Available online at www.sciencedirect.com

SCIENCE DIRECT®

Image and Vision Computing 22 (2004) 609–622

www.elsevier.com/locate/imavis

Contour and boundary detection improved by surround suppression of texture edges

Cosmin Grigorescu, Nicolai Petkov*, Michel A. Westenberg

Institute of Mathematics and Computing Science, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands

Received 18 February 2003; received in revised form 15 December 2003; accepted 16 December 2003

Abstract

We propose a computational step, called surround suppression, to improve detection of object contours and region boundaries in natural scenes. This step is inspired by the mechanism of non-classical receptive field inhibition that is exhibited by most orientation selective neurons in the primary visual cortex and that influences the perception of groups of edges or lines. We illustrate the principle and the effect of surround suppression by adding this step to the Canny edge detector. The resulting operator responds strongly to isolated lines and edges, region boundaries, and object contours, but exhibits a weaker or no response to texture edges. Additionally, we introduce a new post-processing method that further suppresses texture edges. We use natural images with associated subjectively defined desired output contour and boundary maps to evaluate the performance of the proposed additional steps. In a contour detection task, the Canny operator augmented with the proposed suppression and post-processing step achieves better results than the traditional Canny edge detector and the SUSAN edge detector. The performance gain is highest at scales for which these latter operators strongly react to texture in the input image.

Keywords: Edge; Region boundary; Contour detection; Texture; Inhibition; Non-classical receptive field; Surround suppression; Context; Canny; SUSAN

1. Introduction

Edge detection is considered a fundamental operation in image processing and computer vision, with a large number of studies published in the last two decades. In the context of this paper, the term 'edge' stands for a local luminance change for which a gradient can be defined and which is of sufficient strength to be considered important in a given task. Examples of edge detectors are operators that incorporate linear filtering [6,13,25,35,41], local orientation analysis [17,36,59], fitting of analytical models to the image data [8,16,22,40] and local energy [12,24,30,39]. Some of these methods were biologically motivated [24,25,35,39]. Since these operators do not make any difference between various types of edges, such as texture edges vs. object contours and region boundaries, they are known as non-contextual or, simply, general edge detectors [58].

Other studies propose more elaborate edge detection techniques that take into account additional information around an edge, such as local image statistics, image topology, perceptual differences in local cues (e.g. texture, colour), edge continuity and density, etc. Examples are dual frequency band analysis [48], statistical analysis of the gradient field [2,38], anisotropic diffusion [4,9,45,57], complementary analysis of boundaries and regions [32–34], use of edge density information [10] and biologically motivated surround modulation [19,31,46,47]. These operators are not aimed at detecting all luminance changes in an image but rather at selectively enhancing only those of them that are of interest in the context of a specific computer vision task, such as the outlines of tissues in medical images, object contours in natural image scenes, boundaries between different texture regions, etc. Such methods are usually referred to as contextual edge detectors.

The human visual system differentiates in its early stages of visual information processing between isolated edges, such as object contours and region boundaries, on the one hand, and edges in a group, such as those in texture, on the other. Various psychophysical studies have shown that the perception of an oriented stimulus, e.g. a line segment, can be influenced by the presence of other such stimuli

* Corresponding author. Address: Department of Computing Science, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands. Tel.: +31-50-363-7129; fax: +31-50-363-3800.

E-mail addresses: petkov@cs.rug.nl (N. Petkov), cosmin@cs.rug.nl (C. Grigorescu), michel@cs.rug.nl (M.A. Westenberg).

0262-8856/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2003.12.004

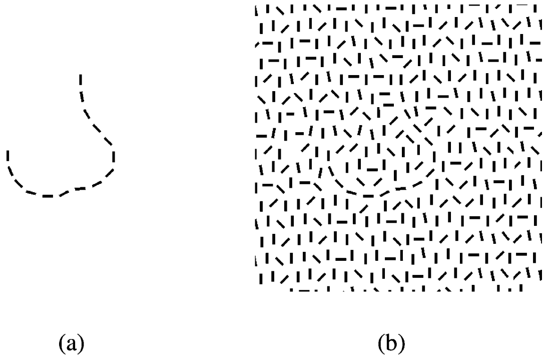


Fig. 1. (a) An isolated contour is more salient than (b) the same contour embedded in texture [11].

(distractors) in its neighbourhood. This influence can, for instance, manifest itself in the decreased saliency of a contour in presence of surrounding texture [11,27] (Fig. 1), in the so-called orientation contrast pop-out effect [43], or in the decreased visibility of letters, object icons, and bars embedded in texture [47,54]. These visual perception effects are in agreement with the results of neurophysiological measurements on neural cells in the primary visual cortex. These studies show that the response of an orientation selective visual neuron to an optimal bar stimulus in its receptive field[1] is reduced by the addition of other oriented stimuli to the surround [28,29,44]. Neurophysiologists refer to this effect as nonclassical receptive field (non-CRF) inhibition [29,44] or, equivalently, surround suppression [26]. Statistical data [26,29,44] reveals that about 80% of the orientation selective cells in the primary visual cortex show this inhibitory effect. In approximately 30% of all orientation selective cells, surround stimuli of orientation that is orthogonal to the optimal central stimulus have a weaker suppression effect than stimuli of the same orientation (anisotropic inhibitory behavior), see Fig. 2(a)–(c). In 40% of the cells, the suppression effect manifests itself irrespective of the relative orientation between the surrounding stimuli and the central stimulus (isotropic inhibitory behavior), see Fig. 2(e)–(g).

In [19,47], it was suggested that the biological utility of non-CRF inhibition is contour enhancement in natural images rich in background texture. In that study, contour detection operators were proposed that combine two biologically motivated steps: Gabor energy edge detection followed by non-CRF inhibition. In the current study, we demonstrate that the usefulness of non-CRF inhibition is not limited to biologically motivated contour detection operators only. We incorporate a non-CRF inhibition step into a typical gradient-based edge detector, the Canny operator that is widely used in image processing and computer vision, and show that this results in better enhancement of object contours and region boundaries in presence of texture. Since the terminology related to receptive fields is less appropriate in this more general computer vision context, throughout the paper, we refer to the mechanism inspired by non-CRF inhibition as *surround suppression*. Furthermore, we propose a new post-processing method based on hysteresis thresholding of the suppression slope, a measure characteristic of the context in which an edge appears (stand-alone contour vs. a texture edge).

The paper is organized as follows. Section 2 reviews gradient-based edge detection, describes two mechanisms of surround suppression, anisotropic and isotropic, and introduces the suppression slope thresholding technique. In Section 3, we use a measure defined elsewhere [19] to evaluate the performance of the proposed contour and boundary enhancement steps. The Canny edge detector augmented with these steps is compared with the traditional Canny edge detector [6] and the SUSAN operator [53]. Finally, in Section 4, we summarize the results, review similar work, and draw conclusions.

2. Surround suppression augmented operators

In the following, we propose two gradient-based contour and boundary detection operators that incorporate surround suppression. As a first step in our method, we compute a scale-dependent gradient, a technique similar to that proposed by Canny [6]. We start by reviewing scale-dependent gradient computation briefly, and then introduce two types of surround suppression and a new post-processing step.

2.1. Scale-dependent gradient computation

Gradient methods for edge detection compute the luminance gradient for each pixel of the image. When using finite differences in a very small neighborhood, the gradient is susceptible to image noise and discretization effects. In order to diminish such influences, it is customary to apply first some type of smoothing. For example, in Canny's original formulation, the input image $f(x,y)$ is first smoothed by convolving it with a two-variate Gaussian function $g_\sigma(x,y)$:

$$f_\sigma(x,y) = (f * g_\sigma)(x,y), \tag{1}$$

where

$$g_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{2}$$

The scale-dependent gradient of $f(x,y)$, defined as the gradient of the smoothed function $f_\sigma(x,y)$,

$$\nabla f_\sigma(x,y) = \left(\frac{\partial f_\sigma(x,y)}{\partial x}, \frac{\partial f_\sigma(x,y)}{\partial y}\right),$$

[1] The concept of receptive field or, more precisely, classical receptive field (CRF) used in neurophysiology corresponds to the concept of support of the impulse response used in image processing. It is the area in which an impulse stimulus affects the firing rate of the neuron. In neurophysiological practice, the CRF of an orientation selective neuron is determined by using a bar stimulus of certain optimal size and orientation.

