2024年度秋学期

# 画像情報処理

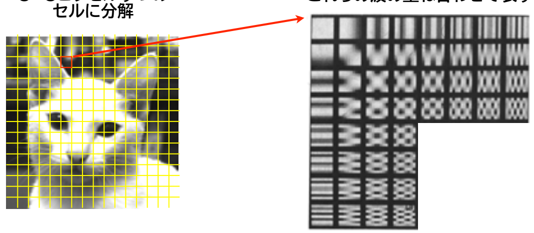
第2部・画像情報圧縮 ／

## 第8回 行列の直交変換と基底画像

関西大学総合情報学部
浅野 晃

# JPEG方式による画像圧縮

画像を波の重ね合わせで表わし，一部を省略して，データ量を減らす




細かい部分は，どの画像でも大してかわらないから，省略しても気づかない

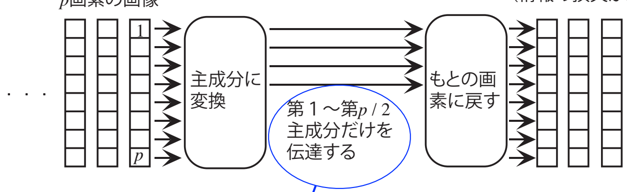
省略すると，データ量が減る

（図はA. K. Jain, Fundamentals of Digital Image Processingより転載）



# Karhunen-Loève変換（KL変換）

画像を主成分に変換してから伝送する




データ量が半分でも
情報の損失は最小



# KL変換の大問題

主成分を求めるには，分散共分散行列が必要

分散共分散行列を求めるには，
「いまから取り扱うすべての画像」が
事前にわかっていないといけない

そんなことは不可能😵
（一応）

じゃあ, 主成分を求めるのはあきらめて,
どういう直交変換をするか「直観的」に🤔



画像をベクトルにしてしまったら,
直観がはたらかない…



行列の直交変換💡



# 画像を行列であらわす

平面のものを素直に表せばいいだけのことですが,
前回はベクトルで考えていたので。

$$z = P'x$$

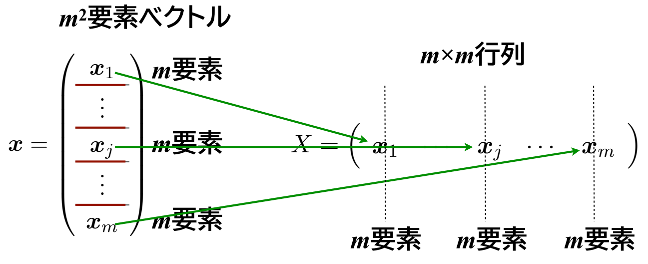
変換後の画像を表すベクトル（$m^2$要素）

原画像を表すベクトル（$m^2$要素）

直交変換を表す行列（$m^2 \times m^2$）

ベクトルから行列に書き換える（戻す）ことを考える

# ベクトルを行列に書き換える




$z$も同じ



# 直交変換行列$P'$は？

$P'$ がこういう形になっているのなら

$$P' = \begin{pmatrix} r_{11}c_{11} & \cdots & r_{11}c_{1m} & & r_{1m}c_{11} & \cdots & r_{1m}c_{1m} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ r_{11}c_{m1} & \cdots & r_{11}c_{mm} & & r_{1m}c_{m1} & \cdots & r_{1m}c_{mm} \\ & \vdots & & \ddots & & \vdots & \\ r_{m1}c_{11} & \cdots & r_{m1}c_{1m} & & r_{mm}c_{11} & \cdots & r_{mm}c_{1m} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ r_{m1}c_{m1} & \cdots & r_{m1}c_{mm} & & r_{mm}c_{m1} & \cdots & r_{mm}c_{mm} \end{pmatrix}$$

こういう形ってどういう形？



# 行列のKronecker積

$$P' = \begin{pmatrix} \underset{r_{11}\times C}{\boxed{\begin{matrix} r_{11}c_{11} & \cdots & r_{11}c_{1m} \\ \vdots & & \vdots \\ r_{11}c_{m1} & \cdots & r_{11}c_{mm} \end{matrix}}} & \cdots & \underset{r_{1m}\times C}{\boxed{\begin{matrix} r_{1m}c_{11} & \cdots & r_{1m}c_{1m} \\ \vdots & & \vdots \\ r_{1m}c_{m1} & \cdots & r_{1m}c_{mm} \end{matrix}}} \\ \vdots & \ddots & \vdots \\ \underset{r_{m1}\times C}{\boxed{\begin{matrix} r_{m1}c_{11} & \cdots & r_{m1}c_{1m} \\ \vdots & & \vdots \\ r_{m1}c_{m1} & \cdots & r_{m1}c_{mm} \end{matrix}}} & \cdots & \underset{r_{mm}\times C}{\boxed{\begin{matrix} r_{mm}c_{11} & \cdots & r_{mm}c_{1m} \\ \vdots & & \vdots \\ r_{mm}c_{m1} & \cdots & r_{mm}c_{mm} \end{matrix}}} \end{pmatrix}$$

