Image Transforms

February 13, 2019

Image Transforms-Need

Transform from spatial domain to another domain and back
What is the need of this kind of transformation?

- Filtering
  - Convert from spatial domain to frequency domain
  - Filter in the frequency domain
  - Convert back to spatial doman
- Data Compression
  - For storage space
  - For transmission, particularly realtime
- Feature extraction-edge pixels, corners etc.

Mathematical Preliminaries-Unitary Matrices

Definition

A matrix A is said to be unitary if its inverse is the complex conjugate of its transpose, i.e., $AA^{\mathsf{T}*} = \mathrm{I}$, where I is the unit matrix.

In this case, every column of vector is orthogonal to conjugate of every column, except itself.

Sometimes $A^{\mathsf{T}*}$ is also written as A^{H}.

If the elements of A are real, then unitary matrix becomes same as orthogonal matrix.

In this case, every column of vector is orthogonal to every column of vector except itself.

Image Transforms

We had defined an image transform by the relation $g = h_c^T f h_r$.
If h_c and h_r are chosen to be unitary, then $h_c h_c^{T*} = I$
$\Rightarrow f = (h_c^T)^{-1} g h_r^{-1}$
In a unitary matrix, the columns form a set of orthonormal vectors.

Elementary Image or Basis image

If f is an image then using a separable transformation H with individual transformations being h_c, h_r, f is transformed to g by the formula $g = h_c^T f h_r$, and hence, $f = (h_c^T)^{-1} g (h_r)^{-1}$

Suppose we partition the matrices $(h_c^T)^{-1}$ and h_r^{-1} as follows:
$(h_c^T)^{-1} = (u_1|u_2|\cdots|u_N)$,

$$h_r^{-1} = \begin{pmatrix} v_1^T \\ -- \\ v_2^T \\ -- \\ \vdots \\ -- \\ v_N^T \end{pmatrix}$$

Elementary Image or Basis Images

Then, $f = (u_1|u_2|\cdots|u_N)g\begin{pmatrix} v_1^T \\ -- \\ v_2^T \\ -- \\ \vdots \\ -- \\ v_N^T \end{pmatrix}$

Since g is an $N \times N$ matrix, it can be written as the sum of N^2 matrices, each matrix having only one non-zero element as

$$g = \begin{pmatrix} g_{11} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} + \begin{pmatrix} 0 & g_{12} & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} + \cdots + \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & g_{NN} \end{pmatrix}$$

Basis images

The previous equation becomes $f = \sum_{i=1}^{N} \sum_{j=1}^{N} g_{ij} u_i v_j^T$
Thus the image f has been expressed as the sum of vector outer products. A vector outer product may be thought of as an "image". Thus this equation gives us an image f as the sum of "some basis images" appropriately weighted by the g_{ij} coefficients.

We can write this transformation equation as $f = UgV^H$
Another consideration which is kept in mind is to choose the matrices U and V as unitary - so that inversion becomes much easier. In such a situation, the above expression which transforms an image f into g is called a UNITARY TRANSFORM of image f.

Image Transforms

How do we choose h_r and h_c matrices?
We may choose them so that the transformed image g can be represented by lesser number of bits.
If we want to represent the image f with fewer than N^2 number of elements, we may choose matrices U and V so that the transformed image g is a diagonal matrix. Then we could represent image f using only the N nonzero elements of g.
This can be achieved with a process called Matrix Diagonalixation and it is called Singular Value Decompostion (SVD) of the image.

Image Transforms-SVD

What is Matrix Diagonalisation?
Determining two matrices U and V such that $UAV^T = I$, where I is diagonal.

When we talk of diagonal, obviously, A has to be square. However, not all images are square. In fact more often than not, the images are rectangular. If a matrix is square and symmetric, it can be diagonalised. If it is not symmetric but has a full set of eigenvectors, then also it is diagonalisable. However, for a general square matrix, we cannot be sure that we can diagonalise.
When the matrix is rectangular, question of diagonalisation does not arise.

Singular Value Decomposition

To bring the general matrix $A_{m\times n}$ as close to a diagonalisation as possible, we use the technique of Singular Value Decomposition. First the result - Form the square matrix $A^T A$. $A^T A$ is always symmetric. Suppose the rank of this matrix is r, then

$$g = U\Lambda V^T$$

where U and V are orthogonal matrices of sizes $m \times m$ and $n \times n$, while Λ is a matrix of size $m \times n$, where for the first r rows, only in the elements in the diagonal positions are non-zero and in fact equal to square root of the r non-zero eigenvalues of $A^T A$.

