Level Set-Based Fast Multi-phase Graph Partitioning Active Contours Using Constant Memory

Filiz Bunyak and Kannappan Palaniappan*

Department of Computer Science,
University of Missouri-Columbia, Columbia, MO 65211 USA

Abstract. We present multi-phase FastGPAC that extends our dramatic improvement of memory requirements and computational complexity on two-class GPAC, into multi-class image segmentation. Graph partitioning active contours GPAC is a recently introduced approach that elegantly embeds the graph-based image segmentation problem within a continuous level set-based active contour paradigm. However, GPAC similar to many other graph-based approaches has quadratic memory requirements. For example, a 1024x1024 grayscale image requires over one terabyte of working memory. Approximations of GPAC reduce this complexity by trading off accuracy. Our FastGPAC approach implements an exact GPAC segmentation using constant memory requirement of few kilobytes and enables use of GPAC on high throughput and high resolution images. Extension to multi-phase enables segmention of multiple regions of interest with different appearances. We have successfully applied FastGPAC on different types of images, particularly on biomedical images of different modalities. Experiments on the various image types, natural, biomedical etc. show promising segmentation results with substantially reduced computational requirements.

1 Introduction

PDE-based segmentation methods such as active contours attract a considerable interest particularly in biomedical image analysis because of their advantages such as mathematical description of the underlying physics phenomena, subpixel precision, and direct extension to higher dimensions [7, 20]. Active contours can use edge-based or region-based information. Recent research is increasingly focusing on the use of statistical region-based image energies introduced in [5, 6, 12, 13, 21]. This interest is partly due to larger basin of attraction in region-based energies compared to edge-based energies. Some efficient edge-based and region-based image energy terms can be found in [8].

Image segmentation can also be formulated as a graph-partitioning problem [3, 9, 10, 16, 19]. The common theme underlying these approaches is formation of a weighted graph where each vertex corresponds to an image pixel or region. Edge weight between two vertices of the graph represents the similarity/dissimilarity between two corresponding pixels/regions. This graph is partitioned by minimizing some application specific cost function which is usually summation of the weights of the edges that

* This work is partially supported by a United States National Institutes of Health grant, NIBIB award R33 EB00573.

J. Blanc-Talon et al. (Eds.): ACIVS 2009, LNCS 5807, pp. 145–155, 2009.

are cut. Graph cut approaches have strong connections to active contours and level sets. The segmentation energies optimized by graph cuts combine boundary regularization with region-based properties in the same fashion as Mumford-Shah style functionals [3]. When image segmentation problem is formulated as the best bi-partitioning of the image into two regions A and B, minimum cut technique minimizes a cost function defined as:

$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v) \tag{1}$$

In a recent paper [17], Sumengen and Manjunath have introduced a framework called graph partitioning active contours (GPAC), that uses the equivalent energy functional in the continuous domain written as:

$$E_{CR} = \iint\limits_{R_i(\mathcal{C}(t)))} \iint\limits_{R_o(\mathcal{C}(t))} w(p_1,p_2) dp_1 dp_2 \tag{2}$$

where $\mathcal{C}$ is a curve, t is the time parameter of the evolution of $\mathcal{C}$, R_i and R_o are the interior and exterior of this curve, p_1 and p_2 are points such that $p_1 \in R_o$, $p_2 \in R_i$, and $w(p_1,p_2)$ is a dissimilarity measure between points p_1 and p_2. The pair of double integrals reflect the integration over a 2D region defined by $R_r(x,y)$.

This approach combines advantages of pairwise pixel (dis)similarity-based cost functions with flexibility of the variational methods [2]. But heavy computational and memory requirements prevent the technique's direct application to large images. Some *approximations* of the GPAC algorithm have been proposed in [17] and [2] to alleviate this problem by partitioning the input image into regular blocks or into "superpixels" and by calculating the dissimilarities at block or superpixel level respectively.

