

# A Clustering Game Based Framework for Image Segmentation

Dan Shen, Erik Blasch, Khanh Pham, and Genshe Chen

**Abstract**—Image segmentation decomposes a given image into segments, i.e. regions containing “similar” pixels, that aids computer vision applications such as face, medical, and fingerprint recognition as well as scene characterization. Effective segmentation requires domain knowledge or strategies for object designation as no universal segmentation algorithm exists. In this paper, we propose a holistic framework to perform image segmentation in color space. Our approach unifies the linear smoothing filter, a similarity calculation in selected color space, and a clustering game model with various evolution dynamics. In our framework, the problem of image segmentation can be considered as a “clustering game”. Within this context, the notion of a cluster turns out to be equivalent to a classical equilibrium concept from game theory, as the game equilibrium reflects both the internal and external cluster conditions. Experiments on image segmentation problems show the superiority of the proposed *clustering game based image segmentation framework (CGBISF)* using both the Berkeley segmentation dataset and infrared images (for which, we need to perform color fusion first) in autonomy, speed, and efficiency.

## I. INTRODUCTION

Image segmentation is usually the first step in many applications of computer vision. It partitions a digital image into multiple regions or sets of pixels with similar characteristics. The goal of image segmentation is to simplify the image and represent the image in a more meaningful way for subsequent image processes. In the literature, several algorithms and techniques have been developed for image segmentation (see surveys [1], [2], [3]). The first category is *region-based segmentation*. Region growing starts from some pixels or seeds representing distinct image regions and the grows regions until the entire image is covered. For region growing, we need a rule to describe a growth mechanism and a rule to check the homogeneity of the region after each growth step [4], [5], [6]. The opposite approach to region growing is region shrinking (or splitting). Region shrinking is a top-down approach and it starts with the assumption that the entire image is homogeneous. If an image is not homogeneous, the image is split into four sub images where the splitting procedure is repeated recursively until the image is split into homogeneous regions. Next, in the second step, all adjacent regions with similar attributes may be merged following other (or the same) criteria. Typical split and merge techniques are covered in [7], [8], [9].

Manuscript received February 15, 2012.

D. Shen and G. Chen are with I-Fusion Technologies, Inc., Germantown, MD 20874, USA (email: {dshen, gchen}@i-fusion-i.com).

E. P. Blasch is with the Air Force Research Lab, Rome, NY USA (e-mail: erik.blasch@wpafb.af.mil).

K. Pham is with the Air Force Research Lab, Albuquerque, NM (e-mail: AFRL.RVSV@kirtland.af.mil).

The second category is *edge-based segmentation*. Edge-based segmentation technology performs image segmentation through region boundary detection, which can be conducted locally or globally. The local technique determines an edge point and only needs information in the neighborhood of that point. The global technique [10], [11], [12] makes a sort of global optimization, and therefore the given edge point could be identified after many optimization steps involving changes over a large area. *Pixel-based segmentation* is another category. Histogram-based methods [13], [14] are very efficient when compared to other image segmentation methods because they typically require only one pass through the pixels. In this technique, a histogram is computed from all of the pixels in the image, and the peaks and valleys in the histogram are used to locate the clusters in the image. For example, color or intensity can be used as the measure. Segmentation using clustering [15], [16], [17], [18], [19] involves the search for image pixels that are similar enough to be grouped together. In this way, a cluster corresponds to a segment of an image.

In this paper, we apply evolutionary cluster games [20] to image segmentation which we develop a *clustering game based color image segmentation framework (CGBCISF)* approach. Unlike the game approach [21] for edge date, in our game engine, pixel clusters are derived as a result of the competition of individuals playing the so-called “clustering game”, which is a symmetric evolutionary game [22], [23] involving two players. Each player simultaneously selects pixels that should be clustered and, after having revealed his choice, he receives a payoff according to the similarity that the selected pixels have with respect to the opponents’ ones. The evolutionary stable equilibrium (ESS) can determine a cluster of pixels for image segmentation.

The rest of the paper is organized as follows. In section II, the evolutionary game theory is reviewed as preliminaries. Section III describes the proposed framework for color image segmentation. Segmentation experiments on both the Berkeley dataset and noisy IR images are demonstrated in Section IV. Finally we draw conclusions in Section V.

## II. EVOLUTIONARY GAME THEORY

Evolutionary game theory [22], [23] originated as an application of the mathematical theory of games to biological contexts where a realization of a frequency dependent fitness function introduces a strategic aspect to evolution. Unlike classical game theory, which deals with individual players who have to decide between different strategies to maximize its own payoff or reward functions, evolution game theory focuses on the dynamics of change strategies for the entire

population of players. Strong or fit options with high payoff will spread within the population via an inheriting process.

