Accurate Junction Detection and Characterization in Natural Images

Gui-Song Xia, Julie Delon, Yann Gousseau

HAL Id: hal-00631609
https://hal.science/hal-00631609

Submitted on 12 Oct 2011

Accurate Junction Detection and Characterization in Natural Images

Gui-Song Xia, Julie Delon, Yann Gousseau
Telecom ParisTech, LTCI CNRS,
46, rue Barrault, 75013 Paris, France
{*xia,delon,gousseau*}@enst.fr

Abstract

Accurate junction detection and characterization are of primary importance for several aspects of scene analysis, including depth recovery and motion analysis. In this work, we introduce a generic junction analysis scheme. The first asset of the proposed procedure is an automatic criterion for the detection of junctions, permitting to deal with textured parts in which no detection is expected. Second, the method yields a characterization of L-, Y- and X- junctions, including a precise computation of their type, localization and scale. Contrary to classical approaches, scale characterization does not rely on the linear scale-space, therefore enabling geometric accuracy. First, an *a contrario* approach is used to compute the meaningfulness of a junction. This approach relies on a statistical modeling of suitably normalized gray level gradients. Then, exclusion principles between junctions permit their precise characterization. We give implementation details for this procedure and evaluate its efficiency through various experiments.

Keywords: Junction detection, scale characterization, *a-contrario* method, contrast invariance.

1   Introduction

Junctions are of primary importance for visual perception and scene understanding. They are parts of the well known *primal sketch*, the schematic representation of images introduced by D. Marr [1]. Recent approaches to the computational computation of this sketch, as proposed by [2], show the key role played by junctions. Depending on the number of edges they connect, junctions are often classified into L-, Y- (or T-) and X-junctions. In particular, the role of T-junctions as cues for the perception of occlusions has been thoroughly studied by G. Kanisza [3]. Later, it has been shown in [4] that junctions are essential *local* cues to initiate contour completion and that their specific configuration (e.g. T- or Y-junction) has to be taken into account in this process. The distinct roles of L- and T-junctions for the perception of motion, in particular through the aperture phenomenon, has been known for long [5]. Junction types and positions are also shown to have a strong impact on the perception of brightness and transparency, as investigated in [6, 7].

Junctions are therefore naturally used as important cues for various computer vision tasks. Since they reveal important occlusion relationships between objects, they are involved in figure/ground separation [8, 9, 10, 11]. Junctions are also used for grouping edges and regions to achieve image segmentations [12, 13, 14]. The role of junction for object recognition has been studied since the early works on automatic scene analysis, see [15, 16, 17].

In the present work, we propose a principled and automatic method for the detection, localization and characterization of junctions. We define junctions as local or semi-local structures at the center of which at least two dominant and different edge directions meet. Junctions are defined locally, and not for instance as a by-product of a global image segmentation, mostly to achieve precise description of the junction. Following observations from psychophysics [18], we nevertheless consider large regions for the assessment of a junction. This is made possible by an automatic scale selection rule. It should be noticed that not all perceived junctions enter this framework, and that some of them are only

1

detected while performing scene analysis [18]. Nevertheless, as advocated above, we believe that a precise and accurate description of junction makes them valuable information in view of a more global scene analysis.

More precisely, we take interest in a junction detection method meeting several requirements. First, junctions have to be clearly related to the geometry of the image. In particular, the description of the junction branches should be accurate. This is in strong contrast with the classical corner detectors such as Harris, that are commonly used a keypoint detectors. Second, an automatic validation rule will be derived. For this, we rely on the *a contrario* methodology [19], in which structures are validated by controlling the number of false detections. A key advantage of this approach is the automatic setting of detection parameters in a way that will prevent the numerous junctions that are usually detected in textured areas. Third, the position and scale of the junctions should be detected precisely and be closely related to the image geometry. This will be achieved thanks to a competition between junctions relying on a sound quality measure, the *number of false alarms* associated with a junction.

1.1 Related works

Corner detectors Automatic junction detection has been a very active research field over the last four decades. One of the earliest methods was introduced by [20], considering corners as points which are not self-similar in an image. The similarity of two points is measured by using the sum of squared differences (SSD) between their associated image patches. Harris and Stephens [21] then proposed to approximate the SSD by the autocorrelation in a patch. The resulted cornerness measure is isotropic and has an analytic expansion, called the *Harris matrix*, and is widely used in practice. A large number of detectors rely on this idea, the detection of corners boiling down to the analysis of the eigenvalues of this matrix, see e.g. [22, 23, 24, 25]. In particular, in order to achieve contrast invariance, Kovesi [24] uses phase congruency to derive cornerness measurement. In this work, the image gradient is normalized over small wedges. Alternative measurements of self-similarity have also been proposed, such as the univalue segment assimilating nucleus (USAN) [26] and its variants [27, 28, 29]. Observe that such methods relying on a measure of cornerness actually do not distinguish between different types of junctions. They are therefore usually used to compute generic interest points but their use as local cues for occlusion analysis or figure/ground separation is less clear. Also observe that in order to identify the characteristic scale of a corner, corner detectors usually make use of the linear scale-space [30, 31, 32, 33, 34]. One of the main shortcomings of such approaches is that they quickly lose precision both in localization and scale. An alternative approach is to rely on some contrast invariant multi-scale analysis. In this direction, Alvarez *et. al.* [35] analyze junctions in images using an affine morphological scale space.

Boundary based methods A second popular and efficient way to detect corners relies on explicit boundaries in images. Corners are interpreted as points in the image where a boundary changes direction rapidly and points with high curvature are selected. Many works have concentrated on different and efficient ways to compute the curvature of curves for detecting corners, see [36, 37, 38, 39, 40, 41]. In a different way, it has been proposed in [42] to detect junctions by extending disconnected edges and by filling gaps at junctions. Junction detection can also rely on the grouping of edges in the neighborhood of a candidate junction [43, 44, 45]. Several works have proposed to rely on the contrast invariant level lines of images to detect junctions. Caselles *et. al.* [46] have proposed a method based on a level-set representation, detecting junctions as points where two level lines meet. Cao [47] detects corners as breaking points under a good continuation criterion on image level lines. Corners are detected as points where the curvature along a level line is abnormally high, in an *a contrario* framework. In other approaches, a global segmentation of the image is first performed, followed by a heuristic grouping of edges to form junctions [48, 49]. It should there be emphasized that the segmentation process in itself is a very challenging task. For instance, before detecting junctions, Maire *et. al.* [49] finds contours by the *global probability of boundary* (gPb) detector learned from a human-annotated database. Such approaches permit to benefit from global image interpretation to find junctions. On the other hand, boundary are often imprecise in the neighborhood of a junction,

so that such approaches do not enable an accurate characterization of junctions, that in turn can be used to refine edges.

Template-based methods   Among existing approaches, the model-based or template-based ones are the most suitable for accurate local junction detection. In [50], Deriche and Blaszka present computational approaches for a model-based detection of junctions. A junction model is a 2D intensity function depending on several parameters. Starting from a poorly localized initialization (e.g. from the Harris detector) parameters are then optimized in view of a precise localization. Parida *et al.* [51] suggest a region-based model for simultaneously detecting, classifying and reconstructing junctions. A junction is defined as an image region containing piecewise constant wedges meeting at the central point. This work relies on a template deformation framework and uses minimum description length principle and dynamic programming to obtain the optimal parameters describing the model. This work also involves junction candidates provided by a preliminary corner detector. Ruzon and Tomasi [52] model junctions as points in images where two or more image regions meet. Regions are described by their color distributions, which allows textured objects with the same mean color to be distinguished. Following the model of Parida, Cazorla and Escolano [53, 54] propose both a region-based and an edge-based model for junction classification by using Bayesian methods. The region-based one formalizes junction detection as a radial image segmentation problem under region competition framework and the edge-based one detects junctions as radial edges minimizing some Bayesian classification error. In this work, the edge-based approach is shown to yields more accurate junction detection, but at the price of a large number of false detections, especially in textured regions. More recently, the work of Sinzinger [55] first detects a set of junction candidates by using a preliminary detector (for instance the Harris detector) and then refines those candidates by relying on a radial edge energy.

Method	Corduerness-based	Contour-based	Template-based	our approach (template-based)
Some representative works	[21, 31, 26, 27, 34, 29, 24]	[44, 46, 48, 49, 41]	[50, 51, 53, 55]	— —
Junction detection cues	Cornerness measurements, such as Harris matrix.	Edge map	Gradients or piecewise region consistence	Amplitudes and phases of gradients
Scale of junctions	Possibility of scale detection relying on the linear scale space ([31, 34]).	No scale definition.	Junctions are detected at a given scale (except : scale selection by MDL in [51])	Each junction is associated with an adaptive scale
Classification of junctions	No	No	Yes	Yes
Contrast changes	Dependent (except in [24])	Dependent	Dependent	Robust
Detection criterion	Thresholds on local cornerness	Heuristic grouping principles	Energy minimization from a junction model	Probabilistic perceptual principles

Table 1: Different approaches for junction detection.

The different approaches for junction detection, as well as their properties, are summarized in Table 1.

1.2   Contributions and outline

As explained earlier in the introduction, the goal of this paper is to design a junction detection procedure involving an automatic decision step and permitting a geometrically accurate description of junction properties, including type, localization and scale. This is made possible thanks to the use of an *a contrario* methodology [19]. The resulting detection method will be coined *ACJ* (for *a contrario* junction detection). In short, we first define (in Section 2) junctions as geometrical structures in discrete images and we associate to each candidate junction a quantity called *strength*,

which is designed to be high at locations where several edges (called branches) meet in the image. This strength relies on a well chosen normalization of the gradient, which makes the whole approach robust to local contrast changes. We then rely in Section 3 on a statistical framework in order to decide automatically which junctions deserve to be detected or not in a given image. More precisely, meaningful junctions are detected as those which could not occur by chance under some null hypothesis $\mathcal{H}_0$. This detection step is followed by several exclusion criteria, designed to rule out redundant junctions and to identify precisely the correct scales, positions and complexities (number of branches) of junctions. As a by-product of the scale estimation, the whole approach is robust to resolution changes. Finally, in Section 4, details about implementation and parameter settings are provided. Section 5 analyzes the proposed algorithm experimentally.

2   Contrast Invariant Junctions

The aim of this section is to define junctions in discrete images (Section 2.2) and to associate to each candidate junction a *strength* (Section 2.3). The strength of a junction is defined through both the gradient orientation and magnitude. Robustness to local contrast changes is achieved by normalizing the gradient magnitude, as explained in Section 2.1. The strength of junctions will be used in Section 3 to decide whether junctions are meaningful or not in a given image.

2.1   Gradient normalization

Let us start with some definitions and vocabulary that will be used throughout the paper. A discrete image is a function $I : \Omega \to \mathbb{R}$, where Ω is a rectangular subset of $\mathbb{Z} \times \mathbb{Z}$. We write $\nabla I = (I_x, I_y)$ for the discrete gradient of the image I. For a pixel $\mathbf{q}$ in Ω, we define $\phi(\mathbf{q}) = (\arctan \frac{I_y(\mathbf{q})}{I_x(\mathbf{q})} + \pi/2) modulo(2\pi)$, the direction perpendicular to the gradient at $\mathbf{q}$ (and set $\phi(\mathbf{q}) = \pi$ when $I_x(\mathbf{q}) = 0$). We call this angle the *direction* of the pixel $\mathbf{q}$.

In order to be robust to local contrast changes, we locally normalize the gradient magnitude by dividing it by its average on a small neighborhood. That is, for $\mathbf{q} = (x, y)$, we define $\widehat{\nabla I} = (\widehat{I_x}, \widehat{I_y})$ as

$$\widehat{I_x}(\mathbf{q}) = \frac{I_x(\mathbf{q})}{\langle \sqrt{I_x^2 + I_y^2} \rangle_{\mathcal{N}_{\mathbf{q}}}} \quad \text{and} \quad \widehat{I_y}(\mathbf{q}) = \frac{I_y(\mathbf{q})}{\langle \sqrt{I_x^2 + I_y^2} \rangle_{\mathcal{N}_{\mathbf{q}}}}, \tag{1}$$

where $\mathcal{N}_{\mathbf{q}}$ is a small neighborhood around $\mathbf{q}$ and $\langle . \rangle_{\mathcal{N}_{\mathbf{q}}}$ is the average operator on $\mathcal{N}_{\mathbf{q}}$. The resulting gradient is robust to contrast changes that can be approximated by affine transformations on each neighborhood $\mathcal{N}_{\mathbf{q}}$. An example of the gradient and normalized gradient of an image, obtained with a square neighborhood of size 5×5 around each pixel, is shown in Figure 1. Observe that the phase of the normalized gradient is the same as the phase of the plain gradient.

In Section 3, we will rely on the empirical distribution of the gradient to detect meaningful junctions. More precisely, we will consider the globally *normalized gradient*

$$\widetilde{I_x} = \frac{\widehat{I_x} - \mu_x}{\sigma_x} \quad \text{and} \quad \widetilde{I_y} = \frac{\widehat{I_y} - \mu_y}{\sigma_y}, \tag{2}$$

where μ_x (resp. μ_y) and σ_x (resp. σ_y) are the empirical mean and standard deviation of $\widehat{I_x}$ (resp. $\widehat{I_y}$) over the whole image. The norm $\|\widetilde{\nabla I}\| = \sqrt{\widetilde{I_x}^2 + \widetilde{I_y}^2}$ will be used in the following paragraphs to define the strength of a junction. As we will see, the distribution of this norm in natural images is well approximated by a standard Rayleigh distribution.

2.2   Discrete junction definition

A *junction* is defined as a discrete image structure $\jmath : \left\{ \mathbf{p}, r, \{\theta_m\}_{m=1}^M \right\}$, characterized by its center $\mathbf{p}$, its scale $r \in \mathbb{N}$ and a set of branch directions $\{\theta_1, \ldots \theta_M\}$ around $\mathbf{p}$. The number M of branches
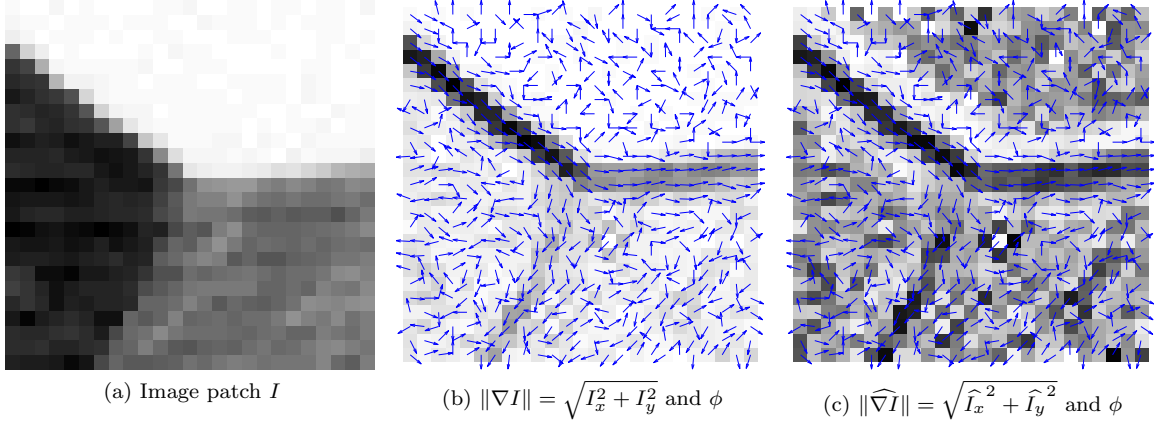
(a) Image patch I (b) $\|\nabla I\| = \sqrt{I_x^2 + I_y^2}$ and ϕ (c) $\|\widehat{\nabla I}\| = \sqrt{\widehat{I_x}^2 + \widehat{I_y}^2}$ and ϕ

Figure 1: Gradient ∇I and normalized gradient $\widehat{\nabla I}$ on an image patch

in the junction is called the order of the junction : when $M = 2$, 3 or 4, we speak respectively of L, Y or X-junctions. The discrete set $\mathcal{D}(r)$ of possible directions at a given scale r is defined as

$$\mathcal{D}(r) := \left\{ \frac{2\pi k}{K(r)} \; ; \; k \in \{1, \dots K(r)\} \right\}, \tag{3}$$

where $K(r)$ is the number of possible discrete directions permitting a one pixel precision, that is $K(r) = \lfloor 2\pi r \rfloor$.

