $\boldsymbol{M}$ の大きい 4 個の固有値に対する単位固有ベクトル $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{u}_3$, $\boldsymbol{u}_4$ を計算する．
5. 3×4 行列 $\boldsymbol{\Pi}_\kappa$ を次のように計算する．

$$\boldsymbol{\Pi}_\kappa = \begin{pmatrix} \boldsymbol{u}_{1\kappa} & \boldsymbol{u}_{2\kappa} & \boldsymbol{u}_{3\kappa} & \boldsymbol{u}_{4\kappa} \end{pmatrix} \tag{3}$$

ただし $\boldsymbol{u}_{i\kappa}$ は $\boldsymbol{u}_i$ の第 $3(\kappa-1)+1$, $3(\kappa-1)+2$, $3(\kappa-1)+3$ 成分を第 1, 2, 3 成分とする 3 次元ベクトルである．

6. 次の計算を $\alpha = 1, ..., N$ に渡って計算する．

(a) 次のように計算した $M \times M$ 行列 $\boldsymbol{A}^\alpha = (A^\alpha_{\kappa\lambda})$ の最大固有値に対する単位固有ベクトル $\boldsymbol{\xi}_\alpha$ を計算する．

$$A^\alpha_{\kappa\lambda} = \frac{\sum_{k=1}^{4} (\boldsymbol{x}_{\kappa\alpha}, \boldsymbol{u}_{k\kappa})(\boldsymbol{x}_{\lambda\alpha}, \boldsymbol{u}_{k\lambda})}{\|\boldsymbol{x}_{\kappa\alpha}\| \cdot \|\boldsymbol{x}_{\lambda\alpha}\|} \tag{4}$$

固有ベクトルの符号は次のように定める．

$$\sum_{\kappa=1}^{M} \xi_{\kappa\alpha} \geq 0 \tag{5}$$

(b) 得られた $\boldsymbol{\xi}_\alpha$ から射影的奥行き $z_{\kappa\alpha}$ を次のように計算する．

$$z_{\kappa\alpha} = \frac{\xi_{\kappa\alpha}}{\|\boldsymbol{x}_{\kappa\alpha}\|} \tag{6}$$

(c) 得られた $z_{\kappa\alpha}$ を用いてベクトル $\boldsymbol{p}_\alpha$ を再計算する．

(d) 3 次元位置 $\boldsymbol{X}_\alpha = (X^k_\alpha)$ を次のように計算する．

$$X^k_\alpha = (\boldsymbol{p}_\alpha, \boldsymbol{u}_k), \qquad k = 1 \sim 4 \tag{7}$$

7. 次のように再投影誤差 E を計算する．

$$E = f_0 \sqrt{\frac{1}{MN} \sum_{\kappa=1}^{M} \sum_{\alpha=1}^{N} \|\boldsymbol{x}_{\kappa\alpha} - Z[\boldsymbol{\Pi}_\kappa \boldsymbol{X}_\alpha]\|^2} \tag{8}$$

8. $E < E_{\min}$ であれば終了する．そうでなければステップ 3 に戻る．

2.2 双対法（原形）

入力: $\boldsymbol{x}_{\kappa\alpha}$, $\kappa = 1, ..., M$, $\alpha = 1, ..., N$
収束判定の再投影誤差 $E_{\min}$(画素)
出力: $\boldsymbol{\Pi}_\kappa$, $\kappa = 1, ..., M$, $\boldsymbol{X}_\alpha$, $\alpha = 1, ..., N$
計算:

1. 射影的奥行きを $z_{\kappa\alpha} = 1$ と初期化する．
2. 次の N 次元ベクトル $\boldsymbol{q}^i_\kappa$ を計算する．

$$\begin{aligned} \boldsymbol{q}^1_\kappa &= (z_{\kappa 1}x_{\kappa 1}/f_0, z_{\kappa 2}x_{\kappa 2}/f_0, ..., z_{\kappa N}x_{\kappa N}/f_0)^\top \\ \boldsymbol{q}^2_\kappa &= (z_{\kappa 1}y_{\kappa 1}/f_0, z_{\kappa 2}y_{\kappa 2}/f_0, ..., z_{\kappa N}y_{\kappa N}/f_0)^\top \\ \boldsymbol{q}^3_\kappa &= (z_{\kappa 1}, z_{\kappa 2}, ..., z_{\kappa N})^\top \end{aligned} \tag{9}$$

そして各 κ ごとに $\boldsymbol{q}^i_\kappa$, $i = 1, 2, 3$ に共通の定数を掛けて，次式が成り立つように正規化する．

$$\sum_{i=1}^{3} \|\boldsymbol{q}^i_\kappa\|^2 = 1 \tag{10}$$

3. 次の $N \times N$ 行列 $\boldsymbol{N}$ を計算する．

$$\boldsymbol{N} = \sum_{\kappa=1}^{M} \sum_{i=1}^{3} \boldsymbol{q}^i_\kappa \boldsymbol{q}^{i\top}_\kappa \tag{11}$$

[1] これは数値計算を安定化させるためである [2]．実験では f_0 = 600 画素とした

4. 行列 $\boldsymbol{N}$ の大きい 4 個の固有値に対する単位固有ベクトル $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$, $\boldsymbol{v}_4$ を計算する.
5. 3 次元位置 $\boldsymbol{X}_\alpha = (X_\alpha^k)$ を次のように計算する.

