

情報デザイン専攻

画像情報処理論及び演習I

-画像合成・類推-

Image Analogy

第11回講義
水曜日1限
教室6218情報処理実習室

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

今日の授業内容

www.riken.jp/briect/Yoshizawa/Lectures/index.html
www.riken.jp/briect/Yoshizawa/Lectures/Lec11.pdf
 レポートの採点結果を取りに来てください!

- ① 画像類推のアルゴリズム・演習.
- ② NumberEditorの演習.
- ③ レポート3.

Shin Yoshizawa: shin@riken.jp

復習: 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!

類似検索

©D. Hoiem, Univ. Illinois.
©H. Yamuchi et al., CGJ 2005.

Shin Yoshizawa: shin@riken.jp

復習: 画像類推: Image Analogy

✓ AがA'になるならば、BはB'になる→B'の画素値(色、明るさ)を類似検索で計算: 局所テクスチャーを特徴ベクトルとし類似検索する事で、類推したピクセルを転写.

©A. Hartmann et al., SIGGRAPH 2001.

入力

出力

Shin Yoshizawa: shin@riken.jp

Image Analogyアルゴリズム

パラメータ(Windowサイズ): $r \geq 2$.

©A. Hartmann et al., SIGGRAPH 2001.

```

CREATEIMAGEANALOGY(A, A', B)
1 Compute Gaussian pyramids for (A, A', B)
2 Compute features for (A, A', B)
3 Initialize search structures
4 for  $\ell = 0$  to  $L$ 
5   for each pixel  $q \in B'_\ell$ , in scan-line order
6      $p \leftarrow \text{BESTMATCH}(A, A', B, B', s, \ell, q)$ 
7      $B'_\ell(q) \leftarrow A'_\ell(p)$ 
8      $s_\ell(q) \leftarrow p$ 
9 return  $B'_L$ 
  
```

Shin Yoshizawa: shin@riken.jp

Image Analogyアルゴリズム2

✓ 検索はANN (Approximate Nearest Neighbor)ライブラリを使う.

✓ ANNはエラー(誤差)を許して高速にn次元空間の近傍をサーチ.

パラメータ(ANNError): $E \geq 1.0$.

©A. Hartmann et al., SIGGRAPH 2001.

```

CREATEIMAGEANALOGY(A, A', B)
1 Compute Gaussian pyramids for (A, A', B)
2 Compute features for (A, A', B)
3 Initialize search structures
4 for  $\ell = 0$  to  $L$ 
5   for each pixel  $q \in B'_\ell$ , in scan-line order
6      $p \leftarrow \text{BESTMATCH}(A, A', B, B', s, \ell, q)$ 
7      $B'_\ell(q) \leftarrow A'_\ell(p)$ 
8      $s_\ell(q) \leftarrow p$ 
9 return  $B'_L$ 
  
```

Shin Yoshizawa: shin@riken.jp

Image Analogyアルゴリズム3

✓ Best Approximate MatchはWindowの半径2のとき5次元ベクトルのガウス相関。
 - ガウス相関: 中心の画素からガウス関数で重みを付けて対応する画素を要素とするベクトルの距離。

✓ Best Coherence MatchはTextureの整合性を加味して既に合成された画素の対応する画素でサーチ。

- Textureの整合性を重視する場合はパラメータ k を大きくする。
 - 大きくすぎるとAとA'だけしか結果に反映されないで注意。

パラメータ(Texture度): $k \geq 0$.

```

BESTMATCH(A, A', B, B', s, t, q)
1 papp ← BESTAPPROXMATCH(A, A', B, B', t, q)
2 pcoh ← BESTCOHERENCEMATCH(A, A', B, B', s, t, q)
3 dapp ← ||Fi(papp) - Fi(q)||2
4 dcoh ← ||Fi(pcoh) - Fi(q)||2
5 if dcoh ≤ dapp(1 + 2-k)
6   then return pcoh
7   else return papp
  
```

CA: Hartmann et al., SIGGRAPH 2001

Shin Yoshizawa: shin@riken.jp

演習: Image Analogyを使ってみよう!

www.riken.jp/brict/Yoshizawa/Lectures/Ex05.zip
www.riken.jp/brict/Yoshizawa/Lectures/Lec11.pdf

Image Analogyでフィルタリング:

- Ex05内に用意されたプログラム群を動かしてみる。
- Ex05内の画像を用いてImage Analogyによる色々なフィルタリング処理を試みる。
- 新しいフィルタリングを考えてみよう!

