

# 距離測定カメラとビデオカメラを併用したステレオ動画生成の一手法

浅山 能久<sup>†</sup> 山口 達也<sup>†</sup> 東海 彰吾<sup>†</sup> 長谷 博行<sup>†</sup>

<sup>†</sup> 福井大学大学院工学研究科 〒910-8507 福井県福井市文京 3-9-1

E-mail: †{asayama,yamaguchi}@monju.fuis.u-fukui.ac.jp, ††{tokai,haseh}@u-fukui.ac.jp

**あらまし** 近年、3D映像の普及により、ステレオ映像生成の様々な手法が研究されている。本研究では、撮影物までの距離を測定できる距離測定カメラの情報を利用し、ステレオカメラではない、通常のビデオカメラを用いて撮影した映像からステレオ映像対を生成する手法を提案する。具体的には、まず、お互いの位置情報がわかっている距離測定カメラとビデオカメラを用いて撮影物のカラー3次元形状を復元する。そして、ステレオ映像対を生成するカメラの位置や方向を指定し、指定した位置、方向から復元したカラー3次元形状を見た映像を生成する。その後、その映像を右目用の映像とし、左目用の映像を右目用の映像の座標軸に対して左に数 cm ずらした位置から見た映像を生成する。

**キーワード** 3次元形状復元, ステレオ映像, カメラキャリブレーション

## 1. はじめに

近年、3D映像の普及により、ステレオ映像生成の様々な手法が研究されている。現在、ステレオ映像を作成する場合、ステレオ映像撮影専用のステレオカメラを用いて撮影するのが一般的である。しかし、高価なステレオカメラを持たない一般ユーザーが普段利用するビデオカメラを用いてステレオ映像を撮影しようとするのは非常に困難である。

本研究では、撮影物までの距離を測定できる距離測定カメラの情報を利用し、ステレオカメラではなく、我々が普段利用するビデオカメラを用いて撮影した映像からステレオ映像対を生成する手法を提案する。例えばサッカーやバスケットボールなどのスポーツ観戦時などに観客が普段利用するビデオカメラを用いて撮影した映像を使用することを想定する。そのような状況で撮影された映像のステレオ映像を生成するという事は、その映像を右目用の映像とすると、撮影に使用したビデオカメラが存在した位置から左へ数 cm 平行にずれた位置から撮影したものと同一映像を左目用の映像として仮想的に生成するという事を意味する。この左目用の仮想的な映像を生成するためには、撮影に使用したビデオカメラが撮影物に対してどのような位置からどのような方向に撮影していたかという情報と、撮影物がビデオカメラからのくらの距離に存在するかという情報を求める必要がある。

そこで、本研究では、赤外線を用いてカメラから撮影物までの距離を測定することができる距離測定カメラを使用して撮影物の3次元形状を復元し、その復元した撮影物の3次元形状をどのような位置、方向から見た際のステレオ映像を生成するかを指定し、指定した位置、方向から復元したカラー3次元形状を見た映像を生成する。

その後、その映像を右目用の映像とし、左目用の映像を右目用の映像の座標軸に対して左に数 cm ずらした位置から見た映像を生成することとする。

以下、2章では撮影物の3次元形状を復元するために使用した機器の情報を示す。そして、3章では撮影物の3次元形状の復元手法の手順を説明し、4章で実際に復元した3次元形状とその精度検証を報告する。さらに、5章では復元したカラー3次元形状を用いたステレオ映像生成について示し、第6章で本報告をまとめる。

## 2. 使用した距離測定カメラとビデオカメラ

### 2.1 距離測定カメラ

本研究で使用する距離測定カメラは、スイス MESA 社製 SwissRanger 「SR4000」というカメラで、赤外線 TOF 方式 3次元距離画像センサである。TOF (Time Of Flight) 方式とは、全面の LED より強度変調して発光された赤外線が撮影物に反射して戻ってくるまでの時間を計測することで、物体までの距離を取得する方法である。

