21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Measuring and Evaluating the Compactness of Superpixels

Alexander Schick* Mika Fischer† Rainer Stiefelhagen*,†
*Fraunhofer IOSB †Karlsruhe Institute of Technology
alexander.schick@iosb.fraunhofer.de {mika.fischer, rainer.stiefelhagen}@kit.edu

Abstract

Superpixel segmentation has become a popular preprocessing step in computer vision with a great variety of existing algorithms. Almost all algorithms claim to compute compact superpixels, but no one showed how to measure compactness and no one investigated the implications. In this paper, we propose a novel metric to measure superpixel compactness. With this metric, we show that there is a trade-off between compactness and boundary recall. In addition, we propose an algorithm that allows to precicely control this trade-off and that outperforms the current state-of-the-art. As a demonstration, we show the importance of considering compactness with the help of an example application.

1. Introduction

The term superpixel was introduced by Ren and Malik [13] and describes the oversegmentation of an image into homogeneous regions that align well with object boundaries. This allows to represent an image with only a couple of hundred segments that function as atomic building blocks instead of tens of thousands of pixels.

There are many different approaches to segment an image into superpixels, for example graph-based approaches using normalized cuts [5, 14] or graph cuts [15]. Other approaches utilize geometric flows [3], geodesic distances [17], or pseudo-boolean optimization [18]. Some algorithms also guarantee that the segmentation conforms to a lattice structure [7, 8]. The currently best results are reported by an entropy rate based approach [4]. One approach that excells with its ease of use and efficiency is the k-means based SLIC algorithm [1].

Does shape matter? Compactness means that each superpixel has a regular shape and size with smooth boundaries. Many authors claim compactness for their superpixels or agree that it is a desirable property [1, 3, 4, 7, 15, 17, 18], but no one ever measured compactness. With this work, we are the first to measure and investigate compactness in the context of superpixels. Besides aesthetical considerations, compactness is also appealing from a practical and theoretical point of view. Compact superpixels better capture spatially coherent information and it is easier to extract information from their boundaries. Non-compact superpixels with irregular shapes, on the other hand, can be compared to overfitting in machine learning: too much data (i.e. boundary pixels) is used to represent the essential information (i.e. object boundaries).

In this paper, we propose a novel metric that measures compactness and investigate its implications. We also introduce an improvement of SLIC [1] to allow for a precise and transparent compactness control.

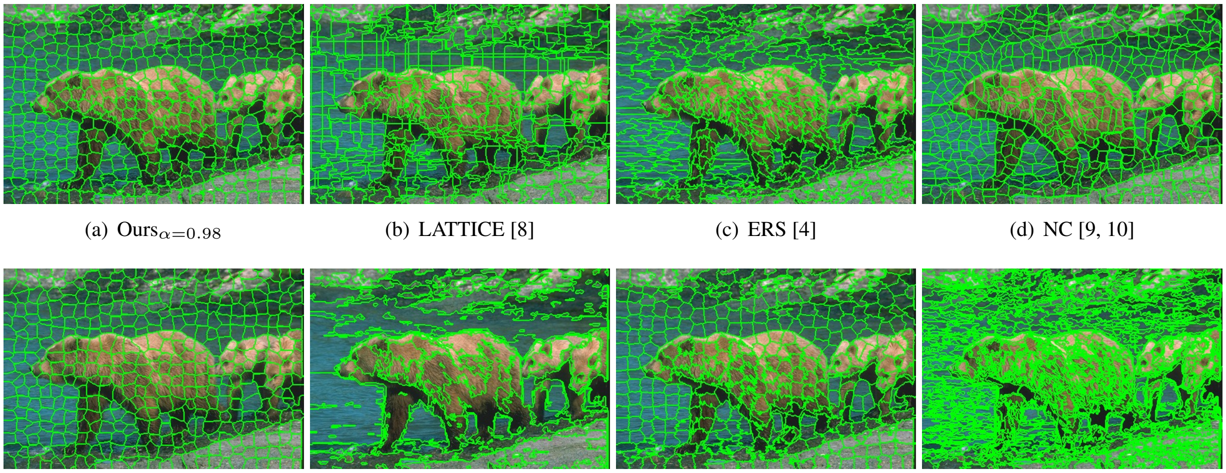
2. Superpixel compactness

Measuring the compactness of a shape is a well-known task in mathematics and one common measure is the isoperimetric quotient. It is related to the isoperimetric problem: finding the two-dimensional shape that has the largest possible area for a given boundary length [12]. The solution to this problem is the circle which is the most compact shape.

