

情報デザイン専攻

画像情報処理論及び演習I

-画像合成-

Texture Synthesis/Inpainting

第9回講義
水曜日1限
教室

吉澤 信
shin@riken.jp, 非常勤講師
大妻女子大学 社会情報学部

独立行政法人
理化学研究所

Shin Yoshizawa: shin@riken.jp

レポートについて

第一回レポートの採点結果を返します！
取りに来てください。

- ✓ 採点に納得がいけない人は講義終了後に交渉可。
- ✓ レポートは7末(予定)まで受け付けますが、 \times 切後は点数に0.8倍。
- ✓ 名前と学籍番号かいてくださいねー(^.^)
- ✓ ファイル名・フォルダー名は半角英数でお願いします日本語(全角)はダメ。
- ✓ 良く書いて(調べて、考えて)いたり、coolな解答にはボーナス点を加算しました。

Shin Yoshizawa: shin@riken.jp

今日の授業内容

www.riken.jp/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Lec06.pdf

- ① 前回の復習とレポートについて。
- ② Texture Synthesis・Inpainting
- ③ 演習：前回(領域抽出)の続き。
+レポート第2・3回の質問。

第2回のレポートは今日 \times 切なので出してねーp(^)q

Shin Yoshizawa: shin@riken.jp

前回の復習：ラベリングとは？

✓ ラベリング(Labeling)：連結領域を抽出する事。
✓ 連結領域：同じ画素値の繋がった領域。

- 4連結：左右上下。
- 8連結：3x3の領域。

二値化 多値化 ラベリング

Shin Yoshizawa: shin@riken.jp

前回の復習：細線化(thinning)

✓ 領域抽出後(二値化)に領域を線状に簡略化する事、ただし通常は入力の二値画像と**同位相**の形状。

✓ 出来るだけ中心に細く、端点でない境界画素を削除していく。

分岐点 端点 通過点

細線化

文字認識等で非常に
よく用いられる！

Shin Yoshizawa: shin@riken.jp

第3回レポートの説明

www.riken.jp/brict/Yoshizawa/Lectures/index.html

✓ 第3回レポートの \times 切は**7月6日**です。

レポート作成の注意点・作成提出方法の説明資料

レポートの提出先

レポートの雛形
(これをWindowsのワードで編集してレポートを作成)

Shin Yoshizawa: shin@riken.jp

復習: 3次元形状を用いた画像合成

2D人顔・人体画像の3D形状モデルを用いたアニメーション・モーフィング:

Application: input → output

3D reconstruction → rendering

©V. Blanz et al., FG 2003

Shin Yoshizawa: shin@riken.jp

画像合成 (Image Synthesis)

✓ 複数(又は局所画像)画像から新しい画像を生成する事.

- 本講義では3D形状は使わない画像合成を扱う.
- Alpha-Blending.
- Dissolve.
- Image Morphing.
- Inpainting.
- Pixel Transfer.
- Image Analogy.
- etc.

次回以降:
Poisson Image Editing.

©CG-ARTS 協会

Shin Yoshizawa: shin@riken.jp

単純な合成

✓ 色の平均: $I^{new}(x) = (I_1(x) + I_2(x)) / 2$

✓ Alpha-Blending: 透明度を画素の位置により線形補間.

©CG-ARTS 協会

Shin Yoshizawa: shin@riken.jp

時間変化の合成: ティームルフ (Dissolve)

✓ 透明度(Alpha)を時間的に変化(線形補間0.0~1.0):

©CG-ARTS 協会

$$A^{new}(x) = tA_1(x) + (1-t)A_2(x), \quad 0 \leq t \leq 1.$$

Shin Yoshizawa: shin@riken.jp

モーフィング (Morphing)

✓ 物体(注: 画像ではない)の平均・補間.

©D. Hoem, Univ. Illinois

©Grey Eden, CML, 2008

Shin Yoshizawa: shin@riken.jp

モーフィング (Morphing) 2

✓ 単純な画素値のディゾルブの結果では物体の平均・補間にはならない!

