

部分画像列からのパノラマ画像生成法

勝山 剛・島田 英之・塩野 充・宮垣 嘉也

岡山理科大学工学部情報工学科

(1998年10月5日 受理)

1. まえがき

近年、画像処理技術の発展が目覚ましく、テレビジョン、ファクシミリ、コンピュータ端末などさまざまな方面に応用されている。また、コンピュータや周辺技術の発展に伴い、画像処理の手法も多様化している。

そこで、自分で撮った写真を合成するために、パソコンのアプリケーションソフトなどで画像を加工し、手作業でひとつずつ画像を合成し、ひとつのパノラマ画像に合成するなどの需要も多くなってきた。しかし、空間的な画像の歪み、明るさやコントラストのずれのために、手作業では正確な合成が不可能な場合が多く、たとえ合成できても大変な手間と時間がかかる。

そこで本研究では、部分画像列を、その位置、大きさ、傾きを指標となる画像に合わせて補整を行う。しかし、それだけでは明るさやコントラストなどが違う画像を合成すると継ぎ目が目立って、違和感のある画像となる。そこで、本稿では明るさやコントラストの違いを解消するための補整を行い自動的に歪みや継ぎ目のないパノラマ画像に合成する手法について述べる。

2. 関連する各種技法

2.1 2D 投影変換

投影変換¹⁾の一般的な表現は、次の通りである。

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1.0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \text{or} \quad \mathbf{u}' = \mathbf{M}\mathbf{u} \quad (1)$$

$$x = \frac{x'}{w'} = \frac{m_0u + m_1v + m_2}{m_6u + m_7v + 1.0} \quad (2)$$

$$y = \frac{y'}{w'} = \frac{m_3u + m_4v + m_5}{m_6u + m_7v + 1.0} \quad (3)$$

平行移動の場合の \mathbf{M}

$$\mathbf{M} = \begin{bmatrix} 1.0 & 0.0 & m_2 \\ 0.0 & 1.0 & m_5 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (4)$$

m_2 は x 方向の移動量, m_5 は y 方向の移動量。

拡大・縮小の場合の \mathbf{M}

$$\mathbf{M} = \begin{bmatrix} m_0 & 0.0 & 0.0 \\ 0.0 & m_4 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (5)$$

m_0 は x 方向の拡大・縮小, m_4 は y 方向の拡大・縮小。

回転の場合の \mathbf{M}

$$\mathbf{M} = \begin{bmatrix} \cos\theta & -\sin\theta & 0.0 \\ \sin\theta & \cos\theta & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (6)$$

原点を中心として反時計回りに θ だけ回転。

せん断の場合の \mathbf{M}

$$\mathbf{M} = \begin{bmatrix} m_0 & m_1 & 0.0 \\ m_3 & m_4 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (7)$$

ここで, w は同次座標であり, 任意の 0 でないスカラー量である。同次座標表現では, n 次元の問題を $n+1$ 次元の問題として取り扱え, 同次座標 (x', y', w') を w' で割ることにより通常座標 (x, y) を得る。

2.2 補間法

画像の大きさの変換を行う場合, 変換前の画像の座標と交換後の座標の対応が必ずつくとは限らない。そこで, 対応付けの時に補間の必要が生じる。

そこで, 変換画像の一つの画素を (x_0, y_0) とする。

- (1) (x_0, y_0) の変換前の座標 (u_0, v_0) を, 変換式を用いて求める。
 - (2) 求めた座標 (u_0, v_0) の近くにある画素の値を使って, 補間により (x_0, y_0) の値を求める。
- 補間の方法としては, 最近傍法や線形補間法などがあるが, 本研究では, 線形補間法を用いた。

