

2005

A new divide and conquer algorithm for graph-based image and video segmentation

Wanqing Li

University of Wollongong, wanqing@uow.edu.au

M. Shi

University of Southern Queensland

P. Ogunbona

University of Wollongong, philipo@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>



Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Li, Wanqing; Shi, M.; and Ogunbona, P.: A new divide and conquer algorithm for graph-based image and video segmentation 2005.
<https://ro.uow.edu.au/infopapers/425>

A new divide and conquer algorithm for graph-based image and video segmentation

Abstract

The concept of the Shortest (or Minimum) Spanning Tree (SST) and Recursive SST (RSST) of an undirected weighted graph has been successfully applied in image segmentation and edge detection. This paper presents a divide-and-conquer approach for (R)SST based image segmentation in order to overcome the problem of high computational complexity associated with conventional graph algorithms. In the simplest form, the proposed approach, block-based RSST (BRSST), first divides the image into rectangular blocks, finds the (R)SST of each block individually using conventional graph algorithms and, then, merges the (R) SSTs of all image blocks to form an (R)SST of the entire image. Efficient merging algorithms are presented in this paper. We proved a theorem showing that the (R)SST obtained by the merging algorithms is one of the (R)SST that would be found by applying to the entire image the same algorithm used for finding the (R) SST of each image block. Theoretical analysis and experimental results have shown that BRSST has significantly reduced the computational cost. In addition, an incremental BRSST is proposed for video segmentation and results are presented.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Li, W, Shi, M & Ogunbonba, P, A New Divide and Conquer Algorithm for Graph-based Image and Video Segmentation, IEEE 2005 7th Workshop on Multimedia Signal Processing, October 2005, 1-4. Copyright IEEE 2005.

A New Divide and Conquer Algorithm for Graph-based Image and Video Segmentation

Wanqing Li, Mingren Shi* and Philip Ogunbona

School of Information Technology and Computer Science, University of Wollongong, Australia

Department of Mathematics and Computing, University of Southern Queensland, Australia

{wanqing.philipo}@uow.edu.au, shi@usq.edu.au

Abstract—The concept of the Shortest (or Minimum) Spanning Tree (SST) and Recursive SST (RSST) of an undirected weighted graph has been successfully applied in image segmentation and edge detection. This paper presents a divide-and-conquer approach for (R)SST based image segmentation in order to overcome the problem of high computational complexity associated with conventional graph algorithms. In the simplest form, the proposed approach, block-based RSST (BRSST), first divides the image into rectangular blocks, finds the (R)SST of each block individually using conventional graph algorithms and, then, merges the (R)SSTs of all image blocks to form an (R)SST of the entire image. Efficient merging algorithms are presented in this paper. We proved a theorem showing that the (R)SST obtained by the merging algorithms is one of the (R)SST that would be found by applying to the entire image the same algorithm used for finding the (R)SST of each image block. Theoretical analysis and experimental results have shown that BRSST has significantly reduced the computational cost. In addition, an incremental BRSST is proposed for video segmentation and results are presented.

I. INTRODUCTION

Segmentation is one of the most fundamental and challenging problems in image/video processing. The problem is often defined as the partitioning an image or video frame into connected homogeneous regions in terms of a set of features, such as intensity, color and/or texture. Xu [1] and Morris [2] applied the concept of Shortest Spanning Tree (SST) to image segmentation by mapping an image onto an undirected and weighted graph. The vertices in the graph correspond to pixels and an edge encodes the spatial connectivity of the two pixels represented by the two vertices of the edge. The weight of an edge measures the similarity between the two pixels with respect to the features. Segmentation is then formulated as a process of finding an RSST of the graph and cutting the RSST into a forest of sub-graphs, each sub-graph being a segment/region. The advantages of the RSST based segmentation over conventional region-growing approaches have been demonstrated in [1], [2], [3], [4].

Finding an RSST of a graph, especially one with a large number of vertices and edges, is expensive in both time and memory requirement. Kwok et al.[5], [6] made a few attempts to overcome the problem by using a region-based process

without a need for sorting while seeking an RSST. In a recent paper [7], they explored the linking properties to produce a truncated RSST.

