



HAL
open science

A New Perceptual Edge Detector in Color Images

Philippe Montesinos, Baptiste Magnier

► **To cite this version:**

Philippe Montesinos, Baptiste Magnier. A New Perceptual Edge Detector in Color Images. ACIVS 2010 Advanced Concepts for Intelligent Vision Systems, Dec 2010, Sydney, Australia. pp.209–220. hal-00807734

HAL Id: hal-00807734

<https://hal.science/hal-00807734>

Submitted on 4 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A New Perceptual Edge Detector in Color Images

Philippe Montesinos, Baptiste Magnier

Ecole des Mines d'ALES,
LGI2P,
Site EERIE, Parc Scientifique G.Besse
30035 Nimes Cedex 1
{Philippe.Montesinos,Baptiste.Magnier}@mines-ales.fr

Abstract. In this paper we propose a new perceptual edge detector based on anisotropic linear filtering and local maximization. The novelty of this approach resides in the mixing of ideas coming both from perceptual grouping and directional recursive linear filtering. We obtain new edge operators enabling very precise detection of edge points which are involved in large structures. This detector has been tested successfully on various image types presenting difficult problems for classical edge detection methods.

Key words: Edge detections, perceptual organization, anisotropic filters.

1 Introduction and Background

This work is motivated by applications needing strongly for a “good” edge detector, providing precise and really informative edge points with the fewest false detection rate as possible. For example, in the domain of object finding or object recognition, in natural scenes or in Internet images, objects may present various shapes and color aspects. For such cases, many local methods involving for example points of interest and color invariants fail.

In such applications, edge detection remains a central key point as it can provide geometrical information. However, commonly used edge detectors does not lead directly to object contours which must be searched among numerous edge points. Then to avoid or simplify a difficult geometrical search, many works on perceptual organisation in computer vision have been carried out [8], [12], [11], with the aim of selecting edge points involved in large “perceptual” structures. Independently of these works, new edge detection techniques using anisotropic filtering have been defined [7], [9]. These methods are able to correctly detect large linear structures.

In Practice, the generalization of perceptual organization methods for the automatic segmentation of various image types remains a difficult task in regard to the necessity of adjusting multiple thresholds and parameters. For the second class of methods (anisotropic filtering), the robustness against noise depends strongly on the filter’s smoothing parameter. If this parameter increases, the

detection is less sensitive to noise (and small structures considered as noise). Consequently, the precision of detected edge points decreases strongly at corners points and for non straight object contour parts.

We describe in this paper a new method for the precise detection of edge points belonging to large structures. Contrary to recent works involving edges, junctions and texture detection [1] [3] this work does not at this time address texture. Consequently the method described here is computationally much simpler and faster. The aim of this method is to obtain the most reliable geometrical information for object contour extraction as possible by just image differentiation. The method described, involves directional linear filtering by means of recursive filters followed by the computation of an edge operator built with a local directional maximization of the response of the filters. This method is inspired from perceptual organisation works [11], and anisotropic filtering [9].

Paper Organization :

In the section 2, we remember first the basics of perceptual organization with saliency networks, and then the basics of anisotropic edge detection with the anisotropic tensor. In the section 3, we present our method implementing a robust edge detector on gray-scale and color images having some features pointed out at section 2. Finally, at section 4, we present the results obtained with our method and we compare our results with other methods.

2 Perceptual Organization and Anisotropic Edge Detection

2.1 Perceptual Organization with Saliency Networks

Based on the works described in [12] and [11], let us consider a path crossing a pixel P of an edge segmented image. We can divide this path into three parts: the path coming from the left of P , the pixel P under consideration and the path leaving on the right (see Fig. 1). A quality function of a path can be defined as the sum of left and right lateral quality terms multiplied by a coupling factor based on the local curvature of the path at P .

$$F(P) = [F_l(P-1) + F_r(P-1)] C_{(P-1, P+1)}. \quad (1)$$

For these terms, it has been shown in [12] that some classes of quality functions involving grey levels and curvature could be optimized with locally connected saliency networks, by a local to global way. In [11] we have differently written these quality functions to establish a link with active contour function used in snakes. These functions are also composed of grey levels and curvature well separated into two different terms.

