

グレイコードパターンを利用したレンズ歪みの補正手法

高辻 誠也[†] 佐川 立昌[†] 越後 富夫[†] 八木 康史[†]

† 大阪大学 産業科学研究所

〒 567-0047 大阪府茨木市美穂ヶ丘 8-1

E-mail: †{takatsuji,sagawa,echigo,yagi}@am.sanken.osaka-u.ac.jp

あらまし 本論文では、レンズ歪みをグレイコードパターンを利用して補正する方法を提案する。具体的には、LCD のような平面ディスプレイ上にグレイコードパターンを表示させて、ディスプレイ座標系とカメラ画像座標系との対応関係を表す変換マップを作成する。本手法には、第一に、キャリブレーションでよく用いられているチェックカーボードなどのマーカーを利用する方法よりも簡単に画像の周辺部でも対応がとれるという利点があり、第二に、ディスプレイと画像との対応関係が密に得られるので、単純な線形補間を用いるだけで元の画像から歪みを補正した画像への変換マップを作成することができるという利点がある。また、変換マップを直接作成して歪みを補正するので、歪みモデルによる制約を受けないという利点も持つ。本手法の精度を評価するために、歪みパラメータを推定する方法と比較して誤差が減っていることを確認した。

キーワード 歪み補正、構造化光、グレイコード、ノンパラメトリック

Calibration of Lens Distortion by Structured-Light Scanning

Masaya TAKATSUJI[†], Ryusuke SAGAWA[†], Tomio ECHIGO[†], and Yasushi YAGI[†]

† The Institute of Scientific and Industrial Research, Osaka University

8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, JAPAN

E-mail: †{takatsuji,sagawa,echigo,yagi}@am.sanken.osaka-u.ac.jp

Abstract This paper describes a new method to automatically calibrate the lens distortion of wide-angle lenses. We project structured-light patterns by using a flat display to generate a map between the display coordinate system and the image coordinate system. This approach has two advantages. First, it is easier to take correspondences of image and marker (display) coordinates around the edge of a camera image than using a usual marker, e.g. a checker board. Second, since we can easily construct a dense map, a simple linear interpolation is enough to create an undistorted image. Our method is not restricted by the distortion parameters because it directly generates the map. We have evaluated the accuracy of our method and the error becomes smaller than a result by parameter fitting.

Key words Calibration of lens distortion, structured-light scanning, non-parametric

1. はじめに

ロボットのナビゲーションシステムなどでは、周囲の環境を観察するために広い視野が必要となる。広い視野を得るには様々な方法があり、例えば広い画角を持ったレンズを用いる方法[1]、全方位ミラーを利用する方法[2], [3]、カメラを移動させる方法[4]、複数のカメラを使用する方法などがある。

魚眼レンズのような画角が大きいレンズを用いると簡単に広い視野を得ることができるが、一般的に、レンズの画角が大きくなればなるほど、レンズを通して得られる画像は大きく歪んでしまう。画像処理で扱うことを考えると、そのような歪みは望

ましくない。歪みを取り除くことができれば、投影モデルをピンホールカメラモデルとして扱うことができるようになり、画像の処理が簡単になるので歪みの補正是重要である。

歪みを取り除く方法は数多く提案されている。多くの手法では歪みモデルに樽型歪み (*radial distortion*) と円周方向への歪み (*tangential distortion*) を含んだものを用いている。円周方向への歪みは *decentering distortion* [5], [6] とも呼ばれている。ここで (x_d, y_d) を歪みを持った画像中の一点とし、 (x_u, y_u) を歪み補正後の画像中で対応する点とする。 (x_u, y_u) は

$$x_u = (1 + \sum_{i=1}^{\infty} K_i r^{2i}) x_d + \quad (1)$$

$$(2P_1\bar{x}\bar{y} + P_2(r^2 + 2\bar{x}^2))(1 + \sum_{i=1}^{\infty} P_{i+2}r^{2i})$$

$$y_u = (1 + \sum_{i=1}^{\infty} K_i r^{2i}) y_d + \\ (P_1(r^2 + 2\bar{y}^2) + 2P_2\bar{x}\bar{y})(1 + \sum_{i=1}^{\infty} P_{i+2}r^{2i})$$

と表現される。ここで $\bar{x} = x_d - C_x$, $\bar{y} = y_d - C_y$, $r^2 = \bar{y}^2 + \bar{x}^2$, C_x と C_y は光学的中心, K_i と P_i はそれぞれ樽型歪み, 円周方向への歪みのパラメータである。実際の計算では、式(1)の右辺において次元の高い項の影響は無視し、以下のように近似される[1], [6]:

$$x_u = (1 + K_1 r^2 + K_2 r^4) x_d + (2P_1\bar{x}\bar{y} + P_2(r^2 + 2\bar{x}^2)) \quad (2)$$

$$y_u = (1 + K_1 r^2 + K_2 r^4) y_d + (P_1(r^2 + 2\bar{y}^2) + 2P_2\bar{x}\bar{y})$$

