SAND2014-2731C

Superpixel segmentation using multiple SAR image products[1]

Mary M. Moya[2], Mark W. Koch, David N. Perkins, R. Derek West
Sandia National Laboratories[3], PO Box 5800, MS 1163, Albuquerque, NM 87185-116

ABSTRACT

Sandia National Laboratories produces copious amounts of high-resolution, single-polarization Synthetic Aperture Radar (SAR) imagery, much more than available researchers and analysts can examine. Automating the recognition of terrains and structures in SAR imagery is highly desired. The optical image processing community has shown that superpixel segmentation (SPS) algorithms divide an image into small compact regions of similar intensity. Applying these SPS algorithms to optical images can reduce image complexity, enhance statistical characterization and improve segmentation and categorization of scene objects. SPS algorithms typically require high SNR (signal-to-noise-ratio) images to define segment boundaries accurately. Unfortunately, SAR imagery contains speckle, a product of coherent image formation, which complicates the extraction of superpixel segments and could preclude their use.

Some researchers have developed modified SPS algorithms that discount speckle for application to SAR imagery. We apply two widely-used SPS algorithms to speckle-reduced SAR image products, both single SAR products and combinations of multiple SAR products, which include both single polarization and multi-polarization SAR images. To evaluate the quality of resulting superpixels, we compute research-standard segmentation quality measures on the match between superpixels and hand-labeled ground-truth, as well as statistical characterization of the radar-cross-section within each superpixel. Results of this quality analysis determine the best input/algorithm/parameter set for SAR imagery. Simple Linear Iterative Clustering provides faster computation time, superpixels that conform to scene-relevant structures, direct control of average superpixel size and more uniform superpixel sizes for improved statistical estimation which will facilitate subsequent terrain/structure categorization and segmentation into scene-relevant regions.

Keywords: superpixel, SAR, oversegmentation, Quick-shift, Simple Linear Iterative Clustering (SLIC), undersegmentation error, boundary recall, radar cross section, subaperture multilook, coherent change detection

1. INTRODUCTION

SAR images provide a wealth of information about structures and activities in an imaged scene. Frequently, however, details of interest are buried amongst mundane structures, vegetation and terrains. While image analysts are adept at extracting significant details, too few analysts are available to examine the multitude of SAR images produced each day. An automated terrain/structure categorization system could augment their ability to examine these large data sets by characterizing commonly occurring objects and backgrounds and recognizing changes over time. Sandia National Laboratories has established 'Pattern ANalytics To support High performance Exploitation and Reasoning' (PANTHER) to develop an automated analyst support system. The first stage of the PANTHER system will process a large database of SAR images, categorize regions according to their content and produce classification maps of relevant scene features for representation in a semantic graph subject to analyst query. Rather than classify individual image pixels, the PANTHER classifier first oversegments the image into superperpixels, which are small groups of pixels that share similar location and intensity. These superpixels facilitate statistical characterization of the terrain/structure classes, as well as reduce computational complexity. This paper describes a variety of options for creating superpixel segmentations of SAR images, which include using different superpixel segmentation (SPS) algorithms, different parameter sets and different SAR image products, and quantitatively compares the quality of the resulting superpixels.

[1] This work was supported by PANTHER, a Laboratory Directed Research and Development (LDRD) Project at Sandia National Laboratories. For additional information about PANTHER, please contact Kristina Czuchlewski, Ph.D., krczuch@sandia.gov.

[2] Contact mmmoya@sandia.gov for reprints, questions or more information.

[3] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000. SAND#

A wide body of research has developed and applied SPS algorithms for optical imagery [1-4]. These applications show that superpixels preserve boundaries of imaged objects, provide self-similar regions for measuring feature statistics, reduce artifacts and boundary misclassifications introduced by sliding window algorithms and improve processing speed by reducing computational complexity.

Recent developments [5-9] have allowed application of SPS segmentation to SAR imagery, as well, even though SAR images contain speckle, a result of coherent combining and cancelling of the multi-path backscattered radar energy. In the formed SAR image, speckle is a multiplicative phenomenon. SPS algorithms designed for optical imagery use a difference-based distance metric, which responds poorly to multiplicative contamination. Thus, researchers have employed different strategies for oversegmenting SAR imagery into superpixels. Caves, et.al. [10] introduce two segmentation algorithms for SAR magnitude images, which both detect edges using mean differences between regions. They test these algorithms on simulated multi-look imagery and use a normalized intensity ratio to combine statistically similar segments to produce their final segmentations. Fjortoft [11] extends work of Caves and Touzi [12] and detects edges from ratios of average intensity in SAR imagery and applies a modified watershed algorithm to produce oversegmented superpixels. Su, et.al. [5] apply multi-scale Meanshift [13] segmentation to decomposed multilook polarimetric data, then implement a random field merging which incoporates a graph-based representation for relationships between regions of multiple scales. Wu, et.al. [14] apply a noise-adjusted quad-tree decomposition to derive superpixels and then combine superpixels with random field merging. Gan et.al. [7] modify the Turbopixels algorithm [4] to incorporate an exponentially-weighted ratio of local mean intensities to represent edge strength. Yu et.al. [8] apply Gaussian smoothing within the Turbopixels algorithm to reduce influence of speckle in producing superpixels. Xiang et.al. [9] incorporate a Gaussian-smoothed intensity ratio distance into a local k-means algorithm to extract superpixels.

We apply widely-used SPS algorithms without modification to speckle-reduced SAR image products. For the backscatter image products, we compute log-magnitude before applying SPS algorithm. The resulting difference-based distance measure for log-magnitude images provides the same advantage of reduced sensitivity to multiplicative contamination as a ratio-based distance measure for magnitude images. We measure the quality of the resulting superpixels against hand-labeled segments using widely-used SPS quality measures. We apply SPS algorithms to combinations of image products as well as individual image products. An important feature of the approach is that we can use superpixels derived from any product to process any other co-registered SAR image product.

