



Published in Image Processing On Line on 2018-02-27.
 Submitted on 2017-11-09, accepted on 2018-02-13.
 ISSN 2105-1232 © 2018 IPOL & the authors CC-BY-NC-SA
 This article is available online with supplementary materials,
 software, datasets and online demo at
<https://doi.org/10.5201/ipol.2018.218>

Contours, Corners and T-Junctions Detection Algorithm

Antoni Buades¹, Rafael Grompone von Gioi², Julia Navarro³

¹ Universitat de les Illes Balears, Spain (toni.buades@uib.es)

² CMLA, ENS Cachan, France (grompone@gmail.com)

³ Universitat de les Illes Balears, Spain (julia.navarro@uib.es)

Communicated by Luis Gómez

Demo edited by Jose-Luis Lisani

Abstract

This article describes the implementation of the method proposed by Buades, Grompone and Navarro in 2017 for the detection of line segments, contours, corners and T-junctions. The method is inspired by the mammal visual system. The detection of corners and T-junctions plays a role as part of the process in contour detection. An a contrario validation is applied to select the most meaningful contours without the need of fixing any critical parameter.

Source Code

The reviewed source code and documentation for this algorithm are available from [the web page of this article](#)¹. Compilation and usage instruction are included in the `README.txt` file of the archive.

Keywords: contour detection; line segment detection; visual system; a contrario validation

1 Introduction

Contour, corner and T-junction detection in digital images is a fundamental problem in computer vision. These features provide important and necessary information for more complex tasks as object detection and segmentation [16], object recognition [17] and depth ordering [5, 9], among others.

We implement the method proposed by Buades, Grompone and Navarro [2]. The aim of [2] is to jointly identify contours, line segments, corners and T-junctions. It is inspired by the first stages of the mammal visual system. Hubel and Wiesel [15, 14] studied the organization and functionality of the primary visual cortex (V1) in macaques. The authors characterized the functionality of the neurons in V1, their interaction and its relationship with their spatial organization in the visual cortex. In the particular case of the hyper-column system of the V1, proximity in the system relates to spatial closeness in the view and similar dominant orientation. The basic operations made by the neurons in V1 are equivalent to directional derivatives of the image.

¹<https://doi.org/10.5201/ipol.2018.218>

The work in [2] is also motivated by a contrario validation, introduced into digital imaging by Desolneux, Moisan and Morel [6]. The authors defined those geometrical configurations that cannot be observed by chance as meaningful. Desolneux et al. [7] applied this formulation for the detection of contrasted boundaries and Cao et al. [4] for detecting good continuation of contours. In both cases, the level lines of the image are employed as potential candidates to be meaningful edges or contours. The Line Segment Detection algorithm (LSD) [11] performs state-of-the-art line contour detection using the a contrario validation. This algorithm groups close pixels sharing a similar orientation. Thus, it is valid only for detecting line segments or short edges.

The method in [2] is capable of dealing efficiently with line segment and contour detection, as well as corner and T-junction identification. Since these features are complementary, a unified procedure should identify all at once. Indeed, corners and T-junctions separate continuous curves into different edges, and edges are the main feature used for identifying corners and T-junctions.

2 Detection Algorithm

The algorithm is divided into two parts. The first part builds and processes the responses of the image to several oriented filters. The responses are organized in a 3D volume depending on the pixel coordinates and the orientation of the filter. We call this 3D volume the *cube*. Then, inhibition and filtering stages help to enhance the structures. The second part is a grouping process in the cube which identifies contours, corners and T-junctions. Figure 1 shows a block-diagram of the algorithm.

Orientation response. The first stage of the algorithm is the convolution, at each scale σ , of the image with several directional filters. We compute $d_\sigma(x, y, \theta)$: the response of the image u at any point (x, y) to the directional derivative of a Gaussian of standard deviation σ . The direction is indicated by the angle $\theta \in [0, 2\pi]$, meaning that directions with a difference of π are taken as different, even when their response at a particular pixel differs only in the sign. We retain only positive values,

$$d_\sigma(x, y, \theta) \leftarrow \max(0, d_\sigma(x, y, \theta)),$$

and the 2π range is sampled with p different angles.

In order to reduce the computational cost of the detection algorithm, only points having $d_\sigma(x, y, \theta)$ larger than a certain threshold τ are retained. Generally, we are able to distinguish a contour when its gradient is larger than 8, for images with range in $[0, 255]$. In order to be conservative, we set this threshold to a low value $\tau = 2.5$. Flat points are thus easily discarded, making the algorithm faster, thanks to the reduction in the number of points for which the further processing must be applied.

The responses at each scale are processed independently. That is, interactions only occur in a 3D volume of fixed scale σ . The data organization is similar to the hyper-column geometry of the V1 visual cortex [18]. Close points in the volume have close pixel coordinates and similar orientation. This permits to combine points with similar orientation and pixel coordinates by 3D convolution.

