

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-
Etcetera

第13回講義
水曜日 1限
教室6218

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/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Lec13.pdf
補講日: 7/27(金)5限: 6218室

- ① 画像合成・類推 Etcetera.
- ② 第3回レポートの説明.
- ③ 演習: 前回に引き続きレポート第3回の内容.


第3回レポートは金曜日7/20が切なので
みなさん頑張ってくださいねーp(^)q

Shin Yoshizawa: shin@riken.jp

Content-Aware Resizing/Retargeting

✓ 画像中の特徴的/象徴的オブジェクトのサイズやAspect Ratioを変えずに画像のサイズやAspect Ratioを変える事.



Shin Yoshizawa: shin@riken.jp

復習: 勾配: Gradient

✓ 勾配(Gradient): スカラー場の各点で変化が最大の方向と変化率を大きさに持つベクトル場.

✓ 勾配作用素: $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$

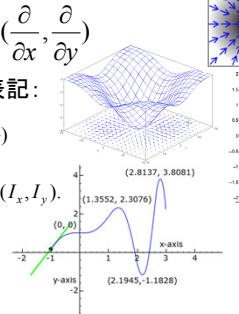
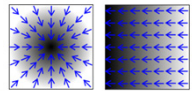
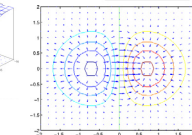
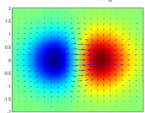
✓ 勾配ベクトルの表記:

$$\nabla I = \nabla I(x, y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$

$$= \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right) = (I_x, I_y)$$

✓ 勾配の大きさ:

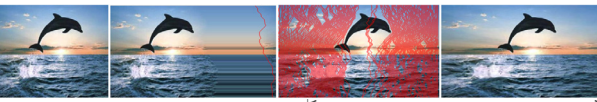
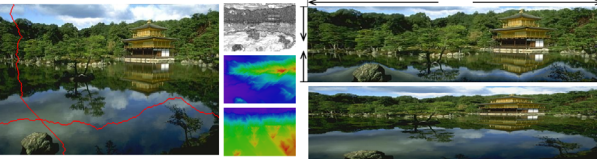
$$\|\nabla I\| = \sqrt{I_x^2 + I_y^2}$$

Shin Yoshizawa: shin@riken.jp

Content-Aware Resizing/Retargeting2

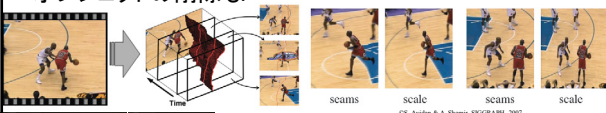
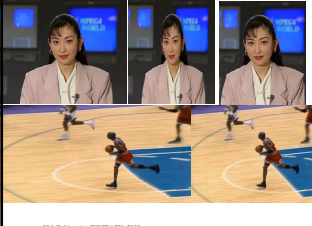

✓ **Seam Carving:** 勾配の強度エネルギーが最小になる曲線群を計算して複製・削除によりサイズ変更.

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right), \quad E(I) = \|\nabla I\|_{L_1} = \left| \frac{\partial I}{\partial x} \right| + \left| \frac{\partial I}{\partial y} \right|$$



Shin Yoshizawa: shin@riken.jp

Content-Aware Resizing3

✓ **Seam Carving:** ビデオへも拡張 & マスクと組み合わせるとオブジェクトの削除も.

Shin Yoshizawa: shin@riken.jp

Content-Aware Resizing / Retargeting4

✓ 勾配強度やマスクと他の画像特徴を組み合わせて局所変形(Warping)をする方法もある。

Gradient Map × Saliency Map = Significance Map

original image using gradient map using significance map

CV: S. Wang et al., SIGGRAPH Asia 2005. CR: Gal et al., ECCV'06.

Shin Yoshizawa: shin@riken.jp

復習: 2-bands Blending

✓ 低周波は滑らかにAlphaを変化 + 高周波はAlpha定数。

低周波成分画像 高周波成分画像

Shin Yoshizawa: shin@riken.jp

イメージモザイク / パノラマ画像生成

- 特徴点・対応点抽出。
- 幾何学的変換の推定。
- 幾何変換&色補正。
- Blending.