The isoperimetric quotient relates the area of a given shape to the area of a circle that has the same perimeter as this shape. It is 1 for a circle and decreases the less compact the shape becomes. Let A_S be the area and L_S be the perimeter of a shape, e.g. of a superpixel S. The circle with the same perimeter as the superpixel has the radius $r = \frac{L_S}{2\pi}$. Let A_C be the area of a circle with radius r. Then the isoperimetric quotient is

$$Q_S = \frac{A_S}{A_C} = \frac{4\pi A_S}{L_S^2}. \tag{1}$$

Based on the isoperimetric quotient, we propose a metric to measure the compactness (CO) of a superpixel segmentation. For a given segmentation, we compute the sum over the isoperimetric quotients of each superpixel normalized by the fraction of the superpixels' size $|S|$ compared to the whole image. Let $\mathfrak{S}$ be the set of



(a) $\text{Ours}_{\alpha=0.98}$ (b) LATTICE [8] (c) ERS [4] (d) NC [9, 10]

(e) TP [3] (f) FH [2] (g) SLIC [1] (h) $\text{Ours}_{\alpha=0.0}$

Figure 1. Visual comparison of the evaluated superpixel algorithms. Note that (h) is only used to demonstrate the importance of compactness. These images are best viewed in color.

all superpixels of a segmentation of the image I. The compactness of the segmentation then is

$$CO = \sum_{S \in \mathfrak{S}} Q_S \cdot \frac{|S|}{|I|}. \tag{2}$$

The compactness is mostly influenced by the regularity of the boundaries and orientation changes in the boundaries have a severe impact on the overall compactness. Therefore, compactness is a desirable property for all applications that require smooth and regular boundaries.

3. Direct compactness control

We propose an improved version of SLIC that integrates a transparent compactness control. At its core, SLIC is based on k-means clustering with distance measure D_S consisting of a distance in Lab color space (d_{lab}) as well as a weighted Euclidean distance (d_{xy}). Please see [1] for all details. It is an efficient algorithm with interesting properties, but it has one major drawback: the superpixels can be ripped apart during the k-means clustering and, therefore, a postprocessing step is required which bypasses the distance measure and, thus, the compactness control.

We improve this by applying the distance measure D_S not to all image pixels, but only to pixels belonging to superpixel boundaries. Thereby, only boundary pixels are reassigned and we achieve a smooth iterative boundary evolution guided by the distance measure D_S which guarantees that the superpixels remain intact. We also improve the balancing of the two components of D_S by introducing a compactness parameter α (instead of just weighting the Euclidean distance). As we will see later, this results in a very transparent compactness control. Let r be the inital side length of the superpixels, then

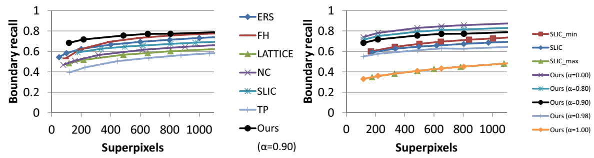
$$D_S = (1-\alpha) \cdot d_{rgb} + \alpha \cdot \frac{d_{xy}}{r}. \tag{3}$$

In contrast to SLIC, we achieved slightly better results for RGB images. Also, the boundary evolution is more sensitive to strong image gradients. This effect can be reduced by low-pass filtering the images.

4. Evaluation

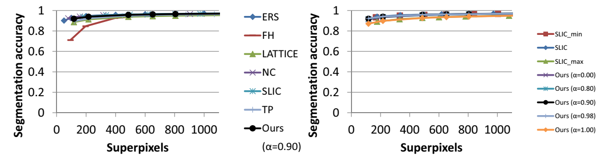
We compared our algorithm to five superpixel algorithms that are all based on different principles: the normalized cuts segmentation from [9, 10] (NC), SLIC [1], TurboPixels [3] (TP), entropy rate superpixels [4] (ERS), and Superpixel Lattices [8] (LATTICE). For SLIC, we also evaluated the weighting range (SLIC_min, SLIC_max). We also compared our algorithm with Felzenszwalb and Huttenlocher [2] (FH). FH does not aim to compute compact superpixels, but its high accuracy makes it interesting for comparison as a baseline algorithm. We used code made available by the authors and chose parameters as proposed by them.

