

Using the \mathcal{P}^n Potts model with learning methods to segment live cell images

Christopher Russell

Dept. Computing, Oxford Brookes University
Oxford OX33 1HX, UK

chris.russell@brookes.ac.uk

Dimitris Metaxas

Rutgers, The State University of New Jersey
Piscataway NJ 08854-8019, USA

dnm@cs.rutgers.edu

Christophe Restif

Dept. Computing, Oxford Brookes University
Oxford OX33 1HX, UK

christophe.restif@centraliens.net

Philip Torr

Dept. Computing, Oxford Brookes University
Oxford OX33 1HX, UK

<http://cms.brookes.ac.uk/staff/PhilipTorr>

<http://cms.brookes.ac.uk/research/visiongroup>

Abstract

We present a segmentation method for live cell images, using graph cuts and learning methods. The images used here are particularly challenging because of the shared grey-level distributions of cells and background, which only differ by their textures, and the local imprecision around cell borders. We use the \mathcal{P}^n Potts model recently presented by Kohli et al. [9]: functions on higher-order cliques of pixels are included into the traditional Potts model, allowing us to account for local texture features, and to find the optimal solution efficiently. We use learning methods to define the potential functions used in the \mathcal{P}^n Potts model. We present the model and the learning methods we used, and compare our segmentation results with similar work in cytometry. While our method performs similarly, it requires little manual tuning and thus is straightforward to adapt to other images.

1. Introduction

Recent advances in Computer Vision have allowed complex energy minimization algorithms to efficiently use graphical models such as Markov Random Fields (MRF) and Conditional Random Fields (CRF) for image segmentation. Such Random Fields usually only include information from single pixels (as *unary potentials*) and pairs of pixels (*pairwise potentials*), and have been used in medical imaging [5]. Such potentials are not suitable for representing texture features though. An approach to include texture information in the graphical model used relies on higher-order cliques (*i.e.* groups of more than two pixels): more complex potentials can be defined this way, at the cost of increasing

the graph complexity. An algorithm to solve such complex graphs efficiently has recently been proposed by Kohli *et al.* [9], for a certain class of higher-order clique potentials, called the \mathcal{P}^n Potts model. In this article we present our application of this framework for the segmentation of live cells images. We use learning methods on texture classification to define the higher-order clique potentials.

Live cells imaging is used in many medical applications, such as gene expression studies or drug tests. By taking snapshots of a sample of living cells using light microscopy, a population of cells can be imaged in controlled conditions over several generations. An application is the estimation of cell population growth and cell division rate. This can be achieved by segmenting all cells in images, and measuring the change of segmented areas over time [7]. With this method, cells merged in clusters do not have to be segmented individually, which is known to be a difficult problem [11, 12, 14, 15], as only the area of the population of cells matters. Significant work in cytometry focuses on the segmentation or identification of individual cells, and the tracking of live cells [14, 15], which are different problems. Other means of image acquisition, such as fluorescent or colour microscopy, result in different segmentation constraints and are not applicable in our context.

An example of an image of live cells is shown in Fig. 1. It illustrates the variations of geometric and texture features among cells. In particular, while some cells are small, bright and round (when in interphase), most are elongated and concave, with sharp corners, and contain all levels of pixel intensities; the background is generally smooth and uniform, but contains small-size dark debris and shows cross-image intensity variations. Such significant variations are not accounted for by many of the other segmentation methods used in cytometry, based on low-level vision, watershed

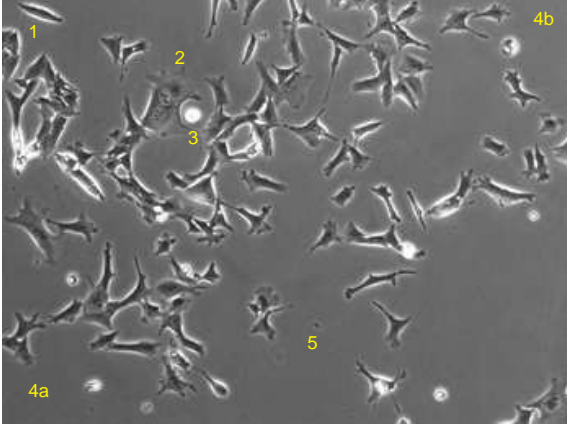


Figure 1. Image of live cells to be segmented. Characteristics that make segmentation difficult are indicated by numbers: (1) pixel values in cells span over the whole range of intensities, (2) cells can be locally hardly distinguishable from background, (3) cells in interphase appear as saturated bright small circles, (4) the background is medium grey, affected by cross-image variations (compare 4a and 4b) and by (5) small dark debris noise.

