

Effective Hyper-Spectral Image Segmentation Using Multi-scale Geometric Analysis

Ofer Levi¹, Shaul Cohen¹ and Ziv Mhabary¹

¹Department of Industrial Engineering and Management, Ben-Gurion University of the Negev,
P.O.B 653, Beer-Sheva 84105, Israel
levio@bgu.ac.il, shaulco@bgu.ac.il, zivmha@bgu.ac.il

Abstract: The wide availability of multispectral images has fostered the development of new algorithms for remote sensing applications. These applications range from agricultural and environmental to military use. Nevertheless, the analysis of such voluminous data requires advanced analysis and computational methodologies as well as advanced hardware and computational methods. In this paper we introduce a new state-of-the-art method for segmentation of Hyperspectral images which uses both spectral and spatial information simultaneously. The proposed methodology is based on a multi-scale geometric transformation, called the Beamlet Transform, and the Beamlet Decorated Recursive Dyadic Partitioning (BD-RDP). The method is applicable for both mono-spectral and multispectral images where each pixel has its corresponding spectral profile vector.

The proposed segmentation method is especially effective when the underlying image consists of relatively large segments with smooth boundaries. In this case, it performs exceptionally well even when the Signal to Noise Ratio (SNR) is extremely low. The method is unsupervised and assumes no prior knowledge of the image characteristics or features. Furthermore, it involves a free sensitivity parameter which allows fine tuning for a specific application, and thus improving segmentation results. Despite of being relatively complex and sophisticated, the proposed segmentation algorithm has a low computational complexity of $O(N \log N)$. This is achieved by implicit computations through the Pseudo-Polar Fast Fourier transform (PPFFT). In order to validate the efficiency of our method, we have used the Lark algorithm which also combines spectral and spatial analysis but lacks the multi-scale property, for segmentation of multi-spectral images and compared its performance to the method proposed in this paper. These comparisons showed that our new proposed method outperforms the Lark algorithm and emphasized the effectiveness of multi-scale analysis. The proposed method was successfully applied to real aerial multi-spectral imagery for the application of estimating nitrogen levels in agricultural areas.

Keywords: Segmentation, Hyper-spectral images, Multi-scale geometric analysis, Beamlet transform, Radon Transform.

I. Introduction

Accurate image segmentation is one of the key problems in computer vision. Before high-level reasoning can be applied to an image, it must be broken down into its major structural components. With The advent of remote imaging spectroscopy [1], it is now possible to classify objects in a

scene based on their spectral properties [2]. There are different approaches for HS image segmentation, such as hierarchical segmentation [3], thresholding techniques [4], end-member extraction [5] and Lark (modified Fuzzy c-mean) [6][7]. All of them have shown good results. However, these types of algorithms are limited with their ability of deal with big amount of data [8]. Supervised methods are almost irrelevant when dealing with such a voluminous data, since the required training set causes problems related to the curse of dimensionality. Unsupervised methods like end-member and convex cone analysis suffers from high computational complexity and for large images the long computation time makes these methods impractical. Furthermore, hyperspectral images have two routes of information: spectral and spatial. Most available Hyperspectral data analysis techniques do not treat the data as a spatial image and focus only on the spectral point of view [9][10]. In the last years, there were several attempts to handle both spectral and spatial information. For instance, several possibilities are discussed in [11], where at the first stage spectral analysis is applied along the spectral dimension and at the second stage the image is spatially analyzed. Despite of combining both spectral and spatial analysis, it is not done simultaneously

The main goal of this paper is to present an effective, unsupervised method for segmentation of HS images which uses both the spectral and spatial information simultaneously. The proposed methodology is based on a multi-scale geometric transformation called the Beamlet Transform, first introduced in [12]. Despite of seeming complex and sophisticated, the proposed segmentation algorithm has a low computational complexity of $O(N \log N)$, Where N is the input image total size. This is achieved by implicit computations through the Pseudo-Polar Fourier transform (PPFFT) [13].

II. Background

A. Hyper/Multi-spectral segmentation

Classically, image segmentation is defined as the partitioning of an image into none overlapping, constituent regions which are homogeneous with respect to some characteristic, such as intensity or texture. If the image domain is denoted by I then the segmentation problem is to determine the sets $Seg_k \subseteq I$ whose union is the entire

image I . Thus, the sets that make up the segmentation must satisfy:

$$I = \bigcup_k Seg_k \quad (1)$$

Where

$$\forall k \neq l \quad Seg_k \cap Seg_l = \emptyset$$

Ideally, a segmentation method finds those sets that correspond to distinct anatomical structures or regions of interest in the image. In typical two dimensional images, the segments are usually visible for the human eye and the purpose of the segmentation is to automate the segmentation procedure. In hyper/multi-spectral images it is not the case. Hyper-spectral images are three dimensional and therefore, contain many layers. Segmentation of this kind of images identifies objects based on the spectral profile and spatial information. By using the Hyper-spectral segmentation, one can identify objects which don't appear in any single layer. Therefore, the segmentation can reveal objects which are not visible to the human eye when viewing one slice at a time.

B. Multi-Scale geometric analyzes

Multi-scale geometric analysis (MGA) is a very effective approach for the analysis and segmentation of digital images. MGA methods are best suited for data containing geometric features such as line segments and filaments buried in high level of noise. The Beamlet algorithm, evaluates a collection of line integrals along a strategic multi scale set of line segments running through the image at different orientations, positions, and lengths. It has been shown [12] that Beamlets are highly relevant to various image processing problems ranging from curve detection to image segmentation.

C. Multi-Scale Geometric Decomposition – Beamlet analysis

In the past fifteen years, multi-scale thinking has become very popular. The main advantage of Multiscale analysis is the ability of detecting objects of different sizes in a single data set.