この距離測定カメラでは、カメラの内部に設定されている座標軸を基準とした撮影物の  $X$ ,  $Y$ ,  $Z$  座標値 ( $Z$  座標値は奥行き値に相当) と、撮影物の反射強度 (グレースケール値として利用)、取得した各画素の距離情報がどれだけ信頼できるかを示す信頼度情報の大きく3つの情報が、距離測定カメラの各画素に対して取得できる。また、この距離測定カメラの解像度は  $176 \times 144$ 、フレームレートは最小 8fps から最大 50fps まで設定可能である。

### 2.2 ビデオカメラ

2.1 で示した距離測定カメラでは、撮影物のカラー情報は取得できないため、他の機器を用いて撮影物のカラー情報を取得する必要がある。本研究で撮影物のカラー3次元形状復元に使用するビデオカメラは、PointGreyResearch

社製 IEEE1394 カメラ「Flea2」である。

このビデオカメラから取得できる情報は、我々が普段利用するビデオカメラと同様の映像のカラー情報である。また、このビデオカメラの解像度は最小 320×240 から最大 1280×960、フレームレートは最小 1.875fps から最大 15.00fps まで設定可能である。

### 3. カラー 3次元形状の復元

#### 3.1 基本的な考え方

本研究では、1 台の距離測定カメラと 1 台のビデオカメラを使用して、撮影物のカラー 3次元形状を復元する手法を提案する。カラー 3次元形状を復元するために、まずは距離測定カメラとビデオカメラの撮影していた位置や向きを求める必要があるため、2 台のカメラのキャリブレーションを行う。次に、算出した 2 台のカメラのキャリブレーション情報から距離測定カメラで取得した撮影物の 3次元座標群をビデオカメラの映像に透視投影変換によって投影し、各 3次元座標群とビデオカメラの画像面上の画素を対応付ける。そして、対応付けした画像面上の点から 4 点を 1 組とし、その 4 点を 4 隅とする四角形のテクスチャを生成する。その後、生成したテクスチャを 4 隅の点と対応付けている 4 点の 3次元座標を 4 隅とする平面に貼り付ける。この処理を距離測定カメラで取得した全ての 3次元座標群に対して行う。

具体的には、(1) ビデオカメラのキャリブレーション、(2) 距離測定カメラのキャリブレーション、(3) 3次元座標群とビデオカメラの画像面との対応付け、(4) テクスチャの貼り付けの 4 段階処理で構成される。以下、それぞれの処理を具体的に述べる。

#### 3.2 ビデオカメラのキャリブレーション

本研究では、使用するビデオカメラのキャリブレーションを算出する際、OpenCV に実装されている Zhang の手法 [6] を用いて算出する。Zhang によるカメラキャリブレーション手法は、図 1 に示すような、チェックパターンなどの幾何特性が既知の平面パターンを多方向から 3 枚以上撮影し、得られた映像中の特徴点をもとにカメラパラメータを推定する手法である。

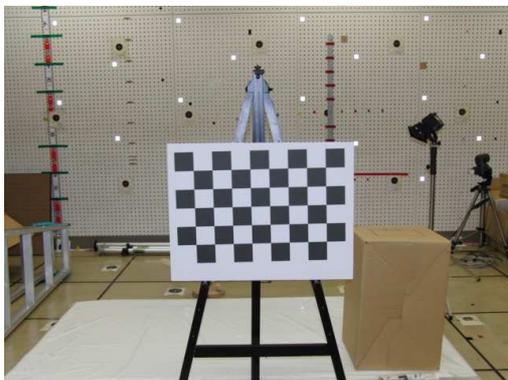


図 1 チェックパターン

#### 3.3 距離測定カメラのキャリブレーション

本研究では、距離測定カメラのキャリブレーションを算出する際、ビデオカメラのキャリブレーションを算出する際に用いた Zhang の手法は用いず、距離測定カメラで取得している撮影物の 3次元座標群の座標値から算出する。これは、Zhang の手法によるキャリブレーションの精度が、ある程度解像度に影響され、本研究で用いる距離測定カメラで取得できる解像度が低いため、推定するカメラのパラメータの精度が悪くなるためである。

