情報デザイン専攻

画像情報処理論及び演習II

-周波数分解-
フーリエ変換、DCTと周波数操作

第3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

今日の授業内容

www.riken.jp/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Lec15.pdf

1. 高校数学の復習.
2. レポート04について (10/21〆切)
3. フーリエ変換と周波数操作.
4. 演習：Discrete Cosine Transform.
(DCT, 離散コサイン変換)によるフィルタ処理.

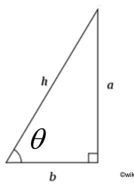
今日の演習はレポート2で出すので、
みなさん頑張ってくださいねーp(^^)q

復習：三角関数

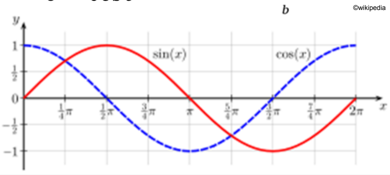
✓ 直角三角形の角度から辺の長さの比を与える.

$$\sin\theta = \frac{a}{h}, \qquad \cos\theta = \frac{b}{h}$$

$$\tan\theta = \frac{a}{b} = \frac{\sin\theta}{\cos\theta}$$



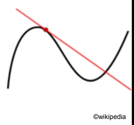
$$\cot\theta = \frac{b}{a} = \frac{1}{\tan\theta}$$
$$\csc\theta = \frac{h}{a} = \frac{1}{\sin\theta}$$
$$\sec\theta = \frac{h}{b} = \frac{1}{\cos\theta}$$



復習：微積分

✓ 微分は関数の微小変化率:
✓ 一階微分は速度・接線.
✓ 二階微分は加速度・曲率

$$\frac{df(x)}{dx} = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

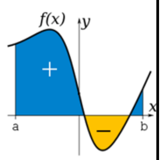


$$y = x^n \Rightarrow \frac{dy}{dx} = nx^{n-1}$$

✓ 積分は微分の逆で「和」の一般化:

$$\frac{d}{dx}\left(\int f(x)dx\right) = f(x)$$

$$y = x^n \Rightarrow \int_a^b x^n dx = \left[\frac{1}{n+1}x^{n+1}\right]_a^b = \frac{1}{n+1}(b^{n+1}-a^{n+1})$$



✓ 面積, 体積など

復習：虚数・複素数

✓ 虚数は平方根の中が負の数:

$$x^2 = -1 \quad \Rightarrow \quad x = \pm\sqrt{-1}$$

- 虚数単位: $i = \sqrt{-1}$
- 例: $\sqrt{-2} = \sqrt{-1}\times\sqrt{2} = i\sqrt{2}$

✓ 複素数は虚数と実数の線形和: $z = x + iy$

- 実数部分: $\mathrm{Re}(z) = x$
- 虚数部分: $\mathrm{Im}(z) = y$
- 例:

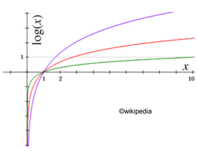
$$z = 3+2i \quad \Rightarrow \quad \mathrm{Re}(z) = 3, \quad \mathrm{Im}(z) = 2$$

復習：指数関数・自然対数

✓ 指数関数: $y = f(x) = a^x$

✓ 対数関数は指数関数の逆関数:

$$x = g(y) = \log_a y$$

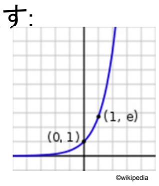


- yはaを底とするxの対数(logarithm).
- 自然対数: $\log_e y = \int_1^y \frac{1}{t}dt, \quad y = e^x$
- ネイピア数: $e = 2.71828\cdots, \quad e = \lim_{n\to\infty}(1+\frac{1}{n})^n$

復習：指数関数・exp()

- ✓ 指数関数：　$y = f(x) = a^x$
- ✓ ネイピア数が底(自然対数)の指数関数を exponential functionの略でexp()と表す：

$$y = e^x \quad \Rightarrow \quad y = f(x) = \exp(x)$$



- 様々な数学的に良い性質：

$$\frac{d}{dx}e^x = e^x, \quad e^{i\theta} \overset{\text{オイラー公式}}{=} \cos\theta + i\sin\theta, \quad etc.$$

- 関数解析・統計→信号・画像処理.

