

全方位カメラを用いた3次元環境計測における GPUを用いた特徴追跡の高速化と動物体の検知

○ 渡辺 和史, 川西 亮輔, 山下 淳, 金子 透

○ Kazushi Watanabe, Ryosuke Kawanishi, Atsushi Yamashita and Toru Kaneko

静岡大学工学部機械工学科, {f0710157, f5945016, tayamas, tmtkane}@ipc.shizuoka.ac.jp

我々はこれまで全方位カメラを用いた周囲環境計測に関する研究を行ってきた。しかし我々の従来手法では、特徴追跡における処理時間が大きく、オフラインで処理を行っていた。そこで本研究では、GPUを用いて特徴追跡の高速化を行う。また、我々の従来手法では、動物体がカメラに接近する場合、動物体がカメラから十分遠ざかるまで手動で撮影を停止させる必要があった。そこで本研究では、動物体が存在する場合でも正常に周囲環境のオンライン計測を完了できる計測システムを提案する。また、提案する計測システムに必要な動物体検知手法を提案する。

<キーワード>特徴追跡, GPU, 動物体検知, 3次元計測, 全方位カメラ

1. 緒言

移動ロボットが自律的に活動する際に必要不可欠な機能として、目的地へ移動するための経路生成や環境中の自己位置同定などが挙げられる。経路生成や自己位置同定を行うためには、ロボットの周囲に存在する物体の形状や位置を把握できる地図が必要となる。このとき、地図が持つべき情報としては、3次元の幾何学的な情報やテクスチャ情報が考えられる。しかし、ロボットが常に環境の3次元地図を持っているとは限らない。また、自律的に活動するロボットは環境の変化に対応できることが望まれる。したがって、ロボット自身が周囲環境を計測し、地図を生成する必要がある。

環境計測を行うためのセンサとして、広い視野を持つ全方位カメラが考えられる。全方位カメラは、周囲360度を1枚の画像に撮像できるという特性により、環境計測に有効であることが示されている[1]。我々はすでに全方位カメラを用いた環境計測手法を提案している[2]。我々の従来手法では1台の全方位カメラを搭載した移動ロボットを用い、Structure from Motionの枠組みによる環境計測を行う。環境計測を行う際は、まず1台のカメラを移動させながら撮影した画像列から特徴点の抽出・追跡（特徴追跡）を行う。そして特徴点の画像間における対応関係から、被写体の計測とカメラ運動推定を同時に行う。

しかし文献[2]では、すべての処理をオフラインで構成していた。ここで、本研究におけるオフラインでの計測とは、処理を始めてから結果を出力するまでの間に人間の操作が必要となる環境計測のことである。ロボットが自律的に活動するためにStructure from Motionの処理の全てをオンラインで行えることが望ましい。

我々の従来手法では、処理時間の中でも特徴追跡の占める割合が大きく、オンライン化の障害であった。そこで本研究では特徴追跡を、GPU (Graphics Processing Unit) を用いた並列演算によって高速化する。

また特徴追跡において、撮影中に動物体がカメラに接近すると静止物体上の特徴点が画像上から大量に損失する。静止物体上の特徴点が画像上から大量に損失すれば、計測結果が実際の周囲環境と違ったものになるため問題である。我々は従来、動物体が接近する場合、動物体がカメラから十分遠ざかるまで、手動で撮影を停止させていた。そのため文献[2]では、動物体が存在する環境でオンライン計測を行っていなかった。そこで本研究では、動物体が存在する環境においても、手動操作を必要とせずに正常な周囲環境の計測を完了できる計測システムを提案する。また、提案する計測システムでは、動物体がカメラに接近する前に、視野内に動物体が存在する

かどうかを検知する必要がある．そこで，本研究では，視野内に動物体が存在するかどうかを検知する手法を提案する．



図1 全方位カメラ（左）と取得画像（右）

2. 処理の概要

本研究で使用する全方位カメラは，ビデオカメラの前方に双曲面ミラーを取り付けたものを用いる（図1）．カメラの内部パラメータおよびレンズの歪曲収差は，あらかじめキャリブレーションによって求める．

