Adaptive Pseudo Dilation for Gestalt Edge Grouping and Contour Detection

Giuseppe Papari and Nicolai Petkov

***Abstract*—We consider the problem of detecting object contours in natural images. In many cases, local luminance changes turn out to be stronger in textured areas than on object contours. Therefore, local edge features, which only look at a small neighborhood of each pixel, cannot be reliable indicators of the presence of a contour, and some global analysis is needed. We introduce a new morphological operator, called adaptive pseudo-dilation (APD), which uses context dependent structuring elements in order to identify long curvilinear structure in the edge map. We show that grouping edge pixels as the connected components of the output of APD results in a good agreement with the gestalt law of good continuation. The novelty of this operator is that dilation is limited to the Voronoi cell of each edge pixel. An efficient implementation of APD is presented. The grouping algorithm is then embedded in a multithreshold contour detector. At each threshold level, small groups of edges are removed, and contours are completed by means of a generalized reconstruction from markers. The use of different thresholds makes the algorithm much less sensitive to the values of the input parameters. Both qualitative and quantitative comparison with existing approaches prove the superiority of the proposed contour detector in terms of larger amount of suppressed texture and more effective detection of low-contrast contours.**

***Index Terms*—Edge and boundary detection, Gestalt grouping, morphological analysis methods.**

I. Introduction

CONTOUR detection is important for many computer vision applications, such as shape discrimination and object recognition. It is essential to distinguish the concepts of edge and contour. We use the term edge to refer to any non-negligible local luminance change. The term contour will be used to refer to the boundary line that delimits (a part of) an object from its background.

Different techniques have been proposed for edge detection: examples are linear filtering [1]–[5], local orientation analysis [6], [7], fitting of analytical models to the image data [8]–[11], local energy [12]–[15], statistical analysis of the gradient field [16]–[19], anisotropic diffusion [20]–[22], complementary analysis of boundaries and regions [23]–[25], and biologically motivated surround suppression [26]–[29]. The main limitation of these operators is that they detect contours by only analysing a small neighborhood around each pixel. Conversely, the human visual system performs a contour integration process, so that long chains of collinear edges are grouped together and recognized as object contours. A large amount of research has been carried out in order to develop contour detection algorithms which include computational models of contour integration. The approaches that have been proposed for this purpose can be classified in two categories: 1) computation of a global saliency map, and 2) grouping edges according to gestalt laws.

Saliency maps are obtained by iteratively increasing the strength of those edges which have collinear edges in their immediate surrounding, and by reducing the strength of edges that are surrounded by random patterns. The best known algorithms are based on relaxation labelling [30]–[32], which works in a graph based framework, and tensor voting schemes [33], [34], which associate a vector or tensor field with each edge pixel. Concerning the latter, recent work [35]–[37] shows that schemes similar to tensor voting can successfully model the facilitation and contour integration processes that are performed in the front-end part of the human visual system. Special extension fields, called *association fields* in the psychophysical literature [38], are defined in order to take into account inhibition and facilitation. For an overview of the related biological background, we refer to [39] and [40]. In [41], an iterative scheme based on uniform networks is developed in order to compute a saliency map from local information about curvature and curvature changes.

On the other hand, Gestalt theory concerns the ability of human observers to group visual stimuli which share certain common characteristics. The best known grouping criteria are *proximity* (stimuli that are close to each other are grouped together), *similarity* (stimuli that have the same shape, color or size are grouped together), and *common fate* (objects that move at the same speed in the same direction are grouped together). Whenever elementary stimuli or objects are grouped together, they form a new larger object, a gestalt. Such a larger object can be grouped together with other similar objects and the process can be repeated hierarchically until only a few large meaningful structures are identified. As a starting point of this contribution we use the ability of humans to group together local edge cues that are nearby and collinear, forming long and smooth lines (Fig. 1). In psychology, this phenomenon is generally referred to as *gestalt law of good continuation* (GLGC) [42]–[45].

Computational models of Gestalt laws can find applications in many computer vision tasks. For instance, Fig. 2 shows the result of a local edge detector [26] for a natural image. It effectively rejects texture while preserving most of the contour

Manuscript received July 12, 2007; revised June 20, 2008. Current version published September 10, 2008. G. Papari was supported by the NWO—Dutch Organisation for Scientific Research. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gaudenz Danuser.

The authors are with the Institute of Mathematics and Computing Science, University of Groningen, 9700 AK, Groningen, The Netherlands (e-mail: g.papari@rug.nl; petkov@rug.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2008.2002306

Fig. 1. Perceptual grouping of short line segments according to the GLGC: (left) a set of segments and (right) the most likely result of human grouping. The edges that are grouped together are shaded.

Fig. 2. (Left) Natural image and (right) the output of the local contextual operator proposed in [26]. The result is a binary map of unrelated edge pixels with many gaps in the contours.

edges. However, the detected contours are incomplete and present many interruptions. As we will show, including a computational model of edge grouping according to the GLGC in the process of contour detection contributes to filling many of the gaps and significantly improves the performance of contour detection. Moreover, for many applications, such as shape recognition, it is mandatory to have long contour segments as connected components instead of a binary map of unrelated edge pixels.

In an early work of Marr [46], a two-stages edge grouping process, called *curvilinear aggregation*, is proposed. First, edges that are close to each other and match in orientation and contrast are grouped into a contour segment. Then, analogous matching is evaluated for pairs of the resulting contour segments (nodes). Successive developments can be found, for instance, in [47]–[55]. However, The main limitations of most of the existing edge grouping algorithms is that typically they are computationally demanding.

In this contribution, we propose a contour detector which deploys a computationally efficient algorithm for edge grouping according to the GLGC. It is based on a new morphological operator, called *adaptive pseudo-dilation* (APD), which works with context-dependent structuring elements and limits dilation to the Voronoi cell of each edge pixel. The grouping algorithm is then embedded in a multithreshold scheme which, at each threshold level, removes small groups of edges and recovers missing contours by mens of a generalized reconstruction from markers. A wide range of experimental results, in comparison with other existing techniques, validates the proposed approach for contour detection. The exposition is organized as follows: In Section II, APD is defined and an efficient algorithm for its computation is presented. In Section III, the ability of APD to group edges according to the GLGC is shown. In Section IV, the grouping algorithm is embedded a multithreshold contour detector. Results are presented in Section V and discussed in Section VI, and conclusions are drawn in Section VII.

II. Adaptive Pseudo Dilation

In this section, we define APD and we provide a simple and fast algorithm for its computation. As well accepted in mathematical morphology, we refer to binary images as the set of their "1" pixels.

Let $b = \{\mathbf{r}_1, \ldots, \mathbf{r}_N\}$ be a point set in the plane and let $V(\mathbf{r}_k)$ be the Voronoi cell of $\mathbf{r}_k$. We also assume that a subset R_k of $\mathbb{R}^2$, which we call local structuring element (LSE), is associated with each point $\mathbf{r}_k$. With this notation, APD is defined as an operator that takes as input the point set b and the LSEs R_k, $k = 1 \ldots N$, and produces as output the union of all the intersections between each LSE and the Voronoi cell of the corresponding point

$$\operatorname{apd}(b) \triangleq \bigcup_k R_k \cap V(\mathbf{r}_k). \tag{1}$$

Let us assume that the Cartesian representation of the collection $\mathcal{R} = \{R_1, \ldots, R_N\}$ is available in the form of a positive *construction function* $\Delta_{\mathcal{R}}(\mathbf{r}, \mathbf{r}_i)$, defined such that $\Delta_{\mathcal{R}}(\mathbf{r}, \mathbf{r}_i)$ is less than one if and only if $\mathbf{r}$ belongs to R_k

$$\mathbf{r} \in R_k \Longleftrightarrow \Delta_{\mathcal{R}}(\mathbf{r}, \mathbf{r}_k) < 1. \tag{2}$$

Let us consider the Voronoi tessellation of the input point set b and let $\mathbf{q}(\mathbf{r}) \in b$ be the centroid of the Voronoi cell that contains point $\mathbf{r}$, where $\mathbf{r}$ is a generic point in the plane. In this notation, we can show that apd (b) is equal to the set of those points $\mathbf{r}$ such that $\Delta[\mathbf{r}, \mathbf{q}(\mathbf{r})] < 1$

$$\operatorname{apd}(b) = \{\, \mathbf{r} |\, \Delta_{\mathcal{R}}[\mathbf{r}, \mathbf{q}(\mathbf{r})] < 1 \}. \tag{3}$$

A formal proof of (3) is given in the Appendix A.

This result leads to a very efficient implementation of APD. Specifically, for each pixel $\mathbf{r}$, we compute its nearest neighbor $\mathbf{q}(\mathbf{r}) \in b$ and we add $\mathbf{r}$ to $\operatorname{apd}(b)$ if $\Delta_{\mathcal{R}}[\mathbf{r}, \mathbf{q}(\mathbf{r})] < 1$. In this process, the most demanding operation is the computation of the first nearest neighbor transform $\mathbf{q}(\mathbf{r})$ of the input image b, which can be computed very efficiently. We remark that while the computation time of the Voronoi tessellation of M points in a continuous domain is $O(M \log M)$, the computation of $\mathbf{q}(\mathbf{r})$ on a square lattice is linear with the size of the input image.

III. Edge Grouping

In this section, we use APD to group edges according to the GLGC. We first present the method with LSE of a fixed size (Section III-A) and then we describe a simple procedure to adaptively compute the size of the LSE for each different edge point (Section III-B).

