Vol.2022-HCI-199 No.33
2022/8/23

顔映像に対するリアルタイムな任意の動作追加

蔵内雄貴[1]　瀬古俊一[1]　山本隆二[1]

概要：笑顔や頷きといった動作の追加により、円滑な会議の進行が可能となると考えられる。このためには、顔映像に対してリアルタイムに任意の動作を追加する手法が必要となる。しかし、既存の動作転写モデルを用いる方法では、任意の表情への変換、任意の表情からの変換、複数動作の同時追加ができない問題がある。そこで、顔表情生成ツールで作成した顔画像を変換先のソース、表情認識結果と同じ表情の顔画像を動作の変換元のソースとし、複数の映像の特徴点の変化を合成することで、各課題を解決する手法を提案した。VoxCeleb データセットを用いた実験では、表情変換結果が変換先のソースと同じ表情となっているかについて、感情推定ツールを用いた定量的な評価と被験者による評価を行った。結果、定量評価では Accuracy が 0.39 から 0.49 に、被験者評価では 0.32 から 0.50 に改善し、提案法の有効性が示された。

キーワード：動作転写、動作追加、ディープフェイク

Real-time addition of actions to face images

YUKI KURAUCHI[†1]　SEKO SHUNICHI[†1]　RYUJI YAMAMOTO[†1]

1. はじめに

コミュニケーションにおいて表情や身振りといった動作は重要な要素である[1]。特に、聞き手の反応と関心が話し手の不安に大きな影響を与える[2]ことから、聞き手が笑顔や頷きなどの動作で反応し関心を示すことが重要である。これらの動作を用いてコミュニケーションに介入する研究もあり、互いに相手の表情を笑顔に変換することで創造性が向上する[3]ほか、頷くアバタを講義動画に追加することで講義を改善する[4]研究がある。以上のことから、web 会議において送付する映像を加工、または対面において AR グラスに表示する映像を加工するなどにより、聞き手に笑顔や頷きといった動作を追加することで、話し手の不安の軽減、ひいては会議の創造性の向上など、円滑な会議の進行が可能となると考えられる。本稿では、この際に必要となる、顔映像に対してリアルタイムに任意の動作を追加する手法について述べる。なお、本稿における動作とは、表情や身振りを含むものと定義する。

顔映像に対するリアルタイムな任意の動作追加を実現しうる手法として、リアルタイムな動作の転写[5, 6, 7]がある。これは、動作を追加される人物の画像または映像（以降、外見ソース）と追加したい動作を行っている人物の画像または映像（以降、動作ソース）の 2 つを入力とし、外見ソースの人物に対して動作ソースの動作を転写した映像をリアルタイムに出力するものである。転写する動作の特徴には、動作前の画像（以降、動作変換元ソース）から動作中の画像（以降、動作変換先ソース）への変化が用いられる。動作前後の画像は、同一人物である方が自然な動作の転写が可能であり、多くは同一の映像から切り出して用いられる。本研究で目指すリアルタイム映像への任意の動作追加を実現するためには、この手法において外見のソースをリアルタイムの映像、動作のソースを事前に用意した映像とし、追加したい動作ごとに映像を 1 つずつ用意しておけばよいと考えられる。しかしこの方法には、以下 3 つの課題があると考える。

1. 任意の表情への変換
2. 任意の表情からの変換
3. 複数動作の同時追加

そこで提案法（図 1）では、表情に関する動作ソースとして用いる顔画像について、顔表情生成ツール[8]を用いて生成した顔画像を用いることで任意の表情への変換を実現する。外見ソースに対して表情認識[9]を行い、その結果と同じ表情の顔画像を顔表情生成ツールで生成して動作変換元ソースとすることで、任意の表情からの変換を実現する。動作ソースを複数の事前に用意した映像としてそれぞれの特徴を抽出し、外見ソースに対して複数の動作を同時に追加することで複数動作の同時追加を実現する。このために、動作を転写する際に明示的に動作ソースの特徴点を抽出する手法[7]を用いる。そして、複数の動作ソースから抽出した特徴点の変化を、外見ソースの特徴点に対してベクトル合成する。VoxCeleb データセット[10]を用いた実験では、表情変換結果が動作変換先ソースと同じ表情となっているかについて、感情推定ツールを用いた定量的な評価と被験者による評価を行った。結果、定量評価では Accuracy が 0.39 から 0.49 に、被験者評価では 0.32 から 0.50 に改善し、提案法の有効性が示された。

