

DSQモデルと時モデル空間勾配法 を用いた3次元動き推定の研究

上野 達矢 前田佐嘉志 鶴田直之
福岡大学大学院 工学研究科

実画像から対象物の位置、姿勢、形状に関する様々なパラメータを獲得し、そのパラメータを用いて仮想空間内に対象物を3次元アニメーションの形でマッピングするシステムの開発を目指している。そのために著者らはDSQ(Deformable SuperQuadrics)モデルと時モデル空間勾配法を用いた動き推定法を提案してきた。これにより非剛体物体の動きを実時間に近い速度で推定することができる。現在は、SIMDアーキテクチャを用いてさらに実行時間を短縮して実時間処理を目指している。本稿では、SIMDアーキテクチャの適用法を明らかにするため、従来手法の実行時間の内わけについて詳細な解析を行った。

3D Motion Estimation using DSQ Model and Temporal Model-Spatial Gradient Method

Tatsuya Ueno, Sakashi Maeda, Naoyuki Tsuruta

Graduate School of Engineering, Fukuoka University

We are developing a 3D animation system using the position, the posture and the shape parameters, which are estimated by image base modeling systems. To realize this system, a 3D-motion estimation method using DSQ (Deformable Super Quadrics) model and temporal model-spatial gradient method is proposed. By this method, motion of non-rigid objects can be estimated at the speed near to the normal video camera frame rates. Now, to reduce the estimation time to the range of normal video camera rates is aimed. In this paper, analysis detailed about an inner division of the execution time of the proposed method is shown.

1 序論

1.1 背景

近年、マルチメディア時代を迎えてコンピュータが画像情報をはじめ様々な情報を扱えるようになってきている。画像は膨大な情報を含んでおり、我々の日常生活で重要な情報伝達の媒体となっている。画像情報を用いて高いリアリティを持つ仮想空間を実現するためには、コンピュータを使った画像処理技術を基本とし、人間が使い易いコンピュータシステムを実現させるための様々な要素技術の研究を行っていかなければならない。その中、著者らは、非接触型のセンサであるビデオカメラからの動画像をもとに、実空間に存在する物体をシームレスに仮想空間に融合することにより、リアリティの高い仮想空間を手軽に構築するシステムを開発することを目指している。具体的なシステムとしては、実空間(多視点動画像)から対象物(剛体・非剛体物体)の位置、姿勢、形状に関する様々なパラメータを獲得(推定)し、そのパラメータを用いて仮想空間内に対象物を3次元アニメーションとしてマッピングするシステムを開発している。

一般に、2次元の画像から3次元の形状や動きを復元する問題は、何らかの事前情報がなければ解けない問題である。そこで、著者らは、問題毎に事前情報を付与するユーザの介入を認め、

仮想空間を提示するために必要な情報 = 事前情報 + 獲得情報

という枠の中で、必要以上に3次元情報の獲得が難しくならず、しかもユーザが手軽に事前情報を与えることができるようなシステムを開発する。

1.2 システムの構成と位置づけ

我々が開発しているシステムの全体図を図2に示す。本システムの構成要素は以下の通りである。

1. 多視点画像入力装置 (Multiple Cameras)
複数のカメラから画像を入力する。現状では、手動で視点を決定し、3視点を獲得している。
2. 初期フレームにおけるモデルフィッティング (Interactive 3D Modeler)
ユーザ介入の基で、多視点画像から半自動に物

体の位置・姿勢・形状を獲得する GUI ベースの形状モデラである。(図1) 現在はこのモデラを用いて、初期フレームにおける対象物の初期モデルを獲得している。

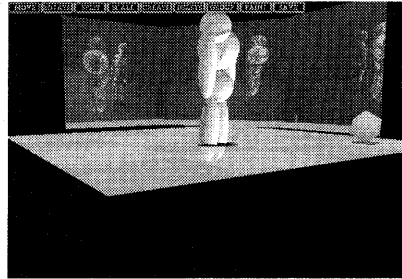


図 1: GUI によるモデルフィッティング

3. 動画像系列におけるパラメータ推定部 (Tracking System)

第2フレーム以降の物体の各フレームにおける対象物の位置・姿勢・形状のパラメータを推定する。

4. 仮想空間現実部 (Animation System)

推定されたシーンパラメータを元に、3次元アニメーションとして仮想空間内に表示する。

本研究は、上記のシステム構成要素のうち、3の改良の研究である。具体的には、動き推定における高速化である。本システムは、ユーザの介入を認め、ユーザとシステムがインタラクティブにモデリングを進めるので、ユーザへの負荷を減らすために実時間性の実現が必要不可欠となる。