歪みパラメータを推定する方法には、キャリブレーションマークを用いる方法[7]~[9]や、それとは対照的に、明示的なマークを用いない方法[1], [10]~[12]がある。後者の方法では、対象シーン中の直線とそれに対応する画像中の曲線に注目して、曲線を本来投影されるべき直線にフィッティングさせて歪みパラメータを推定する。

上記のような手法では歪みモデルを仮定してパラメータの推定を行うが、仮定する歪みモデルを適切に選ばなければ、得られるパラメータは信頼できない値になってしまい。例えば、歪みモデルに樽型歪みだけを仮定してパラメータの推定を行うと、もし画像に円周方向の歪みが存在しても得られたパラメータでは円周方向への歪みを取り除くことはできない。また、式(2)の全ての変数を考慮してパラメータの推定を行ったとしても、光学的中心から離れば離れるほど誤差は大きくなってしまう、という問題が残る。

一方で、パラメータを用いない方法もいくつか提案されている。Greenら[13]は画像上の格子パターンに局所アフィン変換を用いて、実際の格子パターンへと当てはめることによって歪みを取り除く方法を提案した。この方法では変換は線形であるのに対し、格子パターンは離散的であるために滑らかに歪みを取り除くことができなかった。Goshtasby[14]はBezierパッチを用いて歪みパラメータをモデル化した。変形は滑らかになったが、扱う頂点が少なかったので依然として誤差は残った。YingとHu[15]はマークがついた半球を利用するパラメータを用いない歪み補正方法を提案した。

本論文では、グレイコードパターンを用いてレンズ歪みを補正する新しい方法を提案する。提案する手法では、歪みを持った画像から歪みを除去した画像への対応関係を全てのピクセルについて求めるので、パラメータを計算することなく補正された画像を作ることができ、得られる画像は透視投影で得られる画像と同じように扱うことができる。本手法では、全てのピクセルで対応関係を求めるために密なマークを撮影して対応関

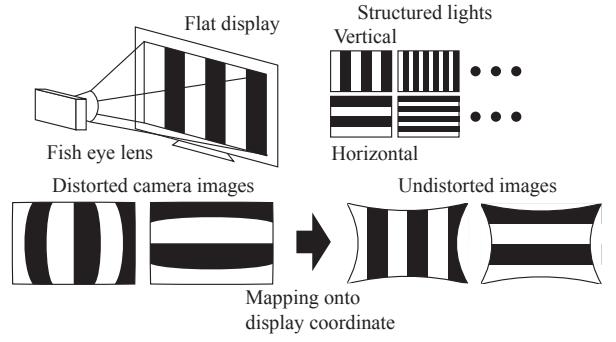


図1 構造化光投影による画像のコード化を利用した歪み補正

係を調べるが、密なマークを作成するために液晶ディスプレイ(LCD)やプラズマディスプレイ(PDP)のような平面ディスプレイを用いる。LCDやPDPは高い解像度を持ち、映像を表示する部分が平面であるため、適切な画像を表示させることによって密なマークを作成することができる。マークとしてチェックボードなどを用いる方法ではレンズ歪みが激しい場合、画像の周辺部で画像とマークの対応関係を調べるのは困難である。しかし本手法では、画像の周辺部においても簡単に画像とマークの対応をとることができる。また、非常に多くの対応関係を密にとることができるので、単純な線形補完を用いるだけで滑らかな補正画像を作成することができる。

以下、第2.節で平面ディスプレイを用いて密なマークを作成する方法と、そのマークを用いた歪み補正の方法を説明する。また、第3.節で実験結果を述べ、最後にまとめを第4.節で述べる。

2. グレイコードパターンを用いた歪み補正

本手法では、歪みパラメータを推定するのではなく、歪みを持った画像から歪みを取り除いた画像への対応関係を表すマップを作成する。密なマップを作成するためには、多くのマークが必要になる。また、用いるマークは簡単に対応点を識別できるものが望ましい。本手法では、密なマークを実現するために構造化光を利用する。構造化光はレンジファインダの光切断法[16]~[18]でよく用いられる技術である。

本手法では構造化光にグレイコード[19]を用いる。時系列的にグレイコードの各ビットで構成されたパターン光を表示し、その組み合わせで対象を領域に分割する。1枚のパターン光において、対応するグレイコードのビットが1となる領域では光がオン、0となる部分ではオフとなっていて、画像中のある点に注目すると、その点はパターン光が変わることにオン、オフと変化する。オンを1、オフを0として符号化すると、どの点がどこに対応するのかを一意に知ることができる。対応点を求めるには、各パターン光を撮影した画像においてオンとオフの部分を判別できればいいので、画像を2値化するだけでよい。1枚のパターン光で1ビットを表し、n枚のパターン光で 2^n の領域に分割できるので、少ないパターン光で多くの領域に分割することができる。