Beamlet analysis is a multiscale geometric decomposition based on dyadically-organized line segments [14]. The Beamlet framework is based on three main components:

1. *The Beamlet dictionary* – a dyadically-organized set of line segments in different locations, scales and orientations which give a multiscale approximation of all line segments.
2. *The Beamlet transform* – a collection of line integrals along the line segments set defined in the *Beamlet dictionary*.
3. *Tree structure Beamlet algorithms* – high level algorithms that rely on the beamlet graphs and on connectivity and good continuation properties in such graphs.

Beamlet analysis is an effective tool that can be used for a wide range of applications, such as approximating an image using line segments. Our discussion will focus on using Beamlet analysis for segmentation purposes.

1) Beamlet dictionary

Before defining the beamlet dictionary, we will start with some terminology and notations. We assume a square image that consists of n by n pixels. As in [10], we will represent

the image as a square $[0,1]^2$ and the pixels are defined by $1/n \times 1/n$ size cells arranged in the $[0,1]^2$ domain. A dyadic square S is a collection of points $\{(x_1, x_2) : [k_1/2^j, (k_1 + 1)/2^j] \times [k_2/2^j, (k_2 + 1)/2^j]\}$ where $0 \leq k_1, k_2 \leq 2^j$ for an integer $J \geq 0$ and $n = 2^J$.

For $v_1, v_2 \in [0,1]^2$. A *Beam* $b = \overline{v_1 v_2}$ is a line segment connecting these two vertices. If we only consider line segments that connect vertices at pixel corners, we get $O(N^2)$ beams from an image with $N=n^2$ pixels. Typical fast image processing algorithms complexity is $O(N)$ and at most $O(N \log N)$. Therefore, it is strongly desired to reduce the number of beams. Beamlets are a strategic efficient subset of beams that includes only $O(N \log N)$ elements. The beamlets are a multi-scale set which is defined using the hierarchical structure of dyadic squares within a digital image.

The dyadic square at scales $0 \leq j \leq J$ has a side length of $\delta = 2^{-j}$, for each dyadic square there are $M_j = 4 \cdot 2^j$ boundary vertices, and therefore if we will consider only beams that connects a pair of such boundary vertices we will get only $\binom{M_j}{2}$ for the given dyadic square, such beams are called beamlets. In Figure 1 we have an example of beamlets at different scales

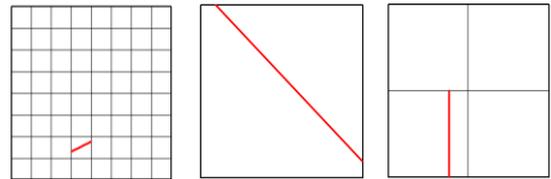


Figure 1. Beamlets at different scales

We note that beamlets only connect vertices on the boundary of each square, so the number of beamlets is much smaller than $O(N^2)$. In fact, the order of the total number of beamlets is $O(N \log N)$. This fact is essential in order to produce fast algorithms using Beamlet analysis.

Despite of the reduced cardinality of the beamlets set, the beamlet dictionary is still expressive, we can represent any line segment as well as any smooth curve using a chain of connected beamlets, as shown in Figure 2.

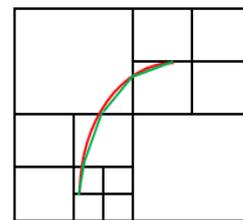


Figure 2. Smooth curve (red lines) approximate by a chain of beamlets (green lines).

2) The Beamlet Transform

Let $f(x_1, x_2)$ be a continuous function on $[0,1]^2$. The beamlet transform of f is a collection of all line integrals

$$T_f(b) = \int_b f(x(l)) dl, \quad b \in B_{n,\delta} \quad (2)$$

Where

$$B_{n,\delta} = \{b = \overline{v_i v_j}, 0 \leq i, j \leq M_j\}$$

Thus, the beamlet transform evaluates a set of line integrals over all $b \in B_{n,\delta}$, where $B_{n,\delta}$ is the group of beamlets defined by δ .

3) Tree structure Beamlet algorithm

There is a variety of different beamlet algorithms [8]. In this work we focus on a tree structure algorithm which relies on a recursive dyadic partitioning (RDP). A recursive dyadic partition is based on combining the following two rules:

- $P = \{[0,1]^2\}$ is an RDP
- If $P = \{S_1, \dots, S_{l-1}, S_l, S_{l+1}, \dots, S_1\}$ is an RDP and S_l can be decomposed into four dyadic squares $S_{l,00}, S_{l,10}, S_{l,01}, S_{l,11}$ then the new partition

$\dot{P} = \{S_1, \dots, S_{l-1}, S_{l,00}, S_{l,10}, S_{l,01}, S_{l,11}, S_{l+1}, \dots, S_1\}$ is an RDP.

In Figure 3 and Figure 4 we can see examples of a balanced RDP tree structure (meaning that all of the tree leaves are the smallest possible dyadic squares) and an imbalanced RDP.

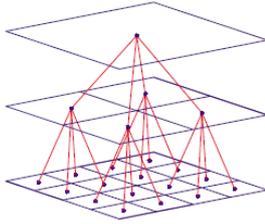


Figure 3. A balanced RDP tree structure

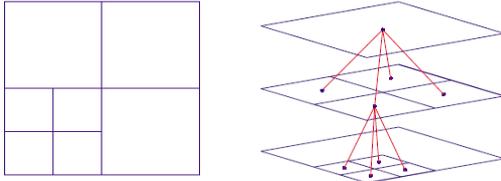


Figure 4. An Incomplete imbalanced RDP and the corresponding tree structure

Now, the stage is set for introducing the Beamlet decorated RDP (BD-RDP). This is an RDP which some of its squares (optionally) are decorated by a beamlet. In Figure 5 we can see an example of a BD-RDP. The motivation for adding a BD option is to make it possible to approximate a smooth curve boundaries in a much more efficient way compare with the ordinary RDP.

