International Journal of Engineering, Science and Mathematics
Vol. 7, Issue 4, April 2018,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Kernal based Integration of Hough Transform and Genetic Algorithm for easy optimization of Two Dimensional Diagrams

Dr. G. S. Katkar[1] , V.R. Nikam[2]

Abstract

.

Abstract: This paper describes the detecting object based on kernels. Objects are detected by background differencing. We have Integrated with Genetic Algorithm, to analyze the flow of pattern and object. The Diagram integration gives the high cutting edge result for hough transform. Low contrast levels can present problems, leading to poor object segmentation and fragmentation, particularly on older analogue tracking system. The object detection can be done through the kernel, which is interacting with the given image by morphology and result will be identified. The model-free tracking or detecting algorithm described in this paper addresses object fragmentation and the masking induces spatially-smooth similarity. The morphological operators are applied with kernel for detecting the specific objects. The this kernel can thus the predict the border of object, thus counting on the iteration basis.

[1]Head, Dept of Computer Science, Taywade College, Koradi, Nagpur
[2]Research Scholar ,Taywade College, Koradi, Nagpur

.

1 INTRODUCTION OF HOUGH TRANSFORM

Morphological based object detection with in image is difficult task. One way to simplify the problem is to change the grayscale image into a binary image, in which each pixel is restricted to a value of either 0 or 1. The techniques used on these binary images go by such names as: blob analysis, connectivity analysis, and morphological image processing. The foundation of morphological processing is in the mathematically rigorous field of *set theory*; however, this level of sophistication is seldom needed.

2. Mathematical Representation of HOUGH TRANSFORM

Most of the mathematical formalism and notations are adopted from Serra (1982)[1]. To understand procedure, certain basic mathematical morphological transformations are detailed along with list of symbols and notations. Mathematical morphology based on set theoretic concepts is a particular approach to the analysis of geometric properties of different structures. The main objective is to study the geometrical properties of a natural feature represented as a binary image by investigating its microstructures by means of "kernels", following Serra's concept (1982)[1]. It aims to extract information about the geometrical structure of an object by mathematical morphological concepts. In this, specific object detection features are subjected to transformations by means of another object called kernel. The main characteristics of kernels are, shape, size, origin and orientation. Different kernels can characterize the topological characteristics spatial distribution. According to Matheron's (1975)[2] approach, each image object is assumed to contain its boundary, and thus can be represented by a closed subset of Euclidean space. In addition many kernels are represented by a compact subset of E, so that constraints which correspond to the four principles of the theory of mathematical morphology such as invariance under translation, compatibility with change of scale, local knowledge and upper semi-continuity will be imposed on morphological set transformations (erosion, dilation, opening and closing).

Dilation, *erosion*, *opening* and *closing* are simplest quantitative morphological set transformations. These transformations are based on Minkowski[3] set addition and subtraction. The Minkowski set addition of two sets, *M* and *S*, is shown in Eq. (1).

$$M \oplus S = \{m + s: m \in M, s \in S\} = \bigcup_{s \in S} M_s \quad (1)$$

M and *S* consist of all points *c* which can be expressed as an algebraic vector addition $c = m+s$, where the vectors *m* and *s* respectively belong to *M* and *S*.

The Minkowski set subtraction of *S* from *M* is denoted as

$$M \ominus S = (M^c \oplus S)^c = \bigcap_{s \in S} M_s \quad 2$$

Let *M* be a binary image where the pixels with 0s are marked with a dot for a better legibility. Kernel *S* will be moved from top to bottom and left to right by applying the criterion of erosion principle to achieve shrinking. When the rectangle, *S* is centered on one point of the frame of the image *M*, then it will be truncated and only its intersection with the shape is kept.

The discrete binary image, *M*, is defined as a finite subset of Euclidean 2-dimensional space, R^2. The geometrical properties of a binary image possessing set (*M*) and set complement (M^c) are subjected to the morphological functions. From geometrical point of view morphological dilations and erosions are defined as set transformations that expand and contract a set. The morphological operators can be visualized as working with two images. Each kernel has a designed shape that can be thought of as a probe of the main feature. The three morphological transformations involved in this study are dilation to expand erosion to shrink, and cascade of erosion-dilation to smoothen the set.

Dilation : Dilation combines two sets using vector addition of set elements. If *M* and *S* are sets in Euclidean space with elements *m* and *s*, respectively, $m = (m1, ..., mN)$ and $s = (s1, ..., sN)$ being *N* -tuples of element co-ordinates, then the dilation of *M* by *S* is the set of all possible vector sums of pairs of elements, one coming from *M* and the other from *S*. The dilation of a set, *M*, with kernel, *S* is defined as the set of all points such that *Sm* intersects *M*. It is worth mentioning here that

Minkowski addition and subtraction are akin to the morphological dilation and erosion as long as the kernel(S) is of symmetric type. As long as the kernels are symmetric (S =) Minkowski's addition and subtraction are respectively similar to morphological dilation and erosion.

$$M \oplus \hat{S} = \{m : S_m \cap M \neq \varnothing\} = \bigcup_{s \in S} M_{-s} \tag{3}$$

This operation enlarges the objects and neighboring particles will be connected.

Erosion: The erosion of an image, M, with kernel, S is defined as the set of points m such that the translated Ss is contained in M. It is expressed as:

$$M \ominus \hat{S} = \{m : S_m \subseteq M\} = \bigcap_{s \in S} M_{-s}$$

where -S = {-s: sS}, i.e., S rotated 180^{0} round the origin. (4)

2.1 Theory of kernels

Kernel is a microstructure of the set with which transformations are to be performed. Broadly these structuring elements are categorized as symmetric and asymmetric types. Moreover, 1-D and 2-D structuring elements can also be defined at will. The role of structuring element is to unravel the hidden morphological properties of the set that is transformed by structuring element according to a particular rule. This functions as an interface between objective and subjective.

2.3 Property of iteration

To generate a large size erosion or dilation, the dilation as well as erosion can be iterated. Instead of using a larger structuring element, with the use of smaller structuring element repeatedly one will get the same effect, although not all dilations with a large structuring element can be so decomposed. is the symbol for the dilation. Carrying out dilation twice is represented as

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \quad \oplus \quad \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \quad = \left| \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix} \right.$$

Fig. 1 Dilation by 3×3 Kernel

The above diagrammatic representation is represented as the following mathematical notation

$$s \oplus s \oplus s \oplus \ldots \oplus s = S_n \tag{5}$$

Decomposition of structuring elements: Several other structuring elements include circle, segment, bi-points, triangle. Moreover, these structuring elements can be defined at will to unravel hidden properties of image under investigation. This choice is based on the type of result to get the purpose of the transformation. The transpose of kernel is shown in Fig. 4.8

$$\begin{array}{ccc} 1 & 1 & 1 \\ & 1 & \end{array} \qquad\qquad \begin{array}{ccccc} & 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 & 1 \end{array}$$

$$S \qquad\qquad\qquad \hat{S}$$

Fig.2 The triangluler and its transpose

S_x will be the kernel centered in x and S, the symmetric of S relative to its center. *Sh* will be translated set by vector h- as shown in Fig. 2. Prior to understand the impact of other types of kernel on the structure it should be noted that there are several types of line kernel or 1-D kernel. Following are the line kernel with direction 0^0, 45^0, 90^0, 135^0, 180^0, 225^0, 270^0, and 315^0 that are available on a square grid, and 0^0, 60^0, 120^0, 180^0, 240^0, and 300^o are on hexagonal grid.

$$\begin{array}{c} 1 \\ 1 \\ 1 \end{array} \qquad \begin{array}{ccc} & & 1 \\ & 1 & \\ 1 & & \end{array} \qquad \begin{array}{ccc} 1 & 1 & 1 \end{array} \qquad \begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \end{array}$$

Fig 3 The line kernel and its transpose

These 1-D kernels can be used to shrink or expand the size of the objects in the given direction and may discard them. These line-kernels are centered on the right side for the erosion, and on the left side for the dilation. Many other objects can be generated as shown in the following representations with the 1-D kernels.

$$\begin{array}{ccc} & & 1 \\ & 1 & \\ 1 & & \end{array} \oplus \begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \end{array} = \begin{array}{ccc} & 1 & \\ 1 & 1 & 1 \\ & 1 & \end{array}$$

$$\begin{array}{ccc} & & 1 \\ & 1 & \\ 1 & & \end{array} \oplus \begin{array}{ccc} 1 & 1 & 1 \end{array} \oplus \begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \end{array} = \begin{array}{ccccc} & 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & \end{array}$$

Fig 4 The Object generation with 1-D kernel

2.4 Multi-scale operations

As already mentioned, the simplest morphological set transformations are *erosion*, *dilation*, *opening* and *closing*. Since, the structuring element is of symmetric type both Minkowski's subtraction and addition are similar to morphological erosion and morphological dilation respectively (Sagar 1996, Sagar *et al*. 1998a, 1998b). The dilation followed by erosion is called closing transformation. Cascade of erosion-dilation is called opening transformation. These cascade transformations are idempotent (Serra 1982)[1]. However, these transformations can be carried out according to the multiscale approach (Sagar *et al*. 2000)[3]. In the multiscale approach, the size of the structuring template will be increased from iteration to iteration. The opening and closing operations are *idempotent*; i.e.,

$$\{[(M \oplus S) \oplus S] \oplus S\} \oplus S = (M \oplus S) \oplus S, \quad (6)$$

and similarly closing (Serra 1982[1], Sagar *et al*. 2000, 2003)[4]. But a variation will be identified while performing either opening or closing as multiscale operations/cycles, i.e.,

$$\{[(M \oplus S) \oplus S] \oplus S\} \oplus S = (M \oplus S_2) \oplus S_2 \quad (7)$$

Theoretically, the above expression is true. Other way of performing opening is the right side notation. In this subsection opening and closing operations are performed on the basis of cycles. As shown in Eq. (5), the size of structuring element will be changed as follows; if *S* is of size 3 x 3 pixels, this means that instead of using a larger structuring element, it is often possible to use a smaller one repeatedly to get the same effect.