Lateral inhibition. The next step is to identify meaningful points to be candidates for edges. This is achieved by a lateral inhibition process. That is, we suppress the points that have lateral neighbors with larger responses. Lateral inhibition is a pervasive mechanism in the visual nervous system [18]. The aim of this step is to reduce lateral redundancy by selecting the set of points that best follow the ridge of the edge. Again, each scale σ is processed separately. A point (x, y, θ) is retained only if

$$d_\sigma(x, y, \theta) > e^{-\frac{(x-x')^2+(y-y')^2}{8\sigma^2}} \cdot d_\sigma(x', y', \theta'),$$

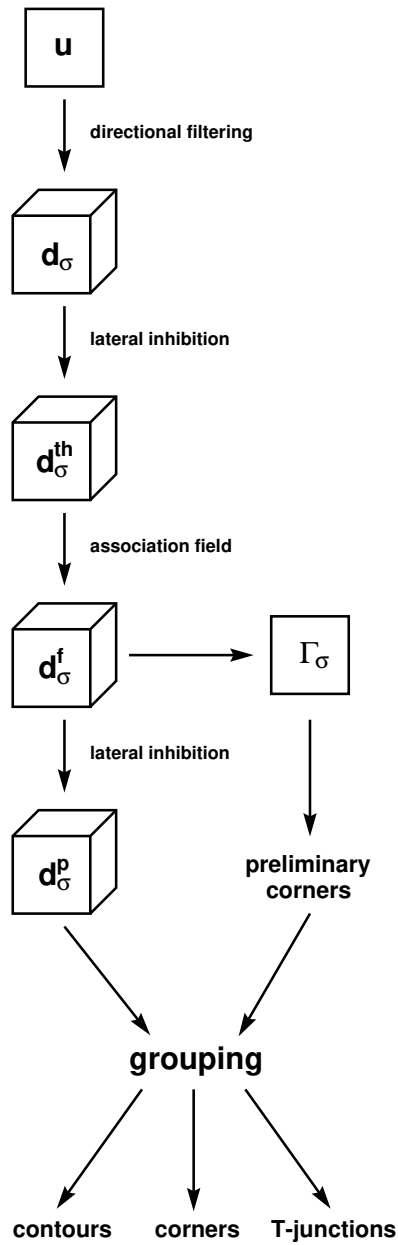


Figure 1: Block-diagram of the novel algorithm.

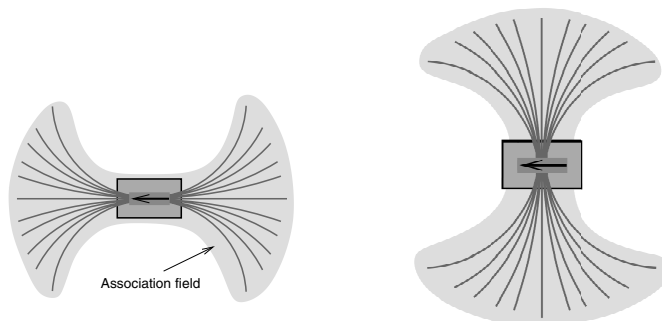


Figure 2: Left: Association field. Right: Inhibition or lateral field. For each reference point, neighbours in its association field reinforce its response while the ones in the inhibition field cancel it. Points in the association field share a similar orientation and are spatially located according to this direction.

for (x', y', θ') in the lateral neighborhood as shown in Figure 2 (right). The lateral neighborhood is defined as

$$\{(x', y', \theta') \mid |\vec{d} \cdot \theta| > |\vec{d} \cdot \theta^\perp| \text{ and } |\theta - \theta'| < \theta_0\},$$

where $\vec{d} = (x - x', y - y')$ and θ_0 is fixed empirically to 15 degrees. This lateral neighbourhood is the complementary of the association field well studied in visual perception. The neurones in the association field enforce the response of the current one, while neurones in the lateral field inhibit its response. A simplified scheme is provided in Figure 2. In practice, the association and inhibition fields form a 3D structure in the cube, since they might react to different orientations.

Retained points are set to one and non retained ones to zero. The response to the oriented filter is no longer used. This means the contrast information is not considered for detecting contours apart from the initial threshold. We will denote the output of this process by $d_\sigma^{\text{th}}(x, y, \theta)$.

Filtering/Density computation. After the inhibition process, points in the cube mutually reinforce each other by averaging their responses. We want responses to interact if they are linked by the association field displayed in Figure 2 (left). The exact solution leads to the computation of geodesics in a space with sub-Riemannian geometry [20, 19].

Two points (x_0, y_0, θ_0) and (x_1, y_1, θ_1) are linked if orientations θ_0 and θ_1 are similar, pixel coordinates (x_0, y_0) and (x_1, y_1) are close and the segment joining them has similar orientation to θ_0 . We write such an operation as a linear convolution in the cube. Each point of the cube is convolved with a 3D Gaussian oriented according to the current point orientation. Since this operation is separable, each 2D component (\cdot, \cdot, θ) is convolved with a directional Gaussian kernel of orientation θ . The resulting response is filtered with a 1D convolution in the third dimension θ .