For a two-player symmetric evolutionary game, each player has the same payoff function. Let  $S = \{1, 2, \dots, n\}$  be the set of pure strategies available to the players and  $A = (a_{ij})$  is a  $n \times n$  payoff or utility matrix, where  $a_{ij}$  is the payoff that a player gains when playing the strategy  $i$  against an opponent playing strategy  $j$ . A mixed strategy is an assignment of a probability distribution  $x = (x_1, x_2, \dots, x_n)$  over the set of pure strategy  $S$ . Clearly, the mixed strategies lie in the standard simplex  $\Delta$  of the  $n$ -dimensional Euclidean space,  $\Delta = \{x \in \mathcal{R}^n | e^\top x = 1, x \geq 0\}$ , where  $e$  is an appropriately sized vector consisting of 1s and  $\top$  is the transpose operator. For the image segmentation problem,  $n$  is the number of the sampled pixels in a picture. For example, the  $n$  value for a  $300 \times 200$  image (with a sample rate 0.005) is 300, and a possible  $x = 1/300 \cdot \text{ones}(300, 1)$ .

The expected payoff or reward,  $\pi$ , that a player obtains by playing the strategy  $y \in \Delta$  against an opponent's  $x \in \Delta$  strategy is (note that two player share a same simplex because of the symmetric game setup),

$$\pi(y|x) = \sum_{i \in S} y_i \sum_{j \in S} a_{ij} x_j = y^\top A x \quad (1)$$

From (1), we have

$$\pi(x|x) = x^\top A x \triangleq \pi(x)$$

$$\pi(y-x|x) = \pi(y|x) - \pi(x|x) = \pi(y|x) - \pi(x)$$

$$\pi(y-x) = \pi(y-x|y-x) = \pi(y-x|y) - \pi(y-x|x)$$

The support of a mixed strategy  $x$ , denoted by  $\sigma(x)$ , defines the set of pure strategies with non-zero probability,

$$\sigma(x) = \{x_i \in S \mid x_i > 0\} \subseteq S \quad (2)$$

For a mixed strategy  $x$ , we define *deteriorating* pure strategies set, *improving* pure strategies set, and *balancing* pure strategies set by

$$\phi_-(x) = \{i \in S | \pi(e^i - x|x) < 0\} \quad (3)$$

$$\phi_+(x) = \{i \in S | \pi(e^i - x|x) > 0\} \quad (4)$$

$$\phi_=(x) = \{i \in S | \pi(e^i - x|x) = 0\} \quad (5)$$

respectively, where  $e^i$  is the  $i^{\text{th}}$  column of the identity matrix with proper size.

The best replies,  $\beta(x)$ , is the set of mixed strategies that maximize the expected payoff when played against  $x$ ,

$$\beta(x) = \{\arg \max_{z \in \Delta} \pi(z|x)\} \quad (6)$$

A strategy pair  $(x, y) \in \Delta^2$  is a Nash equilibrium (NE) if no player has anything to gain by changing only his own strategy unilaterally,  $x \in \beta(y)$  and  $y \in \beta(x)$ . For a symmetric game, only symmetric pairs  $(x, x)$  are of interest. Therefore, we call a mixed strategy a Nash equilibrium if it is the best reply to itself.

$$x \text{ is a NE} \iff x \in \beta(x) \iff \forall y \in \Delta, \pi(y-x|x) \leq 0 \quad (7)$$

A mixed strategy  $x$  is said to be evolutionary stable strategy (ESS) if and only if  $x$  is a NE and for any best reply  $y$  to  $x$  different from  $x$ , we have  $\pi(x-y|y) > 0$ .

$$x \text{ is a ESS} \iff \begin{cases} \text{equilibrium condition:} \\ x \text{ is a NE} \\ \text{AND} \\ \text{stability condition:} \\ \forall y \in \beta(x), y \neq x \Rightarrow \pi(y-x|x) < 0. \end{cases}$$

### III. A HOLISTIC FRAMEWORK FOR IMAGE SEGMENTATION

In this section, we designed a framework with the flowchart shown in Fig. 1. For non-color images (for example IR images), we first convert it to a virtual color images by the channel based color fusion algorithms [25]. Then we apply the linear smooth filters to reduce the image noise. Since the similarity matrix deriving from whole image may be dramatically large, we apply a random sample mechanism of complexity reduction to speed up the segmentation. For the unlabeled sampled pixel set, we calculate the similarity matrix and call a clustering game engine to extract and label one segment. After that, we need to recalculate the similarity matrix for the updated unlabeled sampled pixel set and call the clustering game again until all sampled pixels are labeled. Once the set of clusters has been extracted from the sampled pixel set, the remaining un-sampled pixels are assigned to the most similar cluster.