2.2 Anisotropic Edge Detection

Anisotropic edge detection can be seen as a generalization of color edge detection [6]. Suppose that the image is smoothed with a bank of rotated anisotropic

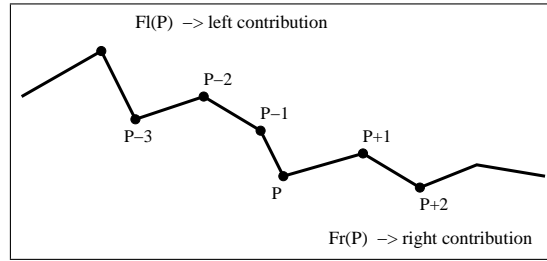


Fig. 1. A path crossing a considered pixel P.

Gaussian kernels :

$$G_{\theta}(x, y) = C e^{-\begin{pmatrix} x & y \end{pmatrix} P_{\theta}^{-1} \begin{pmatrix} \frac{1}{2\sigma_1^2} & 0 \\ 0 & \frac{1}{2\sigma_2^2} \end{pmatrix} P_{\theta} \begin{pmatrix} x \\ y \end{pmatrix}}$$

where C is a normalization coefficient, P_{θ} a rotation matrix of angle θ and σ_1 and σ_2 are the standard-deviations of the Gaussian filter.

By convolution with these rotated kernels, we obtain a collection of smoothed images $I_{\theta} = I * G_{\theta}$ which can be derived along X and Y axis to obtain anisotropic directional derivatives :

$$I_{\theta X} = \frac{\partial I_{\theta}}{\partial X} \quad \text{and} \quad I_{\theta Y} = \frac{\partial I_{\theta}}{\partial Y} .$$

These derivatives can now be combined in an orientation tensor [9]. From this tensor, an anisotropic gradient and its orientation can be computed respectively with the square root of the largest eigenvalue and its associated eigenvector. Extension to color is straightforward [6].

3 A Perceptual Edge Detector

As pointed out in section 1, the anisotropic edge detector described in section 2.2 performs well at linear portions of contours, but near corners, the gradient magnitude decreases as the edge information under the scope of the filter decreases (see Fig. 2.a). Consequently, the robustness to the noise decreases.

The simplest solution to bypass this effect is to consider paths crossing each pixel in several directions. We simply “cut” the smoothing kernel in two parts : a first part along the path coming from the left and a second part along the path leaving on the right as seen in Fig. 2.b). In the direction of the contour, “half smoothing” is performed, in the perpendicular direction a derivation filter is applied to obtain a derivative information. In order to obtain edge information, we have to combine entering and leaving paths. Now, the edge detection problem becomes an optimization problem similar to path optimization in saliency networks.

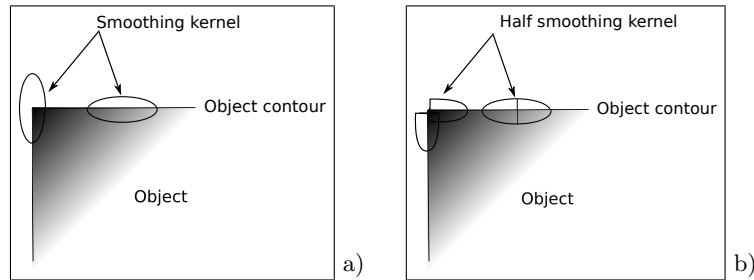


Fig. 2. Anisotropic Gaussian kernels at linear portions of contours and at corners, a) Full anisotropic Gaussian kernels. b) Half anisotropic Gaussian kernels (smoothing along the contour, derivation in the orthogonal direction of the edge).

3.1 Filters and their Implementation

Considering edge detection, it is now well established that the Gaussian and its derivatives form optimal filter in the sense of the SNR, localization and uniqueness of detection [2]. Due to their euclidean invariance and also to the emergence of scale-space theory, these filters have received a great attention in edge detection. But several other filters based on the exponential filter presenting also optimality features have been pointed out by [4], [13]. These filters do not present isotropic invariance but are interesting for their lower computational complexity in regard to Gaussians [5]. The Shen filter can be implemented recursively at order 1 and Deriche filter at order 2, as the Gaussian filter can be recursively implemented at order 4 with a good approximation.

In our case, firstly, we aren't concerned with isotropic property of filters. Secondly, the better SNR of the Gaussian filter in regard to the exponential filters is not fundamental as we are going to use a large standard-deviation in the smoothing direction (in any way this important smoothing will remove the noise). According to these considerations, we can use any of the three filters for smoothing. If we need small computational complexity, we will prefer Shen filter to the Gaussian filter. At the contrary, for derivation, we will preferably use the first derivative of the Gaussian according again to SNR considerations.

