

## GPUを使った高速高精度な平面トラッキング

伊藤 栄介<sup>†1</sup> 岡谷 貴之<sup>†1</sup> 出口 光一郎<sup>†1</sup>

本研究では、Malisらの提案したESMによる画像追跡手法を高速かつ高精度に実現することを目的とし、GPUに実装する方法を述べる。ESMは追跡対象全ての画素を使った計算である為、特徴点がり難い画像では有効な手法である。ESMは元々高速な計算が可能となるように設計されているが、ESMの並列計算可能な部分に着目し、GPUの特性を生かしたメモリの使い分けや並列計算をすることで、さらに高速なトラッキングを実現した。加えて、テスト画像を用いて既存の手法との計算時間およびトラッキング性能の比較を行った。

## GPU based high-speed and high-precision visual tracking

EISUKE ITO,<sup>†1</sup> TAKAYUKI OKATANI<sup>†1</sup>  
and KOICHIRO DEGUCHI<sup>†1</sup>

This paper presents a method of implementing the ESM visual tracker proposed by Malis et al. to realize fast and accurate visual tracking. The ESM tracker is effective especially for the images in which feature points are difficult to obtain, since it uses entire image pixels of the target image region. It is originally designed so as to enable fast tracking, and our GPU implementation enables yet faster tracking; we focus on the parts of the algorithm that can be processed in a parallel fashion, and implement it by appropriately using the memory system of a GPU based on its properties and performing parallel computation. We also show comparison between the GPU and a CPU implementations of the ESM algorithm in terms of computation time and tracking performance.

<sup>†1</sup> 東北大学情報科学研究科

Graduate School of Information Science, Tohoku University

### 1. はじめに

物体を視覚的に追跡することはコンピュータビジョンの主要な課題の一つである。監視カメラのような固定カメラにおいては、視覚的に追跡することで物体の観察や監視が可能となる。また、カメラを搭載したロボットにおいては、得られた情報からロボットをどのように運動させるかを決定するための情報を得られる。このように、視覚的な追跡を実現することにより、監視カメラや、ロボット、乗り物の制御といった様々なアプリケーションへの適用が期待できる。

コンピュータビジョンによる物体追跡手法は大きく分けて、特徴ベーストとアピアランスベーストの二つに分類される。特徴ベーストの方法では輝度勾配のエッジや特徴点を用い画像間のマッチングを行う。一方、アピアランスベーストの方法では画像間の輝度の差を小さくする領域を見つけることで追跡領域を推定する。従来からある計算手法（ガウス・ニュートン法など）では時間がかかり、上記のアプリケーションをリアルタイムで実行することが困難であった。近年ではMalisら<sup>4)</sup>の提案したESM(Efficient Second-order Minimization)と呼ばれる手法がこの問題に対して有効であると知られている。ESMを使うことで従来の手法に比べ高速高精度な平面パターンの追跡が可能となったが、画像サイズが大きくなるほど計算量が二乗のオーダーで増大するため、更なる工夫が必要となる。ESMの計算には、ピクセル毎の計算など並列計算可能な部分が多く存在するため、本研究ではMalisらの手法を改良し、GPU(Graphics Processing Unit)を用いた高速トラッキング手法の実現を目指す。

### 2. ESMの原理

ESMではカメラや対象物体が移動する前の画像から後の画像へのホモグラフィ（二次元射影変換）を求めることで、トラッキング対象がどこに移動したかを推定している。本節では、この位置・姿勢の推定に用いたホモグラフィとその計算方法について述べる。加えて、照明の変化に対応したESMの理論も言及する。

#### (1) ホモグラフィ

一つのカメラと追跡対象（平面パターン）を考え、対象が移動したときの像の変化をモデル化する<sup>5)</sup>。カメラが移動前と移動後で同じ対象を撮影すると、図1のように追跡対象平面のある一点 $\mathbf{P}$ は、2枚の画像中のそれぞれ $\mathbf{p}^*$ と $\mathbf{p}$ に投影される。このときカメラ自身の座標系において、 $\mathbf{p}^*$ と $\mathbf{p}$ は $3 \times 3$ 行列 $\mathbf{H}$ によって対応付けられる。ここで、移動前のカメ

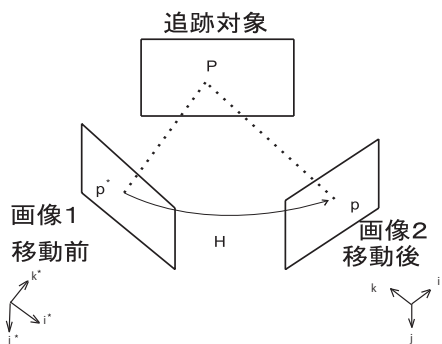


図 1 平面を投影した 2 枚の画像.  
Fig. 1 Projection model between two images of a plane.

