Pattern Recognition Letters 53 (2015) 24–30

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A new algorithm for number of holes attribute filtering of grey-level images ☆

Juan Climent [a,*], Luiz S. Oliveira [b]

[a] Barcelona Tech (UPC), Jordi Girona, 1, 08034 Barcelona, Spain
[b] Federal University of Parana (UFPR), Rua Cel. Francisco Heráclito dos Santos, 100, Curitiba, Brazil

ARTICLE INFO

Article history:
Received 20 February 2014
Available online 30 October 2014

Keywords:
Attribute filtering
Connected operators
Euler number

ABSTRACT

In this paper we present a new algorithm for filtering a grey-level image using as attribute the number of holes of its connected components. Our approach is based on the max-tree data structure, that makes it possible to implement an attribute filtering of the image with linear computational cost.

To determine the number of holes, we present a set of diverse pixel patterns. These patterns are designed in a way that the number of holes can be computed recursively, this means that the calculations done for the components of the image can be inherited by their parent nodes of the max-tree. Since we do not need to re-calculate the attribute data for all connected components of the image, the computation time devoted to the attribute computation remains linear.

1. Introduction

In mathematical morphology, connected filters [1] are operators that modify an image by only suppressing connected components, and thus, preserving its original contours. Attribute filters [2] are connected operators that keep certain shapes in the image based on a wide range of criteria. Different attributes can be used, like elongation, moment of inertia [3], energy [4], noncompactness [5], circularity [6], etc. The idea behind this approach is to think of attribute filtering as selecting shapes in images based on prior knowledge of the objects of interest. A connected component is preserved if it meets the corresponding attribute criterion, or removed otherwise.

The common attributes used for filtering binary images (like area, perimeter, moments, or Euler number) can be extended to grey-level images by threshold decomposition [7]. A grey-scale image can be decomposed in many binary images (called cross sections or level sets) by thresholding it at each grey-scale level. A cross section at level h is given by the set of all pixels greater or equal than h. A straightforward implementation of this idea, filtering each cross section as an independent binary image, leads to very inefficient algorithms. Nevertheless, attribute filtering of grey-level images can be efficiently implemented using Max-Tree algorithms [8–10].

Here, we propose a new algorithm for filtering a grey-level image using as attribute the number of holes of the image connected components. The number of holes of a connected component is a topological property closely related to the Euler number.

The Euler number is defined as the difference between the number of connected components and the number of holes in a binary image (Equation (1)). It is an important attribute, invariant to several image transformations such as translations, rotations, scale, projections, and even some non-linear deformations. Traditionally, it has been used in a great amount of applications, like signature verification [11], detecting malaria parasites in blood images [12] , reflectance-based object recognition [13], etc.

$$\varepsilon = N - H \tag{1}$$

There exist some algorithms [14–17], and patents [18] for computing the Euler number of a binary image. The most popular algorithm [19] is based on counting certain 2×2 pixel patterns called bit-quads.

The Euler number and the number of holes are equivalent concepts for a single connected component. Since for a single connected component, N value is always 1 in Equation (1), we can conclude that ε and H are practically the same thing. The result of computing the value of the Euler number of a connected component is always 1 minus its number of holes. Therefore, the same algorithm used for computing the Euler number can be used for computing the number of holes of a connected component.

Although there exist several algorithms for computing the number of holes (or Euler number) of the blobs in a binary image, to the best of our knowledge, no algorithm has been proposed up until now for computing the number of holes of the connected components of a grey-level image. Classical features used for grey-level attribute filtering are based on shape or size properties (like moments, volume

☆ This paper has been recommended for acceptance by G. Borgefors.

* Corresponding author. Tel.: +34 934137958.
E-mail address: juan.climent@upc.edu (J. Climent).

http://dx.doi.org/10.1016/j.patrec.2014.10.012
0167-8655/

or perimeter), but in this paper, we describe an algorithm that uses a topological attribute (number of holes) for filtering the image.

The rest of this paper is organized as follows: In Section 2 we review connected filters and the attribute filtering algorithm. Section 3 introduces our algorithm for filtering an image using the number of holes as attribute. We provide some results of the algorithm in Section 4, and make some conclusions in Section 5.

2. Attribute filtering

Connected filters are those that preserve the contours of the original image. They cannot create new contours, or modify the position of the existing ones. Therefore, they have very good contour-preservation properties and are able to low-level filtering and higher-level object recognition. These filters modify the image removing only flat zones [20], which are connected components where the image grey-level is constant.

The hierarchical representation of the connected components by means of max and min-tree, made possible the implementation of efficient algorithms to compute connected filters [10]. The use of these trees allows more sophisticated forms of filtering, like the one based on attributes [8]. Component features, called attributes, are computed on the nodes of the max-tree. The max-tree should contain both the hierarchy of connected components in the image, and the attributes for each component to be used as a filter criterion.

Since max-trees are used in many different applications (motion extraction [21], MSER feature extraction [22], segmentation, 3D visualization [6]), several algorithms have been proposed till now to compute the max-tree. Some of them are dedicated to a particular task, and are not useful for other purposes. In our work, we use the algorithm proposed in [10], that uses the max-tree to achieve, specifically, an efficient sequential implementation of attribute filters. It has a worst-case computational cost $O(kN)$ (being k the number of grey-levels in the image), but which is effectively linear for most natural images. In [23] they provide a full and exhaustive comparison of the state-of-the art of algorithms for computing the max-tree.

