

# Interactive Image Segmentation using Particle Competition and Cooperation

Fabricao Breve

Department of Statistics, Applied  
Mathematics and Computation (DEMAC)  
Institute of Geosciences  
and Exact Sciences (IGCE)  
São Paulo State University (UNESP)  
Rio Claro, São Paulo, Brazil  
Email: fabricao@rc.unesp.br

Marcos Gonçalves Quiles

Institute of Science  
and Technology (ICT)  
Federal University of  
São Paulo (Unifesp)  
São José dos Campos, SP, Brazil  
Email: quiles@unifesp.br

Liang Zhao

Department of Computer Science  
and Mathematics (DCM)  
School of Philosophy, Science and  
Literature in Ribeirão Preto (FFCLRP)  
University of São Paulo (USP)  
Ribeirão Preto, São Paulo, Brazil  
Email: zhao@usp.br

**Abstract**—Many interactive image processing approaches are based on semi-supervised learning, which employ both labeled and unlabeled data in its training process. In the interactive image segmentation problem, a human specialist labels some pixels of an object while the semi-supervised algorithm labels the remaining pixels of the segment. The particle competition and cooperation model is a recent graph-based semi-supervised learning approach. It employs particles walking in a graph to classify the data items corresponding to graph nodes. Each particle group aims to dominate most unlabeled nodes, spreading their label, and preventing enemy particles invasion. In this paper, the particle competition and cooperation model is extended to perform interactive image segmentation. Each image pixel is converted into a graph node, which is connected to its nearest neighbors according to their visual features and location in the original image. Labeled pixel generates particles that propagate their label to the unlabeled pixels. The particle model also takes the contributions from the adjacent pixels to classify less confident labeled pixels. Computer simulations are performed on real-world images, including images from the Microsoft GrabCut dataset, which allows a straightly comparison with other techniques. The segmentation results show the effectiveness of the proposed approach.

## I. INTRODUCTION

Image segmentation is the process of dividing a digital image into multiple parts (sets of pixels), identifying objects or other relevant information [1]. Image segmentation is considered one of the most difficult tasks in image processing [2]. Furthermore, fully automatic segmentation is still a big challenge and the techniques are often domain-dependent. Therefore, interactive image segmentation, which takes a partial supervision into account, has emerged as an interesting approach in the last decades [3]–[14].

Many interactive image processing approaches are based on semi-supervised learning (SSL). SSL is an important paradigm in the machine learning field specially in problems where unlabeled data is abundant and labeled data is expensive and/or time consuming, requiring an intense work of human specialists [15], [16]. SSL techniques employ both labeled and

unlabeled data in its training process, overcoming the limitations of both supervised and unsupervised learning approaches, which use only labeled or unlabeled data, respectively. In the interactive image segmentation problem, a human specialist could label some pixels far from the object borders, which is an easier and faster task than defining object boundaries manually. Thus, by taking the labeled and unlabeled pixels into account, the semi-supervised learning algorithm propagates the labels from the labeled pixels to their related remaining unlabeled pixels, or the image segments.

The particle competition and cooperation model [17] is a recent graph-based semi-supervised learning approach. Firstly, it converts vector-based data sets to non-weighted and undirected graphs, so each data item becomes a graph node, and edges are created between nodes corresponding to similar data items. Then, particles, which correspond to labeled data, walk in the graph in cooperation with other particles representing the same class and competing against particles representing other classes. Each particle group aims to dominate most unlabeled nodes, spreading their label, and preventing enemy particles invasion. At the end of the process, particles territory frontiers usually fall near the boundaries between classes, thus achieving high classification accuracy.

Differently from other semi-supervised graph-based methods [18]–[23], which are similar and share the same regularization framework [15], the particle competition and cooperation approach spread labels in a local fashion, through the particles, instead of globally. Therefore, its computational complexity is close to linear ( $O(N)$ ), where  $N$  is the number of samples in the dataset.

Although it is a relatively new approach, particle competition and cooperation was successfully extended and applied to other important machine learning problems, like overlapped community detection [24], [25], learning with label noise [26]–[28], learning with concept drift [29], [30], and combined active and semi-supervised learning [31]–[33]. However, in all those scenarios, the model was applied only to vector-based and graph-based data.

