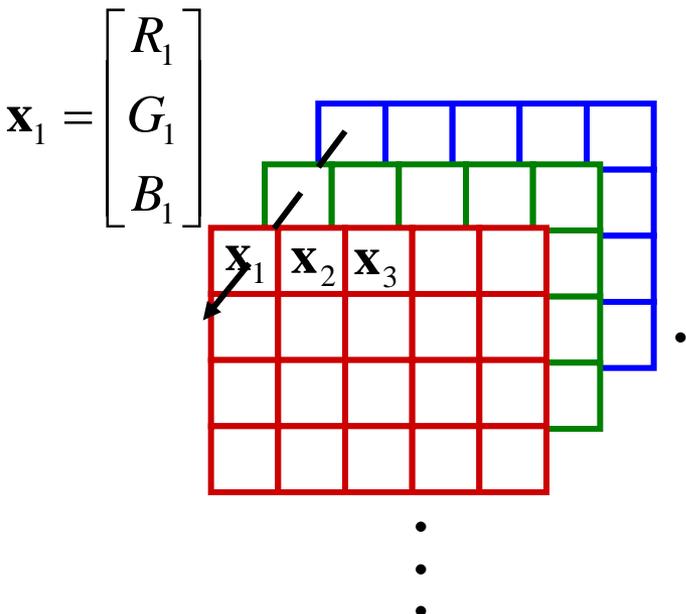
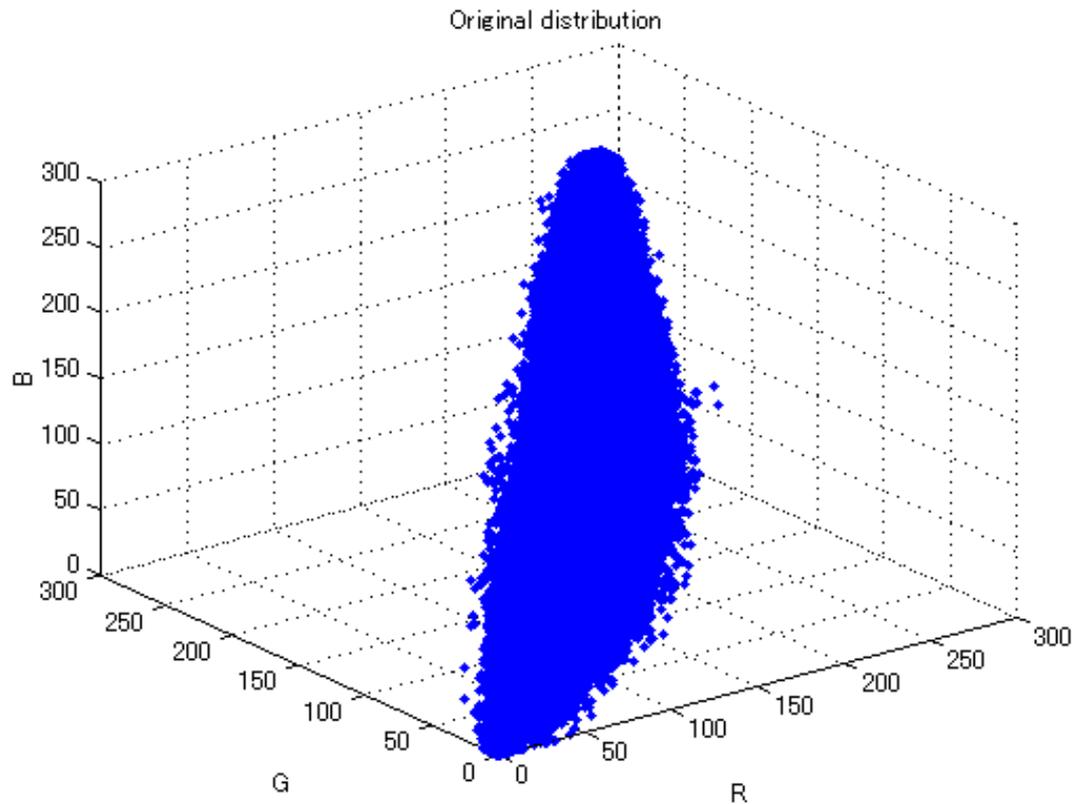


