没入感拡張システムのための pix2pix を用いた周辺視野映像生成

木村 直紀* 暦本 純一*†

概要. 没入感は映像体験において最も重要な要素の一つである. 没入感を高める方法の一つとして, 周辺視野への映像提示を行う「没入感拡張システム」が提案され, これを実現するシステムの開発がこれまで多く試みられてきた. しかしこういった「没入感拡張システム」は十分に普及していない. システムに対応するコンテンツが少ないためである. 対応コンテンツを作るための試みは, 撮影方法やコンピュータービジョンのアルゴリズムなどによって実現が試みられてきたが, 撮影の技術的障壁や処理速度, 生成映像の質といった点で問題が残っている. 一方, 近年ディープニューラルネットワークを用いた画像補完の研究が大きな成功を収めている. 補完と周辺視野映像生成は画像を生成する箇所と面積が異なるものの, 同様の方法で周辺視野映像の生成も実現できることが予想された. そこで, 本研究では, 汎用的な画像変換が可能な pix2pix と呼ばれるモデルを用いて周辺視野映像の生成を試み, 2 つの学習手法とその結果を報告すると共に, 新たに発見された課題とこれからの展望を示した.

1 はじめに

没入感は映像体験において最も重要な要素の一つである. そのため, 没入感を高める方法がこれまでに多く考案されてきた. それらの試みの一つに周辺視野に映像を提示することによる没入感拡張がある. 周辺視野へ映像を提示し, 視野を映像によって埋め尽くすことで, あたかも映像の世界に飛び込んだかのような没入感, 臨場感を与えることができる. この映像体験を実現するため様々なシステムが開発されてきた. 例えば Illumiroom[1]は, テレビ画面の周りの壁や家具に対して周辺視野映像を映し出し, 没入感のあるゲーム体験を実現する. 実用化されているシステムとしては, 正面のメインスクリーンに加えて, 左右にそれぞれ 1 面ずつ設置されたスクリーンにも投影し水平画角 270 度まで投影する映画館システムが存在する[2][3].

しかし, こういった周辺視野への映像提示によって没入感を拡張するシステムは十分に普及していない. その大きな理由として, これらのシステムに対応したコンテンツが少ないことが挙げられる. 対応コンテンツを作る最も基本的な方法は, 周辺まで撮影できる画角の広いカメラを用いる, または同時に複数のカメラを用いて撮影を行うことである. しかし, この方法にはポストプロダクションの複雑さや, 撮影技法の制約・ノウハウの不足などの問題があり, またすでに撮影されてしまった膨大な既存の映像コンテンツに対しては用いることができない.

そこで既存のコンテンツから周辺視野映像を生成する方法やアルゴリズムが考案されてきた. しかし処理速度と生成された映像の自然さの両方を十分なレベルで達成したものは未だ存在していない.

一方, 近年ディープニューラルネットワーク (DNN) を用いた画像補完の研究が大きく進んでいる. 画像補完と周辺視野映像生成は, 作り出す画像の面積と場所が違うものの, ほぼ同じネットワークで処理できることが予想される.

そこで本研究では, DNN を用いた周辺視野映像生成を試み, 2 つの学習手法とその結果を報告すると共に, 新たに発見された課題とこれからの展望を示す. 図 1 は今回の手法で生成できる映像を用いた没入感拡張システムの実装例である.



図 1. 周辺視野映像提示の例. 映像が表示されているモニター画面の周囲に, 今回の手法で生成した周辺視野映像をプロジェクターによって投影している.

Copyright is held by the author(s).

