

Image Segmentation Using Swarm Intelligence Based Graph Partitioning

Dhanya G S^{#1}, Dr. R. Joshua Samuel Raj^{#2}, P.Sivasamy^{#3}

¹Pg Scholar, ²professor, ³assistant Professor,

[#]Department Of Computerscience And Engineering, Anna University-Chennai,
Rajas Engineering College, Tirunelveli, Tamil Nadu, India.

Abstract: The performances of image segmentation methods are dependent on many factors such as intensity, texture, image content. A single segmentation method cannot be applied to all types of images and, all methods do not perform well for one particular image. The graph based methods achieve segmentation on the basis of local properties of image. To segment images for applications where detailed extraction of features is necessary, consideration of global impression along local properties is inevitable. Efficient implementation, energy consumption and computational time are key aspects in defining performance of these methods. In this paper, the graph partitioning scheme Swarm Intelligence based Graph Partitioning (SIBGP) for Image Segmentation is developed. SIBGP works through three steps namely, partitioning, refining and recursive partitioning. This technique allows consideration of both, local as well as global features during normalized cut based segmentation to meet the requirement of precise segmentation. This creates strong weight connections between the identical neighbouring pixels resulting in better segmentation quality. Due to the proper combination of algorithms at appropriate stage for graph partitioning and the fundamental property that entire swarm converges to an optimal best solution, the algorithm produces best segmentation in computationally less time for all types of images.

Keywords: graph partitioning, normalized cut, partitioning, refining, SIBGP

I. Introduction

Digital image processing opened new avenues for interdisciplinary research for reliable personal identification and authentication techniques such as biometrical systems, medical applications, remote sensing and in many others. The major steps in digital image processing shown in Figure 1 include image acquisition, enhancement, restoration, morphological processing, image segmentation, representation and description, object recognition. Image acquisition digitizes the image captured by camera; enhancement is the process of manipulating an image so that the outcomes are more appropriate for explicit applications; restoration increases the appearance of an image which has a tendency to degrade; morphological processes are used to take out the image components that are useful in the narration and appearance of an image; image segmentation is the utmost tough task in digital image processing which splits up objects from the background; representation makes the choice whether to denote data as boundary or as a complete region; recognition allocates label to an object based on information provided by its descriptor. The process of segmentation is a very important step in computer vision for image retrieval, visual summary and image base modelling. Image segmentation is the major phase in any effort to evaluate an image spontaneously which links the gap between low-level and advanced image processing. The goal of segmentation is typically to locate certain objects of interest. The task of segmenting images for separating objects, ignoring the effects of lights and textures on them has been extensively investigated for many years. [1].

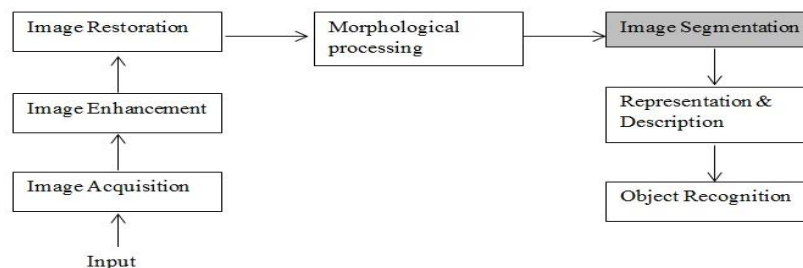


Figure 1: Steps in digital image processing

In graph based methods image is represented by undirected weighted graphs where vertices are pixels. Weight of each node is represented as a function of similarity between nodes. The sets of nodes are partitioned into disjoint nodes having strong affinities between them. But they differ in graph algorithms. Usually applications require computationally fast and efficient algorithms. The local properties of the graph usually influence these segmentation approaches without considering the global impressions of the image[9]. Precise criteria for good calculation and its efficient calculation are the several challenges to achieve segmentation through partitioning. In this context these segmentation problems are converted to graphs and solved as the GPP. Karypis et al. [2],[3],[4], [5], [6] developed single global priority queue. The vertex selected yields maximum improvement in objective function hence improved the balance constraint. Sanders and Schulz [7] achieved balance constraint; multiple local constraints are combined to improve the balance globally. Walshaw et al. [8] showed that recursive bisection failed to generate optimal partitioning due to lack of global information hence paved the way to k-way local search algorithm.