Our main goal is to represent an image that consists if a union of homogenous regions (segments). We can get an approximation by finding the optimal BD-RDP representation of the image. This is done by minimizing a complexity-penalized residual sum of squares and approximating the image using a piecewise constant representation. In order to find such an optimal BD-RDP, we use the data collected in the *Beamlet Dictionary*, which maps all the possible beamlet splits, and the *Beamlet Transform*, which helps us in choosing the best splits.

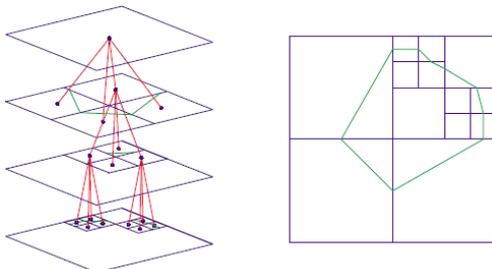


Figure 5. A BD-RDP (right) and its associated tree structure (left)

D. Lark's Algorithm

Lark's segmentation algorithm is based on two fundamental parts. The first one is the fuzzy c-means algorithm (FCM) [15], and the second one is spatial smoothing [6]. In this algorithm both spectral and spatial information is being used in a sequential manner. In the first step (FCM) segmentation is made according to the spectral information, where in the second step a smoothing is applied according to the spatial knowledge.

FCM algorithms are very common in hyperspectral analysis. This is the reason for choosing Lark's algorithm as a reference for this work.

Methodology

The main goal of this work is the development of a new segmentation tool that can efficiently and effectively analyze Multispectral images. The new proposed algorithm is based on the BD-RDP algorithm, which was developed by Donoho and Hue [8]. The innovation in our work is applying the BD-RDP based algorithm to Multi/Hyper-spectral data and adding a new Merge phase which significantly improves the segmentation results. Another significant contribution is the superfast implementation of the algorithm, the complexity of a straight forward naive implementation is $O(N^2)$. In order to decrease the complexity and make real time analysis possible, we have integrated fast methods such as Fast Slant Stack [14], the PPFST [13] and Radon Transforms [16] which reduced the algorithm overall complexity to $O(N \log N)$.

In order to validate and evaluate our segmentation results a reference method was needed. Since one of our major applications of interest is segmentation of agricultural remote sensing HS images, we looked for an algorithm, which is commonly used for these kinds of applications. Lark's algorithm, which is based on FCM, is a natural choice.

We will next define our improved BD-RDP method and its Hyperspectral implementation, afterwards, we will describing the nature of the agricultural application and data.

E. The proposed Beamlet algorithm

In this section, we describe the conflicts and solutions that arose during the implementation of the beamlet algorithm. Furthermore, we introduce the improvements made to the basic BP-RDP algorithm. Our discussion focuses on analysis of three dimensional data. Therefore, all the calculation ahead will be vectorial ones.

The implementation of the algorithm can be divided into three main stages:

1. Splitting phase – in this phase, a tree structure is built out of the multispectral image. The image is partitioned to its smallest parts according to a quad tree structure. While doing so, the best beamlet split of each part is saved.
2. Folding phase – in this phase, we fold up the tree that was built in the first phase. In each folding junction, we need to decide between three options: prune, not prune or prune with a beamlet.
3. Neighbors Merge –. In this phase, we look for segments pairs whose merge was never examined.

These three stages are described in details in the following subsections.

1) *Splitting phase*

In this phase, we split the image and produce a quad tree structure using an iterative step. In each iteration, we divide a dyadic square (node) into four smaller dyadic squares nodes (Figure 3). For each new node, we look for the best beamlet split.

The question ay hand is: what is the best beamlet split? Before answering, let us keep in mind that a beamlet split divides a square A into two regions A_{b1}, A_{b2} by a linear (beamlet) split. For each square A , there are many possible beamlet splits. These splits depend on the orientation and location of the separating line (as shown in Figure 6).

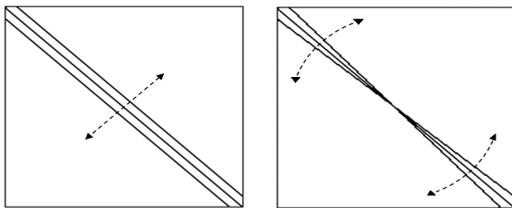


Figure 6. Beamlets at different locations and orientations

The optimal split is determined by minimizing of sum of squared error (SSE).

$$F = SSE(A) = \sum_{j \in A_{b1}} (x_j - \bar{x}_{A_{b1}})^2 + \sum_{j \in A_{b2}} (x_j - \bar{x}_{A_{b2}})^2$$

$$= \sum_{j \in A_{b1}A_{b2}} x_j^2 - \left[\frac{(\sum_{j \in A_{b1}} x_j)^2}{N_{A_{b1}}} + \frac{(\sum_{j \in A_{b2}} x_j)^2}{N_{A_{b2}}} \right] \quad (3)$$

Where $\bar{x}_{A_{b1}}$ is the vector of averages of the spectral intensities in the region A_{b1} and $\bar{x}_{A_{b2}}$ is defined similarly for A_{b2} .

At the end of this phase, we get a tree structure (RDP) of the multispectral image with the best beamlet split for each node. In order to reduce the calculation time of examining all of the beamlet splits, we use the Radon transform (implemented in $O(N \log(N))$ complexity using the PPFFT [10]). Additionally, in order to further reduce the computational complexity, we apply simple updates of the sums and sums of squares in the computation of SSE above by successively computing sums of parallel splits. An additional data, such as average value, sum of pixels squares, number of pixel and etc. is saved for each node of the tree. This data will serve us on the next phase, where parts of the three are folded.

2) *Folding phase*

In this phase, we act bottom-up; starting with the leaves (end-nodes) of the tree and pruning them, until we get an optimal tree structure. The folding phase is iterative. At each step, we have a three way decision to make (Figure 7):

- *No Pruning* – represent the given area as four squares $A_{00}, A_{01}, A_{11}, A_{10}$ resulted from the previous iteration. Note, that these squares can be *Beamlet Decorated* from previous iterations.
- *Pruning*– merge four squares: $A_{00}, A_{01}, A_{11}, A_{10}$ to one uniform square A . This merge is done, by setting all of the square’s pixels as the average value of the square.

