ビデオデータベースを用いた流体画像に基づくアニメーション生成

A Data-Driven Approach for Creating a Fluid Animation from a Single Image

岡部誠 † 安生健一 ‡ 尾内理紀夫 *
Makoto OKABE†, Ken ANJYO‡ and Rikio ONAI*

†* 電気通信大学 †* The University of Electro-Communications
† 科学技術振興機構さきがけ † JST PRESTO
‡ オー・エル・エム・デジタル ‡ OLM Digital, Inc.

E-mail: †m.o@acm.org, ‡anjyo@olm.co.jp, *onai@onailab.com

1 はじめに

静止画を動かすことは、飛び出す絵本に起源を持ち、現在では映像産業においても伝統的に用いられている視覚効果である．コンピュータグラフィックスの活発な研究分野でもある [11, 16, 13]．デザイナーは単一画像とその動きを指定し，コンピュータが効率良く動画を生成する．静止画を動かす問題の難しさは目的画像のシーンや物体の複雑さに依存し，その意味で流体画像を元にしたアニメーション生成は未だ難しい問題である．既存研究 [8, 26] はこの問題に解法を提案したがそれぞれ限界がある．本研究ではビデオデータベースを導入し，より変化に富んだ流体の動きの表現を可能にする．

流体画像を動かすには 3 つの既存手法がある．1 つは流体シミュレーションの適用だが目的画像の見た目を再現する様な物理パラメータの適切な設定が難しい．2 つ目は水面等の比較的穏やかな流体を扱う手法がある [8]．一方，本手法は水飛沫等、もっと激しい流体の動きや煙や炎のアニメーションも試みる．3 つ目は流体ビデオを指定し，その流体特徴を画像上に転送する手法である [26]．この手法は単一の流体ビデオを利用するのみ，またユーザが与える動き場も定常的なので再現可能な流体の動きに限界がある．目的画像に合う流体ビデオの検索作業も手間が掛かる．

これらの問題に対処すべく，ビデオデータベースを用いた流体画像のアニメーション手法を提案する (図 1)．数百のビデオを基にデータベースを構築し，既存手法と比べより質の高いアニメーションをより少ない労力で合成する．入力は目的画像と流体の動きの情報 (流れ方向と速さ) 及び流体領域のアルファマットである．流体の動きの情報は質の高いアニメーションの生成に重要だが，とりあえずの合成には必要なく，ユーザの労力が軽減されている．アルファマットはスケッチベースの画像切抜手法で簡単に作られる [23]．川，滝，煙，炎の様々な流体アニメーションを合成して提案手法の柔軟性を確認する．

本研究は 3 つの技術的提案を行う．1 つ目は既存の画像検索技術を流体画像のアニメーションへ適用する方法．データベースに基づくアプローチなのでなるべく多くのビデオが必要だが，例えば数万の流体ビデオを集めることは実際は難しい．そこで少ないビデオを効率良く利用するため，ビデオを小さな断片に分割する．ビデオ断片をベクトル量子化して bag-of-features の codebook を構築しビデオ断片を効率良く検索することを可能にする．2 つ目は流体ビデオを目的画像に割り当てる方法である．割り当て問題を複数ラベル割り当て問題としてマルコフ確率場 (MRF) を用いて定式化する．エネルギー最小化問題は α-expansion のグラフカットを用いて効率良く解ける．3 つ目はアニメーション合成手法の拡張である．既存手法は流体ビデオを 3 つに分割した．平均画像，動き場と残渣である [26]．本手法は 2 つの分割 (平均画像と差分) で十分である．結果として本手法の動き場は時間変化し結果のアニメーションの質が向上する．

2 既存研究

画像・ビデオデータベース 画像やビデオ検索は活発な研究分野である．近年，研究成果が画像やビデオの合成等，コンピュータグラフィックスに応用されている．何百万もの画像が画像の穴埋めに役立つ [14]．Sketch2Photo はユーザのラフなスケッチを元に画像検索を行い目的画像を合成する [7]．Skyfinder は希望の空画像を検索し見た目の修正や合成画像を生成する [30]．ビデオ合成に関して，SIFT Flow は単一画像の動き場の推定やビデオの一部を別のシーンへ転送することを可能にした [25]．Webcam Clip Art は野外シーンのビデオデータベースをシーンの照明編集に役立てた [20]．しかし，既存研究にビデオデータベースを流体画像のアニメーションに応用した例はない．

画像のアニメーション 単一画像のアニメーションは数多くの既存研究がある [11, 16, 13]．これらは画像の中に入っていく様な 3 次元的な視点の変更やキャラクタアニメーションに便利だが，流体アニメーションは扱っていない．Chuang らは確率的な動きを単一画像に与えた [8]．この手法は水面の揺れ等は扱えるが流れる流体は扱えない．Lin らは単一ではないが複数の高解像度画像から流体アニメーションを合成した [24]．

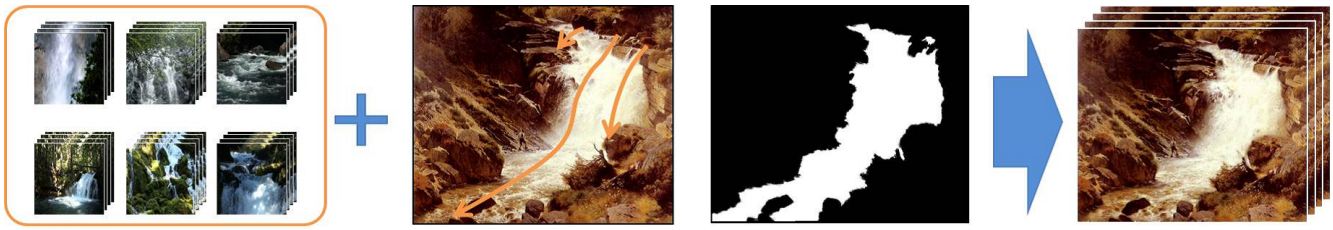


図 1: 提案手法の概要: (a) 流体ビデオデータベースの構築．(b) ユーザは目的画像と流体の動きを指定する．ここではオレンジの矢印で流れ方向を指定している．(c) 流体領域を指定するアルファマット．(d) システムが動画を合成する．

