

大量の映像データを用いる CG手法

2012年7月20日

岡部 誠

電気通信大学 / JST さきがけ

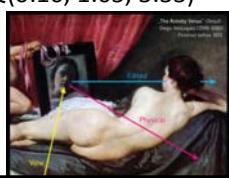
お話の流れ

- ・自己紹介
- ・映像データを分析してCGに応用する技術
- ・ビデオデータベースに基づく映像合成
 - 画像中の流体を動かす
 - 楽曲を演奏しているように見える動画生成



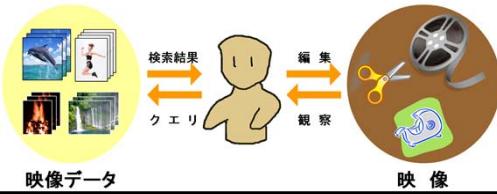
自己紹介

- ・CG製作のためのユーザインターフェース
 - 例:3次元CGの照明編集ツール(0:19, 3:41)
- ・映像製作を効率化したい
 - 人にとって直感的な入力を扱いたい
- ・反射の編集(0:10, 1:05, 3:33)



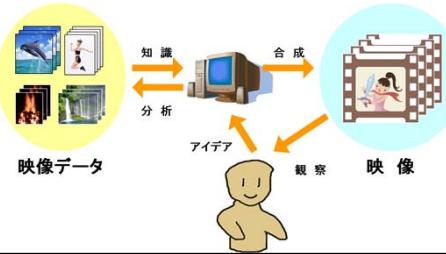
背景

- ・大量の動画が利用可能
 - デジタルビデオカメラ、携帯電話のカメラ
 - YouTube、他の動画データベース
- ・有効利用されていない
 - ダウンロード、切り貼り



やりたいこと

- ・大量の動画を分析、利用
 - 写真を動かす
 - エキストラ合成
 - 背景音合成



既存研究

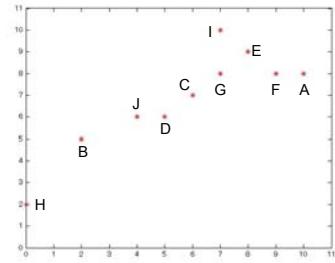
- ・大量音声の利用は高い実用性
 - text to speech、楽曲との合成
- ・大量画像の利用
 - 検索と画像処理による穴埋め
- ・大量動画の利用はこれから



映像データの分析に基づくCG合成手法

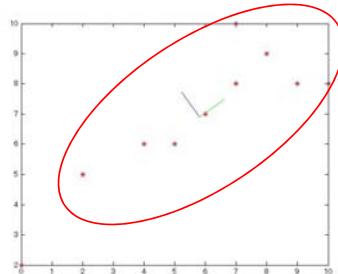
主成分分析

	国語	算数
A	10	8
B	2	5
C	6	7
D	5	6
E	8	9
F	9	8
G	7	8
H	0	2
I	7	10
J	4	6



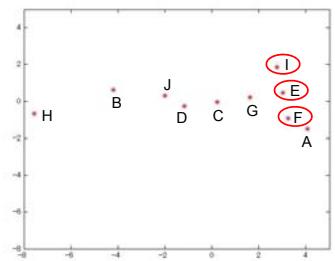
主成分分析

	国語	算数
A	10	8
B	2	5
C	6	7
D	5	6
E	8	9
F	9	8
G	7	8
H	0	2
I	7	10
J	4	6



主成分分析

	国語	算数
A	10	8
B	2	5
C	6	7
D	5	6
E	8	9
F	9	8
G	7	8
H	0	2
I	7	10
J	4	6

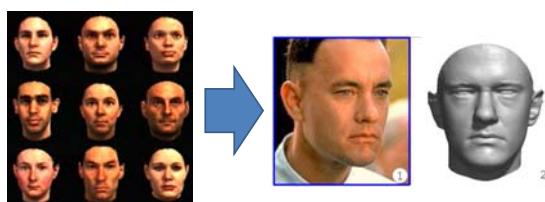


第一主成分のみで学生を評価する。

- 2つの数字を用いなくてよくなる → 次元削減
- より合理的に学生を評価できる

A morphable model for the synthesis of 3D faces
Volker Blanz and Thomas Vetter, SIGGRAPH 99