In order to obtain all the rotated filtering (see Fig. 3 b)), we just proceed in an inverse rotation of the image at discredited orientations from 0 to 360 degrees (of 1, 2, 5, or 10 degrees, depending on the precision needed and the smoothing parameter) before applying non rotated filters. We always use the "half smoothing" filter along the Y direction, and the derivation filter along the X direction (as illustrated in Fig. 3 a)). As the image is rotated instead of the filters, the filtering implementation is quite straightforward. The "half smoothing" uses slightly different implementations that those described in [13], [4] or [5].

It is well known that for implementing a recursive filter, we need to decompose the filter function into causal and anti-causal parts. In this case, we need exactly "half filtering", for normalization considerations. We then decompose

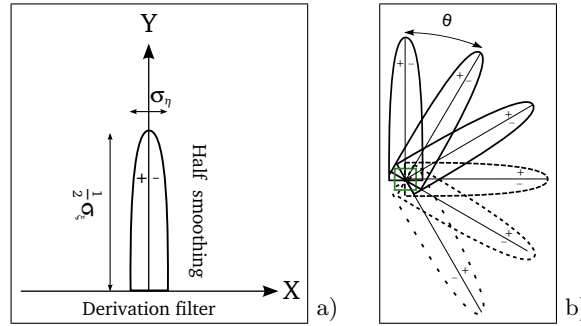


Fig. 3. a) Edge operator (*derivation filter on X and half smoothing filter on Y*). b) Rotated filters with an angle of θ .

the smoothing filter as :

$$F(x) = F^+(x) + F^-(x)$$

where :

$$F^+(x) = \begin{cases} F(x) & \text{if } x > 0 \\ \frac{1}{2}F(x) & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (2)$$

and

$$F^-(x) = \begin{cases} 0 & \text{if } x > 0 \\ \frac{1}{2}F(x) & \text{if } x = 0 \\ F(x) & \text{if } x < 0. \end{cases}$$

Only the filter of equation 2, representing the “half smoothing” will be implemented. If the chosen filter is the Gaussian, we will speak of σ_ξ for the standard-deviation of the smoothing filter and of σ_η for the standard-deviation of the derivation filter. But for a Shen smoothing filter (equation 3), we will continue speaking of σ_ξ , in this case we just define it as : $\sigma_\xi = 1/\sqrt{2\alpha}$ since the Shen filter is defined as :

$$F(x) = C e^{-\alpha|x|} . \quad (3)$$

3.2 Edge Extraction

After the filtering stage, for each pixel, we obtain a quality measure $\mathcal{Q}(x, y, \theta)$ of a path entering this pixel at the orientation θ . This measure is an integration of the slope of the image function in this direction. For obtaining an edge operator $E(x, y)$ and a gradient direction $\eta(x, y)$, we have now to combine two entering paths (inspired by [11]). For a fast practical computation of $E(x, y)$, we first compute local extrema of the function $\mathcal{Q}(x, y, \theta)$, θ_1 and θ_2 (illustrated in Fig. 4 b)). Then two of these local extrema are combined to maximize $E(x, y)$.

$$E(x, y) = \mathcal{Q}(x, y, \theta_1) - \mathcal{Q}(x, y, \theta_2)$$

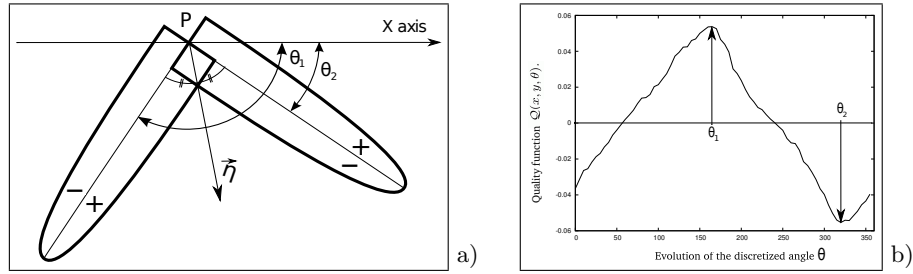


Fig. 4. a) Computation of $\eta(x, y)$ from θ_1 and θ_2 . b) Example of a function $Q(x, y, \theta)$ (the x -axis corresponds to the value of $Q(x, y, \theta)$ and the y -axis to θ).

Then we simply estimate $\eta(x, y)$ by a linear combination of θ_1 and θ_2 with the coefficients $Q(x, y, \theta_1)$ and $Q(x, y, \theta_2)$ (see Fig. 4 a)).