ラ座標と移動後のカメラ座標, そしてホモグラフィをそれぞれ

$$\mathbf{p}^* = (u^*, v^*, 1)^T \quad (1)$$

$$\mathbf{p} = (u, v, 1)^T \quad (2)$$

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \quad (3)$$

と置くと, 射影変換は次の式で表される.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \simeq \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} u^* \\ v^* \\ 1 \end{pmatrix} \quad (4)$$

式 (4) の右辺に表れる  $3 \times 3$  行列に定数倍の不定性があり, 要素数は 9 だが推定しなければならないパラメータ数は 8 である. ESM では, 定数倍の不定性を  $\det(\mathbf{H}) = 1$  となるようなパラメータを設定することで除去している. このように決定された  $\mathbf{H}$  によって, 具体的に式 (4) から座標を決定する計算を  $w(\mathbf{H})$  とすると

$$\mathbf{p} = w(\mathbf{H})(\mathbf{p}^*) = \begin{bmatrix} \frac{g_{11}u^* + g_{12}v^* + g_{13}}{g_{31}u^* + g_{32}v^* + g_{33}} \\ \frac{g_{21}u^* + g_{22}v^* + g_{23}}{g_{31}u^* + g_{32}v^* + g_{33}} \\ 1 \end{bmatrix} \quad (5)$$

と表される. したがって, 移動前と移動後の対象平面は式 5 によって一対一で対応し, ホモグラフィを推定することができれば画像上のどこに追跡対象があるか計算できる.

## (2) ESM の計算手法

ここではホモグラフィの推定方法と, その計算に用いた ESM について述べる. 図 1 において,  $\mathbf{p}^*$  と  $\mathbf{p}$  における画像の輝度をそれぞれ  $I^*(\mathbf{p}^*), I(\mathbf{p})$  とする. ここで照明などの変化がないと仮定すると, 対応する点での輝度値は一致するので

$$I^*(\mathbf{p}^*) = I(\mathbf{p}) \quad (6)$$

が成り立つ. ここで, 1 フレーム目の追跡すべき正方領域を  $q$  ピクセルのテンプレートとすると, テンプレート領域の輝度値と一致する 2 フレーム目の画像領域はホモグラフィを用いて表すことができ

$$I^*(\mathbf{p}_i^*) = I(w(\mathbf{H})\mathbf{p}_i^*) \quad (i = 1, 2, \dots, q) \quad (7)$$

となる. 推定したい値はホモグラフィ  $\mathbf{H}$  であり, この行列は追跡領域が移動する場合, フレーム毎に刻々と変化していく. そこで, フレーム毎に更新される推定値を  $\hat{\mathbf{H}}$  とし, その更新される増分を  $\mathbf{H}(\mathbf{x})$  ( $\mathbf{H}$  の自由度は 8 であるためパラメータ  $\mathbf{x}$  は要素数 8 のベクトルである) と置くと,  $\mathbf{H}$  が  $\hat{\mathbf{H}}$  と近似できる.  $\hat{\mathbf{H}}$  の初期値を  $3 \times 3$  の単位行列とし, 追跡対象が微小移動したホモグラフィの増分を  $\mathbf{H}(\mathbf{x})$  とすることで式 7 は以下のように変形できる.

$$y_i(\mathbf{x}) = I(w(\hat{\mathbf{H}}) \circ w(\mathbf{H}(\mathbf{x}))\mathbf{p}^*) - I^*(\mathbf{p}^*) = 0 \quad (8)$$

$\mathbf{y}(\mathbf{x})$  を, 画素ごとの輝度値の差分を格納した  $q \times 1$  のベクトルとすると

$$\mathbf{y}(\mathbf{x}) = [y_1(x) \ y_2(x) \ \dots \ y_q(x)]^T \quad (9)$$

と置くことができる. このベクトルのノルムを 0 に近づけるようなパラメータ  $\mathbf{x}_0$  を求めることで  $H(\mathbf{x})$  を計算することができ,  $\hat{\mathbf{H}}$  を更新することで求めたいホモグラフィを得ることができる.  $\mathbf{x}_0$  の求解は非線形最適化問題であり, これには様々な解法が存在するが, リアルタイムで実行しなければならない制約があるため, パラメータ  $\mathbf{x}$  に関して線形化した後に, 反復法を使うことで解を求める手法が一般的である.

