

ドローネー三角形を利用した 単眼車載カメラの映像からの障害物検知法

長井 歩^{1,a)} 太田 直哉^{1,b)}

概要：車載単眼カメラを用いて障害物を検知するアルゴリズムを提案する。主に駐車時における障害物検知を目的とする。そのため静止した障害物の検知に主眼を置く。提案法のアイデアは、それまでに車両が走行してきた軌跡上には確実に障害物がないことを利用して蓋然性の高い道路面を求め、仮説検定によって道路面と判断できる領域を拡大し、以後その領域を道路面とみなし教師データとして利用する。次にフレーム内の特徴点をドローネー三角形分割し、三角形を単位に道路面か否かの最終判定を尤度関数によって行う。

キーワード：車載カメラ、障害物検知、単眼カメラ、ドローネー三角形分割、ウェルチの検定

Obstacle Detection from Monocular On-Vehicle Camera in units of Delaunay Triangles

Abstract: An algorithm to detect obstacles by using a monocular on-vehicle video camera is developed. Since our main application is when parking a vehicle, we focus on motionless obstacles. Our key idea is to use the knowledge that there are no obstacles along the trajectory of the vehicle so far. The knowledge helps us to find a part of road surface with high probability and to construct a likelihood function. The image is divided into Delaunay triangles so that we can judge each triangle whether it is a part of road surface or not.

Keywords: on-vehicle camera, obstacle detection, monocular camera, Delaunay triangulation, Welch's test

1. はじめに

近年、自動車の運転支援のためセンサを搭載した車両が増えている。特に安価な単眼車載カメラを備えた自動車の増加は顕著である。その利用方法は実用化しているものだけでも、駐車時のバックモニター、前方の車両への追突防止、前方車両への追従運転など多岐に渡る。本論文では障害物検知に焦点を当てる。自動車による接触事故は人命に関わるだけに障害物の検知は非常に有用であり、検知精度と価格次第では近い将来の需要が期待できる。価格面で有利なセンサはカメラである。そこで障害物との接触事故の予防のため、車載カメラによる障害物検知、特に駐車時における障害物検知に着目する。

検知すべき障害物の定義についてであるが、障害物を移

動体に限定する研究も多い。移動体は間違いなく障害物であり、さらにオブティカルフローや背景差分などの方法で原理的には比較的容易に検知可能なためである。本論文では、駐車時の障害物検知を主な用途と考える。そのため障害物を移動体に限定せず、車両へ衝突する可能性のある物体全般とする。要は道路面以外の物体はすべて障害物と考える。(具体例としては、壁や柱、停車中の他の車両など。)そこで道路面の検知が不可欠となる。

提案法は駐車中の衝突防止を目標に、移動体ではなく静止した障害物の検知に主眼を置く。提案法の入力は、車載カメラの現在の映像とその直近の数十～数百フレームに限る。過去の映像などの事前情報は用いない。その方が未知の駐車場にも柔軟に対応できる。提案法の出力は、各フレームに対して生成した3次元ポリゴンと、そのポリゴンを構成する各三角形が道路面か障害物かという判定結果である(3.5節)。ポリゴンなので、フレーム上の任意の点について障害物か否か判定できる。別の出力方法として、車

¹ 群馬大学 群馬県桐生市天神町 1-5-1
Gunma University, Kiryu-shi, Gunma, 376-8515

a) nagai@cs.gunma-u.ac.jp

b) ohta@cs.gunma-u.ac.jp

載カメラの映像の上に障害物の危険度を色で重ね描きした図も示す (3.6 節). これはカルマン・フィルタによって時間的空間的に安定化させた危険度の表示手段である. また, 提案法の実行速度を度外視した上でオンライン実行するシミュレーション実験も行った (4.6 節). その結果判明した注意点も議論する.

2. 関連研究

オプティカルフローによる障害物検知には多数の研究 [3], [6], [12] がある. 基本的には移動する障害物を検知するが, 静止している障害物を検知する研究 [7] もある.

前方を走行する車両の検知に限定した研究 [2], [5], [11] も多数行われている. これらの研究は, 白線検出やナンバープレート認識, 左右の対称性など, 車道走行時の車両後方からの見え方に特化した工夫がなされている.

フレーム間画像の差を利用した障害物検知 [14] の研究もある. これは移動する障害物を検知する. 過去に撮影した映像との差分によって障害物を検知する研究 [9] もある. この場合の障害物は過去の映像に写っていない物体, つまり移動体や一部の静止物体である. 壁などの恒久的な障害物は検知できない.

被写体が存在するのが道路面上に平行な平面上か, あるいは進行方向に垂直な平面上かの二択を仮定し, 輝度分布にて判定する方法 [8] もある.

障害物除去を目的とした研究の中には障害物検知に転用できるものもある. 例えば [10] などである. 遠方の人工物は平面上のテクスチャで近似できることを利用し, それに当てはまらないものを除去する仕組みのため, 障害物検知として利用するには映像の中に道路面が写り込まないなどの条件を満たす必要がある.