Once we have obtained $E(x, y)$ and $\eta(x, y)$, edges can easily be extracted by computing local maxima of $E(x, y)$ in the $\eta(x, y)$ direction followed by an hysteresis threshold.

3.3 Adaptation to Color Images

A color image is composed of three image planes : red, blue and green. Our perceptual detector above can be applied on gray-scale images, but it can also be adapted separately to the three planes of a color image as three gray-scale images. We can combine them into the following formula :

$$\begin{aligned} \|\nabla I\|_{color} &= \max(\|\nabla R\|, \|\nabla G\|, \|\nabla B\|), \\ \eta_{color} &= \arg \max_{\|\nabla R\|, \|\nabla G\|, \|\nabla B\|} (\eta_R, \eta_G, \eta_B). \end{aligned}$$

After this step, edges can easily be extracted by computing local maxima of $\|\nabla I\|_{color}$ in the $\eta_{color}(x, y)$ direction followed by an hysteresis threshold.

4 Results and Computational Time

We present results obtained both on synthetic and real images using our perceptual edge detector with derivative and half-smoothing Gaussian filters.

4.1 Synthetic Images

The first group of images (Fig. 5) is composed of two synthetic images. The first binary image (Fig. 5 a)) contains geometrical shapes without noise, dotted lines and a perceptual triangle. The image is filtered with $\sigma_\xi = 15$, $\sigma_\eta = 1$ and a discretization angle of 10 degrees. Note that all contours are detected, moreover the triangle on the down right of the image is completely visible with our perceptual edge detector and dotted lines are detected as straight lines. The second synthetic gray-scale image containing small geometrical objects, (Fig. 5

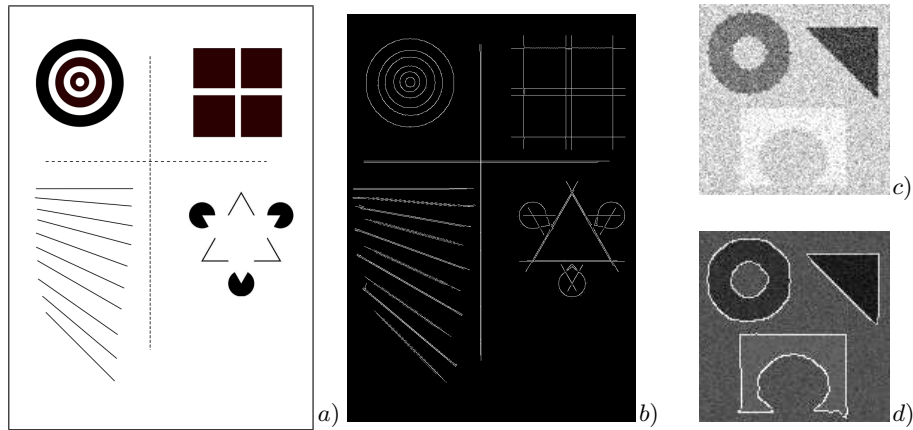


Fig. 5. a) Synthetic input image with lines, dotted lines, nearby squares and circles. b) Result of the perceptual edge operator. c) Noisy synthetic input image. d) Result of the perceptual edge operator.

c)) is small (128×128) and noisy. The amplitude of the noise is greater than the amplitude of the edges (the white figure is very difficult to segment correctly both with classical edge detection and with the method described at section 2.2). The three objects and their corners are easily extracted using our perceptual edge detector with $\sigma_\xi = 10$, $\sigma_\eta = 1$, the hysteresis lower threshold is equal to 0.01, the higher threshold is equal to 0.15 and the discretization angle is equal to 2 degrees. As stated in section 3.1, if objects to segment are small, the parameter σ_ξ should not be too “high” ($\sigma_\xi = 10$ is a good compromise).

