

規範フロー場と実フロー場の差異解析による 路上障害物の検出と危険度提示に関する研究

山中隆 古賀由紀夫 外園宙 山田純一 鹿嶋雅之 佐藤公則 渡邊睦
鹿児島大学 大学院 理工学研究科
〒890-0065 鹿児島市郡元 1-21-40 Tel:099-285-8453

あらまし オプティカルフロー場を解析することによる外界認識の研究が行われている。しかしながら、路上駐車など複数の障害物が存在する状況で、運転者に適宜その危険度を提示するシステムは未だ研究段階である。我々は、障害物が存在しない標準的な環境で想定されるオプティカルフロー場を「規範フロー」として予め計算により求め、ビデオカメラから入力した画像を処理して得たオプティカルフローとの差異を検出、解析することにより静止障害物を検出する方式の研究を行ってきた。本論分において我々は、本年3月に提案した手法に対し、障害物検出、危険度提示の部分に関する改良を行った。屋外シーンによる実験を行い、提案手法の有効性を示す。

キーワード 動画像解析、障害物検出、車載カメラ、オプティカルフロー、規範フロー、KLT法、危険度

Research of the risk degree demonstration and obstacle detection by analyzing a difference of optical flow and model flow

Takashi YAMANAKA, Yukio KOGA, Hiroshi HOKAZONO, Junichi YAMADA,
Masayuki KASHIMA, Kiminori SATO, and Mutsumi WATANABE
Graduate School of Science and Engineering, Kagoshima University
1-21-40 Korimoto Kagoshima, 890-0065, Japan

Abstract. Many researches have been done as for outdoor scene recognition by analyzing optical flow information, however, a practical driver-oriented system has not been proposed yet that notices the "risk degree" when multiple obstacles exist, such as, parking cars and pedestrians. We have been doing research on the method to detect stand still obstacles using the "model flow" corresponding to optical flow field where there is no obstacle. The model flow is calculated using the knowledge as for navigation environments. Stand still objects are automatically detected by analyzing the difference between a model flow field and a real optical flow field which is acquired by an in-vehicle camera image. We improved the method proposed in Mar.2007, as for obstacle detection and the risk degree demonstration. Experimental results in outdoor scenes have shown the effectiveness of the proposed method.

Keyword. motion analysis, obstacle detection, in-vehicle camera, optical flow, model flow, risk degree

1. はじめに

自動車関連技術における主要課題は快適性、走行性から安全性の追求へと変化してきている。既にABSやエアバッグなどで搭乗者の安全を確保する技術はほぼ確

立しており、現在では事前に危険を察知して回避する性能の確立や、歩行者の自動検出など外界への安全性を高めるための研究が盛んに行われている[1]。
外界の安全確保のために認識すべき道路の状況とし

て、カーブなどの道路形状、歩行者等の移動障害物、停止車両などの静止障害物などが挙げられる。我々はこれまでに静止障害物を対象とし、動画画像解析手法を用いて自動的に検出する手法の開発を行ってきた[2] [3]。

2. 研究目的

交通事故件数を法令違反別に見ると、安全不確認が構成率 31.1%を占め最も多く、次に脇見運転が構成率 16.6%と続いている[4]。事故の原因として最も多いのは運転者の危険に対する認識不足だと考えられる。このことからシーン中の危険を提示することにより、運転者の危険に対する認識を補助することが交通事故を未然に防ぐ有効な方法の一つであると考えられる。このことから運転者への障害物提示による運転補助を目的とした。

障害物検出に用いる特徴は距離情報と動き情報に大別される。距離情報を用いるものには、距離センサを利用する方法[5]や、ステレオ視を利用する方法[6]がある。動きの情報を用いるものには FOE を利用する方法[7]が従来研究されている。

我々は、検出した障害物の情報を運転者に確かかつわかり易く提示する手法の開発を目的としている。この目的においては、距離情報を高精度に求めることよりは、広範囲における障害物の状況を認識することがより重要である。また車にセンサを装着する際の簡便性も要求される。以上より、単眼視を用いる手法が適切であると考へ、採用することとした。

