Vol.2018-CG-172 No.2
Vol.2018-DCC-20 No.2
Vol.2018-CVIM-214 No.2
2018/11/7

色と動き情報の学習による静止画像からのシネマグラフ生成

遠藤 結城[1]　金森 由博[2]　栗山 繁[1]

概要：シネマグラフは、画像中の一部の領域のみが動くループアニメーションであり、コンテンツをアーティスティックに表現できる。一般にシネマグラフの制作は、動画を入力として用意することに加え、変化領域の指定や自然なループ生成のためのユーザ入力など、手間のかかる作業を要する。そこで本研究は 1 枚の静止画像を対象に、最小限のユーザ入力で自然なシネマグラフを生成することを目的とする。ニューラルネットワークにもとづく動画生成モデルはこれまで数多く研究されてきたが、高解像度で多様なアニメーションを生成するのは未だ挑戦的な課題である。提案手法では、動画中のアニメーションを、動きと色の変化の 2 つの要素に分解して考え、2 種類のニューラルネットワークモデルにこれらを別々に学習させる。各モデルには画像を直接生成させるわけではなく、入力画像を変換するための色変換マップと流れ場を予測させ、これらに正則化をかけることで、モデルの汎化性能を向上させる。これにより、既存手法よりも格段に高解像度で、長く滑らかなアニメーションを実現する。さらにユーザは、低次元の潜在変数によって出力を制御しながら、多様なアニメーションを生成できる。既存手法との比較を通して、提案手法の有効性を検証する。

1. はじめに

写真は静的なコンテンツであるが、我々人間はその景観が時間の経過とともにどういう風に動き、外見を変えていくのか想像をめぐらすことができる。シネマグラフはそのようなダイナミックな効果を、シームレスなループアニメーションとして静止画像の一部に付与できるコンテンツ表現の 1 つとして知られている。これまで質の高いシネマグラフを簡単に作る方法が研究されてきたが [8], [20]、入力となる動画を用意するのは大きな手間であった。1 枚の画像からシネマグラフを作るためには、Adobe After Effect などの商用サービスが利用できるが、画像中のどの領域をどの方向に動かしたいかなど細かい編集操作がユーザに要求される。

最近は特に深層ニューラルネットワークを用いた画像生成技術が大きな注目を浴びており、これが単一画像からの動画生成にも応用されている [6], [19]。しかし、画像に対しては高解像度で質の高い結果が生成できつつあるが [18]、動画に対しては未だ挑戦的な課題である。動画は複数枚の画像を生成する分、学習すべきバリエーションが増え、学習に多くのリソースを要するため、高い汎化性能を獲得するのは容易ではない。そのため、最先端の手法でも入力が高解像度になるほど結果にノイズがのったり、動きや色の変化を含む多様かつ長いフレームを生成できなかったりするのが問題であった。また、1 枚の画像に対する未来の変化には、周囲の環境に依存した複数の可能性（不確定性)があることへの対処も重要である。例えば入力画像がほぼ同じでも、雲が風向きなどの環境要因の影響を受けて右に動く動画と左に動く動画を与えることを考える。このような場合、一意に定まらない変化を学習するのは難しく、推定時には一対一（one-to-one）の出力しか得られない。従来手法でも変分オートエンコーダなどを利用することで対処しているものの、ユーザが調整しながら多様な結果を生成するには十分ではない。

本稿では、深層ニューラルネットワークを用い、1 枚の静止画像から高品質なシネマグラフを生成する手法を提案する。本研究は空の雲や湖の水の動き、1 日の昼夜変化などを含む風景のシーンを対象に、Web 上で比較的容易に手に入るタイムラプス動画を訓練データとして教師なし学習を行う。図 1 に示す入力画像のような景観は、1 日のうちに数秒単位で雲や水面が動くだけでなく、数分から数時間単位で夜になるにつれて色も変化していく。これらすべての要素を考慮するには長期間のフレームを小刻みに予測する必要があるが、予測できる長さには限界がある。そこでまず 1 つ目のアイディアは、動画中のアニメーションを動き（**モーション**）と外見の色（**アピアランス**）の変化の 2 つの要素に分解して考え、2 種類のニューラルネットワークモデルにこれらを非同期で学習させることである。

[1] 豊橋技術科学大学
[2] 筑波大学

Vol.2018-CG-172 No.2
Vol.2018-DCC-20 No.2
Vol.2018-CVIM-214 No.2
2018/11/7

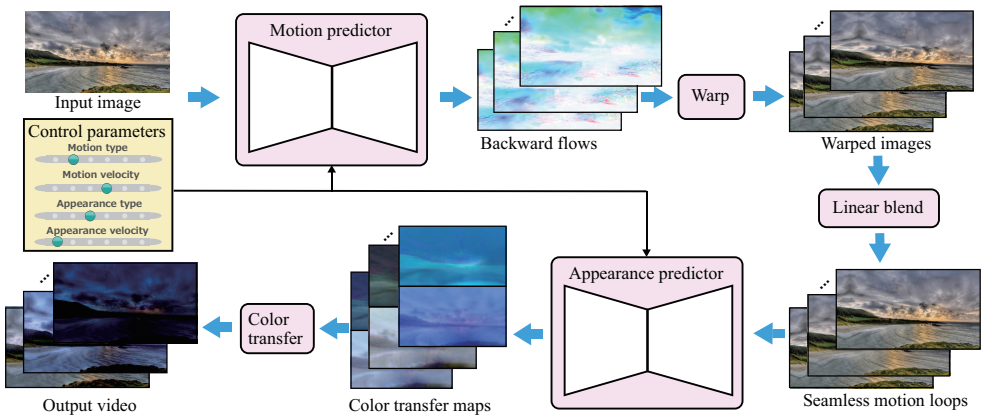

図 1 提案手法のシネマグラフ生成時の処理の概観。

2 つ目のアイディアとして、モーション用とアピアランス用の各畳み込みニューラルネットワーク（CNN）には動画を直接生成させず、入力画像を変換するためのモーションと色変換マップを予測させる。そしてこれらのマップに正則化をかけることでモデルの汎化性能の向上を図る。この際モデルの複雑さを減らし学習コストを削減するために、既存手法 [6], [19] のように 3D 畳み込み層を使うのではなく、2D 畳み込み層を使い複数枚の画像を生成するアプローチを採用する。2D 畳み込みによって計算された各マップにもとづき、入力画像にワーピングと色変換を適用することで、各フレームの出力を生成する。

