A computational framework for labeling spatiotemporal remote sensing datasets

Manu Sethi, Yupeng Yan, Anand Rangarajan, Ranga Raju Vatsavai[1], and Sanjay Ranka
Dept. of Computer and Information Science and Engineering,
University of Florida

1 Introduction

Remote sensing instruments and sensors have made significant progress over the last several decade in spatial, spectral and temporal resolutions. These improvements have led to the collection of synoptic scale data and enabled a variety of new applications. For example, improvements in temporal resolution allows monitoring biomass on a daily basis. Improvements in spatial resolution allows fine-grained classification of urban settlements, damage assessments, and critical infrastructure monitoring. Remote sensing applications have the following characteristics:

- Spatio-temporal Grid: The underlying data sets are gridded—with the grid dimensions ranging across 2D (images, 2D flow fields), 3D (volumetric data) and 4D (spatiotemporal data). Unlike standard machine learning approaches where the data is usually only available in the form of high dimensional feature vectors, the presence of a grid affords us the potential to develop techniques that can interpolate each of the features (using differentiable splines) and generate new feature vectors with different (and more desirable) nearest neighbor properties.
- Large Volume and Velocity: The underlying volume (terabytes to petabytes) and velocity (gigabytes to terabytes per day) of these applications is very large and are responsible for carrying us into the bigdata regime. Effective processing requires low complexity and multiscale algorithms that can exploit modern parallel architectures with deep memory hierarchies.
- Complex Queries: The underlying queries of interest are complex and require segmentation into cohesive regions; change detection; modeling spatial and temporal correlations; detection of rare events; and classification of each pixel into one of the k given labels (e.g. forest, water, urban). spatiotemporal linkages play a prominent role and potentially exhibit evolutionary dynamics rendering prediction and generalization even more difficult.
- Expert Interactivity: These techniques need to be able to incorporate domain expertise and expert involvement in the knowledge discovery process. Though supervised methods are the preferred form of analysis, getting ground-truth (training and/or labeling) data for large spatiotemporal extents is impractical in several application domains. Thus, techniques have to be semi-supervised to address the fact that expert labels may only be available for a small portion of the data.

In this paper we provide a framework for semi-supervised labeling of regions (urban, slums, forests, sea, sand) in high resolution aerial images. Our approach apart from being capable of running in parallel is fast enough despite the large volume of input data. The speed of our framework is realized through clustering the data into oversegmented regions called superpixels which have recently become popular in the computer vision community. Superpixels correspond to coherent patches or areas in 2D (or volumes and hypervolumes in 3D and 4D respectively). These coherent superpixels also reduce the data complexity since processing is moved to the superpixel level from the pixel level. This provides a huge benefit specially in the case of large volumes of data because dealing at per

1. *Oakridge National Laboratories (ORNL)*
. This work is partially supported by NSF IIS 1065081

pixel level can be overwhelming. Superpixels also have a huge advantage over partitioning the image into regular patches because regular patches ignore the local variability of the underlying data w.r.t. the grid. Superpixel estimation follows the path of partitioning a set of pixels using feature distances. In comparison to standard patch based approaches which use square or rectangular patches, superpixels in 2D can be expected to perform better since they take into account the local variability of features. For example, if we obtain 128 dimensional SIFT vectors [15] at each pixel, the resulting superpixels take into account the local variation of the set of SIFT vectors. Consequently, provided superpixel estimation can be made scalable and efficient, superpixel tessellation is an important first step in spatiotemporal semi-supervised learning.