3. 本研究の手法

3.1. 規範フロー

単眼での処理が可能な動き情報を用いて障害物を検出、認識する手法としてオプティカルフローと FOE (Focus of Expansion: 動きの消失点)を利用する手法[7]が提案されている。この FOE の違いを調べることによって背景と違う運動をする移動物体の検出が可能となる。しかし静止障害物は背景と同じ FOE を持つため、背景と静止障害物を識別することはできない。そこで、我々は[2]において移動状態で静止障害物を検出するために、『規範フロー』という概念を提案した。

標準的な環境、標準的な移動状態から予測される、画像中の各場所におけるオプティカルフローを計算により求め、これを規範フローとする。そして動画画像処理により得られた車載カメラ画像のオプティカルフローと

の差異を検出することにより、障害物を検出する。

オプティカルフローの差異検出は局所的なベクトルの減算という簡単な減算処理であるため、処理負荷も軽く、実用的な処理速度で処理できる特長を有する。

3.2. 危険度

運転中の人間に対して過度の情報を与えることにより運転への集中力低下や、判断の遅延などの問題が生じる。そのため運転者に必要不可欠な情報のみに限定し、通知する場合も必要に応じて注意を促さなければならない。我々は障害物検出結果から得られる情報を「危険度」に変換し運転者にわかり易く提示する方式を考案し研究を行ってきた。

3.2.1. 危険度の計算方法

前回の発表[3]では危険度には衝突時における観測者に対する障害物の相対的な運動エネルギーの仮定に基づき危険度を求める手法について述べた。運動エネルギーの計算に必要な質量は、画像中の対象領域の面積に基づいて仮定し、速度は、対象の接地部分までの距離とフローの値により仮定している。しかし複雑な背景や多くの物体を含む実際の利用シーンで実験を行ったところ、対象領域を正確に抜き出すことは困難であり適切に提示できないという問題が生じた。

今回は検出した障害物の位置と自転車の走行状態から対象が自転車の進路上に存在しているかを新たに判別し、自転車と衝突する可能性を提示し、これを危険度として用いた。

4. 処理の流れ

本手法の処理の流れを図 1 に示す。図中の番号は手法を説明している節番号である。

まず自動車の速度やハンドルの切り角などの情報を車載のセンサから取得し、仮想空間上の路面に配置した特徴点の座標を移動させ、その動きを画面に投影したオプティカルフローを算出する。ただし今回は PC 上で実現するためにスライドバーにより値を設定することとした。この生成プログラムから算出されるオプティカルフローを規範フローとし、KLT 法により実画像のフローを抽出する。次に、画面をブロックに分けて各ブロックに規範フローの長さや向きを設定し、実画像のフローの長さや向きと、そのフローの入っているブロックの規範フローとの差異がないかを照合する。

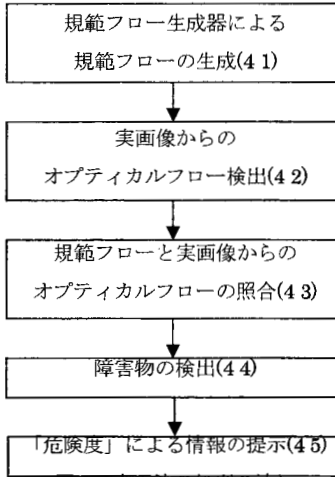


図1 本手法の処理の流れ

差異があると判断したフローがブロックの中に存在する場合、そのブロックに障害物が存在するとして、ブロック周辺を含む領域を障害物領域とする。

障害物領域が検出された位置から算出した「危険度」を基に運転者への情報の提示を行う。

以下各処理について、規範フローの生成、入力画像中の特徴点抽出とフロー検出、規範フローと実フローの照合、障害物の検出、「危険度」による情報の提示の順に説明する。

4.1. 規範フローの生成

4.1.1. 生成アルゴリズムの概要

まず固定パラメータとして、カメラの路面からの高さ、カメラの焦点距離、路面に対するカメラの傾き、車の前輪と後輪の間隔、車の右の車輪と左の車輪の間隔の5つのパラメータを設定することにより、仮想空間上の路面に特徴点を配置する。

