ISSN:0975-9646

K S Chidanand Kumar / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (3) , 2012, 4460 - 4464

A Vision based Application for Virtual Mouse Interface Using Finger-Tip

K S Chidanand Kumar

E&C Department, National Institute of Technology
Surathkal, Karnataka, India

***Abstract*— This paper presents a vision based virtual mouse interface using a single camera. A virtual panel which is a rectangular board with its borders shaded by red color and its four corners shaded by blue color are extracted using HSV color quantization. In consecutive frames of a video, four corners are tracked using color based histogram tracker. The four corners of rectangular panel are mapped to computer screen using Projective mapping. The position and direction of finger tip on virtual panel is estimated using ellipse fitting. The system identifies click events by a V shaped gesture recognized using Fourier descriptor and Nonlinear SVM.**

***Keywords*— Quadrangular Virtual-panel, HSV Color Quantization, Tracking, Homography, Ellipse-fitting, Fourier descriptor, Support Vector Machine.**

I. INTRODUCTION

As computer technology continues to develop, people have smaller and smaller electronic devices and want to use them ubiquitously. There is a need for new interfaces designed specifically for use with these smaller devices. Increasingly we are recognizing the importance of human computing interaction (HCI), and in particular vision-based gesture and object recognition. Touch screens are also a good control interface and nowadays it is used globally in many applications. However, touch screens cannot be applied to desktop systems because of cost and other hardware limitations. By applying vision technology and controlling the mouse by finger tip on virtual panel we can achieve accurate and effective interaction with computer even at larger distances away from camera.

Many researchers in the human computer interaction and robotics fields have tried to control mouse movement using video devices. However, all of them used different methods to make a clicking event. One approach, by Erdem et al, used finger tip tracking to control the motion of the mouse. A click of the mouse button was implemented by defining a screen such that a click occurred when a user's hand passed over the region [1, 2, and 3]. Another approach was developed by Chu-Feng Lien [4]. He used only the finger-tips to control the mouse cursor and click. His clicking method was based on image density, and required the user to hold the mouse cursor on the desired spot for a short period of time. Paul et al used still another method to click. They used the motion of the thumb (from a 'thumbs-up' position to a fist) to mark a clicking event thumb. Movement of the hand while making a special hand sign moved the mouse pointer. Asanterabi Malima et al. [5] used a finger counting system to control behaviours of a robot. He used color calibration method to segment the hand region and convex hull algorithm to find finger tip positions [4].

The remainder of the paper is organized as follows. In Sect. 2, we present system setup followed with an overview of the functional block diagram. In Sect. 3, we extract virtual panel and corners using HSV color quantization scheme. In Sect. 4, we track four corners using histogram based tracker. In Sect. 5, we detect the position and orientation of finger tip followed with click event detection in Section 6. Section 7 & Section 8 concludes the paper with results and references.

II. OVERVIEW

The system set up mainly involves camera mounted on the ceiling behind the user and away from computer. A quadrangular white color cardboard with its border shaded by red color and its four corners shaded with dark blue color circular patches is used as a virtual panel. Virtual panel is extracted using HSV color quantization scheme and four corners of virtual panel are tracked using histogram based color tracker. Users can use their fingertip as a mouse on the virtual panel to simulate a cursor for the remote display. Homography can map any point on the virtual panel to a remote display. Finger tip detector detects position and orientation of finger tip on the virtual panel. Current system implements pressing/clicking events by holding a particular gesture for a while which is recognized using gesture recognition module.

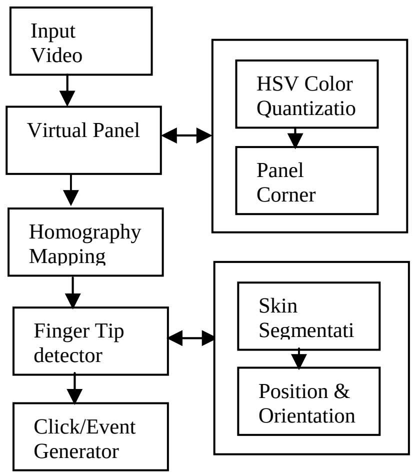


Fig. 1 An Overview of Vision based Virtual Mouse Interface

III. VIRTUAL PANEL DETECTION

An arbitrary quadrangular shaped plane object such as piece of cardboard is used as a panel. Each side of quadrangular shaped object is shaded with Red color. The four corners of quadrangular shaped object are marked with circular shaped blue color. The extraction of virtual panel is mainly done using HSV color quantization scheme [6] as shown below.