# 例 1 RGBカラー信号の主成分分析

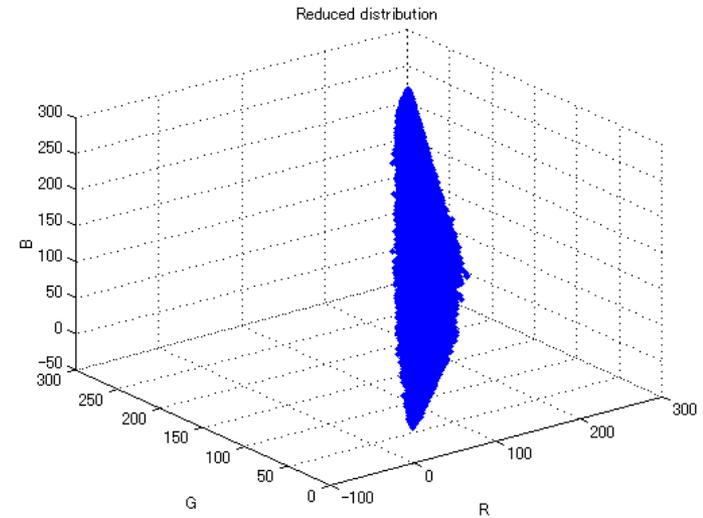
## オリジナル画像



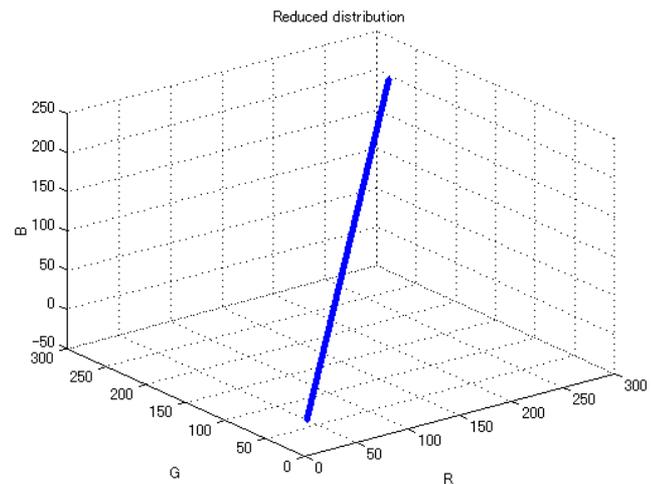
RGB空間での画素値の分布

# 例1 RGBカラー信号の主成分分析

## 第1 および第2主成分のみ



## 第1主成分のみ



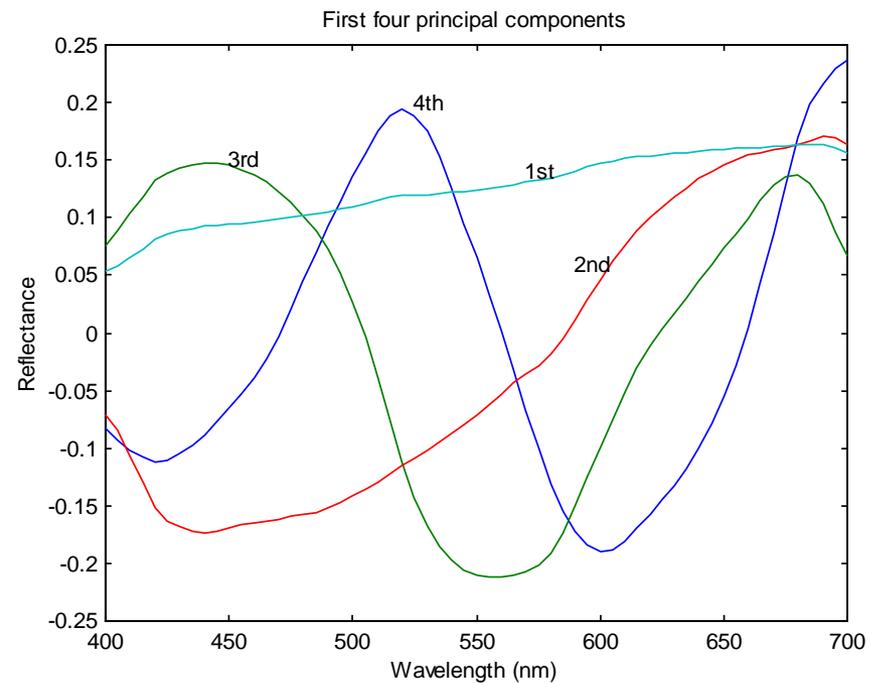
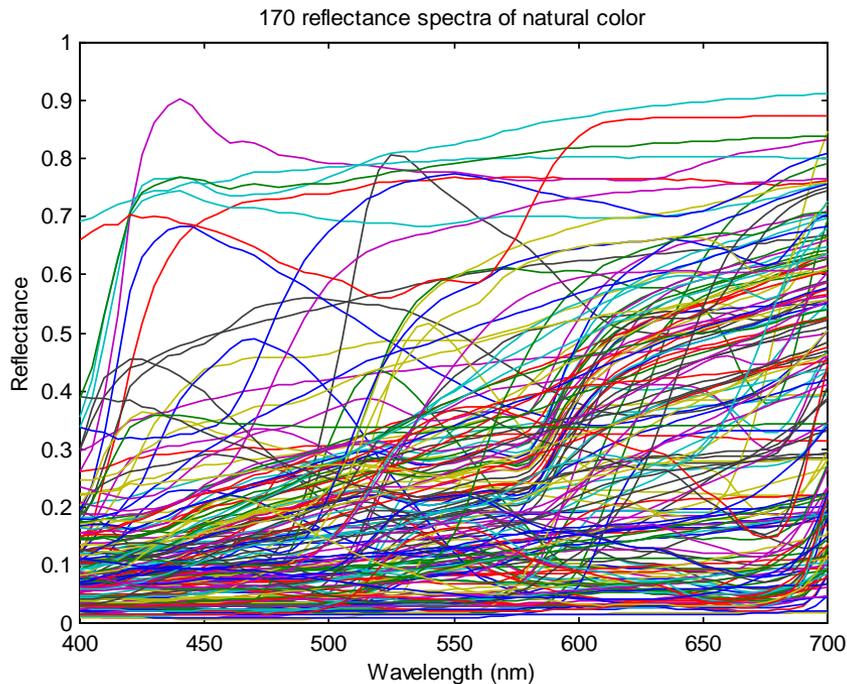
# 例2 ～分光反射率の近似～

波長：400,404,...,700nm -> 61次元

自然界の分光反射率サンプル

主成分分析

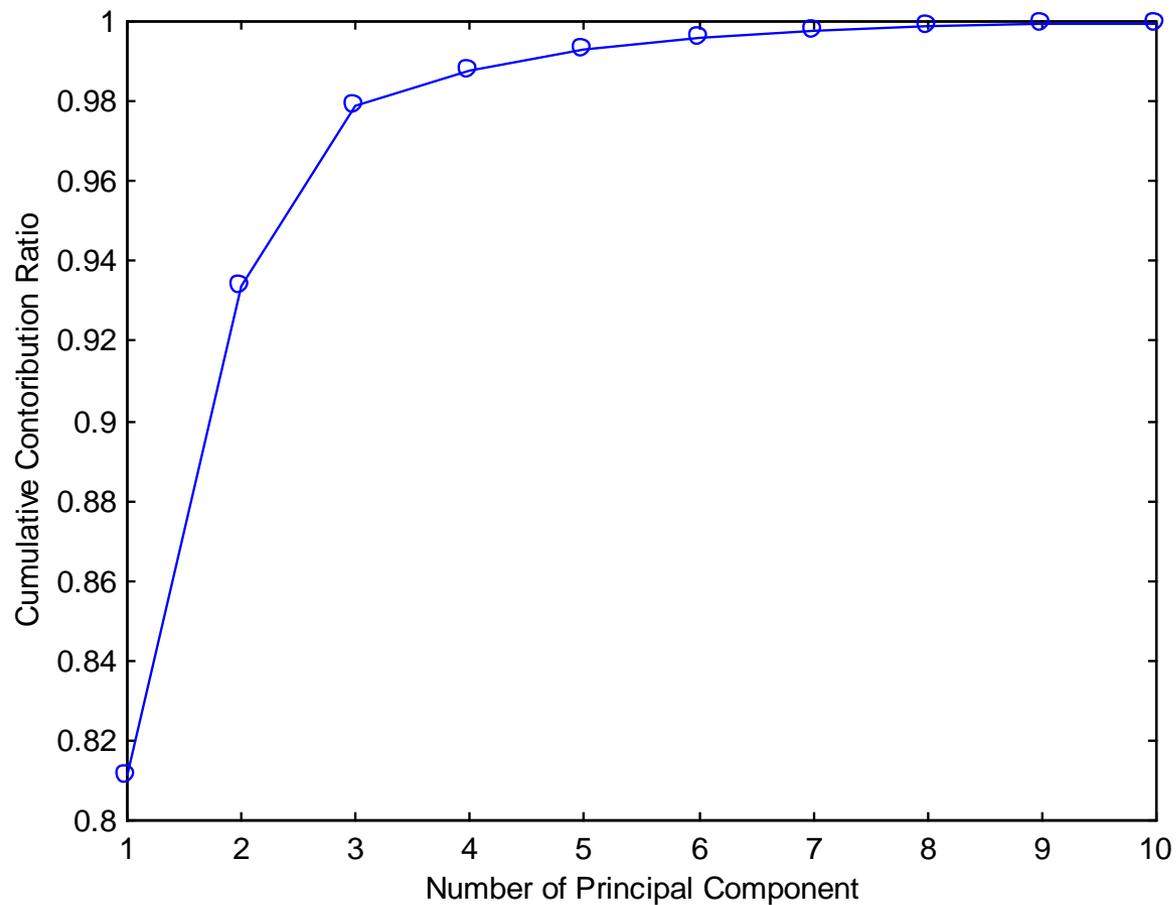
主成分（第1～第4）



$$r(\lambda) = m(\lambda) + \sum_{i=1}^{61} k_i u_i(\lambda) \quad \longrightarrow \quad r(\lambda) \approx m(\lambda) + \sum_{i=1}^4 k_i u_i(\lambda)$$