A. Proposed Approach

Let $b = \{\mathbf{r}_1, \ldots, \mathbf{r}_N\}$ be a set of edge points with local edge orientations $\theta_1, \ldots, \theta_N$. The proposed approach to edge grouping is simple: first, we compute $\mathcal{D}_b = \operatorname{apd}(b)$, by using

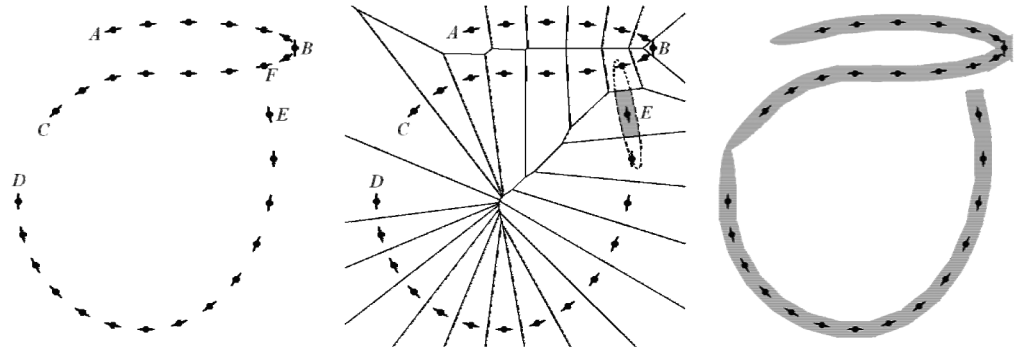

Fig. 3. Illustration of the proposed edge grouping algorithm for a simple edge set. (a) Input set of edge points with indication of the local edge orientation, (b) the corresponding Voronoi tessellation, and (c) the result $\mathcal{D}_b$ of APD, with ellipses as LSEs.

elongated LSEs which are oriented along the local edge direction θ_k. Second, we perform a connected components analysis of $\mathcal{D}_b$, and we group together points of b that belong to the same connected component of $\mathcal{D}_b$.

This process is illustrated by Fig. 3, where the LSEs are ellipses. Each ellipse, centered on point $\mathbf{r}_k$, has the major axis oriented along the local edge direction θ_k. The input point set [Fig. 3(a)] contains a long chain of collinear edges, with a high curvature point B and a large gap between points C and D. The Voronoi tessellation of the input point set is shown in Fig. 3(b) and the result $\mathcal{D}_b$ of APD is shown in Fig. 3(c).

Intersecting each ellipse with the corresponding Voronoi cell has several benefits. First, edges that are close to each other, but which are not collinear, are not grouped together [such edges E and F of Fig. 3(a)]. Classical dilation, or adaptive schemes which do not limit dilation to $V(\mathbf{r}_k)$ [56], [57] would not be effective in this respect. Second, if no such intersection is performed, undesired spikes would be present on high curvature points, which may have a devastating effect on the grouping process (Fig. 4). As a result, very long ellipses can be used without negatively affecting $\mathcal{D}_b$ and, consequently, the grouping process; such long ellipses are necessary for grouping together collinear edges that are quite distant, like points C and D in Fig. 3(a).

B. Automatic Adaptive Selection of the LSE Size

We now describe a simple adaptive procedure that computes the size of the LSE associated with each edge point. For each connected component C_k of the input binary image b, we assign a single value $2a_1^{(k)}$ to the length of the major axis of each ellipse, and a value $2a_2^{(k)}$ to the minor axis. Thus, all edges of the same connected component are dilated by ellipses of the same size.

The length $2a_2^{(k)}$ of the minor axis is computed as follows: let $V(C_k)$ be the Voronoi cell of C_k, defined as the union of all the Voronoi cells $V(\mathbf{r}_i)$ of the points $\mathbf{r}_i$ that belong to C_k

$$V(C_k) \triangleq \bigcup_{\mathbf{r}_i \in C_k} V(\mathbf{r}_i). \tag{4}$$

Let $\mathbf{r} \in C_k$ be an edge point and let $\mathbf{q}(\mathbf{r})$ be the nearest neighbor of $\mathbf{r}$ on the boundary of $V(C_k)$ (Fig. 5). We indicate by $\rho(\mathbf{r})$ the Euclidean distance between $\mathbf{r}$ and $\mathbf{q}(\mathbf{r})$, and by $\rho_H^{(k)}$ the *harmonic mean* of $\rho(\mathbf{r})$, for $\mathbf{r} \in C_k$

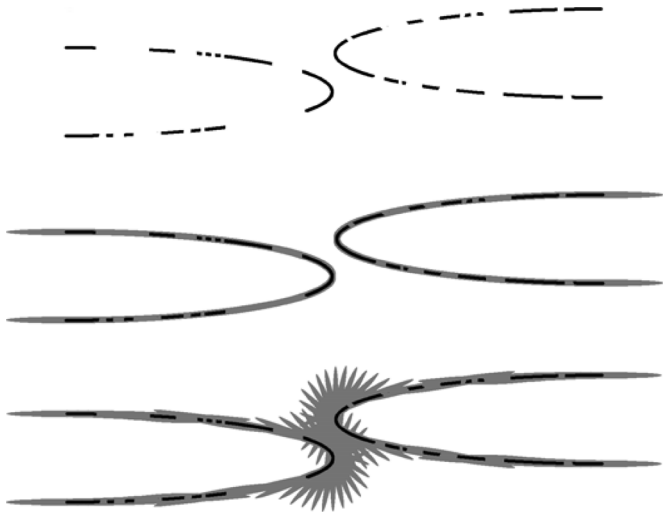

Fig. 4. Benefit that the intersection between each ellipse and the corresponding Voronoi cell brings to the grouping of high curvature edges. From top to bottom: input set of incomplete contours and groupings induced by APD and by the output that would be obtained without the intersection with the Voronoi cells.

$$\rho_H^{(k)} = \left[\frac{1}{N_k} \sum_{\mathbf{r}_i \in C_k} \frac{1}{\rho(\mathbf{r}_i)} \right]^{-1} \tag{5}$$

where N_k is the number of edge points in C_k. In this notation, the length $a_2^{(k)}$ of the minor semi-axis is computed as a monotonically increasing function f of $\rho_h^{(k)}$, which satisfies the conditions $f(0) = 0$ and $f(\xi) < \xi$ for $\xi \geq 0$

$$a_2^{(k)} = f\left[\rho_H^{(k)}\right]. \tag{6}$$

Since smaller values of $\rho(\mathbf{r})$ have higher influence on the harmonic mean, and being $f(\xi) < \xi$, the structuring element is always narrower than the Voronoi cell $V(C_k)$. In this way, the dilation orthogonally to the edge direction θ prevents that noncollinear components that run in parallel are grouped together (such as C_1 and C_3 in Fig. 6). On the other hand, $a_2^{(k)}$ is sufficiently large to compensate for slight misalignment between collinear components, such as C_1 and C_2 in Fig. 6. Pos-

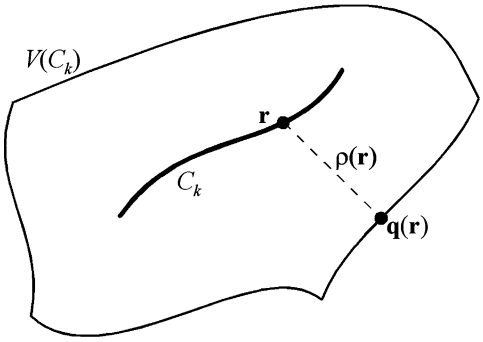

Fig. 5. Computation of the minor axis of an elliptic structuring element.

Fig. 6. Width of the structuring element is adaptively computed such that the slight misalignment between C_1 and C_2 is compensated, while keeping separated components C_1 and C_3 that run in parallel.

Fig. 7. Two different structuring elements are intersected with the Voronoi cell V of a given point. If the structuring element is longer than V, the result of the pseudo-dilation does not change substantially with the length of the structuring element.

sible choices for the function f are $f(\xi) = \log(\xi + 1)$, or $f(\xi) = (\xi + a)^q - a$, with $a > 0$ and $q \in (0, 1)$.

Concerning the length of the major axis, we observe that if it exceeds the length of the corresponding Voronoi cell, the actual value of $a_1^{(k)}$ only has a small influence on the final output (Fig. 7). In our experiments, we found that the choice $a_1^{(k)} = \eta a_2^{(k)}$, with $\eta \geq 3$, is sufficiently accurate for this purpose. Specifically, with such a choice of $a_1^{(k)}$, almost all points that belong to long chains of collinear edge points are dilated with an ellipse that intersects the corresponding Voronoi cell; thus, they will be grouped together. On the other hand, almost all isolated edge points are dilated with ellipses which do not intersect the corresponding Voronoi cells and, consequently, they are not grouped together. This is in agreement with the GLGC.

In Fig. 8, an example of how this procedure works is presented. We observe that, due to the structure induced by the Voronoi tessellation, the LSE is large on isolated contours and small on texture, so that the former have much higher probability to be grouped together than the latter.

IV. An Application: Contour Detection

In this section, APD and the grouping algorithm introduced above are applied to the problem of contour detection in natural images. The proposed approach is depicted in Fig. 9.

Like many existing edge detectors, we first compute a *local quantity* $L(\mathbf{r})$ that measures the visual similarity of a local pattern to an ideal step edge. Examples of such quantities are the gradient magnitude [1], Gabor energy [26], phase congruency [14], template matching [18], and texture gradient [19]. In our experiments, $L(\mathbf{r})$ is a quantity called *contourness*, introduced in [29] and [58] with the purpose of suppressing texture, and successively improved in [59]–[61]. Thinning by nonmaxima suppression, which is also a local operation, is included in the computation of $L(\mathbf{r})$.

