

## 移動物体の影響を低減した単眼モーシヨンステレオによる距離推定

倉貫 芳紀† 若山 涼至‡ 吉田 睦† 藤吉 弘亘‡

†ヤマハ発動機株式会社 ‡中部大学

E-mail: kuranukiy@yamaha-motor.co.jp

### Abstract

単眼モーシヨンステレオとは連続画像から得られるオプティカルフローを使ってカメラ運動パラメータと距離を推定する方法である。移動物体などのカメラ運動とは異なるオプティカルフローが観測された場合に距離推定が困難となる。本研究では、移動体の影響を軽減するために、局所領域のオプティカルフローからカメラ運動パラメータを推定する方法を2つ提案する。中心領域追跡による方法と複数パスを考慮したRegression Forestsによる方法である。評価実験より、中心領域追跡による方法では66.8%、複数パスを考慮したRegression Forestsによる方法では63.9%の距離画像精度を得ることができた。

### 1 はじめに

飲酒運転に対する罰則強化やシートベルト着用率の向上により、交通事故発生件数ならびに自動車運転中の死者数は大幅に減少している。しかし、交通事故死者数を状態別にみると歩行中が全体の37%を占め、近年最も多い要因となっている[1]。より安全安心な交通社会を実現するためには、歩行者を守る取組みが必要不可欠である。

そのようなニーズの高まりとともに、自動車への予防安全装置の搭載が始まりつつある。赤外線レーザやミリ波レーダなどを使った装置がある一方、カメラ画像を用いて障害物を検出する方法も数多く研究されている[2]。EyeSightに代表されるようなステレオカメラ画像を使った予防安全装置は既に市販車に搭載されている。更なる普及のためには、予防安全装置自体のコストダウンは避けられない。ドライブレコーダやバックモニターなどとして世の中に数多くの安価な車載単眼カメラが存在しており、そのような単眼カメラで予防安全装置が実現できれば、大幅なコストダウンが期待できるであろう。単眼カメラによる障害物検出方法も数多く研究されているが、障害物までの距離を推定することは難しい。

単眼カメラの映像から距離を推定するためには、大きく分けて2つの方法が挙げられる。1つ目は事前知識

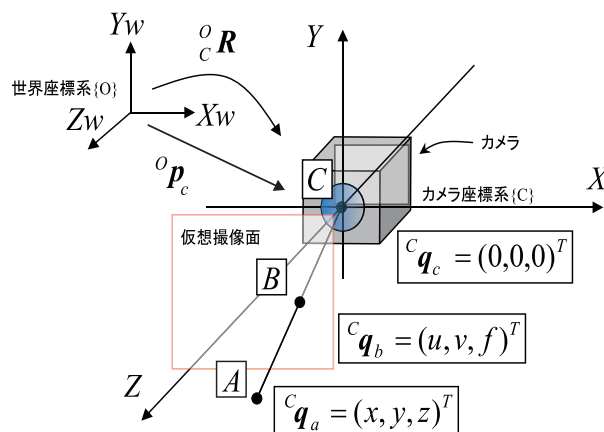


図1 カメラモデル

を用いて障害物までの距離を推定する方法、2つ目は連続して撮影された画像から距離を推定する方法である。前者の方法は、障害物の大きさや路面状況などの事前知識を基に障害物までの距離推定を行う。Steinらは、路面上の車両下端位置から距離推定を行なっている[4]が、カメラ方向と路面とが平行である仮定に基づくため、そのような前提条件が大きく崩れた場合は距離推定精度が悪化する可能性がある。後者の方法は、障害物や路面状況などの事前知識を必要としないが、画像間のフローから距離推定を行うため、移動物体などのフローが存在する場合に距離推定が破綻する可能性がある。本研究では、障害物に関する事前知識や路面状況などの前提条件に依存せずに距離推定可能なシステムを開発するべく、後者の単眼モーシヨンステレオによる方法を採用する。局所領域のオプティカルフローからカメラ運動パラメータを推定することで移動体の影響を低減し、安定的な距離推定を目指す。

### 2 単眼モーシヨンステレオの原理

まず、単眼モーシヨンステレオの関係式を導出する[3]。車載単眼カメラを図1にあるようなピンホールカメラモデルとする。単眼カメラの光軸と車両進行方向と平行になるように取り付けてあるものとし、カメラは事前に校正済みであるとする。世界座標系 $\{O\}$ からカメラ座標系 $\{C\}$ までの座標変換行列を ${}^O_cR$ とする。点Aに静止物体があり、それをカメラで撮影することで仮想撮

