19th Computer Vision Winter Workshop
Zuzana Kúkelová and Jan Heller (eds.)
Křtiny, Czech Republic, February 3–5, 2014

Over-Segmentation of 3D Medical Image Volumes based on Monogenic Cues

Markus Holzer and Rene Donner

Computational Imaging Research Lab,
Department of Radiology and Nuclear Medicine,
Medical University of Vienna, Austria
markus.holzer, rene.donner@meduniwien.ac.at

Abstract *In this paper, we propose a novel approach to compute 3D supervoxels for radiological image datasets. It allows to cope with the high levels of noise and low contrast encountered in clinical data such as Computed Tomography (CT), Optical Coherence Tomography (OCT) and Magnetic Resonance (MR) images.*

The method, monoSLIC*, employs the transformation of the image content to its monogenic signal as primal representation of the image. The phase of the monogenic signal is invariant to contrast and brightness and by selecting a kernel size matched to the estimated average size of the superpixels it highlights the locally most dominant image edge. Employing an agglomeration step similar to the one used in SLIC superpixels yields superpixels/-voxels with high fidelity to local edge information while being of regular size and shape.*

The proposed approach is compared to state of the art superpixel methods on the real-world images of the 2D Berkley Segmentation Dataset[1] *(BSD) converted to gray-scale, as well as challenging 3D CT and MR volumes of the Visceral*[2] *dataset. It yields a highly regular, robust, homogeneous and edge-preserving over-segmentation of the image / volume while being the fastest approach.*

1 Introduction

The goal of over-segmenting an image is to merge pixels into homogeneous groups of superpixels while preserving boundaries of objects in an image. This allows to perform image analysis tasks on the greatly reduced number of superpixels as opposed to every pixel in the volume. While several approaches to computing superpixels have been proposed in recent years, most were developed with the application to typical 2D color photographs in mind. Radiological image data, on the other hand, has quite different characteristics: monochrome data, high levels of noise and low contrast, both finely detailed structures (e.g. the internal structure of bones) and large objects with only subtle texture differences (liver in a CT). Our aim is to provide a method specifically adapted to these characteristics which is also computationally attractive.

[1] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/
[2] http://www.visceral.eu/

All but one (Mori [13]) of the existing methods work directly on the image pixel intensities (see Section 2), causing them to be depended on contrast and brightness of the image. The method of [13] extracts texture and contour information, but with high computational costs [9]. In contrast the proposed method extracts structure information using the computationally attractive approach of the monogenic signal. The phase of the monogenic signal contains the local structural information of the image, from which edge cues are extracted and then used as input for k-means clustering. The number of k-means cluster centers is equal to the number of desired superpixels, resulting in a fast and parameter-free (additional to the desired number of superpixels) over-segmentation method that creates regular and smooth superpixels.

In Section 2 the state of the art is summarized and the new method is described in Section 3, showing the definition and incorporation of the monogenic signal into the proposed approach, as well as its extension to 3D data. Section 4 presents the experimental setup and results, with Section 5 providing a conclusion.

2 State of the Art

The state of the art approaches of the over-segmentation problem can be categorized into *graph-partition-based* and *gradient-ascent-based* [2]. Visual examples for each algorithm are presented in Figure 1 for a cropped part of an abdominal CT slice.

Graph-Partition-Based
The methods listed in this category use a graph to represent the similarity between pixels, where each node refers to a pixel and the weight between nodes refers to the similarity of the pixels. The graph is then cut at weights where the difference between nodes is significant, creating a superpixel segmentation.

Felzenswalb [7] proposed an efficient graph-based image segmentation approach, where the image is represented in a 5-D feature space containing spatial information (*x,y*) and color information (*r,g,b*). A nearest neighbor graph is created, with the weights corresponding to the distance in the feature space. The graph is then cut into components representing a minimum spanning tree of associated pixels. The resulting superpixels are perceptually meaningful but there