The diagrammatic representation is shown as square type of structuring element of two sizes and the cumulative effect through addition. According to the expression (6), two consecutive erosions and dilations can be respectively represented as

$$(M \ominus S) \ominus S = M \ominus S_2, \text{ and } (M \oplus S) \oplus S = M \oplus S_2 \quad (8)$$

The transformations from the field of mathematical morphology such as *erosion*, *dilation*, and *opening* discussed so far are used to extract morphological skeletal network (*MSN*).

Segmentation is to distinguish objects from background. For intensity images (ie, those represented by point-wise intensity levels) four popular approaches are: threshold techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation methods. Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. Because spatial information is ignored, however, blurred region boundaries can create havoc.

3. INTRODUCTION OF GENETIC ALGORITHM (GA)

Genetic algorithms are a stochastic search algorithm, which uses probability to guide the search. It was first suggested by John Halland[5] in the seventies. Over the last twenty years, it has been used to solve a wide range of search, optimization, and machine learning. Genetic algorithms are a class of parallel adaptive search algorithms based on the mechanics of natural selection and natural genetic system.

It can find the near global optimal solution in a large solution space quickly. It has been used extensively in many application areas, such as image processing, pattern recognition, feature selection, and machine learning. It is a powerful search technique that mimics natural selection and genetic operators. Its power comes from its ability to combine good pieces from different solutions and assemble them into a single super solution. Genetic algorithms are initial population of solution called individuals is (randomly) generated, the solutions are evaluated. The algorithm creates new generations of population by genetic operations, such as reproduction, crossover and mutation[7]. The next generation consists of the possible survivors (i.e. the best individuals of the previous generation) and of the new individuals obtained from the previous population by the genetic operations.

The best source of information about Gas is Holland's adaptation in natural and artificial systems, Holland uses terms borrowed from mendelian genetics to describe the process:each position in the string is called a gene. The possible values of each gene are called alleles. A particular string is called a genotype. The population of strings also called the gene pool. The organism or behavior pattern specified by a genotype is called a phenotype. If the organism represented is a function with one or more inputs, these inputs are called detectors.

4. FLOWCHART OF WORKING OF GENETIC ALGORITHM

The concept of spiral genetic algorithm (SGA) is incorporated which decreases the search area of the subsequent GA's that is proportional to the minimal cost function of the previous GA and thus decreasing the cost to provide better convergence.

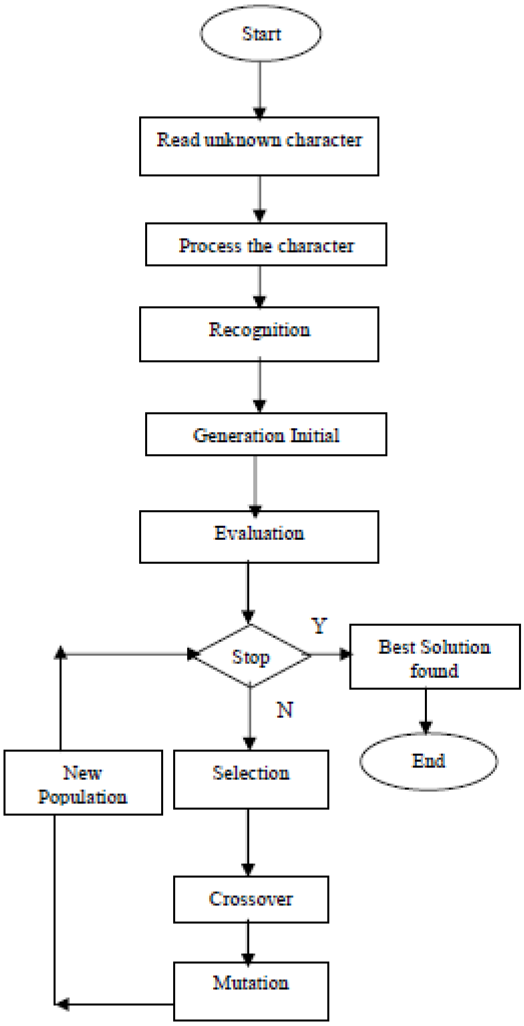


Fig. 5 Circle Life of the Genetic Algorithm

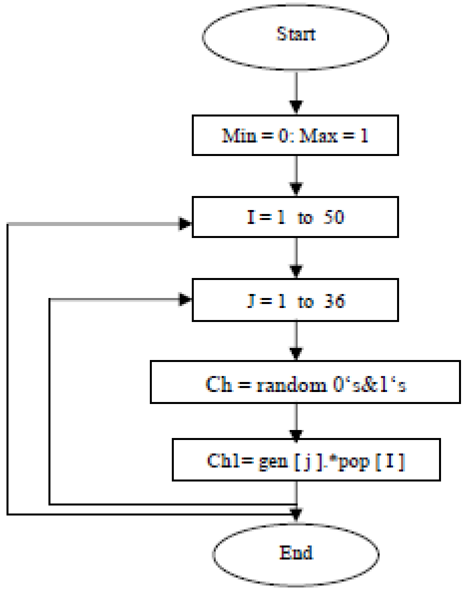


Fig. 6 Evaluate: calculate the fitness of each individuals in the population according to the differences between them and the vector

.

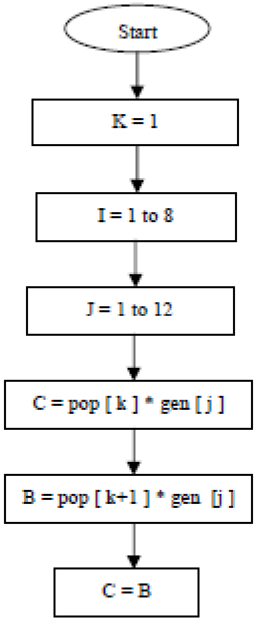


Fig. 4.13

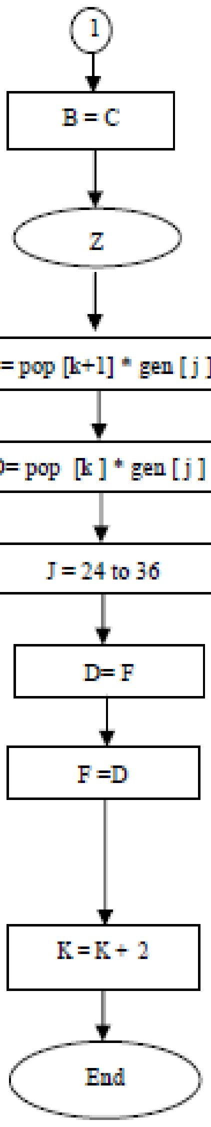


Fig.7 Mutation: it takes the new individuals created by cross2x procedure and randomly changes 15 genes from it, the algorithm goes on:

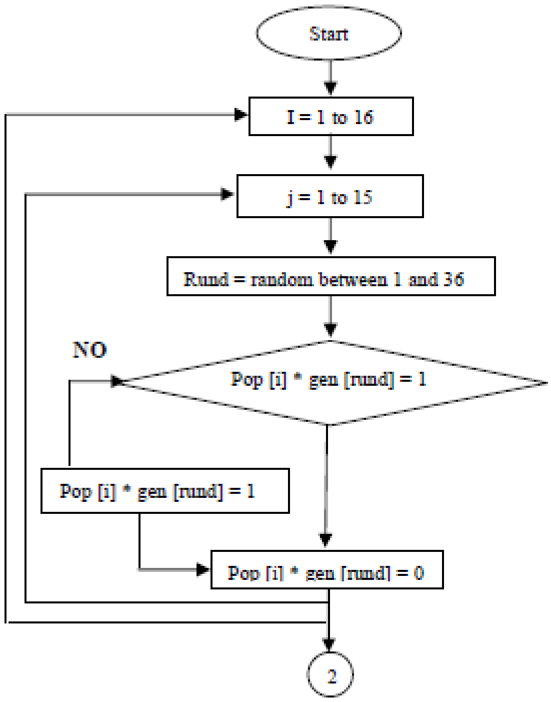
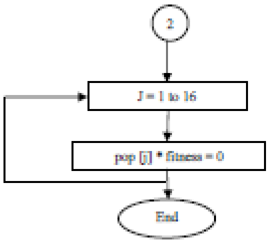


