HAL
open science

# Texture-Aware Superpixel Segmentation

Rémi Giraud, Vinh-Thong Ta, Nicolas Papadakis, Yannick Berthoumieu

## To cite this version:

Rémi Giraud, Vinh-Thong Ta, Nicolas Papadakis, Yannick Berthoumieu. Texture-Aware Superpixel Segmentation. 2019. hal-01995819v2

**HAL Id: hal-01995819**

**https://hal.science/hal-01995819v2**

Preprint submitted on 5 Feb 2019 (v2), last revised 9 Feb 2019 (v3)

**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.

# TEXTURE-AWARE SUPERPIXEL SEGMENTATION

*Rémi Giraud*[1] *Vinh-Thong Ta*[2] *Nicolas Papadakis*[3] *Yannick Berthoumieu*[1]

[1]Bordeaux INP, Univ. Bordeaux, CNRS, IMS, UMR 5218, F-33400 Talence, France.
[2]Bordeaux INP, Univ. Bordeaux, CNRS, LaBRI, UMR 5800, F-33400 Talence, France.
[3]CNRS, Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France.

## ABSTRACT

Most superpixel algorithms compute a trade-off between spatial and color features at the pixel level. Hence, they may need fine parameter tuning to balance the two measures, and highly fail to group pixels with similar local texture properties. In this paper, we address these issues with a new Texture-Aware SuperPixel (TASP) method. To accurately segment textured and smooth areas, TASP automatically adjusts its spatial constraint according to the local feature variance. Then, to ensure texture homogeneity within superpixels, a new pixel to superpixel patch-based distance is proposed. TASP outperforms the segmentation accuracy of the state-of-the-art methods on texture and also natural color image datasets.

***Index Terms*—** Superpixels, Texture, Patch, Segmentation

## 1. INTRODUCTION

Superpixel segmentation approaches that locally group pixels into regions have become very popular in image processing and computer vision applications. The aim is to exploit the local redundancy of information to lower the computational burden and to potentially improve the performances by reducing the noise of a processing at the pixel level. Superpixels can also be considered as a multi-resolution approach that preserves image contours, contrary to standard regular downsampling methods. It is thus a very interesting pre-processing for applications such as visual saliency estimation [1, 2], data association across views [3], segmentation and classification [4, 5, 6] and object detection [7, 8, 9] or tracking [10, 11].

For the past years, most superpixel methods have tended to produce equally-sized regions with homogeneous pixels in terms of color. This paradigm is usually in line with the segmentation of a natural image objects, whose contours can be detected by color changes. Hence, to cluster the pixels into regions, state-of-the-art methods such as [12, 13, 14, 15, 16], only use distance terms in spatial and color (*e.g.*, CIELab) spaces. In [17, 18], more advanced feature spaces are defined to improve the segmentation performances. More recently, [19] proposes to consider contour information in the clustering to ensure the respect of the object boundaries. Nevertheless, such framework requires the need for prior contour detection, at the expense of a global higher complexity.

Most of these state-of-the-art methods only use the pixel information as clustering feature. Therefore, they can be severely impacted by high frequency contrast variations and fail to produce equally-sized regions having the same textural properties. The proposed method TASP is compared in Figure 1 to the state-of-the-art approaches on a synthetic texture image. While TASP produces a relevant segmentation, all other methods are highly misleaded by the texture patterns. We used the regularity parameters recommended by the authors for tunable methods [14, 15, 20, 18, 19], but no other setting would enable to capture texture information. Superpixel methods are indeed generally optimized and evaluated on noise-free natural color images, although specific tasks require to decompose highly textured or low resolution grayscale images, for instance in medical applications [21].

To overcome the limitations of handcrafted color spaces, deep learning approaches have been proposed [22, 23]. Nevertheless, the gain obtained with learned features on a training dataset may come at the expense of usual deep learning limitations, *i.e.*, important learning time, need for a substantial training database and material resources, and direct applicability limited to similar images. Moreover, these approaches do not directly extend to supervoxels and prevent from setting the shape regularity, which can highly impact the performances of superpixel-based tasks. It is thus still necessary to increase the robustness of non-deep learning superpixel methods to textures, while preserving their desired properties: adaptability, low complexity and limited parameter settings.

