



HAL
open science

ASARI: a new adaptive oversegmentation method

Bérengère Mathieu, Alain Crouzil, Jean-Baptiste Puel

► **To cite this version:**

Bérengère Mathieu, Alain Crouzil, Jean-Baptiste Puel. ASARI: a new adaptive oversegmentation method. Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2017), Jun 2017, Faro, Portugal. pp. 194-202. hal-01913673

HAL Id: hal-01913673

<https://hal.science/hal-01913673>

Submitted on 6 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:

<http://oatao.univ-toulouse.fr/190945>

Official URL: https://link.springer.com/chapter/10.1007/978-3-319-58838-4_22

DOI : https://doi.org/10.1007/978-3-319-58838-4_22

To cite this version: Mathieu, Bérengère and Crouzil, Alain and Puel, Jean-Baptiste *ASARI: a new adaptive oversegmentation method*. (2017) In: Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2017), 20 June 2017 - 23 June 2017 (Faro, Portugal).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

ASARI: A New Adaptive Oversegmentation Method

Bérengère Mathieu^{1(✉)}, Alain Crouzil¹, and Jean Baptiste Puel²

¹ Université de Toulouse, UPS, IRIT, Toulouse, France
{berengere.mathieu,alain.crouzil}@irit.fr

² Université de Toulouse, ENFA, IRIT, Toulouse, France
jean-baptiste.puel@irit.fr

Abstract. Using superpixels instead of pixels has become a popular pre-processing step in computer vision. However, there are few adaptive methods able to automatically find the best compromise between boundary adherence and superpixel number. Moreover, no algorithm producing color and texture homogeneous superpixels keeps competitive execution time. In this article we suggest a new graph-based region merging method, called Adaptive Superpixel Algorithm with Rich Information (ASARI) to solve these two difficulties. We will show that ASARI achieves results similar to the state-of-the-art methods on the existing benchmarks and outperforms these methods when dealing with big images.

Keywords: Graph-based oversegmentation · Superpixels · Local Ternary Patterns

1 Introduction

The idea and definition of superpixel is given for the first time by Ren *et al.*, in [9]. The two authors describe a segmentation method including an *oversegmentation* pre-processing grouping pixels into small homogeneous and regular regions called *superpixels*. Currently, oversegmentation is an active research field, with steady publication of new methods [1, 2, 6]. Some examples of oversegmentation results are given in Fig. 1. A good oversegmentation method must satisfy five properties: **validity** (an oversegmentation must be an image partition into connected components), **boundary adherence** (superpixels must not overlap different objects of the image), **conciseness** (an oversegmentation must give as few superpixels as possible), **simplicity** (the number of neighbors of each superpixel must be as small as possible, to avoid a complex adjacency graph), **efficiency** (an oversegmentation algorithm must have an execution time as low as possible). We call **adaptivity**, the fact that an algorithm is able to find the

The work of Bérengère Mathieu was partially supported by ANR-11-LABX-0040-CIMI within the program ANR-11-IDEX-0002-02.

best compromise between these two properties, reducing the number of superpixels in wide homogeneous regions and increasing it to segment correctly thin details.

Oversegmentation algorithms may be region-based [1, 5], when clustering pixels, or boundary-based [2, 3], when focusing on edge evidence. The review of Stutz [12] shows that five methods achieve similar results and outperform other algorithms: the Felzenszwalb *et al.* algorithm (FZ) [3], Quick Shift (QS) [14], Entropy Rate Superpixels (ERS) [5], Simple Linear Iterative Clustering (SLIC) [1] and Contour Relaxed Superpixel [2]. The FZ [3] and ERS [5] algorithms use a graph-based representation of the image $G = \langle V, E \rangle$, where V is the set of elements to be grouped (*i.e.* the pixels) and E , the set of edges linking pairs of neighboring elements. Each edge is weighted using a dissimilarity measure. FZ uses a predicate checking that the dissimilarity between elements along the boundary of two components is greater than the dissimilarity between neighboring elements within each of the two components, to produce a partition of G into K connected components corresponding to superpixels. ERS is a greedy algorithm selecting a subset $A \subset E$ and removing these edges. The result is a partition of G , which maximizes an entropy rate. The QS method [14] is a modification of the medoid-shift algorithm to efficiently find modes of a Parzen density estimate P . Color and location of each pixel are used as feature vectors that are clustered by linking each vector to its nearest neighbor which increases P . The SLIC method [1] is an adaptation of the k-means algorithm. Starting from an image oversegmentation into a regular grid, the average color and location features of each superpixel are computed. Then, each pixel is re-assigned to the most similar superpixel and the average features of superpixels are re-computed. The CRS algorithm [2] finds a partition S into superpixels, which has a high likelihood of having generated the observed image. Starting from an initial segmentation into rectangular superpixels, CRS maximizes the probability function by reallocating some boundary pixels to another superpixel. In our evaluation, we add to these five state-of-the-art methods, WP, a watershed transformation based algorithm recently suggested by Machairas *et al.* [6].