*東京大学大学院情報学環
† ソニーコンピュータサイエンス研究所

2 関連研究

2.1 周辺視野映像生成

映像没入感拡張システムを実現するため，周辺視野映像の生成方法はこれまでにも多く考案されてきた．Philips 社の AmbiLux[4]は，テレビに映る映像を拡大したものを，テレビの裏側に取り付けられたプロジェクターによって壁面に投影するものである．この方法はシステムが単純で実現が容易だが，テレビに映る映像と壁面に投影された映像の境界で映像がずれてしまい，視聴者に違和感を感じさせてしまう．Aides ら[5]は PatchMatch アルゴリズムを映像の周辺領域に対して適用することで，映像の拡張を試みている．この方法による生成映像は場面によって非常に自然だが，1 フレームにつき数分の処理時間がかかるため，スポーツ中継や音楽コンサート中継のようなリアルタイム性が必要とされるコンテンツには用いることができない．これに対して Laura ら[6]はより軽量なアルゴリズムを提案されているが，生成された映像の自然さに改善の余地がある．

このように，現在十分な処理速度と生成映像の自然さを両立する周辺視野映像生成の方法は提案されていない．

2.2 DNN による画像補完

近年ディープニューラルネットワーク，特に畳み込みニューラルネットワークを用いた画像補完技術が大きく進歩している[7][8]．画像補完とは，データが一部欠損した，つまり「穴」の空いた画像データに対して，穴を補完し元のデータの復元を試みるものである．特に飯塚ら[8]の提案したモデルは，一見しただけでは補完されたものとはわからないほど自然な画像補完に成功している．生成したい周辺視野部分に対してマスクを施し，「穴」とすることで，周辺視野映像も同様に生成できると考えられる．

3 pix2pix

3.1 pix2pix の特徴と使用方法

pix2pix とは Isola ら[9]が提案した機械学習のモデル，および論文と共に公開された実装である．今回はこの pix2pix の実装を用いた．pix2pix では，2 つの画像のペアを提示し続けることで，それらの画像間の変換方法を学習・生成することができる．線画から写真の復元，航空写真から地図画像の生成，昼の風景から夜の風景への変換などの用途が変換の例として挙げられており，高い汎用性を持つ．また，簡単に使うことのできる実装が公開されており，画像のペアを用意するだけで使うことができる．実際の使用時には，それぞれ 256x256pixel のサイズの，変換前の画像と変換後の画像を左右に並べ，



図 2. pix2pix で入力する訓練データの例．画像は place2 dataset[10]から使用した．

512x256pixel の画像を用意することが求められる．今回は元の映像に対して周辺視野映像を生成した結果ちょうど 2 倍の大きさの映像になるように，上下左右 25%ずつクロップしたものを変換前，元の画像を変換後として図 2 のように訓練データを作った．

3.2 Generative Adversarial Network

pix2pix は Generative Adversarial Network，通称 GAN の一種である．GAN とは Generator と Discriminator 2 つのネットワークを持つモデルである．Generator は Discriminator を騙せるようになるべく訓練データに近いデータを生成し，Discriminator は送られてくるデータが訓練データか生成データかを識別しようとする．最終的に Generator は訓練データと同じようなデータを生成し，Discriminator の識別率は 50%になることが期待される．

また，pix2pix は GAN の中でも Conditional GAN の一種である．Conditional GAN は意味を持つベクトルや画像，文字など，生成の補助となる条件を入力に加えることで生成精度の向上を図る．pix2pix の場合は画像がこの条件となる．

3.3 パッチサイズの調整

pix2pix には patchGAN という仕組みが用いられている．これは Discriminator に画像すべてを与えるのではなく，16x16 や 70x70 のようなより小さなパッチに切り出して与えることで，大域的な判定を残しながら学習パラメータ数を削減し，より効率的な学習を可能にする仕組みである．このパッチサイズを変更すると生成される結果が異なる．パッチサイズが小さいほど局所的な判定，大きいほど大域的な判定を行う．様々なタスクに対する汎用的な設定として pix2pix のデフォルト設定では 70x70 のパッチサイズとなっているが，今回の外挿タスクに最も適するパッチサイズを探すため，1x1, 16x16, 70x70, 256x256 の 4 つの条件で映像を生成する実験を行なった．結果の一部を図 3 に示す．パッチサイズが大きいほどノイズが少なく，より中心部分との整合性が画像を生成する傾向があったため，今回の全て実

