

# Interactive Video Completion

岡部 誠<sup>1</sup>

野田 啓太<sup>2</sup>

土橋 宜典<sup>3</sup>

安生 健一<sup>4</sup>

<sup>1,2</sup> 静岡大学 <sup>3</sup> 北海道大学 <sup>4</sup> OLM Digital, Inc./CMIC, Victoria University of Wellington

E-mail: <sup>1</sup> m.o@acm.org, <sup>2</sup> noda.keita.17@shizuoka.ac.jp, <sup>3</sup> doba@ime.ist.hokudai.ac.jp, <sup>4</sup> anjyo@acm.org

## 1 概要

“インタラクティブな動画修復手法”を提案する。本手法は、1) 既存手法と比べ修復結果のクオリティが高い、2) 計算が速く、インタラクティブな修復作業が可能、3) キーフレームを与え、結果を更に修正できる、などが特徴である。本手法では、動画の速度場を用いて色の修復を行っている。反復アルゴリズムが速度場と色を交互に最適化するが、速度場については既存手法と同様に  $L^1$  最適化を行い、一方、色については既存手法と異なり  $L^2$  最適化を行う。これによって滑らかな色変化を実現し、結果のクオリティを改善した。GPU を用いて計算した場合、 $854 \times 480$  ピクセル、100 フレームの動画であれば数秒で処理できる。そこで、ユーザがマスクを描きながら、徐々に物体を消去していけるような、インタラクティブな編集ツールを実装した。また、主観評価では4つの既存手法との比較を行い、結果のクオリティについても検証を行った。

## 2 はじめに

本研究の目的は動画中の物体を消去することである。図1に例を示す。図1-左は“dance-jump”というタイトルの動画のフレーム画像であり、図1-右は動画からダンサーを消去した後のフレーム画像である。図1-左の赤いピクセルは消去すべき箇所を指定するためにユーザが与えたマスクを表している。人物や物体以外にも動画中に現れるロゴやアノテーションやノイズを消去するというタスクは、映像制作の現場で日常的に頻出しているが、現在はその作業の多くを人手に頼っているというのが現状である。そこで我々は実践的で、かつ、インタラクティブな編集作業を可能とするようなソフトウェアを開発し作業を効率化したい。



図1: 左) 入力動画とマスク。右) 物体消去の結果。

## 3 提案手法

提案手法は流れ場に基づいて色を推定するアプローチ [1] を採用している。図2-左は“tennis”というタイトルの動画の12フレーム目の画像である。今、この動画からテニスプレーヤー (図2-左の黄色点線) を消去したいとする。ここで、テニスプレーヤーを消去する、ということは、テニスプレーヤーの背景画像 (テニスプレーヤーが居なくなったときに何が見えるか) を推定できれば、その画像でテニスプレーヤーを塗り潰す (修復する) ことで実現できる。従って、解くべき問題はテニスプレーヤーの背景画像の推定、ということになるが、それはこの動画の場合には少し巻き戻してみればよい。例えば、4フレーム目まで巻き戻せば図2-右の画像となるが、この画像ではテニスプレーヤーは移動しているため欲しい背景画像が見えている。即ち、図2-右の黄色点線の箇所を使って、図2-左のテニスプレーヤーを塗り潰せば (修復すれば) 物体消去が達成できる。逆もまた同様で、図2-右のテニスプレーヤーを消去するのに必要な情報は、図2-左の画像にある。

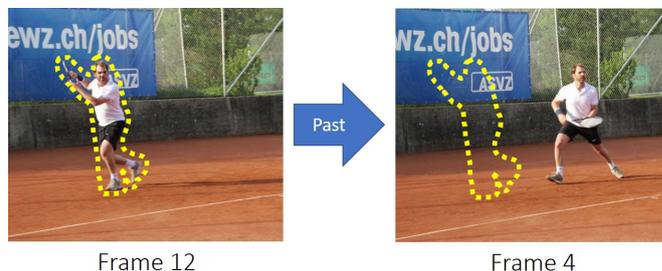


図2: 流れ場に基づく色の推定。物体消去のために必要な背景画像は過去または未来のフレーム画像から推定できる。

詳細は論文 [2] に譲るが提案手法が既存手法と異なるのは、上記の背景画像の推定の際、既存手法が  $L^1$  最適化を採用していたのに対し、提案手法は  $L^2$  最適化を採用した、という点である。図3は“kite-surf”というタイトルの動画からサーファーを消去した結果のフレーム画像である。図3-左に  $L^1$  最適化の結果を示すが、 $L^1$  最適化では時間的な不連続が発生してしまい、それが赤い矩形の箇所のようにアーティファクトとして現れてしまうが、 $L^2$  最適化では図3-右に示すようにそのような時間的な不連続が発生しない。数学的には簡単な変更であるが、結果のクオリティに与える影響は大きい。