基本的な考えは、内部パラメータに関しては、距離測定カメラで取得している 3次元座標群を、透視投影した結果、正確に画像上の画素に当てはまるためのパラメータを最小二乗法を用いて算出する。外部パラメータに関しては図 2 に示すように、まず距離測定カメラの画像中からチェックパターンの 2次元座標  $p_k$  ( $k = 0, \dots$ , 格子点の個数) を算出し、算出した格子点群に対応している 3次元座標  $P_k$  から近似平面を算出し、格子点群から算出した近似平面へ降ろした垂線の足に移動させた 3次元座標  $Q_k$  を算出する。その後、近似平面上に移動させた格子点群  $Q_k$  からワールド座標軸を設定し、設定したワールド座標軸が距離測定カメラの座標軸と重なるように外部パラメータ (平行移動ベクトル  $T$ , 回転行列  $R$ ) を算出する。

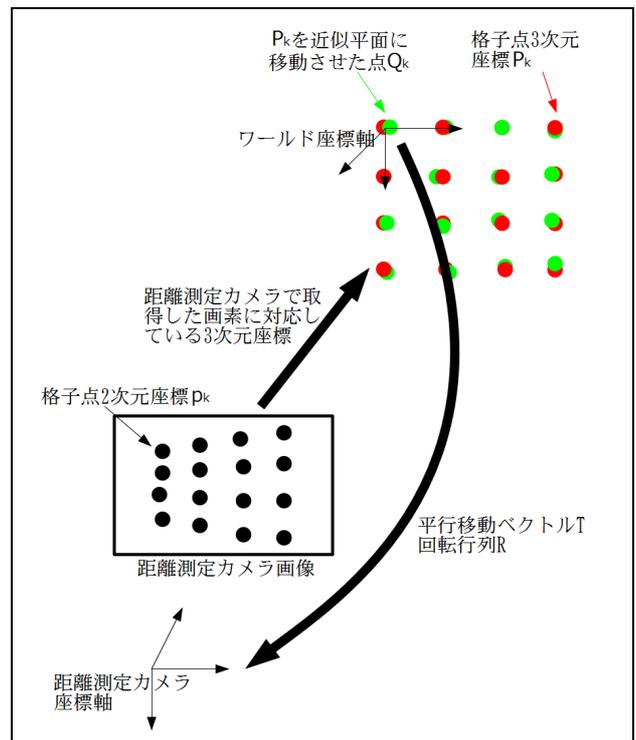


図 2 距離測定カメラのキャリブレーション手順

具体的には、以下の 3.3.1 から 3.3.5 で距離測定カメラのキャリブレーション手法を説明する。

### 3.3.1 内部パラメータの推定

本研究で算出する内部パラメータは、カメラの焦点距離とカメラの光軸中心、半径方向のレンズ歪み係数、円周方向のレンズ歪み係数である。距離測定カメラの内部パラメータは、距離測定カメラで取得した3次元座標群が以下の式を満たすように最小二乗法によって算出する。

$$x' = \frac{x}{z}(1 + k_1 r^2 + k_2 r^4) + 2p_1 \frac{x y}{z} + p_2 (r^2 + 2(\frac{x}{z})^2) \quad (1)$$

$$y' = \frac{y}{z}(1 + k_1 r^2 + k_2 r^4) + p_1 (r^2 + 2(\frac{y}{z})^2) + 2p_2 \frac{x y}{z} \quad (2)$$

$$r^2 = (\frac{x}{z})^2 + (\frac{y}{z})^2 \quad (3)$$

$$u = f_x x' + c_x \quad (4)$$

$$v = f_y y' + c_y \quad (5)$$

式(1)から(5)の中の $X, Y, Z$ は距離測定カメラで取得した空間中の点の3次元座標、 $u, v$ は3次元座標を画像上に投影した際の2次元座標、 $c_x, c_y$ は投影する画像の中心座標、 $f_x, f_y, k_1, k_2, p_1, p_2$ は求めたい距離測定カメラの内部パラメータである。

### 3.3.2 チェックパターンの格子点座標算出

チェックパターンを1枚撮影し、画像中からOpenCVのcvFindChessboardCorners関数を用いて格子点の2次元座標 $p_k$ を算出する。また、距離測定カメラの外部パラメータを算出する際、図3のようにチェックパターン中の一番左上の格子点をワールド座標の原点とし、チェックパターンの最も上の行を $y$ 軸、最も左の列を $x$ 軸となるように算出することとする。

