

# 因子分解法の完全レシピ

金谷 健一<sup>†</sup> 菅谷 保之<sup>†</sup>

<sup>†</sup> 岡山大学工学部情報工学科

あらまし ビデオ画像上を追跡した特徴点からその3次元形状を復元する Tomasi・Kanade の「因子分解法」は“特異値分解によって行列の因子分解を行なう方法”と理解されているが、本論文ではその原理は投影のアフィンカメラモデルによる近似であり、特異値分解は数値計算の手段に過ぎないことを示す。まず問題の幾何学的な構造を示し、次に「平行投影」、「弱透視投影」、「疑似透視投影」を仮定する3次元復元の完全なプログラミング法を記述する。  
キーワード 因子分解法, アフィンカメラモデル, 平行投影, 弱透視投影, 疑似透視投影

## Complete Recipe for Factorization

Kenichi KANATANI<sup>†</sup> and Yasuyuki SUGAYA<sup>†</sup>

<sup>†</sup> Department of Information Technology, Okayama University, Okayama, 700-8530 Japan

**Abstract** The Tomasi-Kanade *factorization* for reconstructing the 3-D shape of the feature points tracked through a video stream is widely regarded as based on factorization of a matrix by SVD (singular value decomposition). This paper points out that the core principle is the approximation of the imaging by an affine camera model and that SVD is merely one means of numerical computation. We first describe the geometric structure of the problem and then give a complete programming scheme for 3-D reconstruction.

**Key words** factorization method, affine camera model, orthographic projection, weak perspective projection, paraperspective projection.

### 1. まえがき

ビデオ画像上で追跡した特徴点からシーンの3次元形状を復元する手法として Tomasi・Kanade [9] の因子分解法がよく知られている。これはカメラの撮像をアフィン変換と仮定してカメラ運動とシーンの3次元形状を計算するものであり、特徴点の軌跡データから線形計算のみで容易に実行できる。復元結果に高精度を要求しなければ十分実用的であり、また透視変換に基づく厳密な復元の反復計算 [1] の初期値としても用いられる。

しかし、発表以来10年以上が経過したにもかかわらず、因子分解法は特異値分解によって行列の因子分解を行なう方法であるという誤解が未だに流布している。この方法の“原理”は投影のアフィンカメラモデルによる近似であり、特異値分解による因子分解は最小二乗解の数値計算の手段に過ぎない。

このことは以前からさまざまな形で指摘されていたが [3], [6], [10], 本論文ではそれを実証するために因子分解法の具体的なプログラミングの詳細を述べる。

もう一つの理由は、ほとんどの文献に“特異値分解に

よって因子分解する”以上のこと、特にその後の核心となる計量条件の処理と運動と形状の最適化があまり書かれていないことである。本論文の目的は次の2点である。

- (1) 因子分解法においては特異値分解による因子分解は本質的な意味を持たないことを示す。
- (2) 文献に書かれていない、具体的なカメラモデルに応じた3次元復元プログラミングの詳細を示す。

### 2. アフィンカメラモデル

$N$  個の特徴点  $\{p_\alpha\}$  を  $M$  枚の画像に渡って追跡し、第  $\kappa$  画像における  $\alpha$  番目の特徴点  $p_\alpha$  の画像座標を  $(x_{\kappa\alpha}, y_{\kappa\alpha})$ ,  $\kappa = 1, \dots, M, \alpha = 1, \dots, N$  とする。そしてその運動履歴を次の  $2M$  次元ベクトルで表し、軌跡ベクトルと呼ぶ。

$$p_\alpha = \begin{pmatrix} x_{1\alpha} & y_{1\alpha} & x_{2\alpha} & y_{2\alpha} & \cdots & x_{M\alpha} & y_{M\alpha} \end{pmatrix}^\top \quad (1)$$

これによって各特徴点の軌跡を  $2M$  次元空間の1点と同一視できる。便宜上、画像番号  $\kappa$  を「時刻  $\kappa$ 」と呼ぶ。

以下、カメラの光軸を  $Z$  軸とするカメラ  $XYZ$  座標系をとり、これに相対的にシーンが運動すると解釈する(静止したシーンに相対的にカメラが運動すると考えても同