Fig.8 Replacement: it take five element randomly from the population, choose worst one and then replace it with a new individuals ,the algorithm goes on:



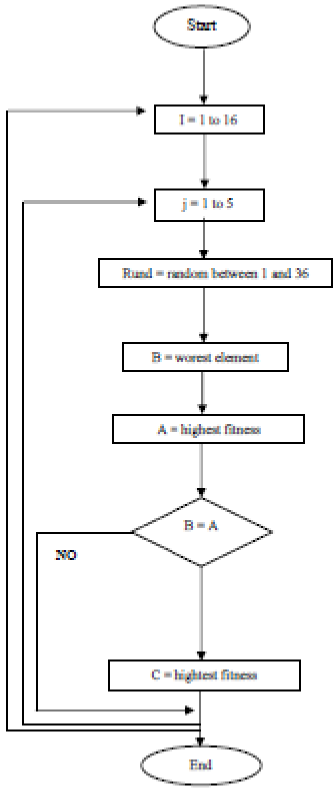


Fig 9 Checkstop : stop the system if best solution found which it's fitness should be less than 3 or the generation no. is more than 500

5. MODEL-BASED RECOGNITION

In model-based recognition problems, a model of an object undergoes some geometric transformation that maps the model into a sensor coordinate system (say, an image plane or a cylindrical coordinate system from a 3D scanner). The development of efficient algorithms for identifying such transformations is central to many model-based recognition systems.

There are a number of other approaches to model-based recognition which employ non-trivial geometric algorithms, and which often draw explicitly on results from computational geometry. The affine hashing method (Lamdan and Wolfson 1988)[8] uses a redundant representation of a set of points in order to locate that point set under an affine transformation, in the presence of extraneous data points. The underlying idea is to use each triple of points in the model as an affine basis, and rewrite the other model points in terms of each basis. In order to recognize an object, triples of image points are selected, and for each triple all remaining image points are expressed in terms of the basis. When a correct basis is found, this will result in affine invariant coordinates that are the same (up to sensing error) as one of the encodings of the model.

Several researchers have developed recognition methods that explicitly account for sensor errors. These methods make considerable use of results on arrangements from computational geometry. Most of these methods represent each point or line segment in an “image" set as a polygonal region (say, the Minkowski sum of the image set with a box). The matching problem is then to search for transformations bringing each point or line segment of the model into such a polygonal region in the image. These problems can be structured as sweeping arrangements, using algorithms from computational geometry (Atherton *et al*. 1987, Baird 1985)[10][11]. A different approach to model-based matching problems involves the development of cost functions for measuring the difference between two sets of points and line segments under various transformations. The applied methods developed in the vision community are provably good approximation schemes for solving the combinatorial problems that were originally investigated in the computational geometry community.

6. KERNEL BASED OBJECT DETECTION INTEGRATED WITH GENETIC ALGORITHM

The detection of an object from the scene is the main purpose of this paper. Binary image has been identified from the given image and then try to apply the morphological operators on it. When we introduce the morphological operations it tries to remove the noise and give the objects based on the kernels. We try to use several kernels like square, rectangle, circle, octagon, triangle and ramous. Any object which is in binary, *opened* with the particular kernel will produce the same kernel after n-1 iterations, so the kernels are playing main role on the object which is used. The kernels are influenced in the object and try to act, which helps to identify the real one.

This procedure includes systematic use of multi-scale opening and simple logical operators. The multi-scale openings and closings are useful smoothing filters because they preserve the shape and location of vertical abrupt signal discontinuities; further, the definition of scale in the openings is identical to the spatial size of geometrical objects. When we use multi-scale openings in the image with kernels, the resultant object is look like the kernel after n-1 times. If we exceed one more openings, the image will be vanished, because the kernel and n-1 time opened image is same, by using this approach, it is vary compact to identify the object based on kernels. In our examples we try to use only few kernels but it is also possible to use more kernels, which will give more specific aspects of object detection based on kernels.

7. IMPLIMENTATION OF WORKING OF HOUGH TRANSFORM (EXPERIMENTAL WORK)

7.1. INPUT DATA (ORIGINAL IMAGE) and (KERNEL)

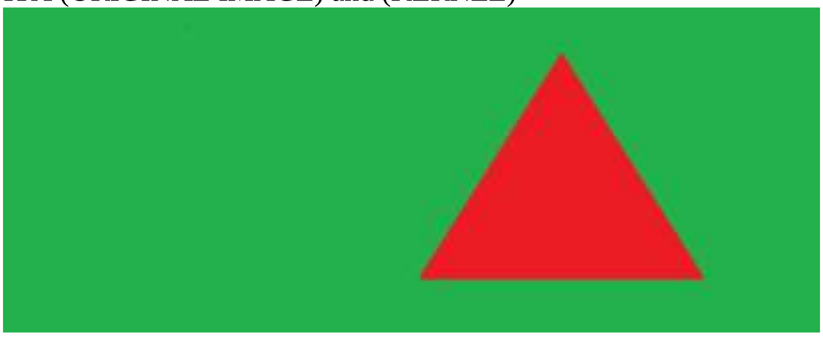


Fig A. Original Single Experimental data

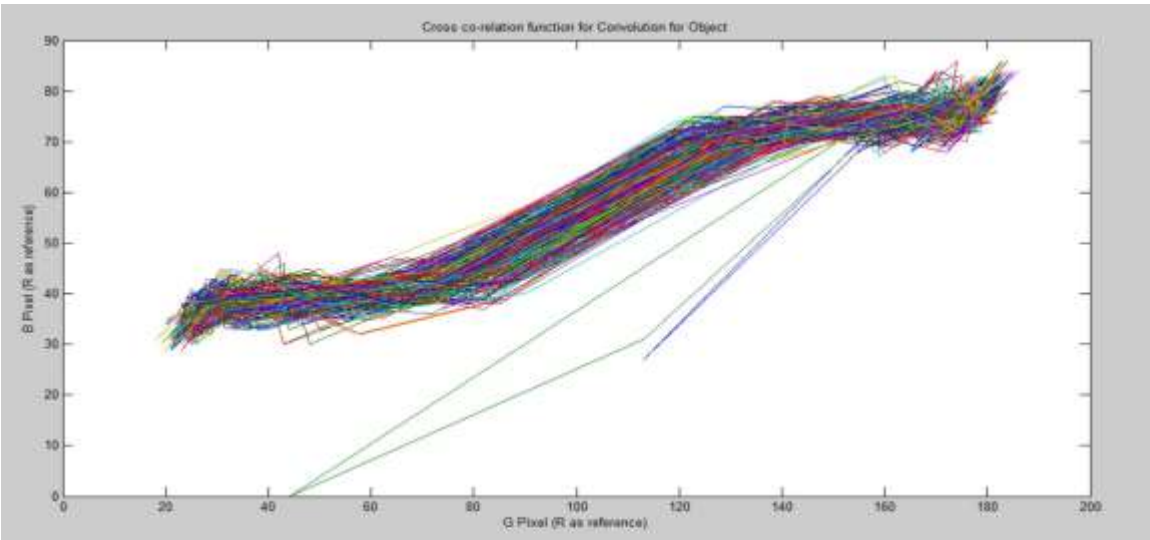


Fig.10:- Single Image Kernal for Triangle Diagram as shown in fig. A

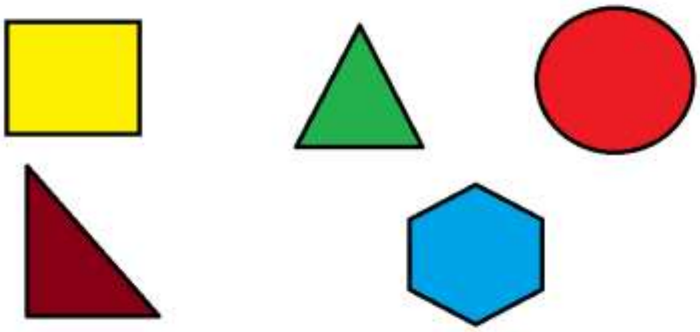


Fig B. Original Single Experimental data

Fig.11, shows the kernel of the fig.A, this result is integrated with hough transform mathematics and we try to superimpose the linear curve on the 2-D diagram (fig 4.21). Similarly we have taken group of 2-D diagram fig.B, and tried to detect them with the hough transform, by integrating with the kernel(fig.4.20) of group, as shown in fig 4.22.