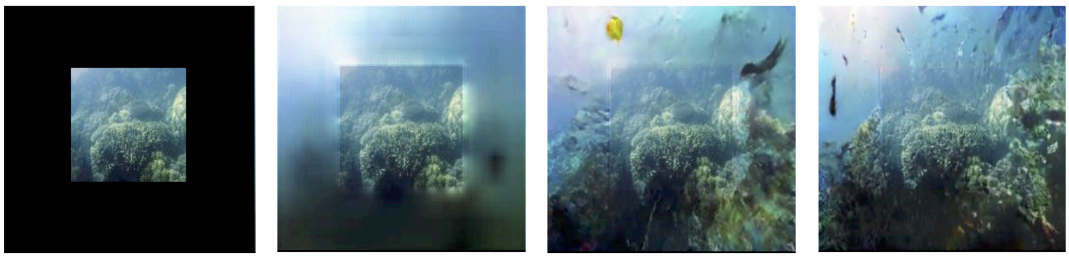


図 3. 各パッチサイズでの画像生成結果.画像は Place2 dataset[10]から用いた. 左から input, パッチサイズ 1, 70, 256 と並んでいる.

験では 256x256 のパッチサイズを用いた. pix2pix をそのまま使用した時, つまりデフォルトの設定である 70x70 のパッチサイズの時, Discriminator の構造は, C64-C128-C256-C512 になっているが, 今回は 256x256 のパッチサイズに変更したため, C64-C128-C256-C512-C512-C512 という構造になっている.

4 実験方法

今回の実験では 2 つの学習方法を試行した. 以下にそれぞれの詳細を示す. pix2pix は 256x256pixel 以上のサイズの画像に対しても使うことができるが, 処理速度も重要であることと, 中心視野に対し解像度が低くなる周辺視野の特性を考慮し, 訓練データ・テストデータ共に 256x256pixel のサイズで実験を行なった. バッチサイズをデフォルトの 1 から推奨値の 4 に変更し, epoch 数は一律で 200 とした. 学習は Intel Core i7-5930K CPU と NVIDIA GeForce TITAN X GPU を搭載した PC で行われた.

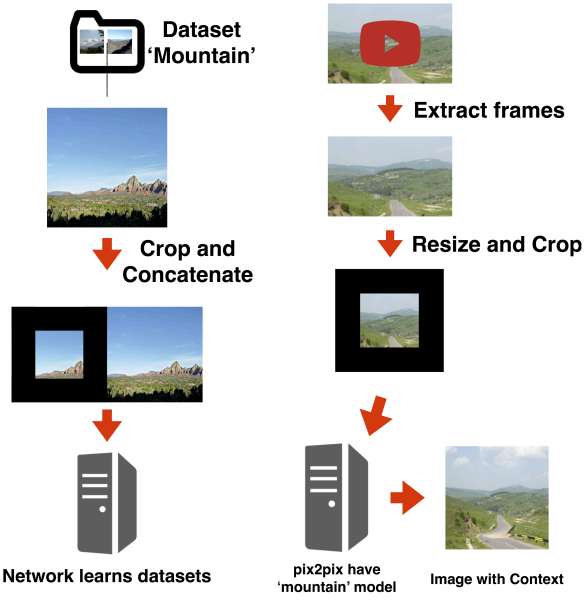


図 4. 手法 1 の手順.

4.1 手法 1：データセットを用いて学習させる

予め画像データセットや動画のデータセットを用いてモデルを生成する方法を試みた. 手順の概要を図 4 に示している. この方法の場合, 予め学習を済ませておくため, 周辺視野映像生成を行う時にはモデルを適用するだけで済む. したがってモデルの適用時間が十分に短ければ, スポーツ中継や音楽ライブ中継のようなコンテンツに対してリアルタイムで周辺視野映像を生成できる可能性がある. しかし, シーンに合ったモデルを選択しなければ自然な映像生成ができないため, 汎用的な利用を実現するには, 自動的に適切なモデルを選択するアルゴリズムを組み合わせる, またはどのようなシーンに対しても自然な外挿が可能な汎用モデルを生成する, などの進展が必要となる. 今回は主に place2 dataset[10]と動画共有サイトから収集した映像を用いて, 5 つのモデルを生成した. 生成されたモデルと使用されたデータセットの対応を表 1 に示した. 動画は多くの場合 16:9, もしくは 4:3 のアスペクト比であったため, 1:1 になるようアスペクト比を変更して成形した. モデル名と学習に使用したデータセットの対応を表 1 に示した. Racing は上下左右 25%ずつクロップすると, 変換前の画像に映るものが地面ばかりになってしまい, 空やコース周辺の風景を生成できな

