

Pix2pix を用いた画像の輪郭線に着目した画像修復
Image Inpainting Focusing on Edge of Images Using pix2pix

皆藤 優太[†]
Yuta Kaito

田村 仁[‡]
Hitoshi Tamura

1. はじめに

1.1 研究背景

画像修復は、長年盛んに研究が行われている。画像修復とは、画像内に入り込んだノイズや画像の欠損箇所を修復する技術の一つである。使用用途は、画像の劣化による傷の修復や、写真の赤目除去、ストリーミングビデオにおける符号化及び伝送過程で失われた部分の置き換え、動画のロゴの削除に用いられたりするなど、多岐に渡る。本稿では、画像の欠損箇所の修復を扱う。

従来の主流だった画像修復メソッドの一部は OpenCV(Open Source Computer Vision Library) [1]に取り込まれている。OpenCV は、Intel 社が開発・公開したオープンソースのコンピュータビジョン向けのライブラリである。C/C++, Java, Python 等で使用可能である。図 1 に OpenCV に



元画像 入力画像 出力画像

図 1 Inpaint 関数による修復

取り込まれた画像修復メソッドによる画像修復を施した画像を示す。図 1 は、OpenCV に取り込まれている Inpaint 関数にて画像修復した画像である。

近年では、従来手法と変わって機械学習を用いた手法が主流となっている。機械学習を用いた手法のほうが、欠損領域が広いほど従来手法よりも自然な画像修復ができるためである。

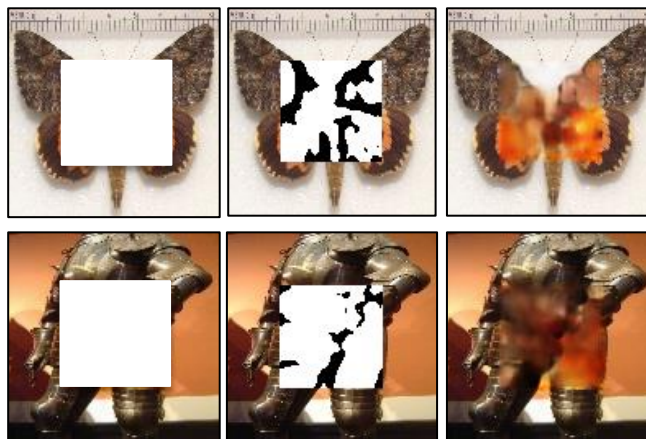


図 2 提案手法による修復の流れ(左→右)

1.2 関連研究

機械学習を用いた画像修復の関連研究として、D.Pathak et al.[6]や S.Iizuka et al.[2], G.Liu et al.[3]などがある。

[6]は、画像修復パイプラインに AlexNet[12]に由来するエンコーダーを持つ。畳み込み層を用いて特徴抽出を行い、画像の修復を行う。

[2]は[6]を元としているが、[6]との違いは大域識別ネットワークと局所識別ネットワークの評価用ネットワークと、欠損領域の補完ネットワークを持つ点である。大域識別ネットワークは画像全体が自然な画像になっているか評価し、局所識別ネットワークは欠損の補完領域周辺をより詳細に整合性を評価している。この 2 つのネットワークを騙すように欠損領域の補完ネットワークを学習させている。

[3]で提案されているのは、[2]を受けて修復用モデルに不足領域と元画像全体の両方を調べさせることでピクセルを再構築するよう学習させることと、その中で Partial Convolution(部分的な畳み込み)を行うことを提案している。不足領域と元画像全体の両方を調べさせることは、[2]と大きな差は無い。欠損領域のマスクを使用者が作成した後、マスクの周辺ピクセルから徐々に修復して行く。周辺ピクセルを修復する度にマスクを縮小させ、

[†] 日本工業大学 機械システム工学専攻,
Nippon Institute of Technology Department of Mechanical
Systems Engineering

[‡] 日本工業大学 先進工学部 ロボティクス学科,
Nippon Institute of Technology Department of Advanced
Engineering Faculty of Robotics

