"RAPID" Regions-of-Interest Detection in Big Histopathological Images

Li Yin Sulimowicz, Ishfaq Ahmad
li.yin@mavs.uta.edu, iahmad@cse.uta.edu

UNIVERSITY OF TEXAS ARLINGTON | DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Introduction

The sheer volume and size of histopathological images (*e.g.*, 10^6 MPixel) underscores the need for faster and more accurate Regions-of-interest (ROI) detection algorithms. In this paper, we propose a novel segmentation algorithm named **RAPID** (Regular and Adaptive Prediction-Induced Detection) to resolve such need. Furthermore, we employ a highly effective **parallelization scheme** to further speedup the processing speed. RAPID is based on a multi-layer coarse-to-fine segmentation algorithm [**?**, **?**], which is shown in Fig 2.

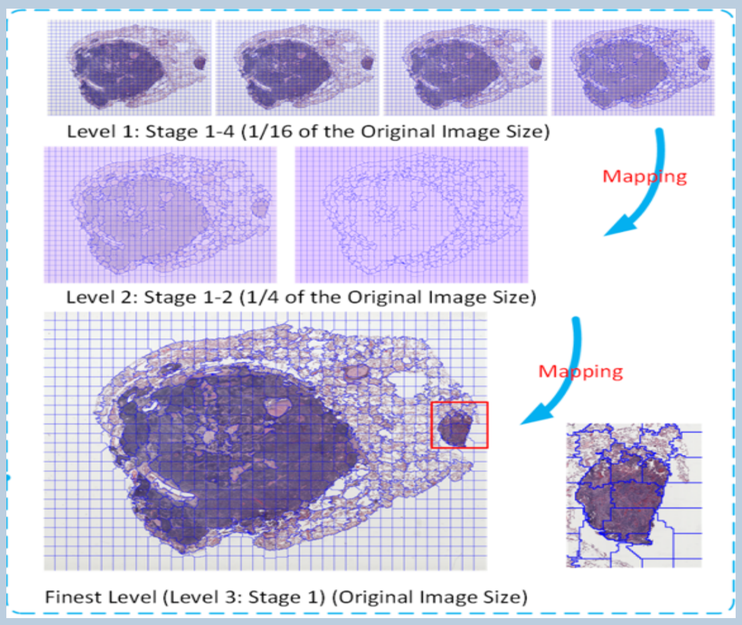

Figure 1: Multi-layer segmentation framework [**?**]

Boundary blocks from baseline algorithm [**?**, **?**] are defined in Eq. 1, which has at least one neighbor that has a different superpixel label.

$$E_b(s_p, s_q) = \begin{cases} 1, & s_p \neq s_q, \\ 0, & otherwise. \end{cases} \quad (1)$$

Conclusion

Experimental results, conducted on the BSD500 [**?**] and 500 whole-slide histological images from the National Lung Screening Trial (NLST) dataset, confirm that RAPID gained 4.6 times speedup compared with the baseline as shown in Table 1. And the parallel RAPID further speed it up to 13 times gain as shown in Table 2. Compared with the most popular used SLIC [**?**], the speedup is around 160 times, without losing accuracy. The visual segmentation effect is shown in Fig **??**, we get fine segmentation at the boundary of ROI and non-ROI, and at all the other parts the algorithms only do coarse segmentation.

References & Acknowlegement

We thank National Cancer Institute for offering the NLST dataset.

