Vol.2021-NL-248 No.19
Vol.2021-CVIM-226 No.19
2021/5/21

複数の補助情報を用いた画像修復精度の向上

山下 陽平[1,a)]　下里 航大[1,b)]　浮田 宗伯[1,c)]

概要：画像修復とは，画像から削除したい領域（マスク領域）を指定し，その領域を周辺領域の見え方に基づいて補間する技術である．近年，画像修復の補助となる画像（補助画像）のマスク領域を修復し，修復された補助画像をヒントにカラー画像を修復する 2 段階の手法が注目されている．これは補助画像がカラー画像よりも修復しやすく，修復された補助画像はマスク領域内の重要な情報を持つことが期待され，画像修復に有効であるためである．これまでの補助画像には画素値の変化が大きな点を抽出したエッジ画像や，物体の領域ごとに分割し，それぞれが何であるかを示したセグメンテーション画像が用いられた．しかし，従来の補助画像では，物体の奥行方向の前後関係を把握できないために，ありえない修復が生じていた．また，従来の手法では補助画像を 1 つしか用いていないため，その画像の作成，修復を精度よくできなければ，不自然な領域が生じてしまう．そこで，エッジ画像に加えて奥行方向の情報である深度画像も補助画像に加えることで修復精度の向上を図る手法を提案する．エッジ画像は画素値が似た物体が重なった領域では機能しない．一方，深度画像は距離情報により画素値が変わるため，エッジ画像の欠点を補うことができる．また，2 つの補助画像を相補的に用いるため，対象領域のコンテクスト情報から，その領域における補助画像の重みを変化させる手法を導入した．これにより，実際の画像が図 1（1）であるマスク付き画像図 1（2）を修復すると，従来手法では図 1（3）のように木と背景が混ざった不鮮明な領域が出現するが，本手法では図 1（4）のように違和感なく鮮明な画像修復を可能にした．また，定性的評価は，平均で PSNR が 0.70，SSIM が 0.01 向上し，平均絶対誤差が 0.20 減少した．

キーワード：画像修復，深度推定，エッジ画像，深度画像

1. はじめに

画像修復とは，画像から削除したい領域（マスク領域）を指定し，その領域を周辺領域の見え方に基づいて補間する技術である．例えば，観光地で写真を撮影するとき，周囲にいる観光客が映り込んでしまうことがある．そこで，画像修復を利用すると，観光客がいなかったかのような画像を生成できる．

近年の画像修復は，深層学習の手法である畳み込みニューラルネットワーク（Convolutional Neural Network；CNN）を使用することで性能が飛躍的に向上している．CNN では，マスク領域周辺の画像の特徴（建物，山など）を抽出し，それを踏まえてマスク領域を修復する．したがって，マスク領域に何が写っているか推定し，周囲の画素値を用いて補間することで，違和感の少ない修復画像を生成できるようになった．しかし，カラー画像のみからマスク領域の画素値を推定，修復するには限界があった．そこで，補助画像が利用されるようになった．

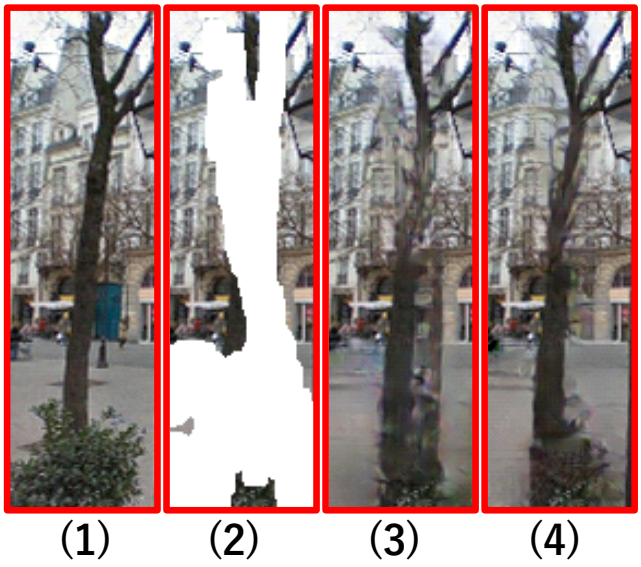

図 1　（1）理想画像，（2）マスク付きカラー画像，（3）従来手法を用いた修復結果，（4）提案手法を用いた画像修復結果．

[1] 豊田工業大学
Toyota Technological Institute
[a)] sd21446@toyota-ti.ac.jp
[b)] sd20425@toyota-ti.ac.jp
[c)] ukita@totota-ti.ac.jp