- 200個の顔のスキャンデータを主成分分析
 - 男女100ずつ、70000頂点のポリゴン(0:26)
- 写真1枚で顔の3次元形状が推定できる(2:10)
 - 推定すべきパラメータが200個に削減(4:20,4:52)



体格の編集 → MovieReshape

Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, Christian Theobalt, SIGGRAPH Asia 2010

- 550個の人体のスキャンデータを主成分分析
- 人体モデルでビデオをトラッキングする
- 体格を編集可能(0:06,0:54,2:50)



A Data-Driven Reflectance Model

Wojciech Matusik et al., *SIGGRAPH 2003*

- リアルな物質表現



A Data-Driven Reflectance Model

Wojciech Matusik et al., *SIGGRAPH 2003*

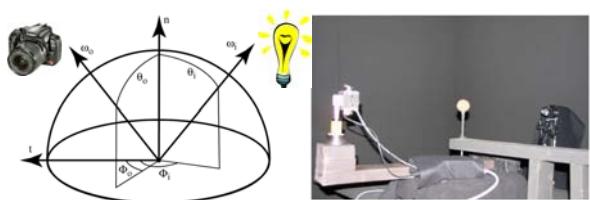
- リアルな物質表現 → 反射データを測定



A Data-Driven Reflectance Model

Wojciech Matusik et al., *SIGGRAPH 2003*

- リアルな物質表現 → 反射データを測定
 - 4次元データ → 360^4 のデータ量
 - 主成分分析とmanifold系非線形分析
 - 15次元



A Data-Driven Reflectance Model

Wojciech Matusik et al., *SIGGRAPH 2003*

- リアルな物質表現 → 反射データを測定
 - 4次元データ → 360^4 のデータ量
 - 主成分分析とmanifold系非線形分析
 - 15次元
 - 拡散具合、輝き具合
 - 金属っぽい
 - プラスチックっぽい
 - 粗い
 - 赤、緑、青っぽい
 - 埃っぽい、ゴムっぽい
 - 布っぽい、etc.



Automatic Photo Pop-up_(0:16,1:16)

Derek Hoiem et al., *SIGGRAPH 2005*

- 空、建物、地面の3種類にセグメンテーション
- 建物と地面の間で、90度に折り曲げる
- セグメンテーション済みの画像データセット
 - 3つのどの領域に入るかを学習



Photo Tourism: Exploring Photo Collections in 3D

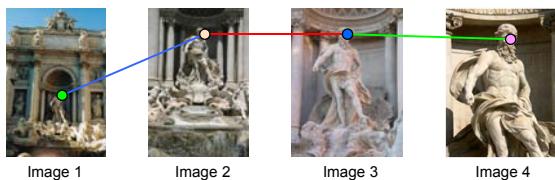
Noah Snavely, Steven M. Seitz, Richard Szeliski, *SIGGRAPH 2006*

- SIFT特徴とRANSACによるマッチング(2:32)
- マッチングを元にstructure from motion
 - 3次元構造物とカメラ位置を特定
- 閲覧のためのユーザインターフェースの提案



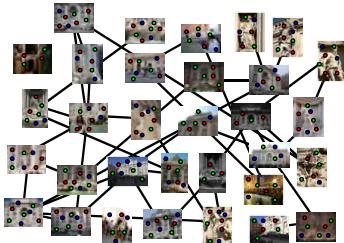
SIFT特徴とRANSACによるマッチング [Snavely et al., 2006]

- 画像内にSIFT特徴を検出 [Lowe, IJCV 2004]



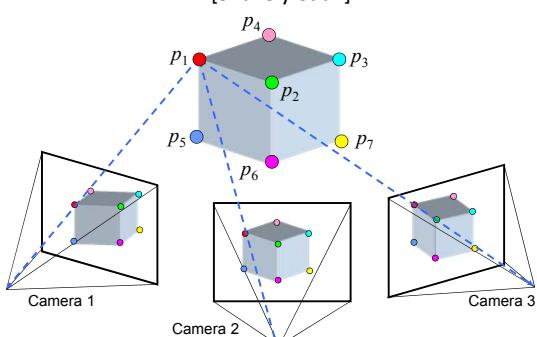
SIFT特徴とRANSACによるマッチング [Snavely et al., 2006]

- SIFT特徴同士を繋いで、
画像データベースをグラフ構造化



Structure from motion

[Snavely et al.]



