動画検索を用いた水シーン画像のアニメーション

Animating Pictures of Water Scenes using Video Retrieval

岡部 誠 ∗　土橋 宜典 †　安生 健一 ‡

Makoto OKABE∗, Yoshinori DOBASHI† and Ken ANJYO‡

∗ 電気通信大学
† 北海道大学
‡ 株式会社オー・エル・エム・デジタル

∗ The University of Electro-Communications
† Hokkaido University
‡ OLM Digital, Inc.

E-mail: ∗m.o@acm.org, †doba@ime.ist.hokudai.ac.jp, ‡anjyo@olm.co.jp

1　はじめに

絵画や写真等の単一画像をアニメーションさせる技術は，CG 分野で活発に研究されている．一方，自然な水のアニメーションは未だ解決されていない．過去にも複数の試みが発表されてきたが，単調な動きのみが対象だったり，計算コストが高い等の問題があった [1, 2, 3, 4]．また，既存手法の結果は何れも実際の水を撮影した映像と比較すると不自然だった．単一画像から自然で高画質な水のアニメーションを生成することは未だ達成されていない．

既存手法は実際の流体の動きについて一部の情報しか扱っておらず，これが不自然な結果をもたらしている．流体の揺らぎに関する統計量のみでは微妙な繰り返しの変動しか表現できない [1]．単一動画のみからでは流体の多様な動きを表現できない [2, 3]．小さなパッチに基づき局所的な情報を扱うのは良いアプローチだが，大局的な構造の情報が欠けていた [5, 4]．これらの問題を解決するため，我々はデータベース中の複数の動画をなるべくそのまま使うことを試みる．提案手法は並列化が簡単で，ユーザは素早いフィードバックが得られる．単一画像から様々の水シーンのアニメーションが生成できることを示す．

2　提案手法

図 1 に示すように，提案手法の入力は 3 つあり，ユーザがアニメーションさせたい画像，水領域を示すアルファマスク， 水の流れ方向のスケッチである．これらが与えられると，提案手法は見た目と動きが入力にマッチするような動画をデータベースから検索する．それぞれの領域に対し，最もマッチした動画を検索し，候補アニメーションを合成してユーザに一覧で見せる．例として，2 つの領域（滝と川）に対する 4 つの候補アニメーションを図 2 に示す．ユーザは候補アニメーションを観察し，最も適切と感じるものを選択する．このインタフェースは [6] を参考に開発した．全ての領域に対して選択を完了した時点でアニメーションが完成する．最後にオプションとして入力画像の見た目を復元するためのテクスチャ合成を適用できる [2, 4]．これにより油絵等のスタイルが転送され質感が維持される．