2.2.1 線形補間法

線形補間法とは, 図 1 のように座標 (u_0, v_0) のまわりの 4 つの画素を用いて, 線形に補間して求めることである。式は, 次のようになる。

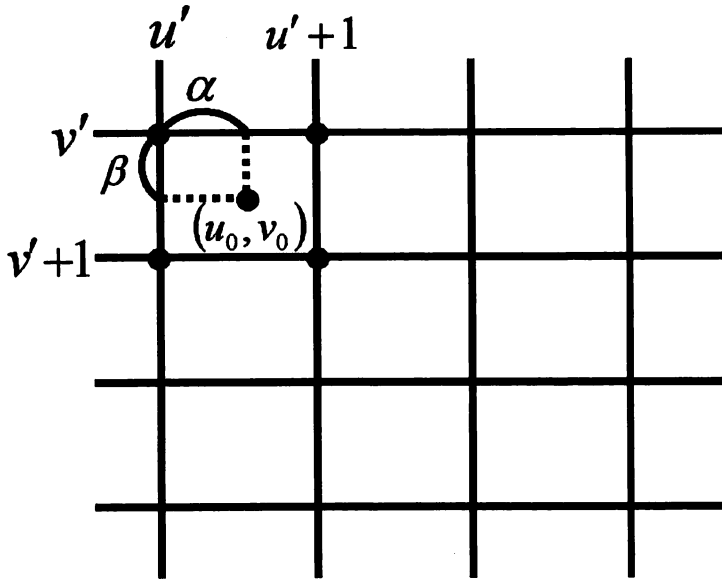


図1 線形補間

$$\begin{aligned}
 f(u_0, v_0) &= f(u', v')(1-\alpha)(1-\beta) \\
 &\quad + f(u'+1, v')\alpha(1-\beta) \\
 &\quad + f(u', v'+1)(1-\alpha)\beta \\
 &\quad + f(u'+1, v'+1)\alpha\beta
 \end{aligned}
 \tag{8}$$

2.3 最小2乗法

最小2乗法²⁾³⁾は、需要予測や回帰分析⁴⁾などの予測をするときなどに多く用いられる。実験や調査などで得られたデータは、種々の誤差を含んでいるが、最小2乗法は、ある程度は、これらの誤差を打ち消すような効果を持つ。

今、 n 個の独立したデータの組を

$$(x_i, y_i) \quad (i = 1, 2, \dots, n; \text{ただし } n > p)$$

とすると、このデータ区間の範囲内 $[x_1, x_n]$ における近似関数が、 x の p 次多項式

$$Y(x) = a_1x + a_2x^2 + \dots + a_px^p + a_0 \tag{9}$$

で表せるとすると、各点における $Y(x_i)$ と、 y_i との残差は、

$$e(x_i) = Y(x_i) - y_i \quad (i = 1, 2, \dots, n) \tag{10}$$

となる。また、この残差の2乗和を E とすると、 E は、

$$E = \sum_{i=1}^n e(x_i)^2 = \sum_{i=1}^n \{Y(x_i) - y_i\}^2 \quad (11)$$

で示される。この E が最小となるように a_j ($j = 0, 1, \dots, p$) を定めるのである。この a_j の値を求めるためには、 E を各 a_j に関してそれぞれ偏微分することによって、次のように得られる。

$$\begin{aligned} \frac{\partial E}{\partial a_j} &= \frac{\partial}{\partial a_j} \sum_{i=1}^n \{Y(x_i) - y_i\}^2 = 2 \sum_{i=1}^n \{Y(x) - y_i\} \frac{\partial Y(x)}{\partial a_j} \\ &= 2 \sum_{i=1}^n \{Y(x) - y_i\} x_i^j = 0 \end{aligned} \quad (12)$$

したがって、次のような係数に関する $p+1$ 元連立 1 次方程式が得られる。

$$\begin{cases} a_0 n + a_1 \sum_{i=1}^n x_i + \dots + a_p \sum_{i=1}^n x_i^p = \sum_{i=1}^n y_i \\ a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 + \dots + a_p \sum_{i=1}^n x_i^{p+1} = \sum_{i=1}^n y_i x_i \\ \vdots \\ a_0 \sum_{i=1}^n x_i^p + a_1 \sum_{i=1}^n x_i^{p+1} + \dots + a_p \sum_{i=1}^n x_i^{2p} = \sum_{i=1}^n y_i x_i^p \end{cases} \quad (13)$$