像平面上の点Bに射影される。点Cはカメラ座標の原点である。ここで、 $\mathbf{p}$ は世界座標系原点からのベクトル、 $\mathbf{q}$ はカメラ座標系原点からのベクトルとし、 ${}^0\mathbf{p}$ は $\{O\}$ 標記、 ${}^c\mathbf{p}$ は $\{C\}$ 標記とする。点Aは ${}^c\mathbf{q}_a = (x, y, z)^T$ と表すことができ、点Bは ${}^c\mathbf{q}_b = (u, v, f)^T$ となる。なお、ここで $f$ はカメラの焦点距離である。このとき、点Aと点Bは直線上にあり $z : f$ の関係にあるため次式のように表すことができる。

$$z {}^c\mathbf{q}_b = f {}^c\mathbf{q}_a \quad (1)$$

また、 ${}^0\mathbf{q}_a$ と ${}^c\mathbf{p}_a$ には次式の関係が成り立つ。

$${}^0\mathbf{p}_a = {}^0\mathbf{p}_c + {}^c\mathbf{R} {}^c\mathbf{q}_a \quad (2)$$

点Aが世界座標系において静止していることに注意しつつ式(2)の両辺を微分すると、

$$\frac{d}{dt} ({}^0\mathbf{p}_a) = \frac{d}{dt} ({}^0\mathbf{p}_c + {}^c\mathbf{R} {}^c\mathbf{q}_a) \quad (3)$$

$${}^0\dot{\mathbf{p}}_a (= 0) = {}^0\dot{\mathbf{p}}_c + [{}^c\boldsymbol{\omega}_c \times] {}^c\mathbf{R} {}^c\mathbf{q}_a + {}^c\mathbf{R} {}^c\dot{\mathbf{q}}_a \quad (4)$$

となる。ここで、 ${}^c\dot{\mathbf{q}}_a$ について解くと、

$${}^c\dot{\mathbf{q}}_a = -{}^c\mathbf{R}^T {}^0\dot{\mathbf{p}}_c - {}^c\mathbf{R}^T [{}^0\boldsymbol{\omega}_c \times] {}^c\mathbf{R} {}^c\mathbf{q}_a \quad (5)$$

$${}^c\dot{\mathbf{q}}_a = -{}^c\dot{\mathbf{p}}_c - [{}^c\boldsymbol{\omega}_c \times] {}^c\mathbf{q}_a \quad (6)$$

となる。ここで ${}^c\dot{\mathbf{p}}_c$ はカメラ座標系においてのカメラの並進速度であり ${}^c\dot{\mathbf{p}}_c = (t_x, t_y, t_z)^T$ となる。 ${}^c\boldsymbol{\omega}_c$ はカメラ座標系においてのカメラの回転速度であり ${}^c\boldsymbol{\omega}_c = (\omega_x, \omega_y, \omega_z)^T$ となる。 $[{}^c\boldsymbol{\omega}_c \times]$ は次のような演算子とする。

$$[{}^c\boldsymbol{\omega}_c \times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (7)$$

${}^c\dot{\mathbf{q}}_a$ はカメラ座標系から見た静止物体の移動量であるため ${}^c\dot{\mathbf{q}}_a = (\Delta x, \Delta y, \Delta z)^T$ とすると次式が得られる。

$$\Delta x = -t_x + \omega_z y - \omega_y z \quad (8)$$

$$\Delta y = -t_y + \omega_x z - \omega_z x \quad (9)$$

$$\Delta z = -t_z + \omega_y x - \omega_x y \quad (10)$$

一方、式(1)の両辺を微分すると、

$$\Delta z {}^c\mathbf{q}_b + z {}^c\dot{\mathbf{q}}_b = f {}^c\dot{\mathbf{q}}_a \quad (11)$$

となり、以上の式(1)(6)(10)(11)から次式が得られる。

$${}^c\dot{\mathbf{q}}_b = -\frac{f}{z} {}^c\dot{\mathbf{p}}_a - [{}^c\boldsymbol{\omega}_c \times] {}^c\mathbf{q}_b + \frac{t_z - \omega_y u + \omega_x v}{f} {}^c\mathbf{q}_b \quad (12)$$

