

ステレオ動画画像を利用した道路パターン抽出による 自車両の運動推定

関 晃仁 奥富 正敏

東京工業大学 大学院理工学研究科

本論文では、車載ステレオカメラを使った自車両の運動推定手法を提案する。カメラを利用して自車両の運動推定を行なうには、画像中での静止領域を特定する必要がある。そこで、静止領域を抽出するため、ステレオ画像を使い、平面領域を抽出する。抽出された平面領域中の道路パターンを用い、その時系列画像でマッチングを行なうことで、運動推定を行なう。マッチングには、並進と回転運動である3つのパラメタが含まれるが、1). 車両の運動モデルを用いることで、簡便に運動推定を行なう方法と 2). 1) で求められた運動推定結果を更新することで、より高精度に求める方法について述べる。最後に、合成画像に対して本手法を適用した結果と、車載カメラによる実画像を用いた実験を通じて、本手法の有効性を示す。

Ego-Motion Estimation by Matching Road Patterns Extracted Using Stereo Images

Akihito Seki Masatoshi Okutomi

Graduate School of Science and Engineering,
Tokyo Institute of Technology

In this paper, we propose the method for ego-motion estimation using the stereo cameras mounted on a vehicle. Estimating ego-motion using the cameras requires to extract the static regions in the images. We first estimate planar regions which might be static area using stereo images. And so then, to match the time series of the road patterns in the extracted regions is equivalent to estimating ego-motion. We use car-movement-model and consider the matching method, so we easily and accurately get the motion. Finally, we present the experimental results of ego-motion estimation with our method.

1 はじめに

車載カメラからの映像を利用して、自車両の運動推定を行うことにより、その推定結果をビジュアルナビゲーションや安全走行のための運転支援、また自車両に対する物体の相対速度を絶対速度へ変換することなどが可能となる。

通常、車両の運動を推定するには、車速センサ、舵角センサやヨーレートセンサからの情報を使う場合がほとんどである。しかし車速センサからの情報では、車輪の滑りが発生した場合、正確な速度を求めることができない。

画像を利用して、車両の運動を求める手法は、単眼を用いる方法と複眼を用いる方法に分けることができる。単眼を用いる手法として、高速道路の白線長さのように規格化されている情報を利用することで車両速

度を算出する方法 [3] が考案されている。また、スリットカメラを利用し、特別な模様を読み取ることで、速度とともに、自車両の位置を測定する方法 [1] も考案されている。しかし、規格化された路面パターンは限られた環境下でしか利用できない。

路面パターンの拘束を用いない方法として、単眼カメラを利用し、画像中のフローを使って自車両の運動を求める方法 [9, 8] が考案されている。しかし、道路領域内に路面模様がないとフローを正しく求めることが難しかったり、画像中の道路領域をどのように特定するのかといった問題がある。また時系列画像間の道路面に対する射影変換行列を求めることで、運動推定をする方法 [2] も考案されている。しかし、道路面とカメラの間の距離を求めることができないため、推定される並進運動には定数倍の不定性を持つ。

次に、複眼を利用する方法として、画像中の複数の

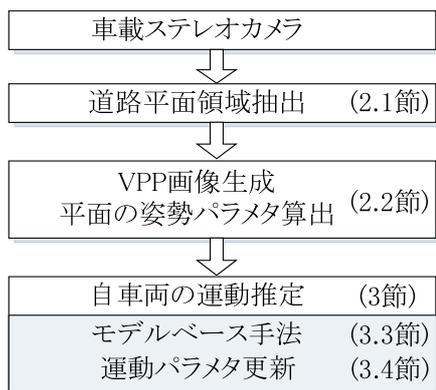


図 1: 本論文の手法の処理全体図

注目点について，ステレオ画像間の対応点から 3 次元位置を計測し，それら注目点の 3 次元位置の時間的変化により，運動推定を行なう方法 [10] が考案されている．しかし，注目点として静止点を選び、その点のトラッキングに成功し，かつそれぞれの点に対して，ステレオ 3 次元計測が成功することなど必須項目が多く，市街地のように，画像中に多くの移動物体が映るような場合には安定性に欠けると考えられる．

さて，車両の運動推定を正確に行なうためには，画像中で静止領域を基準としてカメラの相対的な移動量を求めればよい．そこで，提案手法では，最初にステレオ画像から道路平面領域を抽出することで，静止領域を得る．次に，抽出した平面に含まれる道路パターン全体を使い，時系列に渡ってマッチングすることで，簡便に自車両の並進移動と回転角を算出する．画像間のマッチングには，まず車両の運動モデルを用いることで，計算量を軽減しながら安定に求める方法と，さらにその結果を更新することで高精度に求める方法について提案する．

提案手法では，画像中の静止領域が未決定な部分の局所的なフローを使ったり，ステレオ対応点探索やトラッキングを行なったりする必要がなく，陽に求めた静止領域全体を使って運動推定を行なうため，安定な手法であると考えられる．