2. SAR DATA PRODUCTS

SAR imaging systems deliver complex phase history data, which pass through Fourier transform processing to produce complex-valued backscatter images. Because the multi-path backscattered energy can add and cancel coherently, the resulting images contain speckle, which complicates the application of standard image processing methods to SAR images. Fortunately, exploiting temporal and spatial statistics, we can derive a number of SAR image products designed to reduce speckle. Table 1 lists Ku-band radar collection parameters for the SAR test image used in this work. The following subsections describe the speckle-reduced image products and show small subsections (approx. 1000 x 1000 pixels) of each product.

Table 1. Radar parameters.

Radar Parameter	Value	Radar Parameter	Value
Frequency	16.8 GHz	Wavelength	1.8 cm
Grazing angle	40.4°	Squint angle	-88.4°
Range resolution (ground)	5.5 in	Azimuth resolution (ground)	5.1 in
Image size	3464 x 4864	Range distance	2.9 km

2.1. Subaperture multilook image

Forming a subaperture multilook (SA-ML) image requires transforming a complex-valued SAR image back to the two-dimensional Fourier domain, partitioning the spectrum into non-overlapping pieces and non-coherently averaging the images formed from each piece of the spectrum [15]. The SA-ML image has coarser spatial resolution than an SAR backscatter magnitude image formed from the complete phase history, but also has reduced speckle. Figure **1**a) shows an example of an SA-ML image. We calibrate the SA-ML backscatter and compute the log-magnitude before applying the SPS algorithm. In future sections, SA-ML will refer to $20\log_{10}$(SA-ML).

2.2. Median-over-time of Radar Cross Section (RCS) image

We can use multiple passes of SAR images collected from the same scene to compute image statistics for speckle reduction. We calibrate each RCS image and, from a stack of co-registered RCS images of the same scene, compute a median image to form the Median RCS (MRCS) images, which is a temporal multilook product. We compute the log-magnitude of the MRCS image before applying the SPS algorithm. In future sections, MRCS will refer to $20\log_{10}$(MRCS). Figure **1**b) shows an example of an MRCS image.

2.3. Mean-over-time of Coherent Change Detection (CCD) image

The complex-valued radar backscatter image provides not only magnitude of the backscattered radar return, but phase of the returned signal, which is useful for measuring coherence between multiple passes and forming interferometric height maps [15]. Coherent change detection images, produced from co-registered images of the same scene collected at different times, can detect subtle changes such as tire tracks in a dirt road or breeze-induced shifts in vegetation. Collecting a large number of passes (25 or more) from the same scene allows us to characterize patterns of change over time. We form CCD images from pairs of the image data set, then average many CCDs together to create Mean CCD (MCCD) images. Figure **1**c) shows a mean CCD image formed from an average of 29 CCDs where a dark color indicates regions of low average coherence produced by physical scene changes between collection-times of the image pairs or shadows, regions blocked from the radar signal. The surrounding desert exhibits slightly more change over time than the packed dirt parking lot and driveway. The gravel to the right of the building exhibits relatively small change over time whereas the three trees inside the oval planting area have moderate change likely a result of small displacements from wind and weather. The key point for creating superpixels is that structures made of the same material have similar change properties over time. The MCCD image product represents structures in the image in slightly different ways than do the products based on backscatter magnitude. Some structures are more prominent and others are less so.