ここで、 ${}^c\dot{\mathbf{q}}_b = (\Delta u, \Delta v, 0)^T$ とすると下記の関係式が得られる。

$$\Delta u = -\omega_y f - \frac{t_x f}{z} + \frac{t_z u}{z} + \omega_z v + \frac{\omega_x u v}{f} - \frac{\omega_y u^2}{f} \quad (13)$$

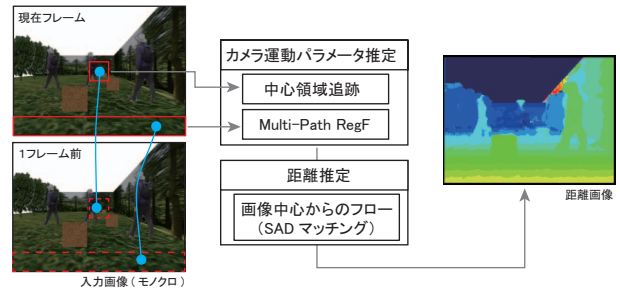


図2 単眼モーシヨンステレオの流れ

$$\Delta v = +\omega_x f - \frac{t_y f}{z} + \frac{t_z v}{z} - \omega_z u - \frac{\omega_y u v}{f} + \frac{\omega_x v^2}{f} \quad (14)$$

これはカメラ運動パラメータ、カメラからの静止物体までの距離、オプティカルフローについての関係式であり、この式をもとに単眼モーシヨンステレオによる距離推定が可能となる。

### 3 提案手法

連続画像間のオプティカルフローと式(13)(14)を使うことにより距離を推定することは可能だが、密な距離画像を生成するためには、画像全体に渡ってオプティカルフローを求めなければならない。さらに、距離分解能を確保するためにはサブピクセルでのオプティカルフロー計算が必要となり、そのようなフロー計算を画像全体で行うことは非常に計算コストが高く、車載向けCPUで高速に処理することは難しい。また、画像中には静止物体のみが存在するわけではなく、移動物体などのカメラ運動とは異なるフローが存在する場合もあり、それらのフローが大きく影響し距離推定が破綻する可能性がある。そこで、カメラ運動パラメータ推定と距離推定を切り分ける。まず、局所領域のオプティカルフローからカメラ運動パラメータを推定することで、移動体のオプティカルフローが存在した場合でも安定的にカメラ運動パラメータの推定が可能となる。距離推定については、画像中心(=消失点)からのフローに限定することで、計算コストの抑制を目指す。以上の流れを図2に示す。

#### 3.1 カメラ運動パラメータ推定

今回は問題を簡単にするために、推定するカメラ運動パラメータはカメラのヨー角 $\omega_y$ に限定する。カメラ進行方向 $t_z$ は車速センサにより計測可能であるとす。ヨー角の推定には中心領域の追跡による方法と、若山[5]らによる複数パスを考慮したRegression Forests (Multi-Path RegF)による方法を用いる。

### 3.1.1 中心領域追跡による方法

中心領域追跡による方法では, 中心領域つまり  $(u, v) = (0, 0)$  周辺領域のオプティカルフローから, カメラ運動パラメータを推定する. 今回はカメラ運動をヨー角に限定しているため, 式 (13) は次のように変形でき, 中心領域のオプティカルフロー  $\Delta u$  から直接ヨー角の推定が可能となる.

$$\Delta u = -\omega_y f \quad (15)$$

中心領域のオプティカルフロー計算には, 画像中心の  $20 \times 20$  pixel の窓を用いて 0.1pixel 精度で行う. 評価関数には SAD を用い, サブピクセル補間にはバイリニア法による輝度値の補間を行う. 中心付近の微小な領域のみを使ってカメラのヨー角推定を行うため, 画像中に移動物体がある場合でも安定的にヨー角推定が可能となる.

### 3.1.2 Multi-Path RegF による方法

Regression Forests は, 分類などに用いられる Random Forests[6] の決定木における末端ノードの出力を連続変数で求め, 回帰分析の問題を解くアプローチである. Regression Forests は Random Forests と同様に学習により回帰木を  $T$  本構築し, テストサンプルを複数の回帰木に入力し平均を取ることで回帰分析を行う. 複数の回帰木を用いることで, 非線形な複雑な回帰を表現することができる. しかし, 各回帰木において, トラバースの際に分岐ノードにおいてノイズを持つ特徴次元を参照した場合, 推定精度が低下するという問題がある.

