Cellular image segmentation using n-agent cooperative game theory

Dimock, Ian, Wan, Justin W.

Ian B. Dimock, Justin W. L. Wan, "Cellular image segmentation using n-agent cooperative game theory," Proc. SPIE 9784, Medical Imaging 2016: Image Processing, 97842T (21 March 2016); doi: 10.1117/12.2216571

SPIE.

Event: SPIE Medical Imaging, 2016, San Diego, California, United States

Cellular image segmentation using n-agent cooperative game theory

Ian B. Dimock[a] and Justin W.L. Wan[b]

[a]Centre for Computational Mathematics in Industry and Commerce, University of Waterloo, Ontario, Canada
[b]David R. Cheriton School of Computer Science, University of Waterloo, Ontario, Canada

ABSTRACT

Image segmentation is an important problem in computer vision and has significant applications in the segmentation of cellular images. Many different imaging techniques exist and produce a variety of image properties which pose difficulties to image segmentation routines. Bright-field images are particularly challenging because of the non-uniform shape of the cells, the low contrast between cells and background, and imaging artifacts such as halos and broken edges. Classical segmentation techniques often produce poor results on these challenging images. Previous attempts at bright-field imaging are often limited in scope to the images that they segment. In this paper, we introduce a new algorithm for automatically segmenting cellular images. The algorithm incorporates two game theoretic models which allow each pixel to act as an independent agent with the goal of selecting their best labelling strategy. In the non-cooperative model, the pixels choose strategies greedily based only on local information. In the cooperative model, the pixels can form coalitions, which select labelling strategies that benefit the entire group. Combining these two models produces a method which allows the pixels to balance both local and global information when selecting their label. With the addition of k-means and active contour techniques for initialization and post-processing purposes, we achieve a robust segmentation routine. The algorithm is applied to several cell image datasets including bright-field images, fluorescent images and simulated images. Experiments show that the algorithm produces good segmentation results across the variety of datasets which differ in cell density, cell shape, contrast, and noise levels.

Keywords: bright-field images, image segmentation, cooperative game theory, cellular images

1. INTRODUCTION

In the biological sciences there is often a need to analyse cellular images. Common tasks include counting the number of cells present in an image or tracking events such as cell splitting. Due to the ever increasing number of images being taken, automatic methods are desired to perform these tasks. The first step in many instances is to produce a segmentation of the image, in which we distinguish between the foreground and background of the images, or rather what is a cell and what is not.

There are multiple imaging techniques used by biologists to capture information about cells. Two general techniques are fluorescent imaging and bright-field imaging. Fluorescent images often have very good contrast, making them easy to segment automatically, however the process of producing them can be harmful to the cells.[1] If we wish to image the cells with minimal interference we can use instead bright-field microscopy. The downside of using bright-field images is that they often have low contrast and imaging artifacts such as halos and broken edges. The artifacts in bright-field introduce challenges to automatic segmentation that most segmentation algorithms fail to overcome.

There has been significant research into the area of cellular image segmentation. Most of this work has been done with fluorescent microscopy, however there has been work done with bright-field images as well. Tscherepanow et al. propose a method using active contour methods for bright-field image segmentation.[2] The method performs well, but is applied to only one type of cell and so questions about its robustness remain.

For further information, please contact Ian Dimock: ibdimock@uwaterloo.ca, or Justin W.L. Wan: justin.wan@uwaterloo.ca.

Medical Imaging 2016: Image Processing, edited by Martin A. Styner, Elsa D. Angelini, Proc. of SPIE Vol. 9784, 97842T · © 2016 SPIE
CCC code: 1605-7422/16/$18 · doi: 10.1117/12.2216571

A different algorithm proposed by Bradbury and Wan uses a spectral k-means method[3] for bright-field image segmentation. This algorithm was also only tested with one series of images. A different approach proposed by Selinummi et al. combines multiple bright-field images of the same cells into a higher contrast projection.[1] The method is successful however requires a specific imaging technique to be used. A common problem in most cases is that the methods are tested on only a single dataset. The methods may not be robust for all the various issues that arise in different image datasets.

This works aims to produce a robust algorithm that is capable of segmenting cellular images including bright-field images and other difficult datasets. We develop a novel algorithm that incorporates both a non-cooperative and cooperative game theoretic model and which performs well on a variety of real world and simulated datasets.

The layout of the paper is as follows. Section 2 will describe the mathematical model underlying the algorithm, the details of the game theoretic segmentation routine as well as several other implementation particulars. Section 3 will present the results of applying the segmentation algorithm to a variety of datasets. Finally we close with some concluding remarks in Section 4.

2. METHODOLOGY

The segmentation problem can be viewed as the process of labelling all the pixels in the image as either foreground or background (0 or 1). While there are many existing methods to perform cellular segmentation, game theory is rarely used for this application, however several examples do exist.[4,5] In this paper we adapt a method presented by Guo, Yu, and Ma.[6] This model has each pixel in the image act as an independent agent or player in an n-agent game. To allow each pixel to make a decision we must provide the pixels with an objective function.