In this paper, the particle competition and cooperation model is extended to perform interactive image segmentation. First, each image pixel is converted into a graph node, which is

---

This work was supported by the São Paulo State Research Foundation (FAPESP) and the Brazilian National Research Council (CNPq).

connected to its  $k$ -nearest neighbors according to their visual features and location in the original image. Then, each labeled pixel generates a particle that will try to propagate its label to the unlabeled pixels. Furthermore, the particle dynamics are improved to decrease storage complexity and to allow the handling of larger images.

A “second phase” is also introduced to pick labels for nodes that are either undecided or that the particle approach have a low confidence regarding to the assigned label/segment. This scenario can be especially observed in border pixels or in pixels representing noise. Noise pixels do not share similar features with their neighborhood, thus they can become isolated nodes. Besides the nodes that represent noise pixels, whether the number of neighbors  $k$  is set to a low value, the graph might be composed of several components, and some of them can be come unreachable by the particles. To avoid those problems, in the second phase, only the undecided pixels take collaboration from their neighbor pixels (in the original image) proportional to their feature similarity, in an iterative fashion until all labels become stable.

Computer simulations are performed on some real-world images, including images from the Microsoft GrabCut dataset [9], so the segmentation accuracy may be compared with those achieved by some of the state-of-the-art algorithms. The results show the effectiveness of the proposed approach.

The rest of this paper is organized as follows. Section II presents an overview of the semi-supervised learning particle competition and cooperation model. In Section III, the interactive segmentation model based on particle competition and cooperation is described. Section IV shows some computer simulations. Finally, some conclusions are drawn on Section V.

## II. PARTICLE COMPETITION AND COOPERATION

Roughly speaking, the semi-supervised learning particle competition and cooperation method [17] works as follows. Firstly, the vector-based data set is converted to a non-weighted and undirected graph. Each graph node represents a sample. Edges between a pair of nodes are created if the Euclidean distance between them is below a threshold. Alternatively, a  $k$ -nearest neighbors approach can be used to establish the edges. Secondly, a particle is created for each labeled node. Particles corresponding to nodes with the same label define a team and cooperate among themselves to dominate the unlabeled nodes. On the other hand, particles corresponding to nodes with different labels compete against each other for the possession of the nodes. Finally, as the system runs, the particles walk in the graph following a random-greedy rule.

Each node has a set of domination levels, a level for each class of the problem. When a particle visits a node, it increases its class domination level and it decreases the other classes domination levels on that node. Each particle has a strength level, which changes according to the domination level of its class in the node being visited. Each particle also has a distance table, which stores the distance from its initial node, or the “home node”, to each node that it has visited. These tables are updated dynamically as particles walk on the graph. At the end of the iterative process, each sample is labeled accordingly to the class associated to the highest domination level.

For further details on the original particle competition and cooperation algorithm, see [17].

## III. INTERACTIVE IMAGE SEGMENTATION MODEL

Given a bidimensional image, the set of pixels are reorganized as  $\mathfrak{X} = \{x_1, x_2, \dots, x_L, x_{L+1}, \dots, x_N\}$ , such that  $\mathfrak{X}_L = \{x_i\}_{i=1}^L$  is the labeled pixel subset and  $\mathfrak{X}_U = \{x_i\}_{i=L+1}^N$  is the unlabeled pixels set.  $\mathfrak{L} = \{1, \dots, C\}$  is the set containing the labels.  $y : \mathfrak{X} \rightarrow \mathfrak{L}$  is the function associating each  $x_i \in \mathfrak{X}$  to its label  $y(x_i)$  in the final segmentation results. The proposed model will estimate  $y(x_i)$  for each unlabeled pixel  $x_i \in \mathfrak{X}_U$ .