本研究の貢献は以下の通りである。

1 NTT 人間情報研究所
NTT Human Informatics Laboratories

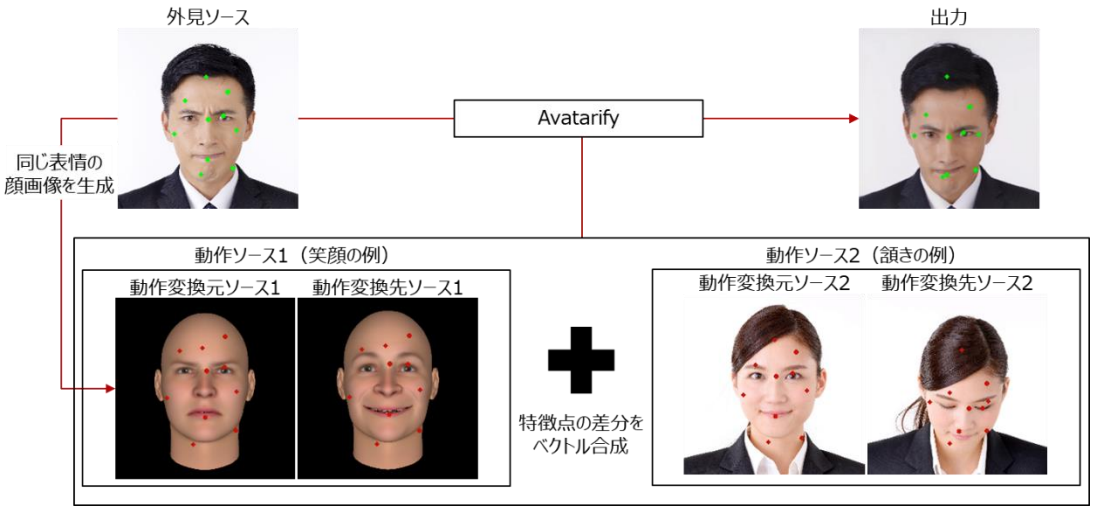

図 1　提案法の流れ。動作ソースが 2 つである場合。顔画像中のドットは Avatarify によって得られた特徴点を示す。
Figure 1　Flow of proposed method with two driving sources. Dots in face image indicate feature points obtained by Avatarify.

- 既存の動作転写モデルを用い、その外見ソースをリアルタイム映像、動作ソースを事前に用意した映像とすることで、リアルタイムに任意の動作を追加できることを確かめた
- 顔表情生成ツールで作成した顔画像を用いて動作転写モデルを再学習し、同顔画像を動作変換先ソースとすることで、任意の表情への変換ができることを確かめた
- 外見ソースに対して表情認識を行い、その結果と同じ表情の顔画像を動作変換元ソースとすることで、任意の表情からの変換ができることを確かめた
- 動作ソースを事前に用意した複数の映像とし、それらの特徴点の変化を、外見ソースの特徴点に対してベクトルとして合成することで、複数の動作を同時に追加できることを確かめた

以降、2 章で関連研究、3 章で提案手法、4 章で実験、5 章で結果と考察、6 章で結論について述べる。

2. 関連研究

2.1 コミュニケーションにおける動作

コミュニケーションにおける動作の役割や効果を調査した研究について述べる。Mehrabian ら[1]は、感情的に矛盾があるような視覚、聴覚、言語情報のメッセージを受け取ったとき、感情を判断する上で視覚情報による影響は 55%であると明らかにした。Macintyre ら[2]は、人前で話す際の不安のうち聴衆に起因するものは、聴衆の関心、反応、評価の 3 つであると明らかにした。さらに、このうち関心や反応は評価よりも重要であると明らかにした。以上から、コミュニケーションを行う上では聞き手が笑顔や頷きなどの動作で反応し関心を示すことが重要であると考え、本研究では表情や身振りといった動作に着目する。

我々と同様に動作に着目し、動作を用いてコミュニケーションに介入する研究がある。Smart Face [3]では、互いに相手の表情を笑顔に変換することで、ブレインストーミングにおける発案数が増加したとしている。藤井ら[4]は、オンデマンドの講義動画に対して頷く動作をとる学生のアバタを表示することで、生徒が学生間のかかわりや積極的な雰囲気を感じたとしている。これらは本研究のモチベーションを補助する研究であり、表情を笑顔だけでなく任意の表情に変換、およびアバタでなく実写の映像に頷きなどの身振りを追加することで、更なる効果が得られることが想定できる。

2.2 動作の追加

画像を加工し動作を追加する研究について述べる。先に述べた Smart Face [3]では、rigid moving least squares [11]を用いて顔画像を歪めて表情を笑顔に変換している。これは先に課題として挙げた複数の動作を追加する場合にも利用可能である。しかし、画像を歪める手法であるために、大きく表情を変化させると不自然になるほか、驚きの表情において口を開けるなどの変換には適用できない。

この問題を解決しうる方法に、深層学習を用いて動作を転写する研究がある。これは、外見ソースと動作ソースとして 2 つの画像または映像を入力とし、動作ソースから動作の特徴を抽出し、外見ソースの人物に対して動作を転写した映像を出力するものである。特徴抽出モデルと映像加工モデルが既に学習されているため、ある動作をさせたい場合に必要な映像は動作ごとに 1 つだけであり、動作ごとの大量の映像を必要としない特徴を持つ。Chan ら[12]は踊りなどの全身動作の転写を実現しているが、動作の特徴として骨格情報を用いているため、表情の転写には利用できない。表情を含む動作をリアルタイムに転写する研究には X2face [5]と Avatarify [7]がある。X2face は動作の特徴を明示的に得ず end2end で学習しているため、動作の特徴に加法性が無い。すなわち、複数動作の特徴を組み合わせるこ

とはできず、先に挙げた課題 3 を解決できない。動作ソースを変えて同じモデルを 2 回適用することも考えられるが、フレームレートが単純計算で 1/2 となり、リアルタイム性が損なわれると想定される。Avatarify は、動作ソースの特徴点を抽出し、動作前後での特徴点の変化を外見ソースに適用する。そのため、動作の特徴に加法性があり、複数動作の特徴を組み合わせることができる。本研究では、複数の動作を入力として組み合わせ、任意の動作追加を実現するために、動作転写手法として Avatarify を用いた。なお、このような動作転写のよくある利用方法には、有名人の画像に自分と同じ動作をさせ、アバタのように用いるなどがある。これは、外見ソースを有名人の画像、動作ソースを自分のリアルタイム映像とし、話している口の動きを含む表情や頷きなどの身振りを転写することで実現される。提案法では外見ソースをリアルタイム映像、動作ソースを事前に用意した映像としており、外見ソースと動作ソースを入れ替えて利用していると考えることができる。

いつどのような動作をするかをモデル化する研究では、過去数秒間の発話状態から頷きの有無およびその種類をモデル化した研究[13]や、発話内容とジェスチャの関係性をモデル化した研究[14]がある。本研究では画像を加工してリアルタイム映像に任意の動作を追加する手法について述べているが、このような動作のモデル化手法を併用することで、動作すべきタイミングで動作を追加できると考えられる。

