ARTICLE IN PRESS
JID: YCVIU [m5G;May 24, 2016;17:55]
Computer Vision and Image Understanding 000 (2016) 1–14
Contents lists available at ScienceDirect
Computer Vision and Image Understanding
journal homepage: www.elsevier.com/locate/cviu

A generalized graph reduction framework for interactive segmentation of large images ☆

Houssem-Eddine Gueziri*, Michael J. McGuffin, Catherine Laporte*

École de technologie supérieure, 1100 Notre-Dame Ouest, Montreal, Québec H3C 1K3, Canada

ARTICLE INFO

Article history:
Received 3 July 2015
Revised 8 March 2016
Accepted 16 May 2016
Available online xxx

Keywords:
Interactive segmentation
User study
Graph-based segmentation
Graph reduction
Random walker
Graph cuts

ABSTRACT

The speed of graph-based segmentation approaches, such as random walker (RW) and graph cut (GC), depends strongly on image size. For high-resolution images, the time required to compute a segmentation based on user input renders interaction tedious. We propose a novel method, using an approximate contour sketched by the user, to reduce the graph before passing it on to a segmentation algorithm such as RW or GC. This enables a significantly faster feedback loop. The user first draws a rough contour of the object to segment. Then, the pixels of the image are partitioned into "layers" (corresponding to different scales) based on their distance from the contour. The thickness of these layers increases with distance to the contour according to a Fibonacci sequence. An initial segmentation result is rapidly obtained after automatically generating foreground and background labels according to a specifically selected layer; all vertices beyond this layer are eliminated, restricting the segmentation to regions near the drawn contour. Further foreground/background labels can then be added by the user to refine the segmentation. All iterations of the graph-based segmentation benefit from a reduced input graph, while maintaining full resolution near the object boundary. A user study with 16 participants was carried out for RW segmentation of a multi-modal dataset of 22 medical images, using either a standard mouse or a stylus pen to draw the contour. Results reveal that our approach significantly reduces the overall segmentation time compared with the status quo approach ($p < 0.01$). The study also shows that our approach works well with both input devices. Compared to super-pixel graph reduction, our approach provides full resolution accuracy at similar speed on a high-resolution benchmark image with both RW and GC segmentation methods. However, graph reduction based on super-pixels does not allow interactive correction of clustering errors. Finally, our approach can be combined with super-pixel clustering methods for further graph reduction, resulting in even faster segmentation.

1. Introduction

Image segmentation consists of delineating specific (foreground) objects from the background of a given image. This task plays a crucial role in biomedical image analysis. Emerging applications, for example tumour measurement, 3D organ reconstruction or cell counting, typically require a segmentation step. This task can be achieved with varying degrees of user involvement, on a continuum from fully-manual to fully-automated. Manual approaches are time consuming and lack repeatability, whereas fully-automated approaches are not applicable in complex scenarios. A compromise between these extremes is interactive segmentation, where the user supervises and adjusts inputs in response to intermediate segmentation results. Because the user can modify inputs as long as the results are unsatisfactory, *human factors* (i.e., usability) are an important consideration.

Recently, graph-based approaches have gained popularity for interactive segmentation (Boykov and Jolly, 2001; Grady, 2006; Li et al., 2004; McGuinness and O'Connor, 2010; Mortensen and Barrett, 1998; Protiere and Sapiro, 2007; Rother et al., 2004). The idea is to represent the image as a graph, where vertices correspond to pixel locations and edges represent pixel adjacency. Edges are weighted as a function of their likelihood of crossing an object boundary. For interactive segmentation to be practical, the computation of the edge weights and segmentation must be fast, enabling a tight feedback loop. However, in graph-based segmentation, computation time increases with graph size, often precluding interactive segmentation of high-resolution images. The total time to perform a segmentation also depends on human factors, such as the input device used and the kind of input required. These chal-

☆ For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article

* Corresponding authors.
E-mail addresses: houssem-eddine.gueziri.1@ens.etsmtl.ca (H.-E. Gueziri), Catherine.Laporte@etsmtl.ca (C. Laporte).

http://dx.doi.org/10.1016/j.cviu.2016.05.009
1077-3142/© 2016 Published by Elsevier Inc.

lenges are addressed in this paper. We present a graph reduction approach that is guided by a rough drawing of the object boundary provided by the user. Our preliminary work (Gueziri et al., 2015) used automatically simulated input drawings to show that this approach speeds up computations for random walker segmentation. This paper further investigates the approach and extends our analysis to make the following additional contributions:

- A controlled experiment compared performance with two input techniques and two input devices (mouse and stylus pen).
- The graph reduction approach is extended to different interactive graph-based segmentations ensuring a precise, high-resolution segmentation.
- We evaluate our approach alongside, and in combination with, graph reduction methods based on single and multi-resolution super-pixels (Achanta et al., 2012) to benefit from further speed-ups.

Section 2 reviews work related to interactive graph-based segmentation. Section 3 describes our user-guided graph reduction approach, and Section 4 discusses some of its key properties. Section 5 presents the user study, and Section 6 presents benchmarks obtained by generalizing our approach to other graph-based segmentation methods and exploiting super-pixel-based reductions. Finally, Section 7 discusses the benefits and limitations of our approach, and future directions.