**Contributions.** In this work, we propose a new Texture-Aware SuperPixel (TASP) clustering method able to accurately segment highly textured images, but also any input image, *e.g.*, natural color ones, using the same parameters.

To be able to generate relevant superpixels on textured images (see Figure 1), TASP adjusts its spatial constraint, according to the feature variance within the superpixel. This way, TASP also addresses the need for fine manual regularity setting. Most recent state-of-the-art methods globally set this parameter according to the image nature, leading default or sub-optimal settings to highly impact the results [24].

Preprint available at: https://arxiv.org/abs/1901.11111

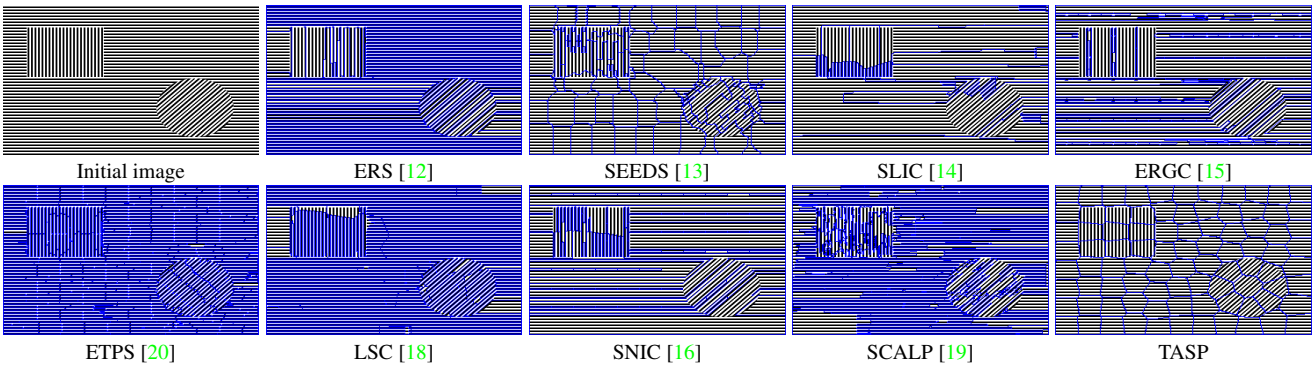



**Fig. 1**. Comparison of the TASP method to state-of-the-art approaches on a synthetic texture image for 100 superpixels. Only TASP succeeds in capturing the textures while other methods are highly misleaded by high frequency contrast variations.

Then, to ensure the texture homogeneity, we introduce a new patch-based framework enabling to easily evaluate the similarity of a pixel neighborhood to a superpixel.

We validate TASP on natural color images from a standard segmentation dataset [25], and on two new datasets proposed to evaluate texture segmentation performances. TASP significantly outperforms the state-of-the-art methods on texture segmentation performances, while performing as well, or better, on natural images, using the same parameters.

## 2. TEXTURE-AWARE SUPERPIXELS

The TASP method improves the superpixel decomposition approach used in [14, 18, 19], that is first presented in this section. Then, we propose a method to locally set the spatial regularity of superpixels, to automatically adapt to the image content. Finally, we introduce a new pixel to superpixel texture homogeneity measure to group pixels in terms of texture.

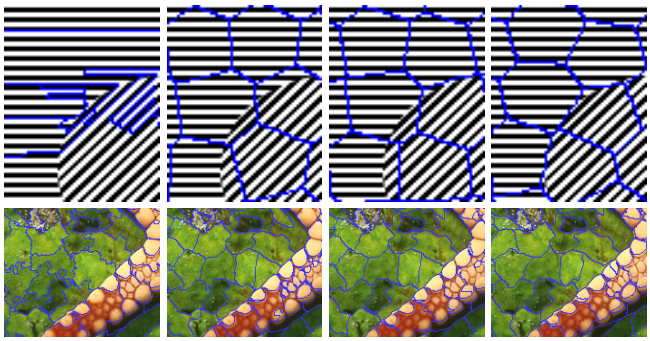
### 2.1. K-means-based Iterative Clustering

