

自己校正法のための射影復元の計算量削減

ハノ・アッカーマン 新妻 弘崇 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効率化する新しい手法を示す。著者らは前報で反復計算を加速する工夫を行って著しい高速化を達成したが、本論文では計算量を注意深く解析し、計算過程を変形することによって演算回数を削減する。そして、シミュレーションや実ビデオ画像によって前報の方法と同等、あるいはそれ以上の効率化が実現されることを示す。

Reducing the Complexity of Projective Reconstruction for Self-Calibration

Hanno Ackermann, Hirotaka Niitsuma, and Kenichi Kanatani

Department of Computer Science, Okayama University, Okayama 700-8530 Japan

We present a new method for efficient computation of projective reconstruction, which dominates the execution time of self-calibration for 3-D reconstruction from feature point tracking over a video sequence. In our previous paper, we achieved significant speedup by accelerating iteration computations. In this paper, we reduce the number of arithmetic operations by carefully analyzing the computational complexity and modifying the procedure. Using simulated and real video images, we demonstrate that our method can achieve efficiency equivalent to or even higher than our previous method.

1. まえがき

前報 [9] 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の高速度の手法を示した。自己校正法は、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段階から成るが、前者の反復計算に多大な計算時間を要する [10]。それは、各反復ステップで大次元行列の固有値問題を解く必要があるためである。

前報 [9] では、解が前回の反復での値とほぼ同じであることに着目して、固有値計算法として前回の値を初期値とするべき乗法を導入した。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計算の加速を試みた。さらに、物理学でよく用いられるSOR法を適用した。そして、シミュレーションや実ビデオ画像実験により、場合によっては実行速度が数1000倍に向上することを実証した。

本論文では射影復元のアルゴリズムの計算量をさらに詳細に解析し、計算の順序を変えるなどの工夫によって演算回数の削減を行う。そしてシミュレーションや実ビデオ画像実験を行い、前報 [9] の高速化技法

に匹敵する高速化が達成できることを示す。

2. 射影復元の方法

射影復元の基本は次式である [2, 6, 9, 10].

$$z_{\kappa\alpha} \mathbf{x}_{\kappa\alpha} = \mathbf{\Pi}_{\kappa} \mathbf{X}_{\alpha}, \quad \mathbf{x}_{\kappa\alpha} = \begin{pmatrix} x_{\kappa\alpha}/f_0 \\ y_{\kappa\alpha}/f_0 \\ 1 \end{pmatrix} \quad (1)$$

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、 $\mathbf{\Pi}_{\kappa}$ は第 κ フレームの 3×4 投影行列、 \mathbf{X}_{α} 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 $z_{\kappa\alpha}$, $\mathbf{\Pi}_{\kappa}$, \mathbf{X}_{α} 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 [8] の方法（「基本法」と呼ぶ）とHeydenら [3] 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。以下、ベクトル \mathbf{a} , \mathbf{b} の内積を (\mathbf{a}, \mathbf{b}) と書く。また、 $N[\cdot]$ は単

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

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

位ベクトルへの正規化を, $Z[\cdot]$ はベクトルの第 3 成分を 1 とする正規化を表す. 以下の導出の詳細は文献 [6, 10] 参照.

2.1 基本法 (原形)

入力: • データベクトル $\mathbf{x}_{\kappa\alpha}$, そのノルム $\|\mathbf{x}_{\kappa\alpha}\|$,
 および正規化データ $N[\mathbf{x}_{\kappa\alpha}] = \mathbf{x}_{\kappa\alpha}/\|\mathbf{x}_{\kappa\alpha}\|$,
 $\kappa = 1, \dots, M, \alpha = 1, \dots, N$
 • 収束判定の再投影誤差 E_{\min} (画素)

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

計算:

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

$$\mathbf{M} = \sum_{\alpha=1}^N \mathbf{p}_\alpha \mathbf{p}_\alpha^\top \quad (2)$$

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

$$\mathbf{\Pi}_\kappa = \begin{pmatrix} \mathbf{u}_{1\kappa} & \mathbf{u}_{2\kappa} & \mathbf{u}_{3\kappa} & \mathbf{u}_{4\kappa} \end{pmatrix} \quad (3)$$

ただし $\mathbf{u}_{i\kappa}$ は \mathbf{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 $\alpha = 1, \dots, N$ に渡って計算する.