For each pixel  $x_i$ , we build a set of 20 features, which are:

- 1) the pixel row location
- 2) the pixel column location
- 3) the red (R) component of the pixel
- 4) the green (G) component of the pixel
- 5) the blue (B) component of the pixel
- 6) the hue (H) component of the pixel
- 7) the saturation (S) component of the pixel
- 8) the value (V) component of the pixel
- 9) the average of R on the pixel and its adjacent pixels
- 10) the average of G on the pixel and its adjacent pixels
- 11) the average of B on the pixel and its adjacent pixels
- 12) the average of H on the pixel and its adjacent pixels
- 13) the average of S on the pixel and its adjacent pixels
- 14) the average of V on the pixel and its adjacent pixels
- 15) the standard deviation of the R on the pixel and its adjacent pixels
- 16) the standard deviation of the G on the pixel and its adjacent pixels
- 17) the standard deviation of the B on the pixel and its adjacent pixels
- 18) the standard deviation of the H on the pixel and its adjacent pixels
- 19) the standard deviation of the S on the pixel and its adjacent pixels
- 20) the standard deviation of the V on the pixel and its adjacent pixels

It is worth noting that for all measures considering the pixel neighborhood, we consider an 8-connected neighborhood, except on the borders where no wraparound is applied. All components are normalized to have mean 0 and standard deviation 1 and then scaled by a constant vector  $\lambda$  in order to emphasize/deemphasize each feature to the upcoming graph generation. The HSV components are obtained from the RGB components, using the method described in [34].

An undirected graph  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$  is generated, in which  $\mathbf{V} = \{v_1, v_2, \dots, v_N\}$  is the set of nodes, and  $\mathbf{E}$  is the set of edges  $(v_i, v_j)$ . Each node  $v_i$  corresponds to a pixel  $x_i$ . Two nodes  $v_i$  and  $v_j$  are connected if  $v_j$  is among the  $k$ -nearest neighbors of  $v_i$ , or vice-versa, using the Euclidean distance between  $x_i$  and  $x_j$  features. Otherwise,  $v_i$  and  $v_j$  are disconnected.

For each node  $v_i \in \{v_1, v_2, \dots, v_L\}$ , corresponding to a labeled pixel  $x_i \in \mathfrak{X}_L$ , a particle  $\rho_i$  is generated and its initial position is set to  $v_i$ . Each particle  $\rho_j$  has a variable  $\rho_j^\omega(t) \in [0, 1]$  to hold the particle strength, which defines how

much a particle impacts a node when visiting it. Particles initial strength are set to the maximum,  $\rho_j^\omega(0) = 1$ .

In [17], each particle also holds a distance table, dynamically update as the particles walk, to store the distance between the particle initial position and each node in the graph. The storage complexity of the whole set of distance tables is  $O(NP)$ , where  $N$  is the amount of nodes and  $P$  is the amount of particles. When handling large images, the amount of nodes and particles can be much higher than those obtained from many traditional semi-supervised learning data sets, so the distance tables may hit system memory limits. Therefore, we changed particles dynamics so particles on the same team (representing the same class) share the same distance table.

The storage complexity is reduced to  $O(NC)$ , where  $C$  is the amount of classes, which is usually much lower than  $P$ . The new distance tables are defined by  $\mathbf{d}_c(t) = \{d_c^1(t), \dots, d_c^N(t)\}$ . Each element  $d_c^i(t) \in [0, N-1]$  holds the distance in hops measured between node  $v_i$  and the closest labeled node from class  $c$ . Particles start knowing only that the distance to labeled nodes of their class is zero ( $d_c^i = 0$  if  $y(x_i) = c$ ). Other distances are set to the largest possible value ( $d_c^i = n - 1$  if  $y(x_i) \neq c$ ).

Each node  $v_i$  has a domination vector  $\mathbf{v}_i^\omega(t) = \{v_i^{\omega_1}(t), v_i^{\omega_2}(t), \dots, v_i^{\omega_C}(t)\}$ , where each element  $v_i^{\omega_c}(t) \in [0, 1]$  corresponds to the domination level from the class  $c$  over the node  $v_i$ . The sum of the domination levels in each node is always constant,  $\sum_{c=1}^C v_i^{\omega_c} = 1$ .