In this paper we present multi-phase FastGPAC that extends our recent dramatic improvement [4] of both memory requirements and computational complexity of two-class GPAC [17], into multi-class image segmentation. This new algorithm makes use of GPAC possible for segmentation of high throughput and high resolution images as in biomedical and geospatial applications. In Section 2, we give an overview of the graph partitioning active contours (GPAC). In Section 3, we present our efficient distribution-based reformulation. In Section 4, we extend our approach to multi-phase segmentation. Experimental results and conclusion are presented in sections 5,6.

2 Overview of Graph Partitioning Active Contours

The across-region cuts energy function (Eq. 2) that maximizes the dissimilarity between regions is not straightforward to extend to multi-class segmentation. In [2], Bertelli *et al.* reformulate this energy function in terms of pairwise dissimilarity within the regions:

$$E_{WR}(\mathcal{C}) = \iint\limits_{p_2 \in R_i(\mathcal{C})} \iint\limits_{p_1 \in R_i(\mathcal{C})} w(p_1,p_2) dp_1 dp_2 + \iint\limits_{p_2 \in R_o(\mathcal{C})} \iint\limits_{p_1 \in R_o(\mathcal{C})} w(p_1,p_2) dp_1 dp_2 \tag{3}$$

Here we use the latter (within regions) formulation. For 2-phase, both energy functions: Eq. 2 and Eq. 3 result in the same curve evolution equation (see [2] for proof).

Curve evolution corresponding to Eq. 3 can be done explicitely using snake-like active contours or implicitely using level set-based active contours. Level set-based approach provide advantages such as eliminating the need to reparameterize the curve and automatic handling of topology changes [15].

In level set-based active contours, a curve $\mathcal{C}$ ($\mathcal{C} = \partial r$, boundary of an open set $r \in \Omega$) is represented implicitly via a Lipschitz function $\phi : \Omega \mapsto \mathbb{R}$ by $\mathcal{C} = \{(x, y)|\phi(x, y) = 0\}$, and the evolution of the curve is given by the zero-level curve of the function $\phi(t, x, y)$ [5]. Ω represents the whole image domain, r represents inside of the curve $\mathcal{C}$ (foreground). , $(\Omega \setminus r)$ represents outside of the curve $\mathcal{C}$ (background). The function ϕ is positive inside and negative outside of the curve $\mathcal{C}$. Heaviside function:

$$H(\phi) = \begin{cases} 1 \text{ if } \quad \phi > 0 \\ 0 \; elsewhere \end{cases}$$

is used as an indicator function for the points inside and outside of $\mathcal{C}$. Using level set representation Eq. 3 is written as:

$$E = \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) H(\phi(p_1)) H(\phi(p_2)) dp_1 dp_2 + \iint_{\Omega} \iint_{\Omega} w(p_1, p_2) \Big(1 - H(\phi(p_1))\Big) \Big(1 - H(\phi(p_2))\Big) dp_1 dp_2 \quad (4)$$

Steepest descent minimization of Eq. 4 leads to the curve evolution equation:

$$\frac{\partial \phi(p_2)}{\partial t} = \delta(\phi(p_2))[\iint_{\Omega} w(p_1, p_2) \Big(1 - H(\phi(p_1))\Big) dp_1 - \iint_{\Omega} w(p_1, p_2) H(\phi(p_1))] \quad (5)$$

discretized as:

$$\frac{\Delta \phi(p_2)}{\Delta t} = \delta_{\epsilon}(\phi(p_2)) \Big[\sum_{p_1 \in R_o(\mathcal{C})} w(p_2, p_1) - \sum_{p_1 \in R_i(\mathcal{C})} w(p_2, p_1) \Big] \quad (6)$$

where δ_{ϵ} is discretized delta function. We refer to the expression inside the brackets as region variability term (RV). For regularization, mean curvature flow which adds to the variational cost function a term proportional to the length of the zero level set contour is used [5]:

$$\mu \cdot Length(\mathcal{C}) = \mu \cdot \iint_{\Omega} |\nabla H(\phi(p))| dp = \mu \cdot \iint_{\Omega} \delta(\phi(p)) |\nabla \phi(p)| dp \quad (7)$$

curve evolution corresponding to this term is:

$$\frac{\partial \phi}{\partial t} = \mu \delta(\phi) \operatorname{div} \frac{\nabla \phi}{|\nabla \phi|} = \mu \delta \mathcal{K} \quad (8)$$

Regularization term is discretized as described in [5, 14].

