

# ビデオ映像からの高速な曲面情報抽出

## Real-Time Surface Extraction from Video Sequence

吉田 壘                      増田 宏  
Lui Yoshida                Hiroshi Masuda

東京大学  
The University of Tokyo

### 1. はじめに

現在、ビデオ映像から特徴点の 3 次元座標を取得する手法が知られており、ビデオ映像から取得された特徴点群を用いて映像と CG を合成したり、特徴点群からの形状モデリングを行う研究がなされている[1][2]。しかし、こうした研究の多くは、撮影している映像に対して、リアルタイムに処理を行うものではない。

リアルタイムに CG 画像を合成するための手法として、Parallel Tracking and Mapping[3] (以後 PTAM) が提案されている。この手法は、カメラ位置を把握するための特殊なマーカを使用せず、特徴点の 3 次元座標を取得する。しかし、平面抽出しかできないため、平らな対象物とのインタラクションしか実現できていない。たとえば、床面に CG オブジェクトを配置するといった限定的な利用しかできない。

本研究では、ビデオ映像から、任意の曲面を抽出することで、対象物の形状を反映したテクスチャマッピングや衝突計算をリアルタイムに実現するための手法を提案する。本手法は、インタラクティブなゲームへの応用などが想定される。

### 2. 関連研究

ビデオ映像から抽出された疎な点群の 3 次元情報を利用して、オフラインで 3 次元モデルの作成をする手法がいくつか提案されている。画像に加えて、3 次元座標を持った点群を付加することで、ステレオ画像によるモデリングよりも効率が大きく向上する。Anton van den Hengel ら[1]はその点群の情報と、ユーザの指示をもとに、効率的に 3 次元のモデルを作成している。その際、曲面フィッティングもしているが、リアルタイムな処理ではない。また、Thorsten Thormahlen ら[2]はビデオ映像から抽出された疎な点群を用

いて対象物を囲む直方体を生成し、物体の 3 面図を作成した。そして、その 3 面図を用いることにより、モデリングの精度を向上させ、所要時間も削減した。

リアルタイムにビデオ映像を解析してカメラの位置・姿勢を推定する手法もいくつか提案されている。拡張現実感の研究のため開発された C 言語ライブラリである ARToolkit[4]を用いれば、容易に特殊なマーカ上に 3 次元オブジェクトを表示することが出来る。その際、得られる環境の情報としてはマーカが存在する平面である。一方 PTAM は、マーカレスで、環境を特徴点の集合として認識し、その環境(マップ)を更新していくといった手法である。マップの更新と特徴点のトラッキングを別スレッドに分けて処理をすること、精度よくトラッキングができたキーフレームを解析に使用することによって、高速で正確で安定しているシステムを実現することを可能にしている。この手法で得られる 3 次元情報は環境の特徴点群である。ただし、これら既存手法は映し出される物体の形状を抽出していない。

### 3. 提案手法

#### 3.1. 概要

ビデオ映像の特徴点を追跡することで、特徴点の 3 次元座標を取得する。そして、取得された特徴点の一部をユーザが指定し、その曲面モデルを作成する。まず 3 次元座標取得の方法について述べ、曲面を抽出する方法を説明する。そして曲面がリアルタイムに抽出可能ということを示す。

#### 3.2. 3 次元情報の取得

PTAM はビデオ映像を解析し、そこに映し出される特徴点群の 3 次元情報を抽出する。本研究では、その点群を曲面フィッティングに利用する。

### 3.3. 曲面フィッティング

ビデオ映像から取得された点群に対して、その一部をユーザが指定し、曲面フィッティングを行う。本研究では曲面としてBスプライン曲面を用いる。

物体形状を正確に把握するため曲面フィッティングの安定性が問題になる。特にビデオ映像より取得される点群は疎であることが多いため、少ない点群に対しても安定に曲面を計算できることが必要である。また、曲面フィッティングは平面に比べてパラメータが多いため、リアルタイムに処理するには、計算時間が問題となる。

具体的な曲面フィッティングにおいては、まず個々の点  $P_k = (x_k, y_k, z_k)$  に対応する曲面のパラメータ  $(u_k, v_k)$  を算出する必要がある。このパラメータの初期値は点群を近似する平面を求めて、その平面上に各点を投影することで計算される。点群を近似する平面を点群と平面との距離の2乗和が最小になるような平面とする。そしてその計算されたパラメータを用いて、Bスプライン曲面の制御点を求める。