最後に 3 つ目のアイディアは、各モデルに出力を制御するための潜在変数を教師なしで学習させることである。これにより入力画像に対する未来の不確定性に対処し学習の安定化を図るとともに、この潜在変数をベースにした少数のコントロールパラメータを用いることで一対多（one-to-many）の画像を生成できるようにする。ユーザはモーションとアピアランスの 2 つの要素について、別々に結果をコントロールできる。

提案手法は先行研究と比べて次の長所を持つ。(i) 数百から数千のタイムラプス動画データを教師なし学習することで、従来よりも格段に鮮明かつ高解像度のアニメーションを生成できる。(ii) モーションだけでなくアピアランスのアニメーションを長期的に生成できる。(iii) それらを組み合わせたアルゴリズムによってシームレスなループ再生が可能なシネマグラフを生成できる。(iv) ユーザは動き方のタイプや速度など少数のパラメータを調節することで、多様なアニメーションを生成できる。以上をふまえた既存手法との比較により、提案手法の有効性を評価する。

2. 関連研究

静止画像をアニメーションさせる技術において本研究が対象とする風景のシーンを扱ったものとしては、波打つ湖、雲の動き、揺れ動く木々などの自然現象を確率的な過程でモデリングしたアニメーション生成システムが提案されている [1]。このシステムは現象ごとに細かいパラメータ調整が必要な一方で、複雑なパラメータ調整なしに流体のモーションを参照動画の高周波成分として転写する手法も提案されている [9], [10]。また、参照動画の位相情報をアニメーションとみなし、画像に転写する手法もある [11]。他方でアピアランスの変化を扱ったものとしては、対象画像と類似したシーンの参照動画との密な画素の対応を計算し、それをヒントに各参照フレームの色味をアフィン変換によって転写する手法がある [15]。さらにより多様なシーンを含む画像データベースを構築し、天候や季節の変化なども再現できるようになった [4]。

このような example-based アプローチは、参照動画の情報を直接転写するおかげで写実的な結果を生成できるが、入力画像と似た参照動画を用意できないと結果が不自然になることがある。例えばアピアランスを扱う既存手法の 1 つは画素単位の対応づけが重要なため、参照動画中に雲のような動的な物体があると生成結果に時間方向のちらつきを生じる制約が報告されている [15]。また参照動画を用意する負担はデータベースを使えば削減できるが、システムの構築や運用には多くのリソースが要求される。例えば流体モーションの生成システムは、高速化前は 1 時間単位の時間を要し [9]、分単位に削減できる高速化も 15 コンピュータ 100 コアの環境が要求される [10]。それに加え流体領域のアルファマットの修正などの手動操作も必要になる。最後に、これらの手法はモーションとアピアランスの両方の要素を含むループアニメーションは扱えない。提案手法はユーザの負担を減らしつつ、両方を対象にシームレスなシネマグラフを生成できる。生成される結果は少数のパラメータを使った潜在変数の調整や内挿によって、滑らかかつ多様に変化する。Learning-based アプローチとして、訓練済みの数百 MB のモデルと潜在変数データを記憶しておけば運用できるのも利点の 1 つである。

ここ数年でニューラルネットワークモデルをデータから学習する learning-based アプローチも急速に発展している。例えば、畳み込み循環型ニューラルネットワーク RNN

Vol.2018-CG-172 No.2
Vol.2018-DCC-20 No.2
Vol.2018-CVIM-214 No.2
2018/11/7

（cRNN） [13] や 3D 畳み込み [19] を利用して動画像を生成する研究がある。最近 Li らは流れ場を予測するためのジェネレータを条件付き変分オートエンコーダに拡張することで、未来の不確かさを考慮した学習を可能にした [6]。予測された流れ場は入力画像のワーピングと GAN による結果の洗練に使われ、最終的に 128×128 サイズの 16 フレームの動画が生成される。

提案手法の流れ場の予測とワーピングによるアプローチは Li らと類似しているが、正解の流れ場を使わずに教師なしで学習する点と、GAN による洗練はせずに出力を入力に循環させて動画を生成する点で異なる。結果は 5 節で比較する。

3. 手法の概要

本研究のゴールは、1 枚の静止画像が所与のもと、画像中の一部の領域が動いたり色が変化したりするループアニメーションの動画（シネマグラフ）を生成することである。先に述べた通り、タイムラプス動画からこのような変化をまとめて捉えようとすると、長期間のフレームを小刻みに予測する必要がある。これを避けるために、短い時間間隔で変化するモーションと、長い時間間隔で変化するアピアランスを非同期で扱う。すなわち、モーションの予測とアピアランスの予測の 2 つのフェーズに分割する。

図 1 に示すように、モーション予測フェーズにおいては、モーション予測器を用いて、入力画像が次のフレームでどのように動くかを表す後ろ向き流れ場の系列を生成する。流れ場の各画素には、前のフレームとの対応を表すオフセットである x 方向と y 方向の 2 次元のベクトルが格納されている。生成された流れ場を利用して入力画像をワーピングで再構成することによって、各フレームの画像を得る。この際、画像をワーピングさせるだけでは、ループ再生した際に最初と最後のフレームにつながりがなくなってしまう。そこで、前後のフレームを線形に混ぜ合わせることによってシームレスなモーションのループを生成する。アピアランス予測フェーズにおいては、アピアランス予測器を用いてモーションループの各フレームに適用する色変換マップを生成する。色変換マップは従来の color transfer [14] と似た要領で、各画素に対するスカラー値の乗算と可算によって各フレームに適用される。最終的に出力される動画は、モーションだけでなくアピアランスの変化も含めてシームレスにループ再生することができる。さらに、ユーザは少数のパラメータ（例えばモーションの動き方のタイプや速度など）を調整することで、最終的に出力されるアニメーションをコントロールすることもできる。