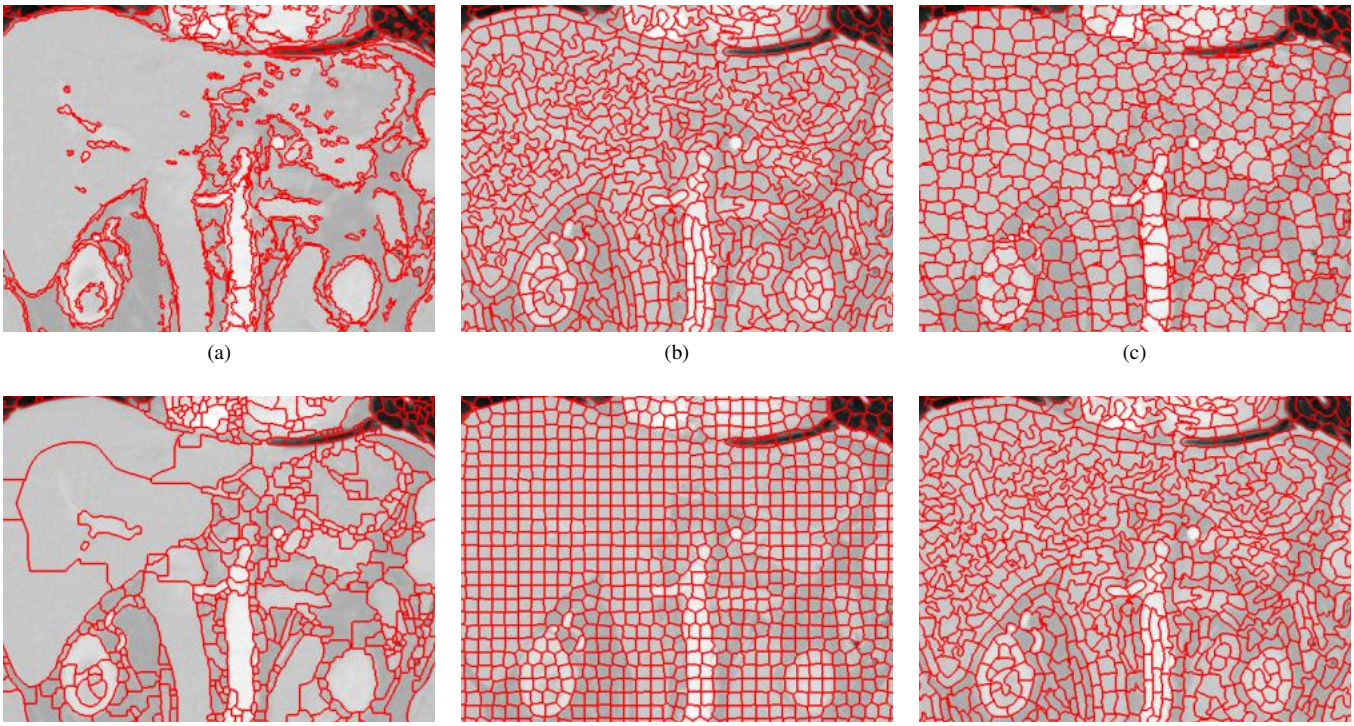


Figure 1: Comparison of the superpixel segmentation on a 2D Computed Tomograph slice of the abdominal region with about 200 pixels per superpixel in average for (a) Felzenswalb [7], (b) Mori [13], (c) Veksler [14], (d) Engel [5], (e) Achanta [1] and (f) MonoSLIC.

are no constraints in terms of superpixel number or size, thus creating irregular ones.

The approach of **Mori** et. al. [13] on the other hand creates regular and visually pleasing superpixels. The quality of its segmentation performance is based on computational intensive extraction of texture and contour features. These are then combined to a final weight matrix creating a graph of size $O(N^2)$ where the normalized graph cut method, proposed by Shi and Malik [10] with a high computational cost of $O(N^{\frac{3}{2}})$ [9], is used to calculate the segmentation.

Similar to the previous method **Veksler** [14] also creates and cuts a graph, but this time an extended grid-graph is created. In this graph each pixel is connected to its neighbors as well as to a number of terminal nodes, each representing a possible superpixel label. A multi-way-graph cut approach is then used to calculate the final segmentation of the image. Two versions of the algorithm are proposed, one for more regular and compact superpixels and another for irregular but more precise superpixels. The latter is named *constant intensity superpixels* and is used for the evalutation in this paper.

Gradient-Ascent-Based In this category methods are used that start by an initial segmentation, which is then improved based on the gradient of the feature space. The boundaries are moved to, or created where, the gradient magnitude of the image is a local maximum.

For the method of **Engel** [5] the initial segmentation is the distribution of seed-points of a watershed transformation. The seed-points and the height-map are calculated by exploiting the properties of the *Gradient Vector Flow* (GVF). Based on the GVF a flux flow field is calculated which serves as the height-map and this height-map is thresholded to generate the initial seed-points. The properties of the superpixel output is similar to [7], with irregular shape and size.

For **Achanta** [2] the initial segmentation is a rectangular grid, which is then iteratively updated to align with local high gradients using the k-means clustering method. The features space for the *Simple Linear Iterative Clustering* (SLIC) method is a 5-D space, which is similar to [7], but instead of the *r,g,b* the *L,a,b* values of the CIELAB color space is used. The distance function for k-means is non-euclidean with a parameter that regulates the weight between *x,y* and *L,a,b* values, giving the user the option to create regular superpixel at the cost of edge precision.

Summary and Additional Approaches A compact summary of the analyzed methods is given in Figure 2. Additional approaches not compared in this paper were proposed, among them the gradient-ascent based approach of Vincent [15], the TurboPixel approach of [9] and the graph-partition-based Lattices approach of [12]. A comparison of these can be found in [2].

3 Methods

The proposed method is designed to over-segment 2D and 3D image data, with a special focus on medical data. Such medical data typically exhibits high levels of noise and low

State of the Art						
Algorithm	Category	Parameter	Preprocessing	Information Representation		Segmentation
Veksler 2010 [13]	graph partition	superpixel count	-	Intensity	Energy Function represented as extended Grid Graph	Energy Minimization as Multi-Way-Graph Cut optimized by α-Expansion
Felzenswalb 2004 [6]	graph partition	significance value	-	Intensity	Grid Graph represented as Boundary Decision Rule	Disjoint-Set Forest optimized by Inverse Ackermann Function
Mori 2005 [12]	graph partition	superpixel count	texture and contour feature extraction	Features	Energy Function represented as Full Graph	Energy minimization as solving a General Eigenvalue System
Engel 2009 [4]	gradient ascent	seed points threshold	edge detection	Edge Information	Gradient Vector Flow Field to compute a Flux Flow Image	Watershed with Seedpoints and Height Map from Flux Flow Field
Achanta 2010 [1]	gradient ascent	superpixel count and regularity	-	Color Information	CIELab Color Space combined with Spatial Information	optimized k-means
Presented Approach						
MonoSLIC 2014	gradient ascent	superpixel count	Monogenic Signal	Local Monogenic Phase	Monogenic Phase combined with Spatial Information	optimized k-means

Figure 2: Overview of the discussed methods.

contrast, as shown on an abdominal CT in Figure 1 and an abdominal MR in Figure 3 (a), as well as the noisy OCT shown in Figure 3 (b). Using the contrast sensitive original gray-level information forces the user to choose a parameter that decides about the level of detail that should be captured. This parameter needs to be tweaked and set appropriately for the methods of [5], [7] and [2]. With the use of the structural information contained in the phase of the monogenic signal, which is brightness and contrast invariant, the proposed method does not require such an parameter. Extracting the structural information also makes the method robust to noise and creates smooth superpixel boundaries.