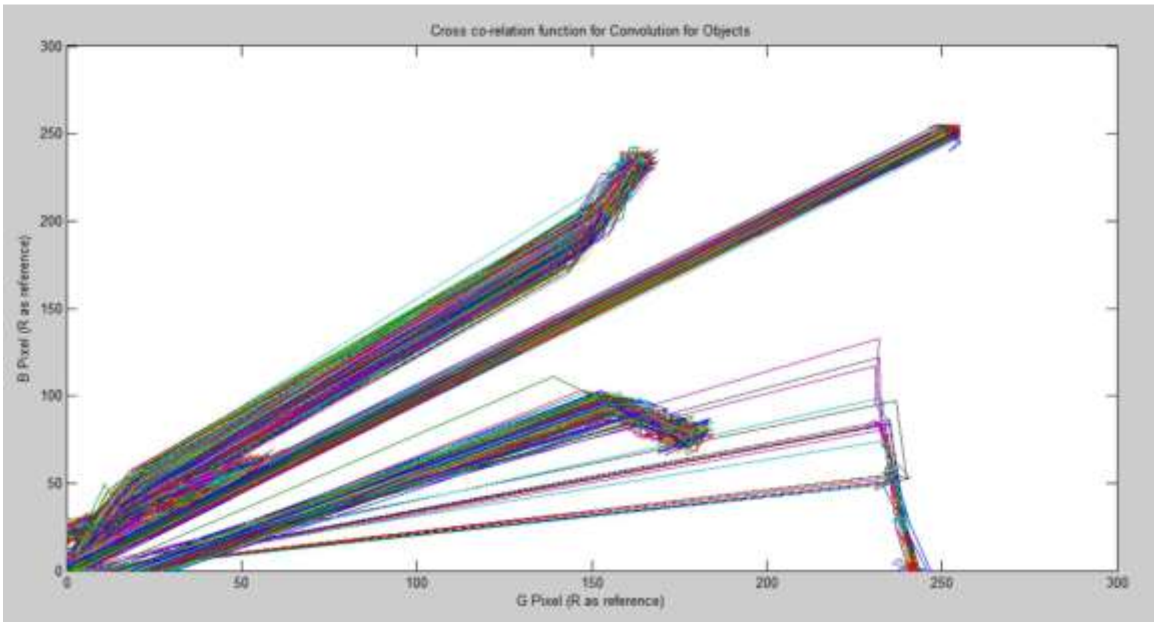


Fig.12 showing:- Multiple object kernal, as shown the magnitude of kernel is directly proportional to the number of objects.

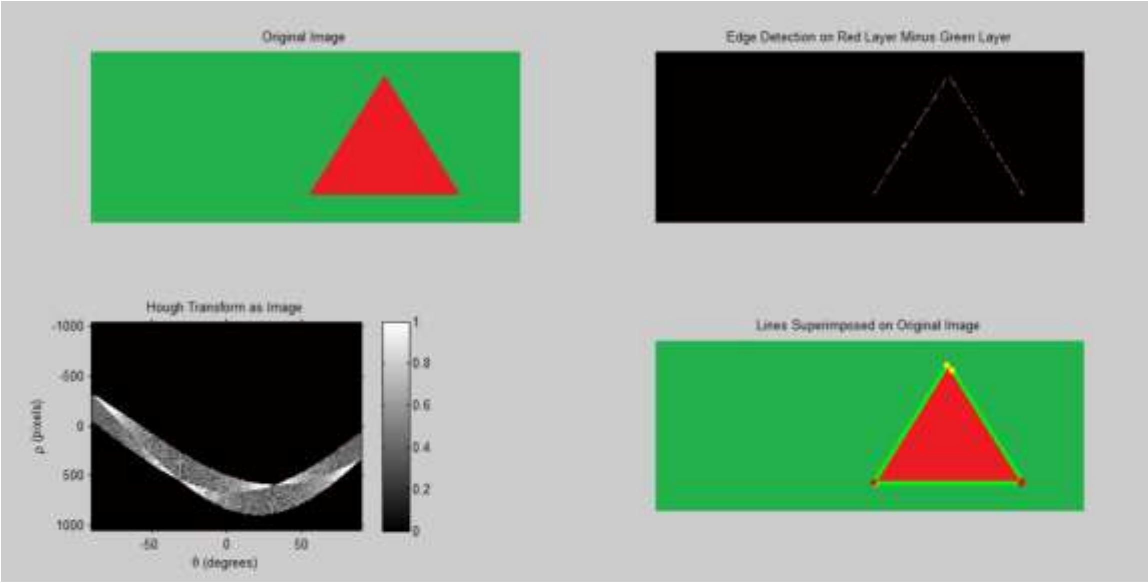


Fig.13:- Test Result for Hough Transform for single Image

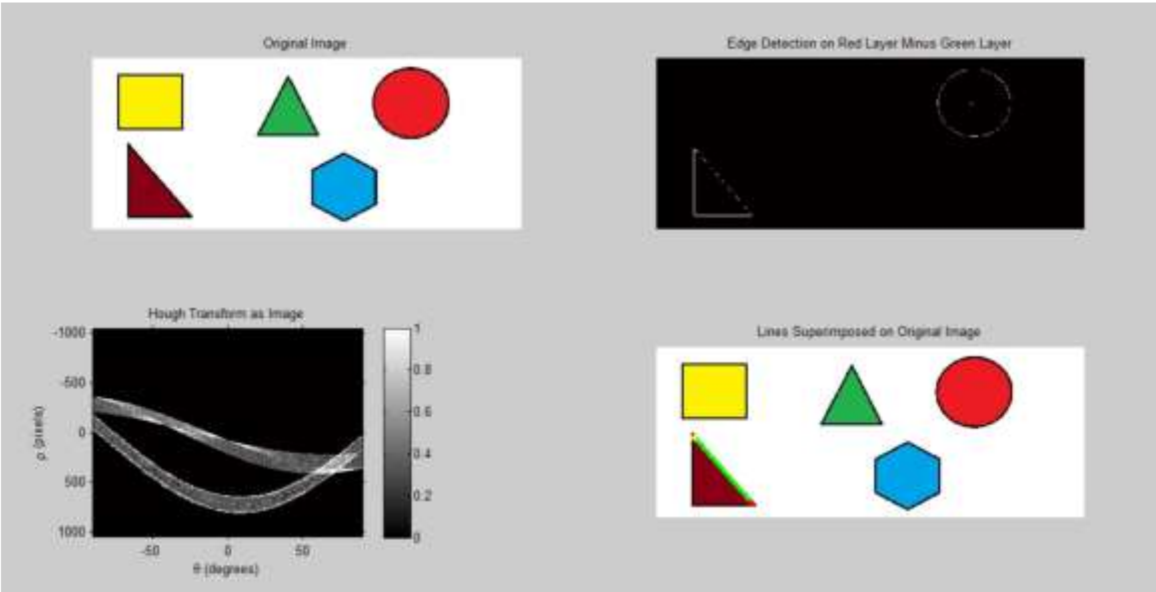
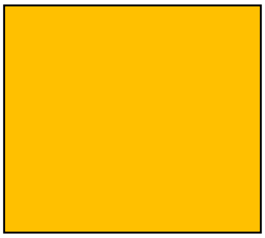
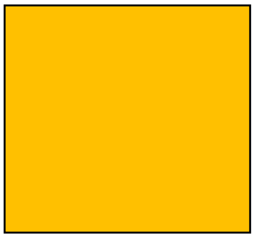
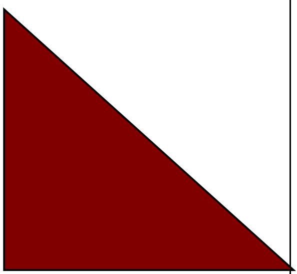
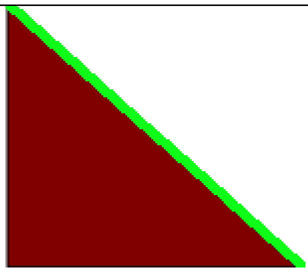
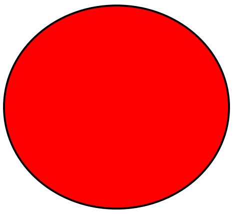
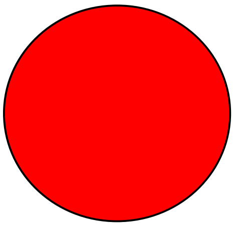
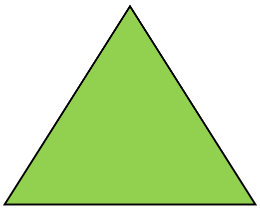
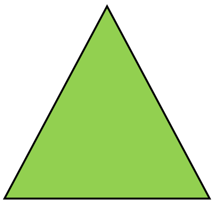
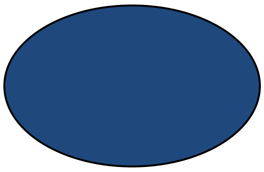
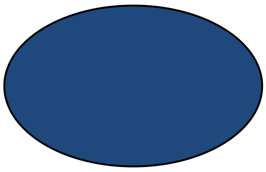
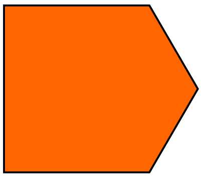
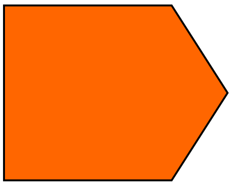
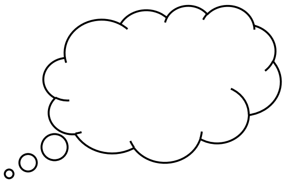
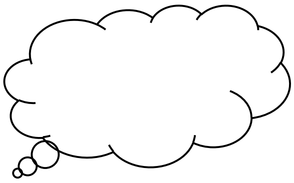
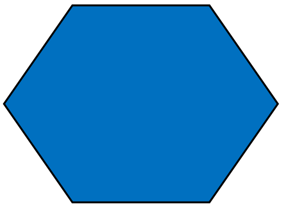
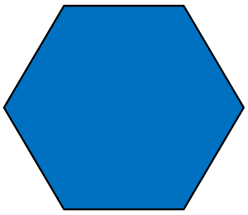
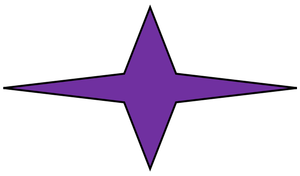
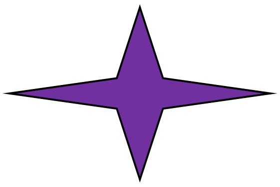
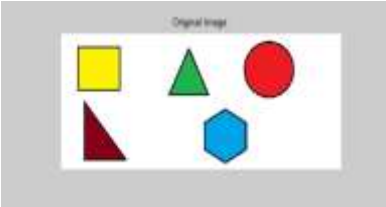
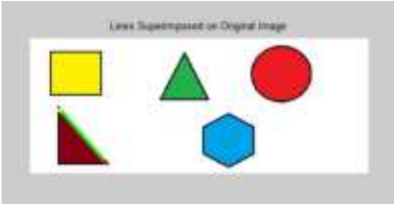