表 1. モデルと使用したデータセットの対応.

モデル名	データセット
Ocean	動画共有サイトで収集した, 合計 5 時間 2 分の動画データから毎秒 1 フレームずつ抽出したもの (6046 枚)
Mountain	Place2 dataset のうち, mountain(5000 枚)
Field	Place2 dataset のうち, field:cultivated (5000 枚)
Racing	レーシングゲームのキャプチャ映像から毎秒 1 フレームずつ抽出したもの (3394 枚)
Desert road	Place2 dataset のうち, desert road (5000 枚)

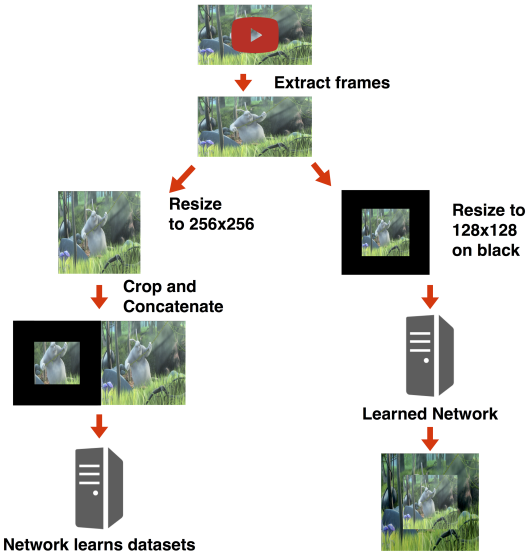


図 5. 手法 2 の手順.

かったため，上下左右 15%ずつクロップした．

4.2 手法 2：周辺視野映像を生成したい映像自体を訓練データとして学習させる

周辺視野映像を生成したい映像を学習データとして使う方法を試みた．まず映像データから 1 秒に 1 フレームずつ画像として抽出する．正方形になるようにリサイズしたのち，上下左右 25%ずつクロップしたものを変換前，元の映像データを変換後とする．学習後，元の映像を 50%に縮小し入力することで周辺視野映像生成を行う．この方法の概要を図 5 に示した．この方法は手法 1 と異なりシーンに合わせたモデルの選択を行う必要がなくどのような映像に対しても同じ処理で完結させることができる．一方，予め学習を済ませておくことができないため，リアルタイム性が求められるようなコンテンツに対して用いることができない．

表 2. 各モデルが 1200 枚の画像を処理するのに要した時間と 1 秒あたりの生成枚数（fps）.

モデル名	処理時間(sec)	fps
Ocean	33.63	35.68
Mountain	35.62	33.69
Field	31.68	37.88
Racing	31.20	38.46
Desert road	30.45	39.40

4.3 生成の結果

手法 1 において 1200 枚の画像を生成するのに要した時間を表 2 に示した．また二つの手法によって生成を行った結果を図 6 に示した．処理は Intel Core i7-5930K CPU と NVIDIA GeForce TITAN X GPU を搭載した PC によって行われた．処理時間は各モデルにおいて約 30 秒であった．1 秒あたり 30