4.2 Real Gray-scale Images

The second group of images (Fig. 6) consists of an omnidirectional image and a real textured image. The first image (Fig. 6 a)) (omnidirectional) contains a lot of warped objects. Within this experiment, we compare the anisotropic method described at section 2.2 and our perceptual edge detector. The first result (Fig. 6 b) is obtained with the anisotropic Gaussian filter with $\sigma_\xi = 10$, $\sigma_\eta = 1$, the hysteresis lower threshold is equal to 0.01, the higher threshold is equal to 0.1 and the discretization angle is equal to 10 degrees. The second result (Fig. 6 c)) is obtained with our perceptual edge detector using the same parameters (except the higher threshold which is equal to 0.5) and gives better results on many small details. Contours of small objects are more accurate, corners also are better defined, even in blurred regions of the image. Also, the second circle at the center of the image is detected almost complete with our perceptual edge operator. All these details (corners, circle in the center) are important for the calibration of this kind of sensor as well as various applications in robotics [10]. At Fig.6 d) we plot the obtained $Q(x, y, \theta)$ function of θ at the pixel indicated by the yellow cross on the Fig. 6 c). In the second image of this

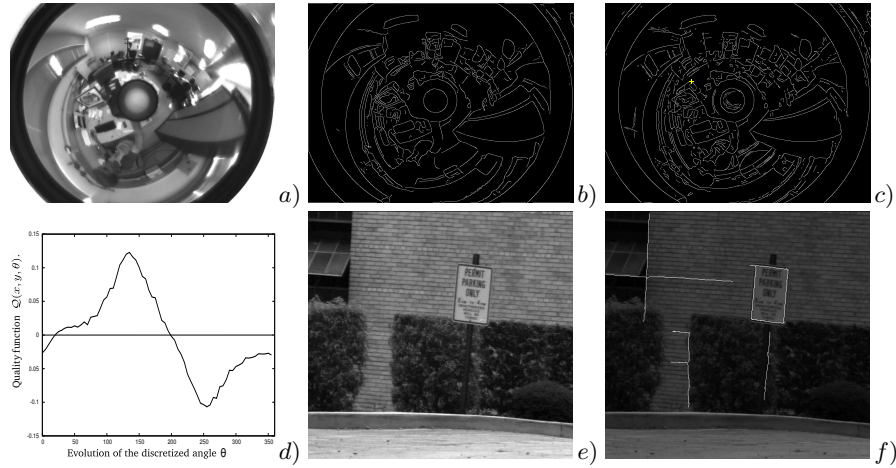


Fig. 6. a) Omnidirectional image. b) Edges obtained with the Gaussian anisotropic edge detector. c) Result of the perceptual edge operator. d) $\mathcal{Q}(x, y, \theta)$ function in one pixel (indicated by the yellow cross in the Fig. c)) of the perceptual edge operator. (the x -axis corresponds to the value of $\mathcal{Q}(x, y, \theta)$ and the y -axis to θ). e) Difficult real image. f) Edges obtained containing a vertical component.

group (Fig.6 e)), we have been interested here in the extraction of the panel and its post. The image is filtered using our perceptual edge operator, with $\sigma_\xi = 50$, $\sigma_\eta = 1$, the hysteresis lower threshold is equal to 0.01, the higher threshold is equal to 0.15 and the discretization angle is equal to 2 degrees. We then have thresholded connex components presenting vertical angles. The panel and its post are correctly extracted (Fig. 6 f)).

4.3 Real Color Images

We have tested our method on color images. The result of the first image is compared with the Gaussian anisotropic edge detector [9], the color Gaussian detector [5], the color Deriche detector [4] and the gPb [1].

The first result (Fig. 7 b)) is obtained with the anisotropic Gaussian filter with $\sigma_\xi = 10$, $\sigma_\eta = 1$, the discretization angle is equal to 5 degrees with hysteresis lower threshold equal to 0.001 and higher threshold equal to 0.15. The second result (Fig. 7 c)) is the result of our perceptual edge detector with $\sigma_\xi = 10$, $\sigma_\eta = 1$, the discretization angle is equal to 5 degrees with hysteresis lower threshold equal to 0.05 and higher threshold equal to 0.15. The Fig. 7 d) is obtained using the color Gaussian edge detector with $\sigma = 1$, hysteresis lower threshold is equal to 0.01 and higher threshold is equal to 0.1. The Fig. 7 e) is obtained using the color Deriche edge detector with $\alpha = 1$, hysteresis lower threshold is equal to 0.001 and higher threshold is equal to 0.04. As compared to the anisotropic Gaussian filter, our perceptual edge detector gives better results on many small details, this can be seen at the arm of the man in the picture, moreover, our perceptual detector

is more robust to noise on complex textures. Results obtained by Gaussian and Deriche filtering are more noisy. If we increase the threshold, many important edge points disappear. We have also compared our result with *gPb* (Fig. 7 f)) and we obtain more details.