これを繰り返すことで欠損領域を修復している。これを取り入れたため、[2]よりも性能が向上させられた。

しかし、これらの手法は高周波成分(濃淡変化の激しさ、つまり輪郭線のこと。)に対して苦勞させられる場面がある。

近年、高周波成分の修復を強化するために、輪郭線に着目した画像修復手法が提案され始めている。多くの輪郭線に着目した手法は輪郭線を修復したのちに着色を行う。

輪郭線を用いた画像修復術として、L. Liao et al. [4]やK.Nazeri et al. [13]がある。

[4]では、Holistically-nested edge detection (HED)[5]法を用いてカラー画像をまず線画化し、その後細線化(線を1ピクセルにすること。)を施す。図3はHED法にて画像を線画化した様子である。線画化された画像内の欠損箇所を、畳み込み層で構成されたニューラルネットワーク(encoder-

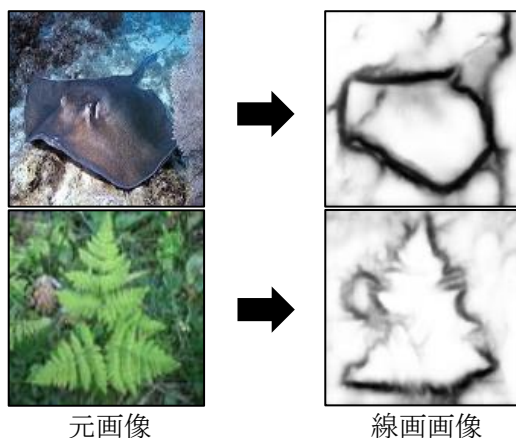


図 3 HED による線画化

decoder)で欠損箇所の輪郭線を修復する。修復済みの線画画像に対してD.Pathakらの手法をもとにしたフレームワークによって着色を行う。輪郭線の修復と着色を一組とする画像修復手法である。

[13]ではCanny法[14]にてカラー画像から輪郭線を抽出した画像を輪郭線修復モデルの学習に用いている。線画として修復してから着色を行う流れは[4]とあまり大きな違いはない。

1.3 研究目的

画像修復において輪郭線に着目する理由は、輪郭線に着目をあまりしていない手法よりも高周波成分の再現を向上させるためである。[13]では、[2]よりもさらに自然な画像修復が可能となっている。

図4は、D.Pathakらの手法において輪郭線がぼ

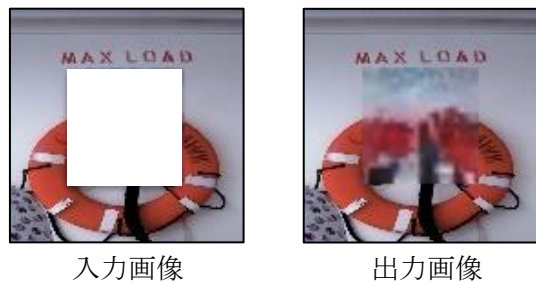


図 4 CE による修復

やけている例である。

輪郭線に着目した画像修復手法では、修復モデルの学習に線画画像のデータセットを用いる。

関連研究では線画化した後に細線化した画像を用いている。細線化された画像を用いた方がぼやけの少ない画像修復が実現できるのだ[4,13]。

しかしながら、細線化は画像の情報を絞りすぎているのではないかと考えた。[13]では細線化画像と共にグレースケール画像も修復時の入力にしている。

本稿では、細線化せずに輪郭線の太さの情報を残した線画の2値化画像を用いる輪郭線に着目した画像修復手法を提案する。

図2は、本稿の提案手法により画像を修復した例である。

また、本稿では輪郭線に着目した画像修復手法を提案するにあたり、画像を線画化する際の細線化の有無、細線化をしない場合に画像の画素階調の制約等を行った画像を用いてデータセットを複数通り作成した。輪郭線修復モデルを学習した場合に、それらを用いてそれぞれの程度の差が出るのかを検証した。

検証ではpix2pix[7]を用いて輪郭線に着目した画像修復フレームワークを作成した。

また比較対象として、線画画像を用いずに学習された画像修復モデルを用いた画像修復手法としてD.Pathakらの手法を用いた。これは輪郭線修復モデルの学習に用いるデータセットの線画画像の違いによる画像の修復結果を、線画画像を学習に用いていない画像修復モデルにより修復された画像と比較することが目的である。