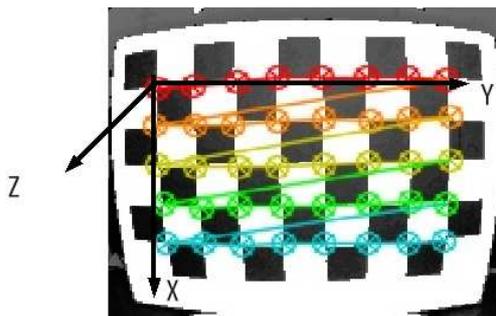


図3 チェックパターンの格子点座標算出

### 3.3.3 格子点の近似平面当てはめ

3.3.2で算出した格子点の座標群からカメラパラメータの基準となるワールド座標軸を設定する。ここで、距離測定カメラで取得した3次元座標群は撮影環境や赤外線集光時間などの距離測定カメラ自体の内部設定により1mm前後の誤差がある。ワールド座標軸を設定する際に、チェックパターンの平面上に座標軸があると仮定しているため、取得した格子点の3次元座標群に近似す

る平面を最小二乗法によって算出し、格子点の3次元座標群を算出した近似平面上の最も近い座標に移動させる。

図4中の赤色の点群は距離測定カメラで取得した格子点群 $P_k$ 、緑色の点群は格子点群 $P_k$ を近似平面上に移動させた点群 $Q_k$ である。

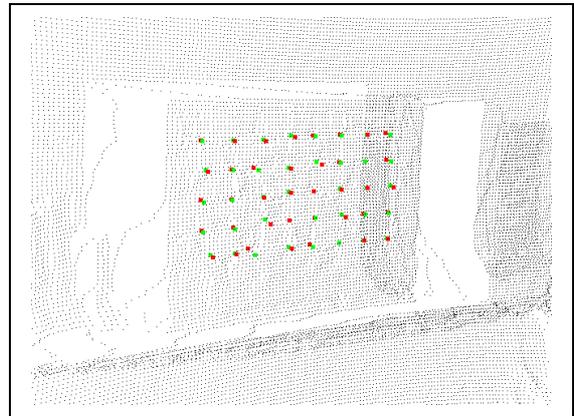


図4 格子点の近似平面当てはめ

### 3.3.4 ワールド座標軸の設定

3.3.3で近似平面上に移動させた格子点の座標群からワールド座標軸を設定する。

まず、近似平面上に移動させた格子点群の中の一番左上の格子点をワールド座標の原点に設定する。そして、ワールド座標の原点を含む1行の格子点群を $y$ 軸、1列の格子点群を $x$ 軸に設定する。その後、この $y$ 軸と $x$ 軸の外積から $z$ 軸を設定する。

しかし、この $x$ 軸と $y$ 軸を設定する際に用いる格子点の座標群は、距離測定カメラの3次元座標の取得精度から必ずしも一直線上には存在するわけではない。そのため、 $x$ 軸と $y$ 軸の設定に用いる格子点群の近似直線を最小二乗法によって算出し、さらに、算出した近似直線が直行することを条件式に加えることによってワールド座標の座標軸を算出する。