3. 提案法

3.1 概要

提案法は 5 つのステップからなる. 最初の 2 つは前処理, 次の 2 つが主要部分, 最後の 1 つは表示方法である.

- (1) **Voodoo による 3 次元特徴点の取得:** 特徴点の 3 次元位置復元ツールである Voodoo[1] を実行し, 3 次元特徴点 (以後「3D 特徴点」と呼ぶ) を取得する.
- (2) **スケール調整:** Voodoo の出力は被写体のスケールが正確ではない. そこで 3D 特徴点に対してスケール調整を行う. ここまでが前処理である.
- (3) **走行軌跡付近の蓋然性の高い道路面を求める:** 現在までの車両の走行軌跡には確実に障害物がないことを利用し, 高い蓋然性で道路面と判定できる領域を求める. 走行軌跡からの距離を少しずつ大きく取り, 3D 特徴点の分布が変わらない領域を道路面と判定する. 判定に用いるのはウェルチの検定 (Welch's test)[13] である. これは 2 つの正規母集団の母平均の差の検定法の

一つで, 母分散が未知で等しいとは限らないときに用いられる検定方法である.

- (4) **ドロネー三角形ごとに道路面判定:** 前の項目で求まるのは, 車両の軌跡に近いために道路面と判定できる 3D 特徴点に限られる. そこで, 軌跡に近くない一般の 3D 特徴点については尤度関数によって判定する. 尤度関数は 3D 特徴点からなるドロネー三角形 [4] に対して定義し, 重心と法線角度を説明変数とする. 前の項目で既に道路面上と判定されている三角形を教師データとし, 他の一般の三角形が道路面上か否かを判定する.
- (5) **カメラ映像への危険度情報の付与:** 危険度は障害物までの距離にて定義し, 対応する色でカメラ映像の上に重ね描きする. このときカルマン・フィルタを通すことによって, 距離のデータを平滑化し安定させる.

実装上の特徴としては, (3) の蓋然性の高い道路面を求める処理は動画 1 本に対し 1 回だけ実行する. 3D 特徴点はフレームから独立しているためである. しかし (4) の三角形ごとの道路面判定は, 三角形分割を 2 次元平面上で行う都合上, 直接的には各フレームに対して実行する. つまり動画に含まれるフレーム枚数だけ実行する必要がある. しかもそれを学習段階とテスト段階の 2 回実行する. つまり教師データを取得するためにフレーム枚数だけ実行し, そのあと道路面判定のために再度フレーム枚数だけ実行する.

3.2 Voodoo による 3 次元特徴点の取得

Voodoo は動画 (を分解して得たフレーム画像群) を入力とし, トラッキングできた特徴点の 3 次元位置 (3D 特徴点) や各フレームにおけるカメラ・パラメータを出力するツールである. トラッキングによって得た 3D 特徴点により, フレーム内の (2 次元) 特徴点と 3D 特徴点が対応付けられる. 提案法では Voodoo である必要はなく, 3D 特徴点とカメラの外部パラメータ (つまりカメラの位置と座標系) を出力できるツールであれば何でも良い.

Voodoo の出力する座標値には 2 つの問題がある. スケールに任意性が残る点と, 座標系が基準化されていない点の 2 点である. 前者については次節で解決し, 後者についても次の標準座標系に変換することによって基準化する. 要はカメラが最初に動き始める方向を x 軸, 鉛直方向を z 軸, 両者の外積を y 軸とする.

$$\mathbf{c} = \text{normalize}(\mathbf{c}[t_e] - \mathbf{c}[t_s]) \quad (1)$$

$$\mathbf{z}_{tmp} = \sum_t \mathbf{c}_v[t] \quad (2)$$

$$\mathbf{z} = \text{normalize}(\mathbf{z}_{tmp} - (\mathbf{c}, \mathbf{z}_{tmp})\mathbf{c}) \quad (3)$$

$$\mathbf{x}_{tmp} = \frac{d}{dt}\mathbf{c}[t_s] \quad (4)$$

$$\mathbf{x} = \text{normalize}(\mathbf{x}_{tmp} - (\mathbf{z}, \mathbf{x}_{tmp})\mathbf{z}) \quad (5)$$

$$\mathbf{y} = \mathbf{z} \times \mathbf{x} \quad (6)$$

ただし, t_s は観測開始時刻, t_e は観測終了時刻である.