Then, N_t binary images $b_1 \subset b_2 \subset \cdots \subset b_{N_t}$ are obtained by thresholding $L(\mathbf{r})$ at N_t different levels $\tau_1 > \tau_2 > \cdots > \tau_{N_t}$

$$b_k = \{\mathbf{r} | L(\mathbf{r}) > \tau_k\}, \quad k = 1, \ldots, N_t. \tag{7}$$

Each threshold τ_k is computed such that a given percentage p_k of the values of $L(\mathbf{r})$ is above τ_k. Examples of such binary images are shown in Fig. 10. As the threshold decreases, the corresponding binary map contains more object contours, but more undesired responses as well.

In order to remove undesired edges, we apply a pruning step to each binary map b_k (block "GLGC pruning" of Fig. 9). The idea is to group edges by means of the grouping algorithm presented in Section III and to remove those groups that contain less than $l_{\min}$ edges, where $l_{\min}$ is a threshold. Specifically, the pruning operator $\mathrm{pru}(\cdot)$ is defined as

$$\mathrm{pru}(b) \triangleq \{\pi_k \in b | \operatorname{card}\{\pi_k\} \geq l_{\min}\} \tag{8}$$

where π_j are the groups detected by our algorithm and $\operatorname{card}\{X\}$ indicates the number of elements of set X. We chose to set $l_{\min}$ proportional to the linear size of the image, $l_{\min} = \xi d$, where d is the length of the diagonal of the input image and ξ is a proportionality factor. In our experiments, we found that for $\xi = 0.03$ small groups π_k, for which card $\{\pi_k\} < \xi d$, do not contain any significant information; therefore, they can be suppressed.

Fig. 11 shows the outputs $\bar{b}_1$, $\bar{b}_2$, and $\bar{b}_3$ of pruning for the binary images b_1, b_2 and b_3 respectively. Remarkably, a large amount of texture edges is removed while preserving most of the object contours; this is due to the fact that edge grouping makes pruning robust to contour fragmentation.

We observe that the pruned contour maps at higher thresholds contain a very little amount of texture, but several useful contours might be missing; such contours are still present in the pruned contour maps at lower thresholds. In order to recover them, we introduce a simple operator (block "APD Reconstruction") that generalizes the well known reconstruction from markers (RFM). By definition, standard RFM takes in input two binary images B_1 and B_2, with $B_1 \subseteq B_2$, and returns the set of those connected components π_i of B_2 which contain at least one point of B_1

$$\mathrm{rfm}(B_1, B_2) \triangleq \{\pi_i \in B_2 | \pi_i \cap B_1 \neq \emptyset\}. \tag{9}$$

We propose to perform RFM based on the connectivity induced by APD. Specifically, our operator $R_{\mathrm{APD}}(\cdot)$ is defined as follows:

$$R_{\mathrm{APD}}(B_1, B_2) \triangleq \mathrm{rfm}[\mathrm{apd}(B_1), \mathrm{apd}(B_2)] \cap B_2. \tag{10}$$

In Fig. 12(a), we show the behavior of $R_{\mathrm{APD}}(\cdot)$, which is applied to the results $\bar{b}_k = \mathrm{pru}(b_k)$ and $\bar{b}_{k+1} = \mathrm{pru}(b_{k+1})$ of pruning. As we see, contours in $\tilde{b}_{k+1} = R_{\mathrm{APD}}(\bar{b}_k, \bar{b}_{k+1})$ are as complete as in $\bar{b}_{k+1}$, while the undesired edges of $\bar{b}_{k+1}$ that

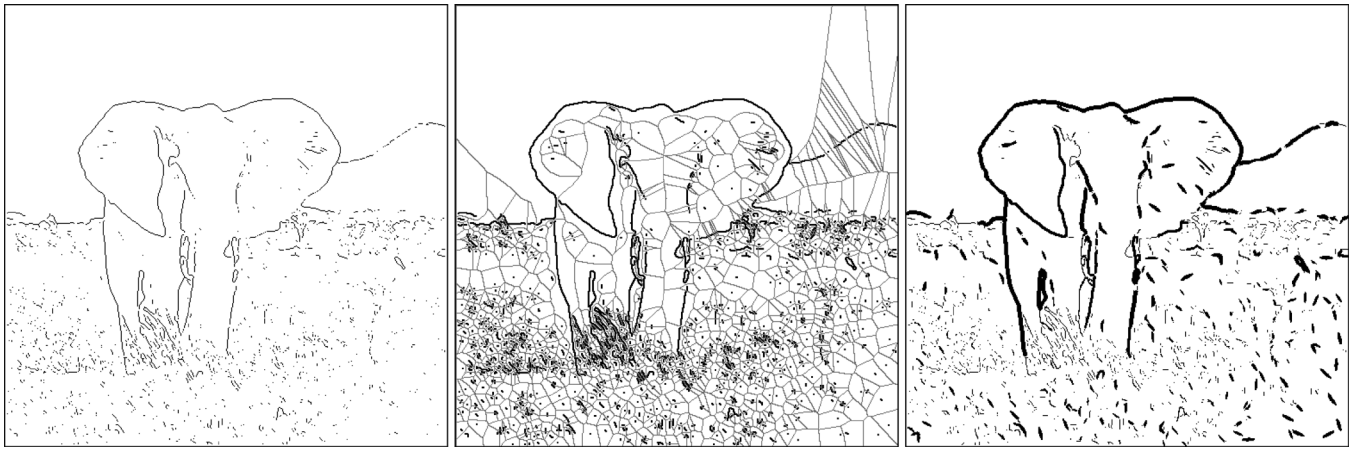

Fig. 8. Illustration of the proposed procedure for adaptively computing the size of the LSE. From left to right: Input binary map, the Voronoi tessellation of its connected components, and the output of APD. The LSE is large on isolated contours and small on texture, so that the former have much higher probability to be grouped together than the latter.

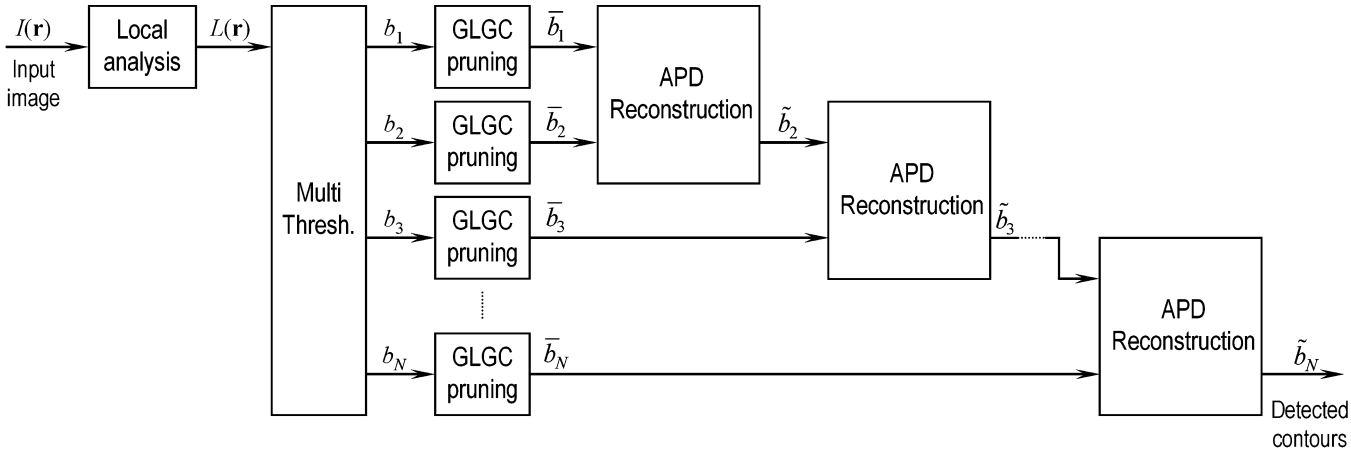

Fig. 9. Overall scheme of the proposed contour detector.

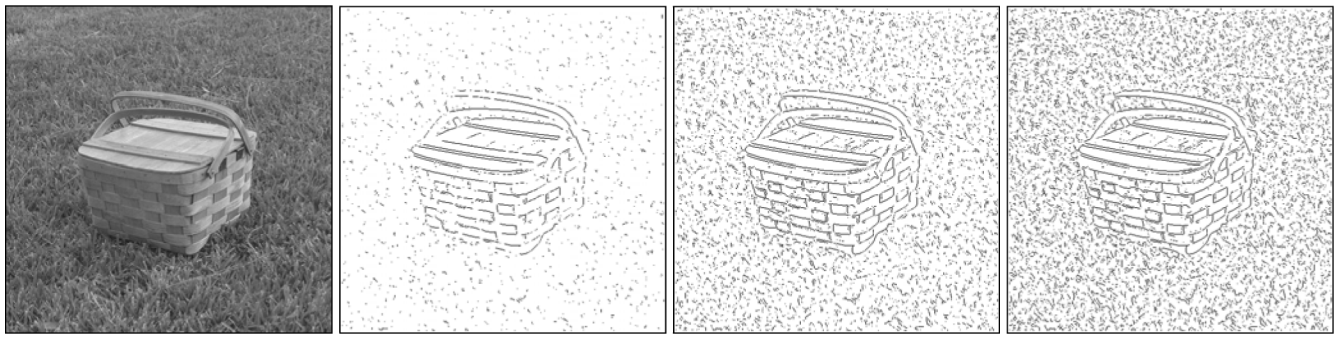

Fig. 10. Input image and the binary maps obtained by thresholding $L(\mathbf{r})$. The thresholds correspond to the values $p_1 = 0.13$, $p_2 = 0.35$, and $p_3 = 0.5$.

