



自動運転ソフトウェアを活用した死角領域の半隠消表示

竹村岩朗¹⁾, 池田聖²⁾, 木村朝子¹⁾, 柴田史久¹⁾

1) 立命館大学大学院 情報理工学研究科 (〒525-8577 滋賀県草津市野路東 1-1-1)

2) 立命館大学 情報理工学部 (〒525-8577 滋賀県草津市野路東 1-1-1)

概要: 道路環境において死角領域を可視化する隠消現実感技術は, 追突事故の防止や同乗者のストレス軽減への応用が期待できる. しかし, 道路環境における隠消現実感に関する先行研究の多くが将来普及する保証のないセンサやインフラを前提とするため, 実現性は不透明である. 本研究では, 自動運転車が普及した環境を前提とし, 自動運転用ソフトウェアモジュール群と自動運転車に搭載される最小限のセンサから得られるデータを用いて死角領域を半透明に可視化 (半隠消表示) するシステムを提案する. 提案システムは, 車両の位置姿勢を推定する SLAM (Simultaneous Localization and Mapping) 機能や三次元点群を画像に投影する機能など可能な限り既存のモジュールを再利用する方針で設計した. 実験では, 自動運転モジュール群に Autoware を, 二組の公道で取得した Tier IV 社の公開データを使用し, 提案システムにより死角領域を可視化できることを確認した.

キーワード: 隠消現実感, 自動運転ソフトウェア, LiDAR

1. はじめに

建物や自動車により生じる死角を自動車内で可視化する[13]隠消現実感(Diminished Reality; DR)技術[9]は, 自動運転車社会が到来しつつある現代においても, 運転者および同乗者の安全性だけでなく安心感を与えることに貢献する. 運転者の自発的な操作やシステムに対する応答が要求される可能性のある運転レベル 0 から 4 の場合は, 遮蔽物の半透明化により, 歩行者や車などの存在を視覚的に伝え, 事故を未然に防ぐことが出来る. 運転手の存在を必要としない運転レベル 5 の場合は, 事故原因となりうる歩行者や車の存在を自動車が検知し急停止できるだろうが, 利用者にその急停止の原因を視覚的に提示することで安心感を与えることができるかもしれない[14].

道路上の死角を可視化する応用に関する従来研究の多くは, それらの実現に必要な専用の機器構成やインフラが利用できることを前提として検証実験がなされている. Kojima らは電柱に設置された道路監視カメラを利用し, サイドミラーやバックミラーなどに映らない死角情報を可視化するシステムを提案した[8]. このシステムでは, カメラで撮影した映像を仮想の鏡により提示する. こうした研究における実験では, 実際の自動車に搭載された環境に設置される可能性が低い機材を利用しており, 実現性が低い. また, Rameau らは利用者が乗る車両 (自車両) の前を走る車両 (前方車両) を視覚的に除去する死角可視化システムを提案した[13]. 具体的には, 前方車両

に搭載されたステレオカメラから取得した点群にテクスチャマッピングし, 自車両と無線共有することで自車両から見た死角の情報を可視化することができる. この研究では既に車両に搭載されているステレオカメラを活用し死角領域を可視化しているため, 実現性がかなり高い. しかし, 可視化する対象は体験者が乗る自車両の前方を走行する車両のみであり応用が限られる.

本研究の目的は, 自動運転用の実験に利用される公開データとソフトウェアモジュールを可能な限り再利用し, 他車両が取得したデータにより道路環境における様々な死角を可視化できることを確認することである. これにより, 自動運転のために設置されるカメラなどのセンサ群, ソフトウェアモジュールの内どの部分が流用でき, どのような機能や性能が不足するのかを明らかにする. そのために実験では, Tier IV 社が公開する自動運転の公開ソフトウェアである Autoware と公開データを用いる. これらのソフトウェアと公開データのみを用いることで, 自動運転車が多数走行する環境で自動運転車に搭載されるセンサのみで死角を可視化できることを示す.