That is $U = \begin{pmatrix} u_1 & u_2 & \cdots & u_m \end{pmatrix}$, $V = \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix}$ and

$$\Lambda = \begin{pmatrix} \sigma_1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & \cdots & \cdots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \cdots & \sigma_r & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{pmatrix}$$

Singular Value Decomposition

Here σ 's are called the singular values of A given by $\sigma_i = \sqrt{\lambda_i}$, where λ_i is an eigenvalue of $A^\top A$. There will be r non-zero eigenvalues where r is the rank A.

LDCOLL Show LDCollege lecture

SVD Problem

Compute the eigenvalues and eigenvectors of $A^T A$ and AA^T

when $A = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

$A^T A = \begin{pmatrix} 5 & 2 & 2 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \end{pmatrix}$

Equation for eigenvalues is

$$\begin{vmatrix} 5-\lambda & 2 & 2 \\ 2 & 1-\lambda & 1 \\ 2 & 1 & 2-\lambda \end{vmatrix} = 0$$

Roots are $\lambda_1 = 6.854, \lambda_2 = 1, \lambda_2 = 0.1459$

SVD Problem

Corresponding eigenvectors normalised are

$$\lambda_1 = 6.854, v_1 = \begin{pmatrix} 0.8156 \\ -0.3568 \\ 0.4178 \end{pmatrix} \lambda_2 = 1, v_2 = \begin{pmatrix} 0.447 \\ 0 \\ -0.894 \end{pmatrix}, \lambda_3 = 0.1459, v_3 = \begin{pmatrix} 0.319 \\ -0.934 \\ 0.160 \end{pmatrix}$$

SVD Problem

Now $A^\top A$ and $AA^\top$ will have same eigen values.
Reason: $A^\top Av = \lambda v \Rightarrow A(A^\top A)v = \lambda ASVv \Rightarrow (AA^\top)Av = \lambda Av$
This means λ is an eigenvalue of $AA^\top$ and the corresponding eigenvector is AV, where λ, v are eigenvalue and eigenvector $A^\top A$. Using this relation, we can find the eigenvalues and eigenvectors of $AA^\top$. Of course eigenvalues are same.

Error in SVD omitting few terms

We have seen that if A has rank r, then A can be expressed as

$$A = \sum_1^r \sigma_i u_i v_i^T$$

Hence, $A_{mn} = \sum_{k=1}^r \sigma_k u_{ik} v_{ik}$
If we retain only the first k eigenvalues, then the error in reconstruction= norm of the difference matrix (original - reconstructed)
Difference Matrix $D_{mn} = \sum_{i=k+1}^r \sigma_i u_{im} v_{in}$
Hence, $||D||= D_{mn}^2 = (\sum_{i=k+1}^r \sigma_i u_{im} v_{in})^2$
$= \sum_{i=k+1}^r \sigma_i^2 u_{im}^2 v_{in}^2 +$
$2 \sum_m \sum_n \sum_{i=k+1}^r \sum_{j=k+1;j\neq i}^r \sigma_i \sigma_j u_{im} j_{in} v_{jm} v_{jn}$
$= \sum_{i=k+1}^r \sigma_i^2 \sum_m u_{im}^2 \sum_n v_{in}^2 +$
$2 \sum_{i=k+1}^r \sum_{j=k+1;j\neq i}^r \sigma_i \sigma_j \sum_m u_{im} u_{jm} \sum_n v_{in} v_{jn}$

SVD error in reconstructing the image

But, u_i's and v_i's are orthogonal and hence,
$\sum_m u_{mi}^2 = \sum_n v_{in}^2 = 1$
and $\sum_n v_{in} v_{jn} = \sum_m u_{im} u_{jm} = 0$ for $i \neq j$
Hence, $||D|| = \sum_{i=k+1}^{r} \sigma_i^2$

Reconstruction of Image from SVD

SVD gives $f = \sum_1^r \sigma_i u_i v_i^T$.
If the σ's are arranged in descending order of magnitude, then discarding some of the last few terms in the above summation will give an approximation to the original image. This approximation is the best in the least square sense. The matrices $u_i v_i^T$ are called the eigenimages and are determined from the original image itself. MAJOR DIFFERENCE FROM OTHER TRANSFORMS, WHERE THE BASE IMAGES ARE FIXED.