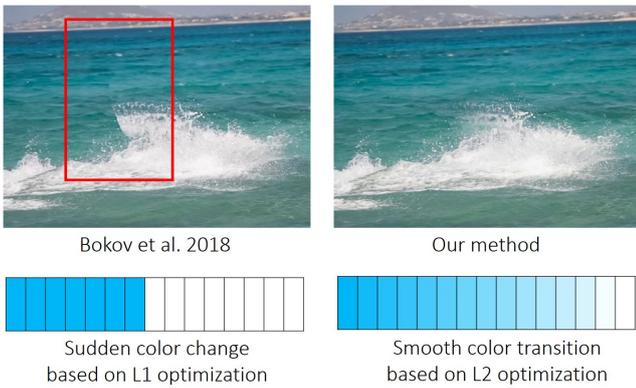


図 3:  $L^1$  最適化 vs  $L^2$  最適化。

## 4 結果と考察

結果については著者らのウェブサイトでは 33 個の動画に対する結果を、既存手法 [3, 4, 5] との比較も合わせて掲載している。また、図 4 に主観評価の結果を示す。ここでは提案手法と 3 つの既存手法の計 4 つに対する比較を行っている。被験者の数は 24 名である。各被験者は、各動画に対し 4 つの手法で物体消去した結果を見比べ、最良の結果だと思ったものに投票する。例えば、図 4 の最上段は “hike” というタイトルの動画に対する投票結果だが、バー全体が黄色い。これは 24 名の被験者全員が提案手法の結果に投票したことを示している。図 4 を見れば、ほとんどの動画で提案手法の結果が最良と判定されていることが分かる。提案手法の結果が最良とならなかったのは、25 個の動画のうち 4 個であるが、この理由については論文 [2] に詳細に記述した。

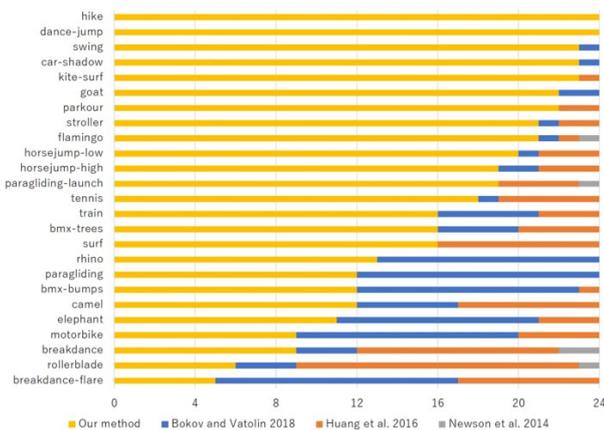


図 4: 主観評価の結果。各バーの長さは、対応する手法の結果に対し、それを最良と投票した被験者の数を表す。

物体消去の結果のクオリティが高いことに加え、提案手法のもう 1 つの利点は計算時間が短いことである。図 5-左に示すように提案手法は同じ動画を処理する際、既存手法に比べて数十倍程度高速である。これは既存手法が CPU

ベースの実装であったのに対し、提案手法がその処理のほとんどを GPU を用いて実装しているから、というのが大きな理由である (図 5-右)。

Method	Time	Our computer	
Huang et al. 2016	3 hours	CPU	Intel Core i7 4.0GHz
Bokov et al. 2018	75 seconds	Memory	32GB
<b>Our method</b>	<b>4.1 seconds</b>	GPU	NVIDIA GeForce GTX 1080Ti

図 5: 計算時間の比較と提案手法の実験環境。

高速な技術が開発できたことによって、図 6 に示すようなインタラクティブな物体消去が実現できる。図 5 の実験までの話は入力動画に加えマスクが全フレームに与えられていることが前提だった。図 6 の例では入力動画のみが与えられ、ユーザはキーフレームに対し赤いピクセルで示したようなラフなマスクを描く。システムはこれらのラフなマスクを補間し全フレームに対してマスクを生成し、提案手法を適用することで物体消去を行う。ユーザはマスクを描いて “update” ボタンを押せば数秒で物体消去の結果が更新されるので、それを確認し再びマスクを修正し… という編集をインタラクティブに繰り返すことが可能である。



図 6: 854 × 480 ピクセル、80 フレームの動画から自転車を消去した。作業時間は 2.5 分。上段はユーザがキーフレームに描いたマスク、下段は物体消去の結果を示す。

## 参考文献

- [1] M. Roxas, T. Shiratori, and K. Ikeuchi, “Video completion via spatio-temporally consistent motion inpainting,” *IPSP Transactions on Computer Vision and Applications*, vol. 6, pp. 98–102, 2014.
- [2] M. Okabe, K. Noda, Y. Dobashi, and K. Anjyo, “Interactive video completion,” *IEEE Computer Graphics and Applications (CG&A)*, vol. 40, pp. 127–139, 2019.
- [3] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, “Video inpainting of complex scenes,” *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, 2014.
- [4] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, “Temporally coherent completion of dynamic video,” in *Proc. SIGGRAPH Asia 2016*, 2016, pp. 196:1–196:11.
- [5] A. Bokov and D. Vatolin, “100+ times faster video completion by optical-flow-guided variational refinement,” in *Proc. IEEE ICIP 2018*, 2018, pp. 2122–2126.