We evaluated all algorithms on the full Berkeley segmentation dataset (BSDS) [6] consisting of 300 natural images and 1,633 human annotated groundtruth images.



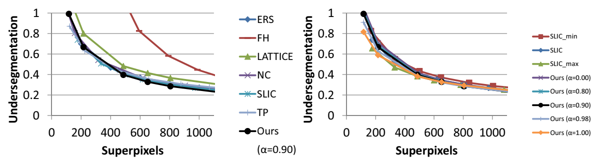
(a) All algorithms (b) Varying compactness

Figure 2. Boundary recall



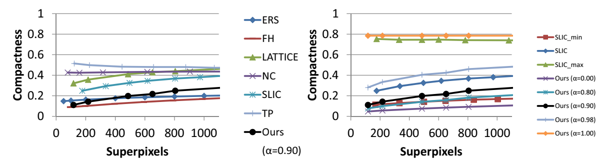
(a) All algorithms (b) Varying compactness

Figure 4. Achievable accuracy



(a) All algorithms (b) Varying compactness

Figure 3. Undersegmentation error



(a) All algorithms (b) Varying compactness

Figure 5. Compactness

We applied the proposed compactness metric CO (Section 2) as well as the commonly used metrics:

Boundary recall (BR) measures the fraction of groundtruth boundaries that overlap with the segmentation boundaries. It is well suited for superpixel segmentations because it does not penalize oversegmentation and is the most commonly used metric for superpixels.

Undersegmentation error (UE) was introduced by Levinshtein et al. [3]. Figuratively speaking, it measures the "bleeding out" [3] of superpixels. The metric was also adopted by [1, 4, 15, 16, 17].

Achievable segmentation accuracy (AA) gives the highest accuracy possible for segmenting objects using superpixels as building blocks [11]. It was also used by [4].

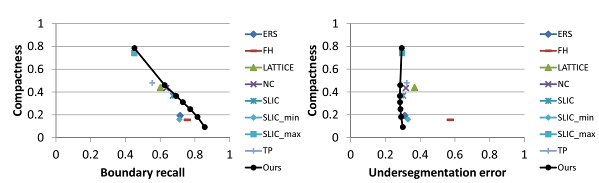
5. Results

Figures 2, 3, 4, and 5 show results for all four metrics, both compared to (a) the current state-of-the-art as well as (b) for different compactness parameters. Please note that our compactness parameter $\alpha \in [0, 1]$ affects the whole distance measure D_S and operates on a more well-defined intervall than the Euclidean distance weight in SLIC ($m \in [0, \text{MAXINT}]$).

We will first discuss the results on the existing metrics before explaining the importance of compactness. The proposed algorithm (with $\alpha = 0.9$) outperforms the current state-of-the-art for BR and achieves comparable or better results for UE and AA. A comparison of the full parameter range (b) also shows that the proposed algorithm outperforms SLIC on all metrics.

We will now motivate the importance of a compactness metric. When taking the extreme case of a non-compact segmentation ($\alpha = 0$, Figure 1h), it becomes apparent that this is not desirable at all. Nevertheless, it achieves the highest BR with good UE and AA results. The reason is that the superpixel boundaries are "overfitted" to capture the finest, but not necessarily meaningful, details while ignoring the representation. This is not sufficiently penalized by any of the current metrics and no metric measures the shape of superpixels. This is our motivation for the compactness metric.