II. Proposed Methodology

We have developed Swarm Intelligence Based Graph Partitioning (SIBGP) for partitioning the graph obtained from the image to be segmented. Figure 2 shows the process flow diagram for image segmentation using the proposed methodology. At first, the image to be segmented is converted to graph where the pixels form the vertices and their intensity differences represent the weight. The proposed method (SIBGP) works after the weighted graph is generated. The proposed method works in three stages namely,

- 2.1 Partitioning
- 2.2 Refining
- 2.3 Recursive partitioning

The graph obtained is coarsened and graph partitioning is applied for initial partitioning. Refinement stage includes Particle Swarm Optimization to generate bisected graph, followed by recursively k-partitioning is generated. This recursively k-partitioned graph is projected back to the segmented image.

2.1 Partitioning

The objective of partitioning is to allocate vertices into blocks having strong interconnections. If $G=(V,E)$ is a weighted undirected graph with nonnegative weights assigned to edges and k is a natural number greater than 1, then the GGP is to divide the vertex V into blocks of vertices v_1, v_2, \dots, v_k such that $v_1 \cup v_2 \cup \dots \cup v_k = V$. If all the blocks have same weight, then partitioning is balanced.

The objective function for GGP is:

$$Cut(v_1, v_2) = \sum_{p \in v_1} \sum_{q \in v_2} w(p, q) \quad (1)$$

Where v_1, v_2 are blocks of vertices; p and q are called boundary vertices.

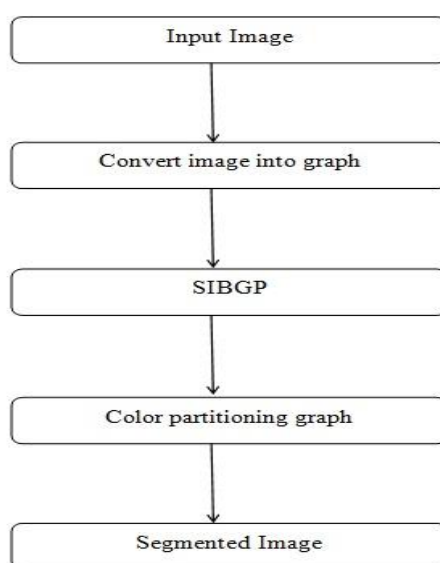


Figure 2: Flow diagram for the proposed method

Graph growing partition arbitrarily selects vertex v and develops region round it till half of the vertex weight has been involved. Quality of partition obtained by this method depends on the starting vertex selected [7]. Greedy Graph Growing Partition (GGGP) also starts with a carefully chosen vertex v and then it enlarges a vertex which leads to minor growth in edge cut. Hence for attaining best partition starting from randomly selected vertices, separate partitions are figured and best one is selected as initial partition. Local search can be combined with this method to improve partitioning. Multiple restarts of the algorithm results in better results. A good starting node can be a node which has maximal distance from random seed node. Graph growing partition algorithm (GGP) introduced by Karypis [6] iteratively generates set A of vertices consisting of half of the vertices such that $w(A) = \frac{1}{2}w(V)$. During implementation of algorithm, vertices of graphs are divided into A, B, C where B contains border of A and C contains remaining vertices. Set is initialized by randomly picking any vertex of the graph:

$$A = A \cup B(2)$$

The process ends when A will contain vertices such that $w(A) = \frac{1}{2}w(V)$. GGP is improved using the concept in [7] and developed Greedy Graph Growing partition (GGGP). In GGGP also A is selected in the same way as in GGP. So partitioning stage in the proposed method includes the following procedures.

For initialization of set A , select the nearest vertex (say u) from B to the randomly selected vertex in set A and then added it to A which is vertex of maximal gain in B . After that, each vertex in set C which is adjacent to u is moved to the set B and calculated its gain. Similarly recalculate the gain of each vertex in B that is adjacent to u and therefore the next iteration starts. This process is continued till the weight of set A extends to half of the total weight. Algorithm ends when $w(A) = \frac{1}{2}w(V)$. GGGP algorithm generates better results for any choice of the initial vertex to move.