The computational bottleneck in GPAC algorithm is the calculations of the regional sums in Eq. 6 that measure the (dis)similarity of each image point p_1 to every points in regions $R_o(\mathcal{C})$ and $R_i(\mathcal{C})$ respectively. For an image of size N^2, this results in $O(N^2 \times N^2)$ pixel-to-pixel dissimilarity computation operations. To avoid recomputation at each contour evolution iteration, these dissimilarities are pre-computed once and kept in memory as a look-up table. For an image of size N^2, this results in a symmetric dissimilarity matrix $\mathbf{W}$ with N^2 rows and N^2 columns, where an element $W(i,j)$ is $w(p_i, p_j)$, dissimilarity of pixels p_i and p_j. Number of such dissimilarities kept in this method is $N^4/2$. Since even for small images $\mathbf{W}$ becomes quite large and hard to fit in memory (1TB for a $1K \times 1K$ image and 2 bytes per $w(p_1, p_2)$), approximations are proposed in [17] and in [2] by partitioning the input image into blocks or into "superpixels" respectively.

3 Fast Graph Partitioning Active Contours (FastGPAC)

While pairwise pixel similarity-based cost functions are powerfull, direct implementation of GPAC is quite costly. For an N^2-pixel image, computational complexity and memory requirements of the look-up table $\mathbf{W}$ are both $O(N^4)$, since (dis)similarity of each N^2 pixels to each N^2 pixels is computed and stored. Use of the original GPAC algorithm for high resolution or high throughput image analysis is not feasible because of these high memory requirements and computational complexity. In this section, we summarize our fast graph partitioning active contours (FastGPAC) approach that dramatically reduces the computational complexity and the memory requirements of GPAC to $O(N^2)$ and $O(1)$ respectively by reducing the 2D regional sums in Eq. 6 into 1D distribution sums. Detailed complexity analysis for GPAC and FastGPAC can be found in [4].

For feature-based (dis)similarity measures $w(p_1, p_2)$ that does not involve spatial distance between pixels p_1, p_2. $w(p_1, p_2)$ can be re-written as:

$$w(p_1, p_2) \equiv D(F(p_1), F(p_2)) \tag{9}$$

where $F(p)$ is a feature extracted at point $p(x, y)$, and D is a dissimilarity measure defined on F (i.e. for absolute grayscale intensity difference, $w(p_1, p_2) = |I(p_1) - I(p_2)|$, feature $F(p)$ is grayscale intensity $I(p)$ and the dissimilarity measure D is L_1 metric.)

Theorem 3.1 (GPAC Region Sum Theorem). For cases where the dissimilarity measure $w(p_1, p_2)$ is feature-based (does not involve spatial distance between pixels), the 2D regional sums, $\sum_{p_1 \in R_r} w(p_2, p_1)$ (for $R_r = R_i$ or R_o) can be reduced to 1D sums independent of the spatial size or shape of the regions $R_r(\mathcal{C})$.

$$\sum_{p_1 \in R_r} w(p_1, p_2) \equiv \sum_{j=0}^{L-1} h_r(j) \times D(F(p_2), j) \tag{10}$$

where $\mathbf{h_r}$ is the histogram of the feature F in the r^{th} region R_r, $D()$ is a (dis)similarity measure, L is number of bins in $\mathbf{h}$, and $h_r(j) = \sum_{p_1 \in R \wedge F(p_1) = j} 1$ is the j^{th} bin of $\mathbf{h_r}$

corresponding to the number of points $p_1 \in R_r$ whose feature $F(p_1) = j$. In the more general case, $D(F(p_2), j)$ is replaced by $D(F(p_2), C(j))$ where $C(j)$ is a representative value for the histogram bin $h(j)$.