2.1 N-agent Game Theoretic Image Segmentation

In this model we represent the $m \times n$ pixel image as a 2D grid of points. We index the points either by a conventional pair (i, j), with $i \in \{1 \dots m\}, j \in \{1 \dots n\}$ or as a single index $s \in S = \{1 \dots N\}$ where $N = m \cdot n$. We can convert from (i, j) to s using the following formula:

$$s = m \cdot (j - 1) + i. \tag{1}$$

The image data is given as $u = (u_1, \dots, u_N)$, where each u_s takes a value between 0 and 1, representing the intensity of a pixel. The goal of image segmentation is to identify foreground and background components. To achieve this we would like to assign each pixel in the image to either the background or the foreground represented by a label 0 or 1. We represent the segmentation as $w = (w_1, \dots, w_N)$, here each w_s is a binary variable taking either 0 or 1. A segmentation w thus provides each pixel with a label giving us potential foreground and background components.

2.1.1 Game Theoretic Objective Function

There are 2^N possible segmentations of a given image with N pixels. We would like to find the best possible segmentation of our image. We call this optimal segmentation w^*. The segmentation w^* is that segmentation which is best explained by the pixels in our image. We can write this as a conditional probability $P(w|u)$. This probability represents the quality at which a given segmentation w is explained by the image u. Finding the segmentation w^* which maximizes this probability is an optimization problem with objective function $P(w|u)$. By applying Bayes' rule we arrive at the following optimization problem:

$$w^* = \arg\max_w P(w|u) = \arg\max_w P(u|w)P(w). \tag{2}$$

Under the random field model we are using, we have that $P(u|w)$ is independent across all points in the grid, so:

$$P(u|w) = \prod_s P(u_s|w_s). \tag{3}$$

We further assume that the distribution of $P(u_s|w_s)$ is Gaussian. We have two classes, foreground and background, represented by 0 and 1. Each class will be represented by its own Gaussian distribution. We specify

these distributions by means and standard deviations: (μ_0, σ_0) for class 0, and (μ_1, σ_1) for class 1. We will let k represent one of the two possible labels for a pixel, 0 or 1, so $k \in \{0, 1\}$. We are now ready to write down the distribution for $P(u_s|w_s)$ for a pixel s taking class k:

$$P(u_s|w_s) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(u_s - \mu_k)^2}{2\sigma_k^2}\right). \tag{4}$$

To simplify our optimization we take the log of the argument. This preserves optimality since log is a monotone function. We get:

$$w^* = \arg\max_w \sum_s \left[\log\left(P(u_s|w_s)\right)\right] + \log\left(P(w)\right). \tag{5}$$

The 2D grid that we have presented with the Gaussian distribution is a Markov Random Field (MRF). It satisfies the conditions of the Hammersley-Clifford Theorem,[7] which allows us to write $P(w)$ according to a Gibbs distribution:

$$P(w) = \frac{1}{Z} \exp\left(-\sum_{X \in \mathcal{X}} V^X(w)\right). \tag{6}$$

Here, Z is called the normalizing constant, and can be ignored after taking the logarithm. The function V^X is called the clique potential for a clique X. A clique is a set of pixels all of which neighbour each other. We consider only four neighbours for each pixel (up, down, left, and right) so cliques in our grid consist only of pairs of pixels. Here $\mathcal{X}$ represents the set of all cliques. Since we only ever consider cliques of size two, we can redefine the potential function correspondingly:

$$V^X(w) \equiv v(w_s, w_r) = \begin{cases} -\beta & \text{if } w_s = w_r \\ \beta & \text{if } w_s \neq w_r \end{cases}, \text{where } X = \{s, r\}. \tag{7}$$

Here β is a model parameter, which controls the scaling of the clique potential.

We can express our optimization in terms of energy. We wish to find the lowest total energy E^{tot}. We express this by expanding $P(w)$ in our objective function and grouping all terms by index s to get:

$$w^* = \arg\min_w E^{tot} \equiv \arg\min_w \sum_s E(u, w, s), \tag{8}$$

where $E(u, w, s)$ represents the local energy for a given pixel s. It is the combination of terms from the Gaussian distribution and the clique potentials of all cliques to which s belongs. Recall, a pixel can be labelled 0 or 1 in our segmentation w. We define the energy when pixel s is labelled 0 to be $E^0(u, w, s)$. Similarly, when pixel s is labelled 1 the energy is $E^1(u, w, s)$. To minimize the total energy, we define the local energy of each pixel to be the minimum of these two:

$$E(u, w, s) = \min\left[E^0(u, w, s), E^1(u, w, s)\right]. \tag{9}$$

The energy $E^0(u, w, s)$ is computed using (μ_0, σ_0) from the Gaussian distribution for class 0, and is defined as follows:

$$E^0(u, w, s) = \log(\sqrt{2\pi}\sigma_0) + \frac{(u_s - \mu_0)^2}{2\sigma_0^2} + \sum_{r \in N(s)} v(0, w_r), \tag{10}$$

where $N(s)$ represents the neighbourhood of pixel s (all pixels belonging to a clique with s). We can similarly define $E^1(u, w, s)$:

$$E^1(u, w, s) = \log(\sqrt{2\pi}\sigma_1) + \frac{(u_s - \mu_1)^2}{2\sigma_1^2} + \sum_{r \in N(s)} v(1, w_r). \tag{11}$$

Equations (10) and (11) give us a way to compare the two possible labels which we can assign to pixel s. They will be the foundation on which we build the cooperative and non-cooperative components described in the following sections.