次に変数パラメータとして、スピード、タイヤの切り角を毎回受け取り、車の状態にあわせて特徴点の座標を変更する。特徴点の位置を、はじめに設定した固定パラメータの情報から計算し、カメラ画像上の投影位置を算出する。前状態と現状態をそれぞれ規範フローの始点と終点として出力する。

4.1.2. 画面投影の計算方法

プログラム上の空間と画面の位置をピンホールカメラに近似させ、図2に示すように視点を原点とし、y軸を奥行きとした右手系座標で、カメラの焦点距離を f とすると、画面(CCD)の位置は(1)式で表される平面で

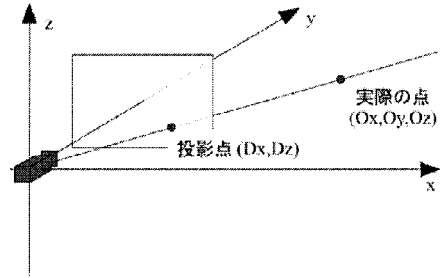


図2 投影座標系

ある。

$$y = f \cdot \cdot \cdot (1)$$

また、画面の範囲は CCD の中心を y 軸とする CCD の幅になる。画面に投影される点は、上記平面、特徴点、原点を結ぶ直線の交点になる。

特徴点の座標を (Ox, Oy, Oz) 、投影点の座標を (Dx, Dz) とすると、 Dx, Dz は(2)式、(3)式で求められる。

$$Dx = \frac{Ox}{Oy} * f \cdot \cdot \cdot (2)$$

$$Dz = \frac{Oz}{Oy} * f \cdot \cdot \cdot (3)$$

この計算により画面の中心を原点とした z-x 平面上で投影点を求める。その後、CCD の大きさを、実際の入力画像の大きさに変換することにより入力画像に応じた規範フローを求める。

4.1.3. 特徴点の配置

特徴点はカメラ座標系上に等間隔に配置された「基準点」を、カメラの設置状態、車両の走行状態などの環境情報によって適応的に路面上の「環境点」に射影し、さらに環境点を画面上に再投影したものである。この模式図を図3に示す。

カメラを路面に対して θ 傾けて設置した場合、図4に示すように環境情報を基に基準点を、 θ 傾けた状態の環境点に射影する。

4.1.4. 特徴点の移動

車両の走行状態からカメラに対する特徴点位置の移動を再現するために、車両の速度情報による基準点の移

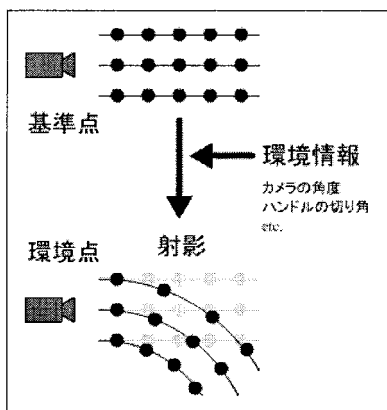


図3 特徴点配置の模式図

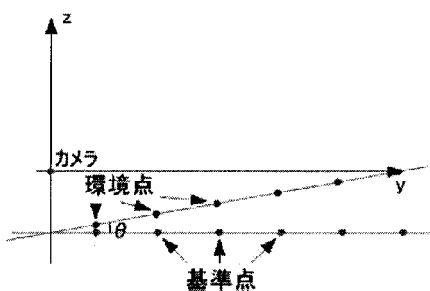


図4 カメラの傾き θ の時の環境点の配置

動を行う。基準点を移動させることで射影される環境点の移動を再現する。

4.1.5. 規範フローの生成

移動前、移動後それぞれの環境点を投影した位置を規範フローの始点、終点としてベクトルを生成する。

4.2. 入力画像中の特徴点抽出とフロー検出

オプティカルフローの検出は、連続する2枚の入力画像の対応点を見つけることにより行われる。まず、1枚目の画像に対してある点を中心とした一定の範囲の画素に対して輝度分布を調べ、 x 軸方向と y 軸方向共に分散が大きくなる部分を特徴点として抽出する手法[8]を用いて特徴点を抽出する。