2 提案手法の概要

本論文で提案する手法の全体図を図 1 に示す．まず奥富らの手法 [4, 5] を利用してステレオ画像間の射影変換行列を動的に求め，道路平面を抽出する処理を行う (2.1 節)．さらに，射影変換行列を分解することで，平面とカメラ間の姿勢パラメータである，基準カメラに対する平面の法線ベクトルと平面までの距離を得る．姿

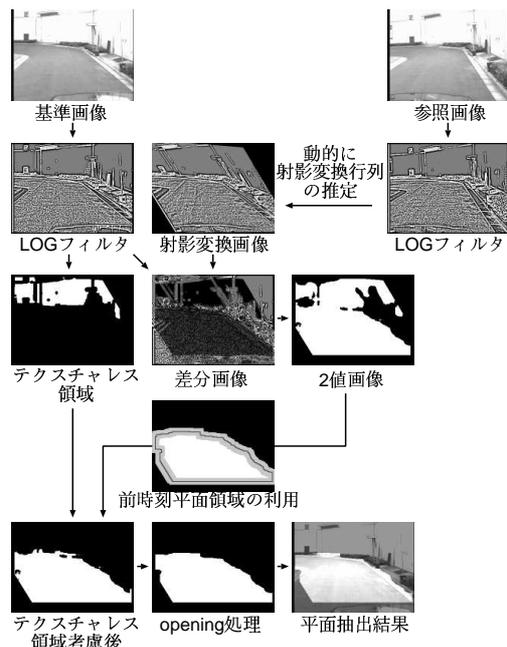


図 2: 平面抽出処理の全体図

勢パラメータは動的に推定されるため，走行時の振動を推定することができる．次に，姿勢パラメータを使って，道路平面を仮想的に上方から見た画像 (VPP 画像) を生成する [6] (2.2 節)．

そして，隣接した時系列 VPP 画像を使い，静止領域である道路平面領域内について，マッチングをすることで，運動推定を行なう (3 節)．マッチングには，車両の運動モデルを用いて，計算量を軽減する方法 (3.3 節) と，さらに高精度に求めるための方法 (3.4 節) について説明する．

最後に，合成画像と実画像に対して本手法を適用した結果を示す (4 節)．

2.1 道路平面抽出

画像中から道路平面を抽出する処理について，簡単に説明する．

図 2 は平面抽出処理の全体図である．まず，基準画像と参照画像を入力し，左右の画像の輝度を一致させるため，LOG フィルタを施し，ヒストグラム平坦化をしてコントラストを上げる．次に道路平面に対する射影変換行列を動的に推定する．その射影変換行列で参照画像を変形し，基準画像との輝度差をとる．平面部分は完全に重なるので，参照画像の平面上にある点は基準画像に変換したときに完全に重なり，その輝度差は小さくなる．平面判定を行うためには輝度差の閾値を用いて決定する．またテクスチャの無い領域では，平

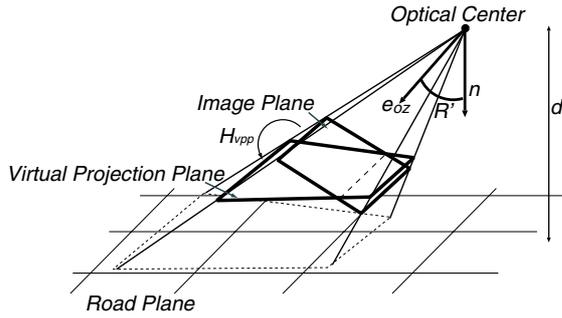
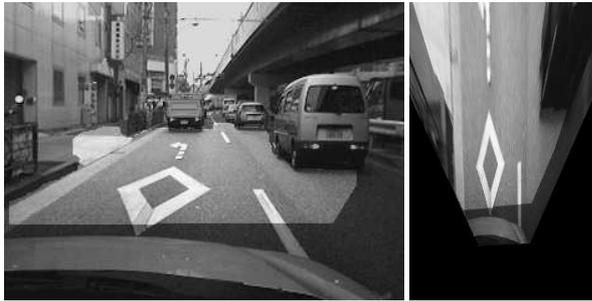


図 3: VPP 画像の幾何学的関係



原画像に平面領域を合成

VPP 画像に平面領域を合成

図 4: 平面抽出結果と VPP 画像例

面上にない点であっても変換後、輝度差がなくなる可能性があるため、別途処理* を加える。最後に Opening 処理を行い、小領域を除去する。

2.2 VPP 画像生成

2.1 節で算出された射影変換行列を用いて、平面の姿勢パラメタを求め、図 3 のように道路平面と平行な仮想投影面画像 (VPP 画像) を生成する。VPP 画像を生成するための仮想カメラは、光軸が平面の法線ベクトルに一致し、VPP 画像の縦軸が実際のカメラの光軸方向に一致したものとなるように座標変換を行ない、生成する。生成された VPP 画像は、縦軸と横軸がそれぞれ実空間の奥行と幅に相当し、かつ縦軸は現実のカメラの光軸方向に一致したものとなる。詳しくは、文献 [6] を参照されたい。