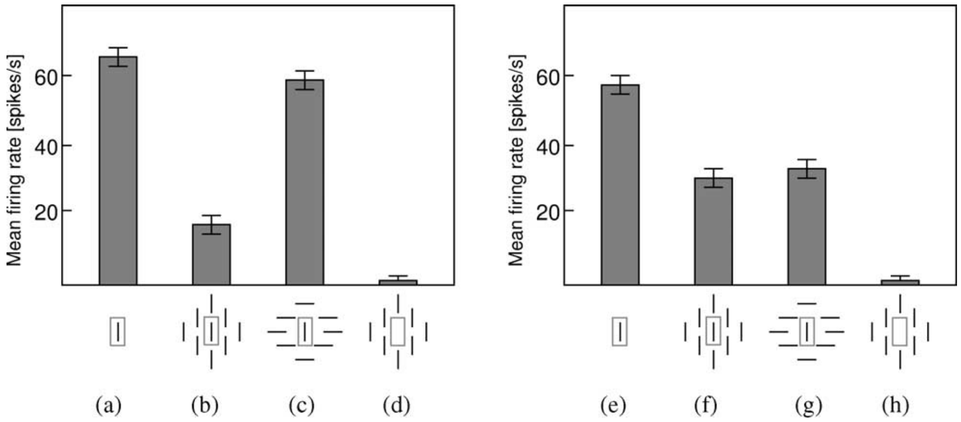


Fig. 2. Responses of two visual neurons showing **anisotropic** (left) and **isotropic** (right) inhibitory behavior, respectively (redrawn from [44], courtesy of H.C. Nothdurft, J. Gallant, D.C. van Essen, and Cambridge University Press). (a),(e) Response to a single bar of optimal size and orientation inside the CRF, delineated by a dotted rectangle. (b),(f) Decreased response is recorded when texture consisting of identical bars with the same orientation is present in the area outside the CRF. (c) For one of the cells (left) the inhibitory effect is small when the orientation of the surrounding bars is orthogonal to that of the optimal stimulus in the CRF (anisotropic surround inhibition). (g) For the other neuron (right), the inhibitory effect does not depend on the relative difference in the orientation between the surrounding bars and optimal stimulus in the CRF (isotropic surround inhibition). (d), (h) In absence of the optimal stimulus, the response of both cells is reduced to the level of spontaneous activity.

is then computed using finite differences. However, this method of differentiation has the drawback of being analytically ill-posed. The derivative of a mathematical distribution (in our case an image), can be obtained by convolving the distribution with the derivative of a smooth test function (e.g. a Gaussian) [49]. In agreement with this proposition and following [56], we choose to compute $\nabla f_\sigma(x,y)$ by evaluating the right-hand side of the following equation:

$$\nabla f_\sigma(x,y) = (f * \nabla g_\sigma)(x,y), \tag{3}$$

which has the advantage that $\nabla g_\sigma(x,y)$ is analytically well-defined and no finite difference computations are needed.

Let $\nabla_x f_\sigma(x,y)$ and $\nabla_y f_\sigma(x,y)$ be the x- and y-components of the scale-dependent gradient Eq. (3):

$$\nabla_x f_\sigma(x,y) = \left(f * \frac{\partial g_\sigma}{\partial x}\right)(x,y), \tag{4}$$

$$\nabla_y f_\sigma(x,y) = \left(f * \frac{\partial g_\sigma}{\partial y}\right)(x,y).$$

The scale-dependent gradient magnitude $M_\sigma(x,y)$ and orientation $\Theta_\sigma(x,y)$ are then given by:

$$M_\sigma(x,y) = \sqrt{(\nabla_x f_\sigma(x,y))^2 + (\nabla_y f_\sigma(x,y))^2}, \tag{5}$$

$$\Theta_\sigma(x,y) = \mathrm{atan}\left(\frac{\nabla_y f_\sigma(x,y)}{\nabla_x f_\sigma(x,y)}\right)$$

The local maxima of the gradient magnitude $M_\sigma(x,y)$ in orientation $\Theta_\sigma(x,y)$ are good indicators of possible edge locations in an image. The derivative of a Gaussian is an optimal step-edge detector in that it maximizes the signal-to-noise ratio in presence of Gaussian noise while maintaining good localization of the response, as first shown by Canny [6] and further studied by Tagare and de Figueiredo [56].

2.2. Surround suppression

Next, we extend the gradient magnitude operator defined above with a term which takes into account the context influence of the surroundings of a given point. Let $\mathrm{DoG}_\sigma(x,y)$ be the following difference of two Gaussian functions:

$$\mathrm{DoG}_\sigma(x,y) = \frac{1}{2\pi(4\sigma)^2}\exp\left(-\frac{x^2+y^2}{2(4\sigma)^2}\right) - \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{6}$$

We define a weighting function $w_\sigma(x,y)$ as follows:

$$w_\sigma(x,y) = \frac{H(\mathrm{DoG}_\sigma(x,y))}{\|H(\mathrm{DoG}_\sigma)\|_1}, \tag{7}$$

where

$$H(z) = \begin{cases} 0 & z < 0 \\ z & z \geq 0, \end{cases}$$

and $\|\cdot\|_1$ is the L_1 norm.

We implement surround suppression by computing an inhibition term for every point of an image. This term is a weighted sum of the values of the gradient in the suppression surround of the concerned point (Fig. 3). The distance between this point and a surround point is taken into account by the weighting function w_σ. In the following subsections, we introduce operators that deploy surround suppression in two different ways: anisotropic and isotropic.

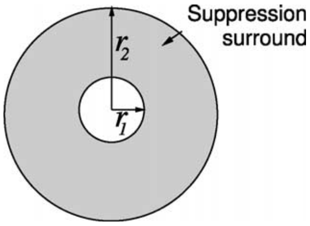


Fig. 3. The central region with radius r_1, $r_1 \approx 2\sigma$, can be considered as the support of the scale dependent gradient operator ∇f_σ. The suppression originates from an annular surround with inner radius r_1. The contribution of points at distances larger than $r_2 = 4r_1$ can be neglected, so that r_2 can be thought of as the outer radius of the suppression surround.

2.2.1. Anisotropic surround suppression

In the case of anisotropic suppression, the difference in the gradient orientations in the central point and a surround point is taken into account by an additional factor. For a point (x, y) in the image with a gradient orientation $\Theta_\sigma(x, y)$ and a point $(x - u, x - v)$ in the suppression surround with a gradient orientation $\Theta_\sigma(x - u, y - v)$, we define this factor as follows:

$$\Delta_{\Theta,\sigma}(x, y, x - u, y - v) \\ = |\cos(\Theta_\sigma(x, y) - \Theta_\sigma(x - u, y - v))|. \quad (8)$$

If the gradient orientations at points (x, y) and $(x - u, y - v)$ are identical, the weighting factor takes a maximum ($\Delta_{\Theta,\sigma} = 1$); the value of the factor decreases with the angle difference $\Theta_\sigma(x, y) - \Theta_\sigma(x - u, y - v)$, and reaches a minimum ($\Delta_{\Theta,\sigma} = 0$) when the two gradient orientations are orthogonal. In this way, edges in the surround of point (x, y) which have the same orientation as an edge at point (x, y) will have a maximal inhibitory effect. The visual cell whose response to various oriented stimuli is illustrated by the left diagram in Fig. 2 exhibits this type of behavior.

For each image point (x, y) we now define an anisotropic suppression term $t^A_\sigma(x, y)$ as the following weighted sum of the gradient magnitude values in the suppression surround of that point:

$$t^A_\sigma(x, y) = \iint_\Omega M_\sigma(x - u, y - v) w_\sigma(u, v) \\ \times |\cos(\Theta_\sigma(x, y) - \Theta_\sigma(x - u, y - v))| \mathrm{d}u\, \mathrm{d}v, \quad (9)$$

where Ω is the image coordinate domain. The two weighting factors ($w_\sigma(u, v)$ and the cosine) take into account the distance and the gradient orientation difference, respectively. This integral can be computed efficiently by convolution, as described in detail in Appendix A.

We now introduce an operator $C^A_\sigma(x, y)$ which takes as its inputs the gradient magnitude $M_\sigma(x, y)$ and the suppression term $t^A_\sigma(x, y)$:

$$C^A_\sigma(x, y) = H(M_\sigma(x, y) - \alpha t^A_\sigma(x, y)), \quad (10)$$

with a half-wave rectification function $H(z)$ defined as in Eq. (7). The factor α controls the strength of the suppression of the surround on the gradient magnitude. If there is no texture in the surroundings of a given point, the response of this operator at that point will be equal to the gradient magnitude response $M_\sigma(x, y)$. An edge passing through that point will be detected by this operator in the same way as it is detected by the gradient magnitude. However, if there are many other edges of the same orientation in the surroundings, the suppression term $t^A_\sigma(x, y)$ may become so strong that it cancels out the contribution of the gradient magnitude, resulting in a zero response. Defined in this way, the operator will respond to isolated lines and edges and to (texture) region boundaries, but it will not respond to groups of such stimuli that make part of texture of the *same orientation*, see Fig. 4(c). The response at texture boundaries is higher than the response in the interior of a texture region because the inhibition term takes smaller values at such boundaries.