そして2枚目の画像の対応点検出によりオプティカルフローを検出する。本システムでは高速かつ高精度で検出することが可能なKLT法[9]を用いて入力画像のオプティカルフローを検出した。

4.3. 規範フローと実フローの照合

4.3.1. フロー照合用ブロックの設定

本システムでは各場所における規範フローと、入力画

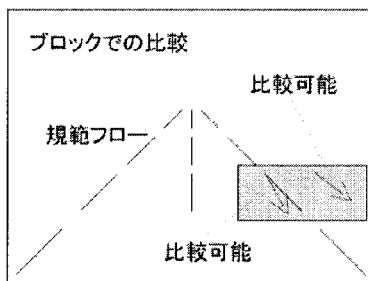
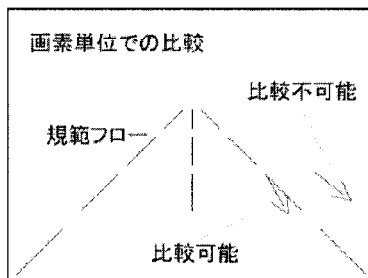


図5 ブロック比較の模式図

像のオプティカルフローを比較して障害物の検出を行うため、場所と規範フローの関連付けを行う必要がある。

規範フローは計算によって画面全体に生成されるが、実フローはテクスチャに依存して特徴の弱いところでは検出されないといった密度の違いが生じる。

そこで規範フローを基にフロー照合用のブロックを設定し、規範フローのパラメータをブロックの範囲に拡張する。この模式図を図5に示す。この手法により、場所と規範フローの関連付けと、処理の効率化を実現した。また、1対1のベクトル比較ではなく、領域での基準との比較により、集中して検出されたベクトルもすべて比較対象を持たせるように構成した。

4.3.2. ブロックの設定方法

ブロックの大きさや配置を固定的に決めると、規範フローの変化によって複数の規範フローが一つのブロックに含まれる可能性や、ブロック中に規範フローが含まれない可能性が生じる。そこでフロー照合用のブロックは各規範フローにおける中点の位置をブロックの中心とし、規範フローの始点、終点の位置をそれぞれブロックの上端、下端の座標に設定することで、ブロックと規範フローが1対1の対応となり、規範フローの変化とともにブロックも配置や大きさを変更するよう設定した。

ただし直進時の規範フローでは水平成分の値が小さ

くなり、規範フローの両端座標をブロックの両端座標とすると、領域単位の比較というブロックの有効性が低下する。そこでブロックの水平成分を定数とすることで一定サイズの領域が確保できるよう設定した。

前回提案したシステム[3]では規範フローを生成するために用いられる環境点の配置に関して、自車の幅情報を用いて自車が通過する領域のみに制限することで処理領域を進路上のみに限定し、模擬環境での有効性を確認した。

しかし実環境においては、進路上の端に存在する障害物に関して、障害物領域の高い部分が処理範囲に含まれず検出が困難となり、また進路からは外れていても近傍で予期しない発進などにより接触の危険性がある停止車両等が検出できないという問題点があることが判明した。今回はこの課題を改善するために、環境点を配置する範囲を車体の通過する領域に加え近傍領域まで拡大することで、障害物検出領域を追加することとした。