本節で述べた平面領域を抽出した結果と VPP 画像を生成した結果を図 4 に示す。これらの画像中で明るく示す領域が抽出された平面領域である。

* テクスチャレス領域ごとにラベリングし、領域ごとに基準画像と参照画像で射影変換行列を使ってマッチングを行い、マッチング結果が完全に平面内に収まっていると判定される領域は道路領域とし、ずれが生じている領域は非道路領域とする。

3 自車両の運動推定

3.1 運動推定と移動ベクトル数

本節では、平面内の回転と並進による 3 つの未知数 (運動パラメタ) を推定するために必要な注目点とその移動ベクトルの個数について考察する。

3.1.1 任意の運動

ある点 (x, y) が、 α 回転し、 t_x, t_y 並進運動したとき、

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (1)$$

となるので、移動ベクトル ($\Delta x = x' - x, \Delta y = y' - y$) は以下のように表される。

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x(-1 + \cos \alpha) - y \sin \alpha + t_x \\ x \sin \alpha + y(-1 + \cos \alpha) + t_y \end{bmatrix} \quad (2)$$

よって、運動を表す未知数が 3 つに対して、1 つの注目点の座標 (x, y) と移動ベクトル $(\Delta x, \Delta y)$ が分かると式が 2 本成立するので、運動している物体 (ここでは道路平面) 上の 2 つ以上の注目点と移動ベクトルを用いることで、運動パラメタを求めることができる。

また、回転角度 α が十分に小さい場合には、 $\cos \alpha = 1$ 、 $\sin \alpha = \alpha$ として、式 (2) より α を消去すると、

$$xt_x + yt_y = y\Delta y + x\Delta x \quad (3)$$

となり、先ほどと同様に 2 つ以上の注目点と移動ベクトルを与えることで、 t_x, t_y を最小 2 乗法で計算した後、 α を算出でき、運動パラメタを求めることができる。

3.1.2 車両の運動モデルの利用

車両の運動モデルを利用することにより、1 つの移動ベクトルと注目点から、運動パラメタを求めることが出来る。以下にその説明をする。

まず、自動車をアッカーマンステアリングジオメトリを想定し、2 車輪モデルに当てはめる。4 車輪モデルを 2 車輪モデルに当てはめるために、カメラの設置位置を通る車両の中心軸を考える。その様子を図 5 に示す。ここで、 l はホイールベース、 L は前輪からカメラの設置位置までの距離、 θ は操舵角、 α は回転角であるヨー角を示している。

時刻 $(t-1)$ における車両上の点 G を原点とした $G-XY$ 座標において、静止点 C を $C_{t-1,G} = (x, y)$ とする。車両の回転中心 O は、 $G-XY$ 座標において、

$$O_G = \left(-\frac{l}{\tan \theta}, -l + L \right) \quad (4)$$

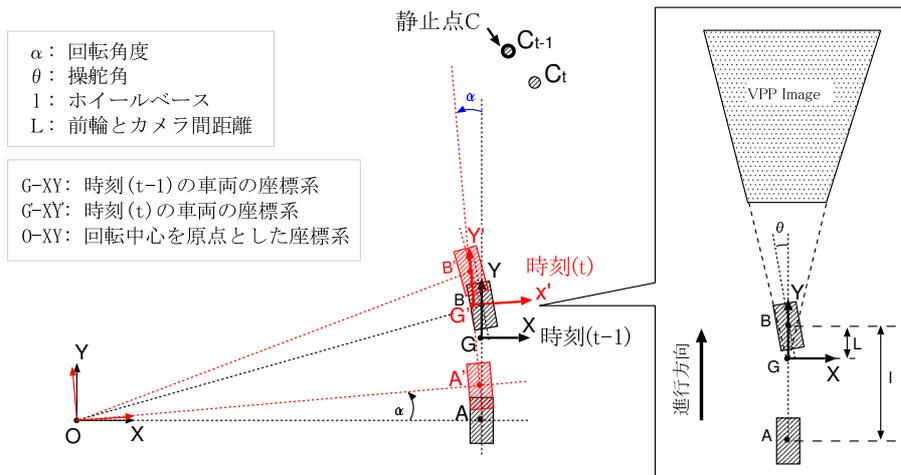


図 5: モデル化した車両の運動

と表される．そこで，点 C を $O - XZ$ 座標に変換すると，

$$C_{t-1,O} = \left(\frac{l}{\tan \theta} + x, l - L + y \right) \quad (5)$$

となる．

次に，点 O を中心として， α 回転し，時刻 (t) とする．静止点 C は， $O - XY$ 座標上では移動しないが，各時刻の車両の座標から観測すると，位置が異なる．そこで，時刻 (t) における車両上の $G' - X'Y'$ 座標で観測される静止点の位置を $O - XY$ 座標に変換する．その位置が図 5 中の点 C_t であり，以下のように示される．

$$C_{t,O} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \frac{l}{\tan \theta} + x \\ l - L + y \end{bmatrix} = \begin{bmatrix} \left(\frac{l}{\tan \theta} + x \right) \cos \alpha + (l - L + y) \sin \alpha \\ -\left(\frac{l}{\tan \theta} + x \right) \sin \alpha + (l - L + y) \cos \alpha \end{bmatrix} \quad (6)$$

