

エッジと背景を利用した車両検出

池田 徹 大中慎一 溝口正典

日本電気(株) 情報メディア研究所

〒216 川崎市宮前区宮崎 4-1-1

あらまし 道路の交通状況を把握するためにTVカメラによる監視が人手で行なわれているが、画像処理による監視機能の自動化への要求が高まっている。本稿では、道路を俯瞰撮影した監視画像から車両を検出し、通過台数を計測するアルゴリズムについて報告する。本アルゴリズムでは、撮影環境の変動のもとで比較的安定なエッジに着目している。まず、背景画像の自動生成手法について述べ、これを利用して車両のエッジを車線毎に背景のエッジから分離して抽出し、各時刻に車線ごとの車両エッジの横方向への投影を適応的に2値化して車両を検出する手法を提案した。また、車両位置データを追跡することにより通過台数を計測する手法を考案した。最後に、実際の道路画像を利用し、車両の通過台数による性能評価実験から本手法の有効性を示した。

和文キーワード 車両、背景画像、エッジ、2値化、追跡

Vehicle Detection Using Edge And Background Image

Toru IKEDA Shin'ichi OHNAKA Masanori MIZOGUCHI

Information Technology Research Laboratories, NEC Corporation

4-1-1, Miyazaki, Miyamae-ku, Kawasaki-shi, Kanagawa 216, Japan

Abstract A method for traffic measurement using image processing technique is in need. In this paper, an algorithm is proposed, which detects vehicles from overlooked road images. In this algorithm, edge segments are used because of their stability under various condition. First, a method, which produces background image automatically, is proposed. Vehicle edges are extracted for every lane using background edges. Next, vehicles' positions are detected by horizontal projections of vehicle edges in each lane, and count them by tracking vehicles. The algorithm is tested using some road traffic images.

英文 key words Vehicle, Background Image, Edge, Binalization, Tracking

1 はじめに

交通計測及び監視の分野では、現在人手に依存しているモニタ監視の自動化を目指し、可搬性があり、多様な交通情報を取得できる視覚センサを用いた交通計測装置への要求が高まっている [1][2][4][6]。そこで本稿では、道路を俯瞰撮影した監視画像から車両を検出するアルゴリズムについて述べる。

本アルゴリズムでは、撮影環境の変動のもとで比較的安定なエッジに着目するが、車両のエッジと背景のエッジが混在しているという問題がある。そこで、背景画像の自動生成手法を用い、車両のエッジを車線毎に背景のエッジから分離して検出するための背景エッジマスクを生成する。背景エッジマスクを用いて検出された車線毎の車両エッジを横方向へ投影し、ノイズや車両エッジの欠落を考慮した適応的な手法により投影データを2値化し、縦方向の位置により車両を検出する。また、各時刻の検出結果から車両を追跡することにより通過台数を計測する。さらに、本アルゴリズムを実際の道路画像に適用し、車両の通過台数による性能評価実験を試みた。

以下、2に背景エッジマスクの生成方法、3に車両の検出手法、4に車両の追跡手法、5に本手法による実験結果について報告する。

2 背景エッジマスク

背景のエッジと車両のエッジとを識別する枠組みについて説明する。まず、背景画像を自動生成し、得られた背景画像から車線毎に背景のエッジに対応する画素を指定する背景エッジマスクを生成する。

2.1 背景画像の生成と更新

入力データとなる道路画像には、屋外のように不安定な条件のもとで撮影されるケースが非常に多い。そこで、背景生成手法には、撮影環境の変動による変化には素早く追従し、しかも移動物体の影響は受けたくないという一見相反する条件を満たす必要がある。

従来法 [3][5][7] には、背景画像に入力画像を一定の割合で足し込む方法や、フレーム間差分や背景差分により動領域と静止領域を判定し、静止領域だけで背景を更新する方法などがある。これらに以下のような傾向があることに注目する。

1. フレーム間差分により動領域判定をすると、移動物体が高速なら質の良い背景画像が得られる。

2. 画素値の更新速度は緩慢な方がよい。
3. 背景差分により動領域判定をすると、更新エラーの修復が困難である。また、初期の背景画像が必要である。
4. 背景に近い画素値を持つ移動物体領域では、移動物体の画素値に更新されやすい。