Vol.2021-NL-248 No.19
Vol.2021-CVIM-226 No.19
2021/5/21

図 2 補助画像一覧. 左上: カラー画像, 右上：エッジ画像, 左下：セグメンテーション画像, 右下：深度画像

従来の画像修復に用いられる補助画像とは，カラー画像（図 2 左上）に対する図 2 右上，左下のような画像である．エッジ画像（図 2 右上）は，隣接する画素値の変化が大きな点を抽出した 2 値画像である．セグメンテーション画像（図 2 左下）は，カラー画像を物体ごと（人，建物など）に分割した画像である．このような補助画像中のマスク領域も消去されるが，補助画像は線や領域内同色で表現されるため，カラー画像に比べて情報量が少なく，修復が容易となる．したがって，補助画像は精度よく修復され，マスク領域の有効なエッジ情報やセグメンテーション情報を持つことが期待される．そこで，2 段階のネットワークを用いて，第 1 段階で補助画像を修復し，第 2 段階で修復した補助画像をもとにカラー画像を修復する手法が提案された．

その手法の 1 つに，EdgeConnect [1] がある．EdgeConnect では，第 1 段階でエッジ画像の作成，修復を行い，修復したエッジ画像をもとに第 2 段階でカラー画像を修復した．しかし，エッジ画像では物体の奥行方向の前後関係を把握できないため，前景と背景の物体が混在する不自然な修復が生じることがある．

そこで，本研究では物体のカメラからの距離を示した深度画像（図 2 右下）を用いることで，従来の補助画像では把握できない物体の奥行方向の前後関係を付与し，マスク領域のより多くの情報を与えることで，より精度の良い画像修復を行う．深度画像は従来の補助画像と比較して作成が難しく，これまで補助画像としてあまり用いられなかった．しかし，深層学習の発達により，深度画像推定精度が向上したため，本研究では深度画像も補助画像として画像修復に利用する．さらに，複数の補助画像の利点を反映させるため，Gated Convolution [2] を用いて，対象領域のコンテクスト情報から，その領域における補助画像の重みを変化させることで，エッジ画像と深度画像のうち画像修復に貢献できる方を強く反映し，修復精度を向上させた．

2. 関連研究

本研究に関連する研究を紹介する．2.1 節で画像修復の手法について述べ，2.2 節で深度画像の作成方法について述べる．

2.1 画像修復

まず，画像修復の基盤となる手法について説明し，深層学習を使用した手法，補助画像を用いた近年の手法，そしてあらゆるマスク形状に対応した手法について述べる．

2.1.1 伝統的な画像修復手法

画像修復の手法は主に拡散ベースとパッチベースの 2 つがある．拡散ベース [3], [4] では，マスク領域近傍の画素値をマスク領域内に伝播することで修復する．しかし，この手法では局所的な情報しか反映できず，画像全体として違和感のある修復となる．

一方，パッチベース [5], [6] では，画像全体を小さな矩形（パッチ）に分割し，マスク領域内の値としてふさわしいパッチを同一画像中の全パッチから探索，コピーすることでマスク領域を埋める．しかし，パッチをコピーするだけではピクセル単位の細かい修復ができず，精度に限界がある．

2.1.2 深層学習を用いた画像修復の初期手法

学習ベースの多くはエンコーダデコーダ構造を使用する．画像修復では，背景画像を用意し，それにマスクをかけた画像をエンコーダに入力する．そして，デコーダから出力された画像と元の背景画像の誤差が最小となるように学習される．

Context Encoder [7] では修復器にエンコーダデコーダ構造を使用し，敵対的生成ネットワーク（Generative Adversarial Network；GAN） [8] を用いて敵対的学習をさせる．GAN とは生成器と識別器の 2 つのネットワークを用いる生成ネットワークである．画像修復においては生成器=修復器であり，マスクをかけた画像から修復した画像を出力する．識別器は修復した画像と元の背景画像を受け取り，背景画像か修復画像かを識別する．識別器は修復画像を識別できるように学習し，修復器は修復した画像が識別器に識別されないように学習する．こうして，修復器が本物の画像に似た修復画像を生成できるようになる．しかし，この手法では，修復領域とマスクの周辺領域との整合性が考慮されず，見た目のひずみやボケを多く含んだ画像が生成されてしまう．

飯塚ら [9] は Context Encoder をベースに，識別器で本物と修復画像の画像全体を識別するだけでなく，それぞれのパッチも入力し，局所的な領域の識別もすることで修復したマスク領域と周辺領域との整合性を保てるようにした．しかし，この手法では学習に時間がかかり，さらに後

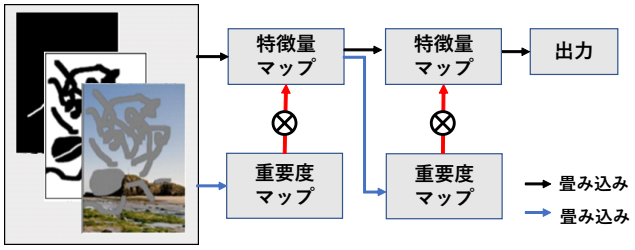

図 3 Gated Convolution [2] の説明．マスク，カラー画像，修復領域の補助線を描いた画像を入力する．青線の畳み込みで重要度マップを作成し，黒線の畳み込みで作成した特徴量マップに掛け合わせて重みづけを行うことで補助線を修復に反映可能となる．