これはガウスの正規方程式と呼ばれている。これを行列—ベクトル記法で表すと、

$$\mathbf{A} \cdot \mathbf{a} = \mathbf{b} \quad (14)$$

となる。 \mathbf{A} は次式の係数行列、 \mathbf{b} はベクトルで既知数である。

$$\mathbf{A} \cdot \mathbf{a} = \begin{bmatrix} n & \sum x_i & \dots & \sum x_i^p \\ \sum x_i & \sum x_i^2 & \dots & \sum x_i^{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_i^p & \sum x_i^{p+1} & \dots & \sum x_i^{2p} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \vdots \\ \sum y_i x_i^p \end{bmatrix} \quad (15)$$

ただし、記号 \sum は $\sum_{i=1}^n$ の意味である。

係数行列の行列式の値が 0 でなければ、すなわち、 $p+1$ 個の方程式が互いに独立であれば、式(15)はただ 1 つの解をもつ。求められた解 a_j ($j = 0, 1, \dots, p$) を式(9)に代入すれば、得られたデータに近似した曲線 $Y(x)$ を求めることができる。

本研究では、2 枚の画像 A, B の対応する画素を 2D 投影変換で求め、画像 B を変形させる。画像 A の濃度値を I 、補間された画像 B の濃度値を I' とする。

2.3.1 濃度差の 2 乗和

次に画像 A, B の対応するすべての画素における濃度差 e_i と濃度差の 2 乗和 E を求める。そのときの式は次の通りである。

$$e = I'(x', y') - I(x, y) \quad (16)$$

$$E = \sum_i [I'(x'_i, y'_i) - I(x_i, y_i)]^2 = \sum_i e_i^2 \quad (17)$$

i は画像 A, B の対応する画素数。

2.3.2 濃度差の偏導関数

非線形最小 2 乗法⁵⁾により、濃度差の 2 乗和 E を最小にするためにアフィン変換行列 \mathbf{M} に関する濃度差の偏導関数を求める必要があり、その式は、次の通りである。

$$\frac{\partial e_i}{\partial m_k} = \frac{\partial I'}{\partial x'} \cdot \frac{\partial m_k}{\partial x'} + \frac{\partial I'}{\partial y'} \cdot \frac{\partial m_k}{\partial y'} \quad (18)$$

$$k = 0, 1, \dots, 7 \quad \frac{\partial I'}{\partial x'}, \frac{\partial I'}{\partial y'} \dots \text{濃度勾配}$$

濃度勾配については次で説明する。

2.3.3 濃度勾配

濃度勾配は、画像の縦方向と横方向の濃度変化である。縦方向は y 、横方向は x である。今回は、中心差分法を用いて濃度勾配を計算している。

計算式は次の通りである。

$$\left. \begin{aligned} \frac{\partial I'}{\partial x} &= \frac{I'(x+1, y) - I'(x-1, y)}{2} \\ \frac{\partial I'}{\partial y} &= \frac{I'(x, y+1) - I'(x, y-1)}{2} \end{aligned} \right\} \quad (19)$$

2.3.4 近似ヘッセ行列と重み付き勾配ベクトル

近似ヘッセ行列と重み付き勾配ベクトルの式は次の通りである。

$$A_{kl} = \sum_i \frac{\partial e_i}{\partial m_k} \cdot \frac{\partial e_i}{\partial m_l} \quad (20)$$

$$b_k = \sum_i e_i \cdot \frac{\partial e_i}{\partial m_k} \quad (21)$$

$$k = 0, 1, \dots, 7 \quad l = 0, 1, \dots, 7$$

式(20)が近似ヘッセ行列で、式(21)が重み付き勾配ベクトルである。

近似ヘッセ行列 \mathbf{A} と重み付き勾配ベクトル b は

$$\mathbf{A} = \begin{bmatrix} a_{00} & a_{01} & \cdots & a_{07} \\ a_{10} & a_{11} & \cdots & a_{17} \\ \vdots & \vdots & & \vdots \\ a_{70} & a_{71} & \cdots & a_{77} \end{bmatrix} \quad b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_7 \end{bmatrix} \quad (22)$$

