A Review of basic Mathematical Transformation used in Image Processing

Dr. Sandeep Mathur[1]
Department of Mathematics
Jodhpur Institute of Engineering & Technology
Jodhpur, India

Dr. Anjali Mathur[2]
Department of Mathematics
Jodhpur Institute of Engineering & Technology
Jodhpur, India

Nitesh Agarwal[3]
Department of Computer Science
Jodhpur Institute of Engineering & Technology
Jodhpur, India

***Abstract*— Mathematics is a very important tool to solve many engineering problems. Almost in every field of engineering mathematics play an important role. Digital Image Processing also such a field that use mathematics as an important tool such as array operation to store image in digital format, to perform geometrical changes in image geometrical transformations are used, in lossy image compression process frequency transformation play an important role. This paper deals with study of basic mathematical transformation (Geometric Transformation, Frequency Transformation) used in image processing.**

***Keywords*: DCT & DST, DST, Transformation.**

1. INTRODUCTION

An image basically a two dimensional figure & can be represented by a 2D pixel matrix in other words an image mathematically can be defined as a two dimensional function $f(x, y)$ where x, y are spatial coordinate & f define the color value of image at x, y in the form of pixel [11].

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Mathematical transformation cannot be performed directly on an image to perform transformation on image, image is stored in the form of a 2D matrix. In image processing system image acquisition is done by some sensing device & then its each coordinate & amplitude value is digitalize using some mathematical sampling & quantization techniques, digitalization of coordinate is known as sampling & digitalization of amplitude is known as quantization. Both these digitalize value stored in the form of 2D matrix on which mathematical transformations are performed. This paper deals with two basic mathematical transformation like geometrical transformation & frequency transformation. The process of basic mathematical transformation is shown in fig. 1

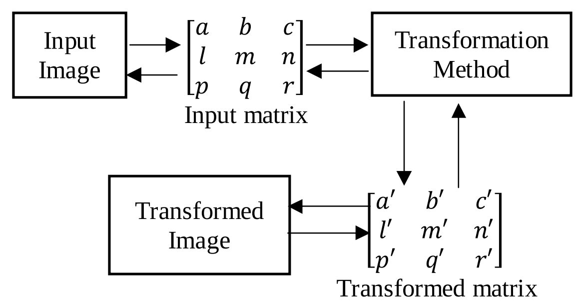


Fig 1: Mathematical Transformation Process in Image Processing

1.1 Geometric Transformation

Geometrical transformation deals with geometrical changes of coordinate values of an input image. It deals with modify spatial relationship between pixels in an image. The geometrical transformation is known as rubber-sheet transformation because all the geometrical transformation can be seen by printing an image on sheet of rubber and then stretching the sheet according to predefined set of rules. In digital image processing geometric transformation consists of two basic operations (1) spatial transformation of coordinate (2) intensity interpolation that assign intensity values to the spatially transformed pixels. The Transformed coordinate can be expressed as

$$(x', y') = T\{(x, y)\} \quad \dots (1)$$

Where (x, y) are pixel coordinate in the original image and (x', y') are the pixel coordinate in transformed image. The affine transform is one of the most commonly used spatial coordinate transformations which is define as

$$[x' \quad y' \quad 1] = [x \quad y \quad 1]T = [x \quad y \quad 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix} \dots (2)$$

This transformations can scale, rotate, translate or sheer an input matrix & can be defined as-

ISSN: 2278-0181
ETRASECT - 2016 Conference Proceedings

Identity
$$x = u \quad y = w \qquad \text{... (3)}$$

Scaling
$$x = c_x v \quad y = c_y w \qquad \text{... (4)}$$

Rotation
$$x = v\cos\theta - w\sin\theta \quad y = v\sin\theta + w\cos\theta \qquad \text{... (5)}$$

Translation
$$x = v + t_x \quad y = w + t_y \qquad \text{... (6)}$$

Shear(vertical)
$$x = v + s_v w \quad y = w \qquad \text{... (7)}$$

Shear(horizontal)
$$x = v \quad y = s_h v + w \qquad \text{... (8)}$$

In the form of affine matrix these transformation can be written as [11]

Transformation Name	Affine Matrix T
Identity	$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$
Scaling	$\begin{bmatrix}c_x&0&0\\0&c_y&0\\0&0&1\end{bmatrix}$
Rotation	$\begin{bmatrix}\cos\theta&\sin\theta&0\\-\sin\theta&\cos\theta&0\\0&0&1\end{bmatrix}$
Translation	$\begin{bmatrix}1&0&0\\0&1&0\\t_x&t_y&1\end{bmatrix}$
Shear(vertical)	$\begin{bmatrix}1&0&0\\s_v&1&0\\0&0&1\end{bmatrix}$
Shear(horizontal)	$\begin{bmatrix}1&s_h&0\\0&1&0\\0&0&1\end{bmatrix}$