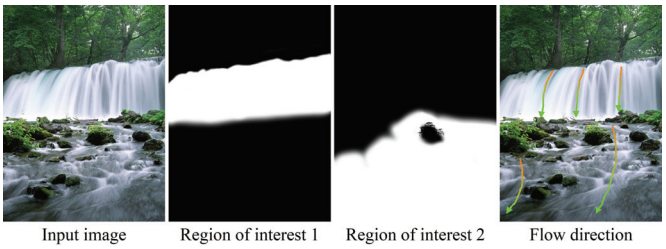

図 1: 入力画像，水領域を示すアルファマスク (滝と川の 2 つの領域)，水の流れ方向のスケッチ．

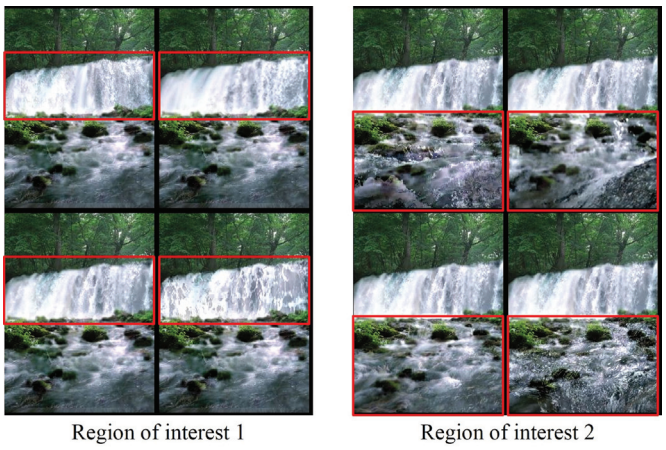

図 2: 滝と川の領域に対する 4 つの候補アニメーション．

2.1　問題設定

図 3 に提案手法のアイデアを示す．図 3-a を入力画像，図 3-b をユーザが水領域に指定したアルファマスクとする．ここで，図 3-c の動画をデータベースから検索し，図 3-d のように入力画像の滝の上に重ね合わせる．2 つの滝は見た目が似ており，また動きも似ていると推察される．よって，両者を合成し，入力画像のアニメーションが得られる．

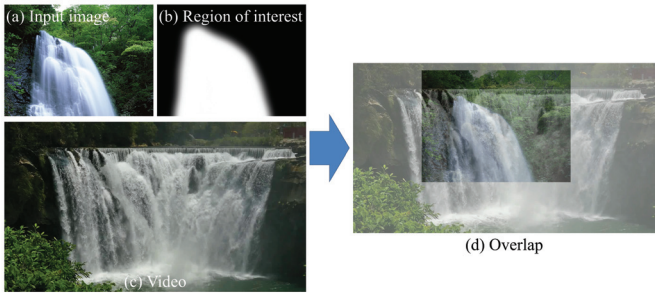

図 3: 提案手法のアイデア.

このアイデアに基づき高画質な結果を得るため，次の 4 つの指標を考える．1) 入力画像と動画の間で水領域の重なりはなるべく大きくあるべき．2) 入力画像と動画の見た目はなるべく類似すべき．3) 入力画像と動画の流れ方向はなるべく類似すべき．4) 入力画像から推察される水の挙動と動画の水の挙動はなるべく類似すべき．

本研究では図 3 の滝のような，大局的な見た目は変化しないが局所的に動きのあるアニメーションを扱う [7]．データベース中の動画も同様の特徴を持つため，今後は各動画の見た目の表現として，単一の代表画像のみを考える．

上の 4 つの指標に基づき，類似度評価関数を導入する．ここで，$\mathbf{I}$，$\mathbf{A}_r$，$\mathbf{F}$ を入力画像，r 番目の領域のアルファマスク，流れ方向のスケッチから得られる流れ場とする．2 値マスク $\mathbf{M}_r$ を次のように定義する:

$$\mathbf{M}_r(\mathbf{p}) = \begin{cases} 1 & \|\mathbf{A}_r(\mathbf{p})\| \geq t^{mask}, \\ 0 & otherwise. \end{cases} \tag{1}$$

ここで $\mathbf{p} = (p_x, p_y)$ はピクセル位置を表す．$\mathbf{V}_i$，$\mathbf{M}^{\mathbf{V}_i}$，$\mathbf{F}^{\mathbf{V}_i}$ は i 番目の動画の最初のフレーム画像，そのフレーム内の水領域，そのフレームでの流れ場とする．類似度評価関数 S を次の様に定義する:

$$\begin{aligned} S \quad = \quad & A(\mathbf{M}_r \cap T(\mathbf{M}^{\mathbf{V}_i}, \mathbf{t})) \times & (2) \\ \sum_{\mathbf{p}} \quad & S^{feature}(f_{\mathbf{p}}(\mathbf{I}), f_{\mathbf{p}-\mathbf{t}}(\mathbf{V}_i)) \times & (3) \\ & S^{flow}(\mathbf{F}(\mathbf{p}), \mathbf{F}^{\mathbf{V}_i}(\mathbf{p}-\mathbf{t})) \times & (4) \\ & S^{motion}(m_{\mathbf{p}}(\mathbf{I}), m_{\mathbf{p}-\mathbf{t}}(\mathbf{V}_i)) \times & (5) \\ & \mathbf{M}_r(\mathbf{p})\mathbf{M}^{\mathbf{V}_i}(\mathbf{p}-\mathbf{t}). & (6) \end{aligned}$$