Since $d_\sigma^{\text{th}}(x, y, \theta)$ has only zeros and ones, the output of the current filtering stage can be interpreted as an anisotropic local density around each point. We will denote the output of this process by $d_\sigma^f(x, y, \theta)$.

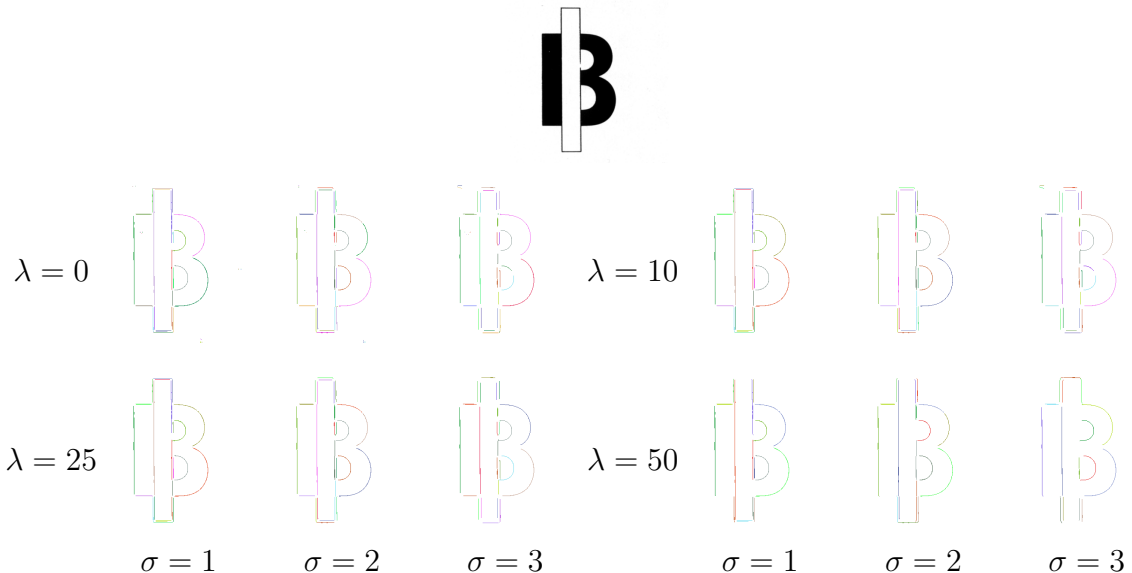


Figure 3: Comparison of detection for different length parameters λ and different scales σ .

High curvature point detection. We define a high curvature point as a pixel that responds meaningfully to a representative set of orientations. For each pixel coordinate (x, y) , we compute the indicator function,

$$\Gamma_\sigma(x, y) = |\{\theta \mid d_\sigma^f(x, y, \theta) > \gamma\}|.$$

Recall that the orientations were quantized into p different values. The compared response is obtained after the inhibition and filtering processes; after these two stages, only meaningful orientations have a non zero value, so γ is set to a small value, $\gamma = 0.01$.

The indicator function $\Gamma_\sigma(\cdot, \cdot)$ is convolved with a 2D Gaussian of standard deviation σ . The high curvature points are defined as local maxima of this convolved image and larger than a certain threshold parameter ρ . This parameter is proportional to the sampling rate of the $[0, 2\pi]$ orientation range, actually $\rho = 0.3p$. That is, a point has to respond to at least 30% of the directions to be considered as possible corner. This percentage is fixed experimentally.

The detection of corners and T-junctions is by itself one of the objectives of the algorithm. In addition, it will serve as part of the contour integration algorithm. Contours will not be allowed to pass through a corner/T-junction, thus stopping the grouping of points in the cube. The analysis of the number of edges arriving at a particular high curvature point permits to differentiate between corners and junctions.

Second lateral inhibition. The same inhibition process of the first stage is now computed on the local densities d_σ^f to obtain d_σ^p . With this second inhibition, the prominent points are detected, highlighting the skeleton of each contour.

Grouping. A $5 \times 5 \times 5$ connectivity grouping is applied to the output of the previous inhibition stage. The detected corners/T-junctions serve to stop the grouping process, thus separating a contour in two edges if it goes through a high curvature point.

Finally, only long enough contours are retained as meaningful. A threshold λ on the length must be applied. Figure 3 illustrates a detection example for different threshold values and scales. As one can see, this length-based threshold turns out to be the main parameter of the algorithm. The next section describes an a contrario formulation for automatically selecting such threshold considering the length and density of points of each curve.

Corner update and T-junctions. We retain only the points identified with high curvature when at least two detected curves of different orientation join in them. At this point, we can differentiate between corners and T-junctions based on the number of different edges arriving at a high curvature point. More than two arriving contours with different orientations indicate a junction.

