

White Paper : CV 演算

全周動画映像を数学的に解析することで、映像取得時のカメラの三次元位置と姿勢を求めることができる。これを CV 値と呼称する。本書では、その解析手法のアルゴリズムについて説明し、さらには実際にコンピュータで解くために行った計算処理についても記載する。

1 概要

CV 値を得るためには、映像のフレーム間関係として同一点の対応をとる特徴点トラッキングと、トラッキング結果から CV 値を計算する処理を、組み合わせて行う。この処理手順アルゴリズムと計算内容について、第 3 章で詳しく記載する。

また、処理時間短縮と精度向上のために、実装時にいくつかの工夫をした。これらの処理について、第 4 章で個別に記載する。

2 定義

2.1 全周画像

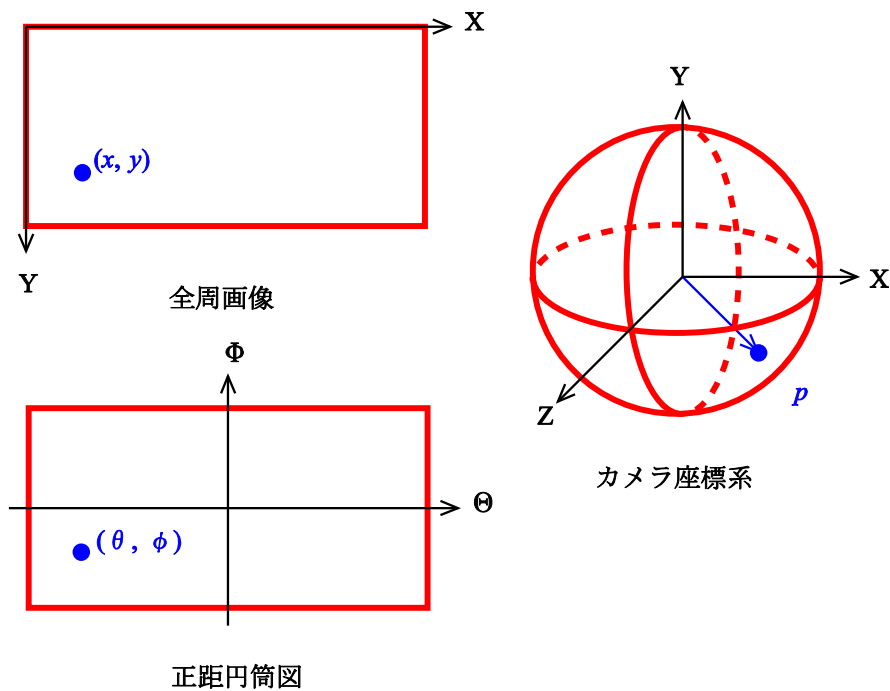


図 1: 全周画像の表現

全周画像は全方位を正距円筒図で表現した画像であり、画像上の各ピクセルの座標は、カメラ中心からの方向を意味している。画像の横幅を w 、縦幅を $h = w/2$ とするとき、画像上のピクセル (x, y) と角度 (θ, ϕ) 及び方向 p について、変換

$$\theta = 2 \left(\frac{x}{w} - \frac{1}{2} \right) \pi,$$

$$\phi = \left(\frac{1}{2} - \frac{y}{h} \right) \pi,$$

$$p = \begin{pmatrix} \cos \phi \sin \theta \\ \sin \phi \\ -\cos \phi \cos \theta \end{pmatrix}$$

により定義して、ピクセル位置と方向を同一視する（図1）。

2.2 CV 値

CV 値とは、あるフレームのカメラの三次元位置を三次元位置ベクトルで表した T と三次元向きを回転行列で表した R を合わせた 6 自由度の量のことである。特徴点の三次元座標を P とおき、これが全周画像上で \tilde{p} に投影されるとき、 ${}^tR(P-T)$ と \tilde{p} が同一方向を指すように定義する（図2）。

