

実画像を用いた3次元物体への正面テクスチャマップの作成

松井 裕司 徐 剛

立命館大学理工学部情報学科コンピュータビジョン研究室, 草津市

E-mail: xu@cv.cs.ritsumeai.ac.jp

http://www.cv.cs.ritsumeai.ac.jp

あらまし CGにおいては高速化のためにテクスチャマッピングをアフィン射影を用いて行なっている。そのため小さな物体ではさほど影響はないが大きな物体で、しかも視線方向と対象平面との角度が小さいときの写真を貼りつけるときには歪みが生じてしまう。そこで、その歪みを極力減少させるために、実写の画像を各面に対応する部分画像に分割し、それぞれの部分画像をあたかもその正面から見たような画像に変換する。まず3次元形状が既知のモデルと撮影した写真との間の射影関係を定める射影行列を求める。次に射影行列からカメラの内部変数及びカメラ座標系と物体座標系との間の回転行列と並進ベクトルを求める。それを利用して各パッチのホモグラフィ変換行列を求め、物体の面に合わせた正面画像を生成する。最後に、得られた正面画像をVRMLで記述された3次元物体にテクスチャマッピングした結果を示す。

Generating Front Texture Maps for 3D Objects from Real Images

Yuji Matsui Gang Xu

Computer Vision Laboratory, Department of Computer Science, Ritsumeikan University, Kusatsu-shi, 525-8577, Japan

Abstract In this paper we introduce an algorithm to generate front texture maps from real images for 3D objects. In computer graphics, texture is usually mapped by affine projection, which is much faster than the exact homography. However, affine projection introduces unrealistic deformation, which becomes unnegligible when the surface is large and its image is taken from a large angle. To reduce the deformation while keeping the efficiency of affine projection, we generate front texture for each 3D plane using homography. We first determine the projection matrix between the known 3D shape and the real image, and then determine the homography matrix for each triangular surface patch. The homography matrix turns out to be very simple. Experimental results are shown.

1 はじめに

CGにおいて、3次元モデリングされた物体へテクスチャを貼りつける場合には、高速化のためにテクスチャをアフィン変換によって変換することによりマッピングが行われている [1].

テクスチャのデータがペイントソフト等を用い自分で作成したものである場合には、正面画像となっているので画像の歪みはあまり問題はないが、実写の画像をそのまま用いると自分で作成したテクスチャのときより大きな歪みが生じてしまう。この画像の歪みを減少させるためには、対象物体の正面から見た画像の入手が必

要となる。

しかし、実写の画像はCGとは違い、マッピングする面に合わせた真正面からの画像を入手することは一般的に困難である。そのため、3次元物体へ実写の画像をマッピングする場合、その3次元物体を構成している各パッチの正面から見た画像に合うように変換してやらなければならない。つまり、実際の写真では真正面からの撮影は困難であり、カメラの内部変数および外部変数を求め、写真の画像を各パッチの正面から見た画像に変換する必要がある。

本論文では、撮影した画像を正面の画像に変

換することで、面に合わせた画像を生成する手法について述べ、それを実際に VRML(Virtual Reality Modeling Language) で記述されたモデルに貼り付けた実験結果を示す。

VRML とは、インターネット上で CG を記述するための言語の 1 つであり、仮想の世界をホームページで作成してその中を歩き回ったりするような、3 次元的な情報を扱うことを実現できるようにした技術である [2]。

2 アフィン変換とホモグラフィ変換

2.1 アフィン変換

アフィン変換は図 1 に示すような左側の画像から右側の画像に変換する手法である。この手

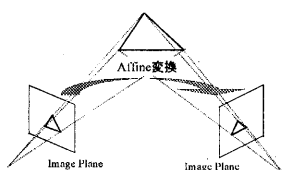


図 1: アフィン変換の概念図

法ではそれぞれの画像の 3 頂点がわかっているならば、三角形内部の点の座標は式 (1) によって容易に変換できる。

$$\begin{aligned} \bar{x}' &= \mathbf{T}\bar{x} & (1) \\ \bar{x} &= [x', y', 1]^T, \bar{x}' = [x, y, 1]^T \\ \mathbf{T} &= \begin{bmatrix} a & b & x_0 \\ c & d & y_0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