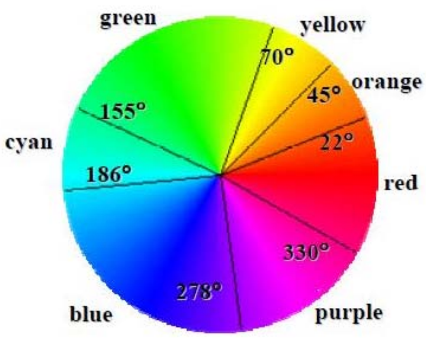


Fig. 2 Hue panel quantization

For $v \in [0,0.2]$, it is a black area, $l = 0$

For $s \in [0,0.2]$ and $v \in [0.2,0.8]$, it is gray area,

$l = [(v - 0.2)*10] + 1$

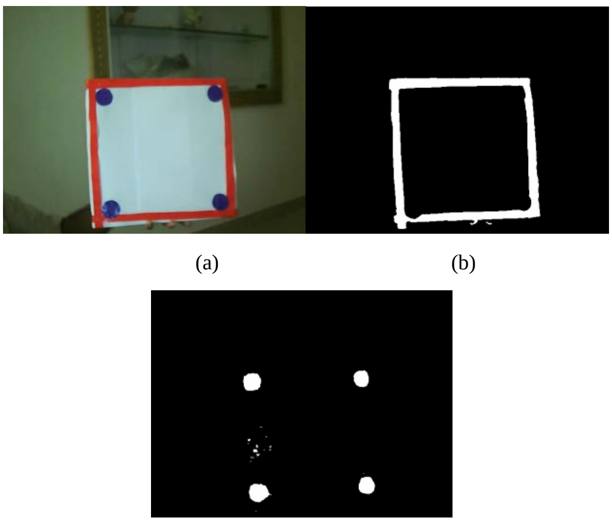
For $s \in [0,0.2]$ and $v \in [0.8,1.0]$, it is white area, $l = 7$

For $s \in [0.2,1.0]$ and $v \in [0.2,1.0]$, it is color area,

$$H = \begin{cases} 0 \; if \;\; hue \in (330,22) \\ 1 \; if \;\; hue \in (330,22) \\ 2 \; if \;\; hue \in (330,22) \\ 3 \; if \;\; hue \in (330,22) \\ 4 \; if \;\; hue \in (330,22) \\ 5 \; if \;\; hue \in (330,22) \\ 6 \; if \;\; hue \in (330,22) \end{cases} \quad S = \begin{cases} 0 \triangleright s \in (0.2,0.65) \\ 1 \triangleright s \in (0.65,1.0) \end{cases} \quad V = \begin{cases} 0 \triangleright v \in (0.2,0.7) \\ 1 \triangleright v \in (0.7,1.0) \end{cases} \qquad (1)$$

$l = 4*H + 2*S + V + 8$

As a result, we can quantize any color into 36 bins. For Red color $l \in [8,11]$ and for Blue color, $l \in [28,31]$. The results of virtual panel extraction and its four corners are as shown in Figure.3 (b) & (c) below from an input image of Figure.3 (a).



(a) (b)

(c)

Fig. 3 Extraction of Red and Blue color from an Input Image

IV. VIRTUAL PANEL CORNER TRACKER

The four corners of virtual panel which are marked dark blue color circular patches were tracked by the below Histogram based color tracker [7].

Given: The target model histogram $\{q_u\}_{u=1,2...m}$ and its location y_0 in the previous frame.