3. 提案法

提案法で利用した Avatarify の詳細と提案法の詳細について順に述べる。

3.1 Avatarify

Avatarify は First order motion model [6]をもとに実装されており、外見ソース画像Sと、動作ソースとして動作変換先ソース画像Dと動作変換元ソース画像DI、計 3 つの入力を用いて変換を行う。処理の流れとしては、まずそれぞれのソースからスパースな数個の特徴点を抽出する。DからDIへの特徴点の変化をSに適用し、DからDIへの各特徴点間における局所的なアフィン変換を算出する。そしてこの特徴点と局所的なアフィン変換からピクセル単位のオプティカルフローとオクルージョンマップを求める。最後にこれらを用いて出力の画像を生成する。このうち、特徴点の抽出、オプティカルフローおよびオクルージョンマップの算出はともに U-Net [15]を用いて、画像の生成は StyleGAN [16]を用いて行われる。残りの特徴点の変化の適用と局所的なアフィン変換の算出について以降で述べる。

Avatarify では、抽象的な参照フレームRを仮定し、DIからDへの変換$\mathcal{T}_{\mathrm{DI}\leftarrow\mathrm{D}}$を、Rを介した変換$\mathcal{T}_{\mathrm{DI}\leftarrow\mathrm{R}}$と$\mathcal{T}_{\mathrm{R}\leftarrow\mathrm{D}}$に分解して独立に扱う。あるソースXにおける特徴点の座標は$\mathcal{T}_{\mathrm{X}\leftarrow\mathrm{R}}(p)$、うち$k$番目の特徴点の座標を$\mathcal{T}_{\mathrm{X}\leftarrow\mathrm{R}}(p_k)$で表す。Rからあるソース X への p_k における局所的なアフィン変換は $\frac{d}{dp}T_{\mathrm{X}\leftarrow\mathrm{R}}(p)\Big|_{p=p_k}$ で求められ、複数のアフィン変換の組み合わせはヤコビアンJ_kで表される。DIからDへの変換$\mathcal{T}_{\mathrm{DI}\leftarrow\mathrm{D}}$をSに合わせて変形してSに適用した特徴点$\mathcal{T}_{\hat{\mathrm{D}}\leftarrow\mathrm{R}}(p_k)$およびヤコビアン$J_k$は以下のように求められる。

$$\mathcal{T}_{\hat{\mathrm{D}}\leftarrow\mathrm{R}}(p_k) = \mathcal{T}_{\mathrm{DI}\leftarrow\mathrm{D}} \circ \mathcal{T}_{\mathrm{S}\leftarrow\mathrm{R}}(p_k)$$

$$J_k = \left(\frac{d}{dp}\mathcal{T}_{\mathrm{D}\leftarrow\mathrm{R}}(p)\Big|_{p=p_k}\right)\left(\frac{d}{dp}\mathcal{T}_{\mathrm{DI}\leftarrow\mathrm{R}}(p)\Big|_{p=p_k}\right)^{-1}\left(\frac{d}{dp}\mathcal{T}_{\mathrm{S}\leftarrow\mathrm{R}}(p)\Big|_{p=p_k}\right)$$

なお、学習においては、データセットにおける同一動画から外見ソースSおよび動作変換先ソースDを選び、SからDへの変換$\mathcal{T}_{\mathrm{S}\leftarrow\mathrm{D}}$をSに適用し再構築した$\hat{\mathrm{D}}$を得る。このDと$\hat{\mathrm{D}}$が近しくなるように学習を行う。

このとき、外見ソースに web カメラなどを用いてリアルタイムの映像を入力し、動作ソースに録画など事前に用意した映像を入力すれば、リアルタイムな任意の動作追加が可能となる。しかしこの場合、以下 3 つの課題がある。

1. 任意の表情への変換
2. 任意の表情からの変換
3. 複数動作の同時追加

各課題について順に詳細を述べる。

課題 1：表情を変換する際、任意の表情へと変換できることが好ましい。しかし、動作ソースを事前に用意した映像とした場合、変換したい表情に近い映像を大量の映像から、変換するたびに探す必要があり、現実的でない。

課題 2：表情を変換する際、外見ソースがどのような表情でも、任意の表情へと変換できることが好ましい。しかし、外見ソースの表情と、動作変換元ソースの表情が異なっていると、うまく表情を変換できない。例えば、外見ソースが怒った表情、動作変換元ソースが無表情、動作変換先ソースが笑顔だと、出力が怒りながら笑うような表情となってしまう。そのため、動作変換元ソースも外見ソースと同じく怒った表情とする必要がある。この際、先に述べたように、動作前後の画像は、同一人物の場合に変換精度が高い。

課題 3：動作を追加する際、表情を笑顔に変換し、かつ頷く身振りを追加するなど、複数の動作を同時に追加できることが好ましい。しかし、この例では動作ソースとして笑顔で頷いている映像が事前に必要となるなど、任意の動作をさせるためにはあらゆる動作の組合せの映像を事前に用意する必要があり、現実的でない。