Our method is similar to the approach of [2], but in our case we perform k-means clustering in a feature space that is spanned by the spatial coordinates *(x,y,z)* and the monogenic phase Λ, which results in 4 dimensional feature space for the 3D case. The properties of the monogenic phase allow for an adaption of the k-means algorithm, that reduces the number of calculations necessary and therefore making our method the fastest 3D approach.

In the following we recapitulate the definition of the monogenic signal, detail how it is employed as additional feature for the k-means clustering and present the changes to the k-means method compared to [2]. In the remainder of this paper the proposed method is called **MonoSLIC**.

The Monogenic Signal The *monogenic signal* proposed by Felsberg [6] originates from the 1D analytic signal [4], which extracts the local phase and local amplitude of a signal. It is based on the Hilbert transform which when applied to $cos(x)$ of a real valued signal x results in $sin(x)$. The Hilbert transform can be therefore understood as a *phase shifter*, shifting every sinosoidal function by -90 degrees. In the frequency domain, with the Fourier transformed signal u the transfer function of the Hilbert transform can be defined as $H(u) = \frac{-iu}{|u|} = -i\,sgn(u)$, where $sgn()$ is the signum function [6]. The kernel can be written as the inverse Fourier transform of the transfer function [17]

$$f_H(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(u)e^{iux}du \tag{1}$$

and its combination with the original signal is called the *analytic signal*

$$f_A(x) = f(x) - if_H(x), \tag{2}$$

The Hilbert transform is generalized in [6] using the Riesz transfrom with the transfer function defined in the frequency domain $R(\mathbf{u}) = \frac{-i\mathbf{u}}{|\mathbf{u}|}$, where $\mathbf{u} = (u_1, ..., u_n)^T$, for n dimensional signals [3]. Similar to the analytic signal the monogenic signal is also a combination of the original signal and, this time, its Riesz transformation $f_R(x)$,

$$f_M(x) = f(x) - \mathbf{i}f_R(x), \tag{3}$$

where is $\mathbf{i} = (i_1, ..., i_n)$ dimensional. The monogenic signal is isotropic and also performs a so-called split of identity [6]. This refers to the fact that the signal is decomposed into the local amplitude A_f, the local phase φ and the local orientation θ.

The local amplitude A_f is defined as the norm of the monogenic signal

$$A_f(x) = |f_M(x)| = \sqrt{f^2(x) + |f_R(x)|^2}. \tag{4}$$

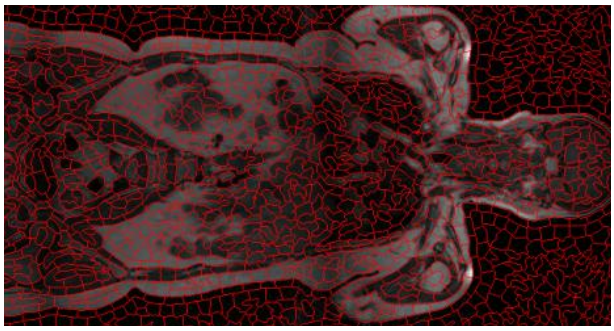
and the local phase

$$\varphi(x) = arg(f_M(x)), \quad \varphi \in [-\pi, \pi] \tag{5}$$

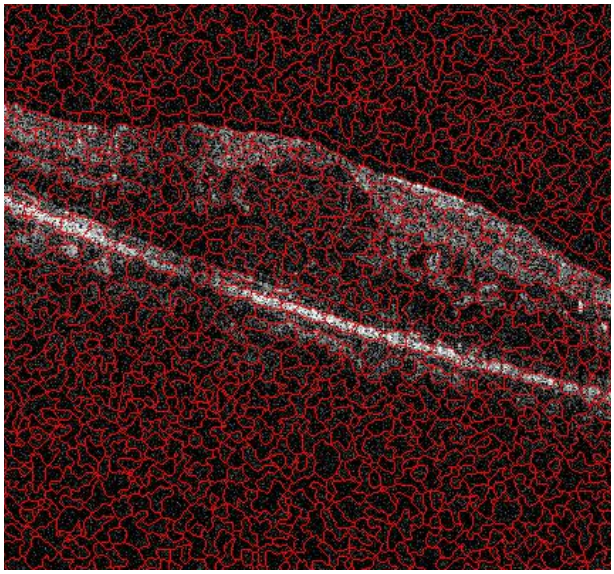
represents the change of local strutural information in the range of $-\pi$ to π [6]. Examples for corresponding structures for a certain φ value are shown in Figure 4 for the 1D case. Finally the geometric information is represented by the orientation in the range of 0 to π

$$\theta(\mathbf{x}) = \arccos\left(f(\mathbf{x})\right)/A_f(\mathbf{x}))\,\theta \in [0, \pi] \tag{6}$$

defining the direction of the structure.