- *Beamlet Decoration* – merge four squares: $A_{00}, A_{01}, A_{11}, A_{10}$ to one *Beamlet Decorated* square.

The decision between these three options is made according to the target function, which is defined in section II.F

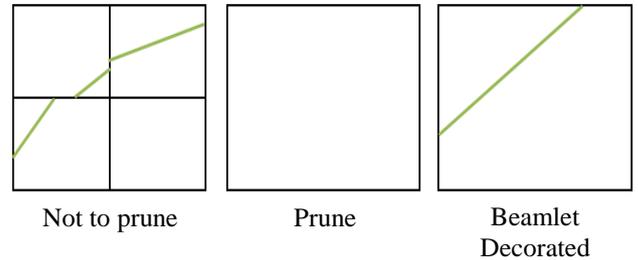


Figure 7. 3-way decision

At the end of this phase, we have a final BD-RDP, which is built according to the *tree structure based Beamlet algorithm* (defined in section II.C.3). This structure defines a segmentation of the multispectral image. In the next section we show how we can further improve the segmentation result by adding a *Neighbors Merge* phase.

3) *Neighbors Merge*

In this section we introduce an additional phase which significantly improves the segmentation results of the BD-RDP. The algorithm as described in the previous sections is based on building a complete quad tree at a first phase. At the second phase, the possibility of merging each four squares into one uniform (*prune*) or *Beamlet decorated* square is evaluated. One can note that in the second phase we only examine the merge of children of the same father in the tree structure. Therefore, there are many segment pairs that their merge is not feasible, even if it could improve significantly the segmentation representation (as shown in Figure 8).

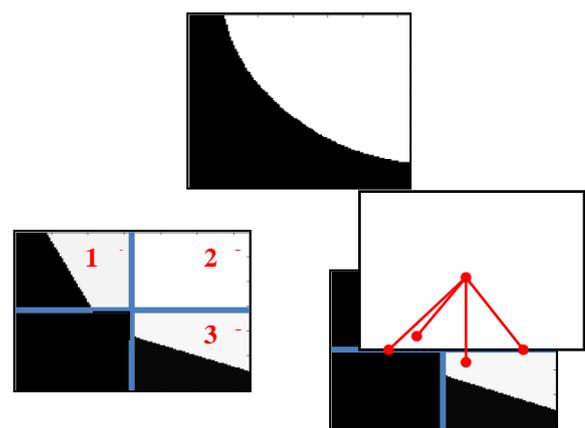


Figure 8. Top: Original 128x128x3 image; Down-Left: segmentation representation; Down-Right: quad tree beamlet decorated representation

Figure 8 illustrates a segmentation representation of an image using BD-RDP. In the left image, according to the algorithm, merge of squares 1 and 2 or 2 and 3 wasn’t considered even though it is clear that the right choice is to merge them.

The first option for neighbors merge that can come to mind is a serial check of all the possible mergers between a pair of

spatially neighbors segments. The complexity of such a procedure can be as high as $O(K^2)$, where K is the number of segments and in an extreme situation of a *complex* segmentation it can even get to the order of $O(N^2)$, where N is the total number of pixels in the image. Therefore, a different more economical approach was called for. Our main application of interest involves identifying homogenous areas in agricultural images, where our objects of interest are relatively large. This assumption led us to start the neighbors merge from the biggest identified segments. We look the largest segment and evaluate all the merge options with its neighbors. After dealing with the largest segment, we move to the second largest segment and so on. The complete algorithm is as follows:

1. Chose the largest segment which was not chosen yet
 - 1.1. Find all of its neighbors
 - 1.2. For each neighbor:
 - 1.2.1. Evaluate the target function after merging the pair of segments.
 - 1.2.2. If the new target function value is better than the best target function value so far – update the best target function value and keep the chosen neighbor index.
 - 1.3. Merge with the last best valued neighbor.
 - 1.4. The merged block is now considered to be a new block which was not chosen yet.
 2. If a merge was applied go to 1 otherwise stop
- Different stopping criteria can be defined such as the number of iterations or biggest block size.



Figure 9. Result of the above example after the Neighbors Merge phase

In Figure 9 we can see an example of executing the Neighbors Merge on the example above (Figure 8). The result is a merge of blocks 1,2 and 3.

F. Target Function

As mentioned above, the goal of our algorithm is to produce segmentation representation of an image. We are approximating the original image using segments. In order to produce an accurate segmentation, we need to look at the *error* resulting by this approximation and try to minimize it. If our target function was based only on minimization of the error, the best resulted segmentation representation was the image itself.

In order to prevent such representation, an addition of another ingredient is needed. As commonly done in data analysis procedures, we have added to our penalty function a *complexity* factor. In our terminology, complexity is measured by the number of blocks in our BD-RDP representation, where complex image is one built from many blocks.

There are many different target functions that could be defined, according to these two contradicting principles (SSE and complexity). One of them is Akaike Information Criterion (AIC) [17].

$$AIC = 2 \cdot K + N \ln \left(\frac{RSS}{N} \right) \quad (4)$$

Where K is the number of blocks (segments), N is the number of pixels and RSS is defined as:

$$RSS = \sum_{k=1}^K \sum_{j=1}^N (x_{k,j} - \bar{x}_k)^2 \quad (5)$$

K represent the *complexity* penalty and RSS represent the *error* factor.

Another example for a target function that can be used is the Bayesian Information Criterion (BIC) [18].

$$BIC = \ln(N) \cdot K + N \ln \left(\frac{RSS}{N} \right) \quad (6)$$

One can note that those target functions differ only on by the weight given to the *complexity* penalty, where in the BIC criterion the penalty weight, $\ln(N)$, is larger (for image with more than 7 pixels).