$$X_\alpha^k = (\boldsymbol{v}_k \text{ の第 } \alpha \text{ 成分}), \quad k = 1 \sim 4 \tag{12}$$

6. 次の計算を $\kappa = 1, ..., M$ に渡って計算する.
   (a) 次のように計算した $N \times N$ 行列 $\boldsymbol{B}^\kappa = (B_{\alpha\beta}^\kappa)$ の最大固有値に対する単位固有ベクトル $\boldsymbol{\xi}_\kappa$ を計算する.

$$B_{\alpha\beta}^\kappa = \frac{(\boldsymbol{v}_\alpha, \boldsymbol{v}_\beta)(\boldsymbol{x}_{\kappa\alpha}, \boldsymbol{x}_{\kappa\beta})}{\|\boldsymbol{x}_{\kappa\alpha}\| \cdot \|\boldsymbol{x}_{\kappa\beta}\|} \tag{13}$$

   ただし $\boldsymbol{v}_\alpha$ は基底ベクトル $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$, $\boldsymbol{v}_4$ のそれぞれの第 α 成分を取り出して縦に並べた 4 次元ベクトルである. 固有ベクトルの符号は次のように定める.

$$\sum_{\alpha=1}^{N} \xi_{\kappa\alpha} \geq 0 \tag{14}$$

   (b) 得られた $\boldsymbol{\xi}_\kappa$ から射影的奥行き $z_{\kappa\alpha}$ を次のように計算する.

$$z_{\kappa\alpha} = \frac{\xi_{\kappa\alpha}}{\|\boldsymbol{x}_{\kappa\alpha}\|} \tag{15}$$

   (c) 得られた $z_{\kappa\alpha}$ を用いてベクトル $\boldsymbol{q}_\kappa^i$, $i = 1, 2, 3$ を再計算する.
   (d) 3×4 行列 $\boldsymbol{\Pi}_\kappa = (\Pi_{\kappa(ij)})$ を次のように計算する.

$$\Pi_{\kappa(ij)} = (\boldsymbol{q}_\kappa^i, \boldsymbol{v}_k) \tag{16}$$

7. 式 (8) の再投影誤差 E を計算する.
8. $E < E_{\min}$ であれば終了する. そうでなければステップ 3 に戻る.

3. べき乗法による高速化

$n \times n$ 半正値対称行列 $\boldsymbol{A}$ の固有値を $\lambda_1 \geq \cdots \geq \lambda_n$ (≥ 0), 対応する単位固有ベクトルを $\boldsymbol{u}_1, ..., \boldsymbol{u}_n$ とすると, $\boldsymbol{A}^k$ の固有値は $\lambda_1^k \geq \cdots \geq \lambda_n^k$ であり, 対応する単位固有ベクトルは変化しない. このため, 任意の線形独立なベクトル $\boldsymbol{v}_1, ..., \boldsymbol{v}_m$ に $\boldsymbol{A}^k$ を掛けると, 大きい固有値に対する固有ベクトルの方向の成分が増幅され, k を大きくすると $\boldsymbol{A}^k\boldsymbol{v}_1, ..., \boldsymbol{A}^k\boldsymbol{v}_m$ の張る m 次元部分空間は $\boldsymbol{u}_1, ..., \boldsymbol{u}_m$ の張る m 次元部分空間に近づく. 特に $m = 1$ の場合, 任意のベクトル $\boldsymbol{v}$ に対して $\boldsymbol{A}^k\boldsymbol{v}$ の方向は $\boldsymbol{u}_1$ に収束する [1]. これがべき乗法の原理であり, 因子分解法の効率化にも利用されている [3, 11, 16].

これを利用して前節の固有値問題を, 次のようにべき乗法に置き換える. 以下, $N[\cdot]$ は単位ベクトルへの正規化を表す.

3.1 基本法 (べき乗法)

入力: $\boldsymbol{x}_{\kappa\alpha}$, $\kappa = 1, ..., M$, $\alpha = 1, ..., N$
収束判定の再投影誤差 $E_{\min}$(画素)
べき乗法の収束判定定数 d, e
出力: $\boldsymbol{\Pi}_\kappa$, $\kappa = 1, ..., M$, $\boldsymbol{X}_\alpha$, $\alpha = 1, ..., N$
計算:

1. 射影的奥行きを $z_{\kappa\alpha} = 1$ と初期化する.
2. 次の M 次元単位ベクトル $\boldsymbol{\xi}_\alpha^0$ を計算する.

$$\boldsymbol{\xi}_\alpha^0 = N[\begin{pmatrix} \|\boldsymbol{x}_{1\alpha}\| z_{1\alpha} \\ \|\boldsymbol{x}_{2\alpha}\| z_{2\alpha} \\ \vdots \\ \|\boldsymbol{x}_{M\alpha}\| z_{M\alpha} \end{pmatrix}] \tag{17}$$

3. $z_{1\alpha}\boldsymbol{x}_{1\alpha}$, $z_{2\alpha}\boldsymbol{x}_{2\alpha}$, ..., $z_{M\alpha}\boldsymbol{x}_{M\alpha}$ を縦に並べた $3M$ 次元ベクトル $\boldsymbol{p}_\alpha$ を計算する.
4. $\boldsymbol{p}_\alpha$ を単位ベクトルに正規化する.
5. 次の $3M \times N$ 行列 $\boldsymbol{P}$ を定義する.

