1 Global Transforms

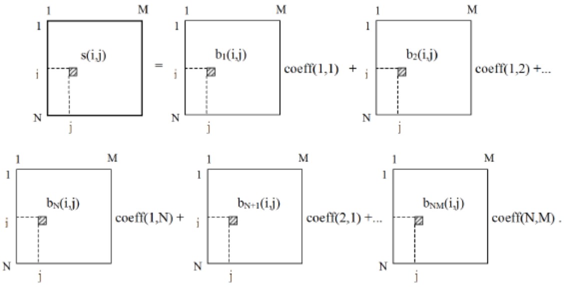
1.1 General Model

Represent an image s (with pixel values s (i,j)) as a weighted sum of basis images

$$b_k$$

with pixel values

$$b_k(i,j).$$



s: spatial domain (pixel domain); coeff: transformed domain (frequency domain)

- Matrix component equation:
  s(i, j) , $coeff_k$, b_k(i,j): scalars, i = 1,. . . , N; j = 1,. . . , M; $L <= NM$
  L should be equal to M x N for a losless transform

$$s(i,j) = \sum_{k=1}^{L} coeff(k) b_k(i,j) \tag{1}$$

- Matrix component equation with the coefficients ordered as matrix:
  s(i, j) , $coeff(k,l)$, b_{kl}(i,j): scalars, i = 1,. . . , N; j = 1,. . . , M;

$$s(i,j) = \sum_{k=1}^{N} \sum_{l=1}^{M} coeff(k,l) b_{kl}(i,j) \tag{2}$$

- forward transform: tool that brings us from the spatial domain to the transformed domain. coefficients used to represent the input
- inverse transform: tool that brings us from the transformed domain to the spatial domain.

1.1.1 Vector Notation

Recipe for matrix to vector conversion:

$$\begin{aligned} s(n,m) &= s(k) \\ k &= m + (n-1)M \end{aligned} \tag{3}$$

inverse:

$$\begin{aligned} s(k) &= s(n,m) \\ m &= mod_M(k) \\ n &= 1 + \lfloor \frac{k}{M} \rfloor \end{aligned} \tag{4}$$

vector/matrix notation:

$$s = (b_1, b_2, \dots, b_L)(coeff_1, coeff_2, \dots, coeff_L)^T = bcoeff \tag{5}$$

dimensions:

$$s : [P, 1], coeff : [L, 1], b_k : [P, 1], b : [P, L] \tag{6}$$

$$s = \begin{bmatrix} s(1) \\ s(2) \\ \vdots \\ s(NM) \end{bmatrix} = \begin{bmatrix} b_1(1) & b_2(1) & \dots & b_L(1) \\ b_1(2) & b_2(2) & \dots & b_L(2) \\ \vdots & \vdots & \ddots & \vdots \\ b_1(P) & b_2(P) & \dots & b_L(P) \end{bmatrix} \begin{bmatrix} coeff_1 \\ coeff_2 \\ \vdots \\ coeff_L \end{bmatrix} \tag{7}$$

1.1.2 Questions

- How to choose the basis images?
  Every transform has its own basis images. The basis images are chosen to be orthogonal to each other.
- How to choose the value of L?
  L should be equal to M x N for a losless transform
- What are the physicical properties of the coefficients?
  (their correlation, their meaning?)
- Numerical problems?
  - calculation of basis images
  - calculation of coefficients for given image
  - storage and speed requirements?
- Under what conditions is the transform invertible?
  L should be equal to M x N for a losless transform
- What can we do with such transforms?

1.1.3 Physical meaning of the expansion in basis images

$$s = (b_1, b_2, \dots, b_L)(coeff_1, coeff_2, \dots, coeff_L)^T = b \times coeff \tag{8}$$

example: let s be a 3D vector

$$s = (s_1, s_2, s_3)^T \tag{9}$$

Projection s' of s on a vector b_1 is given by:

$$s' = b_1 \times coeff_1 \tag{10}$$

($coeff_1$ = projection coefficient)
example in 3D: Projection s' of s on a pkane defined by two vectors b_1 and b_2 is given by:

$$s' = b_1 \times coeff_1 + b_2 \times coeff_2 \tag{11}$$

($coeff_1$ and $coeff_2$ = projection coefficients)
to find the coefficients: if b_1 and b_2 are orthogonal, then:

$$coeff_i = b_i^T \times s' \tag{12}$$

which is scalar product of b_i and s' symbol of scalar product operator: $\cdot$ To recover s from s': we need a number of linearly independent vectors equal to the dimension of s.