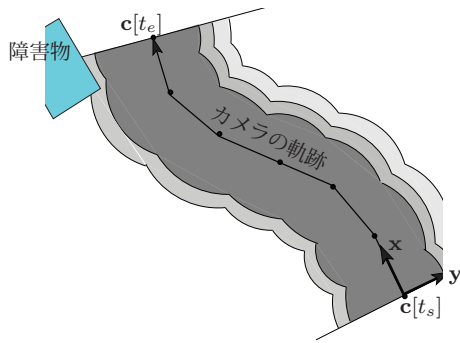


図 1 蓋然性の高い道路面の領域を徐々に拡大して求める。点は Voodoo によって求めたカメラ位置、それをつなぐ折れ線はカメラの時間移動を表す。現在までに道路面とみなしている領域を濃い網掛けで示す。その外側の薄い網掛けの領域が道路面に含まれるかどうかをウェルチの検定により判断する。

また、 $\mathbf{c}[t]$ は時刻 t におけるカメラの位置、 $\mathbf{c}_v[t]$ はワールド座標系で見た時刻 t におけるカメラ座標系の鉛直方向ベクトルであり、共に Voodoo の実行結果から得られる。式 (4) による $\mathbf{x}_{tmp}$ の実際の計算は、時刻 t_s 付近の $\mathbf{c}$ のフレーム間差分を用いる。 $(\cdot, \cdot)$ は内積、 $\times$ は外積、 $\text{normalize}()$ はノルムを 1 にする正規化関数である。

すると 3D 特徴点 $\mathbf{v}$ の標準座標系への変換は次のように表せる。(原点をカメラの初期位置とする。)

$$[\mathbf{x}, \mathbf{y}, \mathbf{z}]^t (\mathbf{v} - \mathbf{c}[t_s]) \quad (7)$$

標準座標系は z 軸が鉛直線に並行となるため、カメラの位置を xy 平面に射影する (x, y 座標値を見る) と車両の水平な動きが把握できる。

3.3 スケール調整

Voodoo の出力は被写体のスケールが正確ではない。これは映像や動画のみを入力とする場合には原理的に避けられない問題である。しかし車載カメラの場合、路面とカメラの距離は一定とみなすことができる。しかもその距離は容易に計測可能である。そこで Voodoo の出力した特徴量に対してスケール調整を行う。

スケール調整は 2 段階に渡って行う。まず xy 平面に射影した点が十分軌跡に近い 3D 特徴点を数十～数百点抽出する。これらの点は道路面上の蓋然性が高いので、 z 座標値 (高さ) の平均を求め、これを Voodoo 座標系におけるカメラ高度とする。これが実測のカメラ高度 (本研究では 1.4[m]) になるように定数倍することによってスケール調整する。これは暫定的なスケール調整である。

さらに、次節の直後に再度スケール調整を行う。次節にて蓋然性の高い道路面を求めると同時にカメラ高度も決まる。それが実測値になるようにスケールを微調整する。

3.4 蓋然性の高い道路面を求める

最初はカメラ軌跡に十分近い (本研究では約 1.4[m]) 3D 特徴点のみを道路面上の点と仮定する。図 1 の濃い網掛け

の領域に含まれる特徴点である。それらの 3D 特徴点の z 座標値 (高さ) の分布を求めておく。次に図 1 の薄い網掛けの領域、つまり現在道路面とみなしている領域よりも軌跡から少し離れた (本研究では 28[cm] 単位) 3D 特徴点の z 座標値の分布 (平均 μ_1 , 標本分散 s_1^2 , 標本数 n_1) を求め、それまでに得ている道路面上の 3D 特徴点の z 座標値の分布 (平均 μ_2 , 標本分散 s_2^2 , 標本数 n_2) と比較する。つまり図 1 の 2 種類の網掛けの領域内の特徴点の分布を比較する。2 つの分布が同じかどうかの検定に通れば、薄い網掛けの領域も道路面とみなす。本研究ではウェルチの検定 [13] を用いる。この処理を、検定に失敗し道路面ではない領域が見つかるまで繰り返す。28[cm] という一定の刻み幅で区切る理由は、平均や標本分散などの統計データの信頼できる値を得るためには、それなりの標本数が存在することが望ましいためである。

まとめると、本節のアルゴリズムは次のようになる。

- (1) カメラ軌跡からの距離に応じ、幅 28[cm] 刻みの小領域ごとに標本数 n , 平均 μ , 標本分散 s を計算。
- (2) カメラ軌跡との距離が約 1.4[m] 以内の 3D 特徴点は強制的に道路面上の点とみなす。
- (3) 領域を幅 28[cm] ずつ広げてはウェルチの検定を実行。カメラ軌跡の右方向に領域を広げて検定し、軌跡の左方向に広げて検定し、右方向にさらに広げて検定、のように繰り返す。検定に失敗したらその方向には領域を広げない。その領域は障害物を含むと解釈する。

3.5 ドロネー三角形ごとに道路面判定

前節により、走行軌跡付近の蓋然性の高い道路面の領域を得た。この領域を道路面とみなし、その情報を教師データとして用い、それ以外の領域の道路面判定を行う。方針としては、まずフレーム内の 2 次元特徴点をドロネー三角形分割 [4] により三角形に分割する。2 次元特徴点を対応する 3D 特徴点に置き換えるとポリゴンを得られる。ポリゴンを構成する各三角形について、それが道路面の一部かどうかを判定する。判定には各三角形の重心と法線角度を説明変数とする尤度関数を用いる。尤度関数は、前節で求めた道路面上の 3D 特徴点のみで構成される三角形を教師データとして用いる。尤度関数の値がしきい値を越える三角形を道路面、つまりその三角形の 3 つの 3D 特徴点は道路面上にあると判定する。