Materials and Methods

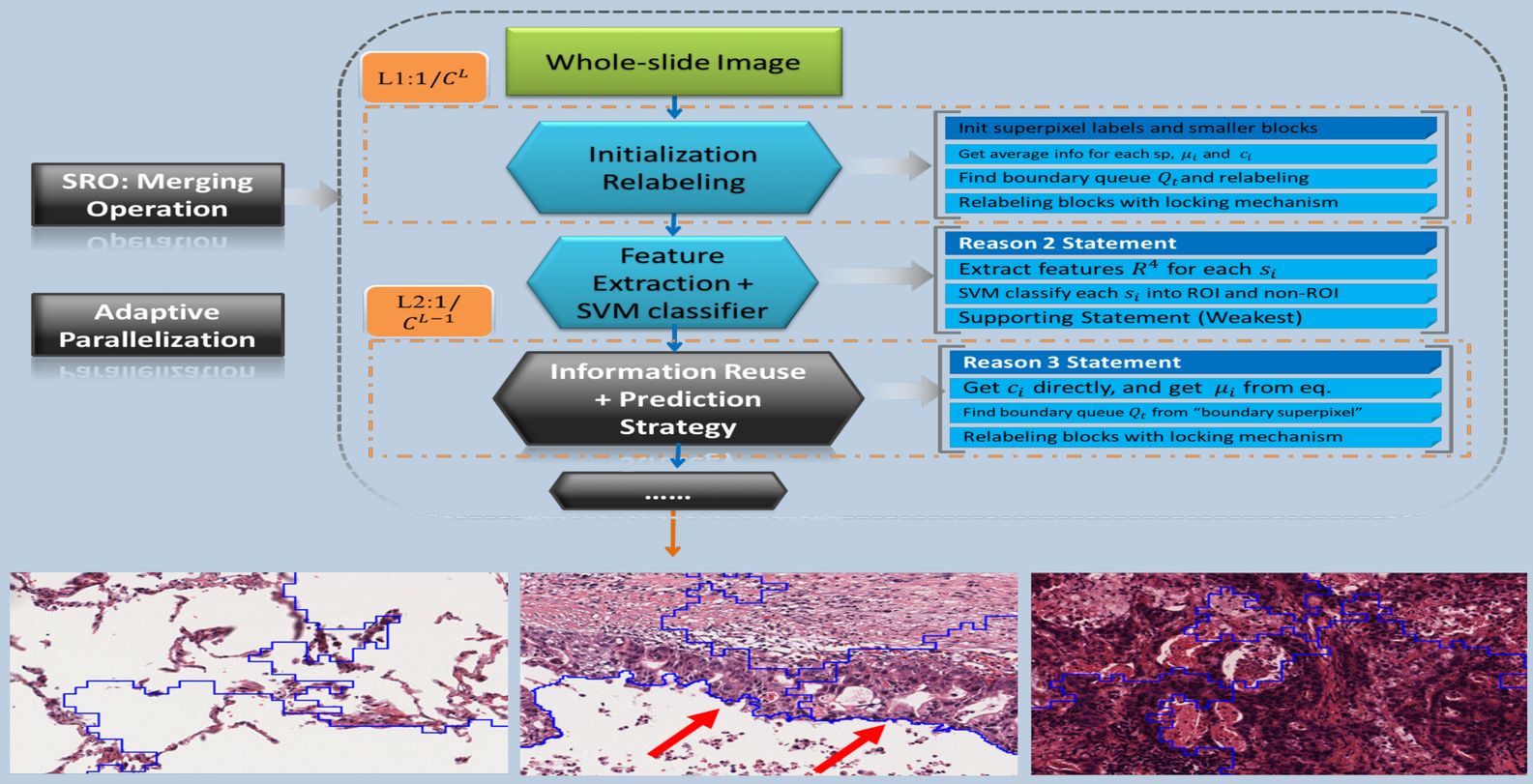

Figure 2: Coarse segment in non-ROI (left); Fine segment at the boundary (middle); coarse segment in ROI (right)

Superpixel Regularity Optimization (SRO) for Accuracy: it defines a merging operation. When the superpixel size after moving a block p_i from superpixel s_{p_i} is smaller than a lower threshold l, this operation searches another superpixel s_{p_m} with conditions: 1) It has the least gross energy after the merging; 2) the new size of s_{p_m} will not exceed the upper bound u to avoid potential problems.

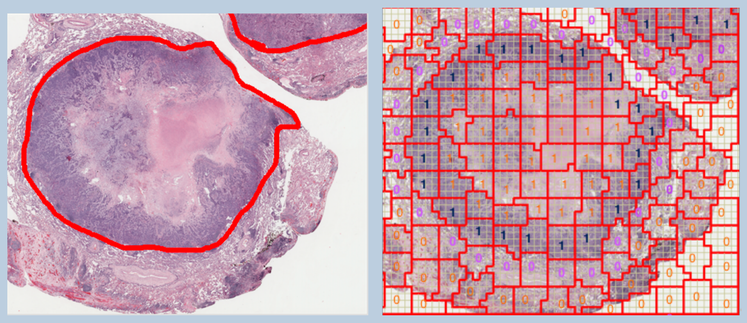

Figure 3: Ground truth; Prediction map (purple 0 and blue 1 represent the "boundary superpixel").

