Feature-Based Image Capturing and Recognition by using Active Contour Models

Chih-Hong Kao[1*], Ko-Min Mao[2], and Lung-Far Hsieh[3]

[1] Aeronautical Systems Research Division, Chung-Shan Institute of Science and Technology, Taiwan, R.O.C.
[2] PH.D Program in Management, Da-Yeh University, Changhua, Taiwan, R.O.C.
[3] School of Management, Da-Yeh University, Changhua, Taiwan, R.O.C.

ABSTRACT

The purpose of establishing a ship recognition system is to research and develop effective ship contour capture in the natural sea environment. We can develop a ship recognition system that is reliable and fast based on a database of ship images. We propose a recognition algorithm for ship images. This system utilizes gradient vector flow to capture the ship image contour first and calculates the geometric eigenvalue using this contour and Fourier descriptor. The eigenvalues are used to perform separated rough and detailed recognition. A graphic user interface is developed and the validity of the proposed technique is demonstrated using identifying images.

Keywords: eigenvalue, Fourier descriptor, ship recognition, gradient vector flow

運用主動輪廓模型於特徵基礎影像擷取與辨識

高誌鴻[1*]　毛格民[2]　謝龍發[3]

[1]中山科學研究院航空研究所
[2]大葉大學管理學院博士班
[3]大葉大學管理學院

摘　　要

建立船艦辨識系統的目的，是為了研發出能在自然海域中有效之船艦擷取技術，配合已建立好之船艦影像特徵資料庫，開發出一套可靠並且能快速辨識的立體船艦辨識系統。在本論文中我們提出了一個應用於船艦辨識的系統。這個系統先使用動態梯度向量流來擷取船艦影像的輪廓，從船艦影像輪廓計算出其幾何特徵值和傅立葉描述子，分別使用這些特徵值進行粗比對和細比對。本系統使用Matlab軟體建立了一個圖形界面，使用者可以透過這個界面，以互動的方式來進行船艦的辨識。

關鍵字：特徵值，傅立葉描述子，船艦辨識，梯度向量流

文稿收件日期 100.2.9;文稿修正後接受日期101.2.13; *通訊作者
Manuscript received February 9, 2011; revised February 13, 2012; * Corresponding author

I. INTRODUCTION

Image analysis is the process of finding, recognizing, realizing and establishing image models. The main purpose of using computer image analysis is to build machine vision ability comparable to that of humans. In the computer vision development field, object recognition is the most important technique, which allows the computer to obtain pure one direction object recognition. These skills are helpful in military applications, including moving target recognition and coastal surveillance. Computer vision recognition permits rapid response and all day reconnaissance. We establish a useful ship recognition system for effective and correct recognition.

The conventional approaches involve the using edge information in one form or another. The edges contain a great amount of the information in a scene. It is also known that the eye pays more attention to edges in an image. Traditionally robot vision systems have utilized the shape of objects for recognition [1]. Many of these methods explicitly exploit the features extracted from only the object's shape (i.e. lines, curves and vertices, also called geometric features) [2]. Approaches that explicitly take these factors into account in object recognition have been arranged as view-based or appearance-based object recognition methods [3].

A more general and extendible approach is view-based recognition in which a set of 2D views of each object is sampled at regular intervals in the viewing sphere and stored in an object-view database. Although view-based recognition is conceptually simple, robust and well-suited to object recognition, many of the techniques developed to date have been demonstrated on small, simplistic object databases requiring, in many cases, large processing training and/or recognition times. In this paper we introduce a new object recognition technique that is a natural extension of the view-based object recognition paradigm and capitalizes on the redundancy of both inter and intra-view features as well as on increased computational efficiency. The technique is also invariant to object scaling and rotation. We use a set of random transects or rectilinear line segments of 2D image views as our training data representation and an efficient matching operation that achieves high generalization recognition rates with significantly reduced computational times compared with current techniques.

Most approaches to view-based modeling represent each view as a collection of extracted features, such as extracted line segments, curves, corners, line groups, regions, or surfaces [4]. In contrast to the feature-based approaches, whose success requires reliable segmentation, a number of image-based view-based recognition systems have emerged [5]. Although these image-based approaches have been shown to work on natural objects, they are sensitive to (one or more of) illumination changes, scaling, image rotation, depth rotation, or occlusion.

The main purpose of this recognition system is to invent the techniques of the ship feature extraction, and establish a fast ship recognition system including the database of the ship image, and replace the traditional method which monitored by human capability. When we use the system to do the ship recognition, the first thing needs to do is the extraction of contour of ship from database. We use the gradient vector flow (GVF) method [6] in the ship recognition system. The system is automatic, but for the reliability, we also add the choice of hand-set operation. The system calculates the geometric character and boundary descriptor of the contour after the shape outline of ship outline of ship was determined. By using these values, we can recognize and establish the characteristic database. We define the template matching into rough matching and detail matching. We also show the results of those two matching methods. To accelerate the template matching process, we use the rough matching results as the basis for detailed matching. The user can do the interactive ship recognition and matching in the GUI, which is written in Matlab.

