Employing Genetic Algorithms for qualitative Shapes Detection
S. M. Abdel- gaied
Department of Mathematics and Science, Faculty of Education, New Valley,
Assiut University 71516, EL kharaga, Egypt
sagaied@yahoo.com

Abstract
The localization of graphical primitives is important task in image processing. This paper discusses problems of automatic computer-based detection and localization of elementary shape in image. A genetic algorithm is used for this task. The using of genetic algorithm allows efficiently reduce time needed to scan a task state space. A modification of genetic algorithm for shape detection is presented. A review of basic algorithms for shape detection is given, followed by short genetic algorithm review. The specifics of using a genetic algorithm for presented task are presented. In addition, some experiment results are shown. The results are demonstrated on circle, ellipse and oblong detection.

Keywords: Genetic algorithms, Hough transform, Shape detection, Optimization.

1. Introduction
Many computer vision problems can be cast as optimization problems. In this paper, we study the applicability of genetic algorithm (GA) as a tool to solve an object location (template matching) problem. Given a template $T$ and an image $I$, it is desired to find the transformation that best matches $T$ to some parts of $I$. In the case that the contains multiple occurrence of (transformed) $T$, the algorithm is expected to find them simultaneously. In this paper we shall assume that we are dealing with an unknown affine transform, which is a six degrees of freedom search problem.

Graphical techniques are presented for estimating the Hough transform for detecting lines in noise image. In [1] some theorems with graphical techniques are introduced, and these theorems are very important for lines detecting with the Hough transform.

Many techniques for solving such object location problem both in 2D and 3D have been proposed such as Hough transform, simulated annealing [2 and 3] and Genetic algorithm [4] are robust but are computationally or memory intensive, or both. Without suitable speedup techniques

2. Related Work
2.1 Template matching
The template matching is approach to shape detection that can be used to locate known object in an image to search for the specific patterns. The best match is based on some optimality criterion, which depends on object properties and object relation [6]. Matched patterns can represent whole object of interest or some of it’s part. Because exact copy of the pattern of interest cannot be expected in the processed image (image is corrupted by noise, by geometric distortion etc.) some match criterion must be defined. The matching criterion can be defined in many ways from simple correlation up to complex approaches of graph matching [5, 7and 8]. The match based segmentation is computational very intensive, but the process can be faster if some performance improvement is found. The most widely used improvements are

- Test firstly image location with a high probability of math
- Detect mismatch before all corresponding pixels have been tested
- Match at lower resolution first

Matching criteria can be defined in many ways. The simplest one is based on summation of pixel difference between a pattern and the searched image. The minimal difference between these is searched. Let $f(x,y)$ be a processed image, and $h(x,y)$ be a searched pattern. Some simple matching optimality criteria describing a match between $f$ and $h$ located at a position $(x,y)$ are:

$$g(x,y) = \max_{i,j} |f(x+i, y+j) - h(i, j)|$$  \hspace{1cm} (1)

Another general matching criterion is correlation between a pattern and the searched image data. When $f$ is the processed image and $h$ is a searched pattern, the correlation is defined:

$$g(x,y) = f(x,y) \cdot h(x,y) = \sum \sum (f(x+i, y+j) - h(i, j))^2$$  \hspace{1cm} (2)
The location where \( g(x,y) \) is minimal identifies a position of the shape. A problem with using correlation is the enormous number of calculation that needs to be performed. The one of the most important strategies for reducing execution time is

### 2.2 Hough Transform

The Hough transform is a technique used to find parameterised shapes in an image. There are many variations of the Hough transform that are differed in efficiency and robustness under certain conditions. In this paper an introduction to Hough transform for lines, circles and ellipses is given. A major variation of these algorithms is also mentioned [9 and 10].

