

Content based Image Search using Rough Set and Representative Graph

G. S. G. N Anjaneyulu^{1*}, C. Kaushika² and Anugrah Kumar²

¹School of Advanced Sciences, VIT University, Vellore-632014, Tamilnadu, India; anjaneyulu.gsgn@vit.ac.in

²School of Computing Science and Engineering, VIT University, Vellore-632014, Tamilnadu, India

Abstract

Until now, the search carried out in devices using the conventional, textual form of search now seems tedious, considering the advances made in human software interaction since the emergence and development in touch-screen interfaces. In the past decade finger-touch or multi-touch interfaces have completely altered the way we interact with the devices. The input method implemented in search mechanisms, use the textual form of string input method that can be extended to graphical search which would enhance the interaction of human to software and also is efficient. This Paper proposes a new content based Image search Algorithm using Rough Set and Relational Graph.

1. Introduction

In the past decade finger-touch or multi-touch interfaces have completely altered the way we interact with the devices. The input method implemented in search mechanisms, use the textual form of string input method that can be extended to graphical search which would enhance the interaction of human to software and is also efficient. As we know that everything is shifting from character user interface to graphical user interface, thus it would be prudent enough if we have a graphical search mechanism instead of traditional text based search mechanism. Thus we propose a graphical search algorithm for searching the nearest matching icon using relational graph. The Graph generated would do using Rough Set Theory. We use Rough Sets to deal with imprecise and vague data involved in an image. The Image has various attributes such as Intensity, Pixels per area, Texture and Shape. We use a Relational Graph (RG) to symbolize each image in the database. Each region in an image is characterized by a node with multiple attributes. The relation between regions is represented by an edge.

2. Rough Set - Notion - Approach

Rough set concept is a novel mathematical tactic which efficiently handles imprecise or vague or inexact or uncertain data. Initially, it was originated by Zdzislaw Pawlak¹. It delivers an excellent alternative solution to the problems comprising imprecise data, where conventional set theory flops. Assume that U is a Universal set and R is an Indiscernibility relation $R \subseteq U \times U$ which characterizes lack of information about the components of U . Suppose that X is a subset of U . By applying Basic Rough Set theory concepts, we can have R is a Lower approximation of X

$$R^*(x) = \{x \in U \text{ for } \{R(x): R(x) \subseteq X\}\}$$

R is an Upper approximation of X

$$R^+(x) = \{x \in U \mid R(x) \cap X \neq \emptyset\}$$

R is a Boundary region of X

$$RN_R(X) = R^+(x) - R^*(x)$$

$$\text{If } RN_R(X) = \emptyset \text{ (Set } X \text{ is Crisp Set)}$$

*Author for correspondence

If $RN_R(X) \neq \emptyset$ (Set X is Rough Set)

The Data is inaccurate as well as superfluous. Therefore many of its redundancy can be detached through calculating Reduct of the sets⁶. Also note that $Core(x) = \cap Red(X)$ where core of X is a finite collection of all indispensable attributes of X ¹. Information is epitomized in form of two attributes Condition and Decision attributes. All the inputs or parameters includes of Conditions whereas Output embraces of Decision. A decision rule is computed by each row. A fixed collection of Decision rules is identified as Decision Algorithm. Once Determining the Reduced set, then respective Decision Algorithm is computed². The Positive Region of U/B subjected to A is the collection of all elements of U that can be uniquely characterized to blocks of partition U/D by means of A^2
 $POS_A(B) = U \mid X \in U/I(D) \mid A^*(X)$

3. Conversion of Image to Relational Graph

In mathematics, every graph is a pictorial demonstration of a set of objects where certain pairs of the objects are connected by links. The interrelated objects are denoted by mathematical abstractions identified as vertices and the links that connect certain pairs of vertices are recognized as edges. Typically, a graph is defined in diagrammatic form as a set of points or nodes for the vertices, joined by lines or curves for the edges. Graphs are one of the important tools of cramming in discrete mathematics. The edges can be either directed or undirected³.

The Steps involved in conversion of an image to graph are:

STEP 1: Distinguishing Between Icon's Foreground and Background

Any icon has a fixed background color and a foreground which does not contain any shade that is different from background color. Thus Image representing the icon is converted to Image having only two distinguishing colors i.e. Fore ground and Background. For Instance we use an icon of very famous video game named pac-man

STEP 2: Formation of Grids

The Image formed should be divided into grids. These Grids would eventually help in generating the required graph for the image. Here for simplicity we use 8 X 8 grid.