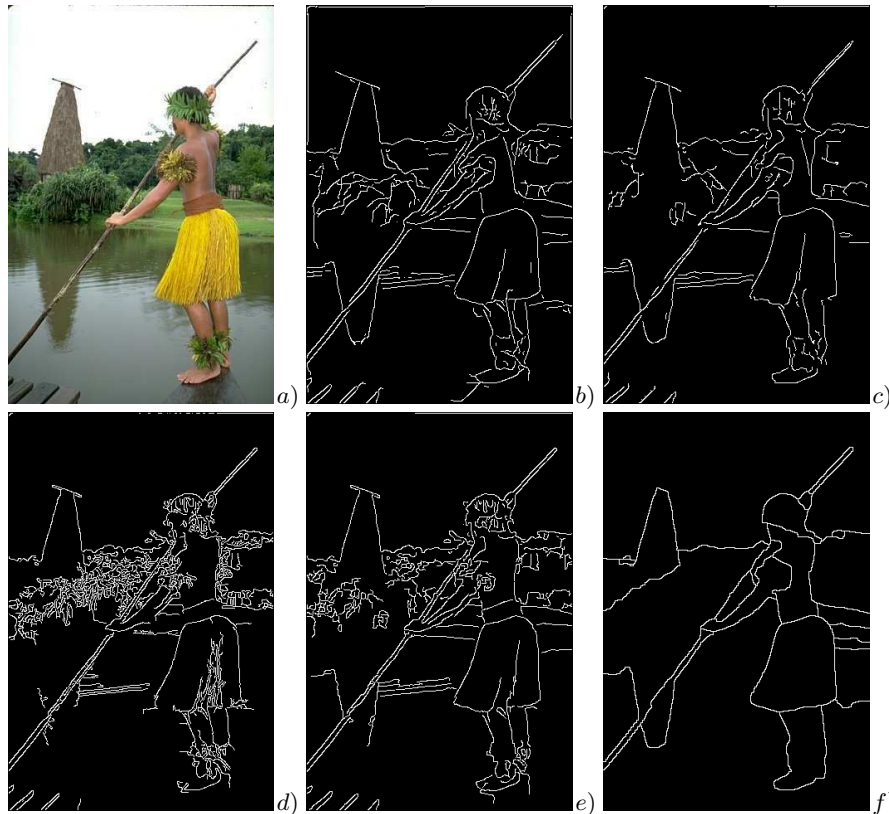


Fig. 7. a) Real color image. b) Edges obtained with the Gaussian anisotropic edge detector. c) Result of the perceptual edge operator. d) Contours obtained with the isotropic Gaussian operator. e) Results of the color Deriche operator. f) Results with the *gPb* operator.

The second image (Fig. 8 a)) contain difficult textures, the aim is to segment the car. The first result (Fig. 8 b)) is obtained with the anisotropic Gaussian filter with $\sigma_{\xi} = 10$, $\sigma_{\eta} = 1$, the discretization angle is equal to 5 degrees with hysteresis lower threshold equal to 0.001 and higher threshold equal to 0.3. If the higher threshold decreases, the noise caused by leaves increases, on the contrary, if the higher threshold becomes greater than 0.3, contours of the car disappear. The second result (Fig. 8 c)) is the result of our perceptual edge detector with $\sigma_{\xi} = 10$, $\sigma_{\eta} = 1$, the discretization angle is equal to 5 degrees with hysteresis

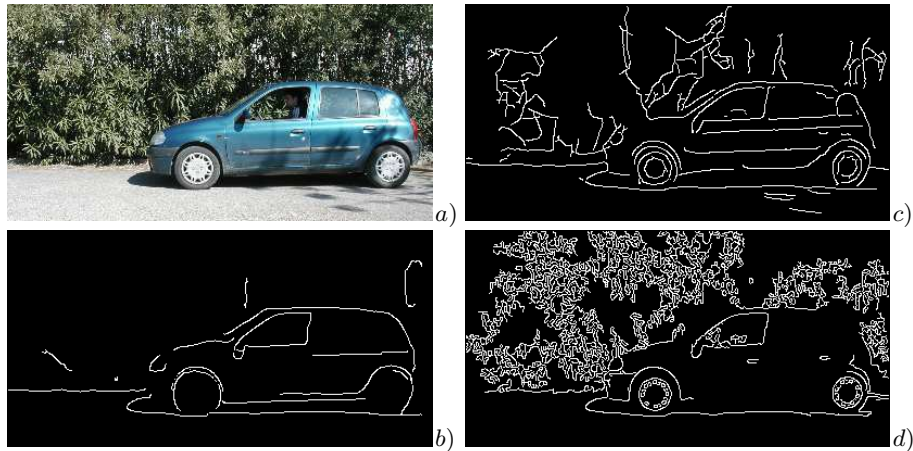


Fig. 8. a) Real color image. b) Edges obtained with the Gaussian anisotropic edge detector. c) Result of the perceptual edge operator. d) Results with the isotropic Gaussian edge operator.