4. モデル

ネットワーク構造の詳細について本稿では割愛するが、基本的にモーション予測器 $G_{\mathcal{M}}$ とアピアランス予測器 $G_{\mathcal{A}}$ は、両方とも全層畳み込みニューラルネットワークによって構成されている。各予測器は、入力としてまず大きさ $w \times h$ 、チャネル数 3 の RGB 画像 $\mathbf{A} \in \mathbb{R}^{w \times h \times 3}$ を受け取る。例えば出力として予測する雲などの動きは 1 枚の画像のみでは決められず、風向きなどの環境要因によっても変化する。そのため未来の不確定要素を上手く考慮し学習させるために、出力をコントロールするための n 次元の潜在変数 $\mathbf{z} \in \mathbb{R}^n$ を導入する。これらの入力に基づいて、各予測器は入力画像を変換するためのマップを出力する。なお、画像をモデルに入力する際に各チャネルの値は $[-1, 1]$ の範囲に正規化する。

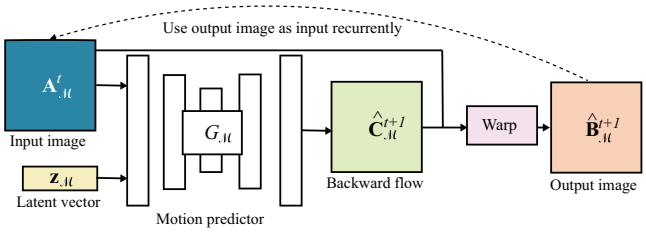

図 2 推定時のモーション予測器の順伝播の流れ。

以降の小節では、入力 $\mathbf{A}$ などの表記にはモーションに対しては $\mathcal{M}$ を 、アピアランスに対しては $\mathcal{A}$ を添え字として利用する。

4.1 モーション予測器

Inference. モーション予測器 $G_{\mathcal{M}}$ は、各画素が前のフレームのどこから来たのかを表す、後ろ向き流れ場 $\hat{\mathbf{C}}^t_{\mathcal{M}} \in \mathbb{R}^{w \times h \times 2} (t = 2, 3, ..., T)$ を推定する。流れ場は x 方向と y 方向の 2 チャネルのオフセット情報を各画素に持ち、値は $[-1, 1]$ の範囲に正規化されている。このとき入力画像の画素座標の空間も水平と垂直方向で $[-1, 1]$ の範囲にパラメータ化されており、推定された流れ場をもとにバイリニア補間によるワーピングによって入力画像を変形させる。各ステップの流れ場を推定するために、入力画像から時間が経過して大きく変化した出力を、入力画像だけから直接推定するのは難しい。そのため、1 ステップずつ小さく変化した出力から繰り返し流れ場を推定する。つまり図 2 に示す通り、タイムステップ t の入力画像 $\mathbf{A}^t_{\mathcal{M}}$ に対して、次のタイムステップ $t+1$ の流れ場 $\hat{\mathbf{C}}^{t+1}_{\mathcal{M}}$ を出力する。その後 $\hat{\mathbf{C}}^{t+1}_{\mathcal{M}}$ を用いて $\mathbf{A}^t_{\mathcal{M}}$ をワーピングさせて $\hat{\mathbf{B}}^{t+1}_{\mathcal{M}} \in \mathbb{R}^{w \times h \times 3}$ を得る。次のステップでは再度 $\hat{\mathbf{B}}^{t+1}_{\mathcal{M}}$ を $\hat{\mathbf{A}}^{t+1}_{\mathcal{M}}$ として利用し、この工程を繰り返すことで、T 枚の出力を生成する。以降ではモデルによって推定された変数にはハットの記号を付ける。

Training. シンプルにモーション予測器のニューラルネットワークを訓練させるためには、正解の流れ場が必要となる。しかしながら、本研究で対象とするような景観画像については、タイムラプス動画としてインターネット上に多くのデータが存在するものの、正解となるような人手で作

Vol.2018-CG-172 No.2
Vol.2018-DCC-20 No.2
Vol.2018-CVIM-214 No.2
2018/11/7

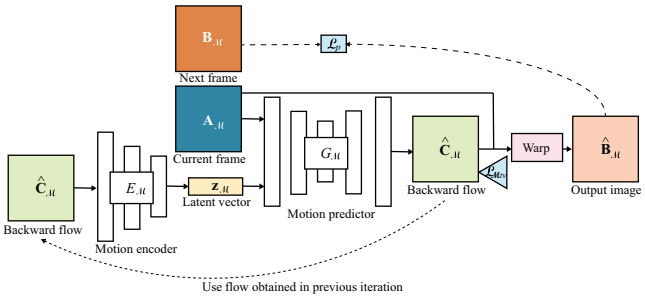

図 3 モーション予測器の学習方法。

成された流れ場のデータは存在しない。SpyNet [12] など 2 枚の画像から流れ場を推定する既存手法で正解データを生成するアプローチ [2], [17] も考えられるが、タイムラプス動画に対しては良好な結果が得られなかった（5.1 節参照）。そこで提案手法では教師なし学習によるアプローチを検討する。

図 3 にモーション予測器の学習方法の概観を示す。まず、ワーピング後の出力 $\hat{\mathbf{B}}_{\mathcal{M}}$ と、入力画像 $\mathbf{A}_{\mathcal{M}}$ に対する正解の次フレーム $\mathbf{B}_{\mathcal{M}}$ との誤差をピクセルごとに測る損失関数を定義する。

$$\mathcal{L}_p = ||\mathbf{B}_{\mathcal{M}} - \hat{\mathbf{B}}_{\mathcal{M}}||_2. \tag{1}$$

加えて、ネットワークから出力される流れ場が、空間的に滑らかに変動するという正則化を施すために、次の損失関数を定義する。

$$\mathcal{L}_{\mathcal{M}tv} = \sum_{i,j,c} \left(\psi(\nabla_x \mathbf{B}^{i,j}_{\mathcal{M}}) \left| \nabla_x \hat{C}^{i,j,c}_{\mathcal{M}} \right| + \psi(\nabla_y \mathbf{B}^{i,j}_{\mathcal{M}}) \left| \nabla_y \hat{C}^{i,j,c}_{\mathcal{M}} \right| \right), \tag{2}$$

$$\psi(\mathbf{w}) = \exp\left(-\frac{||\mathbf{w}||_1}{\sigma} \right), \tag{3}$$