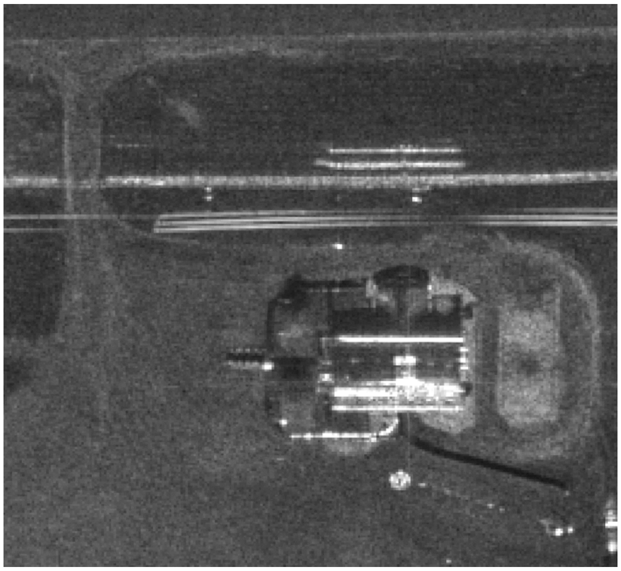

Figure **1**a) Subaperture multilook product

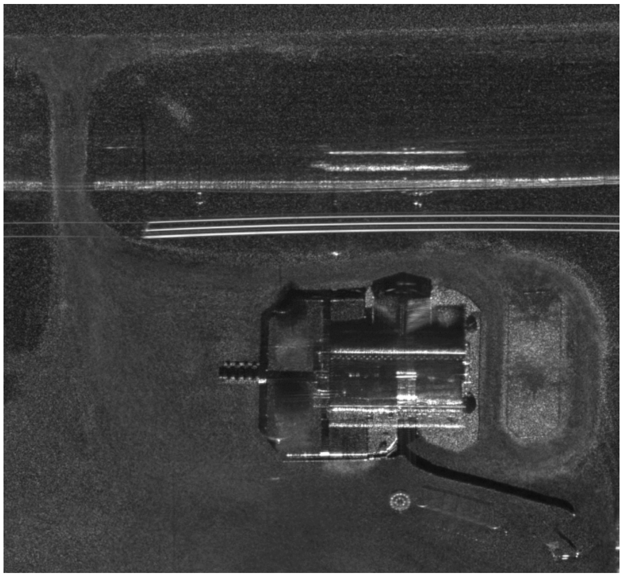

Figure **1**b) Median RCS product

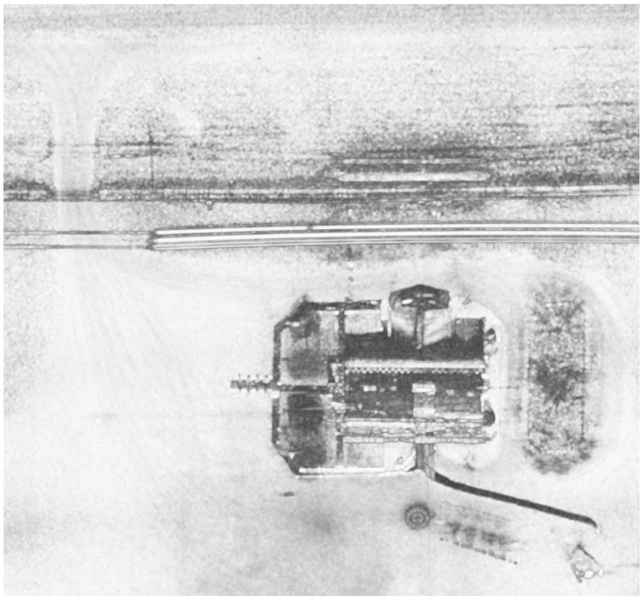

Figure **1**c) Mean CCD product

Figure 1. SAR product images showing driveway, power lines, parking lot, building, oval landscaping area and playground.

3. SUPERPIXEL SEGMENTATION ALGORITHMS

When faced with the challenge of categorizing contents of large SAR images, we apply superpixel segmentation algorithms to exploit spatial correlation between neighboring pixels. Each superpixel represents a small region of the image that contains similar intensities in close spatial proximity. We applied two widely used SPS algorithms, Quick-shift [13] and Simple Linear Iterative Clustering [2, 16] to the SAR image products.

3.1. Quick-shift

The Quick-shift algorithm forms a Parzen density estimate [17] of spatial and intensity distance relationships between neighboring pixels. The distance between pixels *i* and *j*, shown in Equation (1), accounts for both difference in intensity and difference in spatial proximity between the two pixels. Here, r_i is the row index, c_i is the column index and I_i is the intensity of pixel *i*.

$$d_{ij}^2 = (r_i - r_j)^2 + (c_i - c_j)^2 + (I_i - I_j)^2 \tag{1}$$

Equation (2) shows the local Parzen density estimate for a pixel *i* over a neighborhood N_W, where σ is the Gaussian window width and *W* is the search window width. The density estimate produces larger values for pixels whose close neighbors have similar intensities. Figure **2**a) shows a conceptual representation of the Parzen density over the row and column grid space.

$$p_i = \sum_{j \in N_W} e^{-d_{ij}^2 / 2\sigma^2} \tag{2}$$

In contrast to gradient search algorithms, which iteratively compute gradients in the direction of a local maximum, Quick-shift constrains its search to pass through image pixels. For each image pixel, Quick-shift searches a local region to find a 'parent' pixel, one that has higher density than itself and closest distance within the search region [18]. Figure **2**b) shows the Parzen density contours with black dots that indicate the locations of image data pixels. The red arrows indicate the association between each pixel, at the tail, and its parent, at the head. Some pixels are self-parents with no tails because they represent local density maxima. Then, each pixel follows parents-of-parents, creating a chain of associations, until it reaches a parent with largest local density. Figure **2**c) shows a simple example of this second search step, which finds two parent-of-parent pixels indicated by bold red **x**s. All pixels associated with each local maximum form the corresponding superpixel, where pixels labeled with '1' correspond to one local maximum and pixels labeled with ‘2’ correspond to the other local maximum. By searching for local parent pixels with higher density, Quick-shift avoids computationally expensive gradient calculations, but still achieves a constrained locally optimal solution. Computational complexity of Quick-shift is $O(N^2)$ [13].

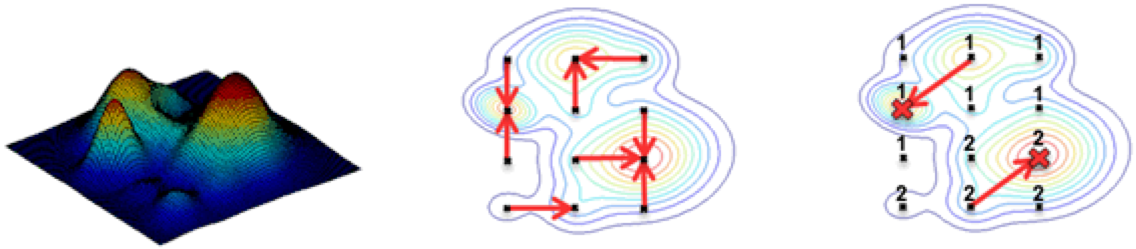

Figure **2**a) Diagram of Parzen density estimate of spatial-intensity distance.

Figure **2**b) Diagram of local search for higher-density, minimum-distance parent.

Figure **2**c) Diagram of local search for parent-of-parent.

Figure 2. Diagrams of Quick-shift search for largest local density clusters.