The contribution of this paper is an intuitive divide-and-conquer approach. The concept is that an image can be divided into small image blocks and an RSST of the image can be found by merging all RSSTs of these image blocks (We refer this as block-based RSST (BRSST)). We provide efficient merging algorithms and proved a theorem showing that the RSST obtained by the merging algorithm is one of the RSST that would be found by applying to the entire image the same algorithm used for finding the RSST of each image block. In addition, we propose an incremental BRSST for video segmentation.

II. GRAPH BASED IMAGE SEGMENTATION

Let $G = (V, E)$ represent an undirected weighted graph that has $n = |V|$ vertices, $v_i \in V, i = 1, 2, \dots, n$ and $m = |E|$ edges, $e_i \in E, i = 1, 2, \dots, m$. We assume that each vertex, v_i , has an associated attribute represented as a vector, $f_i = (f_{i1}, f_{i2}, \dots, f_{ip})$, where p is the dimension of the vector. The weight of the edge, e_i , that links the vertices, v_k and v_l , $w(e_i)$ is defined as a function of the attributes f_k and f_l of the two vertices. Thus, $w(e_i) = \Psi(f_k, f_l)$, $w(e_i) \geq 0 \forall i$, where $w(e_i)$ measures the similarity of the vertices, v_k and v_l , in terms of their attributes or associated features.

A. Graph representation

An image, $I(x, y), x = 1, 2, \dots, N; y = 1, 2, \dots, M$, can be mapped onto an undirected weighted graph, $G(I) = (V, E)$, where the vertex set $V = \{\text{all pixels of } I\}$ and the edge set,

$$E = \begin{cases} (u, v) | u, v \in V & 0 < \text{dist}(u, v) \leq 1 & \text{4-neighbour} \\ (u, v) | u, v \in V & 0 < \text{dist}(u, v) \leq \sqrt{2} & \text{8-neighbour} \end{cases}$$

where $\text{dist}(u, v)$ is the Euclidean distance in terms of the number of pixels. The attribute or feature of each pixel, such as color and motion, shall be used to define the homogeneity of image segments/regions. Similarly, an undirected weighted graph, $G(S) = (V^s, E^s)$ can be constructed from a segmented version of image I , $S, S = \bigcup_{i=1}^R S_i, S_i \cap S_j = \emptyset, \forall i, j, i \neq j$,

where S is comprised of R homogeneous segments/regions. $V^s = S$, consists of all segments and the edge set, $E^s = (u^s, v^s), u^s, v^s \in V^s$ and u^s, v^s are connected, represents the connections among segments. Each segment is characterized by its attribute or associated feature vector. The dimension of the feature vector of the regions need not be the same as that of pixels, though it is a usual practice that the same feature set is used to describe both pixels and regions. The benefit of this formalization is that dynamic/adaptive feature selection is allowed in the process of moving from pixels to segments description [4].

B. Graph based segmentation

With the graph representation of the original image and its segments, the segmentation can be defined as a mapping from the image graph, $G(I)$, to the segment graph, $G(S)$. Based on graph theory, this mapping can be achieved in two ways: non-recursive and recursive. The non-recursive approach [1], [2] considers that a segment is formed by a sub-graph of the original image graph and the mapping from $G(I)$ to $G(S)$ is achieved by first finding an SST of $G(I)$, and then cutting some edges such that connected vertices form the segments. The recursive approach forms a hierarchical representation of the image. At the top level (Level 1), the image has one segment and at the bottom level (Level $N \times M$), each pixel is a segment. At the r 'th level, there are r segments; two segments at the $(r + 1)$ 'th level are merged into one segment. Segments are merged in such way that the sum of the weights of the graphs at all levels is minimized. The recursive mapping approach leads to the RSST based algorithm [2]. Comparison of the SST and RSST based algorithms have been well discussed in [2].

C. Complexity analysis

For an image having n pixels, its 4-connectivity graph has $m = 2n - 2\sqrt{n}$ edges. If we decompose the operation of the SST/RSST mapping into sorting and processing of an edge, the average complexity of the algorithms for finding SST/RSST can be summarized in the Table I, where Kruskal algorithm [8] is assumed to construct the SST and recursively used for constructing RSST [2].