ここで、添え字 i, j は画素の座標、c はチャンネルのインデックスを表す。この損失関数は、流れ場を単純に平滑化するのではなく、流れ場と対応する正解画像 $\mathbf{B}_{\mathcal{M}}$ に応じて、空間的な画素値の変化量が小さいところは近い方向の流れになるような重み関数 ψ がかけられている。σ は $\mathbf{B}_{\mathcal{M}}$ に応じた平滑化度合いを調節するパラメータである。最終的な損失関数は、λ_p と λ_{tv} の重みで比重を調整することで次のようになる。

$$\mathcal{L}_{\mathcal{M}} = \lambda_p \, \mathcal{L}_p + \lambda_{tv} \, \mathcal{L}_{\mathcal{M}tv}. \tag{4}$$

本学習過程において最も特徴的な点は、不確定な出力をコントロールするための潜在変数 $\mathbf{z}_{\mathcal{M}}$ をモーションエンコーダ $E_{\mathcal{M}}$ によって教師なしで学習・生成させる点にある。このような問題には、one-to-many の結果を生成する Bicycle GAN [22] でも取り組まれている。しかし我々のモーション予測器に、この手法はそのまま適用できない。なぜならば入力画像が次以降のタイムステップでどのよう

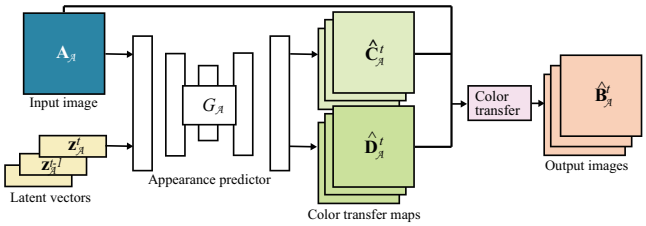

図 4 推定時のアピアランス予測器の順伝播の流れ。

に変化するかは、潜在変数 $\mathbf{z}_{\mathcal{M}}$ によって決められる一方で、そもそもの $\mathbf{z}_{\mathcal{M}}$ は学習時に正解データとして存在しないためである。$\mathbf{z}_{\mathcal{M}}$ は次のステップにおける流れ場に依存すると仮定し、流れ場をエンコードすることを考えたいが、入力となる流れ場も正解データとして存在しないのは明らかであり、シンプルには学習できない。そこで、エンコーダ $E_{\mathcal{M}}$ に渡す流れ場を、以下のように徐々に更新していくことでこの問題に対処する。最初はエンコーダ $E_{\mathcal{M}}$ に渡す流れ場をゼロテンソルとし、学習を繰り返す過程で、シーンごとに前のエポックで得られた出力 $\hat{\mathbf{C}}_{\mathcal{M}}$ を、$E_{\mathcal{M}}$ の入力とすることで学習を進めていく。訓練データから得られたシーンごとの最終的な潜在変数 $\mathbf{z}_{\mathcal{M}}$ は記憶しておき、推定時にはこれらの値をベースにユーザがパラメータを調整することで、複数パターンの動きを生成する。本稿では細かい実装の詳細については割愛するが、実験結果で実証する通り、この方法によってもっともらしい多様な動きを出力する潜在変数 $\mathbf{z}_{\mathcal{M}}$ を学習できる。モーションエンコーダはモーション予測器と同様の損失関数をにもとづいて学習される。

4.2 アピアランス予測器

Inference. アピアランス予測器 $G_{\mathcal{A}}$ は 2 つの色変換マップ $\hat{\mathbf{C}}^t_{\mathcal{A}} \in \mathbb{R}^{w \times h \times 3}$、$\hat{\mathbf{D}}^t_{\mathcal{A}} \in \mathbb{R}^{w \times h \times 3}$ を推定する。推定された色変換マップによる入力画像 $\mathbf{A}$ とのアダマール積 $\circ$ と可算によって、次式の通り $\hat{\mathbf{B}}^t_{\mathcal{A}}$ が出力として得られる。

$$\hat{\mathbf{B}}^t_{\mathcal{A}} = \tanh\left(\hat{\mathbf{C}}^t_{\mathcal{A}} \circ \mathbf{A}_{\mathcal{A}} + \hat{\mathbf{D}}^t_{\mathcal{A}} \right). \tag{5}$$

ニューラルネットワークから出力される $\hat{\mathbf{C}}^t_{\mathcal{A}}$ と $\hat{\mathbf{D}}^t_{\mathcal{A}}$ には特に値の範囲に制約を設けないが、最終的な出力画像 $\hat{\mathbf{B}}^t_{\mathcal{A}}$ については tanh によって $[-1, 1]$ の範囲となるよう制約を設ける。

図 4 に示す通り、アピアランス予測器はモーション予測器とは異なり、繰り返して出力を入力に戻すことで時系列の出力を計算するわけではない。もしこの処理を適用してしまうと、モーションのように入力画像の画素値を直接サンプリングするのではなく、出力に色変換を重ねていくことで徐々に画素値の誤差が蓄積して不自然な色味になってしまう。その代わりにアピアランス予測器では、潜在変数 $\mathbf{z}^t_{\mathcal{A}}$ の時系列データを入力することで、対応する $\hat{\mathbf{C}}^t_{\mathcal{A}}$、$\hat{\mathbf{D}}^t_{\mathcal{A}}$、および $\hat{\mathbf{B}}^t_{\mathcal{A}}$ も時系列で得るアプローチをとる。時系列

Vol.2018-CG-172 No.2
Vol.2018-DCC-20 No.2
Vol.2018-CVIM-214 No.2
2018/11/7

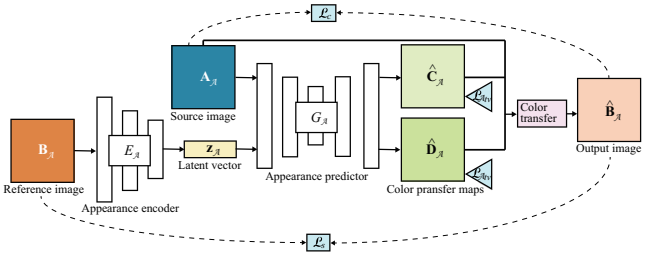

図 5　アピアランス予測器の学習方法。

データ $\{\mathbf{z}_{\mathcal{A}}^t\}$ は学習時に訓練データから得られた情報を記憶しておき、推定時にはこれらの値をベースにユーザがパラメータを調整することで、複数パターンのアピアランス変化を生成する。

