1T-07

Kinectを用いたリアルタイムプロジェクションマッピングシステムの構築

藤澤　春香†　　藤田　悟†

法政大学　情報科学部†

1. 背景

近年、AR 技術の一つとしてプロジェクションマッピングが注目を集めている。本研究では深度センサの Kinect を用いてプロジェクタとカメラのキャリブレーションを行い、対象物の位置推定を行うことで、プロジェクションマッピングを容易に行えるシステムを提案する。提案手法では、静止物体だけでなく、動く物体を追従して映像を投影することができるため、リアルタイムでのプロジェクションマッピングが可能となる。

2.プロジェクションマッピング技術

コンピュータ上で生成した映像をプロジェクタから投影し、実空間中の物体と光学的に重畳させる技術をプロジェクションマッピングという。一方、カメラで撮影した情報に応じてプロジェクタから出力する映像を生成するようなカメラとプロジェクタを組み合わせたシステムとしてプロジェクタカメラシステムが存在する。本研究ではこのプロジェクションマッピングとプロジェクタカメラシステムの融合を提案する。

既存のプロジェクションマッピングシステムは静止空間だけを対象としているため投影する対象に変化が生じた場合に対応できない。一方、提案手法ではプロジェクタに組み合わされたカメラで投影空間を逐次撮影、計測することで動的な物体にも投影することができる。この際に、プロジェクタとカメラの幾何学的整合性と対象物の位置推定方法の 2 つの課題が生じる。プロジェクタとカメラの幾何学的整合性とは、カメラで撮影したある点にプロジェクタから映像を投影するときに、カメラ画像とプロジェクタ画像の対応関係を取ることである。さらに対象物が動く場合、カメラの情報から対象物の位置を正確に推測する必要があり、このときに問題となるのが対象物の位置推定方法である。

3. 提案手法

3.1. システム概要

本システムは 2 つの工程を用いてプロジェクションマッピングを実現する。まず、プロジェクタと Kinect のキャリブレーションを行い、次にマッピングしたい物体をカメラの前に配置して撮影し、物体の位置を推定する。そして、この物体の位置とキャリブレーションパラメータから投影画像を生成し、プロジェクタから投影する。

3.2. Projector-Kinect キャリブレーション手法

本システムでは長岡らの研究[1]に基づいてプロジェクタと Kinect のキャリブレーションを行う。Kinect を中心とした 3 次元空間からプロジェクタ画像への透視投影行列は、以下のように定義される。

$$C = A[R|T] = \begin{bmatrix} fs_u & fk_s & u_0 \\ 0 & fs_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \quad \cdots (1)$$

ここで、A は Kinect の内部行列(既知)、R はプロジェクタ-Kinect 間の回転行列、T は並進行列である。この透視投影行列は 3×4 で 12 個の未知パラメータを持つため、3 次元空間とプロジェクタ画像座標間の対応点を 6 点以上取得すれば行列 C を推定できる。

対応点からの行列の推定には 6 Points Algorithm[2]を用いたあと、ガウス・ニュートン法を適用し最適化することで算出する。プロジェクタ画像座標を求める式は、3 次元座標を$(X_{world} \quad Y_{world} \quad Z_{world})$,プロジェクタ画像座標を$(x_{image} \quad y_{image})$とし、上記の透視投影行列 C を用いて式(2)のように表すことができる。

$$\begin{pmatrix} x_{image} \\ y_{image} \\ 1 \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{pmatrix} \begin{pmatrix} X_{world} \\ Y_{world} \\ Z_{world} \\ 1 \end{pmatrix} \quad \cdots (2)$$