The main contribution of this article is a graph-based oversegmentation method (Sect. 2), called Adaptive Superpixel Algorithm with Rich Information (ASARI). As its name suggest, ASARI is able to adjust the size and the number of superpixels to fit the image complexity, partitioning uniform objects (like the sky) in a few large superpixels and producing a lot of thin superpixels to match with small details. In addition, ASARI groups pixels using both color and texture information, without dramatically increasing the computation time. The only oversegmentation algorithm using texture has been proposed by Ren *et al.* in [9]. Unfortunately, its high execution time discouraged its usage in practical applications. Using Stutz benchmark [12], we show in Sect. 3 that ASARI is competitive with the state-of-the-art methods. However, the benefit of this new algorithm is more significant on a new dataset, that we recently made available: Heterogeneous Size Image Dataset (HSID). Contrary to other oversegmentation evaluation datasets, HSID contains photographs having extremely different sizes

(from some thousand to several millions of pixels) and a majority of big images. This leads us to an interesting application of ASARI, as a preprocessing step in the Superpixel Classification based Interactive Segmentation (SCIS) method [8].

2 ASARI: A Region Merging Algorithm

ASARI is a region-merging algorithm, following the approach of Salembier *et al.* [10], which is still the base of successful works in image analysis [4,15]. Let $G_0 = \langle V_0, E_0 \rangle$ be a graph related to an image I , where V_0 is the set of vertices and E_0 the set of edges connecting them. Each vertex v_i corresponds to a pixel or a small group of pixels and there is an edge (v_i, v_j) between each pair of neighboring pixels or regions. A region merging algorithm with G_0 as input, produces a graph $G_K = \langle V_K, E_K \rangle$, where V_K is a partition of V_0 into connected components and E_K is a subset of E_0 . More concretely, some vertices of G_0 are merged and edges linking them are removed. Salembier *et al.* [10] show that such an algorithm can be defined by a merging order (the order in which the edges are processed), a merging criterion (how to decide if an edge must be removed) and a region model (how to represent the union of two vertices).

Merging order - Merging order in the case of an oversegmentation algorithm must ensure that visually similar regions will be merged, while maintaining regions with reasonably similar areas. We associate to each vertex v_i a variable Δ_i referring to the number of times v_i has been previously selected. At the beginning of each iteration, the algorithm selects the vertex with the smallest Δ_i (when several vertices have the same Δ_i value, one is arbitrary chosen). ASARI merging order keeps among the set of edges linking v_i to its neighbors, the one maximizing a similarity measure f_{sim} . Whether v_i and its most similar neighbor are merged or not¹, Δ_i is incremented.

Similarity measure - When designing ASARI, one of our objectives was to propose a similarity measure using both color and texture features and keeping reasonable execution times (a few seconds on a desktop computer, for images of size 3000×4000). So, we chose the Local Ternary Patterns (LTP) [13] largely for their rapidity. The LTP of a pixel p is a pair of identifiers, (LTP_N, LTP_P) , corresponding to gray level variations of its 8-neighborhood. Let $T = \{g_0, \dots, g_7\}$ be the set the gray levels of the 8 neighbors of p . The identifiers are given by:

$$LTP_N(p) = \sum_{n=0}^7 2^n \delta(p - g_n) \text{ and } LTP_P(p) = \sum_{n=0}^7 2^n \delta(g_n - p) \quad (1)$$

where δ is a threshold function, returning 1 if its input is greater than a threshold ω_{LTP} , 0 otherwise. In the binary representation of LTP_N and LTP_P , digits equal to 0 correspond to neighbors with gray levels similar to the center gray level. So, pixels with a pair of LTP identifiers equals to $(0,0)$, have a neighborhood with similar gray levels. The ratio between the number of this kind of pixels