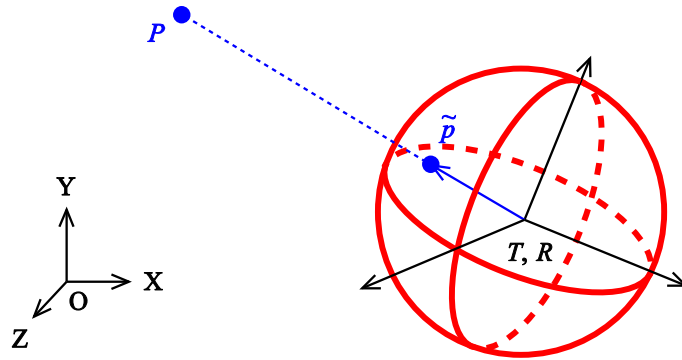


図 2: CV 値の定義

R は通常は 3×3 行列で表現するが、3 変数での表現が必要な計算では、

$$R = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{pmatrix}$$

のようにロール・ピッチ・ヨーの 3 軸回転成分で表現する。

2.3 投影誤差

特徴点の三次元座標 P 、CV 値 T, R および画像上の方向 p は、トラッキングの誤差、ピクセルの量子化誤差などにより誤差を持つ。そのため、計算上の投影位置 $\tilde{p} = \frac{{}^tR(P-T)}{\|P-T\|}$ と、トラッキング位置 p には差が生じる。そこで、本来は重なるはずの 2 つの方向ベクトル \tilde{p} と p の角度差

$$c = c(p, T, R, P) = \cos^{-1}(\tilde{p} \cdot p) = \cos^{-1} \frac{{}^tR(P-T) \cdot p}{\|P-T\|}$$

を投影誤差として定義する（図3）。

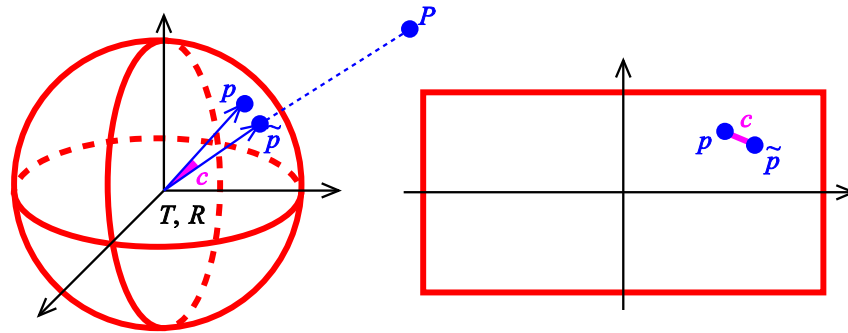


図 3: 投影誤差

以降の計算では、投影誤差の二乗 c^2 を

$$c^2 = c_x^2 + c_y^2$$

と 2 軸成分に分解して利用する。こうすることで、 c_x, c_y は平均をゼロとする誤差分布に従うと考えることができる。

3 計算アルゴリズム

CV 値を得るために、以下のステップを順次行う。

1. トラッキングを行う (3.1 節)
2. CV 計算を行う (3.2 節)
3. CV 値を利用してトラッキングを追加する (3.3 節)
4. 再度、CV 計算を行う (内容は 2. と同じ、3.2 節)

本章では、この各ステップについてアルゴリズムを説明する。

3.1 トラッキング

トラッキング処理は、画像比較により同一特徴点を異なるフレームで見つける処理である。画像の中から特徴点を選び出す処理と、選んだ特徴点を他フレームから探し出す処理の二段階で行う。

特徴点抽出については、FAST アルゴリズム*を使用した。

次に、同一特徴点の発見には、正規化相互相関関数を用いた。正規化相互相関関数は 2 画像の相関値を、1 から -1 までの数値で表す関数で、1 なら一致していることを表す。テンプレート画像を t 、比較画像を f とするとき、相関値 γ は

$$\gamma(u, v) = \frac{\sum_{x,y} (f(x, y) - \bar{f}_{u,v}) (t(x - u, y - v) - \bar{t})}{\sqrt{\sum_{x,y} (f(x, y) - \bar{f}_{u,v})^2 \sum_{x,y} (t(x - u, y - v) - \bar{t})^2}}$$