Nodes corresponding to labeled pixels have constant domination levels, with full domination by the corresponding class. On the other hand, nodes corresponding to unlabeled pixels are variable. They begin with all classes domination levels set equally, but they change as particles visits them. Therefore, for each node  $v_i$ , the domination vector  $\mathbf{v}_i^\omega$  is set as follows:

$$v_i^{\omega_c}(0) = \begin{cases} 1 & \text{if } x_i \text{ is labeled and } y(x_i) = c \\ 0 & \text{if } x_i \text{ is labeled and } y(x_i) \neq c \\ \frac{1}{C} & \text{if } x_i \text{ is unlabeled} \end{cases} \quad (1)$$

When a particle  $\rho_j$  visits a unlabeled node  $v_i$ , the node domination levels are updated as follows:

$$v_i^{\omega_c}(t+1) = \begin{cases} \max\{0, v_i^{\omega_c}(t) - \frac{0.1\rho_j^\omega(t)}{C-1}\} & \text{if } c \neq \rho_j^c \\ v_i^{\omega_c}(t) + \sum_{r \neq c} v_i^{\omega_r}(t) - v_i^{\omega_r}(t+1) & \text{if } c = \rho_j^c \end{cases} \quad (2)$$

where  $\rho_j^c$  represents the class label of particle  $\rho_j$ . Each particle  $\rho_j$  will change the node its visiting  $v_i$  by increasing the domination level of its class ( $v_i^{\omega_c}$ ,  $c = \rho_j^c$ ) while decreasing the domination levels of other classes ( $v_i^{\omega_c}$ ,  $c \neq \rho_j^c$ ). Remember that nodes corresponding to labeled pixels have constant domination levels, therefore (2) is not applied when a particle visits any of them.

A particle strength changes according to the domination level of its class in the node it is visiting. Therefore, at each iteration, a particle strength is updated as follows:  $\rho_j^\omega(t) = v_i^{\omega_c}(t)$ , where  $v_i$  is the node being visited, and  $c = \rho_j^c$ .

When a node  $v_i$  is being visited, the particle updates its class distance table as follows:

$$d_c^i(t+1) = \begin{cases} d_c^q(t) + 1 & \text{if } d_c^q(t) + 1 < d_c^i(t) \\ d_c^i(t) & \text{otherwise} \end{cases} \quad (3)$$

where  $d_c^q(t)$  is the distance from the previous visited node to the closest labeled node of the particle class, and  $d_c^i(t)$  is the distance from the node being visited to the closest labeled node of the particle class. The distance calculation is dynamic. Particles do not have knowledge of the graph connection patterns. They are only aware of which are the neighbors of the node they currently are. Unknown distances are discovered as the particles walk, and the distances may be updated as particles naturally find shorter paths to them.

At each iteration, the particle  $\rho_j$  chooses a node  $v_i$  to visit among the neighbors of its current node. The probability of choosing a node  $v_i$  is given by: a) the particle class domination on it,  $v_i^{\omega_c}$ , and b) the inverse of its distance,  $d_c^i$ , to the closest labeled node from the particle class, as follows:

$$p(v_i|\rho_j) = \frac{W_{qi}}{2 \sum_{\mu=1}^n W_{q\mu}} + \frac{W_{qi} v_i^{\omega_c} (1 + d_c^i)^{-2}}{2 \sum_{\mu=1}^n W_{q\mu} v_{\mu}^{\omega_c} (1 + d_c^{\mu})^{-2}}, \quad (4)$$

where  $q$  is the index of the node being visited by particle  $\rho_j$ ,  $c$  is the class label of particle  $\rho_j$ ,  $W_{qi} = 1$  if there is an edge between the current node and the node  $v_i$ , and  $W_{qi} = 0$  otherwise. A particle stays on the chosen node only if, after applying (2), its class domination level is the largest on that node; otherwise, a shock happens and the particle goes back to the node it was before, and stays there until the next iteration.