or active contours [1, 3, 11, 12, 15]. They are also challenging to encode in simple MRF with unary and pairwise potentials, as features based on single and pairs of pixels hardly discriminate between cells and background. The closest work to ours is by He *et al.* [7], where a learning method is used to locate the cells’ boundaries, which are then refined with multi-step refinement and smoothing. The method we present achieves similar results, but with a single-step segmentation method, theoretically well-grounded, requiring fewer parameters, and more straightforward to adapt to other images. Another issue, common in medical imaging, is the imprecision of cell borders, due either to fading intensities or similarity with background, which makes the definition of a ground truth for segmentation intrinsically ambiguous.

We propose an application of the higher-order-clique based \mathcal{P}^n model by Kohli *et al.* [9]. We use a Bayesian classifier to define the pairwise potentials, and train feed-forward neural networks [4] for the clique potentials, using texture features from [8, 13]. In Section 2 we present how an image can be segmented by minimizing unary and pairwise potentials on a graph, and how the clique potentials are added to the model. We detail in Section 3 the learning process used to define the potentials, and present quantitative results on segmented images in Section 4.

2. Segmentation by inference using a \mathcal{P}^n Potts model

2.1. Binary segmentation as a graph cut

Our goal is to segment an image I into two categories, foreground (cells) and background, which can be seen as a

binary labelling of the image. A common approach is to form a MRF or CRF from the image. A node is created to represent each pixel, and nodes of neighbours pixels p and $q \in N_p$ are connected. Each pixel node p is assigned an energy depending on its labelling S_p , and the link between p and q is assigned an energy depending on the labelling of both p and q . We note $S_p = b$ for labelling p as background and $S_p = f$ for foreground. Let S be a solution to the labelling problem, its total cost or energy is made of the terms:

$$E(S|I) = \sum_{p \in I} \underbrace{E_1(S_p|I)}_{\text{unary terms}} + \sum_{\substack{p \in I \\ q \in N_p}} \underbrace{E_2(S_p, S_q|I)}_{\text{pairwise terms}} \quad (1)$$

where E_1, E_2 represent the unary and pairwise energy functions. By convention, the energies are designed to decrease with the likelihood of the labelling, so that a minimum $E(S|I)$ corresponds to an optimum labelling solution.

While that min-sum problem is NP-hard in its general form, there exists sub-classes of problems which can be solved in polynomial time. One of these subclasses consists of those problems with sub-modular or regular energies [10], for which

$$E_2((S_p, S_q) = (b, b)) + E_2((S_p, S_q) = (f, f)) \leq E_2((S_p, S_q) = (f, b)) + E_2((S_p, S_q) = (b, f)) \quad (2)$$

Using an MRF or CRF structure on the image with non-negative weights, two specialist nodes s (source) and t (sink) are added to the graph, with directed connections to every pixel node. By removing a subset of vertices so that s and t are disconnected, the resulting graph can be interpreted as a binary labelling of the image: all pixels connected to s are foreground, and to t , background. Finding the graph cut which minimizes the values of the cut edges can be solved with the $s-t$ min cut/max flow algorithm proposed in [6], generating a solution corresponding exactly to the original min-sum problem in Eq. 1. The details of this transformation from submodular energy to a graph with positive edge weights is deferred till appendix A.

2.2. The \mathcal{P}^n Potts model

The minimization of the energy $E(S|I)$ in Eq. 1 gives the optimal solution to any problem, providing it can be expressed wholly in terms of unary and pairwise costs. This is a restrictive constraint however. More informative cost functions can be defined in terms of local texture features and associated pixel cliques. While increasing the number of neighbours for each pixel would increase the radius of pixel influence on each other, attempting to express texture-based cost using only pairwise relations is still a highly restrictive way to encode texture. A more powerful solution is to use cliques of more than two pixels. The recent work of Kohli *et al.* [9] presented a way to include such higher-order

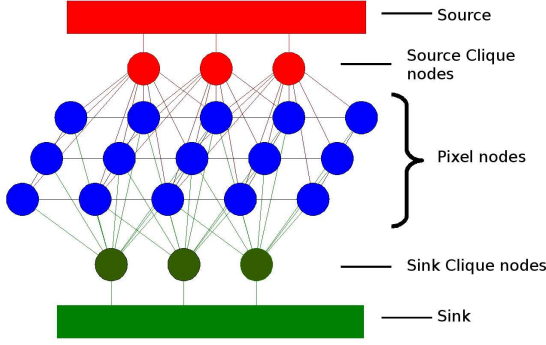


Figure 2. \mathcal{P}^n clique formation. Pixel nodes, in blue, are connected to their neighbours, to many clique nodes, and to the source node (red rectangle) and sink node (green rectangle). These latter links are not shown for clarity. Clique nodes are connected to pixel nodes and to either the source or the sink nodes.

cliques in the $s-t$ min-cut framework presented above, and still solve it efficiently. We follow their method, creating pairs of nodes to represent each higher-order cliques of pixels, as detailed below.