ビデオテクスチャの合成　ビデオテクスチャ合成の研究も多く存在する [31, 3, 27, 9] が，これらで合成した動画の見た目や動きを編集するのは難しい．Bhat らは手書きスケッチに基づいて流体ビデオを編集した [5]．見た目の編集も可能だが画像のアニメーションは扱っていない．Kwatra らはテクスチャをユーザ指定の動き場に沿って動かす手法を提案した [18] が，動き場が定常的で，また高周波な流体特徴が扱えない．岡部らは流体画像のアニメーション手法を提案した [26]．これは手作業で動き場を指定する必要や適切な流体ビデオを検索する必要があった．この手法をビデオデータベースを用いて拡張しユーザの労力の軽減と結果の質の向上を狙う．

3　システム概要

流体の目的画像に対し，本手法は適切なビデオをデータベースから選び，その一部を目的画像の一部に割り当て，最後にそれらを滑らかに統合してアニメーションを合成する．ユーザの入力は単一流体画像と流体領域のアルファマットである．更にスケッチによる流れ方向，ペイントによる速さの指定を行うと割り当て問題を解く際の制約となる．

本手法は 3 つの要素で成る: 1) 流体ビデオデータベースの構築 (図 2-a) では各流体ビデオを小さな断片に切り，2) 最適なビデオ断片を検索し目的画像の一部へ割り当て，3) 割り当てられたビデオ断片を滑らかに統合し見た目を調整してアニメーションを合成する．データベース構築処理のために流体ビデオを収集する (図 2-b)．ビデオの数を増やすため各ビデオを小さな断片に切る (図 2-c)．ビデオ断片のフレームの平均画像を作る (図 2-e)．平均画像とフレームの差分を計算する (図 2-f)．差分は色情報は持たず高周波な流体特徴のみを捉えている．データベース中の全ての平均画像を元に bag-of-features の codebook を構築し，各平均画像を visual word のヒストグラムで表現する (図 2-d)．

目的画像 (図 2-g) はデータベース構築と同様に断片に切られる (図 2-i)．各断片の visual word のヒストグラムを計算し (図 2-h)，ヒストグラム間の近傍探索を行い (図 2-d と h)，目的画像断片に良く似たビデオ断片を割り当てる．ユーザが動き場を指定した時はこの割り当て問題を解く際の制約に使われる．割り当ての結果に基づき差分が対応する目的画像断片にコピーされる (図 2-j)．最後に全ての差分を滑らかに統合し見た目を目的画像に合わせて調整することでアニメーションを作る (図 2-k)．

本手法の流体特徴を目的画像に転送する考え方は *Image Analogies* と似ている [15]．転送される差分と目的画像断片との関係が，その差分と平均画像との関係となるべく等しくなるように転送を行う．Image Analogies との差異は本手法はピクセルベースでなくパッチベースの合成であり，*Image Quilting* に似ている [10]．パッチを統合する際，Image Quilting はパッチ間の境界線を最適化を解いて求め滑らかさを保証するが，本手法は計算量の小さい画像ピラミッドを用いたアルファブレンドを利用する．ビデオ断片は平均画像と差分に分かれ，前者は低周波な見た目情報，後者は高周波な流体特徴を持つ．流体特徴は高周波な情報なのでアルファブレンドで滑らかに統合できる．

4　ビデオデータベース

可能な限り多くの流体ビデオを収集したいが，インターネット上の巨大なビデオ共有サイト等の存在にも関わらず，我々の必要条件を満たす流体ビデオの収集は難しかった．流体ビデオの条件は 3 つある．

- 固定カメラで撮影され流体自身がビデオの主人公．
- アニメーション合成に耐える程度の高解像度．
- 流体以外に顕著に動く物体等がない．

数百のビデオを収集し条件を満たすもの以外を捨て，結局 151 の流体ビデオを水シーンについて得た (図 3)．解像度は全ビデオ 640×360 か 480×360 である．

データ量の増やすため，流体ビデオを局所的に使う．例えば，ある流体ビデオの滝の形状が目的画像の滝の形状と違っても，水飛沫等の局所的な部分が目的画像の一部として使える可能性がある．

データベースのビデオ数を増やすため，左右反転したビデオを作り数を 2 倍にする．更にスケールと回転 (図 4-a) をした後，小さな断片に切る (図 2-c と図 4-b)．スケール