2. 輪郭線に着目した画像修復

深層学習を用いた画像修復フレームワークの多くは、画像内の欠損領域の輪郭線や色などを同時に修復するように学習されている。

輪郭線に着目した画像修復フレームワークでは、

入力画像を線画化し、着色より先に画像内の欠損領域の輪郭線を修復し、その後着色する。

そのため、フレームワーク内に 2 種類の修復モデルを用意する。1 つは輪郭線修復を行うモデル、もう 1 つは着色を行うモデルである。2 種類のモデルを配置することにより、輪郭線に着目した画像修復を実現する。

3. 実装

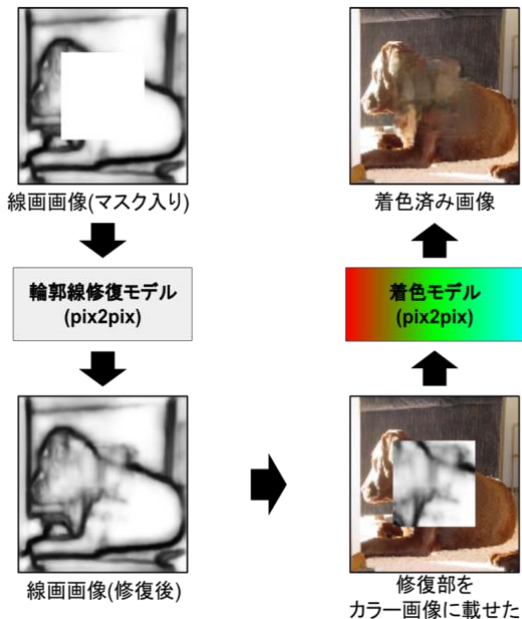


図 5 修復の流れ

3.1 画像修復の流れ

図 5 は、本稿で作成した輪郭線に着目した画像修復フレームワークの画像修復の流れを示す。

まず欠損のある画像を線画にし、それを輪郭線修復用モデルで欠損領域の輪郭線を修復する。次に、欠損領域を修復した線画画像から輪郭線を修復した部分を切り取り、線画化前の色情報が残っている画像に貼り付ける。それを着色モデルにて線画部分を着色する。以上が修復の流れである。

修復の流れをこのようにしたのは、画家が絵を描く過程を参考[15]にしたためである。

多くの画家は、色を塗る前に線画を描く。その後、色を塗る。本稿や[4], [13]でも、同じ理由から輪郭線を先に修復した後に着色をする修復順を採用している。

3.2 フレームワークの実装

それぞれのモデルは、Keras[8]で実装した pix2pix フレームワークで作成した。pix2pix を実

装する際に[9]を参考にした。

また、それぞれのモデルは共通のカーネルサイズや畳み込み層の数をしているが、入出力画像の色情報の有無に違いがある。輪郭線修復モデルは画像の色情報ではなく明るさ情報のみ、着色モデルは色情報を見るようにさせた。

バッチ数は 32、最適化アルゴリズムは Adam を用いた。学習率は 10^{-3} 、 β_1 は 0.9、 β_2 は 0.999、 ϵ は 10^{-8} とした。また、モデル学習中に PC の RAM メモリの使用量を抑えるため、学習用データセットを分割して利用できるようにした。

4. 実験

データセットを用いて各モデルを学習させ、その後、実際に欠損のある画像を修復させ、その出力画像を用いて評価を行った。

本稿の比較対象として Context Encoder(CE)[6]を用いた。CE と比較するにあたり、提案モデルの訓練に用いた画像を用いて CE を学習し直した。

定量的評価指標は Structural SIMilarity(SSIM)[16]、Peak signal-to-noise ratio(PSNR)の 2 種類を用いた。

SSIM, PSNR はその値が大きいほどよい。

4.1 学習用データセットの作成

本稿で作成したデータセットの画像は、ImageNet[11] よりダウンロードした画像から約 10 万枚を選び、それを用いた。また、全ての画像を 128×128 px にリサイズした。



図 6 事前実験の結果

また、事前実験より細線化を施したデータセットよりも細線化を施さないデータセットで学習させた修復モデルの方がわずかながら出力画像の質が良かった。よって細線化データセットは本実験で扱わないことにした。