車両の運動によって引き起こされる，点 C の画像間での移動量は， $C_{t,O} - C_{t-1,O}$ である．ここで， α が十分に小さいものと仮定すると， $\cos \alpha = 1$ ， $\sin \alpha = \alpha$ が成り立つので，

$$C_{t,O} - C_{t-1,O} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} (l - L + y) \alpha \\ -\left(\frac{l}{\tan \theta} + x \right) \alpha \end{bmatrix} \quad (7)$$

よって， θ と α は，式 (7) より，

$$\begin{cases} \alpha = \frac{\Delta x}{l - L + y} \\ \theta = \arctan \left(\frac{l}{-\frac{\Delta y}{\Delta x} (l - L + y) - x} \right) \end{cases} \quad (8)$$

となり，注目点 (x, y) と移動ベクトル $(\Delta x, \Delta y)$ を代入し， θ と α を求める．さらに，式 (7) を用いて，注目点を点 G に一致させることで，回転と並進の運動パラメータを求めることができる．

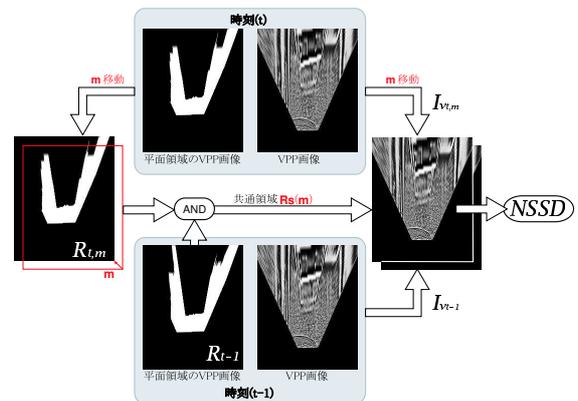


図 6: 移動ベクトル算出手順

3.2 VPP 画像を利用した移動ベクトル算出

自車両の静止座標に対する運動を求めるには，画像中で静止領域を基準としてカメラの相対的な運動量を求めればよい．前述した方法で求められた道路平面領域は，道路面を表しているの静止座標系であると考えてよい．そこで，実座標と同じ座標系である VPP 画像を利用し，平面内の領域について，領域ベースマッチングを行なうことで，移動ベクトルを算出する．ここで，フレーム間での回転角(ヨーレート) α が小さいため，VPP 画像に映る平面領域内は，一様な平行移動であると近似することで，移動ベクトル $m = m(\Delta x, \Delta y)$ を求める．

次に，移動ベクトルの算出方法について説明する．この処理の全体図を図 6 に示す．まず，時刻 (t) の平面領域 R_t を m 平行移動させた $R_{t,m}$ と，時刻 (t-1) の平面領域 R_{t-1} との共通領域 $R_s(m)$ を式 (9) を使い，算出する．この共通領域は両時刻における平面領域に相当

するため、移動体部分を除外して計算することに相当する。

$$\mathbf{R}_s(\mathbf{m}) = \mathbf{R}_{t,m} \cap \mathbf{R}_{t-1} \quad (9)$$

次に、時刻 (t) の原画像に LOG フィルタを掛け、ヒストグラム平坦化処理を行なった後、VPP 画像に変換し、 \mathbf{m} だけ平行移動させた VPP 画像、 $I_{V_t,m}$ を作成する。時刻 (t-1) の VPP 画像は $I_{V_{t-1}}$ とする。

次に、これらの画像の \mathbf{m} に対する類似度を式 (10) により、共通領域 $\mathbf{R}_s(\mathbf{m})$ 内に対して計算する。

$$NSSD(\mathbf{m}) = \frac{1}{N} \sum_{\mathbf{x} \in \mathbf{R}_s(\mathbf{m})} \{I_{V_t,m}(\mathbf{x}) - I_{V_{t-1}}(\mathbf{x})\}^2 \quad (10)$$

N は共通領域 $\mathbf{R}_s(\mathbf{m})$ の面積を示す。

この \mathbf{m} について、

$$\hat{\mathbf{m}} = \arg \min_{\mathbf{m} \in \mathbf{M}_0} \{NSSD(\mathbf{m})\} \quad (11)$$

ただし、 \mathbf{M}_0 は、フレーム間の移動量の探索範囲である。この式から $\hat{\mathbf{m}}$ を探索することで、類似度の高い最も適切な移動量を得る。