Since the advent of Normalized cuts [20] and advances in Graph cut methods [11], [7], there has been a surge in the methods to decompose an image into superpixels. Ultrametric Contour Map (UCM) [2] is another such popular method which uses local and global cues to produce a hierarchy of tessellations at different scales ranging from fine to coarse. These tessellations respect the containment property, that is every finer scale tessellation is contained within the next higher (or coarser) scale tessellation. In this work we lean on the hierarchical output of UCM and leverage it in two different ways – (i) The first usage is more traditional and direct in the sense that a finer scale tessellation is obtained and used for classifying (or labeling) each superpixel (as against each individual pixel). (ii) This second usage of UCM is more subtle in the sense that we obtain the tessellation at a coarser scale and use it as a feature for classifying the superpixels at the finer scale selected in (i). This is because the tessellation at a coarser scale of UCM mostly picks up prominent boundaries thus increasing the probability of detecting urban regions. Distinguishing between different kinds of human settlements for example, slums vs urban, is a challenging task. The key feature that seems to distinguish the two classes is presence of stronger boundaries around the superpixels representing the slums. Traditionally used features like dense sift, HOG etc. can be used towards detecting these regions but these suffer from the problem of determining the appropriate scale and orientation. Further these features are also not able to pick up the prominent boundaries as detected by UCM tessellation at a coarse scales. Because UCM inherently involves a combination of dense features like textons and color histograms and only shows stronger boundaries at coarser scales, it greatly simplifies the task of discriminating the urban regions from the slums. The containment property of UCM further helps in percolating the features obtained at a coarser tessellation down to the all the superpixels of the finer tessellation.

Our classification and label propagation pipeline works by using superpixels to find various features and then uses the sparsely available groundtruth data from the expert to train a rudimentary classifier based on either k-nearest neighbors (kNN) or SVM. We then use Laplacian Propagation to refine the preliminary labels obtained in the previous stage. For both these approaches we build a graph with superpixels as the nodes. These nodes are connected in the feature space in the case of kNN and in the spatial domain in the case of Laplacian Propagation.

Roadmap: In the next section we discuss the related work in classifying remote sensing images using semi-supervised approaches. Section 3 describes our approach in detail. In this section we also discuss the graph theoretical approaches needed for Laplacian propagation and kNN. Section 4 provides experimental validation and section 5 concludes this paper after highlighting the future work.

2 Previous Work

Major steps involved in remote sensing image classification can be abstracted into: (i) extraction of features from the image, (ii) collection of ground-truth (training/test) data for few sample locations, (iii) building a classification model (e.g. naïve Bayes, decision trees, MLPs), and (iv) predicting labels for the entire image. Most of the existing classification approaches work with spectral features (e.g., blue, green, red, thermal infrared) and derived features (e.g., texture, band ratios like Normalized Difference Vegetation Index (NDVI), Histogram of Oriented Gradients (HOG)), extracted at each pixel (spatial location). These classification approaches are called pixel-based or single instance learning (SIL) algorithms. A review of these techniques can be found in [25], [13]. Most classification schemes model the correlations in feature space and often ignore spatial correlations in the image space. An improvement over per-pixel classification schemes are the spatial classification schemes such as MRF [19]. In spatial classification schemes both spatial correlations (context) and feature correlations are modeled simultaneously, as a result the final classified image contains much smoother (spatially) class distributions and eliminates salt and pepper noise. However, it should be noted that spatial classification methods are also essentially single instance learners. One way to overcome single instance limitation is to look at additional features beyond spectral features, because features that exploit spatial contextual information are highly useful in the classification of very high-resolution images. Recent studies [25], [22], [13] show the improved performance of SIL methods when the spectral features are combined with a broad set of extended features such as morphological, texture, and edge density. Although these studies showed that the extended features which exploit spatial contextual information resulted in improved the SIL accuracy, the underlying image complexity and interpixel relationships are still not fully exploited.