ここで、 \bar{x} は変換前の画像の座標、 \bar{x}' は変換後の画像の座標、 \mathbf{T} はアフィン変換行列である。

アフィン変換は高速であるが、実画像において視線方向と平面との角度がないときなどにこの変換を用いると画像に歪みが生じてしまう。そこで、本論文ではアフィン変換時に生じる歪

みを緩和するためにホモグラフィ変換を用いてまず正面画像を生成した。

2.2 ホモグラフィ変換の有効性

ホモグラフィ変換の有効性を確かめるために、図 2 に示すような変換を行なった画像を作成した。まず、視点 P_1 から見た中心射影の画像を作成する (図 3)。次にその画像から視点 P_2 から見た 3 次元平面に正面向きの画像を、ホモグラフィ変換によって作成する (図 4)。そして、視点 P_3 から見た中心射影の画像 (図 5)、視点 P_1 から視

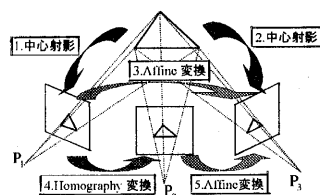


図 2: 有効性確認のための画像変換

点 P_3 へアフィン変換した画像 (図 6)、視点 P_2 から視点 P_3 へアフィン変換した画像 (図 7) を作成し、比較する。

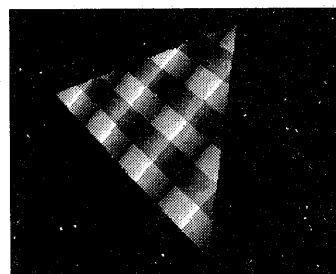


図 3: 視点 P_1 へ中心射影した結果

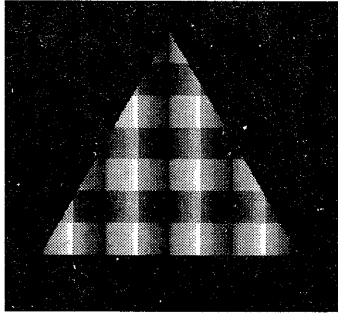


図 4: 視点 P_1 から視点 P_2 へホモグラフィ変換した結果

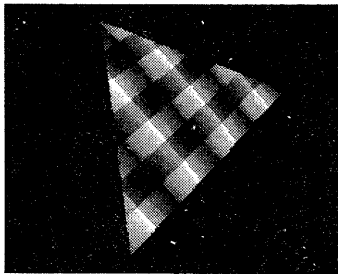


図 5: 視点 P_3 へ中心射影した結果

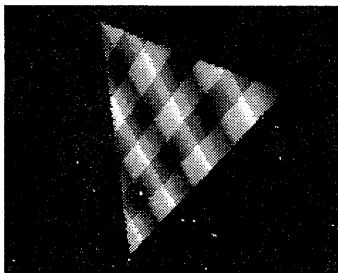


図 6: 視点 P_1 から視点 P_3 へアフィン変換した結果

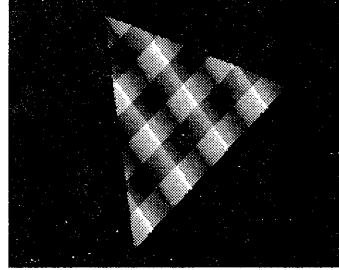


図 7: 視点 P_2 から視点 P_3 へアフィン変換した結果

図 5, 6, 7 の結果から明らかなように, アフィン変換によって直接画像を作成するよりも, いったんホモグラフィ変換で 3 次元平面に正面向きの画像を作成してからアフィン変換を行うほうが, 歪みの少ない画像が得られる.

3 カメラの校正

正面の画像を生成するためには, まず, 3 次元物体と画像の間の射影行列を求め, カメラ座標系と物体座標系との間の回転行列と並進ベクトルを求める必要がある. このプロセスがカメラの校正である [3]. 校正には

1. 3 次元点とその画像の間の射影行列を求める
 2. 射影行列から内部変数と外部変数を求める
- という 2 つのステップがある. 各々のステップについて 3.1 と 3.2 で述べる.