3.2 提案法

提案法（図 1）では、上記 3 つの課題を解決するために以下のような方法をとる。

課題 1：表情に関する動作ソースとして用いる顔画像について、顔表情生成ツール[8]を用いて生成した顔画像を用いる。これにより、任意の表情の顔画像を生成して動作ソースとして任意の表情への変換を実現できる。このとき、元の表情変換モデルの学習データには生成した顔画像が含まれていないため、うまく変換できない。そこで、生成した顔画像を動作ソースとした場合の変換精度の向上のため、生成した顔画像を用いて動作転写モデルを再学習する。
課題 2：外見ソースに対して表情認識[9]を行い、その結果と同じ表情の顔画像を顔表情生成ツールで生成し、動作変換元ソースとする。これにより、動作変換元ソースと外見ソースの表情と、動作変換元ソースと動作変換先ソースの人物を同一とでき、任意の表情からの変換が実現できる。
課題 3：動作ソースを複数の事前に用意した映像としてそれぞれの特徴を抽出し、外見ソースに対して複数の動作を同時に追加する方針をとる。例えば、笑顔の動作ソースから笑顔の特徴を、頷く動作ソースから頷きの特徴をそれぞれ抽出し、外見ソースに笑顔で頷く動作を追加する。この実現のため、動作を転写する際に明示的に動作ソースの特徴点を抽出する Avatarify を用いる。そして、複数の動作ソースから抽出した特徴点の変化を、外見ソースの特徴点に対してベクトル合成する。すなわち、外見ソース S と、動作ソースとしてN個の動作変換先ソース$\mathrm{D}_1, \ldots \mathrm{D}_\mathrm{N}$および動作変換元ソース$\mathrm{DI}_1, \ldots \mathrm{DI}_\mathrm{N}$、計$2\mathrm{N}+1$個の入力を用いて変換を行う。うちn番目の動作変換先ソースをD_n、各動作変換元ソースをDI_nと表す。DI_nからD_nへの変換$\mathcal{T}_{\mathrm{DI}_\mathrm{n}\leftarrow\mathrm{D}_\mathrm{n}}$を、R を介した変換$\mathcal{T}_{\mathrm{DI}_\mathrm{n}\leftarrow R}$と$\mathcal{T}_{R\leftarrow\mathrm{D}_\mathrm{n}}$に分解し、Sに合うよう変形してSに適用する。適用後の特徴点$\mathcal{T}_{\hat{\mathrm{D}}\leftarrow\mathrm{R}}(p_k)$および n回のアフィン変換におけるヤコビアン$J_{k,n}$は以下のように求められる。

$$\mathcal{T}_{\hat{\mathrm{D}}\leftarrow\mathrm{R}}(p_k) = \mathcal{T}_{\mathrm{DI}_\mathrm{N}\leftarrow\mathrm{D}_\mathrm{N}} \circ, \ldots \left(\mathcal{T}_{\mathrm{DI}_\mathrm{n}\leftarrow\mathrm{D}_\mathrm{n}} \circ, \ldots \left(\mathcal{T}_{\mathrm{DI}_0\leftarrow\mathrm{D}_0} \circ \mathcal{T}_{\mathrm{S}\leftarrow\mathrm{R}}(p_k)\right)\right)$$

$$J_{k,n} = \left(\frac{d}{dp}T_{\mathrm{D}_\mathrm{n}\leftarrow\mathrm{R}}(p)\Big|_{p=p_k}\right)\left(\frac{d}{dp}T_{\mathrm{DI}_\mathrm{n}\leftarrow\mathrm{R}}(p)\Big|_{p=p_k}\right)^{-1} J_{k,n-1}$$

$$J_{k,0} = \left(\frac{d}{dp}T_{\mathrm{D}_0\leftarrow\mathrm{R}}(p)\Big|_{p=p_k}\right)\left(\frac{d}{dp}T_{\mathrm{DI}_0\leftarrow\mathrm{R}}(p)\Big|_{p=p_k}\right)^{-1}\left(\frac{d}{dp}T_{\mathrm{S}\leftarrow\mathrm{R}}(p)\Big|_{p=p_k}\right)$$

提案法では外見ソースをリアルタイム映像、動作ソースを事前に用意した映像としており、Avatarify と比べて外見ソースと動作ソースを入れ替えて利用しているが、この場合でも Avatarify と同様にリアルタイムに動作する。また、提案法を用いて複数の動作ソースを適用する場合と各動作ソースについて Avatarify を適用することを複数回繰り返す場合で、結果は同一となる。

表 1　FaceGen データの生成に用いた Action Unit
Table 1 Action Units used for generating FaceGen data.

AU01 Inner Brow Raiser	AU12 Lip Corner Puller
AU02 Outer Brow Raiser	AU13 Sharp Lip Puller
AU04 Brow Lowerer	AU14 Dimpler
AU05 Upper Lid Raiser	AU15 Lip Corner Depressor
AU06 Cheek Raise	AU16 Lower Lip Depressor
AU07 Lid Tightener	AU20 Lip Stretcher
AU08 Lips Toward Each Other	AU22 Lip Funneler
AU09 Nose Wrinkler	AU25 Lips Parted
AU10 Upper Lip Raiser	AU27 Mouth Stretch
AU11 Nasolabial Deepener	AU41 Lid Droop

4. 実験

表情変換結果が動作変換先ソースと同じ表情となっているかを確かめるため、感情推定ツールを用いた定量的な評価と、被験者を用いた評価を行った。

4.1 データセット

データセットには、実際の人間の顔画像データと、人工的なアバタの顔画像データを用いた。

実際の人間の顔画像データとして、YouTube から抽出された顔映像データセットである VoxCeleb [10]を用いた。収集できたのは学習用の 1,211 人ぶんの 18,827 動画、テスト用の 40 人ぶんの 592 動画だった。これらに対し、顔部分を解像度 256×256 で切り取った[17]。その結果、得られたデータ数は、学習用が 98,150 個、テスト用が 3,494 個となった。変換元の表情ごとに評価を行うため、テスト用データについては、感情推定ツールにおいて推定された表情ごとに均等に選んだ 500 枚を用いた。感情推定ツールには MediaGnosis [9]を用いた。MediaGnosis は、顔画像を入力として negative、happy、sad、surprise、neutral の 5 つの表情の確率値を出力する。テスト用データの選び方は、同じ動画内のフレームを使わずに、各感情に対して推定確率の大きい順に 100 個ずつ、計 500 個を選ぶものとした。