等である)。

シーン中に3次元座標系を任意に固定し、特徴点  $p_\alpha$  のそのシーン座標系に関する3次元座標を  $(a_\alpha, b_\alpha, c_\alpha)$  とする。時刻  $\kappa$  でのシーン座標系の原点と各座標軸の基底ベクトルをカメラ座標系で表したものをそれぞれ  $t_\kappa, \{i_\kappa, j_\kappa, k_\kappa\}$  とすると、特徴点  $p_\alpha$  の時刻  $\kappa$  における3次元位置  $r_{\kappa\alpha}$  はカメラ座標系では次式となる。

$$r_{\kappa\alpha} = t_\kappa + a_\alpha i_\kappa + b_\alpha j_\kappa + c_\alpha k_\kappa \quad (2)$$

平行投影や弱透視投影や疑似透視投影を抽象化したアフィンカメラ [2], [8] は、3次元点  $r_{\kappa\alpha}$  が次のように画像上に投影されると仮定するものである [7]。

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = P_\kappa r_{\kappa\alpha} + q_\kappa \quad (3)$$

ここに  $P_\kappa, q_\kappa$  はそれぞれ時刻  $\kappa$  でのカメラの位置や内部パラメータによって定まる  $2 \times 3$  行列および2次元ベクトルである。式 (2) を代入すると、式 (3) は次のように書ける。

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \tilde{m}_{0\kappa} + a_\alpha \tilde{m}_{1\kappa} + b_\alpha \tilde{m}_{2\kappa} + c_\alpha \tilde{m}_{3\kappa} \quad (4)$$

$\tilde{m}_{0\kappa}, \tilde{m}_{1\kappa}, \tilde{m}_{2\kappa}, \tilde{m}_{3\kappa}$  は時刻  $\kappa$  でのカメラの位置や内部パラメータで決まる2次元ベクトルである。これを時刻  $\kappa = 1, \dots, M$  に渡って縦に並べると、式 (1) の軌跡ベクトル  $p_\alpha$  は次のように書ける。

$$p_\alpha = m_0 + a_\alpha m_1 + b_\alpha m_2 + c_\alpha m_3 \quad (5)$$

$m_i, i = 0, 1, 2, 3$  は  $\tilde{m}_{i\kappa}$  を時刻  $\kappa = 1, \dots, M$  に渡って縦に並べた  $2M$  次元ベクトルである。

式 (5) はすべての軌跡ベクトル  $p_\alpha$  が  $2M$  次元空間中で  $m_0$  を通り、 $\{m_1, m_2, m_3\}$  の張る「3次元アフィン空間」に含まれることを表している。これをアフィン空間拘束条件と呼ぶ [7]。

### 3. 処理の流れ

以上より、観測データ  $\{p_\alpha\}$  から3次元復元する計算は次の4ステップから成る。

(1) 軌跡ベクトル  $\{p_\alpha\}$  に3次元アフィン空間を当てはめる。

(2) 当てはめた3次元アフィン空間からベクトル  $m_0, m_1, m_2, m_3$  を計算する。

(3) 得られた  $m_0, m_1, m_2, m_3$  から各時刻のシーン座標系の位置  $t_\kappa$  と向き  $\{i_\kappa, j_\kappa, k_\kappa\}$  を計算する。

(4) 各特徴点の3次元座標  $(a_\alpha, b_\alpha, c_\alpha)$  を計算し、式 (2) により3次元位置  $r_{\kappa\alpha}$  を計算する。

「特異値分解による因子分解」と呼ばれるのは第1ステップである。これは3次元アフィン空間の最小二乗法による当てはめである。

核心は第2ステップである。これは計量条件と呼ばれ、どういうカメラモデルを仮定するかに依存する。

第3ステップは最適化計算であるが  $\{i_\kappa, j_\kappa, k_\kappa\}$  が正規直交系であることから、これらを列として並べた行列  $R = (i_\kappa, j_\kappa, k_\kappa)$  が回転行列であるという非線形拘束条件が介入する。このような回転の最適な当てはめ問題は特異値分解によって解決する [4]。