曲面の計算では、曲面  $S(u_k, v_k)$  と点  $P_k$  との距離の2乗和が最小になるように制御点を求めることができる。しかし本研究で扱う点群は、比較的まばらであり、 $uv$  パラメータ領域全体を覆うものではないため、計算が不安定になる。そこで、本研究では、計算を安定化させるために、滑らかさに関する制約を追加する。

最小化する関数は、Bスプライン曲面  $S$ 、点数  $N$ 、点群  $\{g_k\}$ 、点群に対応するパラメータ  $u_k, v_k$ 、滑らかさに関する制約の重み  $\beta$  を用いて以下のように表せる。

$$\sum_{k=1}^N \|S(u_k, v_k) - g_k\|^2 + \beta \iint (|S_{uu}|^2 + 2|S_{uv}|^2 + |S_{vv}|^2) dudv \quad (1)$$

$$\text{ただし } S_{uu} = \frac{\partial^2 S}{\partial u^2}, S_{uv} = \frac{\partial^2 S}{\partial u \partial v}, S_{vv} = \frac{\partial^2 S}{\partial v^2}$$

実際計算する際は (1) を離散化した数値積分を用いる。つまり最小化する関数として、 $u, v$  方向のサンプリング数を  $M_u, M_v$ 、サンプリングされたパラメータを  $\{u_i, v_j\}$  として

$$\sum_{k=1}^N \|S(u_k, v_k) - g_k\|^2 + \beta \sum_{i=0}^{M_u-1} \sum_{j=0}^{M_v-1} (|S_{uu}(u_i, v_j)|^2 + 2|S_{uv}(u_i, v_j)|^2 + |S_{vv}(u_i, v_j)|^2) \mu S \quad (2)$$

を使用する。

また、曲面の品質を上げるためには、パラメータ  $(u_k, v_k)$  の再設定を行う必要がある。パラメータは、データ点と曲面の距離が最小になるような点をニュートン法を用いて決定する。そして、そのパラメータを用いて式(2)を最小化するような制御点を再度計算する。この処理を複数回繰り返すことで、曲面が計算される。

### 3.4. 反復回数

反復回数は計算時間に対して影響を与える。そのため、リアルタイム処理を実現するには反復回数が少ない方がいい。しかし、反復回数が少なすぎると、曲面の品質が低下する可能性がある。そこで本研究では、出来るだけ曲面の品質を損なうことなく、少ない反復回数を実験的に求めた。

半径1, 高さ3の円柱の表面一部をサンプリングした点群、それらにノイズをのせた点群をそれぞれ A, B とする。A, B それぞれ、フィッティングした曲面と点群の距離の2乗和の平方根、反復回数、制御点数の関係を表すグラフを示す(図1)。ここでデータ点数は100とした。グラフより、制御点数がいずれの場合にも反復回数が5付近を超えると誤差の変化がそれより前に比べて小さくなることが確認できる。実際フィッティング結果をみても反復回数5以降の差は明確ではない。よって本研究では、反復を5回行うことにした。

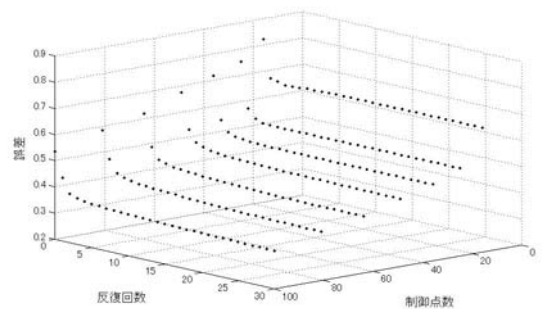
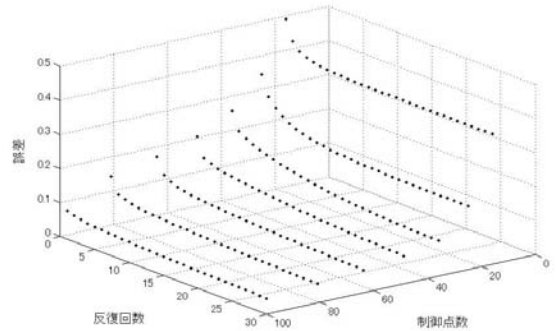


図1 反復回数と誤差(上:A 下:B)

### 3.5. 計算時間

計算時間に大きく影響を与えるものは制御点数・データ点数・反復回数である。複雑な形状を忠実に表すには制御点数が多い方がよいが、計算時間は増加する。また、データ点数が多い場合や、品質を上げるために反復回数を大きくする場合には、計算コストが増大する。リアルタイムに計算を行うためには、これらの相反する要求のバランスを取ることが必要である。