Proof. This equality is derived by grouping the points p into feature class bins $F(p) = j$, and by separating the original sum into two nested sums as follows:

$$\begin{aligned}\sum_{p\in R_r} w(p_2, p_1) \equiv \sum_{p_1\in R_r} D(F(p_2), F(p_1)) &= \sum_{j=0}^{L-1} \sum_{p_1\in R_r \wedge F(p_1)=j} D(F(p_2), F(p_1)) \\ &= \sum_{j=0}^{L-1} D(F(p_2), j) \times \underbrace{\sum_{p_1\in R_r \wedge F(p_1)=j} 1}_{h(j)} \\ &= \sum_{j=0}^{L-1} D(F(p_2), j) \times h(j) \qquad (11)\end{aligned}$$

In this case the discretized region variability term becomes:

$$\begin{aligned}&\sum_{p_1\in R_o(\mathcal{C})} w(p_2, p_1) - \sum_{p_1\in R_i(\mathcal{C})} w(p_2, p_1) \\ &= \sum_{j=0}^{L-1} h_o(j) \times D(F(p_2), j) - \sum_{j=0}^{L-1} h_i(j) \times D(F(p_2), j) \\ &= \sum_{j=0}^{L-1} [h_o(j) - h_i(j)] \times D(F(p_2), j) \qquad (12)\end{aligned}$$

and the curve evolution equation in 6 transforms into:

$$\frac{\Delta\phi(p_2)}{\Delta t} = \delta_\epsilon(\phi(p_2))\Big[\sum_{j=0}^{L-1} [h_o(j) - h_i(j)] \times D(F(p_2), j)\Big] \qquad (13)$$

4 Extension of FastGPAC into Multi-phase Segmentation

The two phase segmentation scheme described above segments the image into two: foreground and background, and is appropriate for segmentation of single class objects/appearances from a single class background. However images often contain multiple classes of objects or multiple regions of interests with different appearances. Various approaches can be used to extend 2-phase segmentation to deal with multiple classes such as hierarchical 2-phase segmentations or N-level sets where each class is assigned a level set. In [18], Chan and Vese presented a multi-phase extension of their two-phase level set image segmentation algorithm [5]. The multi-phase approach enables efficient partitioning of the image into n classes using just $log(n)$ level sets without leaving any gaps or having overlaps between level sets. This ensures that each pixel is properly

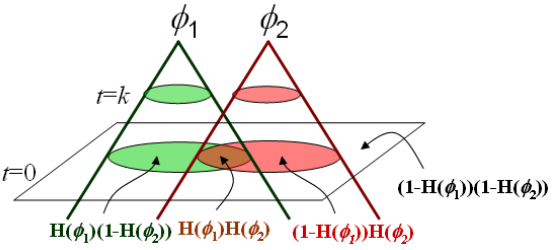

Fig. 1. A four-phase level set partitioning, with two level set functions ϕ_1 and ϕ_2. $H(\phi_1)$ and $H(\phi_2)$ are the Heaviside functions associated with each level set function.

assigned to a unique class during the segmentation process. Figure 1 illustrates the 4-phase (2 level set) case.

In [2], GPAC is extended to multi-phase in a way similar to Vese and Chan's multi-phase extension [18]. In this scheme, the n-phase energy function becomes:

$$E = \sum_{i}^{n} \iint_{\Omega} \iint_{\Omega} w(p_1, p_2)\chi_i(p_1)\chi_i(p_2)dp_1dp_2 \tag{14}$$