Complex object recognition requires investigation of spatial region or image patch. Object based classification schemes [18], [5] seek to segment the image into meaningful objects by exploiting spatial and spectral features. One can build a meta classifier on the features extracted from the objects, for example, area, perimeter, compactness, shape index, and fractal dimension. Or one can aggregate all feature vectors into a single feature vector and then apply any single instance learning algorithm. However, all these approaches lose important structural and spatial properties in the aggregation process. Multi-instance (or Multiple instance) learning (MIL) methods have been developed to overcome some of the limitations of single instance learning schemes. Notable approaches include the seminal work of Dietterich et. al. [10], Diverse Density [16], and Citation-KNN [26]. Recently, MIL algorithms have also been applied to remote sensing image classification as well. For example, in [6], authors have developed an MIL based binary classification scheme for identifying targets (landmines) in Hyperspectral (HS) imagery. While each of these algorithms have advantages and disadvantages over per-pixel based classification schemes, in general they are shown to perform (accuracy) better than single instance learning schemes. In MIL, the training data consists of many bags (image patches) where each bag contains several examples (pixels). A bag is positively labeled if it contains at least one positive instance (e.g., informal settlement) and negative otherwise (e.g., formal settlement). In a recent feasibility study Citation-KNN algorithm was applied for complex settlement mapping [24]. The high computational cost of Citation-KNN has lead to the development of an efficient Gaussian Multiple Instance (GMIL) [23] learning algorithm. Both of these algorithms are shown to perform better than most well-known SIL approaches, however leveraging them for global scale problems is difficult due to their complexity. We believe that our work which utilizes irregular patches or superpixels (which are mainly homogeneous) along with novel and parallelizable machine learning techniques have the potential to address the scale requirements of target applications.

Finally, we summarize the evolution—mainly in the past decade—of graph-based semi-supervised learning methodologies. Note that there is no general literature of graph-based SSL for gridded data. Early work on SSL focused on optimization [9] and relationships to transductive inference [14], [21] and multi-view learning [17]. Since then, the use of graphs in SSL has become standard [12] (while the extension to multivariate graphs for gridded data is not). Graph-based SSL methods attempt to assign node labels using a weighted combination of the neighbors. Different methods use different principles to design objective functions for label propagation. For example, a popular approach [27] iterates a function of the graph affinity matrix until convergence and then uses the sign of the function at each node. A different method adapts the Jacobi iteration for linear systems and obtains a somewhat different weighted combination subsequently used for prediction. Other influential methods [3] use regression to determine the weighted combination. First, they compute the graph Laplacian followed by eigenvector computation. Then a regression objective estimates a weighted combination of the principal eigenvectors on the training samples which is utilized for prediction at the unlabeled nodes. Other methods draw upon random walks on graphs (related to Markovian transition probability estimation) to perform label prediction [8].

3 Approach

We have motivated the importance of adaptive patch generation which respects data regularities prior to construction of the patch graph. Since the decomposition of the data into irregular (but coherent) patches or tessellations using "superpixels" is so crucial for spatiotemporal SSL, we begin by describing our approach to this problem. This is followed by a rudimentary classification using either SVM or kNN and finally smoothing the labels using Laplacian Propagation. Because kNN and Laplacian Propagation are based on a graph datastructure, we will also describe the construction of a superpixel graph with multivariate edges corresponding to the spatiotemporal (grid) and feature-based metrics. The nodes of the graph are the superpixels with the edges being the connections between superpixels. In spatiotemporal processing, multivariate edges comprise spatiotemporal edges (type 1) and feature-based edges (type 2). The latter are obtained from a superpixel similarity metric wherein comparisons are undertaken between pairs of superpixels. To achieve scalability, the number of such edges would be significantly lower than $O(M^2)$ for M superpixels. We develop fast and parallel approaches for implementing the graph-based SSL techniques. The details are provided in the following subsections.

3.1 Superpixel formation

We use UCM [2] to decompose the image into superpixels at multiple scales. UCM outputs a hierarchical structure with dense superpixels at the finest level which gradually merge into one big superpixel for the whole image at the topmost (coarsest) level. At all other intermediate levels in UCM, the superpixels follow the containment property, that is the superpixels at a finer level are contained within the superpixels at the coarser level. In our approach we leverage this hierarchical structure to obtain a fine and a coarse tessellation of the 2D image by thresholding the UCM at two different scales. The finer tessellation is used for classification and label propagation as described in the next section. The coarser tessellation is used to obtain those superpixels which predominantly distinguish urban regions. This is because a coarser tessellation outputs superpixels with prominent boundaries which are more likely to belong to the urban regions.