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{p}_1 & \boldsymbol{p}_2 & \cdots & \boldsymbol{p}_N \end{pmatrix} \tag{18}$$

6. 行列 $\boldsymbol{P}$ を次のように特異値分解する.

$$\boldsymbol{P} = \boldsymbol{U}\mathrm{diag}(\sigma_1, \sigma_2, \cdots)\boldsymbol{V}^\top \tag{19}$$

7. 行列 $\boldsymbol{U}$ の最初の 4 列を $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{u}_3$, $\boldsymbol{u}_4$ とする.
8. 3×4 行列 $\boldsymbol{\Pi}_\kappa$ を式 (3) のように計算する.
9. 次の計算を $\alpha = 1, ..., N$ に渡って計算する.
   (a) 式 (4) のように $M \times M$ 行列 $\boldsymbol{A}^\alpha$ を計算する.
   (b) 次のように $\boldsymbol{\xi}_\alpha$ を計算する.

$$\boldsymbol{\xi}_\alpha = N[\boldsymbol{A}^\alpha\boldsymbol{\xi}_\alpha^0] \tag{20}$$

   (c) $\|\boldsymbol{\xi}_\alpha - \boldsymbol{\xi}_\alpha^0\| < 10^{-d}$ であれば次に進む. そうでなければ $\boldsymbol{\xi}_\alpha^0 \leftarrow \boldsymbol{\xi}_\alpha$ としてステップ (b) に戻る.
   (d) 得られた $\boldsymbol{\xi}_\alpha$ から射影的奥行き $z_{\kappa\alpha}$ を式 (6) のように計算する.
   (e) 得られた $z_{\kappa\alpha}$ を用いてベクトル $\boldsymbol{p}_\alpha$ を再計算する.
   (f) 3 次元位置 $\boldsymbol{X}_\alpha$ を式 (7) のように計算する.
10. 式 (8) の再投影誤差 E を計算する.
11. $E < E_{\min}$ であれば終了する. そうでなければ式 (18) の行列 $\boldsymbol{P}$ を再計算する.
12. 次の N 次元ベクトル $\tilde{\boldsymbol{v}}_i$ と $3M$ 次元ベクトル $\tilde{\boldsymbol{u}}_k$ を計算する $(k = 1 \sim 4)$.

$$\tilde{\boldsymbol{v}}_k = \boldsymbol{P}^\top\boldsymbol{u}_k, \qquad \tilde{\boldsymbol{u}}_k = \boldsymbol{P}\tilde{\boldsymbol{v}}_k \tag{21}$$

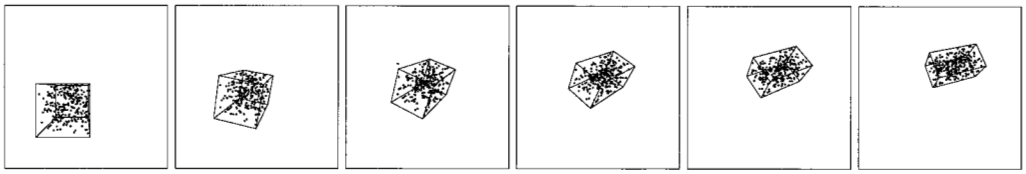

図 1: 256 個の特徴点の 256 フレームのシミュレーション画像系列から抜き出した 6 フレーム.

13. $\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, \tilde{\boldsymbol{u}}_3, \tilde{\boldsymbol{u}}_4$ にシュミットの直交化を施して得られる正規直交系を $\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \hat{\boldsymbol{u}}_3, \hat{\boldsymbol{u}}_4$ とする.
14. $\max_{k=1}^{4} \sqrt{1 - \sum_{l=1}^{4} (\hat{\boldsymbol{u}}_k, \boldsymbol{u}_l)^2} < 10^{-e}$ なら $\boldsymbol{u}_k \leftarrow \tilde{\boldsymbol{u}}_k$, $k = 1 \sim 4$ としてステップ 8 に戻る. そうでなければ $\boldsymbol{u}_k \leftarrow \tilde{\boldsymbol{u}}_k$, $k = 1 \sim 4$ としてステップ 12 に戻る.

3.2 双対法（べき乗法）

入力: $\boldsymbol{x}_{\kappa\alpha}$, $\kappa = 1, ..., M$, $\alpha = 1, ..., N$
収束判定の再投影誤差 $E_{\min}$(画素)
べき乗法の収束判定定数 d, e

出力: $\boldsymbol{\Pi}_\kappa$, $\kappa = 1, ..., M$, $\boldsymbol{X}_\alpha$, $\alpha = 1, ..., N$

計算:

1. 射影的奥行きを $z_{\kappa\alpha} = 1$ と初期化する.
2. 次の N 次元単位ベクトル $\boldsymbol{\xi}_\kappa^0$ を計算する.

$$\boldsymbol{\xi}_\kappa^0 = N[\begin{pmatrix} \|\boldsymbol{x}_{\kappa 1}\| z_{\kappa 1} \\ \|\boldsymbol{x}_{\kappa 2}\| z_{\kappa 2} \\ \vdots \\ \|\boldsymbol{x}_{\kappa M}\| z_{\kappa N} \end{pmatrix}] \qquad (22)$$

3. 式 (9) の N 次元ベクトル $\boldsymbol{q}_\kappa^i$ を計算する.
4. 各 κ ごとに $\boldsymbol{q}_\kappa^i$, $i = 1, 2, 3$ を式 (10) のように正規化する.
5. 次の $N \times 3M$ 行列 $\boldsymbol{Q}$ を計算する.