In our algorithm, we generalize these target function and set the *Penalty weight* (Pw) as a smoothing parameter.

$$TF = Pw \cdot K + N \ln \left(\frac{RSS}{N} \right) \quad (7)$$

The bigger Pw is, the smoother the image is.

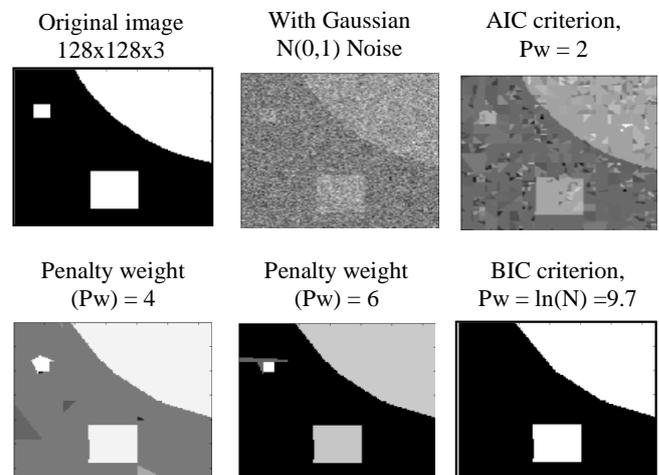


Figure 10. Example of segmentations using different Penalty Weight values.

As we can see in Figure 10, the smoothness of the image is increasing respectively with the increase of the Penalty Weight (PW). We can note that with the BIC criterion the Pw is the largest and therefore it is the smoothest one. On the other hand, as shown on the right bottom image, identification of relatively small objects, such as the small square, is problematic. When using the target function and setting the PW value, we need to take into account the noise level and the objects size

Results

In this section we introduce various segmentation results, which will show the advantages of the new BD-RDP over Lark’s algorithm and the improvement of the BD-RDP. The analysis will start with synthetic data, examples that we produced, in order to examine the efficiency of the algorithm. Later on we continue to real agricultural images.

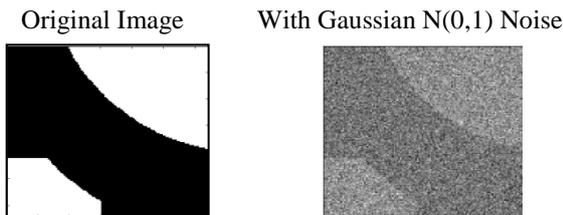
G. Synthetic data

In this section, synthetic images were produced in order to examine the efficiency of the new BD-RDP algorithm. As a first step, its performance was examined on different noise levels and different object size. As a second step, it was compared to Lark’s algorithm.

1) Different noise level

Images of 128x128x3 pixels with round and square objects were produced. In order to examine the effect of a different noise level, Gaussian noise with zero mean value and different variance was added.

For each image the results are gathered in two phases. First, the results of the BD-RDP – marked as *Beamlet*. Second, the results of the new Neighbors Merge step – marked as *Merge*.



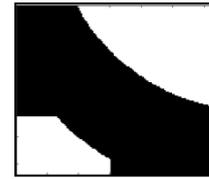
	Penalty Weight = 2	Penalty Weight = 6
<u>Phase 1</u> <i>Beamlet</i>	 930 Segments	 17 Segments
<u>Phase 2</u> <i>Merge</i>	 368 Segments	 3 Segments

Table 1 - Segmentation results of normally noised image, when using different penalty weights.

In Table 1 we can see the effect of the *Merge* phase. The *Merge* phase has significantly reduced the number of segments in each image. At the first example (Penalty weight=2) the number of segments is reduced from 930 to 368. At the second, it reduced from 17 to 3 – which is exactly the number of segments in the original image. One can note that when using a small penalty weight (AIC), our algorithm doesn’t function well.

In the next step we examine the algorithm with different noisy images, when using penalty weight of four.

Original Image



Noisy Image	Penalty Weight = 4	
	<i>Beamlet</i>	<i>Merge</i>
 <i>N(0,9)</i>		
 <i>N(0,25)</i>		
 <i>N(0,100)</i>		

Table 2. Segmentation results of images with different Gaussian noise level.

In Table 2 we can see segmentation results, while using penalty weight of four. First we can notice how the merge phase significantly improves the resulted images. Second, even under extremely noise conditions such as $\sigma^2 = 25$ (SNR=0.2) we get very good results. In our agricultural application of interest, which we’ll show in section II.H, the noise level is quite moderate.

2) Different object size

In the previous section we tried to segment objects in the same size in all of the examples. The relative proportion of the objects in the previous examples was 40% of the image size.

In this section we try to analyze the sensitivity of the algorithm to different object sizes. In each image, the object size was set differently and was marked as a proportion of the image (Table 3). The noise level was set to $N(0,0.5)$ and the Penalty weight to four. We can notice the different representation accuracy of each different object size.

	Object size 1%	Object size 9%
Original Image		

Noised Image $N(0,0.5)$		
Phase 1 Beamlet		
Phase 2 Merge		

Table 3. Segmentation results of noisy images with different object size

In Table 3 we can see segmentation results of noisy images with different object size. It is easy, and quite trivial, to see that as the object's size is increases the segmentation results are better. As mentioned in previous sections, in our application of interest the objects that needed to be segmented are relatively large and their proportion is around 3% - 6% of the image.

3) *Different Object Locations*

The beamlet algorithm is based on a quad-tree structure, as stated before. Therefore, the same object in different locations of the image is represented by a different quad-tree; meaning that the analysis is not completely translation invariant. The difference can be reflected in a different order of the tree branches with the same tree levels and the same amount of leaves as in Figure 11. In such a case, the accuracy of the representation is kept.