4.4. 障害物の検出

4.4.1. 障害物フローの判断

オプティカルフローの持つ情報の中で、カメラの進行方向に対する移動量が危険度に影響すると考えられる。入力画像のオプティカルフローの中で、フロー照合用のブロックに含まれているもの全てについて、そのフローの画像中の z 軸成分を規範フローの同成分と比較し、規範フローよりも大きい場合に障害物のフローとして検出する。

4.4.2. ブロックでの障害物判定

特徴点はテクスチャのある障害物領域に集中して抽出される。すなわち障害物を含むブロック内に多数の障害物のフローが含まれる。これに対して路面上に検出される誤りフローは、一般的に特定領域に集中して検出されることは少ないため、1 ブロック内に含まれた障害物検出誤りに関わるフローは少数であると想定できる。

このことからブロック内に含まれる障害物のフローの数に対して閾値処理をほどこし、ブロック内の障害物フローの数が閾値を越えた場合のみ、障害物領域の一部であると判断することとした。

4.4.3. 障害物を含むブロックの統合

隣接するブロックで障害物が検出された場合、これらの隣接するブロックに含まれる障害物を、まとまった一

つの障害物だと仮定する。ブロックごとに障害物の情報を取得するのではなく、ブロックを統合して一つの障害物として情報を取得する。

4.5. 危険度による情報の提示

4.5.1. 危険度の判定

過剰な警告を避けるために処理領域を限定した結果、特に進路以外の領域で検出に使用できる領域が処理領域に含まれる障害物の一部に限定される。このため、利用できる障害物のフローが制限され、対象を安定に継続検出することが困難となる。

一方、人間が一度に認識・記憶できる対象の数は有限 (7 ± 2 チャンク程度) であることが知られている。検出した全ての障害物を一様に運転者に提示し続けることは過剰な情報提示になりかねず、本来の運転動作に障害を生じる危険性がある。見逃しの排除という観点からは、すでに提示した障害物に比べ、新たに検出された障害物の情報提示がより重要であると考えられる。

このことから新たに検出した障害物領域に対しては、明度の高い警告色で表示し、既出の情報に対しても見逃しを防止するために明度を下げて一定時間表示し続けることとした。

各ブロックにはそれぞれ 1 つの規範フローが対応している。規範フローは生成の説明にある基準点の配置において、カメラ座標系の x 軸方向の範囲として自車の幅情報を用いることによりその特徴点が進路上に配置された点であるかが判別できる。このとき進路上に存在する特徴点から生成された規範フローを基に設置されたブロックにより検出された障害物は、自車の進路上に存在することになり、衝突の危険性が高くハンドル操作や減速などの回避行動が必要となる。この場合は「非常に危険」として提示する。

また停止車両などの障害物は進路上に存在しなくても対象が動くことによって衝突の可能性があるため、通過する位置に存在する特徴点の近傍に存在する特徴点から生成された規範フローを基に設置されたブロックにより検出された障害物にも十分な注意が必要である。この場合は「注意が必要」として提示する。