具体的には、まずドロネー三角形分割によって得た三角形のうち、その 3 つの 3D 特徴点がすべて道路面上のものについて、それらの三角形の重心の高さ h の統計データをとり、平均 μ_h と分散 σ_h^2 を得る。同様に法線角度 (法線と z 軸のなす角度) θ についても平均 μ_θ と分散 σ_θ^2 を得る。 (h と θ については図 2 を参照。)

重心の高さ h , 法線角度 θ の三角形が道路面の一部である確からしさ、つまり尤度 $p(h, \theta)$ を、重心の高さ h の尤

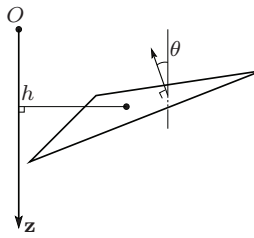


図 2 尤度関数の 2 つの説明変数 h と θ . h は三角形重心の z 座標値. θ は法線と z 軸 (鉛直線) のなす角. 道路面は h がカメラ高度付近の一定値, θ はほぼゼロとなることが期待できる.

度 $p_h(h)$ と法線角度 θ の尤度 $p_\theta(\theta)$ の積にて定義する.

$$p(h, a) = p_h(h) p_\theta(\theta) \quad (8)$$

$p_h(h)$ と $p_\theta(\theta)$ はそれぞれ次のように定義する^{*1}.

$$p_h(h) = \begin{cases} \exp\left(-\frac{(h-\mu_h)^2}{2\sigma_h^2}\right) & (h < \mu_h) \\ 1 & (h \geq \mu_h) \end{cases} \quad (9)$$

$$p_\theta(\theta) = \begin{cases} \exp\left(-\frac{(\theta-\mu_\theta)^2}{2\sigma_\theta^2}\right) & (\theta > \mu_\theta) \\ 1 & (\theta \leq \mu_\theta) \end{cases} \quad (10)$$

尤度 $p(h, \theta)$ がしきい値を越えるドロネー三角形を道路面と判定し, その三角形の 3 頂点を道路面上の 3D 特徴点と判定する. このとき, ある 3D 特徴点を含む三角形のうち一つでも道路面と判定されれば, その 3D 特徴点は道路面上と判定できる.

この手法の利点 フレーム内の任意の点の道路面判定が可能なことである. 前節のウェルチの検定など点を単位とした手法は特徴点以外の任意の点の判定はできない. それに対してこの節の判定法は, 面を単位とした判定法なので, フレーム内の任意の点の道路面判定が可能である.

3.6 カメラ映像への危険度情報の付与

3 次元情報は情報量が多いものの, 実際の運転手への情報提供の方法としては適切かどうかには疑問の余地がある. 運転手には瞬間的な判断が求められるためである. 詳細な情報よりも, 差し迫った危険があるのかどうかの方が重要性が高いと考えられる. そこで本研究では, カメラ映像をベースに危険度情報を色で重ねて表現する. 危険度は障害物への距離にて定義し, 距離に応じて色を変える. 障害物との距離が近く最も切迫した障害物に対しては赤で着色し, 切迫度が低くなるに従って黄色や緑で着色する. 青は道路面と判定された特徴点である.

3D 特徴点はときどき大きな誤差を含む. (スクリーンとは反対側という計算結果を返すことすらある.) そこで画面を小領域に分割し, 小領域を単位に障害物までの距離をカルマン・フィルタで平滑化する. 平滑化の意図は, 3D 特徴点が持つ誤差を時間的空間的に吸収し, 安定化させるた

^{*1} ただし実験では σ_h を 0.15 に強制的に固定している. その理由は, 実際に算出された高さの標準偏差は非常に小さな値になり, 道路面判定の基準が非常に厳しくなってしまうためである.



(a)



(b)



(c)

図 3 実験に用いた動画の一部. 駐車場の空きスペースに駐車するまでの全 15 秒の動画である. 動画そのものは毎秒 30 フレームだが, 10 フレームだけ抽出して利用. 従って実験に用いたフレーム数は 150 枚. (a) はその最初, (b) は 75 枚目, (c) は最後のフレーム. (一部修正しています)

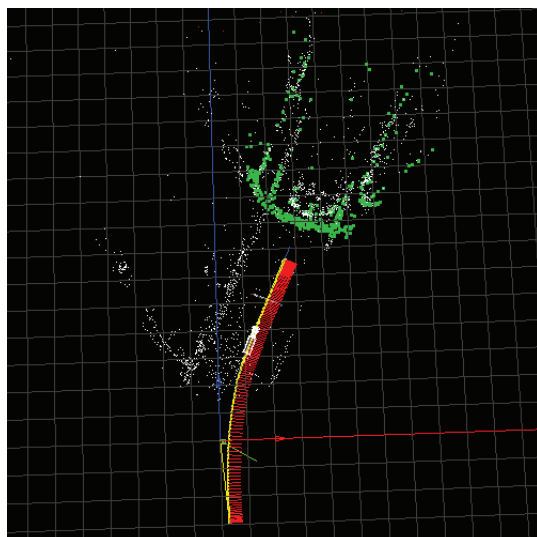
めである. 平滑化した距離に応じて, 赤や青などの色でその小領域を重ね塗りすることによって危険度を表示する.