Scene Completion Using Millions of Photographs

James Hays, Alexei A. Efros, **SIGGRAPH 2007**

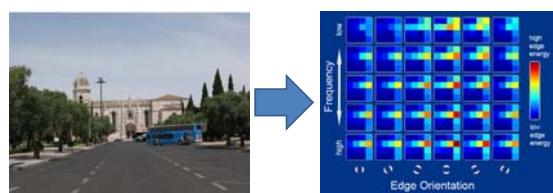
- 画像の気に入らないところを削除 & 穴埋め
- GIST特徴量による高速な検索



Scene Completion Using Millions of Photographs

James Hays, Alexei A. Efros, **SIGGRAPH 2007**

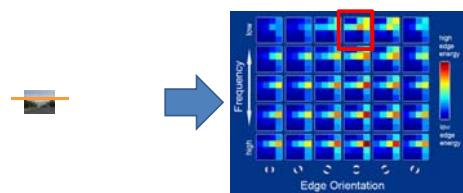
- 画像の気に入らないところを削除 & 穴埋め
- GIST特徴量による高速な検索
– $5 \times 6 \times 4 \times 4 = 480$ 次元 $\leftarrow 1000 \times 1000 \times 3 = 3M$ 次元



Scene Completion Using Millions of Photographs

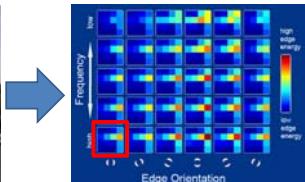
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

- 画像の気に入らないところを削除 & 穴埋め
- GIST特徴量による高速な検索
– $5 \times 6 \times 4 \times 4 = 480$ 次元 $\leftarrow 1000 \times 1000 \times 3 = 3M$ 次元



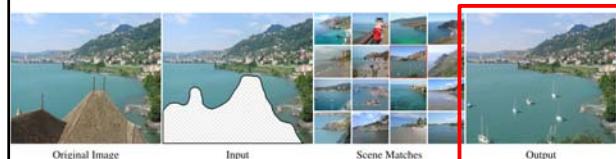
Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

- 画像の気に入らないところを削除 & 穴埋め
- GIST特徴量による高速な検索
 - $5 \times 6 \times 4 \times 4 = 480$ 次元 $\leftarrow 1000 \times 1000 \times 3 = 3M$ 次元



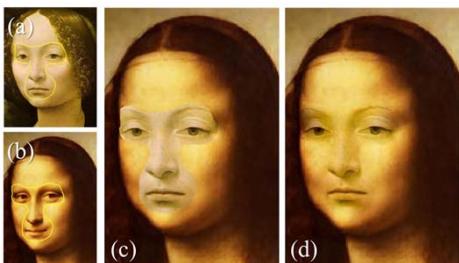
Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

- 画像の気に入らないところを削除 & 穴埋め
- GIST特徴量による高速な検索
- ポアソン画像処理による合成



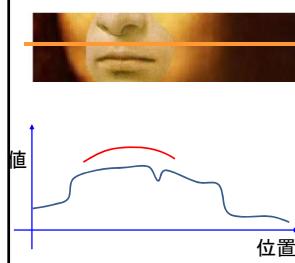
Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

- 2枚の画像の色合いの違いを埋める
 - ポアソン画像処理 [Perez et al. 2003]



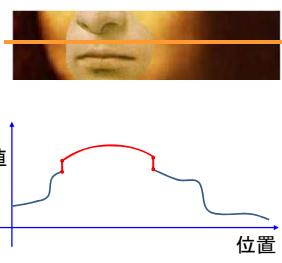
Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

- 2枚の画像の色合いの違いを埋める
 - ポアソン画像処理、勾配画像処理



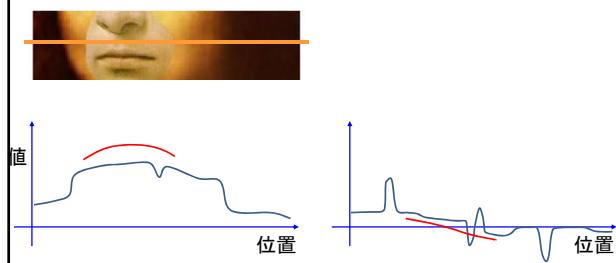
Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