4.5.2. 危険度の表示

「非常に危険」として検出された領域を赤色で表示し、「注意が必要」として検出された領域を黄色で表示する

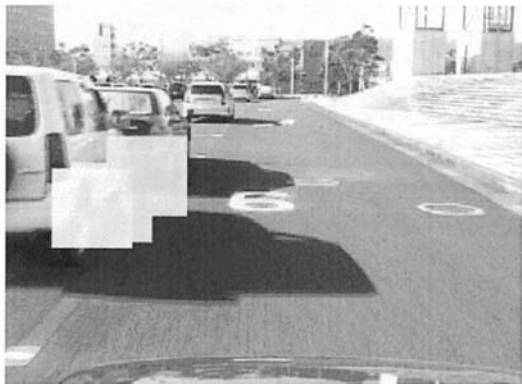


図 6 停止車両検出例

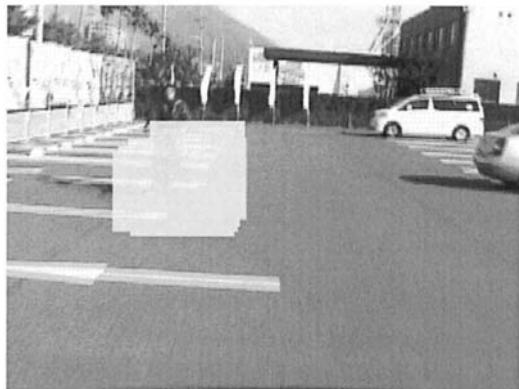


図 9 すれ違う自転車の検出例

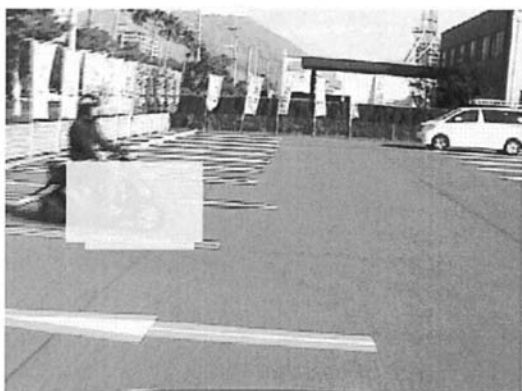


図 7 二輪車検出例

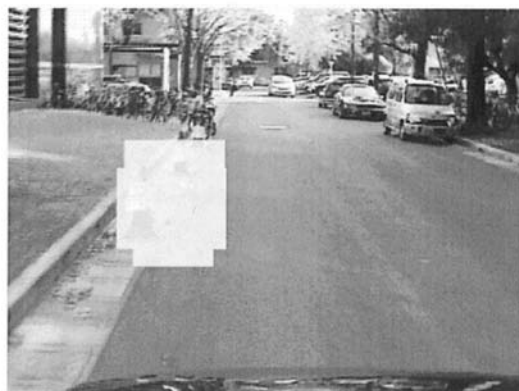


図 10 低い障害物の検出例

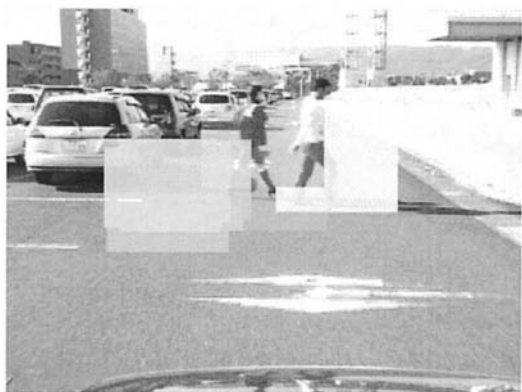


図 8 横断する歩行者及び近傍の停止車両検出例
こととした。

表示された領域は、検出した時点では高い明度で表示し、時間経過とともに警告色の明度を低くして原画像に戻していく。障害物の画像中の位置は時間経過とともに移動するため、領域が検出された位置によって移動する障害物を追跡するように警告領域を移動表示させるこ

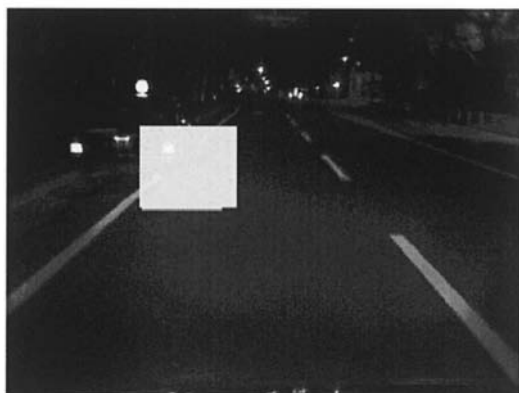


図 11 夜間における車両検出例
こととした。

5. 実験結果

5.1. 実験条件

前章で述べた方式に基づく移動視点からの静止障害

物検出システムを汎用 PC 上のソフトウェアとして試作し、提案手法の妥当性を確認する実験を行った。

実験に使用した計算機の CPU は Intel Xeon processor 3.6GHz、Memory は 2.0GB、OS は Microsoft Windows XP Professional、プログラミング言語は Microsoft 社製 Visual C++ .NET である。パソコンへの動画像の取り込みには Matrox 社製 Meteor II キャプチャボードを用い、処理画像の解像度は VGA(640×480)、カメラは SONY 社製の DCR-TRV30 を使用した。