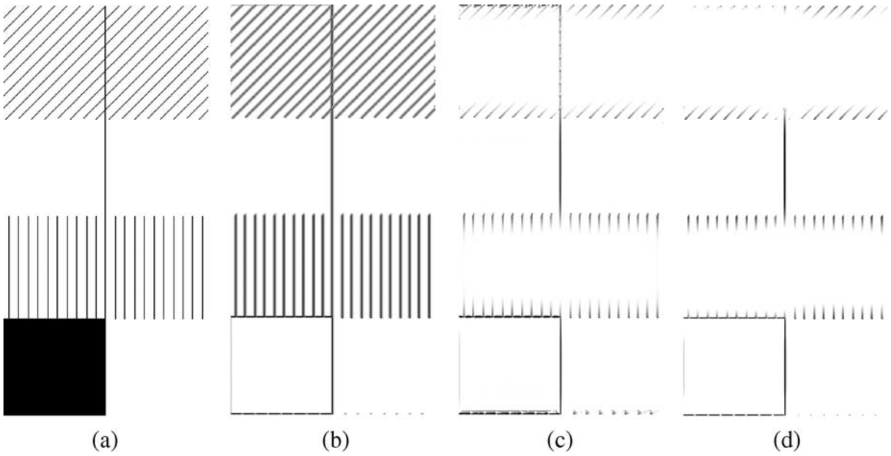


Fig. 4. (a) Synthetic input image. (b) The gradient magnitude operator detects all lines and edges independently of the context, i.e., the surroundings in which these lines and edges are embedded. (c) The gradient magnitude operator augmented with anisotropic surround suppression responds selectively to isolated lines and edges, to lines that are surrounded by a grating of a *different* orientation, and to region boundaries. Interior texture edges are suppressed. (d) The gradient magnitude operator with isotropic surround suppression responds selectively only to isolated lines and edges and also to (texture) region boundaries. Interior texture edges and lines embedded in texture of any orientation are suppressed.

2.2.2. Isotropic surround suppression

We implement isotropic surround suppression by computing a suppression term $t^I_\sigma(x,y)$ that does not depend on the orientation of surround edges; only the distance to such edges is taken into account. The suppression term $t^I_\sigma(x,y)$ is defined as a convolution of the gradient magnitude $M_\sigma(x,y)$ with the weighting function $w_\sigma(x,y)$:

$$t^I_\sigma(x,y) = \int\int_\Omega M_\sigma(x-u, y-v) w_\sigma(u,v) \mathrm{d}u\, \mathrm{d}v. \tag{11}$$

We introduce a second contour operator $C^I_\sigma(x,y)$ which takes as its inputs the gradient magnitude and the isotropic suppression term $t^I_\sigma(x,y)$:

$$C^I_\sigma(x,y) = H(M_\sigma(x,y) - \alpha t^I_\sigma(x,y)). \tag{12}$$

As before, the factor α controls the strength of suppression exercised by the surround on the gradient magnitude. This operator responds to isolated lines and edges in the same way as the operator with anisotropic suppression, but it does not respond to groups of such stimuli of any orientation that make part of the interior of a texture region, see Fig. 4(d). At the boundary of an edge texture region with another region that is not rich in edges, this operator will respond more strongly than to the texture interior. In this way, such boundaries will be enhanced in the operator output. Boundaries of two texture regions that are defined by orientation contrast will, however, not be detected by this operator.

2.3. Binary map computation

Binary contour and boundary maps can be extracted from the surround suppressed responses $C^A_\sigma(x,y)$ and $C^I_\sigma(x,y)$ by non-maxima suppression and hysteresis thresholding similar to the way this is done using the gradient [6,55]. In the following, we will use the shorthand notation $C_\sigma(x,y)$ for either $C^A_\sigma(x,y)$ or $C^I_\sigma(x,y)$. For briefness, we will use the term contour for either an object contour or a region boundary.

2.3.1. Thinning by non-maxima suppression

Non-maxima suppression thins the areas in which $C_\sigma(x,y)$ is non-zero to one-pixel wide candidate contours as follows: For each position (x,y), two responses $C_\sigma(x',y')$ and $C_\sigma(x'',y'')$ in adjacent positions (x',y') and (x'',y'') that are intersection points of a line passing through (x,y) in orientation $\Theta_\sigma(x,y)$ and a square defined by the diagonal points of an 8-neighbourhood are computed by linear interpolation, cf. Fig. 5. If the response $C_\sigma(x,y)$ at (x,y) is greater than both these values (i.e. it is a local maximum along the concerned line), it is retained, otherwise it is assigned the value zero.

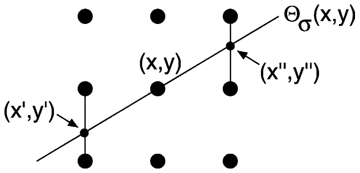


Fig. 5. Interpolated responses at positions (x',y') and (x'',y''). Non-maxima suppression retains the value in the central pixel (x,y) if it is larger than the values at (x',y') and (x'',y'').

2.3.2. Hysteresis thresholding using the contour strength

Next, a binary map is computed from the candidate contour pixels by hysteresis thresholding. This process involves two threshold values t_l and t_h, $t_l < t_h$. Commonly, the high threshold value t_h is computed as a $(1-p)$-quantile of the distribution of the response values at the candidate contour pixels, where p is the minimum fraction of candidate pixels to be retained in the contour map. Candidate contour pixels with responses higher than t_h are definitely retained in the contour map, while the ones with responses below the low threshold t_l are discarded. Candidate contour pixels with responses between t_l and t_h are retained if they can be connected to any candidate contour pixel with a response higher than t_h through a chain of other candidate contour pixels with responses larger than t_l.

2.3.3. Hysteresis thresholding using the suppression slope

An additional processing step that we present in the following further improves contour detection results. Consider the synthetic input image presented in Fig. 6(a) and two points A and B in the image. These are points in which the gradient magnitude (after the application of surround suppression) has local maxima and are, thus, potential contour points.

Note that, where positive, the surround suppressed response $C_\sigma(x,y)$ depends linearly on the suppression factor α, cf. Eqs. (10) and (12). From this linear dependence, it follows that the ratio

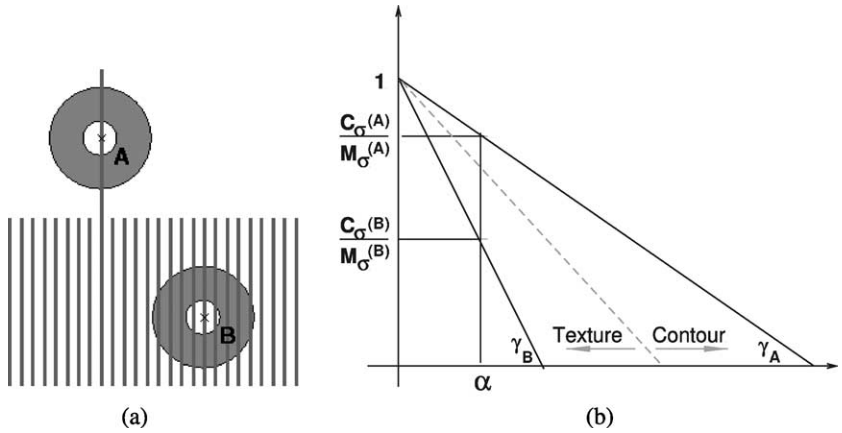
$$\frac{C_\sigma(x,y)}{M_\sigma(x,y)} = H\left(1 - \alpha\frac{t_\sigma(x,y)}{M_\sigma(x,y)}\right), \tag{13}$$

as a function of α, takes values on a line with a slope $\gamma(x,y)$, that we call the *suppression slope*, given by:

$$\gamma(x,y) = \mathrm{atan}\left(\frac{t_\sigma(x,y)}{M_\sigma(x,y)}\right). \tag{14}$$

The suppression slope $\gamma(x,y)$ depends on the amount of texture surrounding the concerned point. For instance, the slope at the contour point A is smaller than the slope at a point in a textured area, like B, see Fig. 6(b).

If the value of the suppression slope is large in a given point, this means that the surround suppression is significant at that point. Consequently, the concerned point is considered to lie in a texture region and can be eliminated from the contour map. A threshold condition can be imposed on the value of the suppression slope $\gamma(x,y)$ to discriminate between contour and texture points: points at which this slope takes values that are larger than a given threshold value can be eliminated form the contour map. A large threshold value will eliminate only a small amount of potential texture edges, while a small threshold value will eliminate such edges more substantially.