3.2 Superpixel descriptor

As mentioned above we obtain superpixel tessellations at two levels – finer and coarser. Because in our framework only the finer scale superpixels are used for classification and label propagation, we only need to provide feature descriptors for superpixels obtained at the finer scale. The coarser scale superpixels in turn act as features to describe the finer superpixels as we will explain here.

Each superpixel at the finer level is described using three kinds of features – intensity histograms, corner density and a binary feature derived from the coarser level of UCM. For the intensity histograms, we quantize the grayscale intensities into 52 bins and obtain a 52 dimensional feature vector for each superpixel. For obtaining the corners we use Harris corner detector and obtain the density as the number of corners per unit area for each superpixel. Corners act as an important feature in discriminating between regions with buildings (for example, slums and urban area) and regions without (for example, forest and sea). The coarser scale UCM provides a binary feature for each finer superpixel as follows. The coarser scale UCM only keeps prominent boundaries and therefore outputs much larger superpixels. Among them, the superpixels which are smaller than a certain size threshold predominantly belong to urban regions. This is because urban regions are usually found with stronger boundaries and hence are more discriminating. The superpixels which are much larger than the size threshold are more likely to be a merger of several different types of smaller superpixels and are often not much discriminating. For example, these superpixels can be a merger of slums and forests or other similar looking regions which do not have as clearly demarcating boundaries as the urban regions have. So to get the binary feature we label the superpixels below the chosen size threshold with ones and the superpixels above this threshold with zeroes. These binary features are then percolated down the UCM hierarchy to the finer scale superpixels. All the finer superpixels contained in the larger superpixels get the same label as that of the larger superpixel they are contained in.

These three different kind of features are then concatenated to form a 54 dimensional feature vector which describes each superpixel of the finer tessellation. Other features like HOG and dense SIFT can also be added to the above framework but for simplicity we only chose the above three features. Further, the coarser level feature derived from UCM itself incorporates textons and size of superpixels as implicit features in arriving at the binary descriptor.

3.3 Building a Patch Graph with Multivariate Edges

Before we move on to describe the semi-supervised learning stage of our pipeline, we discuss methods used to create graph data structures. This is because we employ k-nearest neighbors (kNN) and Laplacian propagation (described in the next section) both of which rely on Graphs. We use graphs with a patch or superpixel at each node with edges connecting superpixels that exhibit strong similarity to each other.

Spatio-temporal datasets have two metrics: (i) pixel and space-time metric (in the case of Laplacian propagation) and the (ii) feature vector metric (in the case of kNN). The pixel space-time metric imposes locality while the feature metric allows distant features to be similar. Thus, similarity between objects can be measured using either of these two metrics generating multivariate edges. Instead of using pairwise comparison that requires quadratic time, whenever possible efficient derivation of these edges is done using k-d trees [4].

3.4 Semi-supervised learning on the Patch Graph

Given the large size of the underlying datasets it is impractical to expect that the ground truth is available except at a small number of grid points since data sets scale but experts do not. Thus, practical approaches have to be semi-supervised (as opposed to supervised or unsupervised) with focus restricted on methods with proven scalability. In this work we achieve SSL through a two stage process of (i) classification using either SVM or kNN followed by (ii) Laplacian smoothing. Our classification pipeline is similar to [1]. As mentioned above the ground truth data is available only for a small number of superpixels as labeled by experts. We use this ground truth to train our classifier. For training the SVM we used a linear kernel and for training the kNN we used the nearest neighbors in the feature space based on the k-d tree method. The model obtained from training either the SVM or the kNN is then used to determine preliminary labels for all other superpixels.

3.5 Laplacian propagation

Because of the semi-supervised nature of the problem, the classification obtained from above is rudimentary because it based on a classifier (SVM or kNN in our case) derived from limited groundtruth data. This classification can lead to artifacts such that neighboring regions which belong to the same class may get labeled incorrectly. To correct this problem and after being inspired by [1] we apply Laplacian propagation method as detailed here.