図5に算出したワールド座標軸を3次元空間に描画したものを示す。青色の軸が $z$ 座標軸、赤色の軸が $x$ 座標軸、緑色の軸が $y$ 座標軸である。

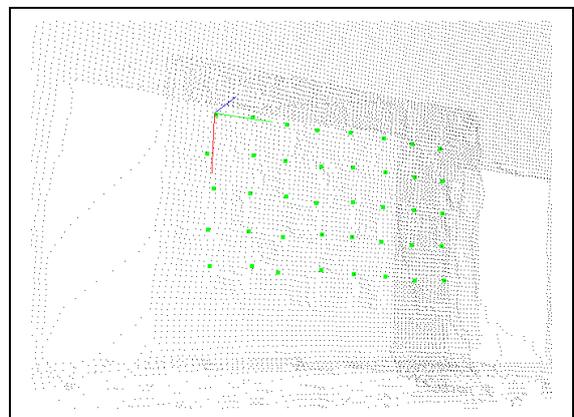


図5 ワールド座標軸の設定

### 3.3.5 外部パラメータの推定

3.3.4で算出したワールド座標軸から，平行移動ベクトル  $T$  と回転行列  $R$  を算出する．距離測定カメラから見たワールド座標の平行移動ベクトル  $T$  は，3.3.4で算出したワールド座標の原点の座標値と同値である．また，回転行列  $R$  は，先程算出した平行移動ベクトル  $T$  によってワールド座標の原点を距離画像カメラの座標軸と一致させ，それぞれの座標軸が一致するように  $x$  軸回転， $y$  軸回転， $z$  軸回転することによって算出する．

図6中の中央の赤色の円で囲まれた箇所に描画されている座標軸がワールド座標軸，左手前に描画されている座標軸が距離測定カメラの座標軸である．

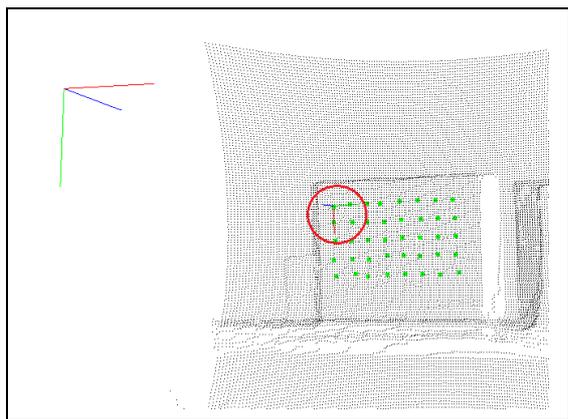


図6 外部パラメータの推定

### 3.4 3次元座標群と画像面との対応付け

3.2と3.3で算出した距離測定カメラとビデオカメラのキャリブレーションから，3.3.1小節で述べた式(1)から(5)と以下の式(6)の透視投影変換行列を用いて，距離測定カメラで取得した3次元座標群をビデオカメラの画像面上に投影することによって3次元座標群とビデオカメラの画像面上の画素の対応付けを行う．

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{T} \quad (6)$$

### 3.5 テクスチャの貼り付け

3.4で対応付けしたビデオカメラの画像面上の座標値  $m_{i,j}$  ( $i = 0, \dots$ , 距離測定カメラの横の画像サイズ,  $j = 0, \dots$ , 距離測定カメラの縦の画像サイズ) から4点を1組とし，その4点を4隅とする四角形のテクスチャを生成する．そして，OpenGLの関数を利用し，テクスチャ生成に用いた4点  $m_{i,j}, m_{i+1,j}, m_{i+1,j+1}, m_{i,j+1}$  を  $\text{glTexCoord2f}$  関数に入力し，対応する3次元座標  $M_{i,j}, M_{i+1,j}, M_{i+1,j+1}, M_{i,j+1}$  を  $\text{glVertex3f}$  関数に入力することによってテクスチャを貼り付ける．

## 4. 復元したカラー3次元形状と精度検証

### 4.1 復元したカラー3次元形状

3.で記述した手法によって復元した撮影物のカラー3次元形状について述べる．距離測定カメラとビデオカメラを1台のパソコンに接続し，同時に2台のカメラを動作させるアプリケーションを開発した．図7から9に開発したアプリケーションの実行画面を示す．

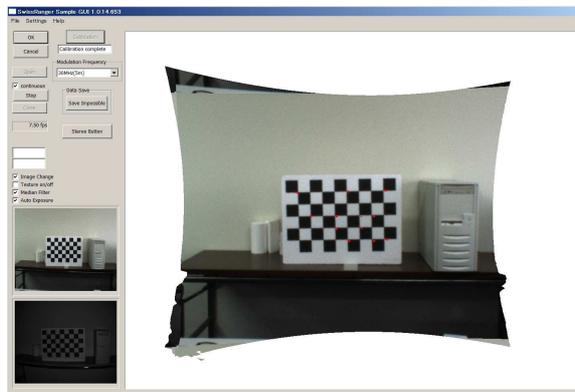


図7 カラー3次元形状復元(正面)