A
B
1
$\frac{C_\sigma(A)}{M_\sigma(A)}$
$\frac{C_\sigma(B)}{M_\sigma(B)}$
γ_B
Texture
Contour
γ_A
α

(a) (b)

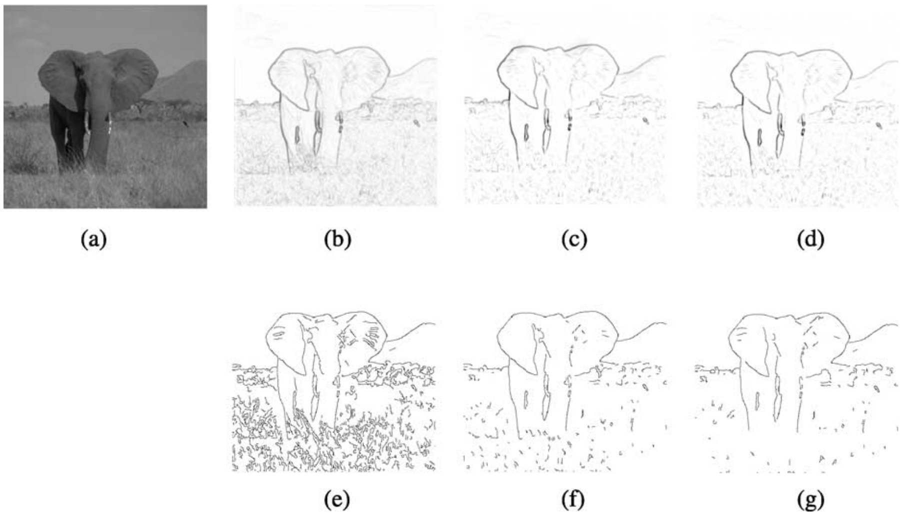
Fig. 6. (a) Points of local maxima of the surround suppressed gradient magnitude response: a point A from a contour and a point B inside a texture region. (b) At each point (x, y), the ratio $C_\sigma(x, y)/M_\sigma(x, y)$ is a linear function of α that has a slope $\gamma(x, y)$ that is determined by the gradient values in the surroundings of (x, y) and is different for a contour point and a texture point: the slope for the contour point A is smaller than the slope for the texture point B, $\gamma_A < \gamma_B$.

Although a single threshold has the advantage of simplicity, it leads in most cases to a streaking effect in the final result (discontinuous segments originating from the same contour). To reduce this effect, we apply hysteresis thresholding on the values of $\gamma(x, y)$. A low hysteresis threshold γ_l is computed as a $p^{(\gamma)}$-quantile of the distribution of suppression slope values, where $p^{(\gamma)}$ is the minimum fraction of contour pixels to be definitely retained in the final contour map. (Only pixels obtained by the first hysteresis thresholding operation are considered). A high suppression slope threshold value γ_h is selected as a multiple of γ_l. In our experiments, we choose a fixed ratio $\gamma_h = 2\gamma_l$. Points with $\gamma(x, y) < \gamma_l$ are labelled as contour points and retained in the final contour map; the points with $\gamma(x, y) > \gamma_h$ are considered texture edge points and are eliminated. Finally, those points with $\gamma_l < \gamma(x, y) < \gamma_h$ which can be connected through a chain of other similar points to a contour point are retained, otherwise eliminated.

To summarize, we perform the following post-processing steps on the surround suppressed response of the gradient magnitude $C_\sigma(x, y)$:

(i) thinning by non-maxima suppression of $C_\sigma(x, y)$;
(ii) binarization by (hysteresis) thresholding applied on the result of (i);
(iii) selection of contour pixels from the result of (ii) by (hysteresis) thresholding of the suppression slope.

This process is illustrated for a natural image in Fig. 7.



(a) (b) (c) (d)

(e) (f) (g)

Fig. 7. (a) Original input image (512 × 512 pixels). (b) Gradient magnitude $M_\sigma(x, y)$ for $\sigma = 1.6$. (c) Anisotropic and (d) isotropic surround suppressed responses for $\alpha = 1.0$. (e) Binary map obtained from (b) by non-maxima suppression and hysteresis thresholding ($p = 0.1$) as in Canny's algorithm. (f), (g) Binary maps extracted from (c) and (d), respectively, by non-maxima suppression and hysteresis thresholding ($p = 0.1$) and subsequent contour pixel selection by hysteresis thresholding of the suppression slope ($p^{(\gamma)} = 0.1$).

3. Experimental results

3.1. Subjectively specified desired output

Most state-of-the-art methods for performance evaluation of edge and contour detectors use natural images (photographs) with associated desired operator output that is subjectively specified by an observer [5]. Some recent studies [50–52] show that the performance of such an operator must be considered task dependent. For object recognition, for example, some operators may perform better than others despite similar performance on synthetic images. The proposed surround suppression mechanisms aim explicitly at better detection of object contours and region boundaries in natural scenes.

We took 40 images which depict either man-made objects on textured backgrounds or animals in their natural habitat. For each image, an associated desired output binary contour map was drawn by hand[2]. A pixel is marked as a contour pixel in the desired output if (i) it is a part of an occluding contour of an object or it belongs to a contour in the interior of an object or if (ii) it makes part of a boundary between two (textured) regions, e.g. sky and grass or water and sky. The desired output is thus defined subjectively similar to the way this is done for image segmentation in [37]. However, our procedure for defining the desired output is different in two aspects: (i) we obtain contour and boundary maps and not region maps; (ii) we use more explicit selection criteria. Fig. 8 presents three natural images from the evaluation database together with their corresponding desired output contour maps.

3.2. Performance measure

We use the performance measure introduced in [19], and first review this measure briefly. Let E_{DO} and B_{DO} be the set of contour and background pixels[3], respectively, of the desired output contour map and E_{D} and B_{D} be the set of contour and background pixels of the contour image generated by a given operator. The set of correctly detected contour pixels is defined as $E = E_{\mathrm{D}} \cap E_{\mathrm{DO}}$. False negatives, i.e. desired output contour pixels missed by the operator, comprise the set $E_{\mathrm{FN}} = B_{\mathrm{D}} \cap E_{\mathrm{DO}}$. False positives, i.e. pixels for which the detector indicates the presence of a contour while they belong to the background of the desired output, define the set $E_{\mathrm{FP}} = E_{\mathrm{D}} \cap B_{\mathrm{DO}}$.

The performance measure introduced in [19] is defined as follows:

$$P = \frac{\mathrm{card}(E)}{\mathrm{card}(E) + \mathrm{card}(E_{\mathrm{FP}}) + \mathrm{card}(E_{\mathrm{FN}})}, \tag{15}$$

where card(X) denotes the number of elements of set X.

The performance measure P is a scalar taking values in the interval [0,1]. If all desired output contour pixels are correctly detected and no background pixels are falsely detected as contour pixels, then $P = 1$. For all other cases, the performance measure takes values smaller than one, being closer to zero as more contour pixels are falsely detected and/or missed by the operator.

Since a subjectively identified contour does not always exactly coincide with a local maximum of the gradient magnitude operator (an effect known from psychophysics), we consider that a contour pixel is correctly detected by the operator if a corresponding desired output contour pixel is present in a 7×7 square neighbourhood centered at the concerned pixel. In our implementation, we take a pixel from a list of contour pixels generated by the operator and look for a matching pixel (within the mentioned neighbourhood) in a list of desired output contour pixels. If such a match is found, both pixels are removed from the corresponding lists. After the whole list of contour pixels generated by the operator is processed in this way, the pixels which remain on that list are considered as false positives. Such a pixel was marked by the operator as a contour pixel while it has no counterpart contour pixel in the desired output. The pixels that remain on the desired output list after the elimination process are the false negatives: these are the positions which the operator wrongly failed to mark as contour pixels.

3.3. Performance evaluation

We compare the performance of the surround suppression augmented operators defined above with two other operators: the traditional Canny and the SUSAN edge detector. Our choice of these detectors is motivated, for the former, by its wide acceptance, and for the latter by a recent study [52], which shows that it performs best in an object recognition task based on edge information when compared with six other operators.

The SUSAN edge detector [53] is based on nonlinear processing performed on a circular neighbourhood. Given an image pixel and a disk of a certain radius centered at that pixel, the method counts the number of pixels inside the disk that have intensity values within a certain threshold difference t from the central pixel. An edge strength is estimated by subtracting this pixel count from a fraction (usually three quarters) of the disk area. When this difference is negative, the edge strength is assumed to be 0. Edge direction is found by computing a local axis of symmetry (second order x-axis and y-axis moments) on the support of the disk. The final binary edge map is computed by thinning and binarization. For noise removal, a nonlinear smoothing operation which preserves edge location can be first applied in a given neighbourhood. We computed SUSAN edges by running the program first in the so-called smoothing mode and then applying the edge detector. In our experiments, the parameters were: d, the radius of

[2] The database of images and their desired output contour maps is available at: http://www.cs.rug.nl/~imaging.

[3] The subscript GT (ground truth) was used in [19] instead of DO (desired output).