2.2 Refining

Refinement stage works using Particle Swarm Optimization (PSO) algorithm [9],[10]. PSO is a computational method which recursively enhances a problem to develop a candid resolution with respect to a given degree of excellence. Each particle's movement is swayed by its local best known position and is also directed in the direction of the best known positions in the search-space, which are reorganised as improved locations are found by other elements. This is expected to move the swarm towards the best solutions. The PSO works as follows;

Generate the entire swarm, Evaluate the initial swarm using the fitness function; initialize the personal best particle and global best of the entire swarm. Update the particle velocity using (3) and (4). Apply velocities to particle positions, evaluate new particle positions. Repeat the above steps till the maximum iterations have reached. The most commonly used stopping criteria for iterations are after a fixed number of iterations or iterations without an improvement in iterations value or when the objective reaches a pre-specific threshold value and thus ultimately a solution is reached. This approach shows improved partitions with less computation time [10]:

$$V_{id}(t+1) = V_{id}(t) + c_1 R_1(p_{id}(t) - x_{id}(t)) + c_2 R_2(p_{gd}(t) - x_{id}(t))(3)$$

$$x_{id}(t+1) = x_{id}(t) + V_{id}(t+1)(4)$$

V_{id} -- Rate of position change of i^{th} particle and t denotes iteration.

x_{id} -- Position of i^{th} particle.

p_{id} -- Historically best position of particle.

p_{gd} -- Position of swarm's global best particle.

R_1 and R_2 -- Two n -dimensional vectors with random numbers uniformly selected between [0, 1].

c_1 and c_2 -- Position constant parameters.

$$p_{id} = p_i \text{ If } f(p_i) > f(p_{id})$$

$$p_{gd} = g_i \text{ If } f(g_i) > f(p_{gd})$$

$F(x)$ is an objective function subjective to maximization.

2.3 Recursive Partitioning

The third step works by applying k -partitioning to the results of the first two steps [11]. k -partitioning minimizes the total distance between data points and cluster centre. The steps involved in k -partitioning are as follows;

Decide the number of k ; randomly set the k -cluster centre at different initial locations in the image. Assign each pixel to the cluster having centre nearest to the respective pixel. Compute each cluster centre which should be average co-ordinates of data points. Repeat the process until no more changes are needed.

III. Results

One of the major difficulties in the field of image segmentation arises from the fact that not even human beings are able to uniquely and unambiguously decide and agree over a correct segmentation. This lack of unique ground truth complicates the comparison of results generated by various segmentation methods. However, efforts were taken to create a framework for objective evaluation of segmentation algorithms. A well-known example is the Berkeley Segmentation Dataset (BSDS) [12] which provides massive collection of natural images along with human marked applications of ground truth. Distinct images along with three different human ground truth suggestions are given in Figure 3.

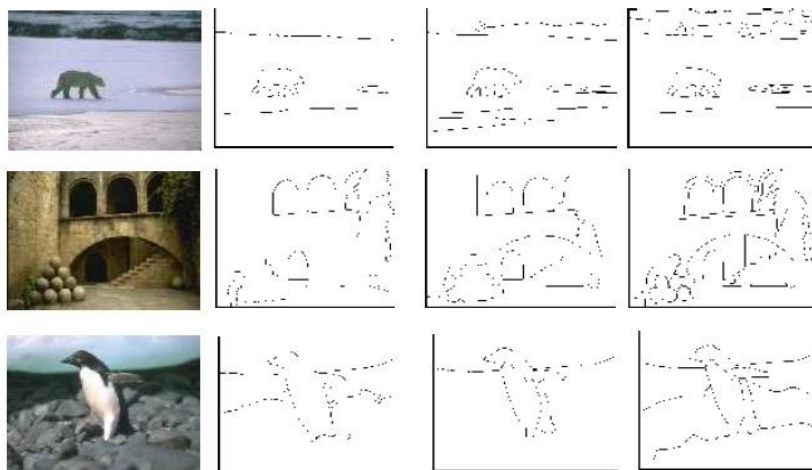


Figure 3: Images from BSDS [12] with three different human ground truth suggestions

In SIBGP, graph generated from an image are coarsened to the smaller graphs which helps to reduce the cut value and PSO applied at the refinement stage optimizes the process. As a result it produces better segmentation quality by detecting most of the image features and negligible texture involvement. Segmentation results generated by SIBGP for few sample images are shown in Figure 4.