Given a clique C , let $E_C(S_C)$ be the cost for labelling every pixel within C as foreground ($S_C = f$), or background ($S_C = b$). We assign a cost $E_C(S_C \notin \{f, b\})$ to an inconstant labelling where the patch contains a mixture of foreground and background pixels. The total energy of a solution S is now:

$$E(S|I) = \sum_{p \in I} E_1(S_p|I) + \sum_{\substack{p \in I \\ q \in N_p}} E_2(S_p, S_q|I) + \sum_{C \subset I} \underbrace{E_C(S_C|I)}_{\text{clique terms}} \quad (3)$$

The new graph structure is illustrated on Fig. 2. The cliques are typically made of $n \times n$ neighbouring pixels, and overlap (each pixel belonging to many cliques). Two nodes are added to the graph for each clique, a source-clique node and a sink-clique one. The source-clique nodes form an extra layer in the graph, in between the source node and the pixel nodes (which are still also connected to the source). They are connected to the source, with a weight set to $E_C(S_C = f)$, and only to the pixels of their clique. A layer of sink-clique nodes is built similarly. The graph-cut algorithm works in the same way as detailed before with unary and pairwise terms. A cut is performed on the graph to separate the sink from the node, while minimizing the energy in Eq. 3. Two examples of cuts are shown in Fig. 3, one where all pixels from a clique are labelled as foreground, and one where pixels of a clique have different labels. As an indication of computation time, the segmentation of a 700×700

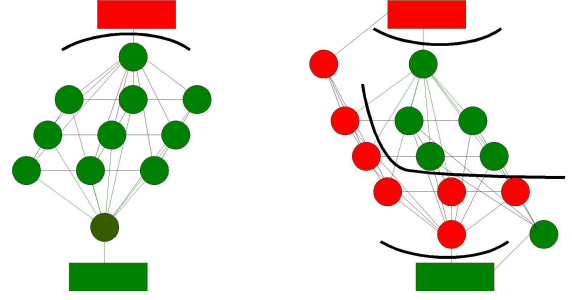


Figure 3. Examples of a source cut and a mixed cut. Left: after the cut, the clique of pixel nodes in the middle layer is entirely connected to the sink, which amounts to labelling all pixels as background. Right: the clique is partially connected to the sink and the source, via some of the clique nodes in the intermediate layers, which corresponds to a mixed labelling of the clique in the middle layer. Note there is still no connected path between source and sink.

pixel image with 5×5 cliques takes less than 20 seconds on a 1.6 GHz laptop with 1Gb of memory.

3. Learning process

In the \mathcal{P}^n Potts model described above, the unary, pairwise and clique potentials have to be generated according to the characteristics of the images to segment. We use learning methods to define them.

3.1. Unary and pairwise potentials

Unary potentials influence the energy function depending on the intensity of individual pixels. In our context though, absolute pixel values are not discriminative enough. As shown in Fig. 4, different images have very different number of cells, and as a result their histograms have little in common. Unary potentials based on the single pixel intensities from some images do not generalize well to other images on our data set. Thus in our model we do not use the term $E_1(p)$ in Eq. 3.

Pairwise potentials correspond to the likelihood of whether two connected pixels share (or not) the same label, based on their relative pixel intensities. We construct a simple Bayesian classifier to generate the pairwise potentials. In an image with a manually segmented ground truth, all pairs of connected pixels are assigned to the corresponding category, one of (f, f) , (f, b) , (b, f) , (b, b) , as illustrated in Fig 3.1. For each pair of pixels (p, q) of intensities $I(p), I(q)$, we use the feature:

$$v(p, q) = \frac{I(p) - I(q)}{I(p) + I(q)} \quad (4)$$

which is a signed version of the normalized variance for two variables x and y with mean μ : $\sqrt{\frac{(x-\mu)^2 + (y-\mu)^2}{\mu^2}}$.