3.3. 物体の位置姿勢推定

本研究では ICP を用いた点群の位置合わせによる推定方法(手法 1)と、RGB 画像から輪郭を抽出する方法(手法 2)の 2 つの位置推定方法を提案する。

手法 1 は、Kinect で計測したデータに物体のモデルデータを重ねることで物体の位置を推定する手法である。ICPによる位置合わせを行うことで、モデルからの座標変換行列が求められる。モデルの初期座標を$(X_{model} \quad Y_{model} \quad Z_{model})$，座標変換後の座標を$(X_{trans} \quad Y_{trans} \quad Z_{trans})$とすると、この変換は

$$\begin{pmatrix} X_{trans} \\ Y_{trans} \\ Z_{trans} \\ 1 \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_{model} \\ Y_{model} \\ Z_{model} \\ 1 \end{pmatrix} \quad \cdots (3)$$

となる。これらの座標は Kinect を中心とした 3 次元座標空間上のものであるので、式(2)を使ってプロジェクタ画像上の座標に変換することで投影画像が得られる。

手法 2 は、RGB 画像を用いた輪郭抽出による位置推定方法である。手法 1 と同様にプロジェクタとキャリブレーション済みの Kinect でマーカーが施された立方体を撮影し、RGB カメラの画像から対象物の輪郭を抽出する。Kinect では各カメラ画素に対応した 3 次元座標を取得できるため、輪郭を抽出してカメラ座標上の特徴点を取得することで特徴点の 3 次元座標が取得できる。この座標からも式(2)を用いてプロジェクタ画像座標が求められ、投影画像が得られる。

4. 投影実験

4.1 システム構成

プロジェクタの上に Kinect を固定し、プロジェクタから映像を投影する。キャリブレーション時には取得する

Real-time projection mapping system with Kinect
†Haruka Fujisawa, Satoru Fujita
†Faculty of Computer and Information Sciences, Hosei University

点の深度にばらつきを出すために凹凸面を配置し、この凹凸面にプロジェクタ画像座標が既知の白円を 35 個投影して対応点を取得する。キャリブレーションが完了したら、対象物を配置し、映像を投影する。実験では対象物を一辺 12cm の立方体とする。

4.2 実験概要

手法 1,2 の 2 つの方法で立方体の各面に映像を投影し、投影精度及び処理速度を比較する。立方体を回転台の上に置いて、$6.0°/s$の速さで立方体の 3 面が最大に見える角度を 0° として、-90° ～90° までの 180° の区間で回転させ、各面それぞれにおいて投影されている画像を観察する。投影精度を比較するために、目視で立方体各面を観察し、面積比を各面の面積に対する投影画像の面積の割合として比較する。投影面の面積は四角形の頂点の座標から座標法により算出する。

4.3 実験結果

4.3.1 ICP を用いた位置姿勢推定

-90° ～90° までの左面･上面･右面への面積比を表 1 に示す。斜線部は、そもそもプロジェクタの光が面に当たらないため、投影できない面である。-90° では 96%となり、高精度な結果が得られたが、-30° では 40%以下の精度となった。速度の面では 1 フレームあたり約 650ms ととても遅いため、回転に投影が追い付かなかった。また、回転を停止させた場合も、ICP が正しい位置に収束するまでに時間を要するため、停止させてから正しく投影が行われるまでに時間差が生じた。

4.3.2 RGB 画像を用いた輪郭抽出による手法

-90° ～90° までの左面･右面への面積比を表 1 に示す。各位置で 90%を超える結果が得られた。面積比が 100%を超えているものは面の範囲外にも映像が投影されたために基準面積より投影面が大きくなってしまったことを表す。速度の面では 1 フレームあたり約 63.2ms であり、フレームレートにすると約 15fps であった。回転の速さにはついていけているものの、動き始めや停止時に若干の遅れがみられた。

4.4 考察

手法 1 では、点群処理の重さがリアルタイムでプロジェクションマッピングする上で問題となった。また、ICP も常に正しい結果が得られるわけではなく、位置推定が失敗してしまうこともある。対象が大きい場合は点群の点の数も増えてしまうためさらに処理が重くなってしまうことが考えられる。これらを解決するためには、位置合わせに使用する点群を精度が落ちない範囲で最小化し、また毎フレームごとに位置合わせをかけなくても済むアルゴリズムが必要である。

