Depth from occlusion: A region-merging approach

Mariella Dimiccoli, *Member, IEEE,* and Philippe Salembier, *Senior member, IEEE*

***Abstract*—This paper addresses the problem of estimating relative depth in single images by relying solely on the depth cue of occlusion. Occlusion is one of the major consequences of the physical image generation process: it occurs when an opaque object partly obscures the view of another object further away from the viewpoint. In this case, the contours of the objects in occlusion intersect in the image plane forming T-shaped junctions. The geometrical configuration of T-junctions (TJs) encodes the relative depths of the objects in partial occlusion. However, TJs may also arise from a reflectance discontinuity or, in an automatic framework, they may correspond to a false detection. In these cases, the local depth interpretation of TJs does not mirror the global depth interpretation. As a consequence, an algorithm aiming to estimate relative depth in single images by relying solely on the depth cue of occlusion should not only globally integrate the local depth information arisen from TJs but must also envisage a mechanism to deal with conflicting situations. The strategy proposed in this paper consists in first detecting TJs, then in segmenting the image preserving the TJs previously detected, and finally in depth ordering the regions of the final partition by relying on the depth information provided by TJs. The global depth ordering is achieved through a graph formalization, which allows to easily detect and solve possible conflicting interpretations. Experimental results demonstrate that the proposed approach gives a correct depth interpretations of a variety of real images.**

***Index Terms*—Image segmentation, T-junctions, occlusion, depth estimation, Binary Partition Tree, monocular depth.**

I. Introduction

THIS paper focuses on the problem of estimating depth ordering information from a single still image, a key issue in image understanding, that in recent years has focused the interest of the community. Motivation behind this tendency is provided by the increasing number of applications that could benefit or became feasible with advances in the field such as automatic foreground object removal, interactive depth-based image editing, 3D display of conventional 2D images. Until recently, most of the works on monocular depth estimation have been motivated by theoretical reasons, mainly aiming to understand the computational mechanism underlying the perception of depth in single images and, as a consequence, they have been tested only on simple synthetic images [1]–[7] or on real images previously segmented by interactive methods [8], [9]. Therefore, the extension of their applicability field to natural images is not straightforward and in most cases impossible.

During the last few years, the problem of recovering depth in single real images has been addressed by learning-based methods [10]–[14], which in general rely on strong assumptions on the image structure [11], [13] or on the image content [14] and in all cases do not produce accurate occlusion boundaries [10]–[14].

Standing in stark contrast to state-of-the-art approaches, this paper proposes a general low-level approach to the problem of estimating depth ordering information from a single still image, in which the depth ordering is directly inferred from the global interaction of local occlusion relationships, without relying on any previously learned information about the structure of the world nor on any assumption on the image structure.

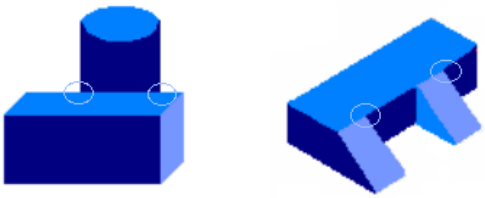
The choice of relying only on the depth cue of occlusion come from the fact that occlusion is pervasive in natural images. In fact, objects spatially separated in the 3D world might interfere with each other in the projected 2D plane and each of them obscures part of the ground. In particular, when an opaque object partly obscures the view of another object further away from the viewpoint, the projection of the object boundaries partially hiding each other creates T-shaped junctions in the image plane. Therefore, TJs represent one of the most primitive trace of the physical image generation process.

Although the importance of TJs in determining how objects and surfaces interact in the scene from their 2D projection has been emphasized by Gestalt psychologists [15] and further investigated in human vision, their role is often downplayed in practical applications. Their usefulness in extracting non-trivial information about 3D scenes has been recently demonstrated in a variety of applications such as stereo vision, multiview geometry and video segmentation [16]–[18], which rely on multiples images. Nevertheless, the potential of TJs in the single image scenario received little attention until now. This is due partly to the lack of robustness of T-junction detection without relying on redundant information in space or time, and partly to the ambiguity of their depth interpretations. In fact, whereas all instances of occlusion produce a T-junction [19] (Fig. 1 (a)), the converse is not true. For instance, the TJs in Fig. 1(b) are likely the result of a reflectance discontinuity and not of an occlusion. In this case, the local interpretation of TJs is not consistent with the global depth interpretation. In addition, in an automatic framework, false positive detections of TJs may also be the cause of a misleading local depth assignment.

As a consequence, an algorithm that uses only the cue of occlusion to infer a global depth ordering must envisage a mechanism to solve possible conflicting interpretations. Another issue that needs to be addressed is that depth information provided by TJs is limited to a small neighborhood of the T-junction centers. Therefore, to reason globally about depth relationships, the local depth information has to be somehow

M.Dimiccoli is with the Laboratory of Physiology of Perception and Action, CNRS-Collège de France, Paris, FRANCE, e-mail: (maria.dimiccoli@college-de-france.fr).

P. Salembier is with the Department of Signal Theory and Communications, Technical University of Catalonia (UPC), Barcelona, SPAIN, e-mail: (philippe.salembier@upc.edu).