SV decomposition of an image

In the problem of the 3×3 matrix done earlier, suppose we retain only the first term. Find the reconstructed image and also calculate the error. Show that the norm of the error image equals the sum of the squares of the discarded singular values

Sol: $A = \sigma_1 u_1 v_1^T = \sqrt{6.85} \begin{pmatrix} 0.319 \\ 0.934 \\ 0.160 \end{pmatrix} \begin{pmatrix} 0.835 & 0.357 & 0.418 \end{pmatrix}$

$= \mathrm{Rec}_F = \begin{pmatrix} 0.697 & 0.298 & 0.349 \\ 2.041 & 0.873 & 1.022 \\ 0.350 & 0.150 & 0.175) \end{pmatrix}$

The error of this reconstruction is

$f - \mathrm{Rec}_F = \begin{pmatrix} -.303 & -0.298 & -0.349 \\ -0.041 & 0.127 & -0.022 \\ -0.350 & -0.150 & 0.825 \end{pmatrix}$

Hence the norm of this matrix =sum of squares of elements = $1.146 = \sigma_2^2 + \sigma_3^2$

SVD reconstruction of images

Perform SVD of the image given by $A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

Identify the eigenimages of the above image.

Practical Problem using SVD

Consider the following matrix:

$$A = \begin{pmatrix} 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 100 & 100 & 100 & 255 & 255 \\ 255 & 255 & 100 & 150 & 150 & 150 & 100 & 255 \\ 255 & 255 & 100 & 150 & 200 & 150 & 100 & 255 \\ 255 & 255 & 100 & 150 & 150 & 150 & 100 & 255 \\ 255 & 255 & 255 & 100 & 100 & 100 & 255 & 255 \\ 255 & 255 & 255 & 255 & 50 & 255 & 255 & 255 \\ 50 & 50 & 50 & 50 & 255 & 255 & 255 & 255 \end{pmatrix}$$

Practical problem using SVD

Eigenvalues of $A^{\mathsf{T}}A$ are
$2593416.5, 111621.508, 71738.313, 34790.875,$
$11882.712, 0.009, 0.001, 0.000$
Since the last three eigenvalues are negligible, we write down the first five eigenvectors v_i

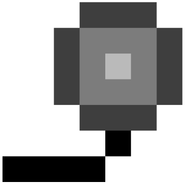
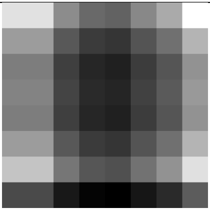
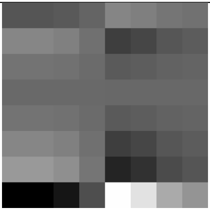
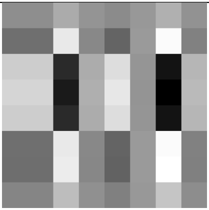
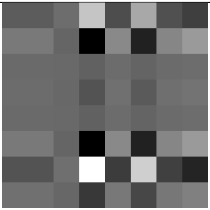
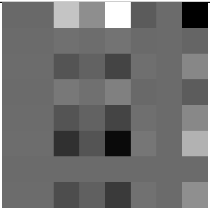
$$\begin{pmatrix}0.410\\0.410\\0.316\\0.277\\0.269\\0.311\\0.349\\0.443\end{pmatrix}\begin{pmatrix}0.389\\0.389\\0.308\\0.100\\-0.555\\-0.449\\-0.241\\-0.160\end{pmatrix}\begin{pmatrix}0.264\\0.264\\-0.537\\0.101\\0.341\\-0.014\\-0.651\\0.149\end{pmatrix}\begin{pmatrix}0.106\\0.106\\-0.029\\-0.727\\0.220\\-0.497\\0.200\\0.336\end{pmatrix}\begin{pmatrix}-0.012\\-0.012\\0.408\\0.158\\0.675\\-0.323\\-0.074\\-0.493\end{pmatrix}$$

Practical problem using SVD

u_i's are given by

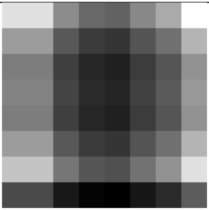
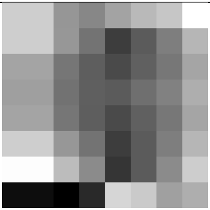
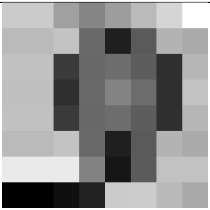
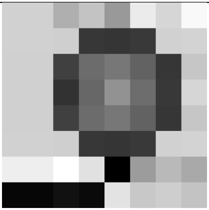
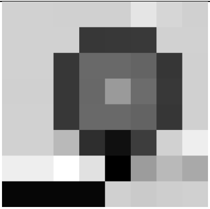
$$\begin{pmatrix}0.441\\0.359\\0.321\\0.329\\0.321\\0.359\\0.407\\0.261\end{pmatrix}\begin{pmatrix}-0.167\\0.252\\0.086\\0.003\\0.086\\0.252\\0.173\\-0.895\end{pmatrix}\begin{pmatrix}-0.080\\-0.328\\0.440\\0.503\\0.440\\-0.328\\-0.341\\-0.150\end{pmatrix}\begin{pmatrix}-0.388\\0.446\\0.034\\0.093\\0.035\\0.446\\-0.630\\0.209\end{pmatrix}\begin{pmatrix}0.764\\0.040\\-0.201\\0.107\\-0.202\\0.040\\-0.504\\-0.256\end{pmatrix}$$

