Vol.2017-MBL-85 No.2
Vol.2017-ITS-71 No.2
2017/11/15

車載ステレオカメラを用いた道路脇壁検出方法に関する研究

情報処理学会　第85回モバイルコンピューティングとパーベイシブシステム
第71回高度交通システムとスマートコミュニティ合同研究発表会
神奈川工科大学　情報学部　情報工学科
樫村　亮多，宮崎　剛

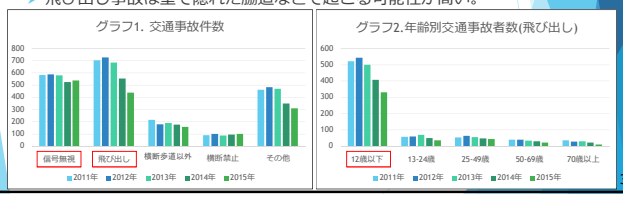
背景

- 近年、多くの企業が自動運転技術の研究を行っている。
- 自動運転技術が発展している。
- 様々な機器や、機器の組み合わせによって多くの自動運転技術が存在する。
  - ステレオカメラ,全方位カメラ,単眼カメラとミリ波レーダー

2

背景

- 近年、交通事故件数は減少してきているものの（グラフ１）、その中で信号無視や飛び出しが高い割合を示している[1]
- 年齢ベルに見た飛び出しによる交通事故では12歳以下が高い割合を占めている[1]（グラフ2）
- 飛び出し事故は壁で隠れた脇道などで起こる可能性が高い。



3

目的

- 壁等により見えにくくなっている脇道の検出。
- 地図情報から脇道の有無を検出する。
- 車載ステレオカメラで脇道手前の壁を認識し、飛び出し事故の減少を図る。
  - 距離データと画像データを利用する。

4

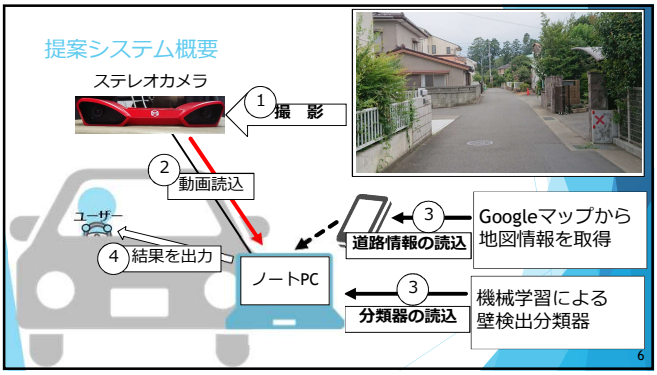
関連研究

『車載ステレオカメラと動的背景差分による歩行者検出の研究』
笠原正樹ら，情報処理学会 第75回全国大会講演論文集,第2分冊, 3U-6,pp.573 - 574,2013.[2]

- ステレオカメラを用いた歩行者検出。
- 背景の物体が当然動くため歩行者でないものが誤検出されるため正確性に問題がある。
- 予測生成した背景と実際の背景の差分に動的背景差分処理を行う事で問題の解決を図る。
- 距離画像を使い、背景差分の精度を向上。
- さらなる精度向上、また上下振動の対策が課題。

5

提案システム概要



6

Vol.2017-MBL-85 No.2
Vol.2017-ITS-71 No.2
2017/11/15

Robovision2

- ZMP社製
- 超高感度CMOSイメージセンサIMX224，2個搭載。
- 距離を色別にて出力可能(図1)
- 開発環境は表1に示す。

【SDK】

- ライブラリ：カメラI/F, 視差画像生成
- アプリケーション：RAW出力，歪補正結果出力
- 視差画像出力

表1. 開発可能環境

OS	WindowsまたはLinux
開発ソフト	Visual Studio 2013
必要ライブラリ	DirectShow
	OpenCV (2.4.10)
	CUDA (Ver.7)