人工的なアバタの顔画像データは、顔表情生成ツール FaceGen [8]を用いて作成した。このデータはテスト用としては用いず、学習用としてのみ用いた。400 人ぶん、それぞれ 33 種の表情の画像、計 13,200 枚を生成した。400 人の顔は、FaceGen の機能を用いて生成した。生成の際の設定値は、Racial Group と Gender を Any、Random variance の Symmetric Shape、Symmetric Shape、Symmetric Color を Typical とした。33 種の表情は、8 つの感情における 4 段階の強さおよび無表情の計 33 個とした。8 つの感情には、基本 6 感情[18]である anger, disgust, fear, sad, smile, surprise およびラッセルの円環[19]における縦横軸である arouse および pleasure を用いた。各表情の顔画像の生成は、FaceGen の機能を用いて表情パラメタである Action Unit [20]を設定することで行った。Action Unit は、上記 8 感情と関連する 20 個（表 1）に絞って用いた。33 個の表情に対する Action Unit の値は、ガウス過程選好学習[21]を用いて、どのような Action Unit の値に対してどのような感情の強さと人が

happy negative neutral sad surprise

図 2 動作変換先および動作変換元ソースに用いた顔画像
Figure 2 Face image used for driving and driving initial source.

感じるかをモデル化することで作成した。作成した画像は顔の大きさと位置が固定であるため、これらを変化させた。具体的には、顔の大きさは 0.7〜1.0 倍、顔の位置は顔が全て画像内に収まる範囲で移動させた。

動作変換先ソース画像および動作変換元ソース画像には、MediaGnosis の出力する 5 つの感情（happy、negative、sad、surprise、neutral）において最も各感情と出力する確率が高い 5 つの人工的なアバタの顔画像を用いた（図 2）。5 つの顔表情およびその Action Unit の値は、人工的なアバタの顔画像の生成と同様の設定において、どのような Action Unit の値に対してどのような確率値を Mediagnosis が出力するかをモデル化することで作成した。

4.2 再学習における設定

再学習におけるパラメタは、epoch 数は 150、バッチサイズは 4、学習率は 0.0002、最適化関数は Adam を用いた。cpu はインテル Core i9-10980HK、gpu は NVIDIA GeForce RTX 3070 を搭載した PC を用いた。

損失関数は、Avatarify の実装[7]と同じく、Reconstruction loss、Equivariance constraint loss [6]、Least squares、Feature matching loss [22]の 4 つを組み合わせて用いた。

Reconstruction loss: 変換前後の画像の、画像特徴における loss。画像特徴を得る方法として、学習済の VGG-19 [23]を用いた。動作変換先ソース D と再構築した $\hat{\mathrm{D}}$ の間の Reconstruction loss を、

$$L_{rec}(\hat{\mathrm{D}},\mathrm{D}) = \sum_{i=1}^{I} \left| N_i(\hat{\mathrm{D}}) \ - N_i(\mathrm{D}) \right|$$

と表す。ただし、$N_i(\cdot)$は VGG-19 で得られたi番目のチャネルにおける特徴、I はこの層におけるチャネルの数を示す。

Equivariance Loss: 変換前後の特徴点およびヤコビアンの loss。あるソース X からあるソース Y へと変換する際の特徴点およびヤコビアンについての制約を、

$$\mathcal{T}_{\mathrm{X}\leftarrow\mathrm{R}}(p_k) \equiv \mathcal{T}_{\mathrm{X}\leftarrow\mathrm{Y}} \circ \mathcal{T}_{\mathrm{Y}\leftarrow\mathrm{R}}(p_k)$$

$$\mathbb{1} \equiv \left(\frac{d}{dp}T_{X\leftarrow R}(p)\Big|_{p=p_k}\right)^{-1}\left(\frac{d}{dp}T_{X\leftarrow Y}(p)\Big|_{p=T_{Y\leftarrow R}(p_k)}\right)\left(\frac{d}{dp}T_{Y\leftarrow R}(p)\Big|_{p=p_k}\right)$$

と表す。ただし、$\mathbb{1}$ は 2×2 の単位行列である。Equivariance loss には、この 2 式における L1 loss である$L_{\mathrm{equi}}^{\mathrm{key}}$、$L_{\mathrm{equi}}^{\mathrm{jaco}}$を用いた。

Least-square loss: GAN を用いた画像の生成において、Discriminator が生成した画像であるかを正しく判別できるか、Generator が生成した画像が入力した画像と比べて生成した画像らしさがあるかの loss。生成部における Discriminator であるG_{disc}と Generator であるG_{gen}における Loss を、

$$L_{\mathrm{gan}}^{G_{disc}}(G_{disc}) = \mathbb{E}_{\mathrm{D}\in\mathrm{D_{all}}}\left[\left(G_{disc}\left(\mathrm{D}\oplus\mathcal{T}_{\mathrm{D}\leftarrow\mathrm{R}}(p)\right)-1\right)^2\right] + \mathbb{E}_{(\mathrm{S,D})\in\mathrm{D}_{all}^2}\left[G_{disc}\left(\hat{\mathrm{D}}\oplus\mathcal{T}_{\mathrm{D}\leftarrow\mathrm{R}}(p)\right)^2\right]$$

$$L_{\mathrm{gan}}^{G_{gen}}(G_{gen}) = \mathbb{E}_{(\mathrm{S,D})\in\mathrm{D}_{\mathrm{all}}^2}\left[\left(G_{disc}\left(\hat{\mathrm{D}}\oplus\mathcal{T}_{\mathrm{D}\leftarrow\mathrm{R}}(p)\right)-1\right)^2\right]$$