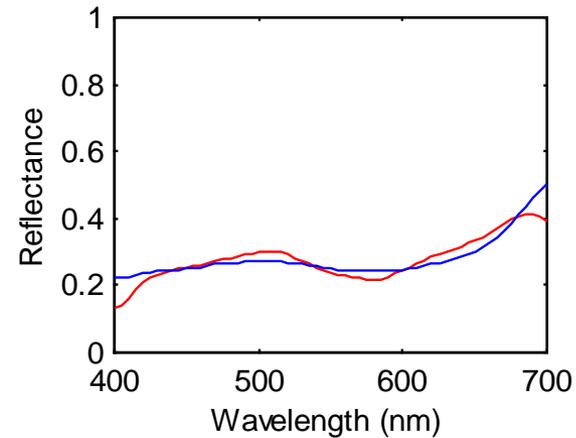
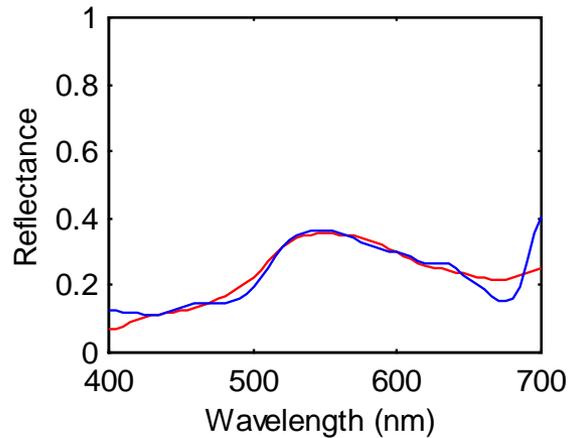
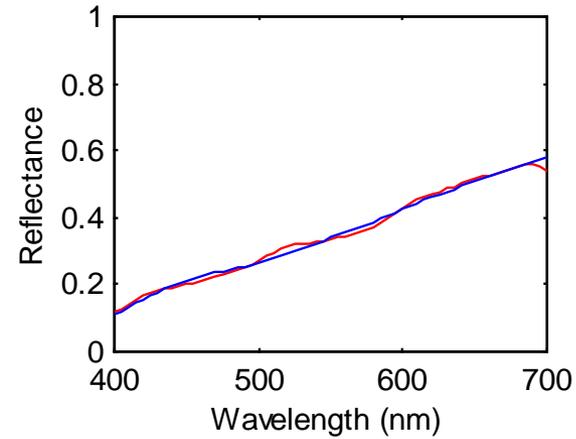
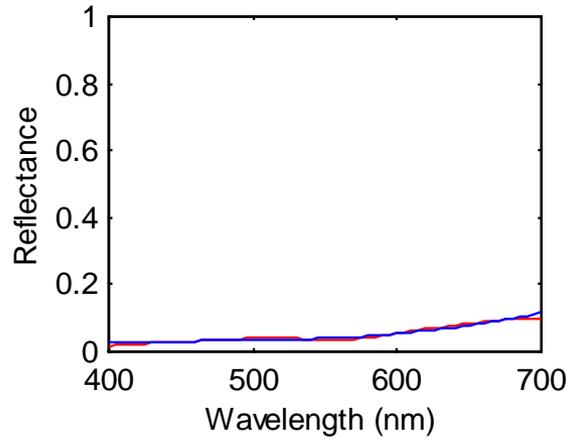
低い次元数で元の信号を表現できる。

# 分光反射率データの累積寄与率



# 4次までの主成分で近似した例

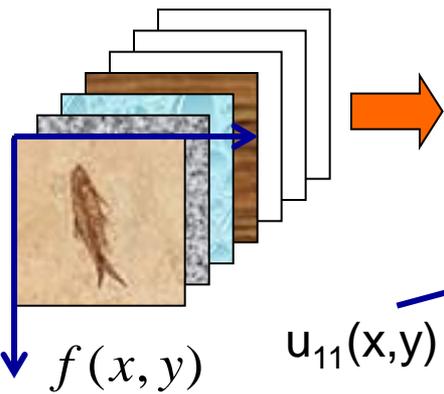
— オリジナル  
— 近似



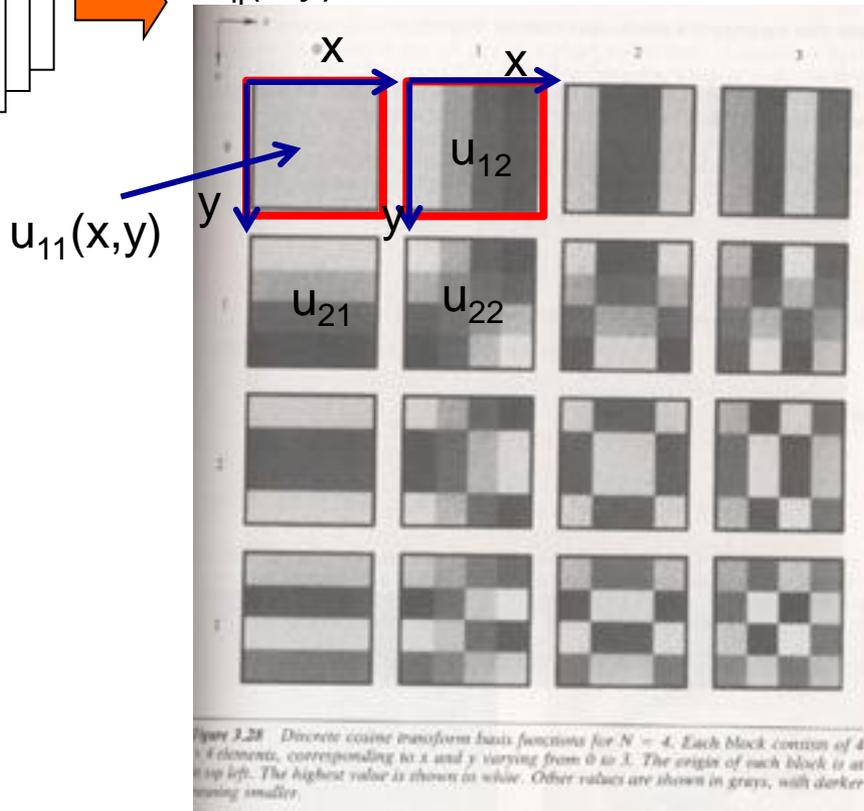
# 例3 画像圧縮

PCA  $\equiv$  Hotelling Transform or Karhunen-Loeve transform

オリジナル画像群

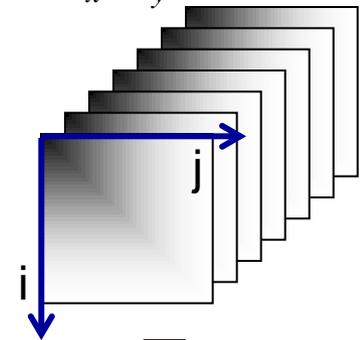
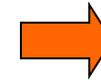


K-L変換（主成分分析）により、分散の大きい順番に基底ベクトル  $u_{ij}(x, y)$  (=基底画像) を算出。

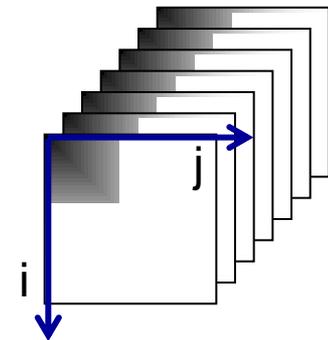


係数算出：（内積演算）

$$k(i, j) = \sum_x \sum_y f(x, y) u_{ij}(x, y)$$



分散の大きい係数のみ保存

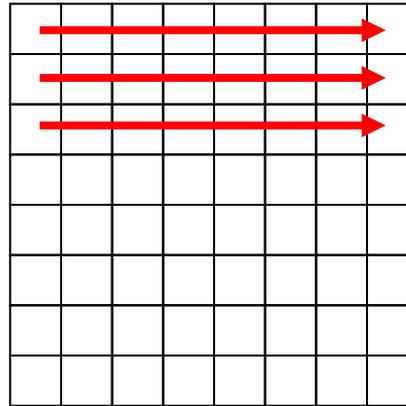


戻すときは逆の手順で

# 画像の主成分分析例



1番目のブロック (8x8=64画素)



$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_{64}^{(i)} \end{bmatrix} \quad \text{画素値}$$