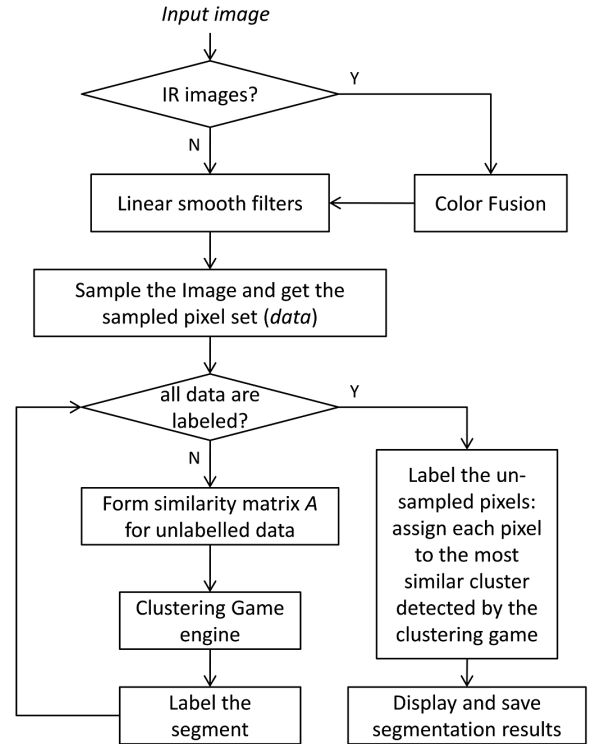


Fig. 1: A framework of the clustering game based image segmentation (CGIS)

### A. Clustering Game Model

The clustering game is an evolutionary game for the clustering, which can be considered as an important unsupervised learning activity aimed at finding a structure in a collection of unlabeled data. A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.

The clustering game model is represented by a 7-tuple  $G_c = \langle S, A, X, P_1, P_2, D_1, D_2 \rangle$ , where

- $S$  denotes the set of objects or data to be clustered,  $S = 1, 2, \dots, n$ .
- $A$  is a payoff matrix or similarity matrix. For binary clustering [24], a similarity matrix, which represents the similarities between objects to cluster, will be used as a payoff matrix in the clustering game model. In the image segmentation, an element  $(A_{ij})$  is the distance in a color space between pixel  $i$  and pixel  $j$ .
- $X$  is the population state space.  $X = \{x \in \Delta\}$ .
- $P_1$  and  $P_2$  are the choices (current and past) of the player 1 and player 2.
- $D_1$  and  $D_2$  are the decisions of player 1 and player 2.

$$D_1 : S \times A \times P_1 \times P_2 \rightarrow \Delta$$

$$D_2 : S \times A \times P_2 \times P_1 \rightarrow \Delta$$

Given a set of objects  $S$ , whose pair-wise similarities are stated by the  $n \times n$  matrix  $A$ , we can design a clustering game between two players with complete knowledge of the game setup. The players simultaneously select an element from  $S$ . After both have shown their choice, each player receives a payoff, monetary or otherwise, proportional to the similarity that the chosen element has with respect to the element chosen by the opponent.

*Remark:* Note that, since every object is by definition strongly similar to itself, setting the diagonal of the payoff matrix to zero, or in general to a sufficiently low value, is of fundamental importance. Otherwise, the best strategy for each player would be to coordinate the selection towards exactly the same object.

### B. Evolutionary dynamics

For evolutionary games, the population state dynamics determine the game solution. In this section, we present three evolutionary dynamics.

1) Best response dynamics and fictitious play: The method of fictitious play (FP) [26] was the first dynamics studied in game theory. In the framework of FP, each player presumes that its opponents are playing stable (possibly mixed) strategies. Each player starts with some initial beliefs and chooses a best response to those beliefs as a strategy in the first round. Then, after observing their opponents' actions, the players update their beliefs according to some learning rule (e.g. reinforcement learning or Bayes' rule). The process is then repeated.

Given the population state (at time  $t$ ),  $x(t) \in \Delta$ , the player chooses the best response  $r(t+1) \in \Delta(x(t))$  at time  $t+1$ .

Then the new population state is determined by

$$x(t+1) = \frac{r(t+1) - x(t)}{t+1} + x(t) \quad (8)$$

where,  $r(t+1) \in \beta(x(t))$ .