The standard framework of [14] only requires the number of superpixels to produce and a regularity parameter. The algorithm is based on an iteratively constrained K-means clustering of pixels. Superpixels $S_i$ are first regularly set over the image domain as blocks of size $s \times s$, and are described by their average intensity feature $F_{S_i}$ (CIELab colors for [14]) and their spatial barycenter $X_{S_i} = [x_i, y_i]$ of pixels in $S_i$. The clustering relies on a feature $d_F(F_p, F_{S_i}) = \|F_p - F_{S_i}\|_2$, and a spatial distance term $d_s(X_p, X_{S_i}) = \|X_p - X_{S_i}\|_2$. At each iteration, each superpixel $S_i$ is compared to all pixels $p$, of feature $F_p$ at position $X_p$, within a $(2s+1)\times(2s+1)$ region around its barycenter $X_{S_i}$. A pixel $p$ is associated to the superpixel $S_i$ minimizing the distance $D$ defined as:

$$D(p, S_i) = d_F(F_p, F_{S_i}) + d_s(X_p, X_{S_i})\frac{m^2}{s^2}, \qquad (1)$$

with $m$ the parameter setting the superpixel shape regularity. A post-processing step finally ensures region connectivity.

Although this method can accurately gather pixels having similar colors, $m$ is globally set and cannot adapt to all local image contours. It also highly fails to capture texture patterns, as it only considers feature information at the pixel level.



(a) SLIC [14] (b) w/ (2) (c) w/ (2), (3) (d) w/ (2), (3), (4)

**Fig. 2**. SLIC [14] with optimal regularity for color images (a) vs TASP contributions (b)-(d), that accurately decomposes both texture and color images with the same parameters (d).

### 2.2. Local Adaptation of Superpixel Regularity

For most methods, including [14, 18, 19], the regularity parameter $m$ must be manually set, according to the dynamic of the feature term $d_F$. Hence, default parameters for natural color images may lead model (1) to generate highly irregular clustering on textures, and the post-processing step enforcing connectivity to irrelevantly merge regions (see Figure 1). We address this issue by using for each superpixel $S_i$, a regularity parameter $m_i$ defined according to the feature variance $\sigma_i(F_p)$ of all pixels $p \in S_i$ such that:

$$m_i = m \exp\left(\frac{\sigma_i(F_p)^2}{\beta}\right), \qquad (2)$$

with a scaling parameter $\beta$. Such regularity term $m_i$ is able to increase the spatial constraint in the TASP model (5) for superpixels having high feature variances, and to reduce it in smooth areas, so the superpixel boundaries can capture image objects that are perceptible from limited feature variations.

This way, without manually adapting $m$ in (2), TASP can compute relevant superpixels on both highly textured images (Figure 2(b)-top), and natural color ones (Figure 2(b)-bottom), since $m_i$ (2) automatically adjusts the trade-off between $d_F$ and $d_s$ in (5). Nevertheless, the clustering accuracy still has to be improved to capture texture information.

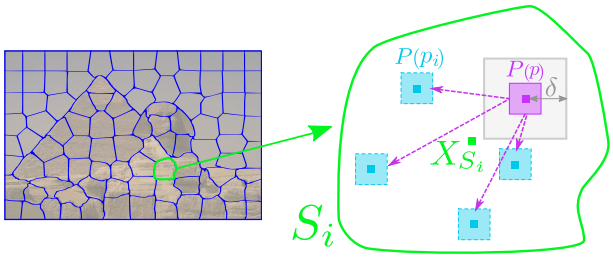



**Fig. 3**. Selection of similar patches $P(p_i)$ in a superpixel $S_i$ of barycenter $X_{S_i}$, outside a $\delta$-neighborhood, to compute the texture homogeneity term (3) for a patch $P(p)$.