3.2. Simple Linear Iterative Clustering (SLIC)

The SLIC algorithm implements a localized *k*-means [19] algorithm with a distance metric that depends on both spatial and intensity differences. Equation (3) gives the Euclidean spatial distance between pixels *i* and *j*.

$$d_{sij} = \sqrt{(r_i - r_j)^2 + (c_i - c_j)^2} \tag{3}$$

Equation (4) gives the Euclidean intensity distance for 3-channel intensities with weighting factors. Achanta, et.al.[2] use the LAB colorspace channels, with $w_1 = w_2 = w_3 = 1$ when applying SLIC to optical color imagery. We select one, two or three of the SAR image products as input channels for the SLIC segmentation. We scale the three channels to similar magnitude ranges and use equal weightings on the channels in our experiments. If we had reason to believe that any channel(s) were more important than other(s), we could adjust the weights accordingly. When applying SLIC to either SA-ML or MRCS, we first compute log-magnitude. Applying the Euclidean distance in the log-magnitude domain is equivalent to a ratio-intensity distance in the magnitude domain, as proposed by Xiang et.al. [9].

$$d_{Iij} = \sqrt{w_1(I_{1i} - I_{1j})^2 + w_2(I_{2i} - I_{2j})^2 + w_3(I_{3i} - I_{3j})^2} \tag{4}$$

Equation (5) shows how SLIC combines spatial and intensity distances together to create its distance measure, which depends on the compactness parameter, *m*, and the initial superpixel grid spacing parameter, $S = \sqrt{N/k}$, where *N* is the number of pixels in the image and *k* is the initial number of superpixels.

$$D_{ij} = d_{Iij} + \frac{m}{S} d_{sij} \tag{5}$$

Because, the *k*-means algorithm can produce different cluster results with different cluster center initializations [19], SLIC deterministically initializes cluster centers to local minima of the intensity gradient. Figure **3**a) shows a conceptual representation of intensity gradient contours. Black dots represent the pixel data locations and bold red **x**s indicate the initial cluster centers located at two local gradient minima. After initialization, SLIC ignores the intensity gradient. It computes distances between every pixel and every cluster center within a local 2Sx2S area. Figure **3**b shows that, for each pixel, it finds the closest local cluster center and assigns the pixel to it. The algorithm iterates between assigning pixels to closest clusters and updating cluster means. The algorithm iterates until the change in cluster center means falls below a preset threshold. Initializing cluster centers to the local intensity gradient facilitates speedy convergence, as does confining each search to a local area. Computational complexity of SLIC is *O(N)* [2].

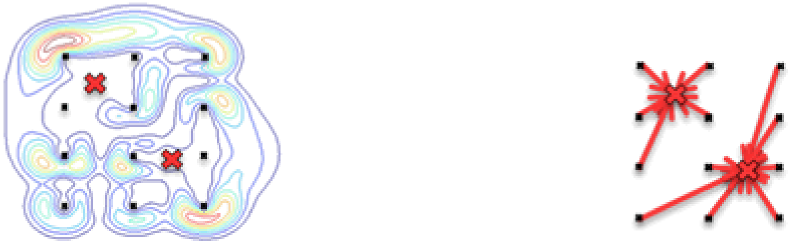

Figure **3**a) Diagram of intensity gradient with initial cluster centers located at local minima

Figure **3**b) Diagram of finding closest local cluster center in local *k*-means.

Figure 3. Diagrams of SLIC local k-means search with minimum intensity gradient initialization.

3.3. Applying SPS algorithms and measuring superpixel quality

Table 2 summarizes the SPS algorithms, parameter sets and SAR product inputs we compare in this paper. We discovered that Quick-shift superpixels were less compact and less accurate than their SLIC counterparts. In addition, it was difficult to find Quick-shift parameters that created smaller superpixel sizes. We conclude that the Quick-shift algorithm is sensitive to subtle image variations, which are likely a result of residual speckle. Thus, we include only one test using Quick-shift algorithm. Furthermore, we conducted a parameter study for the SLIC algorithm to determine which parameter set yielded superpixels that provide a reasonable compromise between fidelity in representing the structures in the scene and sufficient size to provide accurate statistical estimates for later classification and reasonable computational complexity.

This paper reports results of three parameter sets, P1, P2 and P3, which are defined in lines 2 through 4 of the table. Because parameter set P2 provided the best compromise, we use it for the comparing superpixels derived from different image product inputs. MLMC indicates a combination of SA-ML and MCCD product inputs. MLMR indicates a combination of ML and MRCS inputs. We also included interferometric height (IF-H) map data in two of the tests. While IF-H is not a low-speckle product, it does contain valuable information about local height variations, which could provide augment the backscatter and coherent change information. To avoid IF-H dominating the superpixel segmentation, we assign IF-H weights that are eight times smaller than the other products. MLIF indicates a combination of SA-ML and IF-H product inputs, MLMCIF indicates a combination of SA-ML, MCCD and IF-H product inputs.

Table 2. SPS algorithms, parameters and SAR product inputs for quality comparison.

SAR Product Input	**SPS Algorithm**	**Parameter Set**	**# of superpixels**	**Avg. superpixel size (pixels)**
SA-ML	Quick-shift	σ=8, *W*=10	1964	8579
SA-ML	SLIC	P1: S=50K, m=10	51894	325
SA-ML	SLIC	P2: S=25K, m=10	24914	676
SA-ML	SLIC	P3: S=10K, m=10	10304	1635

MRCS	SLIC	P2	24967	675
MCCD	SLIC	P2	24137	698
MLMC	SLIC	P2	24101	699
MLMR	SLIC	P1	24976	675
MLIF	SLIC	P2	25075	672
MLMCIF	SLIC	P2	24440	689

To judge the quality of superpixels produced by these options, we selected two widely-used segmentation quality measures, undersegmentation error and boundary recall [4]. Both quality measures require a hand-labeled ground truth image as input. To compute undersegmentation error, we extract every superpixel that overlaps each ground truth segment. We compute the superpixel area that falls outside the hand-labeled boundary divided by the total area of extracted superpixels. We estimate average undersegmentation error for all ground truth segments. To compute boundary recall, we extract the edges of the hand-labeled segments and the superpixel edges. We shift the edges both horizontally and vertically with respect to one another and compute the overlap between the two sets of edges at each shift value. Because horizontal and vertical differences contribute equally to overall quality, we sum the horizontal and vertical overlaps for each shift value and produce a one-dimensional plot of overlap versus shift. The area under the curve is a measure of the accuracy of placement of the superpixel edges with respect to the hand-labeled edges. Larger areas correspond to more accurate edge placement. We applied each test listed in Table 2 to the full SAR image. Section 4 shows small sections of the image so that we can qualitatively judge the quality of the resulting superpixels. Section 5 reports quantitative quality results for the whole image.