上から見た図

CCG-ARTS15&2

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 & 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. Yamashita et al., CVJ 2003.

Shin Yoshizawa: shin@riken.jp

数百万のExampleがあると...

✓ データベースからの類似検索 + 画像合成.

©D. Hoiem, Univ. Illinois.



Shin Yoshizawa: shin@riken.jp

数百万のExampleがあると...

©D. Hoiem, Univ. Illinois

Shin Yoshizawa: shin@riken.jp

数百万のExampleがあると...

©D. Hoiem, Univ. Illinois

Shin Yoshizawa: shin@riken.jp

Image2GPS

✓ 画像データベースに位置情報が付いている(撮影場所)と画像検索→類似画像の多さを地図にマップ=GPS.

©Hayaj & Etoh, CVPR 2008

Shin Yoshizawa: shin@riken.jp

Image2Real

✓ 画像データベースからCGの類似画像検索+部分合成により写実的(Photorealistic)なCGの生成.

MIT-CSAIL-TR-2009-034

Shin Yoshizawa: shin@riken.jp

カモフラージュ画像合成

Camouflage Art:

Shin Yoshizawa: shin@riken.jp

復習: Image AnalogyのTexture Synthesis

Shin Yoshizawa: shin@riken.jp

カモフラージュ画像合成

quantization, segmentation, luminance assignment, texture synthesis, refinement, synthesis result

©H.-K. Chu, SIGGRAPH 2010

optimized segment map, synthesis result

Shin Yoshizawa: shin@riken.jp

復習:Poisson Image Editingの原理

✓ Idea: 良いBlendingはSource画像の勾配(Gradient=エッジ)を可能な限り保持する事が重要.

Source画像のGradient(マスク内)をTargetにコピーしマスク内だけTargetの境界条件で新しい輝度値 h を解く.

$$\Delta I = \text{div } \nabla g$$

Gradients, Target, Mask, $v = \nabla g$, $h = h(x, y)$

©J. Sun et al. SIGGRAPH 2004

Poisson方程式を解く!

Shin Yoshizawa: shin@riken.jp

Poisson Image Editingの問題点

Shin Yoshizawa: shin@riken.jp

Image Harmonization

✓ Multi-band Blendingの一種: ヒストグラムマッチングにて全体の色を合成し、ソース画像の高周波ノイズを多重解像度解析でBlending.

Source Image, Target Image, Noise Image, Smooth Histogram Matching, Noise Matching, Multi-scale Harmonization, Final Composite, Alpha, Seamless boundaries

©K. Sunkavalli et al. SIGGRAPH 10

(a) Source / Target (b) Seamless Cloning (c) Harmonization (d) Close-ups

Shin Yoshizawa: shin@riken.jp

Image Analogy+PIEでリアルな合成

✓ エッジ保存フィルタリング画像にPIEで合成後、Image Analogyで細部を復元: Aはエッジ保存フィルタリング画像、A'は元画像、BがPIEによる合成画像.

A, A', B, B'

?

Shin Yoshizawa: shin@riken.jp

Image Analogy+PIEでリアルな合成2

Target, Source, Edge保存平滑化, ポアソン画像合成, 画像類推, A:A' :: B:B', Result

Shin Yoshizawa: shin@riken.jp

PIE vs Image Analogy+PIE

Shin Yoshizawa: shin@riken.jp

成績の計算方法について

出席4割、レポート6割.