(a)



(b)

Figure 3: Example images for 3D medical volumes with the segmentation of MonoSLIC, where (a) shows a Magnetic Resonance Tomography of the thorax and head and (b) an Optical Coherence Tomograph of an eye.

Edge-cues from Monogenic Signal and filter scale selection The monogenic phase $\varphi(x)$ picks up the locally dominant structure in an image regardless of contrast and brightness. Its values are between $-\pi, \pi$ and a value of $\pm\frac{\pi}{2}$ correlates with a strong edge, as previously illustrated in Figure 4. In order to detect this change in structure with the k-means algorithm the values have to be mapped such that there is a high difference between two pixels with different values for $sgn(|\varphi(x)| - \pi/2)$. In other words, to get a representation with similar feature values within superpixels and value changes at their boundaries, the monogenic phase needs to be transformed. We first map φ to $\varphi_{new} = |\varphi| - \frac{pi}{2}$ before calculating the final monogenic phase cue with

$$\Lambda\left(x\right) = \frac{sgn\left(\varphi_{new}\right)\exp^{(-|\varphi_{new}|)}}{2}, \Lambda \in [-0.5, 0.5]\,. \quad (7)$$

The wavelength governing the scale of the monogenic signal depends on the number of superpixels and is set to $\lambda = \sqrt[d]{\frac{P}{SP}}$ (for dimension d, the number of pixels P and the number of superpixels SP) which influences the size of the smallest structure that should be detected in the image. λ correlates to the number of superpixels because a small number can only detect large structures and the lower wavelength ensures smoothness balanced with the level of detail, that is desirable at this scale.

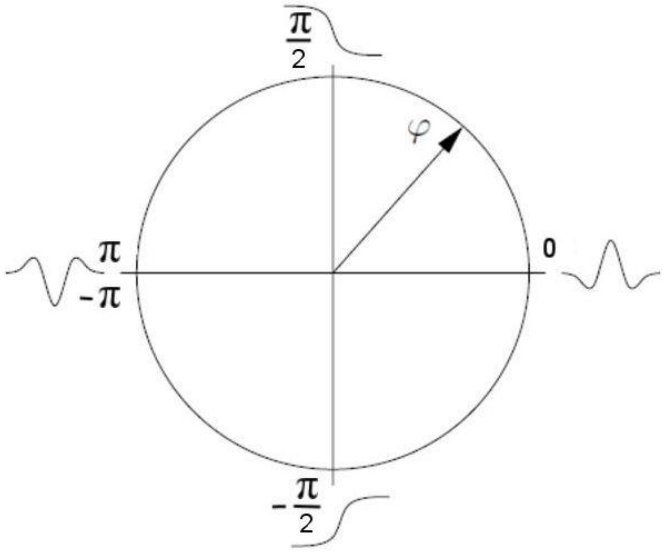


Figure 4: The sketched structure yielding a specific value of φ with a phase wrap from π to $-\pi$. Redrawn from [6].

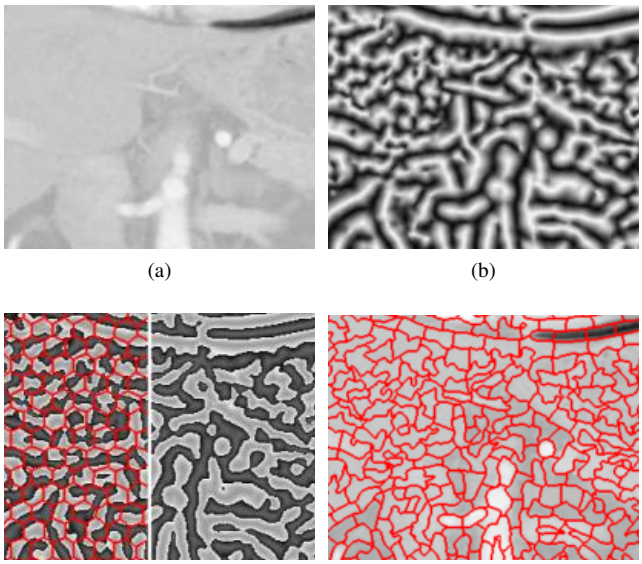
In the example shown in Figure 5 (c) of the transformed monogenic phase a high contrast can be seen at the change of structure of the original image. Using this as input for k-means will create clusters with are boundaries at the changes of structure. This is the only information about the image employed by our approach.

Super-pixels through clustering Similar to [2], we perform k-means clustering to obtain the superpixels, but instead of using the Lab values color space, we cluster based on the monogenic phase. We initialize the cluster centers (seeds) spatially according to a hexagonal grid with a cluster center distance $d_{cc} = \lambda$ corresponding to the number of superpixels, to avoid imposing too much of a directional preference. Each pixel is represented in an $nD + 1$ space, with the coordinates being the original $n = 2$ pixel or $n = 3$ voxel coordinates. The additional dimension is the monogenic phase, which is scaled by two times the cluster center distance

$$\Lambda(x)_{kmean} = 2d_{cc}\Lambda(x). \quad (8)$$

For the k-means computation we restrict the number of potential pixels for each cluster seed to a $\pm 2d_{cc}$ neighborhood and the maximum number of iterations to $I = 10$, reducing the k-means complexity $O(10SR)$ introduced by [2] to $O(9SN + SR), N \ll R$, where S is the number of seedpoints, R a region around S and N a number of randomly chosen pixels from R. The compact information contained in the monogenic phase makes it possible to choose a random subset of pixels from the region R, reducing the number of computations per iteration.