For a given scale r and a given direction θ in $\mathcal{D}(r)$, we define the *branch* of direction θ at $\mathbf{p}$ as the disk sector

$$S_{\mathbf{p}}(r, \theta) := \Big\{ \mathbf{q} \in \Omega; \mathbf{q} \neq \mathbf{p}, \; \|\overrightarrow{\mathbf{pq}}\| \leq r,$$
$$d_{2\pi}(\alpha(\overrightarrow{\mathbf{pq}}), \theta) \leq \Delta(r) \Big\}, \tag{4}$$

where $\Delta(r)$ is a precision parameter, $\alpha(\overrightarrow{\mathbf{pq}})$ is the angle of the vector $\overrightarrow{\mathbf{pq}}$ in $[0, 2\pi]$ and where $d_{2\pi}$ is the distance along the unit circle, defined as $d_{2\pi}(\alpha, \beta) = \min(|\alpha - \beta|, 2\pi - |\alpha - \beta|)$. We note $J(r, \theta)$ the number of pixels in a sector of direction θ at scale r. This size depends on the scale r but also slightly changes with the direction θ since the image is defined on a discrete grid.

Finally, we require that two branches of a given junction do not intersect. It follows that the angle between two directions in a junction $\jmath$ must be larger than $2\Delta(r)$. As a consequence, the number of possible directions in a junction at scale r is smaller than $\frac{\pi}{\Delta(r)}$. An example of a junction with three branches is shown on Figure 2.

2.3   Junction strength

The strength of a junction will be defined from the strength of its branches. The strength of a branch is a measure of how well the corresponding angular sector agrees with the pixels it contains. A pixel $\mathbf{q}$ contributes all the more to the strength as its direction $\phi(\mathbf{q})$ is in agreement with the direction $\alpha(\overrightarrow{\mathbf{pq}})$ and as the normalized gradient $\|\widehat{\nabla I}(\mathbf{q})\|$ is large. Precisely, the strength of a branch is defined as

Definition 1 (Strength of a branch) *Let $\mathbf{p}$ be a pixel, r a positive scale and θ a direction in $\mathcal{D}(r)$. The strength of the branch $S_{\mathbf{p}}(r, \theta)$ is defined as the quantity*

$$\omega_{\mathbf{p}}(r, \theta) = \sum_{\mathbf{q} \in S_{\mathbf{p}}(r, \theta)} \gamma_{\mathbf{p}}(\mathbf{q}), \tag{5}$$
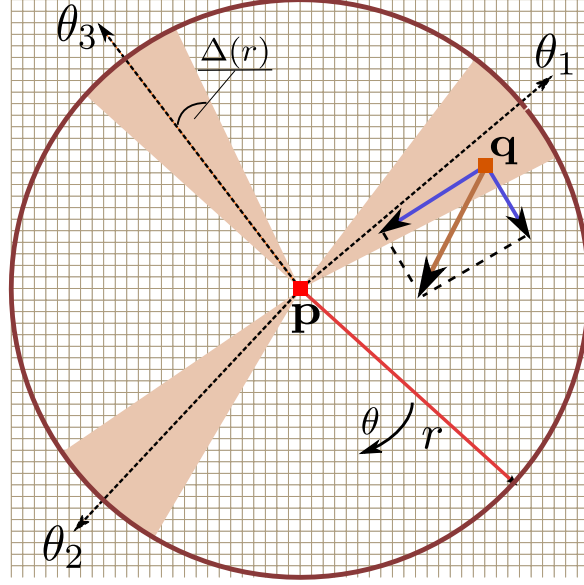
5

Figure 2: A junction with three branches. Each branch is represented by a disk sector $S_{\mathbf{p}}(r,\theta)$ with an angular aperture of $2\Delta(r)$.

where

$$\gamma_{\mathbf{p}}(\mathbf{q}) = \|\widetilde{\nabla I}(\mathbf{q})\| \cdot \max\Big(|\cos(\phi(\mathbf{q}) - \alpha(\overrightarrow{\mathbf{pq}})| -$$
$$|\sin(\phi(\mathbf{q}) - \alpha(\overrightarrow{\mathbf{pq}}))|, 0\Big). \tag{6}$$

The *max* operator has been chosen so that the contribution of the pixel is equal to zero if the difference between α and ϕ is larger than $\pi/4$. Other choices could be made. The larger the strength of a branch, the more likely it is that the branch corresponds to an edge. Figure 3 (c) shows the values of $\gamma_{\mathbf{p}}(\mathbf{q})$ for the image patch shown in Figure 3 (a).

The strength of a junction is then derived from the strength of its branches as follows:

Definition 2 (Strength of a junction) *The strength of a junction $\jmath : \left\{\mathbf{p}, r, \{\theta_m\}_{m=1}^{M}\right\}$ is defined as*

$$t(\jmath) := \min_{m=1\dots M} \omega_{\mathbf{p}}(r, \theta_m). \tag{7}$$

Starting from this definition, a first naive algorithm for junction detection can be developed. The idea is to detect, for a fixed scale r and a given threshold t, all the junctions in I having a strength greater than t. In practice, testing all possible junctions for every point $\mathbf{p}$ in Ω is computationally heavy. Among all the potential branches at a given point $\mathbf{p}$, we restrict ourselves to the directions θ in the discrete set $\mathcal{D}(r)$ where the periodic function $\omega_{\mathbf{p}}(r, .)$ reaches a local maximum. Moreover, in order for branches not to intersect, we impose that the local maximality holds over a length $2\Delta(r)$, that is $\omega_{\mathbf{p}}(r, \theta)$ is greater than $\omega_{\mathbf{p}}(r, \theta')$ for $\theta - \Delta(r) \leq \theta' \leq \theta + \Delta(r)$. The set of these local maxima can be computed quickly, for instance by using a non-maximum suppression (NMS) procedure (see [56, 57]). In practice, if two local maxima are equal and located at a distance smaller than $\Delta(r)$, one of them is discarded.

The case of L-junctions In order not to detect all edge points as L-junctions, junctions for which $M = 2$ and whose branches are opposite (that is, with two directions θ_1 and θ_2 such that $d_{2\pi}(\theta_1, \theta_2 + \pi) \leq 2\Delta(r)$) are discarded.

The overall detection procedure is summarized in Algorithm 1.

Algorithm 1 Junction detection in an image I at a given scale r.

Input: A discrete image I, an order M, a positive scale r, a number K of discrete directions, a precision Δ and a threshold t.

1: compute $\widetilde{\nabla I}$ at each pixel using Equation (1);
2: set $\mathcal{D} = \left\{ \frac{2\pi k}{K} \; ; \; k = 1, \ldots K \right\}$
3: **for** each pixel $\mathbf{p}$ **do**
4:     (1) compute $\omega_{\mathbf{p}}(r, \theta)$ for each θ in $\mathcal{D}$, using Equations (5) and (6);
5:     (2) use a NMS procedure to only keep directions θ such that $\omega_{\mathbf{p}}(r, \theta)$ is locally maximum on $[\theta - \Delta, \theta + \Delta]$; call Θ the set of these directions;
6:     (3) find all the subsets Θ' of size M in Θ such that $\min_{\theta \in \Theta'} \omega_{\mathbf{p}}(r, \theta) \geq t$; for each of these subsets, mark $\jmath : \left\{ \mathbf{p}, r, \Theta' \right\}$ as a junction if $M \neq 2$, or if $M = 2$ and $\Theta' = \{\theta_1, \theta_2\}$ with $d_{2\pi}(\theta_1, \theta_2 + \pi) \leq 2\Delta$.
7: **end for**
8: **return** The set of detected junctions.

The first parts of the procedure are illustrated in Figure 3. In this example, the pixel $\mathbf{p}$ is chosen as the center of Figure 3 (a). Figure 3 (d) shows in blue the values of $\omega_{\mathbf{p}}(r, \theta)$ when the angle θ spans the periodic set $\mathcal{D}(r)$ and shows in red the semi-local maxima kept after the NMS procedure (step (2) of the loop in Algorithm 1). Figures 3 (e) represent respectively a candidate L-junction, a candidate Y-junction and a candidate X-junction at $\mathbf{p}$.

The main drawback of this detection algorithm is that the threshold t on the junction strength remains the same whatever the junction scale and order and whatever the image size. Setting such a threshold globally is not easy and can lead to over-detect at some scales and under-detect at others. The goal of the next section is to provide detection thresholds that adapt to the junction scale and order, as well as to the image size. For this purpose, we resort to an *a contrario* methodology.

3 An *a contrario* Approach for Junction Detection

The *a contrario* detection theory has been primarily proposed by Desolneux *et. al.* [19]. This methodology is inspired by geometric grouping laws governing low-level human vision, known as Gestalt laws [3], and states that meaningful structures in images are structures which are very unlikely under some hypothesis of randomness. The method has been extensively tested and successfully applied to various problems in image processing and computer vision, see e.g. [47, 58, 59, 60]. A complete overview of these methods can be found in [61]. In this section, this methodology is adapted to the detection of meaningful junctions in images.

3.1 Null hypothesis

The goal of the following sections is to set detection thresholds on junction strengths in such a way that no junction will be detected in a "generic random image". Let us precise what "generic random image" stands for here. Let $\mathbf{I}$ be a random image. For each pixel $\mathbf{q}$, we write $\|\widetilde{\nabla \mathbf{I}}(\mathbf{q})\|$ and $\phi_{\mathbf{I}}(\mathbf{q})$ the random variables corresponding to the value and orientation at this pixel. We say that these variables follow the null hypothesis $\mathcal{H}_0$ if

1. $\forall \mathbf{q} \in \Omega, \|\widetilde{\nabla \mathbf{I}}(\mathbf{q})\|$ follows a Rayleigh distribution with parameter 1 ;

2. $\forall \mathbf{q} \in \Omega, \phi_{\mathbf{I}}(\mathbf{q})$ is uniformly distributed over $[0, 2\pi]$;

3. the family $\{\|\widetilde{\nabla \mathbf{I}}(\mathbf{q})\|, \phi_{\mathbf{I}}(\mathbf{q})\}_{\mathbf{q} \in \Omega}$ is made of independent random variables.

Let us comment on the first assumption. In [62], Ruderman *et al.* observe that if we normalize the logarithm of an image intensity by its local mean and standard deviation in a neighborhood $\mathcal{N}_p$ around each pixel, "*the histogram of pixel values has Gaussian tails, and the distribution of*

(a) original image

(b) $\|\widetilde{\nabla I}\|$ and ϕ

(c) $\gamma_{\mathbf{p}}(\mathbf{q})$, the contributions of pixels to the junction strength

(d) branch strength $\omega_{\mathbf{p}}(r, \theta)$ in blue and semilocal maxima in red;

(e) From left to right, a potential L-junction, a potential Y-junction and a potential X-junction respectively;
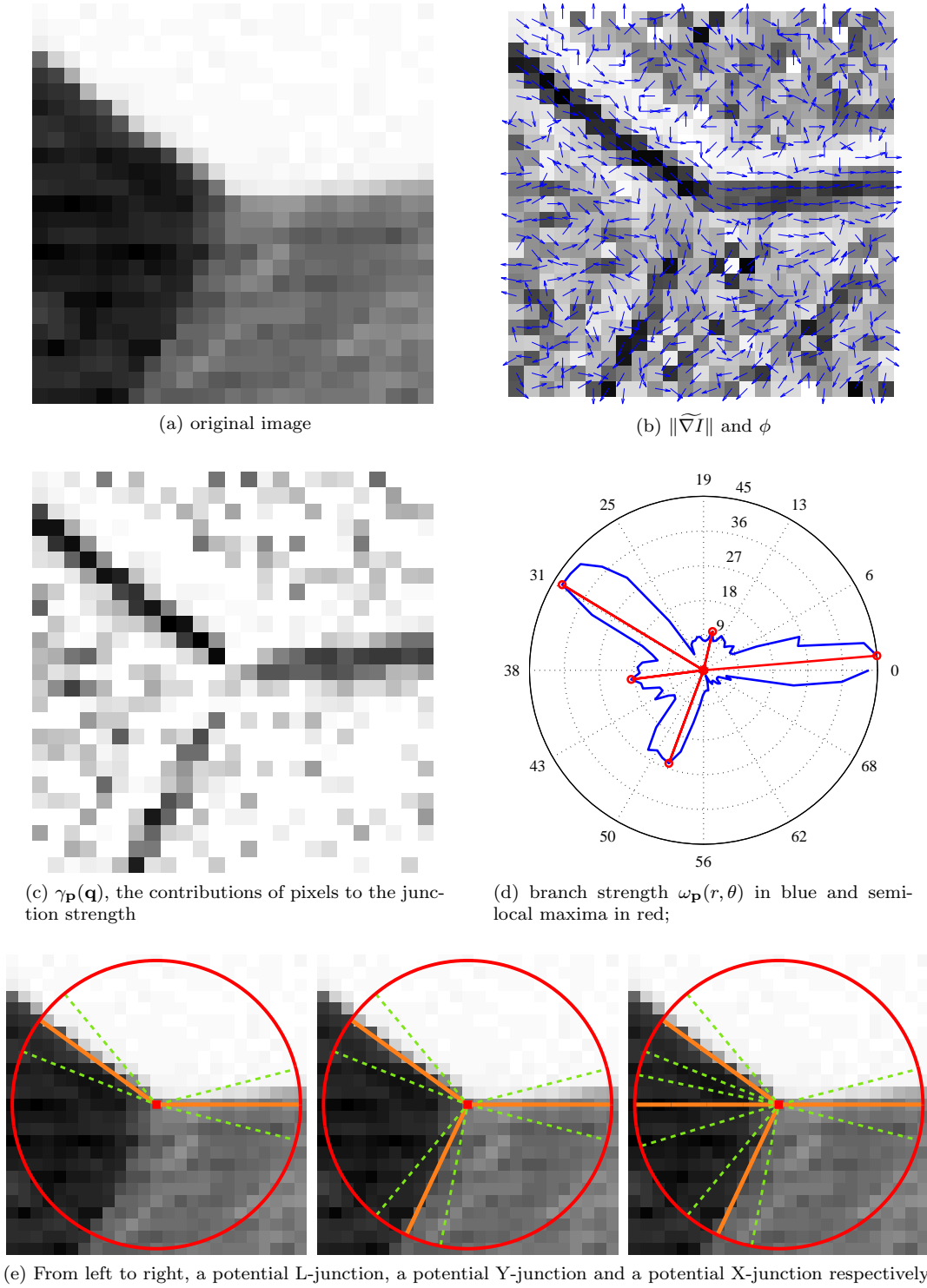
Figure 3: Computation of junction candidates at scale $r = 12$, when $\mathbf{p}$ is the center of the original image (a). The blue curve in (d) shows the strength $\omega_{\mathbf{p}}(r, \theta)$ as a function of the direction θ along the circle. The directions that remain after the NMS procedure are shown in red. For $r = 12$, $K(r) = 75$, and $2\Delta(r) = 0.289$ (see Section 4.1).