2.1.2 Non-Cooperative Strategy

From an initial segmentation w_0, we allow the pixels (agents) to compete in a non-cooperative game where each pixel tries to minimize its local energy, using the labels of its neighbours as information. Agents in a non-cooperative game play greedily, in the sense that they do not seek to form teams or coalitions even if that would be mutually beneficial. The ideal outcome in our game would be to find a Nash equilibrium.[8] A Nash equilibrium is defined such that no agent in the game can improve its objective value by changing strategies. In the segmentation scenario this would represent in a stable segmentation.

There are two difficulties in finding this equilibrium. Firstly, we are not allowing the pixels to pick strategies which are distributions over their actions, and as such the Nash Equilibrium is not guaranteed to exist. Secondly, even if it does exist it may be difficult to find due to the number of agents involved. We turn to a technique called best response dynamics, which is a method by which agents can choose their strategies. It is an iterative procedure where agents update their strategy based on local information. In this case, the pixels will update their label based on the labels of their neighbours. By this we mean that each pixel s will compute E^0 and E^1 based on the previous segmentation, and choose either the strategy 0 or 1 which produced the lowest energy. This process can converge to a Nash equilibrium or sometimes get stuck in a cycle. Our observations are that if a cycle is encountered it is only with a small portion of the total pixels. We can halt the iterations if this cycling occurs and we find in practice that not finding a true Nash equilibrium does not affect the results.

The general procedure for the non-cooperative game is outlined in Algorithm 1. The inputs for the algorithm are the original image, u, some initial segmentation w^0, as well as a limit on the number of iterations, K, which will be a model parameter.

Algorithm 1 Non-Cooperative Segmentation

Input: Image u, initial segmentation w^0
Output: Segmentation $w^{non-coop}$, local energies $E^{non-coop}$

function NON-COOPERATIVE-SEGMENTATION(u, w^0)
  for $i = 1$ to K **do**
    Compute $\mu_0, \sigma_0, \mu_1, \sigma_1$ from w^{i-1} and u
    for Each pixel $s \in S$ **do**
      Compute $E^0 = E^0(u, w^{i-1}, s)$
      Compute $E^1 = E^1(u, w^{i-1}, s)$
      if $E^0 < E^1$ **then**
        Set $w_s^i = 0$
        Set $E_s^{non-coop} = E^0$
      else
        Set $w_s^i = 1$
        Set $E_s^{non-coop} = E^1$
      end if
    end for
    if $w^i = w^{i-1}$ **then**
      return $w^i, E^{non-coop}$
    end if
  end for
  return $w^K, E^{non-coop}$
end function

Each iteration i of the algorithm starts by computing the mean and standard deviation for the two classes 0 and 1. This is done by using the segmentation labels provided by the previous iteration (or the initial segmentation w^0). The algorithm proceeds by computing energies E^0 and E^1 for each pixel s using equations (10) and (11). Finally the algorithm chooses the preferred label w_s^i for each pixel by comparing E^0 and E^1. It also stores the value of the local energy it has chosen in the variable $E^{non-coop}$, which is also indexed by S. These local energies and the final segmentation $w^{non-coop}$ are the outputs of the algorithm.

Segmentation results from this algorithm are generally not very good. In this non-cooperative game the agents make decisions base only on information from their four neighbouring pixels. This tends to produce a lot of noise in the final segmentation. We wish to provide a way for the pixels to incorporate information into their decisions from further away in the image. In the next section we introduce a cooperative component to the game that allows the agents to do this.

2.1.3 Cooperative Strategy

To introduce cooperation into the game that the agents are playing we first introduce the concept of a coalition. Here we define a coalition C to be a collection of pixels $C \subseteq S$, such that all pixels are all spatially connected, meaning all pixels can be reached from one another by travelling amongst shared neighbours. We can split the entire image into coalitions in this way. We introduce the collection of all coalitions, $\mathcal{C} = \{C_1, C_2, \ldots, C_P\}$, with the properties that the entire image is divided amongst the coalitions, $S = C_1 \cup C_2 \cup \ldots \cup C_P$, and that no two coalitions share any pixels, $C_i \cap C_j = \emptyset, \forall i \neq j$.