2. 関連研究

2.1 道路環境における複合・隠消現実感技術

筑波大のグループはこれまで, 道路環境や車両における複合現実感(Mixed Reality; MR)及び DR 技術についてこれまで多数の研究を発表している[15]. 例えば Sasai らは車輪の動きを予測し, その予測軌跡を自車両によって見えない道路の表面上に可視化するシステムを提案した[14]. この研究を通して彼らは, 利用者から見えない領域を可視化することで利用者の不安が軽減されることを示した.

Iwao TAKEMURA, Sei IKEDA, Asako KIMURA
and Fumihisa SHIBATA, Ritumeikan University

また、DR 技術を活用した他のアプリケーションとして自車両の前方を走行している車両を視覚的に消し、より遠くの方の風景を可視化する手法が挙げられる[2,5,11]。現在は、前方を走行している車両が観測した前方の風景をリアルタイムでレンダリングし、実時間で死角領域を可視化した[13]。本研究ではこのような特定の状況を想定していない。

屋外環境の描画技術の研究として LiDAR(Laser Imaging Detection and Ranging)の 3 次元点群を用いた Point-based Rendering に関する研究が挙げられる[7,10]。我々は、このような基本的なレンダリング技術を研究する立場にはなく、既存技術を活用する立場にある。

2.2 自動運転用プラットフォーム

Autoware[6]の他にも、自動運転シミュレーション用のソフトウェアがいくつか公開されている。例えば、自動運転について研究するためのオープンソースのシミュレータとして CARLA Simulator が挙げられる[3]。このソフトウェアは Python によるインタフェースと仮想環境モデルを提供している。また類似したソフトウェアとして Microsoft 社の AirSim がある[16]。Udacity 社の Self-Driving Car Nanodegree プログラムでは、Autoware を基盤にオンライン教育サービスの一環として自動運転プログラムのソース公開している。本研究はこうしたプラットフォームに関する研究ではなく、これらの一部を再利用する立場にある。特に今回は今後追加でデータ取得することも考慮に入れ、日本国内で取得されている Tier IV のデータセットを採用し、実際の自動運転実績がある Autoware を使用する。

また、Autoware には、LiDAR の点群を GPS の地理座標系で統合した 3 次元地図の作成[18]や自己位置推定[1,17]、物体検出[4]、物体認識[19]などの基礎技術を使用されている。本研究では、こうした個々の自動運転用の基礎技術を改善する立場でもなく、実現可能性と再利用性の高いソフトウェアで DR 技術の実装を試みる立場にある。

3. Autoware を基にした DR システム.

3.1 DR に関連する Autoware モジュール

Autoware は Robot Operating System (ROS)上に実装されており、多数の基本的なモジュールを組み合わせた自動運転ソフトウェアである。ROS は publish/subscribe モデルによりモジュール間の情報伝達の実装されており、追加のモジュールを容易に接続することができる。Autoware の主なモジュールは以下の通りである。

センシング : LiDAR, RGB カメラ, GPS, 慣性センサなどのセンサからデータを取得する。

SLAM : LiDAR で取得した点群データから LiDAR の位置姿勢を取得し、3 次元地図を作成する。

自己位置推定 : 3 次元地図と LiDAR で取得する現フレームの点群をマッチングすることで自己位置を推定

動物体検出 : RGB カメラで取得した画像と LiDAR で取得した現フレームの点群によって歩行者、車を検出

静止物体検出 : 自車両の自己位置, RGB 画像, 歩道や

建物, 信号といった静止物体の位置が格納されているベクターマップを使用し, 静止物体を検出する。

経路計画 : ベクターマップと自車両の位置から目的地までの詳細な経路と適切な速度を判断する。

車両制御 : 自車両の自己位置, 適切な速度, 経路選択によって自車両を制御する。

著者らは上記のモジュールの中で、下線で示すモジュールを使用し、DR システムを構築する。またこれらの他にも、静止物体や動物体を検出するために RGB 画像上に点群を投影する投影モジュールを活用する。