- (a) 次のように計算した $M \times M$ 行列 $\mathbf{A}^\alpha = (A_{\kappa\lambda}^\alpha)$ 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.

$$A_{\kappa\lambda}^\alpha = \sum_{k=1}^4 (N[\mathbf{x}_{\kappa\alpha}], \mathbf{u}_{k\kappa})(N[\mathbf{x}_{\lambda\alpha}], \mathbf{u}_{k\lambda}) \quad (4)$$

固有ベクトルの符号は次のように定める.

$$\sum_{\kappa=1}^M \xi_{\kappa\alpha} \geq 0 \quad (5)$$

- (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.

$$z_{\kappa\alpha} = \frac{\xi_{\kappa\alpha}}{\|\mathbf{x}_{\kappa\alpha}\|} \quad (6)$$

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

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

$$X_\alpha^k = (\mathbf{p}_\alpha, \mathbf{u}_k), \quad k = 1 \sim 4 \quad (7)$$

7. 次のように再投影誤差 E を計算する.

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

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: • データベクトル $\mathbf{x}_{\kappa\alpha}$, そのノルム $\|\mathbf{x}_{\kappa\alpha}\|$,
 および正規化データ $N[\mathbf{x}_{\kappa\alpha}] = \mathbf{x}_{\kappa\alpha}/\|\mathbf{x}_{\kappa\alpha}\|$,
 $\kappa = 1, \dots, M, \alpha = 1, \dots, N$
 • 収束判定の再投影誤差 E_{\min} (画素)

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

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 \mathbf{q}_κ^i を計算する.

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

そして各 κ ごとに $\mathbf{q}_\kappa^i, i = 1, 2, 3$ 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.

$$\sum_{i=1}^3 \|\mathbf{q}_\kappa^i\|^2 = 1 \quad (10)$$

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

$$\mathbf{N} = \sum_{\kappa=1}^M \sum_{i=1}^3 \mathbf{q}_\kappa^i \mathbf{q}_\kappa^{i\top} \quad (11)$$

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

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

6. 次の計算を $\kappa = 1, \dots, M$ に渡って計算する.

- (a) 次のように計算した $N \times N$ 行列 $\mathbf{B}^\kappa = (B_{\alpha\beta}^\kappa)$ 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.

$$B_{\alpha\beta}^\kappa = (\mathbf{v}_\alpha, \mathbf{v}_\beta)(N[\mathbf{x}_{\kappa\alpha}], N[\mathbf{x}_{\kappa\beta}]) \quad (13)$$

ただし v_α は基底ベクトル v_1, v_2, v_3, 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。

$$\sum_{\alpha=1}^N \xi_{\kappa\alpha} \geq 0 \quad (14)$$

- (b) 得られた ξ_{κ} から射影的奥行き $z_{\kappa\alpha}$ を次のように計算する。

$$z_{\kappa\alpha} = \frac{\xi_{\kappa\alpha}}{\|\mathbf{x}_{\kappa\alpha}\|} \quad (15)$$

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

$$\Pi_{\kappa(ij)} = (\mathbf{q}_{\kappa}^i, \mathbf{v}_j) \quad (16)$$

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。

2.3 因子分解法との計算量の比較

基本法でも双対法でも、ステップ 3, 4 がデータに 4 次元部分空間を当てはめる計算、ステップ 6 が各々の射影的奥行き $z_{\kappa\alpha}$ を更新する計算であり [6, 10], この二つを反復する。この意味でこれは一種の EM アルゴリズムである。ステップ 1 で $z_{\kappa\alpha} = 1$ と置くことはアフィンカメラ仮定することに等価である。

基本法で、ステップ 5 まで進んで投影行列 $\mathbf{\Pi}_{\kappa}$ を計算した段階で終了し、式 (7) で 3 次元位置 \mathbf{X}_{α} を計算する、あるいは同じことであるが、双対法でステップ 5 まで進んで 3 次元位置 \mathbf{X}_{α} を計算した段階で終了し、式 (16) で投影行列 $\mathbf{\Pi}_{\kappa}$ を計算する²のが因子分解法 [4, 12] の前段 (「アフィン復元」) にほかならない [7]。この意味で、射影復元は因子分解法の自然な反復的拡張である。

因子分解法の後段では平行投影、弱透視投影、疑似透視投影などの特定のアフィンカメラモデルに基づく計量拘束条件を用いてユークリッド化を行うのに対して [7]、自己校正法では透視投影カメラモデルに基づく双対絶対 2 次曲面拘束条件を用いてユークリッド化を行う [6]。しかし、両者の計算の構造は同じであり (基本的に最小二乗法)、ほぼ同じ計算量である。

² どちらも特異値分解による行列の因子分解に帰着することから「因子分解法」という名称が生まれた。

以上から分かるように、自己校正法と因子分解法の計算量の違いは基本法、双対法のステップ 6(a) の固有値計算を行うかどうか、そしてステップ 3-8 を再投影誤差が収束するまで反復するかどうかである。したがって、この計算を高速化することが自己校正法を効率化するための課題となる。

3. 部分空間当てはめの効率化

3.1 基底の計算

まず考えられる工夫は、当てはめる 4 次元部分空間の基底を計算するためのステップ 3, 4 の固有値計算である。そのために次の事実に着目する。それは $3M \times N$ 行列 \mathbf{P} を

