

# アニメーション制作支援のための距離変換を用いた線画フレームの補間 Inbetweens between line drawings of consecutive frames using distance transformation for animation production support

成田 嶺\* 平川 恵悟\*† 相澤 清晴\*  
Rei Narita Keigo Hirakawa Kiyoharu Aizawa

## 1. まえがき

世界ではディズニー映画のような 3DCG を用いたアニメーションが主流になっている中、日本では現在でも 2D の手書きアニメーションが多く見られている。手書きアニメーションでは、まず「原画」と呼ばれる動きの要所となるようなフレームを描き、その後、「中割り」と呼ばれる、動画が滑らかに見えるように原画間を補間するフレームを描く。しかし、この中割りの作業は大きな負担となっており、アニメーターの労働環境悪化につながっている。この中割り作成の過程で技術的なサポートを行うことが出来れば、アニメーション制作の過程を大きく削減することが可能となる。

この中割り生成を目的として、数多くのフレーム間の対応、補間方法が考案されてきた [10] [16]。しかし、このタスクは依然として、チャレンジングな課題である。類似しているタスクとして動画のフレーム間対応、補間があるが、それと比べて線画はテキストチャなどの情報量が少ない、輝度勾配が無い、フレーム間の距離が離れている場合が多く、対応が取りづらく、オクルージョンが発生しやすい、対応が取れたとしても線形補間では難しい場合もある、などの理由で難しいとされる。また、ストロークをベクタライズして処理を行うために、専用ソフトウェア上で作業を行う必要がある研究がほとんどである。

そこで、本論文では今回は線画をラスタ画像のまま用いる手法として、距離変換を用いて線画を変換した後、optical flow を計算し、それを用いて線画フレームの補間を行う手法を提案する。本来、optical flow は輝度勾配のない線画には適さない手法だが、距離変換により勾配を持たせることで、自然動画と同様に補間を行うことができると考えた。chamfer distance を評価指標とし、実験により提案手法と距離変換を用いないベスライン手法との補間結果の評価を行った。また実験結果から、今後の研究についての考察を行った。

## 2. 関連研究

### 2.1. 動画のフレーム間の対応と補間

動画のフレーム間の対応、補間研究は古くからおこなわれてきたが、そのなかでも optical flow を利用した手法が数多くみられる。optical flow とは 2 つの画像間で各画素の動きをベクトルで表したものであり、各フローの中間点を取ることでフレーム補間が可能である。optical flow の精度を評価がフレーム補間のタスクで行われる [4] ほど、両者は密接な関係にあるといえる。optical flow を推定するためのアルゴリズムは数多

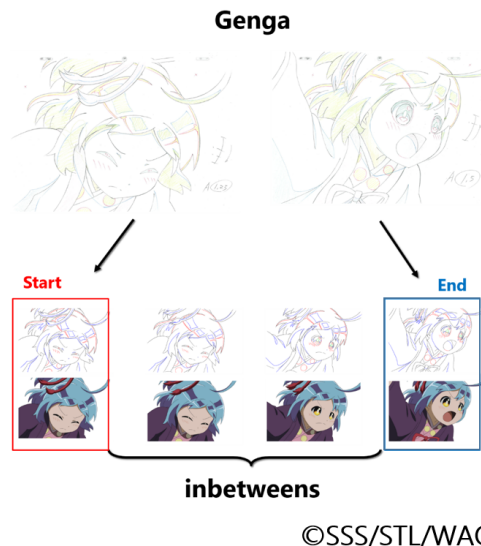


Fig. 1: 原画と中割りの例。このケースでは 2 枚の原画を清書した線画の間に、2 枚の中割りが生成されている。[2]

く提案されてきたが、近年では画像処理分野で大きな成果を上げていた CNN を用いた手法が提案され、こちらでも成果を上げている。

Dosovitskiy ら [8] は、end-to-end で学習することが出来、optical flow を出力する CNN, FlowNet を提案した。Ilg ら [9] は、さらにこれの発展形として FlowNet2 を提案した。しかし、学習の際に optical flow の教師データが必要のため、新しい動画ドメインに fine-tune しようと思った際の障壁になる恐れがあるという問題がある。

optical flow ではなく、直接に補間フレームを出力する手法も増えている。Liu ら [12] の提案した手法では CNN の出力は Deep Voxel Flow と呼ばれる、optical flow に時間成分を加えた 3 次元の flow  $F$  であり、これと入力 of 2 枚の画像  $X$  を用いて補間フレーム  $Y$  が合成される。flow の教師データを必要としないためにあらゆる動画を学習用データとすることが可能である。Niklaus ら [14] は CNN の出力を flow を表現するカーネルとし、対応する入力画像と畳み込むことで補間フレームを出力するネットワークを提案している。