II. The APPLIED ALGORITHMS OF OUR SHIP RECOGNITION SYSTEM

Figure 1 shows the system architecture of our ship recognition system. From the beginning of image contour extraction, we calculate the image characteristic values; then make the rough

matching from the image database. After the rough matching, the detail matching was made to obtain exact results. In the following section, we will introduce the "every steps method" and the related algorithms.

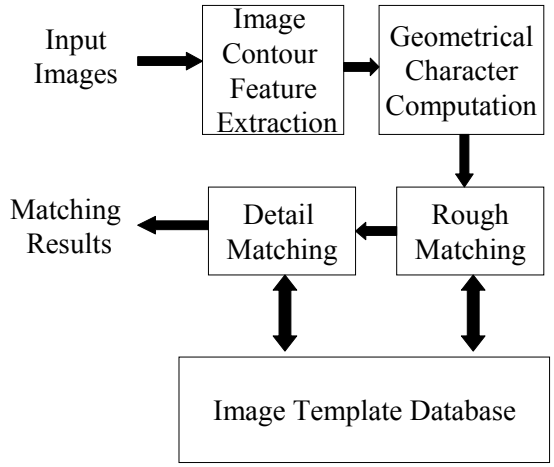


Fig. 1. Ship recognition system architecture.

2.1 Image contour extraction

We use the GVF method to obtain image contour feature extraction [6]. The Canny edge detection method is used to perform edge detection. The following section presents detailed discussion.

2.1.1 Edge detection

Canny edge detection [7] is one of the best edge detection methods in the world. Although its calculation computation processes are more complicated than other edge detection methods, the results are better than others. It can also eliminate incorrect edge points caught by noise. The Canny edge detection methods have the following three rules: (1) Good Detection, (2) Good Localization, (3) Single Response Constraint, which make it to be the best edge detection methods.

2.1.2 The active contour models (snake)

The Active Contour Model (Snake) was first proposed by Kass et al. [8]. A snake is an energy-minimizing curve defined within an image domain that moves under the influences of internal forces coming from within the curve itself and external forces computed from the image data. The internal and external forces are defined such that the snake will be attracted to an object boundary or other features within an image. Snakes are widely used in many applications, including edge detection, shape modeling, segmentation and motion tracking [9]. A two-dimensional dynamic contour called v(s) can be defined in terms of its x and y coordinates, which in turn are parameterized by s, the linear parameter:

$$v(s) = (x(s), y(s)) \quad 0 \le s \le 1 \tag{1}$$

Here we represent the position of the snake parametrically using v(s) = (x(s), y(s)). We define the model as a functional E to be minimized. This functional represents the energy of the model and has the form

$$E_{snake} = \int_0^1 \{E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))\} ds \tag{2}$$

E_{int} represents the internal energy of the snake due to bending, E_{image} serves as external force and is derived from image features. E_{con} serves to impose a smoothness constraint on the snake. E_{image} pushes or pulls the snake toward desired features such as edges. The external energy drives the active contour towards the desired points or boundaries within the image plane. The internal energy tries to keep the snake connected and consistent, preserving characteristics like steadiness, smoothness, tension and stiffness. The iterative minimization process of Eq. (1) dynamically deforms the parametric curve until a minimum is found that corresponds to the final active contour v(s) that better matches the desired boundaries within the image. As the internal and external energies are formulated, we deform the snake by minimizing (1).

2.1.2.1 Internal energy

Generally, the internal energy is the sum of the tension and stiffness (curvature) terms. The internal energy can be written as:

$$E_{int} = (\alpha |V_s(s)|^2 + \beta |V_{ss}(s)|^2)/2 \tag{3}$$

where α and β are weighting parameters that control the snake's elasticity and rigidity, $V_s(s)$ and $V_{ss}(s)$ denote the first and second derivatives of the curve V with respect to s. The first-order term $\alpha |V_s(s)|^2$ makes the snake behave like a string, whereas the second-order term $\beta |V_{ss}(s)|^2$ makes it behave like a rod. Large values of will increase the internal energy of the snake as it stretches more, whereas small values of α will make the energy function insensitive to the amount of stretch. Similarly, large values of β will increase the snake's internal energy as it develops more curves, whereas small values of β will make the energy function insensitive to curves in the snake. Smaller values of both α and β will place fewer constraints on the size and shape of the snake. Kass defined that $V_s(s)$ is the first order derivative of the control point V_i, and $V_{ss}(s)$ is the second order derivative of the control point V_i. Using the calculus definition, we can express $V_s(s)$ and $V_{ss}(s)$ as the following terms:

$$|V_s(s)| = \left|\frac{dv_i}{ds}\right| = |v_i - v_{i-1}| \quad (4)$$

$$|V_{ss}(s)| = \left|\frac{d^2 v_i}{ds^2}\right| = |v_{i-1} - 2v_i + v_{i+1}| \quad (5)$$