When comparing Figures 2 and 5, there is a negative correlation between BR and CO meaning that algorithms with higher BR have, in general, a lower CO. This is true for all algorithms. The trade-off between BR and CO is due to the fact that for a higher BR the superpixel boundaries have to adjust more to the image content which makes them less compact. This is also in accordance with the visual observation (see Figure 1),



(a) CO over BR (b) CO over UE

Figure 6. Combined metrics (800 SP)

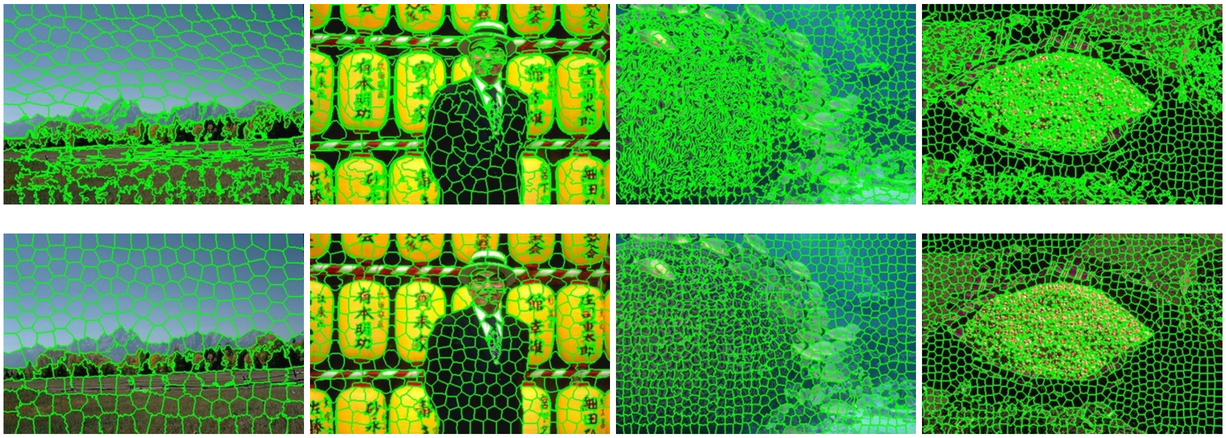


Figure 7. Additional qualitative results of the proposed algorithm. The rows show results for the recommended compactness parameters $\alpha = 0.9$ and $\alpha = 0.98$. The numbers of superpixels from left to right are: 216, 481, 805, and 1107. These images are best viewed in color.

meaning that more irregular and wiggly superpixels are less compact and vice versa.

To reflect the trade-off between shape and function, we recommend to use a new graph: CO-over-BR (Figure 6a). It allows to compare algorithms with respect to this trade-off to chose the best algorithm for a specific application. Similar graphs are also possible for UE (Figure 6b) and AA. However, because the UE and AA metrics show similar results for all algorithms, the combined graphs might not be as meaningful. Currently, the best trade-off is achieved by the proposed algorithm.

Given these results, we recommend $\alpha = 0.9$ for very accurate and $\alpha = 0.98$ for very compact superpixels. Figure 7 shows qualitative results for these settings for different superpixel resolutions.

6. Application

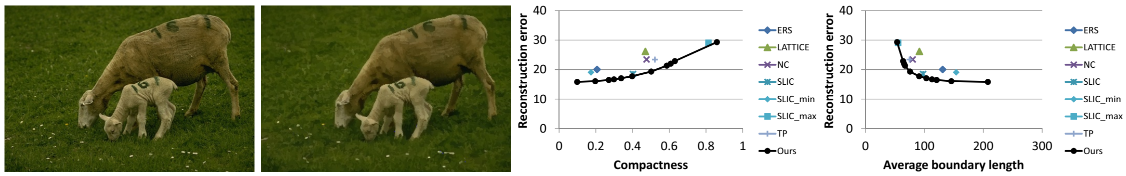
Superpixels are generally used as atomic primitives. Depending on the application, different features are extracted from them. While for some the shape and, therefore, compactness is of no concern (e.g. when extracting color histograms), others rely on spatial coherence or use the boundaries directly. We will now give one example application and point to other applications where compactness is important.

Image representation (or compression) with superpixels has already been show in [3, 17], but with a focus on visual quality. However, when considering image representation, not only the reconstruction quality, but also the size of such a representation is important.

Let an image be represented by a set of superpixels. The superpixel colors can be approximated by a polynomial of nth order [3] with constant encoding size. The encoding size of the shape, however, depends on its regularity. The more compact and regular the shape, the easier it is to encode. In this example application, the encoding is directly represented by boundary pixels leading to the encoding size being a (monotonic) function of the boundary length. While there are certainly more sophisticated encodings, we expect that they will also benefit from regularities in the boundaries.