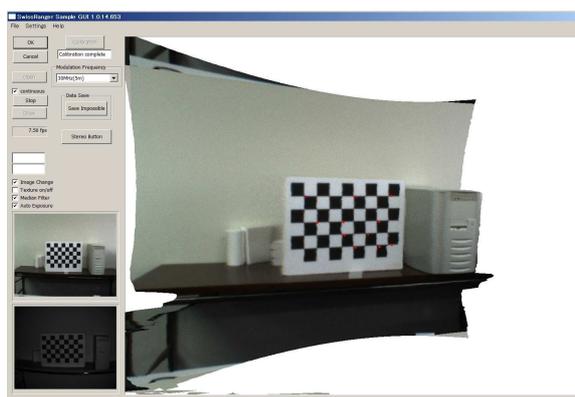


図8 カラー3次元形状復元(右斜め)

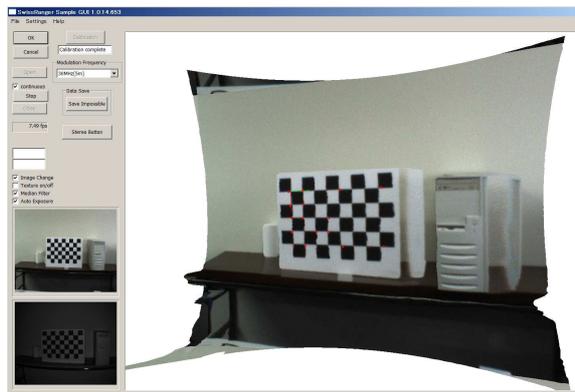


図9 カラー3次元形状復元(左斜め)

このアプリケーションでは，距離測定カメラとビデオカメラの起動，データ取得，2台のカメラのキャリブレーションの推定，カラー3次元形状の復元という必須機能は勿論のこと，その他に以下の機能も付け加えた．

- アプリケーションの現在の実行速度表示 (fps 表示)
- 距離測定カメラが発光する赤外線の変調周波数の変調
- 距離測定の精度に関わる集光時間の設定
- 距離情報やカラー情報などの取得データの保存
- メディアンフィルタによる距離情報の補正

## 4.2 Zhang の手法との精度比較

本研究の 3.3 で提案した手法を用いて復元したカラー 3 次元形状を図 10 から 12 に、Zhang の手法によるキャリブレーションを用いて復元したカラー 3 次元形状を図 13 から 15 に示す。

図 11 と図 14 の正面から見たワールド座標軸の精度を見比べると、図 11 では一番左上の格子点と座標軸の原点がほぼ一致しており、各座標軸の方向も格子点の並びとほぼ一致していることが分かる。図 12 と図 15 の斜めから見比べても同様のことが分かる。

また、実際に距離測定カメラで取得した 3 次元座標群にテクスチャを貼り付けた図 10 と図 13 のチェッカーボードの端を見ると精度の差が分かり易い。図 10 に示す本手法を用いて復元したカラー 3 次元形状では、距離測定カメラで取得したチェッカーボードの 3 次元座標群に綺麗にチェッカーボードのテクスチャが貼られているのに対して、図 13 に示す Zhang の手法を用いて復元したカラー 3 次元形状では、チェッカーボードのテクスチャが正しい位置から右下方向にずれた状態で貼られていることが分かる。

また、実際に距離測定カメラで取得した 3 次元座標群にテクスチャを貼り付けた図 10 と図 13 のチェッカーボードの端を見ると精度の差が分かり易い。図 10 に示す本手法を用いて復元したカラー 3 次元形状では、距離測定カメラで取得したチェッカーボードの 3 次元座標群に綺麗にチェッカーボードのテクスチャが貼られているのに対して、図 13 に示す Zhang の手法を用いて復元したカラー 3 次元形状では、チェッカーボードのテクスチャが正しい位置から右下方向にずれた状態で貼られていることが分かる。