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mm} \end{pmatrix}, \quad R = \begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix}$$

こうなっているのなら

$R$の各要素に
$C$を貼付けたもの

$$P' = R \otimes C$$ Kronecker積



# 行列の変換に書き換える

$$z = P'x$$

ベクトル$x$から
ベクトル$z$への
行列$P'$による変換

証明は…ひたすら計算💦（付録1）

$$Z = CXR'$$

行列$X$から
行列$Z$への
行列$C$と$R'$による変換

# $P'$ が直交行列であるためには

直交行列… 異なる列の内積は0，同じ列同士の内積は1

$$P' = \begin{pmatrix} r_{11}c_{11} & \cdots & r_{11}c_{1m} & & r_{1m}c_{11} & \cdots & r_{1m}c_{1m} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ r_{11}c_{m1} & \cdots & r_{11}c_{mm} & & r_{1m}c_{m1} & \cdots & r_{1m}c_{mm} \\ & \vdots & & \ddots & & \vdots & \\ r_{m1}c_{11} & \cdots & r_{m1}c_{1m} & & r_{mm}c_{11} & \cdots & r_{mm}c_{1m} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ r_{m1}c_{m1} & \cdots & r_{m1}c_{mm} & & r_{mm}c_{m1} & \cdots & r_{mm}c_{mm} \end{pmatrix}$$

$P' = R \otimes C$ なら

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mm} \end{pmatrix}, \quad R = \begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix}$$

***C*, *R*それぞれが直交行列なら，*P'*は直交行列**



# 分離可能性

$$CXR' =$$

$$\begin{pmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mm} \end{pmatrix} \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mm} \end{pmatrix} \begin{pmatrix} r_{11} & \cdots & r_{m1} \\ \vdots & \ddots & \vdots \\ r_{1m} & \cdots & r_{mm} \end{pmatrix}$$

***C*は*X*の列に作用** 　　 ***R*は*X*の行に作用**

縦方向と横方向の作用を分離できることを，
分離可能（separable）という



# 行列の直交変換とユニタリー変換

縦横の作用を区別する必要はない場合，*C*=*R*とする

$$Z = RXR' \qquad X = R'ZR$$

ただし　$RR'=I$　　行列*X*の行列*R*による直交変換

*は複素共役（ *i* を(−*i*)にかえる）

要素が複素数の場合は，*R'*のかわりに *R'**を用いる

行列*X*の行列*R*によるユニタリー変換

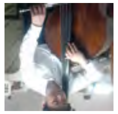


ちょっと余談ですが☕

# 縦横の作用を区別する必要はないのか？

画像処理としてはその仮定はおかしくないが，

現実世界においては，重力があるので，左右と上下は異なる



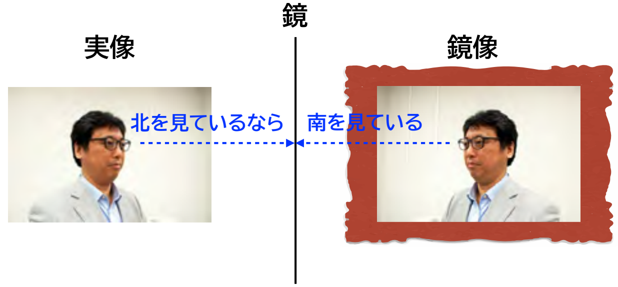
上下反転のほうが違和感が大きい

だから



# 鏡ではなぜ左右だけ逆になるのか？

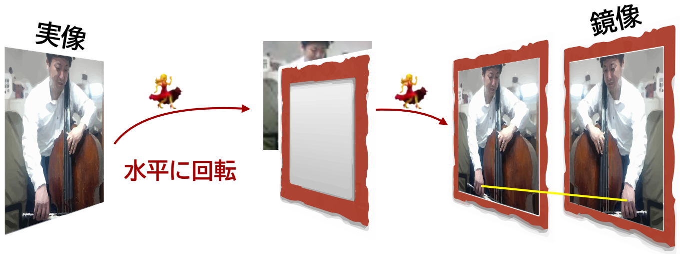
鏡で逆になっているのは，左右でも上下でもなく　前後。






# 鏡ではなぜ左右だけ逆になるのか？

「鏡で逆になる」というなら，「正解」はなにか？




🐻💭 正解はこれしかないでしょう？

左右が反転
上下はそのまま



# 鏡ではなぜ左右だけ逆になるのか？

👨‍🎓💬 いいえ，正解はそれだけではありません


水平回転が正しいと思うのは
重力の都合でしかない

上下が反転
左右はそのまま

ここで参考動画を

基底画像🤔



# 基底画像

$$Z = RXR'$$

どういう$\boldsymbol{R}$を用いれば,
最適に画像データを圧縮できるか？

それは, 依然わからない

しかし, 画像をベクトルでなく行列で表したことで,
直交変換の効果がヴィジュアルにわかる



# 基底画像

変換後の画像$\boldsymbol{Z}$の$\boldsymbol{m^2}$個の要素を, それぞれ行列に分ける

$$Z = \begin{pmatrix} z_{11} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} + \begin{pmatrix} 0 & z_{12} & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} + \cdots + \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & z_{mm} \end{pmatrix}$$