### 2.2. アニメーションの中割りについての研究

アニメーションのフレーム間の対応、補間についても古くから研究が行われてきた [10] [16]。手法の多くはフレーム間のストロークや特徴点のマッチングを取

\*東京大学大学院 情報理工学系研究科 電子情報学専攻

†University of Dayton

る手法が多く、機械学習ベースのものはあまりなかったが、deep learning を使用した手法も登場している。

ストロークの対応を取る手法として、de Juan ら [7] はキャラクターの輪郭のみを抽出し、フレーム間で同じキャラの輪郭を放射基底関数によって補間する手法を提案した。輪郭の内部は elastic image registration [1] という手法を用いて補間している。

Whited ら [17] はストロークの対応に基づく、ユーザーシステムを提案した。まず、線画はベクタ画像としたうえで、ストロークのグラフとしてみなす。このときストロークが交わる点はすべて頂点となる。そしてこのグラフ同士の対応を Correspondence Tracing Algorithm を用いてとるという手法である。対応ミスやオクルージョンはユーザーがインタラクティブに修正する。対応しているストロークの補間は対数螺旋によって行うほか、変形することが可能である。

Yang [19] もストロークの対応に基づく、ユーザーシステムを提案した。線画はベクタ画像とする箇所、ユーザーは対応を修正することが可能である点は Whited ら [17] と同じだが、ストロークを長さ、形、位置関係に基づく類似度によって逐次的に 1 対 1 で対応を取るアルゴリズムを採用している。

これらの手法では、オリジナルのソフトウェアを用いることでストロークを整形されたベクトルの形で用いており、ラスタ画像を入力とするシステムにそのまま組み込むことは難しい。そのような場合は、一度ベクタライズをする必要があるが、ベクタライズそれ自体も非常に難しいタスクとなっている。

一方、学習データが入手しにくい関係で機械学習を用いた手法はなかなか提案されていなかったが、Yagi [18] は、アニメーション制作会社と協力し、実際に使われた動画のデータを用いて、CNN を利用した手法を提案した。基本的な構造としては 29 層の Encoder-Decoder となっており、低解像ネットワークと高解像ネットワークが組み合わせられている。入力として 2 枚の画像を入れると、出力として中割りが直接生成される。こちらもラスタ画像をそのまま入力とすればよいので、今のシステムにそのまま組み込むことが可能である。しかし、訓練画像と同じカット内の中割りのみ生成可能なため、中割りの枚数を増やすという目的にのみ利用可能である。また、生成される画像の質も高くない。

### 3. 提案手法

#### 3.1. 概要

前述のように、線画は 2 値画像のため既存の optical flow の推定が難しい。そこで、入力画像として線画を距離変換したものを使用し optical flow を求め、補間を行う手法を提案する。

距離変換とは、画素値を直近の黒画素までの距離とする変換であり、この変換で得られる画像を距離画像とする。二値画像処理に用いられることが多い。物体のスケルトンを検出する [3]、画像間の変化点を検出する [5] といったタスクに応用される。距離変換することで、画像に勾配が生まれ対応も取りやすくなると考えられる。

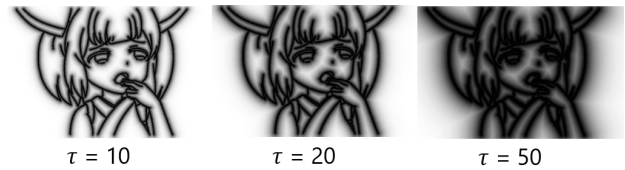


Fig. 2: 距離変換とそのあとの非線形変換のパラメータによる違い。 $\tau$  が大きいほど、勾配が小さくなり、濃い (画素値が小さい) 範囲が広くなる。

距離変換後に、更に以下のように画素値を変換する。

1. パラメータ  $\tau$  を定める
2. 距離画像の画素値  $x$  に対しを、 $\hat{x} = 1 - \exp(-\frac{x}{\tau})$  とする変換を行う
3. 全ての新しい画素値  $\hat{x}$  に対し、上限 255 (白画素)、下限 0 (黒画素) として正規化する。