晴天時の昼間および夜間の実環境シーンを数日に分けて取得した車載カメラの映像に対して実験を行った。実験車両にはスバル製インプレッサを使用し、ルーフ部分にカメラを取り付けた。

5.1. 危険度による障害物領域の提示実験

図 6 に降に様々な対象における障害物検出例を示す。

図 6 は停止車両検出例である。障害物として検出するために自車の横 1m 以内を通過するように走行した。進路近傍領域に存在する障害物として停止車両が検出されていることが確認できる。

図 7 は二輪車の検出例である。路面には白線が引かれているが二輪車の領域のみが進路近傍の障害物として検出されていることが確認できる。

図 8 は横断する歩行者及び近傍の停止車両検出例である。人物領域は進路上にあるため危険度の高い警告色で表示され、同時に検出されている進路の近傍に位置する停止車両は危険度のやや低い色で表示されていることが確認できる。

図 9 はすれ違う自転車の検出例である。停止車両や横向きの二輪車に比べて画像中の領域としては小さいがしっかりと検出されていることが確認できる。

図 10 は低い障害物の検出例である。自動車や歩行者などに比べて高さの低いパイロンが検出されていることが確認できる。

図 11 は夜間における車両検出例である。夜間においても自車のヘッドライトに照らされている領域ではオブティカルフローが検出されるため、停車している障害物を検出できていることが確認できる。

5.2. 検出精度評価実験

表 1 に各シーンにおける対象の位置と検出率を示す。対象数は検出対象となった障害物の数を表し、検出数は

表 1 各シーンにおける対象の位置と検出率

シーン1

種別	危険度	対象数	検出数	検出率
自動車	非常に危険	0	0	-
	注意が必要	17	16	94.1%
人	非常に危険	8	8	100.0%
	注意が必要	0	0	-

シーン2

種別	危険度	対象数	検出数	検出率
自動車	非常に危険	1	1	100.0%
	注意が必要	2	2	100.0%
人	非常に危険	0	0	-
	注意が必要	3	3	100.0%
自転車	非常に危険	0	0	-
	注意が必要	2	2	100.0%
二輪車	非常に危険	0	0	-
	注意が必要	1	1	100.0%

シーン3

種別	危険度	対象数	検出数	検出率
自動車	非常に危険	0	0	-
	注意が必要	31	26	83.9%

シーン4

種別	危険度	対象数	検出数	検出率
自動車	非常に危険	0	0	-
	注意が必要	2	2	100.0%

各障害物を一回以上検出した数を表す。

シーン 1 は晴天時の昼間に進路脇に数台の停止車両が存在し、進路上を横切る人物が存在する状況、シーン 2 は晴天時の昼間に進路脇を自転車や歩行者が通行し、前方を自動車が横切る状況、シーン 3 は晴天時の夕方に日陰になる路上で、進路脇に数台の停止車両が存在する状況、シーン 4 は晴天時の夜間に進路脇にライトを点灯して停止している自動車 1 台とライトの点灯していない停止車両 1 台が存在する状況である。

すべてのシーンを通して障害物をシステムが検出した回数に対する、誤報の割合は 2.0%となった。そのうち昼間であるシーン 1 およびシーン 2 で集計した誤報の割合は 0.9%となった。

6. 考察

検出率について全体では 91.0%という結果になったが、危険度別に見ると非常に危険な進路上での検出率は100%であった。このことから、今回作成したシステムは実環境における障害物の危険度提示に関して十分な性能を有すると考えられる。