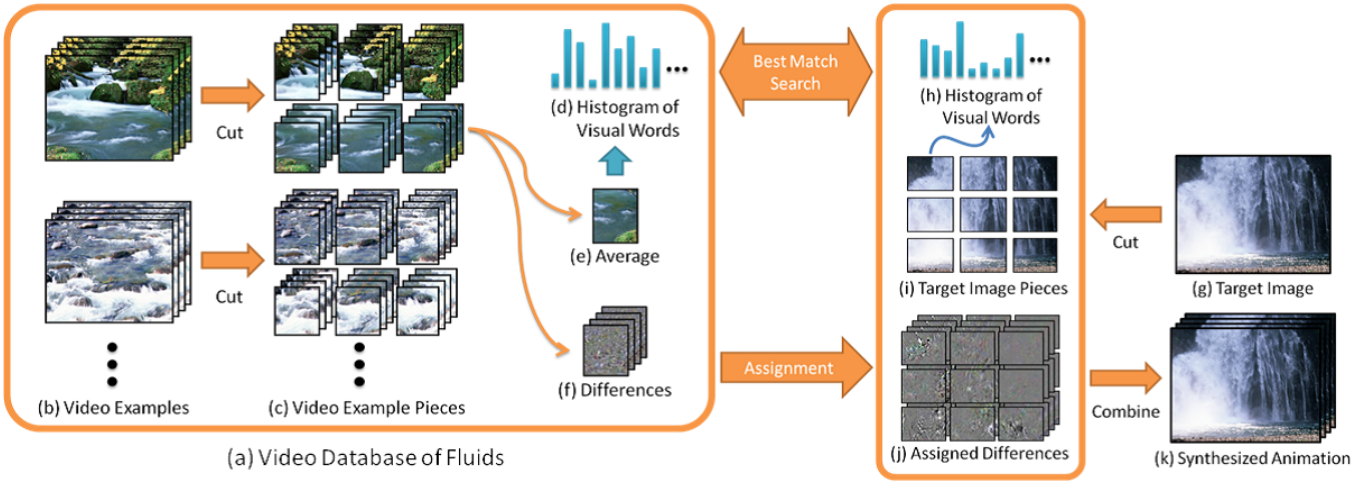


図 2: システム概要.

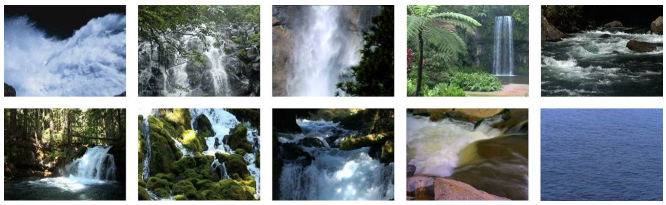
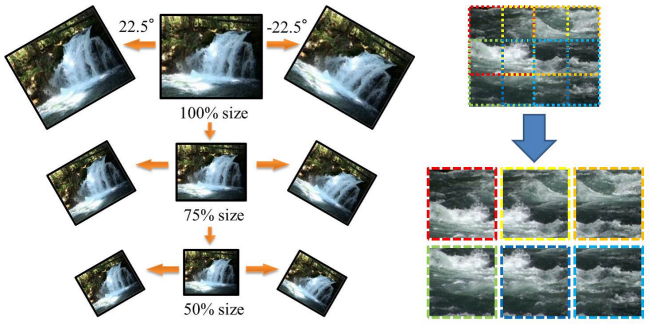


図 3: 水シーンのデータベースのサムネイル．流体ビデオは 3 つの条件を満たす: 1) 流体がビデオの主人公，2) 解像度は 640×360 か 480×360，3) 他に動く物体等がない．

は 3 段階，元のサイズの 50，75，100% を用意する．回転は元画像を -22.5，0.0，$+22.5$ 度回転する．各バージョンを小さな断片に切る．解像度は 48×48．断片は隣の断片と重なりを持つ (図 4-b).



(a) Scaling and rotating a video example. (b) Video example pieces.

図 4: データベース構築.

ビデオ断片は流体だけでなく岩，樹木，空等の他の静止物も含みデータベースの効率を下げる．静止物を除去するため動きを調べる．各断片でフレーム間のオプティカルフローを計算し (図 5-b)，全フレームの平均を求める (図 5-c)．流体ビデオの顕著な動きを持つピクセルのみを使用し他のピクセルはマスクで除く (図 5-d)．どのピクセルも顕著な動きを持たないビデオ断片はデータベースから完全に除去する．オプティカルフローは *OFLib* で計算する [33]．このアルゴリズムでは 3 つのパラメータ，θ，λ と繰り返し計算の回数を指定し，それぞれ 1.0，0.8，25 とした．流体ビデオは高周波な動き情報を持つが，これらのパラメータはそのような動きに敏感にならず，比較的滑らかな結果が得られるように調整したものだ．最終的に水シーンに関して 24 万個のビデオ断片を得た.

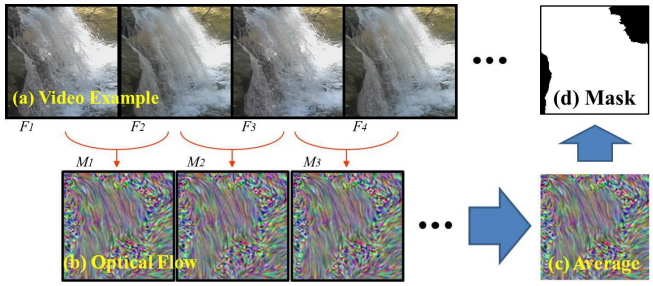


図 5: ビデオ断片の処理.

効率良いビデオ断片の検索及び割り当てのため，bag-of-features の codebook を構築し各断片を visual word のヒストグラムで表現する (図 2-d)．Bag-of-features は画像ピラミッドと合わせて使うことで小さなパッチの検索に有効である [28]．Bag-of-features はテクスチャの存在如何の情報を持つのみだが，画像ピラミッドはテクスチャの位置情報を付加し，これは小さなパッチの検索でも重要である．フレームの平均を取って各ビデオ断片の代表画像を作り，これを基に codebook を構築する (図 2-e)．平均を取ることで流体写真に見られるモーションブラーを考慮している: 写真家はしばしば，流体の流れを軌跡として表現するため意図的にモーションブラーを掛ける．各平均画像上で 128 次元の SIFT 特徴を記述する．48×48 の解像度上の 9×9 グリッドの各頂点で SIFT 特徴を記述する．グリッドで SIFT 特徴を記述するのはガウス差分によって検出されるキーポイントを利用するより画像検索においてしばしば良い精度

を出すことが知られている [21].

抽出した SIFT 特徴に対しベクトル量子化を行い visual word を得て codebook を作る．量子化には repeated cluster bisectioning を用いた．これは K-means より高速で精度が良いことが報告されている [32]．K-means と同様，クラスタ数 k を指定する．本実験では 100，200，300，500 を試した上で，200 を最適値として用いた．codebook を得て，各ビデオ断片の平均画像のヒストグラムを作る．各 SIFT 特徴を最も近い visual word に割り当て，画像ピラミッドを用いてヒストグラムを記述する [22].