$X = R'ZR$ を, 上の各行列で行う。たとえば

$$\begin{pmatrix} r_{11} & \cdots & r_{m1} \\ \vdots & \ddots & \vdots \\ r_{1m} & \cdots & r_{mm} \end{pmatrix} \begin{pmatrix} z_{11} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix}$$



# 基底画像

$r_{11}$が残る　かけ算　かけ算　$r_{11}$が残る

$$\begin{pmatrix} r_{11} & \cdots & r_{m1} \\ \vdots & \ddots & \vdots \\ r_{1m} & \cdots & r_{mm} \end{pmatrix} \begin{pmatrix} z_{11} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix}$$

この列が残る　この行が残る

$$\begin{pmatrix} r_{11} & \cdots & r_{m1} \\ \vdots & \ddots & \vdots \\ r_{1m} & \cdots & r_{mm} \end{pmatrix} \begin{pmatrix} z_{11} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix}$$

ベクトルの直積
（付録3）

$$z_{11} \begin{pmatrix} r_{11} \\ \vdots \\ r_{1m} \end{pmatrix} \begin{pmatrix} r_{11} \cdots r_{1m} \end{pmatrix} = z_{11} \begin{pmatrix} r_{11}r_{11} & r_{11}r_{12} & \cdots & r_{11}r_{1m} \\ r_{12}r_{11} & r_{12}r_{12} & \cdots & r_{12}r_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1m}r_{11} & r_{1m}r_{12} & \cdots & r_{1m}r_{1m} \end{pmatrix}$$

行列, すなわち画像

ここでちょっと
「直積ジョーク」を…😊

# 基底画像

つまり

$$X = z_{11}\boldsymbol{r}_1\boldsymbol{r}'_1 + z_{12}\boldsymbol{r}_1\boldsymbol{r}'_2 + \cdots + z_{mm}\boldsymbol{r}_m\boldsymbol{r}'_m$$

基底画像

原画像$X$は, $m^2$個の基底画像に

それぞれ$Z$の各要素をかけて足し合わせたものになっている



# つまり基底画像とは

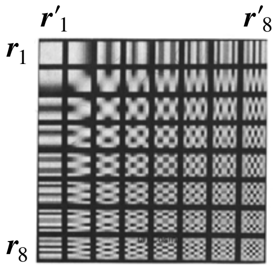
原画像$X$は, $m^2$個の基底画像に

それぞれ$Z$の各要素をかけて足し合わせたものになっている

たとえば, 64個( $m = 8$ )の基底画像が, 右のような

$r_1 \dots r_8$ と $r'_1 \dots r'_8$ の直積になっていると

すると




(図はA. K. Jain, Fundamentals of Digital Image Processingより転載)



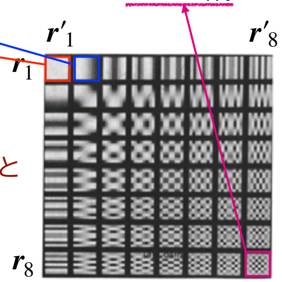
# つまり基底画像とは

原画像$X$は, $m^2$個の基底画像に

それぞれ$Z$の各要素をかけて足し合わせたものになっている

$$X = z_{11}\boldsymbol{r}_1\boldsymbol{r}'_1 + z_{12}\boldsymbol{r}_1\boldsymbol{r}'_2 + \cdots + z_{mm}\boldsymbol{r}_m\boldsymbol{r}'_m$$

64個( $m = 8$ )の基底画像がこれだとすると




(図はA. K. Jain, Fundamentals of Digital Image Processingより転載)



# つづきは

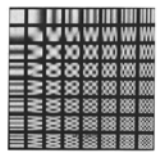
原画像$X$は, $m^2$個の基底画像に

それぞれ$Z$の各要素をかけて足し合わせたものになっている

今日の最初にでてきた

これ(の8×8の1つ1つ)は

基底画像の例です👉



(図はA. K. Jain, Fundamentals of Digital Image Processingより転載)

元の関数は, いろいろな周波数の波に,

各々対応するフーリエ係数をかけて足し合わせたものになっている…

第1部の

👈 これと同じ？

# つづきは

原画像$X$は, $m^2$個の基底画像に
それぞれ$Z$の各要素をかけて足し合わせたものになっている

↕

元の関数は, いろいろな周波数の波に,
各々対応するフーリエ係数をかけて足し合わせた
ものになっている…

つまり, 逆フーリエ変換？

フーリエ変換も, ユニタリー変換の一種

フーリエ変換を基本に,
画像圧縮に適した基底画像（一部を省略しても影響が少ない基底画像）を選ぶ