障害物検出に失敗したのは、障害物領域が処理領域の一部のみしか含まれない場合、手前に存在する他の障害物に隠れているなどして十分なフローが処理領域に検出されなかった場合、画像全体の明度が低くなったために対象にフローが検出十分に検出されなかった場合であった。また、検出された障害物の中には自転車に接近してから検出されるものも存在した。

障害物領域が処理領域の一部のみしか含まれない場合に関しては進路から離れており危険度が低いため、重大な問題にはならないと考えられる。

一方、処理領域の設定に関しては未だ検討の余地があり、処理画像内で障害物フローを検出した領域では高さ方向にも処理範囲を拡大するなどの、設定の改良によって適切な検出が可能になると考えられる。

他の障害物に隠れている場合も、処理領域の配置に関する改良と拡大した処理領域における検出の条件などにより検出率を上げることが可能だと考えられる。明度低下による検出失敗に関しては入力画像に対して明度変化を強調させるフィルタを掛けるとともに、暗い画像用にフロー検出に関わる設定値の調整をすることにより検出率を上げることが可能だと考えられる。

昼間における誤報は 0.9%と低い値になっている。これに対し夜間は自転車の移動によってヘッドライトで照らされている範囲が変化するため、フローの対応付けに誤りが生じてしまい路面や道路標示に対する誤報が増加すると考えられる。この改善も今後の課題である。

7. おわりに

規範フローと実際の環境からのオブティカルフローの比較により静止障害物を検出する手法の改良を行い、運転者に対する危険度提示性能の向上を実現した。実環境において評価実験を行い、進路上および進路近傍に存在する障害物を高精度で検出し適切に危険度を提示することが可能であることを確認した。

今後は更に検出精度を高め、運転者による危険度の評

価を集めるとともに、危険度や処理領域、障害物検出に関わる閾値などの検討を進め、様々な状況に対応できるよう改良を進めていく予定である。

謝辞 本研究の一部は株式会社デンソーアイティージャボラトリの援助による。また、日本学術振興会基盤研究 (C)「居住空間内物体・人物統合認識に基づく親和的情報空間の高度化」の援助を受けた。ここに深謝します。

参考文献

- [1]山口晃一郎,加藤武男,二宮芳樹,“車載単眼カメラによる車両前方の障害物検出”, 情報処理学会研究報告, CVIM-151, pp.69-76, 2005.
- [2]山中隆,中野広樹,下脇克友,渡邊睦,緒方淳,“規範フロー場の差異解析による道路状況認識の研究”, 情報処理学会研究報告, CVIM-154, pp.83-90, 2006.
- [3] 山中隆,古賀由紀夫,外園山,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, CVIM-158, pp.41-48, 2007.
- [4]警察庁交通局,“平成 18 年中の交通事故の発生状況”, 2007
- [5]荒蒔勲,佐生徳実,梅田和昇,“小型距離画像センサを用いた移動ロボットの障害物回避手法の提案”, 日本機械学会ロボティクス・メカトロニクス講演会'01 講演論文集,1P1-N4, 2001.6.
- [6]岡田慧,加賀美聡,稲葉雅幸,井上博允,“色領域分割と両眼ステレオの統合による脚型ロボットの三次元障害物回避” 第 16 回ロボット学会学術講演会予稿集, pp 1505-1506, 1998.
- [7] Mutsumi Watanabe, Nobuyuki Takeda, Kazunori Onoguchi, “Moving obstacle detection and recognition by optical flow pattern analysis for mobile robots”, Advanced Robotics, Vol.12, No.7,8, pp.791-816, 1999.
- [8]Jianbo Shi, Carlo Tomasi: "Good Features to Track", 1994 IEEE Conference on Computer Vision and Pattern Recognition(CVPR'94), pp.593-600, 1994.
- [9]Jean-Yves Bouguet."Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm", OpenCV Documentation, Microprocessor Research Labs, Intel Corporation, 1999.