5　ビデオの割り当て

目的画像はデータベース構築と同じ方法 (図 4-b) で断片に分割する (図 2-i)．目的画像断片にビデオ断片を割り当てる際，3 つの条件を考える: 1) 目的画像断片とビデオ断片の見た目の類似性，2) 隣同士の断片の動きの滑らかさ，3) 隣同士の断片の見た目の類似性．1 つ目の条件によって目的画像断片とビデオ断片の見た目が類似すればするほど，両者は同じ動きを持つ可能性が高くなる．2 つ目の条件は目的画像断片がある動きを持つ場合，その隣の断片にも同様の動きを割り当てようとする．つまり，隣接する断片間で速度及びビデオテクスチャパターンを類似させる．3 つ目の条件は隣同士の断片が一貫性のない見た目を持つことを防ぐ.

1 つ目の条件は bag-of-features のヒストグラムを比較し見た目の類似性を計算する (図 2-d と h)．データベース構築と同様，目的画像断片上で SIFT 特徴を抽出し目的画像断片を visual word のヒストグラムで記述する．見た目の類似性はヒストグラム交差で計算する:

$$I(H_e, H_t) = \sum_{i=1}^{k} min(H_e[i], H_t[i]), \tag{1}$$

ここで H_e と H_t はビデオ断片及び目的画像断片の visual word のヒストグラムである.

2 つ目の条件では，動き場の平均 (図 5-c) を用いて隣接する断片間でこの類似性を測定することが考えられる．しかし，これだけでは図 6 に示す様に不十分である．各ビデオ断片は上から下へ同様の速さで流れる水の動画で似た平均の動き場を持っている．しかし，これらの断片が隣接して割り当てられると結果のアニメーションでビデオテクスチャパターンの違いが不自然に見える．図 6-a は小さな滝を近距離で録画したもので多くの水飛沫が存在しビデオテクスチャは揺らいでいる．一方，図 6-b は大きな滝を遠距離で録画したものでビデオテクスチャは滑らかである．この違いはフーリエ変換で解析できる．図 6-a は全ての周波数で一様に強い係数を持つが，図 6-b は直流成分のみが強く高周波成分は弱い．フーリエ変換を各ピクセル x で計算すると，$\hat{M}(x) = \mathcal{F}(M(x))$．$\mathcal{F}$ は 1 次元離散フーリエ変換とパワースペクトルへの変換．$M(x)$ は x における動き場の列，つまり，$[M_1(x), M_2(x), ..., , M_N(x)]$．$N$ はフレーム数．隣接動き場 M^i と M^j 間の滑らかさの定義は,

$$smooth(M^i, M^j) = d(\hat{M^i}_1, \hat{M^j}_1) + \sigma \sum_{f=2}^{N/2} d(\hat{M^i}_f, \hat{M^j}_f), \tag{2}$$

$$d(F, G) = \sum_{x \in \Omega} (F(x) - G(x))^2, \tag{3}$$

ここで，Ω は隣り合う断片 F と G の重なり合った部分のピクセル集合，σ は平均動き場に相当する直流成分と高周波成分をバランスするパラメータである.

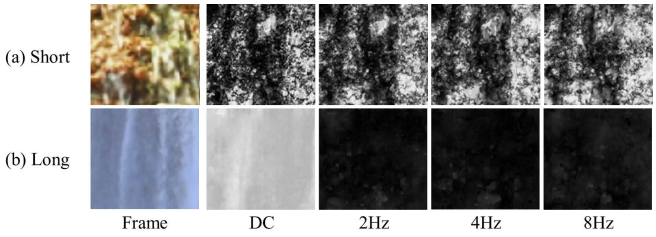


図 6: 動き場のフーリエ解析.

3 つ目の条件は，割り当てられるビデオ断片のテクスチャ度を記述し隣接する断片間でその類似性を測定することで定義する．図 7 にこの条件の必要性を示す．2 つの煙のビデオ断片の見た目が異なる: 1 つは高くもう 1 つは低いコントラストである．この 2 つは同様の動き場を持つため 1 つ目と 2 つ目の条件が隣接することを許し結果として高いコントラストと低いコントラストが隣接する一貫性のない不自然な見た目を作る可能性がある．これは 1 つ目の条件が SIFT 特徴に基づく bag of features であり照明変化を吸収してしまうためである．この問題を解決するため，ビデオ断片の各フレームのテクスチャ度を 4 レベルの Laplacian ピラミッドで測定し全フレームの合計を取って記述する; この処理は図 7-a と b の区別を可能にする．前者は後者に比べ強い係数を全ての周波数帯で持っている．ビデオ断片のテクスチャ度 T は $48 \times 48 \times 4$ 次元で隣接断片間の類似度 $texture(T^i, T^j)$ は Eq. 3 と同様に重なった領域に対しユークリッド距離で計算する.

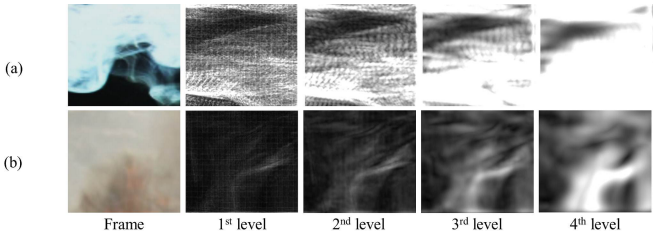


図 7: Laplacian ピラミッドを用いたテクスチャ度の解析.

割り当て問題を解くため，各目的画像断片に見た目の類似度 (Eq. 1) を用いて複数のビデオ断片候補を選択する.