3.1 射影行列の算出

図 8 は既知パターンによる校正の様子を示している. 射影行列は, 3 次元形状が既知のパターンを観察し, 3 次元点 (X, Y, Z) と画像上の点の座標 (u, v) から求めることができ, 次のようになる.

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}} = \mathbf{A}[\mathbf{R}|\mathbf{t}]\tilde{\mathbf{M}} \quad (2)$$

ここで,

$$\tilde{\mathbf{m}} = [u, v, 1]^T, \tilde{\mathbf{M}} = [X, Y, Z, 1]^T$$

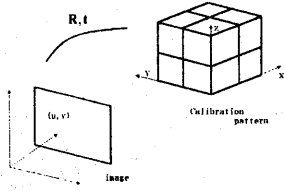


図 8: 3次元形状が既知のパターンを観察し3次元点と画像点の座標から射影行列を求められる

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1^T & p_{14} \\ \mathbf{p}_2^T & p_{24} \\ \mathbf{p}_3^T & p_{34} \end{bmatrix} \quad (3)$$

であり, s はスカラーである.

射影行列 \mathbf{P} は 3×4 の行列で, 12 個の要素があるが, カメラの内部変数と外部変数は合わせて 11 個ある. 式 (2) を展開すれば, 1 つの 3 次元 \mathbf{M}_i 点とその 2 次元画像 (u_i, v_i) から \mathbf{P} の要素に関する 2 つの線型方程式を次のように立てることができる.

$$\begin{aligned} \mathbf{p}_1^T \mathbf{M}_i - u_i \mathbf{p}_3^T \mathbf{M}_i + p_{14} - u_i p_{34} &= 0 \quad (4) \\ \mathbf{p}_2^T \mathbf{M}_i - v_i \mathbf{p}_3^T \mathbf{M}_i + p_{24} - v_i p_{34} &= 0 \end{aligned}$$

N 点あれば, 次の方程式が得られる.

$$\mathbf{Lp} = 0 \quad (5)$$

ここで, $\mathbf{p} = [p_1^T, p_{14}, p_2^T, p_{24}, p_3^T, p_{34}]^T$ は \mathbf{P} の要素を並べたもので,

$$\mathbf{L} = \begin{bmatrix} X_1 & 0 & \dots & X_n & 0 \\ Y_1 & 0 & \dots & Y_n & 0 \\ Z_1 & 0 & \dots & Z_n & 0 \\ 1 & 0 & \dots & 1 & 0 \\ 0 & X_1 & \dots & 0 & X_n \\ 0 & Y_1 & \dots & 0 & Y_n \\ 0 & Z_1 & \dots & 0 & Z_n \\ 0 & 1 & \dots & 0 & 1 \\ -u_1 X_1 & -v_1 X_1 & \dots & -u_n X_n & -v_n X_n \\ -u_1 Y_1 & -v_1 Y_1 & \dots & -u_n Y_n & -v_n Y_n \\ -u_1 Z_1 & -v_1 Z_1 & \dots & -u_n Z_n & -v_n Z_n \\ -u_1 & -v_1 & \dots & -u_n & -v_n \end{bmatrix}^T$$

は式 (5) における N 個の 3 次元点と 2 次元座標から定義される $2N \times 12$ の行列である. 射影の性質上, \mathbf{P} (すなわち \mathbf{p}) のスケールは任意であ

る. 上記で述べたように, \mathbf{P} は 11 個の内部変数と外部変数に依存する. そのため, 6 点の対応があれば \mathbf{P} が求まる. もし N 個の 3 次元点と同じ平面になければ, 一般的に \mathbf{L} のランクは 11 である. よって, 12×1 の \mathbf{p} は $\|\mathbf{p}\| = 1$ の下で $\mathbf{L}^T \mathbf{L}$ の最小の固有値に対する固有ベクトルとして求められる. ただし, 3 次元座標と画像の座標をそのまま用いると数値計算時の誤差が大きくなるため, 正規化を行う必要がある. その方法は付録を参照されたい.