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_N \end{pmatrix} \quad (17)$$

と定義すれば、式 (2) の行列 \mathbf{M} と式 (11) の行列 \mathbf{N} が次のように書けることである。

$$\mathbf{M} = \mathbf{P}\mathbf{P}^T, \quad \mathbf{N} = \mathbf{P}^T\mathbf{P} \quad (18)$$

そこで \mathbf{M}, \mathbf{N} の固有ベクトルを直接に計算する代わりに \mathbf{P} の特異値分解を計算する。

$$\mathbf{P} = \mathbf{U}\text{diag}(\sigma_1, \sigma_2, \dots)\mathbf{V}^T \quad (19)$$

すると、 \mathbf{M} の大きい 4 個の固有値に対する単位固有ベクトル $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ は行列 \mathbf{U} の最初の 4 列に等しく、 \mathbf{N} の大きい 4 個の固有値に対する単位固有ベクトル $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ は行列 \mathbf{V} の最初の 4 列に等しい。そこで計算量を比較する。

まず基本法では \mathbf{M} の計算 (式 (2) と式 (18) は同等) に $(3M)^2N$ 回の乗算³が必要である⁴。固有値計算の計算量は用いるアルゴリズムによって異なるが、 $3M \times 3M$ 行列に対してほぼ $(3M)^3$ と見積ると、前者の計算量 (乗算の回数) は $(3M)^2N + (3M)^3 = (3M)^2(3M + N)$ となる。

特異値分解の計算量もアルゴリズムによって異なるが、特異値分解は行列を転置しても基本的に同じであり (式 (19) の行列 \mathbf{U} と行列 \mathbf{V} が入れ替わるだけ)、計算に有利なほうを選ぶのが普通である。したがって、 $3M \times N$ 行列では $3M \geq N$ ならほぼ $3MN^2$, $3M \leq N$ ならほぼ $(3M)^2N$ である。いずれにしても特異値分解のほうが有利である。

³ 行列やベクトルの計算は積の和の計算からなり、加減算の回数は (項数) - 1 であるが、-1 は無視すると乗算の回数と加減算の回数が等しい。そこで以下、乗算のみを数える。

⁴ \mathbf{M} は対称行列であるから、実際には対角要素と上三角部分のみの計算でよい。下三角部分は上三角部分をコピーすればよい。こうすると計算量は $1.5M(3M + 1)N$ に減る。

双対法でも同様に、計算量は行列 N の計算に $3MN^2$ であり⁵、その固有値計算にほぼ N^3 、合計 $3MN^2 + N^3 = N^2(3M + N)$ となる。しかし、特異値分解の計算量は基本法と同じであり、やはり特異値分解を用いるほうが有利であると結論される。

3.2 基底の更新の計算

上記の固有値計算は反復されるので、前報 [9] のようにべき乗法を用いるのが合理的である。基本法では固有ベクトル $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ の更新が行列 M との積 $M\mathbf{u}_k, k = 1 \sim 4$ に帰着される。これは式 (18) を利用すると、次のように分解できる。

$$\mathbf{v}_k = \mathbf{P}^\top \mathbf{u}_k, \quad \mathbf{u}_k = \mathbf{P}\mathbf{v}_k \quad (20)$$

直接的な計算では、まず乗算が M の計算に $(3M)^2N$ 回⁶必要であり、反復ごとに積 $M\mathbf{u}_k, k = 1 \sim 4$ の計算に乗算が $4(3M)^2$ 回必要である。それに対して式 (20) では反復ごとに $4(3MN + 3MN) = 24MN$ 回の乗算となる。したがって、 $M \geq 2N/3$ であれば反復回数に無関係に (20) の計算が有利となる。しかし、 $M < 2N/3$ の場合でも、直接計算が有利となるのは反復回数が非常の多い場合であり ($N = \infty$ の極限でその最小値が $M/3$ 回となる)、通常はほぼ常に (20) の計算が有利となる (対称性を利用しても同じ結論となる)。

双対法では固有ベクトル $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ の更新が行列 N との積 $N\mathbf{v}_k, k = 1 \sim 4$ に帰着される。これは式 (18) を利用すると、次のように分解できる。

$$\mathbf{u}_k = \mathbf{P}\mathbf{v}_k, \quad \mathbf{v}_k = \mathbf{P}^\top \mathbf{u}_k \quad (21)$$

基本法と同様に、直接計算は行列 N の計算に $3MN^2$ 回⁷の乗算、毎回の反復に $4N^2$ 回の乗算が必要であるが、式 (21) の計算では毎回の反復に乗算が $4(3MN + 3MN) = 24MN$ 回となる。ゆえに $N \geq 6M$ なら反復回数に無関係に式 (21) の計算が有利となり、 $N < 6M$ でも反復回数が多くない限り ($M = \infty$ の極限で最小値が $N/8$ 回)、式 (21) の計算が有利となる (対称性を利用しても同じ結論となる)。