#### Lines

The standard equation for a line in x-y plane is

\[ y = mx + b \]

Let us define a new two-dimensional space, which we will refer as the parameter space. For each pixel \((x,y)\) in the image the set of all possible lines that go through this point have a parameter pairs \((m,b)\) given by linear equation

\[ b = -mx + y \]

For other pixels in image plane, next lines in the parameter space would be defined. If there are some collinear points in the image plane, so the corresponding lines in the parameter space are intersected to give the parameters of the line in the image plane.

\[ \rho = x \cos(\theta) + y \sin(\theta) \]  

For a pixel \((x,y)\) in the image plane the all possible lines that go though this point lie on a sinusoid in the parameter space. Collinear points in the image generate sinusoids, which also intersect in the parameter space, giving parameters of the line.

Unfortunately, for complex images the histogram is rather complicated and finding of the maximums can be very difficult. Another problem is a fact, that maximum in the parameter space don’t need correspond with a line in the image.

#### Circles

A circle can be described by the three parameters \((a,b,c)\) by equation

\[ c^2 = (x-a)^2 + (y-b)^2 \]

In general, a three-dimensional parameter space is required. For each point in the image plane, the every corresponding point in the parameter space must be set. Whereas in the previous section, the task was reduced to finding the intersections of curves in two-dimensional parameter space, for a circle location the intersection of surfaces in a three dimensional parameter space must be found. Unfortunately, this algorithm would be very computationally intensive and memory demanding. There are two major disadvantages of this approach - Three-dimensional accumulator requires a large amount of memory - Increment the accumulator elements and test every element if it is a maximum is computationally intensive It may be noticed that the simplification of this task can be also based on a-priory knowledge of some parameters of detected circles. If there is known circle radius, the task is reduced and only two-dimensional accumulator can be used. For general circle detection task, several variations on the Hough transform have attempted to solve this problem. Some of these are described in following sections. One used solution is to split the circle detection into the following two tasks:

- Finding a potential circle centers in the image.
- Finding a radius corresponding to each centre.

Let it be considered any point on a circle. Note that the centre of the circle lies on the normal to the tangent at that point. If the normal associated with point of circle is founded, so this would intersect the centre of the circle. By recording the normal in an accumulator that is defined over the image plane, the points of intersection can be found by searching for maximum of the histogram. When a noisy image is processed, a problem with normal determination can occur. The position of the local maximum of the histogram is used as the potential circle centre. The circle radiuses corresponding to potential circle centers can be found as a maximum in one-dimensional histogram of distance from potential circle centre to image points. Note that if more than one circle is centered on the same position, so the several histogram maximums correspond to valid radiuses.
**Ellipses**
An ellipse can be uniquely defined by a five parameters – the ellipse centre \((x,y)\), maximum and minimum radiuses of the ellipse \((a,b)\) and angle of the major axis \(\theta\). So a five-dimension parameter space should be used. Direct apply of the Hough transform is become very impractical – it has a large requirement to computation perform and very unreal requirement to memory space. Analogous to circle detection task, the memory and computational demands can be significantly reduced by reorganizing ellipse detection into two subtasks:
- Finding potential ellipse centers in the image
- Finding the remaining three parameters associated with each ellipse centre

To find ellipse centre the geometry feature of the ellipses can be used. In figure 2 is shown construction of centre intersection line. Let \(x_1\) and \(x_2\) be two points on an ellipse boundary. Let \(t\) be an intersection of these two tangents. Define \(m\) to be the midpoint of \(x_1\) and \(x_2\). The line \(tm\) is than passed through the centre of the ellipse.

![Figure 2 construction of line though centre of ellipse](image)

The line \(tm\) is recorded in two dimensional histogram defined over the image plane by incrementing each histogram elements that the line intersects. Note that it is sufficient to record only mid-line originating from \(m\) and oriented away from \(t\). An example of the histogram is shown in figure 3.

![Figure 3 Hough transforms – (a) Image plane (b) histogram](image)

The second part of the task estimates the remaining three parameters. For each ellipse centre the image is translated so that ellipse centre is located at the origin. The remaining three parameters can be obtained by defining a three-dimensional parameter space and using standard Hough transform. However, in this case only a global maximum of the histogram is required. The adaptive Hough transform can be used to reduce memory requirement and computational demands in this case. After determination of the remaining three parameters associated with ellipse, it is necessary to perform a final validation step. To determine whether an ellipse exist in image, the number of pixels lying on (or near) the ellipse must be counted. If there is enough pixels, the ellipse in image is located.