2.1.2.2 External energy

To make snakes effective we need energy functions that attract snakes to salient features in images as long as it takes on its smaller values at these desired features. External energy can be derived from the image, or user defined. It can also be a weighted combination of two or more energy functions. Given a gray-level image I(x,y), a typical energy function is designed as the following in order to let a snake be attracted by edge points:

$$E_{image} = E_{edge} = -|\nabla I(x,y)| \quad (6)$$

where $\nabla I(x,y)$ is the gradient image. When the active contour model is closer to the image edge, the $|\nabla I(x,y)|$ value is larger and the external energy is smaller.

2.1.2.3 Deformation

In the deformation process, the Active Contour Models compute the internal and external energy of every control point and search the minimum energy position around the control points. When all the steps for searching the minimum energy position around the control points are finished, the active contour model deformation is completed. For example, in Fig. 2, from V_{i-2} to V_{i+1} are the active contour model control points. Assume that V_i is a control point. Using V_{i-1}, V_i and V_{i+1} in the energy equation, we can obtain the internal energy E_{int} and external energy E_{image} of the control point V_i and then compute the energy around V_i. In figure 2, we use a 5x5 grid region around V_i, compute and compare the energy values and move the control point V_i to the minimum energy position, then complete the restraint on the control point V_i.

Substituting equations (3) and (6) into equation (2) to mend the energy equation, we get

$$E_{snake} = \int_0^1 \{(\alpha|V_s(s)|^2 + \beta|V_{ss}(s)|^2)/2 + \gamma(-|\nabla I(x,y)|)\}ds \quad (7)$$

where α, β, γ are the weighting parameters.

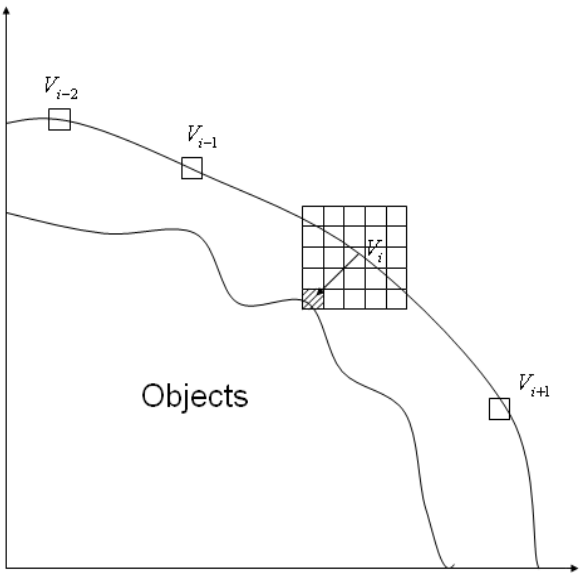


Fig. 2. The deformation of the snake-based contour models.

2.2 Gradient vector flow snake

However, the classical snake models also have limitations. First, most of these methods can only handle topologically simple objects. The topology of the structures of interest must be known in advance because the mathematical model cannot deal with topological changes without adding extra machinery. Second, the classical snake models are too sensitive to the initial conditions due to the non-convexity of the energy functional and contraction force, which arises from the internal energy term. Xu and Prince proposed GVF [6] to solve the capture range and concavity problems encountered in the traditional active contour model. The method they proposed is a new class of external forces for active contour, a dense vector field derived from images by minimizing certain energy functions. The minimization is achieved by solving a pair of decoupled linear partial differential equations that diffuse the gradient vectors of a gray-level or binary edge map computed from the image. We will call this an active contour that uses gradient vector flow as its external force as in a GVF snake. A GVF snake has a wide capture range, can progress into concavities and its insensitive to initialization. Its initialization can be inside, outside, or across an object's boundary. Unlike the classical snake models, a GVF snake does not need to know whether to shrink or to expand. Using the diffusion process the capture range is increase. We define the gradient vector flow field as the vector field F(x,y)=[w(x,y), z(x,y)] that minimizes the energy function

$$E_{image} = -\iint (\mu(w_x^2 + w_y^2 + z_x^2 + z_y^2) + |\nabla f|^2 |F - \nabla f|^2) dxdy \quad (8)$$

where $\nabla f = \left|\nabla C_{canny}(I(x,y))\right|$, $w(x,y) = \sqrt{w_x^2 + w_y^2}$, $z(x,y) = \sqrt{z_x^2 + z_y^2}$, w_x , w_y , z_x , z_y represent the differential gradient vector field. This variation formulation follows a standard principle, making the result smooth when there is no data. We see that when $|\bigtriangledown f|$ is small, the energy is dominated by the sum of the squares of the partial derivatives of the vector field, yielding a slowly varying field. On the other hand, when $|\bigtriangledown f|$ is large, the second term dominates the integrand and is minimized by setting $F = \bigtriangledown f$. This produces the desired effect of keeping F nearly equal to the gradient of the map edge when it is large, but forces the field to be slowly-varying in homogeneous regions. The parameter μ is a regularization parameter governing the tradeoff between the first term and the second term in the integrand. This parameter should be set according to the amount of noise present in the image. It has been shown that the first term in the integrand corresponds to an equal penalty on the divergence and curl of the vector field [10]. Therefore, the vector field resulting from this minimization can be expected to be neither entirely curl-free nor entirely divergent-free.