4. 射影的奥行き計算の効率化

前節の解析にはやや大雑把のところがあったが、詳細に解析しても全体の効率にはそれほど影響しない。

⁵これも N が対称行列であることを利用すると $1.5MN(N+1)$ に減る。

⁶対称性を利用すれば $1.5M(3M+1)N$ 回。

⁷対称性を利用すると $1.5MN(N+1)$ 回。

というのは、部分空間の当てはめは全データに対する計算であり、射影復元の反復で毎回 1 回計算すればよいからである。

それに対して射影的奥行き $z_{\kappa\alpha}$ のための固有値計算は各反復につき基本法では特徴点ごとに N 回、双対法ではフレームごとに M 回計算する必要がある。実測しても、この射影的奥行き計算が全計算の圧倒的な割合を占めることが確認される。

4.1 基本法

次の $C_{\kappa i}^\alpha$ を (κi) 要素とする $M \times 4$ 行列 C^α を定義する。

$$C_{\kappa i}^\alpha = (N[\mathbf{x}_{\kappa\alpha}], \mathbf{u}_{i\kappa}) \quad (22)$$

すると、式 (4) で定義される $M \times M$ 行列 A^α が次のように書ける。

$$A^\alpha = C^\alpha C^{\alpha\top} \quad (23)$$

次の 4×4 行列 D^α を考える。

$$D^\alpha = C^{\alpha\top} C^\alpha \quad (24)$$

この行列の最大固有値に対する固有ベクトルを $\boldsymbol{\mu}_\alpha$ とすると、式 (23) の行列 A^α の最大固有値に対する固有ベクトル $\boldsymbol{\xi}_\alpha$ は次のように計算できる⁸[5]。

$$\boldsymbol{\xi}_\alpha = C^\alpha \boldsymbol{\mu}_\alpha \quad (25)$$

両者の計算量を比較する。式 (22) の $C_{\kappa i}^\alpha$ の計算は両方に共通するので、これは数えない。すると、直接計算の乗算は、式 (23) の A^α の計算に $4M^2$ 回⁹、その固有値計算にほぼ M^3 回、合計 $4M^2 + M^3 = M^2(M+4)$ 回である。

一方、分解する方法の乗算は、式 (24) の D^α の計算に $16M$ 回¹⁰、その固有値計算にほぼ 64 回、式 (25) に $4M$ 回、合計 $20M + 64$ 回となる。これは $M \geq 5$ であれば前者より圧倒的に少ない (対称性を利用しても同じ結論となる)。

自己校正法では通常 5 フレーム以上用いることが多いので、常に後者が有利であるといえる。要するに、5 フレーム以上用いれば 5 次元より大きい行列の固有値計算しなければならないのに、分解する方法では常に 4×4 行列の固有値計算をすればよいからである。

⁸ $\boldsymbol{\mu}_\alpha$ は $D^\alpha \boldsymbol{\mu}_\alpha = \lambda \boldsymbol{\mu}_\alpha$ すなわち $C^{\alpha\top} C^\alpha \boldsymbol{\mu}_\alpha = \lambda \boldsymbol{\mu}_\alpha$ を満たすから、 C^α を両辺に掛けると $C^\alpha C^{\alpha\top} (C^\alpha \boldsymbol{\mu}_\alpha) = \lambda (C^\alpha \boldsymbol{\mu}_\alpha)$ 。すなわち、式 (23) より式 (25) の $\boldsymbol{\xi}_\alpha$ が $A^\alpha \boldsymbol{\xi}_\alpha = \lambda \boldsymbol{\xi}_\alpha$ を満たす。

⁹対称性を利用すれば $2M(M+1)$ 回。

¹⁰対称性を利用すれば $10M$ 回。

その 4×4 行列 D^α の固有値計算にべき乗法やその加速を用いることも可能であるが、普通の計算 (例えば Householder 法) が十分速いので、もはや効率化されないことが実験的に確認される。

4.2 双対法

双対法の固有値計算に現れるのは式 (13) で定義される $N \times N$ 行列 $B^\kappa = (B_{\alpha\beta}^\kappa)$ である。これは式 (1) の x_α の定義より次の変形ができる。

$$B_{\alpha\beta}^\kappa = \left(\frac{x_{\kappa\alpha} v_\alpha}{f_0 \|x_{\kappa\alpha}\|}, \frac{x_{\kappa\beta} v_\beta}{f_0 \|x_{\kappa\beta}\|} \right) + \left(\frac{y_{\kappa\alpha} v_\alpha}{f_0 \|x_{\kappa\alpha}\|}, \frac{y_{\kappa\beta} v_\beta}{f_0 \|x_{\kappa\beta}\|} \right) + \left(\frac{v_\alpha}{\|x_{\kappa\alpha}\|}, \frac{v_\beta}{\|x_{\kappa\beta}\|} \right) \quad (26)$$