8

gradients in the 'variance-normalized' image is almost exactly the Rayleigh distribution". This result can be extended to our modified derivatives $\widehat{I}_x$ and $\widehat{I}_y$: except on a small neighborhood around 0, their distribution is well approximated by a Gaussian distribution. This is illustrated for a particular image I in Figure 4. In this Figure, the empirical distributions of $\widehat{I}_x$ and $\widehat{I}_y$ are drawn in blue, and the best Gaussian fits are drawn in red. If we except a peak around 0, the fit is excellent. This approximation is of particular interest for junction detection since the gradients around a junction usually take large values. Observe that this Gaussian approximation is not valid for the classical gradient (I_x, I_y), whose distribution is quite heavy tailed. It follows that the distributions of $\widetilde{I}_x$ and $\widetilde{I}_x$ are usually well approximated by the standard normal distribution (that is, a Gaussian distribution with zero mean and unit variance) and the norm $\|\widetilde{\nabla I}\|$ in natural images approximately follows a central chi-distribution with 2 degrees of freedom (also known as the Rayleigh distribution of parameter 1). This is confirmed experimentally by the example shown in Figure 4. Moreover, we performed a small scale experiment whose results are shown on Figure 5a. In blue are displayed the empirical distributions of $\|\widetilde{\nabla I}\|$ for 50 different natural images, as well as the density of the Rayleigh distribution of parameter 1 (in red).
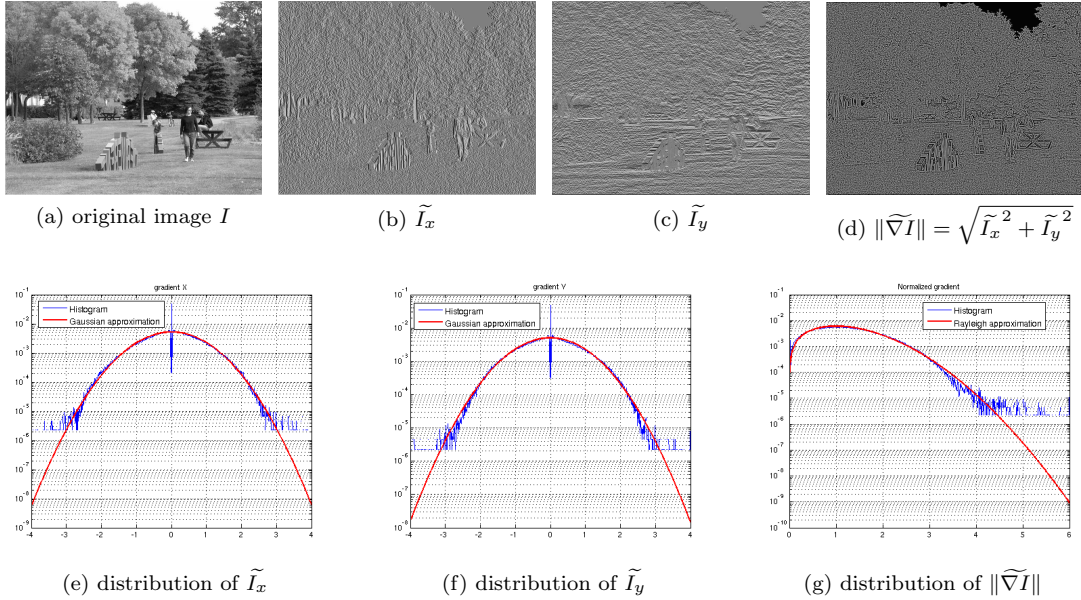

(a) original image I (b) $\widetilde{I}_x$ (c) $\widetilde{I}_y$ (d) $\|\widetilde{\nabla I}\| = \sqrt{\widetilde{I}_x{}^2 + \widetilde{I}_y{}^2}$

(e) distribution of $\widetilde{I}_x$ (f) distribution of $\widetilde{I}_y$ (g) distribution of $\|\widetilde{\nabla I}\|$

Figure 4: Approximations of the distributions of $\widetilde{I}_x$, $\widetilde{I}_y$ and $\|\widetilde{\nabla I}\|$ (given by Equations (1) and (2)). The blue curves are the empirical histograms and the red curves are the approximated distributions. (e) and (f) are Gaussian distributions and (g) is a Rayleigh distribution of parameter 1.

3.2 Distribution of $\mathbf{t}(\jmath)$ under $\mathcal{H}_0$

Let $\jmath : \left\{ \mathbf{p}, r, \{\theta_m\}_{m=1}^M \right\}$ be a junction in $\mathbf{I}$ and assume that the normalized gradients and directions of $\mathbf{I}$ follow the null hypothesis $\mathcal{H}_0$. Then the strengths $\omega_{\mathbf{p}}(r, \theta_m)$ of the different branches $S_{\mathbf{p}}(r, \theta_m)$ are independent random variables (since branches do not intersect). Recall also that the strength of a junction is the minimum of the strengths of its branches. Thus, if we note $\mathbf{t}(\jmath)$ the random variable measuring the strength of $\jmath$,

$$\mathbb{P}_{\mathcal{H}_0}[\mathbf{t}(\jmath) \geq t] = \mathbb{P}_{\mathcal{H}_0}[\forall m, \ \omega_{\mathbf{p}}(r, \theta_m) \geq t]$$
$$= \prod_{m=1}^M \mathbb{P}_{\mathcal{H}_0}[\omega_{\mathbf{p}}(r, \theta_m) \geq t]. \tag{8}$$
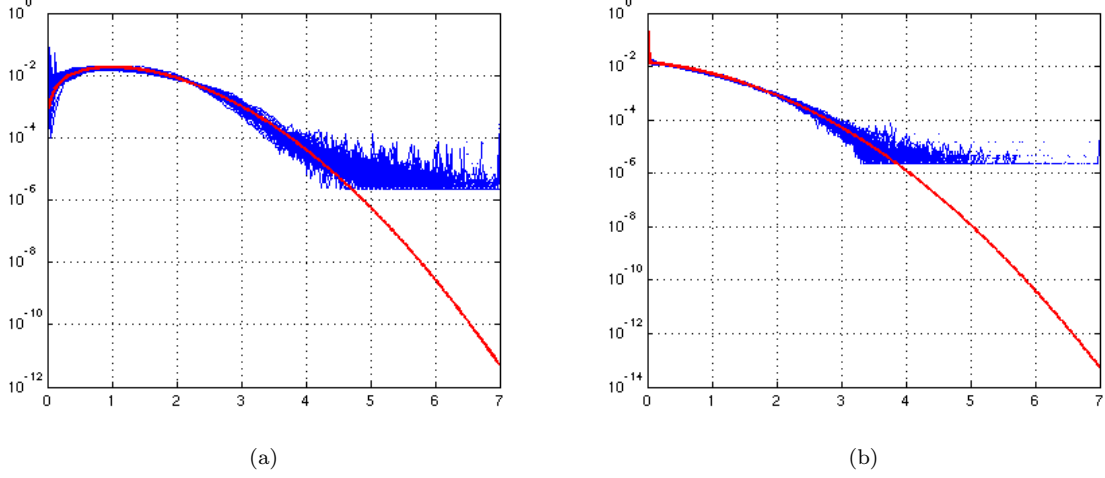
9

Figure 5: Distributions of $\|\widetilde{\nabla I}\|$ (blue curves in (a)) and $\gamma_{\mathbf{p}}(\mathbf{q})$ (blue curves in (b)) for 50 different natural images. $Rayleigh(1)$ density (red curve in (a)) and the corresponding strength distribution $\mu(z)$ (red curve in (b)) are displayed for comparison.

Now, the strength of each branch, $\omega_{\mathbf{p}}(r, \theta_m)$, is itself a sum (over the angular sector $S_{\mathbf{p}}(r, \theta_m)$) of i.i.d. random variables $\gamma_{\mathbf{p}}(\mathbf{q})$. For two given points $\mathbf{p}$ and $\mathbf{q}$ in Ω, the direction $\alpha(\overrightarrow{\mathbf{pq}})$ is a constant in $[0, 2\pi]$. This implies that under the hypothesis $\mathcal{H}_0$, the random angle $\phi_{\mathbf{I}}(\mathbf{q}) - \alpha(\overrightarrow{\mathbf{pq}})$ is still uniformly distributed on $[0, 2\pi]$. As a consequence, each $\gamma_{\mathbf{p}}(\mathbf{q})$ can be written as a product $X \cdot \max(|\cos \theta| - |\sin \theta|, 0)$, where X and θ are independent, X follows a Rayleigh distribution and θ is uniformly distributed on $[0, \pi]$. Finally, the distribution of each $\gamma_{\mathbf{p}}(\mathbf{q})$ under $\mathcal{H}_0$ can be written

$$\mu(z) = \frac{1}{2}\delta_0(z) + H(z) \cdot \frac{1}{\sqrt{\pi}} e^{-\frac{z^2}{4}} \cdot \mathrm{erfc}(\frac{z}{2})dz, \tag{9}$$

where δ_0 is a Dirac mass at 0, H is the Heaviside function ($H(z) = 1$, for $z \geq 0$, and $H(z) = 0$ otherwise) and erfc is the complementary error function, $\mathrm{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty e^{-s^2} ds$.

The empirical distribution of $\gamma_{\mathbf{p}}(\mathbf{q})$ on 50 natural images is displayed in Figure 5b, along with its theoretical distribution μ, showing excellent fit to the model. Under the hypothesis $\mathcal{H}_0$, the law of the strength of a branch $\omega_{\mathbf{p}}(r, \theta_m)$ (that is, of the sum of pixel contributions given by Formula (5)) is obtained by convolving $J(r, \theta_m)$ times with itself the distribution μ, where $J(r, \theta_m)$ is the size of a sector of orientation θ_m at scale r. Finally,

Proposition 1 *Let $\jmath : \left\{ \mathbf{p}, r, \{\theta_m\}_{m=1}^{M} \right\}$ be a junction and suppose that the hypothesis $\mathcal{H}_0$ is satisfied, then the probability that the random variable $\mathbf{t}(\jmath)$ is larger than a given threshold t is*

$$F_{\jmath}(t) := \mathbb{P}_{\mathcal{H}_0}[\mathbf{t}(\jmath) \geq t] = \prod_{m=1}^{M} \int_t^{+\infty} \overset{J(r,\theta_m)}{\underset{j=1}{\star}} \mu(dz), \tag{10}$$

where $J(r, \theta_m)$ is the size of a sector of orientation θ_m at scale r.

3.3 Meaningful junctions

Thanks to the previous computations, we are now in a position to automatically fix detection thresholds on junction strengths. Indeed, thresholds are set so that the average number of false detections under the null hypothesis is controlled. This is obtained by thresholding the probability (10) and by taking into account the number of possible discrete junctions in the image.

Number of tests In this paragraph, we assume that the order M is fixed, and we call $\mathcal{J}(M)$ the set of all possible junctions of order M in the discrete image I. The size of $\mathcal{J}(M)$ depends on several parameters: the minimum and maximum authorized scales r_{min} and r_{max}, the size N (number of pixels) of I and the precision $\Delta(r)$ of branches at each scale r. The practical setting of these parameters will be discussed in Section 4.1.

At a given location $\mathbf{p}$, once the first branch is chosen among the $K(r)$ possible directions, the second direction must be chosen in such a way that the two branches of width $2\Delta(r)$ do not intersect, which means that only $K(r)(1 - 2\frac{\Delta(r)}{\pi})$ directions are authorized. Therefore, at each location $\mathbf{p} \in \Omega$, and for a given scale r, the number of possible junctions of order M is always smaller than

$$\frac{1}{M!} \prod_{m=1}^{M} K(r) \left(1 - 2(m-1)\frac{\Delta(r)}{\pi} \right).$$

It follows that the size of the set $\mathcal{J}(M)$ is upper bounded by

$$\#\mathcal{J}(M) = \frac{N}{M!} \cdot \sum_{r=r_{min}}^{r_{max}} \prod_{m=1}^{M} K(r) \left(1 - 2(m-1)\frac{\Delta(r)}{\pi} \right). \tag{11}$$

ϵ-meaningful junctions The next definition and the following proposition explain how to fix thresholds on junctions strengths in order to control the average number of false detections.

Definition 3 *(ϵ-meaningful junction) Let I be a discrete image. For $\epsilon > 0$, a junction $\jmath$ of order M and at scale r is said to be ϵ-meaningful if*

$$NFA(\jmath) := \#\mathcal{J}(M) \cdot F_{\jmath}(t(\jmath)) \leq \epsilon. \tag{12}$$

The quantity $NFA(\jmath)$ is a measure of the meaningfulness of the junction: the smaller it is, the more meaningful the junction $\jmath$. A junction of order M and scale r is detected as ϵ-meaningful in I if its strength $t(\jmath)$ is larger than the threshold

$$t(r, \epsilon) := \min \left\{ t; \ F_{\jmath}(t) \leq \frac{\epsilon}{\#\mathcal{J}(M)} \right\}. \tag{13}$$

Notice that for a fixed ϵ, this formula yields a different threshold on $t(\jmath)$ for each value of the scale r. The value ϵ is easy to interpret: it corresponds to an expected number of false detections in I. Indeed,

Proposition 2 *Let $\mathbf{I}$ be a discrete random image and assume that the null hypothesis $\mathcal{H}_0$ is satisfied. Let M be a positive integer. The expectation of the number of ϵ-meaningful junctions of order M in $\mathbf{I}$ is smaller than ϵ.*

Proof 1 *First, observe that if X is a random variable and if we define $F(t) = \mathbb{P}[X \geq t]$, then for all β in $[0,1]$, $\mathbb{P}[F(X) \leq \beta] \leq \beta$. Thus, if $\jmath$ is a junction of scale r in $\mathcal{J}(M)$, Formula (10) yields*

$$\mathbb{P}_{\mathcal{H}_0}\left[NFA(\jmath) \leq \epsilon \right] = \mathbb{P}_{\mathcal{H}_0}\left[F_{\jmath}(\mathbf{t}(\jmath)) \leq \frac{\epsilon}{\#\mathcal{J}(M)} \right]$$
$$\leq \frac{\epsilon}{\#\mathcal{J}(M)}. \tag{14}$$

Finally,

$$\mathbb{E}_{\mathcal{H}_0}\left[\#\{\jmath \in \mathcal{J}(M); NFA(\jmath) \leq \epsilon\} \right]$$
$$= \sum_{\jmath \in \mathcal{J}(M)} \mathbb{P}_{\mathcal{H}_0}\left[NFA(\jmath) \leq \epsilon \right]$$
$$\leq \sum_{\jmath \in \mathcal{J}(M)} \frac{\epsilon}{\#\mathcal{J}(M)} \leq \epsilon. \tag{15}$$

Observe that in the definition of the NFA, the correcting factor $\#\mathcal{J}(M)$ is independent of the scale r. We could have used different correcting factors, depending on the junction scale r, in order to favor some particular scales in the detection, and still have a result similar to Proposition 2.