*<http://mi.eng.cam.ac.uk/~er258/work/fast.html>

として定義される。ここで、 $\bar{f}_{u,v}$ は画像 f の (u, v) を中心とするテンプレート画像サイズ領域の平均、 \bar{i} はテンプレート画像の平均である。

この相関値が最大となる点を同一特徴点とするが、閾値を設定して一定値以下の場合には対応点が見つからなかったと判定する。今回は閾値を 0.7 とした。

さらに、誤対応を少なくするため、はじめに粗い画像で大まかに相関値の高いところを見つけ、その後元の画像で該当する箇所の周辺に限って相関値の高いところを見つける処理を行っている。

3.2 CV 計算

p^m を m 番目の特徴点の n フレーム目におけるトラッキングとする。映像から得られたトラッキングデータ $\{p^m\}$ から、各フレームの CV 値 $\{T^n, R^n\}$ と各特徴点の三次元座標 $\{P^m\}$ を計算する。投影誤差の二乗和

$$C(\{T^n, R^n\}, \{P^m\}) = \sum_{m,n} (c^{mn})^2 = \sum_{m,n} ((c_x^{mn})^2 + (c_y^{mn})^2)$$

とおき、これを最小とするような $\{T^n, R^n\}, \{P^m\}$ が求めるものである。ここで、 c^{mn} は投影誤差 $c(p^m, T^n, R^n, P^m)$ である。

この最小二乗問題は、マルカート法により解くことができる。ただし、そのためには、ある程度解に近い初期値が必要となる。

ここで、初期値として使える精度の CV 値を求める方法を説明する。以下の手順で各フレームの CV 値を順次求める。

1. 最初の 2 フレームの CV 値を求める (3.2.1 節)
2. 特徴点の三次元座標を求める (3.2.2 節)
3. 追加で 1 フレームの CV 値を求める (3.2.3 節)
4. 以降は 2. と 3. を繰り返す

以下、順に求め方について記載したのち、マルカート法の適用 (3.2.4 節) について記載する。

3.2.1 2 フレーム間演算

2 フレーム間においては、一般的に知られているエピポラ方程式の解法と同様の手法で行う。1 つ目のフレームの特徴点 $\{p^m\}$ と 2 つ目のフレームの特徴点 $\{q^m\}$ が対応するとき、9 次元ベクトル b の各成分を

$$\begin{pmatrix} b_1^m & b_4^m & b_7^m \\ b_2^m & b_5^m & b_8^m \\ b_3^m & b_6^m & b_9^m \end{pmatrix} = p^{mT} q^m$$

とし、 9×9 行列 B を

$$B = \sum_m b^{mT} b^m$$

とする。 B の最小固有値に対応する固有ベクトルを e とおき、 3×3 行列 E を

$$E = \begin{pmatrix} e_1 & e_4 & e_7 \\ e_2 & e_5 & e_8 \\ e_3 & e_6 & e_9 \end{pmatrix}$$

とおく。すると、 $E^t E$ の最小固有値に対応する固有ベクトル T が CV 値の位置の候補となる。 $-T$ も固有ベクトルのため、どちらかが求めるものとなる。 $\sqrt{2}E/\|E\|$ をあらためて E とし、 $T/\|T\|$ をあらためて T とおき、方程式

$$E = \begin{pmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{pmatrix} R$$

から、回転行列 R を求める。 $-T$ の場合も同様に求めることにより、CV 値の向き候補も 2 つとなる。4 通りの組み合わせのうち、対応する特徴点が正しく画像に対応つくものを、CV 値として採用する。特徴点の計算については次の節で説明する。

3.2.2 三次元計測

CV 値が既知の場合に、トラッキングの結果から特徴点の三次元座標を求めることができる。既知の CV 値を $\{T^n, R^n\}$ 、トラッキングを $\{p^n\}$ とし、三次元座標 P を計算する。