1) Initialize the location of the target in current frame and compute current model histogram $\{p_u(y_0)\}_{u=1,2...m}$

2) Compute the weights of each bin $\{w_i\}_{i=1,2...n}$ using

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{q_u}{p_u(y_0)}} \qquad (2)$$

3) Find the next location of the target y_1 using

$$y_1 = \frac{\sum_{i=1}^{n} w_i * x_i}{\sum_{i=1}^{n} w_i} \qquad (3)$$

4) Compute current model histogram $\{p_u(y_1)\}_{u=1,2...m}$ and evaluate $\upsilon(p_u(y_1), q) = \sum_{u=1}^{m} \sqrt{p_u(y_1) q_u}$ (4)

5) While $\upsilon(p_u(y_1), q) < \upsilon(p_u(y_0), q)$

Do $y_1 \leftarrow (y_0 + y_1)/2$ (5)

Evaluate $\upsilon(p_u(y_1), q)$

6) If $\|y_1 - y_0\| < \in$, stop

Otherwise set $y_0 \leftarrow y_1$ and go to step 2.

V. PERSPECTIVE MAPPING

To control cursor position on remote display we have used quadrangular shaped panel, we need to map a point on panel to a point on remote display which can be achieved using 2D planar homography. Map four corners of quadrangular object to four corners of rectangle in scene, forming the matched point $(0,0) \leftrightarrow (x_1, y_1)$ $(0, \prod_{width}) \leftrightarrow (x_2, y_2)$ $(\prod_{length}, 0) \leftrightarrow (x_3, y_3)$ $(\prod_{length}, \prod_{width}) \leftrightarrow (x_4, y_4)$.

The planar perspectivity between two planes $\prod$ and $\prod'$ is described by 3*3 Homography matrix [8] below

$$\prod = H_{3*3} \prod' \qquad (6)$$

With these four correspondences between world plane assuming $Z = 0$ and image plane, the homography is computed below:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' \\ : & : & : & : & : & : & : & : \\ : & : & : & : & : & : & : & : \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx_n' & -y_nx_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny_n' & -y_ny_n' \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ : \\ : \\ x_n' \\ y_n' \end{bmatrix} \quad (7)$$

This can be written in the form $A'h' = B$. The resulting simultaneous equations for the 8 unknown elements are then solved using a Gaussian elimination in the case of a minimal solution or using a pseudo-inverse method in case of an over-determined system [8]. The result of perspective mapping between virtual panel and remote display is as shown in Figure.4 (b) for an input image in Figure.4 (a) below.

(a) (b)

Fig. 4 Perspective mapping between Virtual panel and Remote display

VI. Finger tip detection

Since a tip pointer is on/off the panel frequently, the system should have the capability of detecting the tip pointer automatically when it appears on the panel. We used skin detection technique in order to detected finger.

A. *Skin Detection*

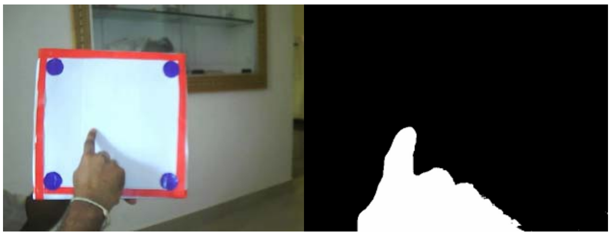
Here we find the skin color distribution by analyzing the color histogram of the images in RGB color space. This is done by two class classifiers. The prior histograms used for classification are pre-computed using the offline database provided by Jones and Rehg [9]. The conditional probabilities of foreground (f_g) and background (b_g) are respectively $p(f_g / rgb)$ and $p(b_g / rgb)$. We give some threshold T to distinguish the classification boundary based on the criteria that the ratio of $p(f_g / rgb)$ and $p(b_g / rgb)$ exceeds T , which is based on a below equation.

$$T < \frac{p(f_g / rgb)}{p(b_g / rgb)} = \frac{p(rgb / f_g) * p(f_g)}{p(rgb / b_g) * p(b_g)} \quad (8)$$

If $p(f_g)$ is the probability of an arbitrary pixel in an image being skin. We can write:

$$T * \frac{(1 - p(f_g))}{p(f_g)} < \frac{p(f_g / rgb)}{p(b_g / rgb)} \quad (9)$$

Where $p(f_g) = 0.09$ in the training database. With given $p(f_g)$ the threshold T can be empirically found by computing "Receiver Operating Characteristics" (ROC curve) over a collection of 100 images [9]. Here a threshold was chosen such that 85% correct classification is achieved while having 25% chance of false alarm. The choice of the threshold is made by the fact that the optimum value of the threshold lies near the bend of the ROC curve. In our experiment the value of T is computed as 0.06.The result of pixel classification gives a binary mask, in which '0's correspond to background pixels and '1's correspond to foreground pixels. Morphological closing operation is done in order to fill the holes. We used maximum area criteria to eliminate noise effects, while extracting the hand. The results of skin detection on remote display as shown in Figure.5 (b) for an input image of Figure.5 (a) below.