まず，我々の提案する3次元計測手法について概要を述べる．全方位カメラは車輪移動型ロボットに搭載する．ロボットは常に移動し，同時にカメラは動画像を取得する．撮影される動画像から，画像列で特徴追跡を行い，対応点を取得していく．ここで，対応点に含まれるアウトライアを検出する．検出されたアウトライアを除いた対応点を用いて，改めてカメラ運動推定と3次元計測を行う．

本研究では，動物体が存在する環境においても，手動操作を必要とせず正常な周囲環境の計測を完了できる計測手法を提案する（図2）．ここで前提として，動物体からカメラにぶつかってくることはないとする．本手法では，対応点のアウトライアを検出した時点で，提案する動物体検知手法により動物体が視野内に存在するかどうかを判定する．動物体を検知しなければ，移動および計測を継続する．動物体を検知した場合，追跡された特徴点情報を保存し，ロボットは停止する．ロボットが停止した状態でカメラは撮影を継続し，フレーム間で大きく座標を変化させた特徴点が存在するときには動物体が視野内に存在すると判定する．動物体が十分遠ざかったと判定すれば，保存した特徴点を用いて計測を再開する．

3. 環境計測

3.1 基本行列 E の算出

カメラから3次元空間中の点に向かう光線を光線

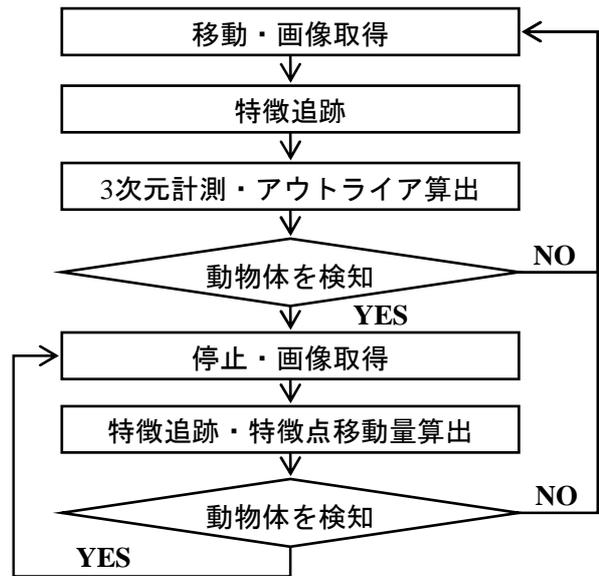


図2 提案する計測システム

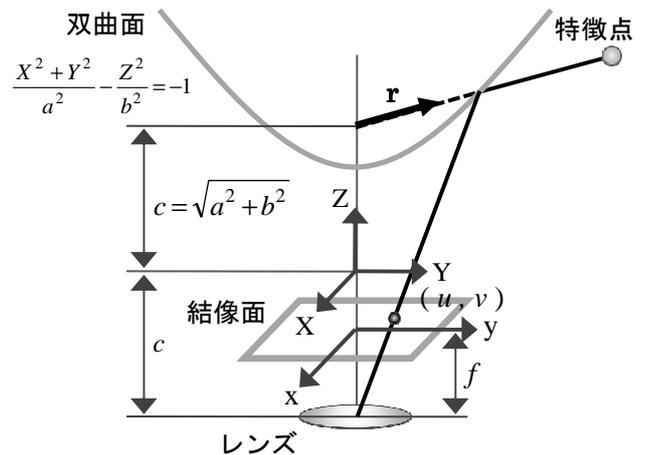


図3 光線ベクトルの算出

ベクトル $\mathbf{r} = [x, y, z]^T$ とする．本研究で用いる全方位カメラはカメラ前方に双曲面ミラーを装着したものである．光線ベクトル \mathbf{r} は，双曲面のミラー側の焦点から3次元空間中の点へ向かう光線が双曲面上で反射する点へのベクトルで表せる（図3）．

特徴点の画像座標 $[u, v]^T$ から式(1)，(2)により \mathbf{r} を求める．ここで a, b, c は双曲面のパラメータであり， f は焦点距離である．

$$\mathbf{r} = \begin{bmatrix} su \\ sv \\ sf - 2c \end{bmatrix} \quad (1)$$

$$s = \frac{a^2 \left(f \sqrt{a^2 + b^2} + b \sqrt{u^2 + v^2 + f^2} \right)}{a^2 f^2 - b^2 (u^2 + v^2)} \quad (2)$$