Let f_i denote the feature vector corresponding to the i^{th} superpixel and let X_i be the label that is required to be found from the Laplacian propagation. Let Y_i be the initial label as obtained from the first stage of either SVM or kNN. To perform Laplacian propagation we construct a graph connecting adjacent superpixels in the spatial domain (and not in the feature domain). The edge weight is given as $W_{ij} = \exp\left(-\frac{\|f_i - f_j\|^2}{2\tau^2}\right)$. Our goal is to minimize the following objective function [1] :

$$C(X) = \sum_{i,j=1}^{N} W_{ij} \left| \frac{X_i}{\sqrt{D_{ii}}} - \frac{X_j}{\sqrt{D_{jj}}} \right|^2 + \sum_{i=1}^{N} \lambda \left| X_i - Y_i \right|^2 \tag{1}$$

where $D_{ii} = \sum_{j=1}^{N} W_{ij}$. The above equation 1 is optimized separately for each category in a one versus rest fashion. $Y_i = 1$ if the superpixel belongs to the category and 0 otherwise. X_i can take a real value and after solving equation 1 for each category, we assign each superpixel the category which corresponds to the maximum value of X_i. The above equation 1 can be directly minimized by solving as a linear system of equations [1]:

$$\left(I - (1 - \frac{\lambda}{1+\lambda}) S \right) X = \alpha Y \tag{2}$$

where $S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.

3.6 Parallelization

The above steps can be easily parallelized by decomposing the high resolution image into smaller pieces and assigning each piece to a different compute node in a distributed system. We obtain UCM for each piece separately and obtain superpixel features from each image. For generating the SVM or kNN classifier we only use one node since only a handful of samples are labeled by the expert. We then distribute this SVM or kNN model to all other nodes in order to classify superpixels in all the image pieces. We next use Laplacian Propagation on each of the sub-images to smooth out the labels.

At this point we have divided the large image data into smaller sub-images and have run UCM and Laplacian propagation individually on each of the sub image. Our future work will focus on parallelizing UCM and Laplacian propagation to run on a distributed system.

4 Experimental results

We now briefly describe an example of superpixel tessellation on a settlement mapping application. The grayscale image of Rio, Brazil is publicly available from DOE (with the color version not currently available in the public domain). This image is of size 10,000 by 10,000 pixels. We have chosen a portion of the image of size 3000 by 3000 to showcase the key ideas. This portion of the image consists of a combination of favelas, forests, urban territories, sea, and sand. We performed Gaussian smoothing followed by UCM [2]. The support of the Gaussian filter was chosen to be a 10 pixels wide square and the standard deviation was set to 15. The resulting tessellation (with appropriate UCM threshold) consisted of 10452 patches [Figure 1b(a)]. This represents a size reduction of a factor of around 1000. Each of the patches represent coherent portions of the image. We note in passing that heavy smoothing of the image as mentioned above was only done for computing UCM. Once the UCM was obtained the original image was used at every other stage of the pipeline for example in computing features or for classification.

Intensity histograms, corner density and coarser scale UCM features were extracted for each of the superpixels. The last two features were weighted by a factor of 100 and concatenated together with the intensity histogram to obtain a 54 dimensional feature vector describing each superpixel. We provided ground truth labels to only 0.24% of the patches to train the SVM. The SVM classifier was used to obtain the preliminary classification [Figure 1c] for all the other superpixels. This was then given as an initialization to the Laplacian Propagation algorithm in order to obtain the final labels [Figure 1d]. The values of τ and λ were kept fixed to be 2 and 0.125 respectively. Additionally, the time required for the overall processing (including UCM) was less than twenty minutes and this processing can essentially be conducted in parallel for different sub-images of size 3000x3000 assigned to each core.

5 Discussion

We have provided a framework to classify different land use regions in spatio-temporal aerial image data. Current results do not showcase the temporal component, however, our work is easily extensible to temporal data as well and is the focus of our future work. Further, our current parallelization only deals with dividing the data into different pieces and work on each piece individually on each compute node. Our immediate future work is to distribute the superpixels on different nodes and run SVM and graph Laplacian algorithm in parallel over a distributed architecture. Further, we also aim to implement UCM to run in parallel by employing the use of parallel watershed transform algorithms. Below we highlight the contributions of this paper in an itemized format.