と表す。ただし、$\mathrm{D_{all}}$は 1 つの動画を、⊕はチャネル軸に沿った結合を表す。

Feature matching loss: 変換前後の画像の、Discriminator の中間層における特徴の loss。

$$L_{\mathrm{feat}} = \mathbb{E}_{(\mathrm{S,D})}\left[\left\| G_{disc_i}\left(\hat{\mathrm{D}}\oplus\mathcal{T}_{\mathrm{D}\leftarrow\mathrm{R}}(p)\right) - G_{disc_i}\left(\mathrm{D}\oplus\mathcal{T}_{\mathrm{D}\leftarrow\mathrm{R}}(p)\right)\right\|_1\right]$$

ただし、G_{disc_i}は Discriminator におけるi番目の層の特徴を示す。G_{disc_0}は Discriminator への入力を示す。

これらの loss について、以下のように結合し用いた。

$$L_{\mathrm{total}} = \lambda\left(L_{\mathrm{rec}} + L_{\mathrm{equi}}^{\mathrm{key}} + L_{\mathrm{equi}}^{\mathrm{jaco}} + L_{\mathrm{feat}}\right) + L_{\mathrm{gan}}^{G_{disc}} + L_{\mathrm{gan}}^{G_{gen}}$$

ただし、Avatarify の実装[7]と同じく$\lambda = 10$とした。

4.3 定量評価の手続き

4.1 節にて述べた 5 つの感情の顔画像（図 2）のうち、neutral を除く negative、happy、sad、surprise の 4 つを動作変換先ソース画像として変換し、この変換結果に対して感情推定ツール MediaGnosis を適用する。そして、動作変換先ソース画像における感情と、変換結果に対する感情推定結果が一致しているかによって評価を行った。評価指標には、accuracy と macroAUC を用いた。

4.4 実験 1

動作ソースとして FaceGen で生成した顔画像を用いる際、

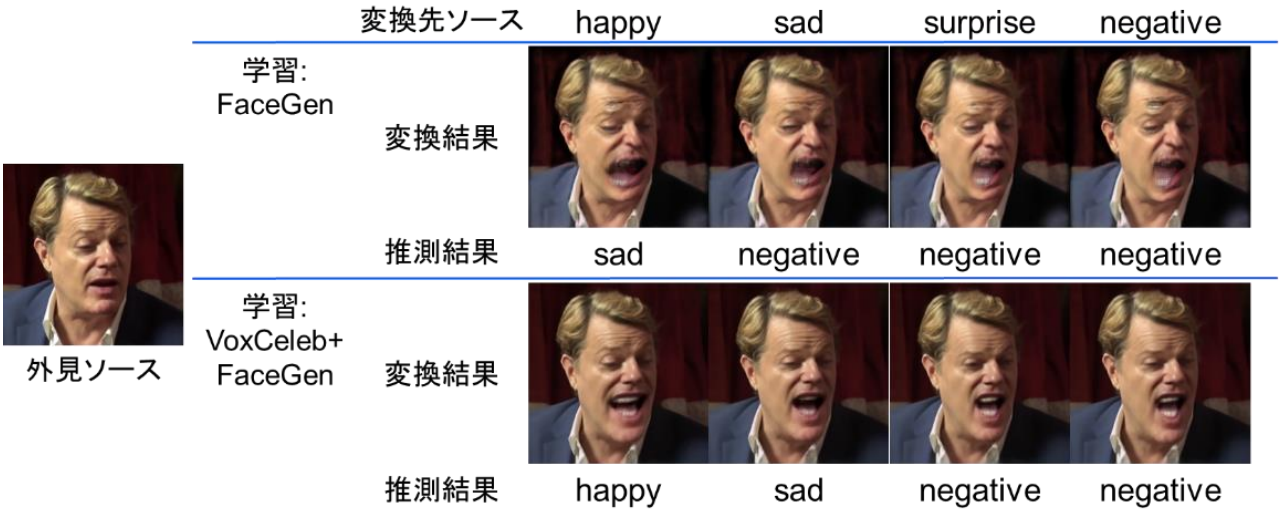

図 3　実験 1 の実行結果の例
Figure 3 Example of result of experiment 1.

どのようなデータセットで動作転写モデルを再学習すれば精度高く変換できるかを確かめるため、以下の 3 種類のデータセットで再学習した場合の精度の比較を行った。

VoxCeleb: VoxCeleb データのみを用いて再学習 (Avatarify の実装をそのまま利用) した場合

FaceGen: FaceGen データのみを用いて再学習した場合

VoxCeleb+FaceGen: VoxCeleb データと FaceGen データをともに用いて再学習した場合

VoxCeleb データのみを用いると、人工的なアバタの顔画像を動作ソースとして設定した際の変換がうまくできないという想定、FaceGen データのみを用いると、実際の人間の顔画像データを外見ソースとして設定した際の変換がうまくできないと想定し、上記の設定とした。このとき、動作変換元ソースは neutral の顔画像(図 2 中央)で固定した。

4.5 実験 2

外見ソースに対して表情認識を行い、その結果と同じ表情の顔画像を顔表情生成ツールで生成し、動作変換元ソースとすることで、どの程度変換精度が改善するかを確かめるため、以下 2 つの方法の比較を行った。

Without DI: 動作変換元ソースに neutral の顔画像(図 2 中央)を設定し変換

With DI: 動作変換元ソースに外見ソースの表情認識結果と同じ表情の画像(図 2)を設定し変換

このとき、動作転写モデルは VoxCeleb+FaceGen で再学習したものを用いた。

また、変換の自然さと、感情推定ツールだけでなく人が正しく感情を認識できる変換となっているかを評価するため、被験者による評価を行った。13 名の被験者に対し、前者を「1:自然でない～7:自然である」の 7 択、後者を動作変換先ソースとして設定した 4 つの感情「happy、negative、sad、surprise」のいずれであるかを 4 択で回答いただいた。without DI と with DI の 2 つの手法について、それぞれを 4 つの感情へと変換する 8 つのパターンについて、実際の人間の顔画像データに変換を適用した結果からランダムに 5 枚ずつを選んだ計 40 枚を用いた。

表 2　実験 1 結果
Table 2 Result of experiment 1.