3.2 内部変数と外部変数の算出

\mathbf{P} が求まれば, 次はそれから $\mathbf{A}, \mathbf{R}, \mathbf{t}$ を求める.

$$\mathbf{A} = \begin{bmatrix} \alpha_u & -\alpha_u \cot \theta & u_0 \\ 0 & \alpha_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

で表し, 式 (2) に代入すると式 (3) の要素は次のようになる.

$$\begin{aligned} \mathbf{p}_1^T &= \alpha_u \mathbf{r}_1^T - \alpha_u (\cot \theta) \mathbf{r}_2^T + u_0 \mathbf{r}_3^T \\ \mathbf{p}_2^T &= (\alpha_v / \sin \theta) \mathbf{r}_2^T + v_0 \mathbf{r}_3^T \\ \mathbf{p}_3^T &= \mathbf{r}_3^T \\ p_{14} &= \alpha_u t_x - \alpha_u (\cot \theta) t_y + u_0 t_z \\ p_{24} &= (\alpha_v / \sin \theta) t_y + v_0 t_z \\ p_{34} &= t_z \end{aligned}$$

ここで, 固有ベクトルとして求められた \mathbf{p} の符号とスケールを決める必要がある. まず, 物体がカメラの前にあることから,

$$t_z = p_{34} > 0$$

である. もし, $p_{34} < 0$ なら, $-\mathbf{p}$ も式 (5) を満たすので, \mathbf{p} の全ての要素の符号を逆にする. また, 式 (6) には $\mathbf{r}_3 = \mathbf{p}_3$ とあるので, $\|\mathbf{p}_3\| = \|\mathbf{r}_3\| = 1$ となるように \mathbf{P} のスケールを合わせる. すると, 式 (6) より次式が容易に得られる.

$$\begin{aligned} \mathbf{r}_3 &= \mathbf{p}_3 \\ u_0 &= \mathbf{p}_1^T \mathbf{p}_3 \\ v_0 &= \mathbf{p}_2^T \mathbf{p}_3 \end{aligned}$$

さらに, $\alpha_u, \alpha_v, \theta$ に関しては, 式

$$\begin{aligned}\alpha_u &= \sqrt{\mathbf{p}_1^T \mathbf{p}_1 - u_0^2 \sin^2 \theta} = \|\mathbf{p}_1 \times \mathbf{p}_3\| \sin \theta \\ \alpha_v &= \sqrt{\mathbf{p}_2^T \mathbf{p}_2 - v_0^2 \sin^2 \theta} = \|\mathbf{p}_2 \times \mathbf{p}_3\| \sin \theta \\ \theta &= \cos^{-1} \left(\frac{(\mathbf{p}_1 \times \mathbf{p}_3)^T (\mathbf{p}_2 \times \mathbf{p}_3)}{\|\mathbf{p}_1 \times \mathbf{p}_3\| \|\mathbf{p}_2 \times \mathbf{p}_3\|} \right)\end{aligned}$$

が得られる. ここで θ が 0 から π の範囲に制限し, $\sin \theta$ を常に正とする. また, 残りの変数は次のように求められる.

$$\begin{aligned}r_2 &= \frac{\sin \theta}{\alpha_v} (\mathbf{p}_2 - v_0 \mathbf{p}_3) \\ r_1 &= \frac{1}{\alpha_u} (\mathbf{p}_1 + (\mathbf{p}_2 - v_0 \mathbf{p}_3) \frac{\alpha_u}{\alpha_v} \cos \theta - u_0 \mathbf{p}_3) \\ t_y &= \frac{\sin \theta}{\alpha_v} (p_{24} - v_0 p_{34}) \\ t_x &= \frac{1}{\alpha_u} (p_{14} + (p_{24} - v_0 p_{34}) \frac{\alpha_u}{\alpha_v} \cos \theta - u_0 p_{34})\end{aligned}$$

4 正面画像の生成

4.1 正面画像への変換

図9に概念図を示す. 3次元点 $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ で

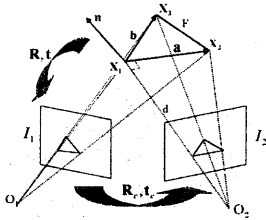


図9: 正面画像生成の概念図

構成される面を O_1 の視点で見た画像を I_1 とする. また, O_2 の視点は $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ で構成される面 F を正面から見ており, その画像を I_2 とする. ただし, 3角形の座標 $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ は反時計回りにとる.