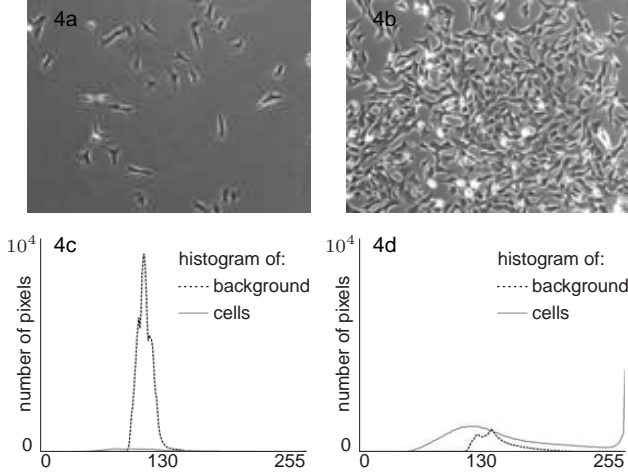


Figure 4. Two images of live cells and their histograms. 4a: early stage of cell culture; 4b: late stage of the same culture; 4c and 4d: histograms of the foreground and the cells, from manual segmentations of 4a and 4b respectively. The histograms of the two regions change significantly during the culture, causing unary potentials to be unreliable over the series of images.

Intensity ratios make this feature invariant to multiplicative variations of intensities, which happens in microscopy imaging depending on slight variations in time of exposure. With our training set, the feature responses are divided into a small number of bins (we use 20). Each bin contains a number of occurrences of the four categories of pairs (f, f) , (f, b) , (b, f) , (b, b) , noted respectively $n_{ff}, n_{fb}, n_{bf}, n_{bb}$, with a total of n occurrences. Then, given any new pair of pixel (p, q) , we compute $v(p, q)$ from Eq. 4, find the corresponding bin, and define:

$$\mathcal{P}((S_p, S_q) = (x, y)) = \frac{n_{xy}}{n} \text{ for } x, y \in \{f, b\} \quad (5)$$

as the probability of pair (p, q) to be in the category $(x, y) \in \{f, b\}^2$. Finally, we define the pairwise terms E_2 in Eq. 3 as:

$$E_2((S_p, S_q) = (x, y)) = -\log \mathcal{P}((S_p, S_q) = (x, y)) \quad (6)$$

While the expression of E_2 may need to be truncated to be sub-modular, in our context there was a sufficiently low proportion of boundary cases, that the cost function was sub-modular.

3.2. Clique potentials

The clique potential indicate how likely a group of pixels (clique) is to be labelled entirely as background, entirely as foreground, or as a mixture of both. An important issue when defining a clique potential is the size of the clique. We use an ROC curve analysis method to select it, as detailed later in this section. Another critical issue is the selection of features which encode the textural information in

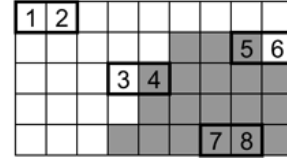


Figure 5. Training values to learn pairwise potentials from ground truth. Background pixels (in white) and cells (in grey) have been manually segmented. All pairs of connected pixels are classified in four categories, depending on the labels of both pixels. In this picture only one of the four directions of pixel connection is shown.

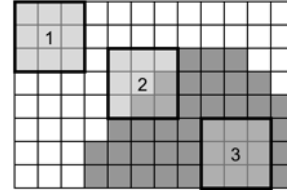


Figure 6. Examples of patches used for learning the clique potential. Patch 1 contains only background, patch 2 contains $\frac{1}{3}$ of foreground, and patch 3 contains only foreground.

the background and foreground. Experiments on our images showed that features based solely on clique and image histograms were not robust enough against the background variations of intensities and noise. To account for the intensity distribution within the clique, we selected the seven moments of Hu [8, 13] as the input features. Their invariance to translation, scaling and rotation is particularly relevant to cytometric images.

Once the clique sizes and features are chosen, three costs are to be assigned to each clique C , namely $E_C(S_C = f)$, $E_C(S_C = b)$ and $E_C(S_C \notin \{f, b\})$. We use a feed-forward neural network to learn the first two costs. The training set is built from a manually segmented image, regarded as ground truth. For each clique C , containing $|C|$ pixels, the proportion of foreground and background pixels is:

$$\mathcal{P}_C(x) = \frac{1}{|C|} \sum_{p \in C} \chi_x(p) \text{ for } x \in \{f, b\} \quad (7)$$

where $\chi_x(p) = 1$ if $S_p = x$ and 0 otherwise. The two proportions $\mathcal{P}_C(b)$ and $\mathcal{P}_C(f)$ are computed for all the cliques from the ground truth, forming the target values to be learnt. Three pairs of target values are shown in Fig. 6, based on 3×3 cliques, one entirely in the background, one with mixed labels, and one entirely in the foreground.