### 2.3. Texture Homogeneity Measure

#### *2.3.1. Pixel to Superpixel Patch-based Distance*

In this section, we propose a method to measure the texture similarity between a pixel neighborhood and the content of a superpixel, thus between two regions of different sizes. A texture descriptor at patch and superpixel levels would yield higher complexity and additional parameter settings. Moreover, texture cannot be preserved as well as for color and spatial information with a global average over the whole superpixel. The framework must preserve its limited complexity, and to be able to adapt to any image content without any prior information. Such constraints also prevent from using costly dictionary or learning-based approaches.

To address these issues, we propose a new framework using square patches to naturally capture texture information. For a pixel $p$, of patch $P(p)$, and a superpixel $S_i$, a nearest neighbor algorithm (see section 2.3.2) is used to find similar patches $P(p_i)$ such that $p_i \in S_i$, and outside a $\delta$-neighborhood around $p$ (see Figure 3). The new term $d_P$ computes the average distance to the selected $p_i \in S_i$:

$$d_P(p, S_i) = \frac{1}{N} \sum_{p_i \in \mathcal{K}_p} d_P(p, p_i), \tag{3}$$

with $\mathcal{K}_p$ the set of $N$ selected pixels $p_i \in S_i$, compared with a patch distance in the feature space, such that $d_P(p, p_i) = \frac{1}{n}\|F_{P(p)} - F_{P(p_i)})\|_2$, with $n$ the patch size.

Any feature can be used in term (3). This way, we propose a general model that can easily evaluate the texture compliance of a pixel neighborhood to a superpixel, while leveraging the need for complex texture classification approaches.

#### *2.3.2. Patch-based Nearest Neighbor Search*

The search for similar patches can be performed by any nearest neighbor (NN) method. We choose to use PatchMatch, a fast iterative approximate-NN algorithm based on the propagation of good matches from adjacent neighbors [26]. The computation of $d_P(p, S_i)$ can be directly performed for all pixels $p$ in the $(2s+1)\times(2s+1)$ area around the barycenter $X_{S_i}$ of $S_i$. The algorithm being partly random, $N$ patches in $S_i$ can be selected in parallel for each pixel $p$, to increase the robustness of the texture homogeneity term (3).

#### *2.3.3. Texture Unicity within Superpixels*

In the texture term (3), the patch similarity is computed regardless of any spatial information. Hence, a pixel $p$ to cluster may find similar local textures in restricted areas, leading a superpixel to potentially group several textures (Figure 2(c)).

To ensure the texture unicity within a superpixel $S_i$, we consider in $d_P$ the spatial distance between the selected patches $P(p_i)$, at position $X_{p_i} \in S_i$, and $X_{S_i}$, the spatial barycenter of the superpixel $S_i$ such that:

$$d_P(p, p_i) = \frac{1}{n}\|F_{P(p)} - F_{P(p_i)})\|_2 + \frac{m_i^2}{s^2}\,\Gamma\big(X_{p_i}, X_{S_i}\big), \tag{4}$$

with $\Gamma$, a scaling function defined such that $\Gamma(X_{p_i}, X_{S_i}) = 2s^2(1 - \exp\left(-\|X_{p_i} - X_{S_i}\|_2^2/s^2\right))$. Such term iteratively contributes to restrict the search area and the diversity of textures within $S_i$ by highly penalizing similar patches found far from $X_{S_i}$. Hence, the barycenter is encouraged to move to a homogeneous textured area and to be contained within the superpixel (see Figure 2(d)). This also increases the shape regularity, which is a desirable property [24].

The clustering distance in TASP is finally computed as:

$$D(p, S_i) = d_F(F_p, F_{S_i}) + d_s(X_p, X_{S_i})\frac{m_i^2}{s^2} + d_P(p, S_i). \tag{5}$$

This way, TASP becomes a very general method, able to handle textures, and efficient on various image types with the exact same parameters, as demonstrated in the next section.

