

Rotation and Scaling Image Using PCA

Hawrra Hassan Abass

Electrical & Electronics Dept., Engineering College

Kerbela University, Kerbela, Iraq

Tel: 964-78-0133-7344 E-mail: hawrrahassan@yahoo.com

Firas Mahdi Muhsin Al-Salbi

Electrical & Electronics Dept., Engineering College

Kerbela University, Kerbela, Iraq

Tel: 964-77-0605-3224 E-mail: firassalbi1974@yahoo.com

Received: October 20, 2011

Accepted: November 14, 2011

Published: January 1, 2012

doi:10.5539/cis.v5n1p97

URL: <http://dx.doi.org/10.5539/cis.v5n1p97>

Abstract

An important aspect of human visual system is the ability to recognize objects despite of the change in their position, size, and orientation. Therefore, any pattern recognition system for many applications, such as for industry or military, must have the ability to recognize geometrically distorted patterns, i.e. scaled, rotated and shifted patterns. In this paper Hotelling transform algorithm has been used for scale and rotation invariant image processing, with the help of MATLAB-v10 simulation program, this algorithm has been implemented and tested on many images with black and white format of different scales and rotation angles.

Keywords: Rotation, Image Processing, Hotelling Transform, Invariant, Pattern Recognition, PCA

1. Introduction

Image processing is a rapidly growing area of computer science. Its growth has been fueled by technological advances in digital imaging, computer processors and mass storage devices. Fields which traditionally used analog imaging are now switching to digital systems, for their reliability and affordability. Important examples for this field are medicine, video production, photography, remote sensing, and security monitoring. These and other sources produce huge volumes of digital image data every day, more than could ever be examined manually. Digital image processing is concerned primarily with extracting useful information from images (Gonzalez, 1991; Pratt, 1978). Ideally, this is done by computers, with little or no human intervention. Image processing algorithms may be placed at three levels. At the lowest level are those techniques which deal directly with the raw, possibly noisy pixel values, with denoising and edge detection being good examples. In the middle are algorithms which utilize low level results for further means, such as segmentation and edge linking. At the highest level are those methods which attempt to extract semantic meaning from the information provided by the lower levels, for example, handwriting recognition.

Another transformation that can be applied to an image is rotation. In this case, the locations of pixels in an image are rotated around a certain origin by a determined rotation angle (Lee, Cho, & Cho, 1995; Adrea & Martina, 2006). Usually the center of the image is selected as origin, and the image is rotated respectively. Figure 1 shows the result of a rotation of 60 degrees in counter clockwise direction. As in the case of translation, some parts of the original image may be lost so that some blank regions will appear in the resultant image. Note that rotation might require interpolation of pixel values, due to transformation characteristics.

2. Hotelling Transform (HT)

The HT is based on statistical properties of an image. The principal uses of this transformation are in data compression and rotation applications (Adrea & Martina, 2006; Schneider & Schneider, 2005). Suppose that an image with $N \times N$, $f(x,y)$, is transmitted M times over some communication channel. Since any physical channel is subject to random disturbances, the set of received images, $\{f_1(x,y), f_2(x,y), \dots, f_M(x,y)\}$, will in general represent a statistical ensemble whose properties are determined by the channel characteristics and the nature of disturbance. Each sample image $f_i(x,y)$ can be expressed in the form of an N_2 dimensional vector x_i as follows:

$$X_i = \begin{bmatrix} X_{i1} \\ X_{i2} \\ \vdots \\ X_{ij} \\ \vdots \\ \vdots \\ X_{iN^2} \end{bmatrix} \quad (1)$$

Where x_{ij} denotes the j th component of vector x_i . One way to construct such a vector is to let the first N component of x_i be the first row of $f_i(x,y)$ image, i.e. $[x_{i1} = f_i(0,0), x_{i2} = f_i(0,1), \dots, x_{iN} = f_i(0, N-1)]$; meanwhile the second set of N components form the second row, and so on. Another way is to use the columns of $f(x,y)$ instead of rows.