lower threshold equal to 0.01 and higher threshold equal to 0.5. We are able to extract the edge of the car from “leaves” texture. The third result (Fig. 8 d)) is obtained with the Gaussian isotropic edge detector, with $\sigma = 1$, hysteresis higher threshold is equal to 0.2 and lower threshold is equal to 0.01. If the hysteresis higher threshold becomes greater than 0.2, we lost totally edges of the car and the noise remains so this result can not be exploited. We present other results with our perceptual detector in Fig. 9 and an images data base with results is available on-line [14].

4.4 Computational Time

The code of our detector is written in C++ on a Linux Operating System. For testing the detector with derivative and half-smoothing Gaussian filter, we have used a 8-core processor running at 2.6 GHz. For processing a 256x256 grey level image (with a discretization angle of 5 degrees), the whole process takes 1.3 seconds. Note that this process is divided in two stages: derivation stage and gradient stage. At the derivation stage, derivation images are computed and written to the disk. At the gradient stage, derivation images are loaded again, to optimize input and output directions, then the gradient and its angle are written to the disk. Within this experiment 72 derivation images are computed and written and loaded again. Only the derivation stage is parallelized on the 8 processors. Many time could be saved firstly by the parallelization of the gradient stage and secondly by avoiding the saving of derivation images. The memory occupation for the whole process is small and the detector is not limited by image size. For example, images of size 1000x1000 can be easily computed with 2 GB of RAM.