For the first iteration of the clustering, the initial values for the $nD + 1$'st coordinate of the cluster center have to be



(a) (b) (c) (d)

Figure 5: The monogenic phase calculated on an abdominal Computed Tomography image (a) with a filter wavelength of 18.3. In (b) the local phase φ of the image is shown. Panel (c) presents the transformed signal Λ as used by our approach with the overlayed initial cluster centers ($d_{cc} = 18.3$) on the left. In panel (d) the final segmentation is overlayed with the original image.

estimated. These are set to the average of $\Lambda(x)_{kmean}$ of all pixels that are closest (distance $< \frac{d_{cc}}{2}$) to that cluster center.

4 Experiments

In Section 4.1 the setup of the experiments is presented, followed by the results in Section 4.2.

4.1 Setup

The methods presented in this paper are run using their respective reference implementations. The parameters governing the number of superpixels was carefully selected for each case to yield the same number of superpixels, where possible. Parameters specifying edge fidelity versus homogeneity were set to a value such that edge responses are captured, while still maintain regularity. The parameter for [1] was set to 15, which provides a trade-off in terms of regularity and recall and was used throughout this paper. The algorithms are compared in terms of recall, the regularity of their superpixels, their robustness to noise and their runtime.

Two datasets are used for evaluation. The first is the BSD [11] with 500, 0.15 Mega Pixels (MP) real-world images, with objects annotated by five different annotators. The second is the Visceral [8] dataset, which consists of 14 CT and 14 MR volumes. In each of the volumes 20 different anatomies are annotated by medical experts. From the 3D volumes also a 2D dataset is created by extracting the 28 coronal center slices of the dataset with an average size of 0.18 MegaPixel (MP). For the 3D test all the 14 CT and but only 5 abdominal MR volumes are taken, as the method of [1] failed to compute on the other MR volumes.

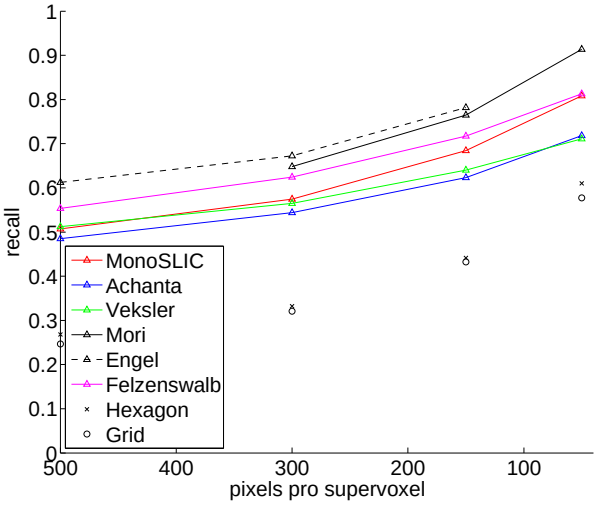


Figure 6: Recall rate for the 2D Visceral image dataset.

4.2 Results

In this section first the results for the 2D Visceral and BSD are presented. The recall rate on the Visceral dataset and the regularity on the BSD dataset are analyzed, before looking at the summary of all acquired statistics. In the following the 3D Visceral results are presented and at the end the run-time performance of the algorithms is evaluated.

2D - Visceral and Berkley Segmentation Dataset The first comparison of the algorithms is the **recall** rate, which describes the percentage of how many of the annotated pixels were detected by the algorithms segmentation. A higher recall value indicates a better detection of the annotated boundaries. The recall rate for the 2D Visceral dataset is measured at defined superpixel sizes shown in Figure 6. The reference over-segmentation using a rectangular and hexagon grid show how a blind over-segmentation would perform. The slightly better results for the hexagon are due to the naturally more meaningful shape compared to a rectangle.

The approaches of Mori [13], Felzenswalb [7] and Engel [5] have the highest recall accuracy followed by MonoSLIC, Veksler [14] and Achanta [1]. The accuarcy of [13] comes at the cost of computation, which is also the reason for the missing first data-point. The methods of [5] and [7] do not allow direct control over the superpixel number. For [5] it was not possible to calculate values for 100 pixels per superpixel due to the properties of the medical images.

The **regularity** of the superpixels is evaluated because smooth superpixel boundaries lower the computation cost of algorithms that compute on the segmentation itself. Another benefit of creating smooth boundaries is that they are visually more pleasing [16], which could be important when presented to physicians. The results are calculated on the BSD, as for regularity comparison this dataset is more representative. That is because medical images contain a high percentage of smooth and homogenous background that would benefit the methods of [1] and [14]. The regluarity is measured

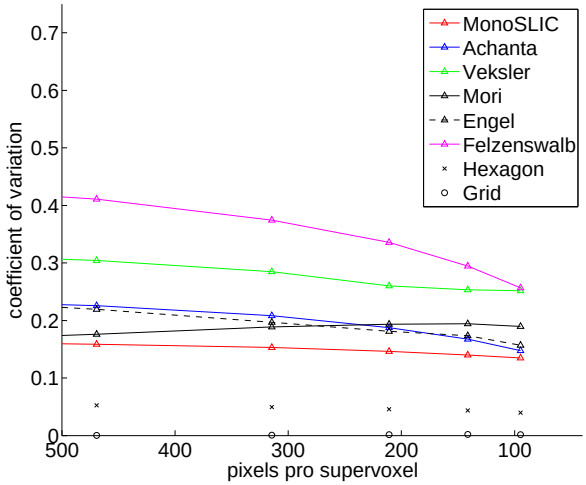


Figure 7: Coefficient of variation for the perimeter over area value on the 2D Berkley Segmentation Dataset images.