そこで、次の立場で新しい方法を考えた。

1. 動領域の判定にフレーム間差分を用いる。また、低速物体に対応するため、フレーム間差分の累積情報を用いる。
2. 画素値の更新速度を1とする。
3. 背景に近い画素値を持つ画素については更新しない。

時刻 t における入力画像を $I(t)$ 、背景画像を $B(t)$ とおく。

まず、フレーム間差分を用いて動領域の判定をする。画素値の一致を判定するための閾値 T を用意し、各時刻のフレーム間差分値を次式により2値化する。

$$m(t) = \begin{cases} 1 & |I(t) - I(t-1)| < T \text{ の場合、} \\ 0 & \text{その他の場合。} \end{cases}$$

低速物体の同一輝度領域の内部が静止領域と判断されて更新されるのを避けるため、次式のように過去 k フレーム間の2値画像の論理積で累積マスク $M(t)$ を定義する。

$$M(t) = \text{AND}_{s=t-k}^t m(s)$$

$M(t)$ の値が1の領域を静止領域、0の領域を動領域と判断する。各時刻の差分情報は各画素1ビットで記憶できるので、大量の画像データを保持する必要はない。

次に、静止領域について次式の操作により背景画像を更新する。ただし、動領域は不変とする。

$$B(t) = \begin{cases} B(t-1) + 1 & I(t) > B(t-1) + T \text{ の場合、} \\ B(t-1) - 1 & I(t) < B(t-1) - T \text{ の場合、} \\ B(t-1) & \text{その他の場合。} \end{cases}$$

すなわち、現在の背景画像と T 程度の差しかない場合には一致していると判断して更新しない。また、画素値の更新速度は1に設定されている。

2.2 背景エッジマスク

生成された背景画像を用いて背景エッジマスクを作る。これは、各車線を指定する車線マスクから、背景に含まれるエッジに対応する画素を除外したものである。車線数が2の画像を対象とした場合を想定して説明する。

まず、左線、中央線、右線の3線を3次元スプラインで近似する。左線と中央線に挟まれた領域を左車線と判断して画素値1を、右線と中央線に挟まれた領域を右車線と判断して画素値2を、それ以外の領域に画素値0を与え、3値からなる車線マスクを生成する。

次に、前節の背景画像の生成アルゴリズムを用いて背景画像を生成し、得られた背景画像からエッジ検出処理を行ない、エッジに属する画素は画素値0、その他の画素は画素値255として2値背景エッジ画像を生成する。エッジ検出およびその2値化の閾値には、後述の車両検出と同じものを用いる。

最後に、車線マスクと2値背景エッジ画像との論理積を背景エッジマスクとする。すなわち、背景エッジマスク画像は左車線、右車線の背景に属するエッジ以外の画素がそれぞれ1、2となり、それ以外の画素は全て0となっている画像である。

更に多くの車線で計測したい場合には、第3の車線には画素値4、第4の車線には画素値8となるように、各ビット毎に車線を指定すれば同様の方法で背景エッジマスクを生成できる。

一例として、図1の入力画像に上記処理を適用したところ、図2に示す車線マスク、図3に示す2値背景エッジ画像、図4に示す背景エッジマスクが得られた。

3 車両の検出

本節では、背景エッジマスクを利用し、各時刻において車線ごとに車両のエッジ投影データを生成する手法について解説する。計測したい車線数は2であると仮定する。ただし、車線数が2より大きい場合にも同様のアルゴリズムを利用できる。

3.1 アルゴリズムの概要

図5に車両検出処理のフローを示す。まず、ノイズを取り除くために入力画像に3×3メディアンフィルタをかける。その後、背景エッジマスクの生成処理に用いたエッジ検出およびその2値化と同様の手法により2値エッジ画像を生成する。このとき、エッジに対応する画素値を255、それ以外の画素値を0とする。