本研究では, トラバースする時に複数パスを考慮することで, 分岐の反転による影響を低減する手法を用いる. 末端ノードまでにノイズの特徴次元を参照した回数により出力される値に重み付けを行い, 出力された全ての末端ノードの値を加算したものを各回帰木の出力とする. 各回帰木の出力を  $\hat{Y}_t$  とすると式 (16) のように表される.

$$\hat{Y}_t = \sum_{k=1}^K \frac{1}{2^{n_k}} \cdot \hat{y}_k \cdot \delta[f_k, 1] \quad (16)$$

ここで  $\hat{y}_k$  は末端ノードの出力を示し  $f_k$  は末端ノードにたどり着いた場合に 1 を持つ.  $n_k$  は末端ノードにたどり着くまでにノイズの特徴次元を参照した回数である.  $K$  は末端ノードの数を表す.  $\delta$  はクロネッカーのデルタ関数を表し, 2つの引数が一致する場合に 1 を, それ以外は 0 を返す関数である. Regression Forests の最終的な出力  $\hat{Y}$  は式 (17) に示すように各回帰木の出力

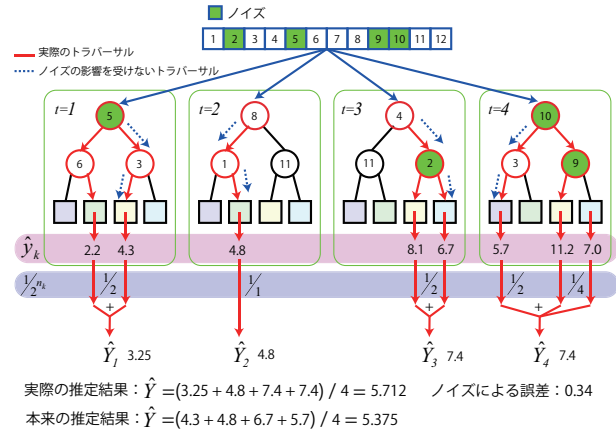


図3 複数パスを考慮した Regression Forests

$\hat{Y}_t$  の平均を出力とする.

$$\hat{Y} = \frac{1}{T} \sum_{t=1}^T \hat{Y}_t \quad (17)$$

図3に示すように, 構築した回帰木にノイズを含むサンプルを入力とした際, ノイズの特徴次元を参照した分岐ノードは左右両方の子ノードへ分岐する. ノイズの特徴次元を  $n$  回参照した場合は, 出力の信頼性が低くなるため,  $\hat{y}_k$  の出力に  $1/2^n$  の重み付けを行う. これにより, ノイズにより間違った末端ノードへ辿りついた場合, その影響を低減させることができる.

以上の手法を用いてカメラのヨー角の推定を行う. まず, 2枚の異なるフレームの画像を入力する. 入力した画像間で SIFT Flow[8] を用いてオプティカルフローを求める. 得られたオプティカルフローの内, 地面から求められるオプティカルフローを用いて Regression Forests で学習し回帰木を構築する. あらかじめ測定された地面までの距離と異なる距離を持つ画素をノイズと判定する. 推定シーン画像を回帰木へ入力することで, カメラのヨー角を推定する.

### 3.2 距離推定

画像中心からのオプティカルフローを用いて画像中の静止物体までの距離を推定する. 今回はカメラの運動パラメータをヨー角に限定しているため式 (13) は次のようになる.

$$\Delta u = -\omega_y f + \frac{t_z u}{z} - \frac{\omega_y u^2}{f} \quad (18)$$

ここで,  $\Delta \bar{u} = \Delta u + \omega_y f + \frac{\omega_y u^2}{f}$  とすると,

$$z = t_z \frac{u}{\Delta \bar{u}} \quad (19)$$

となる. 連続画像が撮影された間隔を  $\Delta t$  とすると  $\Delta u = \frac{u(t) - u(t-1)}{\Delta t}$  となる.  $u(t-1) = \alpha u(t)$  とすると式 (19) は,

$$\begin{aligned} z(t) &= t_z(t) \frac{u(t)}{\Delta \bar{u}(t)} \\ &= \frac{t_z(t) \Delta t}{\alpha - 1} \end{aligned} \quad (20)$$