式 2 は 1) の重なりの大きさに関する指標に相当する．$A(\mathbf{M})$ は 2 値マスク $\mathbf{M}$ の面積を計算する関数である．$T(\mathbf{M}, \mathbf{t})$ は 2 値マスク $\mathbf{M}$ を 2 次元ベクトル $\mathbf{t} = (t_x, t_y)$ で並行移動させる関数である．$\mathbf{t}$ の範囲は $-w^{\mathbf{I}} < t_x < w^{\mathbf{I}}$ と $-h^{\mathbf{I}} < t_y < h^{\mathbf{I}}$ である．ここで，$w^{\mathbf{I}}$ と $h^{\mathbf{I}}$ は入力画像 $\mathbf{I}$ の幅と高さである．式 3 は 2) の見た目の類似性に関する指標に相当する．関数 f は特徴記述子であり，$f_{\mathbf{p}}(\mathbf{I})$ は $\mathbf{p}$ の位置での $\mathbf{I}$ の見た目に関する特徴ベクトルを与える．特徴ベクトルのペアが与えられると，関数 $S^{feature}$ が類似度を計算する．式 4 は 3) の流れ方向の指標に相当する．流れ方向を表す 2 次元ベクトルのペアが与えられると，関数 S^{flow} は類似度を計算する．式 5 は 4) の水の挙動に関する指標に相当する．$m_{\mathbf{p}}(\mathbf{I})$ は $\mathbf{I}$ の水の挙動を $\mathbf{p}$ の位置で推定する関数とする．式 6 は類似度評価が入力画像と動画の水の領域内部で行われることを保証する項である．

我々のアルゴリズムは，まず，r 番目の領域毎に，$\arg\max_{i,\mathbf{t}} S$，を計算する．つまり，関数 S が最大になるような動画のインデックス i と平行移動量 $\mathbf{t}$ を得る．そして，i 番目の動画を $\mathbf{t}$ で並行移動させたものを入力画像と合成することで，r 番目の領域のアニメーションを生成する．

ところが，S の最大化は難しい．式 2，式 3，式 4，式 6 は，そこに登場する全ての関数が適切に定義でき，また必要な情報もユーザによって与えられているため，計算が可能である．しかし，式 5 は単一画像から水の挙動を推定する関数 m を含んでいるが，そのような関数は未知であり，適切に実装するのが難しい．そこで，我々の S を最大化する戦略は，まず，式 2，式 3，式 4，式 6 から成る S' を考え，これを全ての可能な i と $\mathbf{t}$ の組み合わせに対して計算する．次に，高い類似度 S' を与えた i と $\mathbf{t}$ に対し，その i 番目の動画を $\mathbf{t}$ で平行移動させたものを合成して候補アニメーション作ってユーザに見せる．最後にユーザが最も良いと感じた動画を選ぶ，つまり，ユーザ自身に式 5 を最大化してもらうことで S の最大化を行う，というのが我々の戦略である．

2.2 動画データベース

202 個の水の動画を集めた（付録のサムネイル一覧を参照）．解像度は 1280 × 720 ピクセルである．各動画の左右フリップしたバージョンを用意しているので，動画の数は 2 倍の 404 個である．i 番目の動画に対し，その流れ場 $\mathbf{F}^{\mathbf{V}_i}$，2 値マスク $\mathbf{M}^{\mathbf{V}_i}$，アルファマスク $\mathbf{A}_{i,t}$ を予め計算しておく．このために，オプティカルフロー [8] によって流れ場 $\mathbf{F}^{\mathbf{V}_i}$ を得ると同時に，水領域を粗く推定した後，閾値処理によって 2 値マスク $\mathbf{M}^{\mathbf{V}_i}$ を得る．図 4 に動画と 2 値マスクの例を示す．アルファマスク $\mathbf{A}_{i,t}$ は色ヒストグラム

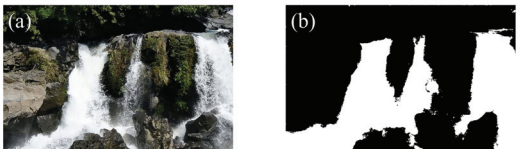

図 4: (a) 動画のフレーム． (b) 2 値マスク $\mathbf{M}^{\mathbf{V}_i}$.