求める三次元座標 P は、誤差が無ければ T^n を通り $v^n = R^n p^n$ を向く直線上に存在する。そこで、この直線と三次元座標の距離を誤差として見積もり、

$$C = \sum_n \|P - (T^n + s^n v^n)\|^2$$

とおき、 C を最小にする時の P が求める解である。ここで、 s^n はカメラ位置から P までの v^n を単位とする距離に相当する変数である。

連立方程式

$$\begin{cases} \frac{\partial C}{\partial s^n} = 0 \\ \frac{dC}{dP} = 0 \end{cases}$$

を解くと、

$$P = \left(\sum_n \left(I - \frac{v^n v^n^t}{\|v^n\|^2} \right) \right)^{-1} \left(\sum_n \left(I - \frac{v^n v^n^t}{\|v^n\|^2} \right) T^n \right)$$

が得られ、 P が計算できる。

より高精度に P を求めるために、今求めた値を初期値として、CV 計算と同様の投影誤差の和

$$C(P) = \sum_n (c^n)^2 = \sum_n ((c_x^n)^2 + (c_y^n)^2)$$

を最小にする P を求める。ここで c^n は投影誤差 $c(p^n, T^n, R^n, P)$ である。

3.2.3 1 フレーム CV 計算

複数の特徴点の三次元座標が既知で、あるフレームでそれらのトラッキング $\{p^m\}$ が与えられたとき、このフレームの CV 値 T, R を計算することができる。投影誤差の和

$$C(T, R) = \sum_m (c^m)^2 = \sum_m ((c_x^m)^2 + (c_y^m)^2)$$

を最小にする T, R が求める解である。ここで c^m は投影誤差 $c(p^m, T, R, P^m)$ である。

ここでも最小二乗問題を解くために初期値が必要となるが、直前のフレームの CV 値はマルカート法の初期値として使うのに十分であるため、その値を採用する。

3.2.4 マルカート法の適用

特徴点数を M 、フレーム数を N とする。初期値として CV 値 $\{T_0^n, R_0^n\}$ と三次元座標 $\{P_0^m\}$ が与えられたときに、投影誤差の和

$$C(\{T^n, R^n\}, \{P^m\}) = \sum_{m,n} (c^{mn})^2 = \sum_{m,n} ((c_x^{mn})^2 + (c_y^{mn})^2)$$

を最小にする解として、CV 値 $\{T^n, R^n\}$ と三次元座標 $\{P^m\}$ を求める。

ここで未知数の数を確認しておく。特徴点の座標に関するものが $3M$ 個、フレームの CV 値に関するものが $6N$ 個であるが、相対関係の解であることから少々自由度が残る。全体の位置および向きは自由であるため、最初のフレームの位置を原点、向きを単位行列と決め、6 自由度減る。さらに写真の原理より、被写体の大きさは決められないことから、1 自由度減る。この 1 自由度として、最後のフレームの位置までの長さを固定して、

$$T^N = \begin{pmatrix} r \cos \phi \sin \theta \\ r \sin \phi \\ r \cos \phi \cos \theta \end{pmatrix}$$

とおき、 r を定数として θ, ϕ の 2 変数として扱う。

$3M + 6N - 7$ 次元の解ベクトルを \mathbf{x} とし、これをマルカート法を用いて数値解を求める。マルカート法は、更新規則

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} - (H + \lambda I)^{-1} \frac{dC(\mathbf{x}_{\text{old}})}{d\mathbf{x}}$$

に従い、収束するまで解を更新するものである。投影誤差の和 $C(\mathbf{x}_{\text{new}})$ が $C(\mathbf{x}_{\text{old}})$ より小さくなれば解を更新し、 λ を $1/\nu$ 倍して反復する。投影誤差の和が小さくならなかったときは解を更新せずに、 λ を ν 倍してやり直す。ここで、 $\nu > 1$ は定数で、今回は ν を 10 とし、 λ の初期値は 10^{-4} とした。 H は

$$H = \begin{pmatrix} \frac{\partial^2 C}{\partial \mathbf{x}_1^2} & \frac{\partial^2 C}{\partial \mathbf{x}_1 \partial \mathbf{x}_2} & \cdots \\ \frac{\partial^2 C}{\partial \mathbf{x}_2 \partial \mathbf{x}_1} & \frac{\partial^2 C}{\partial \mathbf{x}_2^2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