第4ステップは最も単純であり、

$$J = \sum_{\alpha=1}^N \|p_\alpha - m_0 - a_\alpha m_1 - b_\alpha m_2 - c_\alpha m_3\|^2 \quad (6)$$

を最小にすればよい。これは各  $(a_\alpha, b_\alpha, c_\alpha)$  の2次式であるから、線形方程式 (正規方程式) を解いて解が定まる。

### 4. アフィン空間の当てはめ

シーン座標系の原点は任意であるから、 $N$  個の特徴点の重心にとることができる。このとき  $\sum_{\alpha=1}^N a_\alpha = \sum_{\alpha=1}^N b_\alpha = \sum_{\alpha=1}^N c_\alpha = 0$  であるから、式 (5) より  $m_0$  は  $\{p_\alpha\}$  の重心

$$p_C = \frac{1}{N} \sum_{\alpha=1}^N p_\alpha \quad (7)$$

となる。よく知られているように、これを通る最適な3次元アフィン空間は、各  $p_\alpha$  の重心  $p_C$  からの差を

$$p'_\alpha = p_\alpha - p_C \quad (8)$$

と置くと、モーメント行列

$$C = \sum_{\alpha=1}^N p'_\alpha p'^\top_\alpha \quad (9)$$

の大きい3個の固有値  $\lambda_1, \lambda_2, \lambda_3$  に対応する単位固有ベクトル  $u_1, u_2, u_3$  の張る空間である [7]。したがって  $m_1, m_2, m_3$  は次のように  $u_1, u_2, u_3$  の線形結合で表せる。

$$m_j = \sum_{i=1}^3 A_{ij} u_i \quad (10)$$

しかし3次元復元に必要なのは行列  $A = (A_{ij})$  そのものではなく、計量行列

$$T = AA^\top \quad (11)$$

である。これを定める条件がカメラモデルに依存する計量条件である。

## 5. 因子分解法のいわれ

以上のように因子分解法の“原理”は「特異値分解による因子分解」ではないが、もちろん特異値分解を用いてもよい。実際、 $p'_1, p'_2, \dots, p'_N$  を列とする観測行列を

$$W = \begin{pmatrix} p'_1 & p'_2 & \cdots & p'_N \end{pmatrix} \quad (12)$$

と置き、これを次のように特異値分解してもよい。

$$W = U_N \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_N) V_N^\top \quad (13)$$

ここに  $U_N, V_N$  はそれぞれ正規直交系を列とする  $2M \times N, N \times N$  行列であり、 $\sigma_1, \sigma_2, \dots, \sigma_N (\geq 0)$  は特異値である ( $\text{diag}(\dots)$  は  $\dots$  を対角要素とする対角行列)。このとき  $U_N$  の第  $i$  列は式 (9) モーメント行列  $C$  の固有値  $\lambda_i = \sigma_i^2$  に対応する単位固有ベクトルになっている [5]。したがって式 (9) の  $C$  の固有値問題を解くことと観測行列  $W$  の特異値分解は等価である。

$2M \times N$  行列  $W$  の特異値分解の効率は  $\min(2M, N)$  によって定まるので、特徴点数  $N$  に比べてフレーム数 (の2倍)  $2M$  が非常に多いときは特異値分解のほうが計算量が有利になる。

式 (5) より  $p'_\alpha = a_\alpha m_1 + b_\alpha m_2 + c_\alpha m_3$  であるから、これを式 (12) に代入すると次の形に書ける。

$$W = \begin{pmatrix} m_1 & m_2 & m_3 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_N \\ b_1 & b_2 & \cdots & b_N \\ c_1 & c_2 & \cdots & c_N \end{pmatrix} = MS \quad (14)$$

ただし  $M$  は  $m_i$  を第  $i$  列とする  $2M \times 3$  行列であり、 $S$  は  $(a_\alpha, b_\alpha, c_\alpha)^\top$  を第  $\alpha$  列とする  $3 \times N$  行列である。この記法を用いると、式 (10) は

$$M = UA \quad (15)$$

と書ける。ただし  $U$  は  $U_N$  の最初の3列ベクトルから成る  $2M \times 3$  行列である。

Tomasi・Kanade [9] は式 (5) のアフィン空間拘束条件を式 (14) のように書き、ランク3拘束条件と呼んだ。そして  $m_1, m_2, m_3$  を計算するのに式 (13) の特異値分解を利用して式 (15) と置いて  $A$  を定めたことから、「因子分解法は特異値分解によって行列の因子分解を行なう方法である」という解釈が生まれた。しかしこれは幾何学的な構造とそれを計算するための数値解析手段との混同であり、特異値分解はアフィン空間の当てはめの数値計算の一手段に過ぎない。