3.3 移動ベクトルから運動推定

求められた移動ベクトルを 3.1.2 節で述べた方法を利用することで、自車両の運動に変換する。移動ベクトルの成分である、 $(\Delta x, \Delta y)$ を平面とカメラ間距離および仮想カメラの内部パラメタから VPP 画像の 1 画素に相当する実距離を算出し、単位系を一致させた後、式 (8) に代入する。また、 (x, y) として、本論文では、計算領域内の重心の位置を用いた。この変換により、平行移動を仮定したマッチングから回転運動を含めた自車両の運動推定を行なうことができる。

本節の方法により求める運動推定手法をモデルベース手法と呼ぶこととし、合成画像と実画像に対して適用した結果を 4 節で示す。

3.4 運動パラメタの更新

本節では、前節で求められた運動パラメタを初期値として、より高精度に運動推定を行なう方法について述べる。

推定したい運動は回転運動と並進運動であるため、2 枚の画像のユークリッド変換行列を推定することに相当する。ここでは、文献 [7] に述べられている勾配法を利用し、以下のようにユークリッド変換を推定する。

まず、

$$\mathbf{x}' \simeq \mathbf{M}\mathbf{x} \quad (12)$$

$$\mathbf{M} = \begin{bmatrix} \cos \alpha & \sin \alpha & t_x \\ -\sin \alpha & \cos \alpha & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

としたとき、

$$\mathbf{M} \leftarrow \mathbf{M}(\mathbf{I} + \mathbf{D}) \quad (14)$$

として、 \mathbf{M} を更新し、 $\mathbf{x}'' = (\mathbf{I} + \mathbf{D})\mathbf{x}$ と置くと、

$$E(\mathbf{d}) = \frac{1}{N} \sum_{\mathbf{i} \in \mathbf{R}_s} \left[\tilde{I}_{V_t}(\mathbf{x}_i'') - I_{V_{t-1}}(\mathbf{x}_i) \right]^2 \quad (15)$$

を最小にする \mathbf{M} を求める。この \mathbf{M} から運動パラメタを得ることができる。そこで、式 (15) をテーラー展開することで、

$$E(\mathbf{d}) \approx \frac{1}{N} \sum_{\mathbf{i} \in \mathbf{R}_s} \left[\mathbf{g}_i^T \mathbf{J}_i^T \mathbf{d} + e_i \right] \quad (16)$$

ただし、 $\mathbf{g}_i^T = \nabla \tilde{I}_{V_t}(\mathbf{x}_i)$ 、 $e_i = \tilde{I}_{V_t}(\mathbf{x}_i) - I_{V_{t-1}}(\mathbf{x}_i)$ 、 $\mathbf{J}_i = \frac{\partial \mathbf{x}_i''}{\partial \mathbf{d}}$ である。まず、回転成分についての更新量を求めるには、

$$\mathbf{R} \leftarrow \mathbf{R}\hat{\mathbf{R}} \quad (17)$$

Rodriguez の公式を利用することで、

$$\hat{\mathbf{R}} \approx \mathbf{I} + \mathbf{X}(\hat{\boldsymbol{\eta}}) \sin \theta' \quad (18)$$

ただし、 $\theta' = \|\boldsymbol{\omega}\|$ 、 $\hat{\boldsymbol{\eta}} = \frac{\boldsymbol{\omega}}{\theta'}$

$$\mathbf{X}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \quad (19)$$

また、並進移動の更新分も考慮することで、

$$\mathbf{I} + \mathbf{D} = \begin{bmatrix} \mathbf{I} + \mathbf{X}(\boldsymbol{\omega}) & dt \\ \mathbf{0} & 1 \end{bmatrix} \quad (20)$$

として、式 (16) を最小化するので、

$$E(\boldsymbol{\Theta}) \approx \frac{1}{N} \sum_{\mathbf{i} \in \mathbf{R}_s} \left[\mathbf{g}_i^T \mathbf{J}_{\boldsymbol{\Theta}_i}^T \boldsymbol{\Theta} + e_i \right] \quad (21)$$

ただし、 $\boldsymbol{\Theta} = (\omega, dt_x, dt_y)$

$$\begin{aligned} \mathbf{J}_{\boldsymbol{\Theta}_i} &= \frac{\partial \mathbf{x}_i''}{\partial \boldsymbol{\Theta}} = \frac{\partial \mathbf{x}_i''}{\partial \mathbf{d}} \frac{\partial \mathbf{d}}{\partial \boldsymbol{\Theta}} \\ &= \begin{bmatrix} -y & 1 & 0 \\ x & 0 & 1 \end{bmatrix}^T \end{aligned} \quad (22)$$

となるので、

$$\mathbf{A} = \sum_{\mathbf{i} \in \mathbf{R}_s} \mathbf{J}_{\boldsymbol{\Theta}_i} \mathbf{g}_i \mathbf{g}_i^T \mathbf{J}_{\boldsymbol{\Theta}_i}^T \quad (23)$$

$$\mathbf{b} = \sum_{\mathbf{i} \in \mathbf{R}_s} e_i \mathbf{J}_{\boldsymbol{\Theta}_i} \mathbf{g}_i \quad (24)$$



図 7: 合成画像の基準画像列 (左から Frame No.0,10,30)

として,

$$A\Theta = -b \quad (25)$$

を解き,更新していくことで,最適な M をサブピクセル単位で求めることができる. M から,運動パラメタである自車両の並進移動と回転角であるヨーレートを算出することができる.