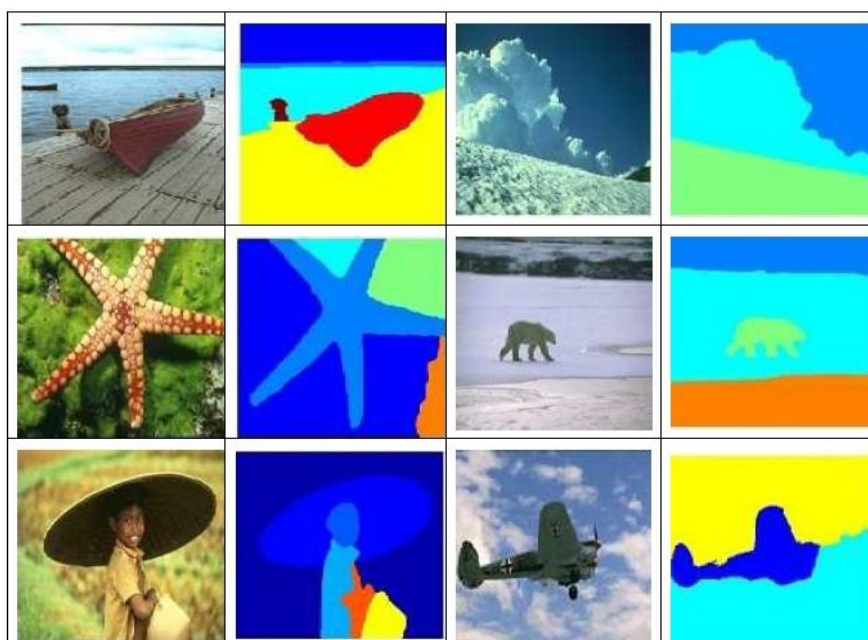


Figure 4: Segmentation results generated by SIBGP

For comparison, we generated segmentation results for the six test images from the Berkeley dataset [12] by SIBGP method, the Normalized cut method [3]. The normalized cut method uses the individual image features for the graph development. Segmentations generated by both methods for six test images are shown in

Figure 5. In results our approach is denoted by SIBGP and Normalized cut by N-cut. Results generated by N-cut are not impressive due to considerable involvement of texture and local features are also ignored.

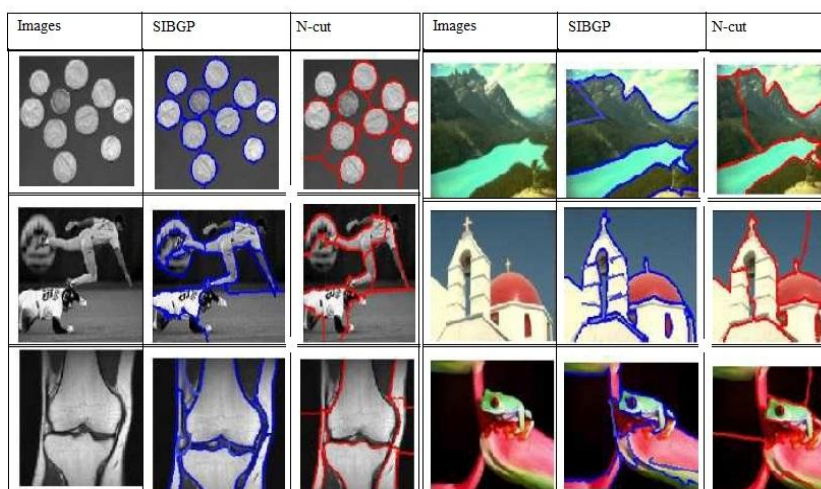


Figure 5: Segmentation results generated by SIBGP and N-cut

We have calculated computational time on these test images using SIBGP and N-cut. In partitioning process time required for refinement is reduced due to the use of PSO, hence the computational time required for overall segmentation process in our approach is comparatively less than that in N-cut. Figure 6 shows the computation time comparison between SIBGP and N-cut.