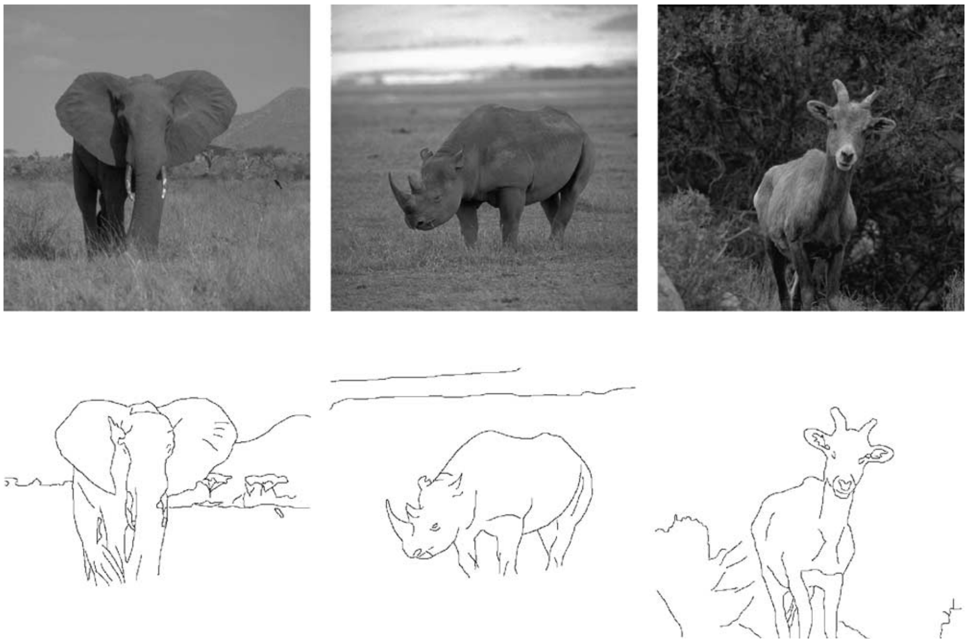


Fig. 8. Natural images (first row) and their associated desired output contour maps (second row).

the neighbourhood in which nonlinear smoothing is applied, called by the authors of SUSAN the distance threshold (in pixels), and the above mentioned threshold luminance difference t.

In our experiments, the Canny edge detector has two parameters: the standard deviation σ of the Gaussian derivative kernel used for gradient computation and p, the minimum fraction of candidate edge pixels which must be retained in the final edge map, further used to compute a high threshold value t_h. We work with a low threshold value $t_l = 0.5t_h$.

Finally, the proposed surround suppression augmented operators have the same parameters as Canny's operator, and additionally, a suppression factor α and a fraction $p^{(\gamma)}$ of the edge pixels (after thinning and gradient strength thresholding) which are definitely considered to be contour pixels. For the additional post-processing step described in Section 2.3, we fixed the value of the parameter $p^{(\gamma)}$ to $p^{(\gamma)} = 0.10$. Notice that the Canny operator can be obtained as a special case of the surround suppression augmented operators for $\alpha = 0$ and $p^{(\gamma)} = 1$.

The values of various parameters were chosen as follows. For the Canny edge detector, we used 8 scales, $\sigma_k = (\sqrt{2})^k$, $k \in \{-1...6\}$. For the surround suppression operator we used 4 scales covering the same domain sampled at even values of k, $k \in \{0, 2, 4, 6\}$ and 2 surround suppression factors, $\alpha \in \{1.0, 1.2\}$. For both methods, we applied 5 high hysteresis threshold values on the gradient, corresponding to $p \in \{0.5, 0.4, 0.3, 0.2, 0.1\}$. This results in 40 parameter combinations for each of these methods.

For the SUSAN edge detector we also chose 40 combinations of parameters, from eight values of the distance threshold d_k twice as big as the values of σ_k used in Canny's case, $d_k = 2\sigma_k$, $k \in \{-1...6\}$, and five threshold luminance difference values $t \in \{5, 10, 15, 20, 25\}$. The distance threshold values d_k lead to a comparable spatial support (in pixels) of the two types of operator.

We evaluated the performance of the operators as formulated in Eq. (15). For each image, we computed a binary contour map for each combination of parameters and calculated the corresponding performance value by comparing this contour map with the subjectively defined desired output. In this way, a set of 40 performance values was obtained for each input image and each operator.

Fig. 9 shows examples of the best contour maps (i.e. the maps with the largest value of the performance) obtained at small, medium, and large scales for all possible values of post-processing parameters. At small scales ($k \in \{-1, 0\}$) that, for these particular images, correspond to the high spatial frequencies present in the texture background, the best contour maps produced by the two suppression augmented operators consist mainly of the real object contours and region boundaries (see Fig. 9, left column). In contrast, the Canny and SUSAN operators produce output that is rich in texture edges. At medium scales ($k \in \{1, 2, 3, 4\}$), this difference is less pronounced (Fig. 9, middle column) and at large scales ($k = 5, 6$) the outputs of the suppression augmented operators and Canny operator are very similar (Fig. 9, right column). As can be seen from Fig. 9, the suppression augmented operators give comparable output across all scales, while Canny and SUSAN operators are vulnerable to texture at scales for which operator support and texture details are in a certain size agreement.

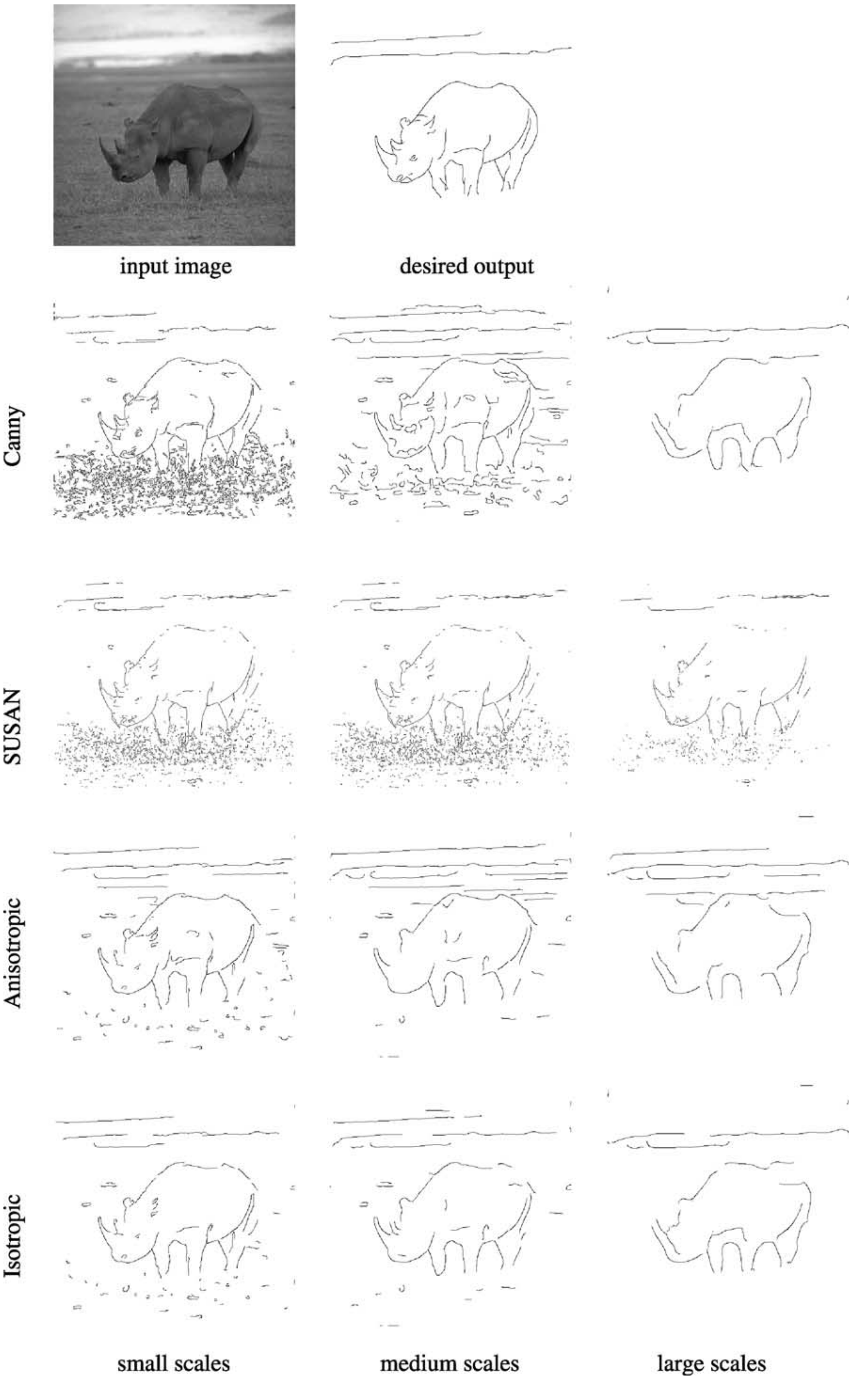


Fig. 9. A natural input image and its desired output contour map (first row). Best binary contour maps obtained for the Canny edge detector (second row), SUSAN edge detector (third row), anisotropic and isotropic surround suppression augmented operators (fourth and fifth row, respectively). The best contour map is the one that results in the best performance value over all combinations of post-processing parameters at small scales ($k \in \{-1, 0\}$, left column), medium scales ($k \in \{1, 2, 3, 4\}$, middle column) and large scales ($k \in \{5, 6\}$, right column).