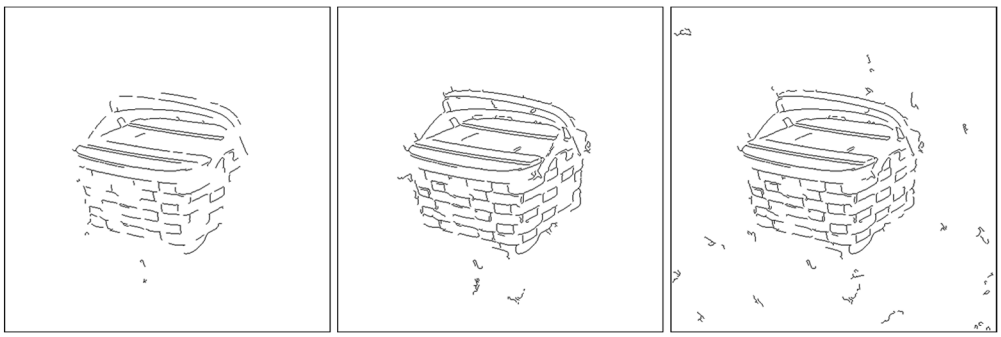

Fig. 11. Outputs $\bar{b}_1$, $\bar{b}_2$, and $\bar{b}_3$ of the block "GLGC pruning" applied to the binary maps b_1, b_2, and b_3 of Fig. 10.

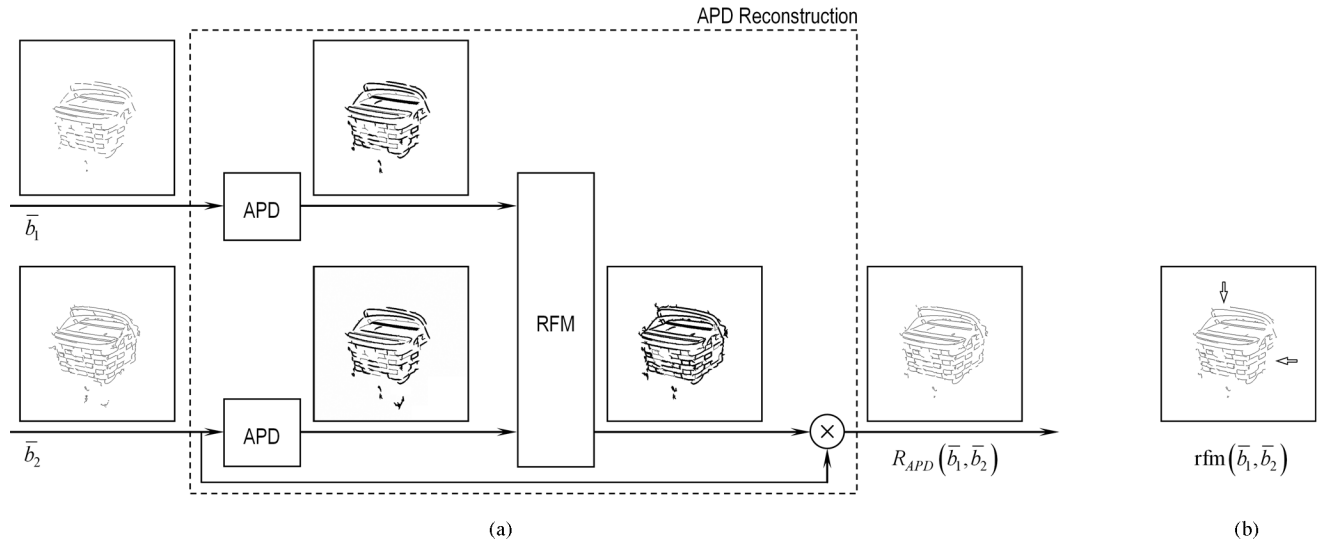

Fig. 12. (a) Illustration of $R_{\mathrm{APD}}(\cdot)$, applied to the images $\bar{b}_k = \mathrm{pru}(b_k)$ and $\bar{b}_{k+1} = \mathrm{pru}(b_{k+1})$. (b) Output of rfm $(\bar{b}_1, \bar{b}_2)$, where the arrows mark contours that would be lost if RFM was applied without APD.

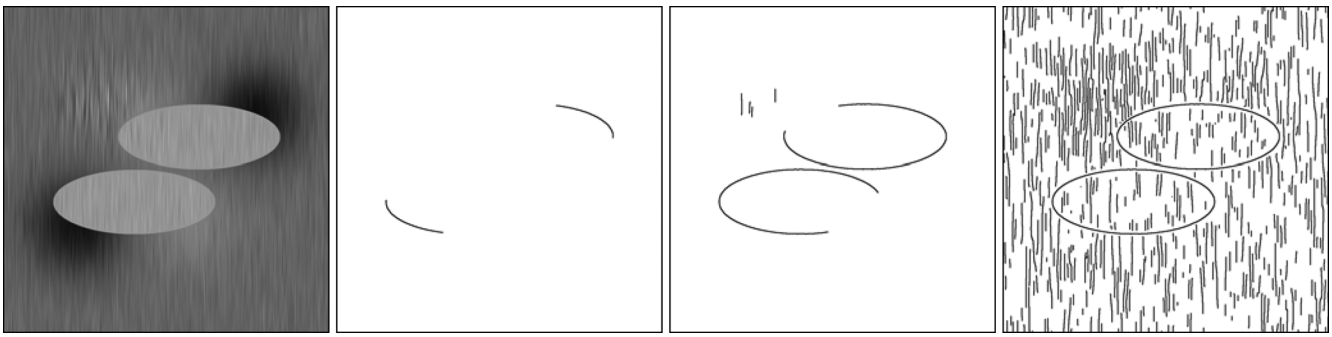

Fig. 13. Synthetic image and the binary maps $b_1 \subseteq b_2 \subseteq b_3$ obtained by thresholding $L(\mathbf{r})$.

are not present in $\bar{b}_k$ are not present in $\tilde{b}_{k+1}$ either. Performing RFM on the dilated images, instead of on $\bar{b}_k$ and $\bar{b}_{k+1}$ directly, makes the operator more robust to contour fragmentation; therefore, more contours can be recovered with respect to the standard RFM [Fig. 12(b)].

An important difference between classical RFM and R_{APD} concerns their behaviors when they are applied in a cascade. Specifically, given three binary maps $b_1 \subseteq b_2 \subseteq b_3$, it is well known that a cascade of two RFMs applied on b_1, b_2, and b_3 is equivalent to a single RFM applied directly to b_1 and b_3

$$\mathrm{rfm}[\mathrm{rfm}(b_1, b_2), b_3] = \mathrm{rfm}(b_1, b_3). \tag{11}$$

However, (11) does not hold for R_{APD}, due to the fact that $b_1 \subseteq b_2$ does not imply $\mathrm{apd}(b_1) \subseteq \mathrm{apd}(b_2)$. Let us consider, as an example, the synthetic image of Fig. 13 and the binary maps $b_1 \subseteq b_2 \subseteq b_3$ obtained by thresholding $L(\mathbf{r})$. Fig. 14 compares a cascade of two R_{APD} applied on b_1, b_2, and b_3 (top), and a single R_{APD} applied on b_1 and b_3 directly (bottom). While in the former case contours are correctly detected, in the latter a considerable amount of undesired response is present, due to the very large LSE which are used to dilate b_1. Large LSE are useful to group sparse objects that typically occur at high thresholds. Therefore, the insertion of an intermediate layer that bridges the large LSE of b_1 and the crowded edges of b_3 proves to be beneficial.

V. Experimental Results

We now present and comment the outputs of the proposed contour detector, tested with natural images, in comparison with other three existing algorithms: the standard Canny edge detector [1], the contour detector with surround inhibition (SI) proposed in [58], and the edge detector EDISON proposed in [18]. Canny edge detector computes the gradient of the input image $I(\mathbf{r})$, and applies nonmaxima suppression and hysteretic thresholding on the gradient magnitude [1]. SI improves the Canny edge detector by replacing the gradient magnitude with a quantity called *contourness* that takes into account the local surrounding context of each pixel [58]. EDISON introduces a two-dimensional feature vector in order to discriminate between contour pixels and undesired edge responses: the first component is the gradient magnitude and the second one is derived from a linear template matching analysis.

We tested the four algorithms on the 40 natural images of the RuG dataset [58] and the 300 images of the Berkley dataset [62]. For all test images, the results of our algorithm are obtained with the *same values* of the input parameters, which are

Fig. 14. Top: Cascade of two R_{APD} applied on b_1, b_2, and b_3. Bottom: Single R_{APD} applied on b_1 and b_3 directly.

the following: the number of thresholds is $N_t = 3$; the values τ_k of the thresholds are computed for each image, such that a fixed fraction p_k of pixels is above τ_k (in our experiment, we used the values $p_1 \cong 0.07$, $p_2 \cong 0.12$ and $p_3 \cong 0.17$). In Section V-B, we show that the performance of the proposed algorithm does not depend very much on the specific values of these thresholds. For the other three algorithms, the parameters have been optimized as suggested in the corresponding references. We present both a qualitative and quantitative comparison of the studied algorithms.