バイナリコードを用いてパターン光を構成することもできる

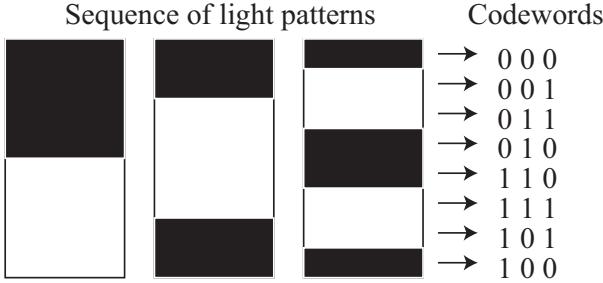


図 2 グレイコードによるコード化

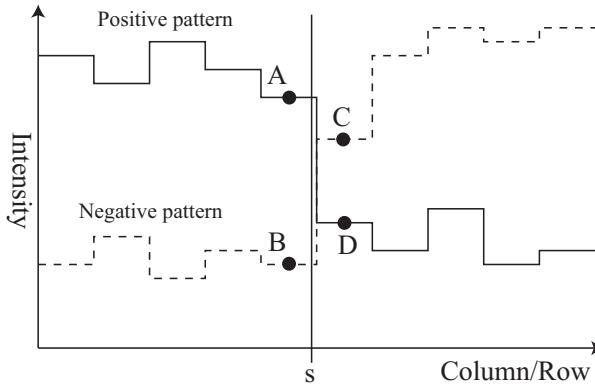


図 3 境界を相補パターン投影を利用してサブピクセルで判定する

が、バイナリコードでは隣接値でビットが反転することがあり、場合によっては 1 ビットの誤りで値が大きくかわってしまい符号化誤りの影響を大きく受けてしまう。グレイコードは隣接値のハミング距離が 1 となっているので、境界上での符号化誤りを最小化できる。

レンジファインダで構造化光を用いる場合は、パターン光を 3 次元計測の対象に投影してそれをカメラで観察するが、本手法では構造化光を補正対象のカメラで直接撮影して画像をコード化する。

図 1 に提案手法の概要を示す。本研究では LCD のような平面ディスプレイにパターン光を表示させて、その様子を補正対象のカメラで撮影する。カメラの歪み補正のためには、ディスプレイ座標の縦軸と横軸にそれぞれ平行な二通りのグレイコードパターンを用いる。カメラで撮影して得られる画像上の全てのピクセルは時分割でコード化され、ディスプレイ座標のどの点が写っているのかという対応関係を計算して得ることができる。画像上の全ての点を対応するディスプレイ座標へ投影することによって、歪みを取り除いた画像を作成することができる。

2.1 画像のコード化

本手法では、時系列的にグレイコードの各ビットで構成されるパターン光をディスプレイに表示して、その様子をカメラで撮影していく。図 2 にグレイコードによるコード化を示す。グレイコードパターンを撮影した画像をデコードするには、ディスプレイ座標の縦軸と横軸に沿って計算する。そのために、それぞれ縦軸と横軸に平行で互いに垂直になっている二通りのグレイコードパターンを撮影して、ディスプレイ座標から画像座標へのマッピング関数 f を得る。ディスプレイ座標上の点 (i, j)

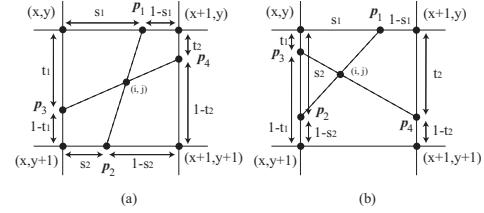


図 4 ディスプレイ座標から画像座標への対応を求める計算方法

はマッピング関数 f によって画像上の点 (x, y) へと変換される。

$$(x, y) = f(i, j) \quad (3)$$

マッピング関数 f は以下のような手順で求めることができる。

(1) パターン光を撮影した画像中でオンとオフの境界をサブピクセルの精度で求める

(2) 互いに垂直な二通りのパターン光で境界が交差する点を求める

(3) 全てのディスプレイ座標 (i, j) について、対応する画像座標 (x, y) を求めるために補間を行う

2.1.1 サブピクセル精度の境界位置の計算

カメラで撮影されたパターン光のオンとオフの境界位置がディスプレイ座標のある i (または j) に対応するが、対応の精度を向上させるために境界が存在する位置をサブピクセル精度で求める。サブピクセル精度で境界を求めるにはそれぞれのパターン光で相補パターンを用いて図 3 のように考える。まずグレイコードの各パターンにおいてポジパターン (1 がオン, 0 がオフ) とネガパターン (1 がオフ, 0 がオン) を表示する。図 3 の s のように輝度値が交差する箇所で境界をサブピクセル精度で計算することができる。 s の計算方法は以下の通りである [17]:

$$s = \frac{A - B}{(A - B) + (C - D)} \quad (4)$$

ここで A, B, C, D は各ピクセルにおける輝度値である。このようにして 1 つの境界線につき画像座標の縦軸、横軸に沿った 2 方向で境界の位置を求める。

2.1.2 境界線の交点の計算

境界線をサブピクセル精度で求めた後は、境界線と境界線の交点を求める。画像座標中でそれぞれ垂直な境界線が交差する点 (x, y) がディスプレイ座標中の点 (i, j) に対応する。図 4 に交点の例を 2 通り示す。点線が境界線を表し、境界線の位置を画像座標中の縦軸、及び横軸に沿った方向にサブピクセル精度で計算しているので、交点は境界線の位置を用いて線形補間にによって計算することができる。

境界線を求めるときに s_1, s_2, t_1, t_2 の値は式 (4) を用いて求まっている。 s_1 と s_2 はディスプレイ座標では i 軸に沿った境界線、 t_1 と t_2 はディスプレイ座標で j 軸に沿った境界線までの距離である。図 4(a) では境界線の交点 $(x_{(i,j)}, y_{(i,j)}) = f(i, j)$ は以下の式を変数 a, b について解くことによって計算できる。

$$p_1 + a(p_2 - p_1) = p_3 + b(p_4 - p_3) \quad (5)$$

ここで p_1, p_2, p_3, p_4 は 2 つの境界線上の点である。図 4(b)

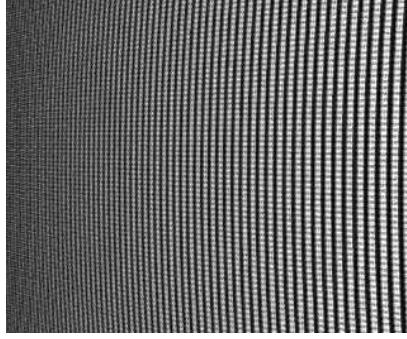


図 5 画像の左端では縞模様が細かくて認識できない

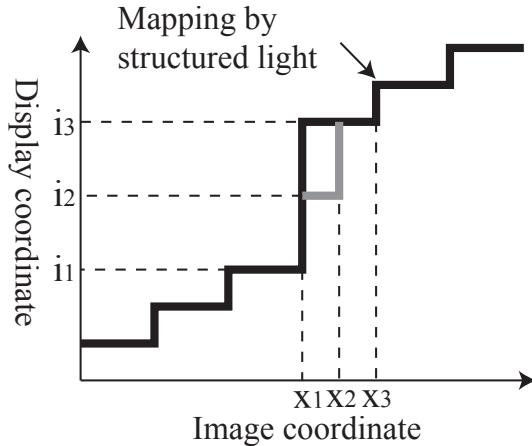


図 6 近傍の点を用いた補間

のように、境界線の交点は他にもいくつかの場合が考えられるが、どの場合も式 (5) を用いて同じようにして交点を求めることができる。

2.1.3 全てのディスプレイ座標点での対応点計算

図 5 のように、画像の周辺部などでは縞の幅が非常に狭くなることがある。このような部分では、縞の境界が識別できないので、 (i, j) に対応する画像中の点 (x, y) を求めることができない。よって全ての (i, j) について関数 $f(i, j)$ で対応点を計算することができなくなってしまう。この場合は近隣の点を用いて補間する。例えば図 6 では、 i_1 と i_3 に対応する x_1 と x_3 が求まっている、 i_2 に対応する点 x_2 は境界が識別できず求まっていないとする。この場合、 i_2 に対応する x_2 を x_1 と x_3 を用いて補間して求める。

2.2 座標系の正規化

歪みを補正した画像を作ることを考えると、マッピング関数 $f(i, j)$ を正規化する必要がある。理由としては 1) 画像とディスプレイの解像度が異なっている、2) カメラの光軸に対してディスプレイが正対していない可能性がある、からである。解像度が異なっていると補正前と後で画像に写っている物の大きさが変わってしまう。またディスプレイがカメラの光軸と垂直でなければ補正後の画像は視点が傾いたものになってしまう。これらの問題を解決するためにホモグラフィー行列を用いて、解像度が等しく光軸からのずれもない平面への変換を考える。ホモグラフィー行列は 3×3 行列で、正規化前と後の対応する点の組み合わせを 3 組以上用いることによって求めることができる。

補正前の画像には歪みが存在するが、画像の中央付近では歪みは無視できるほど小さいと考えられるので、画像中央付近の点とそれに対応するディスプレイ座標上の点の組を 3 組以上用いてホモグラフィー行列を計算できる。求めるホモグラフィー行列は以下の関係式を満たす：

$$a \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H \begin{pmatrix} f_i^{-1}(x, y) \\ f_j^{-1}(x, y) \\ 1 \end{pmatrix} \quad (6)$$