黒画素から遠い画素は周辺の白背景が大部分であり、対応を取る必要がない。そこで画素値を非線形変換することで、線が込み合っている中心部で、距離変化に対して画素値の勾配を急峻にすることが可能となる。距離の測り方はマンハッタン距離、ユークリッド距離等があるが、今回はユークリッド距離を用いる。距離変換の例は Fig. 2 のようになる。このように線が込み合っている部分の勾配は急峻に、外側の背景は平坦になっていることがわかる。

$frame_t$  から  $frame_{t+2}$  への optical flow を用いて  $frame_{t+1}$  を補間する計算式は optical flow の  $x$  成分を  $flow_x$ 、 $y$  成分を  $flow_y$  として以下のように行う。

$$frame_{t+1}(x, y) = frame_t(x - \frac{flow_x(x, y)}{2}, y - \frac{flow_y(x, y)}{2}) \quad (1)$$

#### 3.2. optical flow の計算

optical flow は既存手法である FlowNet2 [9] の PyTorch 実装 [15] を用いて計算する。モデルは公式の学習済みモデルの他に、距離変換を行った画像で fine-tune を行ったモデルを用いる。この fine-tune についての説明を行う。

距離変換を行った画像は輝度勾配を持つが、公式の学習済みモデルに利用されたデータセット、FlyngChairs [8] や FlyingThings3D [13] は自然画像の背景に物体の CG データを合成したものであり、色情報を持つなどドメインが少し異なる。そこで、FlyngChairs の画像からエッジを抽出し、3 で述べた距離変換と非線形変換を行ったものを入力画像、本来の optical flow データを正解データとして fine-tune を行った。エッジ抽出には OpenCV の Canny 法 [6] を用いた。一連の変換の例を Fig. 3 に示す。

FlyngChairs に含まれる全ての画像ペアのうち、7 割を train, 3 割を validation とした。モデルは FlowNet2,

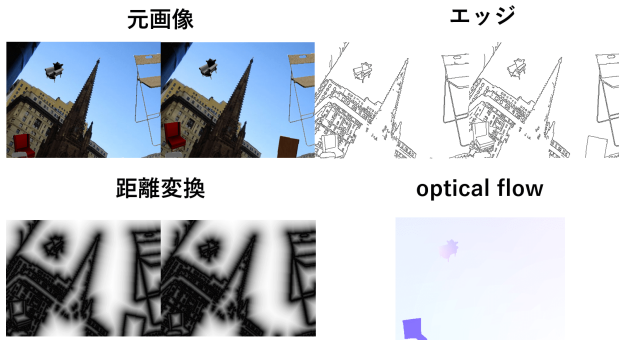


Fig. 3: FlyingChair [8] の元画像, エッジ, 距離変換後の画像および optical flow の可視化例。このようにほとんどが背景である。

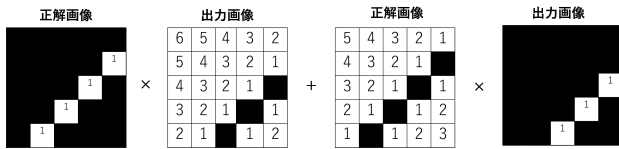


Fig. 4: 距離画像を用いた評価基準の図示

loss には L1 ノルム, optimizer には Adam を使用し, lr はデフォルトより 1 桁低い  $1e-5$  を採用した。

#### 4. データセット

アニメーションのデータセットは著作権の関係から入手が困難だったが, 2017 年, 東北ずん子アニメ「ずんだホライズン」[2] というアニメーションで実際に制作された資料を, クラウドファンディングに出資することで利用することが出来た。

全 469 カット (場面) それぞれにフレームごとの原画, 線画 (中割りを含む), さらにそれを着色したものがデータセット内に含まれている。中割り以下に関してはオブジェクトごとにレイヤーが分けられており, フレーム補間を行う際には, オブジェクトごとの optical flow を求めることが目標となる。

今回の実験では, 線画のみを用いる。このデータセットでは線画に青色, 赤色の線が入っているが, 画像処理をしやすいように, 黒と白の二値化したのち使用する。Fig. 5 に例を示す。

#### 5. 実験

optical flow を用いたフレーム補間について, 入力として, 線画そのままを用いたもの (w/o DT) をベースラインとし, 距離変換を用いたもの (DT(20), DT(50)) を比較手法として実験を行った。距離変換で用いる  $\tau$  は 20 と 50 とした。

optical flow は FlowNet2 の学習済みモデル (pre-trained), エッジ画像で fine-tune したモデル (fine-tune)