第1回レポート04

- ✓ フォーマット変換4問
- ✓ 雛形：

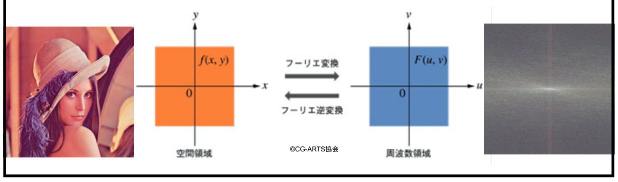
www.riken.jp/brict/Yoshizawa/Lectures/Report04.doc

- ✓ 10月21日〆切.
- ✓ 提出先＆レポートの注意点：

www.riken.jp/brict/Yoshizawa/Lectures

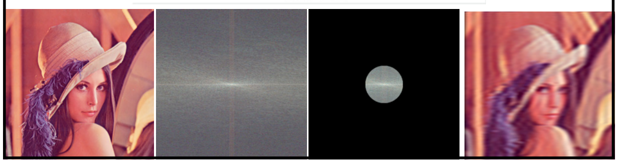
周波数分解

入力画像 ⇨ フーリエ変換、Wavelet変換、KL展開等の関数展開 ⇨ 周波数・係数列



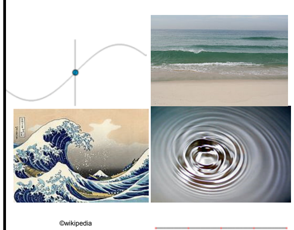
周波数操作

入力画像 ⇨(変換) 周波数 ⇨(処理) 処理後の周波数 ⇨(逆変換) 出力画像

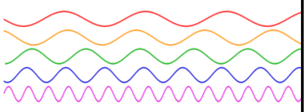


周波数(Frequency)

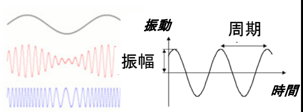
- ✓ 周波数・振動数：波動・振動周期の逆数(1/周期).
- ✓ 周期(Period)：1循環するまでの時間.
- ✓ 振幅（Amplitude）：振動の大きさ.



低周波：ゆるやか＝大きな特徴



高周波：こまやか＝シャープな特徴



周波数操作、実はみなさんよく使ってる！？

入力信号（音楽・音声） ⇨(周波数分解) 周波数 ⇩(周波数操作) 変更された周波数 ⇦(再構成（逆変換）) 出力信号（エフェクト済み）

- ✓ 音楽・音声の再生ソフトウェア・ハードウェアでのイコライザー(Equalizer)など.

意味あるの？役に立つの？

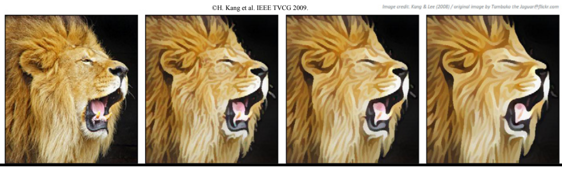
Shin Yoshizawa: shin@riken.jp

✓ デジタル・エンターテイメント応用：画像のアート処理・NPR(Non-Photorealistic Rendering)等に有用.

©J.Collomosse1 and J. Kyprianidis, EG'11.

©R. Fattal et al., SIGGRAPH 2007.

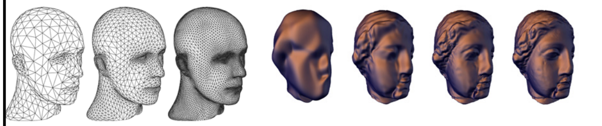
©H. Kang et al. IEEE TVCG 2009.



形状処理への応用

Shin Yoshizawa: shin@riken.jp

✓ ３次元メッシュなど形状への応用：



©D. Zorin et al., SIGGRAPH Corse 2000.

©M. Eck et al, SIGGRAPH'95.

©B. Levy and H. Zhang, SIGGRAPH Course 2010.

復習：三角関数

Shin Yoshizawa: shin@riken.jp

✓ 直角三角形の角度から辺の長さの比を与える.

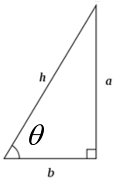
$$\sin\theta = \frac{a}{h}, \qquad \cos\theta = \frac{b}{h}$$

$$\tan\theta = \frac{a}{b} = \frac{\sin\theta}{\cos\theta}$$

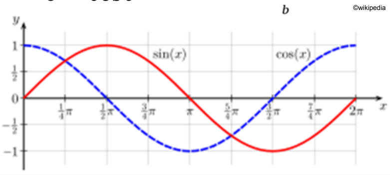
$$\cot\theta = \frac{b}{a} = \frac{1}{\tan\theta}$$

$$\csc\theta = \frac{h}{a} = \frac{1}{\sin\theta}$$

$$\sec\theta = \frac{h}{b} = \frac{1}{\cos\theta}$$



©wikipedia



関数展開

Shin Yoshizawa: shin@riken.jp

✓ 関数を基底と係数の１次(線形)結合で表す事.