with the relation of perimeter *peri* and area *area*, where the final value is $\frac{peri}{\sqrt{area}}$. The mean over all the superpixels of this value is an indicator for the roughness of the boundary, where a lower value means smoother superpixel boundaries. The corresponding standard deviation expresses the consistency of the smoothness, e.g. if there is an high or low variation in superpixel shape. Both measurements are combined in the coefficient of variation, where the mean is divided by the standard deviation. The results are shown in Figure 7 where lower values indicate smoother superpixel boundaries. The MonoSLIC approach has the lowest coefficient of variation and therefor creates the smoothest superpixels for the BSD dataset.

To get a compact view of all the results for the BSD dataset they are **summarized** in Figure 8, where green indicates good performance and desired parameters and red otherwise. The presented columns are recall, recall on images with added Gaussian noise ($mean = 0, std = 0.22$), standard deviation of the superpixel area in pixels, mean and standard deviation for perimeter over area, the runtime in seconds per MP, if there is a parameter P to set (0 equals no parameter) and if the number of superpixels SP can directly be specified. The values are taken for 70 pixels per superpixel.

Going through the algorithms from bottom to top [7] shows very good runtime and recall rates but the user cannot control the number of superpixel, which also causes a high variation in the area and the regularity of the superpixels. The approach of [5] has the same properties, although a lower recall rate and and higher computation time. The method of [13] achieves the highest recall rate with 84% and 79% for noise, but this comes at a high computational cost, taking about 455s to compute one image. The approach of [14] has a low recall rate and is influenced by noise, the superpixels are of irregular shape and also a relative high runtime of 38s. The recall rate of [1] is a moderate 73% for the normal images but is highly influenced by noise, where the recall rate drops to 60%. The average regularity is the highest of the tested methods but also has a low variation. It has

pixel/SP	recall (%)		area (pixel)	perimeter over area		runtime			
90	normal	noise	std	mean	std	s/MP	P	SP	3D
MonoSLIC	0.72	0.72	796	4.56	0.52	1.6	0	1	1
Achanta	0.73	0.60	922	6.03	0.57	1.6	1	1	1
Veksler	0.71	0.63	4316	5.29	0.87	38	0	1	0
Mori	0.84	0.79	497	4.05	0.72	455.4	0	1	0
Engel	0.71	0.75	7618	5.17	0.57	4.8	1	0	0
Felzenswalb	0.77	0.80	8098	5.29	1.24	0.9	1	0	0
Hexagon	0.54	0.54	314	3.50	0.13	-	-	-	-
Grid	0.50	0.50	36	3.92	0.01	-	-	-	-

Figure 8: Summary of the results for the 2D Berkley Segmentation Dataset images and an over-segmentation of 90 pixels per superpixels, where green indicates desired results.

pixel/SP	recall	area (pixel)	perimeter over area		runtime			
70	normal	std	mean	std	s/MP	P	SP	3D
MonogSLIC	0.68	178	4.57	0.63	1.6	0	1	1
Achanta	0.62	136	4.57	0.38	1.6	1	1	1
Veksler	0.64	401	3.90	0.83	38	0	1	0
Mori	0.76	62	3.95	0.76	455.4	0	1	0
Engel	0.78	4364	4.91	0.67	4.8	1	0	0
Felzenswalb	0.72	6072	5.99	1.76	0.9	1	0	0
Hexagon	0.44	56	3.40	0.13	-	-	-	-
Grid	0.43	10	3.82	0.01	-	-	-	-

Figure 9: Summary of the results for the 2D Visceral images and an over-segmentation of 70 pixels per superpixels, where green indicates desired results.

one of the best run-times and is available in 3D. On the other side it has a parameter that has to be tweaked for a trade-off between compactness. The proposed method MonoSLIC has is very robust to noise with a recall rate of 72% for both the normal and noisy images. It creates very regular superpixels and does not require a parameter to be set.

The summary is also presented for the 2D Visceral dataset in Figure 9. There are some differences to the BSD results. The methods of Achanta [1] and Veksler [14] have a better regularity which is due to the smooth background of medical images. For medical images the recall rate of MonoSLIC is higher when compared to the other methods on the BSD.

3D - Visceral Dataset For the 3D case again the **recall** rate is shown for the Visceral dataset. This time the supervoxel size is given in mm^3 as the medical recordings have a defined real-world size for one voxel. The results are shown for a range of about 10^5 to $10^3 mm^3$ in Figure 10. Our method outperforms Achanta [1] for the broader range of superpixel size, for a over-segmentation in the range of $1 - 10cm^3$.

The 3D results are again **summarized** in Figure 11 for a supervoxel size of about $7 * 10^3 mm^3$. The recall rate is 10% higher for MonoSLIC. With a runtime of $0.7s/MP$ the performance of MonoSLIC is 3 times faster than Achanta [1] and it requires no parameter to be set apart of the desired number of superpixels.