2. Related work

2.1. Interactive graph-based segmentation

To preserve the intuitive character of manual segmentation, Mortensen and Barrett (1998) proposed *Intelligent Scissors* (IS), which define a cost function measuring the likelihood of graph edges crossing an obje ct boundary. While the user draws a contour near the object boundary, this contour is adjusted on the fly using Dijkstra's algorithm so as to follow a minimum-cost path in the graph. Extensions of IS, including work by Mishra et al. (2008) and the Magnetic Lasso offered in Adobe's commercial Photoshop software, were proposed to enhance segmentation flexibility. IS and its variants have two drawbacks: since the minimum-cost path must be computed in real-time during user interaction, the approach suffers from interaction feedback lags when applied to large images. Moreover, IS requires relatively high accuracy from the user when drawing the contour, which makes segmentation laborious (Li et al., 2004).

In contrast to the *contour-based interaction* required by IS, *region-based interaction* involves drawing scribbles (labels) on a small set of pixels, in the foreground and/or background regions of the image. Boykov and Jolly (2001)'s *graph cut* (GC) segmentation is a popular approach where pixels (vertices) are typically labeled in this manner. GC segmentation uses these foreground / background labels to remove edges to maximize flow (Boykov and Kolmogorov, 2004), breaking the graph into two sub-graphs (foreground and background).

Variants of GC segmentation have reduced the required user interaction (Gulshan et al., 2010). In *GrabCuts* (Rother et al., 2004), for example, the user first frames the object inside a bounding box, to reduce the search space. A Gaussian mixture model (GMM) is fitted to the cropped image intensities and labels are automatically generated according to the modes of the GMM. An initial segmentation result is then obtained using GC. The user can then add explicit foreground and background labels to adjust the segmentation. GrabCuts fails in the presence of weak boundaries, mostly because of the limited ability of the GMM to capture the true object intensity distribution. Our approach is similar to GrabCuts in that we exploit a user-drawn boundary to reduce the search space. However, our approach relies solely on the graph-based segmentation algorithm; no additional statistical model is required. Moreover, instead of confining the search space to the inside of a bounding box, the rough contour drawing is used to reduce the search space to pixels *near the object boundary*. This has three effects: (i) whereas the bounding box is constrained by object shape (e.g., a large bounding box is needed to surround a thin diagonally-oriented object), our approach is more flexible and optimizes graph reduction for complex shaped objects, (ii) our approach ignores pixels that are sufficiently *far inside* the drawn contour, resulting in a speed-up, and (iii) our approach allows more flexibility in the drawing, i.e., the drawn contour may lie slightly inside and/or outside the object to segment.

In the presence of weak boundaries, GC leads to the "small cuts" miss-segmentation problem. To address this, Grady (2006) proposed random walker (RW) segmentation, wherein unlabeled pixels are assigned *probabilities* of belonging to each label category (foreground or background). Segmentation consists of selecting the most probable label for each pixel. In the absence of edges in the image, an unlabeled pixel is assigned equal probability of belonging to equidistant labels, thereby overcoming the small cuts problem.

Like GC, the RW method segments at interactive speeds for reasonably sized images (Li et al., 2004), but is not fast enough for high resolution images. Grady and Sinop (2008) proposed to pre-compute the eigen-decomposition of the image graph's Laplacian matrix off-line to accelerate the computation of RW probabilities. However, this solution is specific to RW segmentation, while our approach adapts to other graph-based segmentation methods. Moreover, the pre-computation itself is time and memory consuming and unfeasible for live applications.

GPU parallelization has also been used to accelerate RW (Grady et al., 2005) and GC (Delong and Boykov, 2008), but is still constrained by hardware limitations because of the storage required for large datasets. Our approach is compatible with existing GPU approaches and reduces the amount of required GPU storage.

Yang et al. (2010) proposed a *constrained RW* segmentation framework where the user may draw foreground and background labels, as well as *hard* constraint labels to enforce a particular boundary alignment, and *soft* constraint labels to indicate a region where the boundary is expected to pass. This combines *contour-based input* with the traditional *region-based input* of RW segmentation. We leverage a similar combination of user input methods, wherein the user draws a rough contour of the object and then provides additional foreground / background labels to refine the segmentation results. Unlike our approach, Yang et al. (2010)'s approach does not address the issue of computation time. Moreover, their approach relies on additional energy functionals to force the boundary to pass through certain labels, corrupting the probabilities computed by RW segmentation, whereas our approach is entirely consistent with the probabilities generated with RW. Because our approach is solely based on user input to reduce graph size, it generalizes easily to other segmentation approaches that rely on foreground/background labeling, such as GC.

2.2. Interactive segmentation evaluation

Because of human factors, it is difficult to quantitatively assess interactive segmentation, especially when comparing methods involving different types of input. Olabarriaga and Smeulders (2001) enumerated three major criteria that an interactive segmentation should satisfy: (i) *Accuracy* is the degree of similarity between the segmentation and the ground truth. (ii) *Efficiency* relates to the amount of time required to perform segmentation. (iii) *Repeatability* is the extent to which similar results can be obtained from multiple segmentations of the same image. The three criteria