処理として画像合成が必要だった．そこで，Yu ら [10] は，2 つの修復器を用いて，1 つ目でマスク領域を粗く修復し，2 つ目で細部を修復する手法を提案した．しかし，これらの手法では，マスク領域をカラー画像に直接修復するため，修復が難しく，ボケた画像が作成されてしまう．

2.1.3 補助画像を利用した画像修復

Yu ら [10] の手法が提案された後，第 1 段階で画像修復のヒントとなる補助画像のマスク領域を先に修復し，それをもとに第 2 段階でカラー画像を修復する手法が多く提案された．EdgeConnect [1] ではエッジ画像が利用される．エッジ画像は白と黒の線で表示されるため，マスク領域の修復は簡単だが，複雑な画像ではエッジが多く，必要な輪郭を修復することが困難である．一方，Song ら [11] は，セグメンテーション画像を用いた．セグメンテーション画像では同じ物体の領域が同色で表されるため，異なる物体間の境界を推定することに長けているが，物体領域の内部の情報がない．また，これらの手法は，前後関係を把握できていないため，前景と背景が隣接した場所にマスクがかかると，違和感が生じることがある．

2.1.4 任意のマスク形状に対応した画像修復

従来のほとんどの手法は決まった形状のマスクを修復することは可能であったが，様々な形状のマスクを修復するのは困難であった．そこで，畳み込みを工夫することでこれらを改善する手法が提案された．

Partial Convolution [12] では，畳み込みをする際に，マスクをかけたカラー画像とマスクを与えることで，常にマスク領域を考慮しながら画像修復することが可能となり，多用な形状のマスクに対応できるようにした．しかし，Partial Convolution では，追加の入力がマスクだけであり，補助画像などの他の情報を反映させることができなかった．

そこで，Gated Convolution [2] では，2 つの畳み込みを用いて，対象領域における画像の重みづけの仕方も学習させることで，その領域のマスクを埋めるために，どの補助画像が重要であるかを判断し，マスク以外の入力情報を反映できるようにした．実際の Gated Convolution の構造を図 3 に示す．図 3 の黒線の畳み込みからは，入力画像の特徴を表す特徴量マップが作成される．一方，青線の畳み込みからは特徴量マップのどこが重要かを示す重要度マップが作成される．これを特徴量マップに掛け合わせることで，対象領域の重要な情報が強調され，重みづけが可能となる．Gated Convolution に複数の修復補助画像を入力することで，それぞれの利点を反映することができるため，本研究に使用する．

2.2 深度画像生成

深度画像は測定機械を用いて距離を測定して作成される．代表的な深度測定機械に Light Detection And Ranging（LiDAR）がある．LiDAR では，レーザー光を照射し，物体に当たって跳ね返ってくるまでの時間を測定して，その時間に応じて物体までの距離を計算する．こうして求められた距離情報を画像の値に変換することで深度画像を作成する．しかし，LiDAR は高価であるのに加えて，事前に撮影されたカラー画像の深度を計算することができない．そこで，事前に計測して作成されたカラー画像と深度画像がペアとなったデータセットを用いて，深度の推定方法を学習させることで，カラー画像のみから深度画像を推定する深度推定の手法が提案された [13] しかし，こられの手法は精度が粗く，物体の境界で大きな誤推定が生じていた．

一方，Dense Depth [14] では，深度推定に適した特徴を抽出する学習済みのエンコーダのフィルタを初期値としたエンコーダデコーダ構造を用いる．学習済みのモデルを使用することで，最初から重要な特徴を抽出できるようになり，短時間で高性能な推定を行えるようになる．したがって本研究で使用する．

3. 従来手法

本研究の従来手法である EdgeConnect [1] について学習時と修復時に分けて説明する．この手法では，エッジ画像を補助画像として用いることで画像修復精度を向上させた．

3.1 学習時

EdgeConnect では 2 段階の学習を行う（図 4 緑枠）．第 1 段階（図 4 黒線）では，まず，カラー画像から Canny 法 [15] を用いてエッジ画像を作成する．Canny 法では，画像の縦方向と横方向に微分をして画素値の勾配を求め，画素値が勾配方向に対して極大であるものだけを抽出することでエッジ画像が作られる．次にエッジ画像にマスクをかけたマスク付きエッジ画像とマスクをエッジ修復器に入力する．エッジ修復器は図 5 のような構造となっており，エンコーダで入力された画像に畳み込みをして低次元化し，それを基にデコーダで逆畳み込みをして 1 枚の画像が出力される．ここで，修復器は GAN [8] を用いて，敵対的学習を行う．学習に用いる式は以下である．