We use a feed-forward neural network with seven input nodes (one for each moment of Hu), fourteen nodes in one hidden layer, and two output nodes, one for each proportion $\mathcal{P}_C(b)$ and $\mathcal{P}_C(f)$. We use scaled conjugate gradient method for training [4], which we found to be both fast and accurate. It is to be noted that strictly speaking, the output of

the neural network cannot be interpreted as probabilities, as there is no guarantee that any pair of output values will sum up to 1. Thus, we simply use the output from the trained network on clique C , noted $(\mathcal{P}_C(f), \mathcal{P}_C(b))$, to define:

$$E_C(S_C = x) = \mathcal{P}_C(x), \text{ for } x \in \{f, b\} \quad (8)$$

Finally, we define the cost of a mixed labelling as:

$$E_C(S_C \notin \{f, b\}) = \varepsilon + \max(E_C(S_C = b), E_C(S_C = f)) \quad (9)$$

to penalize mixed labels in a patch over consistent labels.

We now explain the selection of a clique size $n \times n$. For each size we need a way to evaluate the quality of the neural network responses themselves. The method we use it to compute the responses for all cliques from an image, average them spatially to perform a complete segmentation, which we then compare to the corresponding ground truth. In more details, for each value of n between 2 and 11, we build inputs and targets from an image with ground truth, as explained above, and train a neural network. On another test image (also manually segmented), we compute the seven moments for each clique C of size n , use them as inputs on the trained neural network, which returns two values $\mathcal{P}_C(b)$ and $\mathcal{P}_C(f)$. We then average the responses from all cliques where a given pixel p belongs. Let n_p be the number of such cliques, we define:

$$\tilde{E}_1(S_p = x) = \frac{1}{n_p} \sum_{C \ni p} \mathcal{P}_C(x) \text{ for } x \in \{f, b\} \quad (10)$$

$$\tilde{E}_1(p) = \frac{1}{2} \left(\tilde{E}_1(S_p = f) + (1 - \tilde{E}_1(S_p = b)) \right) \quad (11)$$

The values $\tilde{E}_1(p)$ are then thresholded, with a value between 0 and 1, which segments the image into foreground and background. For each threshold value, we compute the specificity and sensitivity (respectively true positive and true negative ratios) against the ground truth, which we use to build a receiver-operator characteristics (ROC) curve. Our results are presented in Fig. 7. The value of n which results in the largest area under the ROC curve is selected for the clique size, in our case $n = 7$.

3.3. Finalizing the results

The formulae in Eq. 3 gives equal weight to all potential functions. To account for their relative significance to the segmentation task at hand, we use a constant factor λ :

$$E(S|I) = \sum_{\substack{p \in I \\ q \in N_p}} E_2(S_p, S_q|I) + \lambda \times \sum_{C \subset I} E_C(S_C|I) \quad (12)$$

We use a similar ROC analysis to tune the value of λ , and use 0.98.

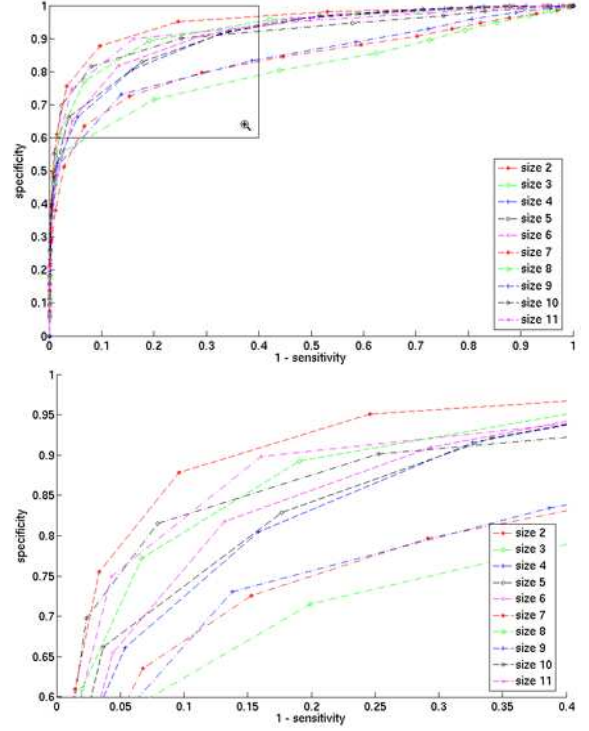


Figure 7. ROC curves for different patch sizes $n \times n$. Top: complete curves; Bottom: zoomed-in region. The size $n = 7$ corresponds to the curve with the largest area under.