3.4 Junctions or nearby edges?

In images, some structures may appear as junctions at large scales without being detected as such when the scale is small. For instance, when three straight edges have close endpoints and are supported by concurring lines, an Y-junction may be detected even if the actual edges do not physically meet, when the possible junction contains a gap at its center. Deciding between a real junction and interacting edges is actually far from trivial. Psychophysical experiments [4] suggest that human may find junctions although there is a small gap at the center. In practice, we observed that detections were visually more satisfying when removing junctions with large gaps at their center. This restriction is imposed by computing for each ϵ-meaningful junction $\jmath : \{\mathbf{p}, r, \{\theta_m\}_{m \in \{1,...,M\}}\}$ a minimum scale of detection, defined as (see Figure 7)

$$r_d[\jmath] = \min \left\{ r' \leq r; \forall s \in [r', r], \ \ \exists \jmath' : \{\mathbf{p}, s, \{\theta'_m\}_{m=1}^M\} \right.$$
$$\left. \text{s.t. } \jmath' \text{ is } \epsilon\text{-meaningful.} \right\}, \tag{16}$$

and by removing all ϵ-meaningful junctions such that $r_d[\jmath] > r_{gap}$. In practice and within the scope of this paper, this threshold is chosen as $r_{gap} = 12$. In order to be fully scale invariant, this threshold could be replaced by a value proportional to the junction scale r.

3.5 Redundant detections and maximality

Redundancy: scale and location As it is common when analyzing geometrical structures in images, junctions are usually detected in a redundant way. A single structure in the image may yield many detections. First, junctions are detected over a range of scale. For a single ideal junction in the image, meaningful junctions will be detected for scales both smaller and larger than the one of the underlying structure, see Figure 6, middle, where several junctions having the same center but different scales, and corresponding to the same ideal Y-junctions, are displayed. Second, several junctions with slightly different locations are detected for a single underlying structure. This is all the more strong as there is blur in the image. An example of such redundant detections is displayed in Figure 6, right. Moreover, both type of over-detections (multiple scales and multiple locations) are usually combined in images. These redundancies are addressed in the next paragraph thanks to an *exclusion* principle.
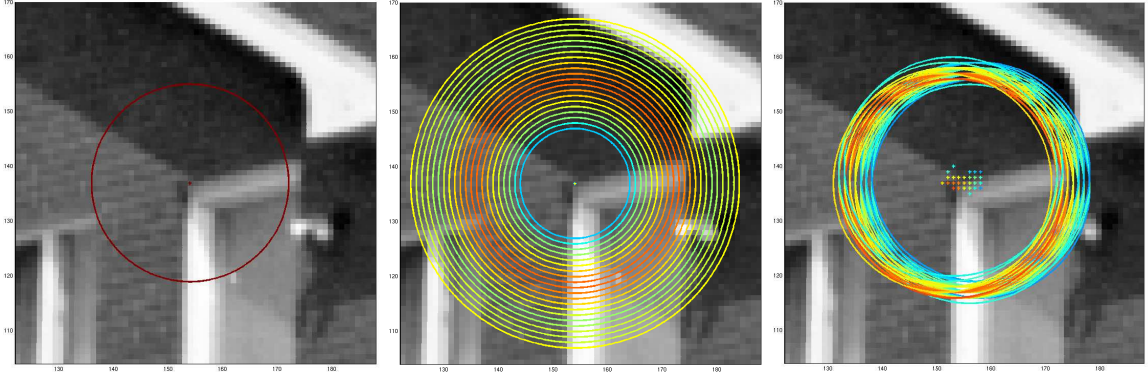

Figure 6: Redundancy of junction detection. For the sake of clarity, each junction is represented by a circle and its center, the radius of the circle is equal to the scale of the junction, and the color of the circle depends on the NFA value (red corresponds to small values, i.e. very meaningful junctions and blue corresponds to high values). **Left**: a junction $\jmath : \{\mathbf{p}, r, \{\theta_m\}_{m \in \{1,...,3\}}\}$, with $r = 20$. **Mid**: all junctions detected at the same point $\mathbf{p}$, with different scales. **Right**: all Y-junctions detected in the neighborhood of $\mathbf{p}$, with the same directions and scale.
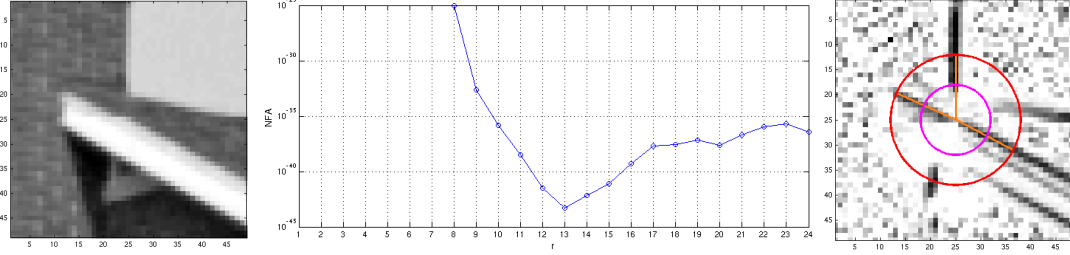
Figure 7: Scale of a junction. Left: image I. The point $\mathbf{p}$ is chosen as the center of I. Middle: for each scale r (in abscissa) is displayed the smallest NFA($\jmath$) observed for a junction $\jmath$ of order 3 centered at $\mathbf{p}$. Right: the smallest scale of detection $r_d = 8$ in magenta and the scale corresponding to the smallest NFA in red (found for $r = 13$), represented on top of the strength $\gamma_{\mathbf{p}}(\mathbf{q})$.

Maximal junctions In order to choose the right representative among all redundant detections, we use an exclusion principle, called maximality. We assign to each junction $\jmath$ a neighborhood $\mathcal{N}'_{\jmath}$. For a given order M, we only keep the junctions not containing any more meaningful junction in its neighborhood. That is, we only keep junctions that are maximal in the following sense.

Definition 4 (Maximal ϵ-meaningful junction of order M) *A junction $\jmath : \{\mathbf{p}, r, \{\theta_m\}_{m=1}^{M}\}$ is said to be a maximal ϵ-meaningful junction of order M if $\jmath$ is ϵ-meaningful and if NFA($\jmath$) $\leq$ NFA($\jmath'$) for any junction $\jmath' : \{\mathbf{p}', r', \{\theta'_m\}_{m=1}^{M}\}$, with $\mathbf{p}' \in \mathcal{N}'_{\jmath}$.*

Observe that in this definition the use of the NFA is the keypoint. Indeed, it permits to compare structures at different scales. Using the strengths $t(\jmath)$ to carry out this comparison would require a well chosen normalization depending on the scale.

In order to select maximal meaningful junctions, the most meaningful junction is first considered. All junctions having it as a neighbor are then removed. Then we proceed to the next most meaningful junction and iterate the same procedure until all junctions have been treated.
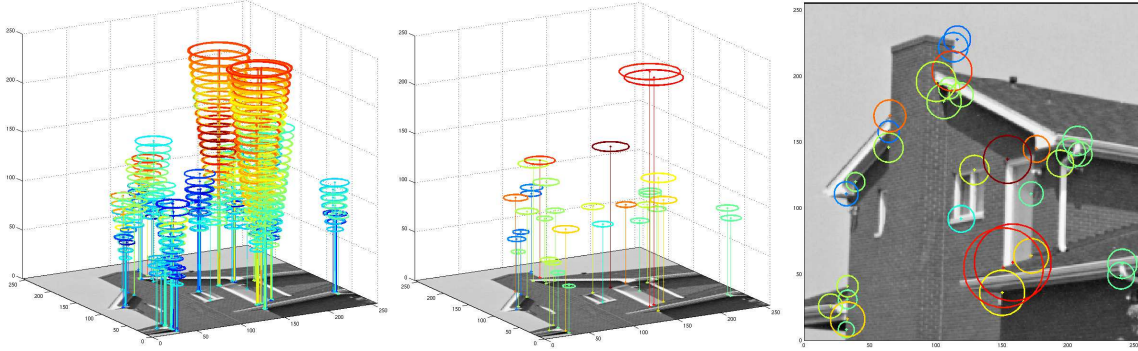

Figure 8: Maximal meaningful junctions of order 3. Each Y-junction is represented by a circle and its center. The radius of the circle indicates the junction scale r, and the color of the circle corresponds to the NFA of the junction (the cooler the color, the larger the NFA). **Left**: all ϵ-meaningful Y-junctions, with $\epsilon = 1$. **Mid**: the maximal meaningful Y-junctions. **Right**: the maximal meaningful Y-junctions displayed over the image.

A result of these selection rules is illustrated in Figure 8 for $M = 3$. In practice, the spatial neighborhood $\mathcal{N}'_p$ used for maximality is chosen as a disk centered at $\mathbf{p}$, with a radius $r_d[\jmath]$, the minimum scale of detection of $\jmath$ as defined by Formula (16).

Masking and junction order When a Y-junction is perceived in a image, the underlying L-junctions are usually not perceived and we decided not to detect them. This masking phenomenon

(for a given junction, no junction made of a subset of its branches should be detected) is easily implemented by the following second exclusion principle: locally, only the more complex junction (the one with the largest order) is kept.

Definition 5 *(Maximal ϵ-meaningful junction)* *A junction $\jmath : \{\mathbf{p}, r, \{\theta_m\}_{m=1}^M\}$ is said to be a maximal ϵ-meaningful junction if $\jmath$ is a maximal ϵ-meaningful junction of order M and if there is no maximal ϵ-meaningful junction of order M' located at $\mathbf{p}'$ with $M' > M$ and $p' \in \mathcal{N}_{\jmath}'$.*

3.6 The three algorithmic steps for junction detection

The different steps of the junction detection procedure are summarized in Algorithms 2 (the *a contrario* detection), 3 (maximality of order M) and 4 (maximality). The complete algorithm pipeline includes a speed-up step and an optional precision refinement and will be described in Section 4 and summarized in Algorithm 7.

Algorithm 2 *A contrario* junction detection

Input: An image I, a maximal order M' and parameters $\epsilon, r_{min}, r_{max}, r_{gap}$.
Output: A list of junctions $Jlist$ and the corresponding list of minimum detection scales $r_d[Jlist]$.

1: Compute $\widetilde{\nabla} I$ at each pixel using Equation (1);
2: For each value of M between 2 and M', compute $\#\mathcal{J}(M)$ by using Equation (11);
3: **for** each pixel $\mathbf{p}$ **do**
4:   Let ρ be a $M' \times r_{max}$ matrix (used to record the values r_d), and fill it with zeros.
5:   **for** $r = 1$ to r_{max} **do**
6:     Compute the branch strength $\omega_{\mathbf{p}}(r, \theta)$ for each direction θ in $\mathcal{D}(r)$, using Equations (5) and (6).
7:     Use a NMS procedure to only keep semi-local maxima of $\omega_{\mathbf{p}}(r, .)$; call $\Theta = \{\theta_m\}_{m=1}^{\#\Theta}$ the set of these directions.
8:     Sort the vector $\omega_{\mathbf{p}}(r, \Theta)$ in a descending order
       $(index, \upsilon) \leftarrow \text{sort}(\omega_{\mathbf{p}}(r, \Theta))$.
9:     *Suppression of alignments*:
         if $M = 2$ and $d_{2\pi}(\theta_{index(1)}, \theta_{index(2)} + \pi) \leq 2\Delta(r)$
           $\theta_{index(2)} = \theta_{index(3)}$
         end if
10:    **for** $M = 2$ to M' **do**
11:      Define the junction $\jmath : \{\mathbf{p}, r, \{\theta_m\}_{m=index(1)}^{index(M)}\}$, with strength $t(\jmath) = \upsilon(index(M))$
12:      Compute $\log \text{NFA}(\jmath)$ by using Equation (24)
13:      **if** $\log \text{NFA}(\jmath) \leq \log \epsilon$ **then**
14:        **if** $\rho(M, r - 1) \neq 0$ **then**
15:          $\rho(M, r) = \rho(M, r - 1)$ and $r_d[\jmath] = \rho(M, r - 1)$
16:        **else**
17:          $\rho(M, r) = r$ and $r_d[\jmath] = r$
18:        **end if**
19:        **if** $r_d[\jmath] < r_{gap}$ **then**
20:          Accept the junction proposal $\jmath$: $Jlist \leftarrow \jmath$
21:        **end if**
22:      **end if**
23:    **end for**
24:   **end for**
25: **end for**

Algorithm 2 does not include the two maximality steps (described in Algorithms 3 and 4). Notice however that line 11 of Algorithm 2 is a first step towards maximality, since only the best junction of a given order M is tested at each point. This permits to speed-up the algorithm by excluding

junctions that will obviously not be maximal. If we wish to compute all ϵ-meaningful junctions in an image, and not only maximal junctions, lines 11 to 22 should be replaced by Procedure 5.

Algorithm 3 Maximality for junctions of order M

Input: $Jlist$: the list of L ϵ-meaningful junctions of order M detected in I and the list $r_d[Jlist]$.

1: Sort $Jlist$ in a descending order according to the NFA of junctions: $Jlist \leftarrow \textbf{Sort}(Jlist, \text{NFA})$
2: Remove non-maximal meaningful junctions
3: **for** $j = 1$ to L **do**
4:     $\jmath_1 \leftarrow$ location of junction $Jlist[j]$;
5:     **for** $k = j + 1$ to L **do**
6:         $\jmath_2 \leftarrow$ location of junction $Jlist[k]$
7:         **if** $\|\mathbf{p}_{\jmath_1} - \mathbf{p}_{\jmath_2}\|_2 \leq r_d[\jmath_1]$ **then**
8:             remove $Jlist[j]$, then **break**;
9:         **end if**
10:     **end for**
11: **end for**

Algorithm 4 Maximality

Input: A maximum order M'. The list $Jlist$ of maximal ϵ-meaningful junctions of all orders smaller than M' in I and the list $r_d[Jlist]$. L is the size of $Jlist$.