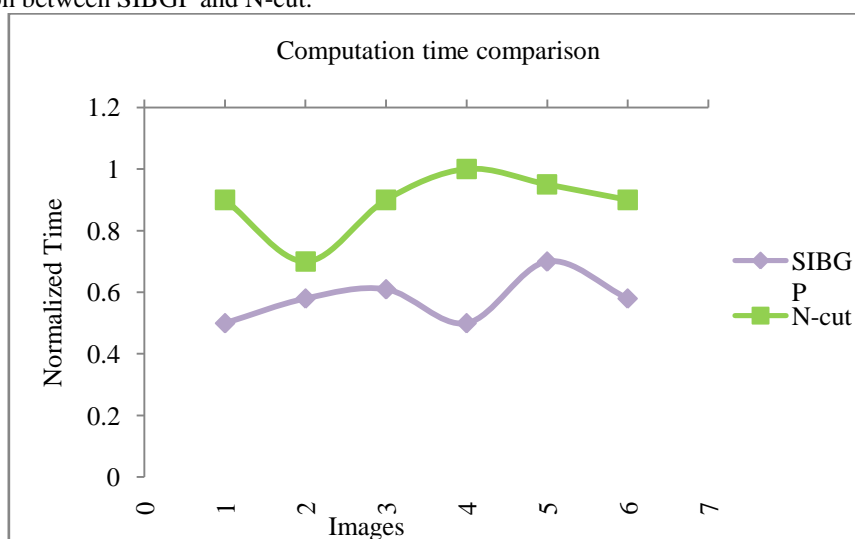


Figure 6: Computation time comparison with N-cut

IV. Conclusion

In this paper, we have presented SIBGP, Swarm Intelligence Based Graph Partitioning approach for balance graph k-partitioning, set to $k=2, 4, 8, 16, 32, 64$. The developed algorithm follows the fundamental concepts of graph partitioning method and integrates PSO refinement procedure. From the results and comparisons, we can observe that the overall performance of the developed SIBGP is remarkable for generating balanced partition in significantly less computing time.

References

- [1]. P. Sander, and C. Schulz, "High Quality Graph Partitioning," BMSW, pp.19 - 36, 2013.
- [2]. G. Karypis and V. Kumar. Analysis of multilevel graph partitioning. Technical Report TR 95-037, Department of Computer Science, University of Minnesota, 1995. Also available on WWW at URL <http://www.cs.umn.edu/~karypis/papers/mlevel analysis.ps>.
- [3]. G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. Technical Report TR 95-035, Department of Computer Science, University of Minnesota, 1995. Also available on WWW at URL <http://www.cs.umn.edu/~karypis/papers/mlevel serial.ps>. A short version appears in Intl. Conf. on Parallel Processing 1995.

- [4]. G. Karypis and V. Kumar. Multilevel k -way partitioning scheme for irregular graphs. Technical Report TR 95-XXX, Department of Computer Science, University of Minnesota, 1995. Also available on WWW at URL <http://www.cs.umn.edu/~karypis/papers/mlevelkway.ps>.
- [5]. R. Woods and R. Gonzalez, Digital Image Processing. Addison-Wesley, 2002.
- [6]. G. Karypis and V. Kumar, "A Fast Quality High Quality Multilevel Scheme for Partitioning Irregular Graphs," SIAM Journal on Scientific Computing, vol.1, no.20, pp. 359 - 392, 1998.
- [7]. L. Safro, P. Sanders, and C. Schulz, "Advanced Coarsening Schemes for Graph Partitioning," in Symposium on Experimental Algorithms, Vol. 7276, pp.369 - 380, Springer 2012.
- [8]. C. Walshaw, Multilevel Refinement for Combinatorial Optimization Problem, in Annals of Operations Research, pp. 325 - 372, 2004.
- [9]. P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient graph based image Segmentation," Int. J of Computer Vision, vol.59, no.2, pp. 167-181, 2004.
- [10]. D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: International Conference on Computer Vision, 2001, pp. 416-423.
- [11]. B. K. Panigrahi, Y. Shi, and M.-H. Lim (eds.), "Handbook of Swarm Intelligence," Series: Adaptation, Learning, and Optimization, Vol 7, Springer-Verlag Berlin Heidelberg, 2011. ISBN 978-3-642-17389-9.
- [12]. J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.22, no.8, pp. 888-905, August 2000.
- [13]. S. Gadde, "Graph partitioning algorithms for minimizing internode communication on a distributed system," Ph. D Thesis, 2013.
- [14]. J. Kennedy and R. Eberhart, "Particle swarm optimization," in the Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, Vol. 4, pp. 1942-1948, 1995.
- [15]. K. Duraiswamy and V. ValliMayil "Similarity Matrix Based Session Clustering by Sequence Alignment Using Dynamic Programming," Computer and Information Science journal, Vol 1, No.3, 2008
- [16]. P. Chardaire, M. Barake, and G. P. McKeown, "A PROBE based heuristic for Graph Partitioning," IEEE Transactions on Computers, Vol.56 (12): pp. 1707-1720, 2007.
- [17]. A.J. Soper, C. Walshaw, and M. Cross, "A combined evolutionary search and multilevel optimisation approach to graph-partitioning," Journal of Global Optimization, Vol. 29(2): pp. 225-241, 2004.
- [18]. C. M. Fiduccia, and R. M. Mattheyses, "A linear Time Heuristic for Improving Network Partitions," in Conf. on Design Automation, pp. 175 - 181, 1982.