PATTERN-BASED ROUGH SEGMENTATION OF SATELLITE IMAGES USING EVOLUTIONARY ALGORITHMS

Radu Andreiu, Radu-Mihai Coliban, Angel Caţaron, Mihai Ivanovici

Machine Intelligence and Vision Laboratory
Department of Electronics and Computers
Transilvania University
Braşov, România

ABSTRACT

Earth Observation applications are becoming more and more spread, thus the automatic interpretation of satellite image content is a key issue. Determining the characteristics of a certain region allows for image segmentation, classification and analysis or further interpretation of satellite image content. Given the very-high resolution of current satellite images, we propose a rapid and rough segmentation approach based on object detection using extended Haar-like features and evolutionary algorithms. The segmentation map is given by the probability of finding a specified pattern and the distance to discovered solutions. We show the results we obtain for satellite image segmentation of built-up areas based on rectangle-shaped house roof detection.

***Index Terms*—** segmentation, extended Haar-like features, evolutionary algorithms, satellite images

1. INTRODUCTION

There exists a plethora of image segmentation approaches [1], but only several frameworks are unanimously accepted by the scientific community as generating good and predictable results. Watershed is the morphological approach, where the image is viewed as a relief or landscape [2]. Pyramidal or multi-scale approaches use a hierarchal structure for representing the image at different resolution levels [3]. Active contours are based on variational calculus and are commonly used to segment one region or a salient object with a priori known characteristics [4]. Active contour implementations are usually very time consuming. Graph-based approaches use graph representations of images, which unfortunately may explode in size with the increase of image spatial resolution [5]. Recently, there is a trend in over-segmenting the image on purpose, then to proceed with fusion of similar regions [6]. Another set of approaches are based on the local description of texture, e.g. Haralick texture features [7]. However, all the aforementioned approaches aim at determining regions with similar properties.

With the increase of image spatial and spectral resolution, all the approaches may become very time consuming. In addition, some regions may exhibit a large variability of the composing elements, making them difficult to segment. In this article we embrace an image information mining approach which combines low level feature extraction with evolutionary algorithms. We aim at determining the regions that contain a certain pattern of interest. The hypothesis that we make is the following: in very high resolution images, such as satellite images, it may take a lot of time to explore the entire image using a deterministic and exhaustive segmentation approach, therefore a probabilistic method that does not explore the entire image and feature space will offer a rapid and rough estimate of the regions of interest. In addition, purely deterministic approaches may fail when applied to objects with a high variability in shape, color or texture.

Genetic algorithms use the principles of evolution and selection to search through a very large set of candidate solutions. This may be an appropriate approach when a set of criteria for the characterization of a convenient solution can be aggregated in a *fitness function*, providing a numerical hierarchy with the most valuable elements on top. When it comes to searching in an image a set of similar objects but with a large variability, the genetic algorithms can be used in a segmentation step. As a result, the areas with a high probability of containing the objects will be highlighted.

Several approaches have been proposed for object detection using genetic algorithms: in [8] distributed genetic algorithms are used for the segmentation of video frames, in order to extract moving objects in MPEG-4 sequences. Genetic algorithms are used in [9] to create cascades of classifiers for object detection in noisy ultrasound images. The authors of [10] integrate basic image processing algorithms in a fuzzy-genetic framework in order to detect buildings in high-resolution satellite images. For the same purpose, there exist alternative approaches like the ones based on SIFT (Scale-Invariant Feature Transform) keypoints and graph theory [11] or texture-derived built-up presence index which is an anisotropic textural co-occurrence measure [12]. In this

doi: 10.2788/25852

paper we show results of rough segmentation of satellite images using an evolutionary method. By searching for simple shapes like rectangles, we aim at quickly identifying image regions with high probability of representing built-up areas.

2. OUR APPROACH

One approach from computational intelligence for object detection in images is emitting hypotheses about the existence and appearance of desired objects inside the analyzed images, and then verifying the validity of those hypotheses. Further on, the hypotheses are refined, thus leading to solutions that best suit the image. One way of improving the discovered solutions is through evolutionary processes, such as genetic algorithms. Using such a method requires a way to encode hypotheses, quantify their fitness and select them accordingly, then reproduce the fittest ones. This last step requires that mutations of the encoding string will occur. Searching objects inside images through an evolutionary process is done by randomly generating a number of hypothesized objects, computing a measure of their fitness and differentially reproducing them according to their fitness, repeating these steps for an empirically-chosen number of generations.