The FP evolutionary dynamics are greedy but myopic strategies. It is proved in [27] that *i*) in fictitious play, Nash equilibria are absorbing states. That is, if at any time period all the players play a Nash equilibrium, then they will do so for all subsequent rounds, and *ii*) if fictitious play converges to any distribution (or mixed strategy), those probabilities correspond to a Nash equilibrium of the underlying game.

2) Replicator dynamics: The Replicator Dynamics (RD), which was introduced in [28] describes a selection process, inspired by Darwinian processes. Consider a population in which individuals, called replicators, exist in several different types. Each type of individual uses a pre-programmed strategy (replicator equation) and passes this behavior to its descendants without modification. The replicator equation differs from other equations used to model replication, such as the quasi-species equation, in that it allows the fitness landscape to incorporate the distribution of the population types rather than setting the fitness of a particular type constant. The population distribution property allows the replicator equation to capture the essence of selection.

For our clustering game model, the evolutionary dynamics is given by

$$x_i(t+1) = x_i(t) \frac{\pi(e^i|x) + \kappa}{\pi(x) + \kappa} \quad (9)$$

where  $x_i(t)$  is the  $i^{\text{th}}$  element of  $x(t)$ ,  $e^i$  is the  $i^{\text{th}}$  column of the identity matrix, and  $\kappa > 0$  is a constant.

The replicator equation satisfies the Folk theorem of evolutionary game theory, (i.e. any outcome of a repeated game can be sustained as an equilibrium if the minimax condition for both players is satisfied) which characterizes the stability of equilibria of the replicator equation.

3) Infection and immunization dynamics: In order to overcome some computational problems afflicting standard evolutionary dynamics (such as FP and RD), Bulò and Bomze [20] introduced a new class of evolutionary dynamics, inspired by infection and immunization processes. For the situation that a share  $\varepsilon$  of mutant population  $y$  invading the incumbent population  $x$ , we define the score function of  $y$  versus  $x$  as

$$\begin{aligned} h_x(y, \varepsilon) &= \pi(y-x|(1-\varepsilon)x + \varepsilon y) \\ &= \varepsilon \pi(y-x) + \pi(y-x)|x \end{aligned} \quad (10)$$

and define  $\delta$  as an invasion barrier:

$$\delta_y(x) = \inf\{\varepsilon \in (0, 1) | h_x(y, \varepsilon) \leq 0\} \cup \{1\} \quad (11)$$

In the clustering game with two players, the invasion barrier can be simplified as

$$\delta_y(x) = \begin{cases} \min\left[\frac{\pi(x-y|y)}{\pi(y-x)}, 1\right], & \text{if } \pi(y-x) < 0 \\ 1, & \text{otherwise} \end{cases} \quad (12)$$

The set of infective strategies for  $x$  is defined as follows

$$\Upsilon(x) = \{y \in \Delta \mid \pi(y - x \mid x) > 0\} \quad (13)$$

Then the general infection and immunization dynamics (*InImDyn*) is

$$x(t+1) = \delta_{\Xi(x(t))}(x(t))[\Xi(x(t)) - x(t)] + x(t) \quad (14)$$

where  $\Xi : \Delta \rightarrow \Delta$  is a strategy selection function, which returns an infective strategy for  $x$  if it exists, or  $x$  otherwise:

$$\Xi(x) = \begin{cases} \text{some } y \in \Upsilon(x), & \text{if } \Upsilon(x) \neq \emptyset \\ x, & \text{otherwise} \end{cases} \quad (15)$$

Depending on how we choose the function  $\Xi(x)$ , we may obtain different *InImDyn*. One in particular, which is simple and leads to nice properties, consists of allowing only infective pure strategies or their respective co-strategies, which is defined as follows.

The co-strategy of  $y$  with respect to  $x$  is given by  $\bar{y}_x$ ,

$$\bar{y}_x = (1 - \bar{\rho})x + \bar{\rho}y \quad (16)$$

$$\bar{\rho} = \min\{\rho \in \mathcal{R} \mid (1 - \rho)x + \rho y \in \Delta\} \leq 0 \quad (17)$$

Specially, if  $y = e^i$  (pure strategy), then  $\bar{\rho} = x_i / (1 - x_i)$ . In this case, *InImDyn* becomes *PureInImDyn*.