✓ 例えば三角関数を基底とすると…

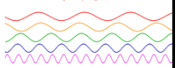
低周波 ⟷ 高周波

$$f(x) = \frac{a_0}{2} + a_1\cos\frac{\pi x}{L} + a_2\cos\frac{2\pi x}{L} + a_3\cos\frac{3\pi x}{L} + \cdots$$
$$+ b_1\sin\frac{\pi x}{L} + b_2\sin\frac{2\pi x}{L} + b_3\sin\frac{3\pi x}{L} + \cdots$$

係数＝周波数成分:a, b

基底:sin, cos

低周波



高周波

©wikipedia

関数 ＝ 係数 × 基底

関数展開2

Shin Yoshizawa: shin@riken.jp

✓ １次元では…(音声・信号処理等)

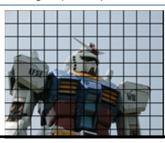
$$\text{関数 } f(x) = \text{係数} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_\infty \end{pmatrix} \times \text{基底} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_\infty \end{pmatrix}$$

低周波

高周波

✓ ２次元では…(画像処理等)

$$\text{関数 } f(u,v) = \text{係数} \times \text{基底}$$



$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1\infty} \\ a_{21} & a_{22} & \cdots & a_{2\infty} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\infty 1} & a_{a_\infty 2} & \cdots & a_{\infty\infty} \end{pmatrix} \quad \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1\infty} \\ v_{21} & v_{22} & \cdots & v_{2\infty} \\ \vdots & \vdots & \ddots & \vdots \\ v_{\infty 1} & v_{a_\infty 2} & \cdots & v_{\infty\infty} \end{pmatrix}$$

低周波

低周波

高周波

高周波

フーリエ級数

Shin Yoshizawa: shin@riken.jp

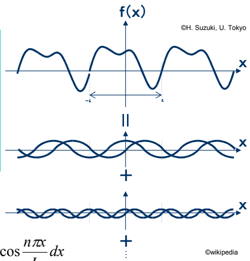
✓ フーリエ級数：[－L、L]のパターンを繰り返す周期関数を、sin(x)とcos(x)の和で表す.

$$f(x) = \frac{a_0}{2} + a_1\cos\frac{\pi x}{L} + a_2\cos\frac{2\pi x}{L} + a_3\cos\frac{3\pi x}{L} + \cdots$$
$$+ b_1\sin\frac{\pi x}{L} + b_2\sin\frac{2\pi x}{L} + b_3\sin\frac{3\pi x}{L} + \cdots$$
$$= \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n\cos\frac{n\pi x}{L} + b_n\sin\frac{n\pi x}{L}\right)$$

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i\frac{n\pi}{L}x}$$

オイラー公式

$$e^{i\theta} = \cos\theta + i\sin\theta$$



©H. Suzuki, U. Tokyo

✓ 周波数成分：基底の係数

a_nとb_nの決め方は、f(x)にcos、sinをかけて積分.

$$\begin{cases} a_n = \frac{1}{L}\int_{-L}^{L} f(x)\cos\frac{n\pi x}{L}dx \\ b_n = \frac{1}{L}\int_{-L}^{L} f(x)\sin\frac{n\pi x}{L}dx \end{cases}$$

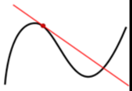
©wikipedia

Shin Yoshizawa: shin@riken.jp

復習:微積分

✓ 微分は関数の微小変化率:
- ✓ 一階微分は速度・接線.
- ✓ 二階微分は加速度・曲率

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

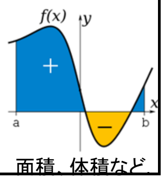


$$y = x^n \Rightarrow \frac{dy}{dx} = nx^{n-1}$$

✓ 積分は微分の逆で「和」の一般化:

$$\frac{d}{dx}\left(\int f(x)dx\right) = f(x)$$

$$y = x^n \Rightarrow \int_a^b x^n dx = \left[\frac{1}{n+1}x^{n+1}\right]_a^b = \frac{1}{n+1}(b^{n+1} - a^{n+1})$$



✓ 面積、体積など

Shin Yoshizawa: shin@riken.jp

復習:虚数・複素数

✓ 虚数は平方根の中が負の数:

$$x^2 = -1 \quad \Rightarrow \quad x = \pm\sqrt{-1}$$

- 虚数単位: $i = \sqrt{-1}$
- 例: $\sqrt{-2} = \sqrt{-1} \times \sqrt{2} = i\sqrt{2}$

✓ 複素数は虚数と実数の線形和: $z = x + iy$

- 実数部分: $\mathrm{Re}(z) = x$
- 虚数部分: $\mathrm{Im}(z) = y$
- 例:

$$z = 3 + 2i \quad \Rightarrow \quad \mathrm{Re}(z) = 3, \quad \mathrm{Im}(z) = 2$$