We also introduce an extension to the local energy functions from (10) and (11) to define energies for an entire coalition:

$$E^0(u, w, C) \equiv \sum_{s \in C} E^0(u, w^{C,0}, s), \tag{12}$$

$$E^1(u, w, C) \equiv \sum_{s \in C} E^1(u, w^{C,1}, s). \tag{13}$$

Here we use special segmentations $w^{C,0}$ and $w^{C,1}$ as input to the local energy functions; $w^{C,0}$ is a modification of the segmentation w in which every pixel $s \in C$ is re-labelled 0, and similarly $w^{C,1}$ is a modification of the segmentation w in which every pixel $s \in C$ is re-labelled 1.

With a set of coalitions, $\mathcal{C}$, we proceed by allowing each coalition to decide as a group which strategy all pixels belonging to that coalition will take. This procedure is detailed in Algorithm 2. Inputs to the algorithm are the image u, an initial segmentation w^0, and a set of coalitions $\mathcal{C}$.

Algorithm 2 Cooperative Segmentation

Input: Image u, initial segmentation w^0, coalitions $\mathcal{C}$
Output: Segmentation w^{coop}, local energies E^{coop}

function COOPERATIVE-SEGMENTATION($u, w^0, \mathcal{C}$)
  Compute $\mu_0, \sigma_0, \mu_1, \sigma_1$ from w^0 and u
  for $C \in \mathcal{C}$ **do**
    Let $w^{C,0} = w^{C,1} = w$
    Set $w_s^{C,0} = 0, \forall s \in C$
    Set $w_s^{C,1} = 1, \forall s \in C$
    Compute $E^0 = \sum_{s \in C} E^0(u, w^{C,0}, s)$
    Compute $E^1 = \sum_{s \in C} E^1(u, w^{C,1}, s)$
    if $E^0 < E^1$ **then**
      $\forall s \in C$, set $w_s^{coop} = 0$
      $\forall s \in C$, set $E_s^{coop} = E^0(u, w^{C,0}, s)$
    else
      $\forall s \in C$, set $w_s^{coop} = 1$
      $\forall s \in C$, set $E_s^{coop} = E^1(u, w^{C,1}, s)$
    end if
  end for
  return w^{coop}, E^{coop}
end function

The algorithm begins by computing the means and standard deviations for the classes 0 and 1 specified by the initial segmentation w^0. It continues by computing coalition energies for each $C \in \mathcal{C}$. It computes the energy E^0 given that the entire coalition C takes label 0 using equation (12) and the energy E^1 given that the entire coalition C takes label 1 using equation (13). By comparing these two coalition wide energies, the coalition makes a decision as to which label all its members should take. It stores the relevant local energies and also the labels that are chosen by each coalition. We are left at the end with the cooperative segmentation w^{coop} and the collection of local energies, E^{coop}, which each pixel computed when choosing its label.

This cooperative game tends to produce large coalitions, and has trouble segmenting fine details in the image because there is no way to reduce coalition size. To achieve a segmentation with strengths from both the the non-cooperative routine as well as this cooperative one, we incorporate the two into a single segmentation algorithm, described in the next section.

2.1.4 Strategy Integration

In this section we describe the procedure by which the non-cooperative and cooperative segmentation routines are integrated into a single procedure. We have segmentations produced by those routines: $w^{non-coop}$ and w^{coop} as well as the collection of local energies computed as each pixel chose its label in Algorithms 1 and 2: $E^{non-coop}$ and E^{coop}.

Normally two segmentations would be hard to compare and combine, however we built each routine upon the same local energy functions (10) and (11). To produce a combined segmentation, w^{comb}, we simply compare the cooperative and non-cooperative energies from $E^{non-coop}$ and E^{coop}. The details of this are presented in Algorithm 3.

Algorithm 3 Combine Non-Cooperative and Cooperative Segmentations

Input: Segmentations $w^{non-coop}$ and w^{coop}, energies $E^{non-coop}$ and E^{coop}
Output: Segmentation w^{comb}

function COMBINED-SEGMENTATION($w^{non-coop}, w^{coop}, E^{non-coop}, E^{coop}$)
  for $s \in S$ **do**
    if $E_s^{non-coop} < E_s^{coop}$ **then**
      Set $w_s^{comb} = w_s^{non-coop}$
    else
      Set $w_s^{comb} = w_s^{coop}$
    end if
  end for
  return w^{comb}
end function

As mentioned, the inputs to the algorithm are the two segmentations and the two collections of local energies: $w^{non-coop}$, w^{coop}, $E^{non-coop}$, and E^{coop}. For each pixel s, the algorithm compares the local energies $E_s^{non-coop}$ and E_s^{coop}. Whichever has the smaller energy, provides the label for the combined segmentation w^{comb}. In this way each pixel in the combined segmentation is provided with a label from wither $w^{non-coop}$ or w^{coop}.

Similar to the iterations in the non-cooperative component, we can iterate this combination procedure until a stable segmentation is reached. At each step we must update the coalitions with information from the new combined segmentation. We use our definition that coalitions are spatially connected regions of pixels that have chosen the same strategy. We can thus construct the coalitions by performing a breadth first search on our segmentation, grouping together pixels with the same label.