4. QUALITATIVE RESULTS

Figure **4** shows an MRCS image of a drainage ditch next to the edge of a road. Figure **4**b) shows Quick-shift superpixels, and Figure **4**c) shows SLIC superpixels. We see that SLIC produces more compact superpixels that conform more readily to the shape of the ditch. The superpixels are also more uniformly sized, which can be an advantage for making statistical comparisons. This image provides a particularly challenging test because the ditch has low contrast compared to its background. We test the ability of improving the superpixels by including height information into the SPS segmentation. **Figure 4**d) shows the resulting superpixels when we use both SA-ML and IF-H products at the input of the SLIC algorithm. Adding the height information improves the ability of the superpixels to represent two of the three arms of the drainage ditch.

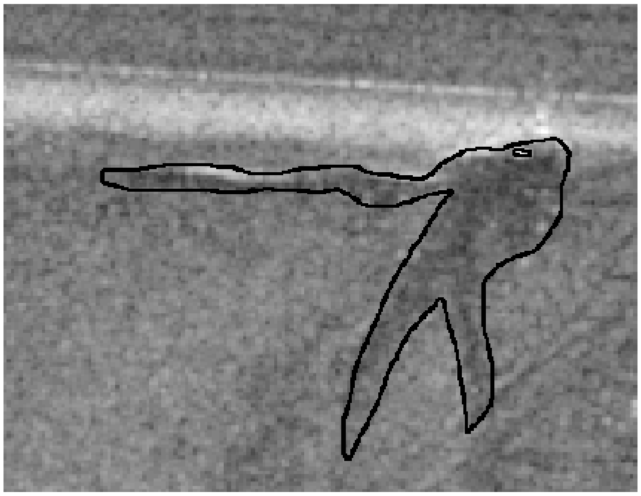

Figure 4a) MRCS image of drainage ditch outlined by hand-labeled ground truth

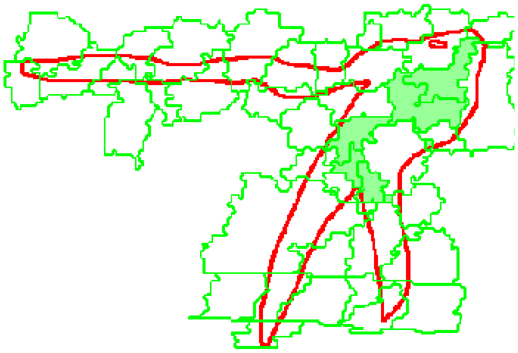

Figure 4b) Quick-shift superpixels

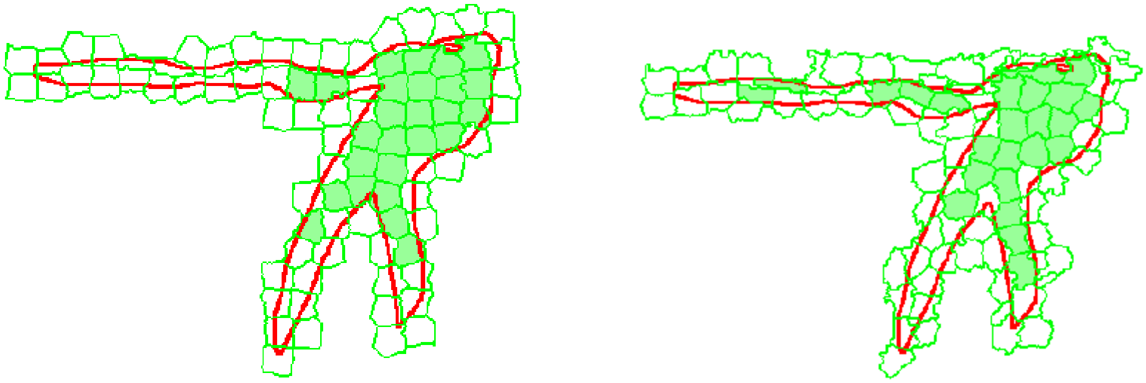

Figure 4c) SLIC superpixels with P2 and SA-ML

Figure 4d) SLIC superpixels with P2 and SA-ML & IF-H

Figure 4. Comparing superpixels to hand-labeled ground truth for image of a drainage ditch

Figure **6** show results of applying SLIC with different parameter sets to an image of a vineyard with a fence. The smaller superpixels on the left accurately represent the rows and fence. The larger superpixels on the right can represent most of the rows, but have multiple instances where several rows are combined.