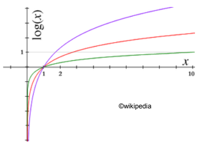
Shin Yoshizawa: shin@riken.jp

復習:指数関数・自然対数

✓ 指数関数: $y = f(x) = a^x$

✓ 対数関数は指数関数の逆関数:

$$x = g(y) = \log_a y$$



- yはaを底とするxの対数(logarithm).
- 自然対数: $\log_e y = \int_1^y \frac{1}{t}dt, \quad y = e^x$
- ネイピア数: $e = 2.71828\cdots, \quad e = \lim_{n \to \infty}(1 + \frac{1}{n})^n$

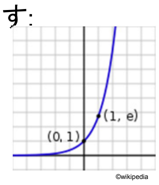
Shin Yoshizawa: shin@riken.jp

復習:指数関数・exp()

✓ 指数関数: $y = f(x) = a^x$

✓ ネイピア数が底(自然対数)の指数関数をexponential functionの略でexp()と表す:

$$y = e^x \quad \Rightarrow \quad y = f(x) = \exp(x)$$



- 様々な数学的に良い性質:

オイラー公式

$$\frac{d}{dx}e^x = e^x, \quad e^{i\theta} = \cos\theta + i\sin\theta, \quad etc.$$

- 関数解析・統計→信号・画像処理.

Shin Yoshizawa: shin@riken.jp

フーリエ変換

✓ もとの関数f(x)から、別の関数F(k)への変換:

順変換

$$F(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ikx} dx$$

逆変換

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(k) e^{ikx} dk$$

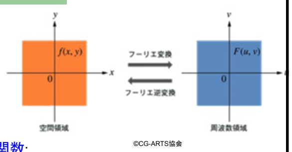
- ✓ 2組の式でフーリエ変換対をなす。フーリエ変換・逆変換という用語を使うこともある.
- ✓ フーリエ変換をf→F→fと2回行えば、元の関数に戻る.

$$e^{i\theta} = \cos\theta + i\sin\theta$$

フーリエ変換の性質

$$f(x) + g(x) \Rightarrow F(k) + G(k), \quad af(x) \Rightarrow aF(k)$$

- ✓ ある関数f(x)をF(k)の積分(〜和).
- ✓ F(k)はf(x)から積分によって計算.
- ✓ f(x)とF(k)の式は対称.
- ✓ f(x)が実数の関数でも、F(k)は一般に複素関数:
  f(x)が偶関数の場合にはF(k)は実関数(cosのみ)



©H. Suzuki, U. Tokyo

$\frac{1}{\sqrt{2\pi}}$ は規格化係数なので、あまり気にしなくて良い

Shin Yoshizawa: shin@riken.jp

離散フーリエ変換

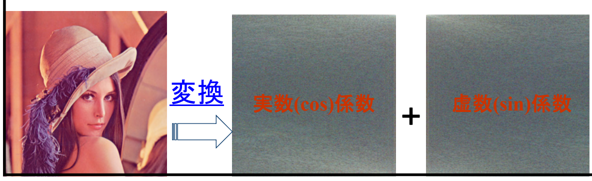
✓ デジタル画像:
- フーリエ変換の式は連続関数に対するもの.
- デジタル画像はサンプリングされて、飛び飛び(離散データ).

✓ 離散フーリエ変換・逆変換:
- 離散データに対するフーリエ変換・逆変換.
- 変換の結果の周波数列も離散的に求まる.
- 1024x1024の画像→2x1024x1024の係数列:

講義資料での周波数画像は全て二乗してlog(1+F)を適用.

$$F(k) = \sum_{i=0}^{N-1} f(s) \exp(-\frac{j2\pi ik}{N})$$

$$(k = 0, 1, \ldots, N-1)$$

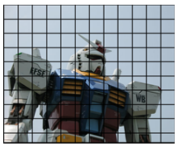


Shin Yoshizawa: shin@riken.jp

離散フーリエ変換2

- 画像では2次元なので…

$$F(u,v)=\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} f(i,j)\exp(\cdots)$$



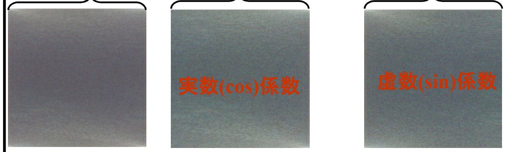
  - 1画素の離散フーリエ変換を計算するのに全ての画素の重み付和が必要！→全ての画素の変換を計算するには入力画素数の2乗に比例する計算量が必要！
- 高速フーリエ変換(FFT):次週少しやります.
  - サンプリング数が2のべき乗(例:512や1,024)の時に高速に計算する方法(Nのべき乗の方法もある).