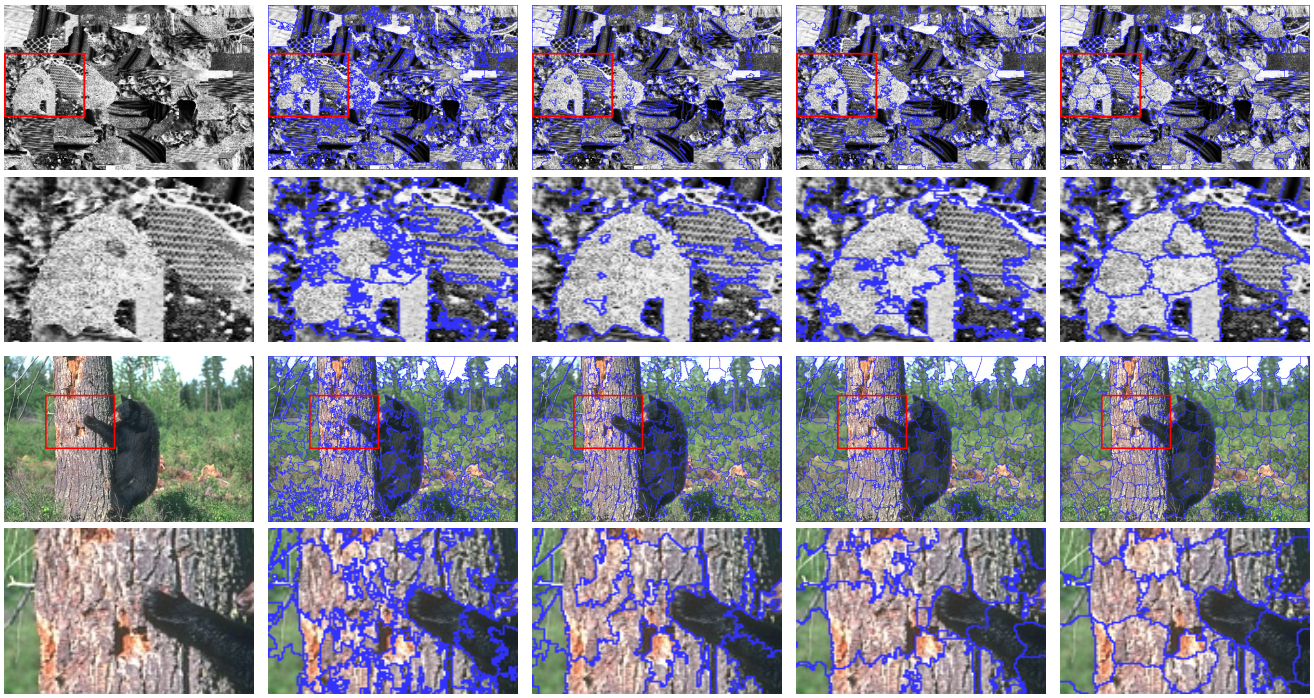
## 3. RESULTS

### 3.1. Validation Framework

Similarly to [27], we create two new datasets to evaluate texture segmentation performances. A highly challenging synthetic stripe (mix-Stripes) dataset of 10 images of size $300\times400$ pixels, is created by putting stripes similar to the ones in Figure 1, in variable shaped regions of minimum size 1000 pixels. Natural textures with normalized intensity are also taken from the Brodatz dataset [28], to create 100 composite images (mix-Brodatz), that can contain up to 10 different textures. Finally, we consider the standard Berkeley Segmentation Dataset (BSD) [25], containing 200 natural color test images of size $321\times481$ pixels.

Most parameters are empirically set once and for all, and their tuning has a moderate impact on performances. For the patch search, $N = 8$ patches of size $5\times5$ pixels are selected outside a $\delta = 3$ neighborhood. In the clustering model, $m$ is set to 0.1 and $\beta$ to 25 in (2). Color features $F$ are computed as in [19]. Finally, the whole clustering process of TASP is performed in 10 iterations as in [14].

TASP is compared to the recent state-of-the-art methods SLIC [14], ERGC [15], ETPS [20], LSC [18], SNIC [16], and SCALP [19]. Performances are evaluated with standard Achievable Segmentation Accuracy (ASA), and contour detection metric F-measure (F) as defined in [24], and we report quantitative results for an average number of 250 superpixels.



