

エンドツーエンドで深層学習に基づく Visual SLAM の 敵対的パッチ攻撃に対する頑健性の分析

田中 成樹^{1,a)} 池西 俊仁¹

概要：Visual Simultaneous Localization and Mapping (vSLAM) は、自動運転や拡張現実感システム等へ活用可能な技術であり、近年では深層学習に基づく手法も多く提案されている。一方で、深層学習モデルには敵対的パッチ攻撃に対する脆弱性が広く知られており、本攻撃は AI システムの実運用において大きな脅威となる。本稿では、エンドツーエンドで深層学習に基づく vSLAM への敵対的パッチ攻撃に対する頑健性を分析する。我々が知る限り、そのような vSLAM に対する敵対的パッチ攻撃手法は存在しないため、敵対的パッチ攻撃を提案することから始める。我々は、従来の画像認識等での攻撃と同様に、vSLAM に対する敵対的パッチ作成を最適化問題として定式化する。実験では、vSLAM の地図作成に関して、提案した攻撃によって敵対的パッチの貼付部分における地図に欠損が生じることを確認した。

キーワード：Visual SLAM, 敵対的パッチ攻撃

Analysis of Robustness against Adversarial Patch Attack on End-to-End Deep Learning-based Visual SLAM

NARIKI TANAKA^{1,a)} TOSHIHITO IKENISHI¹

Abstract: Visual Simultaneous Localization and Mapping (vSLAM) has a wide range of applications, including autonomous driving and augmented reality. In recent years, various methods based on deep learning have been proposed. However, it is widely known that deep learning-based models are vulnerable to adversarial patch attacks, which poses serious concerns for the practical deployment of AI systems. In this paper, we analyze the robustness of end-to-end deep learning-based vSLAM methods against adversarial patch attacks. To the best of our knowledge, there are no attack methods specifically targeting end-to-end deep learning-based vSLAM methods. Therefore, we start our study by proposing an adversarial patch attack on such vSLAM methods. Like existing works in other fields like image recognition, we formulate the generation of adversarial patches for vSLAM methods as an optimization problem. Our experiments show that the proposed attack can lead to some failures in mapping results in the regions where the patches are pasted.

Keywords: Visual SLAM, Adversarial Patch Attack

1. はじめに

Simultaneous Localization and Mapping (SLAM) は、リアルタイムに自己位置推定と周辺の地図作成を同時に行う技術であり、自動運転や拡張現実感システム等へ活用

可能である。入力データを計測するセンサには、カメラ、Light Detection and Ranging (LiDAR), レーダ等が想定される。カメラ画像を扱う Visual SLAM (vSLAM) は、センサの安価性、データの高解像度性等の面で他のセンサによる SLAM に対して利点を持つことから、研究が盛んに行われている [1]。近年の深層学習分野の発展に伴い、深層学習に基づく vSLAM 手法が多く提案されている。

一方で深層学習に基づく手法には、敵対的パッチ攻撃に

¹ 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

a) Tanaka.Nariki@bp.MitsubishiElectric.co.jp

に対する脆弱性が知られており、画像認識や物体検出を始めとする幅広い分野で報告されている [2], [3]。敵対的パッチ攻撃は、画像上の限られた領域（パッチ）内の画素値を変化させることで、深層学習モデルの精度を減少させる攻撃を指す。この脆弱性を利用することで、敵対的パッチをステッカー等として印刷して実環境内に貼付することで、当環境での深層学習モデルの適切な動作を阻害させられる。このようなステッカーは人間から見ると、悪意のあるステッカーとして認識されることは少なく、一種のデザインとして見逃されることが多い。そのため、敵対的パッチ攻撃は AI システムの実運用において大きな脅威となる。敵対的パッチ攻撃に頑健な手法を開発するためには、事前に AI システムの脆弱性を知る必要がある。しかし、深層学習に基づく vSLAM 手法に対する敵対的パッチ攻撃に関する論文は少なく、脆弱性は明らかになっていない。よって、安全性の高い vSLAM 手法の開発が難しい。

深層学習に基づく vSLAM 手法は、一部の機構のみを深層学習手法で置き換える手法と、エンドツーエンドで深層学習に基づく手法の二通りに大きく区分される。前者に関しては、自己位置推定を深層学習に基づく Visual Odometry (VO) で行う方法や、ループ閉じ込み向けのループ検知を深層学習技術で行う方法等が考えられる [4]。深層学習に基づく vSLAM に対して敵対的パッチ攻撃を行う場合を考える。前者に関しては、従来の VO モデルに対する敵対的パッチ攻撃手法 [5] やループ検知に対する敵対的パッチ攻撃手法 [6] 等を直接用いることで、vSLAM の精度を悪化させられることが予想される。しかし後者の場合、エンドツーエンドで深層学習に基づく vSLAM に向けた敵対的パッチ攻撃に関する既存研究は無いため、その脆弱性は未知である。

本稿では、エンドツーエンドで深層学習に基づく vSLAM に焦点を当て、それらの vSLAM 手法に対する敵対的パッチ攻撃手法を提案する。その後、提案攻撃を使用して vSLAM の脆弱性を分析する。動画を撮影するカメラは、比較的安価で小規模に導入可能である単眼の RGB カメラを想定する。実験では、提案した敵対的パッチ攻撃によって大きな影響を与えられなかったが、パッチ貼付部分の地図が欠損することを確認した。

本研究の貢献は以下である。

- エンドツーエンドで深層学習に基づく vSLAM に対する敵対的パッチ攻撃手法を初めて提案した。
- 実験により、提案攻撃は vSLAM が作成する地図において、パッチ貼付部分の欠損を誘発することを確認した。