χ_i is the indicator function that takes value of 1 for phase i, and 0 otherwise. $k = log_2 n$ level set functions are used to represent n phases. 4-phase (2 level set) GPAC energy function is written as:

$$\begin{aligned} E_{WR} = & \iint_{\Omega} \iint_{\Omega} w(p_1, p_2)[\chi_1(p_1)\chi_1(p_2)]dp_1dp_2 \\ & + \iint_{\Omega} \iint_{\Omega} w(p_1, p_2)[\chi_2(p_1)\chi_2(p_2)]dp_1dp_2 \\ & + \iint_{\Omega} \iint_{\Omega} w(p_1, p_2)[\chi_3(p_1)\chi_3(p_2)]dp_1dp_2 \\ & + \iint_{\Omega} \iint_{\Omega} w(p_1, p_2)[\chi_4(p_1)\chi_4(p_2)]dp_1dp_2 \end{aligned} \tag{15}$$

where the indicator functions χ_{1-4} for the four regions are:

$$\chi_1(p) = H(\phi_1(p))H(\phi_2(p)); \qquad \chi_2(p) = H(\phi_1(p))(1 - H(\phi_2(p)))$$
$$\chi_3(p) = (1 - H(\phi_1(p)))H(\phi_2(p)); \quad \chi_4(p) = (1 - H(\phi_1(p)))(1 - H(\phi_2(p)))$$

Curve evolution for this 4-phase energy function is then:

$$\begin{aligned} \frac{\partial \phi_1(p_2)}{\partial t} = & \delta(\phi(p_2))\Big\{H(\phi_2(p_2))\Big[\iint_{\Omega} w(p_1, p_2)\chi_3(p_1)dp_1 - \iint_{\Omega} w(p_1, p_2)\chi_1(p_1)dp_1\Big] \\ & + (1 - H(\phi_2(p_2)))\Big[\iint_{\Omega} w(p_1, p_2)\chi_4(p_1)dp_1 - \iint_{\Omega} w(p_1, p_2)\chi_2(p_1)dp_1\Big]\Big\} \\ \frac{\partial \phi_2(p_2)}{\partial t} = & \delta(\phi(p_2))\Big\{H(\phi_2(p_2))\Big[\iint_{\Omega} w(p_1, p_2)\chi_2(p_1)dp_1 - \iint_{\Omega} w(p_1, p_2)\chi_1(p_1)dp_1\Big] \\ & + (1 - H(\phi_2(p_2)))\Big[\iint_{\Omega} w(p_1, p_2)\chi_4(p_1)dp_1 - \iint_{\Omega} w(p_1, p_2)\chi_3(p_1)dp_1\Big]\Big\} \end{aligned} \tag{16}$$

Note that the normalization and regularization terms are ignored for simplicity of the notation, otherwise these terms needs to be included.

Although FastGPAC is initially defined for binary (2-phase) segmentation [4], GPAC region sum theorem equally applies to the regional sums in Eq. 16. As in the 2-phase case the 2D regional sums $\sum_{p_1 \in R_r} w(p_2, p_1)$ are reduced to 1D sums independent of the spatial size or shape of the regions R_r. In n-phase case R_r refers to the n regions/classes defined by $k = log(n)$ level sets. In 2-phase these regions correspond to R_i, R_o, in 4-phase these regions correspond to $\chi_1, .., \chi_4$ defined above. FastGPAC transforms the 4-phase curve evolution equation Eq. 16 into its more efficient form:

$$\frac{\Delta\phi_1(p_2)}{\Delta t} = \delta_\epsilon(\phi_1(p_2))\Big\{H(\phi_2(p_2)) \sum_{j=0}^{L-1}[h_3(j) - h_1(j)]D(F(p_2), j) + (1 - H(\phi_2(p_2))) \sum_{j=0}^{L-1}[h_4(j) - h_2(j)]D(F(p_2), j)\Big\} \quad (17)$$

$$\frac{\Delta\phi_2(p_2)}{\Delta t} = \delta_\epsilon(\phi_2(p_2))\Big\{H(\phi_1(p_2)) \sum_{j=0}^{L-1}[h_2(j) - h_1(j)]D(F(p_2), j) + (1 - H(\phi_1(p_2))) \sum_{j=0}^{L-1}[h_4(j) - h_3(j)]D(F(p_2), j)\Big\} \quad (18)$$

where $h_1, .., h_4$ denotes the regional histograms corresponding to the regions $\chi_1, .., \chi_4$. While the actual number of operation in FastGPAC increases linearly with number of phases, complexity order remains the same since N^4 point-to-point dissimilarity computation is reduced to N^2 pixel to n histograms dissimilarity computation, where n is number of phases.