のように表す。 \mathbf{A} は 8 行 8 列の行列で、 b は 8 行 1 列の行列になる。

次に、

$$\mathbf{A} \cdot \Delta \mathbf{m} = \mathbf{b} \quad (23)$$

より行列 \mathbf{M} を求める。そのために、 \mathbf{A} の逆行列を求めて解くと、

$$\Delta \mathbf{m} = \mathbf{A}^{-1} \mathbf{b} \quad (24)$$

のようになり、行列 \mathbf{M} が求まる。

3. 合成処理

全体の手順として、画像の取り込み→歪みの補整→明るさの補整→画像の合成という流れで行う。以下では、各手順について述べる。

3.1 画像取り込み

デジタルカメラを用いて重複領域のある数枚の写真を撮る。画像サイズはそれぞれ Width を160, Height を120とする。画像は、白黒濃淡画像とする。

3.2 歪みの補整

歪んだ画像を、そのまま正確に合成することは不可能である。そこで、歪んだ画像を歪みのない画像に補整する必要がある。

歪みの補整の手順を以下に示す。

- (1) 歪みのない画像にするために、2枚の画像のA, Bの対応する画素を2D投影変換により求める。だが2D投影変換行列 \mathbf{M} の値がわからないので、初期値を単位行列として計算する。
- (2) 求まった対応するすべての画素間の濃度差の2乗和を式(17)より求める。
- (3) 非線形最小2乗法により、変換行列 \mathbf{M} に関する濃度差の偏導関数を式(18)より求める。そのために、画像の x 方向と y 方向の濃度勾配を式(19)より求める必要がある。
- (4) 近似ヘッセ行列と重み付き勾配ベクトルを式(20)と式(21)より求め、式(23)に代入し、近似ヘッセ行列の逆行列を求め、重み付き勾配ベクトルとの積を求め、行列 \mathbf{M} の増加分 $\Delta \mathbf{m}$ の値を求める。
- (5) 行列 \mathbf{M} に行列 \mathbf{M} の増加分 $\Delta \mathbf{m}$ を加える。
- (6) 手順1～5を濃度差の2乗和が収束するまで繰り返し、最終的な変換行列 \mathbf{M} の値を決定する。
- (7) 最終的に求まった変換行列 \mathbf{M} で、画像Aの位置に合わせて歪みの補整を行う。このようにして歪みの補整を行った。

3.3 明るさの補整

歪みの補整をし、画像を合成すると、明るさが違う画像の場合、画像の継ぎ目がはっきりと見えてしまう。そこで、継ぎ目をなくすために最小2乗法を用いて片方の画像に合わ

せて明るさを補整する。

明るさの補整の手順を以下に示す。

(1) 画像Aと画像Bの対応する画素の濃度値の関係式を作る。その式は次のようになる。

$$y = a_1 I'(x', y') + a_0 \quad (25)$$

$$e = I(x, y) - y \quad (26)$$

(2) 2枚の対応するすべての画素間の濃度差の2乗和を求める。

(3) 最小2乗法により、濃度差の2乗和を a_1 , a_0 で偏微分する。その値が0の時、濃度差の2乗和が最小となり、その時の a_1 , a_0 を求める。

(4) そのために連立一次方程式を解き、 a_1 , a_0 を求める。

(5) 求めた a_1 , a_0 で画像Aに合わせて明るさの補整をする。

このようにして明るさの補整を行った。

3.4 画像の合成

画像を合成する際には、エッジを目立たなくするために画像中心からの距離に応じて図2のような荷重平均を用いて、最終的な濃度値を決定し、合成する。

次式において、 w は画像中心から離れるにしたがって小さくなる重み、 I は各画像の濃度、 H は最終的な濃度値を示す。

$$H(x, y) = \frac{w(x)w(y)I(x, y) + w(x')w(y')I'(x', y')}{w(x)w(y) + w(x')w(y')} \quad (27)$$