2. 関連研究

vSLAM に関する研究では、ORB-SLAM [7] 等の多くの手法が提案され、高精度な自己位置推定及び地図作成を実

現している。近年では深層学習を取り入れた手法も提案されている。深層学習に基づく vSLAM 手法は、vSLAM を構成する一部の機構を深層学習技術で置き換える手法と、エンドツーエンドで深層学習に基づく手法の二通りに大きく区分される。前者の例として、Bruno と Colombini の LIFT-SLAM が挙げられる [8]。LIFT-SLAM では、従来の vSLAM の特徴点検出を深層学習に基づく手法に置き換えることで、センサノイズに対する頑健性を向上させている。その他、深層学習に基づく VO 手法やループ検知器等も従来の vSLAM 内の機構に入れ替えることができる [4]。後者の例として、Teed と Deng の DROID-SLAM があり、深層学習モデルによってオプティカルフローを算出した後、幾何学的制約により自己位置と周囲の点群を算出することで vSLAM を実現する [9]。彼らの実験では、ORB-SLAM に匹敵する推定精度を達成している。彼らの研究後も、エンドツーエンドで深層学習に基づく手法は数多く提案されているが、DROID-SLAM と比較して大きな精度差は見られない [10], [11], [12]。

敵対的パッチ攻撃に関する研究は、画像認識や物体検出等の様々な分野で報告されている [2], [3]。vSLAM に関する研究もいくつか存在する。Ikram らは、特徴的な模様のパッチを環境内の複数箇所に貼付することで、ループ閉じ込み向けのループ検知を誤検知させされることを報告している [6]。Nemcovsky らは、二枚の画像間の相対位置姿勢を推定する VO に対する敵対的パッチ攻撃を提案している [5]。彼らが提案した攻撃手法では、パッチ作成時に必要なパッチ部分の画像座標を ArUco マーカを用いた事前計測やシミュレータ環境の活用によって算出している。いずれも脅威的ではあるものの、エンドツーエンドで深層学習に基づく vSLAM 手法に対する結果は示されていない。

以上の理由から、本稿ではエンドツーエンドで深層学習に基づく vSLAM 手法に焦点を当て、その脆弱性を分析する。

3. vSLAM に対する敵対的パッチ攻撃

本稿では、エンドツーエンドで深層学習に基づく手法として DROID-SLAM を扱う。本節では、DROID-SLAM の処理を概説した上で、vSLAM に対する敵対的パッチの作成方法を定式化する。次に、パッチ部分の画像座標の算出方法と、パッチに対して必要な前処理を導入する。この際、Nemcovsky らの方法 [5] とは異なり、ArUco マーカを用いる事前計測やシミュレータ環境の構築を必要としない。そのため、提案攻撃の実現には大きな労力を要さない。最後に、パッチ作成時に使用する目的関数の詳細を説明する。

3.1 準備

以下では、動画 $\mathcal{I} = \{\mathbf{I}_t \mid \mathbf{I}_t \in \mathbb{R}^{H \times W \times 3}\}_{t=1}^T$ に対する DROID-SLAM の自己位置推定と地図作成に関する処理

を概説する。DROID-SLAM は、各フレーム I_t に対する自己位置 $G_t = (\mathbf{R}_t, \mathbf{q}_t) \in \text{SE}(3)$ と逆深度 $d_t \in \mathbb{R}_{>0}^{H \times W}$ を推定する。ここで、零より大きい実数の集合を $\mathbb{R}_{>0}$ で表記している。地図は、これら推定値 $\{G_t, d_t\}_{t=1}^T$ によって算出される点群である。時刻 t でフレーム I_t が入力されると、それ以前のフレーム $\{I_{t'}\}_{t'=1}^t$ に関する Covisibility Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ を作成し、隣り合うフレーム間 $(i, j) \in \mathcal{E}$ のオプティカルフローを RAFT [13] によって推定する。RAFT は、Gated Recurrent Unit (GRU) による再帰的処理を含むため、推定値を系列 $\{\hat{f}_{(i,j)}^1, \dots, \hat{f}_{(i,j)}^K\}$ として算出する。ここで各ステップ k における GRU の出力は、直前ステップ $k-1$ のオプティカルフロー推定値 $\hat{f}_{(i,j)}^{k-1}$ からの修正量 $\hat{\Delta}_{(i,j)}^k$ である。カメラ位置姿勢 \hat{G}_i^k と逆深度 \hat{d}_i^k は、ステップ k でのオプティカルフローの仮推定値 $\hat{f}_{(i,j)}^{k-1} + \hat{\Delta}_{(i,j)}^k$ を使用した Covisibility Graph \mathcal{G} 内でのバンドル調整（局所的な最適化）によって算出される。ステップ k での最終的なオプティカルフロー推定値 $\hat{f}_{(i,j)}^k$ は、カメラ位置姿勢 \hat{G}_i^k と逆深度 \hat{d}_i^k を用いた投影変換により算出する。これら推定値 $\{\hat{f}_{(i,j)}^k, \hat{G}_i^k, \hat{d}_i^k \mid (i, j) \in \mathcal{G}\}$ は次ステップ $k+1$ の初期値として使用され、最終ステップ K の推定値 $\{\hat{f}_{(i,j)}^K, \hat{G}_i^K, \hat{d}_i^K \mid (i, j) \in \mathcal{G}\}$ は次時刻 $t+1$ における初期値として使用される。推論時は、キーフレームを判定しながら推定処理を行った後、全キーフレームを用いたバンドル調整（全域的最適化）を行う。最後に、キーフレームではないフレームに対する推定値を線形補間を用いて算出することで、最終的な推定値 $\{\hat{G}_i, \hat{d}_i \mid 1 \leq i \leq T\}$ を算出する。以降、動画 \mathcal{I} に対する推定値 $\{\hat{G}_i, \hat{d}_i \mid 1 \leq i \leq T\}$ を vSLAM(\mathcal{I}) で表す。

