Pyramidal segmentation using higher-order local auto-correlations and its applications to Landsat forestry data

Milos Stojmenovic[1], Takumi Kobayashi[2], Nobuyuki Otsu[2]

[1]Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia 21000 milos.stojmenovic@agrosense.org

[2]National Institute of Advanced Industrial Science and Technology (AIST) 1-1-1, Umezono, Tsukuba, Ibaraki, Japan, takumi.kobayashi@aist.go.jp, otsu.n@aist.go.jp

Abstract. The goal of image segmentation is to partition an image into regions that are internally homogeneous and heterogeneous with respect to neighbouring regions. Recently, a link shifting based pyramidal segmentation method was proposed to resolve existing problems with elongated regions. In this paper, we propose further improvements by replacing pixel intensities at the base level with pixel level higher order local auto-correlation (HLAC) feature vectors over greyscale, RGB, and CIV channels. Thereby, rich texture-like information is incorporated into segmentation. We propose a normalized distance formula between HLAC vectors, where each component contributes with physically same unit. The new algorithms were tested on a set of Landsat images over forested areas, and compared with a non-HLAC variant and several other existing segmentation algorithms. A significant improvement in segmentation quality was achieved compared to non-HLAC variants, and it also gave better results than other existing algorithms on most examples.

Keywords: pyramid segmentation, auto-correlation.

1. Introduction

The motivation for our work here is in the segmentation of satellite imagery to separate relatively homogenous terrain regions such as forest, road and water. Segmentation of this type in an unsupervised fashion is an important, yet difficult first step in categorizing various forest types, and building an accurate database for forest inventory.

Pyramid segmentation was proposed in [9, 3]. 'Unforced linking' [1] allows the exclusion of some vertices from the linking scheme, and the presence of small components. Pyramids are hierarchical structures, which offer segmentation algorithms with multiple representations with decreasing resolution. Each representation or level is built by computing a set of local operations over the level below (with the original image being at base level 0). The value of the node is derived from the values of all its children nodes at the level below. When this is applied recursively, the value of each node at a given level is decided by the set of its descendent pixels at the base level (its receptive field). Each node at level L also represents an image component of level L segmentation, consisting of pixels belonging to its receptive field.

In [7], authors first observe that the connectivity of the receptive field (region in segmentation) is not always a desirable characteristic. For instance, in forestry applications, the same type of vegetation could be present in several parts of the image, and treating them as single segment may in fact be preferred for further processing. In other applications an object that is partially obscured and 'divided' by other objects may also be more naturally identified as a single object instead of processing it in multiple pieces. If, however, region connectivity is essential, it can be achieved by posterior processing, where connected components can be extracted on all derived segments to yield only the connected regions as final product. The segmentation algorithm [7] allows for non-connectivity of receptive fields.

We enhance the segmentation quality of [7, 8] by replacing simple pixel intensity values from the image at the base level with pixel level higher-order local auto-correlation (HLAC) feature vectors [6] extracted from greyscale, RGB, and CIV channels. HLAC extracts local textural features, and is shift invariant. Thereby, the proposed method incorporates texture-like information into segmentation. Furthermore, we employ a normalized distance formula between HLAC vectors, where each component contributes with the same physical unit.

Experiments were conducted to validate the proposed methods on complex forest data. "The ill-defined nature of the segmentation problem" [5] makes subjective judgment the generally adopted method (existing quantitative measurements are based on subjective formulas). In general, it is not clear what a 'good' segmentation is [5]. For this reason, no quantitative evaluation measure is applied to verify the results. Our new algorithms are slower than the basic version from [7]. However, the segmentation quality was significantly improved, as shown on selected images.

2. Pyramidal segmentation

Algorithm LS (Link Shifting) [7] is inspired by the regular pyramid linked approach (PLA) [3]. The algorithm [7] properly handles elongated objects, while preserving shift invariance (the stability of a segmentation algorithm when minor shifts occur), having favourable time complexity (receiving an answer within seconds, depending on the size of the image), and overall simplicity, so that the algorithm can be easily understood, implemented, and used in practice. The authors [7] made some simple changes in the way parent nodes are selected in the regular pyramid framework. Instead of always comparing and selecting among the same four candidate parent nodes as in [3], each vertex at the current level will compare and select the best

among the current parent, its neighbours, and current parents of its neighbouring vertices at the same level. Either of 4- or 8-connectivity neighbours can be used. In case of a tie, parent with the higher number of children is chosen. In [8] the same authors also introduced the concept of a co-parent node for possible region merging at the end of each iteration. The new parents (from 13 or 29 candidates) of the former children are co-parent candidates if they are similar. The co-parent is the one with the largest receptive field among candidate co-parents. Each child then considers the co-parent of the previous parent as additional candidate.