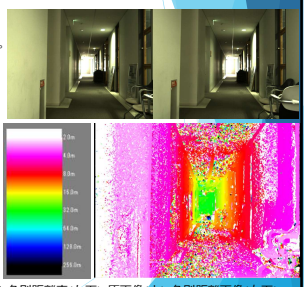
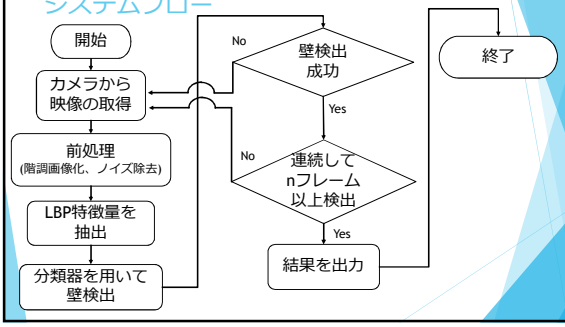


図1.色別距離表(左下)、原画像(上)、色別距離画像(右下) 7

システムフロー



8

壁検出方法

LBP特徴を用いて分類器の作成。

- LBP特徴(Local Binary Pattern)
- 【概要】
  - 3x3画素の中心画素と周辺画素の輝度を比較。
  - 中心画素より明るければ１、暗ければ０とし、１かたまりのバイナリデータにする。
  - バイナリデータの集まりが、1つのヒストグラムとする。
  - ヒストグラムの組み合わせを比較する事により物体を判定。
- 【利点】
  - 照明の変化の影響を受けにくい
  - 高速な特徴計算が可能

9

実験内容

- 実験1.分類器の作成
  - LBP特徴量による分類器の作成
  - 評価方法
    - 作成した分類器で6枚の壁画像から壁の検出率を計測。
    - ネガティブ画像を変更して分類器を作成した場合の検出率の変化を計測。
- 実験2.物体検出のパラメーター(detectMultiScale)
  - OpenCVのdetectMultiScare()を使って壁を検出。
  - 評価方法
    - パラメータ(scaleFactor, minNeighbors)の値を変化させることによる検出率の変化を計測。

今回は地図情報から脇道を検出する実験は対象外。

10

実験環境

CPU	Intel Core i7－7500
GPU	GeForce 1060 6GB
メモリ	8GB
OS	Windows10 Home 64bit
開発言語	C++
開発環境	Microsoft Visual Studio 2013
画像処理ライブラリ	OpenCV 3.1
ステレオカメラ	Robovision2

11

実験1.分類器作成

図2. 正面

図3. 正面（引き）

図4. 斜め

- 図2,図3,図4に示すような3種類の画像427枚を撮影。
  - 正面画像148枚，正面（引き）画像118枚，斜め画像161枚
- 撮影した画像を基に画像を増やし合計1000枚とする。
- 以下の条件で作成した分類器を分類器1とする。
  - 正解画像1000枚，不正解画像430枚。
  - 正解画像のサイズ:400x225
  - 不正解画像に壁のない道路の画像を使用（図5）

図5. 不正解画像

12

実験1.分類器1による壁検出結果

- 分類器１で検出を行った結果を図6に示す。
- 検出数は多いが、壁以外の領域も多く検出している。

図6. 分類器1の検出結果　写真１(左),写真２(真ん中),写真３(右)

13

実験1. ネガティブ画像を追加して再学習

- 以下の作成した分類器を分類器2とする。
  - ネガティブ画像以外の環境は分類器１と同様。
  - 不正解画像430枚に電柱、垣根の画像を追加。
  - 追加した画像の例を図7､図8として示す。

図7. 垣根

図8. 電柱

14

実験1.再学習の分類器2による壁検出結果

- 実行結果を図9として示す。
- 壁以外の検出結果が大きく減少。
- ネガティブ画像の種類により検出数が大きく変化した事がわかる。

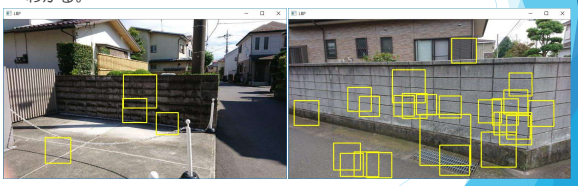


図9. 分類器2の検出結果　写真１(左),写真4(右)