4. 実験結果

実験に用いた動画のフレームの一部を図 3 に示す. 図 3(a) の右下に見える空き駐車スペースに駐車するまでの 15 秒の動画である. この動画をフレームに分解し, 実験にはそのうち 150 枚を用いる.

4.1 Voodoo による 3 次元特徴点の取得

3.2 節の Voodoo の実行結果を図 4 に示す. 図 4(a) は上から見た図である. レンズの放射歪みの影響で駐車場の白線も歪んでいる. 進行方向に並行な白線に影響は見られないが, 進行方向に垂直な白線は大きく影響を受けている. 図 4(b) は横から見た図である. カメラの軌跡の下に, 軌跡に並行する道路面が判別できる. 右側には駐車中の車の 3D 特徴点も判別できる.



(a)



(b)

図 4 Voodoo によって得た 3D 特徴点とカメラ位置. (a) は上から見た図. (b) は横から見た図.



図 5 ウェルチの検定による蓋然性の高い道路面の判定結果. 水色の点が道路面上と判断された特徴点である. 空きスペース内の特徴点は道路面上と判断できるが, 正面に停車中の車の後方は道路面上とは判断できない.



図 6 式 (8) の尤度関数による道路面の判定結果. 水色の点が道路面上と判断された特徴点である. 図 5 では道路面と判断できなかった正面の車の後方も道路面と判断できた.

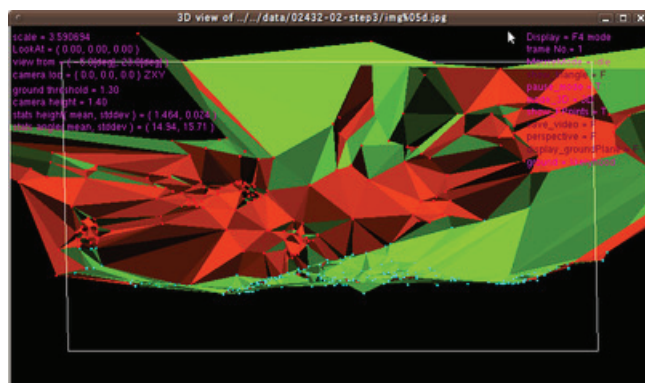


図 7 ドローン三角形分割と 3D 特徴点のデータから得たポリゴンを構成する各三角形について, その法線角度を可視化した図. カメラ・パラメータは図 5 や図 6 と同じ.



図 8 カメラ映像に危険度情報を付加した. 特徴点までの距離や, それを小領域ごとにカルマン・フィルタで平滑化した距離に応じて色で塗り分ける. 青は道路面と判定された点や領域, 障害物は赤, 黄色, 緑の順で切迫度が高い.

4.2 スケール調整

3.3 節のスケール調整の結果は見掛け上は図 4 と変わらない. 座標軸の取り方とスケールの違いだけだからである.

4.3 蓋然性の高い道路面を求める

3.4 節の実行結果を図 5 に示す. 道路面上であると判断された 3D 特徴点を水色で示す. 入力動画は, 右下の空きスペースに駐車するまでの動画である. 空きスペースに侵入する映像があるため, 空きスペース内の特徴点は道路面上と判断できる. しかし正面に停車中の車の後方は道路面上とは判断できない. 正面の車の 3D 特徴点が, 道路面上とは異なる分布をしているためである. 尚, この図は Voodoo によって計算された 3D 特徴点にカメラ映像をテクスチャ・マッピングすることによって作成した 3 次元の図である.

4.4 ドローン三角形ごとに道路面判定

3.5 節の実行結果を図 6 に示す. 道路面上であると判断された 3D 特徴点を水色で示す. 図 5 では道路面と判断できなかった正面の車の後方も道路面と判断できていたことは注目に値する. このような判断が可能になったのは, 正面の車の後方に位置する ドローン三角形の h や θ が, 右

下の駐車スペース内のドロネー三角形の h や θ に似た分布になっていることを利用しているためである。

3.5 節の手法は三角形を単位に道路面判定する。それを示すのが図 7 である。フレーム内で 2 次元のドロネー三角形分割のあと、各頂点に対応する 3D 特徴点からなる 3 次元座標上の三角形を構成する。それが図 7 の三角形である。これらの三角形は全体としてポリゴンを構成する。色は法線角度を表す。法線角度が 45 度以下なら緑系の色、45 度以上なら赤系の色で表示している。45 度に近いほど黒っぽく、45 度から離れるほど明るい。道路面は緑系、停車中の車の側面は赤系の色になっている。