となる. つまり, 現在の画像  $I_m(t)$  を  $\alpha$  だけ拡大した画像  $\alpha I_m(t)$  と 1 フレーム前の画像  $I_m(t-1)$  を画像中心を基準にマッチングした場合の最も適当な拡大率  $\alpha$  と, カメラ進行方向の並進速度  $t_z(t)$  から静止物体までの距離  $z(t)$  を推定することが出来る. マッチング評価関数には SAD を用い SAD 窓サイズは  $17 \times 17$  pixel とする. これにより, 距離の推定に必要なオプティカルフローの計算が画像の縦横方向の 2 次元ではなく, 画像中心を基準とした 1 方向のフロー計算のみで可能となる. 以上の流れを図 4 に示す.

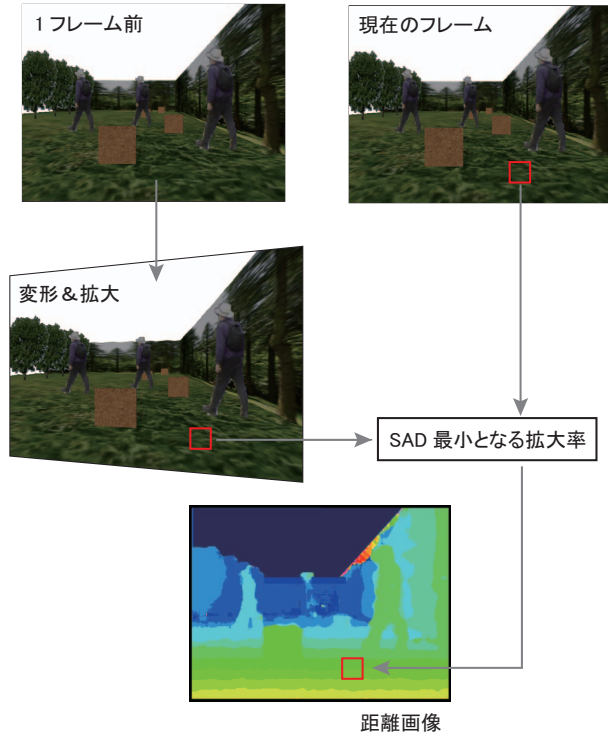


図 4 距離推定の流れ

## 4 評価実験

中心領域追跡による方法と Multi-Path RegF による方法の 2 種類のヨー角推定方法と, それぞれのヨー角推定値を用いた距離推定方法の有効性を確認するために, シミュレーションを用いた評価実験を行う.

### 4.1 評価環境

評価実験に用いるシーケンスは OpenGL を用いて作成する. シーケンスには, 移動物体や路面上の垂直物を配置し (図 5), カメラ運動とは異なるフローを発生させる. カメラは路面から 0.8 m の高さに設置されているものとし, カメラ方向に常に 0.1 m/frame の並進移動させる. ヨー角については  $-0.2 \sim 0.2$  deg の変動幅を 100 frames 周期のコサイン関数で与える. 出力画像サイズは VGA ( $640 \times 480$  pixels) とし, 250 frames のカラー画像を用いて評価を行う. 提案手法では入力画像はモノクロ画像を用いるため, 得られたカラー画像のシーケンスをモノクロ画像に変換し入力とする.