- 2枚の画像の色合いの違いを埋める
 - ポアソン画像処理、勾配画像処理



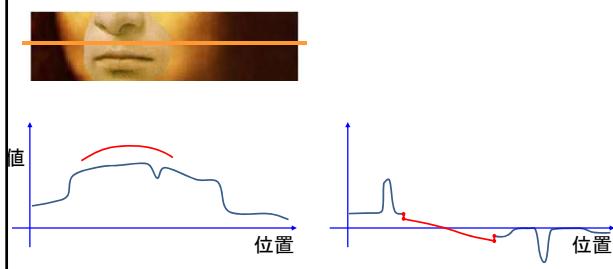
Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

- 2枚の画像の色合いの違いを埋める
 - ポアソン画像処理、勾配画像処理



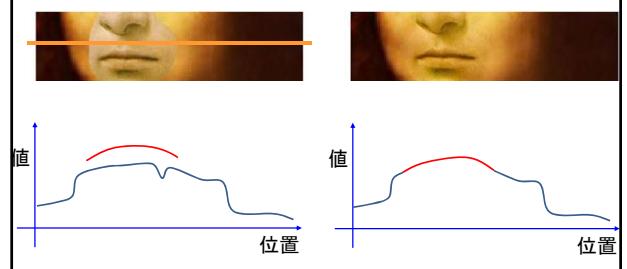
Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

- 2枚の画像の色合いの違いを埋める
 - ポアソン画像処理、勾配画像処理



Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

- 2枚の画像の色合いの違いを埋める
 - ポアソン画像処理、勾配画像処理



Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

「同じ方法を1万枚の写真で実験していた頃は全然ダメだったけど、データベースに200万枚の写真を使うと上手く動きだした」



Scene Completion Using Millions of Photographs
James Hays, Alexei A. Efros, **SIGGRAPH 2007**

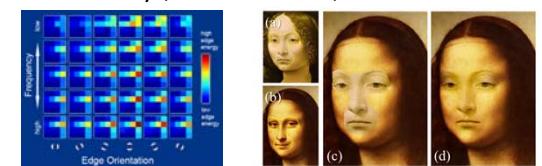


Photo Clip Art

Jean-Francois Lalonde, et al., **SIGGRAPH 2007**

- 別な写真のオブジェクトを cut&pasteしてシーンを合成する
- 画像の合成に使えそうな物体を検索
 - 照明条件やサイズを推定



Photo Clip Art

Jean-Francois Lalonde, et al., **SIGGRAPH 2007**

- 適当にやると結構難しい
 - 特に照明条件が合わない



Photo Clip Art
Jean-Francois Lalonde, et al., *SIGGRAPH 2007*

- LabelMe: ユーザによるアノテーション
 - 人や車等、タグ付けされ、領域も分かっている

Sketch2Photo: Internet Image Montage
Tao Chen et al., *SIGGRAPH Asia 2009*

- キャンバスにスケッチとテキストを入れる
 - 背景画像の検索
 - 形状を考慮した物体検索
- 自然な合成を生成

Sketch2Photo: Internet Image Montage
Tao Chen et al., *SIGGRAPH Asia 2009*

- ~~ヒツジを上手く切り取れるか否か → 難！~~
- ポアソン画像合成で上手く見える
- 前景と背景の「ペア」の検索に力を注ぐ

Data-Driven Object Manipulation in Images
Goldberg et al., *Eurographics 2012*

- 入力はSketch2Photoを変わらない
 - 形のスケッチ
 - 「Race Horse」という言葉を与える
 - ジョッキーの姿勢を曲げて修正する

ShadowDraw: Real-Time User Guidance for Freehand Drawing
Yong Jae Lee, Larry Zitnick, Michael Cohen, *SIGGRAPH 2011*

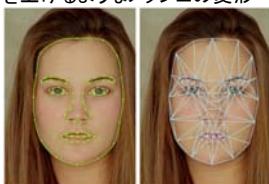
- 大量の画像を使ったお絵描き支援(0:20)
 - スケッチを描くと参考画像が浮き上がってくる
- スケッチに似たエッジ画像の検索
 - パッチに切ってにBiCE特徴(ECCV 2010)を記述
 - 似たパッチをたくさん持つものを候補

Data-driven enhancement of facial attractiveness
Tommer Leyvand et al., *SIGGRAPH 2008*