¹ The decision to merge or not v_i and its most similar neighbor is related to the merging criterion.

and the total number of pixels in a region gives the probability for the region to be textured or not. If this probability is lower than a threshold ω_{tex} , the region is textured. We assume that a similarity measure must give a low value when comparing a textured region to an untextured region. Let $f_c(i, j)$ be a color-based similarity measure between regions v_i and v_j , $f_t(i, j)$ be a texture-based similarity measure, and h be a binary function, returning 1 for a textured region, 0 otherwise. The similarity function is the following:

$$f_{sim}(i, j) = \begin{cases} f_c(i, j) & \text{if } h(i) + h(j) = 2 \\ \frac{f_t(i, j) + f_c(i, j)}{2} & \text{if } h(i) + h(j) = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Let c_i be a vector containing the normalized average color of v_i :

$$f_c(i, j) = \exp\left(-\frac{1}{N_c} \|c_i - c_j\|^2\right) \quad (3)$$

where N_c is the length of c_i . Let t_i be the concatenation of the normalized histograms of LTP_N and of LTP_P , for the region v_i . We have:

$$f_t(i, j) = \exp\left(-\frac{1}{N_t} \sum \frac{(t_i^k - t_j^k)^2}{t_i^k + t_j^k}\right) \quad (4)$$

where N_t is the number of strictly positive bins in t_i or t_j and the superscript k denotes the k -th element of the vector. Notice that f_t is derived from the χ^2 measure. Distinguishing between textured and untextured regions reduces the execution time of ASARI, the computation time for f_c being significantly lower than for f_t .

Merging criterion - A merging criterion $g(i, j)$ must ensure both visual consistency and regularity of the region and can be expressed by the following boolean function:

$$g(i, j) = g_{sim}(i, j) \wedge g_{reg}(i, j) \quad (5)$$

where g_{sim} checks that similarity between the two regions is greater than a threshold ω_{sim} and g_{reg} verifies that the new region size is not disproportionate to the size of the other regions, using a threshold ω_{reg} . So we have:

$$g_{sim}(i, j) = f_{sim}(i, j) > \omega_{sim} \text{ and } g_{reg}(i, j) = (\phi(i) + \phi(j)) < \frac{\omega_{reg}}{M} \sum_{n=0}^M \phi(n) \quad (6)$$

where $\phi(i)$ is the number of pixels belonging to v_i and M the number of regions. The adaptive nature of ASARI results from g_{sim} function, which stops merging in image parts containing small distinct details. The function g_{reg} prevents a segmentation-like behavior and ensures that superpixel sizes remain much smaller than object areas. ASARI stops when any edge cannot be removed any more or when all vertices have been selected 10 times².

² Our experiments show that this number of iterations is sufficient to provide good results.

Initialization - To produce V_0 , we use the SLIC algorithm [1], which has the two benefits of drastically reducing the number of initial regions and simplifying the use of the texture similarity measure, which requires at least one hundred of pixels to give a significant histogram. The drawback of this approach is to introduce two more parameters, ω_{min} and ω_{comp} , which must be designed to produce very small regions. Parameter ω_{min} gives an average region size and parameter ω_{comp} is the SLIC compactness parameter [1].

Parametrization - ASARI has 6 parameters: ω_{LTP} , ω_{tex} , ω_{sim} , ω_{reg} , ω_{min} and ω_{comp} . As these parameters impact different aspects of ASARI, sometime related to contradictory properties (like conciseness and boundary adherence), it is not simple to learn them at once. We processed in three steps, first with parameters linked to untextured region detection (ω_{LTP} , ω_{tex}), then setting SLIC parameters (ω_{min} , ω_{comp}) and finally focusing on thresholds for the merging criterion (ω_{sim} , ω_{reg}). Parameters $\omega_{LTP} = 19$ and $\omega_{tex} = 0.8$ have been learned by solving an intermediary classification problem of small patches extracted both of Wikimedia Common³ images and Brodatz texture images⁴, and then, classified into textured or untextured patches. The value of $\omega_{comp} = 10$ has been set following previous evaluation of SLIC [1]. The parameter ω_{min} is the only one which must be set by the user. In all of our tests we use a value of $0.00015 \times N$, with N the number of pixels in the image. The value of $\omega_{reg} = 4$ is related to an assumption about region growing: in an ideal case, at each step, 4 regions will be merged, to maintain a grid structure. The value of parameter $\omega_{sim} = 0.05$ has been learned on Berkeley dataset, and chosen to allow a satisfactory compromise between conciseness and boundary adherence properties. Experimental results of Sect. 3 show that these parameters allow ASARI to achieve competitive results and that they do not require to be re-learned anymore when dealing with other datasets.

