

規範フローと危険度を用いた障害物検出の研究

山中 隆 古賀 由紀夫 外園 宙 山田 純一 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 我々はオプティカルフロー場を解析することによる外界状況認識の研究を行っている[6]。本稿では、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、移動体に搭載したテレビ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システムにおいて、フロー比較用ブロックの適応的設定の改良、及び新たに障害物領域の抽出結果に基づく「危険性」の提示を試みた結果について報告する。

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 obstacle detection by using a model flow and risk

Takashi YAMANAKA, Yukio KOGA, Hiroshi HOKAZONO, Junichi YAMADA,
and Mutsumi WATANABE

Graduate School of Science and Engineering, Kagoshima University

1-21-40 Korimoto Kagoshima, 890-0065, Japan

Abstract. We have been doing research on outdoor scene situation recognition by analyzing optical flow field. For this purpose we already reports a technique to detect stand still obstacles using the “model flow” which corresponds 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 [6]. This paper newly reports improvements of adaptive block setting and calculation of “risk degree” based on obstacle region detection results.

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。

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。既に静止障

害物を対象とし、動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いる。

2. 従来技術と課題

障害物検出の方法として、距離情報を用いる手法と動きの情報を用いる手法が提案されている。

距離情報を用いるものには、距離センサを利用する方法[1]や、ステレオ視を利用する方法[2]がある。動きの情報を用いるものにはFOEを利用する方法[3]が従来研究されている。

距離センサでは、センサから物体までの距離や位置な

どが検知可能である。物体までの距離を高い精度で得ることができるため、低速での衝突を警告するなどの実用化もなされている。一方、人間が行っているように、障害物が車両か人間かに応じて回避行動を切り換える機能を持たせるためには、検出した障害物の種別を認識することが不可欠であり、距離の情報のみよりも、映像としての情報を用いる方法が適切であると考えられる。

映像から距離情報を求める代表的な方法に両眼ステレオ視があり、風景中の物体認識[2]などに用いられている。両眼ステレオ視の処理では2台のカメラの視差を利用するため、距離精度を得るためにはカメラ同士に距離が必要である。しかし処理領域として利用できる部分は共通視野となるため、近傍を処理するためには2台のカメラを接近させる必要がある、この場合距離精度が低下するトレードオフが生じる。よって1台のカメラで動画画像解析の手法に基づく方式を採用することとした。

3. 本研究の手法

3.1. 規範フロー

映像としての情報から動きの情報を用いる方法に、オブティカルフローとFOEを利用するもの[3]がある。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に使われている。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、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[6]。

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ブ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実際の画像のオブ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。

オブ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性を有する。

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る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、

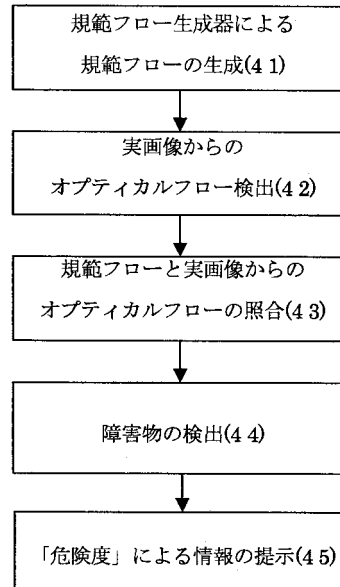


図1 本手法の処理の流れ

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。

このため、検出した障害物領域の大きさ、速度に応じた「危険度」を計算し運転者に提示することにした。

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章番号である。

まず自動車の速度やハンドルの切り角などの情報を受け取り、仮想空間上の路面上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ブティカルフローを算出する。この生成プログラムから算出されるオブティカルフローを規範フローとする。

次にKLT法により実画像のフローを抽出。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。実画像のフローの長さや向きを調べ、そのフローの入っているブロックの規範との差異がないかを照合する。

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、ブロック内での障害物の明度情報を取得し、障害物形状を復元する。

下端エッジから障害物までの距離を算出し、フローか

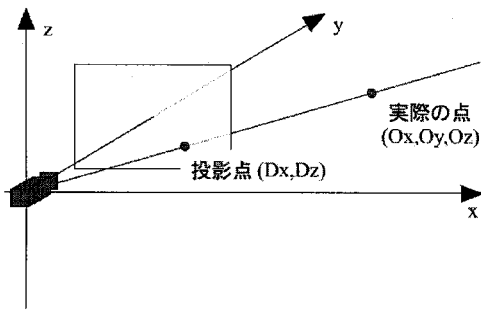


図2 投影座標系

から接近速度を算出する。接近速度と障害物領域の面積から算出された「危険度」を元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. 生成プログラムの概要