A visual comparison in terms of recall rate is presented for the Visceral dataset. In the following two figures the segmentation of the algorithm is shown in red, the annotated ground truth in green and blue corresponds to a matching segmentation of the algorithm to the ground truth. Each figure is divided by a white line, where the left shows the re-

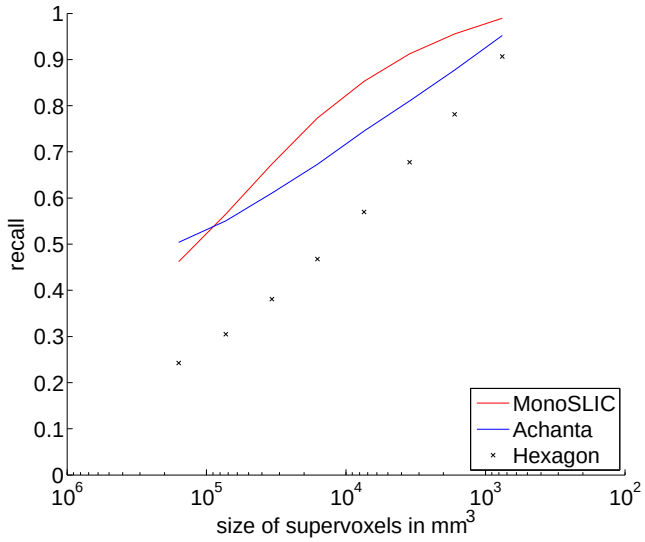


Figure 10: Recall for the 3D Visceral dataset.

SP (mm^3)		runtime		
7*10^3	recall	s/MP	P	SP
MonoSLIC	0.77	0.7	0	1
Achanta	0.67	2.1	1	1

Figure 11: Summary of the results for an over-segmentation into $15 * 10^3 mm^3$ sized supervoxels, where green indicates desired results.

sults of Achanta [1] and the right of MonoSLIC. In Figure 12 a 3D Visceral CT example is presented. If the object boundary has a high contrast it is very well segmented by [1] as seen for the lung and the silhouette of the body, while in the Abdomen region the contrast is very low and the structure is not captured. Tweaking the parameter could increase the structure detected at the cost of regularity. MonoSLIC on the other hand reacts to all the structures which is the reason for the better recall performance and it does so without the need of an additional parameter.

This can also be observed for a 2D Visceral MR example shown in Figure 13 where the uniqueness of the recording technique creates very low contrast volumes. The results for the MR are shown for the 2D case because [1] did not compute on the 3D volume.

Run-Times The run times of the algorithms are displayed in Figure 14. Computations were performed on a 12-core Intel Xeon with the publicly available implementations. Mori [13] not only is the slowest algorithm, but due to the large weight matrix generated for the *NCut* it also scales badly in terms of memory usage. Veksler [14] segments images in about 40 seconds. While Engel [5] has a faster computation time of 4.7 s/MP, Felzenswalb [7] manages to process 1 MP in under 1 second. For images of size 0.15 MP Achanta [1] and monoSLIC take 1.6 s/MP. While [1] has a similar performance for larger images and volumes MonoSLIC segments these in 0.7 s/MP, which is the fastest of all algorithms.

With 12 cores the CPU allows parallel processing, but only a few parts of the methods make use of all the cores.

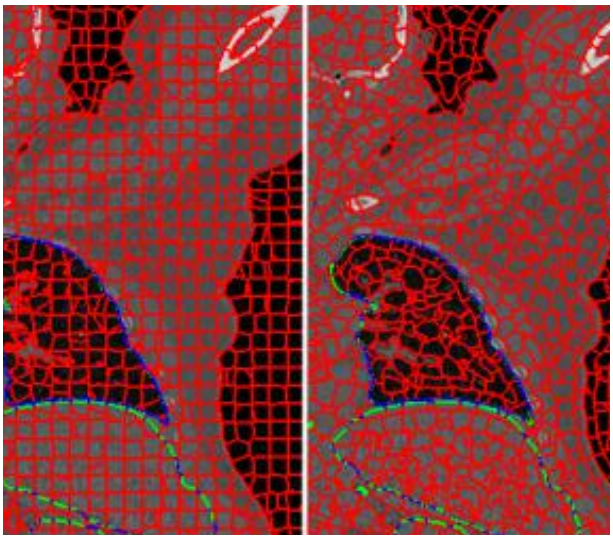


Figure 12: Comparison of Achanta [1] on the left and MonoSLIC on the right on an 3D Visceral Computed Tomography slice example with a supervoxel size of $7 * 10^3 mm^3$.

The features extracted by [13] as well as the NCut segmentation technique use multiple cores. The only other method that has an advantage of the multi-core CPU is the monogenic signal calculation of monoSLIC, as the FFT used for convolution is multi-threaded.

There is more potential for speed improvements. The method of [14] is designed for running in parallel but is only implemented in *C++* for single core. The code of [13] and [5] also have improvement potential as the code is partially implemented in *Matlab* and *C* and only some parts are parallel. The *FFT* used by the monogenic signal could also adapted to make use of the GPU, which for 3D volumes is currently limited by the memory.

5 Conclusion

We have presented *monoSLIC*, a novel method for the computation of superpixels / -voxels, which incorporates the monogenic signal's unique characteristics of being invariant to contrast and brightness as well as robust to noise. It represent the dominant edge information in any given image patch corresponding to a selected scale. The resulting superpixels / -voxels provide a nice balance of of compactness, homogeneity and fidelity to edge information. Combined with a fast runtime in 3D the algorithm outperforms state of the art methods in terms of runtime, recall, regularity and robustness to noise and is therefor particularly well suited for 3D radiological image data.