画像を小ブロックに分割

各小ブロック内の画素を  
ラスタスキャンの順に並  
べて列ベクトルにする

データセット

$$\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$$

に対して主成分分析を行い、主成分ベクトルを求め、画像圧縮に用いる。

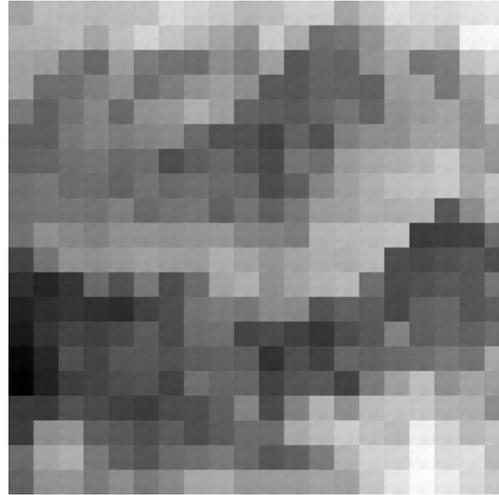
# オリジナルおよび復元画像

オリジナル



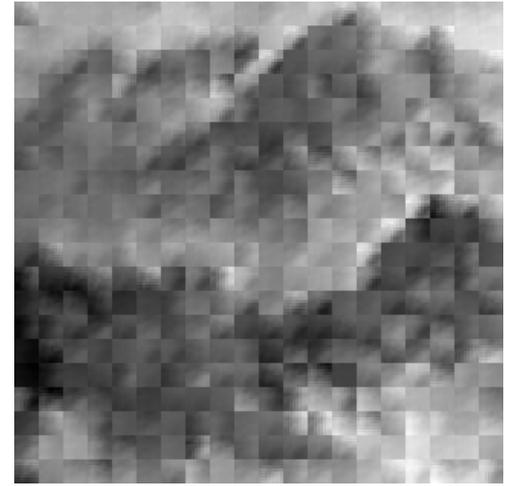
成分数：1

# of synthesized components: 1



成分数：3

# of synthesized components: 3



成分数：10

# of synthesized components: 10



成分数：20

# of synthesized components: 20



成分数：30

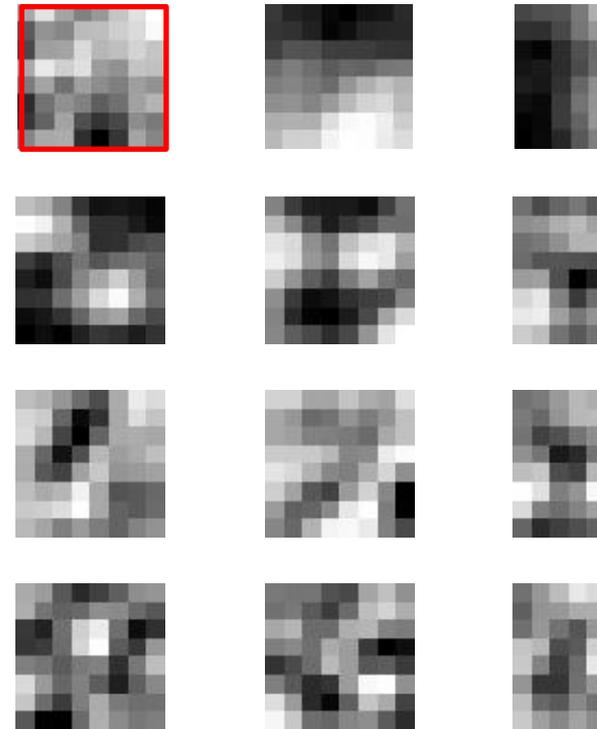
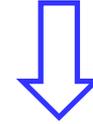
# of synthesized components: 30



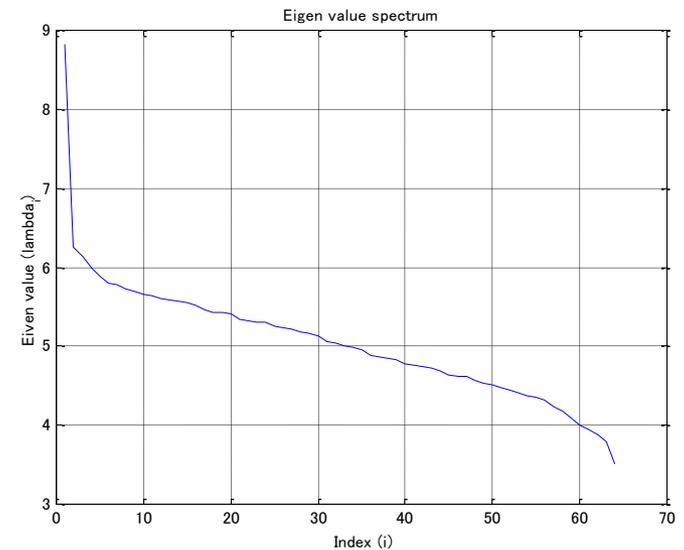
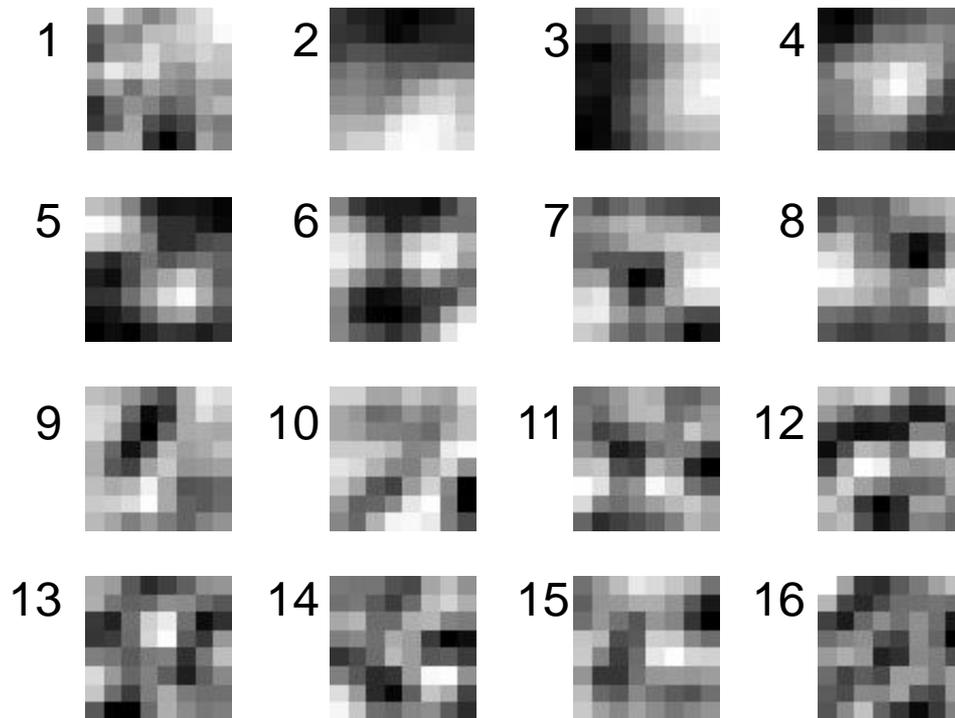
## 第1主成分ベクトルだけで近似した画像の拡大図