単純Alpha-Blending

好ましいモーフィング

©D. Hoem, Univ. Illinois

単純Alpha-Blending

Shin Yoshizawa: shin@riken.jp

モーフィング(Morphing)2

1. 対応点の作成: 特徴点作成+対応付け.
2. 局所変形(Local Warping): **位置合わせ**.
3. クロスディゾルブ(Cross-Dissolve).

対応点作成

変形 ↓ クロスディゾルブ ↓ 変形

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

Shin Yoshizawa: shin@riken.jp

特徴点作成

- ✓ マニュアル、特徴抽出、メッシュ生成(Voronoi図/Delaunay三角形分割)等: Delaunay三角形分割はVoronoi図の双対.

©Stanford Univ.

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

Shin Yoshizawa: shin@riken.jp

局所変形(Local Warping)1

- ✓ 変形・補間法を用いる: **アフィン変換**、スプライン補間、重心座標、一般化重心座標、RBF (Radial Basis Function)等.

回転、拡大・縮小、平行移動、シフト、反転

Mapping $f: Q_i \rightarrow V_i$

$$X = w_1V_1 + w_2V_2 + w_3V_3$$

$w_1 = \frac{\text{area}(Q_2, Q_3, P)}{\text{area}(Q_1, Q_2, Q_3)}$, $w_2 = \frac{\text{area}(Q_1, Q_3, P)}{\text{area}(Q_1, Q_2, Q_3)}$, $w_3 = \frac{\text{area}(Q_1, Q_2, P)}{\text{area}(Q_1, Q_2, Q_3)}$

$w_1 + w_2 + w_3 = 1$

©T. Igarashi et al., SIGGRAPH 2005

©D. Hovem, Univ. Illinois

©T. Ja et al., SIGGRAPH 2005

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

Shin Yoshizawa: shin@riken.jp

局所変形(Local Warping)2

- ✓ 変形・補間法を用いる: **アフィン変換**、スプライン補間、重心座標、一般化重心座標、RBF (Radial Basis Function)等.

©K. Horemann

©T. Ja et al., SIGGRAPH 2005

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

©D. Hovem, Univ. Illinois

Shin Yoshizawa: shin@riken.jp

局所変形(Local Warping)3

- ✓ 変形・補間法を用いる: **アフィン変換**、スプライン補間、重心座標、一般化重心座標、RBF (Radial Basis Function)等.

線形変換 $y = Ax + 1$

非変換した位置の座標を揃った座標系を持って決定(補間)

変換 $y = A'(y - 1)$

©T. Kanai

©Y. Choi and S. Lee, Graphical Models 2000

©Wikipedia

Control Points

Control Polygon

©W. C. Liu et al., SIGGRAPH

Shin Yoshizawa: shin@riken.jp

局所変形(Local Warping)4

- ✓ 変形・補間法を用いる: **アフィン変換**、スプライン補間、重心座標、一般化重心座標、RBF (Radial Basis Function)等.

©K. Durrmann

©N. Ayad, D. Bickel, CGF 1995

Shin Yoshizawa: shin@riken.jp

クロスディゾルブ(Cross-Dissolve) 1

✓ 複数画像に対する変形結果のディゾルブを計算する事.

CG, Woberg, CG'96

Shin Yoshizawa: shin@riken.jp

クロスディゾルブ(Cross-Dissolve) 2

CG, Woberg, CG'96

Shin Yoshizawa: shin@riken.jp

モーフィング(Morphing) 3D

✓ 3D形状のモーフィングもCGではある.

Shin Yoshizawa: shin@riken.jp

フィルタ等の複数の処理を組み合わせる事も

✓ 例えばエンボス画像生成:

- エンボス (Emboss): 板金や紙などに文字や絵柄などを浮き彫りにする加工.