Fig 2: Basic Affine Transformation on pixel matrix

1.2 Frequency Transformation:
Frequency Transformation deals with transform the spatial domain of image into its equivalent frequency domain using some sine & cosine functions. Present paper deals with two basic transformations DCT & DST. Both transformation convert a signal into its equivalent frequency domain & can work with single & multiple variable. DCT & DST convert an image into its equivalent frequency domain by partitioning image pixel matrix into blocks of size N*N, N depends upon the type of image. For example if we used a black & white image of 8 bit then all shading of black & white color can be expressed into 8 bit hence we use N=8, similarly for color image of 24 bit we can use N=24 but using block size N=24 time complexity may increase hence we operate DCT & DST on individual color component for a color image. Color image consist of 8 bit red + 8 bit green + 8 bit blue hence we apply DCT & DST on each color component (Red, Green, Blue) using block size N=8.

1.2.1 One-Dimensional DCT:
If we have one-D sequence of signal value of length N then its equivalent DCT can be expressed as

$$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \qquad \text{...(9)}$$

for $u = 0,1,2,\ldots,N-1$.

& inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1}\alpha(u)c(u)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \qquad \text{...(10)}$$

Where $f(x)$ is signal value at point x & $\alpha(u)$ is transform coefficient for value u.

$$\alpha(u) = \begin{cases}\sqrt{\frac{1}{N}} & for\ \ u = 0\\ \sqrt{\frac{2}{N}} & for\ u \neq 0\end{cases} \qquad \text{...(11)}$$

1.2.1.1 Two – Dimensional DCT

An image is 2-D pixel matrix where each position (i,j) represents a color value for that particular point or position. Hence to transform an image into its equivalent DCT matrix we use 2-D DCT [7].

2-D FDCT can be defined as

$$C(u,v) = \alpha(u)\,\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \text{..(12)}$$

for $u,v = 0,1,2,\ldots,N-1$.

& inverse transformation is defined as (IDCT)

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\,\alpha(v)c(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \text{..(13)}$$

Where $C(u,v)$ represents frequency value for u,v & $f(x,y)$ represents pixel color value at position (x,y).

$$\alpha(u) = \begin{cases}\sqrt{\frac{1}{N}} & for\ \ u = 0\\ \sqrt{\frac{2}{N}} & for\ u \neq 0\end{cases} \qquad \text{...(14)}$$

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & for\ \ v = 0 \\ \sqrt{\frac{2}{N}} & for\ v \neq 0 \end{cases} \qquad ...(15)$$

1.2.2 One-Dimensional DST:
For one-D sequence of signal value of length N then its equivalent DST can be expressed as

$$s(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \sin\left[\frac{\pi(2x+1)(u+1)}{2N}\right] \qquad ...(16)$$

for $u = 0,1,2,\ldots,N-1$.

& inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) s(u) \sin\left[\frac{\pi(2x+1)(u+1)}{2N}\right] \qquad ...(17)$$

Where $f(x)$ is signal value at point x & $\alpha(u)$ is transform coefficient for value u & define as same as one dimensional DCT.

1.1 Two – Dimensional DST

2-D FDST can be defined as

$$s(u,v) = \alpha(u)\,\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \sin\left[\frac{\pi(2x+1)(u+1)}{2N}\right] \sin\left[\frac{\pi(2y+1)(v+1)}{2N}\right] ..(18)$$

for $u, v = 0,1,2,\ldots,N-1$.

& inverse transformation is defined as (IDST)

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\,\alpha(v)\, s(u,v) \sin\left[\frac{\pi(2x+1)(u+1)}{2N}\right] \sin\left[\frac{\pi(2y+1)(v+1)}{2N}\right] ..(19)$$

Where $s(u,v)$ represents frequency value for u, v, $f(x,y)$ represents pixel color value at position (x, y) & $\alpha(u)$ is defines as same as Two Dimensional DCT [8].

2. MAIN RESULTS & OUTPUTS

2.1 Implementation of Geometric Transformation on an Image

Steps involved in this implementation

1. Create pixel matrix of the image.
2. Apply different geometric transformation on each pixel of image as per requirement.
3. Store transformed matrix as output of geometric transformation.
4. Using this transformed pixel matrix get transformed image.