3.2 DR モジュールの入出力

DR モジュールの入力データは三種類のネットワークを介してデータを受け取る。一つ目に、カメラパラメータを含んだ RGB 画像や除去する対象のデータは自車両から直接渡される。二つ目にカメラパラメータを含んだ死角領域の画像は、死角領域を観測する他車両からもしくは、過去の画像であればデータベース(動物体などの情報を含むデータをダイナミックマップ(dynamic map)と呼ぶ)から R2V か V2V ネットワークを介して取得する。最後に、3 次元地図データは V2I ネットワークを介してデータベースから取得する。これらの入力データを DR モジュールに与えることで、死角領域が表示された画像が出力される。

DR モジュールでは、従来の簡易的な Point-based Rendering 手法に基づいて死角領域が描画される。具体的にはまず、3 次元地図内に点在する点群はパッチとして死角領域が写っている RGB 画像に投影される。同様の点群を自車両から観測し取得した RGB 画像に投影し、死角領域に投影されたパッチを重畳する。

奥行きを知覚するために、投影されたパッチは標準偏差が 10 ピクセルのガウスフィルタによってぼかしを付与し、除去対象物体のエッジを重畳する。

3.3 モジュール間の接続関係とその簡略化

我々は図 1 に示すように Autoware ソフトウェア内のモジュールを活用する。またこのとき、各車両からの要求に従って、Dynamic Map の一部分を提供、及び送信するデータベースが存在すると仮定する。Dynamic Map は複数のレイヤーによって構成されており、その中には 3 次元地図、画像、ベクターマップの情報が格納されている。DR モジュールは別車両から観測した死角領域内の動物体の情報とデータベースから送られる画像及び 3 次元点群によって死角領域を可視化するために、SLAM モジュールと点群モジュールを活用する。

現在のところデータベースや死角領域を観測するための車両を用いず、図 1 で示したモジュール間の接続を簡略化する。本稿では、図 2 に示すように 2 台の車両間での直接的な通信を想定する。さらなる単純化のために、一台の車両が取得した同じシーケンス上のデータの一部を他車両が取得したデータとして使用する。具体的には、障害物によって死角領域が遮られているフレームを自車両が取得したデータとし、同じ車両から死角領域を観測すること