(b) (b)

Fig. 5 Skin Extraction from Perspective Corrected image

B. *Position and Orientation of Finger Tip*

After extracting human skin in the virtual panel, the position of finger tip is found. Assuming that the finger tip location lies in the first knuckle of index finger, the length of the elongated axis of the hand region can be computed using below equation.

$$l = \sqrt{\frac{(a + c) + \sqrt{b^2 + (a - c)^2}}{2}} \quad (10)$$

$$a = \frac{M_x}{M_{00}} - C_x^2, b = 2 * \left(\frac{M_{xy}}{M_{00}} - C_x * C_y \right), c = \frac{M_y}{M_{00}} - C_y^2$$

The end point of l is determined as location of the finger tip.

Orientation of finger tip is found using below equation

$$\theta = \frac{1}{2} a\tan\left(\frac{2M_{xy}}{M_x - M_y}\right) \quad (11)$$

$$M_{xy} = \frac{1}{Area} \sum\sum (X - C_X)(Y - C_y) I(X,Y)$$

$$M_x = \frac{1}{Area} \sum\sum (X - C_X)^2 I(X,Y)$$

$$M_y = \frac{1}{Area} \sum\sum (Y - C_Y)^2 I(X,Y),$$

M_x is the inertial moment relative to Y axis in respect to the center of mass; M_y is the inertial moment relative to X axis in respect to the center of mass; is the inertial moment elative to both X and Y axis in respect to the center of mass, C_x and C_y represent the center of mass coordinates of the binarized bounding box. The results of finger tip detection and its centroid is shown in Figure.6 below.

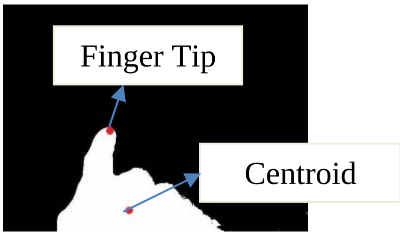


Fig. 6 Finger-Tip detection and its centroid

VII. CLICK EVENT DETECTION

Clicking/pressing event is simulated by a V shaped gesture for a short period of time, and a beep sound is generated as a feedback to indicate that an event is activated. This gesture is illustrated in Figure.7 below.

Fig. 7 Gesture for Click event

In order to recognize the gesture, Fourier descriptor is used as a part of feature extraction process and Non-linear SVM is used for recognition of gesture.

A. *Fourier Descriptor*

Fourier transformation is widely used for shape analysis [10, 11] .The Fourier transformed coefficients form the Fourier descriptors of the shape. These descriptors represent the shape in a frequency domain. The lower frequency descriptors contain information about the general features of the shape, and the higher frequency descriptors contain information about finer details of the shape. It is computed following below:

a) Extract the boundary pixel co-ordinates of hand contour and resample the boundary in order to obtain a uniform resampling along the running arc length of the boundary.

b) Let (x_k, y_k) represents sampled boundary pixels points with $k = 0,1,....N-1$

Express the boundary co-ordinates as a complex signal

$$z(k) = (x(k) - x_0) + i*(y(k) - y_0) \qquad (12)$$

The boundary pixel co-ordinates are subtracted from centroid (x_0, y_0) in order to make translation invariant.

c) Apply Discrete Fourier transform on complex signal $z(k)$ to give

$$z(u) = \frac{1}{K}\sum_{k=0}^{K-1} z(k)e^{-j2\Pi uk/K}, u = 0,1,2..K-1 \qquad (13)$$

d) Rotation and scale invariant of Fourier descriptor is done by using below

$$x = \left[\frac{|F(2)|}{|F(1)|}, \frac{|F(3)|}{|F(1)|}, ... \frac{|F(K/2)|}{|F(1)|}\right] \qquad (14)$$