図 6 は、細線化の有無によるそれぞれの修復画像である。

4.1.1 輪郭線修復モデル学習用データセット

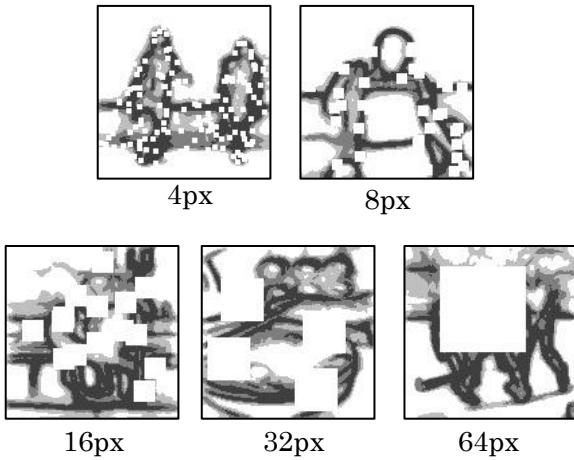


図 7 マスクの配置例

輪郭線修復モデルの学習には線画画像で構成されたデータセットを用いる。線画画像にマスクを載せた画像を訓練画像、マスクを載せる前の画像を正解画像としたペアでデータセットを作成した。

マスクは一辺が 4px, 8px, 16px, 32px, 64px の正方形をランダムに一種類選び、一枚の訓練用画像ごとに合計面積が 4096px^2 前後となるように場所もランダムに敷き詰めた。また、マスクサイズの分布は訓練用画像に対して約 5 等分になるよ

うにした。事前実験より、単純に画像中央にマスクを与えて学習させるよりも、このようにマスクの位置やサイズをばらけさせたりすることで輪郭線の修復能力が向上したため、今回の実験でも用いることにした。図 7 は、マスクを載せた画像の例である。

輪郭線修復モデル学習用のデータセットで用いた線画画像は、全て HED にて線画化を行った。また、線画化後の画像から、2 値化・4 値化画像の 2 種類をさらに作成した。よって、単純な線画画像、それを 2・4 値化した画像で構成した計 3 種類のデータセットを作成した。2 値化に変換時の閾値は 256 階調あるうちの 127 階調目を閾値にし