2 3次元モデル

2.1 3次元モデルの記述法

3次元モデルとして、二次曲面を拡張した超二次曲面 (SuperQuadrics, 以下 SQ と呼ぶ) に、先細り (Tapering)、曲げ (Bending) の拡張を行った変形可能超二次曲面 (Deformable SuperQuadrics, 以下 DSQ と呼ぶ) を用いている。[4]

- SQ の定義

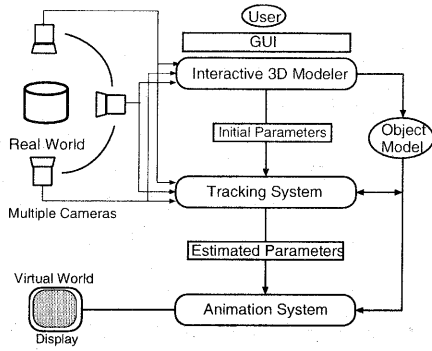


図 2: システム構成

SQ 曲面上の点を記述するために、極座標表現 (マテリアル座標系) を用い、その媒介変数を η, ω とする。モデルの中心を原点とした SQ の点 e の 3 次元座標は以下のように表される。

$$e(\eta, \omega) = \begin{pmatrix} e_1(\eta, \omega) \\ e_2(\eta, \omega) \\ e_3(\eta, \omega) \end{pmatrix} = a \begin{pmatrix} a_1 \cdot C_\eta^{e_1} C_\omega^{e_2} \\ a_2 \cdot C_\eta^{e_1} \cdot S_\omega^{e_2} \\ a_3 \cdot S_\eta^{e_1} \end{pmatrix} \quad (1)$$

ここで、

η ($-\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2}$)	マテリアル座標
ω ($-\pi \leq \omega \leq \pi$)	
a ($a \geq 0$)	スケールパラメータ
a_1, a_2, a_3 ($0 \leq a_1, a_2, a_3 \leq 1$)	アスペクト比
e_1, e_2 ($e_1, e_2 \geq 0$)	形状パラメータ
$C_\omega^{e_2}$	$\text{sign}(\cos \omega) \cdot \cos \omega ^{e_2}$
$S_\omega^{e_2}$	$\text{sign}(\sin \omega) \cdot \sin \omega ^{e_2}$

である。SQ モデルの例を図 3 に示す。

• DSQ の定義

SQ の点 $e = (e_1, e_2, e_3)^T$ を用いて、モデル中心を原点とする DSQ 上の点 $s = (s_1, s_2, s_3)^T$ の 3 次元座標は、以下のように表される。(以下、 η, ω は省略)。

$$s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix}$$

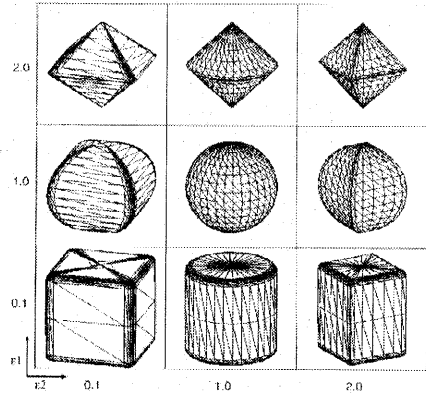


図 3: SQ モデルの例

$$= \begin{pmatrix} \left(\frac{t_1 e_3}{a a_3} + 1 \right) e_1 + b_1 \cos\left(\frac{e_3 + b_2}{a a_3} \pi b_3 \right) \\ \left(\frac{t_2 e_3}{a a_3} + 1 \right) e_2 \\ e_3 \end{pmatrix} \quad (2)$$

ここで、

t_1, t_2 ($-1 \leq t_1, t_2 \leq 1$)	Tapering パラメータ
b_1 (positive or negative)	Bending パラメータ
b_2 ($-1 \leq b_2 \leq 1$)	
b_3 ($0 < b_3 \leq 1$)	