Acknowledgement

This work was partly supported by the European Union FP7 (KHRESMOI FP7-257528, VISCERAL FP7-318068), by the Austrian National Bank Anniversary Fund (BIOBONE 13468, AORTAMOTION 13497, FETALMORPHO 14812), and the Austrian Science Fund FWF (PULMARCH P 22578-B19).

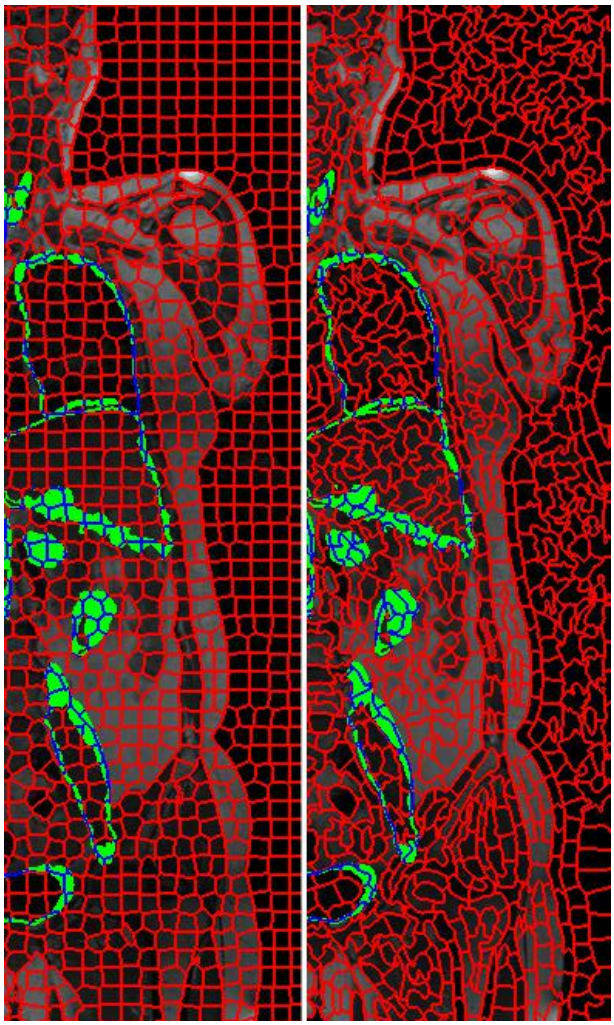


Figure 13: Comparison of the Achanta [1] on the left and MonoSLIC on the right on an 2D Visceral Magnetic Resonance slice example with about 200 pixels per supervoxel.

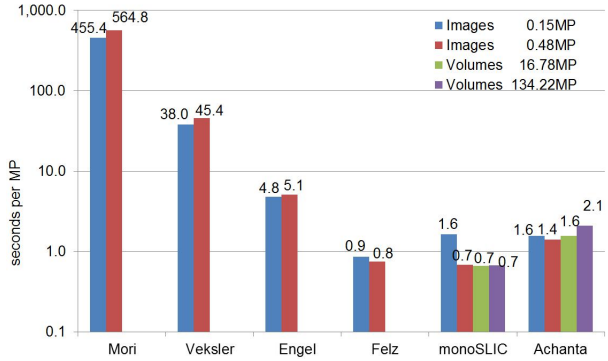


Figure 14: Runtime performance per MP in logarithmic scale on different image and volume sizes.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels. Technical Report 11, EPFL, Nov. 2010.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE PAMI*, 34(11):2274–82, Nov. 2012.

[3] T. Bulow, D. Pallek, and G. Sommer. Riesz transforms for the isotropic estimation of the local phase of moire interferograms. In *22. Symposium fuer Mustererkennung*, pages 333–340. Springer-Verlag, 2000.

[4] L. Cohen. Time-frequency distributions-a review. *Proceedings of the IEEE*, 77(7):941–981, 1989.

[5] David Engel and Cristobal Curio. Scale-Invariant Medial Features based on Gradient Vector Flow Fields. In *19th ICPR*, pages 1–4. IEEE, Dec. 2008.

[6] M. Felsberg and G. Sommer. The Monogenic Signal. *IEEE Transactions on Signal Processing*, 49(12):3136–3144, 2001.

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

[8] G. Langs, H. Müller, B. H. Menze, and A. Hanbury. VISCERAL: Towards Large Data in Medical Imaging - Challenges and Directions. In *Medical Content-Based Retrieval for Clinical Decision Support*, pages 92–98, Nice, France, 2013.

[9] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. TurboPixels: fast superpixels using geometric flows. *PAMI*, 31(12):2290–2297, 2009.

[10] J. Malik. Normalized Cuts and Image Segmentation. *IEEE PAMI*, 22(8):888–905, 2000.

[11] 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 *Proceedings of the 8th ICCV*, volume 2, pages 416–423, 2001.

[12] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel Lattices. In *CVPR*, pages 1–8. IEEE, June 2008.

[13] G Mori. Guiding model search using segmentation. In *10th ICCV*, volume 2, pages 1417–1423. IEEE, 2005.

[14] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and Supervoxels in an Energy Optimization Framework. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, pages 211–224, 2010.

[15] L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithm based on Immersion Simulations. *IEEE Transactions on PAMI*, 13(6):583–598, June 1991.

[16] M. Wertheimer. Untersuchungen zur Lehre von der Gestalt II. *Psychologische Forschung*, 4(1):301–350, 1923.

[17] L. Yi-Wen. Hilbert transform and applications. In *Fourier Transform Applications*, pages 291–300. InTech, 2012.