- 顔写真を入れると自動的に整形する

Data-driven enhancement of facial attractiveness
Tommer Leyvand et al., *SIGGRAPH 2008*

- 顔写真を入れると自動的に整形する
 - データセット: 顔の美しさを7段階で評価したもの
 - 8つの顔パーツの位置を自動検出
 - 特徴ベクトルは特徴点間のエッジ距離(234次元)
 - SVRによる顔空間の学習
 - スコアを上げるようなメッシュの変形



Data-driven enhancement of facial attractiveness
Tommer Leyvand et al., *SIGGRAPH 2008*

- 顔写真を入れると自動的に整形する
 - データセット: 顔の美しさを7段階で評価したもの
 - 8つの顔パーツの位置を自動検出
 - 特徴ベクトルは特徴点間のエッジ距離(234次元)
 - SVRによる顔空間の学習
 - スコアを上げるようなメッシュの変形



Exploring Photobios
Ira Kemelmacher-Shlizerman et al., *SIGGRAPH 2011*

- 最初と最後の顔画像を入れると(1:10)
その間をつなぐようなアニメーションを生成



- 顔の3次元姿勢推定
- 3つの特徴による類似度検索



画像ベースのCG流体表現

2012年7月21日

岡部 誠

電気通信大学 / JST さきがけ

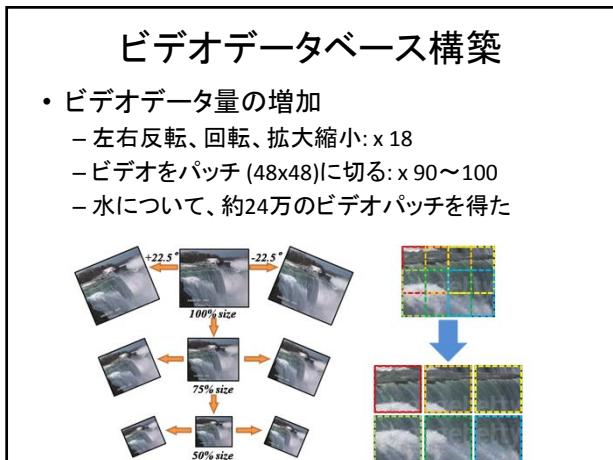
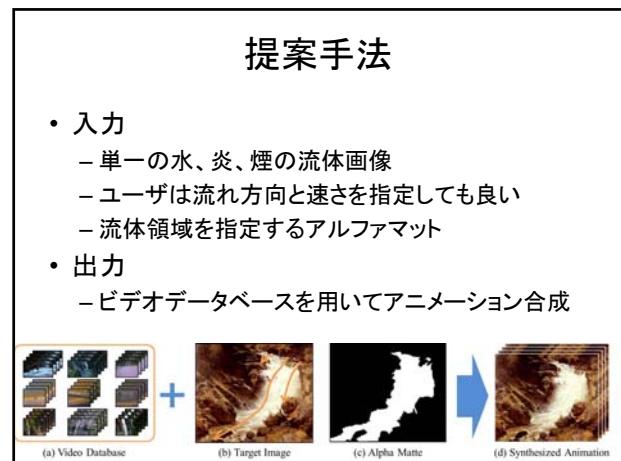
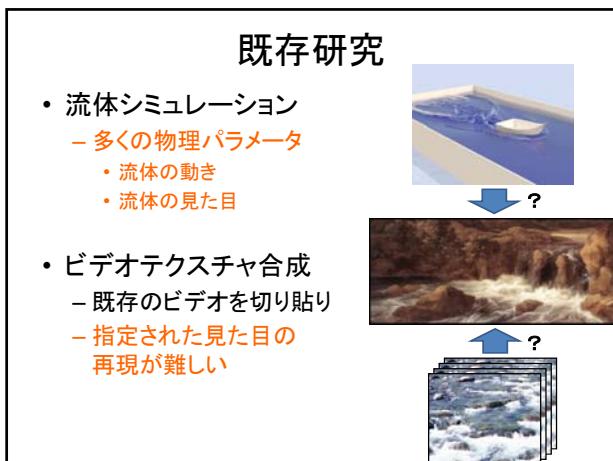
お話しの流れ