This behavior is revealed also by a closer analysis of the performance values at different scales. Fig. 10 shows the performance values as statistical box-and-whisker plots computed for small scales (top part), large scales (middle part) and all scales used in our experiment (bottom part). For each method, the best performance value is represented by the top end of the corresponding whisker. Indeed, at small scales, the isotropic and anisotropic surround suppression augmented operators outperform substantially Canny and SUSAN edge detectors. The same conclusion does not hold for large scales, mainly because the support of the Gaussian function used in the Canny detector or the area of smoothing

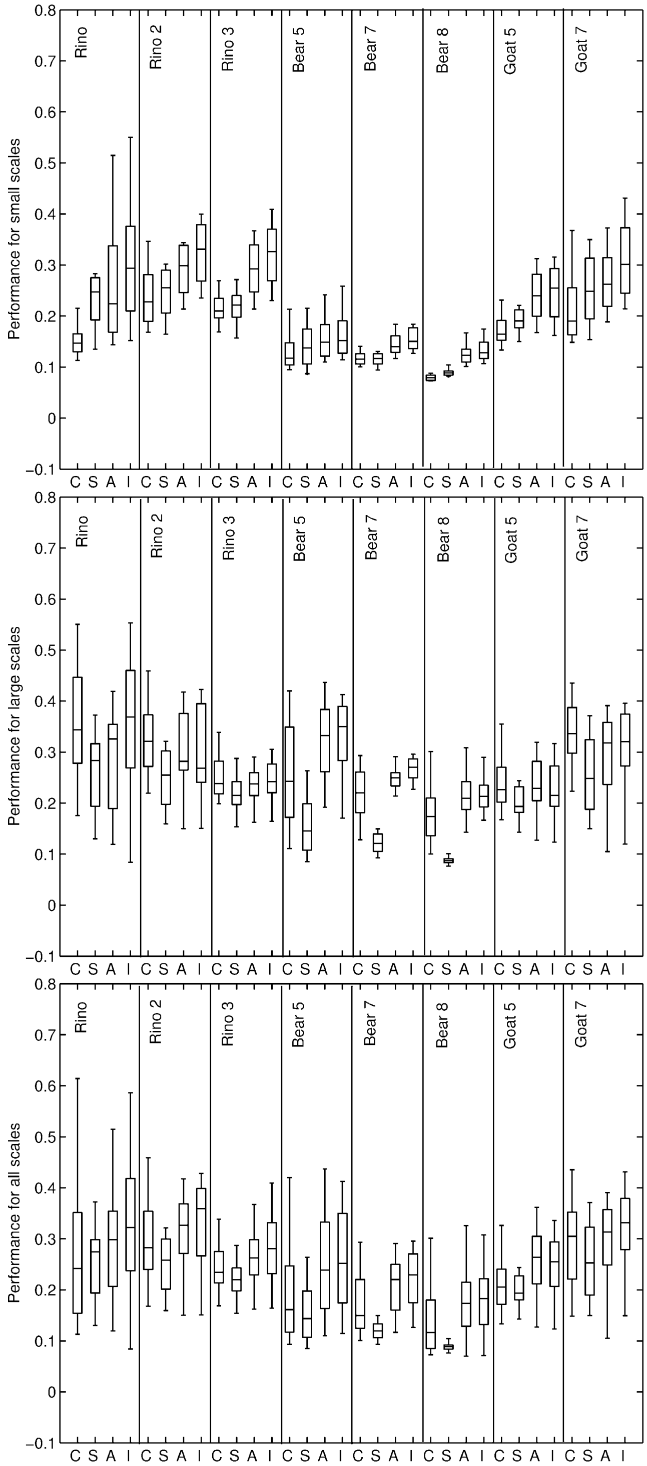


Fig. 10. Box-and-whisker plots of the performance of the Canny edge detector (denoted by C), SUSAN edge detector (denoted by S), the anisotropic (denoted by A), and isotropic (denoted by I) surround suppression augmented operators for some of the test images. Each box is a concise representation of essential features of the statistical distribution of performance values obtained for a given operator and a given input image and all possible parameter combinations. The plots display separately the values of the performance for small scales ($k \in \{-1, 0\}$, top), large scales ($k \in \{5, 6\}$, middle), and across all values of the scales ($k \in \{-1, \ldots, 6\}$, bottom).

used by the SUSAN detector are large enough to average out and thus eliminate the high frequency edges originating from texture areas.

Over all scales, however, the median values obtained for isotropic and anisotropic surround suppression are larger than the ones delivered by the Canny and SUSAN detectors. Thus, in circumstances in which no information regarding the best set of parameters is available, by choosing a random set of parameter values, there is a higher probability that the results delivered by the surround suppression operators will be better than those obtained by Canny and SUSAN.

An interesting case is the synthetic image presented in Fig. 11(a). Our perceptual interpretation of the image, two lines superimposed on a grating of parallel lines of a different orientation, is only mimicked by the anisotropic suppression operator, Fig. 11(e). The traditional Canny operator, SUSAN and the isotropic suppression operator, Fig. 11(c), (d) and (f), respectively, do not deliver results that match human perception.

4. Summary and discussion

4.1. Summary

We have shown how a biologically motivated processing step, called surround suppression, can be added to a traditional gradient-based edge detector to achieve better contour detection in natural images. The model of surround suppression we use is simple and straightforward: the response of an edge detection operator in a given point is suppressed by a weighted sum of the responses of the same operator in an annular neighborhood of that point. In this way, the proposed additional step acts as a feature contrast computation, with edges being the features involved. This step contributes to better contour detection not by means of responding more strongly to contours as compared with a traditional non-contextual edge detector but rather by means of suppressing texture edges. The result of this texture edges suppression is better contour visibility in the operator output. We considered two types of suppression, isotropic and anisotropic, and showed that they give comparable results on natural images. Certain perceptual effects related to orientation contrast can, however, be explained only by the anisotropic suppression mechanism. Furthermore, we introduced a new post-processing step, we called hysteresis thresholding of the suppression slope, aimed at further elimination of the operator response to edges which originate from textured regions.

Our experiments with a large set of natural images show that for images rich in texture background, surround suppression effectively separates contours from texture. This is important at scales for which the spatial support of the deployed edge detection operator is comparable to some characteristic size of the texture available in the input

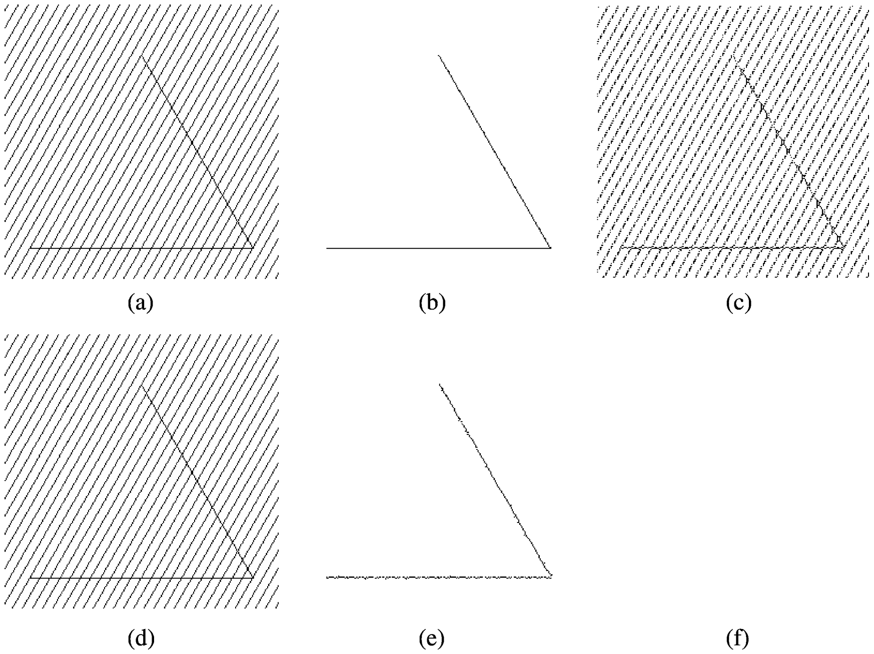


Fig. 11. (a) A synthetic input image (after [14]). (b) Associated desired contour output that agrees with common contour perception. Responses of (c) Canny and (d) SUSAN edge detectors, and the (e) anisotropic and (f) isotropic surround suppression augmented operators. Only the anisotropic suppression operator (e) mimics common contour perception (b).

image. At such scales, a non-contextual edge detector, such as the traditional Canny operator or SUSAN, generates strong responses to the texture regions. Object contours can be difficult to identify in the output of such an operator. In contrast, an edge detector that is augmented with the proposed surround suppression step does not respond strongly to texture edges while it responds to object contours. Consequently, the proposed suppression augmented operators outperform considerably non-contextual edge detectors in terms of a performance measure that favors the detection of contours only. Specifically, we showed that for a broad range of scales, the proposed surround suppression operators perform better than the Canny and SUSAN edge detectors. The performance difference is particularly large at scales for which the latter operators respond strongly to texture available in an input image.