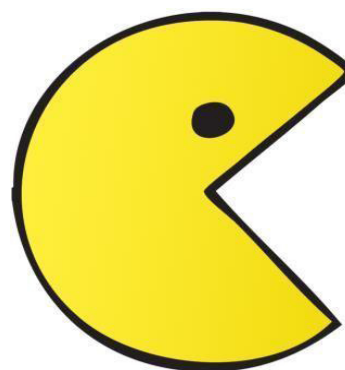


Figure 1. Standard Pac-Man Icon.

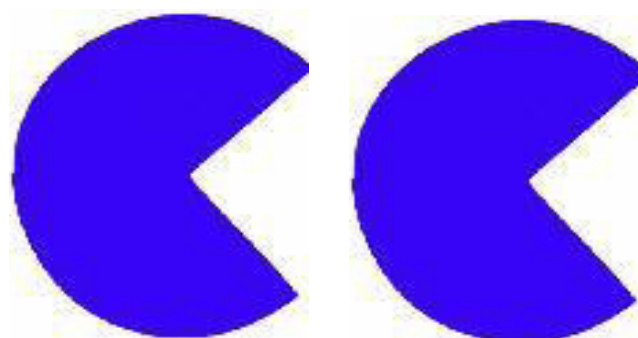


Figure 2. Image generated using Foreground and Background.

STEP 3: Formation of Boundary region

Boundary region are distinguished form other regions through grids. Those grids which contain both foreground and background color are said to be the boundary grid. Thus these grids are searched through exhaustive

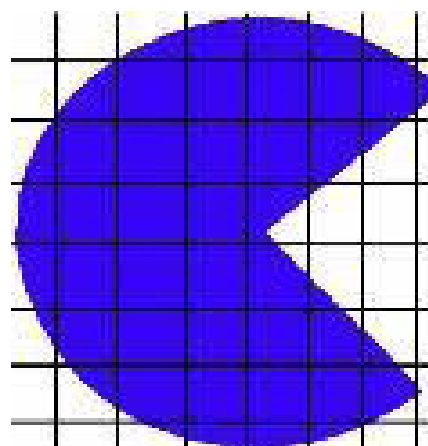


Figure 3. Division of Image into 8 X 8 grids.

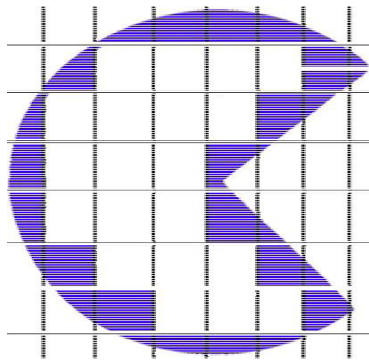


Figure 4. Boundary region.

searching algorithms and other grids are reset to that of background color.

STEP 4: Formation of Boundary

After the boundary region is established a particular strict boundary is generated. Generation of Boundary involves finding the boundary which separates a grid from background to foreground color. If such a boundary formed is then retrieved and particular foreground is converted to background.

STEP 5: Finding Convex Hull

After finding the boundary, the next step would be to find convex hull in mathematics, the convex hull or convex sachet of a collection X of points in the Euclidean plane or Euclidean space is the smallest convex set that contains X . For illustration, when X is a bounded subset of the plane, the convex hull may be visualized as the shape produced by a rubber band stretched around X . We would be by means of the famous graham scan algorithm for finding the convex hull. Algorithm applied here is Graham scan. Algorithm and details can be found in ^{5,6}.

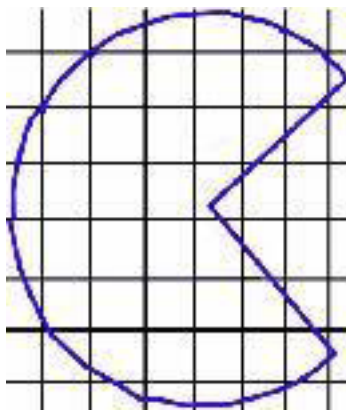


Figure 5. Boundary of the image.