Initial image | LSC [18] | SNIC [16] | SCALP [19] | TASP

**Fig. 4**. Comparison between TASP and most recent state-of-the-art methods. TASP produces a more relevant result on a natural texture composite image (top) and on a natural color image example from the BSD (bottom), for 200 superpixels.

### 3.2. Influence of Contributions

The visual impact of contributions is shown in Figure 2, and ASA and F measures are reported in Table 1 on the three considered datasets. Our new texture homogeneity term (3) and constraint of texture unicity within superpixels (4) both significantly improve performances on each data type.

| | mix-Stripes | | mix-Brodatz | | BSD | |
|---|---|---|---|---|---|---|
| Method | ASA | F | ASA | F | ASA | F |
| TASP w/o (3),(4) | 0.8303 | 0.3498 | 0.7969 | 0.4736 | 0.9484 | 0.4945 |
| TASP w/o (4) | 0.8486 | 0.3882 | 0.8112 | 0.4812 | 0.9493 | 0.4961 |
| **TASP** | **0.8706** | **0.4232** | **0.8139** | **0.4824** | **0.9503** | **0.4992** |

**Table 1**. Influence of contributions in the TASP method.

### 3.3. Comparison to the State-of-the-Art Methods

TASP is compared to state-of-the-art approaches on an image similar to the mix-Stripes dataset in Figure 1, and to the most recent methods on mix-Brodatz and BSD images in Figure 4.

A quantitative evaluation is also performed on the three datasets in Table 2. TASP significantly increases the performances on the synthetic (mix-Stripes) and natural (mix-Brodatz) texture datasets, demonstrating its ability to provide texture-aware superpixels. TASP also obtains the best results on natural color images (BSD), using the same parameters, while other methods fail at providing accurate results on the three data types at the same time. Note that compared methods are used with parameters recommended by the authors. Nevertheless, no other approach explicitly captures texture information, so TASP with default parameters still outperform state-of-the-art methods manually optimized for each dataset.

| | mix-Stripes | | mix-Brodatz | | BSD | |
|---|---|---|---|---|---|---|
| Method | ASA | F | ASA | F | ASA | F |
| SLIC [14] | 0.7256 | <u>0.4048</u> | 0.7784 | 0.4607 | 0.9445 | 0.4706 |
| ERGC [15] | 0.6107 | 0.3717 | 0.7796 | 0.4677 | 0.9477 | 0.4571 |
| ETPS [20] | <u>0.7769</u> | 0.2953 | 0.7568 | 0.4354 | 0.9433 | 0.4710 |
| LSC [18] | 0.6979 | 0.3156 | 0.7908 | 0.4552 | <u>0.9503</u> | 0.4421 |
| SNIC [16] | 0.6659 | 0.3529 | 0.7662 | <u>0.4815</u> | 0.9410 | 0.4617 |
| SCALP [19] | 0.7307 | 0.3290 | <u>0.7977</u> | 0.4759 | 0.9499 | <u>0.4914</u> |
| **TASP** | **0.8706** | **0.4232** | **0.8139** | **0.4824** | **0.9503** | **0.4992** |

**Table 2**. TASP compared to the state-of-the-art methods. Best and second results are respectively bold and underlined.

## 4. CONCLUSION

In this paper, we address the severe non-robustness of superpixel approaches to texture images by proposing a texture-aware decomposition method. A new patch-based framework is introduced to gather pixels having both similar color and local textural properties. The proposed method is general, removes the need for manual setting of regularity constraint, and also naturally can extend to generate supervoxels.

We outperform the segmentation of state-of-the-art methods on color, and synthetic and natural texture datasets, by using the same parameters. This work opens the way for larger use of superpixels and efficient application of our approach to medical image segmentation or video object tracking.