Shin Yoshizawa: shin@riken.jp

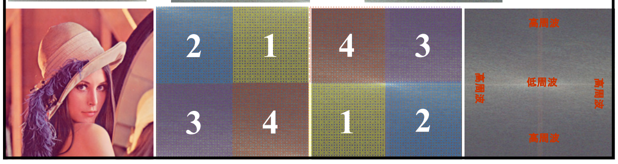
フーリエ変換2

- パワースペクトル: 周波数の強度.

$$\underbrace{|F(u,v)|^2}=\underbrace{\mathrm{Re}\{F(u,v)\}^2}+\underbrace{\mathrm{Im}\{F(u,v)\}^2}$$

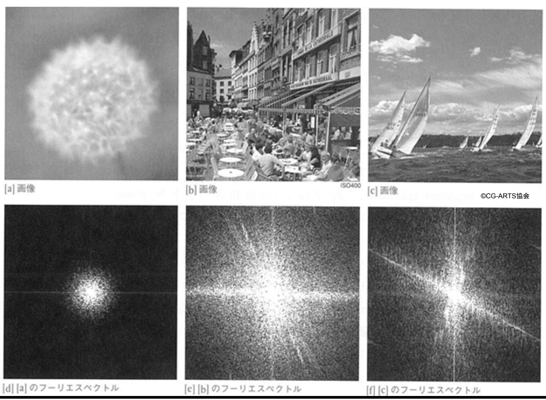


画像処理ではよく象限を1→3, 2→4, 3→1, 4→2と入れ替えた画像を用いる.(周期性を用いた離散変換を行うため)



Shin Yoshizawa: shin@riken.jp

フーリエ変換3



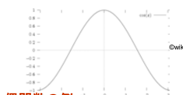
Shin Yoshizawa: shin@riken.jp

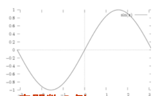
離散コサイン変換(DCT)

- 余弦関数列(cos)のみを基底に用いた変換:
  - 入力を y軸で折り返して偶関数化して離散フーリエ変換する事と同義:離散フーリエ変換は実数に対して複素数を返すのに対して、DCTは常に実数を返す.
  - 低周波成分に集中度が上がるため圧縮やフィルタ処理でよく用いられている.

順変換
$$F(u,v)=C(u)C(v)\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\cos\left(\frac{2(j+1)u\pi}{2M}\right)\cos\left(\frac{2(i+1)v\pi}{2N}\right)f(i,j)$$

逆変換
$$f(x,y)=\frac{4}{MN}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}C(i)C(j)\cos\left(\frac{2(j+1)x\pi}{2M}\right)\cos\left(\frac{2(i+1)y\pi}{2N}\right)F(i,j)$$


偶関数の例

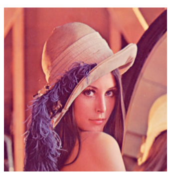


奇関数の例

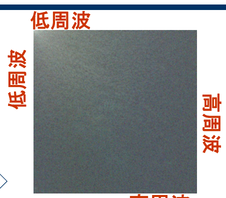
$$C(x)=\begin{cases}\frac{1}{\sqrt{2}} & (x=0)\\ 1 & (x\neq 0)\end{cases}$$

Shin Yoshizawa: shin@riken.jp

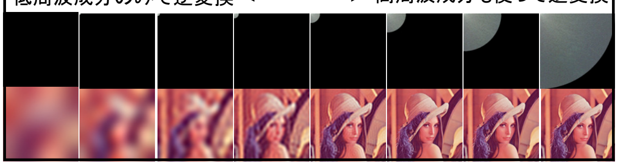
離散コサイン変換(DCT)2



DCT ⇨



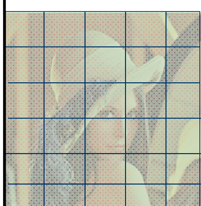
低周波成分のみで逆変換 ←→ 高周波成分も使って逆変換



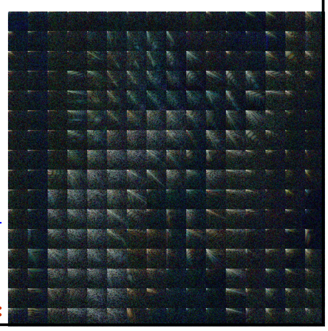
Shin Yoshizawa: shin@riken.jp

離散コサイン変換(DCT)3

- 非常に処理が重いので、FFTを使わない簡単な実装は画像を部分画像(ブロック)に分割してブロック毎に変換する:



ブロックDCT ⇨

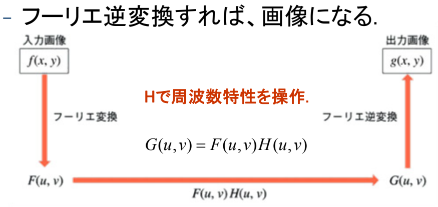


32x32のブロック毎のDCT例:

Shin Yoshizawa: shin@riken.jp

周波数フィルタリング

- ✓ 画像のフーリエ変換:
  - 空間領域から周波数領域へ.
  - フーリエ逆変換すれば、画像になる.

