コンピュータビジョンと
イメージメディア 133−1
（2002. 5. 9）

# プロジェクタ・カメラシステムの キャリブレーションに関する研究

見市 伸裕, 和田 俊和, 松山 隆司
京都大学大学院情報学研究科知能情報学専攻

**概要** プロジェクタの画像を観測者に歪みなく見せるためには, 投影する画像を適切に補正しなければならない. この補正を行うためにカメラを用いる場合, プロジェクタの投影する画像とカメラの撮影する画像に関する画素毎の対応づけを求める必要があり, この対応づけを行うために空間コーディングが一般的に用いられている. しかし, 平面のスクリーンに対して空間コーディングを行う実験をした結果, 通常の実験環境で一般的に生じ得る画像のぼけによって, 空間コーディングの結果に誤差が含まれることを発見した. 本稿では, 画像のぼけによる誤差の大きさとその分布について考察する.

# Calibration of Projector-Camera System

Nobuhiro MIICHI and Toshikazu WADA and Takashi MATSUYAMA

Department of Intelligence Science and Technology,
Graduate School of Infomatics, Kyoto University

**abstract** To show a picture projected by a projector without distortion, it is necessary to correct the picture appropriately. When a camera is used to correct the picture, the picture projected and the picture captured have to be correlated pixel by pixel. Spacecoding is used generally to correlate them. But, when spacecoding for a flat screen we found out that the result includes errors because of being out of focus that can exist in usual experimental situation. In this report, we consider value and distribution of the error.

## 1 序論

プロジェクタは, 3 次元空間中の物体表面に画像を投影するデバイスである. このデバイスを用いれば, 現実世界に仮想的な物体を映し出すことができる. たとえば, 実際にはそこに存在しない 3 次元的な物体の形状を, 画像の投影されている部分に映し出すことが考えられる. このようなことを実現するには, プロジェクタの投影する画像が, 画像を提供する側が希望する形状で観測者に観測される必要がある. そのためには, 観測者の視点や投影する物体表面の形状に応じて投影する画像を補正しなければならない.

投影する画像の補正を行うために, 観測者の位置にカメラを置くことを考える. このような状況で上記のような補正を行うためには, プロジェクタに入力する画像とカメラから得られる画像の画素毎の対応関係を知る必要がある.

今, プロジェクタとその投影する画像を観測するカメラの組を, プロジェクタ・カメラシステムと呼ぶことにする. プロジェクタ・カメラシステムの両画像間の対応関係を求める手法として, 空間コーディング [4] が挙げられる. 空間コーディングは画素の対応づけを高速に数多く求めることができる優れた手法として, レンジファインダなどで一般的に用いられている [3]. 観測者の位置にカメラを置いた場合にも, 空間コーディングを用いて画素を対応づけすることが妥当であると思われる.

しかし, プロジェクタによる投影には被写界深度の違いやレンズの集光能力の限界によるぼけが含まれるので, 空間コーディングの結果には画像のぼけによる誤差が含まれる. 例えばプロジェクタの画像をスクリーンに対して斜めから投影した場合, 画像の場所によって投影される距離が異なるため, スクリーン上の画像の全域に渡ってピントが合うことはあり得ない. またピントを最大限合わせた状態で

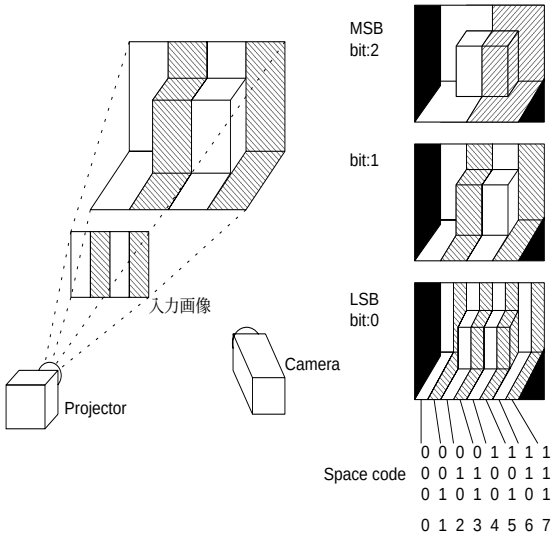



図 1: 空間コーディング

も, 特に対象までの距離が遠い場合にはレンズの集光能力が十分でないためにぼけが生じ得る. カメラに関しても全く同じ理由で, 撮影した画像にぼけが現れる.