データセット	Accuracy	macroAUC
VoxCeleb	0.39	0.65
FaceGen	0.38	0.62
VoxCeleb+FaceGen	**0.41**	**0.67**

表 3　実験 1 および実験 2 の混同行列
Table 3 Confusion matrix of experiment 1 and 2.

データセット	動作変換先ソースの感情推定結果	変換後の感情推定結果 happy	negative	sad	surprise	(参考) neutral	(参考) None	total
VoxCeleb	happy	295	77	55	48	9	16	500
	negative	86	219	54	104	13	24	500
	sad	139	155	81	91	9	25	500
	surprise	95	194	38	136	16	21	500
	total	615	645	228	379	47	86	2,000
FaceGen	happy	233	66	109	24	28	40	500
	negative	61	260	99	22	32	26	500
	sad	106	151	146	28	37	32	500
	surprise	48	273	100	26	31	22	500
	total	448	750	454	100	128	120	2,000
VoxCeleb+ FaceGen (without DI)	happy	362	49	52	15	4	18	500
	negative	97	301	55	25	6	16	500
	sad	143	206	92	31	5	23	500
	surprise	83	338	25	29	8	17	500
	total	685	894	224	100	23	74	2,000
VoxCeleb+ FaceGen (with DI)	happy	380	19	38	26	26	11	500
	negative	52	215	49	54	116	14	500
	sad	110	97	87	28	163	15	500
	surprise	42	232	21	61	128	16	500
	total	584	563	195	169	433	56	2,000

凡例: 400 / 300 / 200 / 100 / 0

5. 結果と考察

5.1 実験 1

結果は表 2 のようになった。Accuracy と macroAUC の両指標において、FaceGen データのみを用いて再学習した場合は VoxCeleb データのみを用いて再学習する場合よりも精度が下がるが、VoxCeleb データと FaceGen データをともに用いて再学習することにより、最も精度が高くなることがわかる。

図 5　実験 2 の実行結果の例
Figure 5 Example of result of experiment 2.

表 4　実験 2 定量評価結果
Table 4 Result of quantitative experiment 2.

手法	Accuracy	macroAUC
Without DI	0.41	0.67
With DI	**0.49**	**0.69**

行を動作変換先ソースの感情、列を変換後の感情推定結果とした混同行列は表 3 の通りである。左上から右下にかけての対角線上の値が正しく推定できたデータ数を意味する。列の変換後の感情推定結果における None は感情推定ツールが感情の推定をできなかったことを表す。定性的には、FaceGen データのみを用いて再学習した場合は表情が崩れている（図 3 上部）ことが多いことから、None が減っているのは、表情が崩れるような変換が減っているためだと考えられる。VoxCeleb では sad への変換精度が低い傾向があり、FaceGen では surprise への変換精度が低い傾向がある。VoxCeleb+FaceGen ではこれらに影響されて sad および surprise への変換精度が低い傾向だった。原因の考察には更なる検討が必要だが、感情推定ツールの出力の偏りなどの影響も考えられる。しかし、全体としては精度が向上したことから、任意の表情への変換ができたと言える。

実行結果の例を図 3 に示す。最も精度の差が大きい FaceGen と VoxCeleb+FaceGen を比較した。FaceGen では、口元に違和感があるほか、額にも口があるような変換が行われている。VoxCeleb+FaceGen では、そのような違和感のある変換は行われていない。変換後の感情推定結果を見ても、VoxCeleb データと FaceGen データをともに用いて再学習することで、より正しく推定されていることがわかる。

以上から、VoxCeleb データと FaceGen データをともに用いて再学習することにより、それぞれのデータのみで学習した場合よりも変換精度が向上し、任意の表情への変換ができることが定量的、定性的に確認できた。

表 5　変換前の感情推定結果ごとの実験 2 の Accuracy
Table 5 Accuracy of experiment 2 for each emotion estimated before conversion.

変換前の感情推定結果	without DI	with DI
happy	0.40	**0.55**
negative	0.42	**0.50**
sad	0.43	**0.54**
surprise	0.39	**0.49**

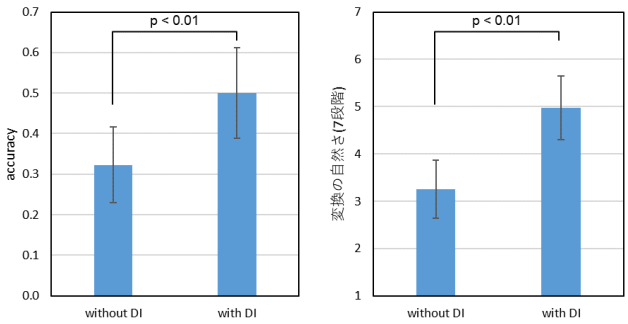

図 4　実験 2 の被験者実験結果。バーは標準偏差を表す。
Figure 4 Result of subject experiment 2. Bar indicate standard deviation.