として定義される二階微分の行列であるが、

$$\frac{\partial^2 (c^{mn})^2}{\partial p \partial q} \simeq 2 \left(\frac{\partial c_x^{mn}}{\partial p} \frac{\partial c_x^{mn}}{\partial q} + \frac{\partial c_y^{mn}}{\partial p} \frac{\partial c_y^{mn}}{\partial q} \right)$$

のように一階微分の積で近似できる。ここで、 C の最小化のかわりに $C/2$ の最小化を考えることで、係数の 2 を省略できる。

一階微分は計算式が煩雑になるので一般的な数値微分式

$$\frac{df(x_0)}{dx} \simeq \frac{f(x_0 + \delta) - f(x_0 - \delta)}{2\delta}$$

で行った。 δ は 10^{-8} とした。

3.3 CV 値を利用したトラッキング

CV 値が既知の場合、この情報を使うことで、精度の高いトラッキングを行うことができる。複数フレームのトラッキングができると、特徴点の三次元座標 P が計算でき、CV 値が分かっていることから、他のフレームでの投影位置 \bar{p} が計算できる。そのため、 \bar{p} の周辺の狭い範囲のみ画像探索すればよく、誤った対応点を見つけてしまう可能性が大きく減る。

4 処理時間短縮と精度向上

4.1 更新式の計算

3.2.4 節で述べた反復演算では、何度も更新式

$$\Delta \mathbf{x} = (H + \lambda I)^{-1} \frac{dC(\mathbf{x})}{d\mathbf{x}}$$

の計算を行う。ここで、行列 H は $3M + 6N - 7$ 次の大きな対称行列であるが、ゼロである成分が多いため、より小さな行列単位で計算を行うことができる。こうすることで、計算速度の向上と、メモリ使用量を抑えることができる。

\mathbf{x} を、特徴点に関する $3M$ 個の変数 \mathbf{x}_s と CV 値に関する $6N - 7$ 個の変数 \mathbf{x}_m に分けて書き、

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_s \\ \mathbf{x}_m \end{pmatrix}, \quad \Delta \mathbf{x} = \begin{pmatrix} \Delta s \\ \Delta m \end{pmatrix}, \quad H + \lambda I = \begin{pmatrix} A & B \\ B & C \end{pmatrix}, \quad \frac{dC(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} d_s \\ d_m \end{pmatrix}$$

とそれぞれおく。すると、 A は特徴点に関する変数の行列となり、特徴点ごとに 3 次対称行列が並んだブロック対角対称行列になる。 C は CV 値に関する変数の行列となり、フレームごとに 6 次対称行列が並んだブロック対角対称行列になる。 B はトラッキングに対応する行列となり、一つのトラッキングデータが 6×3 行列のブロックに対応する。それ以外の部分はゼロとなる。図 4 に示したうち、黒い部分にのみ有意な数値が入り、白い部分は全てゼロである。

