

適応的なパターン変更によるシーンの三次元構造の復元

6G-2

日置尋久† 品川嘉久†

東京大学†

1 はじめに

シーンの三次元構造を理解するためには、二次元画像系列からシーンに存在する物体の形状やそれらの位置関係を復元する必要がある。三次元形状復元の代表的な方法として、三角測量に基づくステレオ法が挙げられるが、一つの画像上に写ったシーン内のある点に対応する他の画像上の点を如何に同定するかということが常に問題となる。またステレオ法は受動的な方法であるため、特にテクスチャの少ない領域においては、対応点の探索は非常に困難になる。

三次元形状復元のもう一つの代表的な方法として、パターン光投影法がある。投影されたパターンとそれ以外の部分とのコントラストが十分であれば、テクスチャの有無に関わらず画像からパターンを抽出することは容易であり、画像に写ったパターン上の各点がそれぞれどこから投影されたかということが同定できれば、ステレオ法と同様に三角測量の原理から、それらの点に対応する三次元座標を復元できる。

本研究では、パターンを抽出した結果をフィードバックとして、次々とパターンを変更していくシーンの三次元構造復元法を提案する。この方法を用いて能動的に情報収集を行えば、より少ない測定回数で、より多くのデータが得られることができる。

2 システム概要

本手法では、図1のようなシステムを用いた。ホストから送ったパターンをビデオプロジェクタでシーンに投影し、それをビデオカメラで撮影して、画像をホストで受け取るというループを構成している。ここでは、カメラとプロジェクタはおおよそ垂直にならんでいるものとする。

撮影時には既知の形状を持つ物体（中空の立方体）を基準として世界座標系を定義し、その物体に特定のパターンを投影しカメラで撮影することにより、画像（カメラ）座標系及びパターン（プロジェクタ）座標系と世界座標系との関係を決定しておく。例えば画像座標系 (x_i, y_i) と世界座標系 (X, Y, Z) との関係は、カメラの位置、姿

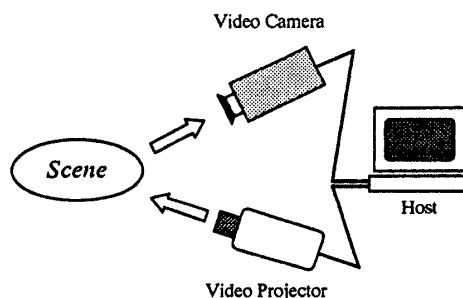


図 1: システム概念図

勢、内部パラメタ等を含めて、

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$x_i = x/z, y_i = y/z \quad (z \neq 0)$$

のように表すことができる。パターン座標系と世界座標系との関係も同様に記述することができる。

3 三次元形状復元アルゴリズム

パターン投影法で問題になるのは、画像中のパターンの同定法である。例えば一本のスリット光を順次走査させながら測定を行えば、パターン同定の問題は容易に解決できる。しかしこの方法では、十分な量のデータを収集するのに要する時間が長くなってしまふ（測定回数が多くなる）。一枚の画像中の複数のパターンを同時に同定するための方法として、色によるパターンのコード化、パターン同士の位置関係によるコード化などがあるが、これらの方法は、シーン中の物体の色や形状に影響されやすい。

またパターンを一定の規則で変化させながら何枚かの画像を撮り、ピクセル値の変化によって、パターンをコード化する方法があり、グレイコード [1] や M 系列符号 [2] が用いられている。しかしこれらの従来法では、ある一定の枚数の画像をとる必要があった。

そこで本研究では、ある時点でのパターンの抽出結果をフィードバックとして見ながらパターンを変更していく方法を提案する。基本的な方針は、まず出せるだけパターンを出して、全パターンが同定できない場合はパター

Recovering Three-Dimensional Shapes by Projectiing Adaptive Patterns
Hirohisa Hioki† and Yoshihisa Shinagawa† The University of Tokyo†