5 Experimental Results

In this section we compare original GPAC [1] and FastGPAC results in terms of computational time and similarity of the segmentation masks. And we present sample two-phase and multi-phase FastGPAC segmentation results for different imaging applications and modalities including natural scenery, various biomedical imagery (MRI, histopathology, fluorescent microscopy), and geospatial images.

In order to make a fair comparison regardless of coding optimizations, GPAC code by Luca Bertelli [1] is used as a base for the FastGPAC code. Both programs (original GPAC [1] and our FastGPAC) are written as a combination of Matlab and C functions. Both programs are tested on Intel Xeon 5140 dual core with dual processor machines with 10 GB RAM . While [2] extends the original 2-phase GPAC to multi-phase, the source code we use for original GPAC [1] is a 2-phase segmentation code, thus the speed-up and mask similarity are given for 2-phase GPAC and FastGPAC results. As discussed in Section 4, similar speed-ups with a constant factor are expected for multi-phase FastGPAC versus multi-phase GPAC.

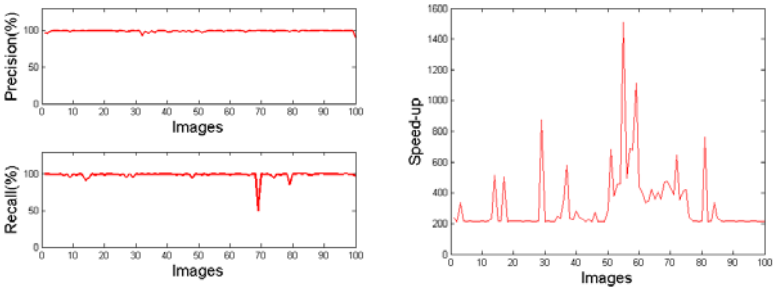

Fig. 2. Quantitative FastGPAC vs. GPAC comparison results on the Berkeley Segmentation Dataset [11]. a) Segmentation mask similarity in terms of recall and precision. b) Speed-up obtained by FastGPAC.

Fig. 3. Sample 2-phase FastGPAC segmentation results for images 42049, 253027,227092, 302008, 376043 from Berkeley segmentation dataset [11]. First row: original image, second row: foreground region.

Figure 2 shows quantitative comparison results on the Berkeley Segmentation Dataset [11]. Detailed complexity analysis for 2-phase GPAC and FastGPAC can be found in our recent paper [4]. Segmentation results for selected images are shown in Figure 3. For both programs input images are converted from RGB color space into YC_bC_r color space and resized to 241×146 pixels (because of the original GPAC's memory requirements). The luma component Y is scaled into [0-100], chroma components C_b and C_r are scaled into [0-200] ranges. Parameters $\mu = 4000$, $\lambda_1 = \lambda_2 = 1$, ($\epsilon = 0.001, \delta_t = 0.5$ for level set) are used for both programs. Segmentation similarity (Figure 2a) is measured in terms of recall and precision defined as below:

$$\text{recall} = \frac{|Mask_G \cap Mask_F|}{|Mask_G|}, \qquad \text{precision} = \frac{|Mask_G \cap Mask_F|}{|Mask_F|} \tag{19}$$

where $Mask_G$, $Mask_F$ are GPAC and FastGPAC segmentation masks respectively. Mean precision and recall of the FastGPAC masks compared to corresponding GPAC masks are 99% and 98.3% respectively. The deep minimum (around 70 in x-axis) corresponds to the image #96073 where the background and foreground color distributions