による画像のセグメンテーション [9] を用いて得た．

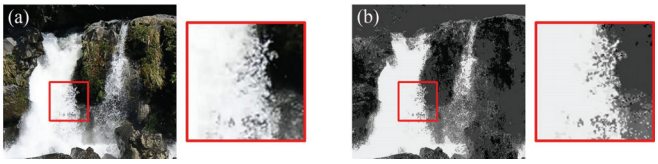

図 5: (a) 動画のフレーム． (b) アルファマスク.

2.3 動画検索

高い類似度評価値 S' を与えるような動画のインデックス i と平行移動量 $\mathbf{t}$ の集合を見つけるために全探索を行う．計算を簡略化するため，入力画像と動画フレームに対し，疎なグリッド上に特徴点を考える（図 6）．特徴点の間隔は本実験では 16 ピクセルとする．従って，平行移動量 $\mathbf{t}$ も 16 ピクセル間隔で考える．また，特徴点はマスク上にのみ存在する．従って，式 6 の項は既に最大化されていると見なすことができる．式 2 の評価は入力画像上と動画上に同時に存在する特徴点を数えることと同値となる．

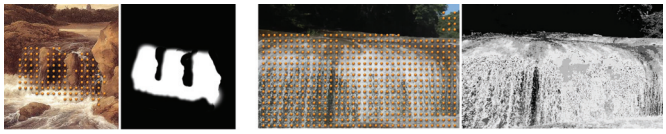

図 6: 疎なグリッド上に特徴点を考える．

式 3 については，各特徴点を中心とする 64×64 ピクセルのパッチに対し画像勾配のヒストグラムを考える．これは，回転不変性とスケール不変性を無視した SIFT 特徴量と同等である [10]．特徴量のペアが与えられると，そのユークリッド距離に基づいて式 3 を与える．

式 4 については, 対応する特徴点の間で流れ場の類似性を計算する．入力画像上には流れ方向のスケッチが疎に描かれているだけなので，同径基底関数 (RBF) を用いて補完し流れ場を得る．2 次元ベクトルのペアが与えられると，その間の角度に基づいて式 4 を与える．

2.4 アニメーションの合成

滝や河川等の典型的な水の動きを見ると，大局的には見た目が変わらないが，局所的には水飛沫等の動きが見られる．そこで，提案手法では大局的な見た目，つまり，低周波成分を入力画像から得て，局所的な動き，つまり，高周波成分を動画から得て合成することで目的のアニメーションを得る．図 7 にプロセスを示す．入力画像（図 7-a）をガウスブラーでぼかし，低周波成分（図 7-b）を得る．次に，動画フレーム（図 7-d）をガウスブラーでぼかした画像（図 7-e）を元の動画フレームから減算することで高周波成分（図 7-f）を得る．最後に低周波成分（図 7-b）と高周波成分（図 7-f）を加算することで結果を得る（図 7-c）.

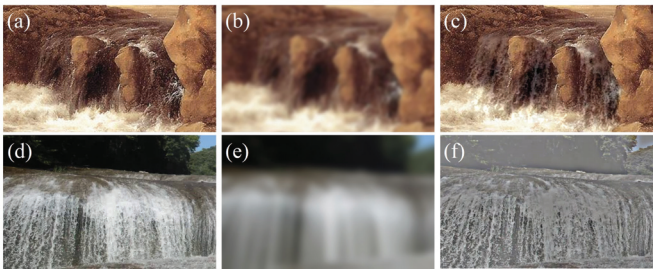

図 7: 合成のプロセス.

3 結果と議論

図 8 に提案手法で作った 11 個の結果を示す．図 8-a から図 8-i は 640×480 ピクセルの写真，図 8-j と図 8-k は 17 世紀に描かれた Jacob van Ruisdael の油絵であり，解像度はそれぞれ 680×878 ピクセルと 990×704 ピクセルである．アルファマスクの作成は Adobe Photoshop CC 2014，流れ方向のスケッチは専用のソフトウェアを開発して用いた．動画検索（S' の計算）は 100 コアの CPU を用いて並列に行い，数十秒の時間を要する．動画検索が終わると，16 個の候補アニメーションが 4×4 のグリッド状に表示され，ユーザに選択を促す (付録の動画参照).