Fig.14:- Test Result for Hough Transform for Multiple Image

7.2. MULTIPLE IMAGE DETECTION USING HT WITH EFFICIENCY

Input Diagram	Detected Diagram	Efficiency
		0%
		44%
		0%
		0%
		0%

Input Diagram	Detected Diagram	Efficiency
		0%
		0%
		0%
		0%
		0%
		9%

Here the Basic difference between the Hough Transform for single and multiple image detection is given, we can analyze that multiple object detection can't be implemented using Hough transform alone, we have to integrate it with some soft computing technique to give us the better result for analysis.

8. IMPLIMENTATION OF WORKING OF GENETIC ALGORITHM (EXPERIMENTAL WORK)

Through Genetic Algorithm, we have first analyze the blue, green and red plane, then we have superimposed this all layer to form one layer, like we do for mutation technique in Genetic Algorithm, and superimposed plane is convolve with kernel of that image we can easily detect the object plus we can also count how many object are present in particular image.

8.1. INPUT DATA (ORIGINAL IMAGE) and (KERNEL)

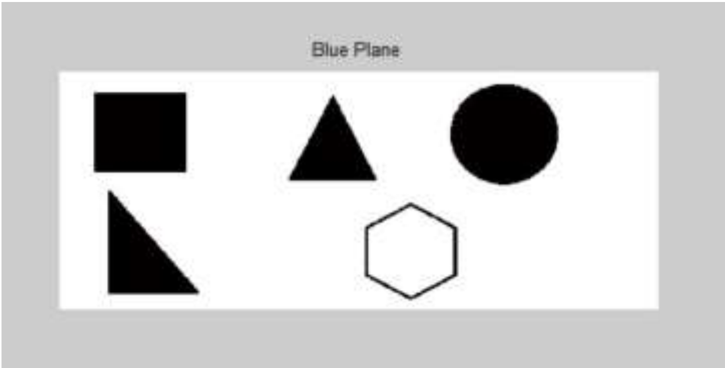


Fig.15:- Showing Blue plane as a input for Genetic Algorithm

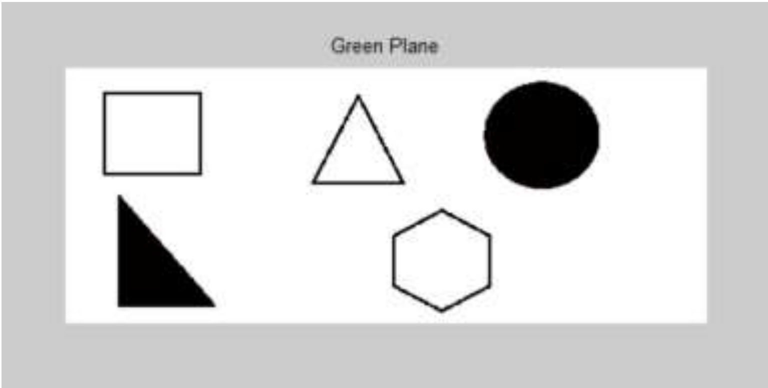


Fig.16:- Showing Green plane as a input for Genetic Algorithm

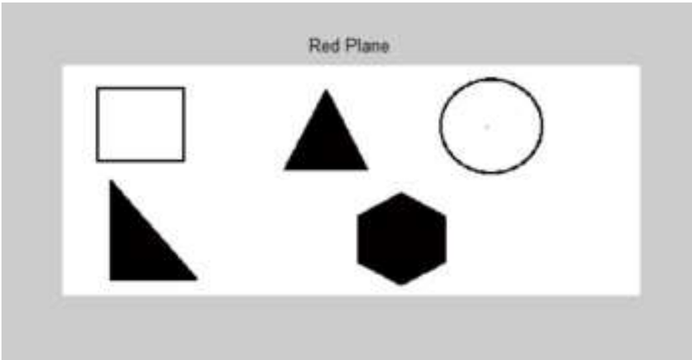


Fig.17:- Showing Red plane as a input for Genetic Algorithm

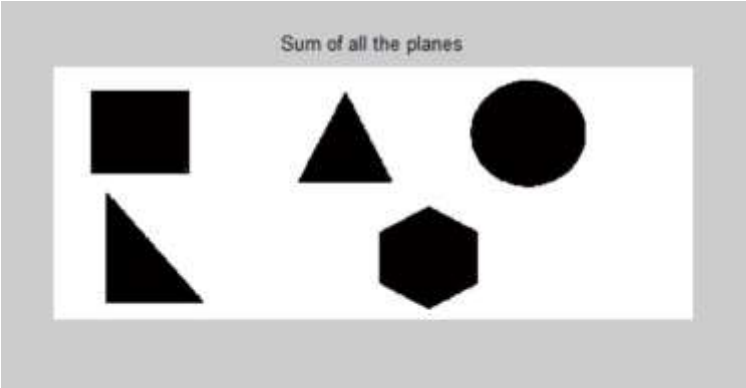


Fig.18:- Showing Sum of all plane as a input for Genetic Algorithm

We also have edge superimposition data, which define the layer for dynamically analyzing the edge and pixel migration. This result seems to be very important, as this is taken as our starting initial parameter for iterations.