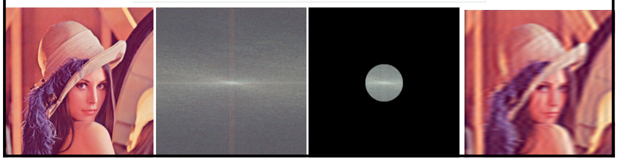


- ✓ フーリエ変換して、画像を周波数領域に変換してしまえば、フィルタリングは、二つの関数を単純に掛け算するだけ.

Shin Yoshizawa: shin@riken.jp

周波数操作

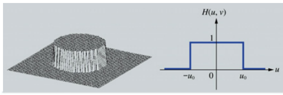
入力画像 → 変換 → 周波数 → 処理 → 処理後の周波数 → 逆変換 → 出力画像



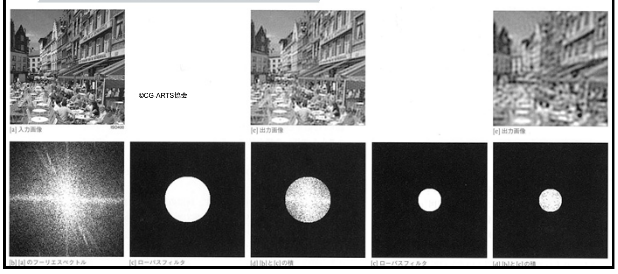
Shin Yoshizawa: shin@riken.jp

ローパスフィルタ

- ✓ −u0から、u0までの低周波数成分だけ残す.

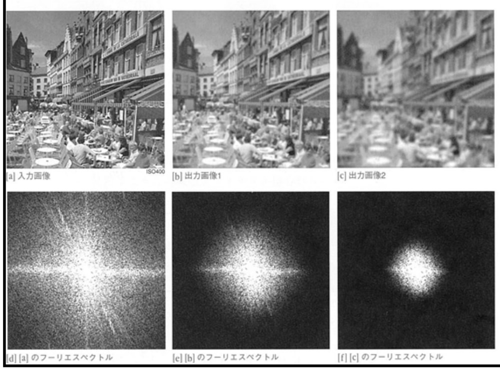


周波数の高い横方向の波(縦縞)を消す.

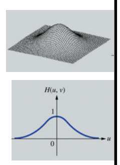


Shin Yoshizawa: shin@riken.jp

ローパスフィルタ2



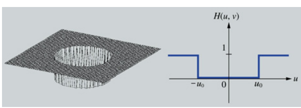
$(u,v)=(0,0)$のフィルタの値が1なので、$(u,v)=(0,0)$の成分が保存される
→画像の平均的な明るさが保持される.



Shin Yoshizawa: shin@riken.jp

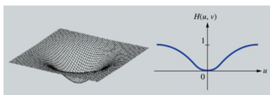
ハイパスフィルタ

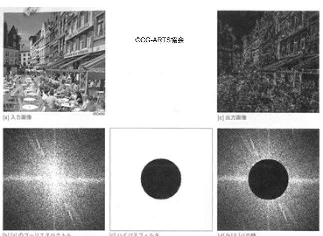
- ✓ −u0から、u0までの高周波数成分だけ残す.



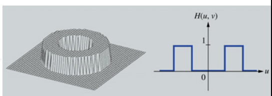
$$H_{high}(u,v) = 1 - H_{low}(u,v)$$

- ✓ 1からロウパスを引く:





- ✓ バンドパスフィルタ:特定周波数成分の抽出.



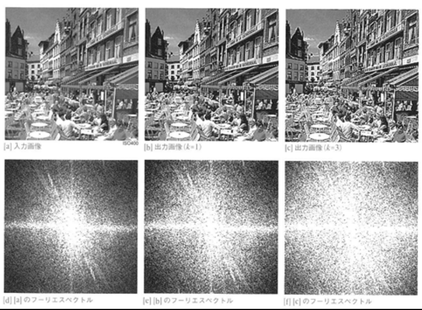
Shin Yoshizawa: shin@riken.jp

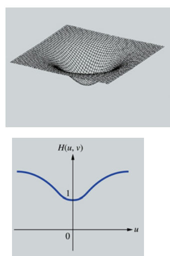
高域強調フィルタ

- ✓ ハイパスフィルタから作る事が出来る.

$$H_{h-emph}(u,v) = 1 + kH_{high}(u,v)$$

エッジ強調!