2.2 Implementation of Frequency Transformation
Steps involved in this implementation

1. Create pixel matrix of the image & divided it into blocks of size 8*8
2. Apply FDCT (Forward Discrete Cosine Transform) or FDST on each 8*8 block of pixel matrix to get equivalent 8*8 DCT or DST blocks respectively.
3. To get Original image we apply IDCT (Inverse Discrete Cosine Transform) or IDST on each 8*8 block DCT or DST respectively & get its equivalent 8*8 IDCT or IDST block respectively.
4. Using 8*8 IDCT or IDST blocks we create original pixel matrix to get original image.
5. Now we Find MSE (Mean Squared Error) & PSNR (Peak Signal To Noise Ratio) to determine quality of image obtain by IDST. MSE & PSNR calculated by following formulas

$$MSE = \frac{1}{H*W} \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} [o(x,y) - m(x,y)]^2 \qquad (20)$$

$$PSNR = 20*\log_{10}(MAX) - 10*\log_{10}(MSE) \qquad (21)$$

Where H=Height of Image, W= Width of Image, variable MAX shows max value of a pixel for example if image is 8 bit then MAX=255.

6. Quality of image obtain by IDCT or IDST is depend on MSE & PSNR value. If as the MSE value increases PSNR value decreases then we get a bad quality of image by IDST or IDCT & if as the MSE value decreases PSNR value increases we get a batter quality image hence a best suitable transformation like DCT, DST, DFT is taken on the basis of this MSE & PSNR value.

2.3 Outputs:

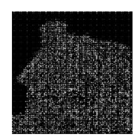
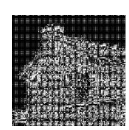
Frequency Transformation	Input Image	Output Image
FDCT		
IDCT		
FDST		
IDST		

Table 1: Forward & inverse frequency transformation

	MSE	PSNR
2D DST	0.37	52.47
2D DCT	0.29	53.52

Table 2: MSE & PSNR value of input image after inverse transformation

Geometric Transformation	Input Image	Output image
Identity		
Scaling	 155%	
Rotation	 180^0 & 45^0	 
Translation	 $t_x = 50$, $t_y = 10$	
Shear(vertical)	 $s_v = 1$	
Shear(horizontal)	 $s_h = 1$	

Table 3: Different Geometric Transformation on an image

3. Conclusion

The result presented in this document shows that

1. The results shows that both DCT & DST transformation add some error to input image.

2. DCT is more efficient then DST to transform an image into frequency domain because it add less error then DST in input image.

3. In the rotation transformation output pixel value goes out of bound from the range of image area.

REFERENCES

[1] A. M. Eskicioglu, and P. S. Fisher, “Image quality measures and their performance,” IEEE Trans. Commun., vol. 43, no. 12, pp. 2959-2965, Dec. 1995.
[2] Andrew B. Watson, “Image Compression Using Discrete Cosine Transform”, NASA Ames Research Centre, 4(1), pp. 81-88, 1994.
[3] Anjali Kapoor and Dr. Renu Dhir, “Image Compression Using Fast 2-D DCT Technique”, International Journal on Computer Science and Engineering (IJCSE), vol. 3 pp. 2415-2419, 6 June 2011.
[4] Harley R. Myler and Authur R. Weeks “The Pocket Handbook of Image Processing Algorithms in C”, ISBN 0-13-642240-3 Prentice Hall P T R Englewood Cliffs, New Jercy 07632
[5] Iain E.G. Richardson “H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia”, ISBN 0470848375, 9780470848371, Wiley, 2003.
[6] L.Dhang, W. Dong D.Zhang and G.Shi “Two stage image denoising by principal component analysis with local pixel grouping” Pattern Recognition, Vol.43, pp1531-1549, 2010.
[7] N.Ahmed, T.Natatarajan, and K.R. Rao, “Discrete Cosine Transform”, IEEE Transactions on Computers, vol. C-32, pp. 90-93, Jan. 1974.
[8] S. Malini. & R.S. Moni. ”Use of Discrete Sine Transform for A Noval Image Denoising Technique”. International Journal of Image Processing (IJIP), Vol, 8, Issue 4, pp. 204-213, 2014.
[9] Swati Dhamija and Priyanka Jain “Comparative Analysis for Discrete Sine Transform as a suitable method for noise estimation” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3pp. 162-164, September 2011.
[10] V.P.S.Naidu, “Discrete cosine Transform based Image Fusion”, Defense Science Journal, Vol.60, No.1, pp.48-54., Jan.2010.
[11] Rafael C. Gonzalez & Richard E. Woods “Digital Image Processing”, 3rd edition, ISBN: 978-93-325-1846-9, Pearson Publication, 2014.