4. Results

To show the importance of the different potential functions, we ran three experiments $\mathcal{E}_{\{1,2,3\}}$, on several images with manually defined ground truth. In \mathcal{E}_1 , we only use the average results of the texture classification, defined in Eq. 11 as $\tilde{E}_1(p)$. This would correspond to using only unary terms. In \mathcal{E}_2 , we use $\tilde{E}_1(p)$ as unary potentials, and E_2 from Eq. 6 as pairwise terms, with no clique potentials. Finally in \mathcal{E}_3 , we use the complete segmentation presented in the previous section, with no unary term E_1 , with E_2 from Eq. 6, and E_C from Eq. 9 as clique potential.

A common issue in evaluating segmentation results, in particular for medical images, is the amount of blur and local ambiguity in the exact location of borders, which causes manual segmentation not to be one-pixel accurate. While manual segmentation remains the reference, care is required when using evaluation measures. In order to compare our results with published work, we use sensitivity and specificity measures, defined as:

$$Sv = \frac{|C \cap M|}{|M|}, \quad Sp = \frac{|I - C \cap M|}{|I - M|} \quad (13)$$

where M is the manual segmentation, C the computed one, I the complete image, and $|X|$ stands for the number of pix-

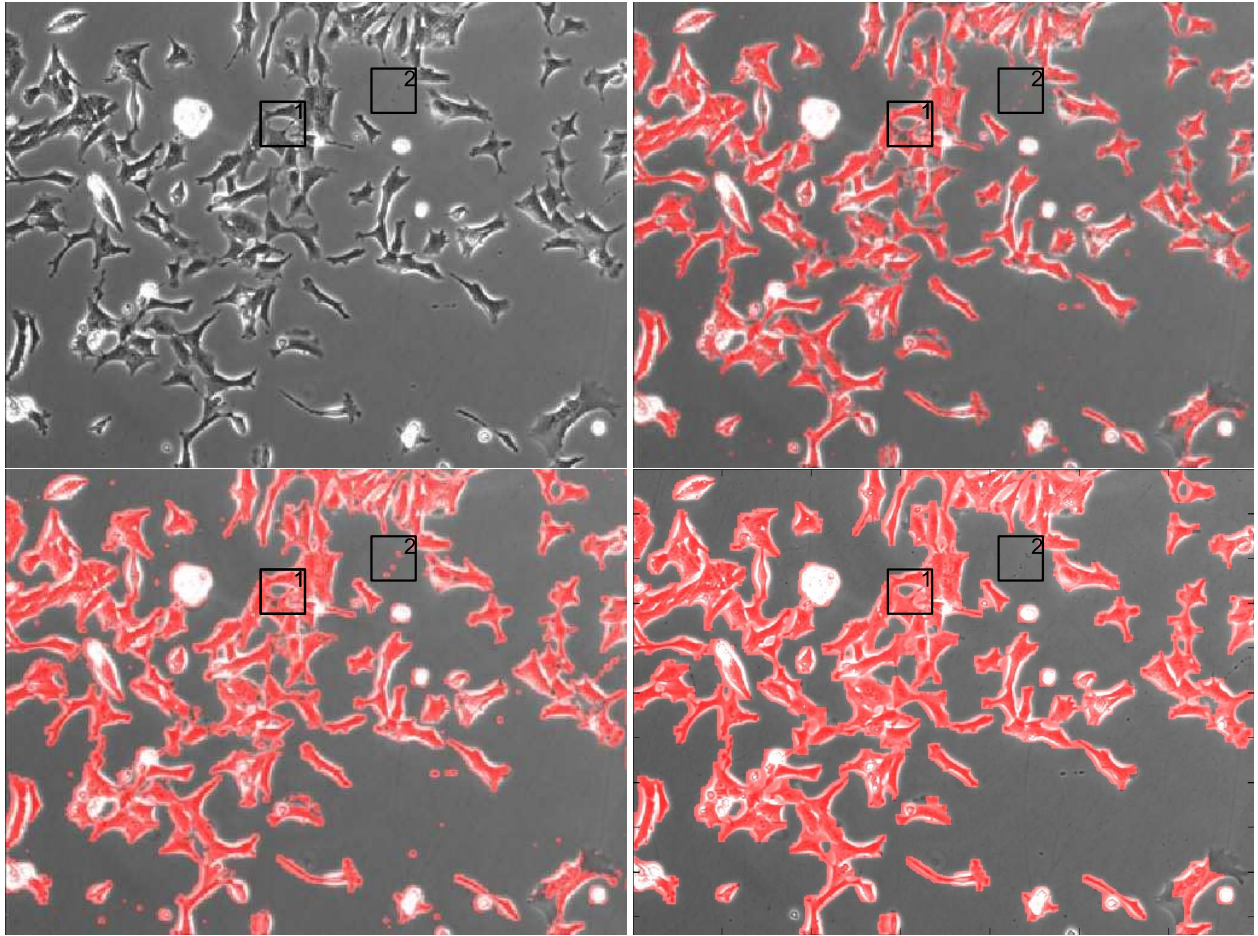


Figure 8. Segmentation results. Top left: original image. Top right: segmentation result from experiment \mathcal{E}_1 , showing many holes in cells and segmented noise in the background. Bottom left: segmentation result from experiment \mathcal{E}_2 , with smaller holes but still noise. Bottom right: segmentation result from experiment \mathcal{E}_3 , where most of the holes are filled in (Box 1), and no noise is segmented (Box 2).

els in set X . These measures are an indication of the overall quality of the segmentation. However, they do not take into account the geometry of the mis-labelled regions, which can be linked to the severity of the segmentation errors. To this end, other measures are based on the Hausdorff distances between the manual and computed segmented regions. We use one such measure, based on a computationally efficient method described in [2]. By taking the distance transform of the ground truth in both foreground and background regions, each pixel in the transform is assigned its distance to the nearest cell border. We only consider the mis-labelled pixels, and take the average of their values in the distance transform, noted \mathcal{D} . This way, mis-labelled pixels around the cells' borders (in the region where ground truth is uncertain) have little influence on the measure, while mis-labelled pixels well inside foreground and background regions increase the measure.

Our results are listed in Table 4, and the results from one