この演習は第3回レポートの内容なので頑張ってくださいねーp(^)q

Shin Yoshizawa: shin@riken.jp

演習: ANNのコンパイル

www.riken.jp/brict/Yoshizawa/Lectures/Ex05.zip
www.riken.jp/brict/Yoshizawa/Lectures/Lec11.pdf

まずはじめに、ANNをコンパイルする。

- Ex05.zipを展開する。
- Ex05内にann_1.1.2.zipがあるのでEx05内で展開する
- 端末でEx05/ann_1.1.2に入る、もしもデスクトップに展開していたら、「cd ~/Desktop/Ex05/ann_1.1.2」。
- コンフィギュレーションを行う4.の後に端末で「sh Make-config」でエンターキー。
- コンパイルする5.の後に端末で「make linux-g++」と打ち込みエンターキーを押す。Ex05/ann_1.1.2/libの下にlibANN.aが出来れば成功。

Shin Yoshizawa: shin@riken.jp

演習: Ex05内の説明

www.riken.jp/brict/Yoshizawa/Lectures/Ex05.zip
www.riken.jp/brict/Yoshizawa/Lectures/Lec11.pdf

Ex05内の説明: コンパイルは端末で「make」Makefile

✓ ImageAnalogyClass.h: Image Analogyの本体。

- ColorImage.h: カラー画像クラス。
- GaussianPyramid.h: ガウスピラミッドクラス。

✓ Image Analogyとは関係ないファイル:

- Image Analogyの入力画像を生成するフィルタで使うヘッダーファイル: Gauss.h: ガウス平滑化用、fastgb.h & gaussfgt1D.h: 高速エッジ保存フィルタ用。
- 前回までに使ったファイル: SimpleImage.h(画像クラス)、otsu.h(大津の二値化)、ppmio.h(カラー画像入出力)、thinning.h(細線化)。

Shin Yoshizawa: shin@riken.jp

演習: Image Analogyとは関係ないファイル

✓ まずは、Image Analogyとは直接関係ないプログラムから。ただし、これらのプログラムを使えばImage Analogyに入力させる画像を簡単に作成可能: [Run_Smoothing.sh](#), [Run_EdgePreserving.sh](#), [Run_EdgeThinning.sh](#)。

✓ EdgeThinning.cxx: エッジ強度画像(勾配強度=Gradientベクトルの大きさ)とエッジの細線化画像を出力するプログラム: 引数3:
 - ./EdgeThinning 入力.ppm 出力エッジ細線化.ppm 出力強度画像.ppm

エッジ細線化画像 入力 エッジ強度画像

Shin Yoshizawa: shin@riken.jp

演習: Image Analogyとは関係ないファイル

✓ Smoothing.cxx: ガウス平滑化を実行するプログラム: 引数3:
 - ./Smoothing 入力.ppm 出力.ppm 平滑化度合(double)
 - 平滑化度合のパラメータは0より大きな実数2.0~20.0ぐらいが実用的。

✓ EdgePreservingFilter.cxx: エッジ保存平滑化を実行: 引数3
 - ./EdgePreservingFilter 入力.ppm 出力.ppm エッジの大きさ(double)
 - エッジの大きさパラメータは0より大きな実数0.5~2.0ぐらいが実用的。

EdgePreservingFilter, 1.0 入力 Smoothing, 5.0

Shin Yoshizawa: shin@riken.jp

演習: Texture Transfer

✓ Image Analogyを用いてTexture Transferを実行するプログラム:
引数 8

- ./TextureTransfer 入力画像A.ppm 入力画像A'.ppm 入力画像B.ppm 出力画像B'.ppm Texture度k(double>=0.0) ANN誤差(double>=1.0) Window半径(int>=2) Blending(1.0>=double>=0.0, 小→元画像強め)
- sh Run_TextureTransfer.shでも実行可能.

Shin Yoshizawa: shin@riken.jp

演習: Texture by Numbers

✓ Image Analogyを用いてTexture by Numbersを実行するプログラム:
引数 7

- ./TextureByNumbers 入力画像A.ppm 入力画像A'.ppm 入力画像B.ppm 出力画像B'.ppm Texture度k(double>=0.0) ANN誤差(double>=1.0) Window半径(int>=2)
- sh Run_TextureByNumbers.shでも実行可能! 実行結果.