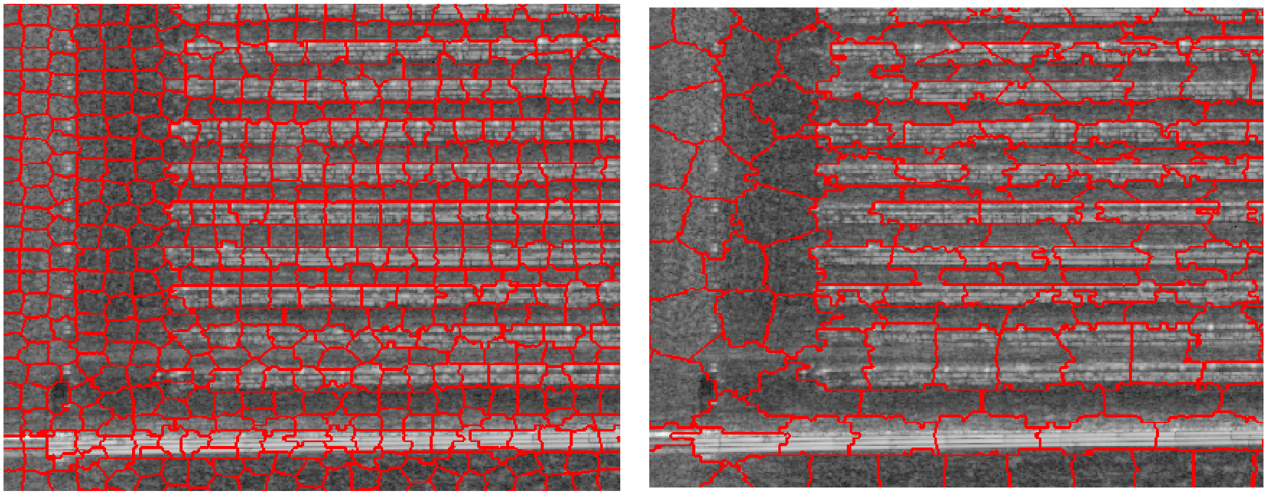

Figure **5**a) SPS with SLIC, P1 and SA-ML.

Figure **5**b) SPS with SLIC, P3 and SA-ML.

Figure 5. Superpixels of vineyard and fence.

Figure **6** compares superpixels derived with different SAR product inputs. Figure 6a) shows that the MRCS-derived superpixels have more wriggly edges than the SA-ML-derived superpixels shown in Figure **5**. Because the SA-ML contains lower spatial frequencies, it yields superpixels with smoother edges. This can be an advantage when we want to represent the edges of man-made structures. Fortunately, we can use SA-ML-derived superpixels to process data from any of the other image products, as long as they are co-registered. For example, if we do not have multiple images from the same site to compute a temporal multilook MRCS, we can still create superpixels from an SA-ML image and then apply them to a co-registered RCS image. In another example, we can compute superpixel histograms from higher resolution MRCS pixels located inside SA-ML superpixels. Such histograms can be useful for classifying surfaces in the SAR image.

Because the MCCD product represents different physical phenomena than the RCS-based products, its corresponding superpixel segmentation, shown in Figure 6b), emphasizes different structural features. The MCCD superpixels contort their boundaries to conform to the local variability of the MCCD image. Figure 6c) shows superpixels derived from combining SA-ML and MCCD together in two channels of the SLIC algorithm. SA-ML and MCCD are scaled to similar ranges and equally weighted. Figure 6d) shows superpixels derived from combining SA-ML, MCCD and IF-H in three channels at the input of SLIC. Adding the additional SAR products produces only subtle changes in the superpixels. Thus, unless we have reason to suspect that the different data products provide an advantage for representing a particular image structure, such as adding height to find a drainage ditch, adding multiple products at the input of the SPS does not provide a significant advantage in the resulting superpixels.