The covariance matrix of the X vector is defined as:

$$C_x = E\{(x - m_x)(x - m_x)^T\} \quad (2)$$

Where

$$m_x = E\{x\} \quad (3)$$

is the mean vector, E is the expected value, and (T) indicates transposition.

Equations (2) and (3) above can be approximated from the samples by using the relations:

$$m_x \cong \frac{1}{M} \sum_{i=1}^M X_i \quad (4)$$

$$C_x \cong \frac{1}{M} [\sum_{i=1}^M X_i X_i^T] - m_x m_x^T \quad (5)$$

The mean vector is of N^2 dimensional and C_x is an $N^2 \times N^2$ matrix.

Let e_i with $(i=1,2, \dots, N^2)$, be the eigenvectors and corresponding eigenvalues of C_x , the transformation matrix (A) whose rows are the eigenvectors of C_x which arranged in descending order of corresponding eigenvalues $\lambda_1 > \lambda_2 > \lambda_3 > \dots > \lambda_{N^2}$ is given by:

$$A = \begin{bmatrix} e_{11} & e_{12} & \dots & e_{1N^2} \\ e_{21} & e_{22} & \dots & e_{2N^2} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ e_{N^2 1} & e_{N^2 2} & \dots & e_{N^2 N^2} \end{bmatrix} \quad (6)$$

Where e_{ij} is the j th component of the i th eign vector.

The HT then consists simply of multiplying a centralized image vector, $(x - m_x)$, by the transformation matrix A to obtain a new image vector y that is:

$$Y = A(x - m_x) \quad (7)$$

The above equation has several properties. Let us first examine the covariance matrix of the y vector. This matrix is given by:

$$C_y = E\{(y - m_y)(y - m_y)^T\} \quad (8)$$

Where m_y is equal to the zero vector 0 as can be shown:

$$\begin{aligned} m_y &= E\{y\} \\ &= E\{A(x - m_x)\} = AE\{x\} - Am_x \\ &= Am_x - Am_x \\ &= 0 \end{aligned} \quad (9)$$

Substitution of equations (7) and (9) in (8) yields the following expression for C_y in terms of C_x . It can be shown that C_y is a diagonal matrix with elements equal to the eigenvalues of C_x , that is:

$$Y = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \lambda_{N^2} \end{bmatrix} \quad (10)$$

The importance of this property is that, since the terms of the main diagonal are 0, the elements of y are uncorrected. In addition, each eigenvalue, λ_i , is equal to the variance of the j th element of y along eigenvector e_i (Matschitsch, Tschinder, *et al.*, 2008; Guo, Zhang, & Zhang, 2010; Lowe, 2004).

2.1 Application of HT to Image Processing

One of the basic concepts underlying the derivation of the HT is that the choice of basis vectors (i.e. rows of A) is made such that these vectors point in the direction of maximum variance of the data, subject to the constraint that all vectors be mutually or orthogonal and the transformed components be uncorrelated. These requirements lead to the solution stated above that the bases are the eigenvectors of the covariance matrix (Gonzalez, 1991; Pratt, 1978).

2.1.1 Rotation Free Image

The above properties have important advantages in image rotation. Consider the problem where an object has been extracted from an image and it is desired to rotate the object so that it's aligned in some standard or invariant direction, the object can be extracted by a horizontal or vertical scan of image so that if the background of image is white and the object is black the coordinates of object are the same coordinates of all black pixels in the image.

Consider the coordinate system with x_1 and x_2 axes as shown in Figure 2. If new coordinates are chosen with a different orientation but the same origin as the original system, then we shall say that there has been a rotation of axes in the plane. Let θ be the angle of rotation from the positive half of the x_1 axis to the positive half of the y_1 axis. It then follows from elementary trigonometry that the new and old axes are related by the equations.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (11)$$

The original coordinates of each pixel in the image can be interpreted as 2D random variables with mean and covariance matrix

$$m_x \cong \frac{1}{P} \sum_{i=1}^P x_i \quad (12)$$

$$C_x \cong \frac{1}{P} [\sum_{i=1}^P x_i x_i^T] - m_x m_x^T \quad (13)$$

where P is the number of pixels in the object to be rotated and x_i is the vector composed of the coordinates of the i th pixel. It is noted that m_x is a 2D vector and C_x a (2*2) matrix. Since the eigenvectors of C_x point in the direction of maximum variance subject to the constraint that they be orthogonal, a logical choice is to select the new coordinates system so that it will be aligned with these eigenvectors.