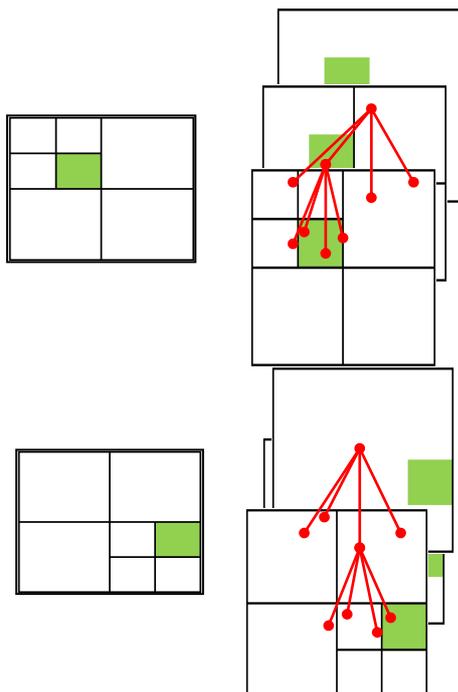


Figure 11. The same object in different location of the image (Up and Down) cause different tree representation

The tree representation is determined by the target function values, which is based on a *complexity* penalty. In Figure 12, the representation accuracy is kept because we have the same number of segments. In other cases, such as shown in Table 4, in order to get to the same accuracy representation, a tree structure with larger number of leaves is needed.

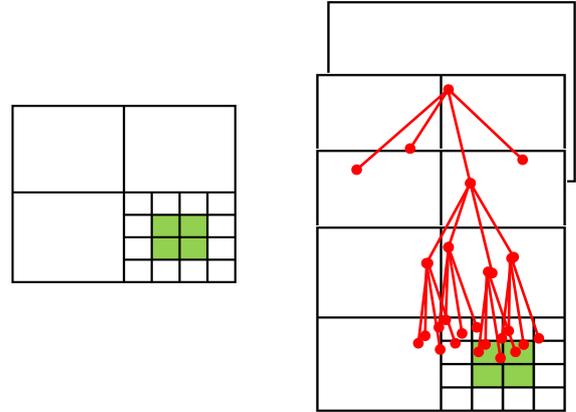


Figure 12. The same object as in the previous figure, with different tree representation

This increased number of leaves result in a larger *complexity*.

Let's look at this observation from another point of view, from the beamlet algorithm point of view. When running the algorithm, a preliminary step is setting the penalty weight and this definition of the target function dictates the resulted tree structure. Therefore, for a given target function, when analyzing images with the same objects but in different locations different results can be obtained.

Original Image	Noised image	Segmentation results
 7 Seg. is needed		 7 Seg. representation
 28 Seg. is needed		 8 Seg. representation

Table 4. Segmentation of images with the same object in different locations, when using $PW=6$

In Table 4, we can see segmentation results of noisy images with the same object in different locations. The top image is represented by 7 blocks. In order to get to the same representation accuracy in the second example, a 28 blocks representation is needed. Using the same penalty weight as in the top image forces an 8 blocks representation, which decreases the accuracy.

There are two possible solutions. One is to tolerate this lack of accuracy, which will cause, in the worst case, approximation using beamlet decorated cells instead of another dyadic split (as in Table 4). Another is to use a different penalty value for the beamlet phase and a different one for the merge phase. The idea is to set lower penalty for the *Beamlet* phase and higher for the *Merge* phase. In the *Beamlet* phase we let the algorithm produce a relatively *complex* image, which doesn't corrupt the object shape, but produce many objects. In the *Merge* phase, we merge the different objects and get larger blocks with the right shape.

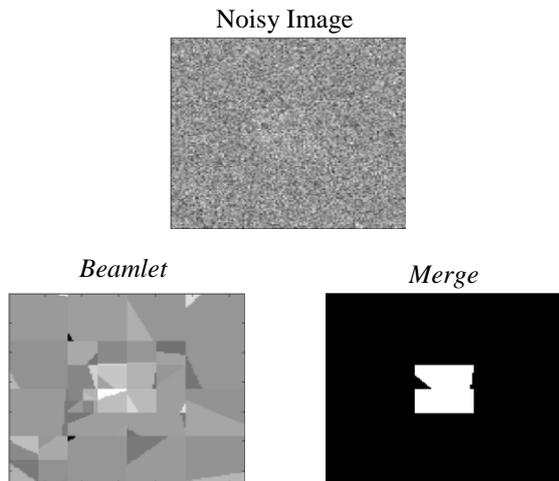


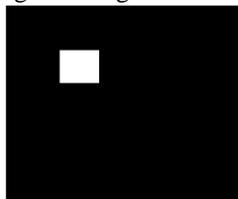
Figure 13. Example of different penalty weight for each phase. Beamlet phase – Pw=2; Merge phase – Pw=8

Figure 13 shows an example of segmentation (the example of Table 4) using different penalty weigh for each phase. The penalty weight of the *Beamlet* phase is two and of the *Merge* phase is eight. With this solution, we improved the segmentation result and got closer to the original image.

4) *Comparison with Lark's algorithm*

After examining the performance and properties of our algorithm, we can compare it with Lark's algorithm. The comparison between these two algorithms is done in two steps. In each step the examination is done under different noise levels. The first comparison step is analysis of performance, when the proportion of object size is 2.5%.

Original Image 128x128x3



Noised Image	Lark	Beamlet + Merge
 N(0,0.25)		

 N(0,1)		
 N(0,4)		

Table 5. Segmentation results of images with different noise level

In Table 5 there are examples of segmentation results of the two algorithms, when analyzing images with different noise level. We can see that Lark's algorithm has difficulties in recognizing a small object even under the presence of low level of noise. On the other hand, our algorithm presents relatively good results even under noise with $\sigma^2 = 4$ (SNR=0.5)

From the examples shown above we can note that the segmentation accuracy is determined by a combination of two properties: object size and noise level. From the segmentation results we can notice of the tradeoff between these two properties. We get the same segmentation accuracy for a large object with high noise level and a small object with low noise level. In statistical terminology, the meaning of small object is small sample size of its class and the meaning of high noise level is that there is a large overlapping between the classes. In these conditions Lark's algorithm has difficulties in producing an accurate segmentation. On the other hand, the new BD-RDP algorithm is less affected by these conditions. The reason lays on its multiscale property. The tree produced by our algorithm is built in a bottom-up approach, therefore the identification of the objects are done in a relatively small spatial sample. In other words, during the recursive operation, we can find the right sample size (square size) in which the object will be relatively big enough (the object's sample size will be big enough) in order for it to be identified. That is one of the reasons for the outstanding results of our method and its superiority over Lark's or similar methods in performing well and in a robust fashion even under extreme conditions.