ガウス・ニュートン法などの一般的な手法では, 効率的に  $\mathbf{y}(\mathbf{x})$  の二階の項まで近似することは困難であり, 解の収束性が低くなるといった問題も存在した. ここで ESM の手法では

ヘッセ行列を計算することなく、効率的に二階の項まで近似することに注目している。式9のノルムを最小化するパラメータ  $\mathbf{x}_0$  は式10を解くことで得られる。

$$\mathbf{J}_{\text{esm}}\mathbf{x}_0 = -\mathbf{y}(0) \quad (10)$$

ただし、 $\mathbf{J}_{\text{esm}}$  の  $i$  列目の成分は以下の式で表される。

$$\mathbf{J}_{\text{esm}} = \frac{1}{2}(\mathbf{J}_{\mathbf{I}^*} + \mathbf{J}_{\mathbf{I}})\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{H}} \quad (11)$$

ここで、 $\mathbf{J}_{\mathbf{I}^*}$  と  $\mathbf{J}_{\mathbf{I}}$  はそれぞれテンプレート画像と変換後画像の空間微分、 $\mathbf{J}_{\mathbf{w}}$  はテンプレート画像の座標情報、 $\mathbf{J}_{\mathbf{H}}$  はリー代数の基底を並び変えた定数行列、 $\mathbf{y}(0)$  はテンプレート画像と変換後画像の差分となっている。これらのうち、 $\mathbf{J}_{\mathbf{I}^*}$  と  $\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{H}}$  については値が一定であり、事前の計算により得られる。そのためトラッキング実行時には  $\mathbf{J}_{\mathbf{I}}$  と  $\mathbf{y}(0)$  のみ計算すればよい。

### (3) 光の変化への対応

ここまで述べた手法は、追跡対象の平面パターンが微小変形したとき、輝度の差の二乗和を最小にする近傍領域を計算していた。そのため弱いノイズがあった場合でもトラッキングが可能であるが、局所的な光の反射などには非常に影響を受けやすく、誤った領域を推定してしまう問題があった。ここではどのように光の影響を ESM の計算に適用するかを述べる。光の反射特性のモデルは、拡散反射、鏡面反射、環境光の変化、そしてカメラのバイアスによって決定され、それは式12とおける。

$$I'(\mathbf{p}_i; \alpha_i, \eta_i, \beta) = \alpha_i I_d(\mathbf{p}_i) + \eta_i I_s(\mathbf{p}_i) + \beta \quad (12)$$

ここで任意の物体を追跡する際、拡散反射や鏡面反射のパラメータは未知である為、Malisら<sup>2)</sup>は拡散反射と鏡面反射の項を一つにまとめた式13を用いて推定を行っている。

$$I' = \gamma I + \beta \quad (13)$$

さらに、平面パターンは一律な光の影響を受けているとは限らない為、図3のように追跡領域のピクセルを数分割し、それぞれの領域に異なる  $\gamma$  を置いて推定を行う必要がある。

画像を  $n$  分割した場合、ESM で推定すべきパラメータ数は、ホモグラフィの自由度8、画像分割の  $\gamma$  の  $n$ 、そして環境光とカメラバイアスの影響を考慮した  $\beta$  の1、合計して  $n+9$  である。このパラメータを  $\theta = [\mathbf{x}^T, \tilde{\gamma}^T, \tilde{\beta}]$  とする。分割した  $j$  番目のブロックにおける  $i$  番目のピクセルの輝度を

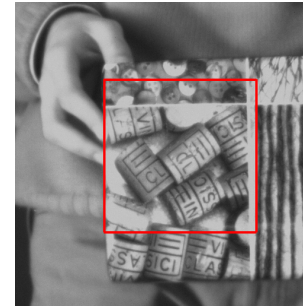


図2 追跡対象の平面パターン。  
Fig.2 Tracking image of a plane

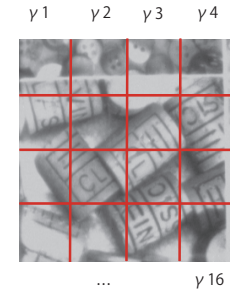


図3 光の影響を考慮して画像を分割したもの。  
Fig.3 Image division considering arbitrary illumination changes

$$I'(w(\mathbf{p}_{ij}^*; \mathbf{H}); \gamma_j, \gamma) = \gamma_j I(w(\mathbf{p}_{ij}^*; \mathbf{H})) + \beta \quad (14)$$