III. THE GEOMETRIC EIGENVALUES

The contour which gets by using the GVF method is not always closed. We should make a closing process while obtaining the edge contour. In the closing process, we use Bresenham's line method to connect the two separate points. We calculate the geometric eigenvalues after we obtain the detailed image contour. We use (1) the complexity of the object image and (2) the long-axis and short-axis object image ratio as the criterion for image recognition eigenvalues. The object image complexity is the length and area ratio of the target image. The long-axis to short-axis object image calculation will be described in the following section. All of these geometrical eigenvalues were for rough matching. We used the Fourier descriptor to perform detailed matching. Fourier descriptor normalization is needed for the ship recognition and will be described in the following section.

3.1 Calculation of the major and minor axes

An object's major axis lies in a straight line that can be represented by an equation in the form $x\sin\theta - y\cos\theta + \rho = 0$, where ρ is the distance from the origin to a straight line. θ is an angle between the straight line and *x*-axis. To find the major axis, we must solve the ρ and θ values for the line. In Fig. 3a, let (x, y) be the coordinates of the original image on the plane. C_x and C_y are the centriods of the desired

identifying object. x' = x - C_x and y' = y - C_y represent the coordinates on which the image at origin moves to the centroid later. A computation form of the centroid of the object may be written using the following form:

$$C_x = \frac{\sum_{(x,y)} \sum X}{\sum_{(x,y)} \sum 1}, \quad C_y = \frac{\sum_{(x,y)} \sum Y}{\sum_{(x,y)} \sum 1} \tag{9}$$

After obtaining the centroid points, three variables a, b and c can be calculated using

$$\begin{aligned} a &= \int\int_{I'} (x')^2 b(x,y)dx'dy' \\ b &= \int\int_{I'} (x'y')\ b(x,y)dx'dy' \\ c &= \int\int_{I'} (y')^2 b(x,y)dx'dy' \end{aligned} \tag{10}$$

where b(x, y) represents all of the points inside of the desired identifying object. I' is the image of the centroid moved to the origin. Solving the resulting equation $\sin 2\theta = \pm b/\sqrt{b^2 + (a-c)^2}$ again, and thus the sinθ and cosθ can be obtained. Substituting the θ and centroid coordinates (C_x, C_y) into xsinθ - ycosθ + ρ = 0, and ρ can be obtained. Substituting all of the points on the contour into xsinθ - ycosθ+ ρ = 0 again, we can obtain the two intersection points with the contour. As for the minor axis, if only the function vertical to xsinθ - ycosθ + ρ = 0 is obtained, the approach is the same as that for the major axis.

3.2 Fourier descriptors

In this section we provide a detailed description of Fourier descriptors. If the object contour is a closed curve C distributed in complex number space as shown in Fig. 3b, then a point surrounds the contour. Assume that this point travels with a velocity of v, the object contour can then be expressed as a function of u(t), where t is the time. In reality, the parameter t is a time parameter, or rather a length parameter following the contour arc length. The u(t) will be a periodic function, thus, presenting a value of L such that

$$u(t + nL) = u(t) \tag{11}$$

Thus, it is feasible that u(t) will be expressed in the complex number exponential form of Fourier series below.

$$u(t) = \sum_{-\infty}^{\infty} a_n \exp(\frac{jn2\pi t}{L}) \tag{12}$$

The Fourier coefficient is

$$a_n = \frac{1}{L}\int_0^L u(t)\exp(\frac{-jn2\pi t}{L})dt \tag{13}$$

For further simplification, we can reasonably assume a value of v so that L=2π, thus this formula can be simplified again as

$$a_n = \frac{1}{2\pi}\int_0^{2\pi} u(t)\exp(-j2nt)dt \tag{14}$$

and

$$u(t) = \sum_{-\infty}^{\infty} a_n \exp(jnt) \tag{15}$$

Obtaining the Fourier coefficient from the above form is not unique for a specified contour. These expressions are described below:

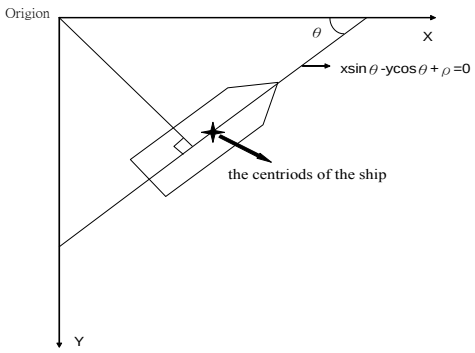


Fig. 3a. The centriods of the ship.