プロジェクタ・カメラシステムではこれらの原因によって, 必ず画像にぼけが生じる. ぼけによる誤差の大きさは, カメラの撮影した画像上で数画素程度である. カメラを観測者の視点とみなしたプロジェクタ・カメラシステムでは投影対象からの距離が数 $m$ 以上である状況が一般的であり, 数画素に相当する領域はスクリーン上で数 cm の大きさになる. この場合のプロジェクタ・カメラシステムは観測者に画像を見せることを目的とするので, この数画素の誤差を無視することはできない. 特に投影対象が平面でない場合には, 見せるべき画像の一部が数画素のずれによって観測者の死角に入ることが考えられるため, ぼけによる誤差は重大な意味を持つ. 本稿では, スクリーンが平面である場合のプロジェクタ・カメラシステムについて空間コーディングを行いホモグラフィーを計算し, 空間コーディングの結果とホモグラフィーの対応づけを比較することによって, 画像のぼけによる誤差の分布を調査し原因を考察する.

## 2 空間コーディング

プロジェクタの投影する画像を補正するために, プロジェクタとカメラの画素間の対応関係を求めたい. プロジェクタの各画素を 1 点ずつ光らせ, それをカメラで撮影する方法をとれば, 確実に全画素間の対応関係を求めることができる. しかし, この方法では画素の数だけ撮影が必要となる. 空間コーディングは, 以下の作業によってより少ない回数でカメラの全画素に対する対応関係を求める手法である.

プロジェクタの画像の縦軸を $v$ 軸, 横軸を $u$ 軸とする. まず, 図 1 のようにプロジェクタの各画素についてその $u$ 座標値の範囲で入力画像をいくつかの領域に分割し, それぞれの領域に白, 黒を割り当てる. 領域を区切る $u$ 座標値の範囲を変化させて入力画像のパターンを変え, 投影された画像を順時カメラで撮影する. それらの撮影画像の任意の画素に対し画像毎の輝度値の変化を観測することで, 対応するプロジェクタの画素の $u$ 座標を復元する. 同様の作業を $v$ 座標についても行う.

この手法を用いれば, たとえば解像度 $1024 \times 768$ のプロジェクタに対し, 縦横 10 枚ずつの入力画像から全ての画素に対して対応関係を求めることができる. 以下ではコーディングに用いる白黒の入力画像を「コーディング画像」と呼び, コーディング画像から座標を復元する作業を「デコード」, プロジェクタとカメラの画素同士の対応関係を「対応点データ」と呼ぶ.

このように少ない撮影回数で全ての対応点データを求めることができる空間コーディングであるが, その結果がどの程度信頼できるものかを評価したい. そこで平面に対して空間コーディングを行い, その結果得られる対応点データとホモグラフィーによる対応づけを比較することによって, その信頼度の評価を行う. 次節ではホモグラフィーについて述べる.

## 3 ホモグラフィーによる検証

ステレオカメラで平面を観測する時, ホモグラフィーによってそれぞれの画像の画素同士を対応づけることができる. プロジェクタ・カメラシステムで平面のスクリーンに画像を投影する場合にも, ステレオカメラと同様ホモグラフィーによって画素を対応づけられる.

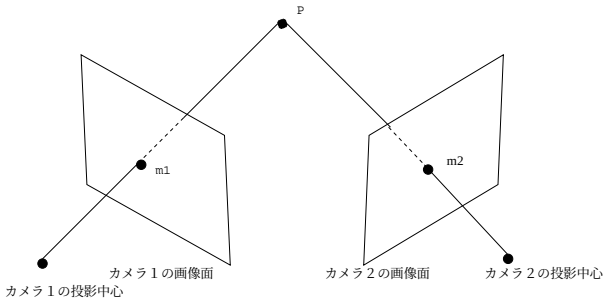



図 2: ステレオカメラの対応点データ $(m1, m2)$

ここでは, ステレオカメラ, プロジェクタ・カメラシステムのホモグラフィーとホモグラフィーの具体的な計算法について述べる.