4.2. Related work

A distinction between different types of luminance transitions, such as texture edges on the one hand vs. edges that arise from surface discontinuities and occluding boundaries on the other, was proposed as early as in 1982 [48]. The authors of that work formulated a method to select only some of the zero-crossings obtained by two difference-of-Gaussians (DoG) filters, one with a high-bandpass and the other with a low-bandpass characteristic. The method is based on the observation that texture edges induce a strong response only in the high-bandpass filter. Only luminance changes, such as a step edge, that induce strong responses in both filters are retained. This method, however, has the drawback that together with texture edges it removes the contours of small objects and lines that are narrow (compared to the support of the low-pass filter). Since the deployed DoG filters involve no orientation dependence, this method will furthermore fail to detect region boundaries defined by orientation contrast.

In [10] it is proposed to make distinction between different types of edges: dust (short isolated line segments), (isolated) curves, flow (dense edge patterns with locally parallel orientation) and turbulence (dense edge patterns with locally random orientation). Two local complexity measures, normal and tangential complexity (essentially the densities of edges in normal and tangential orientation of a given edge), are proposed and deployed for classification of edges in one of these four categories. The authors of that work succeed to explain certain perceptual effects and from their curve map illustrations one can infer that their method can be used for contour enhancement. However, since the goal of that work seems to be quite different, no quantitative analysis and algorithm evaluation for contour detection was made. Furthermore, the referred method is quite complex and it is not clear how crucial parameter selection (e.g. for tangent statistics extraction) is for success.

A very comprehensive model of intracortical interactions in area V1 of the visual cortex was proposed in [31]. Next to inhibition, it incorporates enhancement and dynamical aspects. That model takes into account most of the knowledge available in psychophysics and physiology and is able to explain a number of effects known in these areas. This was also the very purpose of that work that also includes a very good discussion of previous studies in that direction

[7,20,21]. In contrast, the current article is not focused on unveiling the biological role of intracortical interactions; we addressed this problem elsewhere [19,46,47]. In this paper we propose a simple algorithmic step that can be added to almost any edge detector in order to achieve improved *contour* detection. In the context of this study, obtaining a practical computer vision algorithm is an aspect that is considered more important than the original biological motivation. Therefore a performance comparison of the two approaches is not appropriate. Instead, we only point out some essential differences in the two methods. Since our approach has no time dimension, it is computationally less demanding: we compute the result in a single step instead of multiple steps that correspond to a sequence of time steps. Similarly, taking into account enhancement, as this is done in [31], implies considerable additional computational effort in each step that would improve contour detection results only incrementally. Finally, only anisotropic inhibition is taken into account in [31].

Other contextual edge detection techniques based on suppression have been previously proposed within the framework of anisotropic diffusion [4,42,45]. For instance, in [45] a locally adapted smoothing factor controls the amount of suppression applied to the gradient map computed at a given scale. Smoothing is more pronounced at image locations where the gradient magnitude is small, favoring the high-contrast edges over the low-contrast ones. In these approaches, however, suppression has no effect on nearby edges which have equally strong gradients. When applied to images such as the one shown in Fig. 4(a), for instance, they will not suppress the lines which are part of the gratings. Consequently, anisotropic diffusion seems more suitable for edge enhancement regardless of the underlying perceptual context (texture vs. contours). Our technique is particularly intended for texture suppression and better contour detection.

Many methods of comparing edge detection algorithms were proposed in the literature, often deploying a multitude of different evaluation criteria [5,23,50–52]. We used a single comparison method because the inclusion of additional evaluation criteria would, in a way, bring the study out of focus. The additional suppression and post-processing steps we propose are aimed at eliminating texture edges, so that object contours can pop out. Consequently, the performance measure we use is conceived to quantify the improvement in this specific respect. The proposed steps are not intended to improve (or worsen) any of the other properties of edge detectors, e.g. edge localization that is often used for comparison in the edge detection literature [5]. The localization properties of our contour detectors are, in fact, very similar to those of Canny's edge detector.

4.3. Discussion and conclusions

Normally, edge and contour detection are considered to be intermediate operations: the results they provide are used as input to further processing operations aimed at the completion of some more complex task such as object identification. It is of interest how the proposed suppression and post-processing steps would affect the ultimate result. As the proposed steps will eliminate texture edges, it is evidently not appropriate to deploy them in tasks in which texture edges are essential, e.g. texture classification or region based segmentation. In other tasks, such as shape-based object identification, the proposed suppression and post-processing steps can have a very important contribution to the quality of the final result. This is achieved through the simplification effect that these steps have on edge maps (see left column of images in Fig. 9). Clean object contour maps, free of texture edges, are of primary importance for shape-based object recognition methods that rely on contour information. Typically, in such methods some local descriptor is computed for each contour point, determined by the geometrical arrangement of other contour points in the neighborhood [1,3,15,18]. The local descriptors of the contour points of a reference object are compared with the local descriptors of a test object in order to establish point correspondences. Subsequently, a measure of similarity between the two objects is computed and a decision is taken whether they belong to the same category. Texture edges in the background of a test object change the local descriptors to such an extent that no correspondences can be found to the contour points of an identical reference object. Consequently, texture edges in the background have devastating effect on such shape recognition methods.

In conclusion, surround suppression can be incorporated as an additional processing step not only in Canny operator, but also in virtually any edge detection operator that relies on some form of enhancement of luminance transitions based on feature extraction using spatially limited support. The suppression step may be expected to improve contour detection performance in images that contain objects of interest on a cluttered or textured background.

Acknowledgements

We thank the three anonymous reviewers for their comments and suggestions.

Appendix A

In the following we present a method for the efficient computation of the anisotropic suppression term $t_\sigma^A(x,y)$ introduced in Eq. (9). For this purpose we re-write Eq. (9)

as follows:

$$\begin{aligned}
t_\sigma^A(x,y) &= \iint_\Omega M_\sigma(x-u,y-v)w_\sigma(u,v) \\
&\quad \times |\cos(\Theta_\sigma(x,y)-\Theta_\sigma(x-u,y-v))|\mathrm{d}u\,\mathrm{d}v \\
&= \Bigg| \iint_\Omega M_\sigma(x-u,y-v)w_\sigma(u,v) \\
&\quad \times \cos(\Theta_\sigma(x,y)-\Theta_\sigma(x-u,y-v))\mathrm{d}u\,\mathrm{d}v \Bigg| \\
&= \Bigg| \iint_\Omega M_\sigma(x-u,y-v)w_\sigma(u,v) \\
&\quad \times \cos\Theta_\sigma(x,y)\cos\Theta_\sigma(x-u,y-v)\mathrm{d}u\,\mathrm{d}v \\
&\quad + \iint_\Omega M_\sigma(x-u,y-v)w_\sigma(u,v) \\
&\quad \times \sin\Theta_\sigma(x,y)\sin\Theta_\sigma(x-u,y-v)\mathrm{d}u\,\mathrm{d}v \Bigg| \\
&= \Bigg| \cos\Theta_\sigma(x,y) \iint_\Omega M_\sigma(x-u,y-v) \\
&\quad \times \cos\Theta_\sigma(x-u,y-v)w_\sigma(u,v)\mathrm{d}u\,\mathrm{d}v \\
&\quad + \sin\Theta_\sigma(x,y) \iint_\Omega M_\sigma(x-u,y-v) \\
&\quad \times \sin\Theta_\sigma(x-u,y-v)w_\sigma(u,v)\mathrm{d}u\,\mathrm{d}v \Bigg| \qquad \text{(A1)}
\end{aligned}$$

Note that:

$$\begin{aligned}
\nabla_x f_\sigma(x,y) &= M_\sigma(x,y)\cos\Theta_\sigma(x,y) \qquad \text{(A2)} \\
\nabla_y f_\sigma(x,y) &= M_\sigma(x,y)\sin\Theta_\sigma(x,y)
\end{aligned}$$

Substituting Eq. (A2) in (A1), we further obtain:

$$\begin{aligned}
t_\sigma^A(x,y) &= \Bigg| \cos\Theta_\sigma(x,y) \iint_\Omega \nabla_x f_\sigma(x-u,y-v)w_\sigma(u,v)\mathrm{d}u\,\mathrm{d}v \\
&\quad + \sin\Theta_\sigma(x,y) \iint_\Omega \nabla_y f_\sigma(x-u,y-v)w_\sigma(u,v)\mathrm{d}u\,\mathrm{d}v \Bigg| \\
&= |\cos\Theta_\sigma(x,y)(\nabla_x f_\sigma * w_\sigma)(x,y) \\
&\quad + \sin\Theta_\sigma(x,y)(\nabla_y f_\sigma * w_\sigma)(x,y)\Big| \qquad \text{(A3)}
\end{aligned}$$

and the right-hand side of this relation can be evaluated efficiently using convolutions.

References

[1] Y. Amit, D. Geman, K. Wilder, Joint induction of shape features and tree classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (11) (1997) 1300–1305.

[2] S. Ando, Image field categorization and edge/corner detection from gradient covariance, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2) (2000) 179–190.

[3] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–522.