3. Higher-order local auto-correlation (HLAC)

Higher-order Local Auto-Correlation (HLAC) was proposed for extracting "spatial correlation of pixel values", and has been shown to work efficiently in image recognition [6]. HLAC is computed over a local rectangular region, $(2p+1)$ x $(2p+1)$ centered at a reference point, and will be denoted here by HLAC(p). We first explain pixel level HLAC(1). The N-th order HLAC(1) R_k is decided by the corresponding vectors $(a_1, \cdots, a_N)$ and is calculated, for a given pixel r, by the following formula: $R_k(r)= I(r)I(r + a_1) \cdots I(r + a_N)$, where I is the gray-scale image, $r = (x,y)$ is a position vector of pixel coordinates, and $a_i=(\Delta x_i, \Delta y_i)$ are the displacement vectors. These parameter values are restricted to vary as follows: $\Delta x_i, \Delta y_i \in \{\pm 1, 0\}$ and $N \in \{0, 1, 2\}$. The configuration $(r, r+a_1, \cdots, r+a_N)$ is reduced to 35 patterns for greyscale images, as shown in Figure 1, by eliminating duplicates that arise at neighbouring pixels from shifts. Thus, the HLAC(1) feature for a pixel in the image is a 35-dimensional vector. $R_1=I(r)$ is the only component with $N=0$, and it corresponds to the image pixel intensity. In our application, this means that pixel intensity at the base level was replaced by a vector where the first coordinate of the vector is the pixel intensity. R_2-R_5 and R_{26} have N=1 (e.g. $N_2=N_3=N_4=N_5=N_{26}$=1), and the remaining ones have an order of N=2. For instance, $R_2(r)=I(r)I(r+a_1)$, for a_1=(1,0), $R_{10}(r)= I(r)I(r+a_1)I(r+a_2)$, for a_1=(-1,0), a_2=(1,1), $R_{31}(r)= I(r)I(r)I(r+a)$, for a=(-1,1).

HLAC was defined in [6] as the summation over HLAC(1) vectors for all pixels, resulting in a single 35-dimensional vector for the image. Here, however, we consider HLAC features as 35-dimensional vectors over every pixel. Area based HLAC features (HLAC(p)) were proposed in [4]. Elementary pixels are combined into regions by averaging over the pixel values inside each region. The HLAC values are then calculated over such regions. Here we applied only the version with pixels as regions. HLAC(p) features correspond to the summation of HLAC(1) values over a rectangular moving window. We associate each HLAC(p) value with the pixel that is in the center of the corresponding $(2p+1)$x$(2p+1)$ rectangle. The rectangle sizes used in our algorithm were 5x5 and 11x11 pixels (p=2 and p=5). In HLAC(p), the feature value associated with pixel $r=(x,y)$ is, for $g=p$-1:

$$S_k(r)= \sum_{i=-g}^{g} \sum_{j=-g}^{g} R_k(x+i, y+j).$$

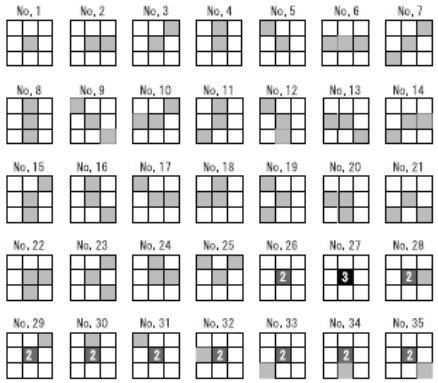

Figure 1. Mask patterns (the numbers in cells indicate duplicated positions for auto-correlations).