### C. Similarity Model in Color Space

A number of color spaces have been used in the segmentation of color image, such as *RGB*, *CIE XYZ*, *CIE*, *HSL*, *YUV* and *YIQ* etc. A color space is uniform, if the equal distance in the color space corresponds to equal perceived color differences. Many color spaces are non-uniform. For example, Red-Green-Blue (RGB) color space is far from exhibiting the perceptual uniformity as it does not model the way that a human perceives colors. From the original Commission on Illumination (CIE), new research [29] concluded that modified CIE  $L^*u^*v^*$  (*mLuv*) performs better than other color spaces. In this paper, similarity matrix is calculated in the *mLuv* space.

Given each pixel  $i$  is represented by a 3-dimensional vector  $C(i)$  in the color space, the similarity  $a_{ij}$  between pixels  $i$  and  $j$  is calculated using the standard Gaussian kernel:

$$a_{ij} = \exp \left[ \frac{-\|C(i) - C(j)\|^2}{\lambda^2} \right] \quad (18)$$

where  $\lambda > 0$  is a user-defined constant.

## IV. IMAGE SEGMENTATION EXPERIMENTS

### A. Color Images

We have tested our clustering game image segmentation (CGIS) approach on the Berkeley segmentation dataset [30]. For the input image (Fig. 3a), we first convert it to *mLuv* color space (Fig. 3b). Then we obtain the partition images (Fig. 3d, Fig. 3f, Fig. 3h) and average color images (Fig. 3c, Fig. 3e, Fig. 3g) of clustering games with *PureInImDyn*, *RD*, and *FP*, respectively. The sample rate is 0.005. In an average color image, the RGB value of a pixel equals to the average

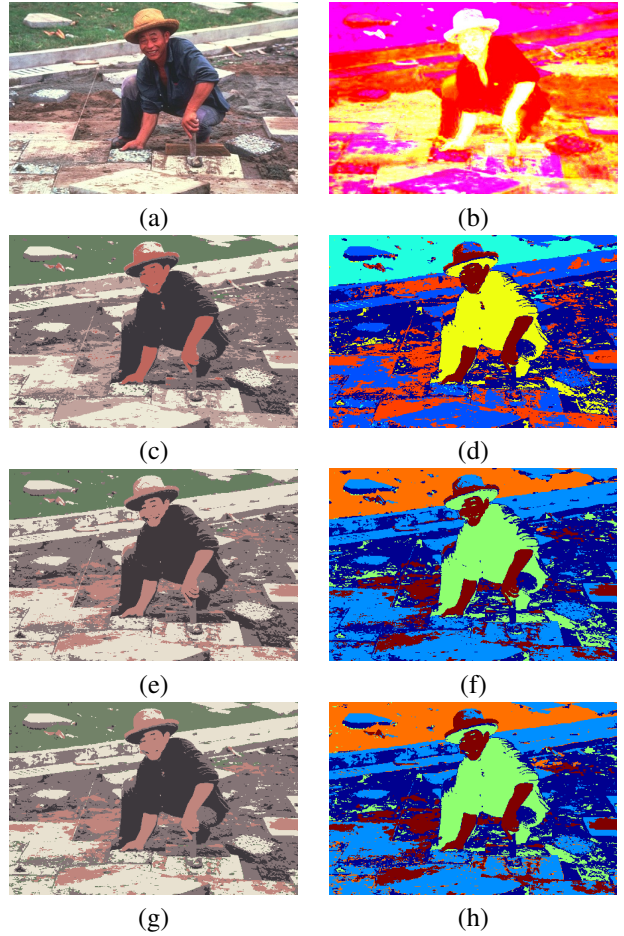


Fig. 2: Image segmentation results. *a.* original (481 321 resolution), *b.* modified CIE  $L^*u^*v^*$  displayed as RGB, *c.* Pure *InImDyn* (average color, 6 clusters), *d.* Pure *InImDyn* (partition image), *e.* Replicator dynamics (average color, 5 clusters), *f.* Replicator dynamics (partition image), *g.* Fictitious play (average color, 5 clusters), and *h.* Fictitious play (partition image)

RGB of all pixels in the same segmentation. So the total number of colors is same as the segment number.

The population state vectors of the clustering game with three different evolutionary dynamics are shown in Fig. 4, with each line representing a pixel. The initial values of all pixels are set to  $1/(\text{No. pixels})$ . After the convergence, we obtain the cluster from its support  $\sigma(x)$ .

We also compared our clustering game approach (with pure *InImDyn*) against the Nyström method [31] on the Berkeley dataset, summarized in Fig. 5 and Table 1.