15

実験1.再学習の分類器による壁検出結果

- 分類器2をネガティブ画像300枚にし、作成した分類器を分類器3とする。
- 分類器2をネガティブ画像600枚にし、作成した分類器を分類器4とする。
- 分類器3の検出結果を図10、分類器4の検出結果を図11とする。

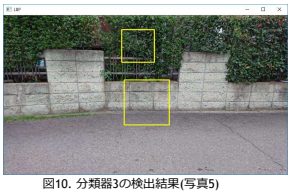


図10. 分類器3の検出結果(写真5)

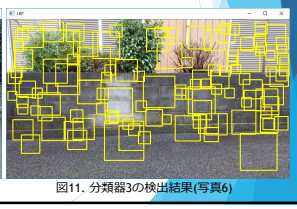


図11. 分類器3の検出結果(写真6)

16

実験1.再学習の分類器による学習結果

- 分類器1から4で写真1を壁検出した結果の正解率を表2に示す。
  - 正解検出は検出時の四角の8割以上を壁が占めるものとする。
  - 誤検出は検出時の四角に壁の占めている割合が7割以下とする。
  - 正解検出をPとし誤検出をEとした時、正解率の計算方法を以下に示す。
    - 正解率$= \frac{P}{P+E}$

表2. 各分類器の正解率

分類器	写真１	写真２	写真３	写真４	写真５	写真６	平均
分類器１	１５%	２０%	７３%	７０%	４５%	３４%	４３%
分類器２	７５%	２４%	７１%	６３%	３０%	３０%	４９%
分類器３	５０%	２１%	１００%	７５%	５０%	３３%	５４%
分類器４	４５%	３４%	７６%	５８%	３８%	５３%	５０%

17

実験2.物体検出パラメータ(detectMultiScale)

- 正解率の一番高かった分類器2を使用。
- detectMultiScaleで用いられるパラメータを調整。
  - 大きさの比率の異なる対象物を検出するパラメータ,画像スケール(scaleFactor)の値を変更
    - デフォルト値が1.1なので、デフォルトより低い1.01と1.01、デフォルトより高い1.15と1.2と1.3で実験。
  - 物体候補最小数を指定するパラメータ(minNeighbors)の値を変更
    - デフォルト値が3なので、デフォルトより低い1と2、デフォルトより高い4と5と9で実験。
- それぞれの値を変えた時の正解率についてまとめる

18

実験2.物体検出パラメータ(detectMultiScale)

- 画像スケール(scaleFactor)の変更
- 標準値よりも低い1.01で検出した結果を図12、標準の1.1での結果を図13、標準よりも高い1.3での結果を図14として示す。
  - 数値が低い1.01では検出数は少なくなり、誤検知は微かに増える。
  - 数値が高い1.3では検出数は多くなり、誤検知は多くなる。

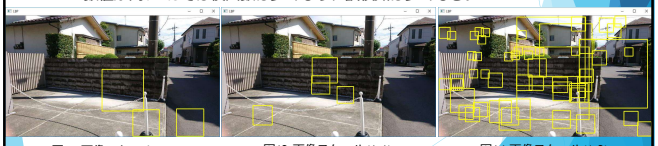


図12.画像スケール(1.01)　図13.画像スケール(1.1)　図14.画像スケール(1.3)

19

実験2.物体検出パラメータ(detectMultiScale)