With a new set of coalitions we repeat the process of computing both the non-cooperative and cooperative segmentations, as well as performing the combination procedure. These steps are iterated a certain number of times specified by parameter L. In general, the algorithm is found to converge in ~ 10 iterations.

2.2 Additional Algorithm Details

There are several extra components involved in the complete algorithm. Firstly, we need an initial set of coalitions. These are created simply by dividing the image up into square blocks of side length B, a model parameter.

We also require an initial segmentation as input to the first iteration. Any segmentation algorithm would do for this purpose. We settled on a two class k-means algorithm, finding that it was both fast and produced a sufficiently good initial segmentation for all the datasets tested. We use the pixel intensities as input to the k-means algorithm.

The final component was a post-processing step. The game theoretic algorithm tends to have some noisy edges when the input image is correspondingly noisy. To help clean up the noisy edges as well as fill in any small holes amongst the cells, we perform a Chan-Vese active contours algorithm[9] using our output segmentation from the game theoretic algorithm as input. Figure 1 shows the outcome of performing our active contours smoothing step, with a noisy input image.

With all these components in place, the resulting algorithm is completely automated and found to be quite robust, performing well on a variety of datasets. The procedure is summarized in Algorithm 4.

Algorithm 4 Game Theoretic Segmentation

Input: Image u
Output: Segmentation w^{gt}

function GT-SEGMENTATION(u)
  Compute w^0 by two class k-means
  Initialize $\mathcal{C}$ as blocks of size B
  $w^{gt} = w^0$
  for $i = 1$ to L **do**
    $w^{non-coop}, E^{non-coop} =$ NON-COOPERATIVE-SEGMENTATION(u, w^{gt})
    $w^{coop}, E^{coop} =$ COOPERATIVE-SEGMENTATION$(u, w^{gt}, \mathcal{C})$
    $w^{gt} =$ COMBINED-SEGMENTATION$(w^{non-coop}, w^{coop}, E^{non-coop}, E^{coop})$
    Recompute $\mathcal{C}$ by breadth first search
  end for
  Use Chan-Vese active contours to smooth w^{gt}
  return w^{gt}
end function

Figure 1: Left: Input image from ISBI 2013 Cell tracking challenge dataset C2DL-MSC 01. Middle: Segmentation result before applying Chan-Vese smoothing procedure. Right: Segmentation after applying Chan-Vese smoothing procedure.

3. NUMERICAL RESULTS

The algorithm as presented was implemented in MATLAB. The largest image sizes segmented were 1024×1024 pixels. We must select several model parameters for our algorithm. Firstly, β, which controls the relative strength in the terms of our energy function. We have the block size of our initial coalitions B. We also have K, the maximum number of iterations performed in the non-cooperative component, and L, the total number of iterations we perform in the combined algorithm. In all cases we select $K = 30$, and $L = 10$. B ranges from $8 - 20$ depending on the input image size.

The Chan-Vese active contour algorithm also requires several parameters to be selected. The only parameter we change between datasets is the parameter $\mu = \mu_{CV}$ from the Chan-Vese formulation, otherwise we set the total number of iterations to $T = 10$ and timestep to $\Delta t = 0.3$. All other parameter values are adapted from the literature: $\nu = 0, \lambda_1 = \lambda_2 = 1, h = 1$.

In summary, only β, μ_{CV}, and B must be chosen for each run of the algorithm, and we found that the same parameters can be used across all images in a given dataset with good results. Even between datasets, the values of these parameters changes minimally.

We now present the results of our algorithm on a set of bright-field C2C12 cell images courtesy of the Genomic Laboratory at McGill University. The size of the images is 512×512 pixels. These images contain several challenges to segmentation including bright halos and low contrast cells. Cells undergoing splitting pose an additional challenge to segmentation. The results of the segmentation algorithm is presented in Figure 2. As can be seen, the majority of the cells have been successfully segmented. The parameters used for these images were: $\beta = 1.5, \mu_{CV} = 6, B = 8$.