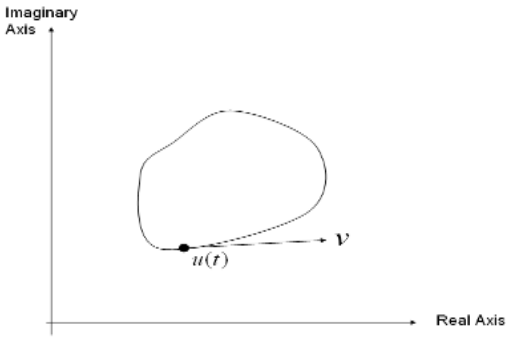


Fig. 3b. Schematic diagram of Fourier descriptors.

3.2.1 Initial point

The Fourier descriptors are closely related to the initial point selection, the Fourier descriptors have a deflection τ in different initial points; that is, as we mentioned before, a point surrounds the contour for each τ, which can generate a function of complex numbers u. When the time equals t, u can be expressed by u = u(t).

The initial point is changed by a small amount, and the initial point for sequence change from t = 0 into t = -τ.

We assume one specified Fourier descriptor for the contour curve as $a_n^{(0)}$, and a specified function exists as follows:

$$u(t) = u^{(0)}(t) \tag{16}$$

The other function set can be given by

$$u(t) = u^{(0)}(t+\tau) \tag{17}$$

where the superscript $^{(o)}$ is a specified contour function (act as a reference standard). The Fourier descriptor for this function is

$$\begin{aligned} a_n &= \frac{1}{2\pi}\int_0^{2\pi} u^{(0)}(t+\tau)\exp(-jnt)dt \\ &= \exp(jnt)a_n^{(0)} \end{aligned} \tag{18}$$

3.2.2 Translation

A translation moves every boundary point a constant displacement in a specified direction. As we assumed before, $a_n^{(0)}$ is a Fourier descriptor of the specified contour. Consider this contour using a vector of complex numbers, Z, to perform a translation. Using the inverse Fourier series, the contour function can be expressed by

$$u(t) = u^{(0)}(t) + Z = \sum_{-\infty}^{\infty} a_n^{(0)}\exp(jnt) + Z \tag{19}$$

Continuing the above form, we can find that Fourier descriptor of the contour after the translation is

$$a_n = \begin{cases} a_n^{(0)}, & \text{for } n \neq 0 \\ a_0^{(0)}, & \text{for } n = 0 \end{cases} \tag{20}$$

Except for a_0, all coefficients are the same as before the translation, without change. In other words, the translation to descriptor is without effect except n = 0.

3.2.3 Rotation

From the basic mathematical analysis, we can know that a rotation of the points of the complex plane about the origin by an angle ϕ obtains a function, which can be realized using the point multiplied by $e^{j\phi}$. $u^{(0)}(t)$ at every point all do the same for a specified contour function, so that the whole sequence rotates around the origin. The sequence after the rotation is

$$u(t) = \exp[j\phi]u^{(0)}(t) \tag{21}$$

Its Fourier descriptor is

$$a_n = \exp[j\phi]a_n^{(0)} \tag{22}$$

so the rotation is simply all coefficients multiplied by $e^{j\phi}$ simultaneously.

3.2.4 Scale change

In the same manner, the scale up and down for the contour can be used to the expression with a factor R. Hence the Fourier descriptor is expressed as follows:

$$a_n = R \cdot a_n^{(0)} \tag{23}$$

Inducting the above discussions, we can arrange a general form for the Fourier descriptor of the contour. After the rotation, initial point change and scale change, the expression of Fourier descriptors is obtained below:

$$a_n = R \cdot a_n^{(0)} e^{j\phi} \cdot e^{jn\tau} \tag{24}$$

It is noteworthy that the translation has an effect on the descriptor using different and known ways.

3.3 Image database comparison

After getting the geometry eigenvalues of the object, we can proceed to the image database comparison. This phase can be classified into two main steps: rough comparison and detailed comparison. The purpose of these two steps is to enhance the recognition efficiency. The detailed collation uses the outcome of the rough collation to make a comparison. Next, we will describe in detail the approach to the rough collation and detailed collation.

3.3.1 Rough collation

The rough collation mainly utilizes the edge complexity and the ratio of major axis to minor axis for the desired identifying object as eigenvalue to make a comparison.

(1) Edge complexity
Edge complexity = the circumference of the target/the area of the target

(2) The ratio of major axis to minor axis
The ratio of major axis to minor axis = major axis/minor axis

To adjust the relative proportions of two eigenvalues, we will multiply edge complexity by α, α = 5. We then calculate their absolute differences using the subtraction between two eigenvalues and the data in image database, adding the two absolute differences again. The smaller the obtained value, the nearer the desired identifying object. Thus, we have completed the course collection procedures. The images of the nine objects closest to the desired identifying object will be listed from the image database in the system.