The beamlet algorithm is not perfect and has its drawbacks compared to Lark. We can notice that because of the quadratic and geometric structure of our segmentation algorithm, in some cases it is difficult to recognize and represent object depending on their location (respectively to the quad tree squares) and shape (such as circular lines). This problem does not occur in Lark's algorithm because there is no attempt to fit the data into geometric shapes.

H. *Real Agricultural Data Analysis*

As a part of a research conducted in collaboration with the Volcani agricultural institute and sponsored by the Israeli space agency, our new segmentation algorithm is being used on aerial and satellite hyperspectral imagery. The goal of the joint project is to develop a precise agricultural method to control the fertilization process. The agricultural motivation

is firm. Adequate assessment of nitrogen level (using fertilization), leads to higher chlorophyll level in the plant, which results in a better photosynthetic process and increase yield [19] [20].

In this research, different areas of a potato field were fertilized with different amount of nitrogen (Figure 14). The multispectral data are images of a potato field, taken by an air plane and contain 11 spectrum channels from 400 nm up to 1000 nm (Figure 15). The goal of the segmentation is to identify each treatment, where high reflectance value corresponds to low nitrogen content.

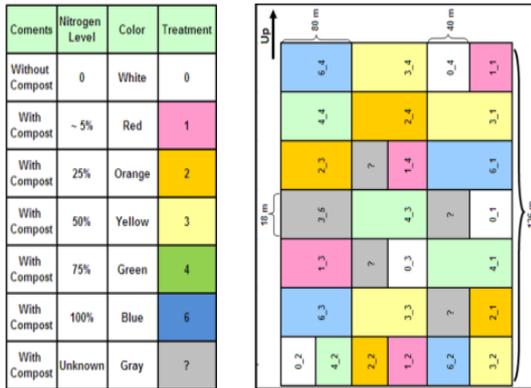


Figure 14. Right: The experiment field scheme with different treatments; Left: Index of the fertilization portions.

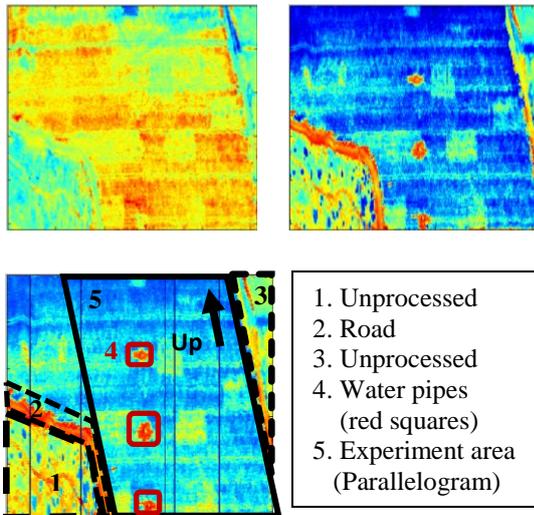


Figure 15. Up: two layers out of the hyperspectral image - 910nm (left) and 420nm (right); Down: mapping of the image areas.

We run our improved Beamlet algorithm using the real agricultural images. In Figure 16, we can see the segmentation results in two steps: before and after the merge stage.

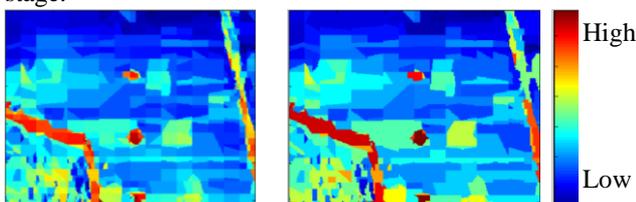


Figure 16. Segmentation results using Pw=30. Left: before Merge phase; Right: after the Merge phase

One can notice that there is a big improvement in the segmentation results. The numbers of segments were reduced significantly from 839 segments to 201.

We can notice that areas: 0_2,1_2,0_3,0_1,1_3 and 1_4 (Figure 14) which has the lowest nitrogen level, differ from the surrounding. Furthermore, these areas got the highest values, which is corresponding the theory described in above. Choosing the “right” penalty weight stays an open question and depends on the segmentation sensitivity which the analysts look for. In our application it seems that (according to our dialogue interactions with the researchers from the Volcani institute) the penalty weight should be between 10 to 20.

Discussion and Conclusion

This research was done in two related planes: algorithmic and practical. The first plane is the theoretical algorithmic one, which its goal was to develop and implement a new improved multiscale segmentation algorithm for Multi/Hyper-spectral images. The second plane is the practical one, which is expressed in using the proposed algorithm for agricultural purposes.

The proposed algorithm is based on the BD-RDP algorithm, which functions as the first step of the new proposed method after being adjusted to multi-channel images. Our improvement is reflected in two main ways. The first one is adding a new merge procedure, which functions as the algorithm’s second step. The second one is generalizing and adapting the algorithm to hyper/multi-spectral data. The implementation of the algorithm is based on the Pseudo Polar Fourier Transform and the projection slice theorem combined with smart updates of sums on the image using successive evaluations of parallel linear splits, which results in reduction of the complexity of the algorithm from N^2 to $N \log N$. In the proposed implementation we used a flexible target function which is based on generalization of the well-known AIC/BIC functions. Due to the target function’s flexibility, which is expressed in the penalty weight, the algorithm performs very well even under extremely noise conditions. This flexibility combined with our two step method, which enables dynamic penalty weight have significantly improved the identification of objects in the image over the original BD-RDP algorithm as well as compare to non multi-scale segmentation methods commonly used for HS images analysis such as the Lark algorithm.