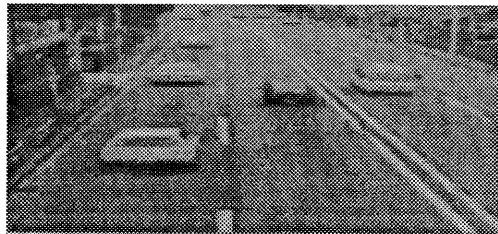


図 1: 入力画像

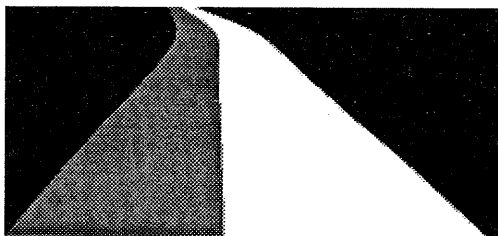


図 2: 車線マスク



図 3: 2値背景エッジ画像



図 4: 背景エッジマスク

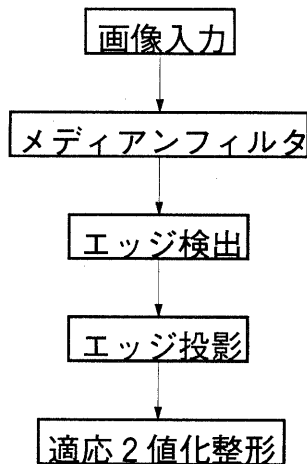


図 5: 車両検出処理フロー

次に、背景エッジマスクを用いて2値エッジ画像から車両のエッジだけを識別し、車線毎にエッジの横方向への投影を求める。さらにエッジ投影の欠落部分を補うため、整形処理を行う。

3.2 エッジの投影

2値エッジ画像を $A_{i,j}$ 、マスク画像を $M_{i,j}$ ($0 \leq i \leq X-1, 0 \leq j \leq Y-1$) とする。また、車線毎のエッジカウンタを用意し、その値を S_1, S_2 とおく。

以下の要領で第 j 行のエッジの投影を求める。

1. $i = 0$
2. $S_1 = S_2 = 0$
3. S_1 に $A_{i,j}$ と $M_{i,j}$ との論理和を加算する。また、 S_2 に $A_{i,j}$ と $M_{i,j}$ との論理和を1ビット右シフトして加算する。
4. $i = i + 1$
5. 3~4の操作を X 回繰り返す。

以上の操作で、第 j 行のエッジの投影が車線毎に求められる。この操作を各行について行なう。

3.3 エッジ投影データの整形処理

前節で得られたエッジ投影をもとに、車両の存在区間を1、それ以外を0とする2値投影データとして車両を検出する。

3 値化後の値	位置づけ
2	車両領域
1	ノイズもしくは車両領域の欠落部分
0	ノイズ

表 1: エッジ投影の3値化

エッジ投影を閾値により2値化したいが、ノイズや車両エッジの欠けのために車両領域に穴領域などの欠落を生じる。そこで、このような欠落部分を補うためにエッジ投影データの整形処理を行う。本処理は4つのステップから構成されているが、これらについて図6を用いて順次説明していく。

3.3.1 適応的2値化

ノイズを分離できる程度の閾値でエッジ投影を2値化すると、前述のように欠落部分を生ずるが、一方でノイズを含んでしまうレベルまで閾値を下げると、この欠落部分にもエッジをかなり検出できることがわかる。そこで、この2種類の閾値を用いてエッジ投影を表1の位置付けのもとで3値化する。図6(a)のエッジ投影を3値化した結果を図6(b)に示す。

次に、1の画素の両隣の2近傍を調べ、2の画素が存在すれば画素値を2に変更する。この操作を N 回繰り返す。ここで、 N は画像空間における車両の x 軸方向の最大長を見積もった数値である。この後、画素値0または1の画素を0に、画素値2の画素を1に書き換える。図6(b)の3値化エッジ投影に、上記の2値化処理を行った結果を図6(c)に示す。

3.3.2 2値投影の穴埋め

適応2値化処理で得られた2値投影の画素値1の領域に閾値未満の穴領域があったら、穴領域の画素値を全て1にする。図6(c)の2値投影に穴埋め処理を適用した結果を図6(d)に示す。

3.3.3 2値投影のノイズ除去

2値投影の画素値1の領域に閾値未満の幅の部分領域があったら、その部分領域の画素値を全て0にする。すなわち、穴埋め処理を、画素値0の領域と画素値1の領域とで反転して行うと考えればよい。図6(d)の穴埋め処理結果に、上記のノイズ除去処理を行った結果を図6(e)に示す。