3.3.2 Detailed collation

The detailed collation major uses the Fourier descriptors of the desired identifying object to make a comparison. As we mentioned in the earlier subsections, we can realize that there are variations in Fourier descriptors for the rotation, initial point change and scale change effects. The normalization consists of the following four major steps:

Step 1: Figure 4 shows a schematic diagram for identifying a desired object. The centroid of the object in the scheme is found first, the equation for the object's major axis using the foregoing method is then solved, thus the angle θ can be obtained.

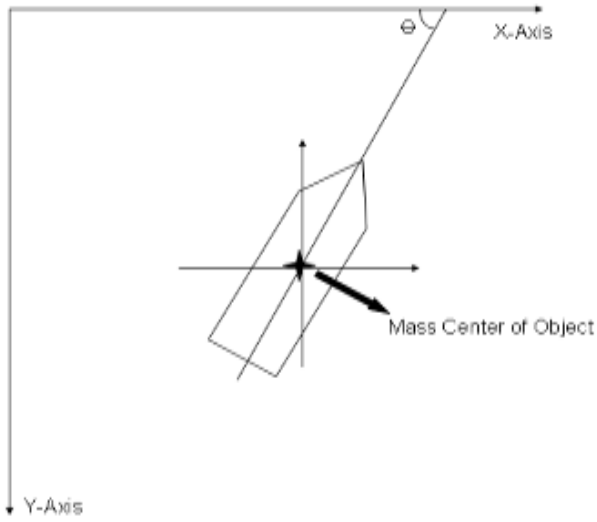


Fig. 4. The desired recognition object.

Step 2: By obtaining the angle θ**,** the object turns positive. Here, the positive turn means that the bow or stern of the ship is perpendicular to the imaging x-axis. The center of rotation is based on the object centroid as the center. The rotation method is uses the following formula:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{25}$$

Step 3: Begin scanning the image from the top and bottom, respectively. Find the two initial points, as shown in Fig. 5. Why we scan from the bow direction of the ship cannot be determined in actual object identification. Using the two initial points as a starting point again, we will find the contour of the whole object by "Contour Following", and the two arrays are formed with different arrangements.

Step 4: Because the arrays obtained consist of point coordinates and in this paper using the average length, we will re-sample 64 points from the original arrays to perform a discrete Fourier transform. Suppose the Fourier descriptors obtained are C_k, k = 0, 1,…, 63, let C_0 = 0 and find $\left[R = \sum_{i=0}^{63} |C_i|^2\right]^{1/2}$, let

$C_k^* = C_k / R$ again, k = 0, 1, 2,…, 63, we can get the normalized Fourier descriptors C_k^*.

While establishing the eigenvalues of imaging database in this article, Fourier descriptors of every image after normalization are stored in a database. So all one has to do is to calculate the mean square error between the Fourier descriptors of the desired identifying object normalized and eigenvalues saved in the imaging database of the rough collated results. The smaller the value, the closer to the desired identifying object, and the system will list the 9 closest images.

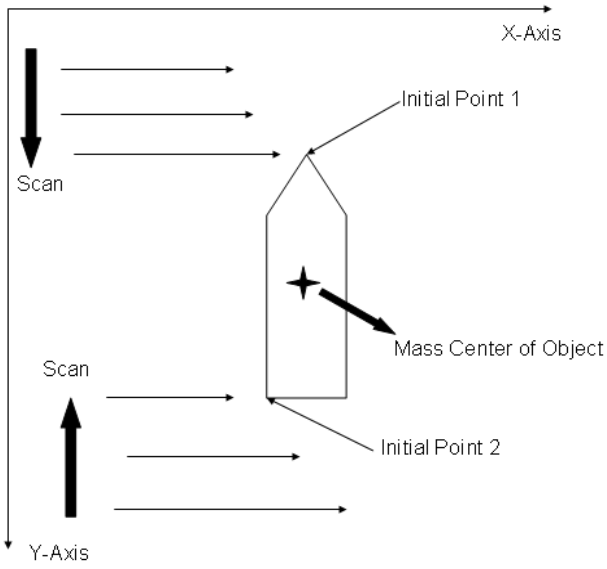


Fig.5. Find the initial points of the image.

3.4 System flowchart

A detailed system flowchart of the proposed recognition system is shown in Fig. 6. The acquisition steps in the system's object contour were divided into two processes: manual acquisition and automatic acquisition. The reason of dividing into two processes is mainly due to the different background complexity for each ship image. The image with complicated background has too much noise while performing the edge detection. That making the edge of the desired identifying object is difficult to be found. In this system the manual capture is greatly different from the automatic capture due to the edge of object was obtained by the different ways. In the step for automation capturing, as long as the user set the profiles. In the step for manual capturing, the user manually marked the edge of the desired identifying object at first, and set the profiles again. Figure 6 shows that the calculation step from geometry features is able to go back to input image step. Its function is like to reset the whole system. When the calculated results from the geometry features are unsatisfactory, the user is able to go back to the input image step to restart the whole processes.