平均ベクトルが加算されているため、右図の  
パターンが顕著に見えるわけではない

第1主成分ベクトル



# 画像の主成分分析例



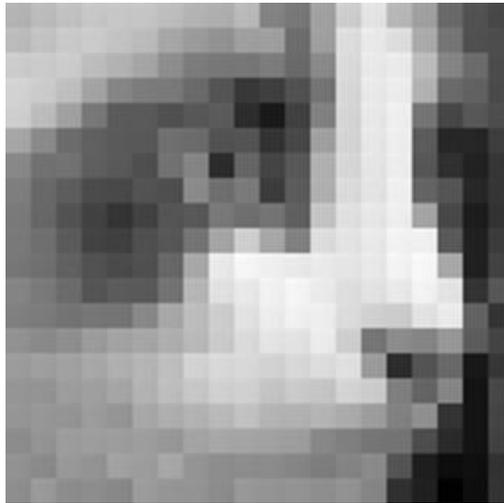
固有値スペクトル

固有値の大きい順に並べた  
最初の16枚の主成分画像

オリジナル



成分数：1  
# of synthesized components: 1



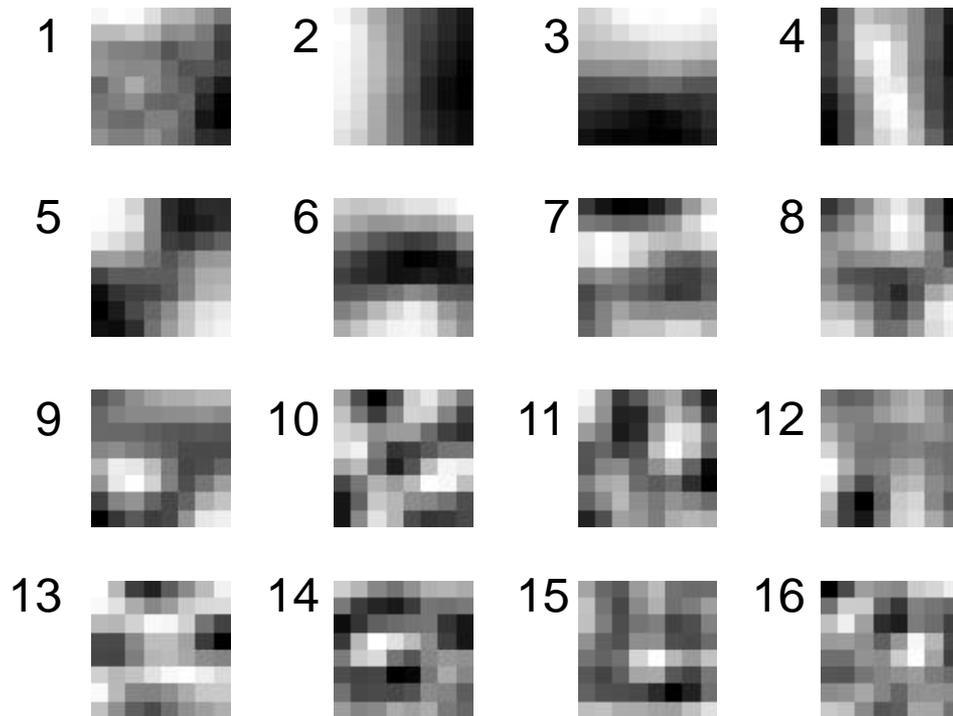
成分数：3  
# of synthesized components: 3



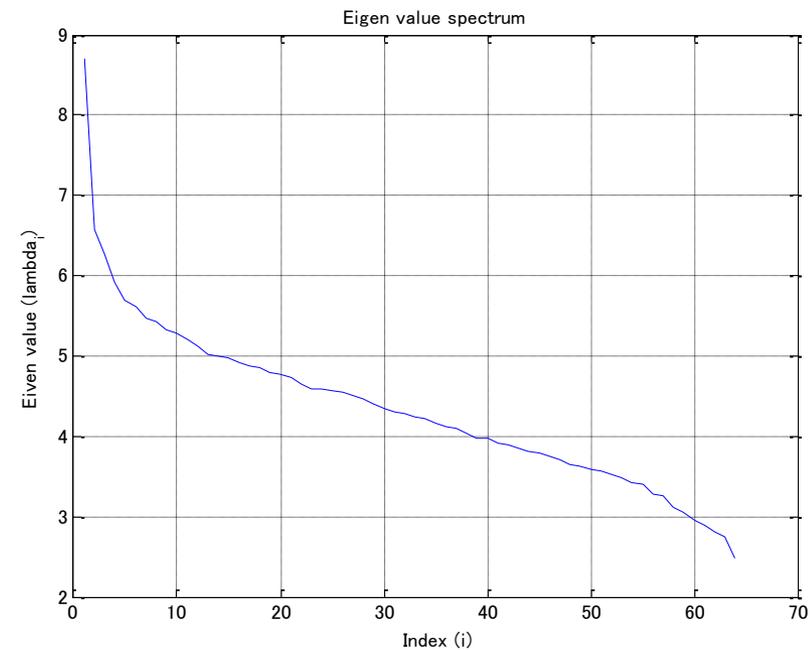
成分数：10

# of synthesized components: 10

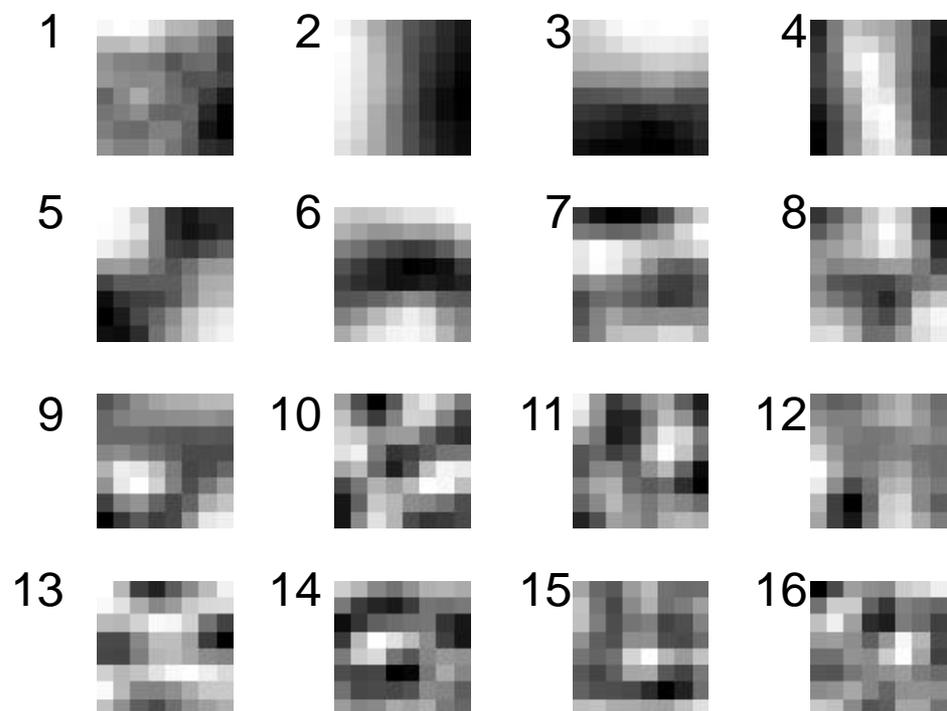
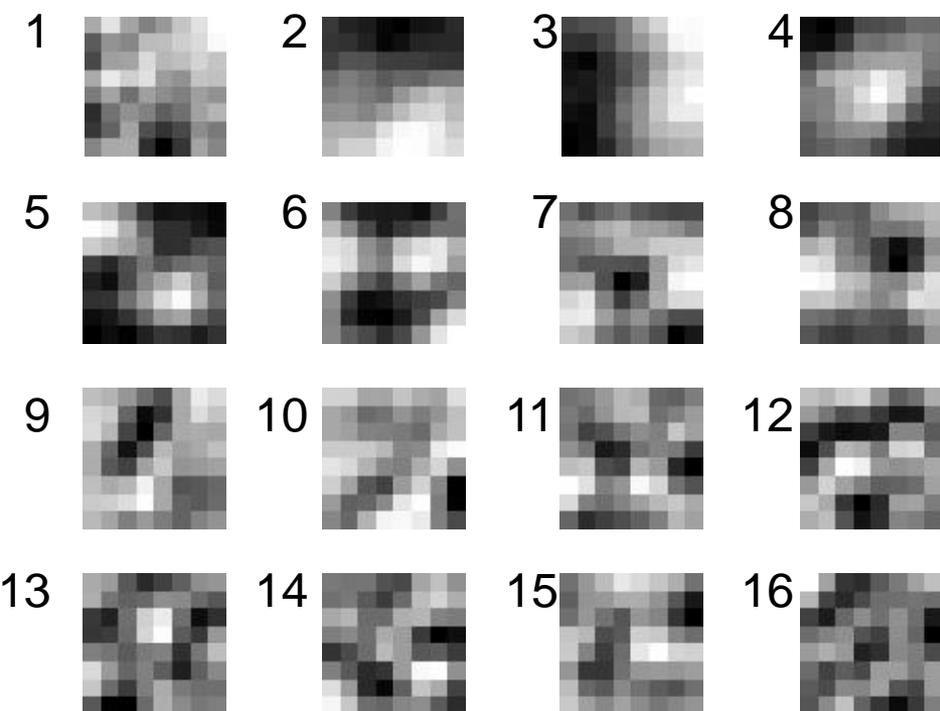