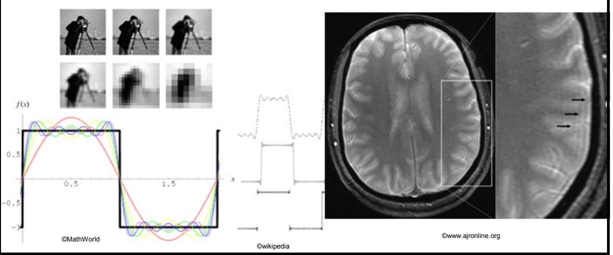




Shin Yoshizawa: shin@riken.jp

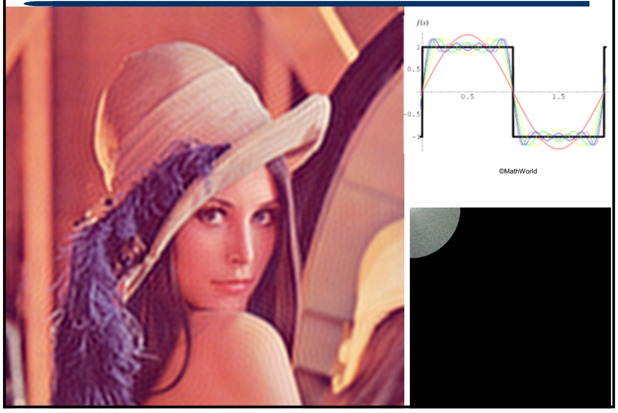
ギブス現象・Overshooting

✓ ローパスフィルタ＝高周波成分の切り捨てはデータにエッジがあった場合に不連続なデータを連続関数で近似するためエッジ周辺での誤差が非常に大きくなる事：画像ではリングアーティファクトと呼ばれている：　圧縮や補間等でのカーネルの打ち切り誤差.



Shin Yoshizawa: shin@riken.jp

ギブス現象・Overshooting2

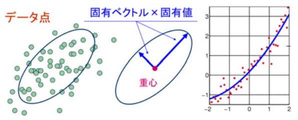


Shin Yoshizawa: shin@riken.jp

その他の変換

✓ フーリエ変換以外にも、様々な基底を用いた関数展開が幅広く周波数解析に用いられている.

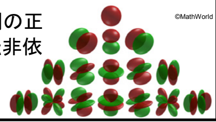
- KL(Karhunen–Loeve)展開：データの共分散行列の固有ベクトルを基底とする. 最小二乗的に最もデータを近似出来る展開.　←主成分分析(PCA)の一般化



PCA: Principal Component Analysis

主成分分析：与えられた点群データに対して最小二乗的に最も相関が強い方向と強度を計算する:
- 直線、平面、Hyperplane 等のデータへの当てはめ(最小二乗近似).
- Covariance matrix(共分散行列=平均からの差)の固有値・ベクトルは Best fit 楕円、ellipsoid等の近似.

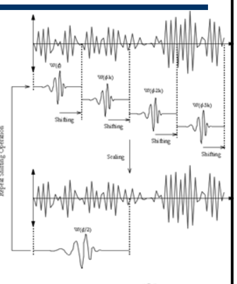
- 球面調和関数:超球面上の関数空間の正規直交基底(円や球への離散化で回転非依存にしやすい).
- Zernike関数、固有関数展開、etc.

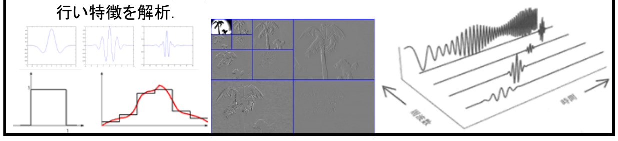


Shin Yoshizawa: shin@riken.jp

その他の変換2

✓ Wavelet:　入力信号を小さな波形の拡大縮小と平行移動の重ね合わせで表現.
- フーリエ変換は時間軸上で常に一定のパターンを持ったデータ解析に有用だが、時刻によってパターンが変化するデータ解析には不向きである. ウェーブレット変換では局所的な波を平行移動と拡大縮小で波を表現するため、有限の区間内にあるデータの特性を解析するには三角関数より適している.
- 多重解像度解析(Multiresolution Analysis)：パターンを周波数分解する作業を繰り返し行い特徴を解析.