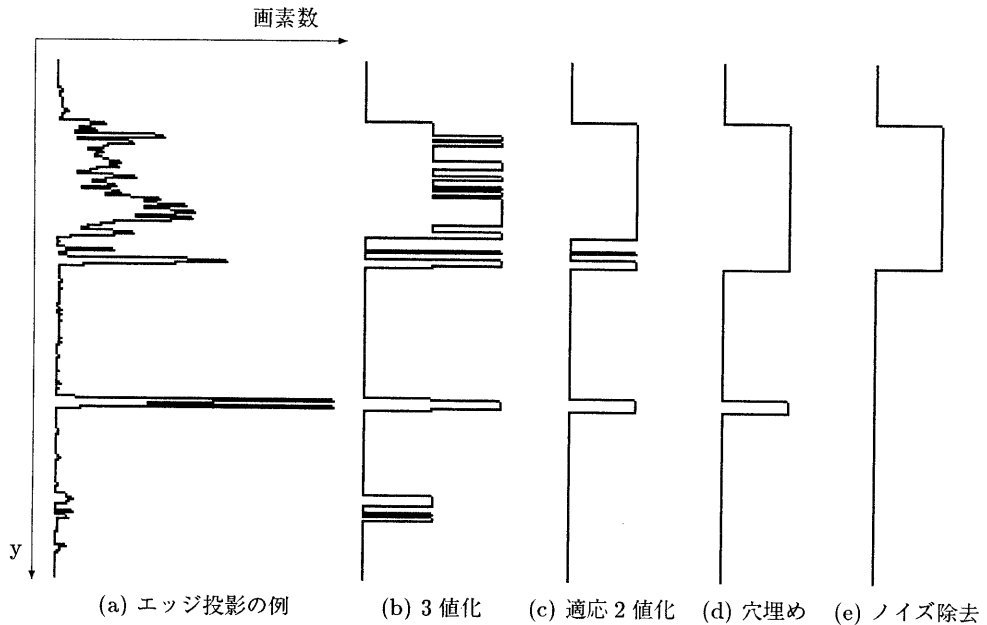


図 6: 投影データの整形処理

4 車両の追跡

車両検出処理の結果、各時刻に 2 値投影が得られる。検出された車両位置情報から種々の計測が可能であるが、今回筆者らが考案し車両検出処理の評価に用いた、車両の追跡により台数をカウントするアルゴリズムについて説明する。

4.1 アルゴリズムの概要

図 7 に車両追跡処理の概要を示す。車両検出処理により得られた 2 値投影から画素値 1 の領域の両端の座標を求め、これを車両の位置データとする。これらの座標は、画像上における車両領域の上端と下端に対応する。前フレームの位置データ群との対応を調べ、位置データに車両識別用のラベルを与える。また、前フレームからの予測データを参照し、検出エラーの補完をする。その際、非検出回数から車両の消滅判定を行ない、車両をカウントする。また、消滅していない位置データに関しては次のフレームの予測データを生成する。

4.2 フレーム間位置データのラベル付け

2 値投影から得られた位置データの追跡を行なう。1 フレーム前までを追跡済みと仮定し、現在の時刻で

得られた位置データに対し、1 フレーム前の位置データ群を参照し、下記要領で車両を識別するためのラベルを付与する。すなわち、1 フレーム前の位置データと対応が見つかった場合には、その位置データのラベルを継承する。

1. 重なりのある位置データが存在する場合、重なりが最大である位置データのラベルを継承する。
2. 1 で対応がつかなかった場合、互いの近い方の端点間の距離が最小かつ閾値以内の位置データのラベルを継承する。
3. 1、2 で対応がつかなかった場合、位置データの下端が、指定された画像下部領域内に存在するときだけ新規のラベルを付与し、それ以外の場合は無視する。