TABLE I
COMPLEXITY OF FINDING SST AND RSST

Complexity	SST	RSST
Sorting	$O(2m \log m)$	$\sum_{i=0}^{n-1} H_r(i)$
Processing	$O(m \log m)$	$O(m \log m)$
Memory	$O(mn) + O(n)$	$O(mn) + O(n)$

$H_r(i)$ in Table I above is the remaining number of edges in the current graph

III. DIVIDE AND CONQUER ALGORITHM

Definition 1: A set of edges in a connected graph the removal of which from the graph would increase the number of connected components in the graph is called a disconnecting set. If the addition of any edge of a disconnecting set would necessarily decrease the number of components in the graph, this set of edges is called a *cutset*.

It is clear that SST/RSST based segmentation is a resource-hungry algorithm, especially, for a large image whose graph has a huge number of vertices and edges. In this section, we describe a divide-and-conquer approach to address the computational problem. We first divide the graph into a set of sub-graphs by some selected cutsets. The size of each sub-graph is small enough such that the (R)SST can be found with affordable resource(memory & processing). Then, the (R)SSTs of all sub-graphs are merged to form the (R)SST of the original graph. We will show that cutsets and merging algorithms exist such that the merged (R)SST is equivalent to an (R)SST of the original graph. Without loss of generality, consider the proposed approach when the original graph is split into two sub-graphs by removing a cutset.

A. SST

Theorem 1: Suppose that a graph $G(X, E)$ is split by a cutset E^* into two connected sub-graphs $G_i = (X_i, E_i)$, $i = 1, 2$ where $X = X_1 \cup X_2$, $E = E_1 \cup E_2 \cup E^*$, and,

$$E^* = \{e_j^* = (x_{1j}, x_{2j}) | x_{ij} \in G_i, i = 1, 2, j = 1, 2, \dots, n\} \quad (1)$$

where n is the number of edges in the cutset. Let $T_i = (X_i, E_i^*)$ be the SST of sub-graph G_i , $E_i^* \subseteq E_i$, ($i = 1, 2$). Then the SST, $T^\#$, of graph $G^\# = (X^\#, E^\#)$, where $X^\# = X$ and $E^\# = E_1^* \cup E_2^* \cup E^*$, is the SST, T , of the original graph $G = (X, E)$.

Proof: Omitted, see [9] ■

Corollary 1: Theorem 1 is also valid for the special case that $G_1 = G \setminus \{e\}$, $e = (x, y)$, and $G_2 = \{y\}$ (i.e. the cutset is only an edge e). In this case, the SST of G is $T_1 \cup \{e\}$.

A simple approach to find an SST of the graph, $G^\# = (X^\#, E^\#)$, is to use the same algorithm applied to each sub-graph. This means that we need to re-sort all edges in the graph, $G^\# = (X^\#, E^\#)$. Obviously, this is not efficient and we propose an algorithm to find the SST of $G^\#$ efficiently.

Algorithm 1: (SST algorithm for $G^\# = (X^\#, E^\#)$, start from $T_1 \cup T_2$ and examine the edges of the cutset in order of non-decreasing weight)

- 1) Add the edge to $e_1^*, T_1 \cup T_2$, $T^{(1)} = T_1 \cup T_2 \cup \{e_1^*\}$ set $i = 1$ and go to (2).
- 2) If $i > n$, stop. $T^{(i)}$ is the SST. Otherwise, $T^{(i)}$ is a spanning tree of $G^\#$, set $i = i + 1$ and then go to (3).
- 3) Add the edge e_i^* to $T^{(i)}$ and remove the edge, e_i , with the maximum weight in the cycle $C^{(i)} = T^{(i)} \cup \{e_i^*\}$,

$T^{(i+1)} = T^{(i)} \cup \{e_i^*\} \setminus \{e_i\}$. And then go to (2). (If there is more than one edge with the maximum weight, we should remove e_i^* .)

Theorem 2: Algorithm 1 constructs the SST of $G^\#$.

Proof: Omitted, see [9] ■

B. RSST

Unlike SST, RSST creates a hierarchical structure of a graph with a large number of interim nodes. For instance, in an n -node graph, the RSST would have at least $(n^2/2)$ interim nodes. Let Level 1 denote the top of the hierarchy where all nodes are merged into one node. Then at Level r , there are $r, r \leq n$ nodes.

Due to the large number of interim nodes in an RSST, it is not feasible to save the entire RSSTs of sub-graphs for the purpose of future merging in our divide and conquer algorithm. For a subgraph with m node, we propose to represent the subgraph using r nodes by cutting its RSST at Level r , $r \ll m$. For image segmentation, this is equivalent to segmenting a subimage into r segments and m is chosen such that the subimage is over-segmented.