訓練時は、事前に Covisibility Graph \mathcal{G} を定義した上で推定処理を行い、得られた出力 $\{\hat{f}_{(i,j)}^k, \hat{G}_i^k, \hat{d}_i^k, \hat{\Delta}_{(i,j)}^k \mid (i, j) \in \mathcal{G}, 1 \leq k \leq K\}$ と真値 $\{f_{(i,j)}, G_i, d_i \mid (i, j) \in \mathcal{G}\}$ を使用して、下式をモデルパラメータに関して最小化する。

$$\mathbb{E}_{\mathcal{I} \sim \mathcal{D}}[w_1 \mathcal{L}_{\text{pose}} + w_2 \mathcal{L}_{\text{flow}} + w_3 \mathcal{L}_{\text{residual}}]. \quad (1)$$

確率分布 \mathcal{D} は、動画 \mathcal{I} の分布を指す。係数 $w_1, w_2, w_3 \in \mathbb{R}$ はハイパーパラメータである。値 $\mathcal{L}_{\text{pose}}, \mathcal{L}_{\text{flow}}, \mathcal{L}_{\text{residual}}$ はそれぞれ自己位置損失値、オプティカルフロー損失値、差分損失値を指す。

$$\mathcal{L}_{\text{pose}} = \sum_{k=1}^K \gamma^{N-i} \sum_{i \in \mathcal{V}} \left\| \text{Log}_{\text{SE}3}(G_i^{-1} \cdot \hat{G}_i^k) \right\|_2, \quad (2)$$

$$\mathcal{L}_{\text{flow}} = \sum_{k=1}^K \gamma^{N-i} \sum_{(i,j) \in \mathcal{G}} \left\| f_{(i,j)} - \hat{f}_{(i,j)}^k \right\|_2, \quad (3)$$

$$\mathcal{L}_{\text{residual}} = \sum_{k=1}^K \gamma^{N-i} \sum_{(i,j) \in \mathcal{G}} \left\| \hat{f}_{(i,j)}^k - (\hat{f}_{(i,j)}^{(k-1)} + \hat{\Delta}_{(i,j)}^k) \right\|_1. \quad (4)$$

3.2 敵対的パッチ攻撃の定式化

一般的に、深層学習モデルに対する敵対的パッチ作成は最適化問題として定式化される。そこで、vSLAM に対する敵対的パッチの作成を最適化問題として定式化する。

パッチ \mathbf{p} が貼られた動画 \mathcal{I}_p に関する推定値 vSLAM(\mathcal{I}_p) と、パッチが貼られていない動画 \mathcal{I} に関する推定値 vSLAM(\mathcal{I}) の違いを測る関数を \mathcal{L} で表す。DROID-SLAM に対する敵対的パッチの作成を、以下の最適化問題として定式化する。

$$\tilde{\mathbf{p}} = \text{argmax}_{\mathbf{p}} \mathbb{E}_{\mathcal{I} \sim \mathcal{D}}[(\mathcal{L}(\text{vSLAM}(\mathcal{I}_p), \text{vSLAM}(\mathcal{I})))]. \quad (5)$$

攻撃者は、算出された敵対的パッチを環境内に貼付することで、その環境での vSLAM の誤作動を図る。

多くの研究において、敵対的パッチ攻撃における最適化問題の解法には Projected Gradient Descent (PGD) [14] 等の勾配を用いた手法が用いられる。しかし、DROID-SLAM を含む一般的な vSLAM の推定処理には推定値の局所最適化と全域最適化の処理が含まれるため、計算量の観点から、パッチ \mathbf{p} に関する推定値 vSLAM(\mathcal{I}_p) の勾配 $\nabla_{\mathbf{p}} \text{vSLAM}(\mathcal{I}_p)$ を計算することが困難である。そこで、動画 \mathcal{I} から連続する 7 フレーム $\mathcal{I}^{\text{sub}} \sim \mathcal{D}_{\mathcal{I}}$ を抽出し、事前に Covisibility Graph \mathcal{G} を定義した上でのフレーム集合 $\mathcal{I}_p^{\text{sub}}$ に対する推定値を扱う。フレーム集合の \mathcal{I}^{sub} に関する分布 $\mathcal{D}_{\mathcal{I}}$ は一様分布とする。バンドル調整はフレーム集合 $\mathcal{I}_p^{\text{sub}}$ 内でのみ行われるため、計算量が減った結果、推定値の勾配が算出可能になる。本処理は、フレームの抽出方法を除き、DROID-SLAM の訓練時に用いられる処理と同じである。本稿では DROID-SLAM を扱うことから、フレーム集合 $\mathcal{I}_p^{\text{sub}}$ に対する出力 $\{\hat{f}_{(i,j)}^k, \hat{G}_i^k, \hat{d}_i^k, \hat{\Delta}_{(i,j)}^k \mid (i, j) \in \mathcal{G}, 1 \leq k \leq K\}$ をパッチ作成に使用する。更新数 K に関して、本稿におけるパッチ作成においては $K = 15$ とする。以下、この出力を vSLAM^{sub}($\mathcal{I}_p^{\text{sub}}$) で表す。よって最適化問題を下式に書き直す。

$$\tilde{\mathbf{p}} = \text{argmax}_{\mathbf{p}} \mathbb{E}_{\mathcal{I} \sim \mathcal{D}}[\mathbb{E}_{\mathcal{I}^{\text{sub}} \sim \mathcal{D}_{\mathcal{I}}}[\mathcal{L}^{\text{sub}}(\text{vSLAM}^{\text{sub}}(\mathcal{I}_p^{\text{sub}}), \text{vSLAM}(\mathcal{I}^{\text{sub}}))]]. \quad (6)$$