A. *Qualitative Comparison*

Some examples of the outputs of the considered algorithms are shown in Fig. 15(a). The results related to the entire datasets are available online.[1] As we see, the outputs of the proposed algorithm (second row) contain the smallest amount of undesired edges and the most complete contours. For instance, in the first picture of Fig. 15(a) the basket has very low contrast contours, whereas the grass presents a very high-contrast texture. Nonetheless, the proposed algorithm succeeds very well in isolating contours and rejecting texture, while the other approaches fail. The second example (elephant) presents a quite low-contrast texture in the grass, but also low contrast region boundaries, such as the contour of the hill in the top-right corner. We see that in such cases the simple Canny edge detector (third row) can achieve a decent texture suppression as well, but at the expense of the loss of low contrast contours. On the opposite, the proposed approach and SI (fourth row) can suppress texture while preserving low-contrast contours. EDISON (fifth row) can preserve low-contrast contours, but is less effective in texture suppression. The third example (bear) exhibits textures that occupy low-frequency regions of the power spectrum, such as the plants in the background of the bear. Such texture is quite difficult to

[1]http://www.cs.rug.nl/~imaging/APD

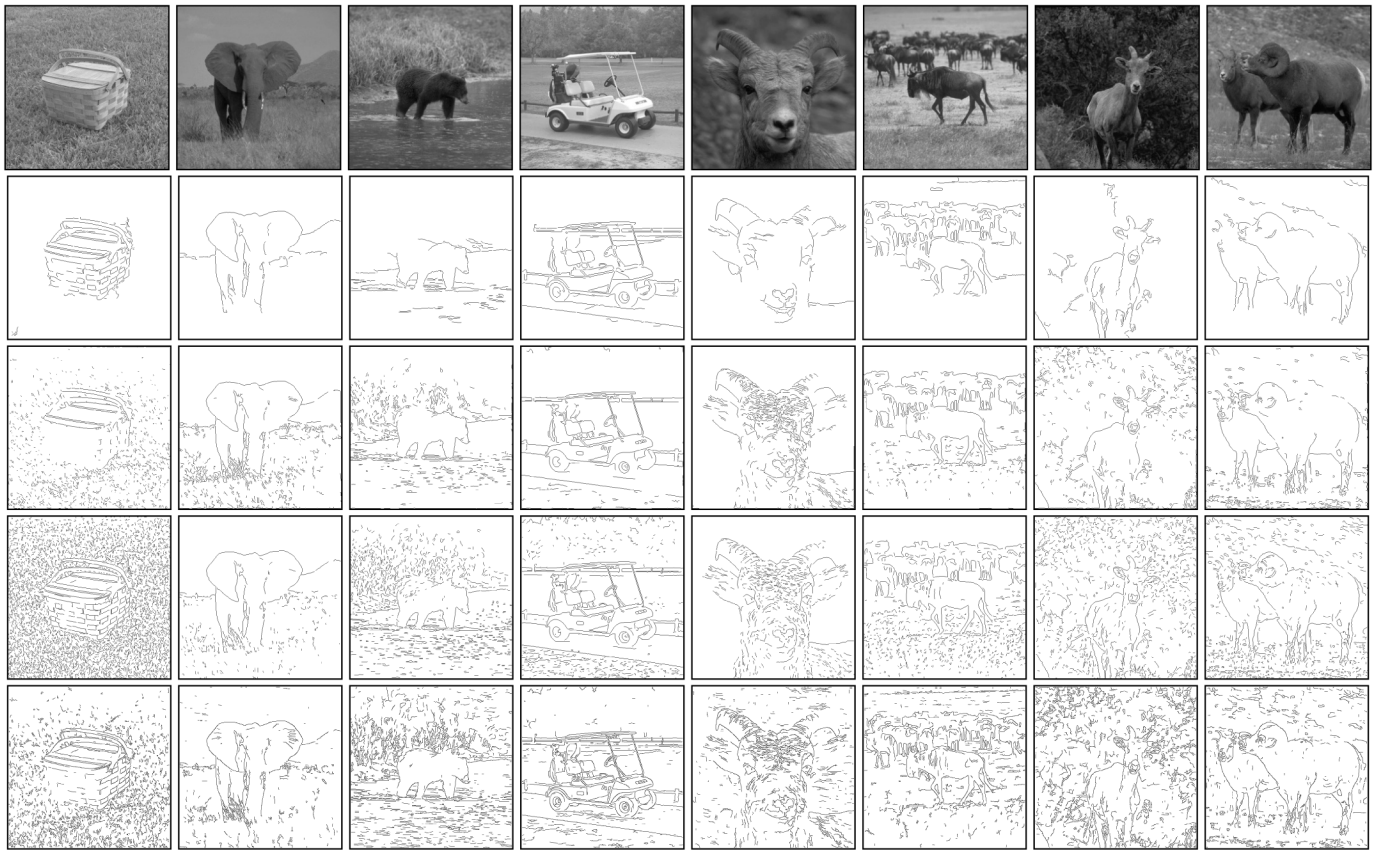

Fig. 15(a). Eight test images (first row) and contours detected with: our approach (second row), the standard Canny edge detector ($p1 = 0.02$) [1] (third row), the contour detector with surround inhibition proposed in [58] (fourth row), and the edge detector EDISON described in [18] (fifth row).

remove. We see that the proposed algorithm can eliminate such undesired edge response, while other approaches are unable to do it. Other examples of images that present low-frequency texture are shown in Fig. 15(a) (second-last column), especially in plants in the background in the picture of the goat.

The last example in Fig. 15(a) shows the ability of the proposed method to extract fragmented contours by means of the GLGC-based edge grouping: the very low contrast diagonal line in the background, which separates two differently textured areas, is detected by our approach quite well, whereas with the other approaches such a line is either missing or masked by many other undesired edges. In general, we can conclude that the proposed operator outperforms the other studied approaches in terms of texture suppression, preservation of low-contrast contours and robustness to contour fragmentation.

B. Quantitative Comparison

We measured the dissimilarity between the output of each considered algorithm and a hand drawn desired output (ground truth). Well known performance indicators are *recall* R and *precision* P, which relate the number of correctly detected contour pixels to the number of ground truth (GT) pixels and the number of contour pixels that are actually detected by an algorithm (DC), respectively

$$R = \frac{\text{card}\{DC \cap GT\}}{\text{card}\{GT\}}; \quad P = \frac{\text{card}\{DC \cap GT\}}{\text{card}\{DC\}}. \quad (12)$$

Following [19], we evaluate the performance in terms of the so-called F-*measure*, defined as the harmonic mean of P and R, $F = 2PR/(P+R)$.

Fig. 16 shows the average values and the standard deviations of F, computed on both datasets for all studied algorithms. The average value of F achieved with the proposed approach is about 20% higher than the values achieved by the other algorithms. In particular, the comparison with SI, which deploys the same local edge strength $L(\mathbf{r})$ that we use here, directly validates the superiority of the multithreshold algorithm presented in Section IV, which is based on APD and edge grouping, with respect to the classical hysteresis thresholding.

Fig. 17 compares the dependence on the thresholds of the performances achieved by the proposed approach and the Canny edge detector: for both algorithms, $\tilde{F}$ is plotted versus the fraction p_1 of pixels that are above the highest threshold. For both algorithms, the values of p that correspond to the lower thresholds are chosen proportional value to p_1. Canny edge detector exhibits a peaked maximum for $p_1 = 0.014$ and decreases quickly as p_1 deviates from that value. In contrast, the proposed approach reaches the maximum values of $\tilde{F}$ for a higher value of $p_1 (p_1 \cong 0.057)$ and the performance decreases much more slowly as p_1 increases.

VI. Discussion

Adaptive mathematical morphology extends classical morphological operators to structuring elements that are context-de-

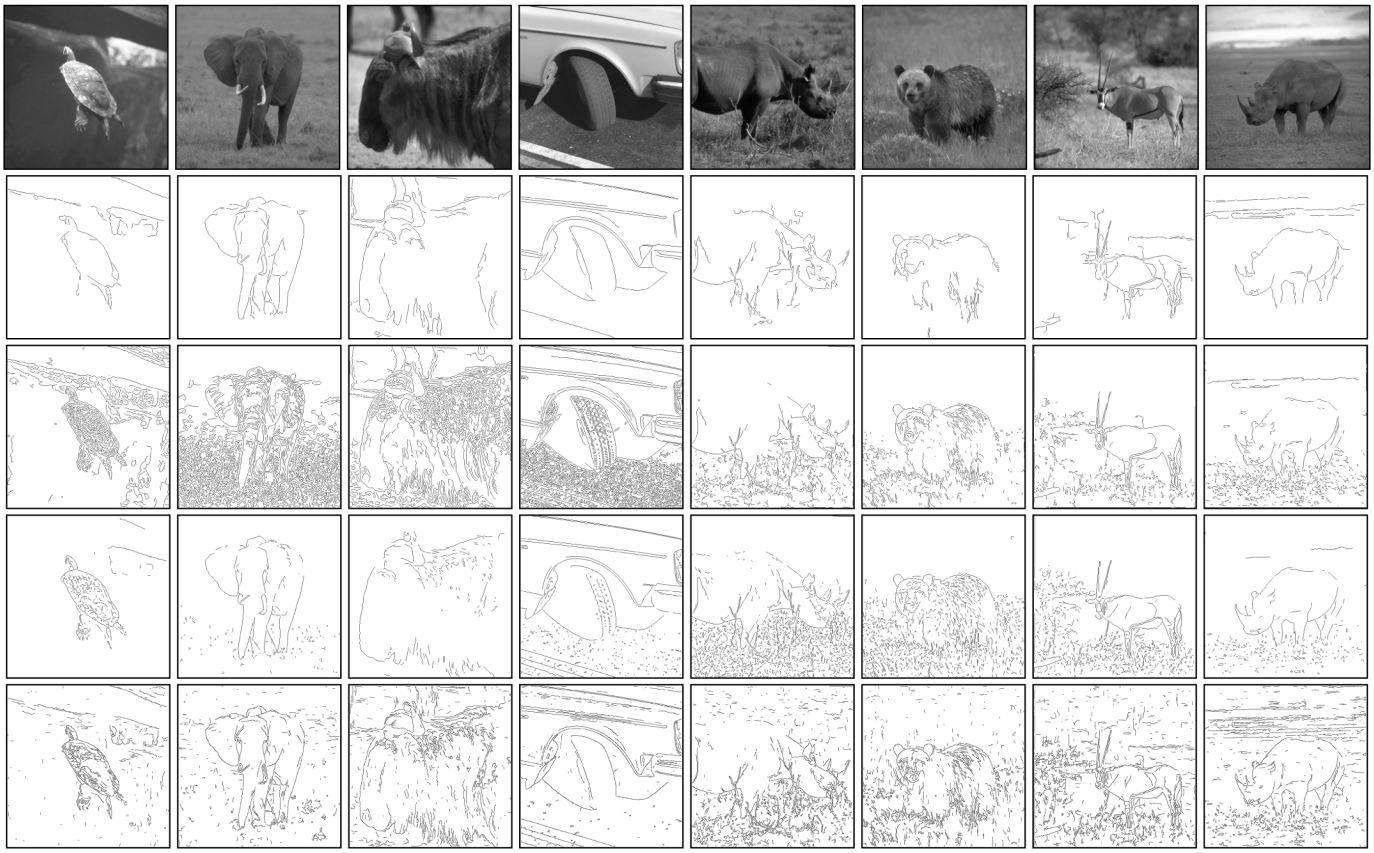

Fig. 15(b). Same as Fig.15(a) for the other eight test images.