ここで、新規のラベルを与える場合に、車両が画像下部から進入することを利用し、それ以外の領域で検出された位置データは検出エラーと判断する。

4.3 検出エラーの補完と車両のカウント

車両が一旦検出され、車両検出のエラーを生じた後、再び同じ車両が検出されることがある。

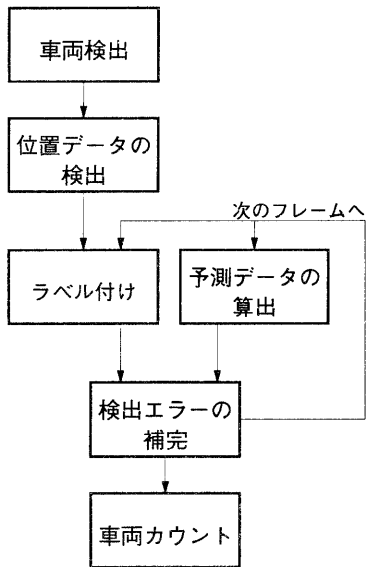


図 7: 車両追跡処理

常に 1 フレーム前のデータだけを参照してラベルを与える方法では、検出エラーを生じた時刻以降の検出データにラベルを伝播できない。そこで、現在の車両データから次の時刻で検出される位置データを予測し、その予測データと実際の検出結果とを比較して情報の欠落を防ぐ。

時刻 t におけるラベル l の位置データに対応する車両領域の画像上の両端の y 座標を $Y_{l,1}(t)$ 、 $Y_{l,2}(t)$ 、画像上の車両の中心位置 $C_l(t) = (Y_{l,1}(t) + Y_{l,2}(t)) / 2$ の移動量を $V_l(t)$ (pixel/frame) とする。移動量の初期値は 0 である。移動量の算出には、過去 2 フレームのみを用いるとノイズの影響により安定しないため、過去の追跡結果を反映できるように次式による逐次的な更新方法を用いる。

$$V_l(t) = \frac{1}{2}V_l(t-1) + \frac{1}{2}(C_l(t) - C_l(t-1))$$

上記の移動量の算出方法を用いて次の手順にしたがい、位置データの予測を行なう。

1. 時刻 $t+1$ における位置データを $Y_{l,1}(t) + V_l(t)$ 、 $Y_{l,2}(t) + V_l(t)$ と予測する。
2. 予測データと検出された現在のフレームから検出された車両の位置データとを比較し、各予測データに対応する位置データが存在しないとき、その予測データを位置データとして追加する。

特定の車両が一定回数連続して検出されなかった場合には、検出エラーではなく車両が消滅したと判断し、予測データによる補完を中止すると同時に、車両カウント値をインクリメントする。

5 実験

本稿のアルゴリズムを実際の道路画像に適用し、実験を行った。実験画像は一般国道を日中に 1 時間俯瞰撮影したものであり、サイズは 512×240 である。

図 1 の入力画像に対する処理の途中結果を図 8 から図 10 に示す。まず、図 8 はエッジ検出の結果得られた 2 値エッジ画像である。この段階では背景のエッジと車両エッジが混在している。図 4 の背景エッジマスクを用いて、車両のエッジだけを抽出した結果を図 9 および図 10 に示す。図 9 が走行車線上の車両エッジで、図 10 が追越車線上の車両エッジである。

また、撮影範囲内には手前と遠方に交差点があり、図 11 に示すように、計測範囲をこれらに挟まれる区間とした。なお、図 11 では計測範囲に指定している領域以外の輝度値を $1/2$ 倍にして表示している。

本手法による車両カウント実験結果を表 2 に示す。ここで、実台数は人間の視覚による確認値である。実台数 (センター別) では、2 車線にまたがって走行した車両を車線ごとの実台数に加えない場合を示しており、実台数 (センター込) では、このような車両を走行車線、追越車線ともに自車線の車両としてカウントした場合を示している。また、計測台数は本手法によるカウント値、正常計測台数は本手法で 1 台としてカウントされた車両の台数、ドロップは本アルゴリズムでカウントされなかった車両の台数、ダブルは本アルゴリズムで二重カウントされた車両の台数、センターは走行車線と追越し車線にまたがって走行した車両の台数である。ここで、本手法の最終出力となるカウント値では、ドロップおよびダブルのエラーがともにある場合にこれらが相殺されることに注意しなければならない。そこで、車両カウントの計測精度を 2 種類の観点から算出した。まず、計測精度 A を、ドロップやダブルのエラーを全て除く正常計測台数の実台数 (センター別) に占める割合で示した。そして、計測精度 B を、本手法による計測台数の実台数 (センター込) に占める割合で示した。

実験の結果、本手法でダブルのエラーを生じた車両は、計測領域内で車線変更するなど 2 車線にまたがって走行したものである。また、本手法でドロップのエラーを生じた車両は、前方の車両との車間距離を短く