## 6. 3次元形状復元アルゴリズム

まず因子分解法による3次元形状復元アルゴリズムの

流れを記述する (導出は省略)。その後で個々のカメラモデルに依存する部分の計算法を述べる。

注意すべきことは、解が二通り得られ、それぞれの復元形状は互いの鏡像になっていることである。アフィンカメラのもとでは、両者を区別する判定法は存在しない (どちらを撮影しても同じ画像が得られる)。

入力:

- $2M$  次元軌跡ベクトル  $\{p_\alpha\}, \alpha = 1, \dots, N$  ( $M$  は画像フレーム数,  $N$  は特徴点数)。
- シーンの平均奥行  $Z_c$  と各フレームの焦点距離  $\{f_\kappa\}$  (未知なら任意に設定してよい)。

出力:

- 各フレームの特徴点の重心の並進ベクトル  $\{t_\kappa\}$ 。
- 第1フレームに関する各特徴点の3次元位置  $\{\hat{r}_\alpha\}, \{\hat{r}'_\alpha\}$  (両者は互いの鏡像)。
- 対応する回転行列  $\{R_\kappa\}, \{R'_\kappa\}$ 。

手順 (\* はカメラモデルに依存することを表す):

【アフィン空間の当てはめ】

(1) 軌跡ベクトル  $\{p_\alpha\}$  の重心  $p_C$  を式 (7) によって計算する。

(2)  $p_C$  の第  $(2(\kappa-1)+1)$ , 第  $(2(\kappa-1)+2)$  要素をそれぞれ  $\hat{t}_{x\kappa}, \hat{t}_{y\kappa}$  とする。

(3) 軌跡ベクトル  $\{p_\alpha\}$  に3次元アフィン空間を当てはめ、その基底ベクトルを  $u_1, u_2, u_3$  とする。

(4) それらを列とする  $2M \times 3$  行列を  $U$  とし、 $U^\top$  の第  $(2(\kappa-1)+a)$  列を  $u_{\kappa(a)}^\dagger$  とする ( $\kappa = 1, \dots, M, a = 1, 2$ )。

【計量行列の計算】

(5)\*  $3 \times 3$  計量行列  $T$  を計算する。

(6)  $T$  の固有値  $\lambda_1, \lambda_2, \lambda_3$  と対応する固有ベクトルの正規直交系  $\{v_1, v_2, v_3\}$  を計算する。

#  $T$  は正値対称行列であり、その固有値  $\lambda_1, \lambda_2, \lambda_3$  はすべて正。

【並進ベクトルの計算】

(7)\* 各時刻  $\kappa$  の並進ベクトル  $t_\kappa = (t_{x\kappa}, t_{y\kappa}, t_{z\kappa})^\top$  を計算する。

【回転行列の計算】

(8) 次の  $2M$  次元ベクトルを計算する。

$$m_i = \sqrt{\lambda_i} \begin{pmatrix} (u_{1(1)}^\dagger, v_i) \\ (u_{1(2)}^\dagger, v_i) \\ (u_{2(1)}^\dagger, v_i) \\ \vdots \\ (u_{M(2)}^\dagger, v_i) \end{pmatrix}, \quad i = 1, 2, 3 \quad (16)$$

(9)  $m_1, m_2, m_3$  を列とする  $2M \times 3$  行列を  $M$  とし、 $M^\top$  の第  $(2(\kappa-1)+a)$  列を  $m_{\kappa(a)}^\dagger$  とする ( $\kappa = 1, \dots, M, a = 1, 2$ )。

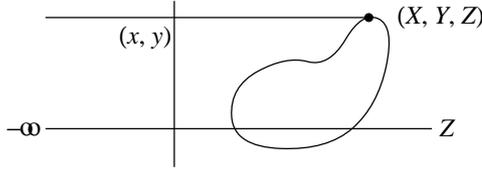


図1 平行投影 .