1: **for** $j = 1$ to L **do**
2:     $\jmath_1 \leftarrow$ location of junction $Jlist[j]$;
3:     **for** $k = 1$ to L **do**
4:         $\jmath_2 \leftarrow$ location of junction $Jlist[k]$
5:         **if** $\|\mathbf{p}_{\jmath_1} - \mathbf{p}_{\jmath_2}\|_2 \leq r_d[\jmath_1]$ and $M_{\jmath_1} < M_{\jmath_2}$ **then**
6:             remove $Jlist[j]$, then **break**;
7:         **end if**
8:     **end for**
9: **end for**

Procedure 5 Computing all ϵ-meaningful junctions

for each junction $\jmath$: $\{\mathbf{p}, r, \{\theta_{m_k}\}_{k=1}^M\}$, with $index(m_1) < \cdots < index(m_M)$ and $t(\jmath) = v(index(m_M))$ **do**
    Compute $\log \text{NFA}(\jmath)$ by using Equation (24);
    if $\log \text{NFA}(\jmath) \leq \log \epsilon$ **then**
        Accept the junction proposal $\jmath$: $Jlist \leftarrow \jmath$.
    end if
end for

---

4 Implementation

The goal of this section is to provide all necessary informations for the practical implementation of junction detection. First, the setting of parameters is addressed in Section 4.1. An optional refinement step to improve the accuracy of the detected branch directions is described in Section 4.2. A pre-selection of the junction candidates for speeding up the method is detailed in Section 4.3. Eventually, the complete detection algorithm pipeline is given in Section 4.4. A practical issue regarding the numerical computation of the NFA is also given in Appendix A.

4.1 Parameter choices

Recall that in the definition of discrete junctions, $K(r)$, the number of possible directions for junction branches as defined by Formula (3), was chosen as $K(r) = \lfloor 2\pi r \rfloor$, in order to have a precision of roughly one pixel along the circle of radius r. The choice of the precision $\Delta(r)$ relies on similar considerations. Visual experiments show that the perceived precision of an angle between two crossing segments in an image is better for long segments than short ones. Now, recall that $2\Delta(r)$ is the angle of a branch (or sector) in a junction at scale r. We consider that the length of the arc defined by this sector should be a constant w and should not depend on r. This length is exactly $2\Delta(r) \times r$, which implies that $\Delta(r)$ should be chosen as inversely proportional to r. In practice, we choose $w = 5$. Thus

$$2\Delta(r) = \frac{w}{r} = \frac{5}{r}. \tag{17}$$

Figure 9 illustrates the corresponding angular sectors for two different scales.

It follows that for a given order M, the number of tests $\#\mathcal{J}(M)$ can be computed as

$$\#\mathcal{J}(M) = \frac{N}{M!} \sum_{r=r_{min}}^{r_{max}} \prod_{m=1}^{M} \lfloor 2\pi r \rfloor \left(1 - \frac{5(m-1)}{r\pi}\right), \tag{18}$$

where N is the total number of pixels in the image. In the experimental section, the maximum order of junctions will be $M = 4$ and the smallest possible radius is $r_{min} = 3$ for all experiments. The maximal scale r_{max} is chosen as 5% of the diagonal of the image. The threshold on the minimum scale of detection is set to $r_{gap} = 12$ for all experiments in this paper.
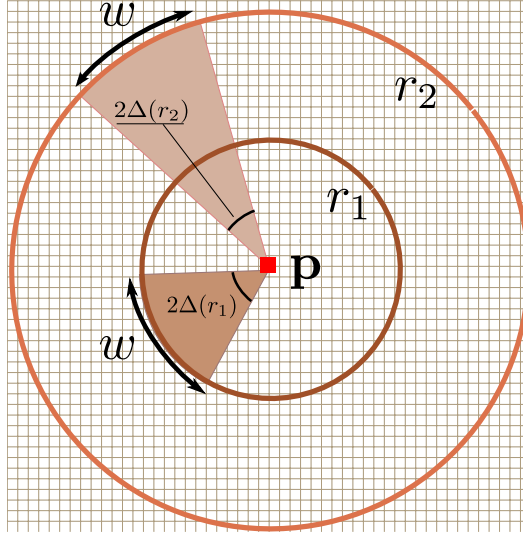

Figure 9: $\Delta(r) = \omega/r$ for two distinct r values. Observe that the length of the arc defined by a sector, w, is a constant and does not depend on r

4.2 Optional direction refinement

Since directions in a junction are bisectors of angular sectors (see Equation (4) for the definition of $S_{\mathbf{p}}(r, \theta)$) and since the set of possible directions is discrete, it may happen that the directions of some branches in a detected junction remain slightly imprecise. In the following, we describe a simple refinement in the computation of junction directions.

For a branch of direction θ centered at $\mathbf{p}$, a refined direction $\widehat{\theta}$ is computed as follows

$$\widehat{\theta} = \arctan \frac{\mathcal{O}_y}{\mathcal{O}_x}, \tag{19}$$

$$\text{with} \quad \mathcal{O}_x = \sum_{\mathbf{q} \in S_{\mathbf{p}}(r,\theta)} \gamma_{\mathbf{p}}(\mathbf{q}) \cos \psi_{\mathbf{q}};$$

$$\mathcal{O}_y = \sum_{\mathbf{q} \in S_{\mathbf{p}}(r,\theta)} \gamma_{\mathbf{p}}(\mathbf{q}) \sin \psi_{\mathbf{q}};$$

$$\text{and} \quad \psi_{\mathbf{q}} = \begin{cases} \phi_{\mathbf{q}} & \text{if } d_{2\pi}(\phi_{\mathbf{q}}, \theta) < \frac{\pi}{2} \\ \phi_{\mathbf{q}} + \pi & \text{otherwise.} \end{cases} \tag{20}$$

Notice that after this refinement, two branches in a given junction may intersect. If this happens, the detected junction is removed. The refinement process is described in Algorithm 6.

Algorithm 6 Direction refinement

Input: A junction $\jmath : (\mathbf{p}, r, \{\theta_k\}_{k=1}^M)$
1: (1) *Refinement of junction branches*:
2: **for** $k = 1$ to M **do**
3:     $\mathcal{O}_x = 0, \mathcal{O}_y = 0$;
4:     **for** $\mathbf{q} \in S_{\mathbf{p}}(r, \theta_k)$ **do**
5:         Compute $\gamma_{\mathbf{p}}(\mathbf{q})$ using Equation (6);
6:         Compute $\psi_{\mathbf{q}}$ using Equation (20);
7:     **end for**
8:     $\mathcal{O}_x \leftarrow \mathcal{O}_x + \sum_{\mathbf{q} \in S_{B_k}} \gamma_{\mathbf{p}}(\mathbf{q}) \cos \psi_{\mathbf{q}}$;
9:     $\mathcal{O}_y \leftarrow \mathcal{O}_y + \sum_{\mathbf{q} \in S_{B_k}} \gamma_{\mathbf{p}}(\mathbf{q}) \sin \psi_{\mathbf{q}}$;
10:     Update the branch direction $\theta_k = \arctan \frac{\mathcal{O}_y}{\mathcal{O}_x}$.
11: **end for**
12: (2) *Check that branches are still disjoint.*
13: **for** $m = 1$ to M **do**
14:     **for** $k = m + 1$ to M **do**
15:         **if** $d_{2\pi}(\theta_m, \theta_k) < \Delta(r)$ **then**
16:             Remove $\jmath$, then **break**;
17:         **end if**
18:     **end for**
19: **end for**

4.3 Speed up

In order to speed-up the algorithm, and following [51, 54, 53, 55], we propose to apply a pre-processing step to select potential junction candidates.

We take advantage of a fast segment detector, the Line Segment Detector (LSD) as introduced in [60] [1], whose complexity is linear in the size of the image. In order to make sure that we won't miss some junction candidates, the detection threshold λ of the LSD is set to a large value. The value $\lambda = 10^4$ has been used for all experiments in this paper. Once all possible line segments have been found, potential junction locations are restricted to a small neighborhood around each endpoint of those line segments. Figure 10 displays all line segments detected with $\lambda = 10^4$ for a given image and shows the corresponding junction candidates in red.

Table 2 shows the number of detections and the running time of the complete junction detection procedure, with and without use of the LSD preprocessing. The comparisons are implemented on three types of images: images containing both geometry and texture, images containing mostly textures

---
[1] The code of LSD can be downloaded from the IPOL website: http://www.ipol.im/pub/algo/gjmr_line_segment_detector/.

Figure 10: Junction candidates using LSD. **Left**: the input image; **Mid**: all line segments detected with $\lambda = 10^4$; **Right**: all junction candidates.

Figure 11: Test images for the speed up evaluation. **Top**: house, Lena and window; **Mid**: autumn, park and branches; **Right**: images of geometric paintings: *Geometric* by John Cooper, *Composition* by Charmion von Wiegand, and *Suprematism* by Kazimir Severinovich Malevich.

Table 2: Number of detections and running time when using or not the LSD pre-selection. The comparisons are implemented on three types of images: images with a strong geometrical content, images containing mostly textures and photographs of abstract paintings. Num (L/T/X) stands for the numbers of detected L/T/X junctions respectively. The experiments are implemented on a computer with a 2.0GHz Intel Core 2 Duo Processor and 2 GB RAM.

Image			Using LSD candidates		Using all candidates	
class	name	size	Num (L/T/X)	Time (s)	Num (L/T/X)	Time (s)
geometry	house	256×256	28/21/3	2.17	29/21/3	8.16
and texture	Lena	256×256	79/44/12	3.90	81/44/12	9.77
	window	768×576	120/224/51	10.36	122/230/52	64.72
	autumn	576×768	52/25/8	38.52	54/27/8	62.07
texture	park	576×768	97/98/15	37.20	100/100/15	62.18
	branches	536×819	78/100/18	35.97	80/101/19	62.08
	geometric	655×518	27/17/2	1.48	28/17/2	44.19
geometry	composition	413×300	3/60/36	1.94	3/62/37	17.46
	suprematism	400×640	40/33/5	1.57	42/34/5	34.66

and photographs of abstract paintings (see Figure 11). Using candidates from LSD clearly reduces the computing time, while the quantity of detections is not affected. Notice that this reduction strongly depends on image structures. The simpler the structures contained within the images, the larger the achieved reduction.

4.4   Algorithm pipeline

The pipeline of the whole a contrario junction detection algorithm is summarized in Algorithm 7. Remark that the parameters to be set in this algorithm are: the detection threshold ϵ, the maximum order of junctions M' and the radiuses $r_{min}, r_{max}, r_{gap}$ (see Section 4.1).

Algorithm 7 Complete junction detection procedure

Input: Image I, threshold ϵ, maximal order M', parameters $r_{min}, r_{max}, r_{gap}$.
  (1) Use LSD to preselect junction candidates $\mathcal{C}$;
  (2) Detect ϵ-meaningful junctions for all orders smaller than M' and for all scales between r_{min} and r_{max} by using Algorithm 2. Store junctions of order M in a list $Jlist(M)$.
  (3) For each $M \leq M'$, refine all junctions in $Jlist(M)$ by using Algorithm 6.
  (4) Keep only maximal junctions in the lists $Jlist(M)$ by using Algorithm 3 and 4.

In the rest of the paper, we refer to this algorithm by the acronym ACJ (for A Contrario Junction detection).

5   Experimental Analysis

This section gathers experiments illustrating the performances of the *a contrario* junction detection (ACJ). When possible, performances will be compared with the classical Harris detector [21] and the recent "*Pj on gPb*" method [49]. For Harris, the cornerness is defined as $det(H) - k \cdot Tr(H)^2$, where H is the Harris matrix and the default value $k = .04$ is chosen. For "*Pj on gPb*", the code kindly provided by M. Maire is used.

Section 5.1 illustrates the stability of the detection for different images. We then investigate the invariance properties of the method: scale and contrast changes are respectively studied in Sections 5.2 and 5.3. In particular, we show the ability of the method to accurately detect both the scale and the

position of a junction. Evaluation on a benchmark is given and discussed in Section 5.4 and some more results are displayed in Section 5.5.

5.1 Stability and choice of ϵ

One great quality of *a contrario* detection methods relies in the fact that the threshold ϵ has an intuitive meaning: it is an upper bound of the average number of false detections in an image following $\mathcal{H}_0$. This makes the setting of ϵ quite easy in practice: by default, and unless otherwise indicated, ϵ is set to 1 in all experiments. We checked that, on the average, less than one false detection occur in a Gaussian white noise image. This could be expected since such images almost follow the hypothesis $\mathcal{H}_0$.

Now, the real strength of the ACJ approach comes from the fact that the value $\epsilon = 1$ also yields very satisfying detections in natural images, whatever their content, size or resolution. In contrast, other approaches such as Harris or "*Pj on gPb*" tend to strongly over-detect in textured areas when using a fixed parameter, see Figure 12. This stability property is all the more interesting as it remains valid through scales. For a given choice of ϵ, Formula (13) yields different thresholds $t(r, \epsilon)$ that adapt to the scales of the junctions. As a consequence, by choosing $\epsilon = 1$, results are simultaneously satisfying at all scales. This would not be possible with a fixed threshold on junction strengths.
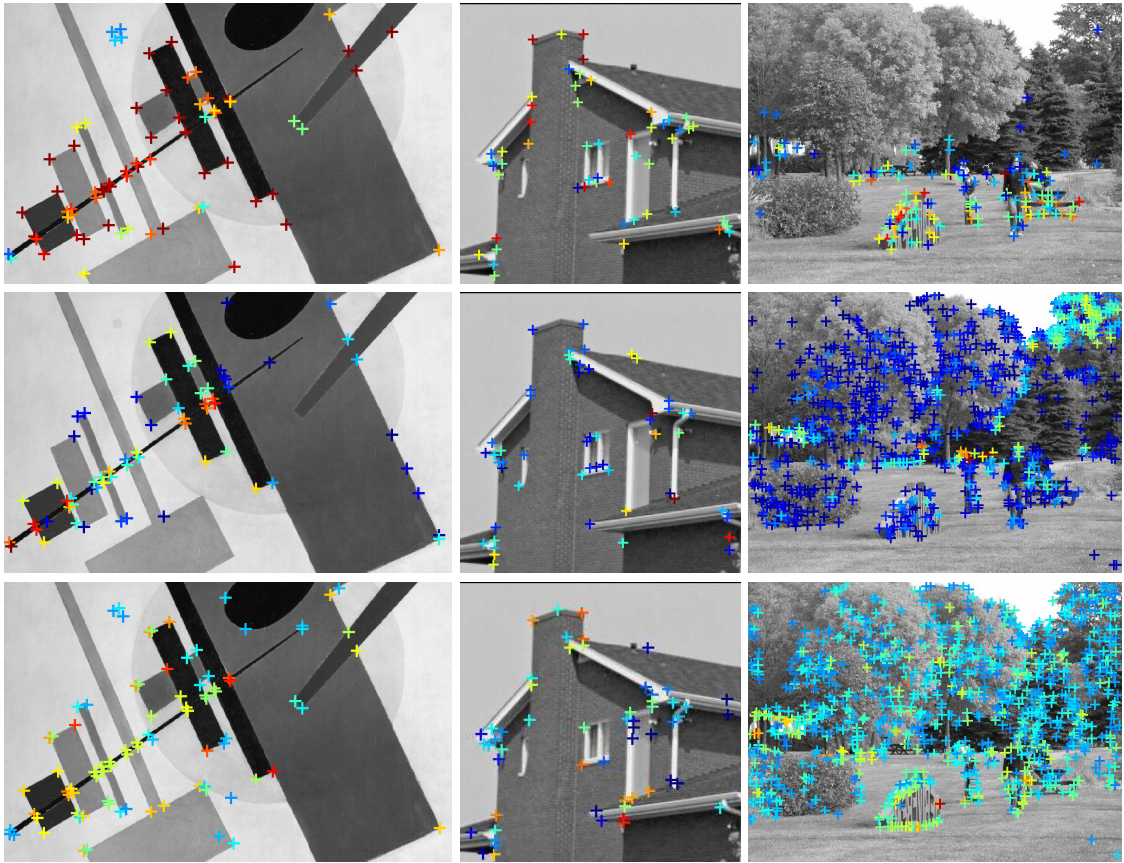

Figure 12: Junctions obtained by ACJ (top row), Harris (mid row) and "*Pj on gPb*" (bottom row). Parameters are fixed for the three images, yielding the same number of detections on the house image (this correspond to $\epsilon = 1$). Observe that only ACJ prevent from over-detection in textured areas. The color of the junction depends on the NFA value (red corresponds to small values, i.e. very meaningful junctions and blue corresponds to high values).