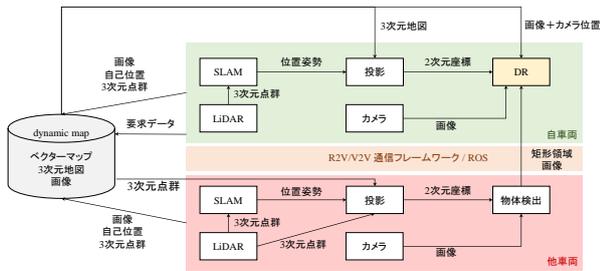


図1 理想のDR モジュール群

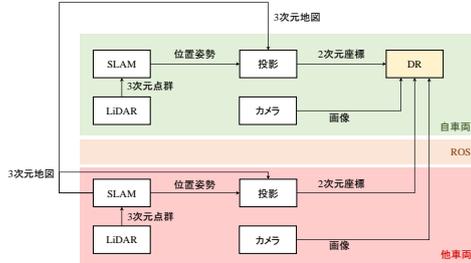


図2 実験に使用するDR モジュール群

ができたフレームを他車両が取得したデータと仮定する。

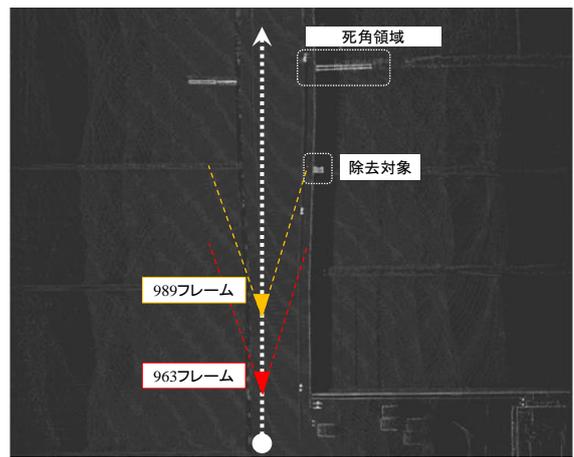
4. 実験

4.1 入力データ

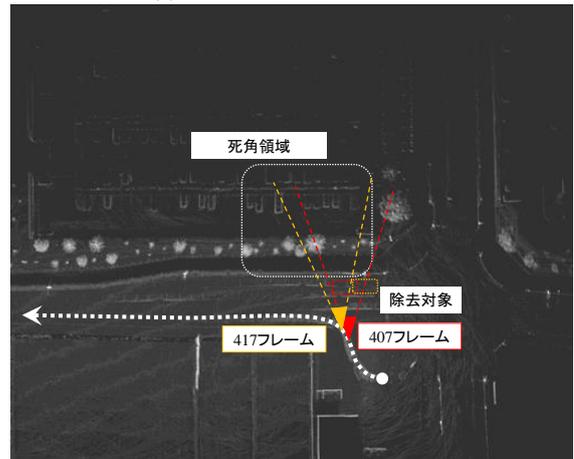
障害物の半透明化画像が自動運転シミュレータ用の実データのみで生成可能かを確認するために、Tier IV社が提供する自動運転用のデータセットを使用する。Tier IV社はBAG形式で自動運転車が取得した道路周辺の3次元点群、画像、そしてGPS位置などの一連のセンサデータを含む多数のデータセットを提供している。2018年7月20日時点では、28個のBAGファイルが用意されており、そのすべてにオブジェクト認識のためのカメラ画像が含まれている。BAGファイル内のデータにはタイムスタンプが含まれており、データを取得した当時の状況をシミュレーションすることができる。

Tier IV社が提供している全てのBAGファイルの画像シーケンスを目視で確認し、次の基準に従って実験に使用するBAGファイルを選択した。(i) 特定のフレームにおいて死角領域が障害物によって見えず、かつ別フレームのカメラ位置からは死角領域を観測することができる。(ii) 死角領域が静的な物体のみ構成されかつ特徴的なテクスチャがある。これらの基準に従い、anjo_01_2017-02-23.bagという名前のBAGファイルから二箇所切り出して使用する。

一箇所目は、図4(a)のように背景の看板(死角領域)を隠している広告板(除去対象)が含まれている963フレーム目のデータを使用した。989フレームは、カメラが死角領域を観測できる位置にある。それゆえ、963フレームを自車両の位置、989フレームを他車両が死角領域を観測している位置と仮定し、死角領域を半透明表示した。二箇所目は、図5(b)のようにガードレールや駐車してある車両(死角領域)を隠している車両(除去対象)が含まれている407フレーム目のデータを使用する。このとき、除去対象である車両は407フレームから417フレームの間で別の位置に移動している。そのため、カメラの動きが非常に小さいが、419フレームでは死角領域を観測出来る。



(a) 除去対象が静的な場合



(b) 除去対象が動的な場合

図3 実験イメージ

4.2 LiDAR・カメラ間のキャリブレーション

LiDAR座標系からカメラ座標系への剛体変換を表すパラメータはTier IV社が提供するキャリブレーションデータを使用する代わりに、複数のフレームを使用しPnP問題を解決することで取得した。Tier IV社のキャリブレーションデータでは、パッチを重畳するには精度が不十分であったためである。さらに、Autoware内にあるキャリブレーション機能は正確なパラメータを取得するために手作業で行う工程が多く、非常に時間がかかったためである。