と定義すると、最小化すべき関数は以下のように書き換えることができる。

$$f(\theta) = \frac{1}{2} \|I'_{ij} - I_{ij}^*\|^2 = \frac{1}{2} \|\mathbf{d}(\theta)\|^2 \quad (15)$$

式15は非線形最適化問題であるため、一般的な ESM の手法と同様に線形化した後に反復法で  $\theta$  を導出する。

### 3. GPU を用いた ESM の実装

今回は GPU として、NVIDIA 社の GeForce GTX295 を用いた。NVIDIA 社の GPU メモリにはいくつかの階層と種類がある。我々はメモリの特性を考慮し、以下のように使い分けることで高速に ESM の計算を実行した。

- 事前に計算できる  $\mathbf{J}_{\mathbf{I}^*}$  と  $\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{H}}$  を画像形式でテクスチャメモリに入れ、読み込みを高速化する。

- 最小二乗解を求める連立方程式  $\mathbf{J}_{\text{esm}}\mathbf{x} = -\mathbf{y}(0)$  を解く際、シェアードメモリを使い  $\mathbf{J}_{\text{esm}}^T \mathbf{J}_{\text{esm}}\mathbf{x} = -\mathbf{J}_{\text{esm}}^T \mathbf{y}(0)$  とに直すことで高速にデータを圧縮する。

ESM の手法で必要な画像の微分や差分はピクセル毎の計算であり、容易に並列化が可能である。これはテクスチャメモリを使うことで高速に実行した。また、正規方程式に直す際に使ったシェアードメモリは容量が小さいため、 $\mathbf{J}_{\text{esm}}$  の行列をブロックに分割し並列計算

を行った。

#### 4. 実画像でのトラッキング実験

ここでは事前に用意された 10 枚の画像を用いて 128×128 ピクセルの正方領域を追跡する。その際に、CPU のみを使った計算時間と、CPU と GPU を使った計算時間とを比較する。次に高速カメラと GPU を用いて 128×128 の領域をリアルタイムに追跡する。最後に任意の光の影響を考慮した ESM についての結果画像を示す。

##### 4.1 実装環境

今回 ESM を実装した環境について示す。10 枚の事前に準備されている画像を用いたトラッキング時間の比較では、CPU は Intel Core i7 920、クロック周波数は 2.67GHz、GPU は NVIDIA 社製の GeForce GTX295 を使用した。CPU の計算では Intel 社から提供されている Intel Performance Primitives や Intel Math Kernel Library を用いて最適化を行った。GPU と高速カメラを用いたリアルタイムトラッキングの実験では、CPU は Intel Core Xeon X5460、クロック周波数は 3.16GHz、GPU は NVIDIA 社製の Quadro FX 5600、カメラは View Plus 社製の Lightning を秒間 500 フレームの撮影に使用した。

##### 4.2 実験結果

図 4 に追跡を実行した画像と追跡領域の画像を、表 1 に CPU のみで行った計算時間と、CPU と GPU を用いて行った計算時間の比較を示す。

次に、図 5 に高速カメラを用いたリアルタイムトラッキング結果を、図 6 に任意の光の影響を考慮した ESM の結果を示す。表 1 から CPU と GPU の長所を組み合わせた提案手法では CPU のみの計算時間と比べ約 43% 速くなったことがわかる。ESM は反復法的一种であり、フレーム間で繰り返しパラメータを推定する必要がある。そのため、GPU と CPU を組み合わせた場合では CPU のみより、最大で 20ms 速く次のフレームの計算に移ることができる。フレーム間で平面パターンの変形が小さければ、安定して物体の追跡ができるという利点もある。図 5 から、平面パターンに平行移動やカメラの光軸方向回転を加えた場合は正しく推定できているが、奥行き方向の回転を加えた場合では推定領域が歪んでしまっていることがわかる。この原因として、計算モデルに照明の影響を考慮していなかったことが挙げられる。平面パターンを傾けた際に発生する光の変化によって、フレーム間で対応する点の輝度が一致しなくなり、誤った推定をしたのだと考えられる。

光の影響を考慮した ESM では画像分割数を 16 に設定し、求めるべきパラメータ数をホモグラフィの自由度  $x$  の 8、領域ごとの光の係数  $\gamma$  の 16、バイアス  $\beta$  の 1 の合計 25 とした。

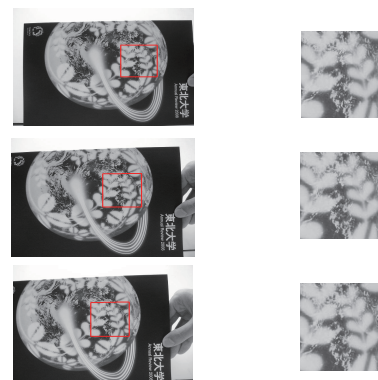


図 4 事前に用意された画像でのトラッキング。

Fig. 4 Tracking images that were prepared beforehand.