$N \times 4$ 行列 $C^{\kappa 1} = (C_{\alpha i}^{\kappa 1})$, $C^{\kappa 2} = (C_{\alpha i}^{\kappa 2})$, $C^{\kappa 3} = (C_{\alpha i}^{\kappa 3})$ を

$$C_{\alpha i}^{\kappa 1} = \frac{x_{\kappa\alpha} v_{\alpha i}}{f_0 \|x_{\kappa\alpha}\|}, \quad C_{\alpha i}^{\kappa 2} = \frac{y_{\kappa\alpha} v_{\alpha i}}{f_0 \|x_{\kappa\alpha}\|}, \quad C_{\alpha i}^{\kappa 3} = \frac{v_{\alpha i}}{\|x_{\kappa\alpha}\|} \quad (27)$$

と定義すると、式 (26) は次のように書ける。

$$B^\kappa = C^{\kappa 1} C^{\kappa 1 T} + C^{\kappa 2} C^{\kappa 2 T} + C^{\kappa 3} C^{\kappa 3 T} = C^\kappa C^{\kappa T} \quad (28)$$

ただし、 $N \times 12$ 行列 C^κ を次のように定義した。

$$C^\kappa = \begin{pmatrix} C^{\kappa 1} & C^{\kappa 2} & C^{\kappa 3} \end{pmatrix} \quad (29)$$

したがって、式 (26) の行列 B^κ の最大固有値に対する固有ベクトル ξ_κ は 12×12 行列

$$D^\kappa = C^{\kappa T} C^\kappa \quad (30)$$

の最大固有値に対する固有ベクトル η^κ を求めれば次のように計算される (脚注 8 参照) [5]。

$$\xi_\kappa = C^\kappa \eta^\kappa \quad (31)$$

両者の計算量を比較する。式 (27) の $C_{\kappa i}^{\alpha k}$, $k = 1 \sim 4$ の計算は両方に共通するので、これは数えない。すると、直接計算の乗算は、式 (28) の B^α 計算に $12N^2$ 回¹¹、その固有値計算にほぼ N^3 回、合計 $12N^2 + N^3 = N^2(N + 12)$ 回である。

一方、分解する方法の乗算は、式 (30) の D^κ の計算は $144N$ 回¹²、その固有値計算にほぼ 1728 回、式 (31) に $12N$ 回、合計 $156N + 1728$ 回となる。これは $N \geq 13$ なら前者より圧倒的に少ない (対称性を利用して同じ結論となる)。

¹¹ 対称性を利用すれば $6N(N + 1)$ 回。

¹² 対称性を利用すれば $78N$ 回。

しかし、特徴点追跡は困難な処理であり、12 個以下しか特徴点が得られないこともある。その場合は直接計算のほうが有利である。要するに、12 個以下しか特徴点がない場合は 12 次元より小さい行列の固有値計算すればよいのに、分解する方法では常に 12×12 行列の固有値計算をしなければならないからである。ただし、少ない特徴点といっても通常 $8 \sim 10$ 個程度は必要であり、直接計算による効率向上の効果はわずかである。したがって、これは考慮しなくてもよいと思われる。

その 12×12 行列 D^κ の固有値計算にべき乗法やその加速を用いることも可能であるが、やはり普通の計算が十分速いので、特にその必要性がないことが実験的に確認される。

5. 効率的な射影復元

前節の解析を考慮すると、射影復元の効率的な計算手順は次のようになる。

5.1 基本法 (効率化)

入力: • データベクトル $x_{\kappa\alpha}$, そのノルム $\|x_{\kappa\alpha}\|$,
および正規化データ $N[x_{\kappa\alpha}] = x_{\kappa\alpha} / \|x_{\kappa\alpha}\|$,
 $\kappa = 1, \dots, M, \alpha = 1, \dots, N$

• 収束判定の再投影誤差 E_{\min} (画素)

• 部分空間当てはめの収束判定定数 e

出力: $\Pi_\kappa, \kappa = 1, \dots, M, X_\alpha, \alpha = 1, \dots, N$

計算:

1. 射影的奥行きを $z_{\kappa\alpha} = 1$ と初期化する。
2. $z_{1\alpha} x_{1\alpha}, z_{2\alpha} x_{2\alpha}, \dots, z_{M\alpha} x_{M\alpha}$ を縦に並べた $3M$ 次元ベクトル p_α を計算し、単位ベクトルに正規化する。
3. 式 (17) の $3M \times N$ 行列 P を計算する
4. 行列 P を式 (19) のように特異値分解し、 U の最初の 4 列 u_1, u_2, u_3, u_4 を計算する。
5. 3×4 行列 Π_κ 式 (3) のように計算する。
6. 次の計算を $\alpha = 1, \dots, N$ に渡って計算する。