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{q}_1^1 & \boldsymbol{q}_1^2 & \boldsymbol{q}_1^3 & \boldsymbol{q}_2^1 & \cdots & \boldsymbol{q}_M^3 \end{pmatrix} \qquad (23)$$

6. 行列 $\boldsymbol{Q}$ を次のように特異値分解する[2].

$$\boldsymbol{Q} = \boldsymbol{V} \mathrm{diag}(\sigma_1, \sigma_2, ...) \boldsymbol{U}^\top \qquad (24)$$

7. 行列 $\boldsymbol{V}$ の最初の 4 列を $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4$ とする.
8. 3 次元位置 $\boldsymbol{X}_\alpha$ を式 (12) のように計算する.
9. 次の計算を $\kappa = 1, ..., M$ に渡って計算する.
   (a) 式 (13) のように $N \times N$ 行列 $\boldsymbol{B}^\kappa$ を計算する.
   (b) 次のように $\boldsymbol{\xi}_\kappa$ を計算する.

$$\boldsymbol{\xi}_\kappa = N[\boldsymbol{B}^\kappa \boldsymbol{\xi}_\kappa^0] \qquad (25)$$

   (c) $\|\boldsymbol{\xi}_\kappa - \boldsymbol{\xi}_\kappa^0\| < 10^{-d}$ であれば次に進む. そうでなければ $\boldsymbol{\xi}_\kappa^0 \leftarrow \boldsymbol{\xi}_\kappa$ としてステップ (b) に戻る.
   (d) 得られた $\boldsymbol{\xi}_\kappa$ から射影的奥行き $z_{\kappa\alpha}$ を式 (15) のように計算する.
   (e) 得られた $z_{\kappa\alpha}$ を用いてベクトル $\boldsymbol{q}_\kappa^i$ を再計算する.
   (f) 3×4 行列 $\boldsymbol{\Pi}_\kappa = (\Pi_{\kappa(ij)})$ を式 (16) のように計算する.
10. 式 (8) の再投影誤差 E を計算する.
11. $E < E_{\min}$ であれば終了する. そうでなければ式 (23) の行列 $\boldsymbol{Q}$ を再計算する.
12. 次の $3M$ 次元ベクトル $\tilde{\boldsymbol{u}}_k$ と N 次元ベクトル $\tilde{\boldsymbol{v}}_k$ を計算する ($k = 1 \sim 4$).

$$\tilde{\boldsymbol{u}}_k = \boldsymbol{Q}^\top \boldsymbol{v}_k, \qquad \tilde{\boldsymbol{v}}_k = \boldsymbol{Q} \tilde{\boldsymbol{u}}_k \qquad (26)$$

13. $\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \tilde{\boldsymbol{v}}_3, \tilde{\boldsymbol{v}}_4$ にシュミットの直交化を施して得られる正規直交系を $\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \hat{\boldsymbol{v}}_3, \hat{\boldsymbol{v}}_4$ とする.
14. $\max_{k=1}^{4} \sqrt{1 - \sum_{l=1}^{4} (\hat{\boldsymbol{v}}_k, \boldsymbol{v}_l)^2} < 10^{-e}$ なら $\boldsymbol{v}_k \leftarrow \tilde{\boldsymbol{v}}_k$, $k = 1 \sim 4$ としてステップ 8 に戻る. そうでなければ $\boldsymbol{v}_k \leftarrow \tilde{\boldsymbol{v}}_k$, $k = 1 \sim 4$ としてステップ 12 に戻る.

4. 実験 1

4.1 べき乗法の効果

図 1 にシミュレーション画像を示す. これは 3 次元空間の直方体領域内に 256 個の特徴点をランダムにとり, 視点を移動させながら 600×600 画素の画像面に焦点距離 $f = 600$（画素）の透視投影によって投影した 256 フレームの画像列から 6 フレームを抜き出したものである[3].

これに対して基本法と双対法による射影復元を行い, 再投影誤差が 0.1 画素になるまで反復した. そして, 2 節の原形と 3 節のべき乗法を用いた場合の実行時間（秒）と反復回数を調べると, 表 1 のようになった. ただし, べき乗法の収束判定定数は $e =$

[2]特異値 $\sigma_1, \sigma_2, ...$ も行列 $\boldsymbol{U}$, $\boldsymbol{V}$ も式 (19) 中に現れるものと同じになる.

[3]図中の枠は見やすくするために示したものであり, 3 次元復元には用いない.

表 1: 図 1 の画像列に対する再投影誤差が 0.1 画素になるまでの実行時間 (秒) と反復回数.