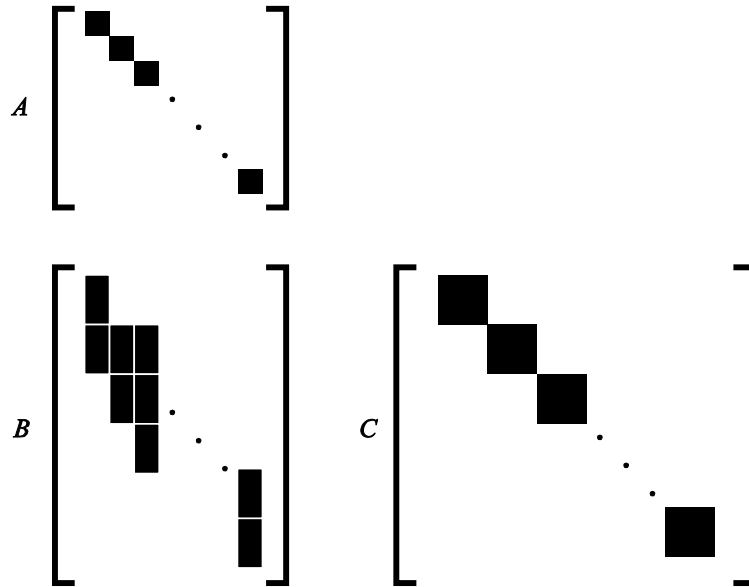


図 4: 行列イメージ

$D = BA^{-1}$ とおき、もとの式を変形すると、

$$\Delta m = (C - D^t B)^{-1} (d_m - D d_s),$$

$$\Delta s = A^{-1} d_s - {}^t D \Delta m$$

が得られ、この式により更新に必要な $\Delta \mathbf{x}$ を計算できる。

なお、計算途中で出現する行列 $C - D^t B = C - BA^{-1} {}^t B$ は $6N$ 次対称行列で、三角行列部分のメモリ量で保持することができる。

4.2 CV 値の連結

4.1 節で述べた計算でも、数千フレーム以上を同時に計算するとなると、計算途中の行列 $C-D'B$ が大きくなりすぎるため一度に計算するのは現実的ではない。そこで、短い区間に区切って計算を行い、それを連結することを考える。

CV 値は相対関係を表しているため、 $\{T^n, R^n\}$ に対し、縮尺 s による拡大縮小、三次元ベクトル T による並進移動、回転行列 R による回転変換

$$\{sRT^n + T, RR^n\}$$

を自由に行うことができる。

ここで、区間を重複させて得られた CV 値 $\{S^l, Q^l\}$ と $\{T^n, R^n\}$ があり、フレーム f_0 と f_1 が両方に含まれているとする。 $\{T^n, R^n\}$ を変換して $\{S^l, Q^l\}$ に連結することを考えると、連立方程式

$$\begin{cases} s \|T^{f_1} - T^{f_0}\| = \|S^{f_1} - S^{f_0}\| \\ sRT^{f_0} + T = S^{f_0} \\ RR^{f_0} = Q^{f_0} \end{cases}$$

から s, T, R を求めて変換を行えばよい (図 5)。

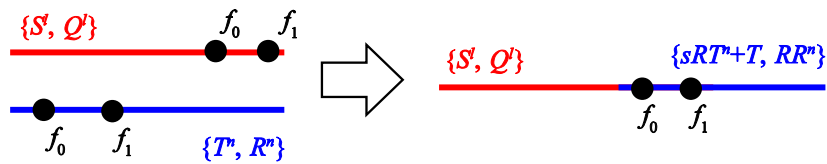


図 5: 連結イメージ

この結果、どれだけ長い映像でも CV 値を計算することができる。

4.3 トラッキングデータの選別

これまでの計算では、各トラッキングデータはピクセル量子化誤差などの微少な誤差を持つ前提であった。しかし、現実には全く違う特徴点を追ったり、移動物体を追うなど、微少ではない誤差を持つものが混じっている。そのため、計算時には逐次このミストラッキングによるデータを取り除くことが必要になる。

ミスを含むデータであっても、大半が正しいデータであれば、CV 計算を行うことである程度正しい CV 値は得ることができ、各トラッキングごとに投影誤差の計量ができることになる。この誤差量をもとにミストラッキングによるデータを選別し、CV 値の精度を高めてゆく。ミストラッキングによる特徴点は正しいものに比べて誤差が大きくなることから、トラッキングの投影誤差総和 $C = \sum_{m,n} (c^{mn})^2$ をトラッキング数で割った平均 \bar{C} と、特徴点ごとのトラッキング投影誤差総和 $C_m = \sum_n (c^{mn})^2$ をトラッキング数で割った平均 \bar{C}_m とを比べ、

$$\bar{C}_m > r\bar{C}$$

の場合に m 番目の特徴点のトラッキングデータを削除する。 r は閾値を表す定数である。現在は 1 フレームを CV 計算に追加して前記マルカート法による解を求めるごとに一度この処理を実施している。