表 1 計算時間の比較  
Table 1 Comparison of calculation time.

計算方法	$J_{esmx} = -y(0)$ の計算時間 [ms]	1 回のパラメータ推定 に必要な時間 [ms]
CPU+GPU	0.58	0.81
CPU	0.86	1.43

図 6 の光の影響を考慮した ESM のトラッキングでは、推定のずれが小さくなり追跡が成功していることがわかる。この実験で用いた光の変化は追跡領域全体に作用していた為、分割数を減らした場合でも追跡することが可能であったが、精度の問題より 16 分割で行っている。分割数を増やすと細かいブロックごとに光の影響を推定することが可能となり、局所的な光の変化やノイズにも対応することができる。ただし、パラメータ数と計算時間はトレードオフの関係にあるため、リアルタイム性を考慮するならばパラメータ数の上限にも注意しなければならない。

#### 5. おわりに

本研究では、高速高精度な平面パターンの追跡をリアルタイムで行うことを目標とした。その有効な手法として Malis らの提案した ESM が良く知られており、これを GPU 上で実装した。近年 GPU を利用することで並列計算能力を劇的に増大させる GPU コンピューティングに注目が集まっている。汎用的な計算に適用できる環境が整ってきたことや CPU よりも演算ユニット数が多いことなどから、より簡単に大規模な並列計算をすることが可能となっている。ESM の計算においても並列計算が多数含まれており、GPU で実装することで 128×128 のピクセルサイズで追跡した場合、約 43% 計算速度が速くすることが可能になった。ここでは GPU のメモリの種類や特性を考慮し、マルチコアの性能を十分発揮でき

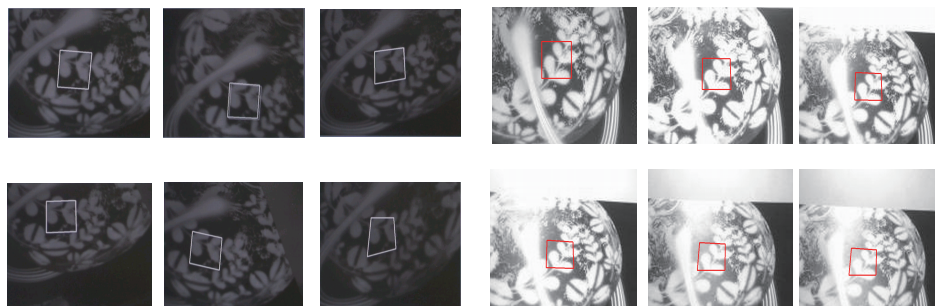


図 5 高速カメラを用いたトラッキング.

Fig.5 Results of real-time tracking when used with a high-speed camera.

図 6 光の影響を考慮した ESM のトラッキング.

Fig.6 Image tracking considering arbitrary illumination changes.

るように留意した. それによってリアルタイムにトラッキングする実験では, 高速な追跡を実現した. さらに, 光の影響や画像のノイズによって精度良く追跡領域を推定できない場合にも追跡を可能とする, 光の影響を考慮した計算モデルについても実現した.

### 参 考 文 献

- 1) S.Benhimane,E.Malis: Homography-based 2D Visual Tracking and Servoing,The International Journal of Robotics Research Vol.26,No.7,July 2007,pp.661-676
- 2) G. Silveira, E. Malis: Real-time Visual Tracking under Arbitrary Illumination Changes, In IEEE Computer Vision and Pattern Recognition, Minneapolis, USA, June 2007,pp.1-6
- 3) E. Malis: An efficient unified approach to direct visual tracking of rigid and deformable surfaces, In IEEE/RSJ International Conference on Intelligent Robots Systems, San Diego, USA, October 2007,pp. 2729 - 2734
- 4) E. Malis: Improving vision-based control using efficient second-order minimization techniques, In IEEE International Conference on Robotics and Automation, Volume 2, Pages 1843-1848, New Orleans, USA, April 2004
- 5) 出口光一郎, ロボットビジョンの基礎, コロナ社, 2000
- 6) NVIDIA.NVIDIA CUDA Programming Guide,Version 2.3.1,2009
- 7) NVIDIA.NVIDIA CUDA C Programming Best Practices Guide CUDA Toolkit 2.3