## 3.1 ステレオカメラのホモグラフィー

ステレオカメラの 2 台のカメラをそれぞれカメラ 1, カメラ 2 とする. それぞれのカメラにはピンホールカメラモデルを適用するものとする.

図 2 のように空間中の同一の点を観測した時, その点がカメラ 1 の画像上で座標 $\mathbf{m_1}$, カメラ 2 の画像上で 座標 $\mathbf{m_2}$ で表されたとする. 3 次元空間中の同一平面上で, かつ両方のカメラの視野内にあるような全ての点について, $\mathbf{m_1}, \mathbf{m_2}$ の同次座標 $\tilde{\mathbf{m_1}}, \tilde{\mathbf{m_2}}$ が

$$\lambda\tilde{\mathbf{m_1}} = \mathbf{H}\tilde{\mathbf{m_2}} \tag{1}$$

を満たすような $\mathbf{H}$ が存在する. ここで, $\mathbf{H}$ は $3 \times 3$ 行列, $\lambda$ は $\mathbf{m_1}, \mathbf{m_2}$ によって決まる定数である. この行列 $\mathbf{H}$ をホモグラフィーという. $\mathbf{H}$ の自由度は 4 であり, 同一平面上の 4 個以上の $\mathbf{m_1}, \mathbf{m_2}$ の組から計算できる.

ホモグラフィー $\mathbf{H}$ は, 2 台のカメラ間の回転・並進を表す回転行列 $\mathbf{R}$ 及び並進ベクトル $\mathbf{T}$ を用いて

$$\mathbf{H} = \mathbf{A_1}[\mathbf{R} + \mathbf{T}\mathbf{n}^\top]\mathbf{A_2^{-1}} \tag{2}$$

のように表現することができる. ここで $\mathbf{n}^\top$ は $\mathbf{m_1}$ または $\mathbf{m_2}$ によって決まるベクトルである. また, $\mathbf{A_1}, \mathbf{A_2}$ は, それぞれのカメラについて

$$\mathbf{A} = \begin{pmatrix} k_u & s & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

の形で表現される行列である. ここで $k_u, k_v, s$ はそれぞれ 1 画素の $u$ 方向の大きさの逆数, 1 画素の $v$ 方向の大きさの逆数, スキューであり, $u_0, v_0$ は光軸と画像面の交点の座標である.

カメラにピンホールカメラモデルが適用できる場合には, $\mathbf{A_1}, \mathbf{A_2}$ が既知であれば, これらの行列で補正した座標系において $\mathbf{R}$ と $\mathbf{T}$(および $\mathbf{n}^\top$) のみからなる $\mathbf{H}$ を求めることができる.

さて, 実際にステレオ視で対象を観測する場合には, それぞれのカメラにピンホールカメラモデルを適用できるように, レンズによる幾何光学的な歪みを補正しなければならない.

レンズの幾何光学的歪みのうちで最も典型的なものにラディアルディストーションがある. ラディアルディストーションはレンズの光軸を中心とする. 光線の不均一な屈折によって生じる歪みである. この歪みはほとんどの場合レンズの光軸に対し等方に現れるケースを想定し. 画像面上の二次元的な歪みとして扱われる. ラディアルディストーションは本来ピンホールカメラでは座標 $(x, y)$ に投影されるべき点が座標 $(x_d, y_d)$ に移動するとして

$$x_d = x + \kappa_1(x - x_{dc})^2 + \kappa_2(x - x_{dc})^4 + \cdots (4)$$
$$y_d = y + \kappa_1(y - y_{dc})^2 + \kappa_2(y - y_{dc})^4 + \cdots (5)$$

のようにモデル化される. ただし $(x_{dc}, y_{dc})$ は歪み中心の座標である. このモデル式中の $\kappa_i (i = 1, 2, \cdots)$ を i 次のレンズ歪み係数と言う. このレンズ歪み係数を求めることにより. 点対称なラディアルディストーションを補正できる.

光軸に対する撮像面の傾きが小さい時, ラディアルディストーションはほぼ点対称となる. 従って, そのような 2 台のカメラについて, $\kappa_i (i = 1, 2, \cdots)$ によって補正した座標系で, 1 つの平面に対して 4 組以上の対応点データを得ることができれば, ホモグラフィーを求めることができる.