Fig. 5: 線画と, それを 2 値化した画像, 着色した画像の例。

を用いた。ただし, fine-tune したモデルでは w/o DT と DT(20) の比較のみを行った。補間の際は OpenCV の remap 関数を使用し, 入力画像は先頭のフレームのみ (距離変換を行っていないもの) を用いている。

ずんだホライズンのデータセットから, 人物が大きく映った 4 つのカットからそれぞれ  $t$  番目のフレームと  $t+2$  番目のフレームを入力画像とした ( $t$  はカットにより異なる)。 $t+1$  番目のフレームを正解画像とする, 評価基準としては, 正解画像と出力画像のストロークのずれを評価するため, 距離画像を用いた距離  $D$  を定義した。二枚の 2 値画像  $A, B$  に対し,

$$D(A, B) = \frac{\bar{A} \text{ と } (B \text{ の距離画像}) \text{ の要素積の和}}{\bar{A} \text{ の白画素数}}$$

とする。ただし  $\bar{A}$  は  $A$  をビット反転させたものである。そして,

$$\frac{D(\text{正解画像}, \text{出力画像}) + D(\text{出力画像}, \text{正解画像})}{2}$$

を最終的な評価基準とする。これを図示すると Fig. 4 のようになる。この距離は以下の式で表される chamfer distance と等価である。

$$\text{chamfer distance} = \frac{1}{2\|A\|} \sum_{a \in A} \min_{b \in B} (\|a - b\|) + \frac{1}{2\|B\|} \sum_{b \in B} \min_{a \in A} (\|b - a\|) \quad (2)$$

$A, B$  は集合,  $a, b$  は各要素である。以下, 論文では上記の評価基準を chamfer distance と呼ぶ。

##### 5.1. 距離変換の有無による比較

chamfer distance の数値は Table 2, 補間の結果は Fig. 6 のようになった。chamfer distance は, どのカットでも, 距離変換を用いた提案手法のときに最も小さくなるのが分かった。また, chamfer distance が小さくなる  $\tau$  の数値はカットにより異なることが分かった。補間の結果をみると, 距離変換を用いていないベースライン手法では Cut B など補間画像と入力画像の差分が, 特に動きの大きい腕の部分に注目すると殆どないことが分かる。一方, 提案手法では腕の大まかな動きの対応はとれていることがわかる。ただし, 今回の補間方法ではオクルージョンの問題は解決できず, 動きの大きい場所ではアーティファクトが多数見受けらおり, 完全な補間は出来ていない。