以上の処理で合成画像を作成できる。

4. 実験結果と考察

4.1 実験結果

本研究での処理結果を図3に示す。

(1) 画像の取り込み

白黒画像で、大きさは Width 160, Height 120 で取り込んだ。その例を図3に示す。

また、図3の画像Bは異なる明るさで取り込んだ。

(2) 歪みの補整

3.2節で述べた手順により画像を処理する。

図3の画像Bを画像Aに合わせて歪みの補整をした画像を図3の(d)に示す。

(3) 明るさの補整

3.3節で述べた手順により画像を処理する。

図3の画像Bを画像Aの明るさに合わせて明るさの補整をした画像を図3の(f)に示す。

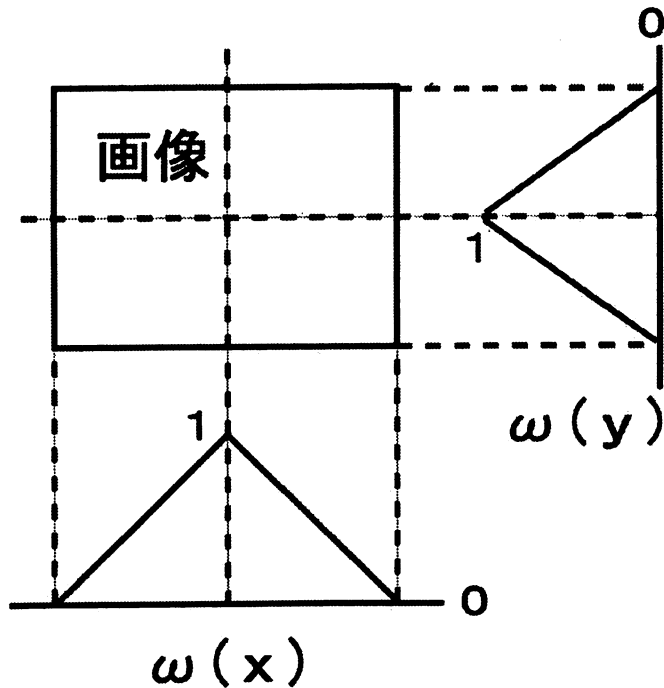


図2 荷重平均

(4) 画像の合成

3.4節で述べた手順により画像を合成する。

図3の画像Aと画像Bを合成した画像を図3の(g)に示す。

4.2 考察

(1) 歪みの補整について

本研究では、2D投影変換を用いて指標とする画像の位置に合わせて歪みのない画像に変形させるという補整を行った。大部分が重なっていないと補整が出来ないが、手動である程度移動させると補整を行うことが出来た。また、明るさがあまりに違う場合、歪みの補整がうまく出来ないことが分かった。

(2) 明るさの補整について

2枚の画像の重なった部分の濃度値を、最小2乗法を用いて近づけることにより、明るさの補整を行った。式(25)の a_1 でコントラストを、 a_0 で明るさを補整することが出来た。

(3) 画像の合成について

歪んだ画像でも、明るさやコントラストの違う画像でも上の2つの補整を行うことにより、きれいな合成画像を作成することが出来た。しかし、画像によっては雑音や継ぎ目が目立つ場合もある。

