

車載カメラ映像の知的解析によるバス乗降客数推定システムの実装

高尾 凌我^{†1} 池内 紀貴^{†2} 鈴木 秀和^{†2} 松本 幸正^{†2}
^{†1} 名城大学理工学部 ^{†2} 名城大学大学院理工学研究科

1 はじめに

with コロナ時代においては、混雑を避けて移動したいという市民の需要が考えられる。この需要に応えるためには、例えばバス事業者がバス車内の混雑情報を利用者へ提供する必要がある。筆者らは、車載カメラ映像の知的解析によるバス乗降客数推定システムを提案してきた [1]。本稿では、提案システムの実装および性能評価について述べる。

2 従来システム

文献 [2] において、単一カメラで画像処理に基づいて、双方向通過人数をカウントする手法が提案されている。間に一定の間隔を保った入口線と出口線を設定し、その通過方法によって入る人と出る人、入らずに引き返す人、出ずに引き返す人の4種類に分類する。しかし、バス車内では線の間隔を確保することが難しい。

文献 [3] において、セルラーネットワークと地理情報を利用して、人の移動形態や軌跡を推定する手法が提案されている。携帯電話が基地局に信号を送信する際のセルラー基地局で生成された制御信号記録と、鉄道網や道路網などの地理情報を組み合わせることで実現している。しかし、通信会社であればその情報は取得可能であるが、バス会社が取得することは難しい。また、携帯電話を所持していない人の動きを把握することができない。

3 提案システム

従来システムの考えを踏襲しつつ、路線バスに適用可能な手法として、車内画像を一度も保存することなく、乗降客数を推定する。図1に提案システムの処理概要を示す。バス停前後の区間での各乗客の特徴点情報を比較することにより、バスの乗降客数を推定する。GPSにより取得したバスの位置情報から、バスの運行状態を推定する。バスが停留所区間を走行中の場合、バス車内の乗客情報から乗客数を推定する。バスが停留所に停車中の

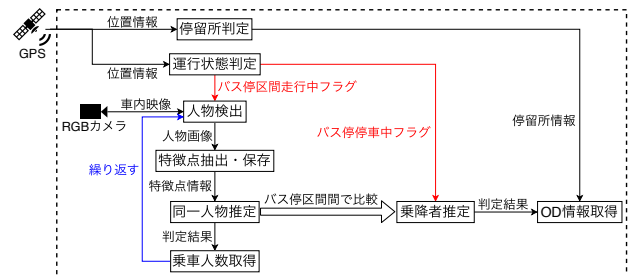


図1 提案手法の処理概要

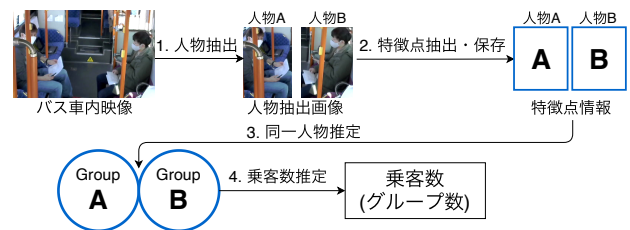


図2 実装した乗客人数推定処理の流れ

場合、停車中の直前のバス停前後のバス停区間の乗客情報から乗降者を推定する。そして、GPSによる位置情報からのバス停推定結果と乗降者推定結果を組み合わせることにより、OD (Origin Destination) 情報を推定する。

4 実装と評価

4.1 実装

図2に乗降客数推定システムの一部である乗客人数推定処理の流れを示す。Raspberry Pi 4 Model B, Raspberry Pi Camera V2 および Edge TPU (Tensor Processing Unit) を搭載した Coral USB Accelerator を用いて、運行中に行う乗客人数推定処理を実装した。TensorFlow Lite のオブジェクト検出モデルを用いて、撮影映像から人物検出を行い、人物画像を抽出する。抽出した人物画像から、OpenCV の AKAZE を利用して局所特徴量に基づく特徴点情報を取得し、pickle ファイル形式で特徴点情報のみを保存する。保存した特徴点情報から、特徴記述子を用いて k 近傍法によるマッチング処理を行い、距離が近い上位 2 個の特徴点を探す。1 番近い距離が 2 番目の距離の半分以下であるものでデータを間引きし、その数を数える。数えた数が設定した閾値以上であれば、同一人物であると推定し、一意なユーザ ID を動的に割り当てる。そして、割り当てたユーザ ID の数を乗客数とし

Implementation of Bus Passenger Estimation System with Intelligent Video Analysis

Ryoga Takao^{†1} Noriki Ikeuchi^{†2} Hidekazu Suzuki^{†2} Yukimasa Matsumoto^{†2}