固有値の大きい順に並べた  
最初の16枚の主成分画像



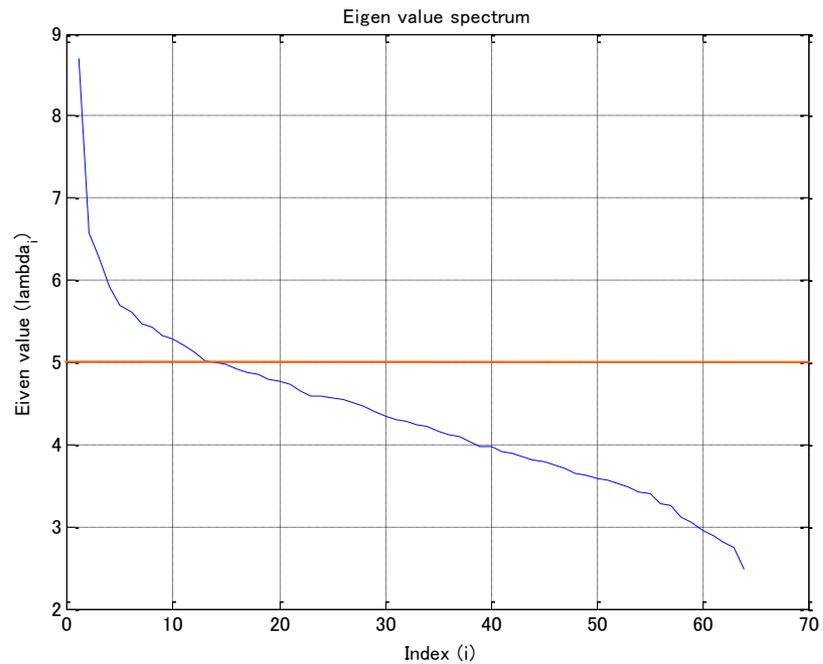
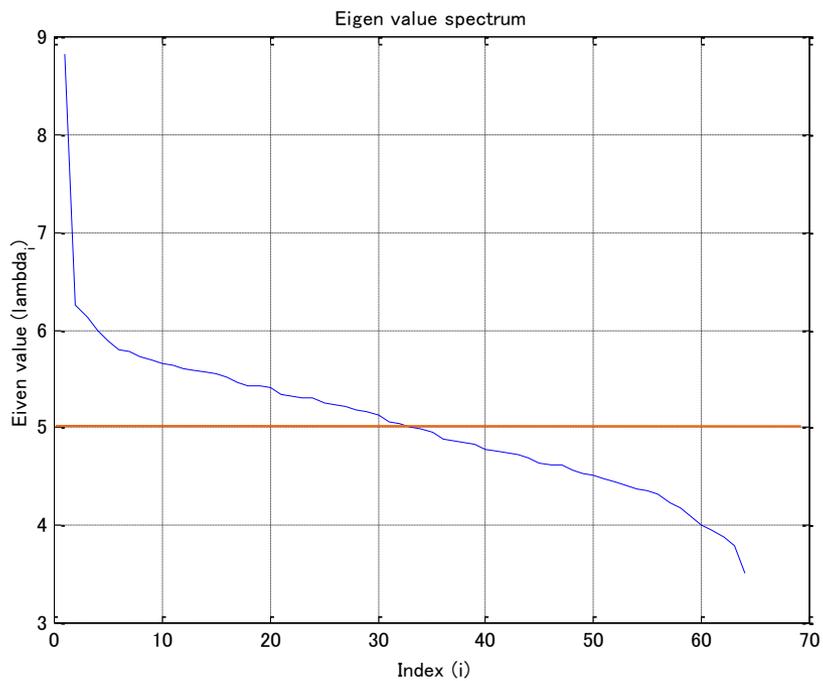
固有値スペクトル



固有値の大きい順に並べた  
最初の16枚の主成分画像

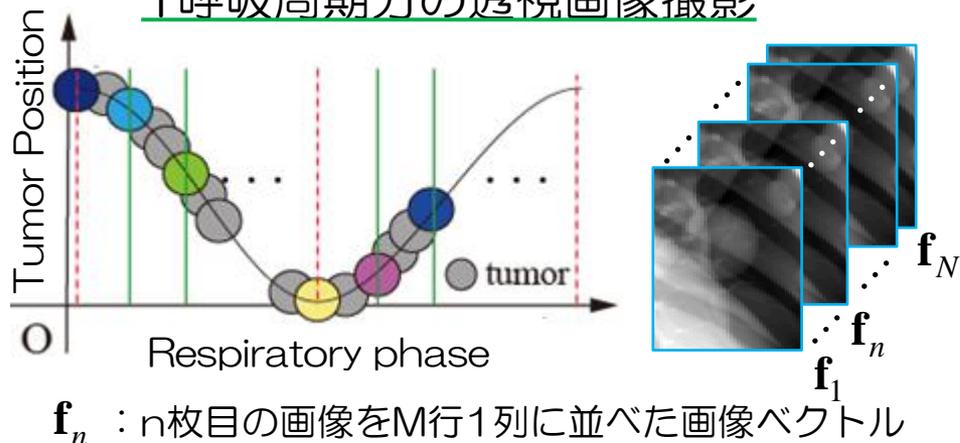
固有値の大きい順に並べた  
最初の16枚の主成分画像

# 固有値スペクトルの比較

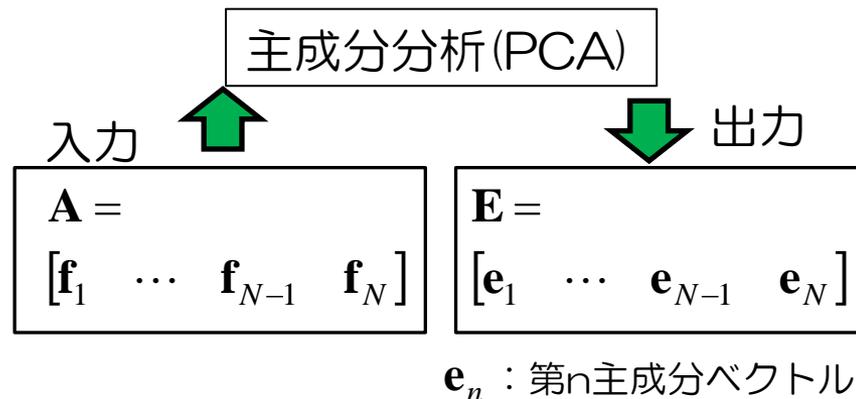


# 腫瘍トラッキング

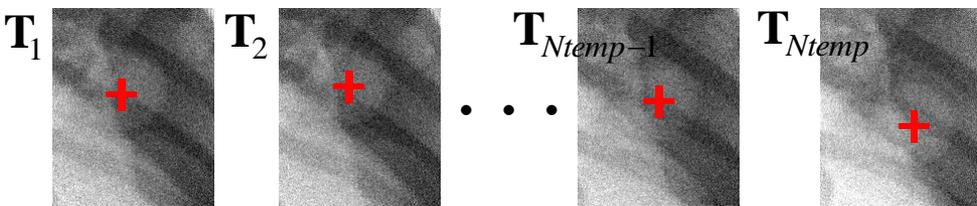
## 1呼吸周期分の透視画像撮影



## 主成分分析



## マルチプルテンプレート



$$\text{CompT}_{ntemp} = \mathbf{K}^T \mathbf{T}_{ntemp}$$

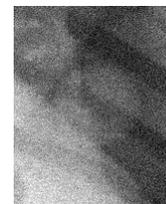
$$\mathbf{K}^T = [\mathbf{e}_1 \ \cdots \ \mathbf{e}_{\text{dim}-1} \ \mathbf{e}_{\text{dim}}]^T \ (\text{dim} < N)$$

$$\text{CompT}_{ntemp} = [t_1 \ \cdots \ t_{\text{dim}-1} \ t_{\text{dim}}]$$

(例)  $\text{CompT}_1$ を視覚化,  $\text{dim} = 50$  (※画像処理済み)