(a) 次の $C_{\kappa i}^\alpha$ を (κi) 要素とする $M \times 4$ 行列 C^α を計算する。

$$C_{\kappa i}^\alpha = (N[x_{\kappa\alpha}], u_{i\kappa}) \quad (32)$$

(b) 次の 4×4 行列 $D^\alpha = (D_{ij}^\alpha)$ を計算する。

$$D_{ij}^\alpha = \sum_{\kappa=1}^M C_{\kappa i}^\alpha C_{\kappa j}^\alpha \quad (33)$$

ただし $i \leq j$ の (ij) 要素のみ計算し、 $i > j$ の (ij) 要素は (ji) 要素の値をコピーする。

- (c) 行列 D^α の最大固有値に対する単位固有ベクトルを μ_α とし、次のように ξ_α を計算する。

$$\xi_\alpha = \pm N[C^\alpha \mu_\alpha] \quad (34)$$

符号を式 (5) が成り立つように選ぶ。

- (d) 得られた ξ_α 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 p_α を再計算し、正規化し、式 (17) の行列 P を再計算する。

- (f) 3次元位置 $X_\alpha = (X_\alpha^k)$ を式 (7) のように計算する。

7. 式 (8) の再投影誤差 E を計算し、 $E < E_{\min}$ であれば終了する。
8. 次の $3M$ 次元ベクトル \hat{u}_k を計算する ($k = 1 \sim 4$)。)

$$\hat{u}_i = P(P^\top u_i) \quad (35)$$

9. $\hat{u}_1, \hat{u}_2, \hat{u}_3, \hat{u}_4$ にシュミットの直交化を施したものを $\hat{u}_1, \hat{u}_2, \hat{u}_3, \hat{u}_4$ とする。
10. $\max_{k=1}^4 \sqrt{1 - \sum_{l=1}^4 (\hat{u}_k, u_l)^2} < 10^{-e}$ なら $u_k \leftarrow \hat{u}_k, k = 1 \sim 4$ としてステップ 5 に戻る。そうでなければ $u_k \leftarrow \hat{u}_k, k = 1 \sim 4$ としてステップ 8 に戻る。

5.2 双対法 (効率化)

- 入力:** • データベクトル $x_{\kappa\alpha}$, そのノルム $\|x_{\kappa\alpha}\|$, および正規化データ $N[x_{\kappa\alpha}] = x_{\kappa\alpha}/\|x_{\kappa\alpha}\|$, $\kappa = 1, \dots, M, \alpha = 1, \dots, N$
• 収束判定の再投影誤差 E_{\min} (画素)
• 部分空間当てはめの収束判定定数 e

出力: $\Pi_\kappa, \kappa = 1, \dots, M, X_\alpha, \alpha = 1, \dots, N$

計算:

1. 射影的奥行きを $z_{\kappa\alpha} = 1$ と初期化する。
2. 式 (9) の N 次元ベクトル q_κ^i を計算し、各 κ ごとに $q_\kappa^i, i = 1, 2, 3$ を式 (10) のように正規化する。
3. 次の $N \times 3M$ 行列 Q を計算する。

$$Q = \begin{pmatrix} q_1^1 & q_1^2 & q_1^3 & q_2^1 & \cdots & q_M^3 \end{pmatrix} \quad (36)$$

4. 行列 Q を次のように特異値分解し,¹³ 行列 V の最初の 4 列 v_1, v_2, v_3, v_4 とする。

$$Q = V \text{diag}(\sigma_1, \sigma_2, \dots) U^\top \quad (37)$$

¹³特異値 $\sigma_1, \sigma_2, \dots$ も行列 U, V も式 (19) 中に現れるものと同じになる。

5. 3次元位置 X_α を式 (12) のように計算する。
6. 次の計算を $\kappa = 1, \dots, M$ に渡って計算する。

- (a) 次のように 12×12 行列 D^κ を計算する。
• 次の 4次元ベクトル w_α を計算する。

$$w_\alpha = \frac{v_\alpha}{f_0 \|x_{\kappa\alpha}\|} \quad (38)$$

- $N \times 4$ 行列 $C^{\kappa 1} = (C_{\alpha i}^{\kappa 1}), C^{\kappa 2} = (C_{\alpha i}^{\kappa 2}), C^{\kappa 3} = (C_{\alpha i}^{\kappa 3})$ を次のように定義する。

$$\begin{aligned} C_{\alpha i}^{\kappa 1} &= x_{\kappa\alpha} w_{\alpha i}, & C_{\alpha i}^{\kappa 2} &= y_{\kappa\alpha} w_{\alpha i}, \\ C_{\alpha i}^{\kappa 3} &= f_0 w_{\alpha i} \end{aligned} \quad (39)$$