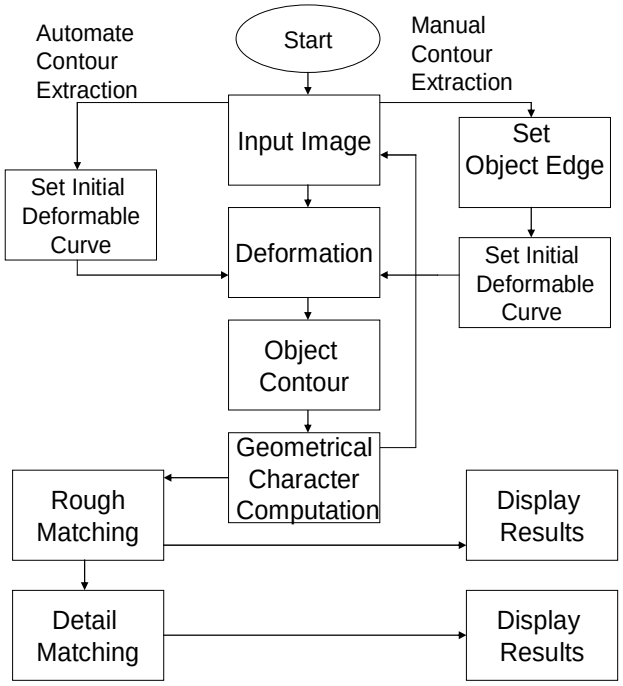
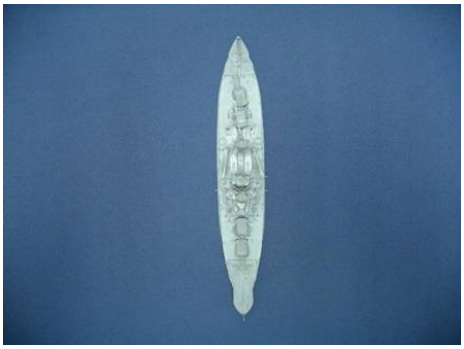


Fig.6. Flowchart of the proposed recognition system.

IV. RESULTS

4.1 Experimental database

The imaging database is used mainly on ship model images. Each ship model was captured an image per 45 degrees with a clockwise direction at bird's-eye view. Therefore, each ship model reached a total of eight kinds of rotation angles. Data were gathered from four types of ship models in this experiment, giving 32 images in the imaging database. Figure 7 shows the partial images of two types of ships.



(a)

(b)

(c)

(d)

Fig. 7. Arizona USA (a) rotate 180 degrees, (b) rotate 225 degrees, (c) rotate 315 degrees, (d) rotate 270 degrees.

4.2 Image interface usage

Using Matlab package software a graphic interface was set up as shown in Fig. 8. The user can utilize this interface to input the desired identifying image and select manual or automatic recognition ways to perform the identification.

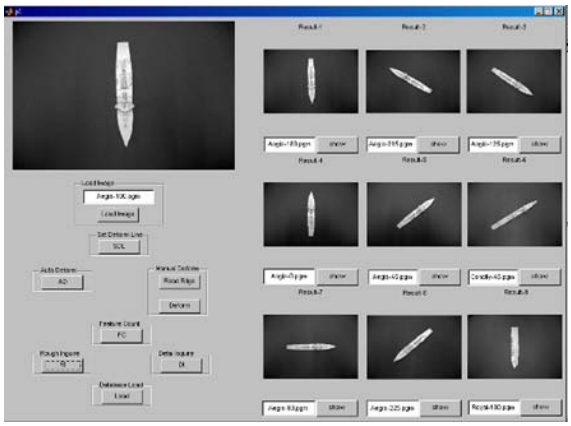


Fig. 8. Full view of the user's interface.

4.3 Experimental results

In our system, when the ship model image is captured from eight various angles, giving eight kinds of images with different angles for every ship model. In this experiment, we used all ship model angles to perform a comparison. One experimental result is indicated in Fig. 9. and Fig. 10.

Result was able to show the right image, this result also indicated that the eigenvalues extracted in the ship comparison is very efficient.

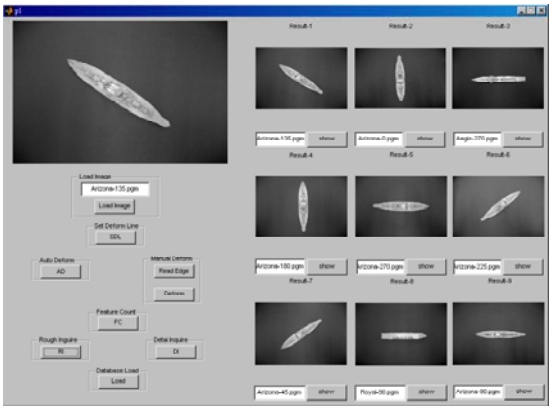


Fig. 9. Results of rough matching of Arizona USA rotated 135 degree.