\mathcal{L}^{sub} は、フレーム集合 \mathcal{I}^{sub} に対する推定値系列 vSLAM^{sub}($\mathcal{I}_p^{\text{sub}}$) と推定値 vSLAM(\mathcal{I}^{sub}) の違いを測る関数である。

式 (5) 及び式 (6) は、動画に関する分布 \mathcal{D} 上の期待値演算により、動画の撮影環境や撮影方法に頑健で強力な敵対的パッチの算出を目指している。この最適化問題を解くためには、様々な撮影方法で収集された多くの動画を準備する必要がある。またエンドツーエンドで深層学習に基づく vSLAM 手法の脆弱性は未知であることから、撮影環境や撮影方法に頑健な敵対的パッチ $\tilde{\mathbf{p}}$ が存在しない可能性がある。そこで我々は、一つの動画に対する敵対的パッチを作

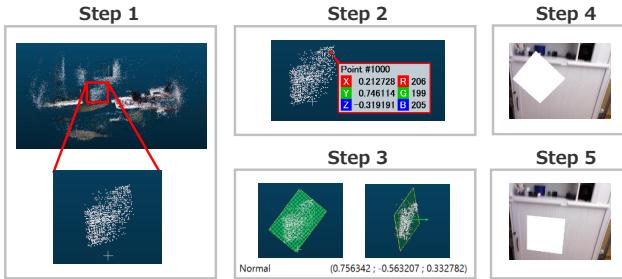


図 1 パッチ部分の世界座標を取得する処理の流れ

Fig. 1 The process flow for obtaining the world coordinates of the patch.

成し、作成されたパッチが貼付された動画に対する結果を観察することで、vSLAM の脆弱性を調査する。これにより、パッチの汎化性は失われるものの、最適化問題は単純化されるため、各動画に対する敵対的パッチの作成は容易になることが期待される。以上を踏まえ、本稿では下式の最適化問題を解くことで各動画に対する敵対的パッチを作成する。

$$\tilde{p} = \operatorname{argmax}_{\mathbf{p}} \mathbb{E}_{\mathcal{I}^{\text{sub}} \sim \mathcal{I}} [\mathcal{L}^{\text{sub}}(\text{vSLAM}^{\text{sub}}(\mathcal{I}_p^{\text{sub}}), \text{vSLAM}(\mathcal{I}^{\text{sub}}))]. \quad (7)$$

3.3 パッチの貼付方法

敵対的パッチを作成する際及び敵対的パッチ攻撃に対する脆弱性を評価する際、パッチが貼付された動画を作成しなければならない。本稿では、動画を撮影したカメラの内部パラメータは既知とする。各フレームにおけるパッチの画像座標の算出には、カメラ位置姿勢とパッチの貼付部分の世界座標を事前に知る必要がある。そこで、事前にパッチが貼付されていない動画を DROID-SLAM で処理することで、各フレームのカメラ位置姿勢と地図を入手することとする。これらの推定値を用いて、図 1 の流れで各フレームにおけるパッチ部分の世界座標を算出する。本処理に含まれる点群処理は CloudCompare [15] を用いて行っている。

- (1) 地図内のパッチの貼付部分付近の点群を切り抜く。
- (2) パッチの右上に相当する位置の世界座標を取得する。
- (3) 切り抜いた点群を平面近似し、近似平面の法線を算出する。本ベクトルはパッチの表向きの法線である。
- (4) 世界座標系において、ステップ 2 で求めた座標値を原点としてパッチの表向き法線を z 軸としたときの xy 平面上で、区間 $0 \leq x < l_{\text{width}}, 0 \leq y < l_{\text{height}}$ でグリッド点を定義する。値 $l_{\text{width}}, l_{\text{height}}$ は世界座標系におけるパッチの幅、高さであり、事前に定める。グリッド点を各フレームへ投影する。
- (5) グリッド点投影後の各フレームを確認しながら、意図した角度になるように前ステップのグリッド点をパッチの法線回りで回転させる。



図 2 低解像度のグリッドで定義されたパッチ（白色）が投影された画像例。

Fig. 2 An Example of image where a patch defined by a low-resolution grid (in white) was projected.



図 3 高解像度のグリッドで定義された（一様ランダムな色）パッチが投影された画像例。

Fig. 3 Examples of images where a patch defined by a high-resolution grid (in uniformly random colors) was projected.

本処理を経て得られたグリッド点の世界座標を、パッチ部分の世界座標とする。パッチ部分に対応する画像座標の算出は、パッチ部分の世界座標を、推定カメラ位置姿勢とカメラ内部パラメータを用いて画像面へ投影することで行われる。ただし各フレームに対して、法線がカメラと逆方向を向いている場合はパッチは画像に投影させていない。

3.4 パッチに対する前処理

世界座標系でパッチを定義するグリッド解像度と動画の各フレームの解像度の関係によっては、二つの問題が生じる。本節では、この現象について述べた上で、対策法について説明する。

パッチのグリッド解像度が画像の解像度に対して過度に低い場合、パッチが貼られるべき部分に複数の小さな穴が生じる場合がある。図 2 は、白色のグリッド点を画像に投影した例であり、パッチ部分が白色で埋め尽くされていないことが確認できる。一方でパッチのグリッド解像度が画像の解像度に対して過度に高い場合、視点によってパッチの色が大きく変化する。図 3 は、各グリッド点の色を一様ランダムな値で決定した上で、視点が異なる二枚の画像に投影した例である。二枚の画像に貼られたパッチは本来同じであるにもかかわらず、画像間でパッチの画素値が大きく異なっていることが確認できる。

我々は、パッチに前処理を施することで、これらの現象を軽減させる。まず、画像の解像度に対して低解像度のグリッドで定義されるパッチを作成する。本稿では画像の解像度 240×320 に対して、パッチのグリッド解像度を 100×100



図 4 双線形補間を含む前処理を施したパッチが投影された画像例.
Fig. 4 Examples of images where a patch preprocessed by some methods including bilinear interpolation was projected.