## 3.2 プロジェク・カメラシステムのホモグラフィー

プロジェクタは, 投影モデルとしてはカメラと同様に考えることができる. したがって投影モデルを考える時, プロジェクタ・カメラシステムは内部パラメータの異なる 2 台のカメラとみなすことができる. ここではプロジェクタ・カメラシステムのホモグラフィーを通常のステレオカメラと同様に求める方法について述べる.

ホモグラフィーを求める際に注意すべきステレオカメラとプロジェクタ・カメラシステムの相違点として, 左右のデバイスの内部パラメータの差が挙げられる.

ステレオカメラの場合は, 左右のカメラにどちらも同じ型のカメラを用いるのが一般的であり, 本来ホモグラフィーは左右のカメラの内部パラメータの投影中心と撮像面の距離や解像度, それらの基準となる画素の大きさがほぼ等しいような場合についての写像である. これに対し, プロジェクタ・カメラシステムでは左右のデバイスが異なるので, それらのパラメータが大きく異なることに注意しなければならない.

以下に内部パラメータのそれぞれ異なるステレオカメラのホモグラフィーを定義する座標系とホモグラフィーの計算法について述べる.

## 3.3 ホモグラフィーの求め方

以下では, カメラは内部パラメータについてキャリブレーションされているものとする.

2 台のカメラをカメラ 1, カメラ 2 とする. ワールド座標系で $(X, Y, Z)$ と表される点は $3 \times 4$ 行列 $\mathbf{P_1'}$ を用いて, 次のようにカメラ 1 のカメラ座標系 $(x_1, y_1, z_1)$ に変換できる.

$$\alpha_1 \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} & \mathbf{P_1'} & \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{6}$$

ただし $\alpha_1$ はワールド座標系におけるカメラ 1 の 1 画素の大きさである.

点 $(x_1, y_1, z_1)$ を, 平面 $Z = f_1$ 上に原点を中心に中心投影した結果, 点 $(u_1, v_1, f_1)$ に写ったとする. ただし, $f_1$ はカメラ 1 の焦点距離である. その時, 次の式が成り立つ.

$$\begin{pmatrix} u_1 \\ v_1 \\ f_1 \end{pmatrix} = \frac{f_1}{z_1} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \tag{7}$$

今, ワールド座標系で平面 $Z = 0$ 上の点について考える. この時式 (6),(7) より

$$\begin{pmatrix} u_1 \\ v_1 \\ f_1 \end{pmatrix} = \frac{f_1}{z_1\alpha_1} \begin{pmatrix} & \mathbf{P_1} & \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{8}$$

が成り立つ. ただし, $\mathbf{P_1}$ は $\mathbf{P_1}'$ の第 3 列目の成分を除いた $3 \times 3$ 行列である. カメラ 2 についてカメラ 1 と同様に $\alpha_2, z_2, f_2, u_2, v_2$ を定義すると, カメラ 2 についても式 (8) と同様の式が成り立つので 2 台のカメラに関する式 (8) をまとめると次のようになる.

$$\frac{f_2 z_1 \alpha_1}{f_1 z_2 \alpha_2} \begin{pmatrix} u_1 \\ v_1 \\ f_1 \end{pmatrix} = \mathbf{P}_1 \mathbf{P}_2^{-1} \begin{pmatrix} u_2 \\ v_2 \\ f_2 \end{pmatrix} \tag{9}$$

この場合, 左辺の $f, z, \alpha$ からなる係数が式 (1) の $\lambda$ に相当する. 従ってそれぞれのカメラの画像座標, 焦点距離から計算できるホモグラフィー $\mathbf{H}$ は,

$$\mathbf{H} = \mathbf{P}_1 \mathbf{P}_2^{-1} \tag{10}$$

であり, 焦点距離や画素の大きさがほぼ等しい通常のステレオカメラのホモグラフィーと同様に表現できることがわかる.