主観評価　結果の質を評価するために主観評価を行った．被験者は 13 人の学生である．図 8 のアニメーションを見て，各結果の質に対し，1 点（最低画質）から 5 点（最高画質）の範囲で採点してもらった．結果を図 9 に示す．最高点は図 8-g が獲得した．複数の被験者から「自然」「美しい」とコメントが得られた．最低点は図 8-c が獲得した．コメントによると，結果中にブロックノイズが見られる，というのが大きな原因である．このノイズは元の動画が持っているものであった．データベースに登録する動画の質に注意すべきであった．

比較　既存手法 [4] と比較を行った．既存手法では図 8-e と図 8-k の結果がある．これらをそれぞれ e' と k' と呼ぶ．上と同様の主観評価を行うと，図 9 の青いバーのようになった．e と e' を比べると，提案手法が高得点を得ており，手法の優位性が伺える．一方，k と k' の比較では，提案手法がやや高い点数を得た．複数の被験者のコメントによると，既存手法の結果 k' は解像度が低くぼやけているが，かと言って不快ではないので低評価としなかった，とのことだった．一方，既存手法は計算に 1 時間程度も費やす．提案手法の方が同等の結果を素早く得られると言える．

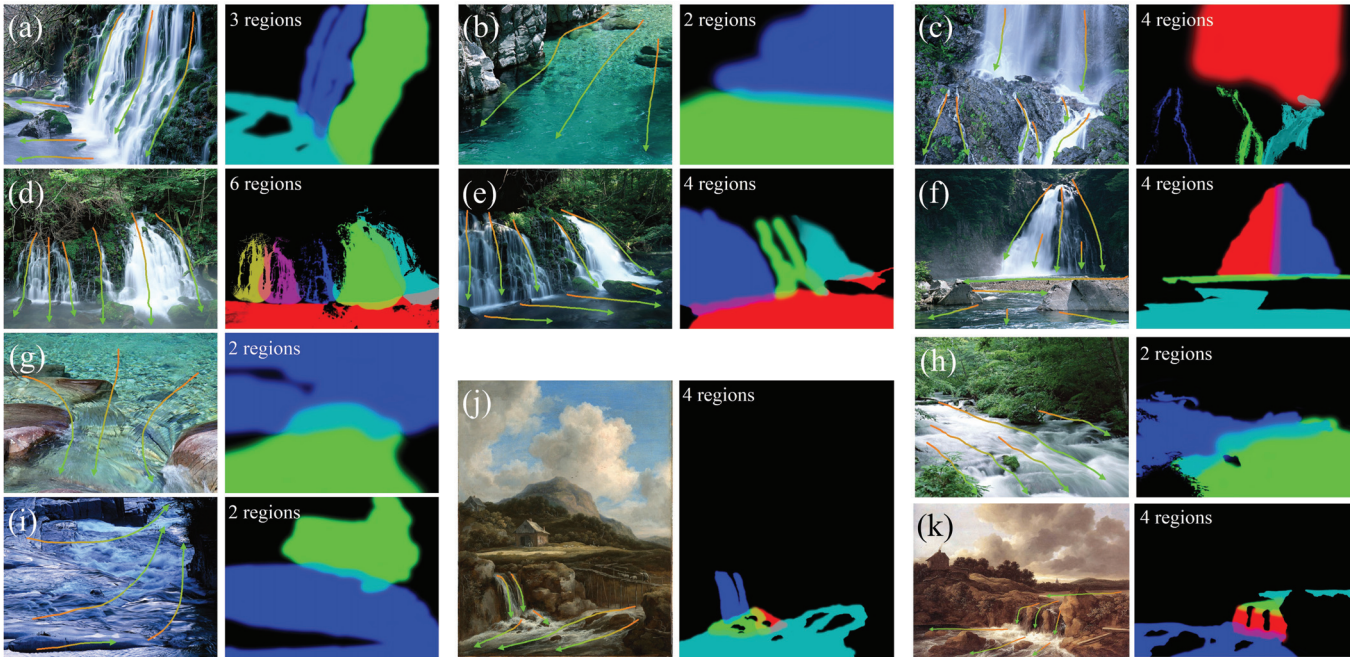

図 8: 各ペアは入力画像（左），アルファマスク（右），流れ方向のスケッチ（矢印）を示す．実際のアルファマスクはグレースケールだが，上では色付けして可視化している．作成したアルファマスクの数も示す．