Suppose that a graph $G(X, E)$ is split by a cutset E^* into two connected graphs $G_i(X_i, E_i), i = 1, 2$ where $X = X_1 \cup X_2, E = E_1 \cup E_2 \cup E^*$. Let $T = rsst(G)$ be an RSST of graph G , and $T^R = seg(T, R)$ be the top R levels of T , then we have,

Proposition 1: There exists two integers R and r , such that

$$seg(T^o, R) = seg(T^m, R) \quad (2)$$

where

$$T_1 = rsst(G_1) \quad (3)$$

$$T_2 = rsst(G_2) \quad (4)$$

$$T^m = rsst(G(seg(T_1, r), seg(T_2, r), E^*)) \quad (5)$$

$$T^o = rsst(G) \quad (6)$$

In other words, the top R levels of the merged RSST T^m is same as the counterpart of one of the RSST, T^o , that are found using a conventional algorithm.

For image segmentation, the proposition implies that the segmentation obtained from T^m would be same as the segmentation obtained from T^o as long as we segment the image into R regions or less.

IV. BLOCK BASED RSST FOR IMAGE AND VIDEO SEGMENTATION

A. BRSST for images

To apply the divide-and-conquer approach to image segmentation, we first partitioned the image into a set of rectangular blocks. Each block is segmented using the RSST algorithm in [4] and the segmentation of the entire image is obtained by merging the segments of all blocks using the same RSST algorithm. We refer this divide-and-conquer RSST algorithm as block-based RSST or BRSST.

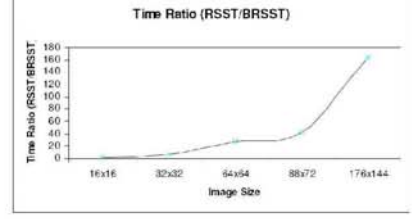


Fig. 1. Comparison of machine time used by RSST and BRSST

Let the image be divided into B blocks $I = \bigcup_{i=1}^B I_i$; T_i be the RSST of the i 'th block, denoted as $T_i = rsst(G_i)$, where $G_i = G(I_i)$ is the weighted graph of block I_i ; $S_i = seg(T_i, r)$ represent to cut the T_i at level r such that the I_i is divided into r regions. Then, BRSST is defined as

$$T^m = rsst(G(\bigcup_{i=1}^B seg(rsst(G(I_i)), r))) \quad (7)$$

Where T^m is an RSST of the image found by the BRSST. The segmentation S of image I into R segments can be obtained by $S = seg(T^m, R)$.

B. Incremental BRSST for videos

An incremental BRSST was developed for segmenting a video sequence. Let I_{k-1} and I_k be respectively the k 'th and $(k-1)$ 'th frame; S_{k-1} be the segmentation of I_{k-1} and M_k be the motion field between I_{k-1} and I_k . M_k can be either optical flow or block-based motion vectors. Incremental BRSST is to build an RSST by making use of the S_{k-1} . It consists of the following steps:

- S1 Predict the segmentation of I_k , $S'_k = P(S_{k-1}, M_k)$, where $P(\cdot)$ represents motion based prediction as well as relabelling the segments since one segment in S_{k-1} may be split into multiple segments after the motion-based prediction.
- S2 Predict the k 'th frame, $I'_k = \Psi(S'_k, I_k)$
- S3 Validate segments in S'_k using $d = |I'_k - I_k|$. A coarse segmentation, \tilde{S}_k of I_k is formed by keeping all segments in S'_k that contain pixels whose d is less than a threshold. Otherwise, the segment is disassembled into pixels.
- S4 Apply the BRSST to \tilde{S}_k , $T_k^m = rsst(G(\tilde{S}_k))$, and the segmentation of I_k is $S_k = seg(T_k^m, R)$.

The incremental BRSST takes advantage of the segmentation of previous frame. On average, the incremental BRSST starts from mixture of pixels and interim segments. In the worst case, it starts from pixels where the actual frame is significantly different from the predicted one. In this case, the incremental BRSST degrades to the BRSST.