3. Genetic Algorithm for Shape Detection
Genetic algorithm (GA) is an adaptive method that mimics the metaphor of natural biological evolution. It is an optimization technique that operates on population of individual solutions. Each individual solution (also called string or chromosome) represents a proposed solution of the solved problem. The theories of the natural selection are applied to this population to find solution. Next generation of the population is obtained by applying mutation, reproduction and selection stochastic operators to the population. With these operators, the population of solutions comes to (pseudo-)optimal solution of the problem. The quality of every individual solution is assessed by objective (fitness) function. This function is problem dependent and it is used to propagate good solutions into next generation. Generally, the genetic algorithm advantages are simple using, low memory demands, using of simple computation algorithm and ability of parallelism. Disadvantage of the genetic algorithm is non-deterministic work time and non-guarantee finding of best solution.

3.1 Edge detection
Such as most shape detection algorithms, we assume that input image was transformed by using any edge detection algorithm. For transformed image \(E(x,y)\) we define:

\[
\begin{align*}
E(x, y) &= 1 \quad \text{for edge point}, \\
E(x, y) &= 0 \quad \text{otherwise}
\end{align*}
\]

Where \(x\) and \(y\) are horizontal and vertical coordinates in the transformed image.

3.2 String Encoding
Let \([a_i,b_i]\) represents of horizontal and vertical co-ordinates of object shape. Example is in figure 4, where the highlighted points can represent a circle.

Let us define a general transform function \(T\) which transform the co-ordinates of the searched shape \([a_i,b_i]\) to the image plane co-ordinates \((x_i,y_i)\)

\[
(x_i, y_i) = T(a_i, b_i, p)
\]

Where \(p\) represents parameters of transformation. The transformation \(T\) can include for example a shape movements, spin, stretch etc. For our shape detection task we consider, that the values \([a_i,b_i]\) are constant and
some instance of shape (given by transformation, e.g. by miscellaneous values of \( p \)) is searched. The transformation parameters is used the natural value encoding.

![Circle Points](image)

**3.3 Fitness Function**

Now let us define fitness (objective) function so that it gives maximal value when the points \((x_i, y_i)\) represent a shape in the image \( E \). It is accomplished if we define function \( F \) as follows in equation 8:

\[
F(p) = \sum_{i=0}^{N-1} E(T(a_i, b_i, p))
\]

Maxima of function \( F \) over all parameters \( p \). Although definition (8) is formally correct but there is a problem with the search for the maximum. Since this fitness function has a spike of maximum only when the transformed shape is accurately equal to shape in image \( E \), no optimization technique (including GA) can’t be successfully used to search for this maximum. This problem can be eliminated by redefine of fitness function \( F \). Let us define a fitness function \( F \) as follows:

\[
F(p) = \sum_{i=0}^{N-1} R(T(a_i, b_i, p))
\]

Where \( R \) is defined as:

\[
R(x, y) = \sum_{\forall q, w} E(x + q, y + w) - \frac{1}{d} \sqrt{q^2 + w^2}
\]

Where \( d \) is a positive constant.

The fitness function takes into account nearest edge points, so as the bigger length of nearest edge points were penalized. Penalization is determined by length constant \( d \). The choice of the constant \( d \) depends on the image structure. Generally better convergence of the solution can be achieved when the constant \( d \) is large, but too large value is not suited for real (and noised) images. Disadvantage of using the too large value is a possibility of detection of non-existing circles with radius smaller or comparable to the length constant \( d \), which can happen especially for images damaged by noise.