Original Image and EigenImages

		
Original Image	Singular Value No 1	Singular Value No 2
		
Singular Value No 3	Singular Value No 4	Singular Value No 5

Original Image and Eigen Images

Reconstructed Images

		
Singular Value No 1	Singular Value No 2	Singular Value 3
		
Singular Value No 4	Singular Value No 5	

Progressively reconstructed Images

Face Recognition Using SVD

Problem: Given a set of images of faces. This forms the database. Given a new image, how to identify the new face with one of the database faces, or say that it is not in the database.
Idea is to find a basis for the database faces.
Suppose we have p face images, each of size $k = m \times n$. Let us call the faces as $Face_i, i = 1, 2, ..., p$
We convert each of these faces into one vector of length $m \times n$ by stacking one column below the other. Let these vectors be called $f_i, i = 1, 2, ..., p$.
Form a matrix A with these f_i's as columns. Hence, A is a matrix of size $k \times p$.

Face Recoginition using SVD

Perform an SVD operation on A. Remember, SVD can be performed on any matrix, square, rectangular, singular, non-singular. Hence, this step is justified.
Thus, $A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$
Here, U is an orthogonal matrix of size $k \times k$, V is an orthogonal matrix of size $p \times p$, where r is the rank of A, and Σ is a matrix of size $k \times p$, where in the first r rows, diagonal elements are the singular values σ_i, and the remaining elements are zeroes.

Face Recognition using SVD

Recall from SVD lecture, we know that v_i's form an orthogonal basis of R^p and u_i's are defined as $u_i = \frac{Av_i}{||Av_i||}$ for $i = 1, 2, ..., r$ and the remaining v's and u's are defined to get orthogonal/orthonormal bases for R^k, R^p respectively.
Hence, $u_i, i = 1, 2, ..., r$ form an orthgonal basis for the column space of A. But the columns of A contain the faces of the database. Hence, $u_i, i = 1, 2, ..., r$ form a basis for the face database.
For any face f_i, the projection on these basis vectors will give the coordinates of the particular face. Thus,
$fc_i = f_i.u_i, i = 1, 2, ..., r$.
Coorindates of $fc_i = (f_{i1}, f_{i2}, ..., f_{ir})$;
Given a new image G, we will form a vector g by stacking all the columns one below the other. Take the dot product of g with the basis vectors $u_i, k = 1, 2, ..., r$ and get its coordinates in the Face Space as $(g_1, g_2, ...g_r)$.

Face Recognition using SVD

Find the distance of the new face from each of the faces in the database using metric norm in r dimensional space. The faces closest to the new face is the likely candidate. We can set a threshold and if the minimum distance exceeds this threshold, we can say that the new face does not belong to the database and can in fact be added to the same.
Show the expansion of A in terms of u and v and explain how u comes out as a basis for A.

Outer product and inner product of itwo vectors

Let u and v be two vectors. The inner product of these two vectors is defined as $u.v = \sum_1^N u_i v_i = u^T v$ and is defined if and only if the two vectors are of same length. The result is a scalar quantity.

Outer product of these two vectors is defined as

$$u \otimes v = uv^T = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix}$$

$$= \begin{pmatrix} u_1v_1 & u_1v_2 & \cdots & u_1v_n \\ u_2v_1 & u_2v_2 & \cdots & u_2v_n \\ \cdots & \cdots & \cdots & \cdots \\ u_nv_1 & u_nv_2 & \cdots & u_nv_n \end{pmatrix}$$

Unlike the inner product, outer product of two vectors is a matrix. Again unlike the inner product, for which the vectors should be of same length, for outer product, there is no such restriction.