ここで $(f_i^{-1}(x, y), f_j^{-1}(x, y)) = f^{-1}(x, y)$ であり、 a は各対応点によって定まる定数である。

また、次の新しいマッピング関数 $g(u, v)$ を定義する：

$$(x, y) = f(H^{-1}(u, v, 1)^T) = f(h(u, v)) = g(u, v) \quad (7)$$

ここで h 、 H^{-1} はそれぞれ $h(u, v) = (i, j)$ 、 $a(i, j, 1)^T = H^{-1}(u, v, 1)^T$ を満たす。この正規化によって、関数 $g(u, v)$ による補正後も画像の中央付近で解像度は変わらなくなる。

2.3 マップの補間

パターン光による対応付けでは、境界線と境界線の交点でディスプレイ座標上の一つの点 (i, j) との対応関係が得られる。よってマッピング関数 f は (i, j) がそれぞれ整数の場合でしか求まらない。しかし本手法では十分密に対応関係をとることができるので、近傍の点を用いて線形補間を行い、対応点を計算すれば十分である。 (u, v) を歪み補正後の画像上の点とすると、

$$(i + \Delta i, j + \Delta j) = h(u, v) \quad (8)$$

より、ディスプレイ座標中の対応する点を計算することができる。対応するディスプレイ座標上の点は、整数 i, j と小数部分の $\Delta i, \Delta j$ ($0 \leq \Delta i < 1, 0 \leq \Delta j < 1$) の和で表現できる。よって、 $(i + \Delta i, j + \Delta j)$ に対応する元の画像上の点 (x, y) は以下のように近隣の 4 点の値を用いて求めることができる。

$$\begin{aligned} (x, y) = & (1 - \Delta i)(1 - \Delta j)f^{-1}(i, j) \\ & + \Delta i(1 - \Delta j)f^{-1}(i + 1, j) \\ & + (1 - \Delta i)\Delta j f^{-1}(i, j + 1) \\ & + \Delta i\Delta j f^{-1}(i + 1, j + 1) \end{aligned} \quad (9)$$

式 (8)、及び式 (9) により、補正後の画像上の全ての点 (u, v) で関数 $g(u, v)$ を用いて対応する座標を計算することができる。

3. 実験

提案手法の精度を評価するために実験を行った。パターン光を投影するために液晶ディスプレイを用いる。用いた液晶ディスプレイは 20.1 インチで解像度は 1600×1200 画素、ピクセルサイズは 0.255 mm、視野角は $\pm 88^\circ$ である。グレイコードパターンはディスプレイ座標の縦軸、横軸に平行なものをそれぞれ 11 パターン用いる。縦軸、横軸方向に対してそれぞれ平行なグレイコードパターンを投影することにより、表現できる座標は最高で $2^{11} \times 2^{11}$ 点となる。

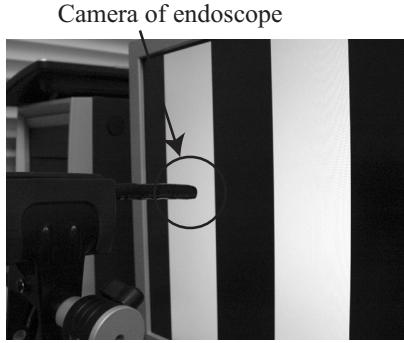


図 7 内視鏡カメラでディスプレイに表示した構造化光を撮影している様子

まず、内視鏡の先端に取り付けられたカメラで実験を行った。実験に用いた内視鏡カメラの画角は対角で 140° である。実験の様子は図 7 に示す。内視鏡カメラの解像度は 676×672 画素とディスプレイの解像度よりも低いので、グレイコードの下位ビットで構成されるパターン光は細かすぎて認識できない。よって 2.1 節で述べたような方法で対応点の補間を行う。図 8 に内視鏡画像の歪みを補正する手順を示す。補正後の画像に対応する元の画像の座標は $f(i, j)$ と $h(u, v)$ によって計算することができる。図 8 で左が歪みを持った入力画像、中央がディスプレイ座標上で対応する領域、右が歪みを取り除き正規化した画像である。最終的な画像サイズは 1736×1736 ピクセルになった。これらの画像から画像の周辺部でも対応関係が得られていることが分かる。

次に歪み補正の精度を評価するための実験を行った。シーン中の直線は歪み補正後の画像でも直線になっているはずなので、補正後の画像の一部を直線にフィッティングさせて誤差を計算する。提案手法の評価を行うために直線との誤差を、(a) 補正前の画像、(b) 提案する補正を行った後の画像、(c) パラメータ推定による歪み補正を行った後の画像の 3 種類で計算した。3 種類の画像における直線フィッティングの様子を図 9 に示す。図 9 で誤差の計算に使われている縞模様は歪み補正に用いたグレイコードパターンとは異なるものである。比較実験であるパラメータ推定による歪み補正には OpenCV [20] のキャリブレーションを行う関数を利用した。OpenCV の関数では歪みのモデルは式 (2) のように定義されている。図 9 の破線で示されている箇所で直線フィッティングを行って誤差の RMS を求めたところ、結果は (a) 16.6 ピクセル、(b) 0.83 ピクセル、(c) 1.18 ピクセルとなった。この実験からパラメータ推定による補正よりも提案手法のほうが精度がよいことが分かる。