2.1. Hypothesis encoding

The first step of the process requires a way of encoding a hypothesis into strings or numbers. The encoded string of an object is named genotype, while the resulting form is called phenotype. The fields of the hypothesis are hereby named chromosomes, which are further divided into smaller fields of information, called genes. Each gene encodes for a trait, or process of generating traits, while one of all possible forms of a gene is called an allele.

Each hypothesis $\mathbf{h}_i$ is of the following format: at position (x, y) in the image there exists a rectangle of width w, height h, orientation α and color c. Colors are represented as 24-bit RGB. Therefore the encoding represents the concatenation of all these fields into a single string $\mathbf{h}_i = [x, y, w, h, \alpha, c]$. For the encoding of the hypothesis we use Haar-like features defined by [13], presented in Fig. 1 where b is the size of the border. Given the large variability of the searched pattern, one can extend these features in order to match various application purposes [14].

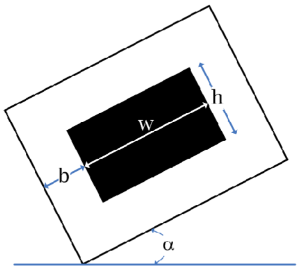


Fig. 1. Illustration of a hypothesis (extended Haar feature).

We do not specifically encode the color of the background (the border surrounding the object), but in our implementation we define an acceptance level for the contrast between the color of the object and the color of the background.

In Figure 2 we show an ideal synthetic image $\mathbf{I}$ with white background and four buildings represented by grey rectangles and four possible hypotheses ($\mathbf{h}_1, ..., \mathbf{h}_4$), generated at a certain step of our approach, overlaid on the original image.

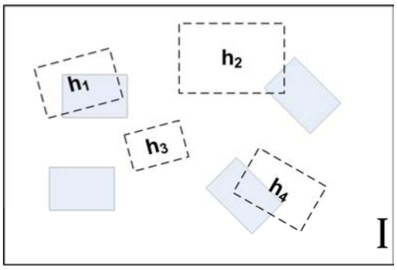


Fig. 2. Illustration of a phenotype overlaid on the image $\mathbf{I}$.

2.2. Selection based on fitness

The second phase of the evolutionary process is analyzing the phenotype and quantifying the correctness of the solution, i.e. assessing the match between the hypothesis $\mathbf{h}_i$ (searched pattern) and the original image $\mathbf{I}$, through the computation of the fitness function $f(\mathbf{h}_i, \mathbf{I})$:

$$f(\mathbf{h}_i, \mathbf{I}) = \begin{cases} \frac{1}{d(\mathbf{h}_i, \mathbf{I})} & , d(\mathbf{h}_i, \mathbf{I}) \leq T \\ 0 & , \text{otherwise} \end{cases} \quad (1)$$

where d is the distance between the hypothesis $\mathbf{h}_i$ and the corresponding (underlying) region of image $\mathbf{I}$ and T is a configurable threshold for the acceptance level of matching. Computing the distance d can be achieved through various approaches: computing the mean squared error, projecting a region of the analyzed image on a basis of functions representing the searched pattern, applying a hit-or-miss transform (e.g. the color hit-or-miss transform for simple patterns [15]), etc. The computation of the mean squared error has to take into account the local contrast between the color of the object and the color of the background. In addition, the information about the spatial arrangement of pixels is lost, but fortunately the information is intrinsic to the definition of the pattern. For our proof-of-concept approach we define $d(\mathbf{h}_i, \mathbf{I}) = \sum_{(u,v) \in Supp(\mathbf{h}_i)} |color(\mathbf{h}_i(u, v)) - I(x + u, y + v)|$.