- 写真中の流体を動かす
- 動画中の流体を抽出する

モチベーション

- 流体画像を元にアニメーションを作りたい
 - 画像の質感を保ったまま
 - 難しい問題
- デザイナは1枚の流体の絵は描ける
 - 独特の面白いスタイル
 - 何枚も描くのは困難 or 時間が掛かる

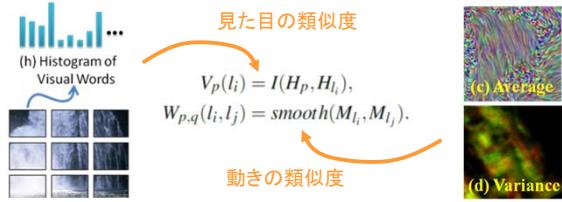




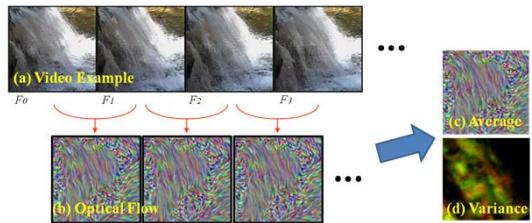
最適なビデオパッチの割り当て

- 動きの滑らかさを考慮した最適化
 - マルコフ・ランダム場を使用する

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

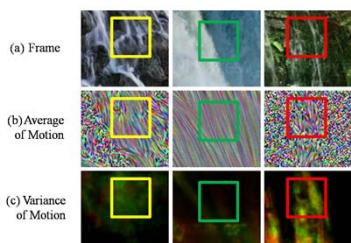


静止画を分割して割り当て



最適なビデオパッチの割り当て

- 隣接するパッチ間の動きの滑らかさ
 - 流れの速度 : 平均
 - 流れの激しさ : 分散



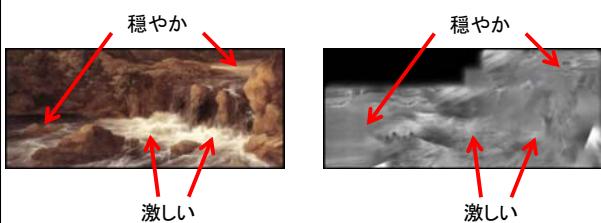
アニメーションの生成

- 入力画像
- 割当てられたビデオパッチ



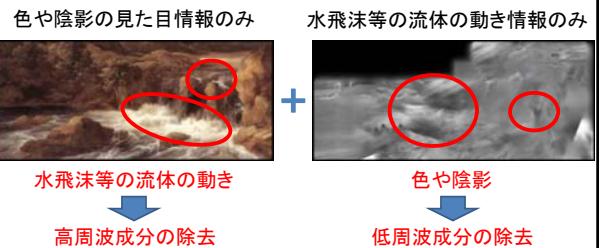
アニメーションの生成

- 入力画像
- 割当てられたビデオパッチ



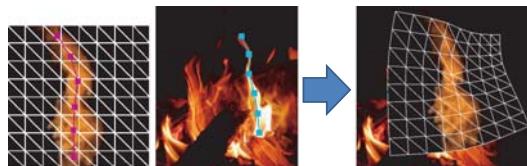
アニメーションの生成

- 入力画像と割当てられたビデオパッチを統合



まとめ

- 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の提案
 - ユーザの手作業の大幅軽減
 - ビデオ素材を崩すことなく利用できる
- ビデオのより直接的な利用法を検討中



お話を流れ

- 写真中の流体を動かす
- 動画中の流体を抽出する

背景

- 爆発、炎、煙の“使える”動画素材を増やしたい
 - 特別なスタジオで撮影する必要がある
 - 実世界の流体動画もたくさんある(ウェブ上)
- 背景を完全に消すことが困難 → 消そう！



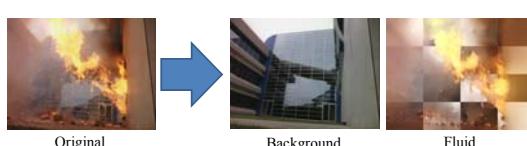
背景

- 野外で撮影した、流体動画素材は使えない
- 背景を完全に消すことが困難



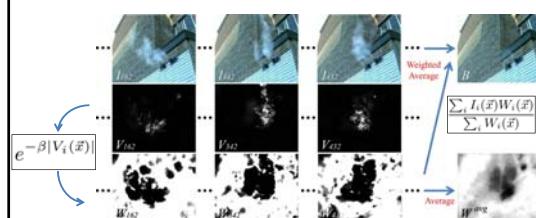
提案手法

- 入力動画を2つに分離
 - まずは背景
 - 次に半透明流体(元フレームと背景間の差分)