ここで行うことは, O_1 の視点から見た画像を O_2 から見た画像に変換することである. I_1 と I_2 を写す2台のカメラの間に次のような関係があるとする.

$$\mathbf{X}'' = \mathbf{R}_c \mathbf{X}' + \mathbf{t}_c \quad (6)$$

ここで $\mathbf{X}', \mathbf{X}''$ は, それぞれ対象の3次元物体のある1点 \mathbf{X} の I_1 と I_2 のカメラ座標系における座標である. $\mathbf{R}_c, \mathbf{t}_c$ はそれぞれ I_1 と I_2 のカメラ間の回転行列及び並進ベクトルである.

対象物体が平面の場合, 両画像における対応点は次式を満たす.

$$s \bar{\mathbf{X}}'' = \mathbf{H} \bar{\mathbf{X}}' \quad (7)$$

ここで, s はスカラーであり, 行列 \mathbf{H} はホモグラフィ行列と呼ばれ次のようになる.

$$\mathbf{H} = \mathbf{R}_c + \frac{\mathbf{t}_c \mathbf{n}^T}{d} \quad (8)$$

ここで \mathbf{n} は O_2 座標系における平面 F の法線ベクトル, d は座標原点 O_2 から平面 F までの距離である [3].

さらに, デジタル画像座標 $\bar{\mathbf{m}}, \bar{\mathbf{m}}'$ を用いる場合, 式 (7) は

$$s \bar{\mathbf{m}} = \mathbf{H}' \bar{\mathbf{m}}' \quad (9)$$

となる. ただし,

$$\mathbf{H}' = \mathbf{A} \left(\mathbf{R} + \frac{\mathbf{t} \mathbf{n}^T}{d} \right) (\mathbf{A}')^{-1}$$

である. ここで, \mathbf{A} は I_1 を撮影したカメラの内部行列で, \mathbf{A}' は I_2 のカメラ内部行列である. ここで, 同じカメラを仮定し, $\mathbf{A}' = \mathbf{A}$ とする.

4.2 H 行列の算出

図9において, $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ は3次元物体の座標である.

まず, これを次式によって I_1 のカメラ座標系に変換する.

$$\mathbf{X}' = \mathbf{R} \mathbf{X} + \mathbf{t}$$

ここで, \mathbf{X}' は I_1 のカメラ座標系における座標, \mathbf{X} は世界座標系における座標で \mathbf{R}, \mathbf{t} は3次元物体と I_1 の画像の関係から得られた回転行列と並進ベクトルである. 座標系 O_1 における $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ の座標を $\mathbf{X}'_1, \mathbf{X}'_2, \mathbf{X}'_3$ とする.

次に, 画像 I_2 が平面 F の正面画像になるように, I_2 のカメラ座標系を決定しなければならない. まず, \mathbf{X}'_1 が画像 I_2 の座標系の原点にくるように

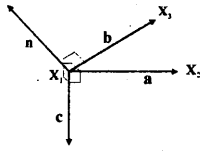


図 10: 法線ベクトルのとり方

する。次に X 軸の方向を定める。 X 軸を表す単位ベクトルを a とすると、次式のようになる。

$$a = \frac{X'_2 - X'_1}{\|X'_2 - X'_1\|}$$

そして、 Z 軸を平面 F の法線方向 n と同じにするために図 10 に示すように 3 次元平面の 2 本のベクトル a, b に垂直になるように b を以下に示される単位ベクトルである。

$$b = \frac{X'_3 - X'_1}{\|X'_3 - X'_1\|}$$

すると n は平面 F の二本のベクトル a, b の外積として求められるので、

$$n = b \times a = \frac{X'_3 - X'_1}{\|X'_3 - X'_1\|} \times \frac{X'_2 - X'_1}{\|X'_2 - X'_1\|} \neq 0$$