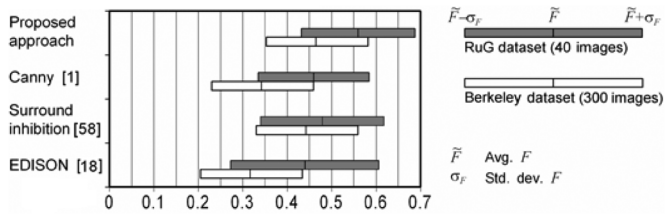

Fig. 16. Statistics of F on the dataset considered in [58] and the Berkeley dataset [62].

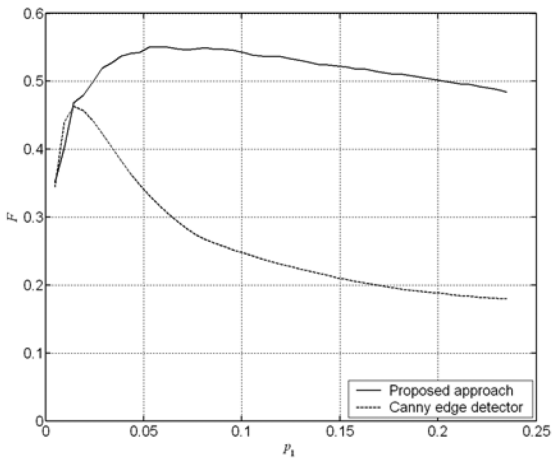

Fig. 17. Values of the F-measure achieved by the proposed approach and the Canny edge detector as function of p_1, averaged over the 40 images of the Rug Dataset [58]. The proposed approach achieves higher performance and exhibits a much less critical dependence on the threshold.

pendent [56], [57], [63], [64]. The novelty of APD is that dilation is limited to the Voronoi cell of each edge point, which brings several advantages since long ray spatial relations between edge points are taken into account in dilation process. Moreover, APD can be computed efficiently by simply thresholding the quantity $\Delta_{\mathcal{R}}[\mathbf{r}, \mathbf{q}(\mathbf{r})]$ as in (3).

In previous literature, different approaches have been proposed for adaptive dilation. In [56], a nonlinear diffusion equation is designed in order to perform dilation only along a preferred direction. While APD only works in the binary case, operators based on nonlinear diffusion can be applied to gray-level images as well. However, they are computationally demanding due to their iterative nature and stopping rules are usually left to heuristics. On the other hand, multithreshold schemes, such as the one presented in Section IV, allow to use APD with grey-level images, too. A noniterative algorithm for adaptive dilation is proposed in [57], with applications to edge linking. It consists in replacing each edge point with an ellipse whose major axis is oriented along the local edge direction. However, it has problems in presence of high curvature points and with edges that are close to each other but which are not collinear (see, for instance, Figs. 3 and 4). These limitations are overcome by the proposed operator, by intersecting the LSE with the corresponding Voronoi cell. More recently, higher degree

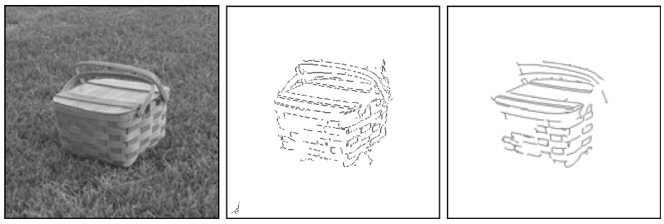

Fig. 18. From left to right: Input image, output of the proposed method, and output that would be obtained if the multithreshold algorithm of Section IV was applied without APD. The loss of object contours in the latter case is evident.

of adaptivity has been achieved by morphological amoeba filters [64]. However, their computation time grows quadratically with respect to the linear size $l_{\max}$ of the LSE, which could be quite large for edge grouping purposes, whereas the complexity of APD does not depend on $l_{\max}$.

APD has some mathematical relationships with the classical structural dilation. Specifically, if the construction function is equal to $\Delta_{\mathcal{R}}(\mathbf{r}, \mathbf{r}_i) = |\mathbf{r}-\mathbf{r}_i|/R$, then APD reduces to a dilation with a disk of radius R as structuring element. In general, an operator $\mathcal{D} : X \mapsto X$, where X is the space of all non-negative digital signals, is called *dilation* if the following requirements are fulfilled:

$$\begin{cases} \mathcal{D}(\alpha x_1) = \alpha \mathcal{D}(x_1) \\ \mathcal{D}[\sup(x_1, x_2)] = \sup[\mathcal{D}(x_1), \mathcal{D}(x_2)] \end{cases} \tag{13}$$

with $x_1, x_2 \in X$ and $\alpha \in \mathbb{R}$. For the binary case, the first condition is replaced by $\mathcal{D}(\emptyset) = \emptyset$ and the supreme reduces to the union. It is easy to show that the proposed operator does not fulfill the second requirement (13). However, it can be easily shown that the only possible operator which fulfills the requirements (13) is the standard non adaptive dilation (see Appendix B). Therefore, as soon as adaptivity is desired, the constraint of commutation with the supreme must be weakened in some way.

We use APD to group edges according to the GLGC: collinearity is taken into account by elongated LSE, while the neighborhood induced by the Voronoi tessellation effectively models the Gestalt law of proximity [65], [66]. In our experiments, we used ellipses as LSEs for reasons of elegance. However, any other elongated shape could be used as well.

The proposed approach for contour detection is a combination of two techniques: a grouping algorithm described in Section III, and the iterative multithreshold process presented in Section IV. Both of them are necessary to achieve good results. In fact, the grouping algorithm itself simply groups edges together, without touching the edge map. Texture suppression and contour completion are achieved by the pruning/reconstruction process, which is not part of the grouping algorithm. On the other hand, the multithreshold algorithm of Section IV would perform very poorly without edge grouping. In fact, in case of a strongly fragmented contour, a pruning process based on the number of pixels of each connected component would be totally unable to distinguish between fragmented object contours and spurious edges. An example is shown in Fig. 18.

Grouping edges based on their Voronoi tessellation has some duality with the algorithm proposed in [50], on which the concept of Constrained Delaunay triangulation is introduced. However, even when the input parameters are optimized for a specific dataset, a considerable amount of useful contours is missed, especially for what concerns internal details of the represented objects, such as the facial details in Fig. 19. In contrast, our method results in a much better contour preservation without increasing the amount of undesired edge response. The reason is that in [50] thresholding and grouping are performed in two separate steps, whereas we propose to group edges at each threshold and to combine together the results of different groupings.

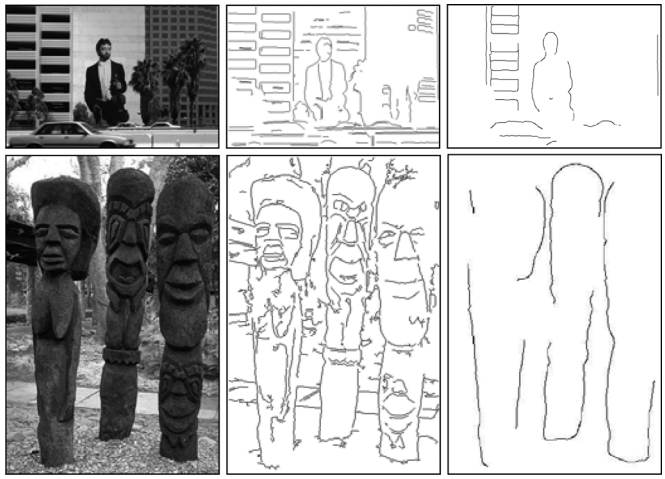

Fig. 19. From left to right: Two input images and contours detected with our approach and with the algorithm proposed in [50]. The method proposed here results in a much better contour preservation.

The algorithm presented in Section IV is a postprocessing step, which operates on a local quantity $L(\mathbf{r})$ that is high in presence of edges. In our experiments, we use the local contourness proposed in [26], which can be computed by means of only two convolutions. On the other hand, higher performance can be achieved by means of local edge strengths based more sophisticated schemes. For instance, the average value of F that is achieved with the approach proposed in [19] for the Berkeley dataset ($\tilde{F} = 0.63$) is higher than the one that we achieve ($\tilde{F} = 0.47$). However, the local edge strength $p_B(\mathbf{r})$ proposed in [19] has two drawbacks: first, it is computationally demanding, as it requires 13 convolutions with Gabor filters and several additional nonlinear machine learning operations. Second, the resulting contours are blurred (Fig. 20). As pointed out in [67], most of the procedures used for quantitative performance evaluation do not penalize sufficiently such undesired smoothing.