1.1.4 How to find the coefficients?

Easiest way: minimize the least square error between the original image and the projected image.

$$\frac{\partial SE}{\partial coeff} = 0 \tag{13}$$

$$\frac{\partial SE}{\partial coeff} = \begin{bmatrix} \frac{\partial SE}{\partial coeff_1} \\ \frac{\partial SE}{\partial coeff_2} \\ \vdots \\ \frac{\partial SE}{\partial coeff_L} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{14}$$

Set of L linear equations. SE:

$$s^T s - s^t bcoeff - coeff^T b^T s + coeff^T b^T bcoeff \tag{15}$$

in general:

$$\frac{\partial(Acoeff)}{\partial coeff} = A^T \tag{16}$$

and

$$\frac{\partial(coeff^T A)}{\partial coeff} = A \tag{17}$$

thus:

$$\frac{\partial SE}{\partial coeff} = 2b^T bcoeff - 2b^T s \tag{18}$$

is zero at the solution:

$$\begin{aligned} (b^T b)coeff &= b^T s \\ coeff &= (b^T b)^{-1} b^T s \end{aligned} \tag{19}$$

$(b^T b)^{-1} b^T$ is called the pseudo-inverse of b.
incase vectors bk are orthonormal, then:

$$coeff = b^T s \tag{20}$$

1.1.5 Specializations of the general model: Separable transforms

$$s(i,j) = \sum_{k=1}^{N} \sum_{l=1}^{M} coeff(k,l) b_{kl}(i,j) \tag{21}$$

with:

$$\begin{aligned} b_{kl}(i,j) &= b_k(i) c_l(j) \quad \text{separable} \\ b_{kl}(i,j) &= b_k(i) b_l(j) \quad \text{symmetric separable} \end{aligned} \tag{22}$$

dimensions:

- s(i,j) coeff(k,l) b_{kl}(i,j) : scalars
- i = 1, ..., N; j = 1, ..., M;

$$
\begin{aligned}
s(i,j) &= \sum_{k=1}^{N}\sum_{l=1}^{M} coeff(k,l) b_k(i) c_l(j) \\
. &= \sum_{k=1}^{N}\left(b_k(i)\sum_{l=1}^{M} coeff(k,l) c_l(j)\right) \\
. &= \sum_{k=1}^{N} b_k(i) z(k,j)
\end{aligned} \tag{23}
$$

s = bz; z = coeff c^T
s = bcoeff c^T
write forward transform as:

$$coeff = a^T sd \tag{24}$$

write inverse transform as:

$$s = bcoeffc^T \tag{25}$$

From these two it follows that the forward transform followed by the inverse transform yields:

$$s = ba^T sdc^T = \left(ba^T\right) s \left(dc^T\right) \tag{26}$$

the perfect reconstruction condition requires that:

$$b = (a^T)^{-1} \quad \text{and} \quad c = (d^T)^{-1} \tag{27}$$

in this case, the direct (forward) transform is given by:

$$coeff = a^T sd = (b)^{-1} s (c^T)^{-1} \tag{28}$$

if b $\neq (a^T)^{-1}$ and d $\neq (c^T)^{-1}$, then the forward-inverse transformation yields an approximation of s:

$$s' = ba^T sdc \approxeq s \tag{29}$$

orthonormal case: in case of Orthonormality:

a = b and d = c; also $b^T = b^{-1}$ and $c^T = c^{-1}$

The direct transform is given by:

$$coeff = b^T sc = b^{-1} sc \tag{30}$$

1.2 Discrete Karhunen Loeve Transform (KLT)