## 腫瘍検出

リアルタイム  
入力画像  
Input



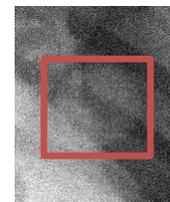
$$\text{CompI} = \mathbf{K}^T \text{Input}$$

$$\mathbf{K}^T = [\mathbf{e}_1 \ \cdots \ \mathbf{e}_{\text{dim}-1} \ \mathbf{e}_{\text{dim}}]^T \ (\text{dim} < N)$$

$$\text{CompI} = [i_1 \ \cdots \ i_{\text{dim}-1} \ i_{\text{dim}}]$$

類似度演算

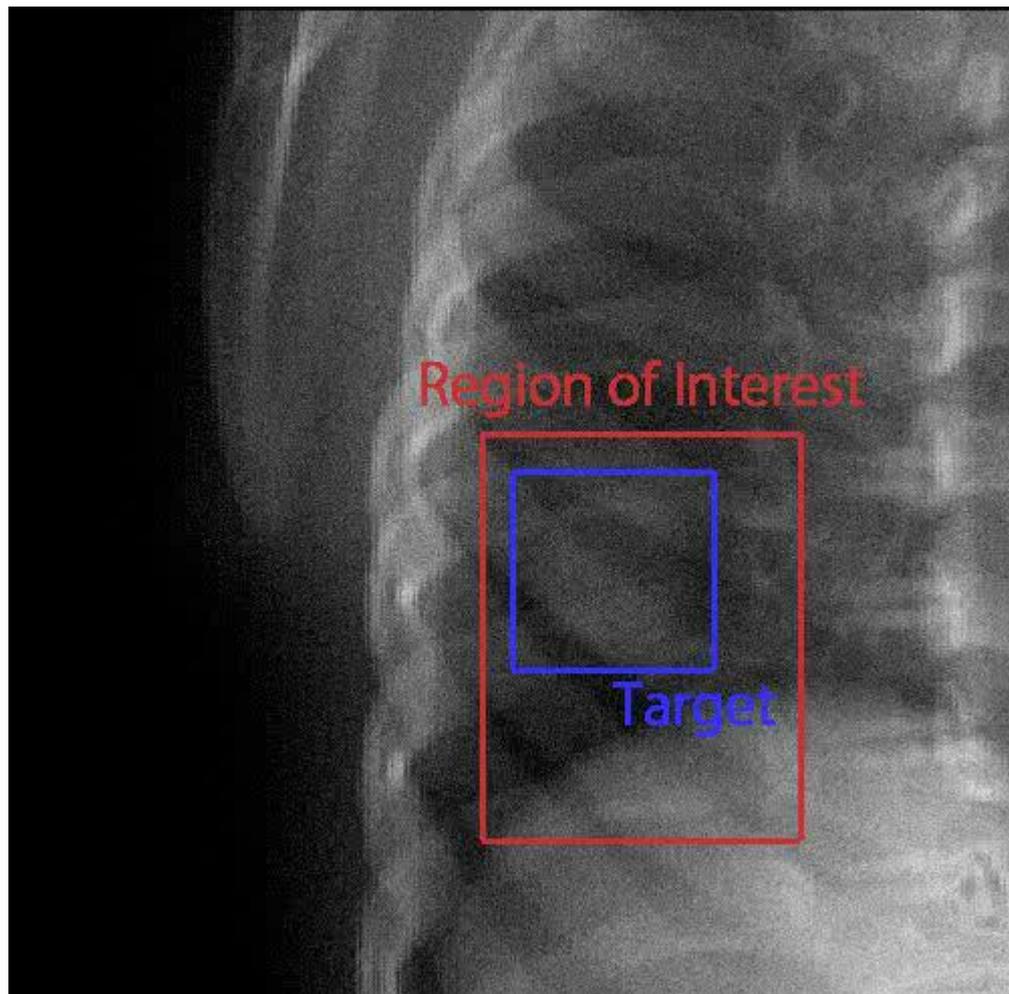
腫瘍位置検出



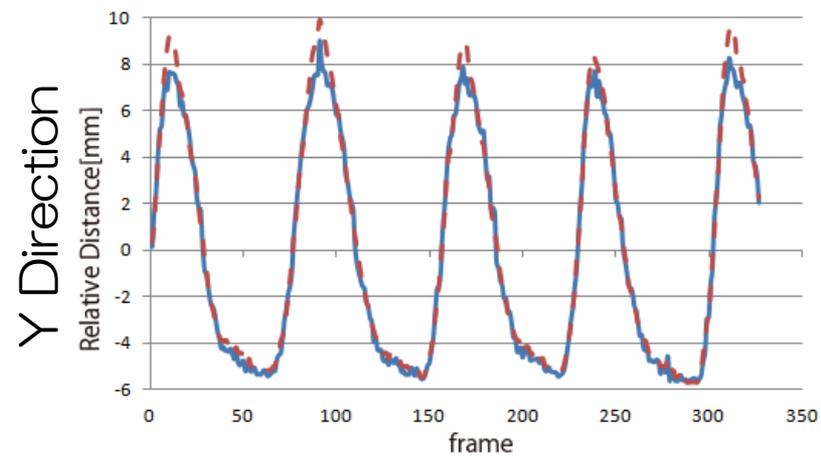
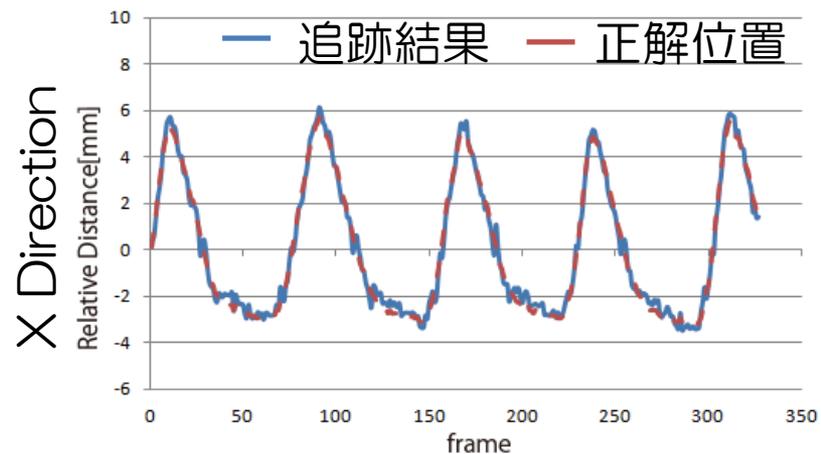
# トラッキング結果

x  
y

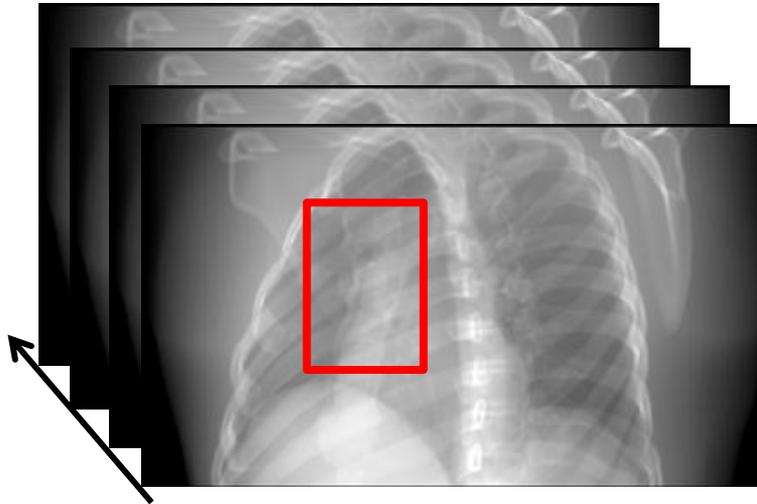
□:トラッキング結果



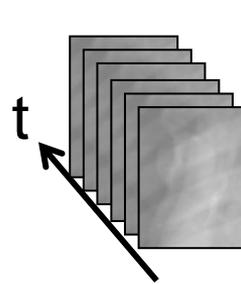
## トラッキング軌跡



## Preparation



Series of the projection image



ROI images

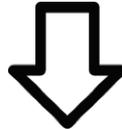
$$f_{learn,t} \triangleq f_{learn}(x, y; t)$$