成績点数(基本は...もしかしたらゲタが...)=
 出席日数 × (20/7) + レポート3回分の合計点 × (1/5)

- ✓ 出席点は40点を休講を抜かした14で割って一回あたり(20/7)点です。補講日も加算します。遅刻した日は(20/7)の代わりに0.8 × (20/7)で計算してください。
- ✓ レポートの点は60点を300点で割ってレポートの1点あたり(1/5)点です。
- ✓ 合計点の小数点以下は切り上げします。合計点100点以上の方は100点です(^.^)
- ✓ S: 100-90点, A: 89-80点, B: 79-70点, C: 69-60点, D: 59-0点。
- ✓ 単位取得ボーダーの人や「あと数点で一つ上の評価なので何とか...」という人は補講日7/27(金):5限に相談可(基本レポートを提出・再提出)。

Shin Yoshizawa: shin@riken.jp

演習: レポート3の内容

www.riken.jp/briect/Yoshizawa/Lectures/index.html
 Ex05及びEx06で端末にて「make clean」 & 「make」.

- ✓ Q4: Image Analogyを用いてオリジナルのエフェクト。
- ✓ Q1: 油絵効果、水彩画効果、線画効果、テクスチャー合成の4種類。
- ✓ Q2: Texture by Numbers.
- ✓ Q3: Poisson Image Editing.

Shin Yoshizawa: shin@riken.jp

Q4: Image Analogyでオリジナル効果

- Ex05内のSmoothingでtexture5.ppmをフィルタリング: 端末で
`./Smoothing texture5.ppm texture5_blur.ppm 5.0`

- ArtisticFilterを使って
`./ArtisticFilter texture5_blur.ppm texture5.ppm lena.ppm lena_r3q4.ppm 2.0 1000.0 2`
`display lena_r3q4.ppm`

Shin Yoshizawa: shin@riken.jp

Q4: Image Analogyでオリジナル効果2

- 例えば「Zebra効果」、もしも以下の様なzebra.ppmを持っていたら、
`./Smoothing zebra.ppm zebra_blur.ppm 5.0`

- ArtisticFilterを使って
`./ArtisticFilter zebra_blur.ppm zebra.ppm lena.ppm lena_r3q4.ppm 2.0 1000.0 2`
`display lena_r3q4.ppm`

Shin Yoshizawa: shin@riken.jp

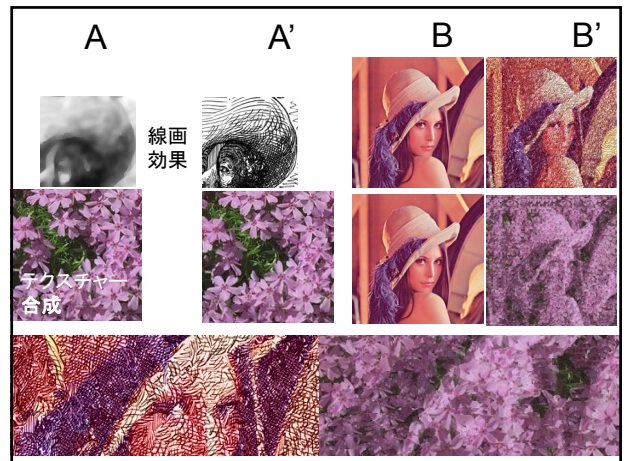
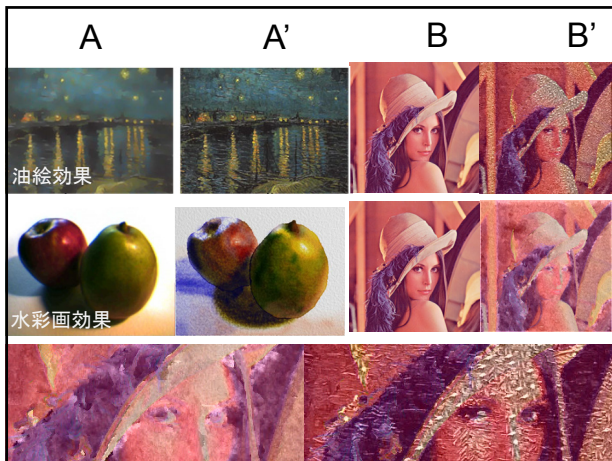
Q1: Image Analogy: Artistic Filters

- ✓ 端末でEx05に移動: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。

- 油絵効果をlena.ppmに適用してみる: 端末で、
`./ArtisticFilter rhone-src.ppm rhone.ppm lena.ppm test1.ppm 2.0 1000.0 2`