B. *Support Vector Machine*

Support Vector Machines (SVM) has been considered as one of the powerful classifiers for character and numeral recognition. SVM is defined for two-class problem and it finds the optimal hyper-plane which maximizes the distance, the margin, between the nearest examples of both classes, named support vectors [12]. Given a training database of M data: $\{x_m \mid m = 0,1,...M\}$, the linear SVM classifier is then defined as: $F(x) = \sum \alpha_j x_j .x + b$ where $\{x_j\}$ are the set of support vectors and the parameters α_j and b have been determined by solving a quadratic problem. The linear SVM can be extended to a non-linear classifier by replacing the inner product between the input vector x and the support vectors x_j , through a kernel function K defined as: $K(x, y) = \phi(x).\phi(y)$. This kernel function should satisfy the Mercer's Condition. The performance of SVM depends on the kernel. We have used polynomial kernel of order 5, which outperformed the other commonly used kernels in the preliminary experiments. The SVM is trained with the training samples using Polynomial kernel. The Fourier descriptor of 62 length feature vector obtained in the previous process will be an input to the SVM for classification.

VIII. RESULTS AND DISCUSSION

Figure 8, 9(a)-(h) shows the various stages in detecting finger tip for Virtual mouse interface from input image. Figure.8, 9(a), (b), (c) shows extraction of Virtual panel and its four corners using HSV color quantization scheme. These four corners marked by dark blue color circular patches are tracked using Histogram based color object tracker. Figure.8, 9(e) shows perspective mapping between four corners of virtual panel to remote display. Figure.8, 9(f) shows skin color segmented output along with finger tip detection and centroid of hand. Figure.8, 9(h) shows click events corresponding to input V shaped gesture Figure.8, 9(g) are recognized using Fourier descriptor and Support Vector Machine (SVM).