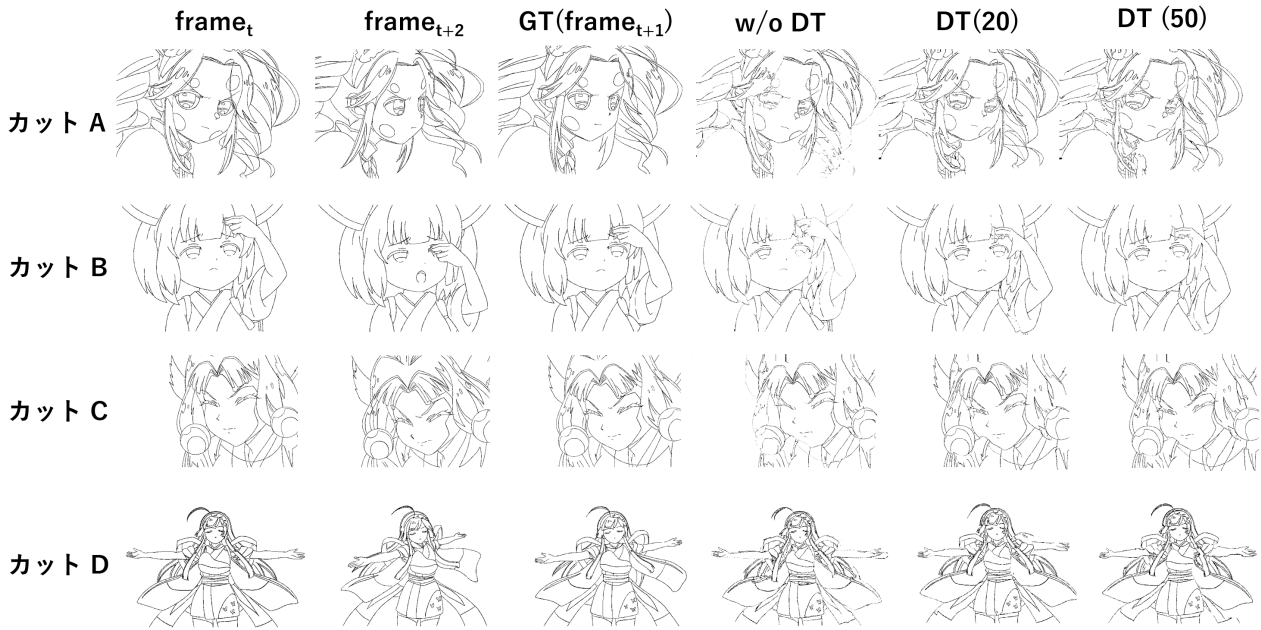


Fig. 6: optical flow を用いたフレーム補間の例. 距離変換を用いることでストロークの動きをある程度終えていることが分かる.

Table 1: 距離変換の有無による chamfer distance の比較

cut	w/o DT	DT(20)	DT (50)
cut A	7.01	<b>5.53</b>	5.73
cut B	3.49	3.47	<b>3.42</b>
cut C	5.71	<b>3.83</b>	3.91
cut D	5.62	<b>4.64</b>	4.76

提案手法では, Cut C のような動きが小さい場合の補間は上手くいっているのに対し, Cut D では補間の結果が正解画像と大きく異なっている. このカットはキャラクターが回転運動を行っている. 回転運動では平行運動と異なり物体の見え方が変わってしまうため, フレーム間の対応が取りづらく, また, 対応が取れたとしても補間が難しいケースとなる.

optical flow の可視化結果は Fig. 7 のようになっている. ベースライン手法では Cut A の髪や, Cut B の腕など, 線分上以外で flow が見られないもの目立つが, 提案手法では線分の間でも flow が見られる. これは画像に勾配が生まれたことで, より対応を取りやすくなったためと考えられる. 一方,  $\tau$  を上げた場合, Cut A の右目周辺, Cut D の左腕周辺などで flow が見られなくなっている場所が発生する. これは線分が混みあっている場所での勾配が平坦になってしまい, 線分だった場所含めて対応が取れなくなってしまうからと考えられる. このように, 画像に合わせて最適な  $\tau$  を選択する必要があると考えられる.

## 5.2. FlowNet2 の学習の有無による比較

chamfer distance の数値は Table 1, 補間の結果は Fig. 8 のようになった. chamfer distance の値は pre-trained, fine-tune ともにあまり差はなく, その大小はカットにより異なるという結果となった.

例えばカット B では fine-tune の場合, chamfer distance が大きいだけでなく, 補間画像, 特に腕のあたりに多くのノイズが残っていることが分かる. また, ほかのカットでもストロークが歪んでいるなど, アーティファクトが多く, pre-trained と比べて補間の精度は大きく向上しているわけではないことが分かる.

optical flow の可視化結果は Fig. 7 のようになっている. fine-tune では全てのカットにおいての境界が明瞭であり, その境界が距離変換前のストロークの場所であることが分かる. 例えばカット B の腕の部分では, pre-trained では黄色の領域が背景にしみだしているが, fine-tune では腕の領域のみにとどまっている. しかし, 前述のように補間の精度は pre-trained の方が勝っている. これは, 距離変換前のストローク周囲の背景部分の動きを正しく計算できていないためと考えられる.

この原因は FlowNet2 の学習方法によると考えられる. 学習の際に, エッジ抽出前の画像についての optical flow を正解データとしたため Fig. 3, もともと背景だった物体の外側の領域の optical flow が上手く学習されていない可能性が高い. そのため, 今後は背景を削除する, 背景領域については loss の計算を行わないといった学習を進める必要がある,