(10)\* 各時刻  $\kappa$  の回転行列  $R_\kappa$  を計算する .

【形状の最適化】

(11)\* 行列  $M$  を次のように再計算する .

$$M = \sum_{\kappa=1}^M \Pi_\kappa^\top R_\kappa \quad (17)$$

#  $\Pi_\kappa = (\Pi_{\kappa(ij)})$  はカメラモデルに依存する  $3 \times 2M$  行列 .

(12) 3次元形状ベクトル  $\{s_\alpha\}$  を次のように計算する .

$$s_\alpha = (M^\top M)^{-1} M^\top (p_\alpha - p_C) \quad (18)$$

【鏡像解の計算】

(13)\*  $\{s'_\alpha\}$  と  $\{R'_\kappa\}$  を次のように計算する .

$$s'_\alpha = -s_\alpha, \quad R'_\kappa = \Omega_\kappa R_\kappa \quad (19)$$

#  $\Omega_\kappa$  はカメラモデルに依存する回転行列 .

【3次元位置の計算】

(14) 2組の解  $\{\hat{r}_\alpha\}$ ,  $\{\hat{r}'_\alpha\}$  を次のように計算する .

$$\hat{r}_\alpha = \frac{Z_c}{t_{z1}} (R_1 s_\alpha + t_1), \quad \hat{r}'_\alpha = \frac{Z_c}{t_{z1}} (R'_1 s'_\alpha + t_1) \quad (20)$$

## 7. 平行投影カメラモデル

前節の\* のついたステップは仮定するカメラモデルに依存する . 平行投影 とは空間の点  $(X, Y, Z)$  が

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \quad (21)$$

の関係によって画像上の点  $(x, y)$  に投影されるとみなすものである (図1) . 平行投影を仮定すると , 前節の\* のステップは次のようになる (導出は省略) .

【計量行列の計算】

(1) 次のように  $3 \times 3 \times 3 \times 3$  テンソル  $B = (B_{ijkl})$  を定義する .

$$\begin{aligned} B_{ijkl} = & \sum_{\kappa=1}^M \left[ (\mathbf{u}_{\kappa(1)}^\dagger)_i (\mathbf{u}_{\kappa(1)}^\dagger)_j (\mathbf{u}_{\kappa(1)}^\dagger)_k (\mathbf{u}_{\kappa(1)}^\dagger)_l \right. \\ & + (\mathbf{u}_{\kappa(2)}^\dagger)_i (\mathbf{u}_{\kappa(2)}^\dagger)_j (\mathbf{u}_{\kappa(2)}^\dagger)_k (\mathbf{u}_{\kappa(2)}^\dagger)_l \\ & + \frac{1}{4} \left( (\mathbf{u}_{\kappa(1)}^\dagger)_i (\mathbf{u}_{\kappa(2)}^\dagger)_j + (\mathbf{u}_{\kappa(2)}^\dagger)_i (\mathbf{u}_{\kappa(1)}^\dagger)_j \right) \\ & \left. \left( (\mathbf{u}_{\kappa(1)}^\dagger)_k (\mathbf{u}_{\kappa(2)}^\dagger)_l + (\mathbf{u}_{\kappa(2)}^\dagger)_k (\mathbf{u}_{\kappa(1)}^\dagger)_l \right) \right] \quad (22) \end{aligned}$$

#  $(\mathbf{u}_{\kappa(a)}^\dagger)_i$  は3次元ベクトル  $\mathbf{u}_{\kappa(a)}^\dagger$  の第  $i$  成分 .

(2) 次の  $6 \times 6$  対称行列を定義する .