20

5.2 Scale Invariance

This section focus on the behavior of our approach with respect to scale changes. Recall that the scale of a junction is defined as the scale at which the junction is most meaningful and therefore strongly differs from classical approaches relying on the linear scale space. In Section 5.2.1, we show that such scales change linearly with the resolution of images, a clearly desirable property. In Section 5.2.2, we discuss the interpretation of these scales in images. Finally, Section 5.2.3 demonstrates that the ACJ approach has a better spatial precision through resolution changes than other classical approaches.

5.2.1 Scale and resolution

In order to investigate the coherence of detections through scale changes, we apply the proposed junction detection algorithm to a sequence of images with different resolutions. An original image I is resized with 8 different zoom factors $s_0 > \cdots > s_7$, using a bilinear interpolation. The set of factors is chosen as $\{1, 0.9, 0.8, 0.7, 0.6,$
$0.5, 0.4, 0.3\}$. Algorithm 7 is then applied to each image $s_i I$ of the sequence and yields a junction list $\mathcal{J}_{s_i}$. Then, each junction $\jmath_0$ of $\mathcal{J}_{s_0}$ is tracked through resolutions. For this purpose, we define an angular distance between two junctions:

$$\mathcal{S}(\jmath, \jmath_0) = \max_{\theta \in \{\theta_m\}_{m=1}^M} \min_{\theta' \in \{\theta_m^0\}_{m=1}^{M_0}} d_{2\pi}(\theta, \theta'). \tag{21}$$

A junction $\jmath_0$ in $\mathcal{J}_{s_0}$ is then matched with j_i in $\mathcal{J}_{s_i}$ if they have the same order M_0, if their centers are close enough, in the sense that $\|\mathbf{p} - s_i \cdot \mathbf{p}_0\|_2 < 3$, and if their angular distance $\mathcal{S}(\jmath_0, \jmath)$ is smaller than $\frac{\pi}{20}$. If several junctions in $\mathcal{J}_{s_i}$ satisfy these properties, $\jmath_0$ is matched with the one minimizing the angular distance $\mathcal{S}(\jmath_0, \jmath)$. If there is no junction in $\mathcal{J}_{s_i}$ satisfying these properties, no junction in $\mathcal{J}_{s_i}$ is matched with $\jmath_0$. For each junction $\jmath_0$ in the original image, we call *trajectory* the list of its corresponding junctions through the different resolutions.

Several such trajectories are shown on Figure 13 (b). In this experiment, we only consider junctions potentialy having complete trajectories in the scale space. Therefore, the maximal scale considered in the coarsest image is $r_{max} \times 0.3 = 30 \times 0.3 = 9$. The red curves show the scales of junctions along all the trajectories as functions of $s_i/0.3$. The baselines $\{y = r \cdot s_i\}$, where r changes from 1 to 90, are displayed in blue. These correspond to an ideal behavior with respect to resolution changes. Notice that the red curves remain close to the baselines: this implies that the scales of maximal meaningful junctions are quite robust to resolution changes. Figure 14 (a) gives some concrete examples of junctions detected by the ACJ algorithm along their trajectories. Once more, their scales increase linearly with resolution. To the best of our knowledge, the only approaches which permit to obtain a similar coherence between scale detection and resolution are those which rely on linear scale-spaces [63, 64]. However, it should be underlined that the use of linear scale-space inevitably leads to poor location precisions at large scales. This aspect will be further developed in Section 5.2.3. Figure 14 (b) illustrates the aforementioned dilemma (see Section 3.4) to detect or not a junction in the presence of nearby edges. In these examples, junctions are not detected at large scale because of the constraint r_{gap} (see Section 3.5). This behavior seems consistent with human perception: when several edges meet around a large gap, we probably disregard it as a junction.

In a second experiment, shown on Figure 13 (c), we compute the repeatability rate of detections with respect to image resolution. More precisely, if we note $\mathcal{J}_{s_0}(s_0 I \to s_i I)$ the list of junctions that are matched for all intermediate resolutions between $s_0 I$ and $s_i I$, the repeatability rate is defined as

$$R(s_i) = \frac{\#\mathcal{J}_{s_0}(s_0 I \to s_i I)}{\#\mathcal{J}_{s_0}}. \tag{22}$$
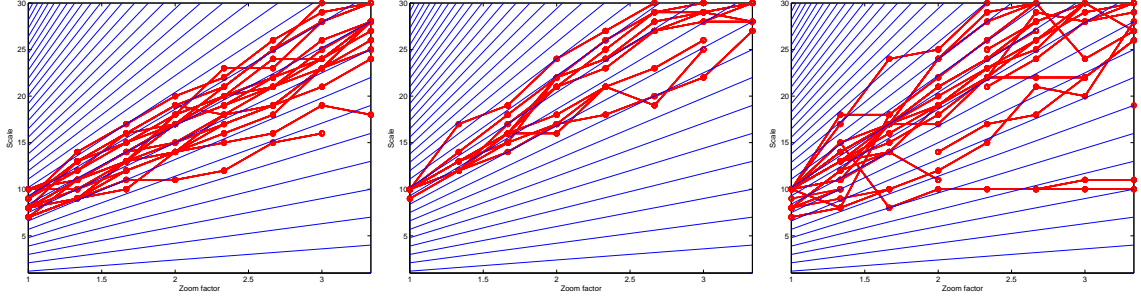
Observe that, for the three images, the repeatability rate always remains above 60%.

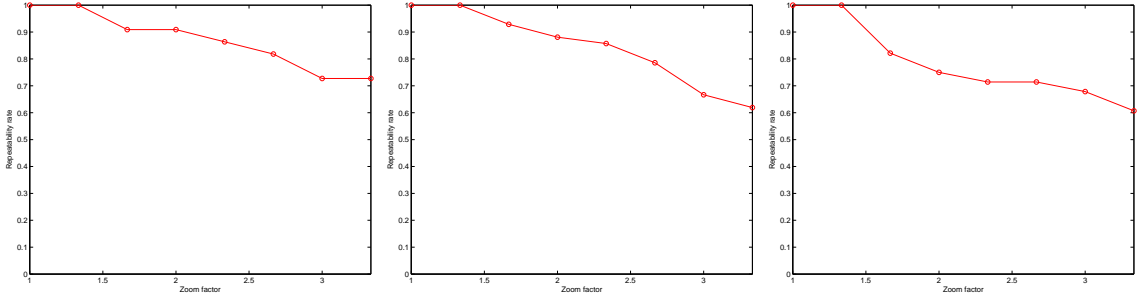### 5.2.2 How to interpret the detected scales ?

Most of the junction detection approaches in the literature do not provide characteristic scales for their junctions. One notable exception is the Harris-Laplace interest point detector, which makes
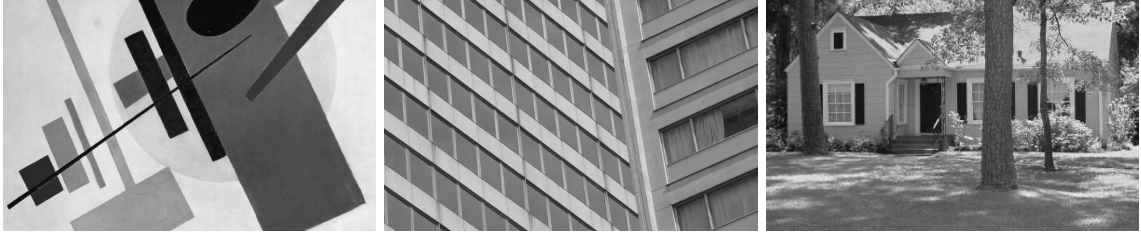
(a) original images

(b) junction scales as a function of the zoom factor s_i

(c) Repeatability rate as a function of the zoom factor s_i

Figure 13: Illustration of the scale invariance along the scale space. The first row shows the tested images. The second row shows the scales of junctions detected by ACJ along all the trajectories (curves in red) as a function of the zoom factor s_i (the abscissa is $s_i/0.3$). The baselines $\{y = r \cdot s_i\}$, where r changes from 1 to 90, are displayed in blue. The bottom row presents the repeatability rate of the junctions as a function of the zoom factor.

use of a linear scale space in order to detect keypoints at different scales. Detected points are those which maximize the Laplacian of Gaussian (LoG) in the scale-space and the Harris corner measure in a local space neighborhood. Some results of this detector on a synthetic and a real image are given in Figure 15 (b) and (d). Observe that the scales of junctions do not have a clear interpretation in these images. In contrast, the scales detected by ACJ (see Figure 15 (a) and (c)) arguably correspond to the optimal size at which one can observe the junction in the image. For instance, the scale of the L-junctions located at the corner of a rectangle is generally chosen as the length of the smaller side of the rectangle, see Figure 15 (a). More generally, we observed that *the scale of a junction usually corresponds to the length of its shortest branch*. The fact that the use of the NFA as a tool to select the most proheminent scale yields such simple geometric behavior is a strong asset of the proposed approach. The same coherence between the characteristic scales of junctions and the sizes of local structures can be observed in the examples of Figure 14.
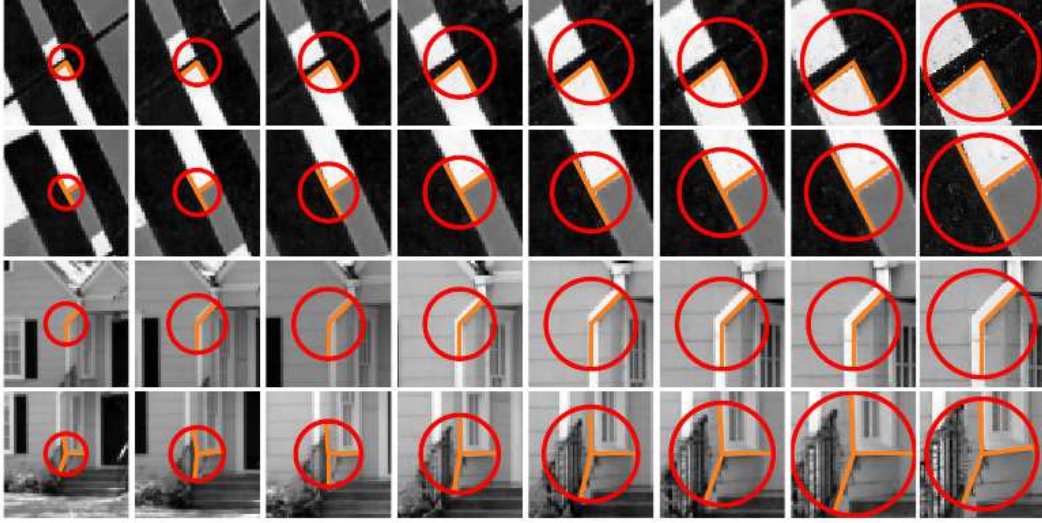
(a)

(b)

Figure 14: Examples of detected junctions along several junction trajectories: each row shows a list of junctions detected at the same relative locations in images, from the coarsest to the finest resolution. (a) In these examples, junctions can be followed along complete trajectories, and their scales remain roughly proportional to the image resolution. (b) These two examples illustrate the difficulty of junction detection in the presence of nearby edges.

5.2.3 Scale and spatial precision

We have shown that the ACJ approach permits to detect scales that behave correctly when images are zoomed. We now show that the location of the junction is accurate and stable when performing such resolution changes, in contrast to linear scale space approaches. We compare ACJ with two classical methods, Harris-Laplace and the "*Pj* on *gPb*" detector proposed by Maire *et al.* [49]. To this aim, we build in the same way as before a sequence of images composed of one image at full resolution (the one displayed on Figure 15 (a)) and 7 sub-sampled versions with zoom factors $\{0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3\}$. The original image is made of simple geometrical patterns, which permit to easily create a ground-truth of junctions. Gaussian noise and blur are added to the original image before creating the sequence. For a given image and a given method, the detected junctions are matched to the junctions of the ground truth in the same way as in Section 5.2.1. The average location error is then defined (for this image and this method) as the average squared Euclidean distance between detected junctions and their corresponding groundtruth junctions.

The average location errors of the different methods are shown in Figure 16 (b). We can see that ACJ has the smallest errors, followed by the Laplace-Harris detector and "*Pj* on *gPb*".

(a) scale of L-,Y- and X-junctions with our approach   (b) scale with Harris-Laplace

(c) scale of L-,Y- and X-junctions with our approach   (d) scale with Harris-Laplace
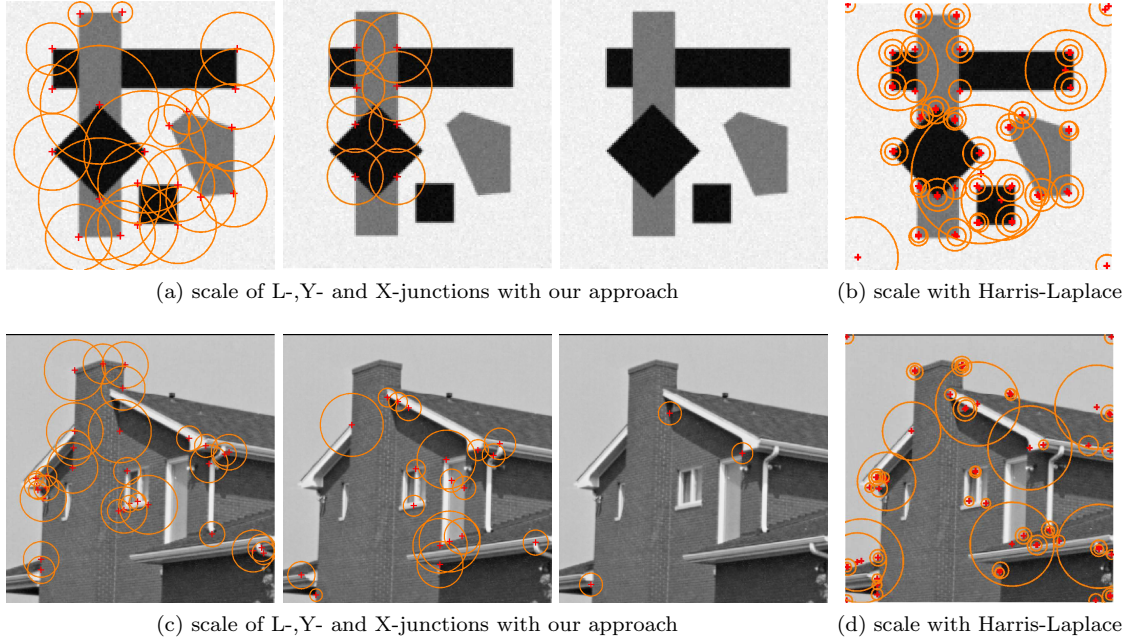
Figure 15: Scale selection. (a) shows the characteristic scale of L-, Y- and X-junctions given by the proposed approach, and (b) shows the characteristic scale of junctions given by Harris-Laplace on a synthetic image. (c) and (d) show the same comparison on the house image. The location of each junctions is indicated by a red cross and circles have a radius equal to the corresponding scale. Refer to the text for more explanations.
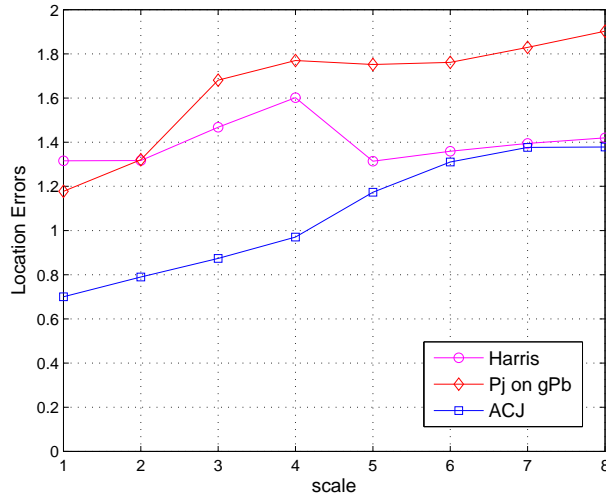

Figure 16: Average location error for different approaches: the proposed ACJ, "Pj on gPb" and the Harris detector. The comparison is led on a sequence of images with different resolutions, starting from the image displayed in Figure 15 (a).