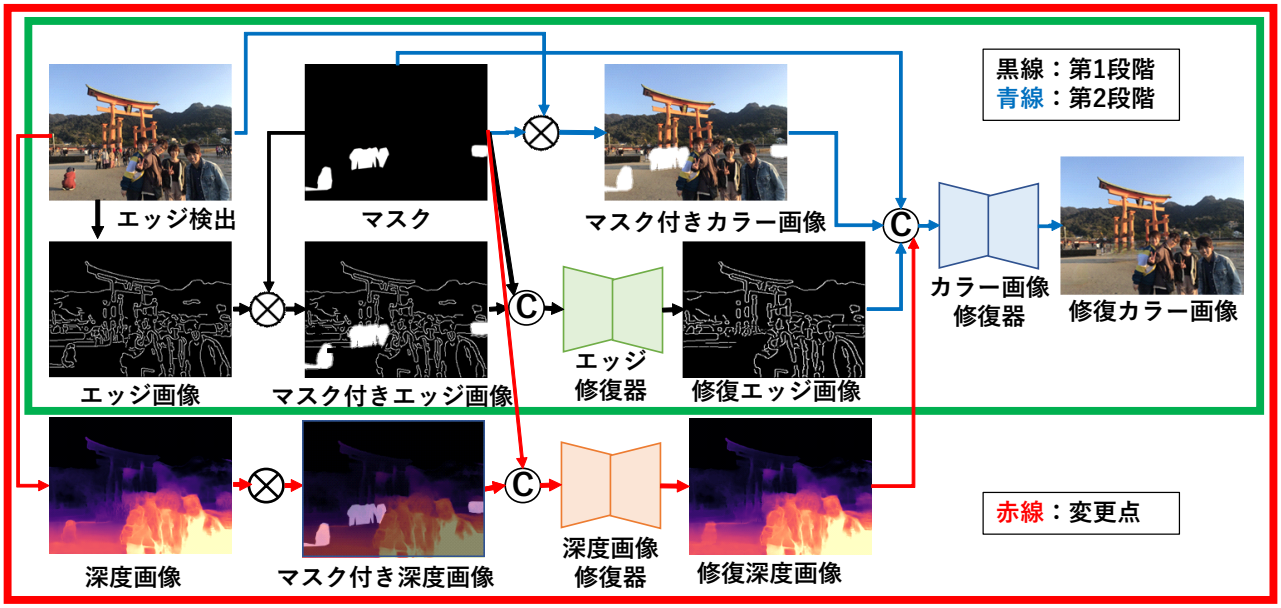

図 4 手法の説明．緑枠が EdgeConnect [1]，赤枠が提案手法．EdgeConnect では，カラー画像からエッジ画像を作成し，これにマスクをかけたマスク付きエッジ画像とマスクをエッジ修復器に入力してエッジを修復する．そして，作成された修復エッジ画像とマスク付きカラー画像，マスクをカラー画像修復器に入力し，カラー画像を修復する．一方，提案手法では，エッジとともに深度画像を作成してマスクをかけ，それぞれのマスク領域を修復器で修復する．そして修復された修復エッジ画像，修復深度画像をマスク付きカラー画像，マスクとともにカラー修復器に入力し，カラー画像の修復を行う．

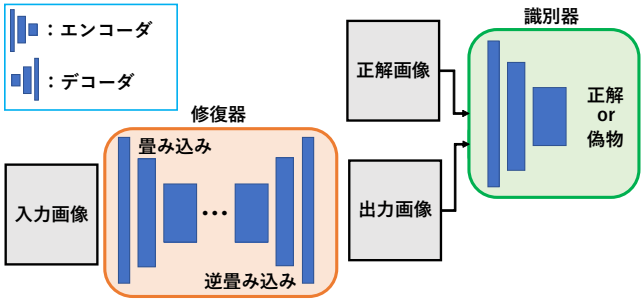

図 5 修復器内部と識別器．修復器では，エンコーダで入力画像に畳み込みをして低次元化し，これをもとにデコーダで逆畳み込みをすることで出力画像を作成する．そして，出力画像と正解画像を識別器に入力し，GAN を用いて修復器，識別器を学習させ，正解画像に近い出力画像が生成される．

$$\mathcal{L}_{\mathrm{G}} = min_{\mathrm{G}}\left(\lambda_{\mathrm{adv}} max_{\mathrm{D}}(\mathcal{L}_{\mathrm{adv}}) + \lambda_{\mathrm{FM}}\mathcal{L}_{\mathrm{FM}}\right) \qquad (1)$$