CG, Woberg, CG'96

Shin Yoshizawa: shin@riken.jp

マスク(領域抽出)画像と画像合成

✓ マスク(領域抽出)画像を用いて対象領域だけ合成する事が主流.

- マスクの境界からの距離等を用いる方法もある.
- 領域抽出の応用.

CG, Woberg, CG'96

Shin Yoshizawa: shin@riken.jp

マスク(領域抽出)画像生成 1

✓ マスク画像は自動領域抽出、クロマキー、マニュアル、半自動(Interactive)等で生成.

- 復習: 自動領域抽出: 大津の二値化法, Snake (Active Contour), Graph Cuts, Mean Shift, Water Shed (Region Growing)等.

CG, Woberg, CG'96

Shin Yoshizawa: shin@riken.jp

マスク(領域抽出)画像生成2

- ✓ マスク画像は自動領域抽出、クロマキー、マニュアル、半自動(Interactive)等で生成。
 - クロマキー(Chromakey): 特定の色からマスクを生成する事。
 - テレビ、映画等の背景合成。
 - 光学式、回路式、デジタル式。

©CG-ARTS 監修
©Shinaka.com/ony.com
©Chalipicoda
©Pophandling.net/jp

Shin Yoshizawa: shin@riken.jp

マスク(領域抽出)画像生成3

- ✓ マスク画像は自動領域抽出、クロマキー、マニュアル、半自動(Interactive)等で生成。
 - 半自動: 最小限のユーザーインタラクションでマスクを生成。
 - 基本的アルゴリズムは全自動の領域抽出法だが、抽出法のパラメータや拘束条件等をユーザーが与える方法。

©T. Iri and H. Yokota PG 10.
©C. Rother et al. SIGGRAPH, 2004.

Shin Yoshizawa: shin@riken.jp

Inpainting・Hole Filling

- ✓ マスク内部の画像を自動生成する事。
 - 周りの画素値を使った補間。
 - Texture合成: Pixel/Texture Transfer, Image Completion.

©H. Yamuchi et al., CGJ 2003.
©D. Hoiem, Univ. Illinois.

Shin Yoshizawa: shin@riken.jp

補間によるInpainting

- ✓ 補間によるInpaintingは、(ほとんどの補間法が滑らかな関数で値を繋ぐため)細かい領域に有効だが、大きなマスクでは不自然な結果。
- ✓ 通常は補間+Texture合成。

©H. Yamuchi et al., CGJ 2003.

Shin Yoshizawa: shin@riken.jp

Texture合成

- ✓ 与えられた画像を敷き詰める事:
 - 境界を出来るだけ意識させない。
 - Textureの繋がり(パターン)を保持。

©D. Hoiem, Univ. Illinois.

Shin Yoshizawa: shin@riken.jp

Pixel TransferによるInpainting

- ✓ 画像から似ている画素・Textureを持つてくる。
 - 局所Windowで類似パターンを検索: Windowサイズに依存。
 - 低周波画像は補間で生成しておくことと影等の効果を反映出来る。
 - 穴(マスク)を埋める順番が重要!

©D. Hoiem, Univ. Illinois.
©H. Yamuchi et al., CGJ 2003.

Shin Yoshizawa: shin@riken.jp

Pixel TransferによるInpainting2

©D. Hoiem, Univ. Illinois.

Increasing window size

Input texture

Block

Random placement of blocks

Neighboring blocks constrained by overlap

Minimal error boundary cut

overlapping blocks

vertical boundary

overlap error

min. error boundary

Shin Yoshizawa: shin@riken.jp

Pixel TransferによるInpainting3

©D. Hoiem, Univ. Illinois.

Input Image

類似検索

Shin Yoshizawa: shin@riken.jp

Pixel TransferによるInpainting4

✓ 補外(Extrapolation)も同じ原理で可能.

©D. Hoiem, Univ. Illinois.

Shin Yoshizawa: shin@riken.jp

Pixel TransferによるInpainting5

©L. Dorri et al., SIGGRAPH 2003.