**3.4 Selection**

Steady-State selection is used for proposed algorithm. Main idea of this selection is that big part of chromosomes should “survive” to next generation. This is one of the simplest versions of genetic algorithm. One step of this algorithm consists of three sub steps:

- Mutation – randomly selected strings are mutated
- Crossover – two randomly selected strings are crossover (theirs parts are swapped)
- Selection – the worst string (solution) from population is replaced by the best string.

**3.5 Mutation**

Several mutation algorithms were tested in proposed genetic algorithm. As a basic method the binary inversion was used. For chosen arithmetic string encoding, an arithmetic mutation was tested. The individual string values are (independently with some probability) pretreated. Perturbation is chosen so as enhance a probability of small variation. Next method is proposed especially for shape detection task. It is partially shape dependent. Proposed mutation method is based on assumption that a shape instance (given by string) has one shape point located correctly. The string (e.g. coefficients of shape transformation) is then randomly changed so as to accomplished condition that chosen point is located on the same place also for new string (transformation). This is demonstrated in figure 5, where the point marked by arrow is chosen to be stable. The grey circles are potential circles after mutation (one is stretched, the other is rotated).

![Circle Mutation](image)

For polygonal shapes (or shape parts) a mutation can be extended to operation of movement in direction of derivation of shape in selected points. The mutation rate (probability of mutation in one step of GA) can also be adapted in according to GA convergence. The proposed approach increases a mutation rate when the best fitness does not improve for some time.

**3.6 Crossover**

Two types of crossover can be used for chosen string encoding – binary crossover or arithmetic crossover. As a binary crossover a 2-point crossover is used. This crossover operator uses two randomly chosen crossover points. Strings swap the segments between these two points. Arithmetic crossover independently with some probabilities swaps individual values from string.

**4. The Overall Algorithm**

The hybrid system of the global search and the local search procedures has become more popular since it preserves both advantages of jump out from the local optima and the effectiveness of search. The execution plan of the proposed method is a hybrid structure consisting of a GA and a local search algorithm. In the GA phase, parameter candidates with fitness values
above a given threshold are copied into a candidates list. Then, in the local search phase, the true candidates which do exist in the image are filtered out from the candidate list.

4.1 The GA Phase
In each generation, the strings with fitness values above a specified threshold are copied into a candidate list. When a new candidate is to be inserted, the candidate list is checked whether there already exists a member which is too close to the new candidate in the parameter space; if so, retain the better one according to their fitness values. In addition, the Elitist model [11], which preserves the best string obtained so far within the current population, is applied to enhance the search. After the GA phase terminates, the candidate list will contain the strings which are close to the regions where the true parameter strings of the existing features reside. The proposed GA procedure is described as follows:

Input: the size of the population \( P \), the candidate threshold \( T_1 \) and the maximal number of generations \( G \).
Output: a candidate list \( L \).
Step 1: Randomly generate the initial population of \( P \) chromosomes.
Step 2: Evaluate the fitness values of all strings and copy the best string \( S_b \) of the initial population in a separate location.
Step 3: Apply the proposed reproduction, crossover and mutation operators to generate the next population.
Step 4: Evaluate the fitness values of all strings in the current population.
Step 5: Add the strings with fitness values greater than \( T_1 \) to \( L \). Retain the better one if two members of \( L \) are too close.
Step 6: Compare the best string \( S_c \) of the current population with \( S_b \). If \( S_c \) has a higher fitness value, then replace substitute \( S_b \) with \( S_c \).
Step 7: Replace the worst string \( S_w \) of the current population by \( S_b \).
Step 8: Go to Step 3 until there are \( G \) generations.
The value of \( T_1 \) can be determined by picking, for example, the top 5% of the individuals in each generation.

4.2 The local search Phase
To find out the true parameters of the existing features, each member of the candidate list should look around its neighborhood and choose the best direction to move. The idea is based on the observation that the fitness values of true candidates can be increased more drastically than those of false candidates by a local search algorithm. The neighborhood of the candidate is determined by two heuristic searches. One is improving the candidates by changing the values of successive parameters, the other is based on copying the parameter values from other candidate members. The details of the local search procedure are given as follows.