4 実験結果

これまで述べた手法を適用し,合成画像と実画像に対して運動推定を行なった結果を示す.

4.1 合成画像実験

まず,合成画像を利用して,運動推定を行なった結果を示す.

合成画像は OpenGL を利用し,道路面に関しては,実画像から道路面を切りだし,合成することで作成した(画像例を図 7 に示す). また画像は, 320×240 pixel として, 30 fps で撮影されたものと想定した. 車両は,ホイールベースが 2.7 m とし,前輪軸上にカメラが搭載されているものとした. カメラと平面間の距離は 1.2 m で一定とし,カメラと平面との姿勢は変動させた. マッチングに用いる VPP 画像は, 20 [pixel/m] とした[†]. 自車両は,一定速度で回転運動をしながら,ピッチ方向に振幅 3° の正弦波で変化している状況とした. この状況は,実際の車両運動と比較すると,厳しい条件であると考えられる. 図 7 は,その運動状況における連続画像を 10 フレームおきに示している.

図 8 が平面抽出処理によって推定された射影変換行列から得られた平面の法線ベクトルをピッチ角について示した実験結果である. ほぼ真値に近い値が推定されている. 次に,図 9 は,図 10 中の r に相当する,真値との距離を示している. この運動で想定した移動量である $|\Delta Y|$ は 0.588 m, $|\Delta X|$ は 0.0576 m である. モ

[†] VPP 画像は,仮想カメラの内部パラメタを任意に決定できるため,解像度を高くすることにより,より高精度に運動を算出することができるが,計算コストが大きくなるため,本論文では,この解像度に設定した.

表 1: 合成画像実験: RMS エラー

平面パラメタ	運動パラメタ推定	RMS	
		移動量	ヨー角
動的に変更	モデルベース手法	$4.672e-2$	$2.803e-3$
	運動パラメタ更新後	$1.902e-2$	$1.141e-3$
法線 85° 固定	モデルベース手法	$6.185e-1$	$1.492e-2$

デルベース手法によって算出した移動量でも,真値に近い値が算出できることがわかる. 運動パラメタを更新することで,より真値に近づくことも確認できる. 図 11 は,ヨーレートを示している. この図からも,モデルベース手法でもヨーレートが求まることが確認できるが,運動パラメタを更新することで,より精度が向上することが示されている. 次に,推定された移動後の位置について,図 12 に示す. この図は, $(X, Y) = (0, 0)$ が真値であり,図 10 中の領域 A に相当する領域を表している. モデルベース手法により運動推定をした結果は,作成した VPP 画像の解像度によって離散化された値となっている. 運動パラメタを更新することで,より真値に近づくことが確認できる.

表 1 は,運動推定結果の RMS エラーを示している. 同表の結果より,モデルベース手法で算出した運動推定より運動パラメタを更新した方が精度が良くなっている. また,平面パラメタは,平面とカメラの位置関係を動的に推定した結果を利用するか,ピッチ角をオフセットである 85° で固定値を利用するかを示している. 平面パラメタを固定すると, RMS エラーが 1 桁大きくなり,エラーが増大している.

4.2 実画像実験

本節では,実画像を利用して,運動推定を行なった結果を示す.

実験環境は,車両に SONY XC003 を 2 台ほぼ平行に配置した. 撮影される画像は, 30 fps で 320×240 pixel である. 実験車両は,ホイールベースが 2.7 m であり,カメラは前輪から 0.85 m 後方に設置されている. また,静止した状態で,カメラと平面との距離は約 1.2 m,カメラの平面に対する俯角は約 5° であるが,走行中は路面の状況などにより変動する.