Thus the center of the moving window is inside the $(2p+1)$ x $(2p+1)$ rectangle. Boundary pixels (whose associated local region is not entirely inside the image) have slightly adjusted calculations with respect to their local regions. Pixels (x',y') outside the image which should contribute to the calculation of the region are replaced with their mirroring pixels within the image (that exist). If $x' >$ image width or $y'>image\_height$, $x'=image\_width-(x'-image\_width)-1$, $y'=image\_height - (y'-image\_height)-1$. If $x'<0$ or $y'<0$, $x'=-x' -1$, $y'=-y'-1$.

4. HLAC Based Pyramid Segmentation

In this section we describe algorithmic extensions to the LS algorithm [7]. We first replace the pixel intensity at the base level of pyramid with a HLAC(1) vector. It has a merit of extracting richer feature information from the local color textural pattern structure. It is a primitive and the true merit of HLAC is cumulative (HLAC(p)). Shift invariance is the key property for the success of the segmentation. The vector forms were employed to represent feature values, corresponding to a single channel, up to 3 channels of image data. In the single channel version, the pixel intensity $I(r)$ is derived from RGB data using one of options. The RGB-HP1 (HLAC(1) Pyramid segmentation) algorithm uses the Red, Green and Blue (RGB) channels. The CIV-HP1 algorithm uses the Infrared, Red, and Green (IRG) channels. It is used in literature [10] to detect vegetation in the images. Grayscale pixel values were computed by $G = 0.299*R + 0.587*G + 0.114*B$, from the RGB image, in the corresponding G-HP1 algorithm. Algorithm G-HP1 uses a 35-dimensional vector while algorithms RGB-HP1 and CIV-HP1 use 105-dimensional vectors at each pixel.

Instead of simple differences in pixel intensity values, our new algorithm must use vector differences. Existing distance formulas for measuring similarities and dissimilarities of feature vectors over 35- or 105-dimensional vectors did not give encouraging data, because the components of HLAC(1) have a scaling bias according to their order (power), with important information such as

the simple difference in intensity values being 'undervalued'. We proposed the following 'fair' distance formula. It is a normalized distance, where each component has a proper physical unit which depends on the corresponding value of N. The summation of over the (N_i+1)-th root of the absolute value of difference of the i^{th} components of the vectors A and B:

$$D(A,B) = \sum_{i=1}^{35} |A_i - B_i|^{1/(N_i+1)}.$$

35 is replaced by 105 for algorithms based on three channels. Further adjustments are made in the thresholds for deciding similarity among components (S) and the application of unforced linking (D). The threshold value S = 15 was replaced by 15 x 35 = 525 for 35-dimensional vectors, and this number was multiplied by the number of channels being considered. The threshold D = 100 was similarly replaced by 100 x 35 x (no. of channels). We found that a threshold of S = 1575 worked best for the RBG-HP1 and CIV-HP1 algorithms.

We have also considered the HLAC(p) feature values associated with pixels at the base level of the pyramid. They resulted in more notable improvements in the segmentation quality than previous enhancements. The corresponding algorithms are labelled RGB-HPp and CIV-HPp where $(2p+1)$x$(2p+1)$ is the size of the HLAC local region. The corresponding thresholds for the new algorithms should also adjusted, by multiplying the above S and D values with approximately $(2p+1)$. For the HLAC test cases, S = 15 x 35 x (no. of channels) x $(2p+1)$, and D = 100 x 35 x (no. of channels) x $2(2p+1)$.

5. Experimental Results

The algorithm presented here was designed to improve upon the quality of segmentation of satellite forest data via an unsupervised approach that considers all possible information given. Our algorithm was applied to Landsat imagery from the Ottawa, Canada region, provided by the Ontario Ministry of Natural Resources. These images were of size 512x512 pixels and came with the near Infra red, Red, Green, and Blue bands. Each pixel represents an area of 40cmx40cm of the earth. Combinations of these bands were used in our experiments. The processing time for such images was roughly 3 minutes per image when considering a single channel, and 9 minutes when considering 3 channels while it was about 1 minute for algorithm [7]. The experiments were run on a single core of the Intel I7 chip at a speed of 2.66 GHz running on a 64 bit windows Vista platform. The implementation was done in C# and used the EMGU OpenCV, C# wrapper classes for handling the image processing functions. Since all of the images are 512x512 pixels in size, their segmentation pyramids contain 8 levels. We will show only the final level of segmentation of the pyramid since it best reflects the desired segmentation results for these images. Although it is relatively easy for us to judge whether or not a simple shape laid against a high contrast background is segmented properly, it is far more difficult to evaluate the precision of a segmentation of forest imagery. In our experiments, the judgement of the quality of the segmentation is subjective since no ground truth data was available. Our goal was to separate as clearly as possible general classes of terrain via unsupervised learning, such as roads, water and forest covered areas. Note that the images were introduced to the algorithms in their original form, which means no pre-processing of any kind was performed (ex: normalization of channels, enhancement…).