そして、後の演算で特徴点ごとの偏差を生じさせないように $\|\mathbf{r}\|$ の大きさを単位ベクトルに変換し、正規化する。

計測に用いる2画像間における対応点の光線ベクトル $\mathbf{r}_{1,i} = [x_i, y_i, z_i]^T$, $\mathbf{r}_{2,i} = [x_{2,i}, y_{2,i}, z_{2,i}]^T$ を用いて幾何関係によりカメラ間の位置・姿勢情報からなる行列を求める。この行列は基本行列 \mathbf{E} と呼ばれ、式(3)で表される。式(3)を変形すると式(4)が得られる。

$$\tilde{\mathbf{r}}^T \mathbf{E} \tilde{\mathbf{r}} = 0 \quad (3)$$

$$\mathbf{u}^T \mathbf{e} = 0 \quad (4)$$

ただし、

$$\mathbf{u} = [x_1x_2, y_1x_2, z_1x_2, x_1y_2, y_1y_2, z_1y_2, x_1z_2, y_1z_2, z_1z_2]^T$$

$$\mathbf{e} = [e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33}]^T$$

(e_{ab} は \mathbf{E} の a 行 b 列成分)

基本行列 \mathbf{E} は8点以上の対応する光線ベクトル対に対する方程式、すなわち式(5)を解いて求める。

$$\min \|\mathbf{Ue}\|^2 \quad (5)$$

ただし、 $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]^T$ である。 \mathbf{e} は $\mathbf{U}^T\mathbf{U}$ の最小固有値に対応する固有ベクトルとして与えられ、基本行列を変形したベクトル \mathbf{e} が求まる。ここで、 n は特徴点の数とする。

3.2 アウトライアの除去

特徴点追跡によって得られた対応点は全てが正しく対応しているとは限らない。対応を誤っている点が計測に含まれると位置・姿勢の推定に悪影響を及ぼす。そこで、位置・姿勢に悪影響を及ぼす点を外れ値として除去する。これにはRANSAC (RANdom Sample Consensus) を用いる[3]。まず基本行列 \mathbf{E} が決定する最小の対応点数である8個の特徴点をランダムで抽出し \mathbf{E}_{rand} を求める。そして光線ベクトル $\mathbf{r}_{1,i}$, $\mathbf{r}_{2,i}$ に対して式(6)を満たす特徴点の数を k とする。

$$|\mathbf{r}_{2,i} \mathbf{E}_{\text{rand}} \mathbf{r}_{1,i}| < q \quad (6)$$

ただし q は閾値である。この \mathbf{E}_{rand} , k を求める処理をあらかじめ設定した回数繰り返す。そして k が最大となったときの式(6)を満たさない特徴点をアウトライアとする。その後、アウトライアを除いた特徴点を用いて式(5)より基本行列 \mathbf{E} を算出する。

4. 特徴追跡

4.1 特徴追跡

全方位画像列で特徴点の抽出・追跡を行うことで、画像間に対応点を得る(図4)。特徴追跡にはKLT法[4]を用いる

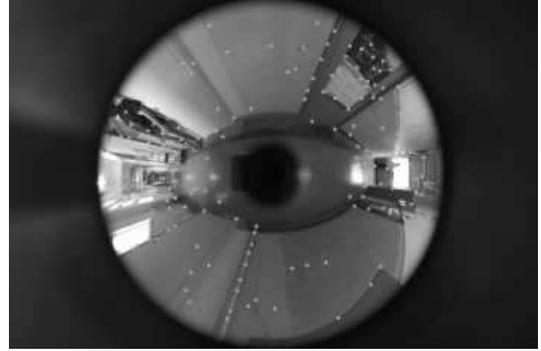


図4 特徴点の抽出

4.2 GPUによる特徴追跡の高速化

KLT法はフィルタリング処理(平滑化, 微分, 固有値計算)を全画素に対して順番に行う。よって計算回数が多く、処理時間も大きい。例えば、 $H \times W$ pixelの画像に $M \times N$ pixelのフィルタで処理を行う場合、総計算回数は $H \times W \times M \times N$ 回となる(図5)。そこで、本研究では特徴追跡を高速化する。