After measuring the fitness of all the existing hypotheses, the selection phase takes place. The goal is the improvement of the existing solutions, meaning that the fittest ones must reproduce more often than the less fit solutions. This can be done either through a deterministic process, or a stochastic one. The former is as simple as greedily selecting the best N solutions, while the latter requires randomly selecting solutions based on their relative fitness, such that better hypotheses have greater chances of reproducing, offering no guarantee of actually achieving the goal. This can even lead to good hypotheses becoming extinct, but can also expand the pool of solutions compared to the former method by allowing non-greedy solutions to be selected. Statistically, better solutions would reproduce more often for both methods. The following overall function has to be maximized: $\arg\max\limits_{i, f(\mathbf{h}_i, \mathbf{I}) \neq 0} \sum\limits_{1}^{N} f(\mathbf{h}_i, \mathbf{I})$.

Selection can be tailored to specific application needs: for instance, if we are looking for roofs of a certain color, a best fit solution is marked as final solution and may not be further used for reproduction, since in the near vicinity (a few pixels) of the solution there cannot exist another best solution - however, further away, the probability of finding another roof is high in a built-up area.

2.3. Evolution

Finally, *reproducing* solutions is only a matter of copying the encoding string and modifying or mutating it, which means altering the values of various fields of the hypothesis encoding. The amount of mutation is an important parameter of the approach, as few mutations per genotype can lead to a longer, yet more stable, progression, while higher rates of mutation preserve too little of the information for selection to achieve its goal of generating a better batch of hypotheses. One way of increasing the amount of variation, while still preserving vital information is through the *combination* of similar solutions, which is called sexual reproduction. This serves to combine features of different hypotheses, yielding solutions that would be unlikely to occur through simple mutation of either of the combined hypotheses. Such combinations of encoding strings are facilitated by the existence of well-separated fields inside the strings, each one encoding specific sets of traits of the phenotype, or processes of the phenotype generator.

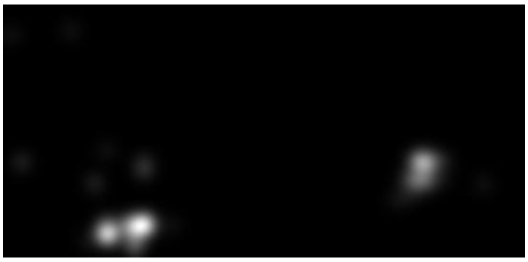
At the moment of reproduction, each parent solution donates its chromosomes, resulting in an offspring which has pairs of homologous chromosomes. Two problems arise: (i) how to generate a single phenotype from two sets of strings and (ii) how to select which chromosome from a pair to donate to the offspring in the reproduction phase. The first problem is addressed by the existence of allele dominance: some alleles are dominant (will get expressed in the phenotype), while others are recessive. If two different homologous alleles have the same dominance, their resulting expression is a mixture of the two. For the second problem, at the moment of reproduction, one of the homologous chromosomes is selected randomly, without bias. Furthermore, each pair of resulting homologous chromosomes can undergo a process of recombination by splitting both chromosomes at one or more locations and recombining the pieces to form new chromosomes that are different from any of the parent chromosomes.

At a first glance, the complexity of such an algorithm might seem unnecessarily high, but these processes can prove important when creating a suitable environment for the development of effective solutions for such general problems as object detection. Precisely what evolutionary processes can achieve in the context of object detection is still an open question and the recipe for an optimal implementation is currently unknown.

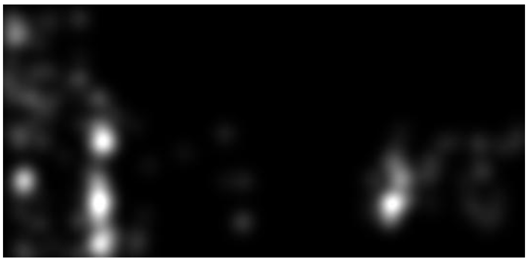
3. EXPERIMENTAL RESULTS

We demonstrate the usefulness of our approach in the context of an Earth Observation application: by searching rectangular patterns, representing house roofs, we aim at identifying and roughly segmenting the built-up areas within satellite images. In Figures 3 and 4 we show the original images, the output of the genetic algorithm (2D Gaussians in the positions of the discovered solutions) and the final segmentation maps.

(a) original image



(b) run #1



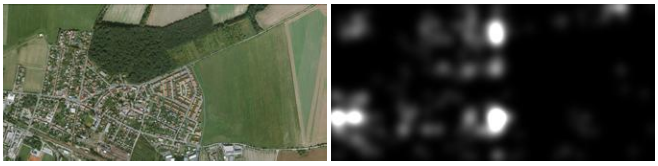
(c) run #2

(d) built-up area rough segmentation

Fig. 3. Experimental results.