	基本法		双対法	
	時間	回数	時間	回数
原形	85,277	579	835	10
べき乗法	4,036	602	257	28

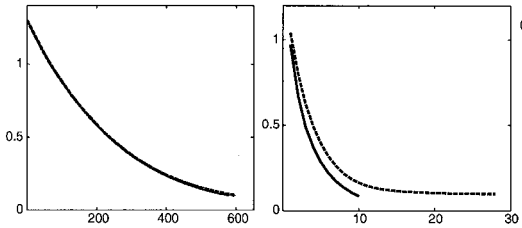

図 2: 反復回数と再投影誤差. 実線が原形, 破線がべき乗法. (a) 基本法, (b) 双対法.

1, $d = 5$ とした. CPU には Pentium 4 3.4GHz, 主メモリ 2GB, OS には Linux を用いた.

これからわかるように, べき乗法によって実行時間が基本法では 5%, 双対法では 31%に削減されている. ただし, 各反復で固有値計算を完全に収束させないために, 全体の反復回数はやや増えている. しかし, 各反復の計算量が減るので, 全体の実行時間は大幅に削減される.

これを確認するために, 図 2 は横軸に反復回数, 縦軸に再投影誤差をプロットした. 図 2(a) が基本法, 2(b) が双対法であり, 実線が原形, 破線がべき乗法である. 共に再投影誤差は単調に減少するが, べき乗法を用いると収束が長引く. これは双対法で著しい. それにもかかわらず実行時間は著しく減少する.

4.2 基本法と双対法の比較

表 1 からわかるように図 1 の例では双対法が基本法より効率的である. しかし, これは特徴点数 N とフレーム数 M に依存するので, 比較にはより詳細な評価が必要である.

基本法は各特徴点に対して固有値計算を行い, 双対法では各フレームに対して固有値計算を行うので, 実行時間は基本法はほぼ特徴点数 N に比例し, 双対法はほぼフレーム数 M に比例する.

しかし, 固有値計算の計算量を正確に見積もるのは困難である. おおざっぱに $n \times n$ 行列の固有値計算にほぼ n^3 に比例する演算回数が必要であるとすると, 原形では各反復に基本法で $O(NM^3)$ 回の演算, 双対法で $O(MN^3)$ 回の演算が必要である. べき乗法はベクトルと行列との積の計算から成るので, これら

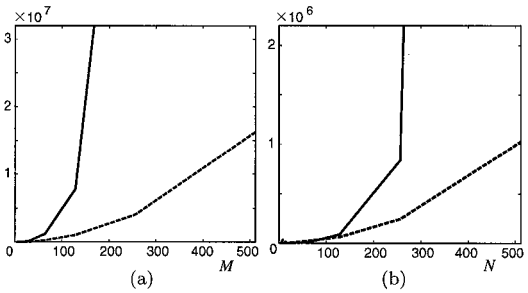

図 3: 原形 (実線) とべき乗 (破線) の実行時間 (秒) の増加の様子. (a) 基本法 ($N = 256$). 横軸はフレーム数 M. (b) 双対法 ($M = 256$). 横軸は特徴点数 N.

はほぼ $O(NM^2)$, $O(MN^2)$ 回になると予想される.

そこで図 1 の例で特徴点数 N とフレーム数 M を変化させて実行時間の M, N への依存性を調べた. 図 3(a) は基本法のフレーム数 M に対する実行時間, 図 3(b) は双対法の特徴点数 N に対する実行時間をプロットしたものである. 実線が原形, 破線がべき乗法によるものである.

M, N のいろいろな値に対する実行時間に指数関数を当てはめると, べき乗法を用いたときの基本法および双対法の実行時間 T_p, T_d はそれぞれほぼ次のようになった (単位は ms).

$$T_p \approx 1.407N^{0.94}M^{1.7}, \quad T_d \approx 0.163M^{0.95}N^{1.7} \quad (27)$$

これは予想がほぼ正しいこと, および特徴点数が多ければ基本法が, フレーム数が多ければ双対法が効率的であることを意味している. しかし, $N \approx M$ なら双対法が極めて有利である.

5. さらなる高速化

5.1 べき乗法の加速

任意の初期値 $\boldsymbol{\xi}^0$ から出発して, べき乗法によって半正値対称行列 $\boldsymbol{T}$ の最大固有値 λ_1 に対する単位固有ベクトル $\boldsymbol{u}_1$ を求めるとき, $\boldsymbol{\xi}^k = N[\boldsymbol{T}^k\boldsymbol{\xi}^0]$ と置くと, 初期値 $\boldsymbol{\xi}^0$ が $\boldsymbol{u}_1$ に十分近い, あるいは k が十分大きいとき,

$$\boldsymbol{\xi}^k \approx \boldsymbol{u}_1 + C\gamma^k\boldsymbol{u}_2, \qquad \gamma = \frac{\lambda_2}{\lambda_1} \quad (28)$$

となっている [1]. ただし, λ_2 は $\boldsymbol{T}$ の 2 番目に大きい固有値, $\boldsymbol{u}_2$ はそれに対する単位固有ベクトルである (C はある定数). 上式と上式の k を $k+1$ に置き換えたものとから $\boldsymbol{u}_2$ を消去すると, 関係

$$\boldsymbol{u}_1 \approx \frac{\boldsymbol{\xi}^{k+1} - \gamma\boldsymbol{\xi}^k}{1-\gamma} \quad (29)$$