を打ち込んでエンターキーを押す。実行が終了したら、端末で、
`display test1.ppm &`
 同様に、

- 水彩画効果をlena.ppmに適用してみる:
`./ArtisticFilter watercolor-src.ppm watercolor.ppm lena.ppm test2.ppm 2.0 1000.0 2`
- 線画効果をlena.ppmに適用してみる:
`./ArtisticFilter squire-blur.ppm squire.ppm lena.ppm test3.ppm 2.0 1000.0 2`
- テクスチャー合成をlena.ppmに適用してみる:
`./TextureTransfer texture1.ppm texture1.ppm lena.ppm test3.ppm 2.0 1000.0 4 0.75`



Shin Yoshizawa: shin@riken.jp

Q2: 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](#)という名前で保存してください.

Shin Yoshizawa: shin@riken.jp

NumberEditor & TextureByNumbers2

- 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

NumberEditor & TextureByNumbers3

- emacsでRun_TextureByNumbers.shを立ち上げて、以下の様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.

こんな感じのが出れば正解⇒

Shin Yoshizawa: shin@riken.jp

Q3: Poisson Image editing & MaskEditor

- PIE用マスク作成GUI (Java): Ex06/MaskEditor/
 - sh Run_MaskEditor.shでMaskEditorを立ち上げてください.
 - Source画像を読み込む: File->Load Sourceで[Ex06/images/Keira02.ppm](#)を開いてください.
 - Target画像を読み込む: File->Load Targetで[Ex06/images/MonaLisa.ppm](#)を開いてください.
 - 左クリックでPolylineを生成して[Keiraの顔領域を作成](#)してみましょう!

Shin Yoshizawa: shin@riken.jp

MaskEditor2

✓ PIE用マスク作成GUI (Java): Ex06/MaskEditor/

- Source画像の大きさと位置を合わせる: [Keiraの顔とMonaLisaの顔の大きさと位置を合わせてみよう!](#)
 - 右クリックでMove Picを選べば平行移動可能.
 - 右クリックでAddを選べばPolyline作成モードに遷れる.
 - マウスの真ん中ホールで拡大縮小.
 - Polylineの頂点は左クリックで移動可能.
 - 下のスクロールバーで表示の透明度を変更可能.



Shin Yoshizawa: shin@riken.jp

MaskEditor3

- マスク画像(pgm)とTargetと同じ大きさのSource画像(ppm)の二つの画像をセーブ: File->Save Masks: [ソースとマスクをKeiraMonaという名前でセーブしてみよう!](#)

注: セーブするファイル名に拡張子はいらない: ファイル名.pgmとファイル

- 端末でPoissonImageEditorを以下の様に動かして合成してみよう!
 - 端末を立ち上げてEx06へ移動: 「cd ~/Desktop/Ex06」.
 - ./PoissonImageEditor ./MaskEditor/KeiraMona.ppm ./MaskEditor/KeiraMona.ppm ./images/MonaLisa.ppm KM_PIE.ppm 1.0 0.0
 - display KM_PIE.ppm & **訂正あり**

Source Mask Target 合成結果



Shin Yoshizawa: shin@riken.jp

演習

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

✓ レポートの内容が出来た人は以下の演習をやってみましょう!

✓ Image Analogy+PIEでリアルな合成.

Shin Yoshizawa: shin@riken.jp

演習: Image Analogy + PIEでリアルな合成

EdgePreservingFilterとArtisticFilterを使ってよりリアルな合成を試みよう!

- EdgePreservingFilterでMonaLisa.ppmをフィルタリング: 端末で
~/Desktop/Ex05/EdgePreservingFilter ./images/MonaLisa.ppm ML_EP.ppm 1.0
- display ML_EP.ppm &



Shin Yoshizawa: shin@riken.jp

演習: Image Analogy + PIEでリアルな合成

EdgePreservingFilterとArtisticFilterを使ってよりリアルな合成を試みよう!