4.5 カメラ映像への危険度情報の付与

3.6 節の実験の実行結果を図 8 に示す。危険度とは特徴点までの距離であり、距離に応じた色で着色した。距離 3m 以下は赤、6m は黄色、9m 以上は緑を対応させた (中間色も線形補間して使用)。道路面は青で表現した。さらに画面を 40×22 の小領域に分け、特徴点までの距離をカルマン・フィルタで平滑化し、同じ基準で着色した。図 8 は道路面をほぼ正確に判定できた。さらに障害物についても、正面の車の右後方のテールランプや後輪付近がオレンジ色、つまり最も距離が近いことを示す。また車体の前方や左隣の車へ向かうにつれて黄色から緑色に変化する。つまり距離がだんだん離れていくことを示す。これは実際の距離をほぼ正確に反映している。

4.6 提案法のオンライン実行のシミュレーション

実際の理想的な障害物検知はオンラインで実行し、運転中の運転手に情報提供することである。そこで、より実際の状況に近い条件で実験する。ここで試みる実際の状況とは、未来のフレームを使わないことである。つまり最も着目すべき現在のフレームとそのわずかに過去のフレームのみを用いて、現在のフレームの障害物を検知する。前節までの結果は、動画の全フレームを用いて障害物検知を行っていた。少なからず未来のフレームから得られる情報を用いた障害物検知になっていた。(しかも検証するときは往々にして時系列の冒頭のフレームに着目しがちであった。)しかし本来の研究課題はオンライン実行であり、未来のフレームを用いることはできない。そこで未来のフレームを全く用いずに障害物検知を行った場合にどのような結果が得られるのか実験した。

この節で用いた動画を図 9 に示す。図 9(a) の左上にわずかに見える空き駐車スペースに駐車するまでの 23 秒の動画である。この動画をフレームに分解し、実験にはそのうち 230 枚を用いる。ただし同時に用いるのは、想定上の現在のフレームを含む直近 100 枚のフレームに限る。

その結果を図 10 に示す。すなわち図 10(a) は 110 番フレームの時点におけるオンライン実行を想定し、11 から



(a)



(b)



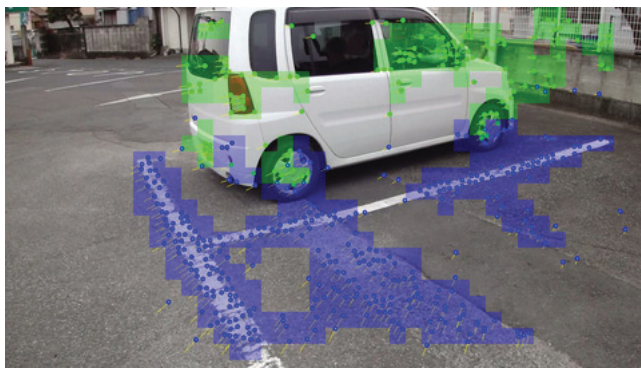
(c)



(d)

図 9 実験に用いた動画 (その 2) の一部。駐車場の空きスペースに駐車するまでの全 23 秒の動画である。動画そのものは毎秒 30 フレームだが、10 フレームだけ抽出した。従って実験に用いたフレーム数は 230 枚。(a) はその最初、(b) は 78 枚目、(c) は 155 枚目、(d) は最後のフレームである。

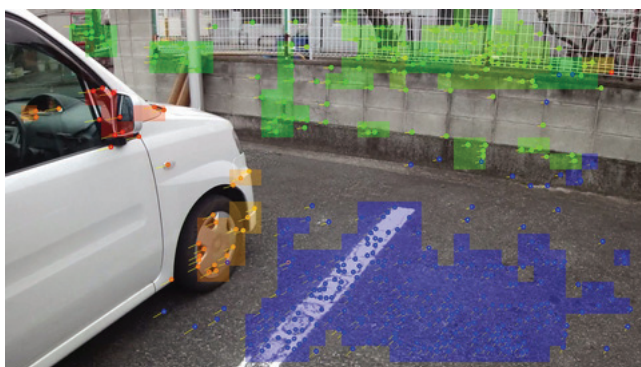
110 番までの 100 フレームを用いた実行結果のうち、最後のフレームの結果を示す。(b) は 130 番フレームの時点想定し、31 から 130 番までの 100 フレームを用いた実行



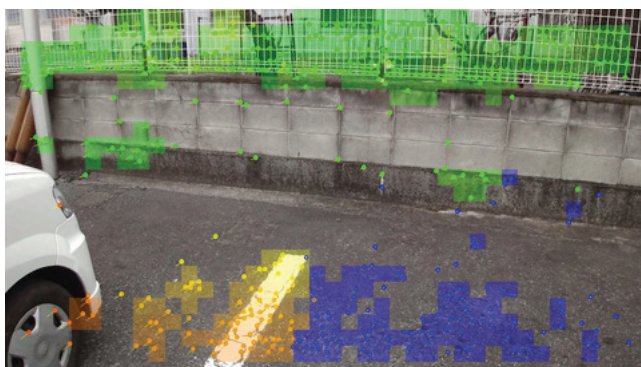
(a)