シーン 1 図 13 は,実画像実験で用いた画像の基準画像を 80 フレームおきに示したものである. このシーンは,自車両の周りに併走する車両が存在するため,画像中に動物体が含まれる. 自車両の運動は,前走車の停止に合わせ,自車両も徐々に減速し,最後には停止す

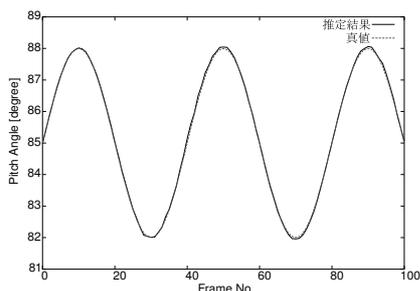


図 8: 合成画像：ピッチ角推定結果

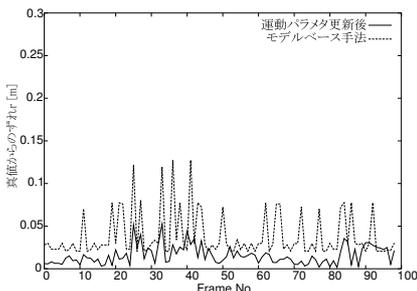


図 9: 合成画像：移動量差分

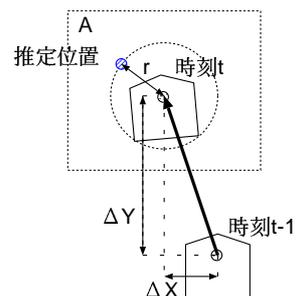


図 10: 自車両位置の移動

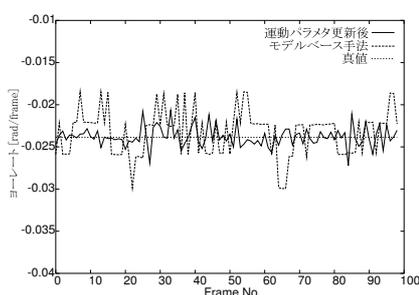


図 11: 合成画像：ヨーレート

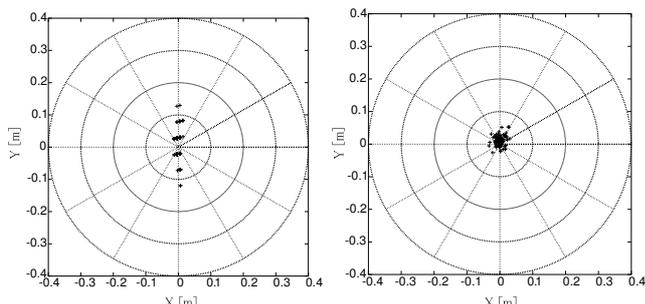


図 12: 合成画像での移動位置算出結果

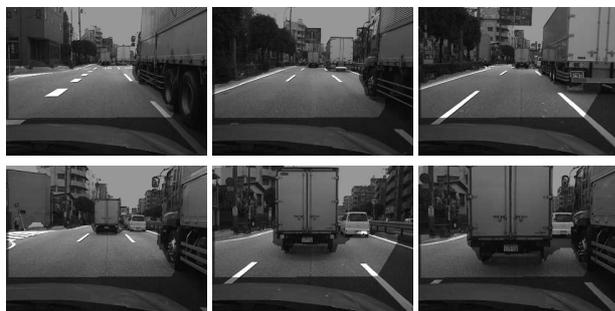


図 13: シーン 1 の画像列 (左上から右へ, 左下から右へ続く 80 フレームごとの連続画像)。ただし, 画像中の色の明るくなっている領域は平面抽出結果を示している。



図 16: シーン 2 の画像列 (左上から右へ, 左下から右へ続く 70 フレームごとの連続画像)。ただし, 画像中の色の明るくなっている領域は平面抽出結果を示している。

る。運動推定を行なった結果を図 14, 15 に示す。実画像を用いた実験では, 真値との比較が困難であるため, 推定した移動量およびヨーレートについて示す。まず, 図 14 は, 直進方向の移動量を示している。シーンが進むに従って, 移動量も少なくなり, 減速していることがわかる。モデルベース手法により求めた結果は, 移動量が階段状になっているが, これは, 生成した VPP 画像の解像度に依存し, サブピクセル推定を行っていないためである。高精度化手法を利用したものは, なんだかになり, 実際の現象により近いことが確認できる。図 15 は, ヨーレートを示している。このシーンでは, ほぼ直進していると考えられる。

シーン 2 図 16 は, 緩いカーブを走行するシーンの一部を示している。このシーンにおける運動推定結果を図 17 に直進方向の移動量を示し, 図 18 は, ヨーレートを示している。走行速度は, 移動量からほぼ等速であることが分り, ヨーレートから, ノイズは交じるものの, ハンドルを切り始めからニュートラルに戻すまでの変化が読み取れる。