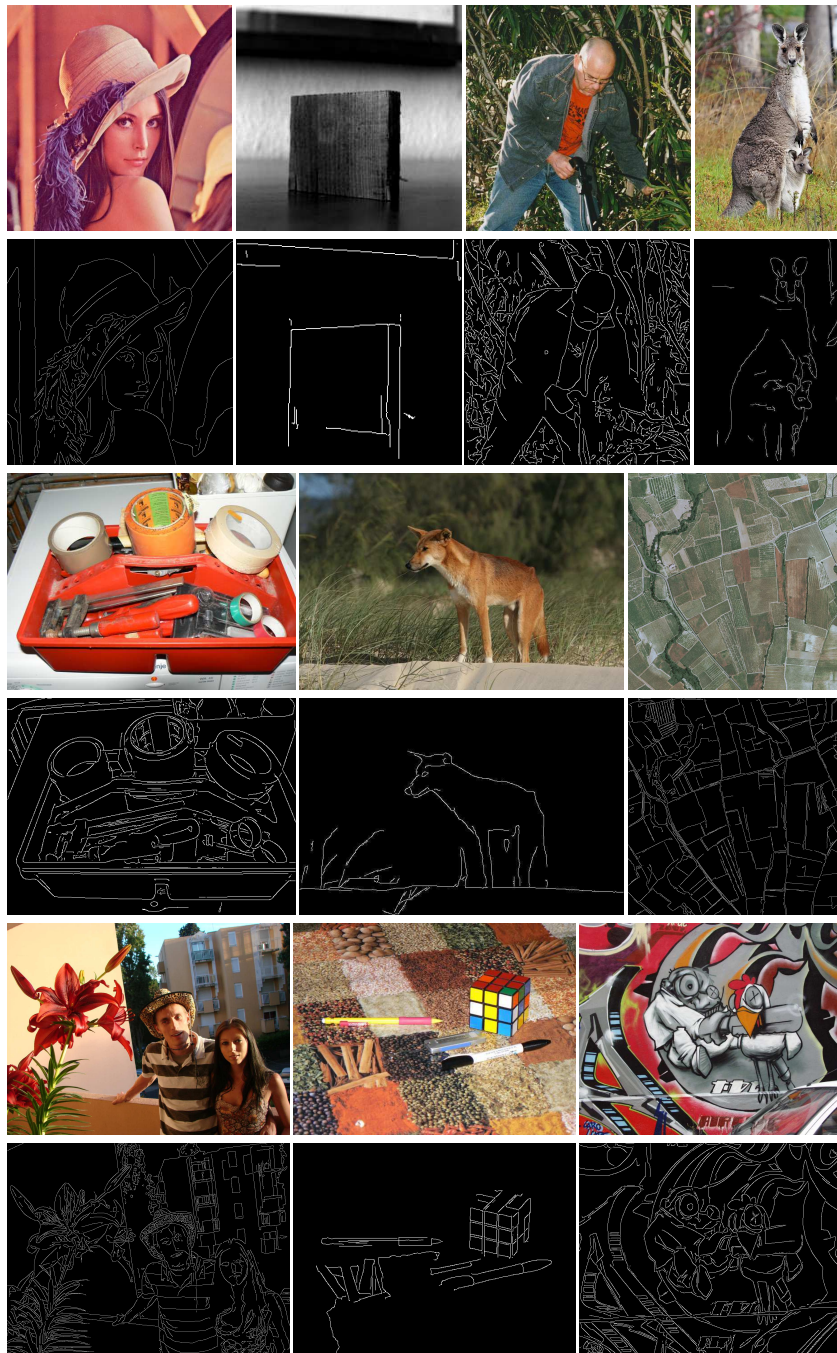


Fig. 9. Image contours selection using our perceptual edge detector with $\sigma_{\xi} = 10$, $\sigma_{\eta} = 1$ and the discretization angle is equal to 5 degrees.

5 Conclusion

We have presented a new precise, robust and fast perceptual edge detector based on anisotropic linear filtering and local maximization. Our method is designed for the detection of edge points which are involved in large structures. It enables a really precise detection for edges even in very noisy cases. This detector has been tested successfully on various image types presenting difficult problems for classical edge and detection methods. At this time, threshold parameters are almost stable, in the future we plan to bring automatic threshold.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From Contours to Regions: An Empirical Evaluation. In: IEEE Computer Vision and Pattern Recognition, pp. 2294–2301. Miami (2009)
2. Canny, J. F.: A Variational Approach to Edge Detection. In: Proceedings 3rd National Conference on Artificial Intelligence, pp. 54–58. Washington, D.C. (1983)
3. Catanzaro, B., Su, B., Sundaram, N., Lee, Y., Murphy, M., Keutzer, K.: Efficient, High-quality Image Contour Detection. In: IEEE International Conference on Computer Vision. Kyoto (2009)
4. Deriche, R.: Using Canny’s Criteria to Derive a Recursively Implemented Optimal Edge Detector. *International J. of Computer Vision*, 1(2), 167–187 (1987)
5. Deriche, R.: Recursively Implementing the Gaussian and its Derivatives. In: IEEE International Conference on Image Processing, pp. 263–267. Singapore (1992)
6. Di Zenzo, S.: A Note on the Gradient of a Multi image. *J. Computer Vision, Graphics, and Image Processing*, 33, 116–125 (1986)
7. Geusebroek, J., Smeulders, A., Van De Weijer, J.: Fast Anisotropic Gauss Filtering. In: 7th European Conference on Computer Vision, pp. 99–112. Springer, Copenhagen (2002)
8. Guy, G., Medioni, G.: Inferring Global Perceptual Contours from Local Features. In: IEEE DARPA Image Understanding Workshop, pp. 881–892. Washington, D.C. (1993)
9. Knossow, D., van de Weijer, J., Horaud, R., Ronfard R.: Articulated-body Tracking Through Anisotropic Edge Detection. In: *Dynamical Vision. LNCS*, vol. 4358, pp. 86–99. Springer (2007)
10. Magnier, B., Comby, F., Strauss, O., Triboulet, J., Demonceaux, C.: Highly Specific Pose Estimation with a Catadioptric Omnidirectional Camera. In: IEEE Int. Conference on Imaging Systems and Techniques. Thessaloniki (2010)
11. Montesinos, P., Alquier, L.: Perceptual Organization of thin Networks with Active Contour Functions Applied to Medical and Aerial Images. In: Proceedings 13th IEEE International Conference on Pattern Recognition, pp. 647–651. Vienna (1996)
12. Sha’ashua, A., Ullman, S.: Grouping Contours Elements Using a Locally Connected Network. *Neural Information Processing Systems*. Morgan Kaufmann (1990)
13. Shen, J., Castan, S.: An Optimal Linear Operator for Step Edge Detection. *Computer Vision, Graphical Models and Image Processing*, 54(2), 112–133. Elsevier (1992)
14. Magnier, B., Montesinos, P.: Perceptual Edge Detector Results. <http://www.lgi2p.ema.fr/~montesin/Demos/perceptualedgedetection.html>