次に Nikon D70 に広角レンズを装着して実験を行った。用いた広角レンズは AF DX Fisheye Nikkor ED 10.5mm F2.8G である。得られる画像サイズは 3008×2000 ピクセルで、画角は対角で 180° である。実験の様子とパターン光の撮影で得られた画像を図 10 に示す。カメラ-ディスプレイ間の距離は約 3 cm である。

図 11 に Nikon D70 で撮影した画像と歪みを補正した画像の例を示す。図 11(a) は図 10(b) の下の画像を補正して得られ

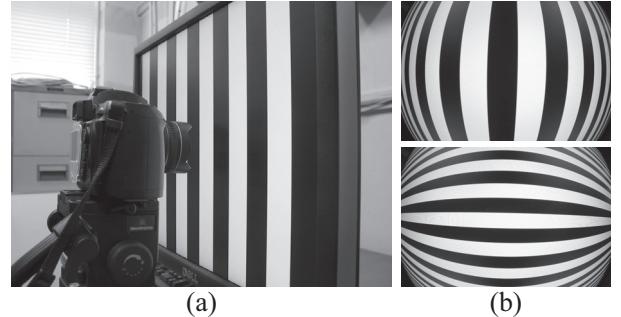


図 10 D70 を用いて実験: (a) D70 で構造化光を撮影、(b) 撮影で得られた画像

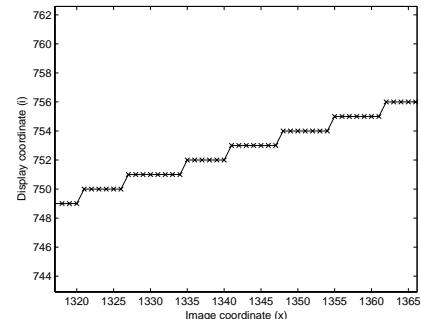
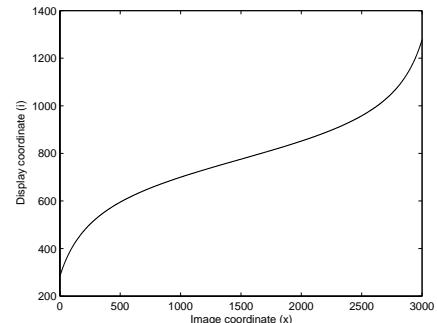


図 12 上: 構造化光投影によって得られた画像座標からディスプレイ座標へのマッピング関数 $i = f_i(x, y)$ の一部。下: 上を中心付近で拡大したもの。

た画像である。補正後の画像サイズは 5328×4288 ピクセルとなった。図 11(b) は Nikon D70 で撮影した画像で、図 11(c) はそれを補正した画像である。我々の提案する手法で歪みの補正に成功していることがわかる。

図 12 では $(i, j) = f^{-1}(x, y)$ で表される対応関係の一例を示している。ディスプレイの解像度よりも画像の解像度のほうが大きいので、画像の中心付近では、複数のピクセルにディスプレイ座標の一点が対応していて、階段状に値が変化していることが分かる。しかし式 (4) や式 (5) では値をサブピクセルの精度で計算しているため、図 13 に示すように、 $f(i, j)$ は滑らかな関数として求めることができる。