Once we have updated the high curvature points, we re-apply the grouping algorithm with the updated high curvature points mask.

3 A Contrario Validation

This section describes the statistical setting used to automatically select the relevant curves, that is, to get rid of the threshold on the curve length.

We follow the a contrario framework introduced in [6] and developed further in [8]. This method is based on *the non-accidentalness principle*, according to which, an observed geometric structure is perceptually meaningful only when its expectation is low under random conditions. This principle guarantees the absence of false positives in the following weak sense: only one false detection would be made on average in a white noise image of the same size. For the control of the expected number of false detections, a geometrical structure is validated only when a test rejects the noise hypothesis. Because a big amount of structures are tested, we need to consider the problem of multiple testing, well known in statistics [13]. The a contrario setting corresponds to the formulation of Gordon et al. [10].

We use the length of the curve and the computed density at each point to compute the meaningfulness of a curve. The density $d_\sigma^f(x, y, \theta)$ implicitly encodes the local curvature. The result is compatible with the fact that one generally perceives line segments as more important than curved contours.

3.1 A Priori Model

It seems appropriate to define the a priori noise model $H(\mu)$ computing the density $d_\sigma^f(x, y, \theta)$ on a noise sample image. This a priori probability must be estimated for each scale at which contours will be evaluated. In order to do so, we first generate a noise image of large size (we used an image of width and height equal to 2048). We then apply the Inhibition/Filtering strategy introduced in Section 2. In order to make the process independent of the noise standard deviation, we set $\tau = 0$. This means we compute the density for all points in the cube.

Finally, we compute the histogram of the density of non-zero points. The values are normalized by the total number of points with non-zero density. The accumulated histogram gives a valid estimate of $H_\sigma(\mu)$.

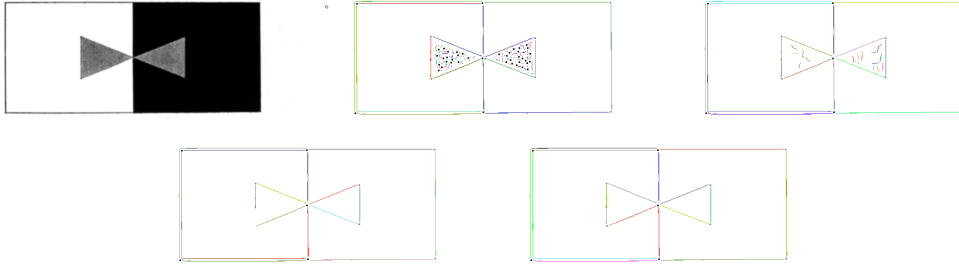


Figure 4: Comparison of detection with fixed length and a contrario method with scale parameter $\sigma = 2.0$. From top to bottom and left to right: original image, length threshold $\lambda = 0, 10, 50$ and a contrario validation. Spurious contours created by noise are detected if a length criterion is used. Long curves might be created in nearly constant areas by chance. Fixing a threshold on contrast or length does not contribute to get rid of these false detections. The non regularity of these curves, interpreted as a fast change in curvature, actually allows to discard these detections. This is naturally performed by the directional density computation, since slowly varying curvatures associate with larger density values. In such a scenario, for the same length, segments or slightly varying curves will be detected, while false alarms in noise will be discarded.

3.2 Detection

The model does not require the whole curve to have large density. Such assumption would suppress many meaningful curves, because the density decreases near its end points. Instead, we count the number of points for which the contrast is higher than a given threshold μ .

Given a curve L of length l and a density threshold μ , the number k will be evaluated as

$$k = \#\{(x, y, \theta) \in L : d_\sigma^f(x, y, \theta) \geq \mu\}.$$

The probability P of observing one such point under the noise hypothesis is $P = H_\sigma(\mu)$. Given the independence assumption, the probability of observing at least k of such points is given by the tail of a binomial distribution of parameter P

$$B(l, k, P) = \sum_{i=k}^l \binom{l}{i} P^i (1 - P)^{l-i}.$$

We define then the quantity

$$\text{NFA}(L) = N_T \cdot B(l, k, H_\sigma(\mu)),$$

where N_T is the number of tests to be performed and corresponds to the Bonferroni correction term in multiple testing [10]. The $\text{NFA}(L)$ is an upper bound to the expected number of detections under the noise model. The larger the NFA value, the more common it is to observe by chance a curve similar to L in the noise model, making it less interesting. Inversely, a curve with small NFA value does not appear frequently in noise images and would require a rare accident to be observed. When the $\text{NFA}(L) < \varepsilon$ the curve is validated. It can be shown [8, 10] that this formulation guarantees on average no more than ε detections on noise. As it is common in the a contrario literature, we will set $\varepsilon = 1$, since obtaining one or less false detections per image is a good compromise.