Training. アピアランス予測器の学習のために、まずアピアランスがある程度変化した隣接フレームを対象に、損失関数を定義する。ここでは、モーション予測器のようにピクセルごとの誤差を測らない。その理由は、アピアランスの変化がモーションと比べ非常に遅いためである。アピアランスが変わる前後のフレームで、物体が大きく動くことがあるため、これをピクセルごとに比べてしまうとアピアランスとは無関係な要素を学習してしまう。そこで図 4 に示す通り、まずは色変換によって得られた $\hat{\mathbf{B}}_{\mathcal{A}}$ と、正解の色変換後画像 $\mathbf{B}_{\mathcal{A}}$ との間で次のスタイル損失を定義する。

$$\mathcal{L}_s = \sum_l ||G(F_l(\mathbf{B}_{\mathcal{A}})) - G(F_l(\hat{\mathbf{B}}_{\mathcal{A}}))||_2, \qquad (6)$$

ここで、関数 F は学習済みの 16 層 VGG ネットワークであり、添え字の l は何層目の特徴マップを出力するかを表す。関数 G は特徴マップのグラム行列を出力する。本手法では、VGG の relu_1_2、relu_2_2、relu_3_3、relu_4_3 層を l の対象とした。スタイル損失に加えて、入力画像の空間的なコンテンツ構造を保持したまま変化させるために次のコンテンツ損失を定義する。

$$\mathcal{L}_c = \sum_l ||F_l(\mathbf{A}_{\mathcal{A}}) - F_l(\hat{\mathbf{B}}_{\mathcal{A}})||_2. \qquad (7)$$

こちらは、入力画像 $\mathbf{A}_{\mathcal{A}}$ のコンテンツ構造として高周波情報を保持したいため、VGG の浅い層 relu_1_2 のみを l の対象とした。さらに、様々なシーンに対する汎化性能を向上させるために、色変換マップ $\hat{\mathbf{B}}_{\mathcal{A}}$ に次の正則化をかける。

$$\mathcal{L}_{\mathcal{A}tv} = \sum_{i,j,c} \left(\psi(\nabla_x \mathbf{A}_{\mathcal{A}}^{i,j}) \left| \nabla_x \hat{C}_{\mathcal{A}}^{i,j,c} \right| + \psi(\nabla_y \mathbf{A}_{\mathcal{A}}^{i,j}) \left| \nabla_y \hat{C}_{\mathcal{A}}^{i,j,c} \right| \right). \qquad (8)$$

この正則化は $\hat{\mathbf{D}}_{\mathcal{A}}$ に関しても同様である。なお、モーション予測器と異なり正解の出力画像 $\mathbf{B}_{\mathcal{A}}$ ではなく、入力画像の構造に合わせて平滑化するために $\mathbf{A}_{\mathcal{A}}$ の空間的な画素値の変化にもとづいて平滑化される。最終的な損失関数は λ_c 、λ_s 、および λ_{tv} の重みによって比重を調整することで次のように定義される。

$$\mathcal{L}_{\mathcal{A}} = \lambda_s \, \mathcal{L}_s + \lambda_c \, \mathcal{L}_c + \lambda_{tv} \, \mathcal{L}_{\mathcal{A}tv}. \qquad (9)$$

アピアランス予測器に与える潜在変数 $\mathbf{z}_{\mathcal{A}}$ の学習には、アピアランスエンコーダ $E_{\mathcal{A}}$ を利用する。アピアランスエンコーダは color transfer 後の画像と対応する正解フレームをエンコードし、アピアランス予測器と同様の損失関数を使って最適化される。

5. 実験

我々は提案手法を Python 言語と PyTorch ライブラリを用いて実装し、プログラムを NVIDIA GeForce GTX 1080 Ti GPU を搭載した PC 上で動かした。学習時の画像サイズ w および h は予測器では 256 、エンコーダでは 128 とし、潜在変数 $\mathbf{z}$ の次元 n は 8 とした。最適化のアルゴリズムには Adam [3] を利用し、学習率は 1.0×10^{-4} 、減衰率に関する 1 次と 2 次のパラメータを 0.5 と 0.999 、バッチサイズは 8 とした。損失関数に関する各パラメータは $\sigma = 0.1$、$\lambda_p = 1$、$\lambda_{tv} = 0.1$ 、$\lambda_s = 1$、$\lambda_c = 1.0 \times 10^{-4}$ とした。モーション予測器とアピアランス予測器で共通するパラメータには同じ値を用いた。なお、推定時には流れ場や色変換マップを得られた後、入力画像サイズに合わせて拡大することで高解像度の結果を生成できる。

各モデルは 2000 エポック学習し、シングル GPU 環境で 3 日程度の時間を要した。推定時は 640×320 サイズの 1 枚の画像に対して、1 フレームのモーションの予測に 0.054 秒、アピアランスの予測に 0.058 秒要した。モーションとアピアランスを含む 1 つのシネマグラフを作成するのに要した時間は、例えば 1010 フレームの生成に対して 98 秒であり、うちモデルのロードに 9 秒、モーションの予測に 11 秒、ブレンドに 6 秒、アピアランスの予測に 59 秒、動画ファイルとしての出力に 16 秒要した。

モーション予測の訓練には、Xiong らのタイムラプス動画データセット [19] を利用した。訓練用に 2,376 のビデオクリップがあり、各ビデオクリップには平均で 159 枚の画像が含まれている。アピアランス予測の訓練には、Youtube で *24 hour timelapse* というキーワードで動画を検索し、全部で 14 クリップ収集した。アピアランスはモーションよりも変化速度が遅いことから、隣接フレームで画素値の平均値の差がしきい値以下のものを不要なフレームとして機械的に除外した。結果として得られる訓練用の各ビデオクリップには平均で 20 枚の画像が含まれている。