$$B = \begin{pmatrix} B_{1111} & B_{1122} & B_{1133} \\ B_{2211} & B_{2222} & B_{2233} \\ B_{3311} & B_{3322} & B_{3333} \\ \sqrt{2}B_{2311} & \sqrt{2}B_{2322} & \sqrt{2}B_{2333} \\ \sqrt{2}B_{3111} & \sqrt{2}B_{3122} & \sqrt{2}B_{3133} \\ \sqrt{2}B_{1211} & \sqrt{2}B_{1222} & \sqrt{2}B_{1233} \\ \sqrt{2}B_{1123} & \sqrt{2}B_{1131} & \sqrt{2}B_{1112} \\ \sqrt{2}B_{2223} & \sqrt{2}B_{2231} & \sqrt{2}B_{2212} \\ \sqrt{2}B_{3323} & \sqrt{2}B_{3331} & \sqrt{2}B_{3312} \\ 2B_{2323} & 2B_{2331} & 2B_{2312} \\ 2B_{3123} & 2B_{3131} & 2B_{3112} \\ 2B_{1223} & 2B_{1231} & 2B_{1212} \end{pmatrix} \quad (23)$$

(3) 次の6次元ベクトル  $c$  を定義する .

$$c = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}^\top \quad (24)$$

(4) 次の6次元連立1次方程式を解いて , 6次元ベクトル  $\tau = (\tau_i)$  を求める .

$$B\tau = c \quad (25)$$

(5) 計量行列  $T$  を次のように計算する .

$$T = \begin{pmatrix} \tau_1 & \tau_6/\sqrt{2} & \tau_5/\sqrt{2} \\ \tau_6/\sqrt{2} & \tau_2 & \tau_4/\sqrt{2} \\ \tau_5/\sqrt{2} & \tau_4/\sqrt{2} & \tau_3 \end{pmatrix} \quad (26)$$

【並進ベクトルの計算】

(1)  $t_{z\kappa} = Z_c$  と置く ( $\kappa = 1, \dots, 2M$ ) .

(2)  $t_{x\kappa} = \tilde{t}_{x\kappa}$ ,  $t_{y\kappa} = \tilde{t}_{y\kappa}$  と置く ( $\kappa = 1, \dots, 2M$ ) .

【回転行列の計算】

(1) 各  $\kappa$  に対して次の特異値分解を計算する .

$$\begin{pmatrix} m_{\kappa(1)}^\dagger & m_{\kappa(2)}^\dagger & \mathbf{0} \end{pmatrix} = V_\kappa \Lambda_\kappa U_\kappa^\top \quad (27)$$

(2) 回転行列  $\{R_\kappa\}$  を次のように計算する .

$$R_\kappa = U_\kappa \text{diag}(1, 1, \det(V_\kappa U_\kappa^\top)) V_\kappa^\top \quad (28)$$

【形状復元の行列と鏡像解の回転行列】

$$\Pi_\kappa = \begin{pmatrix} 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}^{(2\kappa-1)(2\kappa)}, \quad (29)$$

$$\Omega_\kappa = \text{diag}(-1, -1, 1) \quad (30)$$

## 8. 弱透視投影カメラモデル

弱透視変換 [2], [8] とは空間の点  $(X, Y, Z)$  が次のように



$$+(\mathbf{u}_{\kappa(1)}^\dagger)_i(\mathbf{u}_{\kappa(2)}^\dagger)_j(\mathbf{u}_{\kappa(1)}^\dagger)_k(\mathbf{u}_{\kappa(2)}^\dagger)_l \quad (44)$$

$$+(\mathbf{u}_{\kappa(1)}^\dagger)_i(\mathbf{u}_{\kappa(2)}^\dagger)_j(\mathbf{u}_{\kappa(2)}^\dagger)_k(\mathbf{u}_{\kappa(1)}^\dagger)_l$$

$$+(\mathbf{u}_{\kappa(2)}^\dagger)_i(\mathbf{u}_{\kappa(1)}^\dagger)_j(\mathbf{u}_{\kappa(1)}^\dagger)_k(\mathbf{u}_{\kappa(2)}^\dagger)_l$$

$$+(\mathbf{u}_{\kappa(2)}^\dagger)_i(\mathbf{u}_{\kappa(1)}^\dagger)_j(\mathbf{u}_{\kappa(2)}^\dagger)_k(\mathbf{u}_{\kappa(1)}^\dagger)_l$$

$$-\alpha_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(1)}^\dagger)_i (\mathbf{u}_{\kappa(1)}^\dagger)_j (\mathbf{u}_{\kappa(1)}^\dagger)_k (\mathbf{u}_{\kappa(2)}^\dagger)_l$$