In this formulation, one needs to choose the value of μ . Instead of selecting a particular value, several values μ_i are tested. A curve will be declared valid if its NFA is less than ε for any of the μ_i . Formally, this is equivalent to increasing the number of tests, so N_T must be multiplied by a factor M corresponding to the number of tested μ . We do not choose arbitrary values for μ , we rather fix probability values P_i and derive the corresponding μ_i from $P_i = H_\sigma(\mu_i)$. We selected the values $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ for P_i .

To complete the formulation, we need to specify the number of tests N_T . This value depends on the scale parameter σ . The proposed inhibition stage limits the distance between detected curves, thus reducing the area of the image that can be occupied by a contour. This suggests that the number of tests can be written as $\frac{N^2}{\kappa\sigma^2}$, N being the number of pixels in the image. We chose $\kappa = 4$, as the exact theoretical number of curves is not computable. This number of tests is multiplied by M , the number of tested probabilities yielding the family of tests

$$M \cdot N_T \cdot B(l, k, H_\sigma(\mu_i)) < \varepsilon,$$

for $i = 1, \dots, M$. To conclude the process, the corners are updated and the grouping applied again, replacing the length criterion by the above-mentioned a contrario validation.

Computational complexity. We cannot estimate the complexity of the algorithm. It depends on the image being processed since only contrasted points are taken into account during the detection. The most time consuming parts are the lateral inhibition and the directional filtering, having order $O((4\sigma + 1)^2 p)$ for each point in the cube, where p is the number of different orientations. If all points in the cube were processed, the total complexity would be of the order $O((4\sigma + 1)^2 p^2 N)$, N being the number of pixels in the image. The filtering and inhibition stages at each point can be computed at the same time if the algorithm is parallelized. This is the strategy applied in the visual cortex, where all neurons activate and inhibit simultaneously. The execution time for a 600×500 natural image, as those in Figure 9, with $\sigma = 2$ is in average 20 seconds in a 3.3GHZ 6-Core Intel Xeon.

Parameters of the algorithm. Table 1 summarizes the parameters of the algorithm with their default value.

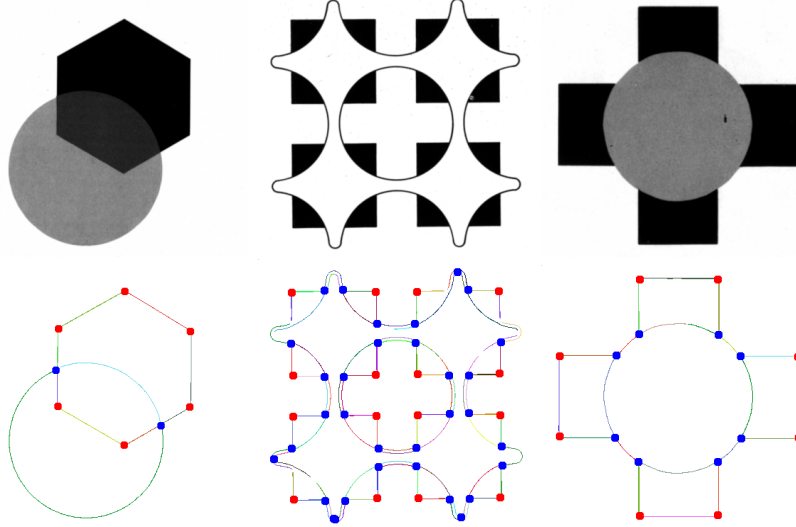
4 Examples

We illustrate the performance of the implemented algorithm with different tests.

The performance of the a contrario validation is illustrated in Figure 4. We compare this criterion with the length-based thresholding of the grouped curves. For low values of the length threshold, spurious curves are detected, whereas for larger values, short but well contrasted line segments are

parameter	definition	value
p	angle precision	7.5
σ	scale	2
τ	minimum contrast	2.5
θ_0	angle difference for filtering and inhibition	15
γ	minimum density for corner	0.01
κ	number of tests multiplier	4

Table 1: Summary of the algorithm parameters.


 Figure 5: Contour and corner detection with $\sigma = 2$. The implemented method extracts all the relevant information, keeping curves as such, while also detecting corners (red dots) and junctions (blue dots).

missed. The length-based criterion may also validate long grouped curves created by chance in noisy parts of the image. The non regularity of these curves, interpreted as a fast change in curvature, allows to discard these false detections. This feature is implicitly considered by the a contrario validation. Indeed, slowly varying curvatures are associated to large density values. In such scenario, segments or slightly varying curves of the same length will be detected by the a contrario criterion, while false alarms in noise will be discarded.

Figure 5 illustrates the contour and junction detection algorithm. The method ($\sigma = 2$) extracts all the relevant information, keeping curves as such. It also detects corners (red dots) and junctions (blue dots). We display these features since they take an active part in the curve detection algorithm. Figure 6 illustrates the detection for natural images.