ただし，$\lambda_{\mathrm{adv}}, \lambda_{\mathrm{FM}}$ は 2 つの損失の値を調整するための正則化定数である．$\mathcal{L}_{\mathrm{adv}}$,$\mathcal{L}_{\mathrm{FM}}$ はそれぞれ敵対的損失，特徴量一致損失 [16] である．敵対的損失は修復器が出力画像と Canny 法で作成したエッジ画像を識別器に見分けられないように，逆に識別器は出力画像と Canny 法で作成したエッジ画像を見分けられるように学習するための損失である．特徴量一致損失は，識別器に入力された正解画像と出力画像に畳み込みをして，特徴量マップにした状態で一致させるための損失関数である．

第 2 段階（図 4 青線）では，第 1 段階で作成した修復エッジ画像を用いてマスク付き背景画像を修復する．まず，カラー画像修復器に修復エッジ画像とマスク付き背景画像，マスクを入力する．修復エッジ画像は情報量が少なく，修復が比較的容易であるため，精度よく修復されることが期待され，自然なカラー画像修復の有効な補助となる．カラー画像修復器もエッジ修復器と同じ図 5 のような構造であるが，損失関数は以下のものを使用する．

$$\mathcal{L}_{\mathrm{G}} = \lambda_{\mathrm{l1}}\mathcal{L}_{\mathrm{L1}} + \lambda_{\mathrm{adv}}\mathcal{L}_{\mathrm{adv}} + \lambda_{\mathrm{per}}\mathcal{L}_{\mathrm{per}} + \lambda_{\mathrm{sty}}\mathcal{L}_{\mathrm{sty}} \qquad (2)$$

$\lambda_{\mathrm{l1}}, \lambda_{\mathrm{adv}}, \lambda_{\mathrm{per}}, \lambda_{\mathrm{sty}}$ は各損失の桁を調整するための正則化定数である．$\mathcal{L}_{\mathrm{L1}}, \mathcal{L}_{\mathrm{adv}}, \mathcal{L}_{\mathrm{per}}, \mathcal{L}_{\mathrm{sty}}$ はそれぞれ平均絶対損失，敵対的損失，知覚的損失 [17]，スタイル損失 [18] である．平均絶対損失は出力画像を正解画像に近づけるための損失関数である．敵対的損失はエッジ修復器で用いたものと同様の理由で使用するが，カラー画像修復器では，出力画像を背景画像と見分けられないように修復するため，識別器の入力には出力画像と背景画像を用いる．知覚的損失は，出力画像と正解画像のピクセルごとの特徴量を一致させるための損失関数である．スタイル損失は，出力画像と正解画像のグラム行列の差分を計算する損失であり，通常の MSE 損失よりも知覚的によい画像が生成できることが知られている．

3.2 修復時

修復時は，「画像修復をしたいカラー画像」と「削除領域

を指定したマスク」を用意する．これらのカラー画像とマスクを EdgeConnect に入力すると，第 1 段階，第 2 段階が順に実行され，入力したカラー画像からマスク領域を修復した修復カラー画像が出力される．

4. 提案手法

ここでは本研究の手法について学習時と修復時に分けて説明する．

4.1 学習時

本研究では，エッジ画像だけでなく，深度画像もカラー画像修復の補助として用いることで，より性能の高い画像修復をすることを目的とする．したがって，EdgeConnect の修復器に深度画像修復器を加えた 3 つの修復器を用いて 2 段階の学習を行う（図 4 赤枠）．

第 1 段階（図 4 黒線，赤線）では，まずカラー画像から補助画像（エッジ画像，深度画像）を作成する．エッジ画像は EdgeConnect と同じく Canny 法を用いて作成し，深度画像は Dense Depth [14] を用いた学習済みの深度推定器を使用して作成する．次に，作成した補助画像にマスクをかけ，マスクとともにそれぞれの補助画像修復器に入力する．補助画像修復器は EdgeConnect と同じく図 5 の構造であり，学習方法も同じである．ただし，深度画像修復器では，出力画像が深度推定器で作成した深度画像と同じになるように学習される．

第 2 段階（図 4 青線）では，マスク付きカラー画像とマスク，補助画像修復器で作成された修復エッジ画像と修復深度画像をカラー画像修復器に入力する．カラー画像修復器も補助画像修復器と同じ構造（図 5）であり，学習方法は EdgeConnect のカラー画像修復器と同じである．ただし，通常の畳み込みでは，修復エッジ画像と修復深度画像を等しく重要な情報として修復に反映させてしまい，双方の利点を活かせない．そこで，すべての畳み込みを Gated Convolution [2] に置き換え，対象領域で画像の重みづけもするようにした．こうしてエッジ画像と深度画像を用いてマスク領域の情報を増やし，有効な情報を画像修復に利用することが可能となる．

4.2 修復時

修復時は，EdgeConnect と同様に，画像修復をしたいカラー画像とマスクを用意し，カラー画像を学習済みの Dense Depth に入力して深度画像を作成する．次に，カラー画像とマスク，作成した深度画像を入力することで，第 1 段階，第 2 段階が順に実行され，修復されたカラー画像が出力される．