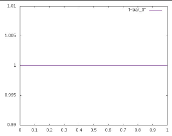
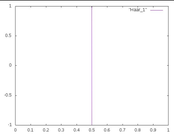
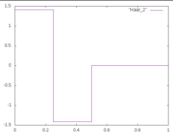
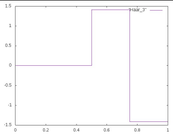
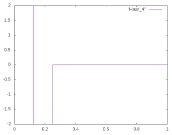
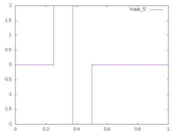
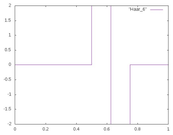
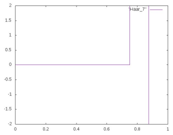
Haar Transform

Haar function defined: Definition recursive:

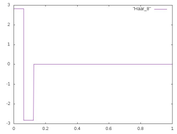
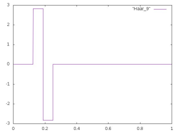
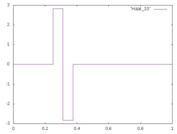
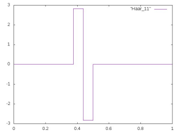
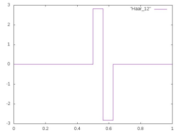
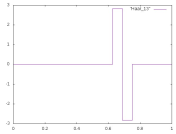
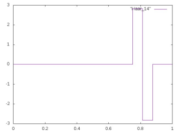
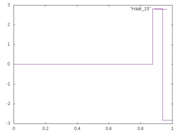
$$
\begin{aligned}
H_0(t) &= 1, 0 \leqslant t < 1 \\
H_1(t) &= 1, 0 \leqslant t < \frac{1}{2} \\
&= -1, \frac{1}{2} \leqslant t < 1 \\
H_{2^p+n} &= \sqrt{2^p}, \frac{n}{2^p} \leqslant t < \frac{n+0.5}{2^p} \\
&= -\sqrt{2^p}, \frac{n+0.5}{2^p} \leqslant t < \frac{n+1}{2^p} \\
&= 0, otherwise
\end{aligned}
$$

where $p = 0, 1, 2, 3, ...$and $n = 0, 1, 2, , , .2^p - 1$

Plots of Haar

			
Haar0	Haar1	Haar2	Haar3
			
Haar4	Haar5	Haar6	Haar7

Plots of Haar

			
Haar8	Haar9	Haar10	Haar11
			
Haar12	Haar13	Haar14	Haar15

Discrete Haar Matrix

To calculate the matrix to evaluate the Haar Transform of a 4×4 image.
We use the definition given above and sample at four points namely, $t = 0, .25, .5, .75$
This gives the matrix

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix}$$

Discrete Haar Transform Matrix

However this matrix is not orthonormal. To make it orthonormal, we multiply by a scale factor so that the columns are orthonormal. The scale factor is $\frac{1}{2}$
Hence the 4×4 Discrete Haar Matrix for transformation of images is

$$\mathrm{H} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix}$$

Haar Transform of an image

Let us consider an image $g = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$

Its Haar Transform is given by $A = HgH^T$

Using the previously derived expression for H, we obtain

$$A = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ \sqrt{2} & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{pmatrix}$$

$$= \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

Haar Transform of an image

If we want to get back the original image g from the its transform A, then the formula is

$$\tilde{g} = H^{-1}A(H^T)^{-1} = H^T A H$$

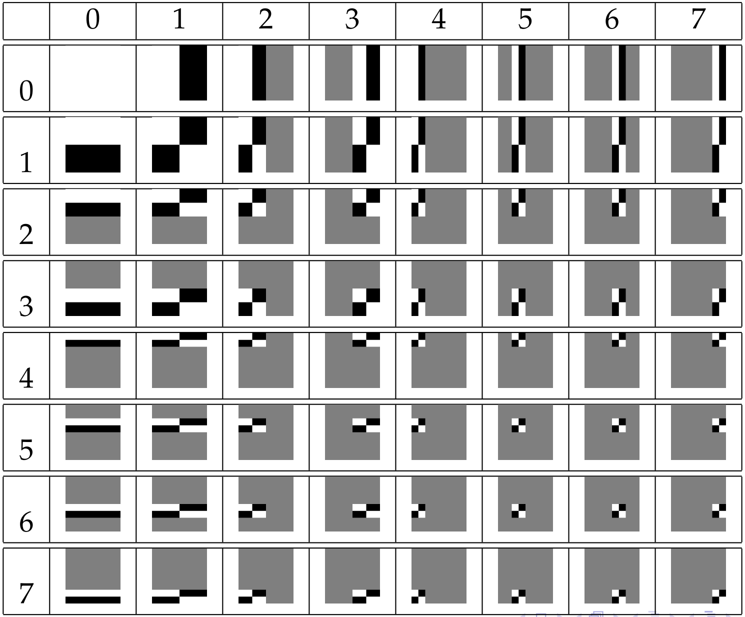
by using orthogonality property of H.
Suppose we use an approximation to the transgform in the form $\tilde{A} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
We have set the bottm right element in the transform as zero.

This when reconsructed gives $\tilde{g} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0.5 & 0.5 \\ 0 & 1 & 0 & 0 \end{pmatrix}$
The error of reconstruction $= 0.5^2 + 0.5^2 + 1^2 = 1.5$.
Note that the error occurs only in the last block of the image - contrast this with Fourier Transforms.