手法 2 では、手法 1 に比べて処理が軽く、リアルタイムに投影することが可能であったが、実際に検出された面の形や、投影した面の形が歪んだ四角形となることがあった。また暗い空間では RGB 画像が撮れないため、ある程度の明るさが必要であり、これは映像投影コンテンツとしては致命的である。

5. 任意形状への投影

5.1 提案手法

手法 1 では点群処理による ICP アルゴリズムを用いた位置推定を行ったが、これを応用した立方体以外の任意形状の対象への投影手法について検討した。投影したい画像を準備し、事前にその画像を対象物に投影する。このとき、対象物の 3 次元座標のほかに RGB 値も計測する。この後対象物を移動させ、対象物に位置合わせされたモデルデータの各点を、計測した RGB 値で式(2)を用いて計算したプロジェクタ画像座標に 1 点ずつ描画する。

表 1. 手法 1,2 の面積比(5 回試行の平均)

手法1	左面	上面	右面
-90°	96%		96%
-60°	92%		79%
-30°		36%	39%
0°	93%	94%	95%
30°	96%	83%	82%
60°		47%	86%
90°	59%		96%

手法2	左面	右面
-90°	92%	98%
-60°	96%	94%
-30°		103%
0°	94%	97%
30°	101%	
60°		98%
90°	98%	96%

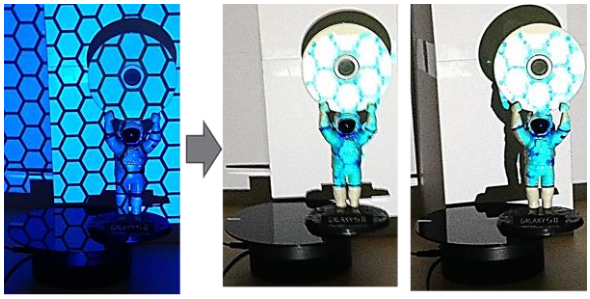

図 1. 任意形状への投影結果

5.2 投影実験

対象物に映像を投影し、見え方及び処理速度を比較する。画像をプロジェクタから対象物に投影しておき、この時対象物に映っている状態を対象が回転しても同じ見え方となるように再現することを目標とする。4 節での実験と同じ回転台の上に対象物を設置し、同様に-60° ～60° までの 120° の区間で回転させ、投影画像の見え方を観察する。

5.3 実験結果と考察

図 1 に投影結果を示す。対象の位置や角度が変わっても、移動前に対象に移っていた柄がそのままの状態で投影されているのが分かる。現在はモデルデータの点のドット絵として投影しているだけであるが、これはテクスチャを投影するのが望ましい。そのためには仮想空間で対象物にテクスチャマッピングを行い、その結果できたテクスチャを投影する必要がある。

6. まとめ

本研究では Kinect を用いて容易にプロジェクションマッピングを行えるシステムを提案し、Kinect とプロジェクタがあれば簡単にプロジェクションマッピングを行えることを示した。ICP アルゴリズムを用いた位置合わせによる手法と、RGB 画像を用いた輪郭抽出による手法の 2 種類の位置推定方法でプロジェクションマッピングシステムを構築し、さらに任意形状への投影についても検討したが、主に処理速度が大きな問題として残り、投影の遅れが目立ってしまう結果となった。リアルタイムで満足な投影を実現するには今後さらなる処理の軽量化、高速化が必要である。

参考文献

[1] 長岡亜耶，橋本直己，”深度センサを用いた手軽なプロジェクタキャリブレーション,” 映像情報メディア学会技術報告,vol.37,No.7,pp.3-6,Feb.2013.

[2] 佐藤淳:”コンピュータビジョン-視覚の幾何学-”,コロナ社，名古屋，1999.