5. 実験

5.1 学習環境

学習データセットには paris-street-view [19] のうち，深度がほとんど均一である画像（空のみや壁のみの画像）や深度の推定が大きく違う画像を除いた 5301 枚のトレーニングデータ（画像サイズ 936 × 537）を，画像サイズ 537 × 537 で左，真ん中，右の 3 つに分割した 14903 枚の画像を用いる．分割するのは EdgeConnect と実験条件をそろえるためである．このうち，14000 枚をトレーニングデータ，1903 枚をバリデーションデータとした．マスクには，あらゆるマスクサイズ，形状に対応するため，ランダムマスクのデータセット [12] を用いた．

各修復器の学習アルゴリズムには Adaptive moment estimation（Adam）[20] を利用し，1 回の学習でどれだけパラメータを更新させるかを決める学習率は 0.0001，バッチサイズは 4 とした．また，学習はバリデーション時のピーク信号対雑音比（Peak Signal-to-Noise Ratio；PSNR）や損失関数が収束するまで行った．

5.2 性能評価方法

提案手法の有用性を確認するため，paris-street-view のテスト画像にランダムマスクをかけ，どれほど元画像と同じように修復できたかを評価する．評価指標には，平均絶対誤差（Mean Absolute Error；MAE）と PSNR，Structural similarity；SSIM [21] を用いる．これらの評価指標は修復した画像と元画像とのずれを表す．

MAE は，テスト画像 n 枚としたとき，修復結果を $x_i(i = 1, 2, 3, ..., n)$，元画像を $y_i(i = 1, 2, 3, ..., n)$ とすると，次の式で定義される．

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_\mathrm{i} - y_\mathrm{i}| \tag{3}$$

式（3）より，修復結果が元画像に近づくほど MAE は小さくなる．つまり，MAE が小さいほど修復精度は良くなる．

PSNR の計算には平均二乗誤差（Mean Squared Error；MSE）が用いられる．画像サイズが $w \times h$ であるカラー画像の修復画像を $x_{i,j}(i = 1, 2, 3, ..., w)(j = 1, 2, 3, ..., h)$，元画像を $y_{i,j}(i = 1, 2, 3, ..., w)(j = 1, 2, 3, ..., h)$ とすると，MSE は次の式で定義される．

$$\mathrm{MSE} = \frac{1}{wh}\sum_{i=1}^{w}\sum_{j=1}^{h}(x_i - y_i)^2 \tag{4}$$

修復画像が元画像に近づくほど MSE は小さくなる．そして，式（4）を用いて，画素値の差が大きすぎると鈍くなる人の目に近づけるため，PSNR は次の式で定義される．

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}} \tag{5}$$

表 1 実験結果．深度画像モデルは EdgeConnect のエッジを深度画像に変更したモデル．マスク領域は画像にマスクをかけた領域の割合を示す．PSNR,SSIM は高いほど良く，MAE は小さいほど良い．最良の結果を赤字，最低の結果を青字で示す．

	マスク領域（%）	EdgeConnect	深度画像モデル	提案手法 1	提案手法 2
PSNR	0-10	35.35	34.96	35.35	35.59
	10-20	30.00	29.48	30.14	30.29
	20-30	27.27	26.94	27.59	27.75
	30-40	25.28	24.82	25.51	25.57
	40-50	23.85	23.49	24.06	24.26
	50-60	20.84	20.70	21.22	21.28
SSIM	0-10	0.984	0.982	0.985	0.985
	10-20	0.952	0.948	0.955	0.956
	20-30	0.915	0.909	0.921	0.928
	30-40	0.873	0.860	0.878	0.882
	40-50	0.825	0.813	0.829	0.838
	50-60	0.704	0.695	0.712	0.722
MAE（%）	0-10	0.49	0.50	0.47	0.45
	10-20	1.23	1.27	1.18	1.13
	20-30	2.19	2.24	2.09	2.01
	30-40	3.09	3.23	2.96	2.88
	40-50	4.02	4.13	3.88	3.72
	50-60	6.28	6.38	6.03	5.89

式（5）より，修復精度がよいほど MSE が小さくなるため，PSNR は大きくなる．

しかし，PSNR ではピクセルごとに値を算出するため，全ピクセルが 1 ずれると見た目は同じだが，値が大きく変化する欠点がある．そこで SSIM では，フィルタを用意し，フィルタがかかった範囲のすべての値を計算に用いることで周囲との相関を取り入れ，PSNR の欠点を補った．フィルタがかかった修復画像の領域を x，フィルタがかかった元画像の領域を y とする．そして，x と y の平均値を μ_x と μ_y，x と y の標準偏差を σ_x と σ_y，x と y の共分散を σ_{xy} とすると，SSIM は次の式で定義される．

$$\mathrm{SSIM} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6}$$

式（6）において，8 ビット画像の場合は $c_1 = (0.01 \times 255)^2, c_2 = (0.03 \times 255)^2$ を用いる．これらは正則化定数であり，フィルタ内の平均，標準偏差がゼロに近い領域で計算できなくなることを防ぐ．SSIM は 1 に近いほど修復精度が良くなる．