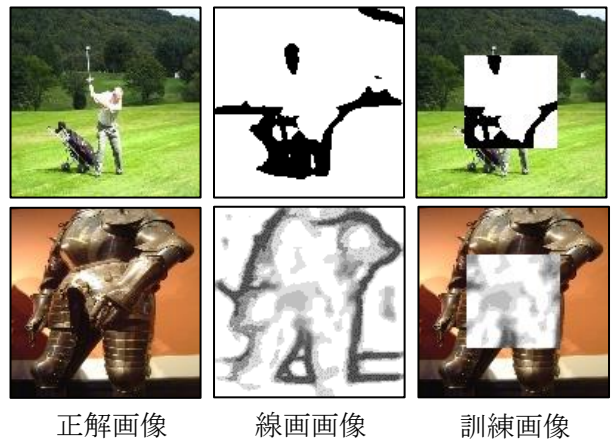


図 8 学習用データセットの例

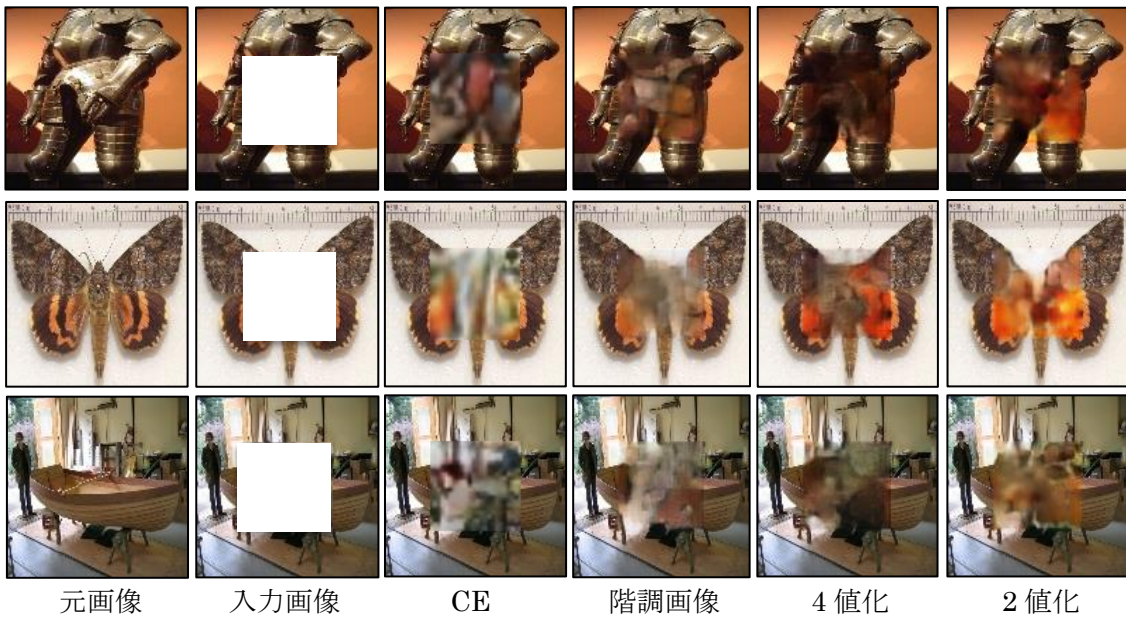


図 9 出力結果

た(固定閾値による 2 値化). また, 4 値化は 256 階調を均等に 4 で割り, それぞれ割り当たった階調を 64 階調ずつ変換した.

4.1.2 着色モデル学習用データセット

着色モデルの学習には, カラー画像上のその一部を線画化したものを訓練画像, 線画化前のものを正解画像としたペアで構成されたデータセットを用いることにした. 線画化を行った領域は画像中央の一辺 64px の正方形サイズの部分である.

輪郭線修復モデル学習用データセットに合わせてこちらにも計 3 種類(単純な線画画像, 2・4 値化した画像用)のデータセットを作成した. 図 8 は, 着色モデル学習用データセット中の正解画像と訓練画像の例である.

4.2 各モデルの学習

実装したフレームワークとデータセットを用いてモデルの学習を行なった. 輪郭線修復モデルと着色モデルの学習時のパラメータは 3.2 章の通りである.

5. 結果

学習を終えたモデルにて実際に画像修復を行った. 評価に使用した画像は, [11]よりダウンロードした画像より選んだ. また, それらは各モデルの学習に用いられていない画像である. また, 評価画像に用いたマスクのサイズは一辺 64px の正方形マスクで, その位置は画像中央に設定した.

5.1 定性的評価

図 9 は, 修復モデルによって修復された画像と比較対象の手法にて修復した画像である.

線画画像の階調を絞っていくにつれて輪郭線の境目がはっきりと修復されているのがわかる. CE の修復画像と我々の手法で修復した画像を見比べると羽の柄や鎧の輪郭がはっきりしている.

輪郭線の修復に着目してみると, 2 値化を施したデータセットで学習したモデルによる修復画像が, 最も輪郭線のぼやけが少ない

色彩の修復に着目してみると, 4 値化を施したデータセットで学習したモデルによる修復画像が最も自然に着色できている.

5.2 定量的評価

実際に修復した画像の PSNR 値と SSIM 値を測定し, その平均を求めた. 表 1 にそれをまとめた.

表 1 各評価値

	CE	細線化なし	2 値化	4 値化
SSIM	0.767	0.786	0.783	0.787
PSNR	18.71	19.10	19.15	18.96

表 1 より, SSIM 値が最も大きかったのは 4 値化のデータセットを用いた修復モデル. PSNR 値が最も大きかったのは 2 値化のデータセットを用いた修復モデルである.

本稿の手法で修復した画像の評価値がどれも僅差となっている. 特に, SSIM については大きく差がついていない.

提案手法の SSIM 値の最大値と最小値の差は 0.004 ほどである. しかし, 提案手法の SSIM 値の最大値と CE を比べると最大で 0.02 程の差がある. PSNR 値については, 最大値と最小値の差は 0.19 ほどの差がついた. 提案手法の PSNR 値の最大値と CE を比べると最大で 0.44 の差がある.

