H-076

カメラ周囲の危険を認識するシステムの開発

Detecting dangerous situations using the video processing technology

山口 諒† 鈴木 啓介† 野崎 祐基† 石井 純一† 柳下 翔太† 山名 徹† 金丸 隆志†
Ryo Yamaguchi Keisuke Suzuki Yuki Nozaki Junichi Ishii Shota Yagishita
Toru Yamana Takashi Kanamaru

1. 目的

近年のコンセプトカーにはすでにサイドミラーの代わりにカメラを使用しているものが圧倒的に多い．そこで，カメラによる危険認識を行うシステムの開発に着目した．現在の危険認識システムはカメラのみでなくセンサーなどを併用して使用しているため高価となる．そこで，より多くの車に危険認識システムを普及させることを念頭に置き，センサーやステレオカメラを使用しないシステムを目指す．

2. 目標と制約条件

下記の目標と制約条件を設定した．

- 車の 1 方向に 1 つのカメラのみ用いる．
- 運転手が画面を見てなくても気づけるよう音声で危険を通知する．
- 実際に車に搭載して運用できるよう，カメラから通知まで，一貫して組み込みシステムを用いる．
- 停止状態と時速 10km 以下で検知できるようにする．
- シンプルにするため当面暗闇での検知は考えないとする．

3. 静止時の処理の流れ

まず，車が静止している時に周囲での動体物を検出するシステムを考えた．このときの処理の流れは下記の通りである．

(1) 元画像にある小さなノイズを除去するためガウシアンフィルタを適用する．

(2) 画像内から物体の動きを検出するため，フレーム差分処理を用いる．このとき変化したと判定された領域を白い画素で，そうでない領域を黒い画素で表す．

(3) (2)の処理で得た画像の白い領域の輪郭をたどり，輪郭の画素数および車に最も近い座標を得る．

(4) (3)で得られた画素数の値が，設定した値 30 ピクセル以上を満している場合は「大きい物体」と考えて(5)の処理に移り，満たしていない場合は輪郭を切り捨てる．

(5) 画素数が 30 ピクセル以上の輪郭に対してのみ，輪郭の画素を赤い画素に変え，移動領域に赤い輪郭を表示させる．

(6) 輪郭上での最接近座標と過去の最接近座標から移動ベクトルを算出する．

(7) 車体からの遠い順に青，黄，赤の線を引く．さらに level0～level3 の 4 通りの危険度を設ける．再接近座標が黄線より遠い場合は level0，黄線と赤線の間にある場合を level1，再接近座標が赤線内でなおかつ移動ベクトルが自車方向を向いていれば level3，向いていなければ level2 とした．

(8) (7)で判定された危険度をシリアル通信により警告信号発生回路に送り，level ごとの警告音を発する．

4. 使用した装置と実験器具

ここまでの処理を下記の装置に対して実装し，動作検証を行った．

- 画像処理回路：東芝製 Visconti 評価ボード
- カメラ：魚眼カメラ
- 警告信号発生回路：PIC16F88 を用いた自作回路

図1 実験風景

図 1 は実験装置を自動車に取り付けて実験を行っている様子である．なお，警告信号発生回路の回路図は図 2 の通りである．

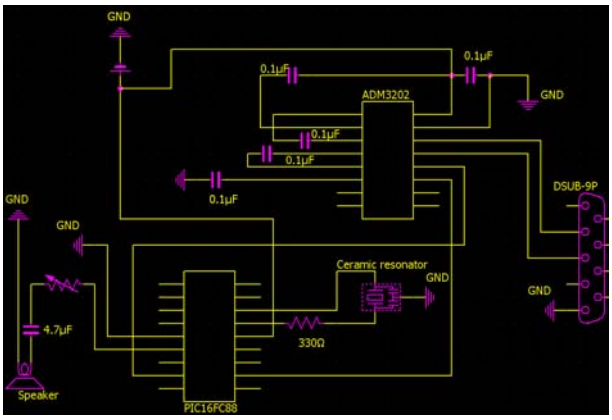

図２警告信号発生回路

前章で述べたように，Visconti からの出力信号は危険度に応じた数字「0」「1」「2」「3」の 4 種類である．この信号をシリアル通信により PIC16F88 に送り，その値に応じて，警告音をスピーカーに対して出力する．それぞれの信号をオシロスコープで計測したものが図 3 である．level0 では無音であるが，level1 以降は「プー，プー，…」という音が鳴り，危険度が高くなるに応じて音の間隔は短くなる．なお，音の周波数は約 440Hz を用いた．