5.2.1 実験結果

テストには paris-street-view のテストデータセット 100 枚のうち，深度がほぼ均一な値になる画像（壁のみの画像など）や深度の推定が大きく間違った画像を除いた 45 枚の画像を用いた．また，マスクにはランダムマスクのデータセット [12] を使用した．本研究では，EdgeConnect と EdgeConnect のエッジ画像を深度画像に変更したモデル，EdgeConnect に深度画像を付加したモデル（提案手法 1），深度画像を付加し，さらに GatedConvolution も用いて重みづけの仕方を学習させるモデル（提案手法 2）の学習をし，それぞれにテスト画像を入力して出力結果の評価を行った．

図 6 提案手法 1 の修復結果．左上：マスク付き画像，中上：修復エッジ画像，右上：元画像，左下：EdgeConnect，中下：修復深度画像，右下：提案手法 1

定量的な評価結果を表 1 に示す．エッジ画像を深度画像に変更したモデルでは，すべてのマスクサイズ，評価指標

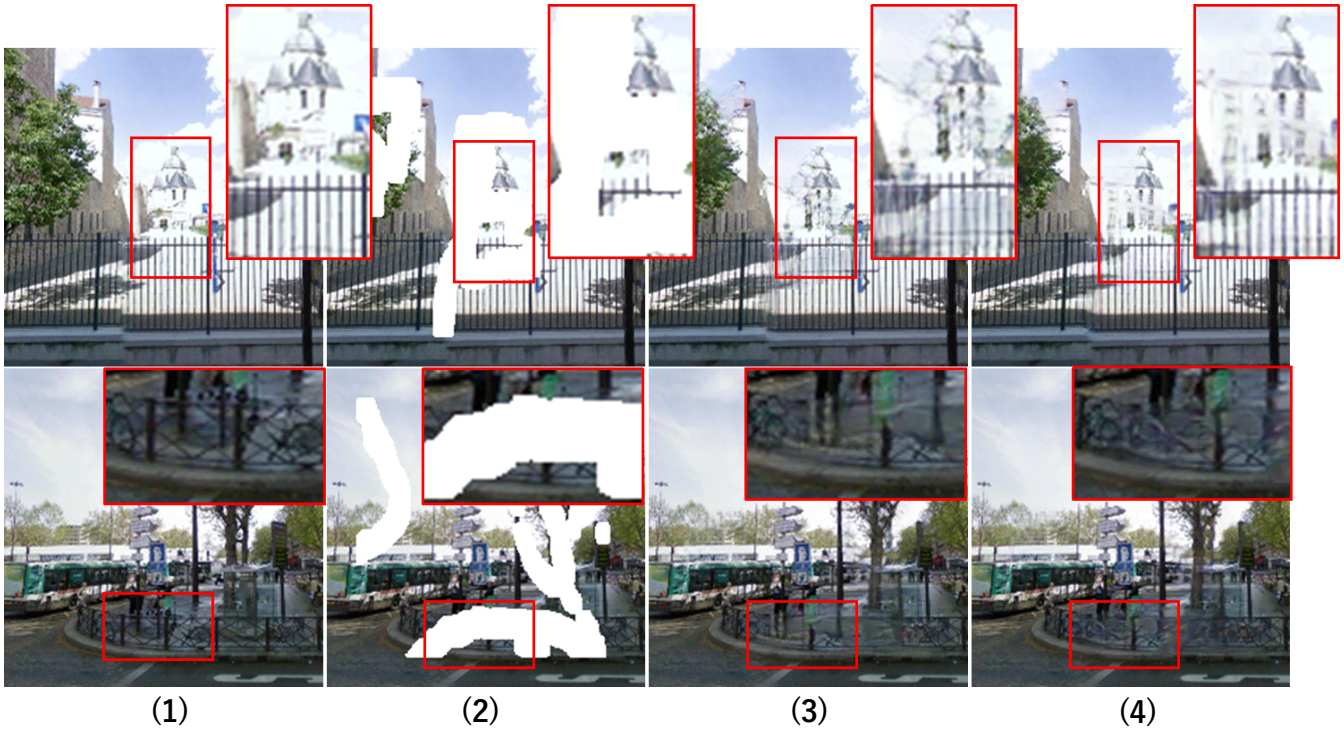

図 7 提案手法 2 の修復結果．(1)：元画像，(2)：マスク付き画像，(3)：EdgeConnect，(4)：提案手法 2

において精度が下落した．これは，深度画像だけではエッジのように明確な物体境界を抽出できず，さらにカメラからの距離により滑らかに変化する深度情報によって修復結果も平滑化され，色が混ざった領域が生成されたためだと考えられる．