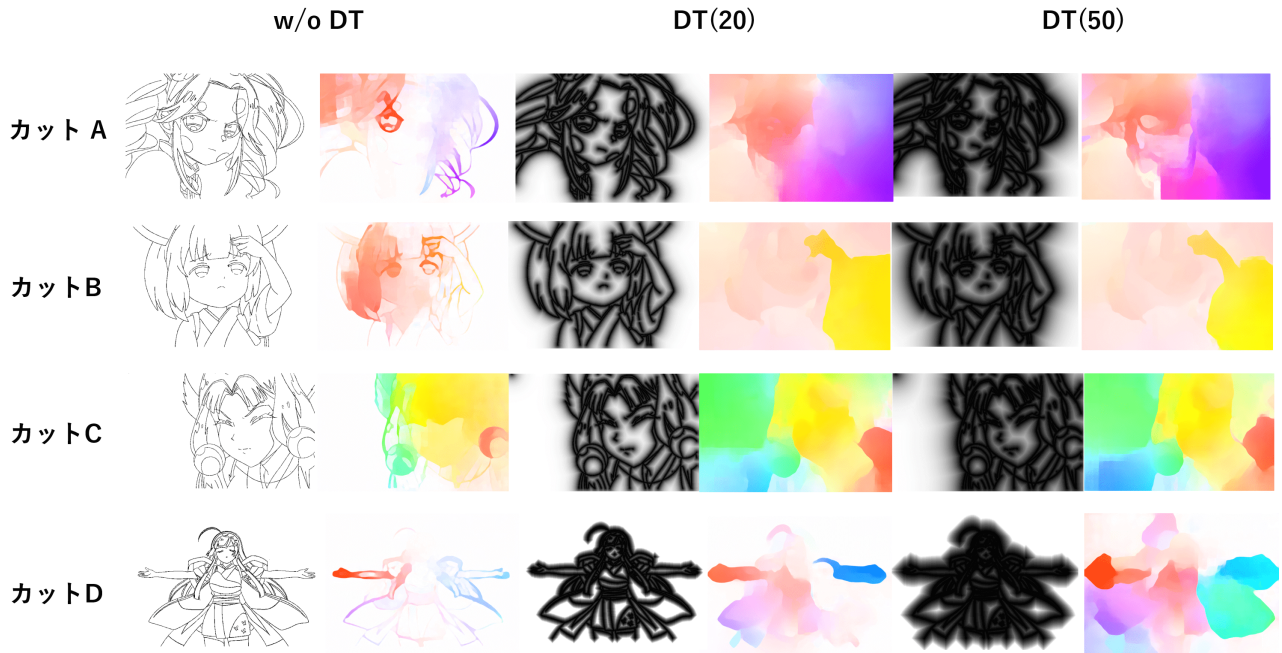


Fig. 7: optical flow の可視化例. 距離変換を用いることで, もともと白色だった領域の optical flow を計算できていることが分かる.

Table 2: FlowNet2 の fine-tune の有無による chamfer distance の比較

DT	w/o DT	w/o DT	DT (20)	DT (20)
FlowNet2	pre-trained	fine-tune	pre-trained	fine-tune
cut A	7.01	6.74	5.53	<b>5.21</b>
cut B	3.49	3.55	<b>3.47</b>	3.71
cut C	5.71	5.33	<b>3.83</b>	3.88
cut D	5.62	5.78	4.64	<b>4.43</b>

## 6. まとめ

本稿ではアニメーションの連続する線画フレームの補間について, 入力としてラスト画像として扱い, 距離変換を行ったのちに, optical flow 取ることで, ベースライン手法と比べ精度の良い補間を行うことが出来ることを示した.

本稿では optical flow 出力のための学習以外は行わなかったが, 今後の課題として, 距離変換を行った線画フレームに特化した, 補間フレームを出力するフレーム間の対応を取るアーキテクチャを開発することが挙げられる. 具体的には, CNN の学習をずんだホライズンデータセットで行う, 損失関数として chamfer distance を用いてこれを最適化する, 双方向の optical flow を用いる, ということが考えられる. また, 距離変換をアーキテクチャの内部に組み込むという手法も考えられる

## 7. 謝辞

本研究の一部は科研費 17K19963 の助成を受けた

## 参考文献

- [1] Super-fast elastic registration of histologic images of a whole rat brain for three-dimensional reconstruction. In *Proc. of SPIE Vol.*, Vol. 5370, p. 329.
- [2] zunda horizon. <https://greenfunding.jp/pub/projects/1852>.
- [3] Carlo Arcelli and Gabriella Sanniti di Baja. Finding local maxima in a pseudo-euclidean distance transform. *Comput. Vision Graph. Image Process.*, Vol. 43, No. 3, pp. 361–367, 1988.
- [4] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, Vol. 92, No. 1, pp. 1–31, 2011.
- [5] Étienne Baudrier, Frédéric Nicolier, Gilles Millon, and Su Ruan. Binary-image comparison with local-dissimilarity quantification. *Pattern Recognition*, Vol. 41, No. 5, pp. 1461–1478, 2008.
- [6] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, No. 6, pp. 679–698, 1986.
- [7] Christina N de Juan and Bobby Bodenheimer. Re-using traditional animation: methods for semi-