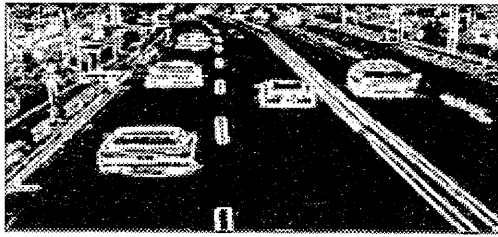


図 8: エッジ検出結果

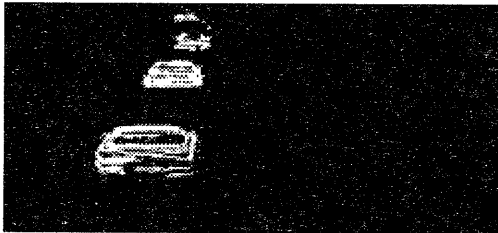


図 9: 走行車線の車両エッジ

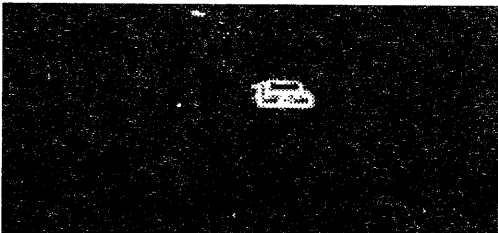


図 10: 追越車線の車両エッジ

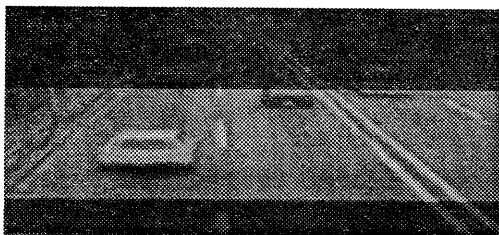


図 11: 計測範囲

	走行車線	追越車線	センター	全体
実台数 (センター別)	599	543	34	1176
実台数 (センター込)	628	569	—	1176
計測台数	628	569	—	1197
正常計測台数	594	535	—	1129
ドロップ	5	8	—	13
ダブル	(34)	(34)	34	34
計測精度 A (%)	99.2	98.5	—	96.0
計測精度 B (%)	99.2	98.6	—	101.8

表 2: 車両カウント実験結果

取ったために車両間に隠蔽を生じ、2台に分離してカウントされなかったものである。

6 おわりに

まず、背景画像の生成アルゴリズムを考案し、車両検出処理へ背景エッジマスクという形で応用した。このアルゴリズムにより、道路監視画像から背景のエッジによる影響を受けずに安定した車両の検出を行うことができた。さらに、俯瞰画像を対象とした車両検出アルゴリズムを車線毎の横エッジ投影法を利用することにより考案した。本アルゴリズムを車両カウントの計測率で評価した結果、98～99%の精度を得た。

今後は、道路が混雑した場合に車両間に生ずる隠蔽への対応策、道路体の側方に設置されたカメラからによる撮影画像への対応策、さらに、エッジ検出の他に、背景差分や時間差分などの他の画像処理情報を利用する方法を検討していきたいと考えている。

謝辞 本研究の機会を下さった NEC 情報メディア研究所パタン認識研究部の天満部長、また、本研究を進めるにあたり、ご指導ご討議頂いた NEC 映像開発本部の皆様にご感謝致します。

参考文献

- [1] 林田, 福島, 外部環境変化に強い画像形車両感知器, OMRON TECHNICS, Vol.32, No.2 (102号), pp. 218-223 (1992).
- [2] 井上, 小畑, 生井, 半場, 車両動態計測の試み, 第17回画像工学コンファレンス, pp.295-298 (1986).
- [3] 黒田, 吉川, 関, 監視システムの画像処理, 映像情報, (1992/9).
- [4] 酒井, 井藤, 根岸, 出口, 西山, 加藤, 監視用ITVカメラを用いたトンネル内交通流計測装置の開発, 住友電気, 第134号, pp.86-92 (1989).
- [5] 関, 黒田, 動き情報を用いた高信頼型侵入監視装置, 三菱電機技報, Vol.67, No.7 (1993).
- [6] 谷口, 古澤, 関, 池端, DTT法を用いた車両認識, 情報CV研72-5 (1991).
- [7] 辻角, 背景画像生成方法及びその装置, 特願平01-250869.