ンを減らして、パターンが消えた部分を利用して再び同定を行なう、というものである。

具体的なアルゴリズムは以下の通りである。なお、投影するパターンは等間隔に並んだ水平なスリットの集合であるとする。また以下で画像 $(I_p - I_q)$ と書いた時には、画像 I_p の各ピクセルの値から画像 I_q の対応するピクセルの値を引いた差分画像を表すものとする。

1. パターンがないときの画像 I_B を撮る。
2. $2^n - 1$ 本のスリットパターンを投影し、画像 I_0 を撮る。
3. 画像 $(I_0 - I_B)$ の各列でパターンを抽出し、パターンの id を new としておく。これらのパターン id は同定されるたびにに対応するスリット番号に置き換えられる。
4. 各列で $2^n - 1$ 個のパターンがとれていれば終了。
5. そうでないときは、 2^{n-1} 番目のスリットを消して、画像 I_1 を撮る。
6. 画像 $(I_0 - I_1)$ から 2^{n-1} 番目のスリットを抽出。
7. 画像 $(I_1 - I_B)$ の各列でパターンを抽出し、(2^{n-1} 番目のスリットの位置を使って) 各列で全ての new が同定できれば終了。
8. 以下同様に $(2i - 1)2^{n-i}$ ($i = 1 \dots 2^{i-1}$) 番目のスリットを消しながら、全ての new が同定されるか、パターンが全て消えるまで、6,7と同様の操作を繰り返す。

4 実験結果

図2のようなシーンを撮影して実験を行なった。ここでは $n = 5$ とした。図3は本手法によってパターンを同定した場合の $t = 3$ の時点での結果である。図4には、上記アルゴリズムの7を行なわず、単純に画像 $(I_{t-1} - I_t)$ のみを使ってパターン同定を試みた場合の同じ $t = 3$ の時点での結果を示した。また図5が最終的に三次元形状を復元した結果である。

5 まとめ

時間的に変化するパターンを投影する三次元形状復元法を提案した。パターン抽出結果をフィードバックとすることにより、より少ない画像でより多くのパターンを同定できることを示した。

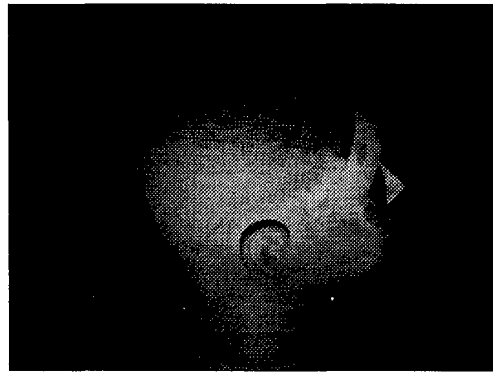


図2: 実験シーン

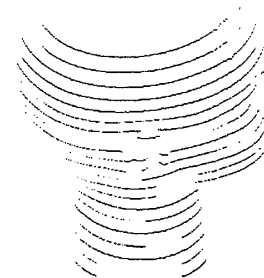


図3: パターン同定: 途中結果 (本手法)

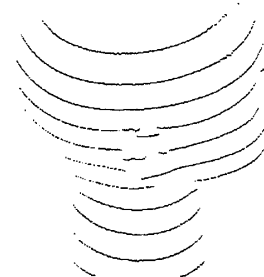


図4: パターン同定: 途中結果 (単純な方法)

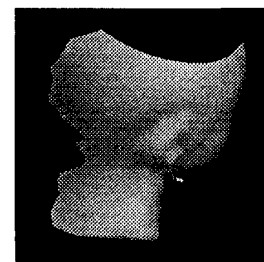


図5: 三次元形状の復元結果

参考文献

- [1] 佐藤宏介, 井口征士. 空間コード化による距離画像入力. 電子情報通信学会論文誌, Vol. J68-D, No. 3, pp. 369-375, 1985.
- [2] 余田茂, 佐藤宏介, 千原國宏. 変調投影光による時空間処理を用いた三次元計測. 情報処理学会研究報告, Vol. 94, No. 104, pp. 1-6, 1994.