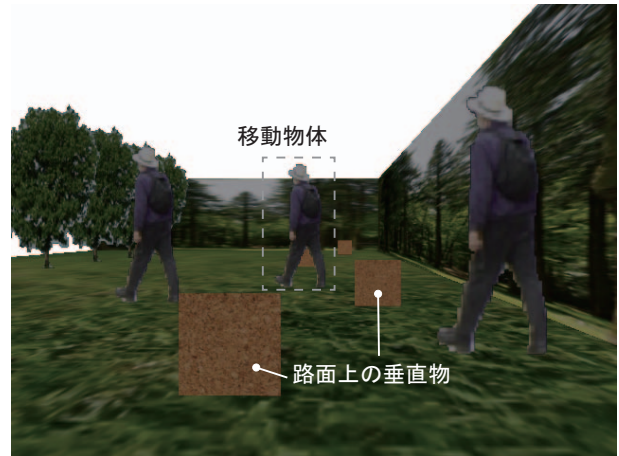


図 5 シミュレーション画像

### 4.2 評価方法

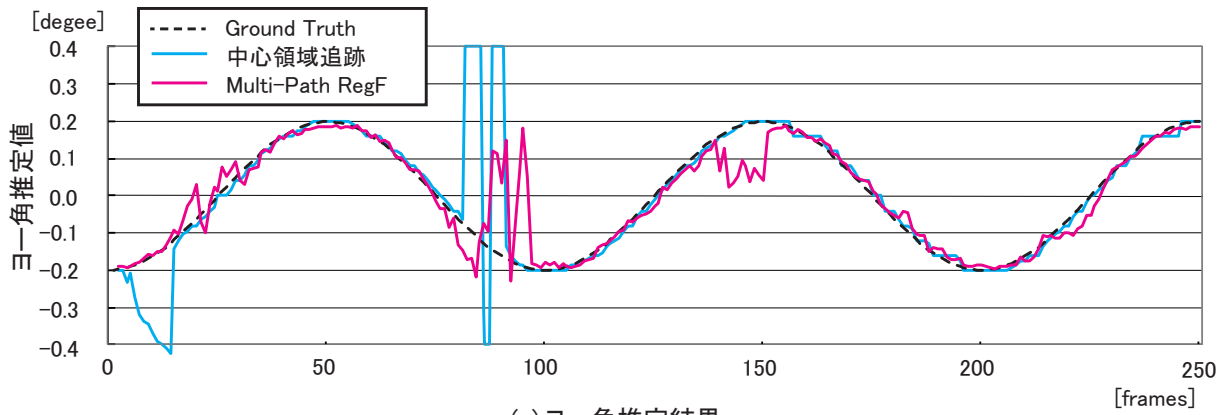
今回提案する単眼モーショステレオでは, まずカメラ運動パラメータとしてヨー角を推定し, その結果をもとに距離推定を行う. そのため, ヨー角推定結果と距離推定結果のそれぞれの評価を行う. ヨー角推定結果については, シミュレーション時に与えられるヨー角を Ground Truth としヨー角推定結果との比較を行い評価する. 距離推定結果については, シミュレーションから得られる倍精度の距離値を単眼モーショステレオの距離分解能まで丸め込んだ値を用いて距離画像を生成する. Ground Truth の距離画像と推定距離画像とを比較し正解率を算出する.

### 4.3 実験結果

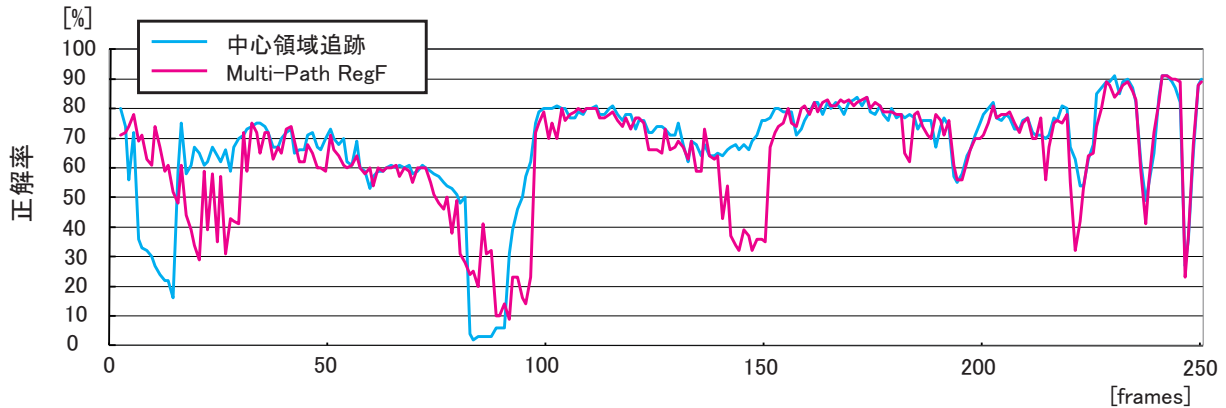
ヨー角推定結果を図 6(a) に示す. 中心領域追跡によるヨー角推定値を見ると Ground Truth から大きく外れるフレームがある. これは, 中心領域を移動物体が通過したためと考えられる. Multi-Path RegF による方法も誤差が大きくなるフレームが存在する. これは, 路面上の静止物や移動物体などフローがノイズとして混入し, ヨー角推定精度が低下したためと考えられる. 表 1 を見ると, 中心領域追跡の方が平均誤差が小さく, 誤差分散は Multi-Path RegF の方が小さい.