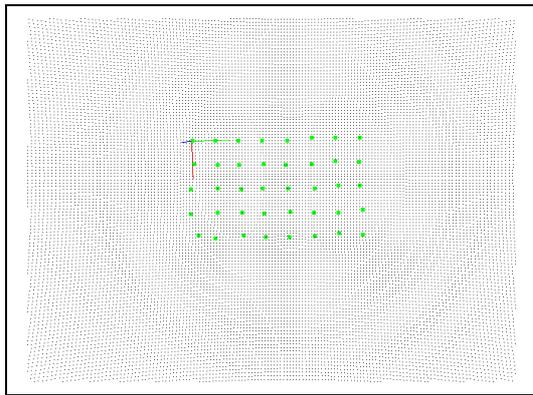


図 10 本手法を用いたカラー 3 次元形状復元

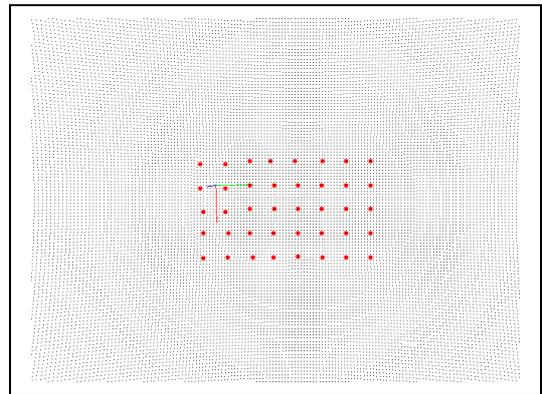


図 13 Zhang の手法を用いたカラー 3 次元形状復元

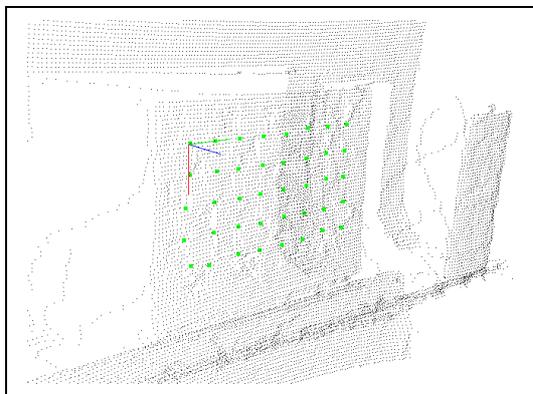


図 11 本手法を用いたワールド座標軸の精度 (正面)



図 14 Zhang の手法を用いたワールド座標軸の精度 (正面)

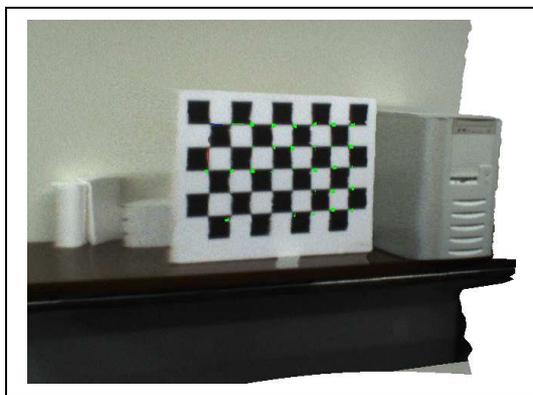


図 12 本手法を用いたワールド座標軸の精度 (左斜め)

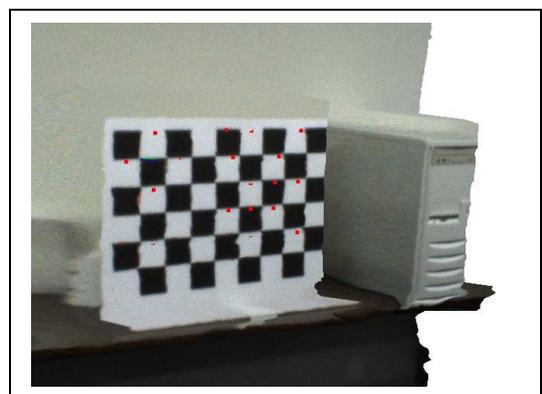


図 15 Zhang の手法を用いたワールド座標軸の精度 (左斜め)