実装の詳細は割愛するが、提案手法によって生成されるアニメーションは、コントロールパラメータ $\mathbf{c} = \{c_{\mathcal{M}}^{type}, c_{\mathcal{M}}^{velocity}, c_{\mathcal{A}}^{type}, c_{\mathcal{A}}^{velocity}\}$ を用いて変えられるようなっている。$\mathbf{c}$ の各次元はそれぞれ、モーションの種類と速度、およびアピアランスの種類と速度を表す。訓練データから学習した潜在変数の次元削減及び内挿にもとづいて、ユーザはパラメータ $\mathbf{c}$ で潜在変数を操作できる。

Vol.2018-CG-172 No.2
Vol.2018-DCC-20 No.2
Vol.2018-CVIM-214 No.2
2018/11/7

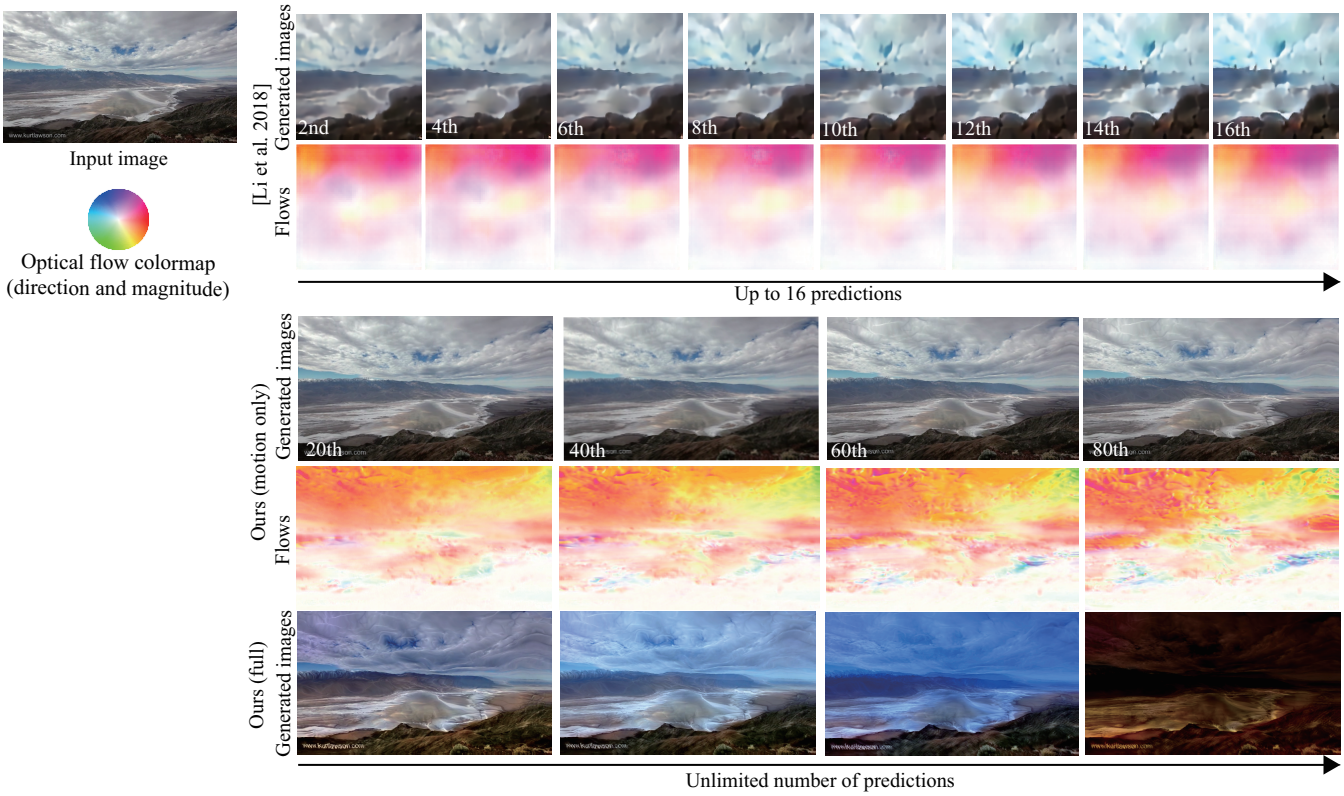

図 6　既存手法と提案手法によって生成された動画の比較。

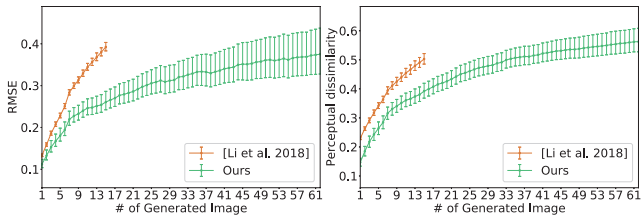

図 7　1 枚の入力画像から生成された動画の各フレームと ground-truth との定量的評価。

5.1　動画生成に対する既存手法との比較

提案手法の有効性を確認するために、生成された動画の質を既存手法と比較する。比較対象としては最新の動画生成手法である、3D 畳み込みを用いた流れ場予測と GAN による画像生成モデル [6] を用いる。訓練データにはタイムラプス動画データセット [19] を用い、この手法に必要な学習用の正解の流れ場は著者らと同じ SpyNet [12] を用いて作成している。

図 6 で定性的に結果を比べる。左側の入力画像に対して右側に各手法で予測した結果が示されている。右側 1、2 行目の画像はそれぞれ [6] によって最終的に生成された画像と、途中で生成された流れ場である。3、4 行目の画像はそれぞれ提案手法のモーション予測器で生成されたワーピング画像と流れ場である。まず 2、4 行目の流れ場について比べると、既存手法は粗い動きしか予測できていない。これに対して、提案手法はシーンの構造をふまえたうえで詳細に流れ場を予測している。例えば雲の領域と水面の領域で異なる動きで、さらに手前の岩場は静止している。1、3 行目の生成画像で比べると、既存手法は GAN による生成が上手くいっておらず、後の予測になるほど結果が抽象化されてしまっている。提案手法の結果はより高解像度にも関わらず、より鮮明かつ長期的なフレームの生成が可能である。最後に 5 行目に示す通り、提案手法はこの結果に線形ブレンドとアピアランス予測を適用することで、色味も変化するループ動画を生成できる。既存手法の結果はまとめて学習した色味の変化も確認できるものの質は低く、シネマグラフも生成できない。