In Figure 7 the performance of the implemented algorithm is tested against noise. Images have been corrupted with additive white Gaussian noise with $std = 10$ and $std = 25$ (for images with range in $[0, 255]$). The detections of contours and junctions from the original and the noisy versions of the image are compared. As it can be seen in the figure, for $std = 10$ the detections are similar and, in general, the detection from the noisy image does not produce new false positives and still finds relevant contours. For $std = 25$ strong edges are still detected, they however are usually fractioned into several segments. Thus, we conclude that the method is robust up to a certain noise amount.

A quantitative comparison with Canny [3] is illustrated in Figure 8. The empirical ROC curve proposed in [1] is plotted using the 50 images of the Objects dataset. The curve represents, at



Figure 6: Detection illustration with natural images ($\sigma = 2$).

different parameter settings, the points (% Unmatched GT edges, % FP edges). These are,

$$\% \text{Unmatched GT edges} = 1 - \frac{C_{\text{TP}}}{C_{\text{P}}}, \quad \% \text{FP edges} = \frac{C_{\text{FP}}}{C_{\text{N}}},$$


where C_{TP} stands for true positives, C_{P} for positives, C_{FP} refers to false positives and C_{N} to negatives. The different parameters are selected to vary the sensitivity of the result. In Canny, these parameters are the scale and lower and upper thresholds. In our method, the sensitivity can be adjusted by tuning the minimum contrast τ , the allowed number of false alarms ϵ and the scale σ . Under this format, the ideal point is (0, 0) and a ROC curve standing to the lower left of another curve is preferred. As seen in Figure 8, our method yields better results than Canny in terms of ROC curve.


In Figure 9, we compare the corner and junction detection algorithm with the classical Harris detector [12]. The Harris detector does not differentiate corners from junctions. For that reason, we plot all their detections in red. The implemented method differentiates T-junctions from X-junctions. For simplicity, we plot all junctions detected by these methods in blue (corners are displayed in red).


Acknowledgment

This work was partially funded by the Ministerio de Economía y Competitividad under grant TIN2014-53772-R.

Image Credits


 “Grammatica del vedere” by Gaetano Kanizsa, Società editrice il Mulino, 1980.


 by the authors.


Suprematism painting by Kazimir Malévich.