である。DSQ の Bending(右) および Tapering(左) の形状変化の例を図 4 に示す。

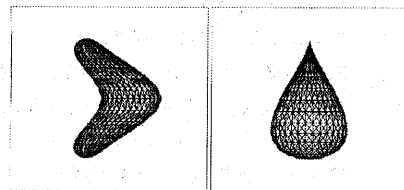


図 4: DSQ の形状変化の例。

3 動き推定の流れ

動き推定では、形状推定で求めた初期フレームにおけるモデルパラメータを初期状態として、過去

のフレームの推定結果をもとにして、時系列の順番に1フレームずつ推定を行う。

1. 推定するフレームの画像より追跡可能な画像特徴点を獲得する。
2. 時系列で一つ前のフレームの推定結果から、モデル表面上のモデル点と獲得した画像特徴点とを対応付け、モデルパラメータを推定するために用いられるモデルサンプル点を獲得する。
3. 獲得したモデルサンプル点を用いて、時モデル空間勾配法の拘束式(後述)をもとに、モデルパラメータを推定する。
4. モデルパラメータを推定結果に基づいて更新し、1へ戻って処理を繰り返す。

処理時間の大半を占める処理は2のサンプル点獲得と3のパラメータ推定の2つである。以下、それぞれについて述べる。

3.1 モデルサンプル点獲得の手順

画像から得られた画像特徴点に対し、それらに対応するモデル表面のマテリアル座標を求め、それらをモデルサンプル点とする。

1. 推定するフレームの画像から追跡可能な領域中央の特徴点を獲得する。
2. モデルの投影画像をZ-bufferアルゴリズムを用いて作成し、モデル表面の可視領域を決定する。
3. 2によって可視と判定されたモデルの各微小領域(ポリゴン)について、領域内に画像特徴点が存在するかを調べる。(図5)
4. 領域内全ての画像特徴点について、各点の2次元座標 (x, y) と対応するモデル表面上のマテリアル座標 (η, ω) をリストに加える。その際、マテリアル座標は画像特徴点を囲むポリゴンの3つの頂点の既知のマテリアル座標から近似を行うことにより求める。

この処理に要する処理時間は特徴点数に比例する。

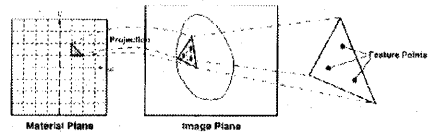


図5: モデルサンプル点獲得の流れ

3.2 時モデル空間勾配法によるモデルパラメータ推定

時モデル空間勾配法は、画像のオプティカルフローの算出に用いられる時空間勾配法の考え方を拡張したものであり、フレーム間の画像特徴の対応を陽に求めず、画像の明度の勾配をもとにモデルパラメータを推定する手法である。以下、その詳細について述べる。

今、物体が移動・変形しても物体表面の点 j のみかけの明るさ I_j は、隣接フレーム間では、変化しないと仮定する。今、時刻 t においてモデルパラメータが $\mathbf{p} = (p_1, p_2, \dots, p_n)$ であった物体が、時刻 $t + \Delta t$ においてモデルパラメータ $\mathbf{p} + \Delta \mathbf{p}$ に変化したとすると、先の仮定より、

$$I_j(\mathbf{p} + \Delta \mathbf{p}, t + \Delta t) = I_j(\mathbf{p}, t) \quad (3)$$

が成立する。

更に、モデルパラメータが \mathbf{p} のモデル表面上の点 j の画像座標 (x_j, y_j) を得る関数を $\mathbf{v}_j(\mathbf{p})$ とすると、式(3)は、

$$I(\mathbf{v}_j(\mathbf{p} + \Delta \mathbf{p}), t + \Delta t) = I(\mathbf{v}_j(\mathbf{p}), t) \quad (4)$$

と書ける。ただし、関数 $I(\mathbf{v}, t)$ は、時刻 t の画像座標 $\mathbf{v} = (x, y)$ における画素の明度を表す。

ここで、物体表面の明度分布が滑らかに変化すると仮定すると、式(4)の左辺をTaylor展開し、一次の項までで近似することにより、

$$\Delta p_1 \left(\frac{\partial I}{\partial x} \frac{\partial x}{\partial p_1} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial p_1} \right) + \dots +$$

$$\Delta p_n \left(\frac{\partial I}{\partial x} \frac{\partial x}{\partial p_n} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial p_n} \right) + \frac{\partial I}{\partial t} = 0 \quad (5)$$

を得る。

ここで、

$$\frac{\partial I}{\partial x} \frac{\partial x}{\partial p_k} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial p_k} = \frac{\partial I}{\partial p_k} \quad (6)$$

と簡略化することにより、式(5)から、

$$\Delta p_1 \frac{\partial I}{\partial p_1} + \dots + \Delta p_n \frac{\partial I}{\partial p_n} + \frac{\partial I}{\partial t} = 0 \quad (7)$$

を得る。式(7)は、画像の時間方向の勾配とモデル空間方向の勾配(以下、時モデル空間勾配と呼ぶ)とフレーム間のパラメータの変化分の関数を表した式である。以下、式(7)を「時モデル空間勾配の拘束式」と呼ぶ。