を得る．γが既知ならこれによって$\boldsymbol{\xi}^k, \boldsymbol{\xi}^{k+1}$から$\boldsymbol{w}_1$を予測すれば収束が加速される．定数$\gamma$は式(28)より次のように推定できる．

$$\gamma \approx \frac{\|\boldsymbol{\xi}^{k+1} - \boldsymbol{\xi}^k\|}{\|\boldsymbol{\xi}^k - \boldsymbol{\xi}^{k-1}\|} \tag{30}$$

したがって，$\boldsymbol{\xi}^{k-1}, \boldsymbol{\xi}^k, \boldsymbol{\xi}^{k+1}$から式(30)によって$\gamma$を求め，$\boldsymbol{\xi}^{k+1}$を$N[(\boldsymbol{\xi}^{k+1} - \gamma\boldsymbol{\xi}^k)/(1-\gamma)]$に置き換えれば偶数回目の解が加速される（$\boldsymbol{\xi}^0, \boldsymbol{\xi}^1$から$\boldsymbol{\xi}^2$を加速し，$\boldsymbol{\xi}^2, \boldsymbol{\xi}^3$から$\boldsymbol{\xi}^4$を加速し，...）．これを射影的奥行きを表すベクトル$\boldsymbol{\xi}_\alpha, \boldsymbol{\xi}_\kappa$のべき乗法に適用する．

5.2 SOR法による部分空間当てはめの加速

SOR (successive overrelaxation) 法とは反復の収束を加速する方法であり，数列$\xi^1, \xi^2, \ldots$に対して，

$$\xi^k \leftarrow \xi^{k-1} + \omega(\xi^k - \xi^{k-1}) \tag{31}$$

とするものである．ω (≥ 1)は「加速定数」と呼ばれる．これが多くの反復問題で有効であることは経験的に知られている．しかし，性質のよく知られた線形反復問題以外は加速定数を適切に定めることが困難であり，経験的に定めるのが普通である．

これを射影的奥行きを表すベクトル$\boldsymbol{\xi}$ ($= \boldsymbol{\xi}_\alpha, \boldsymbol{\xi}_\kappa$)に適用する．すなわち，反復の各ステップで計算した$\boldsymbol{\xi}$を前ステップでの値$\boldsymbol{\xi}'$を用いて$N[\boldsymbol{\xi}' + \omega(\boldsymbol{\xi} - \boldsymbol{\xi}')]$に置き換える．

6. 実験2

表2は図1の画像列に対してべき乗法の加速を行なった場合，およびべき乗法はそのままで部分空間当てはめをSOR法で加速した場合，および両方を行なった場合の実行時間（秒）および反復回数を示す．ただし，加速したべき乗法に対しては収束判定定数を$e = 1, d = 1$とした．また部分空間当てはめをSOR法の加速定数は$\omega = 1.9$とした．なお，「効率化指数」とは基本形の時間がかかるほうと最も高速化されたものの実行時間の比である．

表2: 図1の画像列に対する再投影誤差が0.1画素になるまでの実行時間（秒）と反復回数．

	基本法		双対法	
	時間	回数	時間	回数
べき乗法	4,036	602	257	28
べき乗法の加速	4,083	599	**48**	11
SOR法	2,146	315	139	7
両方	2,112	312	76	14

効率化指数 = 1,777

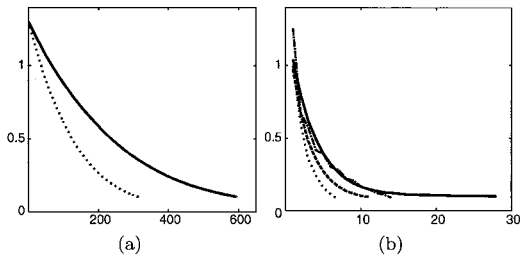

図4: 基本法(a)と双対法(b)に対する反復回数と再投影誤差．実線がべき乗法，破線がべき乗法の加速，点線が部分空間当てはめのSOR法，鎖線が両方を行なった場合．ただし(a)では破線は実線と，鎖線は点線と重なるので，実線と点線のみを示す．

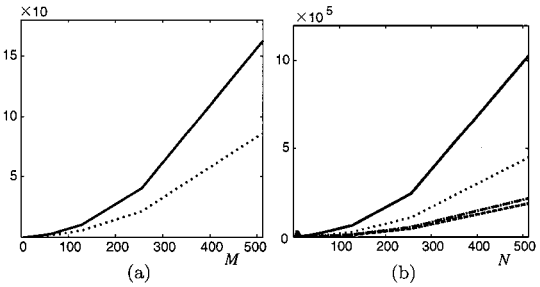

図5: 実行時間（秒）の増加の様子．実線がべき乗法，破線がべき乗法の加速，点線が部分空間当てはめのSOR法，鎖線が両方を行なった場合．(a) 基本法($N = 256$)．横軸はフレーム数M．(b) 双対法($M = 256$)．横軸は特徴点数N．ただし(a)では破線は実線と，鎖線は点線と重なるので，実線と点線のみを示す．

図4は反復回数による再投影誤差の減少を基本法と双対法に対してプロットしたものである．図4(a)の基本法では実線がべき乗法およびべき乗法の加速であり（両者に差はない），点線は部分空間当てはめのSOR法およびべき乗法の加速と併用した場合である（両者に差はない）．図4(b)の双対法では実線がべき乗法，破線がべき乗法の加速，点線が部分空間当てはめのSOR法，鎖線が両方を行なった場合である．