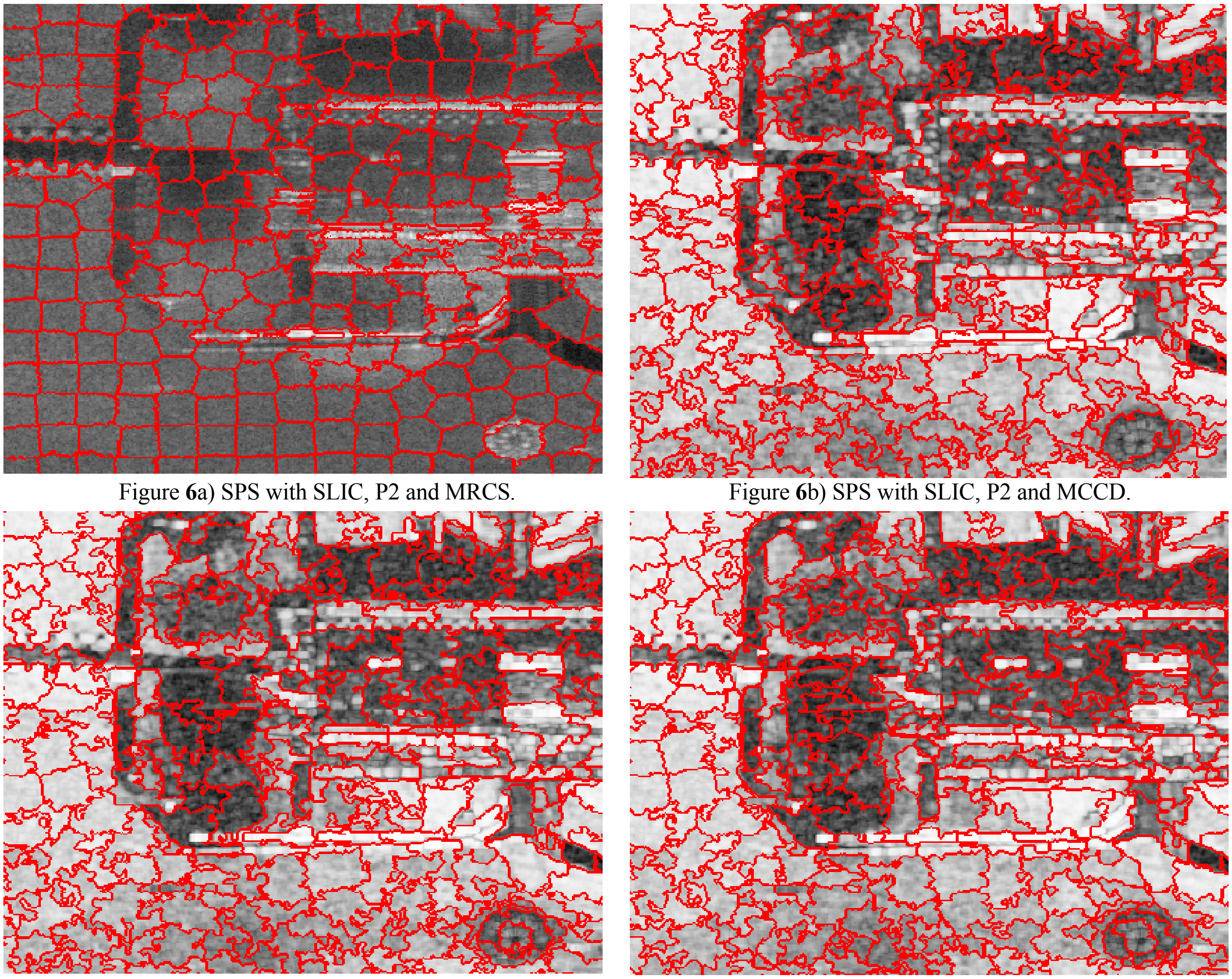

Figure **6**a) SPS with SLIC, P2 and MRCS.

Figure **6**b) SPS with SLIC, P2 and MCCD.

Figure **6**c) SPS with SLIC, P2 and SA-ML & MCCD.

Figure **6**d) SPS with SLIC, P2 and SA-ML, MCCD & IF-H.

Figure 6. Superpixels of building and playground.

5. QUANTITATIVE RESULTS

We computed both undersegmentation error and boundary recall for the ten quality comparison experiments listed in Table 2. Because the two quality measures produce similar conclusions, we only report boundary recall results in this paper. Figure 7a) shows that SLIC produces better boundary recall than that produced by Quick-shift. Figure 7b) shows that adjusting the SLIC parameters allows one to directly control the number and average size of superpixels. (Table 2 designates parameter values and average superpixel sizes that correspond to sets P1, P2 and P3.) Smaller superpixel sizes produce higher boundary recall and superpixels that conform better to structures shapes in the image, but they require more computation because the algorithm must compare each pixel against a larger number of local cluster centers. Because SLIC allows direct control of superpixel size, it allows the operator to control the trade-off between fidelity, computational cost and statistical representation. Figure 7c) and Figure 7d) shows that the type and number of input product(s) have only small effects on the global superpixel quality. ML-, MCCD- and MRCS-derived superpixels produce similar boundary recall. Combining SAR products at the SPS input, such as ML & MRCS, ML & MCCD, SA-ML & IF-H and SA-ML & MCCD & IF-H, also produces only small differences in the boundary recall measurement. All overall boundary recall results fall below 0.5, which reflects the difficulty of the test image which contains low contrasting structures and terrains.

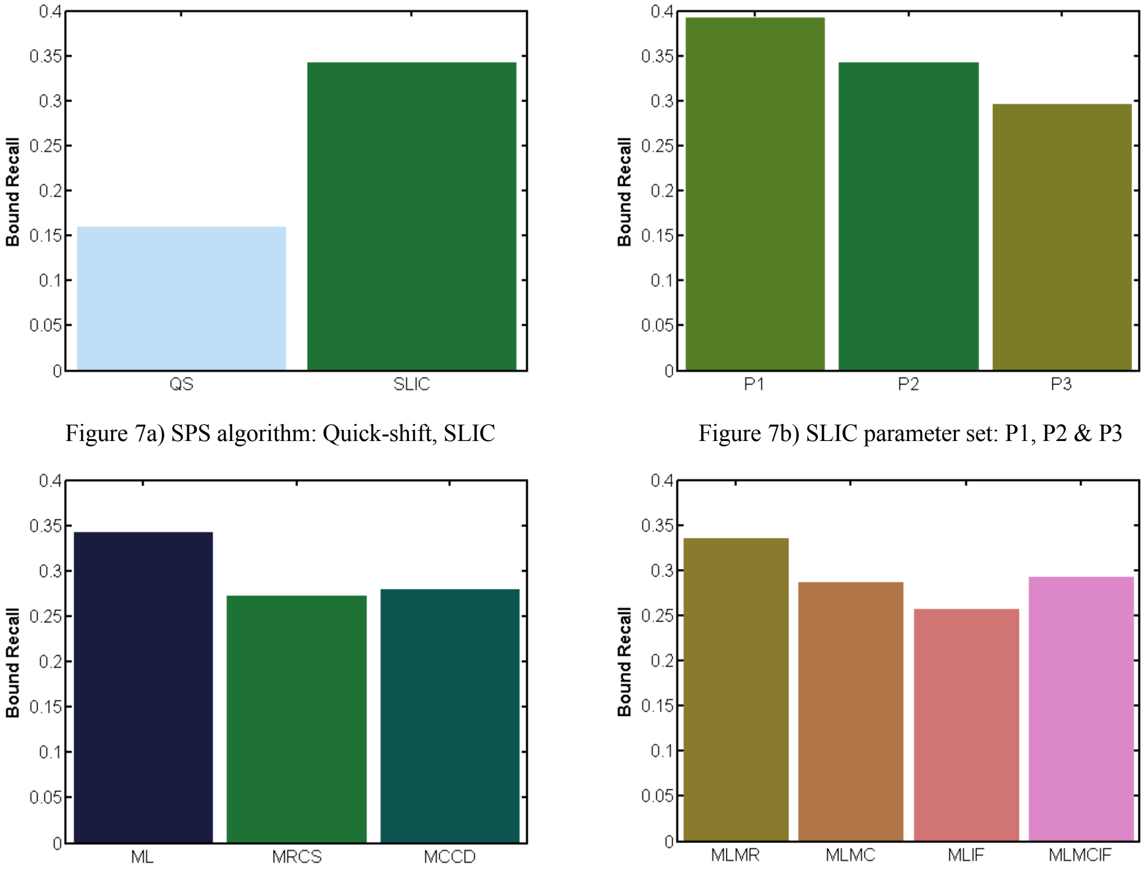

Figure 7a) SPS algorithm: Quick-shift, SLIC

Figure 7b) SLIC parameter set: P1, P2 & P3

Figure 7c) SLIC single-product inputs: ML, MRCS, MCCD

Figure 7d) SLIC multiple-product inputs: MLMR, MLMC, MLIF, MLMCIF

Figure 7. Boundary recall for superpixel quality comparison.