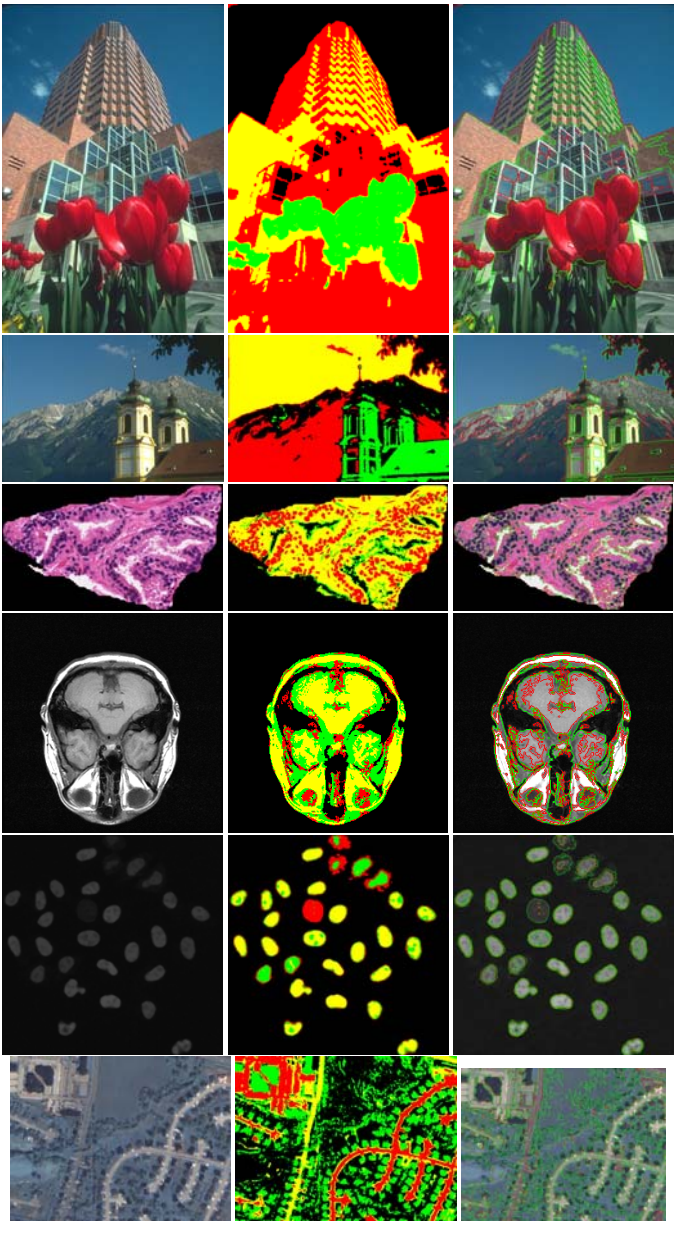

Fig. 4. Sample segmentation results for different imaging applications and modalities including natural scenery (rows 1,2), various biomedical imagery (histopathology (row 3), MRI (rows 4), fluorescent microscopy (row 5)), and geospatial images (row 6). First column: original image, second column: segmentation mask, third column:segmentation contours superimposed on the original image, first phase as red, second phase as green lines.

are very similar and texture is the distinctive feature that should be used for segmentation. For this image both GPAC and FastGPAC programs used with only color features fail dramatically producing segmentation masks far from the desired mask. Figure 2b shows speed-up obtained for images of the same dataset. FastGPAC results in an average of $322\times$ speed-up in elapsed time and $226\times$ speed-up in CPU time. CPU time is the time for which the CPU was busy executing the task. It does not take into account the time spent in disk I/O. Because of its large memory requirements, $O(N^4)$ for an image of size N^2, GPAC uses more virtual memory than FastGPAC increasing its disk access and elapsed time. For larger image sizes the difference between CPU time and elapsed time becomes even larger. These speed-up numbers illustrate that even for quite small images there is a significant timing difference between original GPAC and FastGPAC algorithms, which increases dramatically for larger images considering $O(N^2)$ complexity for FastGPAC vs. $O(N^4)$ complexity for original GPAC.

FastGPAC's dramatic improvements in memory requirements and computational complexity enable use of GPAC on high throughput and high resolution images (e.g. biomedical and geospatial). Figure 4 presents sample multi-phase segmentation results for different imaging applications and modalities including natural scenery (rows 1,2), various biomedical imagery (histopathology (row 3), MRI (rows 4), fluorescent microscopy (row 5)), and geospatial images (row 6).