5.3 Contrast Invariance

The goal of this section is to evaluate the robustness to contrast changes of different junction detection approaches. To this aim, we create a sequence of images by applying different gamma corrections to

an original image I, with γ in $\{\frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{2}{3}, 1, 1.5, 2, 2.5, 3, 4\}$. For each image in the sequence, we detect junctions with

- our ACJ approach, with $\epsilon = 1$,

- Harris corner detector, with a threshold 0.06 on local maxima,

- Maire's approach using "*Pj on gPb*" [49],

- a totally contrast invariant version of ACJ (denoted TACJ), obtained by removing the normalized gradient amplitudes in the definitions (5),(6) of the branch strength [2].

In order to evaluate the different results, we compute the repeatability rate of each method on the image sequence. More specifically, if $\mathcal{J}_0$ is the list of junctions in the original image ($\gamma = 1$), and $\mathcal{J}_i$ the list of junctions in the i-th image, we note $\mathcal{J}_0 \cap \mathcal{J}_i$ the set of junctions in $\mathcal{J}_i$ that can be matched with junctions in $\mathcal{J}_0$, where the matching criterion is the one defined in Section 5.2.1. Following the same protocol as in [32], the repeatability rate of the i-th image is then calculated as

$$Repeat(i) = \frac{\#\{\mathcal{J}_0 \cap \mathcal{J}_i\}}{\#\mathcal{J}_0}. \tag{23}$$

The curves of repeatability rates for the different methods are shown on Figure 17. We can see that the Harris detector has the worst performance with respect to contrast changes, which could be expected since it is the most contrast dependent. Maire's approach [49] gives better results, possibly because it relies on an edge detector that is tuned to match boundaries annotated by humans, which may somehow eliminates contrast variations. Both ACJ and TACJ perform better than the other ones, TACJ providing (as could be expected) slightly better results than ACJ. This could advocate for the choice of TACJ in practice. However, we observed that this totally contrast invariant approach appears to be less robust than ACJ to small image modifications, such as those created by JPEG compression. The choice of ACJ results in a compromise between contrast invariance requirements and robustness.
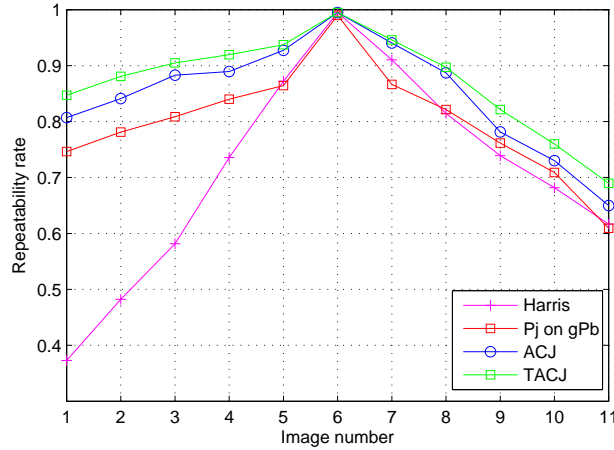

Figure 17: Repeatability rate of different approaches regarding contrast changes. The curves are averaged over 9 image sequences. Each sequence is obtained by applying different gamma corrections (as specified in the text) to a test image. The 9 test images are those presented in Figure 11.

---

[2]This boils down to replace the distribution of the strength at a pixel with $\mu = \frac{1}{2}\delta_0 + \frac{2}{\pi}\frac{1}{\sqrt{2-z^2}}H(z)dz$ in Equation (10). The resulting junction detection algorithm is similar to Algorithm 7.

5.4 Junction benchmark

A practical way to evaluate the consistency of our approach with human perception would be to rely on a human annotated junction database. To the best of our knowledge, such a database does not exist. Nevertheless, the well known Berkeley segmentation dataset[3] has been used by the authors of [49] in order to evaluate different junction detectors. In their paper, Maire et al. use the human annotated boundaries to create a ground truth of junctions in the following way: L-junctions are locations of high curvature along these boundaries and Y-junction are locations where more than three regions intersect.
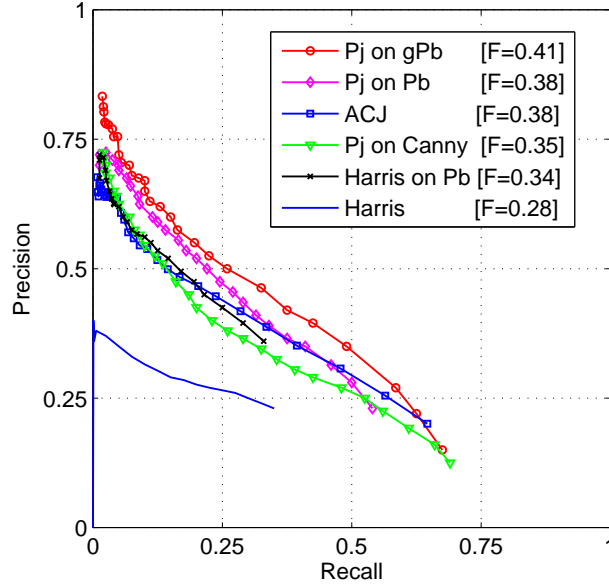
(a) Precision-Recall curves of different methods

Figure 18: The precision-recall curves of different methods on the BSDS-based junction benchmark.

As mentioned before, they also propose a procedure called Pj to detect junctions from any given set of boundaries. This procedure is applied to the results of different edge detectors, including their detectors Pb and gPb. The quality of the resulting junctions is assessed by comparing them to the previously created ground truth. We applied the same protocol to the ACJ method proposed in this paper. In order to overcome the strong compression artifacts of the Berkeley database, to which we are sensitive because of strong contrast robustness, we apply a small amount of blur and noise to images before proceeding. Figure 18 presents the results for several junction detection algorithms on the benchmark. The ACJ approach yields a performance of F= 0.38 [4], which is better than the baseline Harris detector (F=0.28) and Pj on $Canny$ (F=0.35) but remains below the performance of Pj on Pb (F=0.38) and Pj on gPb (F=0.41). The human agreement on the database yields F=0.47. The performance of our approach in comparison to Pj on gPb has to be tempered by two facts. First the junctions of the ground truth stem from a set of human annotated boundary, and the junctions detected by Pj stem from a computer segmentation. Therefore, both are boundary-based. It is likely that any detector using a more local junction definition (including ACJ or even Harris) will be penalized in the benchmark. As explained in the introduction and shown by psychophysical studies [4], both local and global cues are at play for the perception of junctions. Second, and most important, the parameters of Pb and gPb are optimized in order to match human detected edges on the same database, which introduces a bias in favor of these methods.

---

[3]The Berkeley segmentation dataset can be downloaded from www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.

[4] F is defined as the highest value of the quantity $2\frac{Precision \cdot Recall}{Precision + Recall}$ along the curve.

5.5 More Results

This last section aims at illustrating the proposed approach with several more visual experiments. Figure 19 presents all the junction detected on a synthetic image. We can observe that the junctions are found with high accuracy and are mostly correctly classified. Figures 20 and 21 present the results on two natural images. Observe that junctions are accurately characterized through their type, localization and scale. Again, we emphasize that the proposed ACJ algorithm yields very litlle detections in textured areas, see Figure 21.
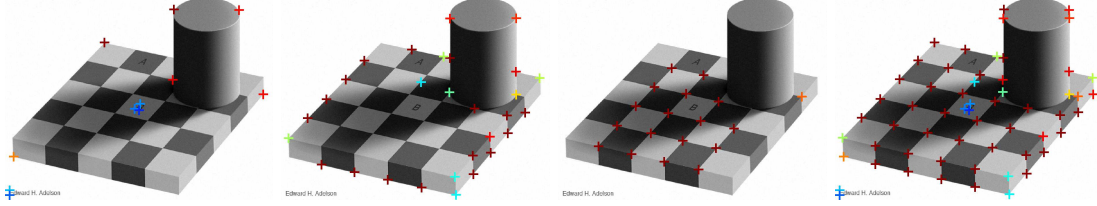

Figure 19: Some results of the proposed ACJ approach, with $\epsilon = 1$. From left to right: L-, Y- and X- junctions. On this simple synthetic image (from E. H. Adelson), junctions are well detected and classified. The color of the junction depends on the NFA value (red corresponds to small values, i.e. very meaningful junctions and blue corresponds to high values).

6 Conclusion and perspectives

In this work, we have introduced a generic and principled approach for the detection and characterization of junctions in digital images. The proposed procedure is able to inhibit junction detection in textured areas, in contrast with classical approaches. Junctions are accurately characterized through their type, localization and scale. In particular, the method does not rely on the linear scale-space for scale computation, permitting geometric precision.

This works opens several perspectives. First, the accuracy of junction characterization is of importance for depth recovery or motion interpretation. The characterization of junction's type, the difference between T- and Y- junctions or their precise localization claim to be tested in the framework of such applications. Second, and as advocated in the introduction, several psychophysical studies show that the perception of junctions relies on both local and global cues. Therefore the proposed, local, procedure for junction detection could benefit from more global image analysis schemes. The modeling of interactions between junction detection and segmentation procedures is far from trivial, but similar principled approaches could be applied to junction definitions building from color and textured-based region analysis. In the opposite direction, segmentation methods can benefit from meaningful junction detections. Preliminary tests show that the boundary saliency, as for instance defined in [65], can benefit from a meaningful detection of junctions, in particular by solving ambiguities in boundary connections near these junctions.

A Appendix: Computing NFA in practice

The NFA of a junction $\jmath : \left\{ \mathbf{p}, r, \{\theta_m\}_{m=1}^M \right\}$ has been defined as $\mathrm{NFA}(\jmath) := \#\mathcal{J}(M) \cdot F_\jmath(t(\jmath))$, where

$$F_\jmath(t) = \prod_{m=1}^{M} \int_t^{+\infty} \overset{J(r,\theta_m)}{\underset{j=1}{\star}} \mu(dz),$$

with

$$\mu(z) = \frac{1}{2}\delta_0(z) + H(z) \cdot \frac{1}{\sqrt{\pi}} e^{-\frac{z^2}{4}} \cdot \mathrm{erfc}(\frac{z}{2})dz.$$

27

(a) original image

(b) L-junctions

(c) Y-junctions
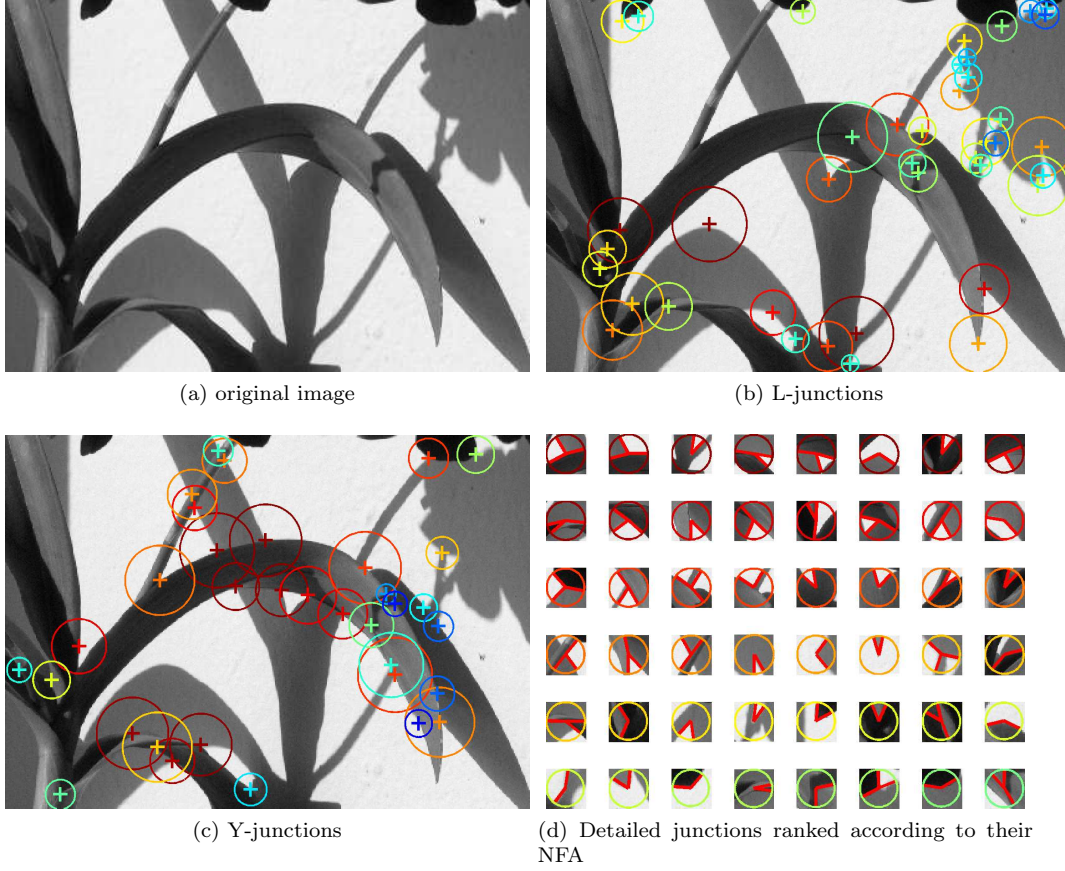
(d) Detailed junctions ranked according to their NFA

Figure 20: Specific junction detections on a natural image. The color of the junction depends on the NFA value (red corresponds to small values, i.e. very meaningful junctions and blue corresponds to high values).