Letting the y_1 and y_2 axes be aligned with the normalized eigenvectors e_1 and e_2 , it then follows from Figure 2 that:

$$\begin{aligned} \cos \theta &= e_{11}, \\ \sin \theta &= e_{12}, \\ -\sin \theta &= e_{21}, \\ \cos \theta &= e_{22}, \end{aligned}$$

Where e_{11} and e_{21} are the projections of e_1 and e_2 along the x_1 axis, and e_{12} and e_{22} are the projections of these two vectors along the x_2 axis. The rotation from the original original coordinate system to this new system is then given by the relation.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (14)$$

Which is seen to be in the familiar form $y=Ax$, where

$$A = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \quad (15)$$

$$= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (16)$$

It is noted that Equation 14 is analogous to a two dimensional HT where the mean has not been subtracted from the original vectors. Subtraction of the mean vector simply centralizes the object so that its centre of gravity is at the origin of the new coordinate system. In this case, the transformation assumes the form:

$$y = A\{x - m_x\} \quad (17)$$

Interpretation of Equation 14 is different in another respect from our earlier discussion of the HT, where each vector was formed from pixels in entire image. In the present application, the transform concerned with the coordinates of pixels in a single image, and the ensemble of vectors used to arrive at the transformation matrix is derived from an object in the image.

2.1.2 Shift Free Image

Shift free of an object is depending on the centre of gravity of an object which is aligned with the centre of original image. When the object is extracted from an image and its coordinates have computed, the object become more flexible to be translated to any position inside the image.

2.1.3 Scale Free Image

Scale free image can be performed for the rotated output process (image) by using normalization to the eigenvalues of the covariance matrix (Mikolajczyk & Schmid, 2005).

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{B}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{B}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (18)$$

Where B is the normalization factor.

The normalization factor has a robust effect on the output of the preprocessing stage because of the size of the output image changes according to it. A high value gives a large size image with high resolution. Meanwhile a low value gives a small size image with low resolution. The choice of (B) value depends on the type or the nature of an image. Figure 3 shows the effect of the normalization factor on the output image for different values of the normalization factor.

2.2 HT Algorithm

This algorithm was implemented using MATLAB version 10 simulation program with the following steps:

Step 0: Present the image.

Step 1: Extract the coordinates of object

Step 2: Compute the covariance matrix.

$$C_x \cong \frac{1}{P} [\sum_{i=1}^P x_i x_i^T] - m_x m_x^T \quad (19)$$

Step 3: Compute the eigenvalues and the eigenvectors of the covariance matrix (using eighn command).

Step 4: Rotation free the coordinates of object.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (20)$$

Step 5: Scale free the coordinates of object using the adaptive algorithm.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{B_x}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{B_x}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (21)$$

Step 6: Express an object coordinates by a bipolar data image.

3. Algorithm Implementation & Results

The algorithm was implemented using MATLAB software package on a personal computer. The algorithm was tested for different cases.

Example 1: Figure 4 illustrates the input image of aircraft with a 60 degree rotation angle and the output image of the HT for normalization factor (B=10) of the output after median filter.

Example 2: Figure 5 illustrates the input image of a heart with a 90 degree rotation angle, and the output image of the HT for normalization factor (B=10).

Example 3: Figure 6 illustrates the input image of a box with a 90 degree rotation angle and the output image of the HT for normalization factor (B=10).

Example 4: Figure 7 illustrates the input image of a hand with a 90 degree rotation angle and the output image of the HT for normalization factor (B=10).

Example 5: Figure 8 illustrates the input image of a triangle with a 90 degree rotation angle and the output image of the HT for normalization factor (B=10).