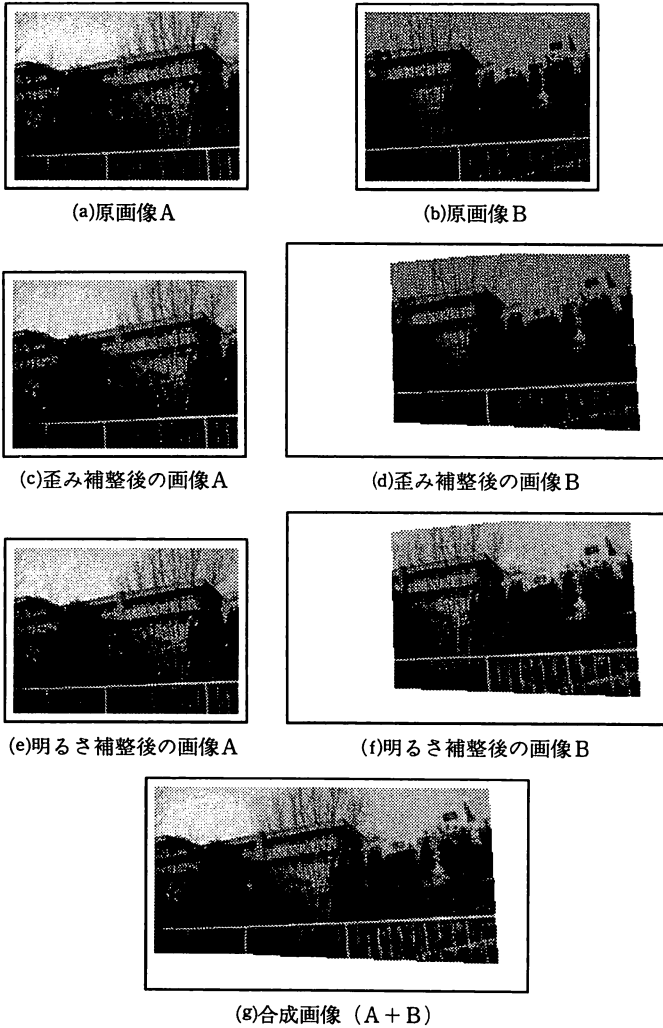


図3 合成結果

5. ま と め

本研究では、いくつかの画像を合成して大きな1枚の画像を作成することを目的とする部分画像列からのパノラマ画像生成法について述べた。

本研究で得られた結論を以下に述べる。

本研究では、2D投影変換と非線形最小2乗法を用いて、歪みの補整を行った。歪み補整する際に、濃度勾配によって補整を行っているため、明るさの異なる画像である場合も歪み補整を行うことが出来た。明るさの補整には、最小2乗法を用いた。歪みの補整が正しく行えると明るさの補整もうまく行うことができた。本研究では、2つの補整をすることにより、自動的に歪みや継ぎ目のない合成画像を得ることができた。

本研究の改良点として、次のことが挙げられる。

- (1) 本研究で使用した歪みの補整の手法では、画像の重なりが少ない場合正しく補整が出来ない。現状では手動により移動させているので、画像のおおまかな位置合わせを行う前処理が必要である。
- (2) 歪みの補整では、濃度差の2乗和 E が収束すると補整が終わる。しかし、収束の判定を行っていないので、 E の値や反復回数にしきい値を設けるなどの方法で収束判定をするプログラムが必要である。
- (3) 本研究では、白黒画像だけを取り扱っていて、カラー画像は、取り扱っていない。そこで、カラー画像にも対応できるプログラムが必要である。
- (4) 画像を合成するまでの時間が非常にかかる。そこで、計算速度の高速化が必要である。

参考文献

- 1) 小堀研一, 春日久美子著: "基礎から学ぶ図形処理", 工業調査会 (1996).
- 2) 竹本宜弘, 荒 実著: "Cによる数値計算", 朝倉書店 (1987).
- 3) 川上一朗著: "数値計算", 岩波書店 (1989).
- 4) 有馬 哲, 石村貞夫著: "多変量解析のはなし", 東京図書株式会社 (1987).
- 5) Richard Szeliski: "Video Mosaics for Virtual Environments", IEEE Computer Graphics and Application, March 1996, pp. 22-30 (1996).

A Method of Making a Panoramic Image from Partial Scenes

Go KATSUYAMA, Hideyuki SHIMADA,
Mitsuru SHIONO and Yoshiya MIYAGAKI
*Graduate School of Engineering,
Department of Information & Computer Engineering,
Faculty of Engineering,
Okayama University of Science,
Ridai-cho 1-1, Okayama 700-0005 Japan*
(Received October 5, 1998)

It is very difficult to make an exact panoramic image manually from warped image and taken under different light scenes. And this work takes a great deal of time.

In this paper, we have described a method of making an exact panoramic image from partial scenes.

- (1) The user can get an exact and fine panoramic image easily from different light scenes with this method.
- (2) Gathering a lot of pictures make it possible to create a high resolution image. We can get a panoramic image with no joint by this method.