候補の数は 100，つまり各目的画像断片に 100 のラベルを持たせ最適なラベル割り当て問題を MRF で定式化し解く:

$$\arg\min_i E = \sum_p V_p(l_i) + \lambda \sum_{(p,q)} W_{p,q}(l_i, l_j), \qquad (4)$$

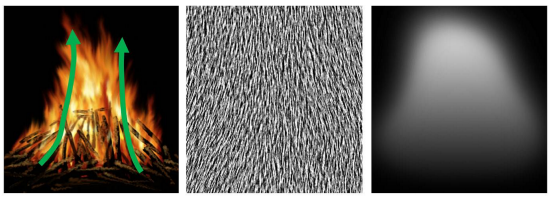
ここで，ラベル l_i と l_j が隣接する目的画像断片 p と q に割り当てられる．V_p はデータ項で $W_{p,q}$ は滑らかさ項で，ρ パラメータを使って定義される:

$$V_p(l_i) = I(H_p, H_{l_i}), \qquad (5)$$

$$W_{p,q}(l_i, l_j) = smooth(M^{l_i}, M^{l_j}) + \rho \cdot texture(T^{l_i}, T^{l_j}). \qquad (6)$$

このエネルギー最小化問題は α-expansion のグラフカット [6] で効率良く解くライブラリがある [29]．グラフカット処理後，各目的画像断片に 3 つの条件を満たす最適なビデオ断片が割り当てられている．

ユーザ指定の動き場　ユーザは手作業で動き場を指定し割り当て処理を制御することで流体の動きをデザインしたり間違った割り当てを防ぐことができる．動き場の編集作業を簡潔にするため，動き場は方向と速さの 2 要素に分割される [26]．方向マップの指定にはスケッチを描き (図 8-a)，この疎な情報は同径基底関数で補間する (図 8-b)．速さマップはグレースケール画像なのでペイントソフトで簡単に編集できる (図 8-c)．方向マップと速さマップは両方指定しても良いし 1 つだけでも良い．速さマップはゼロから作ると難しいが，割り当て処理を一度行い，その結果を編集するとやりやすい．編集された動き場は 100 の候補選択に影響する．選択前に指定された動き場と著しく異なる動きのビデオ断片を予め捨てることで絞込みを行う．



(a) Target Image　(b) Orientation Map　(c) Speed Map

図 8: ユーザ指定の動き場．ここでユーザは炎の動きを下から上へ緑の矢印で指定 (a と b)．速さマップで下部は遅く上部は速く指定している (c)．

6　レンダリング

割り当てられたビデオ断片の統合　割り当て処理後，各目的画像断片にビデオ断片が割り当てられている．各断片は異なるビデオの異なる部分であり，いいかげんに統合すると断片間の切れ目が見えてしまう．滑らかな統合に向け，ビデオ断片を平均画像と差分に分割する．各ビデオ断片の全フレームを平均する: $A = \frac{1}{N}\sum_{i=1}^{N} F_i$．$F_i$ はビデオ断片のフレーム，N はフレーム数である．差分は，$D_i = F_i - A$ である．差分は高周波な流体特徴のみを持つのでアルファブレンドで切れ目を生じることなく統合できる (図 9)．隣接するビデオ断片の重なり部分を滑らかに繋ぐため，画像ピラミッドを適用し各周波数帯でアルファブレンドする [1]．低周波を強くブレンドし高周波は保存しようとするため普通のアルファブレンドより良い結果を得る．

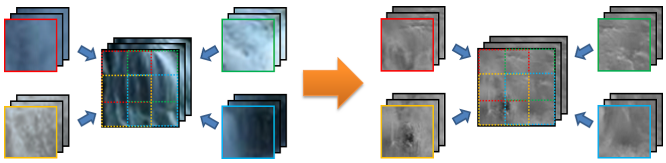


図 9: 隣接ビデオ断片の統合．左に目的画像へのビデオ断片の割り当てを示し，右に対応する差分のコピーと画像ピラミッドを用いたアルファブレンドによる統合を示す．

見た目の復元　統合された差分は色情報を持たない．見た目を復元するため平均画像 (図 2-e) に対応する画像を合成し統合した差分に加える．この近似的な平均画像は目的画像に擬似的モーションブラーを掛けることで得る [2]．割り当てられたビデオ断片から動き場を得てモーションブラーに使う．この時、擬似シャッタースピードのパラメータを適宜設定する必要がある．ノイズ除去のためガウスブラーを追加で掛ける．カーネルサイズは動き場の大きさに比例した値を用いた．

見た目のマッチング　最後に合成されたアニメーションの各フレームと目的画像の間でヒストグラムマッチングを取り見た目のマッチングを行う [26]．Heeger と Bergen のテクスチャ合成技術を用いる [12]．局所的な見た目のマッチングを行うため画像空間を一様なグリッドに切って各領域毎に独立にマッチングを行う．

動的な流体の境界　上記の手法で遠距離から見た滝など全体的な形状に変化のない流体アニメーションは合成できるが，炎や煙の場合は不自然な結果になるので動き場に沿ってアルファマットを動かす [4] ことで動的な流体の境界を表現する．上記の手法で合成したアニメーションの動き場を $\phi_i(i$ は時刻) として，画像のワーピング関数 $W_i(I)$ を画像 I を ϕ_i に従って変形させる関数として定義する．アルファマット α を元に時間的に変化するアルファマット B_i を合成する: $B_{i+1} = \tau W_i(B_i) + (1-\tau)\alpha$．$B_1$ は α に等しく τ は B_i の動きをコントロールする．また B_i の面積が一定になるように毎フレーム調整を掛ける．動的な境界は手作業での背景画像の合成が必要となる．流体の外形が変わる時，今まで見えなかった背景部分が見えるためである．本実験では Adobe Photoshop のスタンプツールを用いて 10 分以内の手作業で合成したものを使用している．