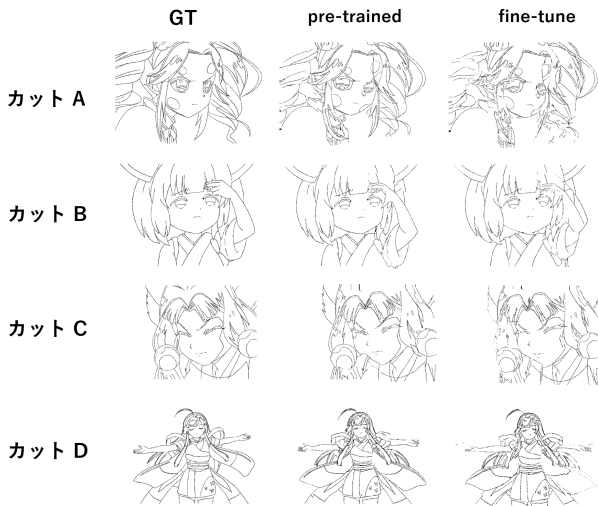


Fig. 8: FlowNet2 の fine-tune の有無による optical flow を用いたフレーム補間の比較. fine-tune では補間結果にアーティファクトが目立つ.

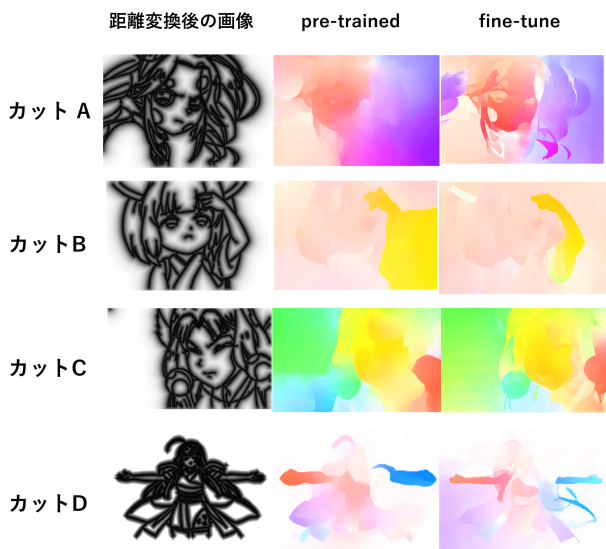


Fig. 9: FlowNet2 の fine-tune の有無による optical flow の可視化比較. fine-tune では optical flow の境界がはっきりしている.

automatic segmentation and inbetweening. In *Eurographics*, pp. 223–232. Eurographics Association, 2006.

- [8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, pp. 2758–2766, 2015.
- [9] Saikia Keuper Dosovitskiy Brox Ilg, Mayer. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*.
- [10] Alexander Kort. Computer aided inbetweening. In *Proceedings of the 2nd international symposium on*

*Non-photorealistic animation and rendering*, pp. 125–132. ACM, 2002.

- [11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012.
- [12] Tang Liu Agarwala Liu, Yeh. Video frame synthesis using deep voxel flow. In *ICCV*, 2017.
- [13] Nikolaus Mayer, Eddy Ilg, Philipp Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, pp. 4040–4048, 2016.
- [14] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017.
- [15] Fitsum Reda, Robert Pottorff, Jon Barker, and Bryan Catanzaro. flowNet2-pytorch: Pytorch implementation of flowNet 2.0: Evolution of optical flow estimation with deep networks. <https://github.com/NVIDIA/flowNet2-pytorch>, 2017.
- [16] William T Reeves. Inbetweening for computer animation utilizing moving point constraints. *SIGGRAPH*, Vol. 15, No. 3, pp. 263–269, 1981.
- [17] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, pp. 1385–1392, 2013.
- [18] Yuichi Yagi. A filter based approach for inbetweening. *arXiv preprint arXiv:1706.03497*, 2017.
- [19] Wenwu Yang. Context-aware computer aided inbetweening. *TVCG*, Vol. 24, No. 2, pp. 1049–1062, 2018.