(b)



(c)

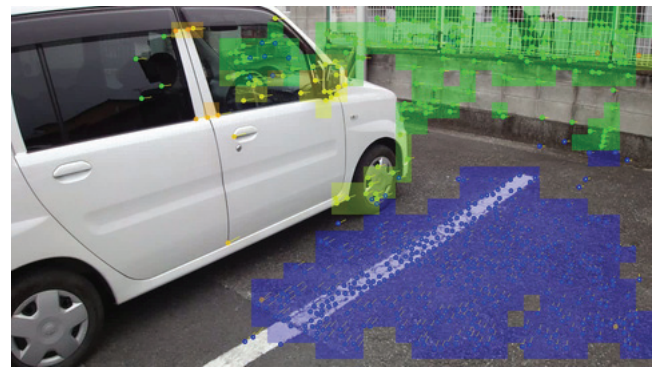


(d)

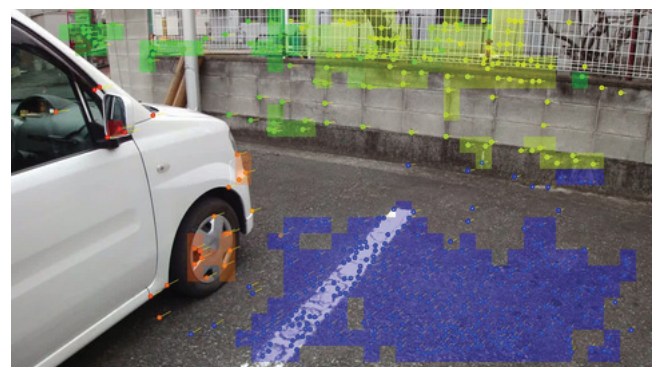
図 10 図 9 の動画 (その 2) から 100 フレームの短い動画を切り出して実行. すなわち (a) は 110 番フレームの時点におけるオンライン実行を想定し, 11 から 110 番までの直近 100 フレームを用いた結果のうち, 最後のフレームの結果. 同様に (b) 以降も 20 枚ずつフレームをずらした 100 枚のフレームを用いた, 最後のフレームの結果. 道路面の誤判定が見られる.



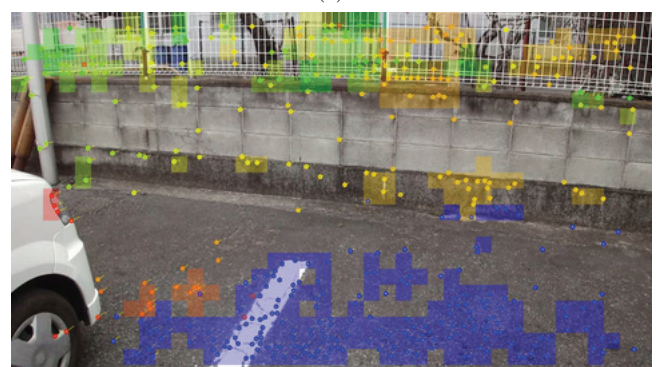
(a)



(b)



(c)



(d)

図 11 図 10 と同じく, 図 9 に示す動画 (その 2) から 100 フレーム (10 秒) ずつ切り出して実行した結果. 図 10 との違いは, Voodoo を実際の時系列とは逆向きに実行したことである. 道路面の誤判定は殆ど見られない.

結果のうち、最後のフレームの結果を示す。同様に (c) 以降も 20 枚ずつフレーム番号をずらした 100 枚のフレームを用いた、最後のフレームの結果である。図 10 の結果は、障害物検知が一部不安定になっている。例えば、図 10(b) や (d) に映っている白線の端である。道路面上であるにもかかわらず、障害物と誤判定してしまっている。

このような誤判定が生じる理由は、Voodoo が画像の特徴点を切り出しそれをトラッキングする際、時系列順に追跡することが原因と考えられる。その根拠^{*2}となるのが図 11 である。図 11 は、図 10 と同じく図 9 の動画 (その 2) を用いた結果である。いずれも 100 枚のフレームを用いており、その区間も図 10 に対応する。違いは Voodoo を実行させる際に、フレーム画像群を実際の時系列とは逆向きに番号付けして入力^{*3}したことである。それだけの違いで安定した障害物検知の結果を得た。道路面の誤判定は殆ど見られない。もし Voodoo が入力フレーム画像群の時系列に依らず本質的に同じ結果を返すなら、図 10 と図 11 の道路面判定も同じ結果となるはずである。しかし実際には同じ結果にはならない。これは Voodoo の実行結果が入力フレーム画像群の時系列に依存することを示している。しかもフレーム番号の若いフレームにとって都合の良い依存性がある。オンライン実行では逆に、フレーム番号順では最後のフレームの重要性が高く、それは Voodoo のようなトラッカーにとっては必ずしも力を発揮できる状況にはない。