We also compared the computational speed of algorithm execution. We compiled publically available C++ versions of both Quick-shift and SLIC running on a six-core Intel Xeon 2.9 GHz processor. We tested the execution speed for generating superpixels on a 3465 x 3443 image. Processing time for Quick-shift with complexity $O(N^2)$ was 109 min, and processing time for SLIC with complexity $O(N)$ was 12 sec.

6. CONCLUSIONS AND FUTURE WORK

We have shown that it is possible to apply widely-used SPS algorithms, developed for optical imagery, to speckle-reduced SAR image products. Taking the logarithm of SA-ML and MRCS before applying SLIC with a difference-based distance measure reduces SPS sensitivity to multiplicative contamination. The SLIC algorithm produces higher quality superpixels than does the Quick-shift algorithm for SAR product images. Furthermore, SLIC is computationally faster than the Quick-shift algorithm. SA-ML-derived superpixels have straighter edges and produce slightly better segmentations than other products. We can use superpixels derived from one product to process data from any other co-registered product, which could include an individual RCS magnitude. Thus, this approach does not depend on having temporal multilook data available. The SLIC algorithm parameters directly control the number and average size of resulting superpixels to allow an operator control of trade-offs between superpixel fidelity, computational cost and statistical representation. Combining multiple products as multi-channel inputs to SLIC does not reduce global superpixel errors, but it could provide more flexible segmentations for specific objects, such as including height information to create superpixel segments for the drainage ditch.

We will apply SA-ML-derived superpixels to develop a terrain/structure classifier for SAR imagery. These superpixels provide small local self-similar regions for estimating backscatter statistics and can significantly reduce computational requirements for a classification algorithm.

ACKNOWLEGEMENTS

The authors gratefully acknowledge Andrea Vedaldi, Stefano Soatto and Radhakrishna Achanta, et.al. for providing publicly available code for Quick-shift and SLIC algorithms, respectively. The authors would also like to thank Jeremy Goold for producing and co-registering the SAR image products used in this paper and Matt Stosnick for introducing us to the utility of Quick-shift superpixels.

REFERENCES

1. Fulkerson, B., A. Vedaldi, and S. Soatto. *Class segmentation and object localization with superpixel neighborhoods*. in *Computer Vision, 2009 IEEE 12th International Conference on*. 2009. IEEE.

2. Achanta, R., et al., *SLIC superpixels compared to state-of-the-art superpixel methods.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2012. **34**(11): p. 2274-2282.

3. Ren, X. and J. Malik. *Learning a classification model for segmentation*. in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. 2003.

4. Levinshtein, A., et al., *Turbopixels: Fast superpixels using geometric flows.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2009. **31**(12): p. 2290-2297.

5. Su, X., et al., *A supervised classification method based on conditional random fields with multiscale region connection calculus model for sar image.* Geoscience and Remote Sensing Letters, IEEE, 2011. **8**(3): p. 497-501.

6. Wu, Y., et al., *Fast algorithm based on triplet Markov fields for unsupervised multi-class segmentation of SAR images.* Science China Information Sciences, 2011. **54**(7): p. 1524-1533.

7. Gan, L., et al., *Triplet Markov fields with edge location for fast unsupervised multi-class segmentation of synthetic aperture radar images.* IET Image Processing, 2012. **6**(7): p. 831-838.

8. Yu, H., et al., *Context-Based Hierarchical Unequal Merging for SAR Image Segmentation.* Geoscience and Remote Sensing, IEEE Transactions on, 2013. **51**(2): p. 995-1009.

9. Xiang, D., et al., *Superpixel Generating Algorithm Based on Pixel Intensity and Location Similarity for SAR Image Classification.* Geoscience and Remote Sensing Letters, IEEE, 2013. **10**(6): p. 1414-1418.

10. Caves, R., Q. Shuan, and R. White, *Quantitative comparison of the performance of SAR segmentation algorithms.* Image Processing, IEEE Transactions on, 1998. **7**(11): p. 1534-1546.

11. Fjortoft, R., et al., *An optimal multiedge detector for SAR image segmentation.* Geoscience and Remote Sensing, IEEE Transactions on, 1998. **36**(3): p. 793-802.

12. Touzi, R., A. Lopes, and P. Bousquet, *A statistical and geometrical edge detector for SAR images.* Geoscience and Remote Sensing, IEEE Transactions on, 1988. **26**(6): p. 764-773.

13. Vedaldi, A. and S. Soatto, *Quick shift and kernel methods for mode seeking*, in *Computer Vision–ECCV 2008*. 2008, Springer. p. 705-718.

14. Wu, Y., et al., *Region-Based Classification of Polarimetric SAR Images Using Wishart MRF.* Geoscience and Remote Sensing Letters, IEEE, 2008. **5**(4): p. 668-672.

15. Jakowatz, C.V.J., et al., *Spotlight-mode Synthetic Aperture Radar: A Signal Processing Approach*. 1996, Norwell, MA: Kluwer Academic Publishers.

16. Achanta, R., et al., *SLIC superpixels.* École Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep, 2010. **149300**.

17. Parzen, E., *On estimation of a probability density function and mode.* Annals of mathematical statistics, 1962. **33**(3): p. 1065-1076.

18. Vedaldi, A., et al. *VLFeat*. 2012; Available from: http://www.vlfeat.org/about.html.

19. Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern classification*. 2012: John Wiley & Sons.