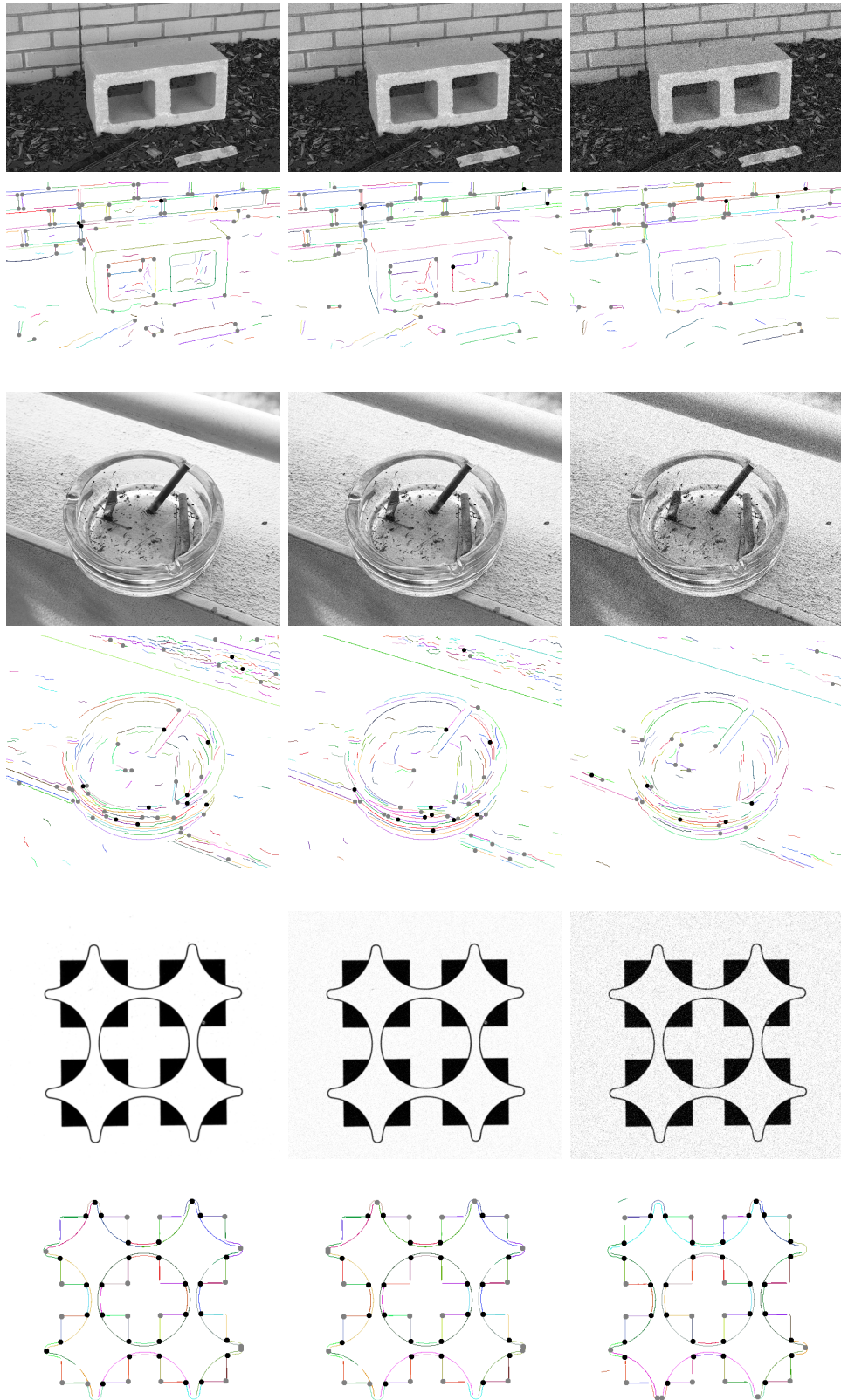


Figure 7: Robustness to additive white Gaussian noise. From left to right: original image, original image corrupted with additive white Gaussian noise with $std = 10$, and $std = 25$.



Standard *Lab* image.

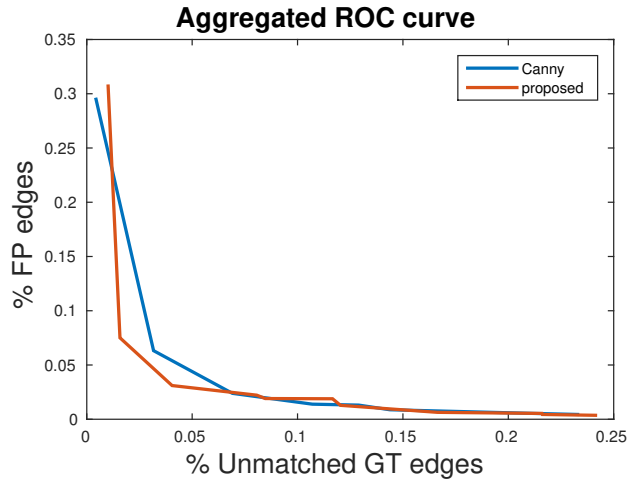


Figure 8: Quantitative comparison with Canny [3]. Aggregated ROC curve using the 50 images of the Objects dataset from [1]. A ROC curve standing to the lower left of another curve is preferred.

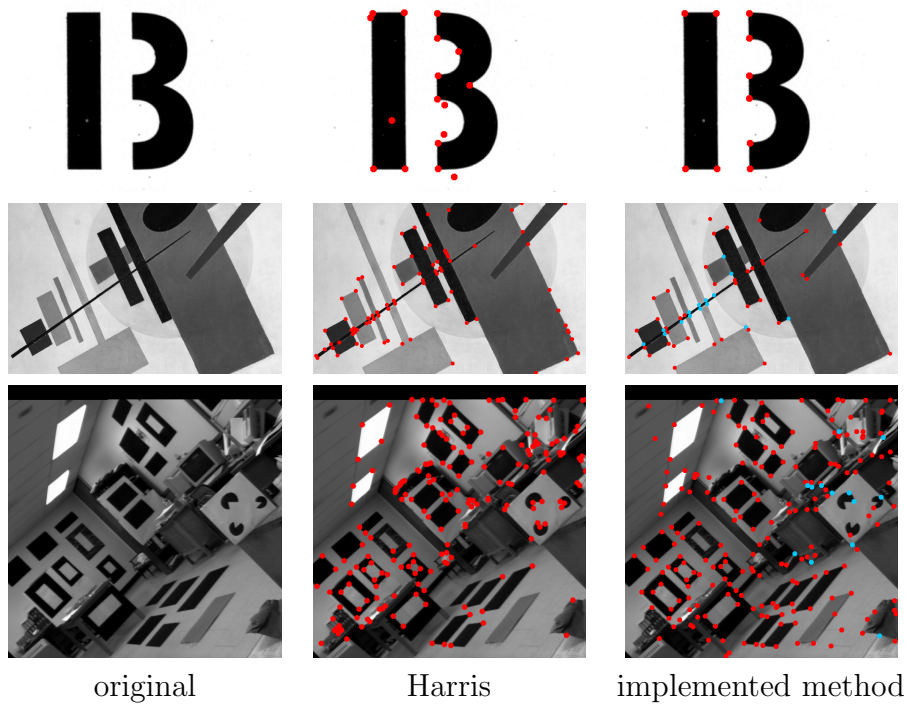
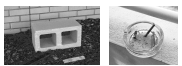


Figure 9: Corner and junction detection. Junctions are plotted in blue while corners are plotted in red. From left to right: original image, Harris detector [12] and the implemented method.