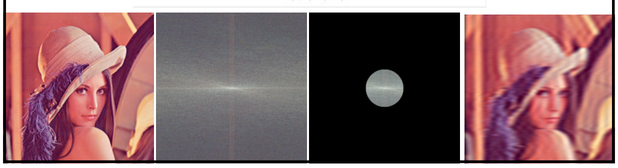




Shin Yoshizawa: shin@riken.jp

周波数分解と操作

入力画像 ⇒ 変換 ⇒ 周波数 ⇒ 処理 ⇒ 処理後の周波数 ⇒ 逆変換 ⇒ 出力画像



Shin Yoshizawa: shin@riken.jp

演習: DCT

www.riken.jp/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Lec15.pdf

離散コサイン変換による周波数フィルタ
www.riken.jp/brict/Yoshizawa/Lectures/Ex08.zip

前回の演習(BMPとPPMの相互変換)が分からなかった or 出来ていない or 欠席した人は、<u>前回の演習から始める事！</u>

Shin Yoshizawa: shin@riken.jp

演習:Ex08の説明

- ✓ DCT.h: 離散コサイン変換のブロック実装:
- ✓ 順変換:Image *DCT(Image *in,int X,int Y): 入力画像inをX×YブロックでDCTを実行し周波数画像を戻り値で返す.
  - 注意点:周波数画像は入力画像サイズがブロックサイズで割り切れない場合は入力画像サイズより少し大きなサイズで作成される.
- ✓ 逆変換:

void InverseDCT(Image *dct,Image *out, intX,int Y): DCT()にて変換された画像dctをX×Yブロックで逆変換し出力画像outへ保存.

Shin Yoshizawa: shin@riken.jp

演習:Ex08の説明2

- ✓ testDCT.cxx: 離散コサイン変換の例.
- ✓ makeでコンパイル.
- ✓ 引数4つ:
  1. 入力BMPファイル名.
  2. 出力ファイル名(ただし拡張子.bmpなし).
  3. ブロックサイズ(int).
  4. 周波数の閾値(int): 高周波を四角にカット.

./testDCT 入力BMP 出力ファイル名(.bmp抜き) ブロックサイズ(int) 周波数の閾値(int)

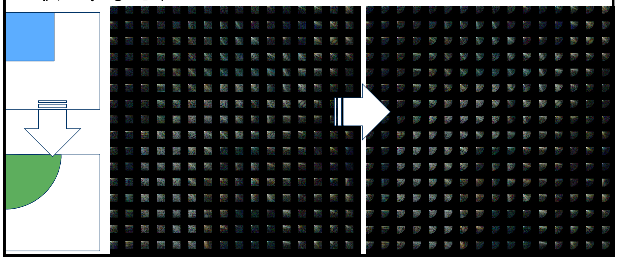
- ✓ 出力は3つのbmp画像ファイル:

出力ファイル名_spectrum.bmp
出力ファイル名_smooth.bmp
出力ファイル名_smooth_spectrum.bmp

Shin Yoshizawa: shin@riken.jp

演習:周波数フィルタ

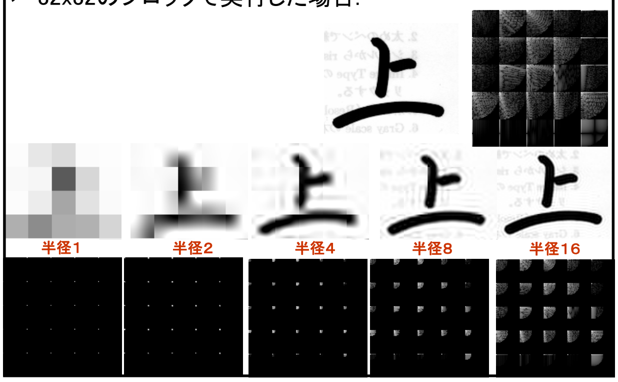
- ✓ testDCT.cxxを編集して円状に高周波をゼロにするローパスフィルタを作ってみましょう!
- ✓ 16x16のブロックで半径1,2,3,4,8で実行してみてください.
- ✓ ヒント:testDCT.cxxは四角に低周波を残しているので、円状にするだけ.



Shin Yoshizawa: shin@riken.jp

演習の正解例

- ✓ 32x32のブロックで実行した場合:



Shin Yoshizawa: shin@riken.jp

演習の正解例2

- ✓ 32x32のブロックで実行した場合:

- ✓ Ex08中のSeikai.zip内にStrasborug2.bmpでの正解画像が入っています.

Shin Yoshizawa: shin@riken.jp

演習:周波数フィルタ2

- ✓ 以下の周波数フィルタのプログラムを作ってみましょう!
  1. ローパスフィルタ: 円状に高周波をゼロにする方法とガウス関数を使う方法両方.
  2. ハイパスフィルタ:円状に高周波をゼロにする方法.
  3. バンドパスフィルタ:円状に高周波をゼロにする方法.
  4. エッジ強調フィルタ:円状に高周波をゼロにする方法とガウス関数を使う方法両方.
  - ヒント:ガウス関数の画像を作って正規化(画素の和で割る)し、DCT後に入力のDCT画像とかけて逆変換.

$$g(x,y) = \exp(-\frac{x^2+y^2}{2\sigma^2})$$

第2回レポートは↑を含むので頑張ってーp(^^)q