- Determined the appropriate scale factors to obtain coarse and fine tessellations from UCM.
- Obtained features which depict regions surrounded by strong boundaries by combining the coarse scale UCM tessellation with the size of superpixels.
- Developed features which can discriminate between urban, slums, and forest regions.
- Developed a fast SSL approach based on SVM, kNN and Laplacian Propagation.

References

[1] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 811–818.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[3] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Machine Learning*, vol. 56, no. 1-3, pp. 209–239, 2004.

[4] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[5] T. Blaschke, S. Lang, and G. Hay, *Object-Based Image Analysis*. Springer, 2008.

[6] J. Bolton and P. Gader, "Application of multiple-instance learning for hyperspectral image analysis," *Geoscience and Remote Sensing Letters, IEEE*, vol. 8, no. 5, pp. 889 –893, sept. 2011.

[7] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International journal of computer vision*, vol. 70, no. 2, pp. 109–131, 2006.

[8] O. Chappelle, B. Schölkopf, and A. Zien, Eds., *Semi-supervised learning*, ser. Adaptive Computation and Machine Learning. The MIT Press, 2010.

[9] A. Demiriz and K. Bennett, "Optimization approaches to semisupervised learning," in *Complementarity: Applications, Algorithms and Extensions*. Springer, 2001.

[10] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, no. 1-2, pp. 31–71, 1997.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[12] A. B. Goldberg, X. Zhu, and S. J. Wright, "Dissimilarity in graph-based semi-supervised classification," in *AISTATS*, 2007, pp. 155–162.

[13] J. Graesser, A. Cheriyadat, R. Vatsavai, V. Chandola, J. Long, and E. Bright, "Image based characterization of formal and informal neighborhoods in an urban landscape," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 5, no. 4, pp. 1164 –1176, Aug. 2012.

[14] T. Joachims, "Transductive learning via spectral graph partitioning," in *Intl. Conf. Machine Learning*, 2003, pp. 290–297.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

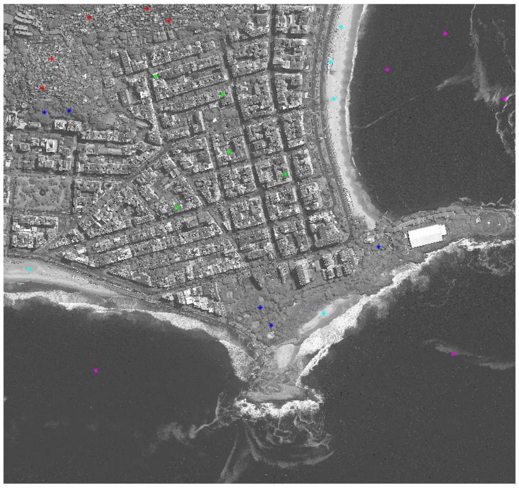
[16] O. Maron and T. Lozano-Perez, "A framework for multiple-instance learning," in *Advances In Neural Information Processing Systems*. MIT Press, 1998, pp. 570–576.

[17] I. Muslea, S. Minton, and C. A. Knoblock, "Active + semi-supervised learning = robust multi-view learning," in *Intl. Conf. Machine Learning*, 2002, pp. 435–442.

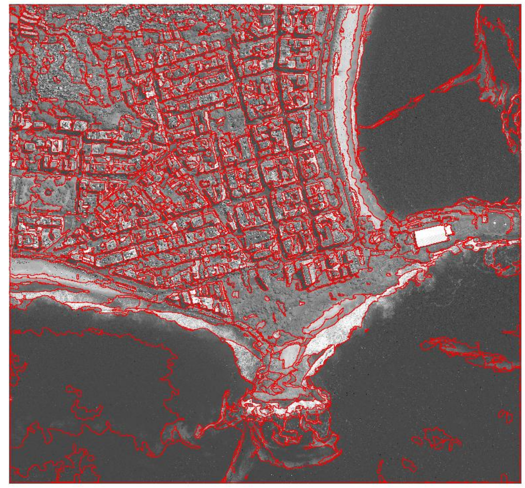
[18] S. Nussbaum and G. Menz, *Object-Based Image Analysis and Treaty Verification*. Springer, 2008.