Prediction Strategy: The Eq. 2 is used to detect blocks that locating in the "boundary superpixel" (shown in Fig 3), which is a superpixel that is classified as ROI or non-ROI, and has at least a neighbor which is classified otherwise.

$$E_b(s_p, s_q) = \begin{cases} 1, & s_p \neq s_q, y_{s_p} \neq y_{s_q}, \\ 0, & otherwise. \end{cases} \quad (2)$$

Only put boundary blocks from the "boundary superpixels" in the queue at later stages.

Information Reuse:

$$\mu_{i+1} = C \times \mu_i - \frac{C-1}{2} \quad (3)$$

C is the compression ratio, i refers to the image level, and L is the total levels. Finally the cost of information computation is only $1/C^L$ of the original cost.

Parallelization Scheme:

$$startR = totalR * i/totalThreads \quad (4)$$
$$endR = totalR * (i+1)/totalThreads$$

Results

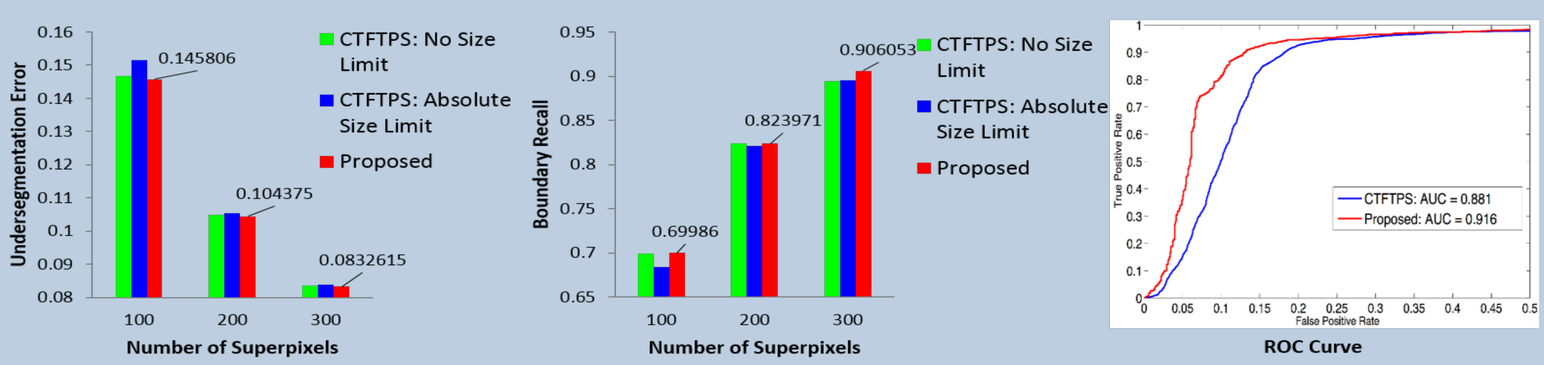

Figure 4: Comparison between CTFTPS with and without superpixel regularity optimization.

Table 1: Comparison of the average time cost.

Average Image Size	Grain	SLIC	CTFTPS	Multi-CTFTPS	**RAPID**
4712×5867	1×1	$12.90s$	$3.29s$	$3.54s$	**1.56s**
23561×29335	4×4	N/A	$27.26s$	$28.98s$	**6.96s**
	1×1	$809s$	$79.46s$	$66.46s$	**17.54s**

Table 2: Average time cost of Parallel RAPID with different number of threads.

Average Image Size	Grain	2T	4T	8T	14T	24T
4712×5867	1×1	$1.09s$	$0.63s$	$0.49s$	$0.42s$	$0.34s$
23561×29335	4×4	$4.39s$	$3.79s$	$1.95s$	$1.69s$	$1.54s$
	1×1	$12.13s$	$7.83s$	$6.29s$	$5.83s$	$5.20s$