- 画像スケール(scaleFactor)の成功率を以下の表3にまとめる。

表3. 各scaleFactor値の正解率

Scale Factor値	写真１	写真２	写真３	写真４	写真５	写真６	平均
1.01	３３%	３０%	１００%	３３%	０%	１００%	４９%
1.05	３３%	３６%	１００%	４２%	５０%	２４%	４８%
1.1	７５%	２４%	７１%	６３%	３０%	３０%	４９%
1.15	１６%	１７%	８０%	８１%	８%	１６%	３６%
1.2	２５%	３１%	４２%	７１%	３９%	２０%	３８%
1.3	２７%	２６%	６７%	６１%	４０%	４９%	４５%

20

実験2.物体検出パラメータ(detectMultiScale)

- 物体候補最小数を指定するパラメータ(minNeighbors)の変更
- 標準の数値よりも低い１で検出した結果を図15、標準の３の結果を図16、標準よりも高い9の結果を図17として示す。
  - 数値が低い1では検出数は多くなるが、誤検知が多くなる。
  - 数値が高い9では検出数は少なくなるが、誤検知は少なくなる。

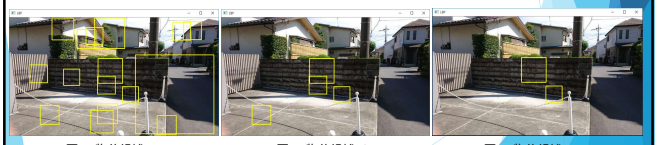


図15.物体候補(１)　図16.物体候補(３)　図17.物体候補(9)

21

実験2.物体検出パラメータ(detectMultiScale)

- 画像スケール(minNeighbors)の検出結果を以下の表4にまとめる。

表4. 各minNeighbors値の正解率

minNeighbors値	写真１	写真２	写真３	写真４	写真５	写真６	平均
1	２５%	１５%	７１%	６７%	１５%	２８%	３７%
2	５０%	２０%	７２%	７３%	１５%	２０%	４２%
3	７５%	２４%	７１%	６３%	３０%	３０%	４９%
4	１００%	３３%	７１%	７２%	１３%	１９%	５２%
5	１００%	３５%	８３%	７１%	１７%	２３%	５５%
9	１００%	３６%	１００%	７１%	３３%	３３%	６２%

22

考察

- 実験１の結果から、現時点では画像ごとに正解率の差が大きく不安があるため、ポジティブ画像とネガティブ画像を増やす必要がある。
- 実験２の結果から、detectMultiScaleの２つの数値を片方変えるだけでは決して正解率が上がるわけではない事が分かったため、壁検出に適した組み合わせを見つけることが必要であると考えられる。
- これらを行えば検出率が上がるのではないかと思われる。

23

まとめ

- 飛び出しの危険性のある脇道の検出を行うため、壁認識を用いた脇道検出方法を提案した。
- 壁認識の正解率を高めるため、ポジティブ画像とネガティブ画像を調整することで成功率を高める事ができるかを調べた。
- 検出時のパラメータを調整することで正解率が高める事ができるかを調べた。

24

Vol.2017-MBL-85 No.2
Vol.2017-ITS-71 No.2
2017/11/15

参考文献

1. 交通事故総合分析センター:統計資料
http://www.itarda.or.jp/materials/publications.php?page=4(参照 2017/10/19)
2. 笠原正樹ら:車載ステレオカメラと動的背景差分による歩行者検出の研究,情報処理学会第75回全国大会講演論文集,第2分冊, 3 U-6,pp.573 – 574,2013.
3. EeePCの軌跡：OpenCVでLBP特徴を使った"物体検出器"を作成してみた,http://arkouji.cocolog-nifty.com/blog/2017/04/opencvlbp-b3cc.html (参照 2017/8/03)
4. Workpiles：物体検出（detectMultiScale）をパラメータを変えて試してみる,http://workpiles.com/2015/04/opencv-detectmultiscale-minneighbors (参照 2017/10/16)
5. DESIGNER BLOG: OpenCVで物体検出器を作成する③ ～LBP特徴～, https://www.pro-s.co.jp/engineerblog/opencv/post_6256.html (参照2017/8/01)

25