On the practical point of view, our algorithm was used as a part of a research conducted with Volcani’s institute for agricultural engineering. The research’s goal was to make fertilization decisions based on analysis of multispectral images of a potato field. Our segmentation method was able to identify the low nitrogen level treatments and distinguish them from the others. As a part of the multi-spectral data analysis we have identified the most influential wavelength bands and used PCA in order to reduce dimensionality and decrease the running time of the algorithm. The collaboration with the Volcani institute continues where the goal is that the new segmentation method will serve as a pre-stage for classification to different nitrogen levels.

Being general and flexible, the proposed methods can be adjusted and fitted to many other image processing applications in various fields of applications, especially in multi-channel and other types of high dimensional images.

References

- [1] G. Vane and a Goetz, "Terrestrial imaging spectroscopy," *Remote Sensing of Environment*, vol. 24, Feb. 1988, pp. 1-29.
- [2] A. Ifarraguerri, "Hyperspectral Image Segmentation with Convex Cones."
- [3] J.C. Tilton, W.T. Lawrence, and A.J. Plaza, "Utilizing Hierarchical Segmentation to Generate Water and Snow Masks to Facilitate Monitoring Change with Remotely Sensed Image Data," *GIScience & Remote Sensing*, vol. 43, Mar. 2006, pp. 39-66.
- [4] M. Azarbad, A. Ebrahimzade, and V. Izadian, "Segmentation of Infrared Images and Objectives Detection Using Maximum," *International Journal of Computer Information Systems and Industrial Management Applications (IJCSIM)*, vol. 3, 2011, pp. 026-033.
- [5] a Plaza and C.-I. Chang, "Impact of Initialization on Design of Endmember Extraction Algorithms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, Nov. 2006, pp. 3397-3407.
- [6] R.M. Lark, "Forming spatially coherent regions by classification of multi-variate data: an example from the analysis of maps of crop yield," *International Journal of Geographical Information Science*, vol. 12, Jan. 1998, pp. 83-98.
- [7] R. Lark and J. Stafford, "Classification as a first step in the interpretation of temporal and spatial variation of crop yield," *Annals of Applied Biology*, 1997.
- [8] J.S. Borges and A.R.S. Marçal, "Hyperspectral image segmentation using FSMLR with Jeffreys prior," *Europe*, 2009, pp. 378-385.
- [9] D. Landgrebe and W. Lafayette, "CLASSIFICATION OF HIGH DIMENSIONAL DATA WITH LIMITED TRAINING Saldju Tadjudin," *Computer Engineering*, 1998.
- [10] A. Plaza, J.A. Benediktsson, J.W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, and A. Gualtieri, "Recent advances in techniques for hyperspectral image processing," *Remote Sensing of Environment*, vol. 113, Sep. 2009, p. S110-S122.
- [11] D. Landgrebe, *Signal theory methods in multispectral remote sensing*, Hoboken: John Wiley and Sons, 2003.
- [12] D. Donoho and X. Huo, "Beamlets and multiscale image analysis," *Multiscale and Multiresolution Methods*, vol. 72890, 2002.
- [13] a Averbuch, R. Coifman, D. Donoho, M. Elad, and M. Israeli, "Fast and accurate Polar Fourier transform," *Applied and Computational Harmonic Analysis*, vol. 21, Sep. 2006, pp. 145-167.
- [14] A. Averbuch, R. Coifman, D. Donoho, and M. Israeli, "Fast Slant Stack: A notion of Radon transform for data in a Cartesian grid which is rapidly computable, algebraically exact, geometrically faithful and invertible," *SIAM Scientific*, vol. 72890, 2003.
- [15] J. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, 1984, pp. 191-203.
- [16] S.R. Deans, *The Radon transform and some of its applications*, 1983.
- [17] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, Dec. 1974, pp. 716-723.
- [18] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, 1978, pp. 461-464.
- [19] D. Haboudane, "Integrated narrow-band vegetation indices for prediction of crop chlorophyll content for application to precision agriculture," *Remote Sensing of Environment*, vol. 81, Aug. 2002, pp. 416-426.
- [20] Y. Cohen, V. Alchanatis, Y. Zusman, Z. Dar, D.J. Bonfil, a Karnieli, a Zilberman, a Moulin, V. Ostrovsky, a Levi, R. Brikman, and M. Shenker, "Leaf nitrogen estimation in potato based on spectral data and on simulated bands of the VEN μ S satellite," *Precision Agriculture*, Nov. 2009.

Acknowledgments

This research was funded by the Israeli space agency. The fertilization study stated in section H was done in collaboration with Agricultural Research Organization – Volcani Center researches: Dr. Victor Alchanatis and Dr. Yafit Cohen.

Authors Biographies

Ofer Levi. Born in Israel. Received his Ph.D in Scientific Computing and Computational Mathematics from Stanford University in 2004. . Has been a lecturer in the department of Industrial Engineering Management of Ben-Gourion University Since 2004. Main interests - mathematical and statistical modeling of physical systems, inverse problems, compressed sensing, signal and image processing, parallel and distributed computing, biologically inspired analysis methods, physiological signal processing.



Shaul Cohen. Born in Israel. Ph.D student at Ben-Gurion university of the Negev in the field of Image Processing. Received his B.Sc in math in 2007. Has been a lecturer in the Industrial Engineering Management of Ben-Gourion University Since 2009. His main scientific interests include: Algorithms design for image processing, Mathematical modeling of complex systems, Hyper-spectral image analysis.



Ziv Mhabary. Born in Israel. Ph.D student at Ben-Gurion university of the Negev in the field of Image Processing. Received his Master degree in 2009. Has been a lecturer in the department of Industrial Engineering Management of Ben-Gourion University Since 2009. His main scientific interests include: Algorithms design for image processing and computer vision. Hyper-spectral images, multi-scale analyzes.