The average maximum domination levels of the nodes ( $\langle v_i^{\omega_m} \rangle$ ,  $m = \arg \max_c v_i^{\omega_c}$ ) is used to identify when the algorithm reached stability. This value may never converge, as there may be intense dispute on the nodes in classes' frontiers. Therefore, the iterations are stopped when there is no increase in this measure for a considerable amount of iterations. When it happens, most nodes are dominated by a single class. Therefore, for every node  $v_i$ , if  $v_i^{\omega_c} > 0.9$ , the class  $c$  is assigned to the corresponding pixel  $y(x_i) = c$ , and the remaining nodes (if any) are left unlabeled for now, as the algorithm has low confidence in the label it would assign to them.

If unlabeled nodes still exist, the algorithm begins its second phase. At this point, the unlabeled pixels are mostly those that were under intense dispute by two or more classes, and/or those that were unreachable by all the particles. The second scenario may happen if a pixel (or set of pixels) are too different from any labeled pixels, specially when  $k$  is low, which may lead to a disconnected graph.

During the second phase, at each iteration, each unlabeled pixel  $x_i$  will set its corresponding  $\mathbf{v}_i^\omega$  as follows:

$$\mathbf{v}_i^\omega(t+1) = \frac{1}{a} \sum_{j \in \eta} \mathbf{v}_j^\omega(t) \text{dist}(x_i, x_j), \quad (5)$$

where  $\eta \subset \mathcal{X}$  is the subset of the  $a$  adjacent pixels of  $x_i$ .  $a = 8$ , except on pixels from the border of the image, which have only 3 or 5 adjacent pixels.  $\text{dist}(x_i, x_j)$  is a function that returns the Euclidean distance between  $x_i$  and  $x_j$  features. So, each unlabeled pixel will get contributions from its adjacent pixels, which are proportional to the similarity it has to each

of them. The second phase ends when  $\langle v_i^{\omega_m} \rangle$  stabilizes. The remaining unlabeled pixels are then labeled using  $y(x_i) = \arg \max_c v_i^{\omega_c}(t)$ .

#### IV. COMPUTER SIMULATIONS

In this section, computer simulation using real-world images are presented in order to show the effectiveness of the proposed method. The parameters  $k$  and  $\lambda$  are optimized for each image using the genetic algorithm available in Global Optimization Toolbox of MATLAB, with its default parameters.

Figure 1a shows an original image with  $576 \times 432$  pixels. A trimap providing seed regions is presented in Figure 1b. Black (0) represents the background, ignored by the algorithm; dark gray (64) is the labeled background, which generates background class particles; light gray (128) is the unlabeled region, which labels will be estimated by the proposed method; and white (255) is the labeled foreground, which generates the foreground class particles. Figure 1c shows the close-up foreground segmentation results achieved by the proposed method.

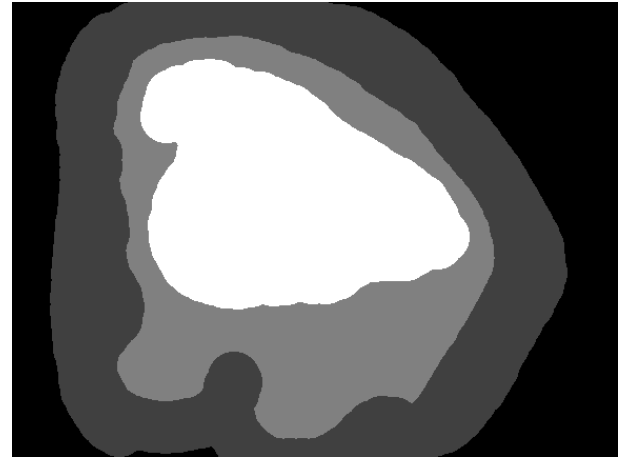
In order to compare the proposed method segmentation accuracy with those obtained by state-of-the-art methods, computer simulations are performed on five images from the Microsoft GrabCut dataset<sup>1</sup> [9]. The Grabcut dataset is used to evaluate interactive segmentation methods in many papers, thus the proposed method results may be compared with those achieved by other methods. Those specific five images were selected because the error rate and close-ups foreground images obtained on them by a few state-of-the-art algorithms are presented in [7], therefore one can compare those results with the results achieved by the proposed method, both numerically and visually. The five images, the trimaps providing seed regions, and the original ground truth are shown on Figure 2.