The algorithm proposes a separation of the filtering process into two stages: a tree construction stage, and a filtering stage.

In the first stage, the tree is built sequentially, structuring the pixels in a way that leads to efficient implementations of the further filtering process. As seen in Fig. 1, each node of the max-tree points to its parent (which has a strictly lower grey-level). The nodes corresponding to the components with the highest intensity are the leaves. The root, having the lowest grey-level, points to itself. This way of linking nodes simplifies the computation of component attributes since every parent inherits the data of its descendants. When dealing with increasing attributes (like area or volume), inheritance is a simple accumulation. The attribute considered in this paper is non-increasing (case of shape attributes), then the inheritance needs a more complex handling, described in next section.

The tree construction uses a flood-filling approach based on hierarchical FIFO queues, where each queue corresponds to a particular grey-level value. These queues are used to define an appropriate order for scanning and processing the pixels. It performs a depth-first sweep of the tree, starting at the lowest grey level of the image (the root), and moving upward in grey scale. When a pixel is processed, the algorithm updates the data of the node currently being flooded. After that, it inspects its neighbours and places them in their respective queues. If a neighbouring pixel is at a higher level, flooding proceeds recursively at this new level. Once a node is completely flooded, the process is completed by determining its parent and attribute fields. The function returns the node attribute information to its parent and the flooding process goes on at the new level.

The max-tree creation relies on a recursive flooding procedure. The algorithm has basically two steps: the first one performs the propagation and the updating of the status of each pixel being processed, and computes the attribute data of each node, whereas the second step defines the parent/child relationships. The process finishes when all pixels have been processed. The details of this algorithm can be found in [10].

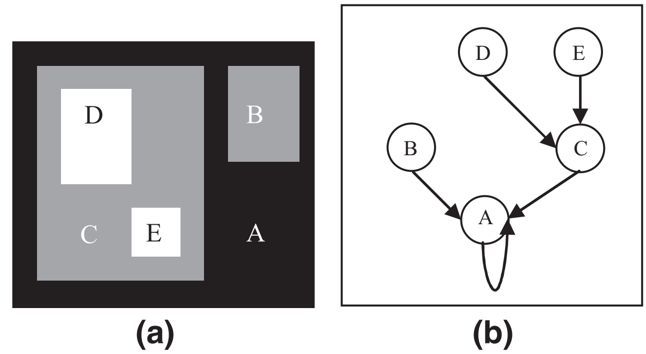

Fig. 1. A synthetic image and its corresponding max-tree. Region grey-level values: A = 0, B = 1, C = 1, D = 2, E = 2.

In the filtering stage, it is decided which nodes (connected components) will be suppressed from the tree. The pruning process is governed by a criterion that may involve simple features such as size, contrast, or more complex ones such as texture, motion, or even a criterion of similarity to predefined shapes. Finally the filtered image is obtained from the pruned tree.

Attribute filters use a criterion T to determine which connected components (CC) are preserved. If $T(CC)$ is true, then $Filter(CC) = CC$; otherwise, $Filter(CC) = NULL$. Usually, T is defined by using some attribute of the connected component, and comparing it to a threshold value. The criterion T can be that attribute value is either higher or lower than the threshold. Any node can be removed from the tree. There are four typical strategies for removing the filtered nodes:

- the max rule prunes the branches from the leaves up to the first node that has to be preserved.
- the min rule prunes the branches from the leaves up to the last node that has to be removed.
- the direct rule consists of removing the selected nodes even if this does not create a pruning. The pixels belonging to the connected components that have been removed are merged to the node of their first ancestor that has to be preserved.
- the subtractive rule [24] is the same as the direct rule except that the grey levels of surviving descendants of removed nodes are also lowered, so that the contrast with the local background remains the same.

3. Algorithm for attribute filtering by number of holes

3.1. Computing the number of holes for a binary component

Our algorithm for computing the number of holes of a connected component is originally based on the Euler number. The most popular algorithm for computing the Euler number of an image is the well known algorithm based on locally binary patterns [19], used in the MATLAB image processing toolbox. The local binary patterns consist in the following set of 2×2 binary pixel quads:

$$Q_1 = \left\{ \begin{matrix} 0\ 0 \\ 1\ 0 \end{matrix}, \begin{matrix} 0\ 0 \\ 0\ 1 \end{matrix}, \begin{matrix} 0\ 1 \\ 0\ 0 \end{matrix}, \begin{matrix} 1\ 0 \\ 0\ 0 \end{matrix} \right\}$$

$$Q_2 = \left\{ \begin{matrix} 1\ 1 \\ 0\ 1 \end{matrix}, \begin{matrix} 1\ 1 \\ 1\ 0 \end{matrix}, \begin{matrix} 1\ 0 \\ 1\ 1 \end{matrix}, \begin{matrix} 0\ 1 \\ 1\ 1 \end{matrix} \right\}$$

$$Q_3 = \left\{ \begin{matrix} 0\ 1 \\ 1\ 0 \end{matrix}, \begin{matrix} 1\ 0 \\ 0\ 1 \end{matrix} \right\}$$