In order to measure the speed of the proposed contour detector, we run its Matlab implementation 1000 times for a gray-level image (321 $\times$ 481 pixel) on a Pentium 4 CPU 3.00 GHz, under Windows XP. Table I details the time required for each step, averaged over the 1000 runs.

VII. Summary and Conclusions

The main contribution of this paper is a new morphological operator, called *adaptive pseudo-dilation*, that we use for edge grouping according to the GLGC. A simple and computationally efficient algorithm for APD is provided. The operator defined in (4) is a *generalized Minkowski sum*, where a context-dependent structuring element is intersected with the Voronoi cell of

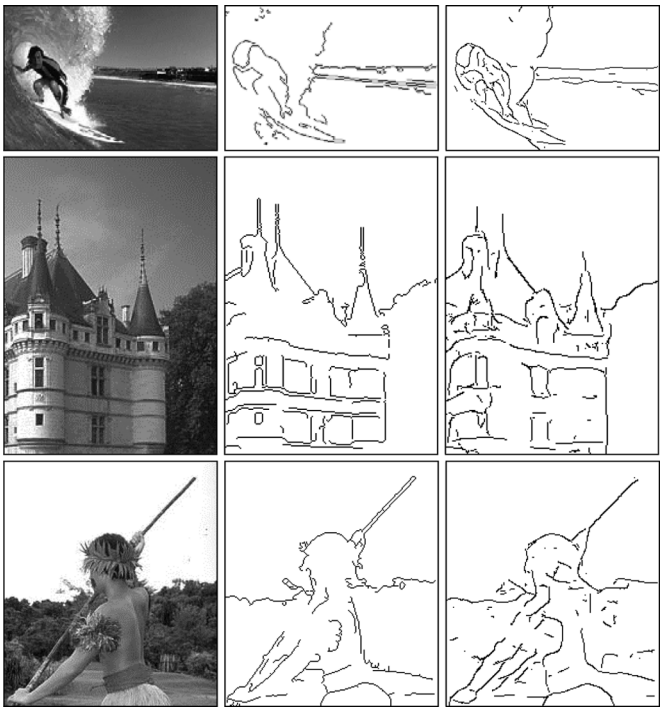

Fig. 20. From left to right: Input images and contours detected with our approach and the one proposed in [19]. Concerning the latter one, the local edge strength $p_B(\mathbf{r})$ has been thresholded at the level $p_{\min} = 0.1$.

TABLE I
TIME REQUIRED BY EACH STEP PERFORMED BY THE PROPOSED ALGORITHM

Operation	Time (s)
Local edge strength *L*(r) (including thinning)	**1.754**
GLGC Pruning (T_1 + T_2 + T_3)	**0.603**
Adaptive size structuring element (T_1)	0.269
Adaptive pseudo-dilation (T_2)	0.237
Elimination of small groups (T_3)	0.097
APD Reconstruction	**1.208**

the corresponding point. Such an intersection brings several advantages with respect to other adaptive morphological operator, especially in presence of high curvature points.

We used APD for contour detection in combination with a multithreshold algorithm that generalizes hysteresis thresholding. We tested the proposed approach on a large set of natural images. Both qualitative and quantitative comparison show the superiority of the proposed approach with respect to the other studied techniques in terms of texture suppression, preservation of low contrast contours, and insensitiveness to the values of the input parameters.

APPENDIX A

We formally prove that APD defined in (1) is equal to the set A of those points for which $\Delta_{\mathcal{R}}[\mathbf{r}, \mathbf{q}(\mathbf{r})] < 1$. Specifically, we prove the following.

Theorem:

Hypothesis:

$$\mathbf{r} \in A \Longleftrightarrow \Delta_{\mathcal{R}}(\mathbf{r}, \mathbf{r}_k) < 1, \quad r \in \mathbb{R}^2, \ \ \mathbf{r}_k \in b$$
$$\mathbf{r} \in A \Longleftrightarrow \Delta_{\mathcal{R}}[\mathbf{r}, \mathbf{q}(\mathbf{r})] < 1$$

Thesis:

$$A = \bigcup_k R_k \cap V(\mathbf{r}_k).$$

Proof: By the definition of $\mathbf{q}(\mathbf{r})$, we have $\mathbf{q}(\mathbf{r}) = \mathbf{r}_k, \forall\, \mathbf{r} \in V(\mathbf{r}_k)$. Therefore, we have

$$\Delta_{\mathcal{R}}(\mathbf{r}, \mathbf{r}_k) < 1 \Longleftrightarrow \Delta_{\mathcal{R}}[\mathbf{r}, \mathbf{q}(\mathbf{r})] < 1, \quad \forall\, \mathbf{r} \in V(\mathbf{r}_k). \quad \text{(A1)}$$

Or, equivalently, by the hypothesis

$$A \bigcap V(\mathbf{r}_k) = R_k \bigcap V(\mathbf{r}_k). \quad \text{(A2)}$$

Let us take the union of both the sides, on all the points $\mathbf{r}_k \in b$

$$\bigcup_k A \cap V(\mathbf{r}_k) = \bigcup_k R_k \cap V(\mathbf{r}_k). \quad \text{(A3)}$$

With basic calculus, and the observation that the union of all the Voronoi cells $V(\mathbf{r}_k)$ is the entire plane, we can show that the first member of (A3) is equal to A. This completes the proof.

APPENDIX B

Let $x[k]$ be the kth sample of a positive digital signal x, $x[k] > 0$. We prove that under the hypothesis (13), any shift-invariant operator $\mathcal{D}$ must be expressed by a generalized convolution in the following form:

$$\{\mathcal{D}(x)\}[k] = \sup_s x[s] w[k - s] \quad \text{(B1)}$$

where $w[k]$ is the structuring element.

Proof: Combining the two requirements (13), we obtain

$$\mathcal{D}\left(\sup_k \alpha_k x_k\right) = \sup_k \alpha_k \mathcal{D}(x_k). \quad \text{(B2)}$$

We observe that, for *any* signal x, the following equality holds:

$$x = \sup_k x[k]\, \delta_k, \quad \delta_k[s] = \begin{cases} 1, & s = k \\ 0, & s \neq k. \end{cases} \quad \text{(B3)}$$

Therefore, combining (B2) and (B3), we have

$$\mathcal{D}(x) = \mathcal{D}\left(\sup_k x[k]\, \delta_k\right) = \sup_k x[k]\, \mathcal{D}(\delta_k). \quad \text{(B4)}$$

As decomposition (B3) is applicable to *any* signal, it holds for $\mathcal{D}(\delta_k)$ as well. Therefore, we can write $\mathcal{D}(\delta_k) = \sup_s w_{k,s} \delta_s$, where $w_{k,s}$ is the sth sample of $\mathcal{D}(\delta_k)$. Since $\mathcal{D}$ is shift-invariant, it must be $w_{k,s} = w[s - k]$. Therefore, (B4) becomes

$$\mathcal{D}(x) = \sup_{k,s} x[k]\, w[s - k]\, \delta_s \quad \text{(B5)}$$

which is equivalent to (B1). This completes the proof. This proof holds for n-dimensional signals as soon as the indexes k and s are meant to be n-dimensional.

REFERENCES

[1] J. F. Canny, "A computational approach to edge detection," *IEEE Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Jun. 1986.

[2] W. Frei and C. Chen, "Fast boundary detection: A generalization and a new algorithm," *IEEE Trans. Computers*, vol. 26, no. 10, pp. 988–998, Oct. 1977.

[3] E. C. Hildreth, "The detection of intensity changes by computer and biological vision systems," *Comput. Vis., Graph., Image Process.*, vol. 22, pp. 1–27, 1983.

[4] J. B. Martens, "Local orientation analysis in images by means of the hermite transform," *IEEE Trans. Image Process.*, vol. 6, no. 8, pp. 1103–1116, Aug. 1997.

[5] R. Nevatia and K. Babu, "Linear feature extraction and description," *Comput. Graph. Image Process.*, vol. 13, pp. 257–269, 1980.

[6] P. H. Gregson, "Using angular dispersion of gradient direction for detecting edge ribbons," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 7, pp. 682–696, Jul. 1993.

[7] O. A. Zuniga and R. M. Haralick, "Integrated directional derivative gradient operator," *IEEE Trans. Syst. Man, Cybern.*, vol. 17, no. 3, pp. 508–517, Mar. 1987.

[8] G. Chen and Y. H. H. Yang, "Edge detection by regularized cubic B-spline fitting," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 4, pp. 635–642, Apr. 1995.

[9] S. Ghosal and R. Mehrotra, "Detection of composite edges," *IEEE Trans. Image Process.*, vol. 3, no. 1, pp. 14–25, Jan. 1994.

[10] R. M. Haralick, "Digital step edges from zero-crossings of second directional derivatives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 1, pp. 58–68, Jan. 1984.

[11] V. S. Nalwa and T. O. Binford, "On detecting edges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 699–714, Jun. 1986.

[12] T. Folsom and R. Pinter, "Primitive features by steering, quadrature and scale," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1161–1173, Nov. 1998.

[13] F. Heitger, "Feature detection using suppression and enhancement," Tech. Rep. TR-163, Commun. Technol. Lab., Swiss Fed. Inst. Technol., Lausanne, 1995.

[14] P. Kovesi, "Image features from phase congruency," *Videre: J. Comput. Vis. Res.*, vol. 1, no. 3, pp. 2–27, 1999.

[15] M. C. Morrone and R. A. Owens, "Feature detection from local energy," *Pattern Recognit. Lett.*, vol. 6, pp. 303–313, 1987.

[16] Y. T. Zhou, V. Venkateshwar, and R. Chellappa, "Edge detection and linear feature extraction using a 2-D random field model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 1, pp. 84–95, Nov. 1989.