Vector representation

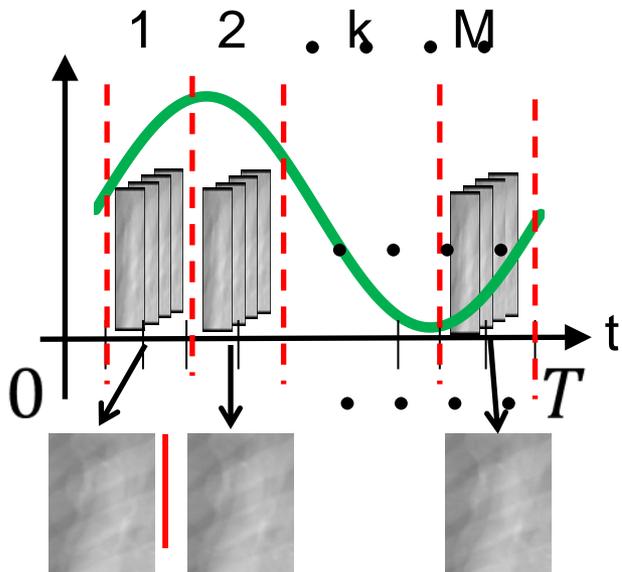
$M$ : Number of bins  
 $T$ : A cycle of breath  
 $t_w$ : duration of a bin  
 $t_w = T/M$

Calculate PCA using

$$\{f_{learn,t} | t = 1 \sim T\}$$


$$E = [e_1 \quad \dots \quad e_L]$$

Prior to radiotherapy, each patient breathes on the treatment table. A series of X-ray projection images are captured during breathing. Then multiple templates are generated.

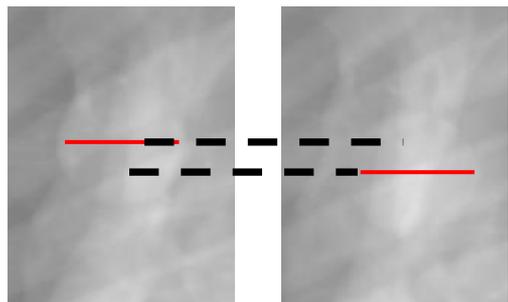


Average the images in  $k$ th respiratory phase bin.

$$\bar{f}_{learn,k} \triangleq \sum_{m=1}^{tw} f_{learn,tw(k-1)+m/tw}$$



Generate more templates by shifting the averaged image by small amounts for each bin:



$$\bar{f}_{learn,k,i,j} \triangleq \bar{f}_{learn,k}(x + i\Delta x, y + j\Delta y)$$

$$i = 1, \dots, I$$

$$j = 1, \dots, J$$

$$\bar{\mathbf{f}}_{learn,k,i,j} \triangleq \bar{\mathbf{f}}_{learn,k}(x + i\Delta x, y + j\Delta y)$$

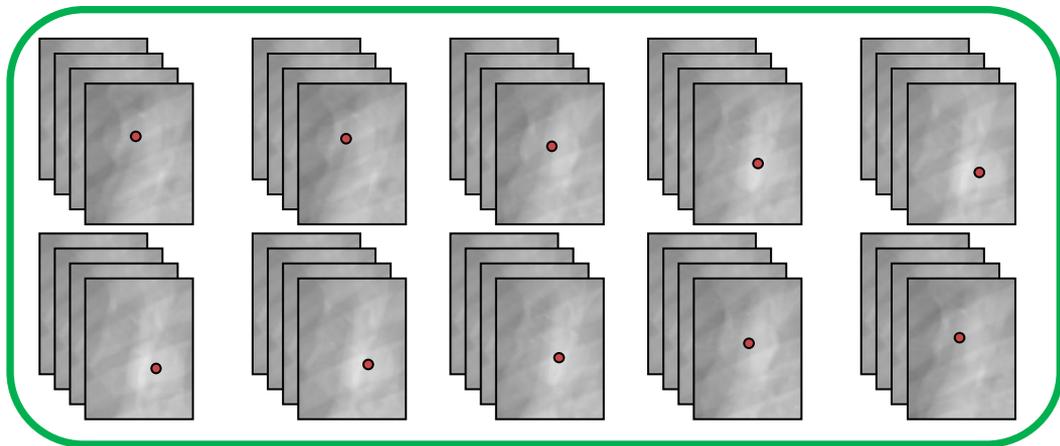
$$i = 1, \dots, I$$

$$j = 1, \dots, J$$



Define the target position for each template

$$\mathbf{r}_{k,i,j} = (x_{k,i,j}, y_{k,i,j})$$



Reducing template image dimension using principal component analysis (PCA)



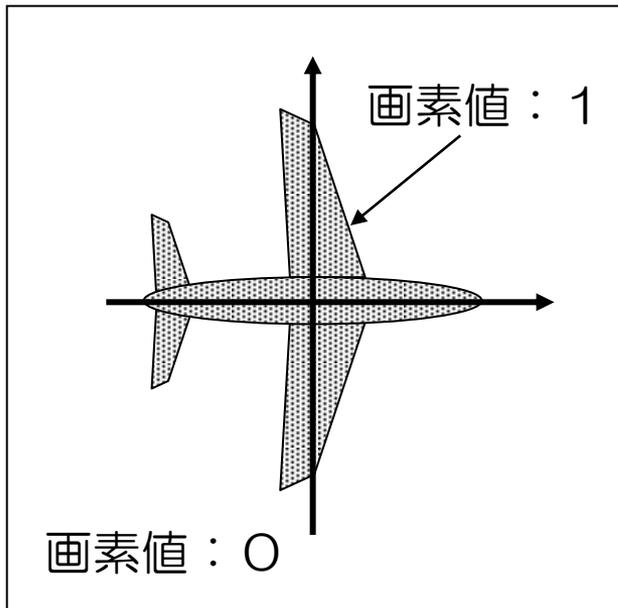
Multiple template

Use  $\{\mathbf{g}_{learn,k,i,j}\}$  as templates

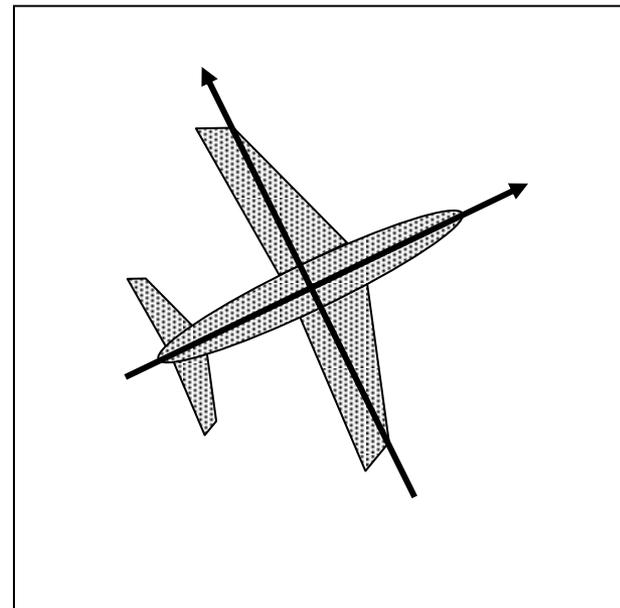
$$\mathbf{g}_{learn,k,i,j} = \mathbf{E}^t \bar{\mathbf{f}}_{learn,k,i,j}$$

$$\mathbf{g}_{learn,k,i,j}$$

2値画像



2値画像



画素値1のプロットの共分散行列から主成分方向とばらつきを算出

画素値1のプロットの共分散行列から主成分方向とばらつきを算出



- 固有値 ( $\lambda_1, \lambda_2$ ) からパターンの一貫性がわかる
- 固有ベクトルの方向から物体の回転が検出できる