距離画像精度を図 6(b) に示す. どちらの方法とも, ヨー角推定誤差と距離画像精度に相関があることが確認できる. 表 2 に平均距離画像精度を示す. 全体として中心領域追跡の方が, 精度が高くなることが分かった. これは中心領域追跡の方が平均ヨー角推定誤差が小さいためと考えられる.

距離推定の結果画像を図 7 に示す. ヨー角推定結果と同様に中心以外に移動物体 (破線部) が存在した場合は, どちらの方法とも安定して距離推定が出来ているが (図 7 上段), 中心付近を移動物体が通過する場合は, 中心領域追跡による方法での距離画像精度が著しく低下した (図 7 下段).



(a) ヨー角推定結果



(b) 距離画像精度

図 6 評価結果

結果として、中心領域追跡による方法は Multi-Path RegF による方法に比べて移動体の影響を受けやすいが、平均ヨー角推定精度は Multi-Path RegF に比べて良く、平均距離推定精度も良くなるなどの結果になった。中心領域追跡による方法と Multi-Path RegF による方法の2つの方法どちらにおいても、移動体の影響を低減することで安定的に距離推定を行うことができた。

表 1 ヨー角の推定精度 [deg]

	平均誤差	誤差の分散
中心領域追跡	0.0442	0.0237
Multi-Path RegF	0.1187	0.0033

表 2 距離画像精度 [%]

	平均正解率
中心領域追跡	66.8
Multi-Path RegF	63.9

#### 4.4 実環境評価

今回提案した距離推定方法を用いて、実環境でのテストを実施した。ゴルフカート前方の車両中心にカメラを取り付け、地面から 0.8 m の高さとなるようにした、さらに、進行方向とカメラ光軸が平行になるように調整した。距離推定結果を図 8 に示す。

## 5 おわりに

本研究では、移動体の影響を低減するために局所領域のオプティカルフローからカメラ運動パラメータを推定する方法を用いた。中心領域追跡によるものと Multi-Path RegF による2つの方法を実施し、どちらの方法においても移動体の影響をある程度低減してヨー角推定および距離推定を行うことができた。

今回はカメラ運動パラメータ推定をヨー回転に限定していたため、今後はピッチ回転やカメラの鉛直および水平方向の並進運動への拡張を目指す。また、移動体そのものの距離推定については原理的に不可能なため、物体検出との統合を今後進めていき障害物検出+距離推定のシステム開発を目指す。また、車載カメラの映像など実環境のデータセットで実験を行い、汎用性を確かめる予定である。

## 6 謝辞

実験データの取得に協力頂いた葛城ゴルフ倶楽部に、感謝の意を表す。

## 参考文献

- [1] 警察庁, 平成 24 年中の交通死亡事故の特徴及び道路交通法違反取締状況について, 2013.