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。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、カメラ画像のどこに投影されるかを算出する。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

$$y = f$$

となる平面である。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、この平面と、特徴点の座標と原点を結ぶ直線の交点にな

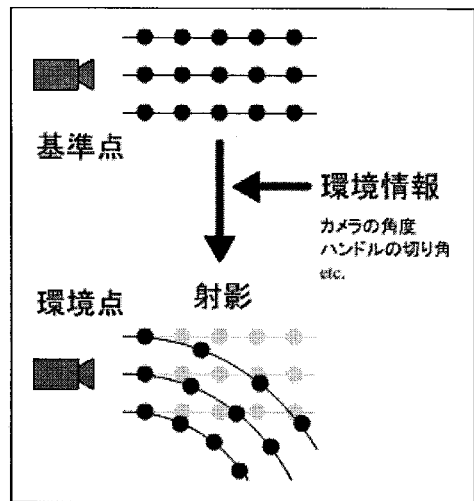


図3 特徴点配置の模式図

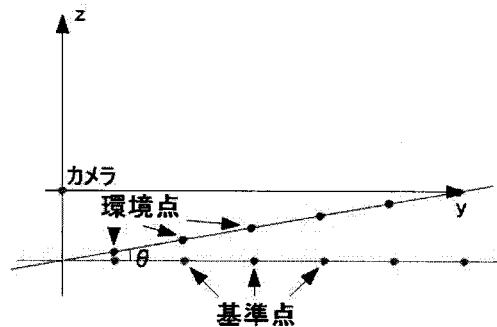


図4 カメラの傾きθの時の環境点の配置

る。

特徴点の座標を (Ox, Oy, Oz) 、環境点の座標を (Dx, Dz) とすると、 Dx 、 Dz は次の式で求めることができる。

$$Dx = \frac{Ox}{Oy} * f$$

$$Dz = \frac{Oz}{Oy} * f$$

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ことができる。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

特徴点の配置はカメラの設置状態、車両の走行状態に応じ配置を変える。特徴点はすべての状態の基準となる

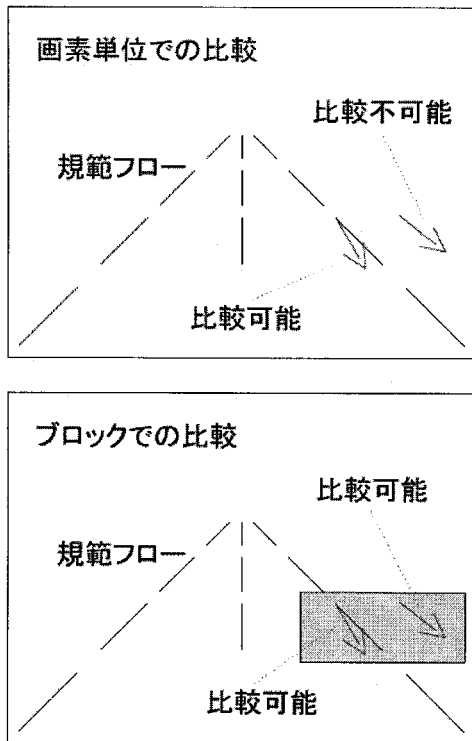


図5 ブロック比較の模式図

「基準点」を基に、環境情報によって適応的に「環境点」に射影され、環境点を投影したものが画面上の特徴点となる。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 Good Features to Track[4]に基づき

特徴点抽出を行う。

そして2枚目の画像との対応点を見つけてオプティカルフローを検出する。この対応点の決定方法を、本システムではオープンライブラリを利用して、比較の実装の容易であったKLT法[5]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。

規範フローは生成プログラム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まれる。

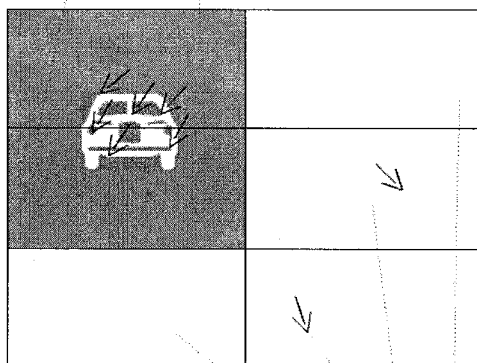
そこで規範フローを基にフロー照合用のブロックを設定し、規範フローのパラメータをブロックの範囲に拡張する。この模式図を図5に示す。この手法により、場所と規範フローの関連付けと、処理の効率化を両立した。また、1対1のベクトル比較ではなく、領域での基準との比較により、集中して検出されたベクトルもすべて比較対象を持つことが出来るようになる。