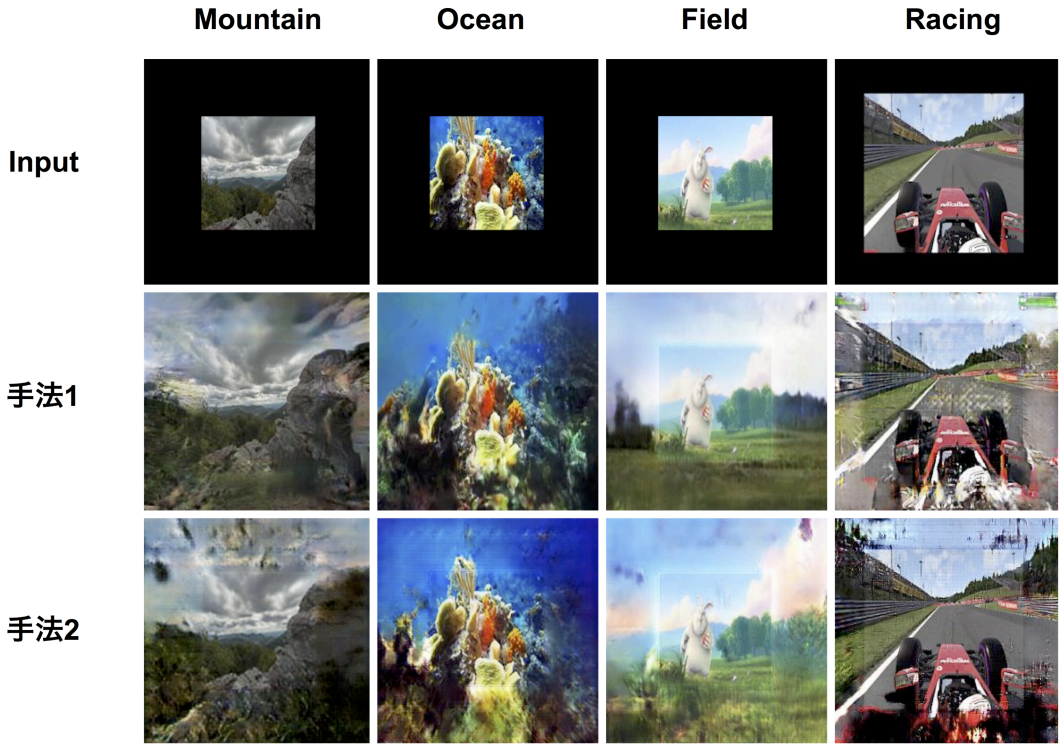


図 6. 手法 1 と手法 2 による画像生成結果. テストデータはそれぞれ， Mountain : ©2015, Garrett Martin / https://vimeo.com/139927695, Ocean : ©2017, Eric Falconi / https://vimeo.com/199482708, Field : ©2008, Blender Foundation / www.bigbuckbunny.org , Racing : F1(TM)2016 より.

枚以上の処理が行われており, 24fps や 30fps を用いる多くの映像コンテンツにおいてリアルタイムに周辺視野映像を生成できる可能性が示唆された.

5 考察

図 6 の Mountain, Ocean のように, 手法 1 はテストデータに対して適切なモデルが与えられた時, 自然な生成映像が得られた. 一方 Field は Mountain, Ocean と比較して入力部分と生成された部分の差異が大きいことが見て取れる. これは データセットが田や畑のような人工的な田園風景を多く含むものであった一方, テストデータは森や平原を写したものだったためだと考えられる. 入力映像とモデルの相性によって生成される映像の自然さに差が出たと言える. また手法 2 は手法 1 よりもノイズが多い一方, 色や質感についてはどの映像においても再現されていた.

手法 1 と既存手法による結果を図 7 に示した. 各論文[5][6]で生成している箇所やサイズ, アスペクト比が異なるため, 元の映像を約 2 倍にするという今回の目標に合わせてサイズを整え, アスペクト比を 1:1 にして整理した. それらの論文に掲載されていない結果については図に載せておらず穴になっている. Aides ら[5]の手法による結果を論文の題目から”Multiscale”, Turbanr ら[6]の手法による結果を論文の題目から”Extrafoveal”として掲載した. “Semitrailer”では元の映像の範囲より外に出たトラックについて, Multiscale[5]では予測生成できているが, Extrafoveal[6]と今回の手法 1 では生成できていない. これは予測生成の根拠となるフレームが, 予測生成しようとするそのフレームのみであるためであると考えられる. これに対しては, ベクトル情報を入力として入れることや, 予測・生成したいフレームだけでなく, 前後のフレーム数十枚を入力することで解決できる可能性がある.