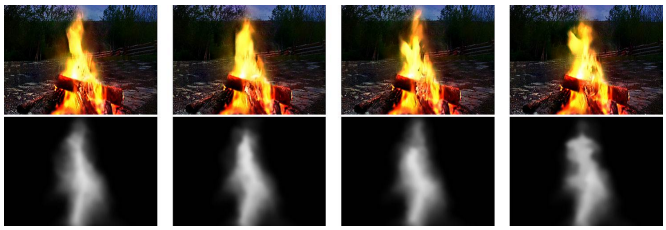


図 10: 炎のアニメーションに使用した動的なアルファマット．

7　結果と考察

水，炎，煙の各シーン毎に独立したビデオデータベースを構築した．それぞれ 151，96，89 の流体ビデオを持ち，24 万，22 万，19 万のビデオ断片を得た．写真と絵画を含む目的画像に対しアニメーション合成を実験した (図 1 と 11)．図 1 は *tour into the picture* と合わせてアニメーション合成を試みた [11]．スケッチベースの画像切抜手法を用いアルファマットを作成した [23]．この処理は 5 分程度を要した．流体の流れ方向マップは全ての目的画像に指定した．方向マップの作成は何本かのストロークをスケッチするのに数十秒〜1 分を要した．速さマップは図 1， 11-d， 11-g に用意した．速さマップをゼロから作るのは難しいので，まずアニメーションを合成して得た速さマップをペイントソフトに読み込んで編集した．

図 11: 実験した目的画像．写真 (a,b,c,e,h) と絵画 (d,f,g)．

ビデオ断片の割り当て　図 11-e と f の結果では提案手法がビデオ断片を上手く割り当てたことが分かる．割り当てられたビデオ断片の結果を見ると既に目的画像と類似の見た目を持っている．図 11-e では上部に強い炎，薪の周辺に小さい炎が割り当てられた．図 11-f は滝，白い水飛沫，穏やかな水面等の領域があるが，各々の領域に適切なビデオ断片が割り当てられた．滝や炎はユーザ指定の方向マップや速さマップの入力がなくても正しい割り当てを得ることが多い．これは流体の動きが画像のエッジ情報と強い相関を持つためである．例えば，滝や炎は常に上から下，下から上へそれぞれ流れる．高周波な領域は遅く詳細な動きを持つ傾向がある．一方，図 11-a と h の様な水面は流れ方向が曖昧だ．また，煙の流れ方向も決めることが難しい．理由は画像のエッジ情報が流れ方向や速さと相関を持たないためである．見た目と動きの間の相関は絵画の場合に欠如する場合が多い．この様な場合，方向マップ，速さマップの指定が特に重要になる．

計算時間　アニメーションの第 1 バージョンの合成に 1 時間程度掛かる．*Intel(R) Xeon(R) 3.33 GHz* を計算に使い，リモートサーバをデータベースに使用している．割り当て問題を解くのに半時間，残りの半時間でレンダリングする．割り当て処理はグラフカット等の計算ではなく Eq. 6 の滑らかさ項の計算がボトルネックである．MRF の各ノードが 100 の候補を持つので，100×100 のエッジがノード間に張られている．各エッジの計算に 2 つの動き場を読み込んで処理する際のディスクアクセスがボトルネックである．この効率化は将来課題で，現在は断片化している全てのデータを 1 つの巨大ファイルにまとめ，ディスクのランダムアクセスを軽減することを検討している [19]．

パラメータ　割り当て処理に 3 つのパラメータがある．Eq. 2 では σ で直流と高周波成分をバランスする．Eq. 4 の λ はデータ項と滑らかさ項をバランスする．Eq. 6 の ρ は動き場とテクスチャ度をバランスする．実験で λ は結果に大きな影響を与えないことが分かった．100 のビデオ断片を選択した時点でデータ項の条件は既にある程度満たされるためである．σ は結果に影響を与え，大きいと直流成分つまり平均動き場が無視され結果のアニメーションの動きが乱雑になる．ρ も重要だが σ に比べると影響が小さい．実験では λ を 0.01，σ を 100 か 1000，ρ を $\sigma \times 100$ とした．

既存手法との比較　岡部らの手法 [26] との違いは流体ビデオの分割である．既存手法は差分を更に動き場と残渣に分割した．このような流体アニメーションのモデルはユーザが定常的な動き場を自由にデザインすることを可能にするが，定常的な動き場は与えた流体ビデオが持つオリジナルの流体特徴を再現できない．加えて，この分割は動き場と残渣の間の重要な相関を捨ててしまい，結果のアニメーションが粘性を持ったようになる．図 11-a における比較結果を見ると，流体ビデオの水飛沫等の特徴が結果のアニメーションに生かされておらず，無関係な合成ノイズが定常的な動き場に沿って流れているように見える．一方，本手法の結果では割り当てられたビデオ断片の流体特徴が結果のアニメーションでも良く再現されていることが分かる．ユーザの労力も削減されており，図 1 の結果を得るため，既存手法は画像を滝の部分と川の部分に分けて，それぞれに流体ビデオを独立に割り当てる必要があった．一方，本手法では画像全体を一度に処理できた．

7.1　ユーザテストと提案手法の限界

アニメーションの質を評価するためユーザテストを行った．2 人のプロのアニメータと 14 人の情報科学科の学生が参加した．各被験者は図 1 と 11 の結果を見て不自然な部分がないかに注意しながら質を評価した．スケールは 5 段