Haar Transform of Images

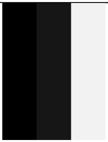
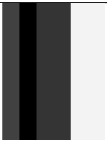
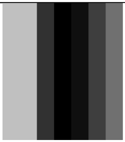
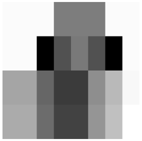
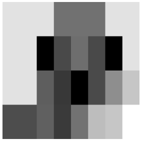
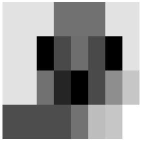
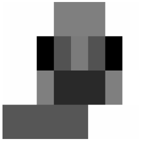
As before, $\tilde{g} = HgH^T$
If g is of $n \times n$ matrix then it can be written as the sum of n^2 matrices of same size, with each matrix having just one element non-zero, all other entries being zeroes.
That is, let $g = \sum_{i=1}^{n} \sum_{j=1}^{n} G^{ij}$, where $G^{ij}_{kl} = \delta_{ik}\delta_{jl}g_{kl}$
Then $\tilde{g}$ will be the sum of n^2 terms, each term being the product of one element and the outer product of one column of H and one row of H^T.
These outer products of columns of H and rows of H^T are called basis images. Since rows of H^T are columns of H, the basis images are the outer products of columns of H with themselves.

Basis images of Haar Transform

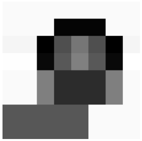
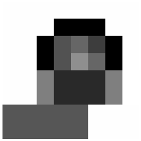
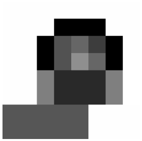
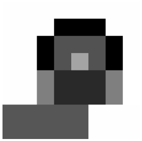
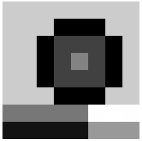
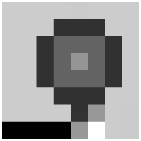
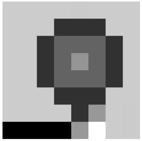
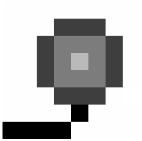


8 × 8 Haar basis images

Reconstructed image with Haar Transform

			
0_0	0_3	0_5	0_7
			
3_0	3_3	3_5	3_7

Flower reconstructed by Haar

			
5_0	5_3	5_5	5_7
			
7_0	7_3	7_5	7_7

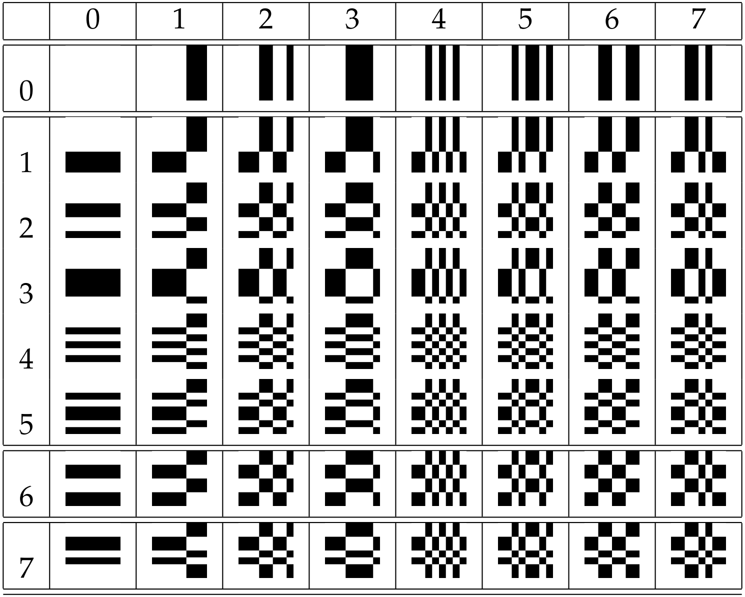
Walsh Transform

There are so many ways of defining Walsh Transform. We shall define recursively.

$$W_0(t) = \begin{cases} 1 & \text{if } 0 \leqslant t \geqslant 0 \\ 0 & \text{otherwise} \end{cases}$$

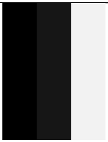
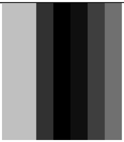
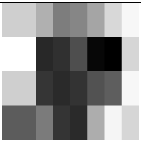
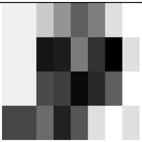
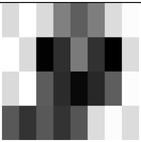
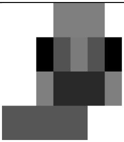
$$W_{2j+q}(t) = (-1)^{\lfloor \frac{j}{2} \rfloor + q}\{W_j(2t) + (-1)^{j+q} W_j(2t-1)\}, \quad j = 0, 1, 2, ..., q = 0, 1$$

