Image Transform: DFT, DCT, DWT

- Processing images in the frequence domain, or called transform domain

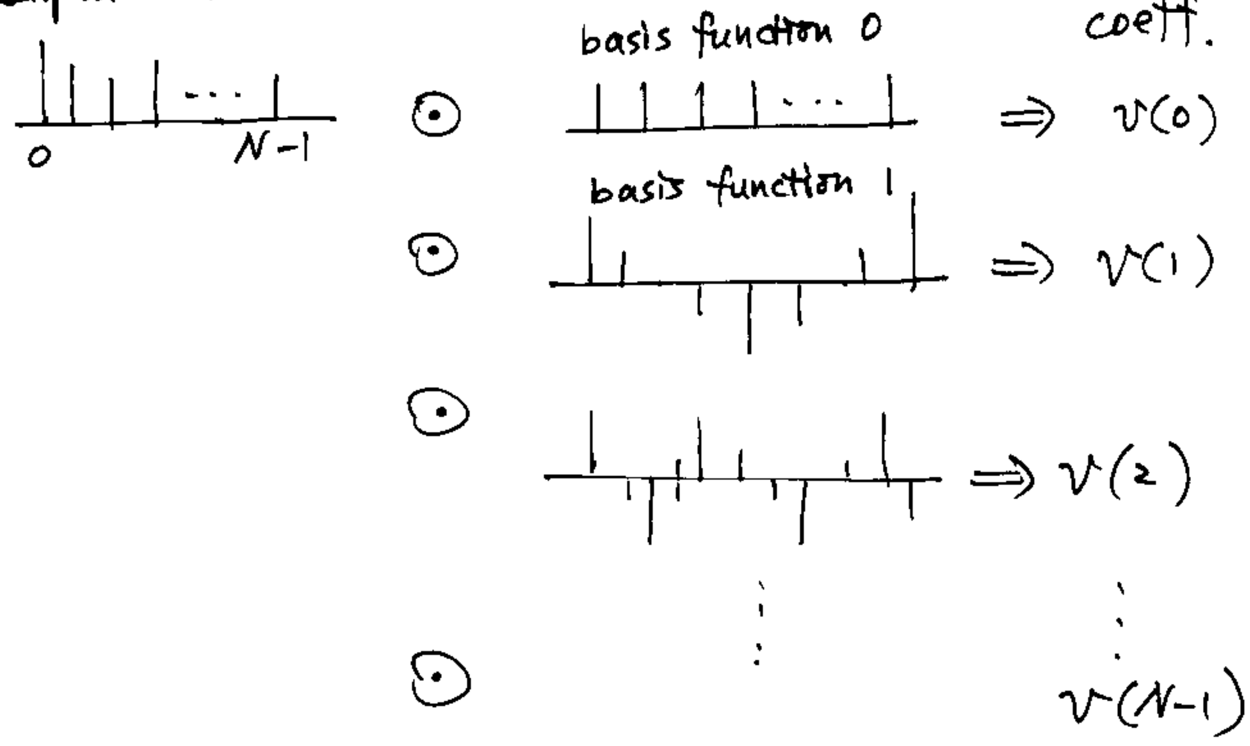
$$Im \rightarrow \boxed{Tx} \rightarrow \boxed{Process} \rightarrow \boxed{\text{Inverse } Tx} \rightarrow Im'$$

Reasons:

- A lot of transform coefficients are small and can be removed. $\Rightarrow$ save space
- Processing can be done efficiently in the transform domain. Number of coefficients is small.
  Fast transforms exist.

<u>Intuition of the transform process:</u>

Input $u(n)$ $n: 0, \ldots, N-1$



Inner Product
or "Project"

basis function

$b_0(n)$ — will have large response value when $u(n)$ is smooth

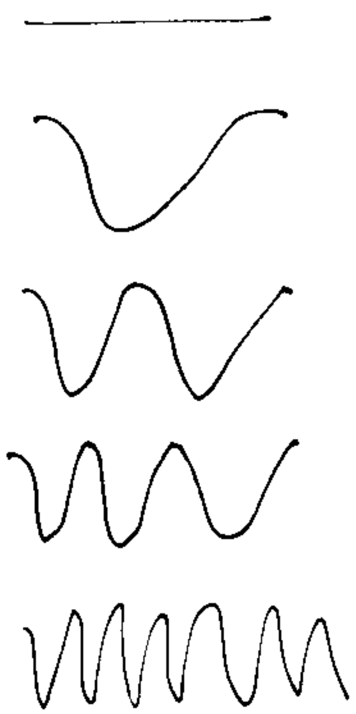
$b_1(n)$ — have large response value when $u(n)$ ~~is~~ has some variation

$b_2(n)$

$b_3(n)$ — large response when $u(n)$ has high variation

$\vdots$

$b_{N-1}(n)$



* Reconstruction & superposition

$$\vec{u} = \sum_{k=0}^{N-1} v(k)\,\vec{b}_k \qquad \text{(superposition of basis functions)}$$

$$b_k(n) = \frac{1}{\sqrt{N}}\left(\cos\frac{2\pi kn}{N} + j\sin\frac{2\pi kn}{N}\right)$$

$$= \frac{1}{\sqrt{N}}\left(e^{j\left(\frac{2\pi}{N}\right)kn}\right)$$

$$= \frac{1}{\sqrt{N}}\left(W_N^{-1}\right)^{kn}$$

where $W_N = e^{-j2\pi/N}$ "N-th complex root of 1"

$$(W_N)^N = 1$$

Represent in the vector-matrix form

$$\vec{u} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix} = \left[\begin{array}{c:c:c} \vec{b_0} \cdot v(0) & \vec{b_1} \cdot v(1) & \cdots \end{array} \right]$$

$$= \left[\begin{array}{c:c:c} \vec{b_0} & \vec{b_1} & \cdots \end{array} \right] \begin{bmatrix} v(0) \\ v(1) \\ \vdots \\ v(N-1) \end{bmatrix}$$

$$= B_{N \times N} \, \vec{v}$$ Backward Transform

$$\vec{v} = A_{N \times N} \, \vec{u}$$ Forward ~~Matrix~~ Transform

Note $A = B^{-1} = B^{*t}$ ($\because$ B is a unitary matrix)

$\cancel{A =}$

<u>Regular Fourier Transform Equations</u>

$$v(k) = \frac{1}{\sqrt{N}} \left(\sum_{n=0}^{N-1} u(n) \, W_N^{kn} \right)$$

where $W_N = e^{-j 2\pi / N}$

$$u(n) = \frac{1}{\sqrt{N}} \left(\sum_{k=0}^{N-1} v(k) \, W_N^{-kn} \right)$$