$$-\alpha_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(1)}^\dagger)_i (\mathbf{u}_{\kappa(1)}^\dagger)_j (\mathbf{u}_{\kappa(2)}^\dagger)_k (\mathbf{u}_{\kappa(1)}^\dagger)_l$$

$$-\alpha_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(1)}^\dagger)_i (\mathbf{u}_{\kappa(2)}^\dagger)_j (\mathbf{u}_{\kappa(1)}^\dagger)_k (\mathbf{u}_{\kappa(1)}^\dagger)_l$$

$$-\alpha_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(2)}^\dagger)_i (\mathbf{u}_{\kappa(1)}^\dagger)_j (\mathbf{u}_{\kappa(1)}^\dagger)_k (\mathbf{u}_{\kappa(1)}^\dagger)_l$$

$$-\beta_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(2)}^\dagger)_i (\mathbf{u}_{\kappa(2)}^\dagger)_j (\mathbf{u}_{\kappa(1)}^\dagger)_k (\mathbf{u}_{\kappa(2)}^\dagger)_l$$

$$-\beta_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(2)}^\dagger)_i (\mathbf{u}_{\kappa(2)}^\dagger)_j (\mathbf{u}_{\kappa(2)}^\dagger)_k (\mathbf{u}_{\kappa(1)}^\dagger)_l$$

$$-\beta_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(1)}^\dagger)_i (\mathbf{u}_{\kappa(2)}^\dagger)_j (\mathbf{u}_{\kappa(2)}^\dagger)_k (\mathbf{u}_{\kappa(2)}^\dagger)_l$$

$$-\beta_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(2)}^\dagger)_i (\mathbf{u}_{\kappa(1)}^\dagger)_j (\mathbf{u}_{\kappa(2)}^\dagger)_k (\mathbf{u}_{\kappa(2)}^\dagger)_l$$

$$+(\gamma_\kappa^2 - 1) \alpha_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(1)}^\dagger)_i (\mathbf{u}_{\kappa(1)}^\dagger)_j (\mathbf{u}_{\kappa(2)}^\dagger)_k (\mathbf{u}_{\kappa(2)}^\dagger)_l$$

$$+(\gamma_\kappa^2 - 1) \alpha_\kappa \gamma_\kappa (\mathbf{u}_{\kappa(2)}^\dagger)_i (\mathbf{u}_{\kappa(2)}^\dagger)_j (\mathbf{u}_{\kappa(1)}^\dagger)_k (\mathbf{u}_{\kappa(1)}^\dagger)_l \quad (40)$$

(3) 式(23)の $6 \times 6$ 対称行列 $B$ を計算する.

(4)  $B$ の最小固有値に対する6次元単位固有ベクトル $\tau = (\tau_i)$ を計算する.

(5) 計量行列 $T$ を式(26)によって計算する.

(6)  $\det T < 0$ であれば $T \leftarrow -T$ と符号を換える.

【並進ベクトルの計算】

(1)  $t_{z\kappa}$ を次のように計算する.

$$t_{z\kappa} = f_\kappa \sqrt{\frac{2}{\alpha_\kappa (\mathbf{u}_{\kappa(1)}^\dagger, \mathbf{T} \mathbf{u}_{\kappa(1)}^\dagger) + \beta_\kappa (\mathbf{u}_{\kappa(2)}^\dagger, \mathbf{T} \mathbf{u}_{\kappa(2)}^\dagger)}} \quad (41)$$

(2)  $t_{x\kappa}, t_{y\kappa}$ を式(34)によって計算する.