一般に、モデルパラメータの数は複雑であるので、明らかに単独の式(7)のみからではモデルパラメータを決定することはできない。しかし、画像中の同一部位の全画素に対し、パラメータ \mathbf{p} は共通であるという制約があるので、同一部位の投影領域に属する画素を $k(k \gg n)$ 個を選び、最小二乗法により、

$$\| \mathbf{A}_I \cdot \Delta \mathbf{p} - \mathbf{b}_I \| ^2 \quad (8)$$

を最小にする $\Delta \mathbf{p}$ を求める。[2]

実際には、求めた $\Delta \mathbf{p}$ を用いて新たに

$$\mathbf{b}_I = \mathbf{b}_I - \mathbf{A}_I \cdot \Delta \mathbf{p} \quad (9)$$

として式(8)を繰り返し計算することによって精度を高める。

ただし、

$$\mathbf{A}_I = \begin{bmatrix} \frac{\partial I_1}{\partial p_1} & \dots & \frac{\partial I_1}{\partial p_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial I_k}{\partial p_1} & \dots & \frac{\partial I_k}{\partial p_n} \end{bmatrix}, \Delta \mathbf{p} = \begin{bmatrix} \frac{\partial p_1}{\partial t} \\ \vdots \\ \frac{\partial p_n}{\partial t} \end{bmatrix},$$

$$\mathbf{b}_I = \begin{bmatrix} -\frac{\partial I_1}{\partial t} \\ \vdots \\ -\frac{\partial I_k}{\partial t} \end{bmatrix} \quad (10)$$

式(8)の最小解を求める処理は、MMX など整数演算しか使えない SIMD アーキテクチャでも十分な精度を保ったまま高速化できることが分かっている。

[1]

4 動き推定の処理時間測定

4.1 モデルサンプル点選択法の評価

従来の提案手法[2]ではパラメータ推定に有効なモデルサンプル点を選別するために、画像特徴点において空間微分の自己相関行列の固有値を計算し、その評価値の高いものから順番に選択している。そのため画像特徴点選択時間にかかなりの時間がかかり全体の特徴点選択時間としても時間がかかる。そこで、今回は式(8)を解く際に特異値分解法を使用し、特徴点選択は折込み済みと考え、特徴点選択処理を省いた。その精度評価(合成画像)を図6に示す。

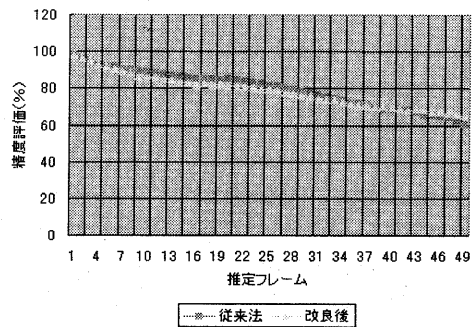


図6: 精度評価

図6から比較できるように従来法の画像特徴点選択の処理を省いても精度には影響しない。その時、モデルサンプル点の獲得時間を62.1%に短縮できた。

4.2 パラメータ推定時間の内訳

モデルサンプル点の獲得時間とパラメータ推定時間はともにモデルサンプル点数に比例する。本研究では、パラメータ推定部の並列化を検討しているのでその効果を見積もるために2つの処理時間の内訳を調べた。

4.2.1 モデルサンプル点獲得

モデルサンプル点獲得処理の実験方法としては、特徴点の最大数を600、1200、…、9600としていく。その時のモデルサンプル点の獲得時間を測定した。それぞれ測定を5回行った時の実行時間の中間データ3回を平均して求めた。

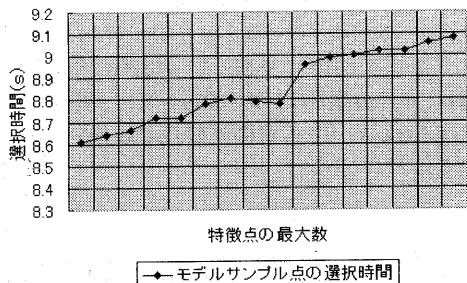


図 7: モデルサンプル点の獲得時間

図 7 は、特徴点選択処理のモデルサンプル点選択時間をグラフにしたものである。グラフよりモデルサンプル点選択時間の関係式は次のように導けた。

$$t_m = 0.03x \quad (\text{sec}) \quad (11)$$

ここで、 x は特徴点の最大数、 t_m はモデルサンプル点の選択時間である。

4.2.2 モデルパラメータ推定

モデルパラメータ推定処理の実験方法としては、特徴点の最大数を600、1200、…、9600としていく。まず、その時のモデルパラメータの推定時間を測定した。それぞれのモデルパラメータの推定時間は、それぞれ測定を5回行った時の実行時間の中間データ3回を平均して求めた。