となる。 n は画像 I_2 のカメラの光軸の方向と同じである。また、 Y 軸方向を表す単位ベクトルを c とすると、これは次式のように n と a の外積によって求められる。

$$c = n \times a$$

よって、回転行列 R_c は次のように決定できる。

$$R_c = [a \ c \ n]$$

次に d を求める。画像 I_1 と画像 I_2 のスケールを同じにするため、平面 F を画像 I_1 も画像 I_2 も等距離から見る必要がある。そのため

$$d = \|X_1\|$$

とする。また、 t_c は式 (6) より

$$t_c = X'_1 - R_c \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix}$$

となる。これら R_c, d, t_c を式 (8) に代入して変形すると次式が得られる。

$$H = [a \ c \ \bar{X}'_1]$$

ただし、

$$\bar{X}'_1 = \frac{X'_1}{\|X'_1\|}$$

これを用いることによって、正面画像の生成ができる。しかし、 X'_1 の計算に誤差が含まれているので計算される X'_1 の代わりに画像 I_1 のカメラ座標系の原点 O_1 から画像 I_1 における X'_1 の座標へのベクトル u_1 を用いる。すると、 H 行列は次のようになる。

$$H = [a \ c \ \bar{u}_1]$$

これを用いて、正面画像の生成を行う。

5 実験結果

実験は 3 次元物体として立命館大学内にある建物を VRML 上に定義して図 11 に示す画像を貼り付けることを行った。図 11 の建物の部分を

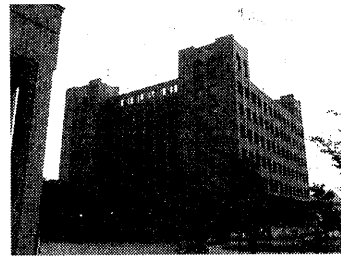


図 11: 原画像

アフィン変換によって張り付けた結果と建物の部分から正面画像を生成して、それを定義されたモデルに貼り付けた結果を図 12~図 15 に示す。ここで、図 12, 13 はアフィン変換のみで張り付けた結果であり、図 14, 15 はホモグラフィ変換を用いて正面画像を生成し、マッピングした結果である。

図 12~図 15 に示されているように、原画像をそのままマッピングするよりも、正面画像を生

成してそれを3次元物体に貼り付けた方が、画像の歪みが減少されている。ただ、誤差が含まれているため建物の窓枠のようなまっすぐにならないところがある。

また、正面画像を生成した際に、それをそのまま用いると斜辺の部分でピクセルが欠けていて生成した画像同士がうまく接合されない、というエイリアシングの問題が起きる。そこで生成された画像を一回りか二回り大きくしてから3次元物体に貼り付けてやる必要がある。