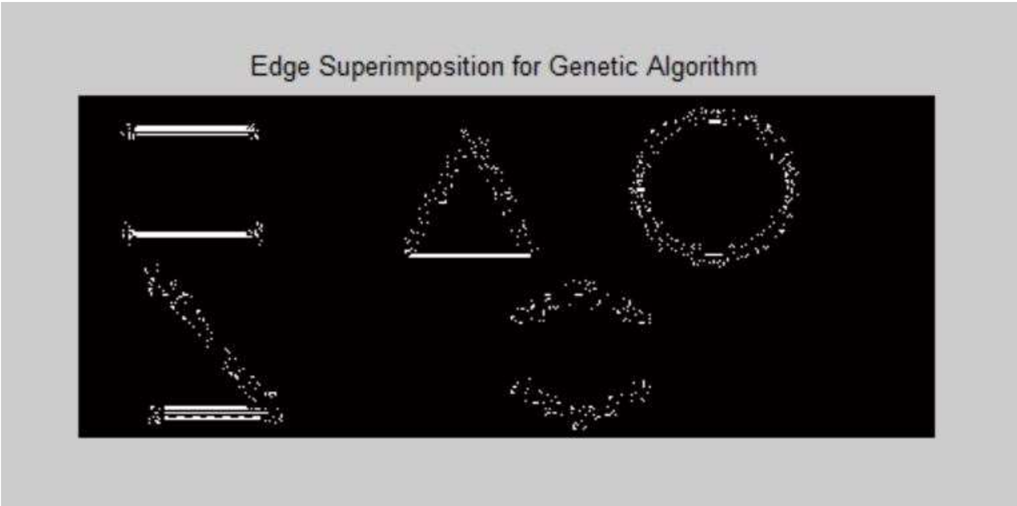


Fig.19 showing edge superimposition for GA

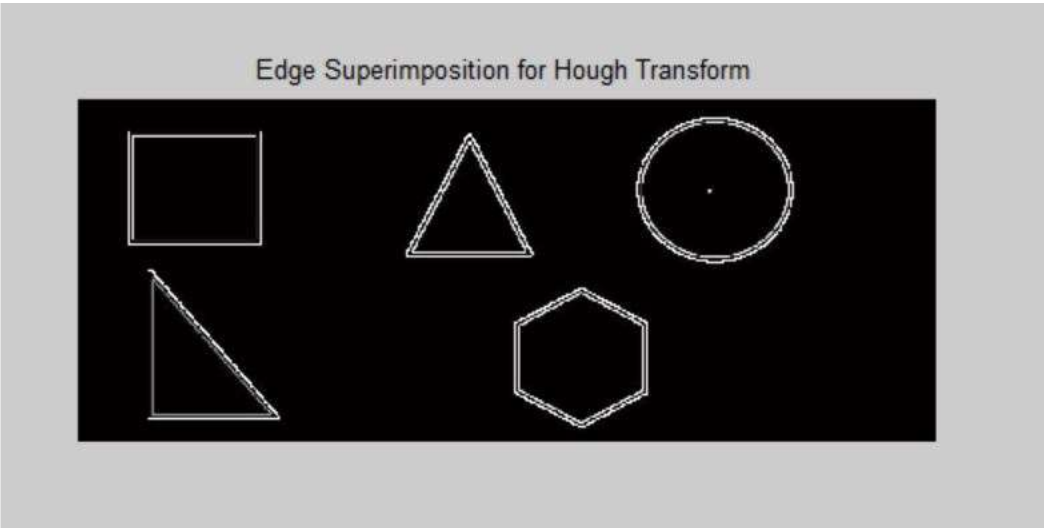


Fig.20 showing edge superimposition for Hough

This is convolve with kernel of the image, and when integrated with Genetic Algorithm, we get priority analysis (figure) and then after few iteration, we get final result of Genetic Algorithm as shown in the figure:-

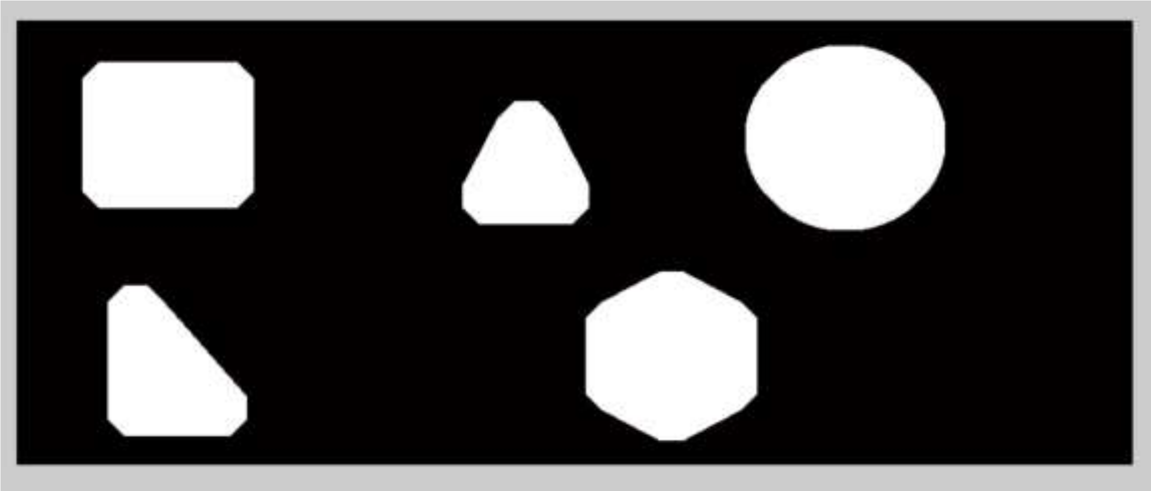


Fig.21 :- Showing the priority analysis just before the GA result.

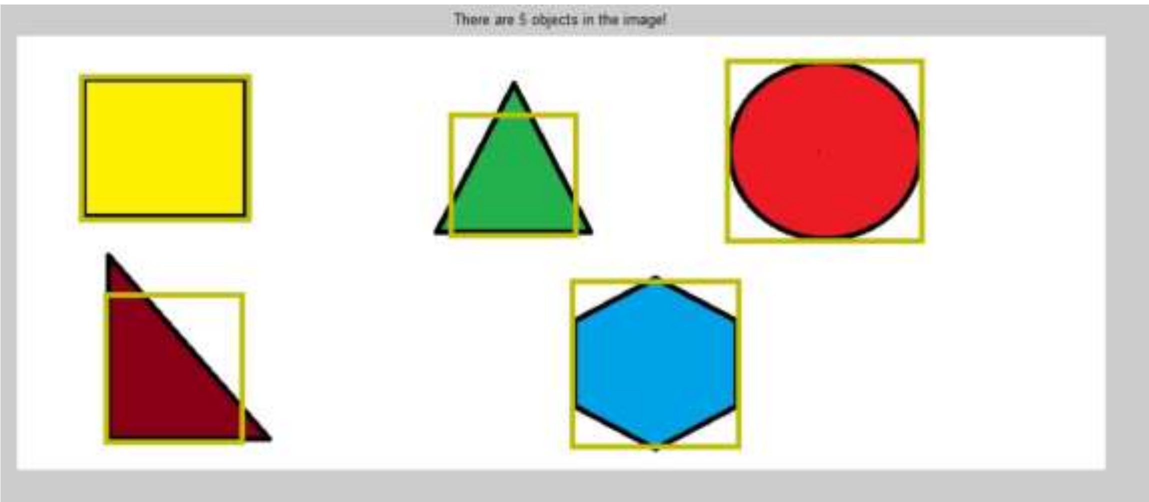
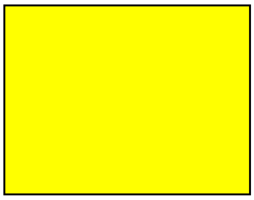
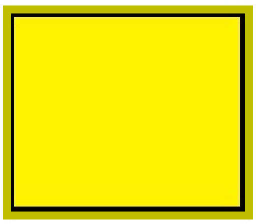
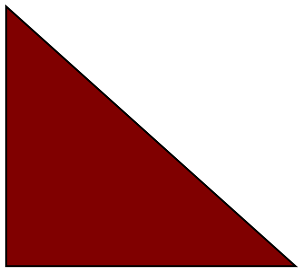
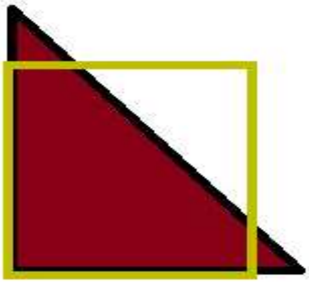
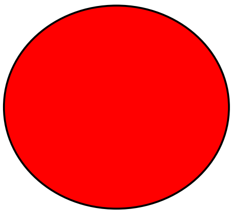
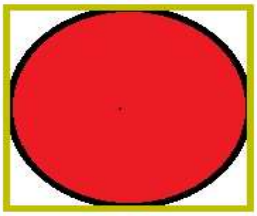
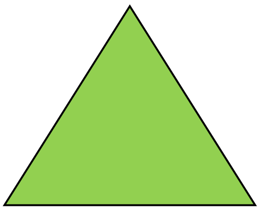
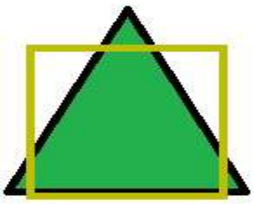
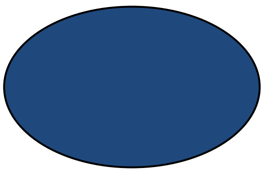
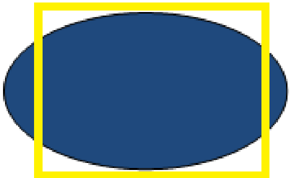
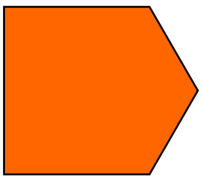
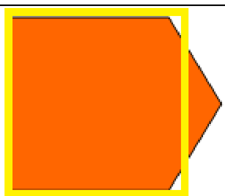
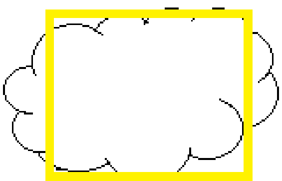