- $C^{\kappa 1}, C^{\kappa 2}, C^{\kappa 3}$ を並べた式 (29) の $N \times 12$ 行列 C^κ を定義する。
• 式 (30) の行列 D^κ を計算する。

$$D^\kappa = C^{\kappa \top} C^\kappa \quad (40)$$

ただし $i \leq j$ の (ij) 要素のみ計算し、 $i > j$ の (ij) 要素は (ji) 要素の値をコピーする。

- (b) 行列 D^κ の最大固有値に対する単位固有ベクトルを η_κ とし、次のように ξ_κ を計算する。

$$\xi_\kappa = \pm N[C^\kappa \eta_\kappa] \quad (41)$$

符号は式 (14) が成り立つように選ぶ。

- (c) 得られた ξ_κ から射影的奥行き $z_{\kappa\alpha}$ を式 (15) のように計算する。
(d) 得られた $z_{\kappa\alpha}$ を用いて式 (9) のベクトル q_κ^i を計算し、式 (10) のように正規化し、式 (36) の行列 Q を計算する。
(e) 3×4 行列 $\Pi_\kappa = (\Pi_{\kappa(ij)})$ を式 (16) のように計算する。

7. 式 (8) のように再投影誤差 E を計算し、 $E < E_{\min}$ であれば終了する。
8. 次の N 次元ベクトル \tilde{v}_k を計算する ($k = 1 \sim 4$)。)

$$\tilde{v}_k = Q(Q^\top v_k) \quad (42)$$

9. $\tilde{v}_1, \tilde{v}_2, \tilde{v}_3, \tilde{v}_4$ にシュミットの直交化を施したものを $\hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_4$ とする。
10. $\max_{k=1}^4 \sqrt{1 - \sum_{l=1}^4 (\hat{v}_k, v_l)^2} < 10^{-e}$ なら $v_k \leftarrow \hat{v}_k, k = 1 \sim 4$ としてステップ 5 に戻る。そうでなければ $v_k \leftarrow \hat{v}_k, k = 1 \sim 4$ としてステップ 8 に戻る。

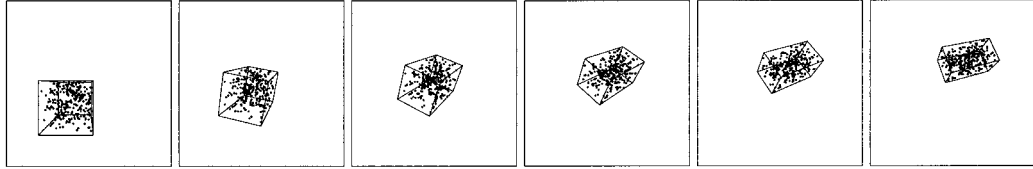


図 1: 直方体領域中の点の運動.

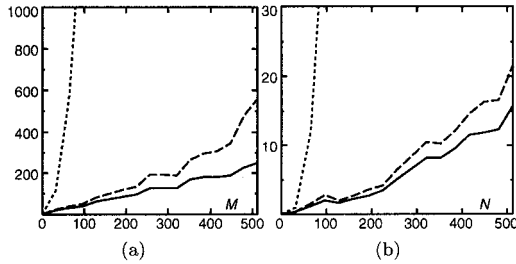


図 2: 原形 (点線), べき乗 (破線), および効率化法 (実線) の実行時間 (秒) の増加の様子. (a) 基本法 ($N = 256$). 横軸はフレーム数 M . (b) 双対法 ($M = 256$). 横軸は特徴点数 N .

6. 実験 1

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¹⁴ に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画像列である.

図には 265 個の特徴点, 6 フレームの場合を表示しているが, 特徴点とフレーム数を変化させ (第 1 画像と最終画像は図 1 と同じ), 基本法と双対法による射影復元を行い, 実行時間 (秒) を計測した. CPU には Core2Duo 1.8GHz, 主メモリ 1GB, OS には Linux を用い, 再投影誤差が 0.1 画素になるまで反復した.

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. 図 2(a) は基本法のフレーム数 M に対する実行時間, 図 2(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.

点線は原形であり, 破線は前報 [9] で最も高速であることが確認された部分空間の当てはめにべき乗法, 固有値計算にべき乗法の加速を用いる方法 (以下「加速法」と呼ぶ), 実線は本論文の計算を効率化する方法 (以下「効率化法」と呼ぶ) である. 部分空間の収

¹⁴ 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.

束判定定数は加速法, 効率化法共に $e = 1$ とし, 加速法のべき乗法の収束判定定数は $d = 5$ とした (文献 [9] と同一条件).

図 2 からわかるように, 原形の実行時間は基本法ではフレーム数 M が, 双対法では特徴点数 N が増えると急速に増大する. それに対して加速法は前報 [9] に示したように著しい高速化を達成している. しかし, 本論文の効率化ではさらに高速化されている. そして, 基本法ではフレーム数 M に, 双対法では特徴点数 N に関してほぼ線形に増大している.