Figure 2 below shows the test set and segmentation results of all of the considered algorithms. The first column shows the original image, followed by the segmentation results of each algorithm respectively. The ERDAS (www.erdas.com), LS [7, 8], and AGBB [2] algorithms are competitors and the rest of the columns are methods we propose here.

The Erdas Imagine 9.1 unsupervised classification (ERDAS) is required to have a manually set input of the number of anticipated segments per image (in our examples, each image can be divided into 2 classes of terrain). The rest of the algorithms have preset parameters which remain constant throughout testing. The subsequent algorithms are listed as follows LS [7, 8], G-HP1, RGB-HP1, RGB-HP2, CIV-HP2, RGB-HP5, CIV-HP5 and AGBB. RGB-HP1, RGB-HP2 and RGB-HP5 refer to HLAC based methods with local regions 3x3, 5x5 and 11x11, respectively. CIV-HP2 and CIV-HP5 refer to area based HLAC with p=2 and p=5 respectively applied to the CIV representation.

We see in Figure 2 that the non-area based algorithms over segment the image, whereas the area based methods generally well isolate the water from the land and the road. The ERDAS, G-HP1 and RGB-HP1 are the worst offenders of over segmentation, yet the remaining algorithms mainly also include small inclusions (segments) in areas that should be generally homogenous. To the best of our judgement in segmentation quality, the first image is best segmented by the AGBB algorithm [2], the second by CIV-HP5, the third by RGB-HP5, the fourth is segmented equally well by almost all of the algorithms, and the fifth by RGB-HP5 again.

5. Discussion

In our examples, the most homogeneous regions (water) were all segmented out very well, because the water regions were relatively homogenous in colour. Textured areas were more or less segmented depending on the algorithm used. A disadvantage to the algorithms proposed here are the long execution times associated with even relatively small images. To speed up the method, some parallelism is considered. The software may need to be redesigned to take advantage of multi CPU machines, and the algorithms need to be adjusted to execute more rapidly. The proposed method has an advantage for such a parallelism because HLAC values and pyramid linking can be executed in parallel. The language of implementation should also be taken into consideration.