とする。その後、このパッチのグリッド解像度を、双線形補間によって画像の二倍の解像度にリサイズする。図 4 は、各グリッド点の色を一様ランダムな値で決定した上で本前処理を適用したパッチを画像に投影した例である。図 2 の現象を抑え、図 3 の現象も軽減されていることが確認できる。以降では、パッチを画像に貼付する際、この双線形補間によるリサイズを適用した上で画像に投影している。

3.5 パッチ作成に向けた目的関数

式(7)の目的関数 \mathcal{L}^{sub} の設計は、敵対的パッチ攻撃の強度に大きく影響することが予想される。本稿では、目的関数 \mathcal{L}^{sub} として以下の 3 つの関数を用いて、各目的関数による攻撃を扱う。

$$\mathcal{L}_{\text{train-based}} = w_1 \mathcal{L}'_{\text{pose}} + w_2 \mathcal{L}'_{\text{flow}} + w_3 \mathcal{L}'_{\text{residual}}, \quad (8)$$

$$\mathcal{L}_{\text{position-based}} = \sum_{k=1}^K \sum_{i \in \mathcal{V}} \|\hat{q}_i^k - \tilde{q}_i^k\|_2, \quad (9)$$

$$\mathcal{L}_{\text{flow-based}} = - \sum_{k=1}^K \sum_{(i,j) \in \mathcal{G}} \text{CosSim}(\hat{f}_{(i,j)}^k, \tilde{f}_{(i,j)}^k). \quad (10)$$

値 $\mathcal{L}'_{\text{pose}}, \mathcal{L}'_{\text{flow}}, \mathcal{L}'_{\text{residual}}$ はそれぞれ、式(2), (3), (4)において、出力 $\{\hat{f}_{(i,j)}^k, \hat{G}_i^k, \hat{d}_i^k, \hat{\Delta}_{(i,j)}^k \mid (i,j) \in \mathcal{G}, 1 \leq k \leq K\}$ をパッチが貼られた動画に対する推定値 $\text{vSLAM}^{\text{sub}}(\mathcal{I}_p^{\text{sub}})$ で置き換え、真値 $\{f_{(i,j)}, G_i, d_i \mid (i,j) \in \mathcal{G}\}$ を推定値 $\text{vSLAM}(\mathcal{I})$ と、当推定値から計算されるオプティカルフロー推定値 $\{f_{(i,j)} \mid (i,j) \in \mathcal{G}\}$ で置き換えて計算される値である。式(9)は従来の VO モデルに対する攻撃 [5] が使用する目的関数と同じであり、式(10)は従来のオプティカルフロー推定器に対する攻撃 [16] が使用する目的関数である。以降、式(8), (9), (10)による攻撃をそれぞれ訓練損失ベースの攻撃、位置ベースの攻撃、オプティカルフローベースの攻撃と呼称する。

各目的関数で定義される式(7)の最適化は PGD によって解くこととする。PGD は下式によるパッチ更新を繰り返す。

$$\mathbf{p}^{l+1} = \text{Proj}_{[0,1]}(\mathbf{p}^l + \alpha \cdot \text{sgn}(\nabla_{\mathbf{p}^l} g(\mathbf{p}^l))). \quad (11)$$

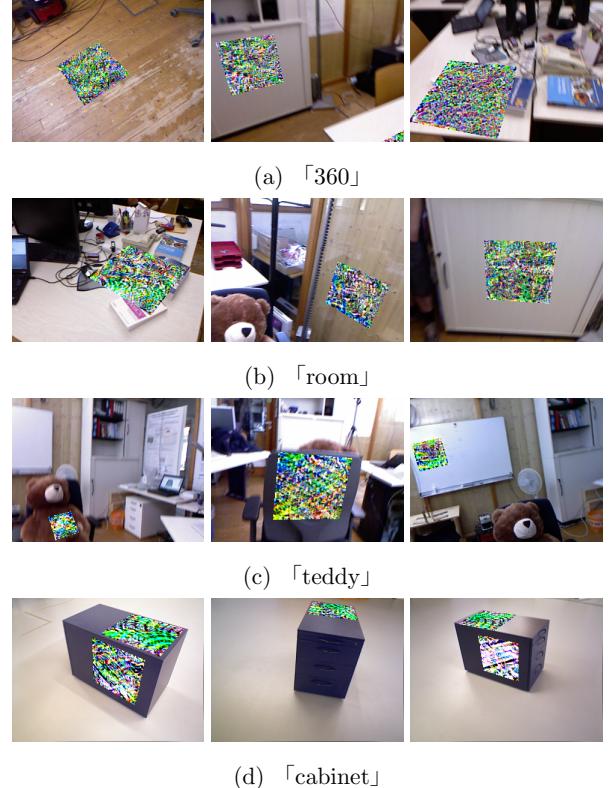


図 5 各動画において、敵対的パッチが投影されたフレームの例.
Fig. 5 Examples of frames from each video where adversarial patches were projected.