Basis images of Walsh Transform

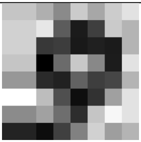
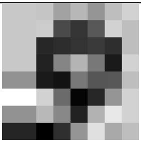
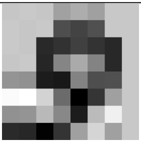
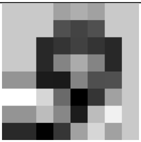
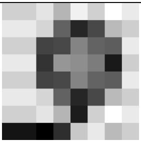
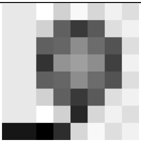
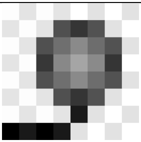


8 × 8 Walsh basis images

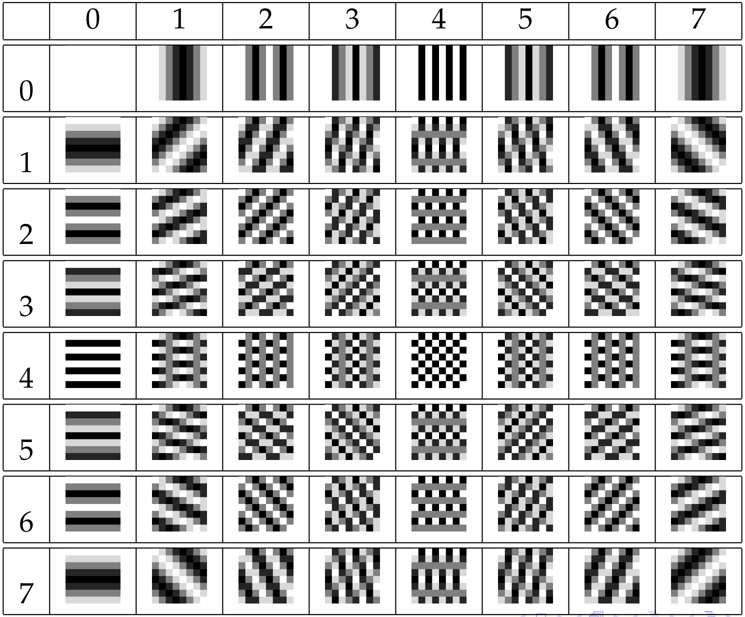
Flower Reconstructed by Walsh

			
0_0	0_3	0_5	0_7
			
3_0	3_3	3_5	3_7

Flower Reconstructed by Walsh

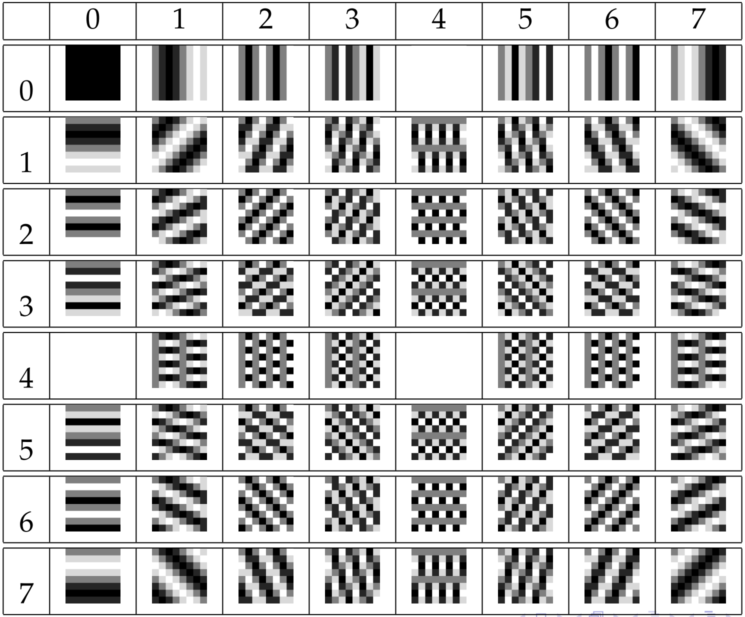
			
5_0	5_3	5_5	5_7
			
7_0	7_3	7_5	7_7

Basis images of Real Fourier Transform



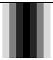
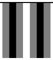
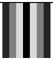
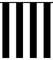
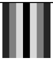
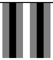
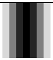
8 × 8 Real Fourier basis images