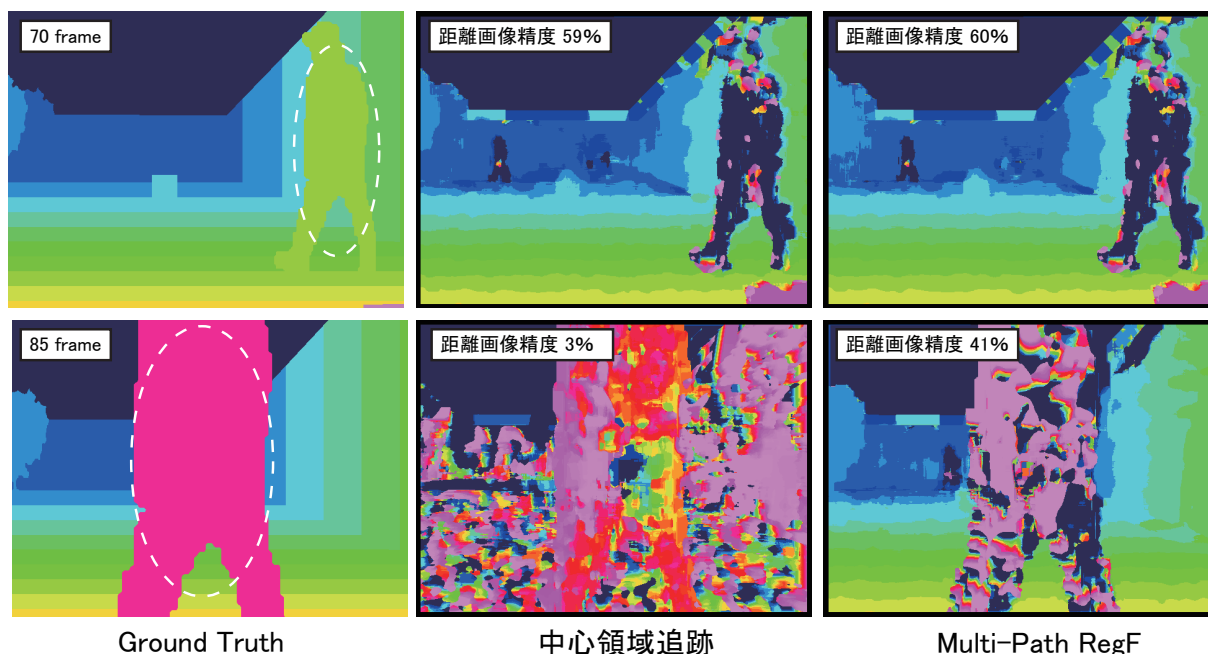


図 7 距離推定結果

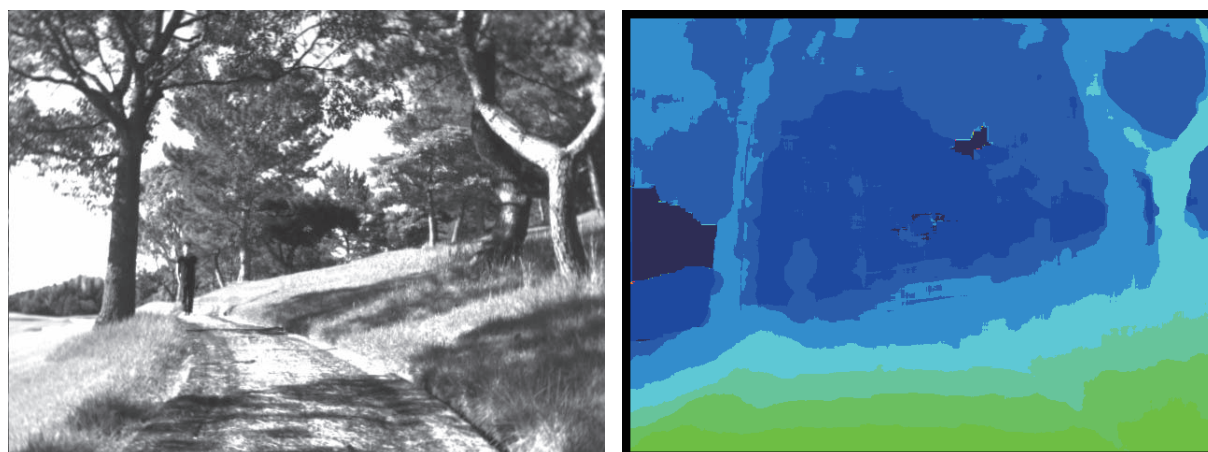


図 8 実環境評価

- [2] S. Zehang, et al., On-road vehicle detection using optical sensors: A review. Intelligent Transportation Systems, 2004.
- [3] H. Longuet-Higgins, et al., The interpretation of a moving retinal image. Proceedings of the Royal Society of London B, Vol. 208, pp. 385–397, 1980
- [4] GP Stein, et al., Vision-based ACC with a Single Camera: Bounds on Range and Range Rate Accuracy, IV2003, pp. 120–125, 2003.
- [5] 若山涼至, et al., 複数パスを考慮した Regression Forests によるカメラのヨー角の推定, 信学技報, Vol. 112, No. 441, pp. 111–112.
- [6] L. Breiman, Random Forests, Machine Learning, vol. 45, pp. 5–32, 2001.
- [7] A. Criminisi, et al., Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning, Foundations and Trends in Computer Graphics and Vision. Vol. 7, No. 2–3, pp. 81–227, 2012.
- [8] C. Liu, et al., SIFT Flow: Dense correspondence across scenes and its applications. Trans. Pattern Analysis and Machine Intelligence, Vol. 33, No 5, pp. 978–994, 2011.