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出てくる。そこでフロー照合用のブロックは規範フローの変化とともに動的に設定される。フローの位置、大きさに合わせてブロックを設定することにより一対一の対応ができる。

規範フローは障害物のない路面を仮定して生成されたオプティカルフローであり、FOEより上部には生成されない。規範フローの比較に用いるブロックであるから、規範フローの生成される部分にブロックを配置するために、規範フローの位置情報を用いて、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。フローベクトルの中点をブロックの中心として配置し、規範フローの生成幅の設定により、路肩の標識や建物な

ブロック内に多く検出される



検出されない
または少ない

図6 ブロック内フロー検出数の模式図

どの進行の妨げにならない領域にもブロックを生成しない。

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、領域での比較というブロックの有効性が低下する。領域を確保するためにブロックの水平成分は定数で設定する。

4.4. 障害物の検出

4.4.1. 障害物フローの判断

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入っているもの全てについて、そのフローの存在するブロックの規範フロー情報と比較する。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、検出されたフローの画像中の z 軸成分が規範フローの同成分を超える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。これに対してノイズの場合は、ある領域に集中して出るわけではないからブロック内には障害物のフローだと判断されたノイズが少数含まれる。この模式図を図6に示す。

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。

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、それぞれのブロックに含まれる障害物は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。このことによりブロックの重複による情報取得処理の重複も防ぐことができる。

4.4.4. 障害物の明度情報取得

ブロックで検出した障害物の領域は障害物の一部であり、必ずしも障害物全体を示す領域とは一致しない。そこで障害物を示す情報の一つとして障害物の明度情報を取得し、同一明度を持つ領域を障害物領域として補完する。

統合されたブロックの領域は、そのすべてのブロックに渡り障害物と判定されるフローが検出されることから、障害物の領域が多くを占める。よって統合されたブロックの領域内の明度ヒストグラムを作成し、閾値以上の明度値を障害物の明度値とする。この閾値はブロックの水平成分の定数より設定した。

4.4.5. 下端エッジの検出

入力画像にエッジ処理をかけた画像に対して、障害物として検出された特徴点の座標から下方へエッジを探索する。エッジを検出すると、そこからさらに左右を探索し、横方向のつながりを補完する。

検出されたエッジの幅を障害物の幅とする。障害物の明度情報と一致したピクセルでも、エッジの両端よりも外側で検出された場合は障害物領域としない。この範囲の内側のみで障害物領域を検出することで、無駄な範囲の検索処理を省くことができ、障害物領域の誤検出も防ぐことができる。

4.5. 危険度による情報の提示

危険度に使う基準として検出した障害物の持つ相対的な運動エネルギーを用いることとした。物体の質量を m 、物体の速さを v とすると、運動エネルギー E は次の

式で表すことができる。

$$E = \frac{1}{2}mv^2$$

4.5.1. 障害物の速度情報取得

障害物の速度を V 、オプティカルフローを v 、障害物までの距離を y とすると次の関係式が成り立つ。

$$V = v \cdot y$$

障害物までの距離 y は、カメラの高さ h と障害物に対する角度 θ を使い、次の式から算出できる。

$$y = \frac{h}{\tan \theta}$$

4.5.2. 障害物の大きさ情報の取得

物体の質量をカメラからの入力で計測することは不可能であるから、障害物領域の面積で代用する。

4.5.3. 危険度の計算

オプティカルフローの値は障害物領域に含まれるフローの平均値を利用し、計算に利用するフローの値はカメラの進行方向成分とする。障害物領域のピクセル数を S 、障害物の速度における進行方向成分を V_y とすると、危険度 R は次の式で定義する。

$$R = S \cdot V_y^2$$

4.5.4. 警告の表示

障害物を検出すると「やや危険」「非常に危険」など段階的に表示を変える。表示の切り替えは計算によって得られた危険度に関する閾値を設定し、閾値を超えると「非常に危険」とする。閾値は画面上の障害物領域の占める割合から実験的に設定した。

5. 実験結果

5.1. 実験条件

前章で述べた方式に基づき移動視点からの静止障害物検出システムを汎用パソコン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。