Figures 8a and 8b show an example image and its reconstruction. Figure 8c shows the average reconstruction error per pixel over all images of the BSDS for a superpixel resolution of 800 with a quadratic polynomial for color representation. (FH is omitted because it is not truly a superpixel algorithm and, thus, achieves very bad results for this task.) As expected, the reconstruction error increases with compactness because of the trade-off with its accuracy. Figure 8d makes this even more apparent. Depending on the compactness, the boundary length changes, but the trade-off is not linear. In fact, with a doubling of the boundary length from 100 to 200 pixels, there is almost no change of the reconstruction error. This illustrates the ”overfitting” argument which says that a more accurate segmentation does not necessarily imply a better overall performance. By controlling compactness, the (application-dependent) optimal trade-off between error and encoding size can be chosen. Here, no superpixel algorithm achieves a better trade-off than the proposed one.

Besides this example application, we expect similar results for applications that analyze the shape of super-



(a) Original (b) Reconstruction (c) Error over compactness (d) Error over boundary length

Figure 8. Image representation with superpixels.

pixels and that rely on smooth boundaries (e.g. [9, 10]). We expect that algorithms that offer a direct compactness control are likely to achieve better results.

7. Conclusion

Compactness is a desirable property of superpixels [1, 3, 4, 7, 15, 17, 18]. We introduced the first metric measuring superpixel compactness and showed its implications both in comparison with existing metrics as well as for an example application. Further, we presented an algorithm that achieves best or comparable results on all metrics compared to the current state-of-the-art and that offers the best trade-off between recall and compactness. This algorithm also allows to transparently control compactness and, thereby, this trade-off, thus making it a good choice for a wide range of applications.

Acknowledgments

This work was partially supported by the FhG Internal Programs under Grant No. 692026 and by OSEO, French State agency for innovation, as part of the Quaero Programme.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels. *EPFL Technical Report 149300, June*, 2010.

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[3] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. TurboPixels: Fast Superpixels Using Geometric Flows. *Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009.

[4] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy Rate Superpixel Segmentation. In *Proc. Computer Vision and Pattern Recognition*, pages 2097–2104, 2011.

[5] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[6] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *Proc. International Conference on Computer Vision*, pages 416–423, 2001.

[7] A. P. Moore, S. J. D. Prince, and J. Warrell. Lattice Cut - Constructing superpixels using layer constraints. In *Proc. Computer Vision and Pattern Recognition*, pages 2117–2124, 2010.

[8] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel Lattices. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[9] G. Mori. Guiding Model Search Using Segmentation. In *Proc. International Conference on Computer Vision*, pages 1417–1423, 2005.

[10] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering Human Body Configurations: Combining Segmentation and Recognition. In *Proc. Computer Vision and Pattern Recognition*, pages 326–333, 2004.

[11] S. Nowozin, P. V. Gehler, and C. H. Lampert. On Parameter Learning in CRF-based Approaches to Object Class Image Segmentation. In *Proc. European Conference on Computer Vision*, pages 98–111, 2010.

[12] G. Polya. *Mathematics and Plausible Reasoning, Volume 1: Induction and Analogy in Mathematics*. 1990.

[13] X. Ren and J. Malik. Learning a Classification Model for Segmentation. In *Proc. International Conference on Computer Vision*, pages 10–17, 2003.

[14] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[15] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and Supervoxels in an Energy Optimization Framework. In *Proc. European Conference on Computer Vision*, pages 211–224, 2010.

[16] S. Xiang, C. Pan, F. Nie, and C. Zhang. TurboPixel Segmentation Using Eigen-Images. *Transactions on Image Processing*, 19(11):3024–3034, 2010.

[17] G. Zeng, P. Wang, J. Wang, R. Gan, and H. Zha. Structure-sensitive Superpixels via Geodesic Distance. In *Proc. International Conference on Computer Vision*, pages 447–454, 2011.

[18] Y. Zhang, R. Hartley, J. Mashford, and S. Burn. Superpixels via Pseudo-Boolean Optimization. In *Proc. International Conference on Computer Vision*, pages 1387–1394, 2011.