



# jSLIC: superpixels in ImageJ

Jiří Borovec & Jan Kybic,

Faculty of Electrical Engineering, Czech Technical University in Prague

jiri.borovec@fel.cvut.cz

<http://cmp.felk.cvut.cz/~borovji3/>



## Abstract

We present the implementation and improvements of the superpixel clustering algorithm - SLIC (Simple Linear Iterative Clustering). The implementation is called jSLIC. Our main contributions are: a significant speed-up of the original clustering method, a new post-processing step (after clustering) which now gives more reliable superpixels - the newly established segments are more homogeneous, and transforming the compactness parameter such that the value is superpixel size independent. The improvements of speed and quality are shown on real images. We implemented the new jSLIC in Java and made the source code publicly available. Also we created a plugin for ImageJ/Fiji which is a commonly used research and development platform in biology and medical imaging.

## SLIC overview

SLIC [1] is an algorithm for calculating superpixels based on k-means. Its key ideas are:

1. the distance between pixels

$$D = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \approx d_c^2 + (d_s^2 \cdot f)$$

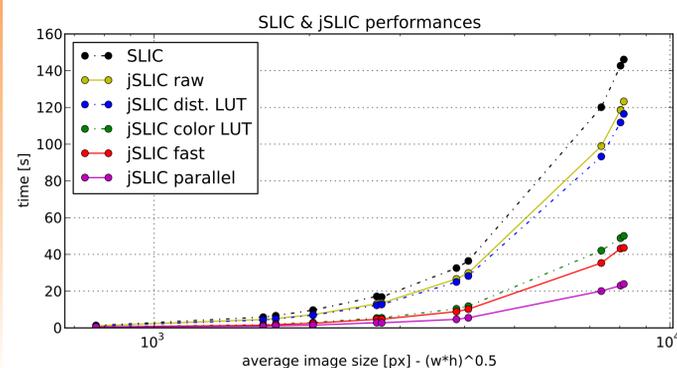
combines Euclidean colour distance  $d_c$  (using the CIELAB colour space), Euclidean pixel distance  $d_s$  and regularisation  $f = \left(\frac{m}{S}\right)^2$

2. the search space is reduced by limiting to a spatial region  $2S \times 2S$ ,

proportional to the desired superpixel size  $S$ .

## Speed improvements

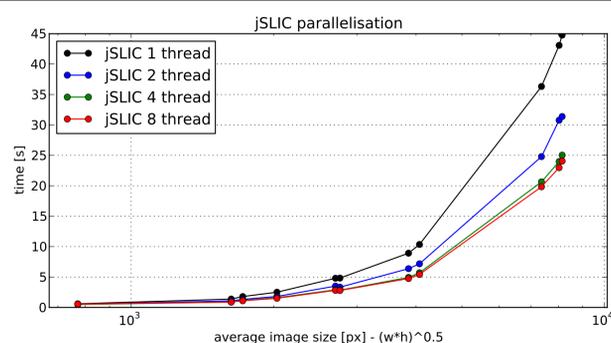
1. Java-based implementation. (*speed-up 1.27 times*)
2. Precomputed spatial distance in regular grid. (*speed-up 1.06 times*)
3. Colour conversion RGB to CIE Lab using LUT. (*speed-up 2.48 times*)
4. Parallelisation. (*for 4 threads - speed-up 4.68 times*)