実験に使用したパソコンの 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

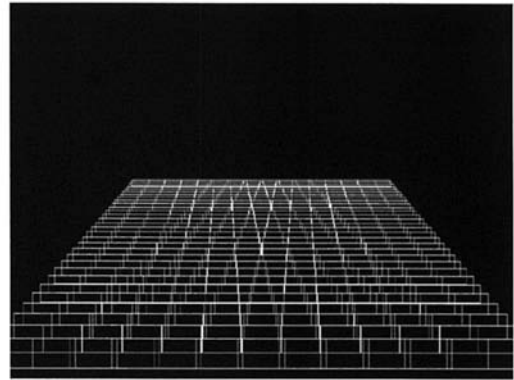


図7 時速 10km のブロック設定例

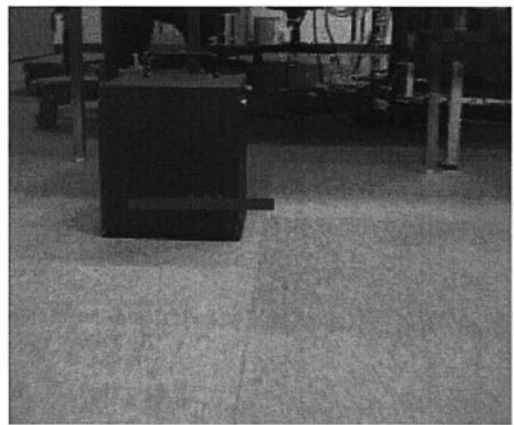


図8 ブロックでの障害物フロー検出例

ャーボードを用い、処理画像の解像度は VGA (640×480) で行い、カメラは SONY 社製の DCR-TRV30 を使用した。

5.2. ブロックの設定実験

規範フロー生成プログラムに、初期パラメータとして、カメラの高さを 120cm、特徴点の配置幅を 2m、特徴点の数を 360、特徴点の間隔を 20cm、特徴点を配置する列の数を 12 列、1/3 インチ CCD カメラの焦点距離を 8mm と設定し、実験を行った。

速度を時速 10km に設定し、実行した場合の実行画面を図 7 に示す。路面に配置した特徴点からなる規範フローの生成位置にブロックが配置されていることがわかる。速度、ハンドルの切り角などの変数を変化させるとブロックの配置、大きさもその情報に応じて変化することが確認できた。

5.3. 障害物検出とブロックの統合実験

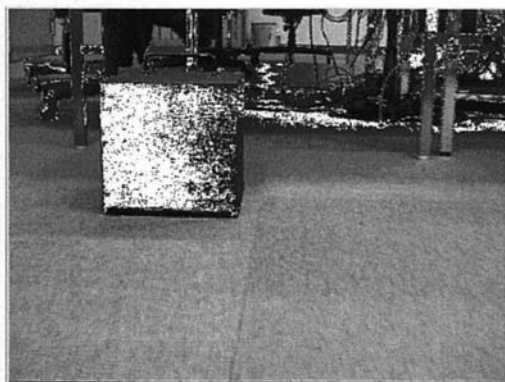


図 9 明度情報による障害物領域検出例

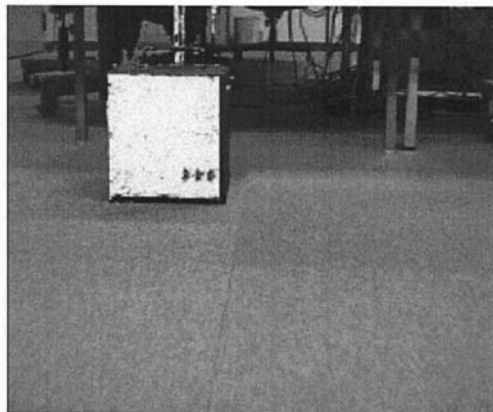


図 11 危険度の低い状態の障害物例

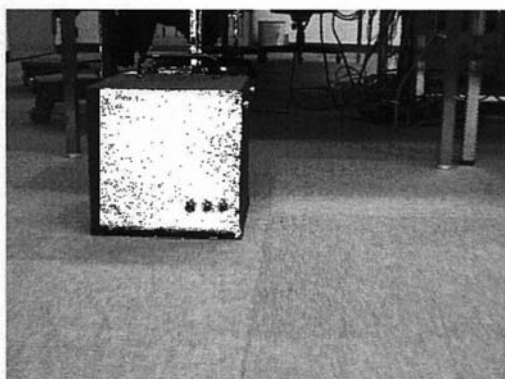


図 10 下端エッジを利用した障害物領域の表示例

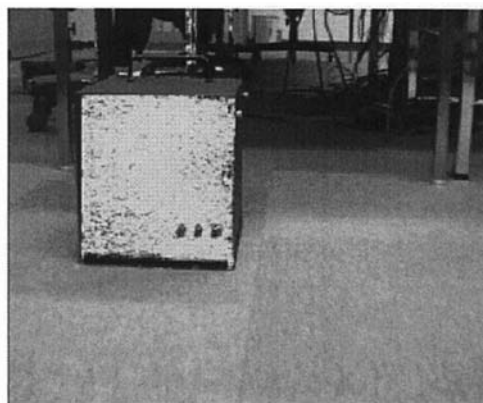


図 12 警告状態の障害物例

室内に構築した模擬環境用にパラメータの値を、カメラの高さ 33cm、特徴点の配置幅を 50cm、特徴点の数を 360、特徴点の間隔を 10cm、特徴点を配置する列の数を 12 列、カメラの傾きを水平から下向きに 9 度に設定した。模擬環境上に障害物に見立てた箱を設置し、カメラを移動させたときの障害物検出実験を行った。