Figure 3 shows the close-up foreground segmentation results obtained using the proposed method, as well as the classification error rates. The error rates are computed as the ratio of the number of incorrectly classified pixels to the total amount of unlabeled pixels (light gray on the trimap images). Notice that the ground truth images (Figure 2c) have a thin band of gray pixels which corresponds to uncertainty, i.e., pixels that received different labels by the multiple persons who annotated them. These pixels are not counted in the error rate calculation, as done in [7], to ensure consistency. By comparing the results on Figure 3 with those presented in [7], the proposed method achieved lower error rates in three of the five images. Table I shows the error rate comparison between the directed hypergraph model(DINH) [7] and our proposed particle competition and cooperation method (PCC).

Table II shows the error rates and optimized parameters values used for each of the segmented images on Figure 3. Even though the variation is high, by observing this table one can notice that the two most important features are probably those related to the pixel location on the image, followed by the V (value) component on HSV. It is interesting to observe



(a)



(b)



(c)

Fig. 1. (a) Original image, (b) trimap providing seed regions, and (c) close-up foreground segmentation results by the proposed method.

<sup>1</sup>Available at <http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm>



Fig. 2. (a) Original images from the GrabCut dataset, (b) the trimaps providing seed regions, and (c) the original ground truth.





(a) Error rate: 6.19%

(b) Error rate: 7.67%



(c) Error rate: 1.53%



(d) Error rate: 1.24%



(e) Error rate: 1.09%

Fig. 3. Close-up foreground segmentation results from the proposed method.

TABLE I. COMPARISON OF THE ERROR RATES OBTAINED BY THE DIRECTED HYPERGRAPH MODEL (DINH) [7] AND THOSE OBTAINED BY OUR PROPOSED PARTICLE COMPETITION AND COOPERATION METHOD (PCC)

Image	DINH	PCC
208001 (mushroom)	<b>3.65%</b>	6.19%
271008 (children)	<b>5.12%</b>	7.67%
person7	2.82%	<b>1.53%</b>
sheep	5.85%	<b>1.24%</b>
124084 (flowers)	3.51%	<b>1.09%</b>

that, in fact, many earlier segmentation methods actually rely only on pixel coordinates and luminance values, ignoring other color components.

## V. CONCLUSION

This paper proposes a new nature-inspired interactive segmentation method based on the semi-supervised learning method known as particle competition and cooperation. In this method, particles walk in a graph generated from the image to be segmented. Particles representing the same object/class cooperate with each other to dominate the unlabeled pixels, while particles representing different object/classes compete against each other to avoid invasion from enemy particles in the nodes they already dominated. Pixels that are labeled with low confidence by the particle competition and cooperation method pass through a second phase, where they get contribution from their adjacent pixels in the original image, in order to get their definitive label.

Some computer simulations using real-world images were performed. The proposed method has showed foreground segmentation accuracy, which is comparable to those, achieved by some state-of-the-art methods, and even slightly better accuracy in some specific images. As future work, we intend to segment a larger set of images, to refine the model, and to find some pattern that may arise in the optimal choice of parameters, so they could be automatically selected based only on image properties.

## REFERENCES

- [1] L. Shapiro and G. Stockman, *Computer Vision*. Prentice Hall, 2001.
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2008.
- [3] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, 2001, pp. 105–112 vol.1.
- [4] L. Grady, "Random walks for image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 11, pp. 1768–1783, Nov 2006.
- [5] A. Protiere and G. Sapiro, "Interactive image segmentation via adaptive weighted distances," *Image Processing, IEEE Transactions on*, vol. 16, no. 4, pp. 1046–1057, April 2007.
- [6] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive gmmrf model," in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Springer Berlin Heidelberg, 2004, vol. 3021, pp. 428–441. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-24670-1\\_33](http://dx.doi.org/10.1007/978-3-540-24670-1_33)