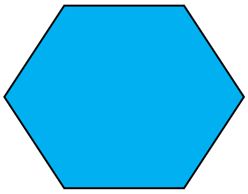
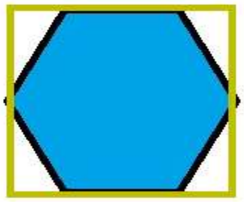
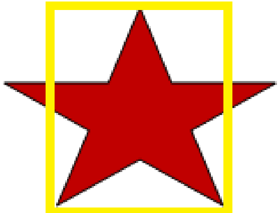
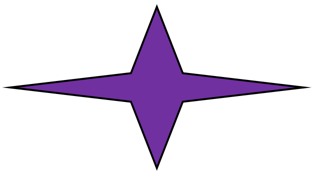
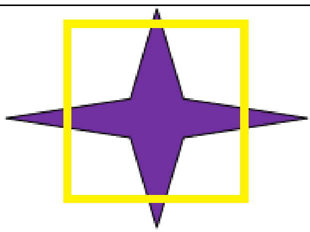
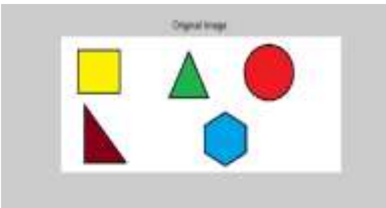
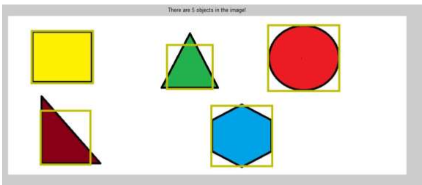
Fig.22 Final Result of Genetic Algorithm for multiple object together.

8.2. MULTIPLE IMAGE DETECTION USING HT WITH EFFICIENCY

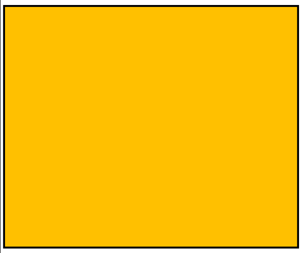
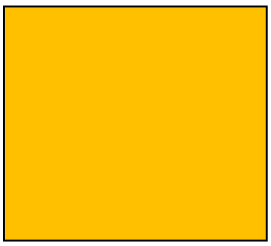
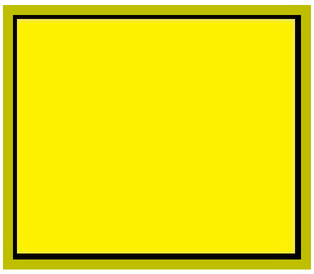
Input Diagram	Detected Diagram	Efficiency
		100%

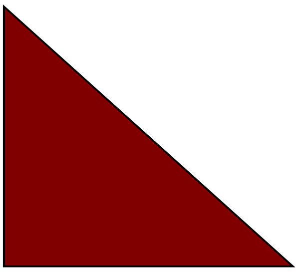
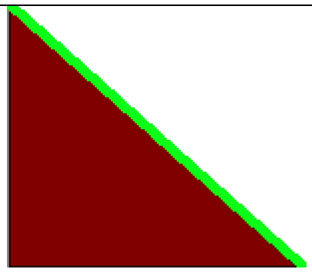
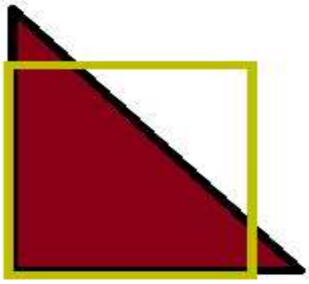
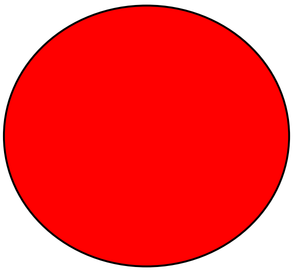
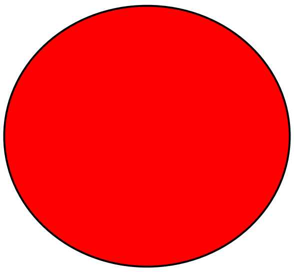
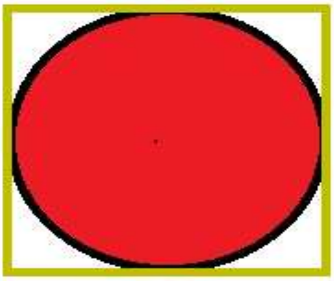
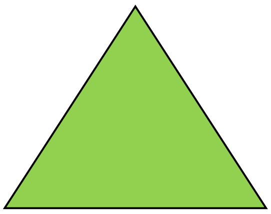
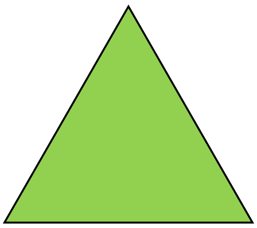
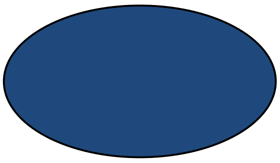
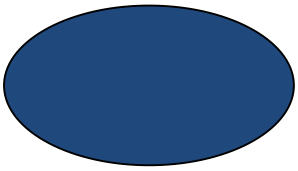
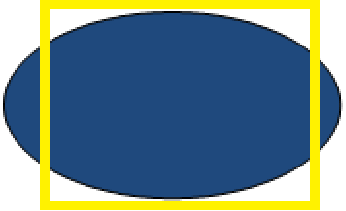
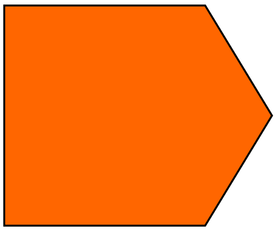
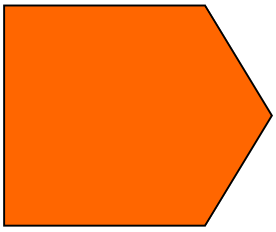
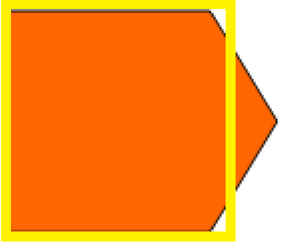
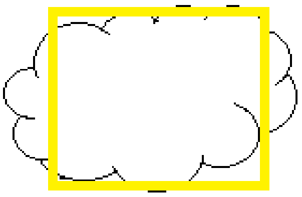
		64%
		24%
		55%
		30%

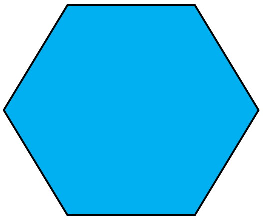
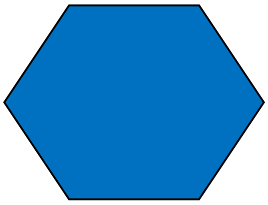
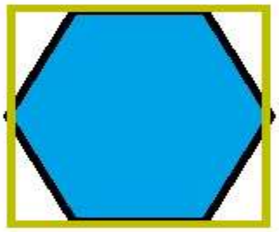
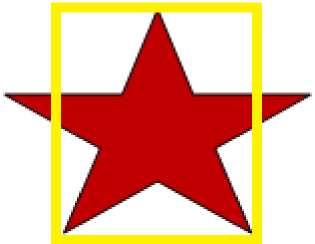
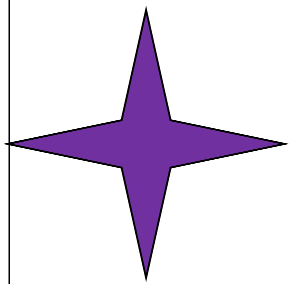
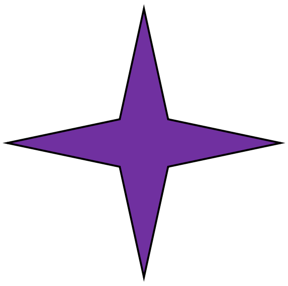
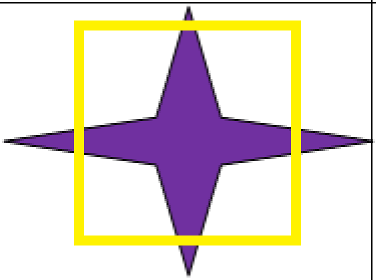
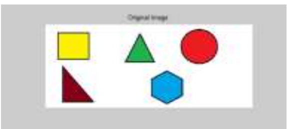
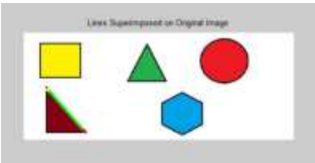
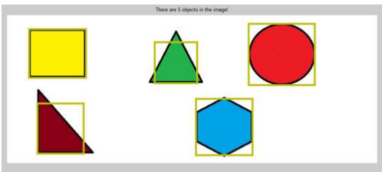
Input Diagram	Detected Diagram	Efficiency
		78%
		33%

		65%
		40%
		34%
		70%

9. COMPARISON OF HOUGH TRANSFORM AND GENETIC ALGORITHM

Input Diagram	Results of HT		Results of GA	
	Detected Diagram	Efficiency	Output	Efficiency
		0%		100%

		44%		64%
		0%		24%
		0%		55%
		0%		30%
		0%		78%
		0%		33%

		0%		65%
		0%		40%
		0%		34%
		9%		70%

It has been observed that efficiency of hough transform comes to be very low, when the edges of the object are more than three, and thus this algorithm for 2-D diagram detection is not so much efficient, we have also shown the result of multiple image, and the result are not so efficient. The efficiency range for HT is from 0% to 48% Same input 2-D diagram we have integrated with GA, and the efficiency is promising and the diagram detection probability increases for more than 15 to 20%, totaling up to 60% to 75 % efficiency.