### B. IR Images

For an IR image (Fig. 6a), we apply the channel based color fusion [25] with a similar RGB picture and obtained the virtual color image (Fig. 6b). The segmentation results of *PureInImDyn* clustering game method without linear smoothing filters are shown in Fig. 6c and Fig. 6d.

We can see a lot of noisy segments. After applying the linear smoother filters, we obtain the partition results in

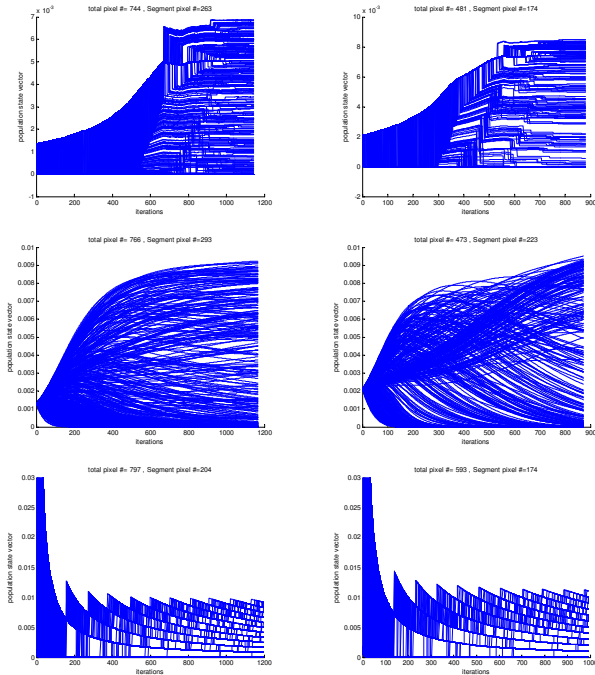


Fig. 3: The population state vectors of the clustering game with  $pr$  (first row),  $RD$  (second row) and  $FP$  (third row). only first two segments are shown here.

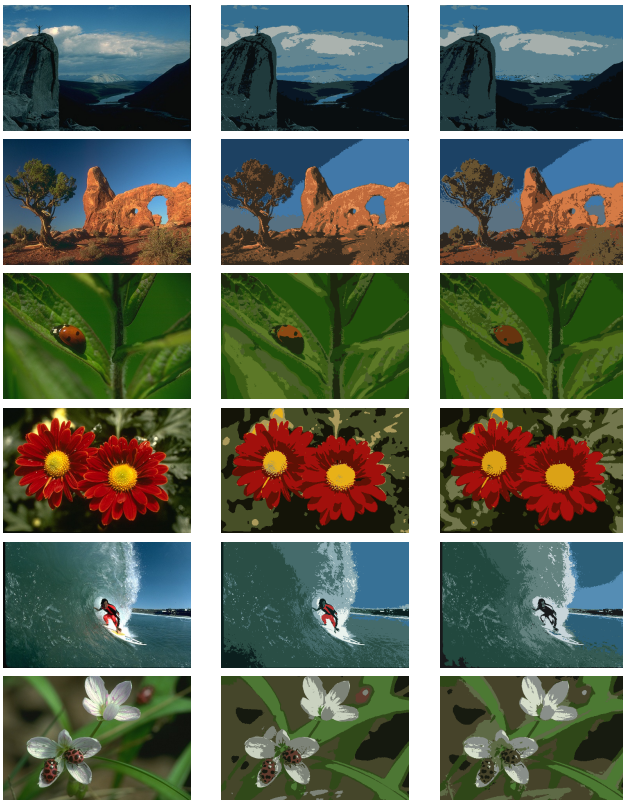


Fig. 4: Segmentation comparison (left, middle, right columns are for the original, CGIS, and Nyström).

TABLE I: Computing time of clustering game approach vs. Nyström method

Image name with resolution and segment# (sample rate = 0.005)	Computing time (second)	
	CGIS	Nyström
Hill (6 clusters)	13.426447	36.266262
Arch (7 clusters)	13.941974	37.739345
Ladybug (7 clusters)	14.647467	37.214642
Sunflower (6 clusters)	14.454268	36.530463
Surf (7 clusters)	14.435891	38.488570
Bugs on flowers (7 clusters)	15.534748	38.701645

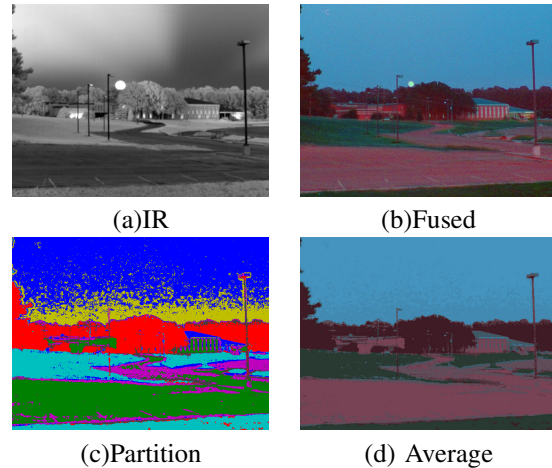


Fig. 5: Fused IR image and the partition results before the noise reduction.