しかし, ホモグラフィーはもともと平面上の点を一台のカメラが視点を変えて観測した時の画像座標を対応づける変換である. そのため, 焦点距離の大きく異なる 2 つのカメラの間のホモグラフィーを計算する手法はあまり一般的ではない. そこで焦点距離の等しいステレオカメラと同様にホモグラフィーを計算するために, 画像座標をそれぞれ焦点距離で正規化する. この時, 式 (9) は

$$\frac{z_1 \alpha_1}{z_2 \alpha_2} \begin{pmatrix} u_1/f_1 \\ v_1/f_1 \\ 1 \end{pmatrix} = \mathbf{P}_1 \mathbf{P}_2^{-1} \begin{pmatrix} u_2/f_2 \\ v_2/f_2 \\ 1 \end{pmatrix} \tag{11}$$

のようになり, この座標系を用いれば両カメラとも焦点距離を 1 と考えることによって, 焦点距離の等しいステレオカメラと同じようにホモグラフィーを計算できる.

## 3.4 ホモグラフィーの計算

同一平面上の点に関する対応点データは取り終えたものとして, それらのデータからホモグラフィーを計算する方法を簡単に説明する.

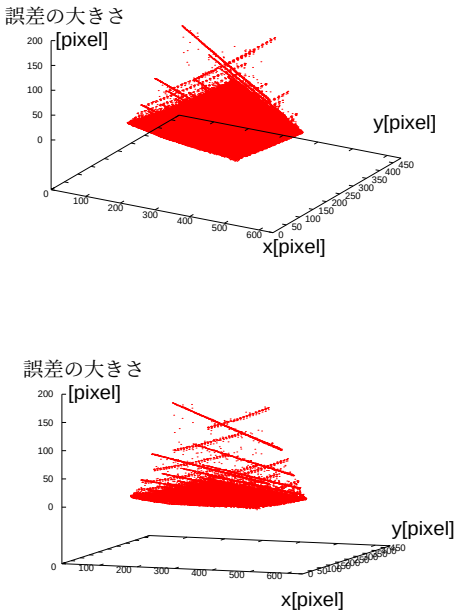



図 3: カメラ画像誤差の大きさの分布

得られた対応点データが全部で 4 組である場合は式 (6) についてその 4 組の座標値をそれぞれ代入し, 連立方程式を解くことで行列 $\mathbf{H}$ を計算する.

本研究のように 5 組以上の対応点データからホモグラフィーを計算する場合には, 式 (1) において $\sum_{k=1}^{N} ||\lambda \mathbf{m_1} - \mathbf{H m_2}||^2$ を最も小さくする最小 2 乗法を用いて計算する. ただし, ここで $N$ はカメラ画像上でプロジェクタの出力画像が占める領域内に含まれるカメラの画素の数である. 従って対応点データの数が多ければ多いほど, 正確に画像面同士を関係づけるホモグラフィーを計算できる[1].

# 4 実験

## 使用器具及びそのセッティング

空間コーディングの結果をホモグラフィーによって検証する実験を行った. 実感に用いた器具は以下の通りである.