近年、GPUによる高速な画像処理が提案されている。GPUはCPUと比較して多数のコアを有する(表1)。多数のコアを用いた演算の並列化により、処理の高速化が可能である。表1でGPUは、クロック数においてCPUに劣るが、コア数においては大きく上回る。よって演算を並列化する際にはGPUを用いた方が処理時間は短縮されると予想される。

特徴追跡は、同様の演算を繰り返し行う部分が多く、演算の並列化が処理時間の短縮に効果的である。そこで本研究では、KLT法による特徴追跡をGPUを用いて高速化する。

表1 CPUとGPUの性能比較

	CPU	GPU
コア数(個)	4	112
クロック(GHz)	2.67	1.62

※CPU: Intel Core i5-750,
GPU: NVIDIA GeForce GTS240

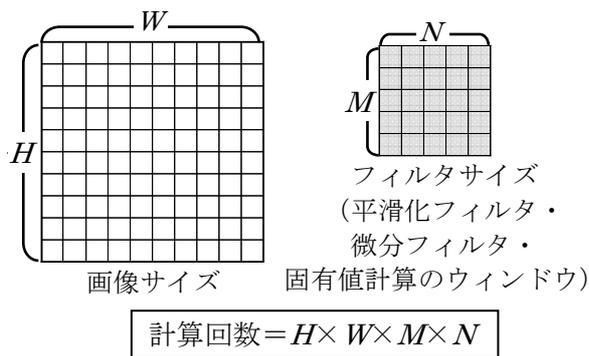


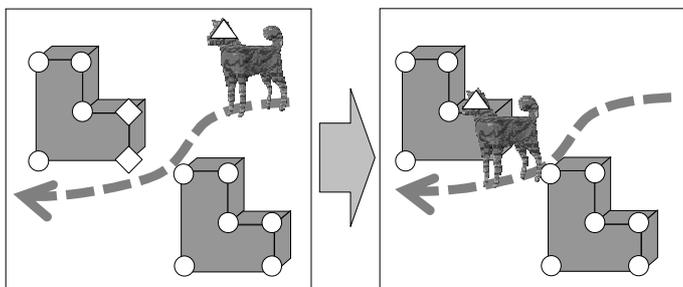
図5 フィルタリング処理における計算回数

5. 動物体検知

動物体の存在する環境において周囲環境の計測を行うことは、以下の問題により困難である。

まず、視野内に動物体が存在する場合、動物体上の特徴点を追跡してしまうことがある(図6)。動物体上の特徴点を計測に用いれば、正常に周囲環境の計測を行うことができない。

また、動物体の存在により、静止物体上に存在する特徴点の追跡を失敗してしまうことがある(図6)。静止物体上の特徴点を追跡できなければ、その特徴点が存在する領域の計測を行うことは困難である。



- ① 静止物体上の特徴点追跡 (○)
- ② 動物体上の特徴点追跡 (△)
- ③ 静止物体上の特徴点追跡の失敗 (◇)

図6 計測の際に問題となる特徴点

5.1 動物体の接近による特徴追跡の失敗

我々の従来手法では、特徴点に含まれるアウトライアを、カメラ運動推定および計測に使用しないことで、動物体による計測への障害を防止していた。

しかし、動物体がカメラに接近すると、静止物体上の特徴点が大量に損失する(図7)。静止物体上の特徴点を大量に損失すれば、正常な計測が困難である。そこで、本研究では動物体が接近する前に、動物体を検知する。

5.2 動物体による妨害の回避

本研究では、動物体を検知し、ロボットの移動および計測を一旦停止させることで、動物体存在下での計測への障害を回避する。動物体を検知して一旦停止したロボットは、撮影および特徴追跡を継続する。フレーム間で大きく座標を変化させた特徴点が存在するときには動物体が視野内に存在すると判定し、すべての特徴点が静止した時点で、移動および計測を再開する。

5.3 検知方法

動物体の検知には文献[2]の手法に基づき、カメラ運動推定とともに、特徴点に含まれるアウトライアを検出する。データ中のアウトライアの数と正解データと比べて十分に少ないとき、アウトライアは動物体上もしくは誤対応の特徴点であると考えられる。ここで、動物体が視野内に存在すれば、アウトライアとみなされた特徴点が画像上で動物体周辺に偏在すると予想される(図8)。そこで本研究では、アウトライアとみなされた特徴点の画像上での分布から、動物体の検知を行う。