5.2 実験 2

定量的な評価の結果は表 4 のようになった。Accuracy と macroAUC の両指標において、動作変換先ソースを設定することで、精度が向上することがわかる。

表 3 の混同行列を見ると、None が減っていることから、動作変換先ソースを設定することで、表情が崩れる変換が減っていると考えられる。neutral が増加していることから、動作変換先ソースを設定することで、変換の際の表情の変化が減っていることが想定される。これは、neutral からの変換よりも、何らかの表情からの変換の方が顔の特徴点における変化が少なかったためだと考えられる。その他の項目においては精度が向上していることから、動作変換先ソースを設定することで、任意の表情への変換の精度が向上したと言える。

変換前にあたる動作変換元ソースの感情ごとの accuracy は表 5 の通りである。全ての感情からの変換において精度が向上していることから、動作変換先ソースを設定することで、任意の表情への変換だけでなく、任意の表情からの変換の精度が向上したと言える。

被験者による評価の結果は図 4 のようになった。被験者実験においても、動作変換先ソースを設定することで、精度が向上したことが見てとれる。感情推定ツールによる評価と比べて、被験者による評価において精度の向上幅が大きい傾向であった。また、変換の自然さについても、動作変換先ソースを設定することで向上したことがわかる。

実行結果の例を図 5 に示す。この例では感情推定ツールによって sad と推定された画像を外見ソースとして設定している。withoutDI では、例えば happy への変換において、neutral から happy への変換を sad の画像に対して適用しているため、泣き笑いのような表情となってしまっている。withDI では、sad から happy への変換を sad の画像に対して適用しているため、自然な笑顔へと変換できている。

以上から、動作変換先ソースを設定することで、設定しない場合よりも変換精度が向上し、任意の表情から任意の表情への変換ができることが定量的､定性的に確認できた。

6. 結論

本稿では、顔映像に対してリアルタイムに任意の動作を追加する手法を提案した。既存の動作転写モデルを用いる方法には、任意の表情への変換、任意の表情からの変換、複数動作の同時追加ができないという 3 つの課題があった。そこで提案法では、顔表情生成ツールで作成した顔画像を動作変換先ソース、表情認識結果と同じ表情の顔画像を動作変換元ソースとし、複数の映像の特徴点の変化を合成することで、それぞれの課題を解決した。

VoxCeleb データセットを用いた実験では、表情変換結果が動作変換先ソースと同じ表情となっているかについて、感情推定ツールを用いた定量的な評価と被験者による評価を行った。結果、定量評価では Accuracy が 0.39 から 0.49 に、被験者評価では 0.32 から 0.50 に改善し、提案法の有効性が示された。

参考文献

[1] Mehrabian, A.. Silent messages. Wadsworth, Belmont, California. 1971.
[2] Macintyre, P. D., Thivierge, K. A., and MacDonald, J. A.. The effects of audience interest, responsiveness, and evaluation on public speaking anxiety and related variables. Communication research reports, 1997.
[3] Nakazato, N., Yoshida, S., Sakurai, S., Narumi, T., Tanikawa, T., and Hirose, M.. Smart Face: enhancing creativity during video conferences using real-time facial deformation. Conference on Computer supported cooperative work & social computing (CSCW). 2014.
[4] 藤井 亮哉、広瀬 隼人、青柳 西蔵、山本 倫也。うなずく学生キャラクタがかかわりを実感させるオンデマンド授業の配信。ヒューマンインタフェース学会論文誌、Vol. 23、No. 3、2021。
[5] Wiles, O., Koepke, A., and Zisserman, A.. X2face: A network for controlling face generation using images, audio, and pose codes. In Proceedings of the European conference on computer vision (ECCV), 2018.
[6] Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., and Sebe, N.. First order motion model for image animation. Advances in Neural Information Processing Systems (NeurIPS), 2019.
[7] Aliev, A., Avatarify, https://github.com/alievk/avatarify-python (accessed 2022/07/27).
[8] Singular Inversions, “FaceGen Modeller”, https://facegen.com/modeller.htm, (accessed 2022/07/27).
[9] Takashima, A., Makishima, N., Ihori, M., Tanaka, T., Orihashi, S., and Masumura, R.. Unsupervised Domain Adversarial Training in Angular Space for Facial Expression Recognition. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). 2020.
[10] Nagrani, A., Chung, J. S., and Zisserman, A.. VoxCeleb: a large-scale speaker identification dataset, INTERSPEECH, 2017.
[11] Schaefer, S., McPhail, T., and Warren, J.. Image deformation using moving least squares. ACM Transactions on Graphics (TOG), 2006
[12] Chan, C., Ginosar, S., Zhou, T., and Efros, A. A.. Everybody dance now. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
[13] Giannopulu, I., and Watanabe, T.. Give Children toys robots to educate and/or neuroreeducate: the example of PEKOPPA. Springer New Trends in Medical and Service Robots. 2016.
[14] Ahuja, C., Lee, D., Ishii, R., and Morency, L. P.. No gestures left behind: Learning relationships between spoken language and freeform gestures. Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP). 2020.
[15] Ronneberger, O., Fischer, P., and Brox, T.. U-net: Convolutional networks for biomedical image segmentation. Springer International Conference on Medical image computing and computer-assisted intervention, pp. 234-241, 2015.
[16] Karras, T., Laine, S., and Aila, T.. A style-based generator architecture for generative adversarial networks. The IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp. 4401-4410, 2019.
[17] Siarohin, A., video-preprocessing, https://github.com/AliaksandrSiarohin/video-preprocessing
[18] Ekman, P., and Friesen, W. V.. Pictures of facial affect, Palo Alto: Consulting Psychologist. Press, 1976.
[19] Russell, J. A.. A circumplex model of affect, Journal of Personality and Social Psychology, vol. 39, no. 6, pp. 1161-1178, 1980.
[20] Ekman, P., and Friesen, W. V.. Facial action coding system, Environmental Psychology & Nonverbal Behavior, 1978.
[21] Chu, W., and Ghahramani, Z.. Preference learning with Gaussian processes, ACM international conference on machine learning (ICML), 2005.
[22] Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., and Sebe, N.. Animating Arbitrary Objects via Deep Motion Transfer. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[23] Simonyan, K., and Zisserman, A.. Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations (ICLR), 2014