図5(a)は基本法($N = 256$)のフレーム数Mに対する実行時間の増加の様子を示し，図5(b)は双対法($M = 256$)の特徴点数Nに対する実行時間の増加の様子を示す．ともに実線がべき乗法，破線がべき乗法の加速のみ，点線が部分空間当てはめのSOR法，鎖線が両方を行なった場合である．これは図4の反復回数の差をそのまま反映している．

これらから，基本法ではべき乗法の加速はほとんど効果が見られないが，双対法では著しく効率化されていることがわかる．これは基本法ではべき乗法の収束がそもそも速く（γが小さく），双対法ではべ

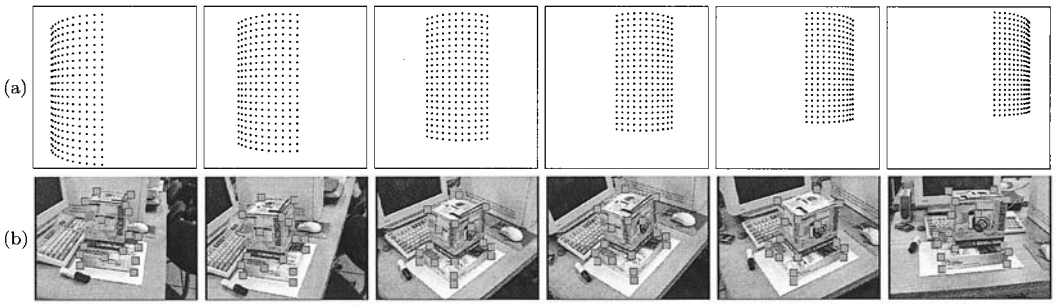

図 6: (a) シミュレーション画像系列（6 フレームを抜き出したもの）．フレーム数 11，特徴点数 231 個．(b) 実ビデオ画像（6 フレームを抜き出したもの）．フレーム数 200，特徴点数 16 個．

き乗法の収束が遅い（γ が大きい）という差によるものである．実際に γ の大きさを調べて，このことが確認された．

一方，SOR 法は基本法，双対法ともに効果が見られる．しかし，双対法ではべき乗法の加速の効果には及ばず，べき乗法の加速と併用するとべき乗法の加速の効果が打ち消される．

全体として，図 1 の画像列に対してはべき乗法を加速した双対法が最も効率的であり，実行時間が基本法の原形に対して約 0.06%（実行速度が約 1,800 倍）になっている．

7. 実験 3

図 6(a) は円筒面上に 231 個の特徴点をとり，視点を移動させながら 600×600 画素の画像面に焦点距離 $f = 600$（画素）の透視投影によって投影した 11 フレームの画像列である (抜き出した 6 フレームのみを示している)．

図 6(b) は 200 フレーム実ビデオ画像列 (640×480 画素) から抜き出した 6 フレームを示している．画像中には追跡した 16 個の特徴点をマークしている．これらは初期フレームに手動で指定し，以降のフレーム上を Kanade-Lucus-Tomasi の方法 [14] によって追跡したものである．ただし，追跡が途絶えたら手動で再追跡を開始した．

表 3 は図 6(a) の画像列にこれまでに述べた方法を適用した場合の，再投影誤差が 0.1 画素になるまでの実行時間（秒）と反復回数である．

図 6(a) のような特徴点数が多く ($N = 231$)，フレーム数が少ない場合 ($M = 11$) では原形を用いると，双対法のほうが反復回数が少ないにもかかわらず，1 回の反復の計算量が多いので，反復回数の多い基本法のほうが効率的になっている．しかし，べき乗法を用いると 1 回の反復の計算量が減り，双対法のほうが効率的になっている．

表 3: 図 6(a) の画像列に対する再投影誤差が 0.1 画素になるまでの実行時間（秒）と反復回数．

	基本法		双対法	
	時間	回数	時間	回数
原形	3.84	89	7.15	3
べき乗法	2.24	90	1.25	3
べき乗法の加速	2.37	90	**0.51**	3
SOR 法	1.12	47	10.76	21
両方	1.27	47	5.73	31

効率化指数 = 14

表 4: 図 6(b) の画像列に対する再投影誤差が 2.01 画素になるまでの実行時間（秒）と反復回数．

	基本法		双対法	
	時間	回数	時間	回数
原形	2,153.3	300	0.26	5
べき乗法	82.8	315	0.87	8
べき乗法の加速	81.3	314	**0.26**	5
SOR 法	43.4	165	1.85	15
両方	42.6	165	1.48	31

効率化指数 = 8,282

また実験 2 で観察したように，基本法ではべき乗法の収束が速いので，これを加速しても効果がない．しかし，SOR 法を用いるさらに効率化される．それに対してべき乗法の収束が遅い双対法では，べき乗法の加速が効果的である．その結果，実行時間が双対法の原形に対して約 0.7%（実行速度が約 14 倍）になっている．

しかし，SOR 法を用いるとかえって大幅に悪化し，べき乗法の加速と併用しても，べき乗法の加速のみに場合に及ばない．

表 4 は図 6(b) の実ビデオ画像列にこれまでに述べた方法を適用した場合の，再投影誤差が 2.01 画素[4]

[4]Kanade-Lucus-Tomasi の方法 [14] による特徴点追跡の精度には限界があり，2 画素程度のふらつきがある．このため反復回数をこれ以上に増やしても再投影誤差はこの値以下にはならなかっ