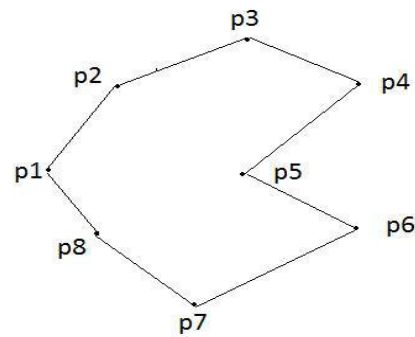


Figure 6. Representative Graph.

STEP 6: Finding the graph (Figure 6)

After we have done with the convex hull we may proceed to our final step i.e. finding the representative graph for an image. For sake of simplicity we consider here an eight node graph. Convex hull so obtained is optimized to get a 2d undirected graph which represents our icon.

4. Rough Set for Searching the Graph

Now after we have obtained the graph for particular icon our next step would be to search the graph formed by user to that of representative graph. Now for obtaining this we would require to form a matrix of x, y coordinates representing the position of particular pixel. For simplicity here we use 8×8 Matrix. Now the user drawn graph is exhaustively searched and matched with pixels all the positions in the matrix. Suppose the user drawn graph is Q with nodes $Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8$. Now the pixels are noted and stored in matrix.

Since the graph is undirected all the points are exhaustively compared with representative graph. This helps us

	X							
Y	0	1	2	3	4	5	6	7
0			Q8					Q1
1							Q2	
2						Q3		
3	Q7						Q4	
4								Q5
5			Q6					
6								
7								

in establishing the adjacency of the graph⁴. Now all the 8 nodes are exhaustively permuted against representative graph's nodes. For eg. – The Permutations so obtained for 8 nodes are 40320 different graphs in 8 X 8 pixel density. Instead of exhaustively comparing each and every point of a graph with representative graphs node the boundary condition using rough set. The Data is inaccurate as well as redundant. Thus many of its redundancy can be truncated through computing Reduct of the Sets. Also Core (X) = $\cap$ Red(X) where Core of X is a finite collection of all indispensable attributes of X. An Adjacency matrix is obtained which compares one node with other adjacency. We applied the following data for approximation of rough set X.

CONNECTION MATRIX OF P1

Pt	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Match
1	1	1	1	0	0	0	0	0	Yes
2	1	0	1	0	0	1	0	0	No
3	0	1	1	1	0	0	0	0	No
4	0	0	0	1	1	1	0	0	No
5	1	0	0	1	0	0	1	0	No
6	0	0	0	0	1	1	1	0	Yes
7	0	0	0	0	0	1	1	1	No

CONNECTION MATRIX OF P2

pt	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Match
1	1	1	1	0	0	0	0	0	No
2	1	1	1	0	0	1	0	0	No
3	0	0	1	1	0	0	0	0	No
4	0	0	0	1	1	1	0	0	Yes
5	1	1	0	1	0	0	1	0	No
6	0	0	0	0	1	1	1	0	Yes
7	0	0	0	0	0	1	1	1	No

CONNECTION MATRIX OF P3

pt	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Match
1	1	1	1	0	0	0	0	0	Yes
2	1	0	1	0	0	1	0	0	No
3	0	1	1	1	0	0	0	0	No
4	0	0	0	1	1	1	0	0	No
5	1	0	0	1	0	0	1	0	No
6	0	0	0	0	1	1	1	0	Yes
7	0	0	0	0	0	1	1	1	Yes

CONNECTION MATRIX OF P4

Pt	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Match
1	1	1	1	0	0	0	0	0	Yes
2	1	0	1	0	0	1	0	0	Yes
3	0	1	1	1	0	0	0	0	No
4	0	0	0	1	1	1	0	0	No
5	1	0	0	1	0	0	1	0	No
6	0	0	0	0	1	1	1	0	No
7	0	0	0	0	0	1	1	1	Yes

CONNECTION MATRIX OF P5

pt	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Match
1	1	1	1	0	0	0	0	0	No
2	1	0	1	0	0	1	0	0	No
3	0	1	1	1	0	0	0	0	No
4	0	0	0	1	1	1	0	0	Yes
5	1	0	0	1	0	0	1	0	Yes
6	0	0	0	0	1	1	1	0	Yes
7	0	0	0	0	0	1	1	1	No