前節の結果より反復回数を5に固定し、制御点数・データ点数の組み合わせによって、計算時間がどう推移するかを実験によって検証した。その結果(100回計算した際の平均時間)のグラフが図2である。制御点数が100、データ数1000、反復回数5でも70msを越えず、リアルタイム処理ができています。

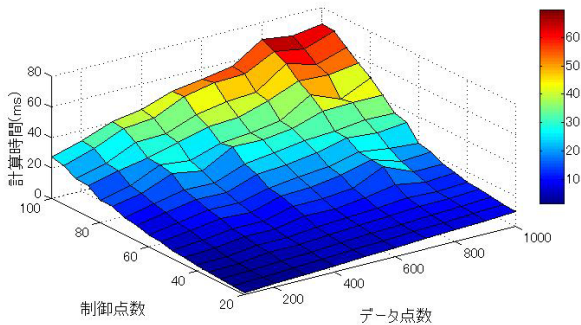


図2 制御点数とデータ点数と計算時間の関係

ただし、この範囲を越える場合には、反復回数の削減やデータの間引きが必要となる。本システムでは、ユーザが制御点数と最大反復回数を指定できるようにしている。なお、実験環境は OS: Windows XP SP3, CPU: Intel Core 2 Duo P9600 2.66GHz, メインメモリ: 4GB である。

## 4. 応用システム

本手法の有効性を検証するために、曲面抽出と物理シミュレーションを組み合わせた応用システムを作成した。物理シミュレーションには、PhysX[5]を用いた。

まず、曲面抽出のインターフェースを説明する。ユーザの初期化処理により、システムはビデオ映像から特徴点を抽出する。そして、ビデオ映像にその点をリアルタイムに描画する(図3)。ユーザは、そこから曲面として認識させたい領域を矩形で指定する。その際、指定された点群をさらに

矩形範囲選択することによって、フィッティングさせたい点群を絞り込むことが出来る(図4)。そして、ユーザがボタンを押すことによって、システムはその点群に対して曲面フィッティングを行う(図5)。

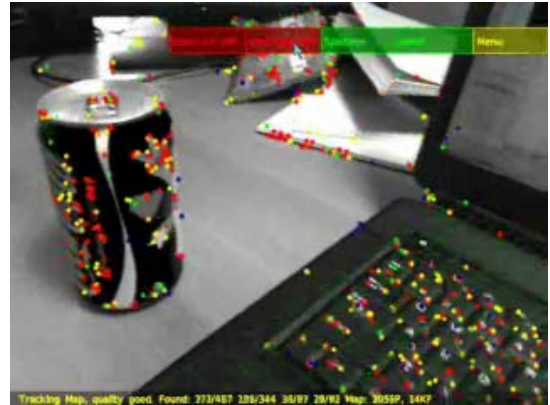


図3 抽出された特徴点群

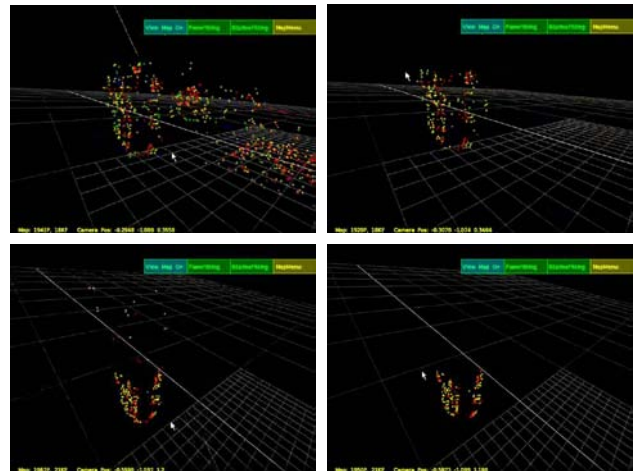


図4 点集合の指定

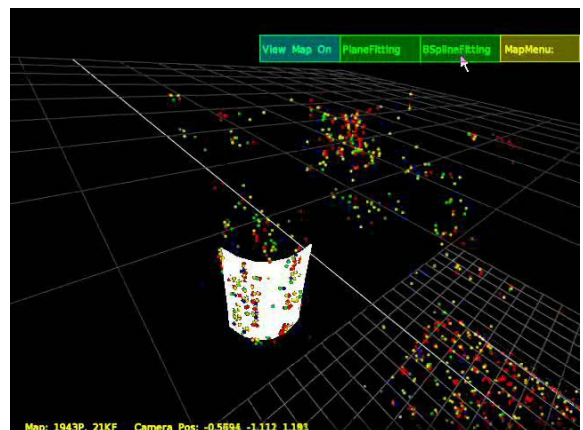


図5 曲面フィッティング

そのように曲面情報を得ることによって、曲面から構成される物体とのインタラクションが可能になり、既存ツールでは表現できなかった曲面形状を反映したリアルタイムでのオクルージョン(図 6)、衝突判定(図 7)、テクスチャマッピング(図 8)などが可能になっている。