我々は20個以上のLiDARの3次元点とそれに対応する画像内の2次元点を手動で対応付けし、この処理を複数のフレームで行う。2次元点と3次元点の対応によりPnP問題を解くことができる。これには、OpenCVライブラリ内にあるsolvePnP関数を使用した。このような手動による処理は自動でキャリブレーションする手法[12]に置き換えることができる。

4.3 結果と考察

図4には静的、図5には動的な除去対象に対して死角領域を半透明表示するまでの生成過程を示す。図4(a)内にある四角の囲いは手動で指定された除去対象領域を表している。赤い曲がった矢印を含む白い広告板が半透明化する除去対象である。図4(b)は同様の除去対象領域内でエッジを抽出した結果である。図4(c)内の赤いドットは投

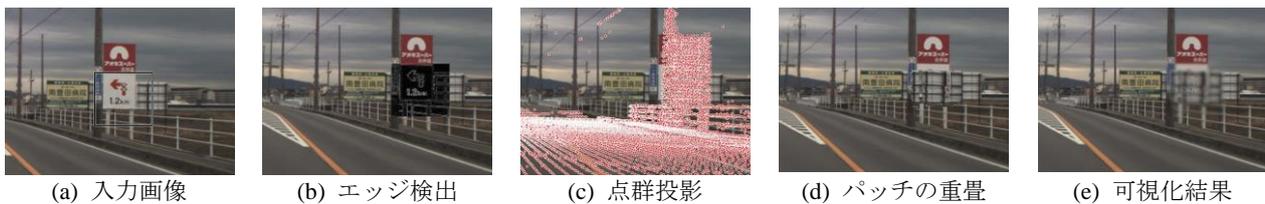


図4 死角領域の半透明表示の生成過程 (除去対象が看板の場合)

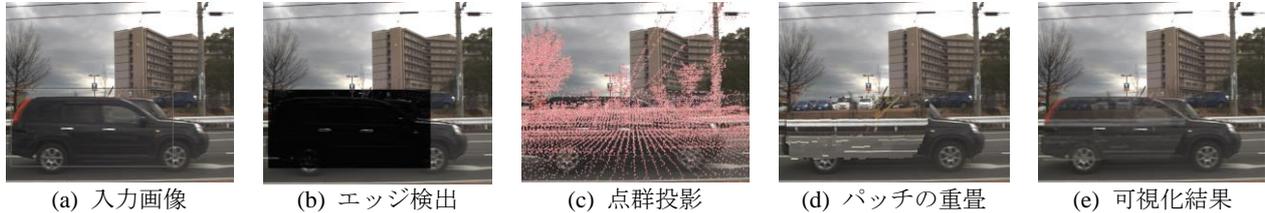


図5 死角領域の半透明表示の生成過程 (除去対象が車の場合)

影した点群を表す。図4 (d) は死角領域のパッチを投影した画像で、図4 (e) は除去対象領域内でガウスフィルタをかけ、テクスチャ付きのエッジを重畳描画した画像である。

図4 (c) の結果を確認すると、電柱に対応した点群がわずかにずれた位置に投影されている。このことから、LiDAR・カメラ間のキャリブレーションパラメータの精度が不足していることが分かる。図4 (d) の結果を確認すると、単純なパッチベースの手法で描画したことによりパッチの境界線にズレが発生している。このようなズレは図4 (e) のようにぼかし、除去対象物体を半透明で重畳することで殆ど目立たないことが分かる。図4 (e) の結果から除去対象領域内の死角領域の概要を把握できる。図5でも同様の結果が確認された。

しかし、各実験では一つの画像シーケンスの中から二つのフレームを取得し、これらのフレームを自車両、他車両がそれぞれ観測し取得した画像として扱った。このような実験条件は画像生成において有利に働いてしまう可能性がある。具体的には、取得した二つのフレームは時間差が小さいため、照明条件やカメラ位置の変化が小さい。そのため今後は、完全に異なる二つの車両からそれぞれ取得したセンシングデータを使用し、今回提案した実験条件と同様の方法で実験する必要がある。