V. EXPERIMENTAL RESULTS

Figure 1 shows the comparison of the time spent on conventional RSST and BRSST. As the size of the image



Fig. 2. Segmentation results by RSST (first row) and BRSST. The image was segmented into 9 regions

increases, BRSST has substantial computational saving and thus results in a faster method than RSST. We note that the conventional RSST of [2] requires a complexity of $O(n^2)$. Our divide-and-conquer method halves this complexity if the image is partitioned into two blocks. The argument can be extended favorably to multiple partitioning of the image. For example given a QCIF image, a block size of 8×8 , and four (4) segments per block there is a computational gain about 155. This analytical gain is consistent with the experimental result shown in Figure 1.

Furthermore, the memory requirement of both the fast RSST (FRSST) of [5] and RSST of [2] is $O(n)$. For a QCIF image with similar configuration as described above, our method requires only one-seventeenth of the memory required by the methods in [5], [2]. We note that our divide-and-conquer method, BRSST, can be tailored to fit available memory by partitioning the image appropriately. This makes this method suitable for implementation on devices with small memory specification.

Figure 2 shows the segmentation results of RSST and BRSST respectively. The visual difference between the two segmentation results is minimal. The difference is attributed to the fact that the SST obtained is one of the SSTs of the graph generated from the image.

Figure 3 shows the video segmentation results of the incremental BRSST without enforcing consistent labels to the regions over the frames during the relabelling step. We applied the BRSST to the first frame and then incremental BRSST to the rest frames using 8×8 block based motion vectors. As shown in the figure, the face, hat and the background were successfully segmented. On Average, the incremental BRSST took about one tenth time required for BRSST.

VI. DISCUSSION

We proposed a divide-and-conquer approach for graph-based image segmentation. The BRSST is a special case where each sub-graph corresponds to a rectangular image block. This partition of the image is mainly due to the nature of the pixel grid: (a) every vertex except those corresponding to boundary pixels in an image graph has same number of connections to



Fig. 3. Segmentation results by the incremental BRSST

its neighbors. There is no favorite cutsets with less number of edges, (b) the cutsets could be implicitly remembered, and hence save memory when a regular partition, such as rectangular block is used. It is interesting to note that our method is not restricted to rectangular block configuration; it can be line based. This is particularly useful for limited memory devices.

Experimental results have shown that BRSST has significant advantage in performance over conventional RSST and produce almost identical segmentation to its counterpart RSST. In addition, the divide-and-conquer algorithms proposed in this paper suggest a possible parallel architecture for graph-based image segmentation, where sub-graphs can be processed concurrently.

REFERENCES

- [1] Y. Xu and E. C. Uberbacher, "2D image segmentation using minimum spanning trees," *Image and Vision Computing*, vol. 15, pp. 47–57, 1997.
- [2] O. J. Morris, M. de J. Lee, and A.G. Constantinides, "Graph theory for image analysis: An approach based on the shortest spanning tree," *IEE Proceedings*, vol. 133, no. 2, pp. 146–152, 1986.
- [3] A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image sequence analysis for emerging interactive multimedia services-the european cost 211 framework," *IEEE Trans on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 802–813, 1998.
- [4] W. Li, P. Ogunbona, L. Ye, and I. Kharitonenko, "Visual process model and object segmentation," in *The 7th International Conference on Signal Processing*, Beijing, Sept. 2004, pp. 753–756.
- [5] S.H. Kwok and A.G. Constantinides, "A fast recursive shortest spanning tree for image segmentation and edge detection," *IEEE Trans on Image Processing*, vol. 6, no. (2), pp. 328–332, 1997.
- [6] S.H. Kwok and A.G. Constantinides, "A parallel recursive shortest spanning tree algorithm for image segmentation in distributed computing environment," *Journal of Parallel and Distributed Computing*, vol. 56, pp. 181–207, 1999, vol. 56, pp. 181–207, 1999.
- [7] S.H. Kwok, A.G. Constantinides, and W.-C. Sui, "An efficient recursive shortest spanning tree algorithm using linking properties," *IEEE Trans on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 852–863, 2004.
- [8] J. Gross and J. Yellen, *Handbook of Graph Theory*, CRC Press, 2004.
- [9] W. Li, M. Shi, and P. Ogunbona, "On SSTs of graphs and subgraphs," Tech. Rep. CVIPCM-TR-01-2004, CVIPCM, TITR, University of Wollongong, Sydney, Australia, March 2004.