TABLE II. ERROR RATES AND OPTIMIZED PARAMETERS VALUES FOR EACH SEGMENTED IMAGE

Image	208001 (mushroom)	271008 (children)	person7	sheep	124084 (flowers)	Mean	Standard Deviation
<b>Error Rate</b>	0.0619	0.0767	0.0153	0.0124	0.0109	<b>0.0354</b>	<b>0.0314</b>
$k$	128	70	90	250	648	<b>237.20</b>	<b>240.04</b>
$\lambda(1) - X$	0.8265	0.9044	0.1108	0.4362	0.7283	<b>0.6012</b>	<b>0.3265</b>
$\lambda(2) - Y$	0.9578	0.8023	0.9885	0.9929	0.7566	<b>0.8996</b>	<b>0.1117</b>
$\lambda(3) - R$	0.1842	0.0119	0.2068	0.9232	0.3846	<b>0.3421</b>	<b>0.3506</b>
$\lambda(4) - G$	0.3626	0.2187	0.2740	0.0373	0.4162	<b>0.2617</b>	<b>0.1470</b>
$\lambda(5) - B$	0.2635	0.0103	0.4578	0.7513	0.2974	<b>0.3561</b>	<b>0.2730</b>
$\lambda(6) - H$	0.7822	0.8711	0.4061	0.0581	0.1098	<b>0.4455</b>	<b>0.3738</b>
$\lambda(7) - S$	0.1054	0.4775	0.2212	0.2590	0.0442	<b>0.2214</b>	<b>0.1672</b>
$\lambda(8) - V$	0.1971	0.0605	0.5401	0.8498	0.9371	<b>0.5169</b>	<b>0.3869</b>
$\lambda(9) - R-M$	0.0516	0.0434	0.6069	0.9529	0.1565	<b>0.3623</b>	<b>0.4029</b>
$\lambda(10) - G-M$	0.0864	0.0859	0.3610	0.3075	0.9922	<b>0.3666</b>	<b>0.3715</b>
$\lambda(11) - B-M$	0.1837	0.0452	0.6721	0.2743	0.9179	<b>0.4186</b>	<b>0.3637</b>
$\lambda(12) - H-M$	0.0703	0.1742	0.1875	0.6234	0.0014	<b>0.2113</b>	<b>0.2428</b>
$\lambda(13) - S-M$	0.0607	0.1096	0.2475	0.2876	0.1326	<b>0.1676</b>	<b>0.0959</b>
$\lambda(14) - V-M$	0.2094	0.0003	0.1361	0.3478	0.0536	<b>0.1494</b>	<b>0.1365</b>
$\lambda(15) - R-S$	0.2920	0.9367	0.0912	0.1853	0.2503	<b>0.3511</b>	<b>0.3360</b>
$\lambda(16) - G-S$	0.9764	0.3095	0.5375	0.5450	0.0838	<b>0.4905</b>	<b>0.3315</b>
$\lambda(17) - B-S$	0.0835	0.1138	0.4314	0.8362	0.2273	<b>0.3384</b>	<b>0.3099</b>
$\lambda(18) - H-S$	0.5134	0.3830	0.0602	0.0471	0.2915	<b>0.2590</b>	<b>0.2035</b>
$\lambda(19) - S-S$	0.0242	0.0827	0.4700	0.5837	0.5223	<b>0.3366</b>	<b>0.2624</b>
$\lambda(20) - V-S$	0.0190	0.0551	0.1894	0.0851	0.0457	<b>0.0788</b>	<b>0.0662</b>