For producing the final segmentation maps we compute at each image location the distance to the closest solutions and if the distance is smaller than a specified radius, then we include the pixel in the region of interest. For the results presented in Figs. 3 and 4 the radius was 150 and 100 pixels,, respectively.

Given the randomness intrinsic to our approach, some runs will lead to better results. However, this method can benefit from parallel runs and then determine statistically which region is commonly present in all results. The average time of execution for the C++ implementation of our approach on a color image of approximately 2 Mpixels, using 100 hypotheses for 100 generations is approximately 8 seconds on a single core CPU running at 2.2 GHz.



(a) original image (b) run #1

(c) run #2 (d) built-up area rough segmentation

Fig. 4. Experimental results (2).

4. CONCLUSIONS

In this article we present a proof-of-concept framework for rough image segmentation based on Haar-like extended features and evolutionary processes. We show how this approach can be used for a preliminary analysis of built-up areas in satellite images, by searching simple patterns, i.e. rectangles of specified color range, based on the hypothesis that they represent roofs of buildings. The results of this image information mining approach are rough segmentation maps.

Given the probabilistic nature of our approach and the way the evolutionary process works, there are two uncertainties related to the solutions: (i) the absolute fitness between the image and the searched pattern is not known and the relatively best solutions are chosen at a certain moment; (ii) the exact position of the best solution is not known, thus we are looking at the closest solutions within a specified range. Once the rapid and rough segmentation is performed by using our approach, one may use dedicated tools for a better discovery of the pattern of interest.

5. REFERENCES

[1] M. Ivanovici, N. Richard, and D. Paulus, "Color image segmentation," in *Advanced Color Image Processing and Analysis*, C. Fernandez-Maloigne, Ed., pp. 219–277. Springer New York, 2013.

[2] S. Beucher, "Watersheds of functions and picture segmentation," *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '82.*, vol. 7, pp. 1928 – 1931, may 1982.

[3] A. Rosenfeld, *Some pyramid techniques for image segmentation*, pp. 261–271, Springer-Verlag, London, UK, 1986.

[4] M. Kass, Andrew W., and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[5] A. K. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," in *Computer Vision, IEEE International Conference on*, Los Alamitos, CA, USA, 2007, pp. 1–8, IEEE Computer Society.

[6] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows.," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 2290–2297, 2009.

[7] R.M. Haralick and L.G. Shapiro, "Image segmentation techniques.," *Computer Vision, Graphics, & Image Processing*, vol. 29, no. 1, pp. 100–132, 1985.

[8] S. W. Hwang, E. Y. Kim, S. H. Park, and H. J. Kim, "Object extraction and tracking using genetic algorithms," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, 2001, vol. 2, pp. 383–386 vol.2.

[9] J. Masek, R. Burget, J. Karasek, V. Uher, and S. Guney, "Evolutionary improved object detector for ultrasound images," in *Telecommunications and Signal Processing (TSP), 2013 36th International Conference on*, 2013, pp. 586–590.

[10] E. Sumer and M. Turker, "An adaptive fuzzy-genetic algorithm approach for building detection using high-resolution satellite images," *Computers, Environment and Urban Systems*, vol. 39, pp. 48 – 62, 2013.

[11] B. Sirmacek and C. Unsalan, "Urban-area and building detection using sift keypoints and graph theory," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 47, no. 4, pp. 1156–1167, April 2009.

[12] M. Pesaresi, A. Gerhardinger, and F. Kayitakire, "A robust built-up area presence index by anisotropic rotation-invariant textural measure," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 1, no. 3, pp. 180–192, Sept 2008.

[13] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2002, vol. 1, pp. I–900–I–903 vol.1.

[14] S. Vural, Y. Mae, H. Uvet, and T. Arai, "Multi-view fast object detection by using extended haar filters in uncontrolled environments," *Pattern Recognition Letters*, vol. 33, no. 2, pp. 126 – 133, 2012.

[15] J. Weber and S. Lefévre, "Spatial and spectral morphological template matching," *Image and Vision Computing*, vol. 30, no. 12, pp. 934 – 945, 2012.