Table 1. Results of the quantitative evaluation of the segmentation results.

<i>Experiment</i>	<i>Sv</i>	<i>Sp</i>	<i>D</i>
\mathcal{E}_1	93.3%	94.0%	10.5 pxl
\mathcal{E}_2	93.3%	93.1%	11.3 pxl
\mathcal{E}_3	93.7%	97.0%	3.4 pxl

slide are illustrated in Fig. 8. It appears that in all three experiments, the proportions of true and false positive (given by the sensitivity and specificity) are about the same. The benefit of using the clique potentials is shown with the distance measure \mathcal{D} . It shows that while as many pixels are mis-segmented, they are located much closer to the actual cell boundaries, on average within 3 pixels of the ground truth. Given the intrinsic uncertainty of the actual cell borders, which is about 2 to 4 pixels for a user performing manual segmentation, this average distance is very good.

The closest published work to ours is by He *et al.* [7]. They achieve 96.8% recall and 98% precision on four images. While recall is the same as sensitivity, the precision measure is used for images with a high proportion of background, which is not always the case in our images. These results are comparable with ours, with a smaller proportion of mis-classified pixels but no information on their location. The method in [7] uses a learning method to initialize a curve around the cell's boundaries, and then uses low-level vision and curve evolution methods to segment the cells. While their method is highly successful on their data set, the use of several steps and many parameters for the segmentation makes it difficult to adapt to other images. By comparison, our method uses only one step, the graph cut, and few parameters to weight the relative importance of the potentials.

5. Conclusion

We have presented an application of recent work on the inclusion of higher-order cliques to perform image segmentation via energy minimization on CRF. Cliques are assigned potential functions, which use local texture features as a basis for assigning the cost of labelling the pixels. We have applied this method to segment live cell images. In our context, unary potentials were discarded as unreliable features for individual pixel labelling, while both pairwise and clique potentials were defined using learning techniques. Based on manually segmented images, we use a Bayesian classifier for the pairwise potentials and a neural network for the clique potentials. To the best of our knowledge this is the first use of learning methods combined with \mathcal{P}^n Potts model in medical imaging. Our quantitative results are comparable with state-of-the-art segmentation methods published in cytometry. While such methods are specialized for specific images, and require the tuning of many parameters, our approach is based on a generic model, with few and intuitive parameters to tune manually, and we believe is straightforward to adapt to other images.