Shin Yoshizawa: shin@riken.jp

Shape Completion

✓ CG: 3D形状への応用.

©V. Kravny and A. Sheffer, SIG'05

©A. Sheef et al., SIGGRAPH'04

Shin Yoshizawa: shin@riken.jp

Image Analogy / Example-based

✓ 例題や類推による画像編集・合成:原理は同じ類似検索.

©A. Hermans et al., SIGGRAPH 2003.

source texture

target image

correspondence maps

Shin Yoshizawa: shin@riken.jp

Image Analogy2

✓ 例題や類推による画像編集・合成: 原理は同じ類似検索

©A. Hermans et al., SIGGRAPH 2011.
©D. Hoiem, Univ. Illinois.

Shin Yoshizawa: shin@riken.jp

Image Analogy3

✓ 様々なフィルタ処理が可能!

©A. Hermans et al., SIGGRAPH 2011.

Shin Yoshizawa: shin@riken.jp

Image Analogy4

✓ 様々なフィルタ処理が可能!

©A. Hermans et al., SIGGRAPH 2011.

Shin Yoshizawa: shin@riken.jp

Image Analogy5

✓ 様々なフィルタ処理が可能!

©A. Hermans et al., SIGGRAPH 2011.

Shin Yoshizawa: shin@riken.jp

Image Analogy6

©A. Hermans et al., SIGGRAPH 2011.

Shin Yoshizawa: shin@riken.jp

Image Analogy7

Example-based Painting:

データ入力
画像とその領域の分類

Userの入力
Painting

出力: 合成画像

©A. Hermans et al., SIGGRAPH 2011.

Shin Yoshizawa: shin@riken.jp

前回は言いましたが、レポートについて1

演習の時間に第一回レポートの採点結果を返します！

- ✓ 採点に納得がいかない人は講義終了後に交渉可.
- ✓ レポートは7末(予定)まで受け付けますが、×切後は点数に0.8倍.
- ✓ 名前と学籍番号かいてくださいね(^.^);
- ✓ ファイル名・フォルダー名は半角英数をお願いします日本語(全角)はダメ.
- ✓ 良く書けて(調べて、考えて)いたり、coolな解答にはボーナス点を加算しました.

Shin Yoshizawa: shin@riken.jp

前回は言いましたが、レポートについて2

前回取りに来ていない人は来てください.

- ✓ **レポートは何度出してもOK.** 解けなかった問題が分かったりプログラムが書けたら、その部分だけ追加で提出OK. ただし(何回目の提出を)明記する事 & ×切すぎたら0.8倍はかわらず.

今回(Q9,Q10出来なくて)点数低かった人もQ9&Q10やれば50*0.8点を加算しますよー.

- ✓ 誰もやってくれる人がいなかったで...°・(//)°、第一回レポートExtra問題の点数を100点に引き上げます!、プログラム得意な人は挑戦してね(´∀`)ノ! 第一回Extraだけ×切過ぎても0.8倍はなし!

Shin Yoshizawa: shin@riken.jp

演習: 二値化のプログラムを作ってみよう!

www.riken.jp/briect/Yoshizawa/Lectures/Ex04.zip
www.riken.jp/briect/Yoshizawa/Lectures/Lec04.pdf

前回の続き(二値化ラベリング、細線化):

1. Ex04内に用意されたプログラム群を動かしてみる.
2. 大津法の代わりに閾値をマニュアルで与えて二値化した結果を用いてRun_ex04_1.sh(又はRun_ex04_2.sh)と同様の結果を出力するプログラムの作成.
3. クラス間分散を計算して大津の方法をプログラムしてみる([otsu_Chi.h](#)を完成させる).

Shin Yoshizawa: shin@riken.jp

来週の予定

www.riken.jp/briect/Yoshizawa/Lectures/index.html



©Perez et al. SIGGRAPH 2003.

- ① 画像合成・Inpaintingその2
- ② 演習: [画像合成](#).