図 8 は、モデルパラメータの推定時間をグラフにしたものである。モデルパラメータの推定時間の関係式を次のように導けた。

$$t_p = 0.05x \quad (\text{sec}) \quad (12)$$

ここで、 x は特徴点の最大数、 t_p はモデルパラメータの推定時間である。

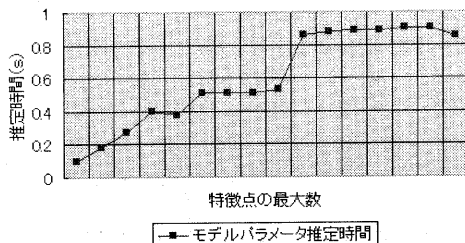


図 8: モデルパラメータの推定時間

この結果から、 $\beta (= 0.05) > \alpha (= 0.03)$ が成立し、モデルパラメータの推定時間は、特徴点選択時間より、時間に対する増加量が多いことが分かる。

更に、その時の最小二乗法を行なう回数に対するモデルパラメータの推定時間を測定する。

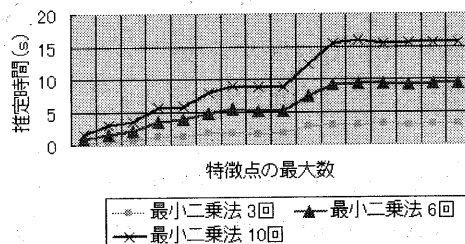


図 9: 最小二乗法の回数によるモデルパラメータの推定時間

図 9 は、最小二乗法の回数によるモデルパラメータの推定時間をグラフにしたものである。

以上の結果より実用的な仕様での実行時間の内訳を算出することができる。例えば、特徴点の最大数を6600、最小二乗法を行なう回数を10回とした場合の実行時間の内訳を表 1 に示す。使用した計算機は pentiumIII である。

表 1: 動き推定の処理時間の割合

	特徴点 選択時間	モデルパラメータ 推定時間
処理時間の割合	16.5(%)	28.7(%)
処理時間	9.63(s)	15.8(s)

MMX アーキテクチャを実装し高速化を図るほうが効率が良いことが分かる。特徴点の最大数を 6600、特徴点どうしの最短距離を 3、最小二乗法を行なう回数を 10 回とした場合、モデルパラメータの推定時間は、15.8 (s) であり、実際に MMX アーキテクチャが実装できた場合、モデルパラメータの推定時間は、1.96 (s) の時間短縮が見込まれ、全体の時間としては、55.08 (s) から 25.1% の時間短縮が見込まれる。

5 結論

動き推定の高速化を図るために、MMX アーキテクチャの実装を提案した。その MMX アーキテクチャを実装にするにあたり、動き推定の 2 つの処理について時間測定を行った。その結果、実行時間の割合がモデルパラメータ推定時間の方が大きいことが分かった。この部分に並列処理を行うことで、実行時間の高速化が見込まれ、我々の目指す実時間処理に近づく。今後は、MMX アーキテクチャを実装するにあたり、モデルパラメータ推定処理部分での整数化と精度評価を行い、どこまで精度を落とさずに整数データ型を落とせるのかを検討し実装する。

参考文献

- [1] 川島 護: “MMX アーキテクチャを用いた時空間勾配法の高速化に関する研究”, 福岡大学工学部電子工学科卒業論文, 2000.
- [2] T. Nunomaki, S. Yonemoto, N. Tsuruta, R. Taniguchi, Proc. of 4th Japan-France Congress and 2nd Asia-Europe Congress on Mechatronics, 2 (1998) 540 – 545
- [3] T. Nunomaki, S. Yonemoto, D. Arita, R. Taniguchi, N. Tsuruta, Proc. of 1st International Workshop on Articulated Motion and Deformable Objects (2000) 72 – 82.
- [4] D. Metaxas and D. Terzopoulos: “Shape and Nonrigid Motion Estimate through Physics-Based Synthesis,” IEEE trans. on PAMI, Vol.15, No.6, pp.580-591.1993
- [5] 熊谷 昭彦: “3次元非剛体物体の非接触型モーションキャプチャに関する研究”, 福岡大学大学院修士論文, 2001.