Fig. 7, which also shows the results of the Nyström method for comparison purpose. Two methods obtain similar results except that the order of segments are different. During the simulation, we found that the Nyström method has memory problems for sample rate greater than 0.002. In addition, the clustering game approach is more computational efficient than the Nyström method. For example, it takes 14.454268 seconds for game approach to finish the segmentation of the “sunflower” image while 36.530463 seconds for the Nyström method for the same image.

### C. Summaries

Basically, the performance of the clustering game approach is better than the available Nyström method’s in three aspects. First, clustering game approach is unsupervised clustering and does not need the desired number of segments in advance. The optimal segment number will be determined within the algorithm and is based on the sampled pixel number ( $n$ ). Second, cluster game is fast and computing resource efficient. To run Nyström to partition bigger images, we have to set a low sample rate otherwise it will run of memory in Matlab. Finally, in some cases (see the last two images in Fig. 5), the Nyström method will phase out the color information in average color images due to the wrong partitions.

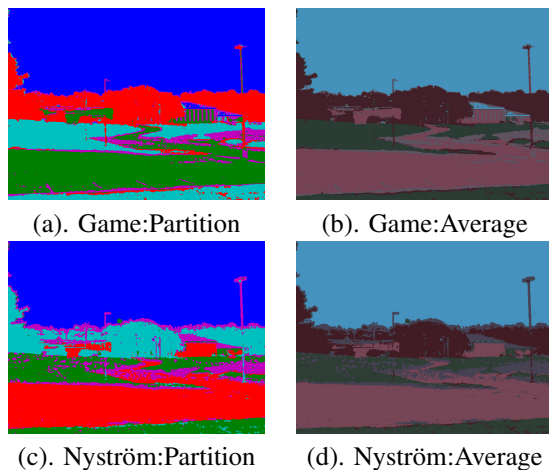


Fig. 6: Comparison of clustering game approach (sample rate = 0.005) and the Nyström method (sample rate = 0.002).

## V. CONCLUSIONS

In this paper, we presented a holistic framework, *CGBISF*, for clustering based image segmentation. The framework integrated color fusion for IR images, linear smoothing filters, and clustering game engine, for which we implemented three main evolutionary dynamics methods (fictitious play, replicator dynamics, and *PureInImDyn*). We tested our approach on the Berkeley color images database as well as fused IR virtual color images. We compared the performance with the Nyström method and the results we obtained showed the superiority of the clustering game based approach in autonomy, speed, and efficiency. Future work would include integrating gradient information to judge the edge boundaries (like non-linear diffusion), trying different color spaces, like Luminance-Alpha-Beta, merging the clustered segments together using region growing and/or morphological processing, assessing the proposed method over different image qualities, using the the method for object identification addressing other metrics such as the Borsotti criteria and Levine-Nazif measure [32], and development of a multi-modal method for image fusion [33]. Using the *CGBISF*, we will also study the effects of image segmentation on image fusion performance assessment [34].

## REFERENCES

- [1] K. Fu and J. Mui, "A Survey on Image Segmentation," *Pattern Recognition*, vol. 13, 1981, pp. 3-16.
- [2] R. Haralick and L. Shapiro, "Image Segmentation Techniques," *Computer Vision, Graphics and Image Processing*, vol. 29, 1985, pp. 100-132.
- [3] N. Pal and S. Pal, "A Review on Image Segmentation Techniques," *Pattern Recognition*, vol. 26, 1993, pp. 1277-1294.
- [4] S. Zucker, "Region Growing: Childhood And Adolescence," *Computer Graphics and Image Processing*, vol. 5, 1976, pp. 382-399.
- [5] M. Amadasun and R.A. King, "Low-Level Segmentation Of Multi-spectral Images Via Agglomerative Clustering Of Uniform Neighborhoods," *Pattern Recognition*, vol. 21, 1988, pp. 261-268.
- [6] I.A. Ismaili and D.F. Gillies, "Color Image Segmentation Using Regression Analysis in RGB Space," *Machine Graphics & Vision*, vol. 3, 1994, pp. 373-384.