In practice, the numerical values taken by $F_\jmath(t(\jmath))$ can become smaller than the precision of the computer when the strength $t(\jmath)$ is too high. For this reason, we write the distribution μ as

$$\mu(z) = \frac{1}{2}\delta_0(z) + \frac{1}{2}R(z)dz,$$

where

$$R(z) = 2H(z) \cdot \frac{1}{\sqrt{\pi}}e^{-\frac{z^2}{4}} \cdot \operatorname{erfc}(\frac{z}{2}).$$

When this distribution is convolved k times with itself, it becomes

$$\underset{j=1}{\overset{k}{\star}}\mu = \left(\frac{1}{2}\right)^k \cdot \left(\delta_0 + \sum_{j=1}^{k}\binom{k}{j}(\overset{j}{\star}R)\right).$$

By integrating this function between $t > 0$ and $+\infty$, we obtain

$$\int_t^\infty \underset{j=1}{\overset{k}{\star}}\mu = \left(\frac{1}{2}\right)^k \cdot \sum_{j=1}^{k}\binom{k}{j}\int_t^{+\infty}(\overset{j}{\star}R).$$

28

(a) original image

(b) all junctions

(c) L-junctions

(d) Y-junctions

(e) X-junctions

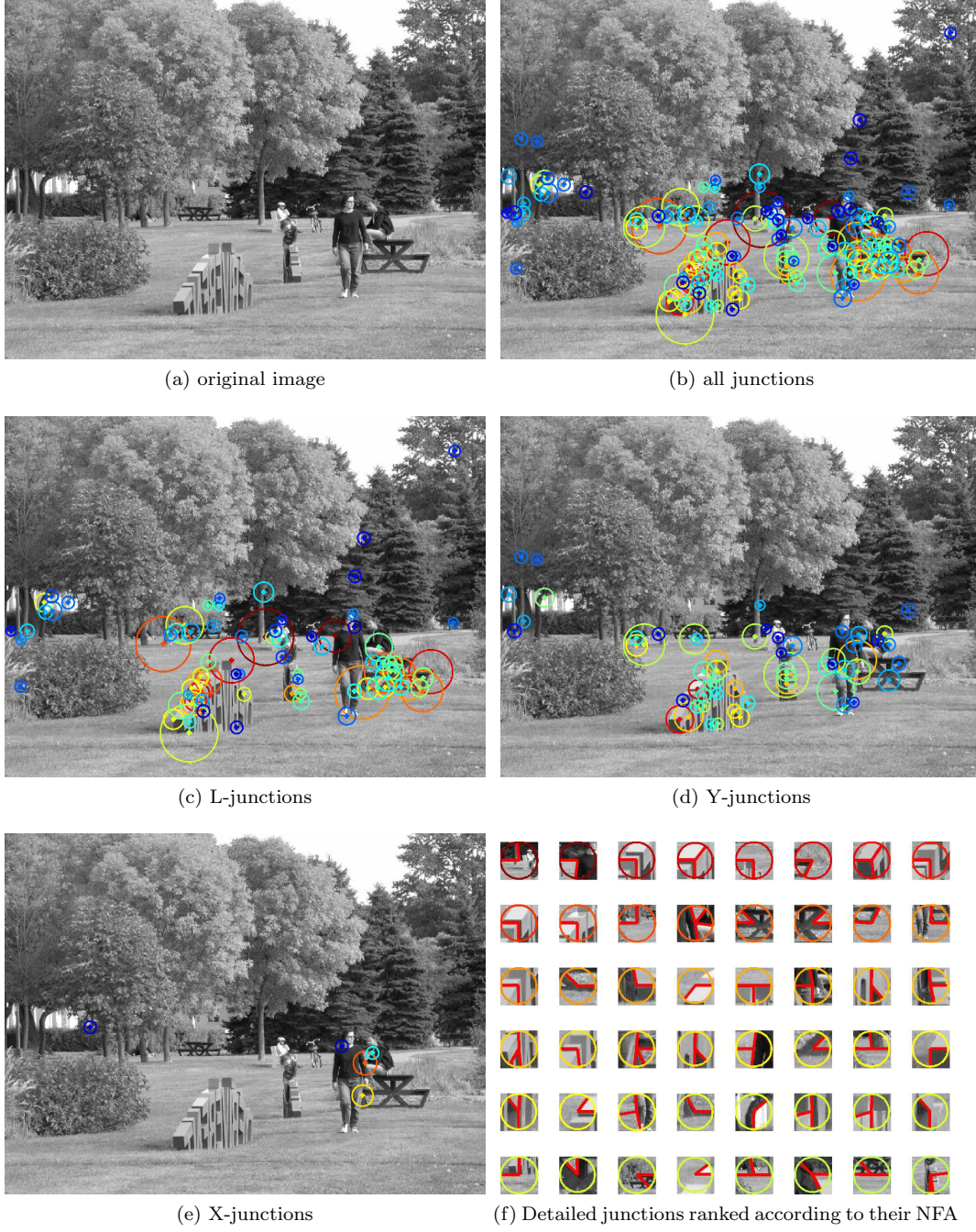(f) Detailed junctions ranked according to their NFA

Figure 21: Specific junction detections on a natural image. The color of the junction depends on the NFA value (red corresponds to small values, i.e. very meaningful junctions and blue corresponds to high values).

Since $\left(\frac{1}{2}\right)^k$ and the integral are both very small, we compute instead

$$\log \int_t^\infty \overset{k}{\underset{j=1}{\star}} \mu = -k \log(2) + \log \left(\sum_{j=1}^k \binom{k}{j} \int_t^{+\infty} (\overset{j}{\star} R) \right).$$

Finally,

$$\log \mathrm{NFA}(\jmath) = \log(\#\mathcal{J}(M)) - \log 2 \sum_{m=1}^{M} J(r, \theta_m)$$
$$+ \sum_{m=1}^{M} \log \left(\sum_{j=1}^{J(r,\theta_m)} \binom{J(r, \theta_m)}{j} \int_{t(\jmath)}^{+\infty} (\overset{j}{\star} R) \right). \tag{24}$$

This formula is used in practice and compared to $\log(\epsilon)$ in order to select ϵ-meaningful junctions.

References

[1] D. Marr, *Vision. A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman and Company, 1982.

[2] C.-E. Guo, S.-C. Zhu, and Y. N. Wu, "Primal sketch: Integrating structure and texture," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 5 – 19, 2007, special issue on Generative Model Based Vision.

[3] G. Kanizsa, *Organization in vision : essays on gestalt perception.* Praeger, New York, 1979.

[4] N. Rubin, "The role of junctions in surface completion and contour matching," *Perception*, vol. 3, no. 30, pp. 339–366, 2001.

[5] H. Wallach, "Ueber visuell whargenommene bewegungrichtung," *Psychologische Forschung*, 1935.

[6] F. Metelli, "The perception of transparency," *Sci. Am.*, vol. 4, no. 230, pp. 90–98, 1974.

[7] E. H. Adelson, "Lightness perception and lightness illusions," in *M. Gazzaniga, ed., The New Cognitive Neurosciences, 2nd ed.*, 2000, pp. 339–351.

[8] D. Geiger, K. Kumaran, and L. Parida, "Visual organization for figure/ground separation," in *Proc. of Computer Vision and Pattern Recognition*, vol. 0, 1996, pp. 155–161.

[9] X. Ren, C. C. Fowlkes, and J. Malik, "Figure/ground assignment in natural images," in *Proc. of European Conference on Computer Vision*, vol. 2, 2006, pp. 614–627.

[10] I. Leichter and M. Lindenbaum, "Boundary ownership by lifting to 2.1d," in *Computer Vision, 2009 IEEE 12th International Conference on*, 29 2009-oct. 2 2009, pp. 9 –16.

[11] M. Dimiccoli and P. Salembier, "Exploiting t-junctions for depth segregation in single images," in *Proc. of ICASSP*, 2009, pp. 1229–1232.

[12] C. Fuchs and W. Forstner, "Polymorphic grouping for image segmentation," in *Proc. of International Conference on Computer Vision*, 1995, pp. 175–182.

[13] T. Lindeberg and M.-X. Li, "Segmentation and classification of edges using minimum description length approximation and complementary junction cues," *Comput. Vis. Image Underst.*, vol. 67, no. 1, pp. 88–98, 1997.

[14] H. Ishikawa and D. Geiger, "Segmentation by grouping junctions," in *Proc. of Computer Vision and Pattern Recognition*, vol. 0, 1998, p. 125.

[15] A. Guzmán, "Decomposition of a visual scene into three-dimensional bodies," in *Proc. of Fall Joint Computer Conference, part I*, 1968, pp. 291–304.

[16] R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible objects: The local-feature-focus method," *Int. J. Robotics Research*, vol. 1, no. 3, pp. 57–82, 1982.

[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 *Proc. of International Conference on Computer Vision*, 2007.

[18] J. McDermott, "Psychophysics with junctions in real images," *Perception*, vol. 33, pp. 1101–1127, 2004.

[19] A. Desolneux, L. Moisan, and J.-M. Morel, "Meaningful alignments," *Int. J. Comput. Vision*, vol. 40, no. 1, pp. 7–23, 2000.

[20] H. P. Moravec, "Towards automatic visual obstacle avoidance," in *Proc. of IJCAI*, 1977, p. 584.

[21] C. Harris and M. Stephens, "A combined corner and edge detection," in *Proc. of The Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[22] W. Forstner, "A feature based correspondence algorithm for image matching," in *ISP Comm. III*, 1986.

[23] J. Shi and C. Tomasi, "Good features to track," in *Proc. of Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[24] P. Kovesi, "Phase congruency detects corners and edges," in *Proc. of DICTA*, 2003, pp. 309–318.

[25] C. S. Kenney, M. Zuliani, and B. S. Manjunath, "An axiomatic approach to corner detection," in *Proc. of Computer Vision and Pattern Recognition*, 2005, pp. 191–197.

[26] S. M. Smith and J. M. Brady, "SUSAN: A new approach to low level image processing," *Proc. of International Journal of Computer Vision*, vol. 23, pp. 45–78, 1995.

[27] M. Trajkovic and M. Hedley, "Fast corner detection," *Image and Vision Computing*, vol. 16, no. 2, pp. 75–87, 1998.

[28] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking." in *Proc. of International Conference on Computer Vision*, vol. 2, 2005, pp. 1508–1511.

[29] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105 –119, 2010.

[30] R. Deriche and G. Giraudon, "A computational approach for corner and vertex detection," *Int. J. Comput. Vision*, vol. 10, no. 2, pp. 101–124, 1993.

[31] T. Lindeberg, "Junction detection with automatic selection of detection scales and localization scales," in *Proc. of International Conference on Image Processing*, 1994, pp. 924–928.

[32] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vision*, vol. 37, no. 2, pp. 151–172, 2000.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[34] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. Comput. Vision*, vol. 60, pp. 63–86, 2004.

[35] L. Alvarez and F. Morales, "Affine morphological multiscale analysis of corners and multiple junctions," *Int. J. Comput. Vision*, vol. 25, no. 2, pp. 95–107, 1997.

[36] A. Rosenfeld and E. Johnston, "Angle detection on digital curves," *IEEE Trans. Computer*, vol. 22, no. 9, pp. 875–878, 1973.

[37] C. H. Teh and R. T. Chin, "On the detection of dominant points on digital curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 8, pp. 859–872, 1989.

[38] A. Rattarangsi and R. T. Chin, "Scale-based detection of corners of planar curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 430–449, 1992.

[39] H. Wang and M. Brady, "Real-time corner detection algorithm for motion estimation," *Image and Vision Computing*, vol. 13, no. 9, pp. 695 – 703, 1995.

[40] F. Mokhtarian and R. Suomela, "Robust image corner detection through curvature scale space," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 1376–1381, 1998.

[41] A. Mohammad and G. Lu, "An improved curvature scale-space corner detector and a robust corner matching approach for transformed image identification," *IEEE Trans. Image Processing*, vol. 17, no. 12, pp. 2425–2441, 2008.

[42] D. J. Beymer, "Finding junctions using the image gradient," in *Proc. of Computer Vision and Pattern Recognition*, 1991, pp. 720–721.

[43] W. T. Freeman, "Steerable filters and local analysis of image structure," Ph.D. dissertation, MIT, Cambridge, MA, USA, 1992.

[44] K. Rohr, "Recognizing corners by fitting parametric models," *Int. J. Comput. Vision*, vol. 9, no. 3, pp. 213–230, 1992.

[45] P. Perona, "Steerable-scalable kernels for edge detection and junction analysis," *Image and Vision Computing*, vol. 10, no. 10, pp. 663 – 672, 1992.

[46] V. Caselles, B. Coll, and J.-M. Morel, "Junction detection and filtering: A morphological approach," in *Proc. of International Conference on Image Processing*, 1996, pp. I: 493–496.

[47] F. Cao, "Good continuations in digital image level lines," in *Proc. of International Conference on Computer Vision*, 2003, pp. 440–446.

[48] T. F. Wu, G.-S. Xia, and S.-C. Zhu, "Compositional boosting for computing hierarchical image structures," in *Proc. of Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[49] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," in *Proc. of Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[50] R. Deriche and T. Blaszka, "Recovering and characterizing image features using an efficient model based approach," in *Proc. of Computer Vision and Pattern Recognition*, 1993, pp. 530 –535.

[51] L. Parida, D. Geiger, and R. A. Hummel, "Junctions: Detection, classification, and reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 7, pp. 687–698, 1998.

[52] M. A. Ruzon and C. Tomasi, "Edge, junction, and corner detection using color distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1281–1295, 2001.

[53] M. A. Cazorla and F. Escolano, "Two Bayesian methods for junction classification," *IEEE Trans. Image Processing*, vol. 12, no. 3, pp. 317–327, 2003.

[54] M. A. Cazorla, F. Escolano, D. Gallardo, and R. Rizo, "Junction detection and grouping with probabilistic edge models and Bayesian A*," *Pattern Recognition*, vol. 35, no. 9, pp. 1869–1881, September 2002.

[55] E. D. Sinzinger, "A model-based approach to junction detection using radial energy," *Pattern Recognition*, vol. 41, no. 2, pp. 494 – 505, 2008.

[56] L. Kitchen and A. Rosenfeld, "Non-maximum suppression of gradient magnitudes makes them easier to threshold," *Pattern Recognition Letters*, vol. 1, no. 2, pp. 93 – 94, 1982.

[57] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. of International Conference on Pattern Recognition*, 2006, pp. 850–855.

[58] L. Moisan and B. Stival, "A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix," *Int. J. Comput. Vision*, vol. 57, pp. 201–218, 2004.

[59] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J.-M. Morel, "An a contrario decision method for shape element recognition," *Int. J. Comput. Vision*, vol. 69, no. 3, pp. 295–315, 2006.

[60] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 722–732, 2010.

[61] A. Desolneux, L. Moisan, and J.-M. Morel, *From Gestalt Theory to Image Analysis: A Probabilistic Approach.* Springer Publishing Company, Incorporated, 2008.

[62] D. L. Ruderman and W. Bialek, "Statistics of natural images: Scaling in the woods," *Physical Review Letters*, vol. 73, no. 6, pp. 814–817, 1994.

[63] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. Comput. Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[64] B. Luo, J.-F. Aujol, Y. Gousseau, S. Ladjal, and H. Maître, "Resolution-independent characteristic scale dedicated to satellite images," *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2503–2514, 2007.

[65] R. Golubchyck and M. Lindenbaum, "Improving the saliency algorithm by cue optimization," Technion, Tech. Rep. MSC-2006-07, 2006.