テストデータ 224 シーンに対する定量的な評価も実施した。テストデータの各シーンにおける最初のフレームを入力として、以降 1 フレームおきの予測結果と、正解画像 (ground-truth) との違いを比べた。評価指標には、Li ら [6] と同様に RMSE および perceptual dissimilarity [21] を使った。RMSE は単に空間的に対応する画素値の差を測るのに対し、perceptual dissimilarity は人間の知覚をふまえたハイレベルな画像特徴を測ることができる。提案手法は 5.3 節で結果を示すように、コントロールパラメータ $c_{\mathcal{M}}^{type}$ によって異なる動きを予測できる。そこでパラメータを変更して、$\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ それぞれに対して結果を生成し、各パラメータで最も良い値、悪い値、平均を比較した。また既存手法 [21] は出力の不確かさを考慮するために VAE による学習を行っているため、正規分布から 6 パターン潜在変数をサンプリングすることで、同様に結

Vol.2018-CG-172 No.2
Vol.2018-DCC-20 No.2
Vol.2018-CVIM-214 No.2
2018/11/7

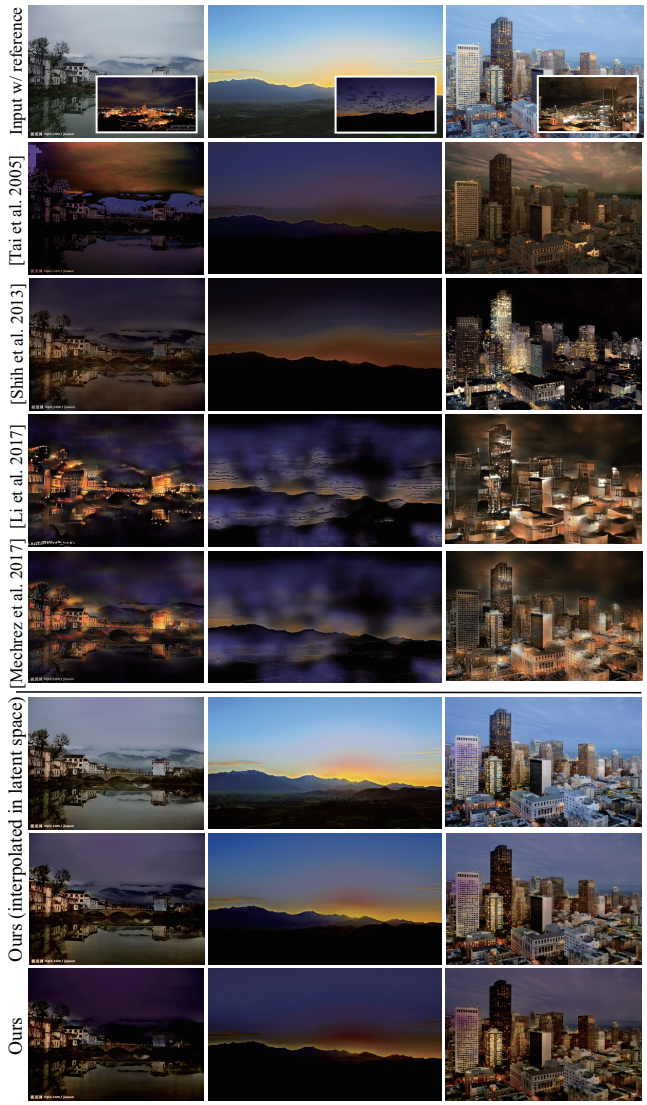

図 8 従来手法と提案手法に対するアピアランス予測の比較。1 行目が入力画像と参照画像、2-5 行目が既存手法の結果、6-8 行目が提案手法の結果を表す。

果をプロットした。

図 7 に RMSE と perceptual dissimilarity の結果を示す。実線が各パラメータで生成された結果の平均値、エラーバーの下側が最小値、上側が最大値を表す。横軸は生成された画像の順番であり、両方の評価指標とも遠い未来の画像になるほど悪くなる傾向にあるが、これは入力から時間的に遠いほど予測が難しいことを示唆している。既存手法は 16 枚の画像しか生成できないが、提案手法は 60 枚以上の画像を生成して評価値を算出した。既存手法と比べて提案手法は良好な数値を示しており、特に perceptual dissimilarity において大きさ差があることから、より正解画像と写実性の近い結果を生成できているといえる。また、提案手法はパラメータによる結果の変動を考慮しても、既存手法より良好な結果であることもわかる。

5.2 その他の比較

前節でニューラルネットワークを用いた動画生成の比較は実施したものの、既存手法は主にモーションの予測に重点を置いているため、アピアランス予測の有効性についてはさらに図 8 を通して関連手法と比較検証する。アピアランスの予測については、参照動画を利用した動画生成手法 [15] しか存在しない。そこで color transfer や style transfer のいくつかの手法 [5], [7], [16] とも比較した。各列のシーンの 1 行目の入力画像に対して、参照画像を使って各手法を適用した結果が 2 行目以降に示されている。[15] 手法は著者らと同じデータベースとシステムを構築するのは困難なので、プロジェクトページに公開されている結果画像を利用した。

まず、2 行目の color transfer [16] については色味の変化が全体的に単調になっている。4 行目の WCT を用いた style transfer [5] は色味の変化は豊かなものの、シーンの構造を崩してしまっている。これを緩和するために後処理で screened poisson [7] を適用して平滑化したものの、5 行目の通り良好な結果にはならなかった。3 行目の Shih らの手法 [15] は、入力シーンの構造を保持しながら自然に色味を変えられるが、前処理として入力画像と参照画像との対応をとる必要があり、このために参照画像と同じシーンかつ入力画像と同じアピアランスの画像を必要とする。対する提案手法は、8 行目に示す通り入力画像と参照画像をエンコードした 2 つの入力のみから、Shih らの結果に匹敵する画像を生成できる。なお、3 列目の結果については、提案手法が参照画像に近い色味を生成できなかった。しかし提案手法は本来参照画像すら必要とせず、6、7 行目の結果に示す通り学習済みの低次元潜在変数の内挿により滑らかなアピアランスの変化を推定できる。そのため、この例でもパラメータ調整によってより暗いアピアランスを再現でき、かつ Shih らの手法のように参照動画に依存したちらつきも抑えられる。