## 5. ステレオ映像の生成

4. で示した復元したカラー3次元形状を利用して、任意の視点からのステレオ映像を生成する。

### 5.1 ステレオ映像生成の基本的な考え方

ステレオ映像を生成するためには、撮影物までの距離の情報と撮影物をどのような位置からどのような方向に撮影したかという情報を求める必要がある。

そして、撮影物までの距離の情報に関しては、4. で示した復元した3次元形状を利用することによって、任意の位置から撮影した映像を作り出すことで求めることができる。

また、どのような位置からどのような方向に撮影するかはこちらから指定し、その情報から右目用の映像の位置と方向、左目用の映像の位置と方向を算出する。

こちらから指定した平行移動ベクトルを  $\mathbf{T}$ 、回転行列を  $\mathbf{R}$ 、右目用映像の位置に対する左目用映像の位置のずらし量を  $t$  とすると、右目用の映像の位置と方向は式(7)で算出でき、左目用の映像の位置と方向は式(8)で算出することができる。

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{T} \quad (7)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \left( \begin{bmatrix} -t \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \right) + \mathbf{T} \quad (8)$$

### 5.2 ステレオ映像生成

5.1 で示した考え方によって生成したステレオ映像の1フレームを図16と図17に示す。

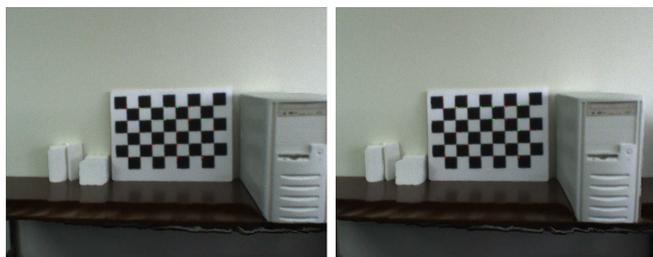


図 16 左目用画像

図 17 右目用画像

図16と図17を見ると、机の端や机の上に置いてある石膏の端などで、実際の物にはない不自然な形に見える箇所が見られた。

これは、距離測定カメラで取得した3次元座標の精度の問題やビデオカメラで撮影できていない箇所のためにカラー情報が取得できていないことが原因であると考えられる。

## 6. ま と め

本論文では、1台の距離測定カメラと1台のビデオカメラを使用して撮影物のカラー3次元形状をリアルタイムで復元する手法を提案した。

本手法では、キャリブレーション時にチェックパターンを撮影しなければいけないという制限もあるが、精度の高いカラー3次元形状の復元ができた。そして、復元したカラー3次元形状を利用して任意視点からのステレオ映像を生成することができた。

しかし、情報を取得する機器が距離測定カメラ1台、ビデオカメラ1台ということもあり、それぞれのカメラから見えない箇所のカラー3次元形状復元については改良の余地が見られ、カラー3次元形状復元に複数の距離測定カメラとビデオカメラを用いる場合への拡張は今後の課題である。

さらに、復元した撮影物のカラー3次元形状とどこから撮影したか分かっていないビデオカメラで撮影した映像から、そのビデオカメラのキャリブレーションを推定し、その映像のステレオ映像を生成する手法の開発も今後の課題である。

謝 辞

本研究の一部は(独)情報通信研究機構の委託研究として行われた(No.143イ105)。

## 文 献

- [1] 株式会社日本クラビス:スイス mesa 社 Swissranger ホームページ, <http://www.j-clavis.co.jp/Mesa-sr>
- [2] PointGreyResearch 社ホームページ, <http://www.ptgrey.com>
- [3] OpenCV-1.0 リファレンスマニュアル (日本語訳), <http://opencv.jp>
- [4] 奈良先端大 OpenCV プログラミングブック製作チーム, "OpenCV プログラミングブック", 毎日コミュニケーション出版(2007)
- [5] 出口光一郎, "ロボットビジョンの基礎", コロナ社(2000)
- [6] Zhengyou Zhang, "A flexible new technique for camera calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334(2000)