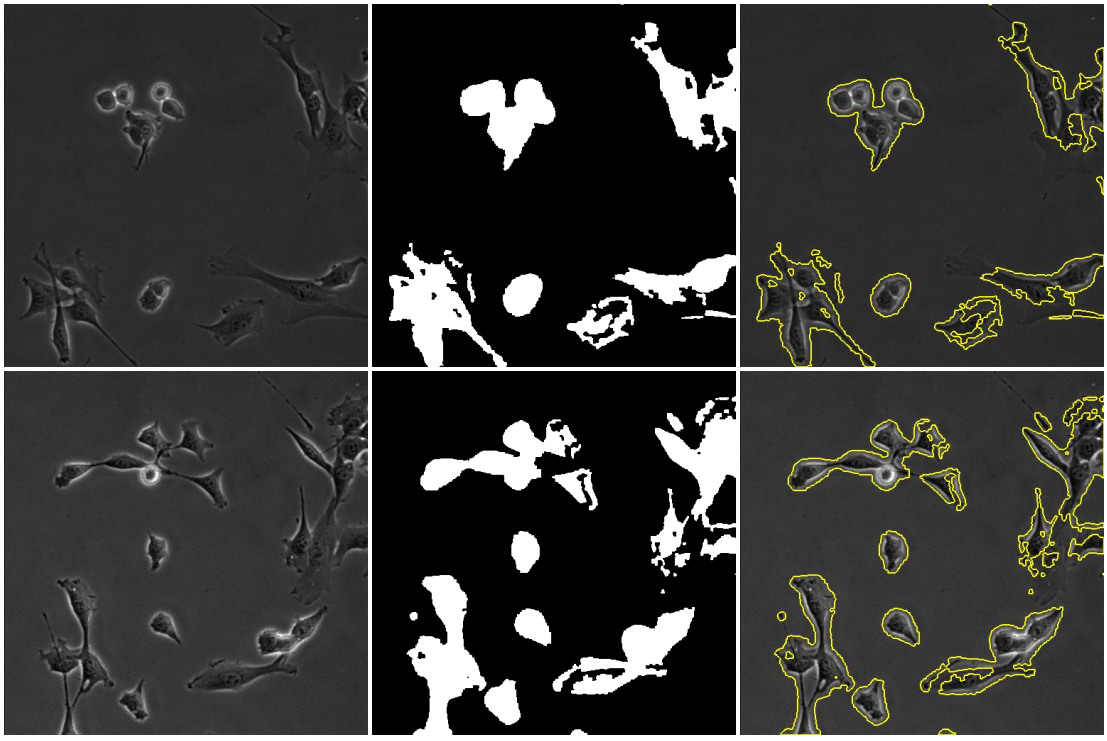

Figure 2: Segmentations for bright-field C2C12 cell images. First column: input images. Second column: game theoretic segmentaion results. Third column: visualization of the segmentation.

To further test the robustness of the algorithm we tested it on datasets from the International Symposium on Biomedical Imaging (ISBI) 2013 Cell Tracking Challenge.[10] The datasets here cover a variety of cell imaging techniques including simulated images. To compare segmentation results across submissions to the challenge the

Jaccard Similarity Index was used. Given the set of pixels representing the ground truth for a cell, R, and the set of pixels representing the segmentation of a cell, S, the Jaccard score is computed as: $J(R,S) = \frac{|R \cap S|}{|R \cup S|}$. This score lies between 0 and 1 for a given cell, and the score across all cells in an image is averaged to provide a score for that image. The Jaccard scores listed for the game theoretic algorithm we have presented are those with smoothing included. We computed the Jaccard scores on several examples both before and after applying the final active contour smoothing operation. There was minimal improvement or even slightly lower scores after smoothing.

We compared our segmentation algorithm against the six entrants[11–16] of the challenge across the twelve 2D datasets provided. The results are summarized in Table 1. It can be seen that the game theoretic algorithm we have presented is competitive with the stronger entries in the competition. The KTH-SE submission was arguably the strongest entered, and compared head to head, our game theoretic approach out performs it on 9 of 12 datasets. Another challenge submission, LEID-NL, performs very well on the simulated N2DH-SIM datasets. Comparing head to head again, our game theoretic algorithm outperforms it on 5 of 6 of these simulated images. The game theoretic algorithm performs particularly well on the two C2DL-MSC series. As can be seen in the table, these were the hardest images to segment amongst all the challenge datasets. This is because of their high noise level and irregularly shaped cells.

Table 1: Segmentation scores from ISBI 2013 Cell Tracking Challenge as well as game theoretic algorithm for datasets provided by the Cell Tracking Challenge. Bolded entries are the highest score for that dataset. Adapted from http://www.codesolorzano.com/celltrackingchallenge/Cell_Tracking_Challenge/Results_First_CTC.html.

Dataset	C2DL-MSC		N2DH-GOWT1		N2DL-HeLa		N2DH-SIM					
Series	01	02	01	02	01	02	01	02	03	04	05	06
COM-US	0.06	0.15	0.69	0.39	0.33	0.3	0.75	0.77	0.65	0.64	0.64	0.75
HEID-GE	0.34	0.43	0.84	0.89	0.83	0.8	0.91	0.9	0.82	0.83	0.81	0.89
KTH-SE	0.49	0.66	**0.86**	0.91	**0.88**	**0.89**	0.91	0.89	**0.84**	0.83	0.8	0.83
LEID-NL	N/A	N/A	0.84	0.88	0.74	N/A	**0.93**	0.91	0.83	0.84	0.84	**0.91**
PRAG-CZ	0.15	0.21	**0.86**	0.92	0.78	0.76	0.8	0.8	0.75	0.78	0.76	0.79
UPM-ES	0.41	0.19	0.55	0.63	0.63	0.65	0.91	0.89	0.81	0.82	0.77	0.86
GameTheoretic	**0.55**	**0.75**	**0.86**	**0.93**	0.69	0.73	0.9	**0.92**	**0.84**	**0.86**	**0.94**	0.89