関数 $\text{Proj}_{[0,1]}$ は引数の要素が区間 $[0, 1]$ を超えた場合に区間 $[0, 1]$ へ投影する関数であり、値 l, α はそれぞれイテレーション数とステップサイズである。式(11)は、事前に定めた最大イテレーション数 L 繰り返される。本稿では、 $\alpha = 1/255, L = 10000$ とする。

4. 実験

本節では、前節で提案した攻撃手法を DROID-SLAM に適用することで、DROID-SLAM の脆弱性を調査する。DROID-SLAM の重みは、著者が提供する TartanAir データセット [18] で訓練済みの重みを用いる。

4.1 データセット

本節では、TUM RGB-D データセット [17] に含まれる freiburg1_360 (360), freiburg1_room (room), freiburg1_teddy (teddy), freiburg3_cabinet (cabinet) を使用する。

4.2 評価方法

各動画に対して三つのパッチを貼ることによる敵対的パッチ攻撃を行う。三つのパッチ作成は、前述した式(8), (9), (10)を目的関数とした最適化問題（式(7)）を各動画で三つのパッチに関して同時に解くことで行われる。作成された敵対的パッチを貼付したフレームの例を図 5 に示

す。以降、三つの最適化による攻撃結果を示すが、いずれの場合も、パッチの大きさと位置は図 5 が示す場合と同様である。これら攻撃に対する脆弱性の評価は、攻撃前後での自己位置推定と地図作成の結果を可視化し、両者を比較することで行う。

4.3 結果

まず、提案した敵対的パッチ攻撃が自己位置推定へ及ぼす影響を観察する。攻撃前後の動画に対する自己位置推定結果を図 6 に示す。動画「360」に対する訓練損失ベースの攻撃を除いて、推定結果に大きな影響を与えることなかった。本結果に対して、二つの要因が考えられる。一つ目は、DROID-SLAM に含まれる RAFT がオプティカルフローの誤推定に頑健な構造を持っていることである。これは、Schrodi らの研究 [19] でも述べられており、RAFT のエンコーダの広い受容野や、GRU による推定誤差の再帰的な修正が RAFT の頑健性を向上させていると考えられる。DROID-SLAM は RAFT によるオプティカルフロー推定値を用いて自己位置推定と地図作成を行うため、RAFT の頑健性が DROID-SLAM の頑健性に大きく寄与した結果、自己位置推定に大きな影響を与えることなかったと考えられる。二つ目は、本稿での敵対的パッチ作成時に使用した最適化問題（式 (7)）と、実際に解くべき最適化問題（式 (5)）の違いが大きい点である。式 (7) におけるパッチが貼付された動画に対する推定値は、特定の 7 フレームを用いる局所最適化のみを経て算出される。しかし、実際に解くべき最適化問題（式 (5)）では、推定値は Covisibility Graph に従って複数回の局所最適化と全域最適化を経て算出される。そのため、式 (7) における推定値 $v\text{SLAM}^{\text{sub}}(\mathcal{I}_p^{\text{sub}})$ に大きな影響を与える敵対的パッチが作成できた場合でも、式 (5) における実際の推定値 $v\text{SLAM}(\mathcal{I}_p)$ にはほとんど影響が与えられていない可能性がある。

次に、敵対的パッチ攻撃が地図作成へ及ぼす影響を観察する。攻撃前後の動画に対する地図作成結果を図 7 に示す。攻撃による大きな影響は確認されなかった。しかし、パッチを貼付した部分の作成地図に関しては攻撃の影響が確認された。図 8 は、図 7 の地図作成結果に関して、各動画でパッチが貼られた三箇所の内の一つを拡大した図である。いずれの動画に対しても、訓練損失ベースと位置ベースの攻撃に対する結果には、パッチ部分の点群が推定できている一方で、オプティカルフローベースの攻撃の場合は、パッチが貼付された部分の点群が欠落していることが確認できる。本結果は、攻撃者がオプティカルフローベースの攻撃によって作成されたパッチを特定の物体に貼付することで、 $v\text{SLAM}$ が作成する地図上からその物体を欠落させられる可能性があることを示唆している。オプティカルフローベースの攻撃のみ地図作成に影響を与えた要因として、本報告で扱った DROID-SLAM の

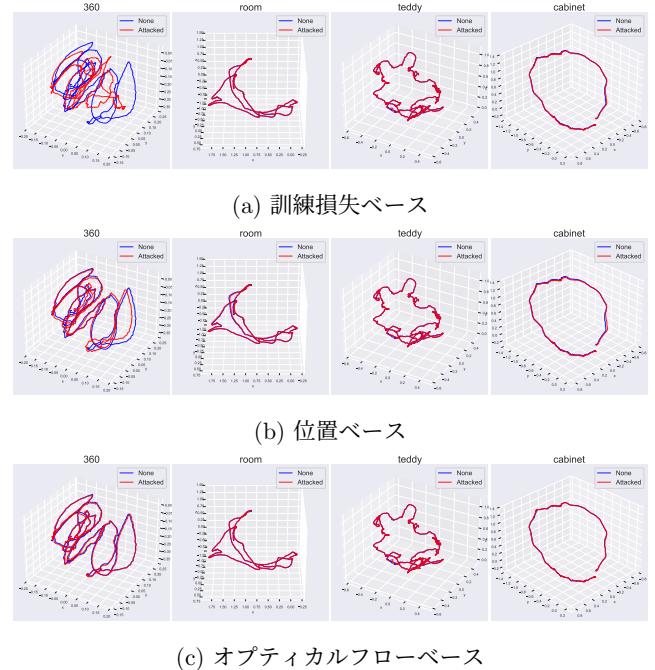


図 6 DROID-SLAM による自己位置推定結果（青：攻撃前、赤：攻撃後）。

Fig. 6 Localization results by DROID-SLAM (blue: before attack, red: after attack).