Example 6: Figure 9 illustrates the input image of Up arrow with a 90 degree rotation angle and the output image of the HT for normalization factor (B=10).

Example 7: Figure 10 illustrates the input image of a clock with a 90 degree rotation angle and the output image of the HT for normalization factor (B=10).

As it can be seen from these tests, the algorithm works in a very good agreement with what was expected.

4. Conclusions

In this research, a free scale and rotation image algorithms have presented. These algorithms have been tested for different types of images. The following can be concluded:

- (1) Hotelling Transform is an efficient method for freeing the input image from scale, shift, and rotation effect. This method is carried by extracting the eigenvalues and eigenvectors from the coordinates of an object in the image.
- (2) The proposed system can be used to shift, scale, and rotate invariant images. The algorithm is able to manipulate shift, scale, and rotation pattern in which the rotation angle range from (-90° to +90°).

References

- Adrea, Gavlasov'a, & Martina, Mudrov'a. (2006). Rotation Invariant Transforms In Texture Feature Extraction. 7th International Scientific – Technical Conference – PROCESS CONTROL.
- Gonzalez, R. (1991). *Digital Image Processing*. Addison-Wesley Publishing Company.
- Guo, Zhenhua, Zhang, Lei, & Zhang, David. (2010). Rotation invariant texture classification using LBP variance (LBPV) with global matching, *Pattern recognition*, 43, 706-719. <http://dx.doi.org/10.1016/j.patcog.2009.08.017>
- Lee, B., Cho, Y., & Cho, S. (1995). New Invariant Pattern Recognition System Based on Preprocessing and Reduced Second order Neural Network. *IEEE Transactions on Neural Network*, 2099-2012.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant key points. *International Journal of Computer Vision*.
- Matschitsch, Stefan, Tschinder, Martin, et al. (2008). Comparison of Compression Algorithms Impact on Iris Recognition Accuracy. *Scientific Commons*.
- Pratt, W. (1978). *Digital Image Processing*, John Wiley and Sons.
- Schneider, Jasper, & Schneider, Skyler. (2005). Hardware Parallelization of the Scale Invariant Feature Transform Algorithm.