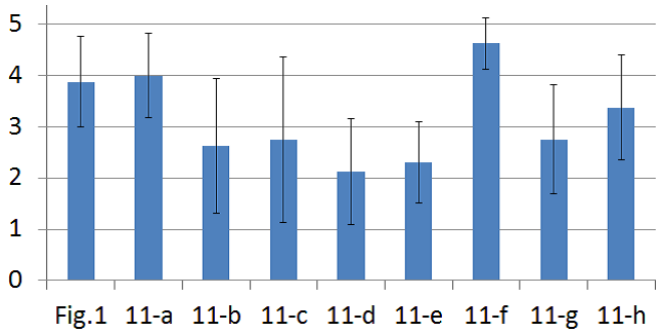


図 12: ユーザテスト．青棒は平均点，誤差棒は標準偏差．

階で 1 と 5 がそれぞれ最低点と最高点である．1 と 5 を少なくとも 1 つの結果に割り振ってもらうことで評価は相対的に行った (図 12)．最高平均点は図 11-f が得た．標準偏差も小さく，この水のシーンのアニメーションは自然であることにほぼ全員の被験者が同意したと言える．他の全ての水のシーンも 1 点は付かず，5 点をいくつか貰えたので，我々の手法は水のシーンについては自然なアニメーションを合成できると言えそうだ．一方，炎と煙のシーンは点数が低かった．この結果を踏まえ，結果の質について議論する．図 11-d は本手法の典型的な失敗例と考えている．

ビデオ断片割り当ての失敗　図 11-d は最低点を得た．複数の被験者はアニメーションが粗いとコメントした．煙の見た目が場所によって異なって見え動きも乱雑である．これはビデオ断片の割り当ての失敗によるもので 2 つの大きな理由がある．1 つは目的画像は抽象的な油絵で平坦な部分が多く 100 のビデオ断片候補の選択に失敗している．2 つ目は割り当てられたビデオ断片が乱雑な動きを作ったことである．流れの方向マップと速さマップは指定したので煙の動きは全体的には下から上への流れとなったが，局所的な動きはこれらのマップで制約できずランダムになっている．このことは単一の動き場では不十分で，より洗練された動きの制御機能が必要であることを示唆している．

ノイズ　本手法の別の限界はビデオデータベースにノイズがあると結果のアニメーションに継承されてしまうことだ．実験で用いたビデオは quicktime コーデックで圧縮されている．また効率良く実験するため各フレームを抽出して JPEG 形式で保存している．結果，ブロックノイズやインターレースノイズがデータベースに存在する．これらは継承されるだけでなく，しばしば目的画像の見た目を復元する処理がコントラストを上げようとする場合は誇張される．例えば複数の被験者が図 11-d の下部の煙にノイズが見えるとコメントした．実際に JPEG ブロックノイズが見える (図 13)．本手法の実用化の際は，大きなストレージを用意しロスレス画像圧縮を用いることが望ましい．

目的画像の見た目の再現　複数の被験者が結果のアニメーションの見た目が目的画像と異なるとコメントした．実際，

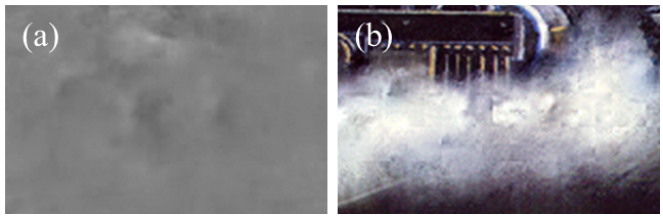


図 13: (a) 割り当てられたビデオ断片はコントラストが低い．(b) 見た目の復元処理がコントラストを上げ，ビデオ断片に隠れていたブロックノイズが浮き上がった．

目的画像の見た目を再現するのは難しい場合があり，特に強いエッジという重要な高周波情報がある場合は難しい．例えば，図 11-c に煙の軌跡の特徴的な見た目がある．しかし，結果のアニメーションはこの形を再現せず，軌跡は消えてしまう．同じことが図 11-e の炎にも言え, 元の炎の形状が正確に再現されない．炎や煙の形状を保存できないことは本手法の限界である．この様な目的画像の見た目の詳細な部分まで再現できるほどデータベースが大きくない．

照明効果　しばしば流体部分のみ動き，他の部分が静止しているのは不自然だとコメントされた．特に図 11-e の炎が動くと地面の照明効果を変化させるはずである．しかし，照明効果や他の物体の動きは本研究の目的範囲外である．

8　結論と将来課題

ビデオデータベースを利用して単一流体画像のアニメーションを合成した．ユーザは目的画像とアルファマットを入力する．流体の流れ方向や速さを指定し流体アニメーションを制御できる．既存手法と比べ本手法はより高い質のアニメーションを小さな労力で作成することが出来た．提案手法はいくつかの将来課題を残している．現在，3 つの方向性を検討中である．

より効率的なビデオデータベースの利用　限界の 1 つに目的画像の詳細な見た目が保存できないことがある．データベース中の流体ビデオの不足が原因であるが, ビデオの数を増やすことに加え，より柔軟なビデオの利用を検討中である．例えば現在はパッチサイズが 48×48 と固定であるが，もしビデオの変形を考慮出来れば [5]，目的画像の見た目がより良く再現できるはずである．

より知的なシステム　提案手法はビデオを切って割り当てるが，多くの割り当ての失敗は大局的な情報の欠如に起因する．もしシステムがシーン構造や流体の状態を理解すれば，手法はもっと自動的になる．シーン構造を理解して流体の速度を決定できれば，例えば見た目が同じ煙でも煙突のものか爆発のものか判断できる．既存手法に SIFT Flow があり [25]，キャンプファイアの画像にキャンプファイアの動画をフィットさせて動き場を推定している．この様な

技術を組み込んだ，より知的なアニメーション合成手法を検討中である．

より動的な流体アニメーション　提案手法は水飛沫等の高周波な流体の合成に強い．しかし，流れる雲の形状等，流体の大きな動きの再現は困難だ．このような大きな動きは将来実現したく，そのために画像中の物体抽出の技術 [17] を利用して大きく意味のあるビデオ領域を効率良く取り出し目的画像に割り当てるような手法を検討中である．

参考文献

[1] Burt P. J., Adelson E. H. A multiresolution spline with application to image mosaics. ACM Trans. Graph. 2, 4, pp.217–236, 1983.