4. おわりに

本論文では広角レンズによって生じる歪みを補正する新しい方法を提案した。本手法では平面ディスプレイに構造化光を表

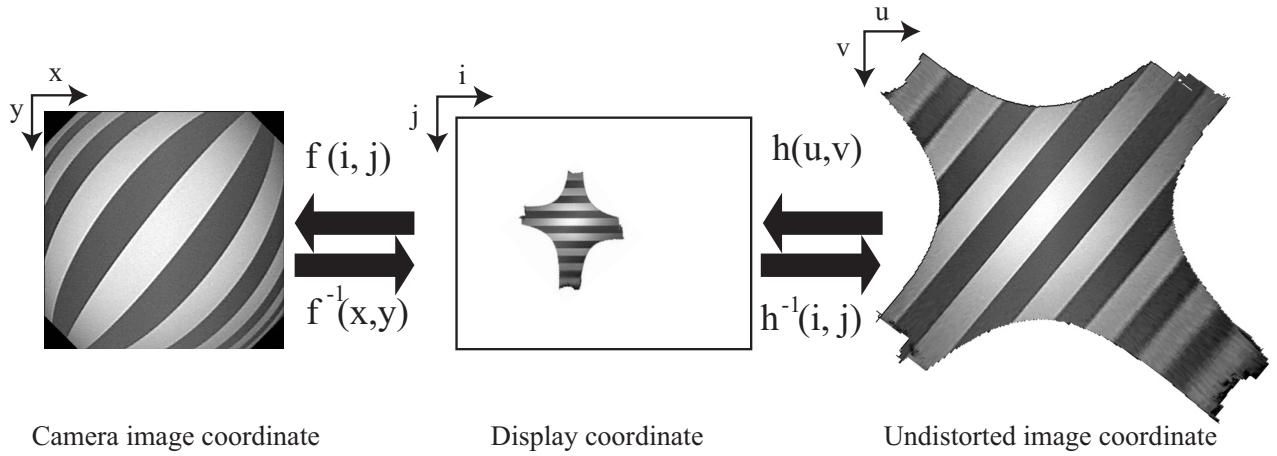


図 8 内視鏡で撮影した画像から歪み補正した画像を作成する手順と各段階で座標系の対応関係

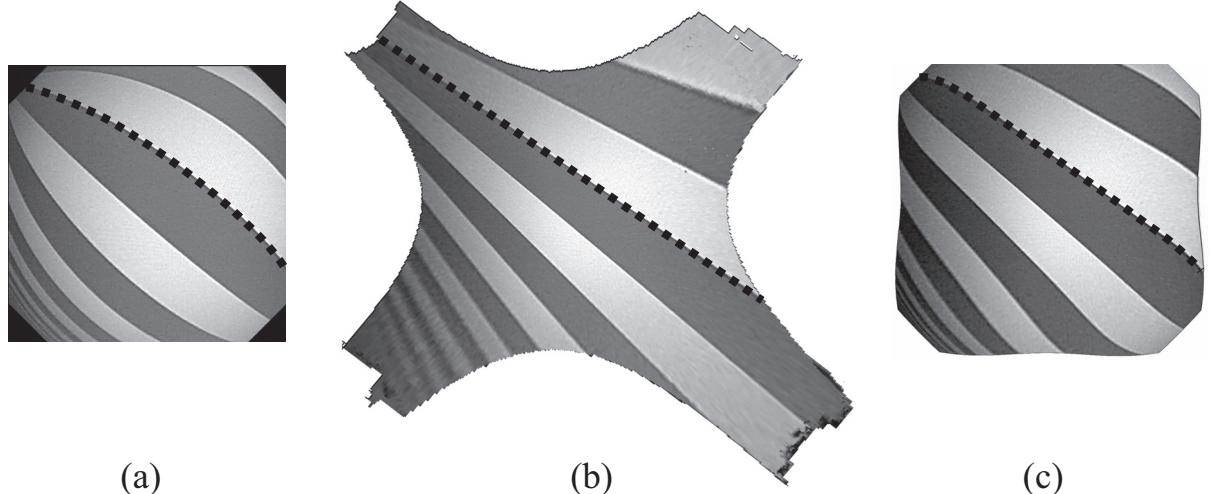


図 9 歪み補正を評価するための比較: (a) 内視鏡で撮影した画像, (b) 提案手法で補正した画像, (c) パラメータ推定を行って補正した画像

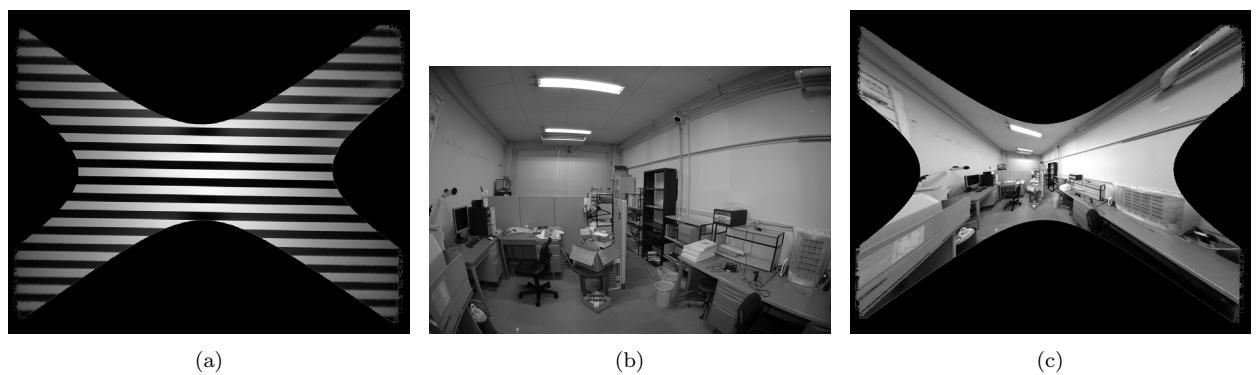


図 11 D70 で撮影した画像と補正した画像: (a) 提案手法で補正した画像(図 10(b)の下の画像を補正), (b) 補正前の風景画像, (c) 図 11(b)を提案手法で補正した画像