Shin Yoshizawa: shin@riken.jp

演習: Artistic Filters

✓ Image Analogyを用いて様々なArtisticフィルタを実行: 引数 7

- ./ArtisticFilter 入力画像A.ppm 入力画像A'.ppm 入力画像B.ppm 出力画像B'.ppm Texture度k(double>=0.0) ANN誤差(double>=1.0) Window半径(int>=2)
- sh Run_ArtisticFilter.sh, sh Run_Etc1~3.sh, sh Run_Smoothing_Sharpring.shでも実行可能.

Shin Yoshizawa: shin@riken.jp

演習 11-1 (テクスチャー合成): シェルの説明

✓ 端末にて「sh シェルスクリプト名.sh」で実行、中にコンパイル+実行+表示のコマンドが書いてある.

- Run_TextureTransfer.sh: 5種類のテクスチャーをテクスチャー度を1,3,5,7,9の五種類で実行.

Shin Yoshizawa: shin@riken.jp

演習: Run_TextureByNumbers.sh

Shin Yoshizawa: shin@riken.jp

演習: Run_Smoothing_Sharpring.sh

入力

Shin Yoshizawa: shin@riken.jp

演習:Run_ArtisticFilter.sh

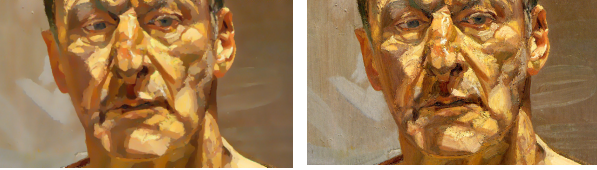

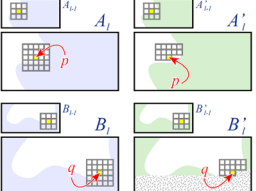




Diagram illustrating the Artistic Filter process with grid operations $A_{i,j}$, A'_i , $B_{i,j}$, and B'_i .

Shin Yoshizawa: shin@riken.jp

演習:Run_ArtisticFilter.sh

✓ テクスチャー度の違い:6種類実行.



Shin Yoshizawa: shin@riken.jp

演習:Run_ArtisticFilter.sh

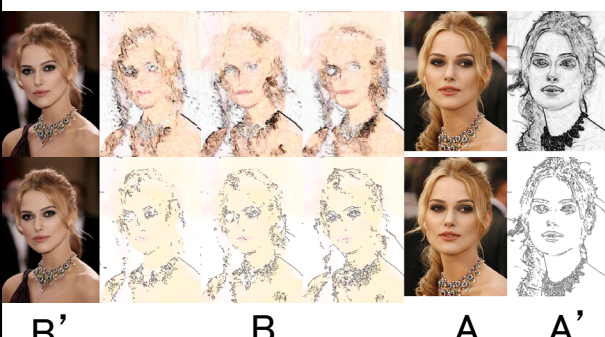
✓ Window半径の違い:2種類実行. **注意点:ANN誤差は全て1000.0で実行、1.0に近づければ綺麗な結果だが、計算時間が大:数十分~数十時間かかる可能性あり!**



Shin Yoshizawa: shin@riken.jp

演習:Run_Etc1.sh

✓ ArtisticFilterにEdgeThinningの出力(エッジ強度)を使った結果.



B' B A A'

Shin Yoshizawa: shin@riken.jp

演習(線画)11-2:Run_Etc2.sh



Shin Yoshizawa: shin@riken.jp

演習11-3:Run_Etc3.sh

✓ 油絵的フィルタ効果
入力→



A A'

Shin Yoshizawa: shin@riken.jp

演習 11-4: Run_Etc3.sh

✓ 水彩画的フィルタ効果
入力→

Shin Yoshizawa: shin@riken.jp

演習: シェルスクリプトを動かしてみよう!

端末にて「sh シェルスクリプト名.sh」

- ✓ Run_ArtisticFilter.sh
- ✓ Run_EdgePreserving.sh
- ✓ Run_EdgeThinning.sh
- ✓ Run_Smoothing.sh
- ✓ Run_Smoothing_Sharpning.sh
- ✓ Run_TextureTransfer.sh
- ✓ Run_TextureByNumbers.sh
- ✓ Run_Etc1.sh
- ✓ Run_Etc2.sh
- ✓ Run_Etc3.sh

✓ シェルスクリプトはテキストなので、emacsで中身を編集する事で自分の画像を実行可能。

Shin Yoshizawa: shin@riken.jp

演習: シェルスクリプトを変えてみよう!