ひとつのブロックでのみ障害物が検出された場合ブロックの枠を表示し、いくつかのブロックが統合された場合はブロックを塗りつぶして表示する。

複数のブロックが統合されて検出されている画像を図 8 に示す。箱の右下で複数のブロックが統合されて検出されていることがわかる。

またブロックを設定した範囲の中でも床面に検出されたフローを障害物と認識していないことを確認した。画面奥に検出されている机などのフローはブロックの処理範囲外にあるため検出されていない。

5.4. 障害物の明度情報取得実験

図 8 に示されたように統合して検出したブロック部分をマスクとして範囲内の明度情報ヒストグラムを作成する。ヒストグラムの値が設定した閾値以上の明度値を障害物の明度として取得し、画面中の同一明度部分検出した画像を図 9 に示す。閾値はブロックの幅の値とした。

一部対象物の明度変化により検出されない部分があるが、ブロックの設定範囲外となりブロック領域では検出不可能な障害物領域まで検出されており、図 8 に比べて対象物の形を復元できていることがわかる。

しかし同時に背景部分の同一明度部分を障害物領域と誤検出してしまっていることもわかる。

5.5. 下端エッジ検出による障害物領域の限定実験

障害物領域を明度情報で補完する際、下端エッジの情報から、その両端よりも内側の範囲に処理領域を限定し

た場合の障害物領域の表示をしたものを図 10 に示す。障害物のほぼ全体を障害物領域として塗りつぶしており、かつ図 9 で誤検出した物の影などの部分は検出していないことがわかる。

図 10 の画像に対して実際の障害物の領域中で、システムが障害物として検出した領域のピクセル数の割合は約 80%となった。

5.6. 危険度による情報提示実験

危険度による警告通知の閾値を 800 と設定して表示の変化を観察した。警告を通知する前の検出された障害物の画像を図 11 に示す。このとき障害物は検出されているが警告は出ていないことが確認できた。

図 11 からさらに接近し、危険度が閾値を超えた場合の画像を図 12 に示す。障害物領域は警告を意味する色へ変化していることが確認できた。

6. 考察

従来手法[6]に比べてブロックの適応的な設定により、処理領域を制限してフローの比較処理を少なくしたが、検出精度に著しい低下は見られなかった。また、誤検出の割合が低下しているが、ブロックの統合や下端エッジを適応することによりノイズを吸収したことが要因だと考えられる。これらのことからブロックの適応的な設定、統合の処理が性能の向上につながっていると言える。

ブロックの適応的な設定により画面全体への処理ではなくなったが、障害物の部分的な検出の場合も明度情報と下端エッジによる補完により障害物領域を補完可能で、ブロックのみによる領域検出よりも検出精度は向上したといえる。

障害物領域の面積と距離情報、フローの大きさから、危険度を算出し、必要に応じた警告をする方式を確立することができた。距離による危険度の変化は未実装だが、接近による画面中の障害物領域の増加により、警告が可能であることが確認できた。

7. おわりに

規範フローと実際の環境からのオプ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について、照合するためのブロックの設定方法の変更や、障害物領域の検出機能の追加により性能が向上することを示した。

今後は実環境での実験などを通して、危険度に対する適当な閾値の設定、危険度を導出する関数や変数の検討、システム全体の効率的な処理を取り入れることにより更なる改良を進める予定である。

謝辞 本研究の一部は株式会社デンソーアイティラボラトリーの援助による。ここに深謝します。

参考文献

- [1]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.
- [2]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.
- [3] 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.
- [4]Jianbo Shi, Carlo Tomasi: "Good Features to Track", 1994 IEEE Conference on Computer Vision and Pattern Recognition(CVPR'94), pp.593-600, 1994.
- [5]Jean-Yves Bouguet:"Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm", *OpenCV Documentation*, Microprocessor Research Labs, Intel Corporation, 1999.
- [6]山中隆,中野広樹,下脇克友,渡邊睦,緒方淳, "規範フロー一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.