結論として、オンラインでの障害物検知を Voodoo を用いて行う場合、時系列とは逆向きのフレーム群を入力として実行すべきである。

5. まとめ

単眼車載カメラの情報から障害物を検知する方法を提案した。その手順は、まず前処理として Voodoo のようなカメラ・トラッカーの実行とスケール調整によって 3 次元特徴点を得る。次に車両の走行軌跡周辺には確実に障害物がないことを利用して高い蓋然性で道路面と判断できる領域を求める。その際ウェルチの検定を用い、道路面上の特徴点の分布を変えない範囲で軌跡からの距離を大きく取れるだけ大きく取る。最後に、車両の走行軌跡以外の領域でドロネー三角形ごとに道路面判定を行う。その際、既に道路面と判断している領域のドロネー三角形の高さと法線角度のデータを教師データとして構成した尤度関数によって判定する。ドロネー三角形を単位とした道路面判定のため、フレーム内の任意の点に対する道路面判定も可能である。

さらに、検知した障害物の表示手段として、瞬間的な視

認性を上げるためカメラ映像の上に危険度を色で着色した。3D 特徴点までの距離と、それをカルマン・フィルタで安定化させた距離を危険度としてカメラ映像に重ね描きした。

また、オンライン実行をシミュレーションした。実際の車載カメラによる障害物判定はオンラインで実行し、未来のフレームを参照することはできない。そこでそのような条件のもとで障害物判定の実験を行った。その結果は必ずしも芳しくはなかったが、入力画像群を時系列とは逆向きに与えると改善した。これは Voodoo のトラッキングの方向性に大きく依存していると考えられる。最新のフレームの情報量が高いオンライン実行の際には、最新のフレームを起点に時系列順とは逆向きにトラッキングすべきである。

参考文献

- [1] Voodoo camera tracker. <http://www.digilab.uni-hannover.de/docs/manual.html>.
- [2] Margrit Betke, Esin Haritaoglu, and Larry S. Davis. Real-time multiple vehicle detection and tracking from a moving vehicle, 2000.
- [3] Christophe Braillon, Cédric Pradalier, James L. Crowley, and Christian Laugier. Real-time moving obstacle detection using optical flow models. In *in Proc. of the IEEE Intelligent Vehicle Symp., Tokyo (JP), IEEE*, pp. 466–471, 2006.
- [4] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 1992.
- [5] Zehang Sun, Ronald Miller, George Bebis, and David DiMeo. A real-time precrash vehicle detection system. In *IEEE International Workshop on Application of Computer Vision*, pp. 171–176, 2002.
- [6] Ashit Talukder and Larry Matthies. Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. *IEEE Int. Conf. on Intelligent Robots and Systems (IROS 2004)*, Vol. 4, pp. 3718–3725, 2004.
- [7] 伊東敏夫, 内藤崇行, 金田悠紀夫. 画像中のオブティカルフローによる障害物検出手法. 自動車技術会論文集, Vol. 40, No. 2, pp. 441–446, 2009.
- [8] 岡田隆三, 谷口恭弘, 小野口一則. 仮想平面追跡法を用いた単眼車載カメラによる障害物検出. 電子情報通信学会信学技報 (PRMU), Vol. 09, pp. 29–36, 2001.
- [9] 久徳遙矢, 山口大輔, 高橋友和, 目加田慶人, 井手一郎, 村瀬洋. 過去の車載カメラ映像との差分による道路上の不特定障害物検出. 画像の認識・理解シンポジウム (MIRU) 2010 (OS13-3), pp. 1547–1544, 2010.
- [10] 栗林宏輔, 子安大士, 前川仁, 小野晋太郎, 川崎洋, 池内克史. 平面群構造を利用した車載カメラ映像からのオンライン前景障害物除去システム. 第 9 回 ITS シンポジウム, pp. 121–126, 2010.
- [11] 五十部宏幸, 中村靖. 単眼車載カメラ画像を利用した自動車運転支援. 広島工業大学紀要研究編, pp. 317–322, 2009.
- [12] 山中隆, 古賀由紀夫, 外園宙, 山田純一, 渡邊睦. 規範フローと危険度を用いた障害物検出の研究. 情報処理学会研究報告 2007-CVIM-158, pp. 41–48, 2007.
- [13] 東京大学教養学部統計学教室 (編). 統計学入門. 東京大学出版会, 1991.
- [14] 楊長輝, 本郷仁志, 谷本伸一. 単眼車載カメラを用いた路面動き補正による障害物検知. 精密工学会誌, Vol. 75, No. 2, pp. 278–283, 2009.

^{*2} 他の根拠として、図 10 には実験に用いた各 100 枚のフレームの最後のフレームの結果を載せているが、最初のフレームの結果は実はもっと期待に沿ったものになることから分かる。

^{*3} 図 1 で進行方向の前後の領域を切り捨てていたが、その理由は時系列とは逆向きの入力画像を与えた場合にも障害物に惑わされないためである。