Original Image

Rotated Image

Figure 1. Image Sample

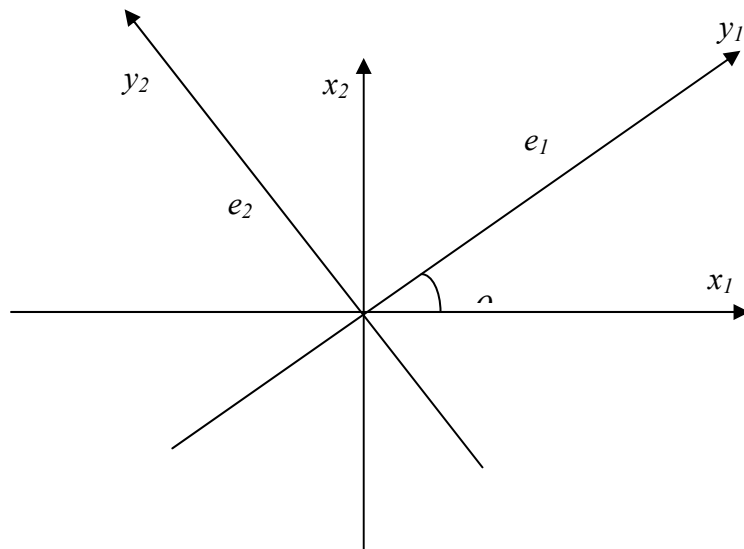


Figure 2. Rotation of coordinate system

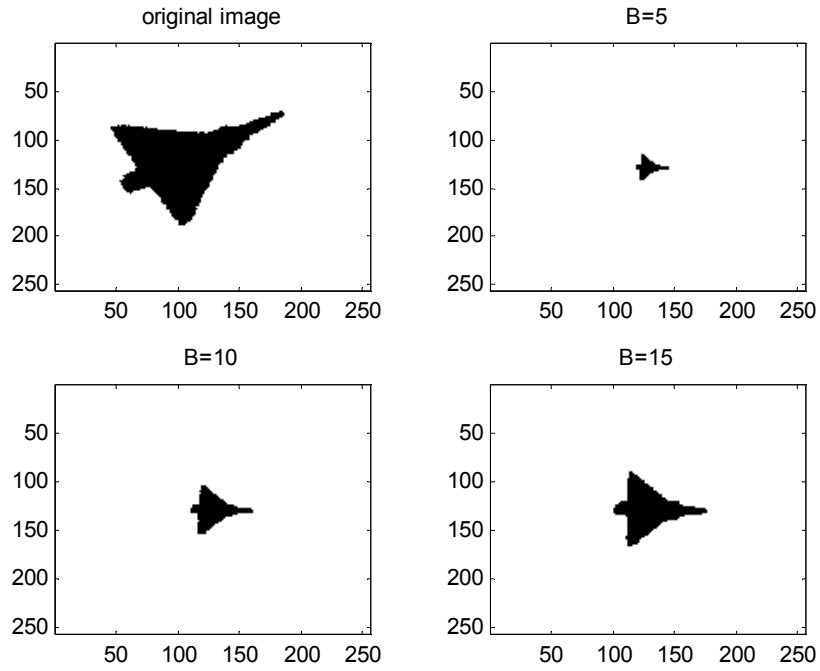


Figure 3. The input image after Hotelling Transform with different normalization factor

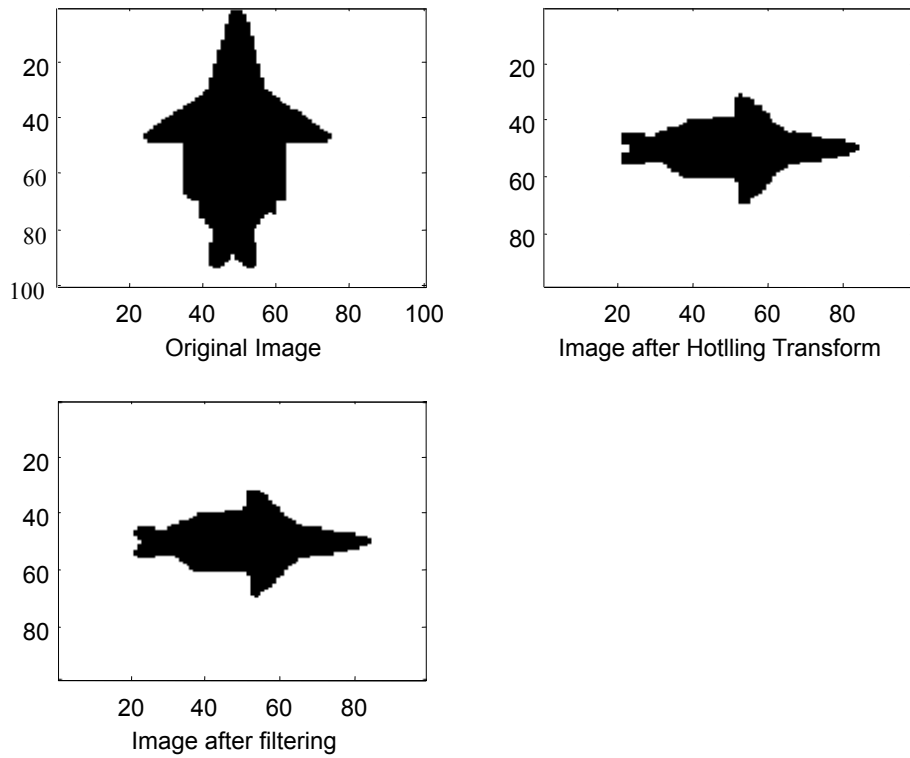


Figure 4. Image of an air craft with 60 degree rotation angle.