5.3 コントロールパラメータの影響

提案手法は予測の不確定性を考慮するために潜在変数を学習しているが、この潜在変数にもとづくパラメータを用いて出力を制御することで結果がどう変わるかを検証した。まずモーションの変化を、図 9 に示す。図からわかるように、$c_{\mathcal{M}}^{type}$ を 0 から 1 に変えるにつれて、流れの方向は無秩序に変わるのではなく滑らかに変化しているのがわかる。生成されたビデオにおいては、図中の赤丸箇所などからもわかるように、雲の動きは徐々に向きを変えており、どれも写実的なアニメーション効果が付与されている。一方、図 10 はアニメーションの種類 $c_{\mathcal{A}}^{type}$ を調節した結果であり、昼から夜への変化においても異なる結果が見て取れる。以上の結果から、提案手法はモーションやアピアランスの種類に関するパラメータをランダムに決めても良い

Vol.2018-CG-172 No.2
Vol.2018-DCC-20 No.2
Vol.2018-CVIM-214 No.2
2018/11/7

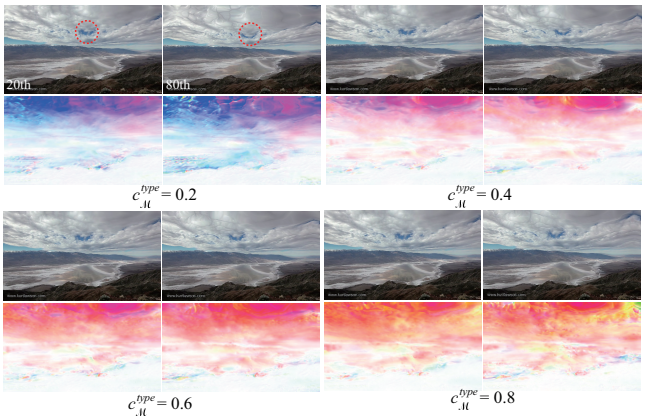

図 9 モーションのタイプをコントロールするパラメータ $c_{\mathcal{M}}^{type}$ を変えたときの結果の比較。赤丸はモーションの特徴的な箇所を示す。

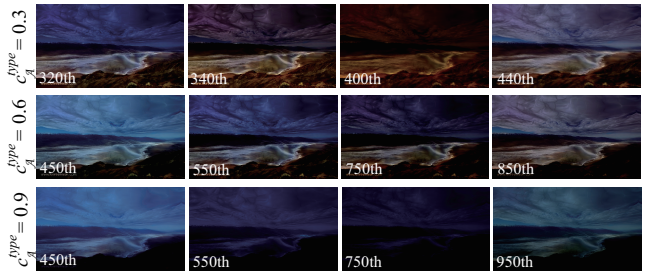

図 10 アピアランスのタイプをコントロールするパラメータ $c_{\mathcal{A}}^{type}$ を変えたときの結果の比較。

が、これを調節することで結果を調整できることも確認した。また、速度に関するパラメータ $c_{\mathcal{M}}^{velocity}$ や $c_{\mathcal{A}}^{velocity}$ の調整によって得られる結果は、外見が大きく変わるわけではないため割愛する。

6. おわりに

提案手法は動きと色の変化の 2 つの要素を非同期で別々のモデルに学習させることで、既存手法よりも多様な変化を扱え、かつ写実的なシネマグラフを生成できることを確認した。今後は本研究で扱った空や海のシーンのほかに、煙や炎などの流体に対する提案手法の適用可能性も検討したい。アピアランスについては 1 日の変化だけでなく年間を通しての季節 [4] などの時系列変化も扱えるように、データセットやモデルを拡張することも課題の 1 つである。

参考文献

[1] Y. Chuang, D. B. Goldman, K. C. Zheng, B. Curless, D. Salesin, and R. Szeliski. Animating pictures with stochastic motion textures. *ACM Trans. Graph.*, 24(3):853–860, 2005.

[2] R. Gao, B. Xiong, and K. Grauman. Im2flow: Motion hallucination from static images for action recognition. *CoRR*, abs/1712.04109, 2017.

[3] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[4] P. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph.*, 33(4):149:1–149:11, 2014.

[5] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. Yang. Universal style transfer via feature transforms. In *NIPS 2017*, pages 385–395, 2017.

[6] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Flow-grounded spatial-temporal video prediction from still images. In *European Conference on Computer Vision*, 2018.

[7] R. Mechrez, E. Shechtman, and L. Zelnik-Manor. Photo-realistic style transfer with screened poisson equation. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*, 2017.

[8] T. Oh, K. Joo, N. Joshi, B. Wang, I. S. Kweon, and S. B. Kang. Personalized cinemagraphs using semantic understanding and collaborative learning. In *ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5170–5179, 2017.

[9] M. Okabe, K. Anjyo, and R. Onai. Creating fluid animation from a single image using video database. *Comput. Graph. Forum*, 30(7):1973–1982, 2011.

[10] M. Okabe, Y. Dobashi, and K. Anjyo. Animating pictures of water scenes using video retrieval. *The Visual Computer*, 34(3):347–358, 2018.

[11] E. Prashnani, M. Noorkami, D. Vaquero, and P. Sen. A phase-based approach for animating images using video examples. *Comput. Graph. Forum*, 36(6):303–311, 2017.

[12] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *CVPR 2017*, pages 2720–2729, 2017.

[13] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *CoRR*, abs/1412.6604, 2014.

[14] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

[15] Y. Shih, S. Paris, F. Durand, and W. T. Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph.*, 32(6):200:1–200:11, 2013.

[16] Y. Tai, J. Jia, and C. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *CVPR 2005)*, pages 747–754, 2005.

[17] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *ICCV 2015*, pages 2443–2451, 2015.

[18] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR 2018*, 2018.

[19] W. Xiong, W. Luo, L. Ma, W. Liu, and J. Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *CVPR 2018*, 2018.

[20] M. Yeh. Selecting interesting image regions to automatically create cinemagraphs. *IEEE MultiMedia*, 23(1):72–81, 2016.

[21] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR 2018*, June 2018.

[22] J. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *NIPS 2017*, pages 465–476, 2017.