9.Conclusion

T1he object and pattern detection was successfully carried out for hough and Genetic Algorithm. We have analyzed for Single as well as multiple object. Objects are detected by background differencing in both Hough and GA. We have Integrated with Genetic Algorithm, to analyze the flow of pattern and object. The Diagram integration gives the high cutting edge result for hough transform. Low contrast levels can present problems, leading to poor object segmentation and fragmentation, particularly on older analogue tracking system. The object detection can be done through the kernel, which is interacting with the given image by morphology and result will be identified. The model-free tracking or detecting algorithm described in this paper addresses object fragmentation and the masking induces spatially-smooth similarity.

The kernel when integrated with hough transform, gives the perfect result for single object, but same kernel when integrated for multiple image, it fails to detect it. This gives the upper hand to genetic algorithm for multiple object detection. We have set different plane for object detection, this we have consider as the subjects, and after mutation result for all the plane together we have done the priority analysis. Thus the normal superimposition is done for object detection as well as count how many object are there in a particular diagram or image.

10.REFERENCE

[1]Badi, K. and Shimura, M., “Machine recognition of Arabic cursive script,” Trans. Inst. Electron. Commun. Eng., vol.65(E), pp. 107-114 (1982)

[2]. Matheron, G. (1975) Random Sets and Integrated Geometry Wiley: New York

[3] Borell, C., The Brunn-Minkowski inequality in Gauss space. Invent. Math. 30 (1975), 207–216.

[4] Sagar, B.S.D., Venu, M., Srivinas, D., 2000. Morphological operators to extract channel networks from digital elevation models. International Journal of Remote Sensing 21 (1), 21–30

[5] Halland J. Adaptation In Natural and Artificial Systems University of Michigan Press

[6] Lihua Zhang, Wenli Xu , Cheng Chang, Genetic algorithm for affine point pattern matching , Pattern Recognition Letters 24 ,9–19,2003

[7]. Mauldin, M. L., "Maintaining Diversity in Genetic Search", The National Conference on Artificial Intelligence (AAAI- 84), August 1984.

[8] Lamdan, Y. and H. Wolfson, \On the Error Analysis of Geometric Hashing", IEEE Computer Vision and Pattern Recognition Conference, June 1991, pp. 22-27

[9] P. Radhakrishnan College of Computer Science, King Khalid University, Abha, Kingdom of Saudi Arabia

[10]Atherton, P., Earl, C., Fred, C. A graphical simulation system for dynamic five-axis NC verification, Autofact Show of the Society of Manufacturing Engineers, Detroit, Nov. 1987, 2-1 - 2-12.

[11]. Baird, H.S. Mode-Based Image Matching Using Location, MIT Press, 1985

[12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge Press, University 2000

[13]Sun, Y., “Hopfield neural network based algorithms for image restoration and reconstruction-Part I: Algorithms and Simulations”, IEEE Transaction on Signal Process vol. 48(7), pp. 2105-2118 (2000)

[14]Szu, H., Yang, X., Telfer, B. and Sheng, Y., “Neural network and wavwlet transform for scale invariant data classification”, Phys. Rev. E 48, pp. 1497-1501 (1993)

[15]Nagy, G., “Classification Algorithms in Pattern Recognition,” IEEE Transactions on Audio and Electroacoustics, vol. 16(2), pp. 203-212 (1968)

[16]Hoppensteadt, F.C. and Ihikevich, E.M., “Synchronization of Laser Oscillators, Associative Memory, and Optical Neurocomputing,” Phys. Rev., vol. 62(E), pp. 4010-4013 (2000)

[17]Keith, L.P., “Classification Of Cmi energy levels using counterpropagation neural networks,” Phys. Rev., vol. 41(A), pp. 2457-2461 (1990)
[18]Carlson, J.M., Langer, J.S. and Shaw, B.E., “Dynamics of earthquake faults,” Reviews of modern physics, vol. 66(2), pp. 657-670 (1994)
[19]Palaniappan, R., “Method of identifying individuals using VEP signals and neural networks,” IEE Proc. Science Measurement and Technology, vol. 151(1), pp. 16-20 (2004)
[20]Zhao, H., “Designing asymmetric neural networks with associative memory,” Phys. Rev. vol. 70(6), pp. 137-141 (2004)
[21]Schutzhold, R., “Pattern recognition on a quantum computer,” Phys. Rev. vol. 67(A), pp. 311-316 (2003)
[22]Impedovo, S., “Fundamentals in Handwriting Recognition.” NATO-Advanced Study Institute, vol. 124, Springer-Verlag (1994)
[23]Mori, S., Suen, C.Y. and Yamamoto, K., “Historical review of OCR research and development,” Proceeding of the IEEE, vol.80 (7), pp. 1029-1058 (1992)
[24]Fukushima, K. and Wake, N., “Handwritten alphanumeric character recognition by the neocognitron,” IEEE transaction on Neural Networks, vol. 2(3), pp. 355-365 (1991)
[25]Blackwell, K.T., Vogl, T.P., Hyman S.D., Barbour, G.S. and Alkon, D.L., “A New Approach to Handwritten Character Recognition,” Pattern Recognition vol. 25, pp. 655-666 (1992)
[26]Ie Cun, Y., Boser, B., Denkar, J.S., Henderson, D., Howard, R.E., Hubbard, W., and Jackel, L.D., “Handwritten Digit Recognition with a Back-Propagation Network,” Advances in Neural Information Processing Systems, vol. 2, pp. 396-404 (1990)
[27]Kharma, N.N., and Ward, R.K., “A novel invariant mapping applied to hand-written Arabic character recognition,” Pattern Recognition vol. 34(11), pp. 2115-2120 (2001)
[28]Badi, K. and Shimura, M., “Machine recognition of Arabic cursive script,” Trans. Inst. Electron. Commun. Eng., vol.65(E), pp. 107-114 (1982)
[29]Suen, C.Y., Nadal, C., Lagault, R., Mai, T.A., and Lam, L., “Computer recognition of unconstrained handwritten numerals,” Proc. IEEE, vol. 80(7), pp. 1162-1180 (1992)
[30]Knerr, S., Personnaz, L., and Dreyfus, G., “Handwritten digit recognition by neural networks with single-layer training,” IEEE Trans. on Neural Networks, vol. 3, pp. 962-968 (1992)
[31]Lee, S.W., and Song, H.H., “A New Recurrent Neural Network Architecture for Visual Pattern Recognition,” IEEE Trans. on Neural Networks, vol. 8(2), pp. 331-340 (1997)
[32]Urbanczik, R., “A recurrent neural network inverting a deformable template model of handwritten digits,” Proc. Int. Conf. Artificial Neural Networks, Sorrento, Italy, pp. 961-964 (1994)
[33]Hagan, M.T., Demuth, H.B. and Beale, M.H., “Neural Network Design,” PWS Publishing Co., Boston, MA (1996)
[34]Rumelhart, D.E., Hinton G.E., and Williams R.J., “Learning internal representations by error propagation.”, MIT Press, Cambridge, vol. 1,pp. 318–362 (1986).
[35]Sprinkhuizen-Kuyer, I.G., and Boers, E.J.W., “The Local Minima of the error surface of the 2-2-1 XOR network,” Annals of Mathematics and Artificial Intelligence, vol. 25(1-2), pp. 107-136 (1999)
[36]Zweiri, Y.H., Seneviratne, L.D., and Althoefer, K., “Stability Analysis of a Three-Term Backpropagation algorithm,” Neural Networks Journal, vol. 18(10), pp. 1341-1347 (2005)
[37]Abarbanel, H., Talathi, S., Gibb, L., and Rabinovich, M., “Synaptic plasticity with discrete state synapses,” Phys. Rev., vol. E, 72:031914 (2005)
[38]Shrivastava, S. and Singh, M.P., “Performance evaluation of feed-forward neural network with soft computing techniques for hand written English alphabets”, Journal of Applied Soft Computing, vol. 11, pp. 1156-1182 (2011).

***