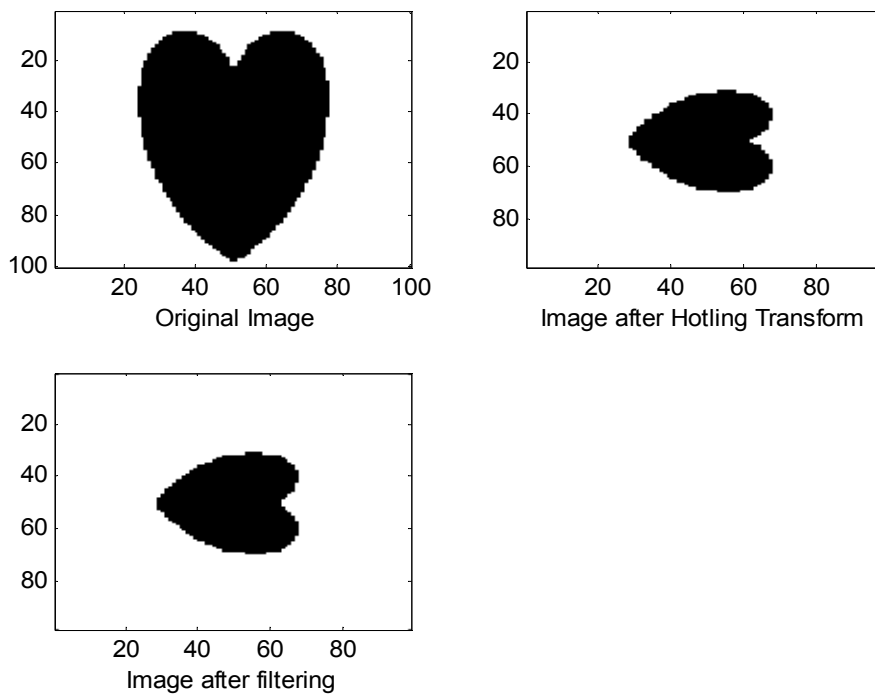


Figure 5. Image of a heart with 90 degree rotation angle

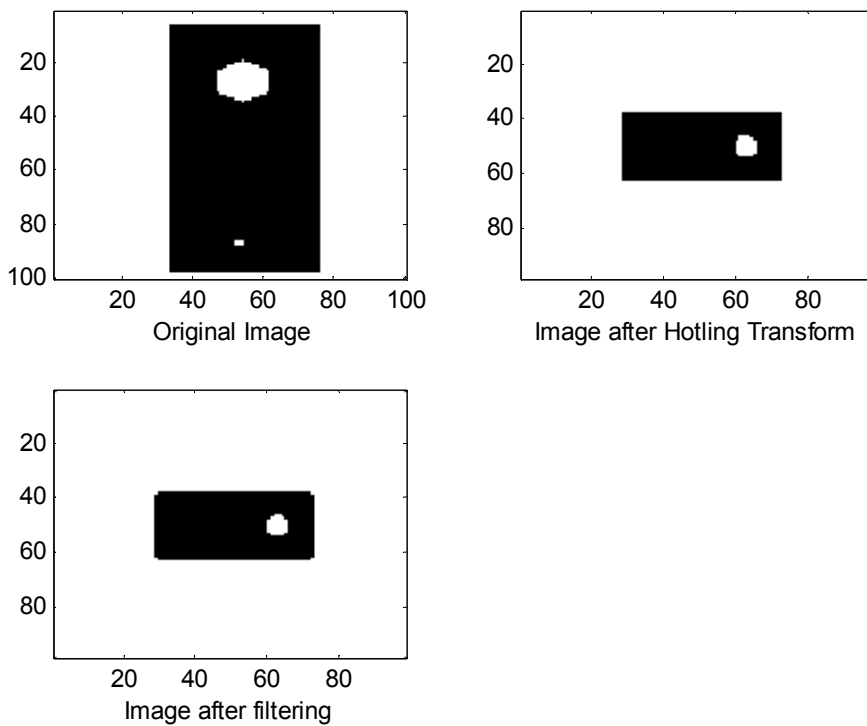


Figure 6. Image of a box with 90 degree rotation angle