示し、ディスプレイ座標系と画像座標系の対応関係を表すマップを作成する。本手法には、カメラ画像の周辺部でも画像座標とディスプレイ座標の対応関係を求めやすい点、対応関係を表すマップを密に作成することができるので簡単な線形補間を用いるだけで歪みを補正した画像を作成することができる点という二つの利点がある。本手法は作成したマップを用いて直接歪

みを補正するので、パラメータによる拘束を受けないという利点も持っている。提案手法の精度を評価するために実験を行い、誤差がパラメータ推定による歪み補正よりも小さくなっていることを確認した。

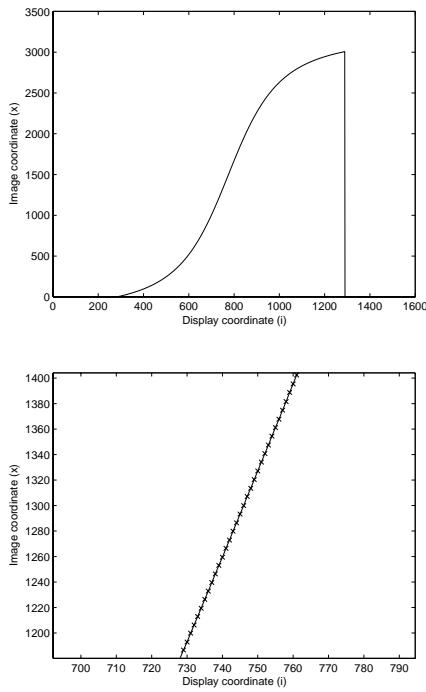


図 13 上: 補間された逆関数 $x = f_x(i, j)$ の一部 . 下: 上を中央付近で拡大したもの。

文 献

- [1] R. Swaminathan and S. Nayar, "Nonmetric calibration of wide-angle lenses and polycameras," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1172–1178, 2000.
- [2] K. Yamazawa, Y. Yagi, and M. Yachida, "Omnidirectional imaging with hyperboloidal projection," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, July 1993, pp. 1029–1034.
- [3] S. K. Nayar, "Catadioptric omnidirectional camera," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.
- [4] S. Peleg and M. Ben-Ezra, "Stereo panorama with a single camera," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Ft. Collins, Colorado, June 1999, pp. 395–401.
- [5] A. Conrady, "Decentering lens systems," *Monthly Notices of the Royal Astronomical Society*, vol. 79, pp. 384–390, 1919.
- [6] D. Brown, "Decentering distortion of lenses," *Photogrammetric Eng.*, vol. 32, no. 3, pp. 444–462, 1966.
- [7] R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.
- [8] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, 1992.
- [9] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [10] S. B. Kang, "Radial distortion snakes," in *IAPR Workshop on Machine Vision Applications*, Tokyo, 2000, pp. 603–606.
- [11] S. Becker and V. B. Bove, "Semiautomatic 3-d model extraction from uncalibrated 2-d camera views," in *SPIE Visual Data Exploration and Analysis II*, vol. 2410, Feb 1995, pp. 447–461.
- [12] G. Stein, "Lens distortion calibration using point correspondences," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 602–608.
- [13] W. Green, P. Jespen, J. Kreznar, R. Ruiz, A. Schwartz, and J. Seidman, "Removal of instrument signature from mariner 9 television images of mars," *Applied Optics*, vol. 14, no. 1, pp. 105–114, 1975.
- [14] A. Goshtasby, "Correction of image deformation from lens distortion using bezier patches," *Computer Vision, Graphics, and Image Processing*, vol. 47, no. 3, pp. 385–394, September 1989.
- [15] X. Ying and Z. Hu, "Distortion correction of fisheye lenses using non-parametric imaging model," in *Proc. Asian Conference on Computer Vision*, Jeju, Korea, 2004, pp. 527–532.
- [16] J. Posdamer and M. Altschuler, "Surface measurement by space-encoded projected beam systems," *Computer Graphics and Image Processing*, vol. 18, no. 1, pp. 1–17, 1982.
- [17] K. Sato and S. Inokuchi, "Range-imaging system utilizing nematic liquid crystal mask," in *Proc. International Conference on Computer Vision*, 1987, pp. 657–661.
- [18] J. Gühring, "Dense 3-d surface acquisition by structured light using off-the-shelf components," in *Videometrics and Optical Methods for 3D Shape Measurement*, vol. 4309, 2001, pp. 220–231.
- [19] F. Gray, "Pulse code communication," US Patent, March 17 1953, 2,632,058.
- [20] Open Source Computer Vision Library, <http://www.intel.com/research/mrl/research/opencv/>.