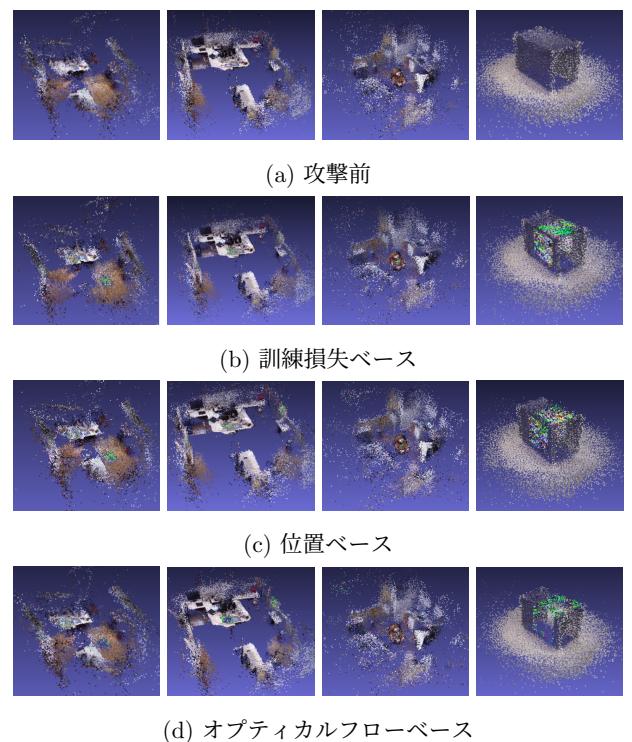


図 7 DROID-SLAM による地図作成結果。左から順に「360」、「room」、「teddy」、「cabinet」に対する結果を示す。

Fig. 7 Mapping results by DROID-SLAM. From left to right, the results are for “360”, “room”, “teddy”, and “cabinet”.

構造が挙げられる。DROID-SLAM は内部でオプティカルフローを推定しているため、オプティカルフローの誤推定

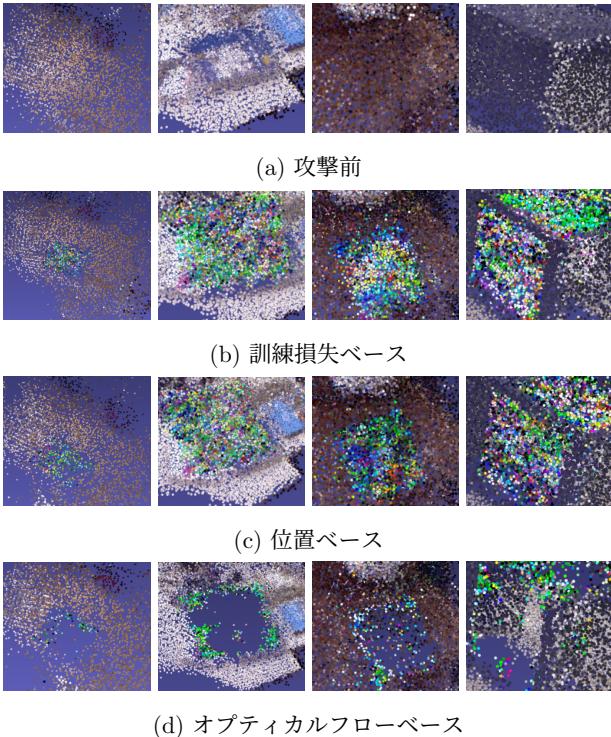


図 8 DROID-SLAM による作成地図に対するパッチ貼付部分の拡大図。

Fig. 8 The zoomed-in views of the patched areas from the mapping results by DROID-SLAM.

定を直接的に促す攻撃のみ、一定の攻撃力を発揮したと考えられる。これを確かめるために、DROID-SLAM が推定するオプティカルフローを可視化する。以下では、動画 \mathcal{I} に対する DROID-SLAM の推定値として、式(7)と同様に $v\text{SLAM}^{\text{sub}}(\mathcal{I}^{\text{sub}}) = \{\hat{f}_{(i,j)}^k, \hat{G}_i^k, \hat{d}_i^k, \hat{\Delta}_{(i,j)}^k \mid (i,j) \in \mathcal{G}, 1 \leq k \leq 15\}$ を用いる。図 9 は敵対的パッチが貼付された「360」の特定の二つのフレーム I_i, I_j に関して再帰的に推定されたオプティカルフロー推定値の内、推定値 $\hat{f}_{(i,j)}^1, \hat{f}_{(i,j)}^5, \hat{f}_{(i,j)}^{10}, \hat{f}_{(i,j)}^{15}$ を可視化している。オプティカルフローベースの攻撃のみ、推定値に大きな影響を与えられていることが確認できる。以上の結果から、学習損失ベースや位置ベースの攻撃はオプティカルフロー推定に大きな影響を与えられなかった結果、地図作成に対してもほとんど影響を与えられなかったと考えられる。

本稿では DROID-SLAM のみを扱ったが、オプティカルフロー推定を含む他の vSLAM [12] に対しても類似する結果が予想される。一方で、提案した攻撃手法は NeRF ベースの vSLAM [10] や 3D Gaussian Splatting ベースの vSLAM [11] に対して適用できない課題がある。