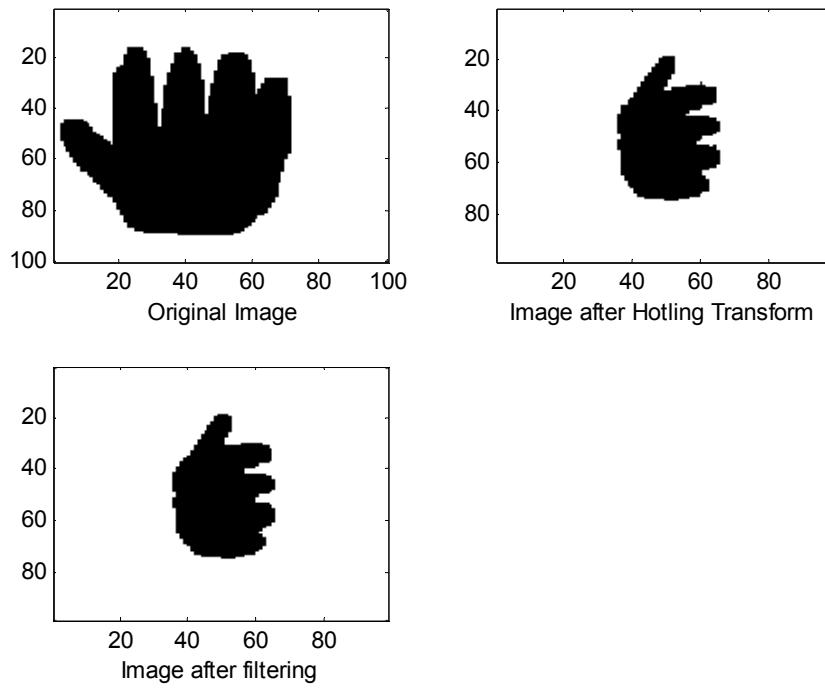


Figure 7. Image of a hand with 90 degree rotation angle

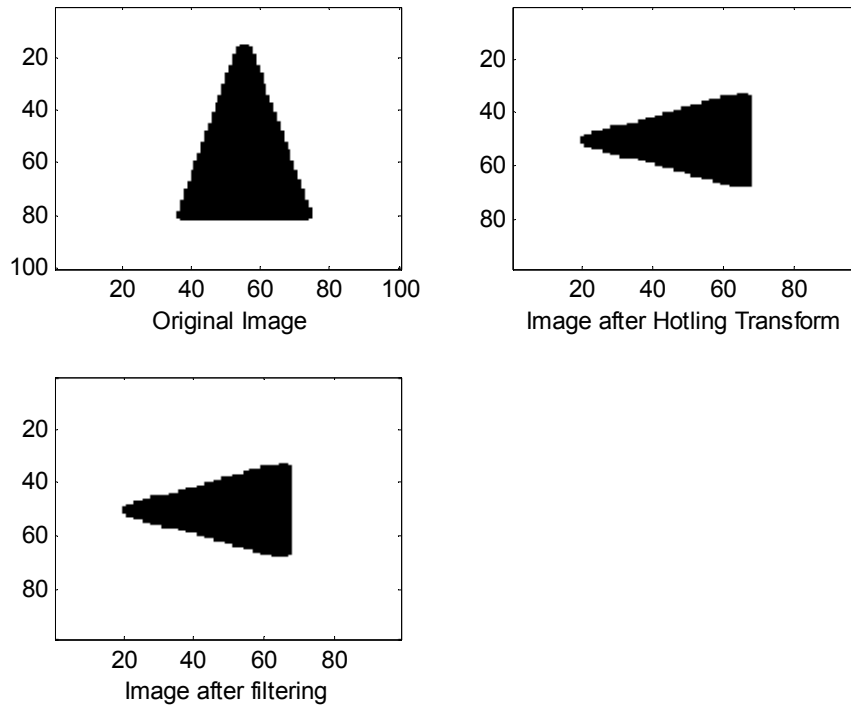


Figure 8. Image of a Triangle with 90 degree rotation angle