提案手法の問題点として, 生成したフレームを画像シーケンスとして動画に戻すと, フレームごとに明度や生成した物体の形が異なり, 強いちらつきが発生してしまうことが挙げられる. これはフレームごとに予測生成を行うために, 隣接するフレームでも生成結果が微妙に異なってしまうことによる. これに対する解決策として, 各フレームを前後 2 枚ず

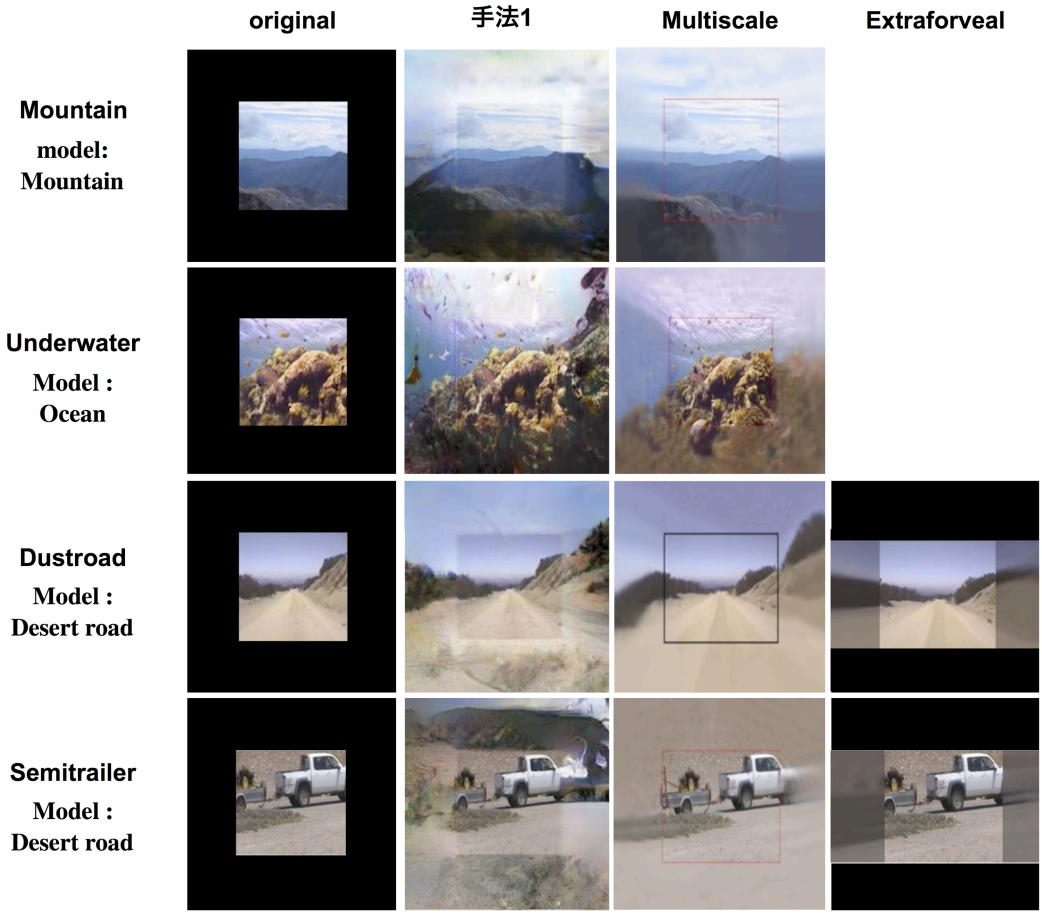


図 7. 既存の手法との比較. Mountain, Underwater, Dustroad, Semitrailer のテストデータは Aides ら[5]が公開しているテストデータを用いた. Model は手法 1 による生成で使われたモデル（表 1）を示している.