3 Experimental Results

We evaluated ASARI using both Stutz experimental design [12] and a new benchmark, Heterogeneous Size Image Dataset (HSID) that we proposed recently⁵. Contrary to Stutz datasets, *i.e.* the 100 images of the Berkeley Segmentation Dataset⁶ (BSD) [7] and the 400 images of the New York University⁷ (NYU) [11] which only contains small images of thousand pixels, HSID is made up of 100 Wikimedia Common photographs, with a majority having several millions of pixels. In addition, variations of image size are significantly more important in HSID than in BSD or NYU.

In Stutz review [12], boundary adherence of evaluated methods is analyzed using two measures: the Undersegmentation Error (UE) and the Boundary Recall

³ https://commons.wikimedia.org/wiki/Main_Page.

⁴ <http://www.ux.uis.no/~tranden/>.

⁵ <http://image.ensfea.fr/hsid/>.

⁶ <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>.

⁷ <http://cs.nyu.edu>.

(BR). Let G be a ground truth segmentation for an image I , S be an oversegmentation of I , B_G be the set of boundary pixels in G , B_S be the set of boundary pixels in S and N be the number of pixels. The undersegmentation error gets for each object G_i in G the set of superpixels required to cover it and counts the number of pixels leaking of G_i . The boundary recall checks whether boundaries of superpixels contain boundaries in G :

$$UE(S, G) = \frac{1}{N} \sum_{G_i \in G} \sum_{S_j \cap G_i \neq \emptyset} \min(|S_j \cap G_i|, |S_j - G_i|) \text{ and } BR(S, G) = \frac{|B_S \cap B_G|}{|B_G|} \quad (7)$$

In fact, even for a human, it is sometimes difficult to know exactly whether a pixel belongs or not to a boundary, Stutz allows a distance of $0.0075 \times \text{diag}$ (where diag is the image diagonal length) pixels between boundary points in G and in S .

Unfortunately these two measures are not suitable to evaluate boundary adherence on HSID. To put it in a nutshell, the average score of UE is not significant due to the important variations in foreground areas in HSID images and the Stutz threshold is not suitable for big images. So, we suggest a modification of BR using fuzzy set theory. From B_G we define the fuzzy set $B_{G \cap S}^*$ with membership function $f_{G \cap S}(p_i) = \exp(-\frac{d(p_i - p'_i)^2}{2\sigma^2})$ where $d(p_i - p_j)$ is the distance between p_i and p_j locations, and $p'_i = \arg \min_{p_j \in B_S}(d(p_i - p_j))$. The function

$f_{G \cap S}$ returns a value in the range $[0, 1]$, a value of 1 meaning a perfect coincidence between an element in B_G and an element in B_S . Finally, we propose the fuzzy boundary recall measure $FBR(S, G) = \frac{1}{|B_G|} \sum_{p \in B_G} f_{G \cap S}(p)$.

Comparison to the state-of-the-art methods - By design, ASARI satisfy validity property, the only requirement is that vertices in G_0 are connected components. Then, because only adjacent regions are merged, the resulting superpixels are always a partition of I into connected components. The average number of neighbors by superpixel equals to 6 on BSD, NYU and HSID dataset, that is similar to all state-of-the-art oversegmentation methods.

Table 1 compares the results of ASARI to those achieved by the methods QS [14], FZ [3], SLIC [1], ERS [5] and CRS [2], in Stutz review [12]. On BSD, UE and BR scores of ASARI are similar to those of the state-of-art methods,

Table 1. Comparison between ASARI and the state-of-the-art oversegmentations methods reviewed by Stutz [12]. K is the average number of superpixels and T the execution time in seconds.