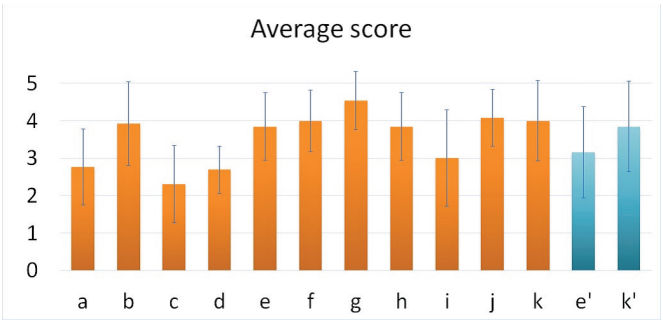

図 9: 主観評価の結果．各バーは平均点を表す．エラーバーは標準偏差．

失敗例　図 10 は典型的な失敗例である．図 10-a は水の泡が発生している大きな白い領域があり，この部分は激しく動くはずであるが，結果にはそのような水の挙動が見られない．これは検索された動画にそのような挙動を持ったものがなかったのが原因である．また，図 10-b は小さな滝が大きな滝の背後に見えているが，これらも動かすことが出来ていない．理由は同じく検索された動画がそのような小さな滝を含んでいなかったためである．

3.1　限界と今後の展望

今回は水のみを扱ったが，今後は炎や煙等，他の流体にも手法を拡張したい．また，近年の目覚しい画像理解技術の進歩を見れば，水領域や流れ方向の自動推定も可能だと思われる．知的なシステムを構築し，自動化を目指したい．

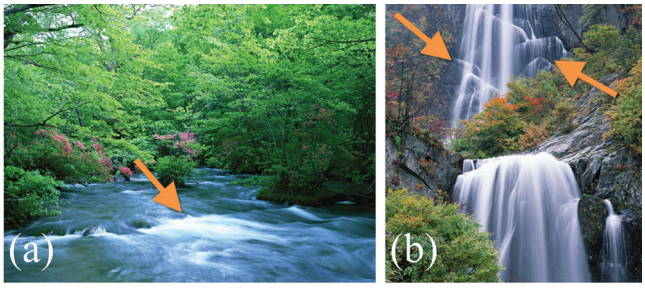

図 10: 典型的な失敗例．

参考文献

[1] Y.-Y. Chuang, D. B. Goldman, K. C. Zheng, B. Curless, D. H. Salesin, and R. Szeliski, “Animating pictures with stochastic motion textures,” in *Proc. SIGGRAPH 2005*, 2005, pp. 853–860.

[2] M. Okabe, K. Anjyo, T. Igarashi, and H.-P. Seidel, “Animating pictures of fluid using video examples,” *Computer Graphics Forum (Proc. EUROGRAPHICS)*, vol. 28, no. 2, pp. 677–686, 2009.

[3] Y. Gui, L. Ma, C. Yin, and C. Z, “Preserving global features of fluid animation from a single image using video examples,” *Journal of Zhejian University - Science C*, vol. 13, no. 7, pp. 510–519, 2012.

[4] M. Okabe, K. Anjyo, and R. Onai, “Creating fluid animation from a single image using video database,” *Computer Graphics Forum (Proc. Pacific Graphics 2011)*, vol. 30, no. 7, pp. 1973–1982, 2011.

[5] L.-Y. Wei and M. Levoy, “Fast texture synthesis using tree-structured vector quantization,” in *Proc. SIGGRAPH 2000*, 2000, pp. 479–488.

[6] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[7] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Int. J. Comput. Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[8] M. Werlberger, T. Pock, and H. Bischof, "Motion estimation with non-local total variation regularization," in *Proc. of CVPR 2010*, 2010, pp. 2464–2471.

[9] D. Ramanan, "Learning to parse images of articulated bodies," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 1129–1136. [Online]. Available: http://papers.nips.cc/paper/2976-learning-to-parse-images-of-articulated-bodies.pdf

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, 2004.