Input: the candidate list \( L \) produced by the GA phase, a small amount of adjusting value \( v \) and a threshold \( T_2 \).
Output: the parameters of all detected features.

Step 1: Select each member from \( L \), do Steps 2 and 3.
Step 2: For each parameter of the current member, do the following:
  1. Change the parameter value by each of the three ways: adding \( v \), subtracting \( v \) and copying the parameter value from other members.
  2. If the fitness is improved, retain the best one as the new parameter value; otherwise, stay unchanged.
Step 3: Repeat Step 2 if any parameter value of the current member has been changed.
Step 4: Check any two members of \( L \). If they are too close, retain the better one according to their fitness values.
Step 5: Output the members with fitness values greater than \( T_2 \).

The local search algorithm is successful in separating the true candidates from the candidate list. The reason is as follows. The true candidates will receive large gains in improving their fitness values by adjusting the parameter values locally, while the false ones are scarcely improved by looking to their neighbors. The value of \( T_2 \) can be determined automatically by clustering the fitness histogram into two groups.

5. Experimental Results
We implement our method on PC with Pentium-1600 MHz CPU inside to test robustness of the proposed method. In the GA phase we use a population of 50 strings and run for 20 generations. Figure 6(a) shows a synthetic image (200 x 200) of a circle. The edge image of this image is shown in Fig 6(b). The genetic algorithm is applied on image 6(b) to get Figure 6(c) shows the detected circle superimposed on the original image. Figure 7(a) shows the original image for recognition. It contains three floppy disks in different orientations. Figure 7(b) is the template used for corresponding edge detected image and Figure 7(c) is the template used for recognition the experiment is terminated when the fitness of the 10 best fit chromosomes has no chance in 100 generations. This termination criterion is used for this experiment. Figure 7(d) shows the recognition Results of the 3 best fit chromosomes. The experimental results show that GA can locate all the three floppy disks in the image correctly and simultaneously.

Figure 8 (a) and 9 (b). Show the image with the presence of 75% and 97% outliers, respectively. Figure 8 (b) and 9 (b) show the line extracted by GA. The extraction works well up to approximately 97% outliers.
We applied the algorithm on real image. Figure 10(a) shows a real image (240 x 240) of Egyptian coins. The edge image of this image is shown in Figure 10(b). The genetic algorithm is applied on Figure 10(b) to get Figure 10(c) which shows the detected circle superimposed on the original image.

6. Conclusion
We present an efficient way of fitness evaluation based on the geometric symmetries. We also propose specific genetic operators for the application. The candidates with fitness value greater than a threshold are added into a candidate list. In the local search phase, the true candidates are filtered out from the list by iterative local search algorithm. We conclude the distinct features of the proposed method from the Hough-based methods and previous GA-based methods as follows:

- The proposed method does not use tangents of edge points which are hard to be correctly estimated.
- The proposed method does not decompose the parameterization process into multiple stages, so no propagation error is produced.
- The time complexity and the storage complexity of the proposed method are linear to the number of the parameters of the detected features.
- Fast fitness evaluation and little memory storage
- No need of existence verification process after the true candidates is output.

Figure 6: (a) The original image; (b) edge detection and (c) The detected circle superimposed on the original image.

Figure 7(a) the original image; (b) the edge detection image; (c) the template; (d) the objects that are recognized by GA; and (e) the recognized objects onto the original image.

Figure 8 Detection of a line with 75% outlier
Figure 9 Detection of a line with 97% outliers

(a) Image space
(b) Extraction line

Figure 10 A real image of Egyptian coins: (a) The original image; (b) edge detection and (c) The detected circles superimposed on the original image.

(c)

7. References


Sahar M. Abdel-Gaied received the B.Sc. degree from Assiut University, Egypt, in 1994, the M.Sc. degree from South Valley University, Egypt, in 1998, and the Ph.D. degree from Assiut University Egypt in 2004. She is an Lecture at Assiut University-Egypt. Her research interests include information retrieval, genetic algorithms, image processing and networks.