【回転行列の計算】

(1) 各 $\kappa$ に対して次の $\{\mathbf{r}_{\kappa(1)}^\dagger, \mathbf{r}_{\kappa(2)}^\dagger, \mathbf{r}_{\kappa(3)}^\dagger\}$ を計算する.

$$\mathbf{r}_{\kappa(3)}^\dagger = \frac{t_{z\kappa}}{f_\kappa \left(1 + (t_{x\kappa}/t_{z\kappa})^2 + (t_{y\kappa}/t_{z\kappa})^2\right)}$$

$$\left( \frac{t_{z\kappa}}{f_\kappa} \mathbf{m}_{\kappa(1)}^\dagger \times \mathbf{m}_{\kappa(2)}^\dagger - \frac{t_{x\kappa}}{t_{z\kappa}} \mathbf{m}_{\kappa(1)}^\dagger - \frac{t_{y\kappa}}{t_{z\kappa}} \mathbf{m}_{\kappa(2)}^\dagger \right),$$

$$\mathbf{r}_{\kappa(1)}^\dagger = \frac{t_{z\kappa}}{f_\kappa} \mathbf{m}_{\kappa(1)}^\dagger + \frac{t_{x\kappa}}{t_{z\kappa}} \mathbf{r}_{\kappa(3)}^\dagger,$$

$$\mathbf{r}_{\kappa(2)}^\dagger = \frac{t_{z\kappa}}{f_\kappa} \mathbf{m}_{\kappa(2)}^\dagger + \frac{t_{y\kappa}}{t_{z\kappa}} \mathbf{r}_{\kappa(3)}^\dagger \quad (42)$$

(2) 次の特異値分解を計算する.

$$\begin{pmatrix} \mathbf{r}_{\kappa(1)}^\dagger & \mathbf{r}_{\kappa(2)}^\dagger & \mathbf{r}_{\kappa(3)}^\dagger \end{pmatrix} = \mathbf{V}_\kappa \mathbf{\Lambda}_\kappa \mathbf{U}_\kappa^\top \quad (43)$$

(3) 回転行列 $\{\mathbf{R}_\kappa\}$ を式(28)によって計算する.

【形状復元の行列と鏡像解の回転行列】

$$\mathbf{\Pi}_\kappa = \frac{f_\kappa}{t_{z\kappa}} \begin{pmatrix} & (2\kappa-1) & (2\kappa) & & \\ 0 \cdots 0 & 1 & 0 & 0 \cdots 0 & \\ 0 \cdots 0 & 0 & 1 & 0 \cdots 0 & \\ 0 \cdots 0 & -t_{x\kappa}/t_{z\kappa} & -t_{y\kappa}/t_{z\kappa} & 0 \cdots 0 & \end{pmatrix},$$

$$\mathbf{\Omega}_\kappa = \frac{2t_\kappa \mathbf{t}_\kappa^\top}{\|\mathbf{t}_\kappa\|^2} - \mathbf{I} \quad (45)$$

## 10. まとめ

本論文では Tomasi・Kanade の因子分解法の原理が投影のアフィンカメラモデルによる近似であり, 特異値分解は数値計算の一手段に過ぎないことを示した. そして, 問題の幾何学的な構造を示し, 「平行投影」, 「弱透視投影」, 「疑似透視投影」を仮定する3次元復元の完全なプログラミング法を記述した. これにより, 誰でも直ちに因子分解法が実装できる.

本論文では導出の詳細を省略したが, すべてのステップの意味, 理由, 導出を付録として補足した本論文の完全版を Web 上に公開している<sup>(注1)</sup>.

謝辞: 本研究の一部は文部科学省科学研究費基盤研究C(2) (No. 15500113), テレコム先端技術センター, 栢森情報科学財団の助成によった.

## 文献

- [1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, U.K., 2000.
- [2] 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.
- [3] 金谷健一, 因子分解を用いない因子分解法: 平行投影から透視変換へ, 電子情報通信学会技術報告, PRMU98-26 (1998-6), 1-8.
- [4] K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, U.K., 1993.
- [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, 黒澤典義, 松永力, モデル選択によるランク推定と複数運動の分離, 情報処理学会研究報告, 2001-CVIM-126-3 (2001-3), 17-24.
- [7] 黒澤典義, 金谷健一, アフィン空間分離法による運動物体の分離, 情報処理学会研究報告, 2001-CVIM-125-3 (2001-3), 25-32.
- [8] 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.
- [9] 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.
- [10] 坪内貴之, 菅谷保之, 金谷健一, 3次元アフィン復元のための途切れた特徴点追跡の延長情報処理学会研究報告, 2003-CVIM-137-17 (2003-3), 133-140.

(注1): <http://www.suri.it.okayama-u.ac.jp/~kanatani/j/>