背景の推定

- 動きの少ないピクセルは背景
 - 動きはオプティカルフローで計算(速度場 V)
 - V の逆数としての重み W

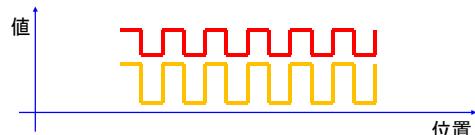
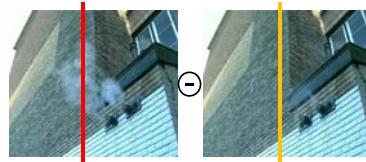


流体の抽出

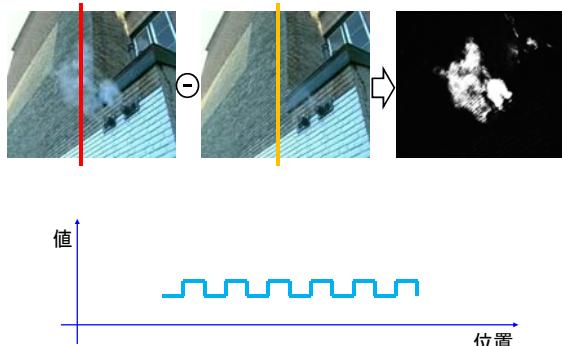
- Idea : 背景差分 (After Effects「異なるマット」)
- Pros : ただの引き算なので処理が高速
- Cons : 背景のレンガの模様の影響が出る



流体の抽出

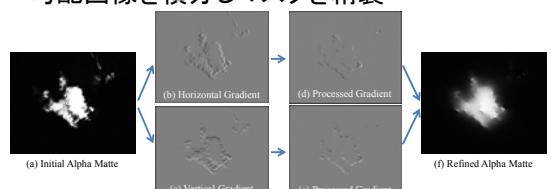


流体の抽出



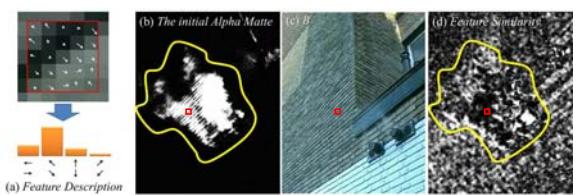
勾配画像処理

- レンガ模様の影響を消す
- 縦横方向の勾配 (=エッジ) 画像を計算
- レンガ模様に関する勾配を弱める
- 勾配画像を積分しマスクを精製



勾配画像処理

- 5×5 パッチ内の勾配 \rightarrow 4次元のヒストグラム
- テクスチャ模様の類似度マップ
 - 黒い値が高い類似性を表わし
そこでは勾配がより除去される。



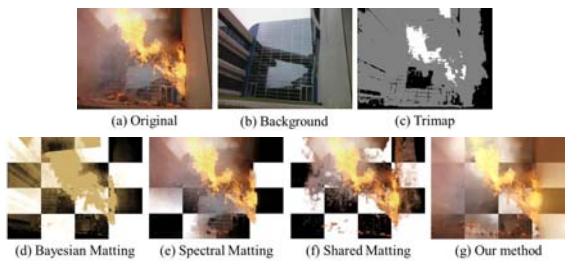
結果と考察

- A notebook PC
 - An Intel(R) Core(TM) i7 1.87 GHz CPU
 - An NVIDIA Quadro FX 3800M GPU
- ボトルネックは積分計算 (ポアソン方程式)

Fluid Video	Frames	Resolution	Time
Smoke1	883	352×288	2.92
Explosion1	209	480×360	6.31
Explosion2	947	640×360	5.0
Explosion3	609	640×360	6.0

結果と考察

- 既存手法との比較
 - 霧の除去法やtrimapは広い半透明部分が苦手



結果と考察

- Pros: 半透明部分が上手く取れている
- Cons: 全体的にボケてしまう
 - コントラストの復元等の後処理?...難しい
 - 勾配を弱める処理の改善