Run_TextureByNumbers.shを使って以下の画像に対して処理してみよう!

A A' B

Shin Yoshizawa: shin@riken.jp

演習: 自分で新しいエフェクトを作ってみよう!

Run_ArtisticFilter.shを使って以下の画像の様に自分のオリジナルのエフェクトを処理してみよう!

注意点: A, A'の画像サイズは同じでないとダメ!

ヒント: 模様・エフェクトが付いた画像を平滑化するとよい?

Shin Yoshizawa: shin@riken.jp

演習 11-5: NumberEditor

Shin Yoshizawa: shin@riken.jp

重要: 演習: NumberEditor

www.riken.jp/briect/Yoshizawa/Lectures/Ex06.zip

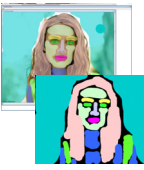
1. 端末にて「tosh」と打ち込んでエンターキー。
2. 端末にて「setenv LANG C」と打ち込んでエンターキー
3. 「sh Run_NumberEditor.sh」

Shin Yoshizawa: shin@riken.jp

演習: NumberEditor

- Image Analogy用TextureByNumbersのお絵かきGUI (Java).
- Ex06/NumberEditor/
- コンパイル: 端末で「javac NumberEditor.java」
- 実行: 端末で「java NumberEditor」
- sh Run_NumberEditor.shでもOK!

- 画像を読み込む: File->Load ppm Image. ppm画像
- お絵かき: 左クリック(ドラッグ)
 - 色を変える: 右下のSelectボタン.
 - ブラシのサイズを変える: 右のスクロールバー or マウスホイール.
 - 表示の透明度を変える: 下のスクロールバー.
- マスク画像(ppm)をセーブ: File->Save Number Image. 拡張子. ppm要る画像ファイル名.
- Ex06/TextureByNumbersの第2引数及び第3引数(もちろん違うお絵かきの結果)へ.



Shin Yoshizawa: shin@riken.jp

演習: NumberEditor & TextureByNumbers

- Image Analogy用TextureByNumbersのお絵かきGUI (Java).
- Ex06/NumberEditor/

- sh Run_NumberEditor.shで立ち上げてください.
- 画像を読み込む: File->Load ppm Image. Ex05/darkclouds.ppmを開いてみてください.
- お絵かき: 左ドラッグ: 木、岩、草原、空を違う色で塗ってみてください.
 - 色を変える: 右下のSelectボタン.
 - ブラシのサイズを変える: 右のスクロールバー or マウスホイール.
 - 表示の透明度を変える: 下のスクロールバー.
- セーブ: File->Save Number Image. A.ppmという名前前で保存してください.



Shin Yoshizawa: shin@riken.jp

演習: NumberEditor & TextureByNumbers


- 追加のお絵かき: 木、岩、草原、空で使った色とほぼ同じ色で書き足してみてください.
- マスク画像(ppm)をセーブ: File->Save Number Image. B.ppmという名前前で保存してください.
- A.ppmとB.ppmをEx05の下に移動(コピーでもカット&ペーストでもOK)してください.
- 端末を新たに立ち上げて、Ex05にcdで移動してください。もしもEx05をデスクトップで立ち上げていたら「cd ~/Desktop/Ex05」又はファイルブラウザのパスをコピーして端末に張り付けて「cd パス」でエンターキーを押す.



Shin Yoshizawa: shin@riken.jp

演習: NumberEditor & TextureByNumbers

- emacsでRun_TextureByNumbers.shを立ち上げて、以下の様に書き換えてください。./TextureByNumbersの後の
 - 第一引数oxbow-mask.ppm は A.ppm
 - 第二引数oxbow.ppm は darkclouds.ppm
 - 第三引数oxbow-newmask.ppm は B.ppm
 その後のppmファイル名も上のルールで変更してください.
- Run_TextureByNumbers.shをセーブ(上書き保存)してください.
- 端末にて「sh Run_TextureByNumbers.sh」で実行してみてください.



こんな感じなのが
出れば正解⇒

Shin Yoshizawa: shin@riken.jp

来週の手配

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



©Perez et al. SIGGRAPH 2003.

- 画像合成・Inpaintingその3
- 演習: 画像類推・合成.