CONNECTION MATRIX OF P6

pt	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Match
1	1	1	1	0	0	0	0	0	No
2	1	0	1	0	0	1	0	0	No
3	0	1	1	1	0	0	0	0	No
4	0	0	0	1	1	1	0	0	No
5	1	0	0	1	0	0	1	0	Yes
6	0	0	0	0	1	1	1	0	Yes
7	0	0	0	0	0	1	1	1	Yes

CONNECTION MATRIX OF P7

pt	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Match
1	1	1	1	0	0	0	0	0	No
2	1	0	1	0	0	1	0	0	Yes
3	0	1	1	1	0	0	0	0	No
4	0	0	0	1	1	1	0	0	Yes
5	1	0	0	1	0	0	1	0	No
6	0	0	0	0	1	1	1	0	Yes
7	0	0	0	0	0	1	1	1	No

CONNECTION MATRIX OF P8

pt	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Match
1	1	1	1	0	0	0	0	0	Yes
2	1	0	1	0	0	1	0	0	No
3	0	1	1	1	0	0	0	0	Yes
4	0	0	0	1	1	1	0	0	No
5	1	0	0	1	0	0	1	0	No
6	0	0	0	0	1	1	1	0	No
7	0	0	0	0	0	1	1	1	Yes

After finding the boundary region by using these eight tables we get

R-Lower approximation of X;

$$R^*(x) = \cup x \in U \{ R(x) : R(x) \subseteq X \}$$

R- Upper approximation of X;

$$R^*(x) = \cup x \in U \{ R(x) : R(x) \cap X \neq \emptyset \}$$

R-Boundary region of X;

$$RNR(X) = R^*(x) - R^*(x)$$

If $RNR(X) = \emptyset$ (Set X is Crisp Set);

If $RNR(X) \neq \emptyset$ (Set X is Rough Set)

BOUNDARY TABLE

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
P1					1	1	1	
P2						1	1	1
P3	1						1	1
P4	1	1						1
P5	1	1	1					
P6		1	1	1				
P7			1	1	1			
P8				1	1	1		

Thus Graph that matches with representative graph is (Q6 Q7 Q8 Q1 Q2 Q3 Q4 Q5). All the nodes of this graph are compared with pixel density of representative graph and a positive region is calculated using Rough Set theory. If Positive region lies in the region greater or equal to 0.5 then accept. Since all the points are accepted hence Graph matching successful and resultant icon is activated.

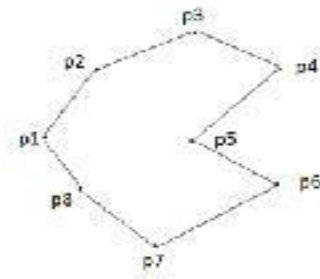
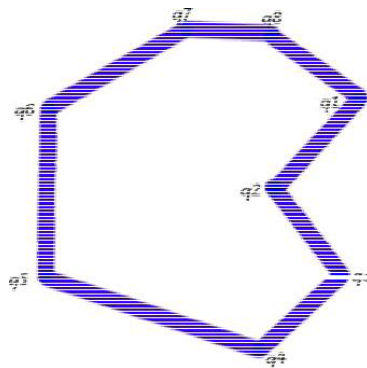


Figure 7. User drawn Graph.

5. Conclusion

In this paper we have implemented rough set theory and convex hull method to match and construct and search the graphs for context based graphical searching. The user Generated graph is then matched with the representative graph and if it matches successfully then user is directed to following application of which icon is a part, although user interactivity increases using this approach but it may not be as efficient as other text based Algorithms. The complexity of this algorithms is calculated to be $O(n \log(n))$. This algorithm becomes more and more complex as the pixel density increases. However this method proves too beneficial to those users who are hard in remembering textual data.

6. References

1. Pawak. Rough sets. International Journal of Computer and Information Sciences. 1982.
2. Polkowski L. Rough sets: mathematical foundations. advances in soft computing. Physica-Verlag, A springer-Verlag Company; 2002.
3. Gross JL, Yellen J. Graph theory and its applications. CRC Press; 2005.
4. Gabow HN. Faster scaling algorithms for general graph matching problems. J ACM. 1991.
5. Graham. Graham scan algorithm. Available from: http://en.wikipedia.org/wiki/Graham_scan
6. Graham RL. An efficient algorithm for determining the convex hull of a finite planar set. Inform Process Lett. 1972; 1:132-3.