- MonaLisa.ppmの代わりにML_EP.ppmで合成(ソース、マスクはそのまま):
./PoissonImageEditor ./MaskEditor/KeiraMona.ppm ./MaskEditor/KeiraMona.ppm ML_EP.ppm KM_PIE_EP.ppm 1.0 0.0
- display KM_PIE_EP.ppm &

Source Mask Target 合成結果

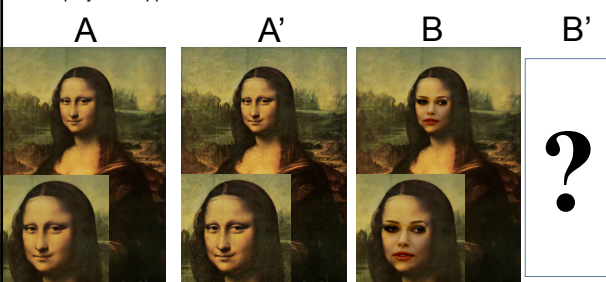


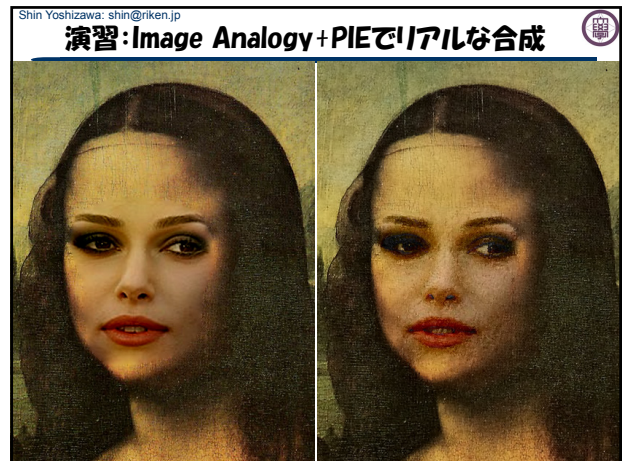
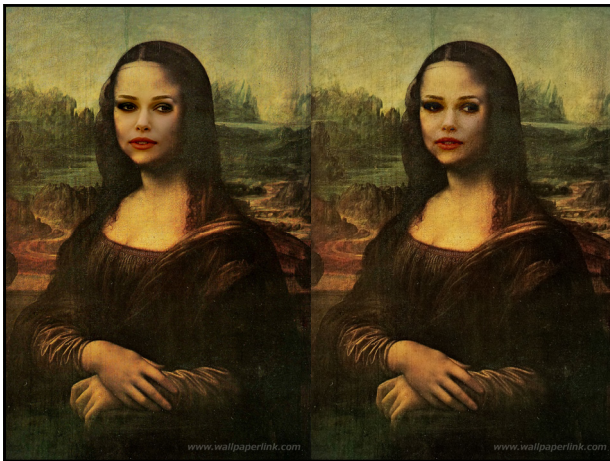
Shin Yoshizawa: shin@riken.jp

演習: Image Analogy + PIEでリアルな合成

- Artistic Filterで細部を復元: A: ML_EP.ppm A': ./images/MonaLisa.ppm B: KM_PIE_EP.ppm
~/Desktop/Ex05/ArtisticFilter ML_EP.ppm ./images/MonaLisa.ppm KM_PIE_EP.ppm result.ppm 2.0 1000.0 2
- display result.ppm &

A A' B B'





Shin Yoshizawa: shin@riken.jp

来週の予定

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

次回(7/25)は

- ① 成績のチェック.
- ② 後期の内容紹介.
- ③ 演習: [レポート1~3の内容](#).

補講日: 7/27(金)5限: 6218室

Shin Yoshizawa: shin@riken.jp

補足資料(Lec10.pdf): ANNのコンパイル

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

まずはじめに、ANNをコンパイルする。

1. Ex05.zipを展開する。
2. Ex05内にann_1.1.2.zipがあるので**Ex05内で展開する**。
3. 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」。
4. 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。
5. 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。
6. **Ex05で端末にて「make clean」&「make」**。

Shin Yoshizawa: shin@riken.jp

補足資料: Lec10.pdf

- ✓ 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 平滑化割合(double)
 - 平滑化割合のパラメータは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