図 6 では缶の側面の特徴点を指定し、曲面をフィッティングさせている。その曲面情報を用いることによって、右から左に移動する球が缶の後ろを通過する、という表現を可能にしている。

図 7 では、異なる位置から発射された球が楽譜のページに衝突後、それぞれ異なる方向へ跳ね返っている。これは曲面を使用しない限り表現できないものである。



図6 オクルージョン

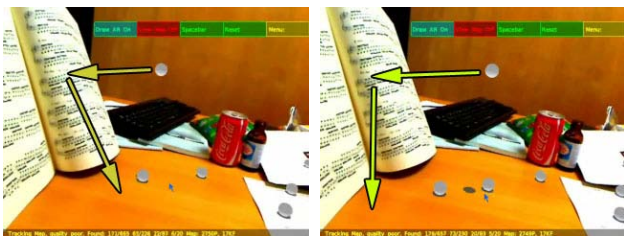


図7 衝突判定

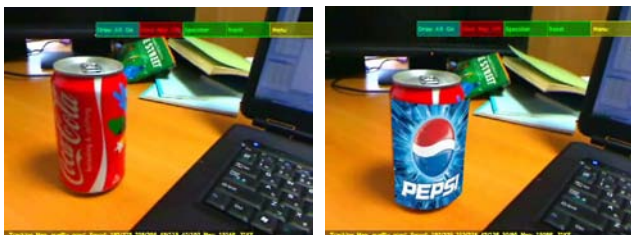


図8 テクスチャマッピング

## 5. 結論

ビデオ映像から特徴点を抽出して、リアルタイムに B スプライン曲面フィッティングをする手法を示した。また、リアルタイムに曲面を表現可能にすることによって、現実世界に即したオクルージョン、衝突などを表現することが可能であることを示した。

なお、今後の展開として、本研究では 3 次元情報を取得するデータは特徴点のみであるが、線の 3 次元データも取得すれば、映し出される物体の輪郭を得ることも可能になり、より適切な形状を計算できると考えられる。また、ビデオ映像に映し出される物体をプリミティブとして抽出することにより表現の幅が広がると考えられる。さらに、幾何形状としてメッシュを利用することも考えられる。メッシュを用いることによって曲面では表現しにくい、角張っている凸凹のある物体を表現することが可能である。

## 参考文献

- [1] A. van den Hengel, A. Dick, T. Thormählen, B. Ward, and P. H. S. Torr, VideoTrace: Rapid interactive scene modeling from video, ACM Transactions on Graphics, Volume 26 Issue 3, Article No. 86, July 2007
- [2] T. Thormählen, H.-P. Seidel, 3D-Modeling by Ortho-Image Generation from Image Sequences, ACM Transactions on Graphics, ACM SIGGRAPH 2008 papers, Volume 27, Issue 3, August 2008
- [3] Georg Klein and David Murray, Parallel Tracking and Mapping for Small AR Workspaces, Proc. International Symposium on Mixed and Augmented Reality, pp1-10, November 2007
- [4] Kato, H., Billinghamurst, M., Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System, Proc. International Workshop on Augmented Reality, pp.85-94, October 1999
- [5] [http://www.nvidia.com/object/physx\\_new.html](http://www.nvidia.com/object/physx_new.html)
- [6] Carlson, Nils, Surface fitting with NURBS: a gauss Newton with trust region approach, Proc. international conference on Applied mathematics, pp169-174, December 2008
- [7] David A. Forsyth, Jean Ponce, コンピュータビジョン, 大北 剛(訳), 共立出版株式会社, 東京, 2007
- [8] Jiang Dan, Wang Lancheng : An algorithm of NURBS surface fitting for reverse engineering. In Springer London(2006)
- [9] 斎藤 隆文ら, コンピュータグラフィックス, 藤代 一成ら(編), CG-ARTS 協会, 東京, 2006