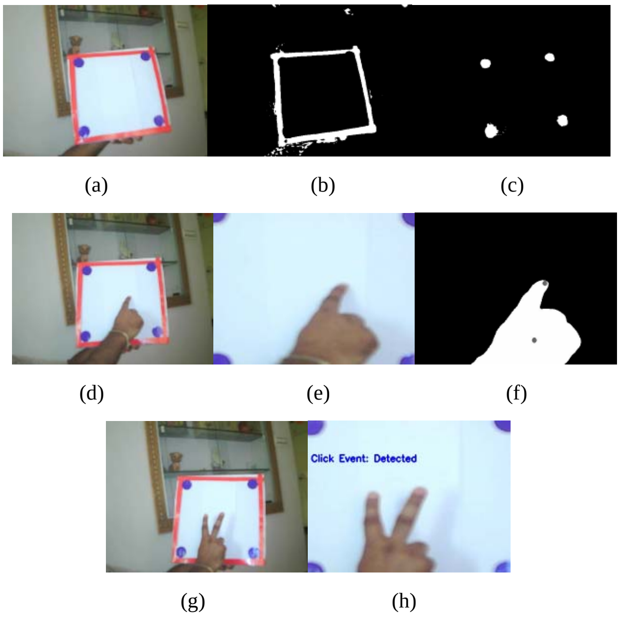


Fig. 8

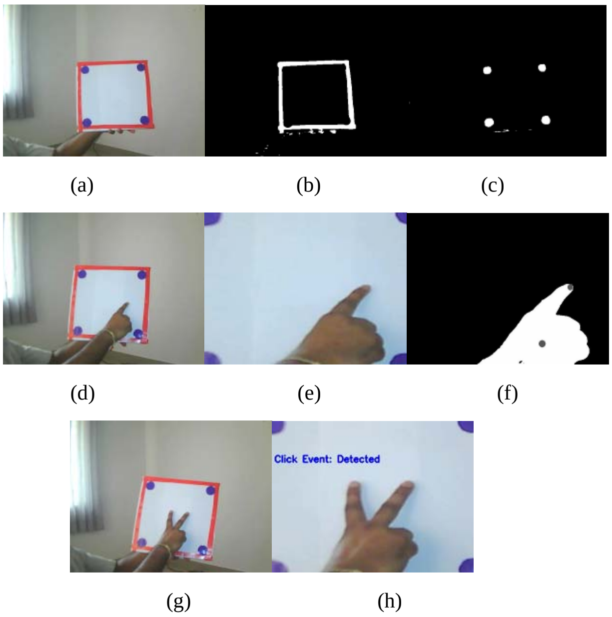


Fig. 9

The proposed method has been implemented on Intel Dual Core processor with 1.6GHZ, CPU 256 MB RAM running on windows vista operating system. The program was developed using Visual C++ language and OpenCV2.0. Colour videos of resolution 1280*720 are captured using Microsoft HD camera mounted on ceiling behind the user. To evaluate the performance of the above proposed approach, we tested with three videos and performance of the system is tabulated in Table 1 &2 below.

TABLE I
FINGER TIP DETECTION ACCURACY

	# Total frames	# Detected frames	% Accuracy
Video1	895	803	89.72
Video2	760	716	94.21
Video3	954	927	97.17
%Average	93.7		

TABLE III
CLICK EVENT DETECTION ACCURACY

	#Total frames	#Click event frames	#Detected frames	% Accuracy
Video1	895	68	56	82.36
Video2	760	46	41	89.13
Video3	954	94	83	88.3
%Average				86.6

Though the system accuracy is quite good in controlled environments, we find several drawbacks and limitations when user behaviour and environment are unpredictable.

a) High error rate on fast moving motion
b) If the environmental lights are changing, false segmentation of fingers resulting in misclassification of gesture for Click events.

IX. CONCLUSIONS

The proposed method presents a vision based application for virtual mouse interface using finger tip. Virtual panel which represents remote display is first extracted using HSV color quantization. The four corners of virtual panel marked by dark blue color circular patches are tracked using Histogram based color object tracker. Homography maps four corners of virtual panel to four corners of remote display. Finger tip detector module detects finger tip position and orientation using ellipse fitting. Finally Click/Pressing events are simulated by a V shaped gesture recognized using Fourier descriptor and Support Vector Machine. Future work of this project includes making the Finger tip detector module invariant to illumination changes and 3D pose estimation of panel which can be used for Augmentation reality of 3D objects.

REFERENCES

[1] A. Erdem, E. Yardimci, Y. Atalay, V. Cetin, "*Computer vision based mouse*", Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASS). IEEE International Conference.
[2] Robertson P., Laddaga R., Van Kleek M," *Virtual mouse vision based interface*", 2004.
[3] *Vision based Men-Machine Interaction* http://www.ceng.metu.edu.tr/~vbi/
[4] Chu-Feng Lien, "*Portable Vision-Based HCI - A Real-time Hand Mouse System on Handheld Devices*"
[5] Asanterabi Malima, Erol Ozgur, and Mujdat Cetin," *A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control*"
[6] Zhang, L., Lin, F., Zang, B," *A CBIR method based on color-spatial feature*", In: IEEE Region 10 International Conference TENCON, pp. 166–169 (1999) 18
[7] Dorin Comaniciu , Visvanathan Ramesh , Peter Meer, "*Kernel-Based Object Tracking*", IEEE Transactions on Pattern Analysis and Machine Intelligence, v.25 n.5, p.564-575, May 2003
[8] R. I. Hartley and A. Zisserman, "*Multiple View Geometry in Computer Vision*", 2nd ed. Cambridge University Press, ISBN: 0521540518, 004.
[9] Jones and Rehg , "*Statistical color models with application to skin detection*", Proc. Computer Vision and Pattern Recognition, IEEE, 1999
[10] Eric Persoon and King-sun Fu 1977, "*Shape Discrimination Using Fourier Descriptors*", IEEE Trans. On Systems, Man and Cybernetics, Vol. SMC- 7(3):170-179.
[11] Fethi Smach, Cedric Lemaître, Jean-Paul Gauthier Johel Miteran, Mohamed Atri 2008. "*Generalized Fourier Descriptors with Applications to Objects Recognition in SVM Context*", J Math Imaging Vis, Springer Science+Business Media 30: 43–71.
[12] Vapnik, V.N. 1998," *The Statistical Learning Theory*", Springer, Berlin.