**カメラ** SONY DFW VL500

**プロジェクタ** 東芝 LCD DATA PROJECTOR TLP771

[1]詳しい計算については [2] を参照

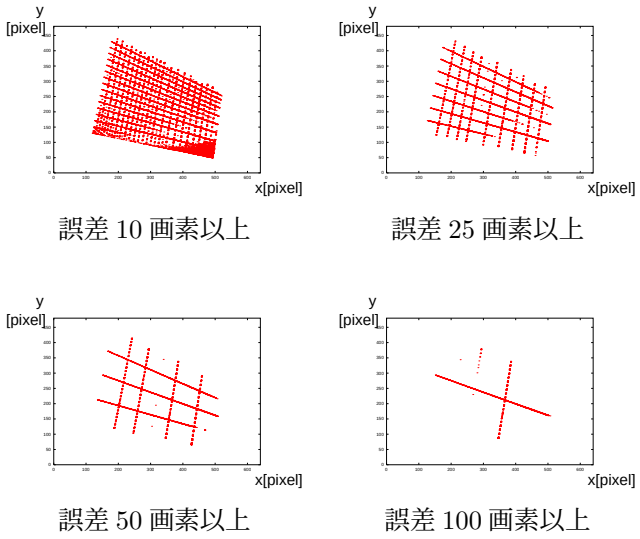



図 4: カメラ画像誤差の大きいデータの分布

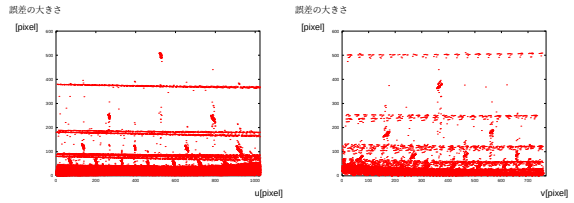
上記のカメラの画像の解像度は $640 \times 480$, プロジェクタの画像の解像度は $1024 \times 768$ である. スクリーンは通常の OHP 用のものを用いた. スクリーンを平面と仮定し, プロジェクタの出力する画像の全領域がスクリーン上に映し出されるようプロジェクタとスクリーンを設置した. その後, スクリーン上のプロジェクタの画像領域が視野内に全て含まれるようにカメラを設置した. このような状況でカメラとプロジェクタのズームとフォーカスを調整し, 固定した後にそれぞれの内部パラメータ (レンズ歪みも含む) をキャリブレーションし, 空間コーディングを行い, その結果を用いてホモグラフィーを計算した.

プロジェクタの内部パラメータは Z. Zhang の方法 [1] によって求めた. また, カメラに関しては文献 [5] の方法でキャリブレーションを行った[2]. 空間コーディングでは, 縦横軸の座標値に二進コードを割り当てるような入力画像を使用した.

## 計算結果の評価

空間コーディングの結果が正確であるならば, その結果とそれから計算できるホモグラフィーによる対応づけが完全に一致するはずである. 空間コーディングの精度は, その結果がホモグラフィーによ

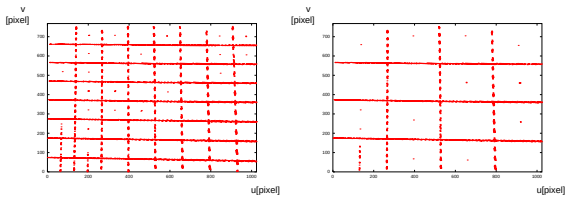
[2][5] を参照




(a) $u$ 座標に対する誤差の大きさ

(b) $v$ 座標に対する誤差の大きさ

図 5: プロジェクタ画像誤差の分布




(a) 誤差 64 画素以上

(b) 誤差 128 画素以上

図 6: プロジェクタ画像誤差の大きいデータの分布

る対応づけといかによく一致するか, で評価できる. 今回の実験ではカメラの全ての画素に対して対応点データをとったので, 計算で求めたホモグラフィーによるカメラ座標値の変換が, その計算の基となった対応点データとどの程度一致するかによって計算結果を評価できる. そのために, 対応点データ中のプロジェクタ座標 $(u, v)$ をホモグラフィー $\mathbf{H}$ でカメラ座標 $(x', y')$ に変換する. 対応点データ中で プロジェクタ座標 $(u, v)$ に対応するカメラ座標 $(x, y)$ に対する $(x', y')$ のずれを「カメラ画像誤差」と定義し, このカメラ画像誤差を用いて計算結果を評価する.

全対応点データについてカメラ画像誤差を求め, 誤差ベクトルの平均とその大きさの平均を計算した. その結果, 誤差ベクトル $(x' - x, y' - y)^\top$ の平均は pixel 単位で (-0.009692,-0.003822) であった. また, 誤差ベクトルの大きさの平均は 7.05pixel であった. カメラ画像上の座標値と誤差ベクトルの大きさの関係を調べたところ, 図 3, 図 4 のような分布が得られた. この図によると, 平均値に比べはるかに大きい誤差ベクトルをもつデータが複数存在する.

このような誤差の起こる原因について, 次節で考察する.

## 4.1 誤差の分布に対する考察

### デコードミス

空間コーディングではコーディングをした後デコードを行うが, コーディングの段階で図 4 にあるような大きな誤差が生じることは実験環境の安定性から考えてあり得ない. もし空間コーディングの過程で誤差が生じているとするならば, それはデコードミスによるものである.

デコードミスとは輝度値の判定ミスである. この判定ミスは, ホモグラフィーの計算過程においてプロジェクタ画像上でのずれとなって現れる. 対応点データ毎にプロジェクタ画像上のずれの大きさを評価するために, 前章で求めたホモグラフィーが理想値であると仮定する. このホモグラフィーを用いてカメラ座標 $(x, y)$ を変換したプロジェクタ座標を $(u', v')$ とする. $(x, y)$ の対応点データ $(u, v)$ の $(u', v')$ に対するずれをプロジェクタ画像誤差と定義する. この誤差は, デコードミスが上位のコーディング画像で起こるほど大きくなる.