Bowyer et al.'s Objects dataset [1].

References

- [1] K. BOWYER, C. KRANENBURG, AND S. DOUGHERTY, *Edge detector evaluation using empirical ROC curves*, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 1999, pp. 354–359. <https://doi.org/10.1006/cviu.2001.0931>.
- [2] A. BUADES, R. GROMPONE VON GIOI, AND J. NAVARRO, *Joint contours, corner and T-junction detection: An approach inspired by the mammal visual system*, Journal of Mathematical Imaging and Vision, (2017), pp. 1–14. <https://doi.org/10.1007/s10851-017-0763-z>.
- [3] J. CANNY, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (1986), pp. 679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>.
- [4] F. CAO, *Application of the Gestalt principles to the detection of good continuations and corners in image level lines*, Computing and Visualization in Science, 7 (2004), pp. 3–13. <https://doi.org/10.1007/s00791-004-0123-6>.
- [5] V. CASELLES, B. COLL, AND J.-M. MOREL, *A Kanizsa programme*, in Variational Methods for Discontinuous Structures, Springer, 1996, pp. 35–55. <https://doi.org/10.1007/978-3-0348-9244-5>.
- [6] A. DESOLNEUX, L. MOISAN, AND J.-M. MOREL, *Meaningful alignments*, International Journal of Computer Vision, 40 (2000), pp. 7–23. <https://doi.org/10.1023/A:1026593302236>.
- [7] —, *Edge detection by Helmholtz principle*, Journal of Mathematical Imaging and Vision, 14 (2001), pp. 271–284. <https://doi.org/10.1023/A:1011290230196>.
- [8] —, *From Gestalt Theory to Image Analysis, a Probabilistic Approach*, Springer, 2008. <https://doi.org/10.1007/978-0-387-74378-3>.
- [9] M. DIMICCOLI, J.-M. MOREL, AND P. SALEMBIER, *Monocular depth by nonlinear diffusion*, in Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP), IEEE, 2008, pp. 95–102. <https://doi.org/10.1109/ICVGIP.2008.97>.
- [10] A. GORDON, G. GLAZKO, X. QIU, AND A. YAKOVLEV, *Control of the mean number of false discoveries, Bonferroni and stability of multiple testing*, The Annals of Applied Statistics, 1 (2007), pp. 179–190. <https://doi.org/10.1214/07-AOAS102>.
- [11] R. GROMPONE VON GIOI, J. JAKUBOWICZ, J.-M. MOREL, AND G. RANDALL, *LSD: A fast line segment detector with a false detection control*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (2010), pp. 722–732. <https://doi.org/10.1109/TPAMI.2008.300>.
- [12] C. HARRIS AND M. STEPHENS, *A combined corner and edge detector*, in Alvey vision conference, vol. 15, Citeseer, 1988, pp. 10–5244. <https://doi.org/10.5244/c.2.23>.
- [13] Y. HOCHBERG AND A. C. TAMHANE, *Multiple comparison procedures*, John Wiley & Sons, New York, 1987. <https://doi.org/10.1002/9780470316672>.
- [14] D. H. HUBEL AND T. N. WIESEL, *Receptive fields of single neurones in the cat's striate cortex*, The Journal of Physiology, 148 (1959), pp. 574–591. <https://doi.org/10.1113/jphysiol.1959.sp006308>.

- [15] —, *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*, The Journal of Physiology, 160 (1962), pp. 106–154. <https://doi.org/10.1113/jphysiol.1962.sp006837>.
- [16] H. ISHIKAWA AND D. GEIGER, *Segmentation by grouping junctions*, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 1998, pp. 125–131. <https://doi.org/10.1109/CVPR.1998.698598>.
- [17] L. LIN, S. PENG, J. PORWAY, S. C. ZHU, AND Y. WANG, *An empirical study of object category recognition: Sequential testing with generalized samples*, in IEEE International Conference on Computer Vision (ICCV), IEEE, 2007, pp. 1–8. <https://doi.org/10.1109/ICCV.2007.4408873>.
- [18] S. E. PALMER, *Vision Science: Photons to Phenomenology*, The MIT Press, 1999. [https://doi.org/10.1016/S1364-6613\(99\)01430-8](https://doi.org/10.1016/S1364-6613(99)01430-8).
- [19] G. SANGUINETTI, G. CITTI, AND A. SARTI, *Implementation of a model for perceptual completion in $R^2 \times S^1$* , in Computer Vision and Computer Graphics. Theory and Applications, Springer, 2009, pp. 188–201. https://doi.org/10.1007/978-3-642-10226-4_15.
- [20] A. SARTI, G. CITTI, AND M. MANFREDINI, *From neural oscillations to variational problems in the visual cortex*, Journal of Physiology-Paris, 97 (2003), pp. 379–385. <https://doi.org/10.1016/j.jphysparis.2003.09.014>.