Methods	UE	BR	K	T
ASARI	0.04	0.99	899	1.38
QS	0.03	1	≈ 1000	1.24
FZ	0.03	0.99	≈ 1000	0.059
ERS	0.04	0.99	≈ 1000	1.11
SLIC	0.04	0.99	≈ 1000	0.09
CRS	0.04	1	≈ 1000	0.9

Methods	UE	BR	K	T
ASARI	0.1	0.99	899	2.89
CRS	0.09	1	≈ 1897	1.19
ERS	0.08	0.99	≈ 1500	2.21
SLIC	0.09	0.99	≈ 1500	0.17
FZ	0.07	0.99	≈ 1500	0.1
QS	0.07	0.99	≈ 1500	1.24

Methods	FBR	K	T
ASARI	0.61	1528	6
WP parameters of [6]	0.51	1587	27
CRS parameters of [2]	0.34	1601	21
ERS parameters of [12]	0.60	1500	36
SLIC parameters of [1]	0.58	1648	3
FZ parameters of [8]	0.59	1596	10
QS parameters of [12]	0.53	1676	97

(a) Scores achieve on BSD. (b) Scores achieve on NYU. (c) Scores achieve on HSID.



(a) Image 42049 of BSD: $K = 472$; $UE = 0.03$; $FBR = 0.99$. (b) Image 00001315 of NYU: $K = 905$; $UE = 0.07$; $BR = 1$. (c) Image : img-010 of HSID: $K = 1445$; $FBR = 0.66$. (d) Image : img-025 of HSID: $K = 928$; $FBR = 0.61$.

Fig. 1. Results achieved by ASARI on BSD, NYU and HSID.

with a lower average number of superpixels (about 100 superpixels below). The adaptive nature of ASARI allows it to increase the conciseness of the produced oversegmentation, with some images partitioned into about 500 superpixels. An example is given in Fig. 1a. On NYU, UE of ASARI is slightly less good but BR is similar. The average number of superpixels is more important (about 300 superpixels above). Figure 1b shows that, despite these difficulties, ASARI keeps its ability to adapt. Table 1c shows that ASARI is the method providing the best compromise between boundary adherence and compactness properties on the HSID dataset. The gap is significant with WP (+10% on FBR, for a similar number of superpixels), CRS (+17% with 100 superpixels fewer), SLIC (+2% with 100 superpixels fewer) and QS(+10% with 200 superpixels fewer). The differences with FZ (+2% with a similar number of superpixels) and ERS (+1% with a similar number of superpixels) are less important, but execution times of ASARI are better (near to 2 times faster than FZ and more than 6 times faster than ERS). Figure 1c and d show some examples of ASARI results. In Fig. 1d, the use of texture information allows a clear separation between the foreground and the background, which have similar colors.

Practical application - Interactive segmentation is a semi-automatic segmentation process. The user chooses some pixels (named seeds) and indicates for each of them the element to which it belongs. Features of desired regions are deduced by analyzing these seeds. Usually, adding or removing some seeds can improve the produced result, allowing the user to get any desired segmentation results. Suggested in 2016, Superpixel Classification-based Interactive Segmentation (SCIS) [8] creates a segmentation of the image by learning features of superpixels labeled by the user and classifying the remaining superpixels. The image is first oversegmented using FZ, then the average color and the center of mass location of each superpixel are computed. The learning and classification steps are made using a Support Vector Machine (SVM). Currently, despite its promising result, SCIS is not suitable to segment a dataset like HSID. The oversegmentation step using FZ produces a lot of small superpixels, often misclassified by the SVM. The resulting segmentation is noisy and correcting these errors by adding seeds is tedious. An example of this problem