This work was supported by an EPSRC research grant, by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, and by funding from the European Commission for the Special Non-invasive Advances in Fetal and Neonatal Evaluation (SAFE) Network of Excellence (LSHB-CT-2004-503243).

References

[1] D. Anoraganingrum. Cell segmentation with median filter and mathematical morphology operation. In *International Conference on Image Analysis and Processing*, pages 1043–1046, Venice, Italy, September 1999.

[2] P. Bamford. Automating cell segmentation evaluation with annotated examples. In *APRS Workshop on Digital Im-*

age Computing, pages 21–25, Brisbane, Australia, February 2003.

[3] P. Bamford and B. Lovell. Method for accurate unsupervised cell nucleus segmentation. In *IEEE Engineering in Medicine and Biology*, volume 1, pages 133–137, Istanbul, Turkey, October 2001.

[4] C. M. Bishop. *Neural Networks for Pattern Recognition*. Number ISBN 0-19-853864-2. Oxford University Press, 1995.

[5] Y. Y. Boykov and M.-P. Jolly. Interactive organ segmentation using graph cuts. In *Medical Image Computing and Computer-Assisted Intervention*, pages 276–286, Pittsburgh, PA, USA, October 2000.

[6] Y. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004.

[7] W. He, X. Wang, D. Metaxas, R. Mathew, and E. White. Cell segmentation for division rate estimation in computerized video time-lapse microscopy. *Proceedings of the SPIE*, 6431:643109–643113, March 2007.

[8] M.-K. Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8(2):179–187, February 1962.

[9] P. Kohli, M. P. Kumar, and P. H. Torr. \mathcal{P}^3 & beyond: Solving energies with higher order cliques. In *Computer Vision and Pattern Recognition*, Minneapolis, June 2007.

[10] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004.

[11] C. Restif. Towards safer, faster prenatal genetic tests: Novel unsupervised, automatic and robust methods of segmentation of nuclei and probes. In A. Leonardis, H. Bischof, and A. Prinz, editors, *European Conference on Computer Vision*, volume LNCS 3954, pages 437–450, Graz, Austria, May 2006. Springer-Verlag.

[12] C. D. Ruberto, A. Dempster, S. Khan, and B. Jarra. Segmentation of blood images using morphological operators. In IEEE, editor, *International Conference on Pattern Recognition*, volume 3, pages 397–400, Barcelona, Spain, December 2000.

[13] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Elsevier Academic Press, second edition, 2003. ISBN 0-12-685875-6.

[14] F. Yang, M. A. Mackey, F. Ianzini, G. Gallardo, and M. Sonka. Cell segmentation, tracking, and mitosis detection using temporal context. In *Medical Image Computing and Computer-Assisted Intervention*, pages 302–309, Palm Springs, USA, October 2005.

[15] C. Zimmer, E. Labruyère, V. Meas-Yedid, N. Guillén, and J.-C. Olivo-Marin. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing. *IEEE Transactions on Medical Imaging*, 21(10):1212–1221, October 2002.

A. Reparameterization of sub-modular energies

The sub-modular energy E_2 can be reparameterize to obtain a graph with positive weights, by defining new unary and pairwise potentials noted E_1^* and E_2^* , with the method detailed in [10], as follows.

An arbitrary well ordering (denoted $<$) on the pixels is imposed on the graph. We iteratively define E_1^* : it is initially set to 0, then $\forall p \in I, \forall q \in N_p$ such that $q > p$, it is updated as:

$$\begin{aligned} E_1^*(p) &+= E_2((S_p, S_q) = (f, b)) - E_2((S_p, S_q) = (b, b)) \\ E_1^*(q) &+= E_2((S_p, S_q) = (f, f)) - E_2((S_p, S_q) = (f, b)) \end{aligned} \quad (14)$$

We generate one directed edge from p to q of cost:

$$\begin{aligned} E_2^*(p, q) &= (E_2((S_p, S_q) = (f, b)) + E_2((S_p, S_q) = (b, f))) \\ &\quad - (E_2((S_p, S_q) = (b, b)) + E_2((S_p, S_q) = (f, f))) \end{aligned} \quad (15)$$

which is non-negative provided E_2 satisfies the criteria of sub-modularity.

The remaining edges of the graph are built as follows. If $E_1^*(p) < 0$, we add a directed edge from the source to the node p of cost $-E_1^*(p)$, else we add a directed edge from the node p to the sink of cost $E_1^*(p)$.