[19] S. Shekhar, P. Schrater, R. Vatsavai, W. Wu, and S. Chawla, "Spatial contextual classification and prediction models for mining geospatial data," *IEEE Transaction on Multimedia*, vol. 4, no. 2, pp. 174–188, 2002.

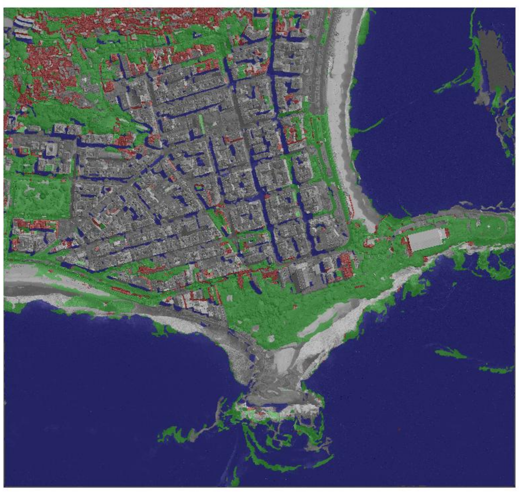
[20] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.



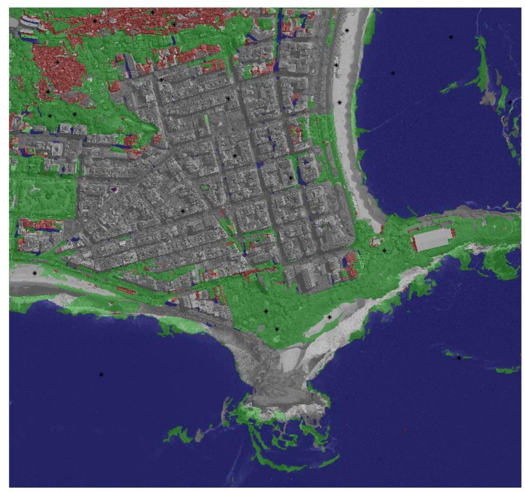
(a) Original 3000 x 3000 subimage consisting of favelas (slums), urban settlement, forests, sea, and sand. The handful ground-truth labels provided by the expert are also shown.



(b) Decomposition into superpixels after running UCM



(c) Rudimentary classification obtained by SVM. The 5 categories are overlayed on the original image in different colors. Slums are shown by red, forests by green, urban by gray, sea with blue and sand shown as white.



(d) The final result after Lapalcian Propagation. Some superpixels inside urban regions which were incorrectly marked as sea in the previous figure 1c are not correctly labeled as urban. Similar improvements can be noticed for other categories and regions.

Figure 1

[21] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: from transductive to semi-supervised learning," in *Intl. Conf. Machine Learning (ICML)*, 2005, pp. 824–831.

[22] R. R. Vatsavai, "High-resolution urban image classification using extended features," in *ICDM Workshops*, 2011, pp. 869–876.

[23] ——, "Gaussian multiple instance learning approach for mapping the slums of the world using very high resolution imagery," in *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2013, pp. 1419–1426.

[24] R. R. Vatsavai, B. Bhaduri, and J. Graesser, "Complex settlement pattern extraction with multi-instance learning," in *Joint Urban Remote Sensing Event*, apr. 2013, pp. 1 –4.

[25] R. R. Vatsavai, E. A. Bright, V. Chandola, B. L. Bhaduri, A. Cheriyadat, and J. Graesser, "Machine learning approaches for high-resolution urban land cover classification: a comparative study," in *COM.Geo*, 2011, p. 11.

[26] J. Wang, "Solving the multiple-instance problem: A lazy learning approach," in *In Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 1119–1125.

[27] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.