Efficient and Scalable Image Segmentation Using Bag-of-Features and Stochastic Region Merging

R. S. Medeiros — Federal University of Rio Grande do Sul, RS, Brazil
Alexander Wong — University of Waterloo, ON, Canada
Jacob Scharcanski — Federal University of Rio Grande do Sul, RS, Brazil

Abstract

This work presents an efficient and scalable texture segmentation algorithm based on bag-of-features and stochastic region merging. The image is partitioned into blocks and processed independently to obtain regions, which are then merged to obtain the final segmentation. Experimental results shows the proposed method achieves an overall speed improvement of at least 4.5x and requires 6.5x less memory, while still improving segmentation accuracy for large images.

(a) STRS [1] (b) Proposed

Fig. 1: Comparison between the segmentations produced by STRS[1] and the proposed algorithm.

1 Introduction

A big challenge faced in image segmentation is the segmentation of large images, which is particularly relevant given that digital images nowadays can reach resolutions greater than 15 megapixels ($\approx 4800 \times 3200$). Current state-of-the-art methods typically deal with significantly smaller images ($\leq 480 \times 320$), and are computationally intractable for handling large images due to either computational time or memory limitations.

This paper presents a novel image segmentation method that allows for efficient and scalable texture segmentation that can handle large images. Inspired by the Multi-scale Stochastic Texture Representation and Segmentation (STRS) framework proposed in [2, 1], we introduce a novel algorithm extending significantly upon STRS in the following ways:
1) computing stochastic texture representation (STR) and bag-of-features (BoF) region histograms in a memory-efficient way,
2) determine texture boundaries by region merging technique in a more efficient manner, without lowering segmentation quality, and
3) partitioning the image into blocks for independent processing to obtain regions, which are then merged to obtain final segmentation result.

2 Proposed Methodology

The proposed segmentation approach extends upon the STRS framework [1], which has 4 main steps:
1) multi-scale image decomposition in N levels;
2) extraction of patch features of size $\mathcal{N}$ and projection into the STR space of dimension M;
3) regional description via BoF, using texton dictionaries of size K;
4) iterative stochastic region merging (SRM), starting with regularization term Q and color weights W_g.

In the proposed method, the input image I is converted to CIE LAB color space and is partitioned in a set of b blocks B_i, $1 \leq i \leq b$, of size $B_h \times B_v$. Each block is then processed using the same steps as the STRS framework to produce a partial segmentation S_i for this partition I. Note that the structures for the region size and histogram descriptors can be updated arithmetically after each merge, with a sum and a weighted average, respectively.

In this fashion, it is easy to see that intermediate segmentation $S_B = \{\bigcup_{i=1}^{n} S_i\}$ can be treated as an partial segmentation, therefore interpreted as an region graph G_B, and analyzed and merged by the same framework to produce a more refined segmentation. This approach, while robust, poses a practical problem: the histograms of regions in different blocks are derived from different texton dictionaries $\mathcal{D}_i$ and therefore they cannot be compared. To overcome this issue, we rebuild the texton dictionaries and recompute the descriptors for all regions in S_B. This texton dictionary rebuilding step could be expensive, specially in terms of memory, but as in natural images neighbor pixels tend to be very similar, and given the high resolution nature of large images, rebuilding the dictionaries at a lower resolution can be used to mitigate memory cost of analyzing the STR feature vectors. Therefore, the dictionaries $\mathcal{D}$ are built from a subset of the STR features uniformly sampled every κ pixels (in each direction). In Section 3 we show that even with a large κ will largely decrease the memory cost.

The final step of the proposed algorithm is to perform SRM on G_B to obtain the final segmentation result. The same initial Q and color weights W_g as the blocks segmentations are used during the SRM process. This step is repeated until the segmentation no longer changes, and at each iteration Q is lowered by half.

3 Experiments and Conclusions

Experiments were conducted using a set of 5 images collected from the Internet, with hand-annotated segmentation ground truth used for evaluation. The image resolution for the set of images vary from 1000×650 to 1000×800. The segmentation parameters are set to the optimal values reported in [1], i.e., $N = 2$, $\mathcal{N}_s = 5 \times 5$, $M = 20$, $K = 50$, $Q = 400$, $W_g = [2, 1, 1]$. The block size is set experimentally to $B_h = B_v = 350$, and sampling rate to $\kappa = 1/3$.

The results in Figure 1 visually show that the proposed approach have improved segmentation quality compared to STRS, where the same strong boundaries are captured properly while being less prone to over segmentation. The proposed method achieved PRI $= 0.8523$ and F-measure $F = 0.58@\{0.5582; 0.6522\}$, while the STRS achieved PRI $= 0.585$ and $F = 0.19@(0.10; 0.89)$.

In terms of computational cost, STRS took an average of $2401s \approx 40$min to compute each image, being that the STR features, histogram initialization and RM used $120s$, $344s$ and $1936s$ respectively. In contrast, the proposed method took and an average of $466 \approx 7,7$min to process an image, divided in $36s$, $108s$ and $355s$ for the same 3 stages (combining pixel-wise and the block parts). In terms of memory, the STRS used up to 1.82GB, when initializing the region histograms, while the proposed method used 286MB, in the same stages of each the block analysis and 8.33MB when recomputing them for regions in S_B. Both algorithms were implemented in Matlab, using C++ MEX files for the region merging stage. All experiments where conducted in a computer with a Intel Core i7 2.5GHz and 16GB RAM, using Matlab profiler to analyze the time and memory performance.

We can conclude that the proposed method achieve similar segmentation quality to STRS, but with a speed-up of 6.8x in the RM step, a overall speed-up of 5.1x for the whole process, and requires 6.5x less memory. This shows that the proposed method hold a lot of promise for dealing with large images, and future work involves automatically selecting optimal block sizes as well as parallel processing.

References

[1] R. Medeiros, J. Scharcanski, and A. Wong. Image segmentation via multi-scale stochastic regional texture appearance models. *Computer Vision and Image Understanding*, (2015).

[2] A. Wong, J. Scharcanski, and P. Fieguth. Automatic skin lesion segmentation via iterative stochastic region merging. *IEEE Trans. Information Technology in Biomedicine*, (2011).