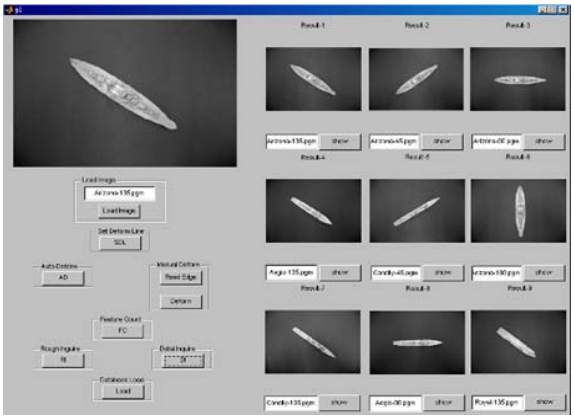


Fig. 10. Results of detailed matching of Arizona USA rotated 135 degrees.

We also computed the Fourier descriptor root mean square for the ship image database, and calculated the mean values for all the root mean squares for the ship image database. Table 1 shows the results. From Table 1, we can get that the characteristic value of every ship is the same with the same type of ship. This shows that the characteristic value of the ship image extracted by our algorithms is very effective.

Table 1. The mean values of all the root mean square for the ship image database

	The mean values of all the root mean square for the ship image database			
Target Image	Ark Royal	Arizona	Arleigh Burke	Connelly
Ark Royal rotated in 225 degree	0.0895	0.1213	0.1206	0.1279
Arizona rotated in 135 degree	0.0974	0.0819	0.1026	0.1007
Arleigh Burke rotated in 135 degree	0.0974	0.2049	0.0886	0.0959
Connelly rotated in 315 degree	0.1112	0.0852	0.1949	0.0799

V. CONCLUSIONS

We proposed an application for a ship recognition system. The proposed system is based on the GVF capturing the ship image contour, computing the geometric eigenvalues and Fourier descriptors from the ship image contour. These eigenvalues were used to perform rough collation and detailed collations. This system used the Matlab software to set up a graphical user interface. The user is able to perform ship recognition processing with an interactive approach through the proposed interface. From the ship recognition result, the system performance was quite satisfactory.

A further direction of this study will be to provide some improvements for this ship recognition system, including effective image contour capture, the establishment of an image database and the normalization of Fourier descriptors. We used shooting ship models as the image recognition database in these experiments. Such image background taken is a more simple approach, but the actual background is often very complicated for the image. Effectively capturing the image contour is the focus of future improvement. We will need to build a more extensive image database to verify the robustness of the recognition system. The method presented here used the ship bow as the Fourier descriptor normalization. Because the general bow has a sharp profile, importing a more generalized normal approach is also a focus of improvement.

REFERENCES

[1] Dyer, R. T. and Dyer, C. R., "Model-based recognition in robot vision, " ACM Computing Surveys, Vol. 18, No. 1, pp. 67-108, 1986.

[2] Chen, H. and Bhanu, B., "3D free-form object recognition in range images using local surface patches, " Pattern Recognition Letters, Vol. 28, Issue10, Pages 1252-1262, 2007.

[3] Weiss, I. and Ray, M., "Model-Based Recognition of 3D Objects from Single Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 2, pp. 116-128, 2001.

[4] Ullman, S. and Basri R., "Recognition by Linear Combinations of Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 10, pp. 992-1006, 1991.

[5] Schmid, C. and Mohr, R., "Combining Greyvalue Invariants with Local Constraints for Object Recognition," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 872- 877, 1996.

[6] Xu, C. Y. and Prince, J. L., "Snakes, shapes,

and gradient vector flow," IEEE Transactions on Image Processing, Vol. 7, No. 3, pp. 359-369, 1998.

[7] Wang, B. and Fan, S. S., "An improved Canny edge detection algorithm," Computer Science and Engineering, Vol. 1, pp. 497-500, 2009.

[8] Kass, M., Witkin, A., and Terzopoulos, D., "Snakes: Active Contour Models," International Journal of Computer Vision, Vol. 1, pp. 321-331, 1987.

[9] Leymarie, F. and Levine, M. D., "Tracking Deformable Objects in the Plane Using an Active Contour Model," IEEE Transactions on Pattern Analysis Machine Intelligence, Vol. 15, pp. 617-634, 1993.

[10] Gupta, S. N. and Prince, J. L., "Stochastic Models for DIV-CURL Optical Flow Methods," IEEE Signal Processing Letters, Vol. 3, pp. 32-35, 1996.