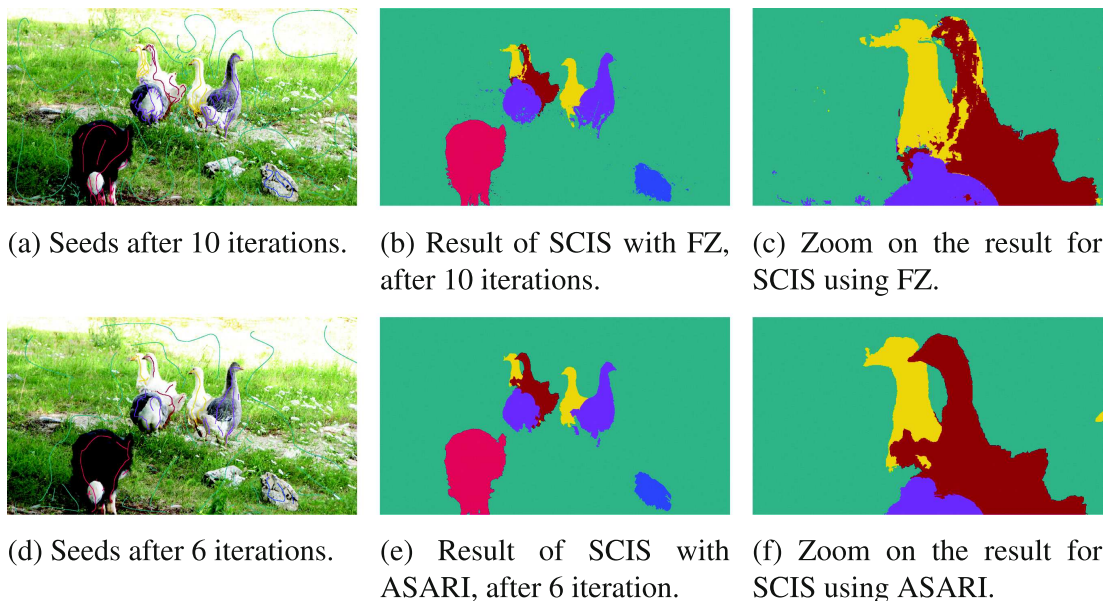


Fig. 2. SCIS with FZ and ASARI: a comparison.

is shown in Fig. 2. As shown in this figure, ASARI, by producing more larger superpixels, considerably reduces this noise. In addition, ASARI is faster than FZ.

4 Conclusion

The main contribution of this paper is the description and the evaluation of a new oversegmentation algorithm, ASARI, which is suitable for datasets with important variations on image sizes and image complexities. The adaptive nature of ASARI and the fact that ASARI creates color and texture homogeneous superpixels while keeping reasonable execution times, allow it to outperform the state-of-the-art methods in such dataset. On other dataset, ASARI obtains results greater or equal to the state-of-the-art methods. The trade-off between boundary adherence, conciseness and efficiency achieved by ASARI offers interesting perspectives. We show an example of ASARI application as a preprocessing step of SCIS method. An implementation of ASARI is available online⁸.

References

1. Achanta, A., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2274–2282 (2012)
2. Conrad, C., Mertz, M., Mester, R.: Contour-relaxed superpixels. In: Heyden, A., Kahl, F., Olsson, C., Oskarsson, M., Tai, X.-C. (eds.) *EMMCVPR 2013*. LNCS, vol. 8081, pp. 280–293. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40395-8_21

⁸ <http://image.ensfea.fr/asari/>.

3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **59**(2), 167–181 (2004)
4. Li, M., Stein, A., Bijker, W., Zhan, Q.: Region-based urban road extraction from VHR satellite images using binary partition tree. *Int. J. Appl. Earth Obs. Geoinformation* **44**, 217–225 (2016)
5. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2097–2104 (2011)
6. Machairas, V., Faessel, M., Cárdenas-Peña, D., Chabardes, T., Walter, T., Decencière, E.: Waterpixels. *IEEE Trans. Image Process.* **24**(11), 3707–3716 (2015)
7. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *8th International Conference on Computer Vision*, vol. 2, pp. 416–423 (2001)
8. Mathieu, B., Crouzil, A., Puel, J.B.: Segmentation interactive pour l’annotation de photographies de paysages. *Reconnaissance des Formes et l’Intelligence Artificielle* (2016). (in French)
9. Ren, X., Malik, J.: Learning a classification model for segmentation. In: *IEEE International Conference on Computer Vision, Proceedings*, vol. 1, pp. 10–17 (2003)
10. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Trans. Image Process.* **9**(4), 561–576 (2000)
11. Silberman, N., Hoiem, D., Pushmeet, K., Fergus, R.: Indoor segmentation and support inference from RGBD images. pp. 746–760 (2012)
12. Stutz, D.: Superpixel segmentation: an evaluation. In: Gall, J., Gehler, P., Leibe, B. (eds.) *GCPR 2015. LNCS*, vol. 9358, pp. 555–562. Springer, Cham (2015). doi:[10.1007/978-3-319-24947-6_46](https://doi.org/10.1007/978-3-319-24947-6_46)
13. Tang, D., Fu, H., Cao, X.: Topology preserved regular superpixel. In: *IEEE International Conference on Multimedia and Expo*, pp. 765–768 (2012)
14. Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008. LNCS*, vol. 5305, pp. 705–718. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-88693-8_52](https://doi.org/10.1007/978-3-540-88693-8_52)
15. Xu, Y., Géraud, T., Najman, L.: Connected filtering on tree-based shape-spaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(6), 1126–1140 (2016)