[4] M.J. Black, G. Sapiro, D. Marimont, D. Heeger, Robust anisotropic diffusion, IEEE Transaction on Image Processing 7 (3) (1998) 421–432.

[5] K.W. Bowyer, C. Kranenburg, A. Dougherty, Edge detector evaluation using empirical ROC curves, Computer Vision and Image Understanding 84 (2001) 77–103.

[6] J.F. Canny, A computational approach to edge detection, IEEE Transaction on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.

[7] G. Carpenter, S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Computer Vision and Graphical Image Processing 37 (1987) 54–115.

[8] G. Chen, Y.H.H. Yang, Edge detection by regularized cubic B-spline fitting, IEEE Transactions on Systems, Man, and Cybernetics 25 (4) (1995) 635–642.

[9] Y. Chen, C.A.Z. Barcelos, B.A. Mair, Smoothing and edge detection by time-varying coupled nonlinear diffusion equations, Computer Vision and Image Understanding 82 (2) (2001) 85–100.

[10] B. Dubuc, S.W. Zucker, Complexity, confusion and perceptual grouping. Part II: mapping complexity, International Journal on Computer Vision 42 (1/2) (2001) 83–105.

[11] D.J. Field, A. Hayes, R.F. Hess, Contour integration by the human visual system: Evidence for a local association field, Vision Research 33 (2) (1993) 173–193.

[12] T. Folsom, R. Pinter, Primitive features by steering, quadrature and scale, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (11) (1998) 1161–1173.

[13] W. Frei, C. Chen, Fast boundary detection: A generalization and a new algorithm, IEEE Transactions on Computers 26 (2) (1977) 988–998.

[14] A. Galli, A. Zama, Untersuchungen über die Wahrnehmung ebener geometrischer Figuren, die ganz oder teilweise von anderen geometrischen Figuren verdeckt sind, Zeitschrift für Psychologie 31 (1931) 308–348.

[15] D.M. Gavrila, Multi-feature hierarchical template matching using distance transforms, IEEE International Conference on Pattern Recognition (ICPR'98), Brisbane Australia (1998) 439–444.

[16] S. Ghosal, R. Mehrotra, Detection of composite edges, IEEE Transactions on Image Processing 3 (1) (1994) 14–25.

[17] P.H. Gregson, Using angular dispersion of gradient direction for detecting edge ribbons, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (7) (1993) 682–696.

[18] C. Grigorescu, N. Petkov, Distance sets for shape filters and shape recognition, IEEE Transactions on Image Processing 12 (10) (2003) 1274–1286.

[19] C. Grigorescu, N. Petkov, M.A. Westenberg, Contour detection based on nonclassical receptive field inhibition, IEEE Transactions on Image Processing 12 (7) (2003) 729–739.

[20] S. Grossberg, E. Mignolla, Neural dynamics of perceptual grouping: textures, boundaries, and emergent segmentation, Perceptional Psychophysics 38 (1985) 141–171.

[21] S. Grossberg, E. Mignolla, W. Ross, Visual brain and visual perception: how does the cortex do perceptual grouping?, Trends in Neuroscience 20 (1997) 106–111.

[22] R.M. Haralick, Digital step edges from zero-crossings of second directional derivatives, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1) (1984) 58–68.

[23] M. Heath, S. Sarkar, T. Sanocki, K.W. Bowyer, A robust visual method for assessing the relative performance of edge-detection algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (12) (1997) 1338–1359.

[24] F. Heitger. Feature detection using suppression and enhancement. Technical Report TR-163, Communication Technology Laboratory, Swiss Federal Institute of Technology, 1995.

[25] E.C. Hildreth, The detection of intensity changes by computer and biological vision systems, Computer Vision, Graphics and Image Processing 22 (1983) 1–27.

[26] H.E. Jones, K.L. Grieve, W. Wang, A.M. Sillito, Surround suppression in primate V1, J. Neurophysiology 86 (10) (2001) 2011–2028.

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

[28] M.K. Kapadia, G. Westheimer, C.D. Gilbert, Spatial distribution of contextual interactions in primary visual cortex and in visual perception, Journal of Neurophysiology 84 (4) (2000) 2048–2062.

[29] J.J. Knierim, D.C. van Essen, Neuronal responses to static texture patterns in area V1 of the alert macaque monkey, Journal of Neurophysiology 67 (1992) 961–980.

[30] P. Kovesi, Image features from phase congruency, Videre: Journal on Computer Vision Research 1 (3) (1999) 2–27.

[31] Z. Li, Visual segmentation by contextual influences via intra-cortical interactions in the primary visual cortex, Network: Computational Neural Systems 10 (1999) 187–212.

[32] W.-Y. Ma, B.S. Manjunath, Edgeflow: A technique for boundary detection and image segmentation, IEEE Transactions on Image Processing 9 (8) (2000) 1375–1388.

[33] J. Malik, S. Belongie, T. Leung, J. Shi, Contour and texture analysis for image segmentation, International Journal on Computer Vision 43 (1) (2001) 7–27.

[34] B.S. Manjunath, R. Chellappa, A unified approach to boundary perception: edges, textures, and illusory contours, IEEE Transactions on Neural Networks 4 (1) (1993) 96–107.

[35] D. Marr, E.C. Hildreth, Theory of edge detection, Proceedings of the Royal Society, London, B 207 (1980) 187–217.

[36] J.-B. Martens, Local orientation analysis in images by means of the Hermite transform, IEEE Transactions on Image Processing 6 (8) (1997) 1103–1116.

[37] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, Proceedings of International Conference on Computer Vision (2001) 416–423.

[38] P. Meer, B. Georgescu, Edge detection with embedded confidence, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (12) (2001) 1351–1365.

[39] M.C. Morrone, R.A. Owens, Feature detection from local energy, Pattern Recognition Letters 6 (1987) 303–313.

[40] V.S. Nalwa, T.O. Binford, On detecting edges, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 699–714.

[41] R. Nevatia, K. Babu, Linear feature extraction and description, Computer Graphics and Image Processing 13 (1980) 257–269.

[42] M. Nitzberg, T. Shiota, Nonlinear image filtering with edge and corner enhancement, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (8) (1992) 826–833.

[43] H.C. Nothdurft, Texture segmentation and pop-out from orientation contrast, Vision Research 31 (1991) 1073–1078.

[44] H.C. Nothdurft, J. Gallant, D.C. van Essen, Response modulation by texture surround in primate area V1: Correlates of "popout" under anesthesia, Visual Neuroscience 16 (1) (1999) 15–34.

[45] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7) (1990) 629–639.

[46] N. Petkov, P. Kruizinga, Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells, Biological Cybernetics 76 (2) (1997) 83–96.

[47] N. Petkov, M.A. Westenberg, Supression of contour perception by band-limited noise and its relation to non-classical receptive field inhibition, Biological Cybernetics 88 (2003) 236–246.

[48] W. Richards, H.K. Nishihara, B. Dawson, CARTOON: A biologically motivated edge detection algorithm. MIT A.I. Memo No. 668, see also, W. Richards (Ed.), Natural Computation, MIT Press, 1988, pp. 55–69, Chapter 4.

[49] L. Schwartz. Théorie des Distributions. Vol. I, II of Actualités scientifiques et industrielle. L'Institute de Mathématique de l'Université de Strasbourg, 1950-51.

[50] M.C. Shin, D. Goldgof, K.W. Bowyer, An objective comparison methodology of edge detection algorithms for structure from motion task, In Empirical Evaluation Techniques in Computer Vision, IEEE CS Press, 1998, pp. 235–254.

[51] M.C. Shin, D.B. Goldgof, K.W. Bowyer, Comparison of edge detectors using an object recognition task, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, Fort Collins, CO, 1999, pp. 360–365.

[52] M.C. Shin, D.B. Goldgof, K.W. Bowyer, Comparison of edge detectors using an object recognition task, Computer Vision and Image Understanding 84 (1) (2001) 160–178.

[53] S.M. Smith, J.M. Brady, SUSAN—a new approach to low level image processing, International Journal on Computer Vision 23 (1) (1997) 45–78.

[54] J.A. Solomon, D.G. Pelli, The visual filter mediating letter identification, Nature 369 (1994) 395–397.

[55] M. Sonka, V. Hlavac, R. Boyle, Image processing, analysis, and machine vision, Brooks/Cole Publishing Company, Pacific Grove, CA, 1999.

[56] H. Tagare, R.J.P. de Figueiredo, On the localization performance measure and optimal edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (12) (1990) 1186–1190.

[57] J. Weickert. A review of nonlinear diffusion filtering. In Scale-space Theory in Computer Vision, volume 1252 of Lecture Notes in Computer Science, pages 3-28, 1997.

[58] D. Ziou, S. Tabbone, Edge detection techniques–An overview, International Journal on Pattern Recognition and Image Analysis 8 (4) (1998) 537–559.

[59] O.A. Zuniga, R.M. Haralick, Integrated directional derivative gradient operator, IEEE Transactions on Systems, Man, and Cybernetics 17 (3) (1987) 508–517.