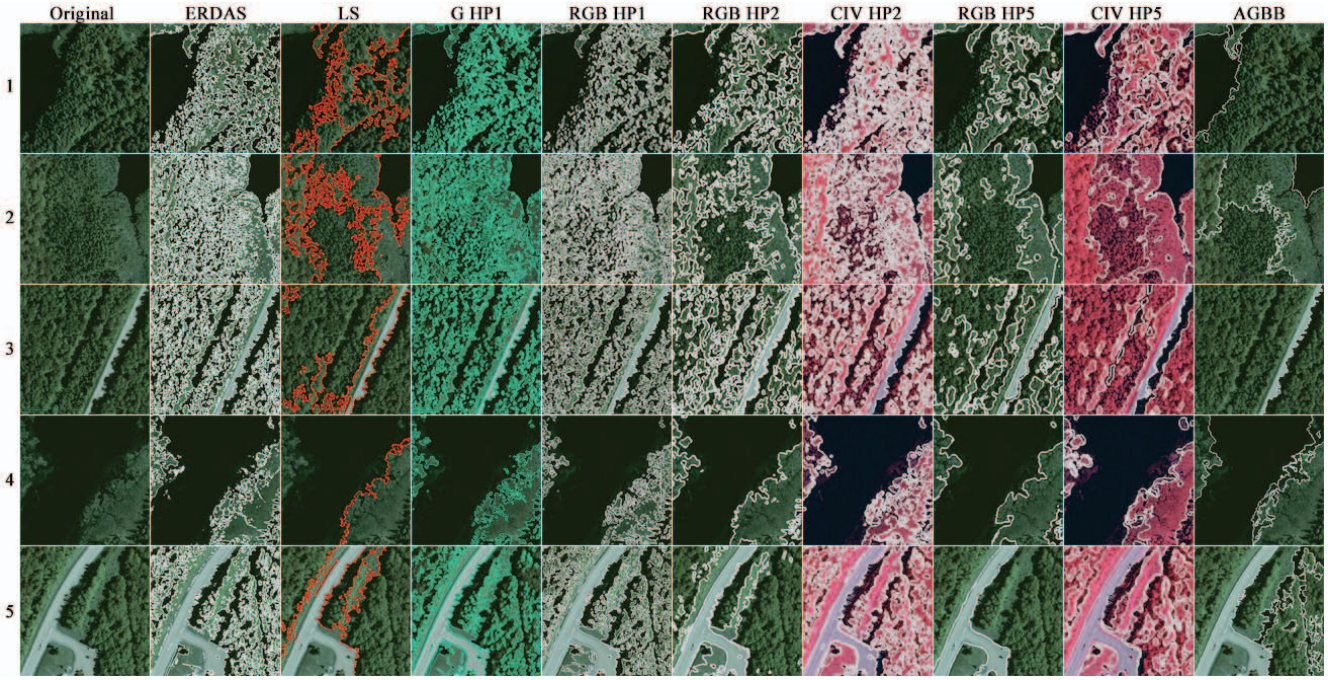

Figure 2 – Test set

The role of HLAC features was to correctly incorporate similarly textured areas into the same segment. Pixel-level information tends to form local clusters and to result in over-segmentation. An example in forest imagery is any area of trees where the tops of trees have a lighter colour than their bodies. Such an area would group the lighter colours of the tops together and the darker colours of the bodies would form at least another segment. This results in too many segments which both HLAC(1) and HLAC(*p*) avoid. HLAC features include color information in corresponding components of HLAC vectors. They therefore offer richer information and are more discriminative than simple colour features and as such are more useful in the segmentation task.

HLAC(*p*) features introduce a smoothing property to the overall segmentation. This characteristic can both be an advantage or a disadvantage. In the segmentation of satellite vegetation data, it can be considered an advantage since the resulting segmentation results look more like they have been done by a human. As the size of the mask in the HLAC(*p*) algorithms grows, these smooth boundaries may start to join too many segments and produce unrealistic results. On the other hand, images that contain many exact line segments with very sharp contours may suffer from the application of this technique for the opposite reasons.

5. References

[1] H. Antonisse, Image Segmentation in Pyramids, *Computer Graphics and Image Processing*, Vol. 19, 367-383, 1982.

[2] S. Alpert, M. Galun, R. Basri, A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-07), 2007.

[3] P. Burt, T. Hong, A. Rosenfeld, Segmentation and estimation of image region properties through cooperative hierarchical computation, *IEEE Trans.Systems, Man, & Cybernetics*, 11, 12, 1981.

[4] A. Hidaka, T. Kurita, N. Otsu, Object detection by selective integration of HLAC mask features, *ECSIC Symposium on Bio-inspired, Learning, and Intelligent Systems for Security (BLISS2008),* Aug. 2008.

[5] R. Marfil, L. Molina-Tanco, A. Bandera, J. Rodriguez, F. Sandoval, Pyramid segmentation algorithms revisited, *Pattern Recognition,* 39, 1430-1451, 2006.

[6] N. Otsu and T. Kurita. A new scheme for practical flexible and intelligent vision systems. *IAPR Workshop on Computer Vision*, 431–435, 1988.

[7] M. Stojmenovic, A. Solis-Montero, A. Nayak, Link shifting based pyramid segmentation for elongated regions, Int. Symp. Signal Processing, Image Processing and Pattern Recognition (SIP 2009), Jeju Island, Korea, Dec. 2009, *LNCS,* to appear.

[8] M. Stojmenovic, A. Solis-Montero, A. Nayak, Co-parent selection for fast region merging in pyramidal image segmentation, submitted for publication.

[9] S. Tanimoto, T. Pavlidis, A hierarchical data structure for picture processing, *Computer Graphics and Image Processing*, Vol. 4, pp. 104-119, 1975.

[10] http://edc2.usgs.gov/pubslists/factsheets/fs12901.pdf