[17] S. Ando, "Image field categorization and edge/corner detection from gradient covariance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 2, pp. 179–190, Feb. 2000.

[18] P. Meer and B. Georgescu, "Edge detection with embedded confidence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 12, pp. 1351–1365, Dec. 2001.

[19] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 530–549, Jan. 2004.

[20] M. J. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 421–432, Mar. 1998.

[21] Y. Chen, C. A. Z. Barcelos, and B. A. Mair, "Smoothing and edge detection by time-varying coupled nonlinear diffusion equations," *Comput. Vis. Image Understand.*, vol. 82, no. 2, pp. 85–100, 2001.

[22] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

[23] W. Y. Ma and B. S. Manjunath, "Edgeflow: A technique for boundary detection and image segmentation," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1375–1388, Aug. 2000.

[24] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 7–27, 2001.

[25] B. S. Manjunath and R. Chellappa, "A unified approach to boundary perception: Edges, textures, and illusory contours," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 96–107, Apr. 1993.

[26] C. Grigorescu, N. Petkov, and M. A. Westenberg, "Contour detection based on nonclassical receptive field inhibition," *IEEE Trans. Image Process.*, vol. 12 , no. 7, pp. 729–739, Jul. 2003.

[27] Z. Li, "Visual segmentation by contextual influences via intra-cortical interactions in the primary visual cortex," *Network: Comput. Neural Syst.*, vol. 10, pp. 187–212, 1999.

[28] N. Petkov and P. Kruizinga, "Computational models of visual neurons specialized in the detection of periodic and aperiodic oriented visual stimuli: Bar and grating cells," *Biol. Cybern.*, vol. 76, no. 2, pp. 83–96, 1997.

[29] N. Petkov and M. A. Westenberg, "Supression of contour perception by band-limited noise and its relation to non-classical receptive field inhibition," *Biol. Cybern.*, vol. 88, pp. 236–246, 2003.

[30] R. A. Hummel, A. Rosenfeld, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 6, pp. 420–433, Jun. 1976.

[31] O. Ben-Shahar and S. W. Zucker, "The perceptual organization of texture flow: A contextual inference approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 4, pp. 401–417, Apr. 2003.

[32] L. Matalas, R. Benjamin, and R. Kitney, "An edge detection technique using the facet model and parameterized relaxation labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 328–341, Apr. 1997.

[33] M. S. Lee and G. Medioni, "Grouping…into regions, curves, and junctions," *Comput. Vis. Image Understand.*, vol. 76, no. 1, pp. 54–69, 1999.

[34] W. S. Tong and C. K. Tang, "Robust estimation of adaptive tensors of curvature by tensor voting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 434–449, Mar. 2005.

[35] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vis. Res.*, vol. 40, no. 10–12, pp. 1489–1506, 2000.

[36] M. Ursino and G. E. L. Cara, "A model of contextual interactions and contour detection in primary visual cortex," *Neural Netw.*, vol. 17, no. 5–6, pp. 719–735, 2004.

[37] T. N. Mundhenk and L. Itti, "Computational modeling and exploration of contour integration for visual saliency," *Biol. Cybern.*, vol. 93, no. 3, pp. 188–212, 2005.

[38] D. J. Field, A. Hayes, and R. F. Hess, "RF "Contour integration by the human visual system: Evidence for a local" association field," *Vis. Res.*, vol. 33, no. 2, pp. 173–193, 1993.

[39] S. C. Yen and L. H. Finkel, "Extraction of perceptually salient contours by striate cortical networks," *Vis. Res.*, vol. 38, no. 5, pp. 719–741, 1998.

[40] Z. Li, "A neural model of contour integration in the primary visual cortex," *Neur. Comput.*, vol. 4, no. 10, pp. 903–940, 1998.

[41] A. Shashua and S. Ullman, "Structural saliency: The detection of globally salient structures using a locally connected network," *Proc. ICCV*, pp. 433–435, 1988.

[42] P. Moore and C. Fitz, "Gestalt theory and instructional design," *J. Tech. Writing Commun.*, vol. 23, no. 2, pp. 137–157, 1993.

[43] K. Koffka, *Principles of Gestalt Psychology*. London, U.K.: R. & K., 1935.

[44] W. Koheler, *Gestalt Psychology*. New York: Mentor, 1947.

[45] G. Kanizsa, *Organization in Vision, Essays on Gestalt Perception*. New York: Praeger, 1979.

[46] D. Marr, "Early processing of visual information," *Phil. Trans. Roy. Soc. London B, Biol. Sci.*, vol. 275, pp. 483–524, 1976.

[47] J. H. Elder and R. M. Goldberg, "Ecological statistics for the Gestalt laws of perceptual organization of contours," *J. Vis.*, vol. 2, pp. 324–353, 2002.

[48] W. S. Geisler, J. S. Perry, B. J. Super, and D. P. Gallogly, "Edge co-occurrence in natural images predicts contour grouping performance," *Vis. Res.*, vol. 41, pp. 711–724, 2001.

[49] Z. Gigus and J. Malik, "Detecting curvilinear structure in images," CSD Tech. Rep./619, Univ. California, Berkeley 1991.

[50] X. Ren, C. Fowlkes, and J. Malik, "Scale-Invariant contour completion using conditional random fields," *Proc. ICCV*, pp. 1214–1221, 2005.

[51] A. K. Gupta, S. Chaudhury, and G. Parthasarathy, "A new approach for aggregating edge points into line segments," *Pattern Recognit.*, vol. 26, no. 7, pp. 1069–1086, 1993.

[52] D. Crevier, "A probabilistic method for extracting chains of collinear segments," *Comput. Vis. Image Understand.*, vol. 76 , no. 1, pp. 36–53, 1999.

[53] J. August, K. Siddiqi, and S. W. Zucker, "Contour fragment grouping and shared, simple occluders," *Comput. Vis. Image Understand.*, vol. 76, no. 1, pp. 146–162, 1999.

[54] A. Amir and M. Lindenbaum, "Ground from figure discrimination," *Comput. Vis. Image Understand.*, vol. 76, no. 1, pp. 7–18, 1999.

[55] M. W. Pettet, S. P. McKee, and N. M. Grzywacz, "Constraints on long range interactions mediating contour detection," *Vis. Res.*, vol. 38, pp. 865–879, 1998.

[56] D. Gil and P. Radeva, "Extending anisotropic operators to recover smooth shapes," *Comput. Vis. Image Understand.*, vol. 99, no. 1, pp. 110–125, 2005.

[57] F. y. Shih and S. Cheng, "Adaptive mathematical morphology for edge linking," *Inf. Sci.*, vol. 167, 2004.

[58] C. Grigorescu, N. Petkov, and M. A. Westenberg, "Contour and boundary detection improved by surround suppression of texture edges," *Image Vis. Comput.*, vol. 22, no. 8, pp. 609–622, 2004.
[59] G. Papari, P. Campisi, N. Petkov, and A. Neri, "A biologically motivated multiresolution approach to contour detection," *EURASIP JASP*, vol. 2007, pp. 28–28, 2007.
[60] G. Papari, P. Campisi, N. Petkov, and A. Neri, "Contour detection by multiresolution surround inhibition," in *Proc. Int. Conf. Image Processing*, 2006, pp. 749–752.
[61] G. Papari, P. Campisi, N. Petkov, and A. Neri, "Contour detection by surround inhibition in the CHF domain," presented at the EUSIPCO, 2006.
[62] Berkley Segmentation Data Set, 2002.
[63] J. Debayle and J. C. Pinoli, "Spatially adaptive morphological image filtering using intrinsic structuring elements," *Image Anal. Stereol.*, vol. 24, no. 3, pp. 145–158, 2005.
[64] R. Lerallut, E. Decenciere, and F. Meyer, "Title: Image filtering using morphological amoebas," *Image Vis. Comput.*, vol. 25, no. 4, pp. 395–404, 2007.
[65] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern Recognit.*, vol. 12, pp. 261–268, 1980.
[66] G. Papari and N. Petkov, "Algorithm that mimics human perceptual grouping of dot patterns," *Proc. BVAI*, pp. 497–506, 2005.
[67] B. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "A robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 12, pp. 1338–1359, Dec. 1997.

Giuseppe Papari was born in Rome, Italy, in 1979. He received the M.S. degree in electrical engineering in 2003 from the Universita' degli Studi di Roma Tre (University of Rome III). He is currently pursuing the Ph.D. degree at the Departement of Mathematics and Computing Science, University of Groningen, The Netherlands.

He has published several papers about biologically motivated contour detection, perceptual grouping, artistic imaging, and multiresolution analysis, one of which was invited to CODEC 2006. His main research interests are nonlinear filtering, contour detection, image segmentation, multiresolution analysis, clustering, pattern recognition, and morphological analysis.

Dr. Papari received one of the four IBM best student paper awards at the IEEE International Conference on Image Processing in 2006.

Nicolai Petkov received the Dr.sc.techn. degree in computer engineering (Informationstechnik) from the Dresden University of Technology.

He is a Professor of computer science and Head of the Research Institute of Mathematics and Computing Science, University of Groningen, The Netherlands. Prior to joining the University of Groningen in 1991, he held research positions at the University of Wuppertal, the University of Erlangen-Nürnberg, the Academy of Sciences in Berlin, and the Dresden University of Technology. In 1989, he was awarded an Alexander von Humboldt scholarship of the Federal Republic of Germany. He is the author of two monographs and (co-)author of another book, four patents, and over 100 scientific papers. He is a member of the editorial boards of several journals. His current research focuses on computer simulations and understanding of the visual system of the brain and using the obtained insights for the development of effective computer vision algorithms. He is also interested in using computers for artistic expression.