- [7] P. Chen and T. Pavlidis, "Image segmentation as an estimation problem," *Computer Graphics and Image Processing*, vol. 12, 1980, pp. 153-172.
- [8] C.L. Huang, T.Y. Cheng, and C.C. Chen, "Color images' segmentation using scale space filter and Markov random field," *Pattern Recognition*, vol. 25, no. 10, pp. 1217-1229, 1992.
- [9] R. Schettini, "A segmentation algorithm for color images," *Pattern Recognition Letters*, vol. 14, pp. 499-506, 1993.
- [10] M. Yachida and S. Tsuji, "Application of colour information to visual perception," *Pattern Recognition*, vol. 3, pp. 307-323, 1971.
- [11] T.L. Huntsberger and M.F. Descalzi, "Colour edge detection," *Pattern Recognition Letters*, vol. 3, pp. 205-209, 1985.
- [12] D.C. Baker, S.S. Hwang, and J.K. Aggarwal, "Detection and segmentation of man-made objects in outdoor scenes: concrete bridges," *Journal of the Optical Society of America A*, vol. 6, no. 6, pp. 938-951, 1989.
- [13] Y.I. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Computer Graphics and Image Processing*, vol. 13, pp. 222-241, 1980.
- [14] L.G. Shapiro and G.C. Stockman, *Computer Vision*, New Jersey, Prentice-Hall, ISBN 0-13-030796-3, 2001, pp 279-325.
- [15] T. Hofmann and J. Buhmann, "Pairwise data clustering by deterministic annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 1, pp. 1-13, 1997.
- [16] C. Fraley and A. Raftery, "How many clusters? which clustering method? - answers via model-based cluster analysis," *Computer Journal*, vol. 41, pp. 578-588, 1998.
- [17] Y. Leung, J. Zhang, and Z. Xu, "Clustering by scale-space filtering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 12, pp. 1396-1410, 2000.
- [18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888-905, 2000.
- [19] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 10, pp. 1053-1074, 2001.
- [20] S.R. Bulò and I.M. Bomze, "Infection and immunization: a new class of evolutionary game dynamics," *Games and Economic Behaviour (Special issue in honor of John F. Nash, Jr.)*, vol. 71, pp. 193-211, 2011.
- [21] A. Chakraborty, "Game-theoretic integration for image segmentation," *IEEE Trans. on PAMI*, vol. 21, no. 1, pp. 12-30, 1999.
- [22] L. Samuelson, "Evolution and game theory," *Journal of Economic Perspectives*, vol. 16, no. 2, pp. 46-66, 2002.
- [23] J.W. Weibull, *Evolutionary game theory*, Cambridge University Press, 1995.
- [24] J. Barthelemy and F. Brucker, "Binary clustering," *Discrete Applied Mathematics*, vol. 156, no. 8, pp. 1237-1250, 2008.
- [25] Y. Zheng, "An Exploration of Color Fusion with Multispectral Images for Night Vision Enhancement," *Image Fusion and Its Applications*, ISBN 978-953-307-173-2, June 2011.
- [26] G.W. Brown, "Iterative Solutions of Games by Fictitious Play," *In Activity Analysis of Production and Allocation*, T.C. Koopmans, Ed., New York: Wiley, 1951.
- [27] D. Fudenberg and D.K. Levine, *The Theory of Learning in Games*, Cambridge: MIT Press, 1998.
- [28] P.D. Taylor and L. Jonker, "Evolutionary stable strategies and game dynamics," *Math. Biosciences*, vol. 40, pp. 145-156, 1978.
- [29] T. Riemersma, *Color Metric*, 1998, Available: <http://www.compuphase.com/cmtric.htm>
- [30] *The Berkeley segmentation dataset and benchmark*, (BSDS500, and extended version of BSDS300), Available: <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>
- [31] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nystrom method," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 2, pp. 214-225, 2004.
- [32] M. Levine and A. Nazif, "Dynamic measurement of computer generated image segmentations," *IEEE Trans. on PAMI*, vol. 7, no. 25, pp. 155164, 1985.
- [33] E. Blasch, X. Li, G. Chen, and W. Li, "Image Quality Assessment for Performance Evaluation of Image fusion," *Fusion08*, 2008.
- [34] Z. Liu, E. Blasch, Z. Xue, J. Zhao, R. Langanier, and W. Wu, "Objective Assessment of Multiresolution Image Fusion Algorithms for Context Enhancement in Night Vision: A Comparative Survey," *IEEE Trans. on PAMI*, vol. 34, no. 1, pp. 94-109, 2012.