Basis images of Imag Fourier Transform



8 × 8 Imaginary Part Fourier basis images

FT Real components of Flower

	0	1	2	3	4	5	6	7
0								

8×8

Real Part Fourier Component of Flower images

General Properties of Image Transforms

Image Transforms represents a class of UNITARY MATRICES used for representing an image
What is a UNITARY MATRIX?
A matrix $A_{n \times n}$ is said to be inverse if $A^{-1} = A^{*T}$
We will try to represent an image as a series sum of such UNITARY MATRICES
aa WHY DO THIS?
Let us consider a one-dimensional signal
Put the graph (Sengupta Lecture 12)
Represent the signal by a series summation of orthogonal basis functions
Refer to Inner Product space- $\{u_i(x)\}, 1 \leqslant i \leqslant n$
u_i is perpendicular to u_j, $i \neq j$.
An image is a two-dimensional signal. This can be represented as a series summation of a set of two-dimensional orthogonal basis functions, which we call the basis images.

Basis images

One dimensional discrete signal. $\{u(n), 0 \leqslant n \leqslant N-1\}$
This is a set of samples representing a one dimensional sequence, of size N.
Call this as a vector $\underline{\mathbf{u}}$
Let A be an $n \times n$ matrix. Premultiplying $\underline{\mathbf{u}}$ by A results in a vector $\underline{\mathbf{v}}$, which is again of size N.
Hence, $\underline{\mathbf{v}} = A\underline{\mathbf{u}}$
$\underline{\mathbf{v}}$ is the transformed vector and A is the transformation matrix.
This is same as $v(k) = \sum_{n=0}^{N-1} a(k,n)u(n), 0 \leqslant k \leqslant (N-1)$
This is a series summation to obtain $\underline{\mathbf{v}}$.
We assume that the transformation matrix A is unitary, so that $A^{-1} = A^{*T}$
Hence to get back $\underline{\mathbf{u}}$ from v, we can write
$u(k) = \sum_{n=0}^{N-1} a^{-1}(k,n)v(n), 0 \leqslant k \leqslant (N-1)$
But, $a^{-1}(k,n) = a^{*}(n,k)$
Hence, $u(k) = \sum_{n=0}^{N-1} a^{*}(n,k)v(n), 0 \leqslant k \leqslant (N-1)$
This is same as $\underline{\mathbf{u}} = A\underline{\mathbf{v}}$

Image Transforms

Define $\underline{a}_k^* = \{a^*(k,n), 0 \leqslant n \leqslant N-1\}$
These $\underline{a}_k^*$'s are called basis vectors of A
These basis vectors form the columns of a matrix.
v_k's are nothing but the coefficients to multiply the basis vetors.
Compare with the a_n's of Fourier Series.

Image Transforms 2D

An image of size $N \times N$ can be represented as a one-dimensional signal of size N^2. However, we would like to represent the same as a series summation of two dimensional basis functions.

$$v(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m,n) a_{k,l}(m,n), 0 \leqslant k,l \leqslant N-1 \quad (1)$$

Thus for each and every element of the transformed image, we need to compute over the entire given image.
This gives a transformed image.
$a_{k,l}(m,n), 0 \leqslant m,n \leqslant N-1$ for a particular k,l represents a matrix. Now we need such matrices for each and every element of v. Hence, consider the set $\{a_{k,l}(m,n), 0 \leqslant m,n \leqslant N-1\}$. This is a set of N^2 matrices each of size $N \times N$.
This set is the image transform and is a set of complete orthonormal discrete basis functions.

Image Trasforms

The inverse transform is given by

$$u(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k,l) a^*_{k,l}(m,n), 0 \leqslant m,n \leqslant N-1 \qquad (2)$$

The set of image transforms $\{a_{k,l}(m,n)\}$ should fulfill two properties:
1. Orthonormality

$$\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} a_{(k,l}(m,n) a^*_{(k',l')}(m,n) = \delta(k-k', l-l')$$

2. Completeness

$$\sum_{k=0}^{N-1} \sum_{l=0}^{N-1} a_{(k,l}(m,n) a^*_{(k,l)}(m',n')m = \delta(m-m', n-n')$$

Image Transforms

Elements of $\{v_{k,l}\}$ are the transform coefficients. Define $V = \{v(k,l)\}$ is itself a transformed image
To get a single element of $v(k,l)$, we need to do N^2 operations, and hence to get the complete image transform the computational complexity is N^4. If the transformation is a separable transformaton, then we can reduce the complexity.

$$|x| = \begin{cases} x & \text{if } x \geqslant 0 \\ -x & \text{if } x < 0 \end{cases}$$