Figure 3 shows select segmentations by our game theoretic algorithm on datasets from the ISBI challenge. The datasets from top to bottom: C2DL-MSC series 02, 1200 $\times$ 782 pixels using parameters $\beta = 2.5, \mu_{CV} = 2.5, B = 16$; N2DH-GOWT1 series 01, 1024 $\times$ 1024 pixels using parameters $\beta = 2.6, \mu_{CV} = 10, B = 20$; N2DH-SIM series 06, 480 $\times$270 pixels using parameters $\beta = 1.2, \mu_{CV} = 10, B = 10$; N2DL-HeLa series 01, 1100 $\times$ 700 pixels using parameters $\beta = 2.2, \mu_{CV} = 10, B = 14$; and C2DL-MSC series 01, 992 $\times$ 832 pixels using parameters $\beta = 2.5, \mu_{CV} = 10, B = 16$.

4. CONCLUSION

In this work we have developed a segmentation algorithm using game theoretic models not often seen in image segmentation. The algorithm combines two paradigms of game theory, cooperative and non-cooperative games, into a single iterative procedure. With the addition of a k-means initialization stage and active contour smoothing, the result is a robust segmentation algorithm that requires very little parameter tuning and performs well on a variety of datasets. In particular, the algorithm performs well on challenging bright-field images and other difficult to segment cellular images.

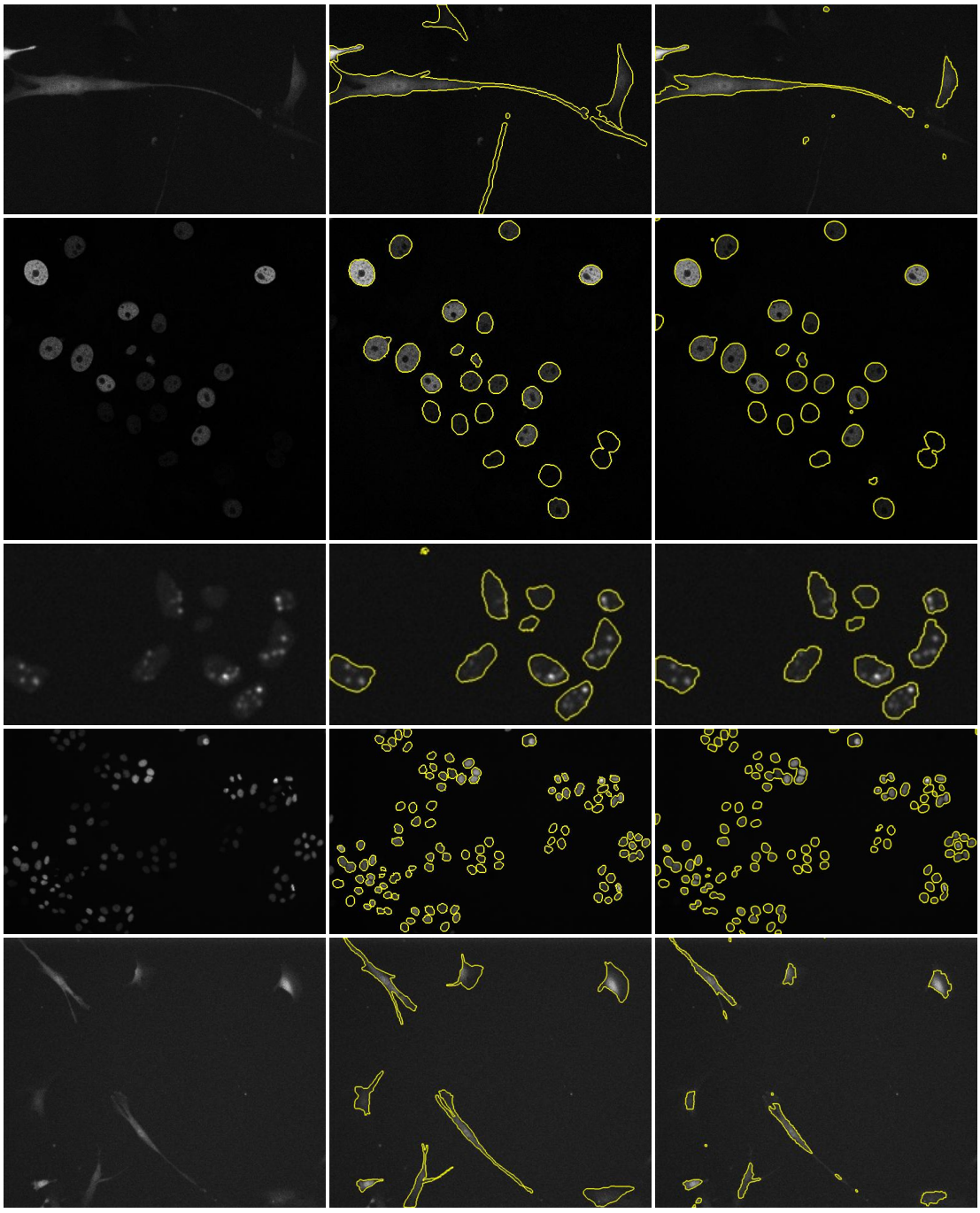

Figure 3: Segmentations for ISBI Cell Tracking Challenge datasets. First column: input images. Second column: ground truths. Third column: game theoretic segmentations.