提案手法 1 では，すべてのマスクサイズ，評価指標で Edge Connect 以上の精度を得られた．一方で，実際の修復結果を図 6 に示す．修復エッジ画像では木と背景をつなげて修復されているが，深度画像では木が手前にあるように修復できている．しかし，提案手法 1 の修復結果を見ると，木の下部は透過したような不自然な修復がされている．また，木の上部はぼかされており，定性的精度があまり向上していないことが分かる．これは，マスク領域の情報が増えたことで修復が容易になったが，エッジ画像と深度画像の情報を同様に反映させたため，不要な情報まで修復に使用された結果だと考えられる．

そこで，領域ごとに画像の重みづけも行う提案手法 2 を用いた．表 1 より提案手法 2 はすべての評価指標で提案手法 1 よりも精度が向上した．実際の修復結果を図 7 に示す．図 7 は (1) が元画像，(2) がマスク付き画像，(3) が Edge Connect の修復結果，(4) が提案手法 2 の修復結果である．上の画像を見ると提案手法 2 のほうが手前の柵を鮮明に修復でき背景の建物も違和感が少なく修復されている．また，下の画像では，(3) は人の足と柵が混在して不自然な修復がされているが，(4) では柵が手前にあるように修復され，奥行方向の前後関係を考慮した修復ができている．

次に，画像の重みづけもする提案手法 2 では深度画像が有効でない画像に対しても修復精度が下落しないと考えられる．したがって，元の paris-street-view データセットを用いて学習し，テストを行った．結果を表 2 に示す．EdgeConnect の結果は公開されている学習済みモデルでテストした結果である．表 2 よりすべての評価指標で精度の向上が確認された．

また，実際に図 8 左の黄色枠内の観光客を除去した結果が図 8 右である．観光客を違和感なく除去することができていることが分かる．

6. まとめと今後の展望

本研究では，画像修復精度の向上を目的とし，補助画像としてエッジ画像を使用した EdgeConect に深度画像を付加したモデル（提案手法 1）と，提案手法 1 に Gated Convolution という畳み込み方法を用いて画像の重みづけの仕方も学習させたモデル（提案手法 2）の 2 つを使用して実験を行った．提案手法 1 の結果から，複数の補助画像を画像修復に用いると定量的評価の精度は向上するが，ただ入力するだけでは定性的精度は向上しないことが分かった．一方，提案手法 2 では画像の重みづけの仕方も学習させたことで，2 枚の補助画像のうち，カラー画像の修復に有効な情報を自動的に反映し，修復精度の向上につながった．

表 2 元の paris-street-view データセットを使用した実験結果．マスク領域は画像にマスクをかけた領域の割合を示す．PSNR，SSIM は高いほど良く，MAE は小さいほど良い．

マスク領域	PSNR		SSIM		MAE（%）	
（%）	EdgeConnect	提案手法 2	EdgeConnect	提案手法 2	EdgeConnect	提案手法 2
0-10	35.49	36.37	0.985	0.988	0.48	0.42
10-20	30.17	31.06	0.954	0.961	1.23	1.11
20-30	27.23	27.99	0.912	0.923	2.13	1.95
30-40	25.41	26.10	0.870	0.884	3.05	2.82
40-50	23.64	24.26	0.809	0.827	4.17	3.86
50-60	21.44	21.78	0.708	0.720	6.03	5.76

図 8 画像修復結果． 左: カラー画像，右：本手法の修復結果

今後の方針としては，セグメンテーション画像などのほかの補助画像も追加することでさらなる修復精度の向上を目指す．

また，本手法は画像サイズの違いに脆弱であり，高解像度画像対して精度が保証されない問題点がある．したがって，今後は画像サイズによらない修復精度の向上を目指す．

参考文献

[1] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *ICCV*, 2019.

[2] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019.

[3] Marcelo Bertalmío, Luminita A. Vese, Guillermo Sapiro, and Stanley J. Osher. Simultaneous structure and texture image inpainting. In *CVPR*, 2003.

[4] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *ICCV*, 2003.

[5] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[6] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B. Goldman, and Pradeep Sen. Image melding: combining inconsistent images using patch-based synthesis. *ACM Trans. Graph.*, 31(4):82:1–82:10, 2012.

[7] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014.

[9] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4):107:1–107:14, 2017.

[10] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *CVPR*.

[11] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C.-C. Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. In *BMVC*, 2018.

[12] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018.

[13] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018.

[14] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *CoRR*, abs/1812.11941, 2018.

[15] John F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[16] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, 2016.

[18] Leon A. Gatys. *Texture synthesis and style transfer using perceptual image representations from convolutional neural networks.* PhD thesis, University of Tübingen, Germany, 2017.

[19] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A. Efros. What makes paris look like paris? *ACM Trans. Graph.*, 31(4):101:1–101:9, 2012.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[21] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.