5. おわりに

本稿では、エンドツーエンドで深層学習に基づく vSLAM の敵対的パッチ攻撃に対する頑健性を分析した。エンドツーエンドで深層学習に基づく vSLAM の敵対的パッチ攻撃に関する既存研究が存在しないことから、攻撃手法を提

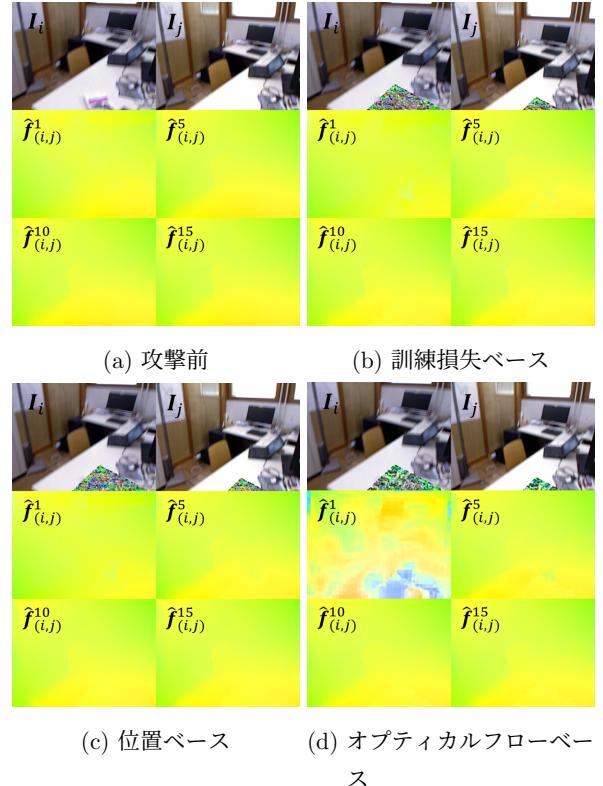


図 9 DROID-SLAM による二枚の画像間のオプティカルフローの推定結果。

Fig. 9 The estimation results of optical flow between two images by DROID-SLAM.

案した。実験では、攻撃対象として DROID-SLAM を扱い、オプティカルフローベースの攻撃のみ、パッチ部分の地図を欠落させる傾向が見られた。しかし、位置姿勢の誤推定や地図作成の大きな乱れは確認されなかった。今後の課題として、提案攻撃は自己位置推定に対してほとんど影響を与えない点が挙げられる。また、提案攻撃は NeRF ベースや 3D Gaussian Splatting ベースの vSLAM に直接適用できない点も課題である。

参考文献

- [1] Cheng, J., Zhang, L., Chen, Q., Hu, X., and Cai, J. A review of visual slam methods for autonomous driving vehicles. Eng. Appl. Artif. Intell., vol. 114, no. 104992, 2022.
- [2] Brown, T. B., Mané, D., Roy, A., Abadi, M., and Glimm, J. Adversarial patch. arXiv preprint arXiv:1712.09665, 2017.
- [3] Liu, X., Yang, H., Liu, Z., Song, L., Li, H., and Chen, Y. Dpatch: an adversarial patch attack on object detectors. In AAAI workshop, 2019.
- [4] Mokssit, S., Licea, D. B., Guermah, B., and Ghogho, M. Deep learning techniques for visual slam: A survey. IEEE Access, vol. 11, pp. 20026-20050, 2023.
- [5] Nemcovsky, Y., Jacoby, M., Bronstein, A. M., and Baskin, C. Physical passive patch adversarial attacks on visual odometry systems. In ACCV, pp. 1795-1811, 2022.
- [6] Ikram, M. H., Khaliq, S., Anjum, M. L., and Hussain, W. Perceptual aliasing++: Adversarial attack for visual

- slam front-end and back-end, IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 4670-4677, 2022.
- [7] Mur-Artal, R., Montiel, J. M. N., and Tardós J. D. Orb-slam: A versatile and accurate monocular slam system. IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, 2015.
 - [8] Bruno, H. M. S., and Colombini, E. L. Lift-slam: A deep-learning feature-based monocular visual slam method. Neurocomputing, vol. 455, pp. 97-110, 2021.
 - [9] Teed, Z., and Deng, J. Droid slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In NeurIPS, vol. 34, pp. 16558-16569, 2021.
 - [10] Zhu, Z., Peng, S., Larsson, V., Cui, Z., Oswald, M. R., Geiger, A., and Pollefeys, M. Nicer-slam: Neural implicit scene encoding for rgb slam. In 3DV, pp. 42-52, 2024.
 - [11] Matsuki, H., Murai, R., Kelly, P. H. J., and Davison, A. J. Gaussian splatting slam. In CVPR, pp. 18039-18048, 2024.
 - [12] Lipson, L., Teed, Z., and Deng, J. Deep patch visual slam. In ECCV, pp. 424-440, 2024.
 - [13] Teed, Z., and Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In ECCV, pp. 402-419, 2020.
 - [14] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In ICLR, 2018.
 - [15] <https://www.cloudcompare.org>
 - [16] Ranjan, A., Janai, J., Geiger, A., and Black, M. J. Attacking optical flow. In ICCV, pp. 2404-2413, 2019.
 - [17] Strum, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. A benchmark for the evaluation of rgb-d slam systems. In IROS, pp. 573-580. 2012.
 - [18] Wang, W., Zhu D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., and Scherer, S. Tartanair: A dataset to push the limits of visual slam. In IROS pp. 4909-4916, 2020.
 - [19] Schrodi, S., Saikia, T., and Brox, T. Towards understanding adversarial robustness of optical flow networks. In CVPR, pp. 8916-8924, 2022.