- [7] A. Ducournau and A. Bretto, "Random walks in directed hypergraphs and application to semi-supervised image segmentation," *Computer Vision and Image Understanding*, vol. 120, no. 0, pp. 91 – 102, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1077314213002038>
- [8] L. Ding and A. Yilmaz, "Interactive image segmentation using probabilistic hypergraphs," *Pattern Recognition*, vol. 43, no. 5, pp. 1863 – 1873, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320309004440>
- [9] C. Rother, V. Kolmogorov, and A. Blake, "'grabcut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, Aug. 2004. [Online]. Available: <http://doi.acm.org/10.1145/1015706.1015720>
- [10] A. Paiva and T. Tasdizen, "Fast semi-supervised image segmentation by novelty selection," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 1054–1057.
- [11] J. Li, J. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 11, pp. 4085–4098, Nov 2010.
- [12] Y. Artan and I. Yetik, "Improved random walker algorithm for image segmentation," in *Image Analysis Interpretation (SSIAI), 2010 IEEE Southwest Symposium on*, May 2010, pp. 89–92.
- [13] Y. Artan, "Interactive image segmentation using machine learning techniques," in *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, May 2011, pp. 264–269.
- [14] J. Xu, X. Chen, and X. Huang, "Interactive image segmentation by semi-supervised learning ensemble," in *Knowledge Acquisition and Modeling, 2008. KAM '08. International Symposium on*, Dec 2008, pp. 645–648.
- [15] X. Zhu, "Semi-supervised learning literature survey," *Computer Sciences, University of Wisconsin-Madison, Tech. Rep.* 1530, 2005.
- [16] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA: The MIT Press, 2006.
- [17] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, and J. Liu, "Particle competition and cooperation in networks for semi-supervised learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 9, pp. 1686 –1698, sept. 2012.
- [18] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2001, pp. 19–26.
- [19] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the Twentieth International Conference on Machine Learning*, 2003, pp. 912–919.
- [20] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, vol. 16. MIT Press, 2004, pp. 321–328. [Online]. Available: <http://www.kyb.tuebingen.mpg.de/bs/people/weston/localglobal.pdf>
- [21] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semisupervised learning on large graphs," in *Conference on Learning Theory*. Springer, 2004, pp. 624–638.
- [22] M. Belkin, N. P., and V. Sindhwani, "On manifold regularization," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*. New Jersey: Society for Artificial Intelligence and Statistics, 2005, pp. 17–24.
- [23] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proceedings of International Conference on Machine Learning*. AAAI Press, 2003, pp. 290–297.
- [24] F. Breve and L. Zhao, "Fuzzy community structure detection by particle competition and cooperation," *Soft Computing*, vol. 17, no. 4, pp. 659–673, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s00500-012-0924-3>
- [25] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, and J. Liu, "Particle competition and cooperation for uncovering network overlap community structure," in *Advances in Neural Networks – ISNN 2011*, ser. Lecture Notes in Computer Science, D. Liu, H. Zhang, M. Polycarpou, C. Alippi, and H. He, Eds. Springer Berlin Heidelberg, 2011, vol. 6677, pp. 426–433. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-21111-9\\_48](http://dx.doi.org/10.1007/978-3-642-21111-9_48)
- [26] F. A. Breve, L. Zhao, and M. G. Quiles, "Semi-supervised learning from imperfect data through particle cooperation and competition," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, July 2010, pp. 1–8.
- [27] —, "Particle competition and cooperation for semi-supervised learning with label noise," *Neurocomputing*, 2015, article in Press.
- [28] F. Breve and L. Zhao, "Particle competition and cooperation to prevent error propagation from mislabeled data in semi-supervised learning," in *Neural Networks (SBRN), 2012 Brazilian Symposium on*, Oct 2012, pp. 79–84.
- [29] —, "Particle competition and cooperation in networks for semi-supervised learning with concept drift," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, June 2012, pp. 1–6.
- [30] F. A. Breve and L. Zhao, "Semi-supervised learning with concept drift using particle dynamics applied to network intrusion detection data," in *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC), 2013 BRICS Congress on*, Sept 2013, pp. 335–340.

- [31] F. Breve, "Active semi-supervised learning using particle competition and cooperation in networks," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Aug 2013, pp. 1–6.
- [32] F. A. Breve, "Combined active and semi-supervised learning using particle walking temporal dynamics," in *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC), 2013 BRICS Congress on*, Sept 2013, pp. 15–20.
- [33] —, "Query rules study on active semi-supervised learning using particle competition and cooperation," in *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, São Carlos, 2014, pp. 134–140.
- [34] A. R. Smith, "Color gamut transform pairs," in *ACM Siggraph Computer Graphics*, vol. 12, no. 3. ACM, 1978, pp. 12–19.