5. 結論と今後の展望

本稿では動的/静的物体によって遮られていた静的な死角領域の風景を半透明表示するDRモジュールとそれを利用したDRシステムについて提案した。DRモジュールは自動運転車用の公開ソースソフトウェアAutowareを基盤に設計した。実験では自動運転をシミュレーションするために提供されている既存のデータセットのみ使用し、DRシステムが実現可能であることを実証した。このことから、将来自動運転が普及している環境下では、DRモジュールを追加するだけで死角領域を可視化することが出来ると言えるだろう。今後は死角領域に動物体が存在した場合に対応したDRモジュールを設計する。

参考文献

[1] P. Biber and W. Strasser: The normal distributions transform: a

new approach to laser scan matching, Proc. IROS, vol. 3, pp. 2743 - 2748, 2003.

- [2] H.-I. Chen, *et al.*: Integrating dashcam views through inter-video mapping, Proc. ICCV, pp. 3110 - 3118, 2015.
- [3] A. Dosovitskiy, *et al.*: CARLA: An open urban driving simulator, Proc. CoRL, pp. 1 - 16, 2017.
- [4] P. F. Felzenszwalb, *et al.*: Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 32, No. 9, pp. 1627 - 1645, 2010.
- [5] P. Gomes, *et al.*: The see-through system: From implementation to test-drive, Proc. VNC, pp. 40 - 47, 2012.
- [6] S. Kato, *et al.*: An open approach to autonomous vehicles, IEEE Trans. Micro, vol. 6, No. 6, pp. 60 - 68, 2015.
- [7] A. Kelly, *et al.*: Real-time photorealistic virtualized reality interface for remote mobile robot control, SAGE Trans. IJRR, vol. 30, No. 3, pp. 384 - 404, 2010.
- [8] K. Kojima, *et al.*: Naviview: Visual assistance by virtual mirrors at blind intersection, Proc. IEEE ITS, pp. 592 - 597, 2005.
- [9] S. Mori, *et al.*: A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects, IPSJ Trans. CVA, 2017.
- [10] S. Nebiker, *et al.*: Rich point clouds in virtual globes - A new paradigm in city modeling?, Computers, Environment and Urban Systems, vol. 34, No. 6, pp. 508 - 517, 2010.
- [11] C. Olaverri-Monreal, *et al.*: The see-through system: A vanet-enabled assistant for overtaking maneuvers, Proc. IV, pp. 123 - 128, 2010.
- [12] G. Pandey, *et al.*: Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information, Proc. AAAI, pp. 2053 - 2059, 2012.
- [13] F. Rameau, *et al.*: A real-time augmented reality system to see-through cars. IEEE Trans. on Visualization and Computer Graphics, vol. 22, No. 11, pp. 2395 - 2404, 2016.
- [14] S. Sasai, *et al.*: MR visualization of wheel trajectories of driving vehicle by seeing-through dashboard, Proc. ISMAR, pp. 40 - 46, 2015.
- [15] A. Sato, *et al.*: I. Kitahara, Y. Kameda, and Y. Ohta: Visual navigation system on windshield head-up display. In Proc. ITSWC, 2006.
- [16] S. Shah, *et al.*: AirSim: High-fidelity visual and physical simulation for autonomous vehicles. Proc. FSR, 2017.
- [17] A. Sujiwo, *et al.*: Localization based on multiple visual-metric maps, Proc. MFI, pp. 212 - 219, 2017.
- [18] E. Takeuchi and T. Tsubouchi: A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping, Proc. IROS, pp. 3068-3073, 2006.
- [19] M. Yoshioka, *et al.*: Real-time object classification for autonomous vehicle using lidar, Proc. ICIIBMS, pp. 210 - 211, 2017.