各対応点データについてプロジェクタ画像誤差を計算し, プロジェクタ画像上の位置とその誤差ベクトルの関係を図 5 に, その中で特に誤差の値の大きいデータの分布を図 6 に示した. この図より, コーディング画像の境界線付近で誤差が特に大きく, 上位ビットの境界線付近ほど誤差が大きくなっていることが確認できる. この結果から, コーディング画像の縞の境界線付近でデコードミスが起きていると推測できる.

### グレイコードによる再実験

縞の境界線付近で生じるデコードミスに対し頑健なコーディングパターンとして, グレイコードが挙げられる. グレイコードでは, 隣り合う数とのハミング距離が常に 1 に保たれる. 従って縞の境界で起きたデコードミスは, プロジェクタの 1 画素分のずれとしてしか影響しない. また図 7 からわかるように, 通常の二進コードでは同じ部分に何度も縞の境界が現れるが, グレイコードではあるビットの画像で一度境界の現れた部分に他のビットの画像で再び

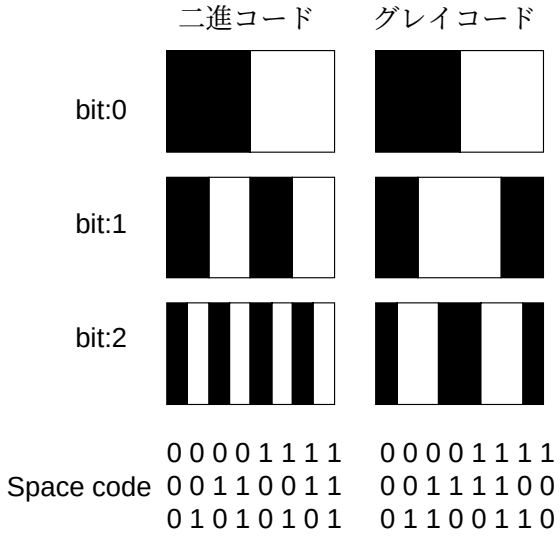



図 7: グレイコードと二進コードの比較

境界が現れることはない. つまりデコードミスによるビットの反転は, 各画素について $u, v$ 方向にそれぞれ最大でも 1 ビットしか起こり得ない.

これらのことから, 理論的にはグレイコードのデコード結果にプロジェクタ画像誤差が 2 画素を越えるようなデータは含まれない. 空間解像度はプロジェクタの方がカメラよりも高いので, カメラ画像誤差は最大でも 1 画素であると期待できる.

実際にグレイコードを用いて対応づけを行い, その結果からホモグラフィーを計算したところ, そのカメラ画像誤差は最大でも 5 画素以下であった. グレイコードによるコーディングによってカメラ画像誤差が大幅に小さくなったことから, 縞の境界付近でデコードミスが起きていることが確認できた. しかし一方でグレイコードでは生じないはずの, プロジェクタ画像誤差が 2 画素を越えるような対応づけが多数存在した. その中で特に誤差の値の大きなデータの分布は図 8 のようになった.

このような誤差の原因として, スクリーンが完全な平面でないことや各デバイスの内部パラメータの測定誤差である. しかしそれらのような原因によって図 8 のように格子上の領域に誤差をもつデータが分布することは考えにくい. この格子は, その方向がコーディング画像の縞の境界と一致する. この分布の原因がデコードミスであるとすると, 図 8 の格子上の領域では複数ビットでデコードミスが生じていると考えられる.