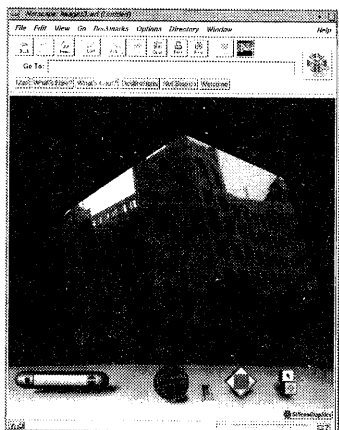


図 12: 原画像をそのまま貼りつけた結果 1

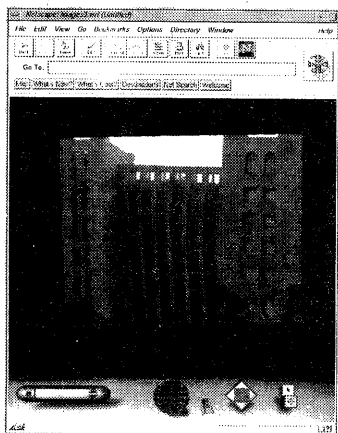


図 13: 原画像をそのまま貼りつけた結果 2

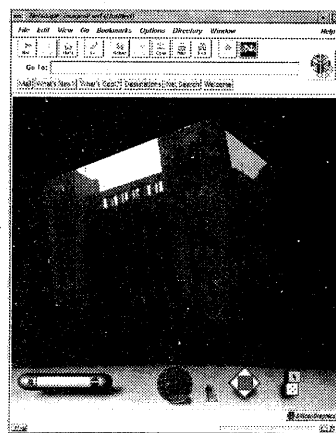


図 14: 原画像から正面画像を生成して貼りつけた結果 1

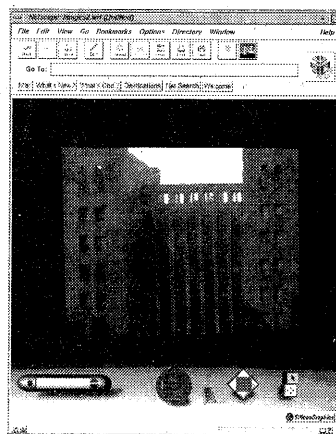


図 15: 原画像から正面画像を生成して貼りつけた結果 2

6 むすび

本論文では、実画像を3次元モデルに貼りつけるときに生じる歪みを減少させる手法を提案した。本手法では、まず3次元モデルと撮影した写真からとの間の射影関係を定める射影行列を求める。次に射影行列からカメラの内部変数及びカメラ座標系と物体座標系との間の回転行列と並進ベクトルを求める。それを利用して各パッチのホモグラフィ変換行列を求め、物体の面に合わせた正面画像を生成する。実画像に本手法を適用し、3次元物体へテキストチャマッピングを行った結果、本手法の有効性が確認できた。

参考文献

- [1] 杉原 厚吉, “グラフィックスの数理”, 共立出版, 1996
- [2] “Software Design”, 技術評論社, pp.148-155, May, 1998
- [3] 徐 剛, 辻 三郎, “3次元ビジョン”, 共立出版, 1998

A 正規化を用いたPの算出

画像に写っている物体の重心を $[u_0, v_0]^T$ とすると,

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \frac{1}{n} \sum_{i=1}^n \begin{bmatrix} u_i \\ v_i \end{bmatrix}$$

となる。さらに、 u, v の分散を次式で計算する。

$$\sigma_u = \sqrt{\frac{1}{n} \sum_{i=1}^n (u_i - u_0)^2}$$

$$\sigma_v = \sqrt{\frac{1}{n} \sum_{i=1}^n (v_i - v_0)^2}$$

正規化した後の座標を \tilde{e} とすると、 \tilde{e} は次式で表される。

$$\tilde{e} = B\tilde{m}$$

ここで,

$$B = \begin{bmatrix} \frac{1}{\sigma_u} & 0 & -\frac{u_0}{\sigma_u} \\ 0 & \frac{1}{\sigma_v} & -\frac{v_0}{\sigma_v} \\ 0 & 0 & 1 \end{bmatrix}$$

また、3次元物体の重心を $[X_0, Y_0, Z_0]^T$ とすると,

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = \frac{1}{n} \sum_{i=1}^n \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$$

となる。さらに、 X, Y, Z の分散を次式で計算する。

$$\sigma_X = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - X_0)^2}$$

$$\sigma_Y = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - Y_0)^2}$$

$$\sigma_Z = \sqrt{\frac{1}{n} \sum_{i=1}^n (Z_i - Z_0)^2}$$

正規化した後の座標を \tilde{E} とすると、 \tilde{E} は次式で表される。

$$\tilde{E} = C\tilde{M}$$

ここで,

$$C = \begin{bmatrix} \frac{1}{\sigma_X} & 0 & 0 & -\frac{X_0}{\sigma_X} \\ 0 & \frac{1}{\sigma_Y} & 0 & -\frac{Y_0}{\sigma_Y} \\ 0 & 0 & \frac{1}{\sigma_Z} & -\frac{Z_0}{\sigma_Z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

これらを式(2)に代入すると次のようになる。

$$sB\tilde{e} = PC\tilde{E}$$

これを变形すると,

$$s\tilde{e} = P'\tilde{E} \quad (11)$$

が得られる。ただし,

$$P' = B^{-1}PC$$

すると,

$$P = BP'C^{-1} \quad (12)$$

となる。式(11)より、正規化された座標を用いて P' を求めることができる。 P' を先に求めて、式(12)より本来の P を求めると、数値計算の誤差が小さくなる。