[2] Brostow G. J., Essa I. Image-based motion blur for stop motion animation. In Proc. SIGGRAPH 2001, pp.561–566, 2001.

[3] Bar-Joseph Z., El-Yaniv R., Lischinski D., Werman M. Texture mixing and texture movie synthesis using statistical learning. IEEE Trans. Vis. Comput. Graph. 7, 2, pp.120–135, 2001.

[4] Bousseau A., Neyret F., Thollot J., Salesin D. Video watercolorization using bidirectional texture advection. In Proc. SIGGRAPH 2007, pp.104, 2007.

[5] Bhat K. S., Seitz S. M., Hodgins J. K., Khosla P. K. Flow-based video synthesis and editing. ACM Trans. Graph. 23, 3, pp.360–363, 2004.

[6] Boykov Y., Veksler O., Zabih R. Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23, 11, pp.1222–1239, 2001.

[7] Chen T., Cheng M.-M., Tan P., Shamir A., Hu S.-M. Sketch2photo: internet image montage. In Proc. SIGGRAPH Asia 2009, pp.1–10, 2009.

[8] Chuang Y.-Y., Goldman D. B., Zheng K. C., Curless B., Salesin D. H., Szeliski R. Animating pictures with stochastic motion textures. In Proc. SIGGRAPH 2005, pp.853–860, 2005.

[9] Doretto G., Chiuso A., Wu Y. N., Soatto S. Dynamic textures. Int. J. Comput. Vision, 51, 2, pp.91–109, 2003.

[10] Efros A. A., Freeman W. T. Image quilting for texture synthesis and transfer. In Proc. SIGGRAPH 2001, pp.341–346, 2001.

[11] Horry Y., Anjyo K., Arai K. Tour into the picture: using a spidery mesh interface to make animation from a single image. In Proc. SIGGRAPH '97, pp.225–232, 1997.

[12] Heeger D. J., Bergen J. R. Pyramid-based texture analysis/synthesis. In SIGGRAPH1995, pp.229–238, 1995.

[13] Hornung A., Dekkers E., Kobbelt L. Character animation from 2d pictures and 3d motion data. ACM Trans. Graph. 26, 1, pp.1, 2007.

[14] Hays J., Efros A. A. Scene completion using millions of photographs. In Proc. SIGGRAPH 2007, pp.4, 2007.

[15] Hertzmann A., Jacobs C. E., Oliver N., Curless B., Salesin D. H. Image analogies. In Proc. SIGGRAPH 2001, pp.327–340, 2001.

[16] Igarashi T., Moscovich T., Hughes J. F.: As-rigid-as-possible shape manipulation. ACM Trans. Graph. 24, 3, pp.1134–1141, 2005.

[17] Kuo Y.-H., Chen K.-T., Chiang C.-H., Hsu W. H. Query expansion for hash-based image object retrieval. In ACM Multimedia, pp.65–74, 2009.

[18] Kwatra V., Essa I., Bobick A., Kwatra N. Texture optimization for example-based synthesis. In Proc. SIGGRAPH 2005, pp.795–802, 2005.

[19] Ke Y., Sukthankar R., Huston L. Efficient near-duplicate detection and sub-image retrieval. In ACM Multimedia, pp.869–876, 2004.

[20] Lalonde J.-F., Efros A. A., Narasimhan S. G. Webcam clip art: appearance and illuminant transfer from time-lapse sequences. ACM Trans. Graph. 28, 5, pp.1–10, 2009.

[21] Li F.-F., Perona P. A bayesian hierarchical model for learning natural scene categories. In Proc. of CVPR '05, pp.524–531, 2005.

[22] Lazebnik S., Schmid C., Ponce J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proc. of CVPR '06, pp.2169–2178, 2006.

[23] Li Y., Sun J., Tang C.-K., Shum H.-Y. Lazy snapping. ACM Trans. Graph. 23, 3, pp.303–308, 2004.

[24] Lin Z., Wang L., Wang Y., Kang S. B., Fang T. High resolution animated scenes from stills. IEEE Trans. Vis. Comput. Graph. 13, 3, pp.562–568, 2007.

[25] Liu C., Yuen J., Torralba A., Sivic J., Freeman W. T. Sift flow: Dense correspondence across different scenes. In ECCV 2008, pp.28–42, 2008.

[26] Okabe M., Anjyo K., Igarashi T., Seidel H.-P. Animating pictures of fluid using video examples. Computer Graphics Forum, 28, 2, pp.677–686, 2009.

[27] Schödl A., Szeliski R., Salesin D. H., Essa I. Video textures. In Proc. SIGGRAPH 2000, pp.489–498, 2000.

[28] Sun M., Su H., Savarese S., Fei-Fei L. A multi-view probabilistic model for 3d object classes. In CVPR 2009, pp.1247–1254, 2009.

[29] Szeliski R., Zabih R., Scharstein D., Veksler O., Kolmogorov V., Agarwala A., Tappen M., Rother C. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. IEEE Trans. Pattern Anal. Mach. Intell. 30, 6, pp.1068–1080, 2008.

[30] Tao L., Yuan L., Sun J. Skyfinder: attribute-based sky image search. In Proc. SIGGRAPH 2009, pp.1–5, 2009.

[31] Wei L.-Y., Levoy M. Fast texture synthesis using tree-structured vector quantization. In Proc. SIGGRAPH 2000, pp.479–488, 2000.

[32] Zhao Y., Karypis G., Fayyad U. Hierarchical clustering algorithms for document datasets. Data Min. Knowl. Discov. 10, 2, pp.141–168, 2005.

[33] Zach C., Pock T., Bischof H. A duality based approach for realtime tv-l1 optical flow. In Pattern Recognition, pp.214–223, 2007.