5.4 動物体領域の判定

各特徴点の評価を行うため、画像を一定の大きさのブロックに分割し、各ブロックに番号を設定する(図9)。画像左下をブロック(0,0)とし、原点から右に*i*番目、下に*j*番目をブロック(*i, j*)とする。 $O(i, j)$ をブロック(*i, j*)に存在するアウトライアの数、 $F(i, j)$ をブロック(*i, j*)に存在する追跡が成功した特徴点、アウトライアとみなされた特徴点、追跡失敗した特徴点を合計した数とする。そして式(7)を満たす場合、動物体が存在すると判定する。ここで、 α は閾値である。画像中にひとつでも動物体を検知したブロックがあれば、カメラの視野内に動物体が存在すると判定する。

$$\frac{O(i, j)}{F(i, j)} > \alpha \quad (7)$$

5.5 対応点過少領域における誤検知の防止

アウトライアは、静止物体上にも少数検出される。また、領域によっては対応点の数が極端に少ない部分がある。よって静止物体上の領域においても、対応点中のアウトライアの割合が大きくなる可能性がある。静止物体上の領域においてアウトライアの割合が大きくなることは、動物体の誤検知の原因となり

問題である。そこで、本研究では、領域内に存在する対応点の個数に閾値を設定した。領域内に存在する対応点の個数が閾値未満であるとき、その領域では動物体が存在するかどうかを判定しない。対応点の個数に閾値を設定することで、個数が極端に少なく、かつアウトライアが存在する領域における誤検知を防止する。



移動前画像 (円：対応点)



移動前画像 (円：対応点)

図7 動物体の接近による特徴点の損失

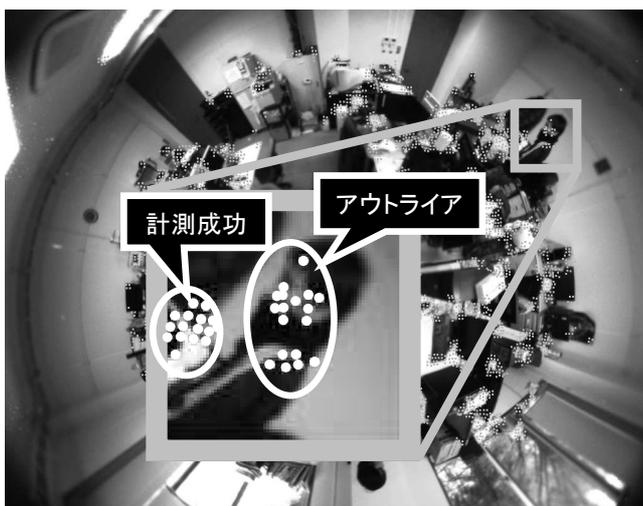


図8 局所的に存在するアウトライア



図9 領域分割

6. 実験

6.1 特徴追跡の高速化

特徴追跡の処理時間を、CPU と GPU で比較した (表2)。画像サイズは 1024×768 pixel, 追跡は画像 100 枚通して行った。実験に用いたプロセッサのスペックは、CPU が Intel 社製 Core i5-750 プロセッサ (コア数 4, クロック数 2.67GHz) で、GPU が NVIDIA 社製 GeForce GTS240 (コア数 112, クロック数 1.62GHz) である。ただし、CPU による処理で用いるコア数は 1 である。

結果より、速度向上は 5.71 倍となり、特徴追跡の高速化に成功した。

また、平滑化、微分、固有値計算における速度向上が処理全体の速度向上を上回る。これは、CPU で処理を行う部分が存在するためである。分岐処理など GPU 処理に向かない部分を除けば、CPU による処理も GPU 処理化により高速化可能である。

表2 処理時間

	CPU	GPU	速度向上
平滑化	1820ms	163ms	11.2 倍
微分	1240ms	196ms	6.33 倍
固有値計算	39.7ms	4.40ms	9.03 倍
上記以外の処理	219ms	-	-
処理全体	3319ms	580ms	5.71 倍
フレームレート	3.01	17.2	5.71
	frames/s	frames/s	倍