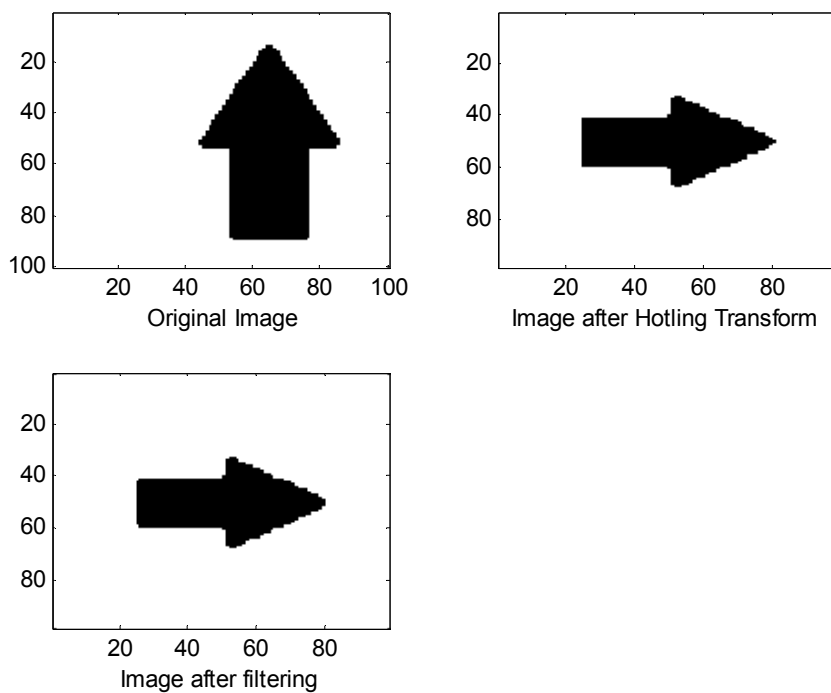


Figure 9. Image of an Up arrow with 90 degree rotation angle

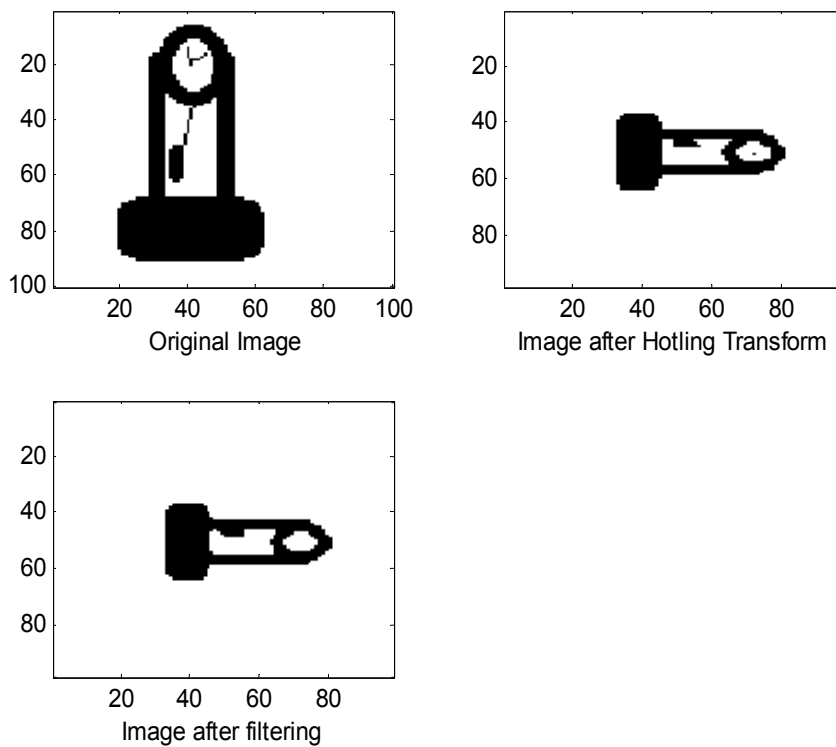


Figure 10. Image of a clock with 90 degree rotation angle