6. 考察

図 9 より, 輪郭線修復モデルの学習用データセットの画像の階調を絞るほど輪郭線をはっきり修復できていることがわかる. はっきりとした輪郭線を修復するには, 輪郭線修復モデルの学習には細線化を施さない 2 値化画像を用いるのが良い, というのが図 9, 表 1 から言える.

しかしながら, 着色に関してはどの手法も似たようなものだが, 学習用データセットの階調を絞らずに学習した修復モデルのほうが自然に着色しているように見える.

このことから, 輪郭線の濃淡が着色時の手がかかりになっているのではないかと考えられる. しかし, 輪郭線に濃淡を与えると, 修復した輪郭線がぼやける傾向がある. このことより, 着色モデルに欠損外部の色情報と輪郭線情報, これに加えて輪郭線を修復した後に輪郭線の濃淡を修復するモデルによって修復された濃淡情報を入力することで, はっきりとした輪郭線を修復しながら着色もより自然に施せるのではないかと考える.

7. おわりに

本稿では, 輪郭線に着目した画像修復手法の提案と, 修復モデルの学習に用いるデータセットの変化による修復結果の検証を目的とした.

結果として, 2 値化した線画画像を修復モデルの学習に用いる pix2pix を用いて輪郭線に着目し

た画像修復手法の提案と、修復モデルの学習に用いるデータセットの変化による修復結果の検証を行った。

検証は、

- ・事前実験にて線画画像の細線化の有無による修復結果への影響
- ・線画画像の階調を2値化・4値化することによる修復結果への影響

についてである。結果は図9、表1の通りである。表1より、SSIM値は僅差だったため、PSNR値が他よりも最も大きい線画の2値化画像を修復モデルの学習に用いるのがよいと考える。

本稿の実験を通して、pix2pixのパラメータの大きな変更なしにデータセットの変更を行うことで修復結果にばらつきが出ることをよく知ることができた。

pix2pixの畳み込み層のカーネルサイズ等を変更して輪郭線修復モデルや着色モデル用に最適化するべきと考えたが、変更を加えない状態においてデータセットの影響が学習結果・画像の修復結果にどのように出るか不明であったため、それらの変更は行わなかった。

今後は、畳み込み層のカーネルサイズや2値化時の閾値の変更を施した際の修復結果の評価等を行い、さらなる性能の向上を目指す。

謝辞

本稿を作成するにあたり、ご指導、ご鞭撻を賜りました指導教員の田村仁先生に感謝申し上げます。

参考文献

- [1] "OpenCV library". <https://opencv.org/>, (参照 2019-4-11).
- [2] Satoshi Iizuka and Edgar Simo-Serra and Hiroshi Ishikawa, "Globally and Locally Consistent Image Completion", ACM Transactions on Graphics (Proc. of SIGGRAPH 2017) 2017.
- [3] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, Bryan Catanzaro, "Image Inpainting for Irregular Holes Using Partial Convolutions", ECCV 2018.
- [4] L. Liao, R. Hu, J. Xiao, and Z. Wang, "Edge-aware context encoder for image inpainting," 2018.
- [5] S. Xie and Z. Tu. "Holistically-nested edge detection," In IEEE International Conference on Computer Vision (ICCV), pp: 1395-1403, 2016.
- [6] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp: 2536-2544, 2016.
- [7] P. Isola, Jun-Yan Zhu, T. Zhou, A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks", University of California, Berkeley In CVPR 2017.
- [8] Chollet, François and others, "Keras", 2015. <https://keras.io>
- [9] pix2pix-keras-byt, "<https://github.com/tommyfms2/pix2pix-keras-byt>", (参照 2018-10-12)
- [10] A. Telea, "An Image Inpainting Technique Based on the Fast Marching Method," J. Graphics Tools, vol. 9, no. 1, pp.23-34, 2004.
- [11] ImageNet, "<http://image-net.org/index>", (参照 2018-11-16).
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.
- [13] K.Nazeri et al. "EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning", arXiv:1901.00212v3 [cs.CV] 11 Jan 2019.
- [14] J. Canny, "A computational approach to edge detection", IEEE Transactions on pattern analysis and machine intelligence, (6):679-698, 1986.
- [15] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?", ACM Transactions on graphics (TOG), 31(4):44-1, 2012.
- [16] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity", IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.