そこであるコーディング画像について縞の境界を

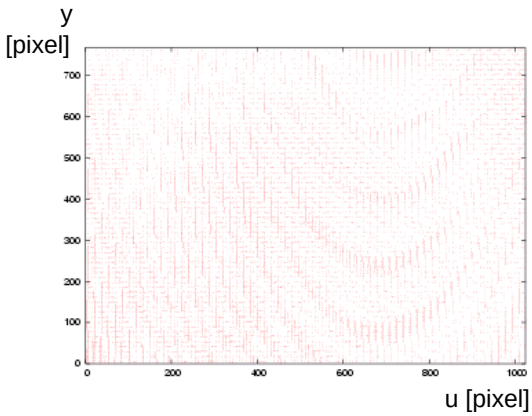



図 8: プロジェクタ画像誤差が 5 画素を越える点の分布

撮影した部分を拡大して見た所, 図 9 のようになった. カメラの画素のうち, 縞の境界が通過する画素の付近の画素についても輝度値に影響が出ていることがわかる. このことは撮影した画像がぼけていることを意味する. このぼけのために, 下位ビットに対応する縞の細かいコーディング画像のデコードにミスが生じ, 縞の境界とぼけの両方のデコードミスによって複数のビットが反転するような画素が存在すると考えられる.

プロジェクタ, カメラのピント調整を十分行って撮影したにも関わらず, このような画像のぼけが撮影画像の全域に渡って存在した. このぼけの原因は, 画素毎にプロジェクタやカメラからの距離が異なることであると思われる. その他の原因として, カメラのピントをできる限り合わせた状態であってもレンズの集光の性能が十分でないために撮影画像にぼけが生じることも考えられる. これらを原因とするぼけは通常の実験環境では一般的に起きる.

実験の結果, プロジェクタ・カメラシステムの画素の対応づけに空間コーディングを持ちいた場合には, 上記のような原因で一般的に起きる画像のぼけによって, 誤差が含まれるを確認した.

## 5 結論

平面のスクリーンに対し, 空間コーディングを用いてプロジェクタ・カメラシステムの両方の画像の対応関係を求めたところ, その結果には大きな誤差を含む対応点データが含まれていることがわかった.

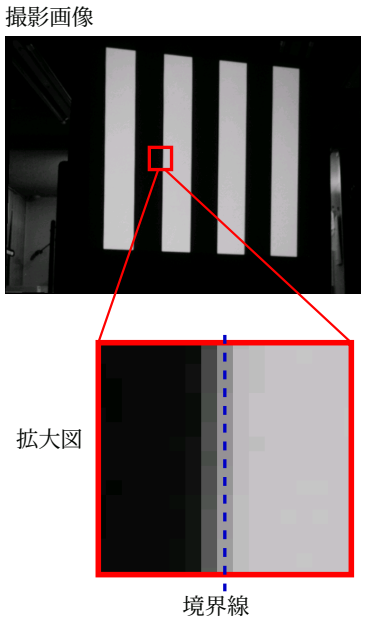



図 9: 境界線上の画素の拡大図

また, その誤差の原因がコーディング画像の縞の境界線上で生じるデコードミスであることもわかった.

この誤差に対する解決法としてコーディングパターンにグレイコードを用いたところ, デコードミスによる誤差の影響は小さくなったが, それでも誤差の大きさが予想を越えるようなデータが多数存在した. その誤差は, 撮影した画像のぼけによるデコードミスによるものと思われる.

カメラを観測者の位置に置いたプロジェクタ・カメラシステムは投影対象が平面でない場合にも応用できるべきであるが, その場合には画像のぼけによる影響が大きい. なぜなら, その場合にはホモグラフィーのような写像で画素同士を対応づけることはできないため、ぼけによるデコードミスの影響が投影する画像に直接現れるからである。今後の課題は、ぼけによるデコードミスに対する解決法を提案し, 提供すべき画像をより精度の高い状態で観測者に見せることである.

なお, 本研究は科学研究費補助金特定領域研究 13224051 の補助を受けて行った.

## 参考文献

[1] Z. Zhang. *A Flexible New Technique for Camera Calibration.* http://research.microsoft.com/zhang/Calib/.

[2] 金谷健一. 画像理解 3 次元認識の数理. 森北出版, 1990.

[3] 服部数幸, 佐藤幸男. レーザーパターンシフトによる高精度空間コード化法. 画像の認識・理解シンポジウム (MIRU'96), 1996.

[4] 井口征士, 佐藤宏介. 三次元画像計測. 昭晃堂, 1990.

[5] 物部祐亮. 視点固定型パン・チルト・ズームカメラを用いた実時間対象検出・追跡. Master's thesis, 京都大学, 1997.