6 Conclusion and Future Works

We present a fast formulation of graph partitioning active contours (GPAC) that combines accuracy and dramatic speed-up. The new formulation reduces memory requirements from $O(N^4)$ to $O(1)$ and computational complexity from $O(N^4)$ to $O(N^2)$ for an $N \times N$ image. We further extend this fast formulation from two-phase segmentation to multi-phase segmentation, enabling segmentation of multiple regions of interests with different appearances. FastGPAC algorithm provides flexibility of the GPAC algorithm enabled by possibility of using different image features and different dissimilarity measures while being computationally feasible for large images, high spatial resolutions, and long image sequences. We have successfully applied FastGPAC to biomedical, geospatial and natural imagery. Experiments show promising segmentation results. Additional speedup can be obtained through parallelization on GPU or Cell processors, and will be explored as future work.

References

1. Bertelli, L.: GPAC implementation, `http://vision.ece.ucsb.edu/~lbertelli/soft_GPAC.html`
2. Bertelli, L., Sumengen, B., Manjunath, B., Gibou, F.: A variational framework for multiregion pairwise similarity-based image segmentation. IEEE Trans. on Patt. Anal. Mach. Intell. 30(8), 1400–1414 (2008)
3. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. Int. J. Comp. Vision 70(2), 109–131 (2006)
4. Bunyak, F., Palaniappan, K.: Efficient segmentation using feature-based graph partitioning active contours. In: Int'l Conf. Computer Vision (September 2009)

5. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Proc. 10(2), 266–277 (2001)
6. Cohen, L.: On active contour models and balloons. Comput. Vis., Graphics, Image Processing 53, 211–218 (1991)
7. Dejnozkova, E., Dokladal, P.: Embedded real-time architecture for level-set-based active contours. EURASIP Journal on Applied Signal Processing 2005(17), 2788–2803 (2005)
8. Jacob, M., Blu, T., Unser, M.: Efficient energies and algorithms for parametric snakes. IEEE Trans. Image Process. 13(9), 1231–1244 (2004)
9. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Patt. Anal. Mach. Intell. 26(2), 147–159 (2004)
10. Malcolm, J., Rathi, Y., Tannenbaum, A.: A graph cut approach to image segmentation in tensor space. In: IEEE Conf. Comp. Vision and Patt. Rec., pp. 1–8 (2007)
11. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision, July 2001, vol. 2, pp. 416–423 (2001)
12. Paragios, N., Deriche, R.: Geodesic active regions for motion estimation and tracking. In: Proc. Int. Conf. Computer Vision, Corfu, Greece, pp. 688–694 (1999)
13. Ronfard, R.: Region-based strategies for active contour models. Int. J. Comput. Vision 13, 229–251 (1994)
14. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D 60, 259–268 (1992)
15. Sethian, J.: Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry. In: Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, Cambridge (1999)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Patt. Anal. Mach. Intell. 22(8), 888–905 (2000)
17. Sumengen, B., Manjunath, B.S.: Graph partitioning active contours (GPAC) for image segmentation. IEEE Trans. Patt. Anal. Mach. Intell., 509–521 (April 2006)
18. Vese, L., Chan, T.: A multiphase level set framework for image segmentation using the Mumford and Shah model. Int. J. Computer Vision 50(3), 271–293 (2002)
19. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. IEEE Trans. Patt. Anal. Mach. Intell. 15(11), 1101–1113 (1993)
20. Wang, X., He, W., Metaxas, D., Matthew, R., White, E.: Cell segmentation and tracking using texture-adaptive snakes. In: Proc. IEEE Int. Symp. Biomedical Imaging, Washington, DC, April 2007, pp. 101–104 (2007)
21. Zhu, S., Yuille, A.: Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. IEEE Trans. Patt. Anal. Mach. Intell. 18, 884–900 (1996)