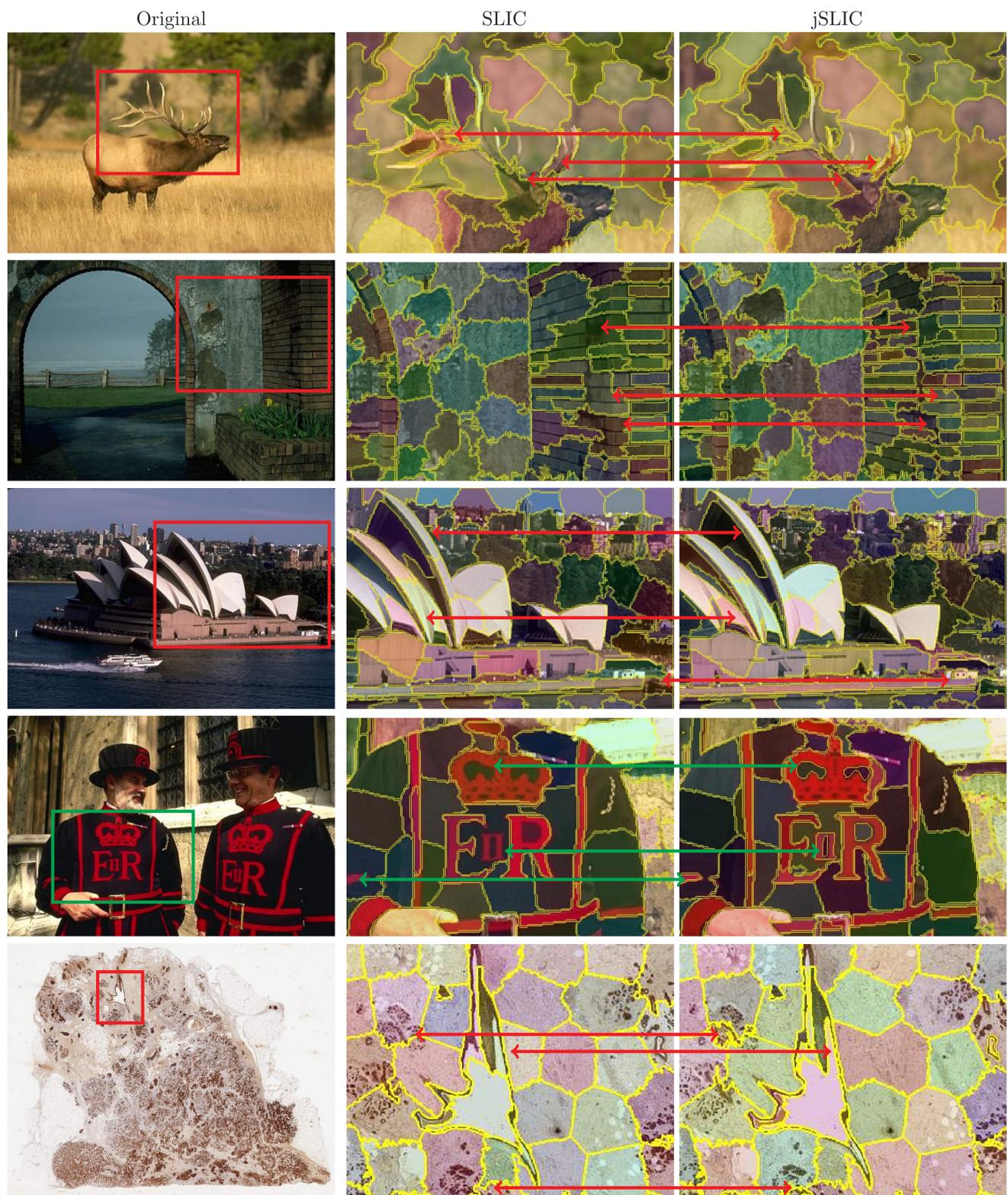
## Regularisation

SLIC contains a regularisation  $f = \left(\frac{m}{S}\right)^2$  which influences the compactness of clustered superpixels, where  $m \in [0, \infty)$ . We propose instead  $m$  to use a parameter  $r$  so  $f = (S \cdot r)^2$  and  $r \in (0, 1)$  meaning 0 as the minimal and 1 as the maximal compactness. Default regularisation is  $r = 0.2$ .

## Parallelisation



## Comparison between SLIC & jSLIC superpixels



Berkeley Segmentation Dataset [2] and some stained histological images.

## Post-processing

To eliminate small regions we take into account their size and colour similarity of neighbouring regions  $\Omega_i$  (connected components) to each region of index  $i$ .

1. Compute mean colours  $c_i^{lab}$  and relative area  $s_i$  for all components  $i$ .
2. Computing the similarity  $d_i(j)$  between each  $i$  and  $j \in \Omega_i$

$$d_i(j) = \frac{\|c_i^{lab} - c_j^{lab}\|_2}{s_j}$$

3. Find the most similar component

$$i^* = \arg \min_{j \in \Omega_i} (d_i(j))$$

4. Regions are merged  $i \leftarrow i^*$  if both  $i, i^*$  are similar enough

$$\left(\frac{s_i}{4}\right)^2 \cdot (1 + d_i(i^*)) < \varepsilon$$

Default threshold is  $\varepsilon = 0.25$ .

## Conclusion

- jSLIC has better performance than the original SLIC.
- Java-based open source implementation of jSLIC and a plugin jSLIC<sup>a</sup> superpixel clustering for Fiji.
- Regularisation parameter invariant to superpixel size in range  $r \in (0, 1)$
- New post-processing step gives more relevant superpixels shapes.

<sup>a</sup>[http://fiji.sc/CMP-BIA\\_tools](http://fiji.sc/CMP-BIA_tools)

## References

- [1] R. Achanta and A. Shaji. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *Pattern Analysis and Machine Intelligence, IEEE*, 34(11):2274 – 2282, 2012.
- [2] D. Martin and C. Fowlkes. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision, IEEE*, number July, 2001.