† 工学院大学 Kogakuin University

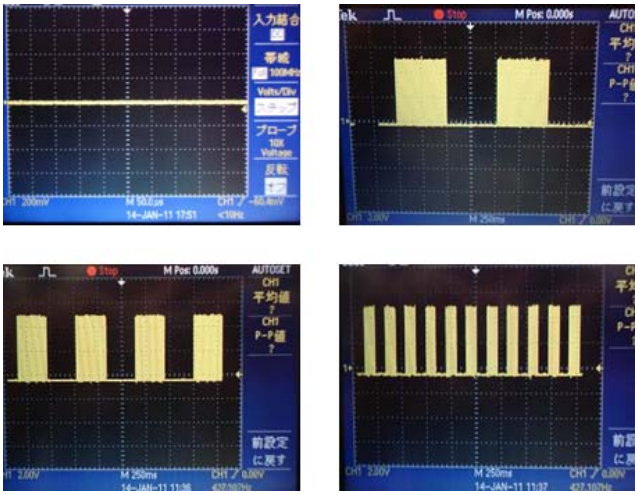

図 3　危険度に応じた音声信号の様子

5. 結果

ここまでのシステムを用い，静止した車の周囲の移動物を検知できるか実験を行った．図 4(左)のフロントカメラの映像に関しては服が地面の輝度値と近く輪郭が少なかったが靴と服の輝度値の差よりシステムがしっかり反応しており，音による警告も成功した．図 4(右)はリアカメラの映像の様子であるが，移動ベクトルと輪郭が表示されていることがわかる．

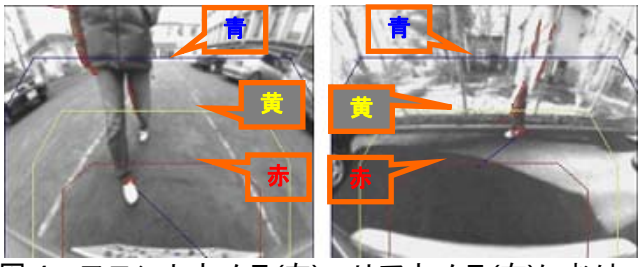

図 4　フロントカメラ(左)，リアカメラ(右)における移動物検出の様子

さらに，図 5 は右方向のカメラにおいて，複数の移動物が同時にカメラの視野に入った場合であるが，しっかりとベクトルと輪郭が表示されていることがわかる．

6. 車が動いたときの対応

上記のように，車が静止しているときに動作するシステムは組み込みシステムにより実現できているが，車が動いている場合に動作するシステムは現在 Windows 上で動作するプロトタイプを作成している．

いま，時刻 N，位置 (i, j) における画素値を $p_N(i, j)$とおく．例えば図 5 のようにサイドカメラに対する動きを検出することを考えよう．変化を検出するためのフレーム差分に対し，以下のように車の移動分を相殺するようなずらし量 $s(i)$ を導入する．

（時刻 $N+1$ の変化量）$= p_{N+1}(i, j-s(i)) - p_N(i, j)$

$s(i)$に i 依存性があるのは，カメラからの距離に対してずらし量を変化させる必要があるためである．$s(i)$の決定には，横座標 i に対する任意の局所領域を設定し，その領域で j をずらしながらのテンプレートマッチングを行い，

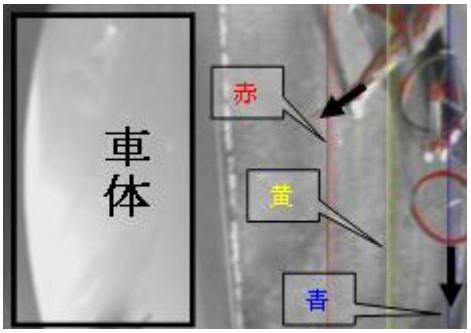

図 5 複数移動物の検知成功例

変化量が最小となるような移動量を $s(i)$として決定している．

Windows 上で作成したプロトタイプの動作を示したのが図 6 である．通常のフレーム差分では移動する自転車だけではなく，地面の石や風景なども移動物として検知されてしまうが，ずらし量 $s(i)$を導入すると，石や風景の変化の検出はかなりの程度抑えられていることがわかった．

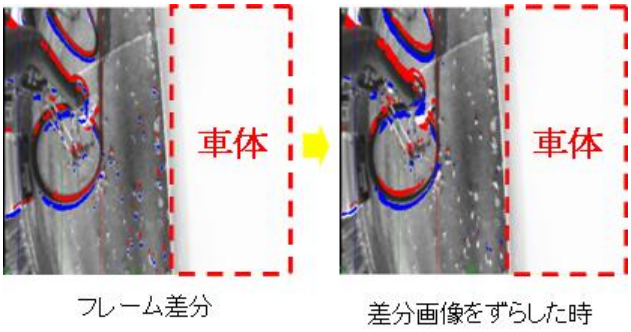

図 6　車が移動している際のフレーム差分(左)と，ずらし量 $s(i)$を導入したフレーム差分(右)

ただし，全ての風景の移動の検出が抑えられているわけではない．これには様々な理由が考えられるが，一つには，このプロトタイプは Windows 上で動作しており，1 秒間に 30 フレームの画像を全て処理できているわけではないこと，それゆえ，一フレームあたりの物体の移動量が大きくなってしまうことが理由として考えられる．

7. 今後の課題

前章で紹介した Windows 上でのプロトタイプを Visconti に移植し，風景の誤検出を抑えるのが今後の課題である．

参考文献

[1]昌達 慶二，“[詳解] 画像処理プログラミング," ソフトバンククリエイティブ株式会社 (2008).
[2]後閑 哲也，“改訂版 電子工作のため PIC16F 活用ガイドブック," 技術評論社, (2006).
[3]Jesse Liberty，“第 4 版 プログラミング C#," O’REILLY, (2007).