## 5. REFERENCES


[1] Z. Liu, X. Zhang, S. Luo, and O. Le Meur, "Superpixel-based spatiotemporal saliency detection," *TCSVT*, vol. 24, no. 9, pp. 1522–1540, 2014. 1

[2] S. He, R. W. Lau, W. Liu, Z. Huang, and Q. Yang, "SuperCNN: A superpixelwise convolutional neural network for salient object detection," *IJCV*, vol. 115, no. 3, pp. 330–344, 2015. 1

[3] R. Sawhney, F. Li, and H. I. Christensen, "GASP: Geometric association with surface patches," in *3DV*, pp. 107–114, 2014. 1

[4] S. Gould, J. Zhao, X. He, and Y. Zhang, "Superpixel graph label transfer with learned distance metric," in *ECCV*, pp. 632–647, 2014. 1

[5] R. Gadde, V. Jampani, M. Kiefel, D. Kappler, and P. V. Gehler, "Superpixel convolutional networks using bilateral inceptions," in *ECCV*, pp. 597–613, 2016. 1

[6] R. Giraud, V.-T. Ta, A. Bugeau, P. Coupé, and N. Papadakis, "SuperPatchMatch: An algorithm for robust correspondences using superpixel patches," *TIP*, vol. 26, no. 8, pp. 4068–4078, 2017. 1

[7] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," vol. 33, no. 5, pp. 898–916, 2011. 1

[8] G. Shu, A. Dehghan, and M. Shah, "Improving an object detector and extracting regions using superpixels," in *CVPR*, pp. 3721–3727, 2013. 1

[9] J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li, "Object detection by labeling superpixels," in *CVPR*, pp. 5107–5116, 2015. 1

[10] J. Chang, D. Wei, and J. W. Fisher, "A video representation using temporal superpixels," in *CVPR*, pp. 2051–2058, 2013. 1

[11] M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann, "Temporally consistent superpixels," in *ICCV*, pp. 385–392, 2013. 1

[12] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *CVPR*, pp. 2097–2104, 2011. 1, 2

[13] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," in *ECCV*, pp. 13–26, 2012. 1, 2

[14] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *PAMI*, vol. 34, pp. 2274–2282, 2012. 1, 2, 3, 4

[15] P. Buyssens, I. Gardin, S. Ruan, and A. Elmoataz, "Eikonal-based region growing for efficient clustering," *IVC*, vol. 32, no. 12, pp. 1045–1054, 2014. 1, 2, 3, 4

[16] R. Achanta and S. Süsstrunk, "Superpixels and polygons using simple non-iterative clustering," in *CVPR*, pp. 4895–4904, 2017. 1, 2, 3, 4

[17] Y.-J. Liu, C.-C. Yu, M.-J. Yu, and Y. He, "Manifold SLIC: A fast method to compute content-sensitive superpixels," in *CVPR*, pp. 651–659, 2016. 1

[18] J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," *TIP*, vol. 26, pp. 3317–3330, 2017. 1, 2, 3, 4

[19] R. Giraud, V.-T. Ta, and N. Papadakis, "Robust superpixels using color and contour features along linear path," *CVIU*, vol. 170, pp. 1–13, 2018. 1, 2, 3, 4

[20] J. Yao, M. Boben, S. Fidler, and R. Urtasun, "Real-time coarse-to-fine topologically preserving segmentation," in *CVPR*, pp. 2947–2955, 2015. 1, 2, 3, 4

[21] Z. Tian, L. Liu, Z. Zhang, and B. Fei, "Superpixel-based segmentation for 3D prostate MR images," *TMI*, vol. 35, no. 3, pp. 791–801, 2016. 1

[22] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Superpixel sampling networks," in *ECCV*, 2018. 1

[23] W.-C. Tu, M.-Y. Liu, V. Jampani, D. Sun, S.-Y. Chien, M.-H. Yang, and K. Jan, "Learning superpixels with segmentation-aware affinity loss," in *CVPR*, 2018. 1

[24] R. Giraud, V.-T. Ta, and N. Papadakis, "Evaluation framework of superpixel methods with a global regularity measure," *JEI*, vol. 26, no. 6, 2017. 1, 3

[25] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *PAMI*, vol. 26, no. 5, pp. 530–549, 2004. 2, 3

[26] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ToG*, vol. 28, no. 3, 2009. 3

[27] T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," *PAMI*, vol. 21, no. 4, pp. 291–310, 1999. 3

[28] P. Brodatz, "Textures: A photographic album for artists and designers," *Dover Publications*, 1966. 3