6.2 動物体検知

実験は屋内環境で行った(図 10)。実験に用いたのは動画像中の 15 フレームである。画像サイズは 1920×1080 pixel である。15 フレーム間における対応点からアウトライアを算出し(図 11)、分割した各領域(i, j)の評価式(7)の値を求めた。領域のサイズは縦が 180pixel, 横が 120pixel で行った。評価式における閾値 α は 0.5 とした。検出されたアウトライアとその他の対応点の座標から、設定した領域ごとの評価式の値を求めると、動物体の存在を検知した(図 11)。動物体を検知した領域は複数あった。検知した領域の座標を確認すると、すべて動物体領域上のブロックであった。

実験結果より、アウトライアは静止物体上にも検出された。本研究では、決められた閾値を超えた数の対応点が存在する領域のみ動物体検知の評価を行うことで、動物体の誤検知に対応した。実験では閾値を 10 として行ったところ、静止物体上の領域で動物体を検知することはなかった(図 11)。

また、対応点の個数が閾値を超える領域にもアウトライアは存在する。しかし、そのような領域では、評価式が閾値を下回ると予想される。実際に、動物体が誤って検知されることはなかった(図 11)。

7. 結言

本研究では全方位カメラによる周囲環境計測における特徴追跡の高速化と、動物体が視野内に侵入する場合の計測方法の提案を行った。実験結果より、特徴点抽出・追跡の高速化を行うことができた。現在、高速化されたのは特徴追跡のみであり、今後は 3 次元計測の高速化をする必要がある。

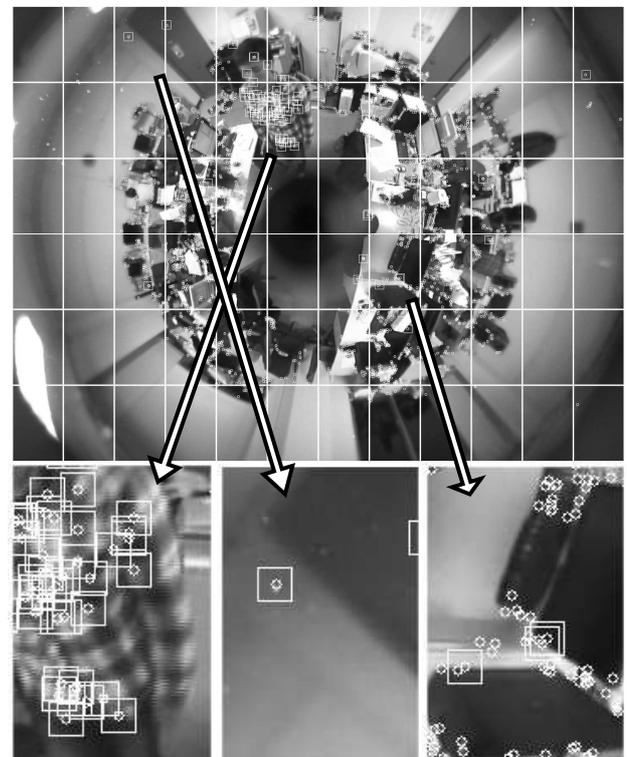
また、本研究では、動物体が存在する場合でも正常に周囲環境のオンライン計測を完了できる計測システムを提案した。また提案した計測システムに必要な動物体検知手法を提案し、実験を行った。しかし、今回の実験は動物体検知手法についての実験を行ったのみで、実際にロボットを用いた実験は行っていない。よって、実機を用いたオンライン計測も課題として挙げられる。

参考文献

[1] J. Gluckman and S. K. Nayar: "Ego-motion and Omnidirectional Cameras", *Proceedings of the 6th International Conference on Computer Vision*, pp. 999-1005, 1998.



図 10 実験環境



動物体検知領域 データ数過少領域 評価値過小領域

図 11 動物体検知画像

(円：対応点，矩形：アウトライア)

- [2] R. Kawanishi, A. Yamashita, and T. Kaneko: "Three-Dimensional Environment Model Construction from an Omnidirectional Image Sequence", *Journal of Robotics and Mechatronics*, Vol. 21, No. 5, pp. 574-582, 2009.
- [3] M. A. Fischler and R. C. Bolles: "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the ACM*, Vol. 24, No. 6, pp. 381-395, 1981.
- [4] J. Shi and C. Tomasi: "Good Features to Track", *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.