つと該当フレームを含めた，計 5 枚の平均とする時間平滑化フィルタを適用することで明度のちらつきについてはほぼ解消することができた．しかし生成された物体の形状の差異によるちらつきは十分に解消することができなかった．生成されたフレーム間の差異がより小さくするために，条件となる入力画像の枚数を増やすことが解決策として考えられる．

6　まとめと今後の展開

6.1 まとめ

本論文では pix2pix を用いた周辺視野映像の生成について検証を行った.今回は２つの手法とその結果について紹介した．Discriminator のパッチサイズは映像全体を覆うサイズにした時最も映像が生成されること，生成したフレームを映像に戻す時，時間平滑化処理を施すことで明度の違いによるちらつきを抑えられることを明らかにした．一方で，今回の方法は予測・生成したいフレーム１枚から外側の領域を予測したため，動いて画面外に出たもの，または画面内に入ってくるものを予測することができなかった．これに対しては，ベクトル情報を入力として入れることや前後数枚の画像を入力として入れることによる対策が考えられる．手法１では場面に応じてあらかじめ学習された適切なデータセットを選び，処理を行った．また 256x256pixel の画像を入力した時，映像生成実行時１秒に約 30 フレームを処理することができた．つまり，予め各データセットを用いて学習を行ったモデルを複数用意し，シーンに対して適切なモデルを選ぶアルゴリズムを開発することで，スポーツ中継を始めとするリアルタイム性が求められるコンテンツに対しても適用できる可能性がある．手法２はコンテンツごとに学習時間が必要であった．しかし，手法１のようにシーンによってモデルを変える必要がない利点があった．また，映像独特の色やテクスチャを再現することができた．

6.2 今後の展開について

フレームイン・フレームアウトする物体に対応するためには，ベクトル情報を入力できる，または複数の画像を入力できるネットワークが有効である可能性がある．また，映画のようにカットが多くシーンが多数登場する映像に対しても適用でき，また視聴時にリアルタイムで処理できるような汎用的な仕組みを作るためには，シーンを判別して適切なモデルを選ぶアルゴリズムを考案する必要がある．

映像にした時のちらつきの問題も解決する必要がある．明度の違いによるちらつきは時間平滑化を施すことで解決することができたが，予測生成した物体の形の違いによるちらつきは十分に解決できていない．今回はユーザースタディを行うことができなかったが，第三者に映像を視聴してもらい感想を求めたところ，「生成した部分が周辺視野に来るように視聴しているときはあまり気にならないが，少し離れて全体を見ると気になってしまう.」という評価を得た．ちらつきによる影響も含めてユーザースタディを行い，今回の手法によって生成された周辺視野映像の提示によって，実際に没入感を高めることができるのか確認する必要がある．

参考文献

[1] Jones, Brett R., et al. "IllumiRoom: peripheral projected illusions for interactive experiences." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013.

[2] CJ CGV, ScreenX http://screenx.co.kr/en/ (2017/9/01 確認)

[3] Barco,BarcoEscape https://ready2escape.com/(2017/9/01 確認)

[4] Philips, AmbiLux TV, https://www.philips.co.uk/c-m-so/televisions/p/ambilux (2017/9/01 確認)

[5] A. Aides, T. Avraham and Y. Y. Schechner, "Multiscale ultrawide foveated video extrapolation" *2011 IEEE International Conference on-Computational Photography (ICCP)*, Pittsburgh, PA, 2011, pp. 1-8.

[6] L. Turban, F. Urban and P. Guillotel, "Extrafoveal Video Extension for an Immersive Viewing Experience," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 5, pp. 1520-1533, May 1 2017.

[7] Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[8] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. "Globally and Locally Consistent Image Completion". ACM Transaction on Graphics (Proc. of SIGGRAPH), 2017.

[9] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *arXiv preprint arXiv:1611.07004* (2016).

[10] Places: A 10 million Image Database for Scene Recognition, B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017