^{†1} Faculty of Science and Technology, Meijo University

^{†2} Graduate School of Science and Technology, Meijo University

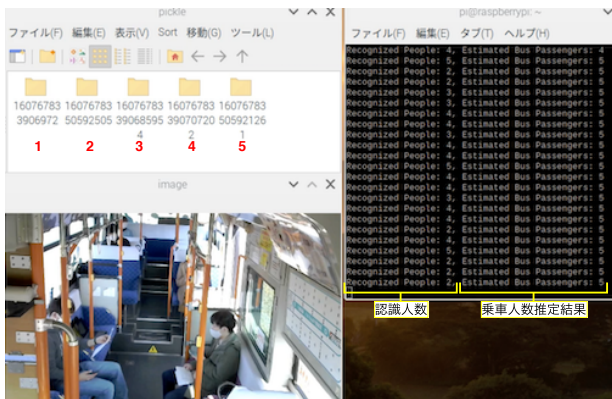


図3 実装した乗車人数推定処理の動作確認

て推定する。

図3に5人が乗車したバス車内動画で動作確認をした結果を示す。左下は乗車人数推定処理を行っているバス車内映像の対象画像であり、左上のフォルダリストには同一人物推定処理の結果を示している。フォルダ数が推定人数を意味するため、5名分の特徴点情報が保存されていることが確認できる。右のコンソール画面は、左側に処理時点における認識人数を、右側に乗車中に推定したトータル的人数である。処理のタイミングによっては人物が障害物と重なって映るため、認識人数に誤差は生じるが、トータルの乗車人数を正しく推定することができた。

4.2 処理時間の評価

5人がバス車内に乗車している映像から50枚のフレーム画像について前述の処理に要した時間を評価した。表1に実装した乗車人数推定処理の処理時間を示す。乗車人数推定処理の平均処理時間は、99.47[ns]であった。筆者らが提案および実証実験を行っているIoTバスロケーションシステム[4]では、3秒程で位置情報とバス停への接近状態の送信処理を行い、GPS取得間隔として2.5秒空けている。つまり、5~6秒に1回の頻度でバスの位置情報を送信している。そのため、5秒以内に実行が可能であれば、現在の送信間隔を維持したまま本システムを組み込むことができる。実装した乗車人数推定処理は、71.86~291.29[ns]で処理が完了していたため、IoTバスロケーションシステムに組み込むことが可能である。

表2に実装した乗車人数推定処理における各機能の平均処理時間を示す。5人がバス車内に乗車している写真1枚の物体検出に平均23.43[ns]を要していた。また、1人分の人物抽出に0.47[ns]、特徴点抽出に6.04[ns]、同一人物推定に0.86[ns]の処理時間で完了していた。表1の平均処理時間に基づくと、50人乗車した場合は約10倍の1[μs]かかると考えられる。5秒間に乗車人数推定処

表1 乗車人数推定処理における処理時間

Max [ns]	Min [ns]	Avg. [ns]
291.29	71.86	99.47

表2 各機能の平均処理時間

処理内容	平均処理時間 [ns]
1枚の画像の物体検出	23.43
1人の人物抽出	0.47
1人の特徴点抽出	6.04
1人の同一人物推定	0.86

表3 アクセラレータの有無による物体検出時間の比較

Coral 使用時 [ns]	Coral 未使用時 [ns]
23.43	191.31

理を 5×10^6 回実行できることになるため、処理速度は十分であると考えられる。今後、バス車内の環境に適した人物検出モデルを構築および適用することにより、更なる高速化と人物認識精度の向上が図れると考えられる。

表3にCoral USB Acceleratorの使用有無における物体検出の平均処理速度の比較結果を示す。5人がバス車内に乗車している写真1枚の物体検出に平均処理時間は、Coral USB Accelerator使用時が23.43[ns]、未使用時が191.31[ns]であった。Coral USB Acceleratorを使用すると、物体検出処理時間が約12.2%まで短縮された。送信するまでの5秒間により多くの乗車人数推定処理を行い、乗車人数推定の精度を上げるために、物体検出処理時間は100[ns]以内に抑えたい。そのため、Coral USB Acceleratorは使用すべきであることが確認できた。

5 まとめ

本稿では、提案している乗降客数推定システムの実装と評価を行なった。乗車人数推定処理時間を評価し、IoTバスロケーションシステムに組み込むために必要な処理時間を満たしていることを確認した。今後は、バス車内の環境に適した人物検出モデルを作成し、バス車内のあらゆる利用状況で動作確認を行う。

謝辞

本研究の一部は総務省SCOPE(受付番号191506001)の委託を受けたものである。

参考文献

- [1] 高尾. 他: 電気・電子・情報関係学会東海支部連合大会講演論文集, Vol. 2020, No. H2-3, 2020.
- [2] Cao, J., et al.: CCDC 2016, pp. 2048-2051, 2016.
- [3] Yamada, Y., et al.: 9th IFIP WMNC 2016, pp. 138-144, 2016.
- [4] Boshita, T., et al.: IEEE ITSC 2018, pp. 933-938, 2018