ACKNOWLEDGMENTS

The authors would like to thank Robert Sladek and Haig Djambazian from the Department of Medicine and Human Genetics at McGill University for the provision of the C2C12 microscopy images. This work was supported by the National Sciences and Engineering Research Council of Canada.

REFERENCES

[1] Selinummi, J., Ruusuvuori, P., Podolsky, I., Ozinsky, A., Gold, E., Yli-Harja, O., Aderem, A., and Shmulevich, I., "Bright field microscopy as an alternative to whole cell fluorescence in automated analysis of macrophage images," *PLOS ONE* **4**(10) (2009).

[2] Tscherepanow, M., Zoellner, F., Hillebrand, A., and Kummert, F., "Automatic segmentation of unstained living cells in bright-field microscope images," in [*Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry, Proceedings*], *Lecture Notes in Artificial Intelligence* **5108**, 158–172 (2008). 3rd International Conference on Mass Data Analysis of Signal and Images in Medicine, Biotechnology, Chemistry and Food Industry, Leipzig, Germany, July 14, 2008.

[3] Bradbury, L. and Wan, J. W. L., "A spectral k-means approach to bright-field cell image segmentation," in [*2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*], *IEEE Engineering in Medicine and Biology Society Conference Proceedings*, 4748–4751 (2010). 32nd Annual International Conference of the IEEE Engineering-in-Medicine-and-Biology-Society (EMBC 10), Buenos Aires, Argentina, Aug 30-Sep 04, 2010.

[4] Chakraborty, A. and Duncan, J. S., "Game-theoretic integration for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(1), 12–30 (1999).

[5] Yu, S. and Berthod, M., "A game strategy approach for image labeling," *Computer Vision and Image Understanding* **61**(1), 32 – 37 (1995).

[6] Guo, G., Yu, S., and Ma, S., "An image labeling algorithm based on cooperative game theory," *Signal Processing Proceedings, 1998* **2**, 978–981 (1998).

[7] Besag, J., "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society. Series B (Methodological)* **36**(2), 192–236 (1974).

[8] Nash, J. F., "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences of the United States of America* **36**(1), 48–49 (1950).

[9] Chan, T. F. and Vese, L. A., "Active contours without edges," *IEEE Transactions on Image Processing* **10**(2), 266–277 (2001).

[10] Maska, M., Ulman, V., Svoboda, D., Matula, P., Matula, P., Ederra, C., Urbiola, A., Espana, T., Venkatesan, S., Balak, D. M. W., Karas, P., Bolckova, T., Streitova, M., Carthel, C., Coraluppi, S., Harder, N., Rohr, K., Magnusson, K. E. G., Jalden, J., Blau, H. M., Dzyubachyk, O., Krizek, P., Hagen, G. M., Pastor-Escuredo, D., Jimenez-Carretero, D., Ledesma-Carbayo, M. J., Munoz-Barrutia, A., Meijering, E., Kozubek, M., and Ortiz-de Solorzano, C., "A benchmark for comparison of cell tracking algorithms," *Bioinformatics* **30**(11), 1609–1617 (2014).

[11] Coraluppi, S. and Carthel, C., "Modified scoring in multiple-hypothesis tracking," *Journal of Advances in Information Fusion* **7**(2), 153–164 (2012).

[12] Harder, N., Mora-Bermudez, F., Godinez, W. J., Wuensche, A., Eils, R., Ellenberg, J., and Rohr, K., "Automatic analysis of dividing cells in live cell movies to detect mitotic delays and correlate phenotypes in time," *Genome Research* **19**(11), 2113–2124 (2009).

[13] Magnusson, K. E. G. and Jalden, J., "A batch algorithm using iterative application of the Viterbi algorithm to track cells and construct cell lineages," in [*Proceedings of the 9TH IEEE International Symposium on Biomedical Imaging*], 382–385 (2012).

[14] Dzyubachyk, O., van Cappellen, W. A., Essers, J., Niessen, W. J., and Meijering, E., "Advanced level-set-based vell tracking in time-lapse fluorescence microscopy," *IEEE Transactions on Medical Imaging* **29**(3), 852–867 (2010).

[15] Ovesny, M., Krizek, P., Borkovec, J., Svindrych, Z. K., and Hagen, G. M., "ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging," *Bioinformatics* **30**(16), 2389–2390 (2014).

[16] Pastor-Escuredo, D., Luengo-Oroz, M. A., Duloquin, L., Lombardot, B., Ledesma-Carbayo, M., Bourgine, P., Peyrieras, N., and Santos, A., "Spatio-temporal filtering with morphological operators for robust cell migration estimation in "in-vivo" images," in [*2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*], 1312–1315 (2012). IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI 2012), Barcelona, Spain, MAY 2012.