5 まとめ

本論文では, 車載ステレオカメラから得られた時系列画像を利用して, 高精度に自車両の運動を推定する

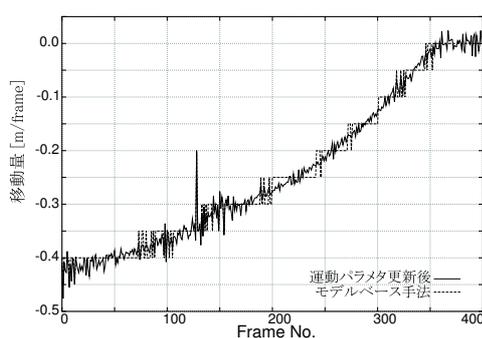


図 14: シーン 1 : 移動量

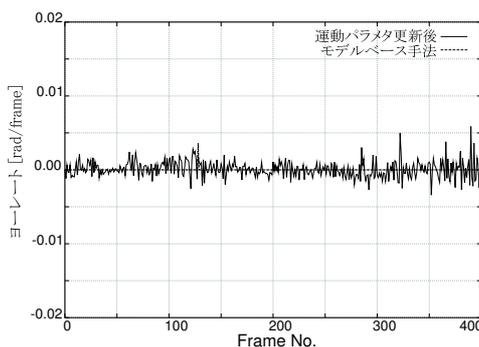


図 15: シーン 1 : ヨーレート

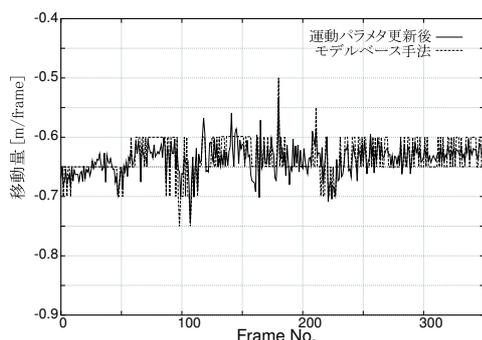


図 17: シーン 2 : 移動量

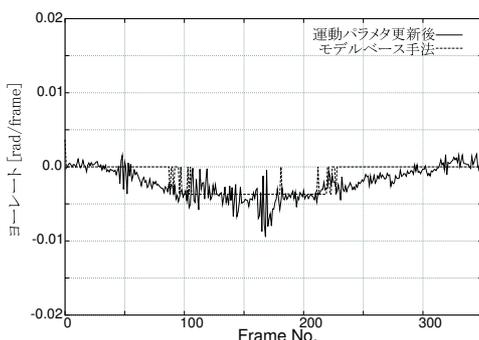


図 18: シーン 2 : ヨーレート

手法を提案した。まず、ステレオ画像から静止領域である平面領域を抽出するとともに、平面に対し仮想的に正対して見た画像である VPP 画像を生成する。VPP 画像と抽出された平面領域を利用し、マッチングを取ることで自車両の運動を推定した。マッチングを行なう際、並進と回転運動をする場合には、3つのパラメタが存在するが、車両の運動をモデル化することで、平行移動成分のみのマッチングから回転も含めた3つの運動パラメタを算出する手法について説明した。さらに、より高精度に推定するために、先に得られた運動パラメタを初期値として、ユークリッド変換行列を勾配法を利用して算出し、並進と回転運動を推定する方法について述べた。

最後に、合成画像に対して本手法を適用し、自車両に大きな振動が発生している場合にも、運動推定ができることを示した。また、実画像に対しても実験を行ない、運動が推定できることを示した。

今後の課題として、運動推定結果にフィルタリング処理を行なうことで、より安定に求める方法について検討していきたい。

参考文献

[1] 川又正太, 片原俊司, 青木正喜. 車載スリットカメラと路面マークを用いた位置計測システムとその応用. 第 8 回

画像センシングシンポジウム, pp. 71–76, 2002.

- [2] Qifa Ke and Takeo Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *IEEE CVPR*, June 2003.
- [3] 中森卓馬, 石川直人, 中島真人. 動画像処理による車両前方監視システム. 電子情報通信学会技術研究報告, pp. 1–6, 2002.
- [4] 奥富正敏, 中野勝之, 丸山純一, 原智章. ステレオ動画像を用いた視覚誘導のための平坦部の連続推定. 情報処理学会論文誌, Vol. 43, No. 4, pp. 1061–1069, 2002.
- [5] Okutomi, Nakano, Maruyama, and Hara. Robust estimation of planar regions for visual navigation using sequential stereo images. In *Proc. IEEE ICRA*, pp. 3321–3327, 2002.
- [6] 関晃仁, 奥富正敏. ステレオ動画像を利用した平面抽出による障害物検出. 情報学研報, Vol. CVIM-143, No. 3, 2004.
- [7] Heung-Yeung Shum and Richard Szeliski. Panoramic image mosaics. *Microsoft Research MSR-TR-97-23*, 1997.
- [8] Gideon Stein, Ofer Mano, and Amnon Shashua. A robust method for computing vehicle ego-motion. In *Procs. IEEE Intelligent Vehicles Symposium 2000*, pp. 362–368, Detroit, USA, oct 2000.
- [9] T. Suzuki and T. Kanada. Measurement of vehicle motion and orientation using optical flow. In *Procs. IEEE Intelligent Transportation Systems*, pp. 25–30, 1999.
- [10] Wannes van der Mark, Dani el Fontijne, and Leo Dorst. Vehicle ego motion estimation with geometric algebra. In *Procs. IEEE Intelligent Transportation Systems*, pp. 58–61, 2002.