7. 実験 2

図 3(a) は円筒面上に 231 個の特徴点を取り, 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 3(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-Lucas-Tomasi の方法 [11] 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 1 にそれぞれの画像列に対する原形と加速法と効率化法の実行時間 (秒) を示す. 収束条件は再投影誤差が図 3(a) では 0.1 画素以下としたが, 図 3(b) では 2.1 画素以下とした. これは Kanade-Lucas-Tomasi の方法 [11] によって追跡した特徴点には 2 画素程度のふらつきがあり, いくら反復しても再投影誤差がこれ以上減少しなかったからである [10]. これは実画像上の特徴点追跡の精度の限界とみなせる.

図 3(a) の画像列はフレーム数が少ないので ($M = 11$), 原形を用いると基本法のほうが効率的である. 双対法は各反復ステップの計算がほぼ N^3 に比例する計算量が必要であり, この場合は特徴点数が多いので ($N = 231$), 実行速度が遅い. これを加速すると, 各反復ステップの計算が重い双対法の改善が著

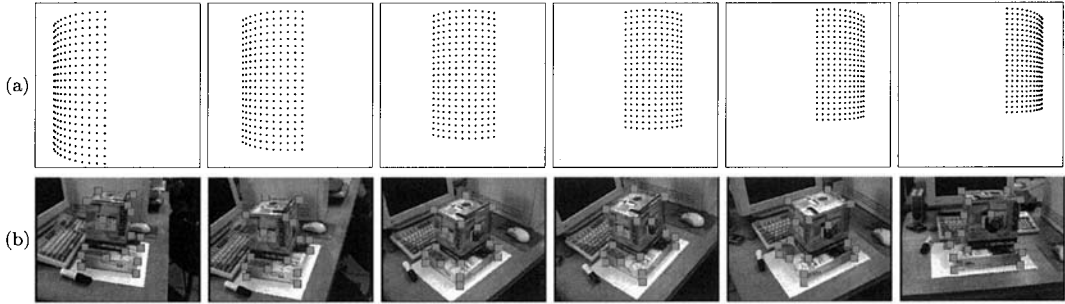


図 3: (a) シミュレーション画像系列. フレーム数 11, 特徴点数 231 個. (b) 実ビデオ画像. フレーム数 200, 特徴点数 16 個. いずれも抜き出した 6 フレームを示す.

表 1: 図 3 の画像列に対する実行時間 (秒). 収束条件は再投影誤差が図 3(a) では 0.1 画素以下, 図 3(b) では 2.1 画素以下.

	図 3(a)		図 3(b)	
	基本法	双対法	基本法	双対法
原形	1.648	4.096	113.415	0.076
加速法	1.024	0.060	0.972	0.076
効率化法	1.256	0.040	0.804	0.060

しい. 一方, 本論文の効率化法では加速法にはやや及ばないが, それに近い高速化が得られ, 双対法では加速法よりも高速化されている.

一方, 図 3(b) の画像列では特徴点数が少ないので ($N = 16$), 原形では双対法が圧倒的に高速である. 基本法は各反復ステップの計算がほぼ M^3 に比例する計算量が必要であり, この場合は特徴点数が多いので ($M = 200$), 非常に多くの計算時間を要する. これを加速すると, 基本法の改善が著しいが, 双対法には及ばない. 一方, 本論文の効率化法では基本法, 双対法ともにさらなる高速化が達成されている.

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効率化する新しい手法を示した.

前報 [9] では反復計算を加速する工夫を行って著しい高速化を達成したが, 本論文では計算量を注意深く解析し, 計算過程を変形することによって演算回数を削減した. そして, シミュレーションや実ビデオ画像によって, 前報 [9] の方法 (加速法) とほぼ同等で多くの場合にそれ以上の高速化が実現されることを示した.

本論文の方法は実行を加速するというより, 計算をより少ない変数の演算に置き換えているので, 計

算時間の短縮とともに, 実行時のメモリ領域の削減にもなっている. これは応用によっては非常に大きなメリットになる.

謝辞: 本研究は三菱プレジジョン (株) との共同研究の成果の一部である.

参考文献

- [1] R. I. Hartley, In defense of the eight-point algorithm, *IEEE Trans. Patt. Anal. Mach. Intell.*, **19-6** (1997-6), 580–593.
- [2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, U.K., 2000.
- [3] 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.
- [4] 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.
- [5] 金谷健一, 「これなら分かる応用数学教室—最小二乗法からウェーブレットまで—」, 共立出版, 2003.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 森 昭延, ハノ・アッカーマン, 金谷健一, 高速射影復元: 徹底的な効率化を目指して, 情報処理学会研究報告, 2007-CVIM-157-15 (2007-1), pp. 109–116.
- [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] 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/klf/>.
- [12] 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.