になるまでの実行時間（秒）と反復回数である．

図 6(b) のような特徴点数が少なく ($N = 16$)，フレーム数が多い場合 ($M = 200$) では基本法の原形は多大な実行時間を要する．べき乗法を用いるとこれは大幅に効率化され，実行時間が約 4%（実行速度が約 26 倍）になる．また他の例と同様に，べき乗法は加速しても効果がないが SOR 法によってさらに効率化される．

しかし，それでも双対法の原形のほうが圧倒的に効率的である．これは収束が極めて速いので，べき乗法を用いるとかえって非効率になる．ただし，べき乗法を加速すると原形と同等になり，実行時間が基本法の原形に対して約 0.01%（実行速度が約 8,000 倍）になっている．

一方，SOR 法を用いると悪化し，べき乗法の加速と併用してもあまり改善されない．

8. まとめ

本論文では，未校正カメラで撮影した画像列上の特徴点の追跡から 3 次元形状を計算する自己校正法において最も計算時間を要する射影復元の反復を高速化する手法を示した．そして，ミュレーションや実ビデオ画像によって次のことを確認した．

1. 固有値計算をべき乗法に置き換えると反復回数は増加するが，実行時間が大幅に削減される．特に基本法に対して効果が著しい．
2. 基本法はべき乗法の収束が速い（第 2 固有値が小さい）ので加速しても効果がない．それに対して，双対法ではべき乗法の収束が遅い（第 2 固有値が大きい）ので加速するとさらに効率化される．
3. 部分空間の当てはめの反復に SOR 法を適用すると収束が早まるが，双対法では効果が少ない．
4. 基本法は特徴点数が著しく多く，フレーム数が少ないとき有利であり，双対法は特徴点数が少なく，フレーム数が著しく多いとき有利である．
5. 実際的な状況では双対法のほうが著しく効率的であり，加速したべき乗法と組み合わせるのが最善である．

ただし，実行時間は部分反復の収束判定定数や加速定数の選択にも依存する．これらの真に最適な値を定めることは未解決の問題である．

謝辞: 有益なコメントを頂いた NTT の宮川勲氏に感謝します．本研究の一部は文部科学省科学研究費基盤研究 C (No. 17500112) の助成によった．

た [10].

参考文献

[1] 甘利俊一, 金谷健一, 「線形代数」, 講談社, 1987.

[2] R. I. Hartley, In defense of the eight-point algorithm, *IEEE Trans. Patt. Anal. Mach. Intell.*, **19**-6 (1997-6), 580–593.

[3] R. Hartley and F. Schaffalitzky, PowerFactorization: 3D reconstruction with missing or uncertain data, *Proc. the Australia-Japan Advanced Workshop on Computer Vision*, September 2003, Adelaide, Australia, pp. 1–9.

[4] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, U.K., 2000.

[5] A. Heyden, R. Berthilsson, and G. Sparr, An iterative factorization method for projective structure and motion from image sequences, *Image Vision Comput.*, **17**-13 (1999-11), 981–991.

[6] 金出武雄, コンラッド・ポールマン, 森田俊彦, 因子分解法による物体形状とカメラ運動の復元, 電子情報通信学会論文誌 D-II, **J74-D-II**-8 (1993-8), 1497–1505.

[7] 金谷 健一, 森 昭延, 菅谷 保之, 自己校正法の最新レシピ, 情報処理学会研究報告, 2006-CVIM-153-31 (2006-3), 199–206.

[8] 金谷健一, 菅谷保之, 因子分解法の完全レシピ, 電子情報通信学会技術報告, PRMU2003-118 (2003-10), 19–24.

[9] S. Mahamud and M. Hebert, Iterative projective reconstruction from multiple views, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, June 2000, Hilton Head Island, SC, U.S.A., Vol. 2, pp. 430–437.

[10] 森 昭延, 金谷健一, 菅谷保之, 最新の自己校正法の性能評価, 情報処理学会研究報告, 2006-CVIM-154-37 (2006-5), pp. 347–354.

[11] T. Morita and K. Kanade, A sequential factorization method for recovering shape and motion from image streams, *IEEE Trans. Patt. Anal. Mach. Intell.*, **19**-8 (1997-8), 858–867.

[12] C. J. Poelman and T. Kanade, A paraperspective factorization method for shape and motion recovery, *IEEE Trans. Patt. Anal. Mach. Intell.*, **19**-3 (1997-3), 206–218.

[13] Y. Seo and H. Heyden, Auto-calibration by linear iteration using the DAC equation, *Image and Vision Computing*, **22**-11 (2004-9), 919–926.

[14] C. Tomasi and T. Kanade, *Detection and Tracking of Point Features*, CMU Tech. Rep. CMU-CS-91-132, Apr. 1991;
`http://vision.stanford.edu/~birch/klt/`.

[15] C. Tomasi and T. Kanade, Shape and motion from image streams under orthography—A factorization method, *Int. J. Comput. Vision*, **9**-2 (1992-10), 137–154.

[16] R. Vidal and R. Hartley, Motion segmentation with missing data using PowerFactorization and GPCA, *Proc. IEEE Comput. Vision Patt. Recog.*, June-July 2004, Washington, D.C., Vol. 2, pp. 310–316.