(a) (b)

Fig. 1: (a) TJs corresponding to depth discontinuities. (b) TJs corresponding to reflectance discontinuities.

propagated globally.

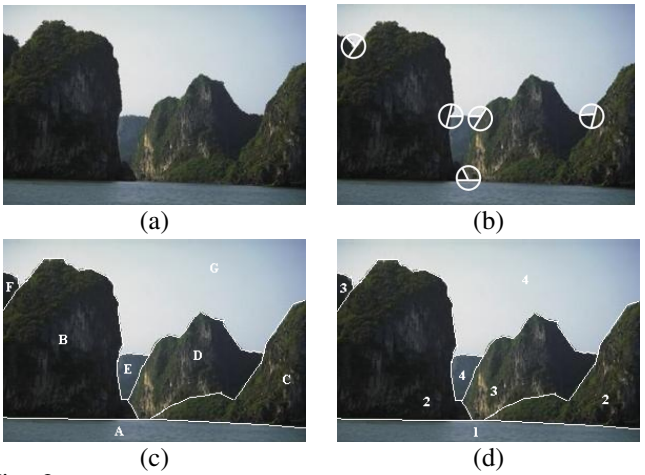
To convince us that TJs, and in turn occlusion, can be used to successfully assign depth in images, let us consider the image in Fig. 2 (a): it contains a certain number of TJs (see Fig. 2 (b)). In a local neighborhood of each T-junction, the region delimited by the roof appears to be in front to the regions delimited by the stem. By extending the branch of each T-junction following the depth discontinuity, which corresponds to the color discontinuity, the image can be partitioned in a limited number of regions (see Fig. 2 (c)). Due to the presence of TJs, the region A appears to be in front of the regions B, D and C; the region B appears to be in front of the regions F, E and G; region D appears to be in front of the regions E and G; finally, region C appears to be in front of the region D and G. By assigning depth locally and by detecting and solving possible conflicting interpretations, a global depth interpretation can be inferred (see Fig. 2 (d)).

Taking into account the above mentioned issues, we have developed an automatic algorithm that uses only the cue of occlusion to infer global, consistent depth ordering [**?**], [20]. Our strategy involves three main steps. First, occlusion relations signaled by TJs are detected; second, the image is segmented by using a BPT-based statistical region-merging algorithm which preserves the previously detected TJs; third, the depth relations between the regions of the final partition are encoded through a Directed Graph (DG). This formalization allows to easily detect and solve possible conflicting interpretations leading to a global depth ordering.

The organization of this paper is as follows. Before detailing each of the above mentioned steps, section II introduces the Binary Partition Tree (BPT), which constitutes the basic tool of the proposed region merging-based framework. The following three sections are devoted respectively to the detection of occlusion (III), to the segmentation preserving TJs (IV) and to the global depth ordering through a graph formalization (V). Section VI discusses experimental results. Finally, section VII summarizes the proposed region-merging approach, discusses limitations and proposes possible future lines of research.

II. Binary Partition Tree

A BPT [21] is a structured representation of a set of hierarchical partitions in which the finest level of detail is given by the initial partition. The set of the regions of the initial partition may coincide with the set of image pixels



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

Fig. 2: (a) An image of a natural scene. (b) T-junction branches are marked in white. (c) Result of extending the T-junction branches following the color discontinuities. (d) Smaller numbers correspond to regions closer to the viewpoint.

or the partition of flat zones, or any other set of regions obtained using any other pre-computed partition. The nodes of the tree are associated to regions that represent the union of two children regions and the root node represents the entire image support. Starting from the initial partition of all image pixels, pairs of neighboring regions are iteratively merged until a termination criterion (usually the number of regions of the final partition) is reached. The general region merging algorithm used for creating a BPT requires the specification of the *region model*, *merging order*, and *merging criterion*.

- *Region model*: the region model defines how to represent each region. When two regions are merged, the region model defines how to model its union and what are the main characteristics that should be kept in order to continue the merging process. For instance, if objects are assumed to be homogeneous in color, a region model based on the mean color of each region may be used or, if objects are assumed to be generated by the same probability distribution, a region model based on the color histogram of each region may be used.
- *Merging order*: the merging order determines the order in which pairs of neighboring regions (links) are processed. It is a real valued function of each pair of neighboring regions and is usually based on a similarity criterion between the region models. Each time a link is processed its associated nodes, which correspond to the neighboring regions, are merged together. The merging order is closely related to region model. As the region model, the merging order is related to the notion of objects, that is to the notion of homogeneity with respect to a defined property. It can be seen as a measure of the likelihood that two neighboring regions belong to the same object. For

instance, if objects are assumed to be homogeneous in color, a similarity measure based on the color difference should be used or, if objects are assumed to be generated by the same probability distribution, a similarity measure based on the color histogram should be used.

- *Merging criterion*: each time a link is processed, the merging criterion decides if the merging has actually to be done or not. It is a binary valued function (”merge” or ”do-not-merge”) of each pair of neighboring regions. The merging criterion allows to decide which of the mergings proposed by the merging order should be really done. An example of simple merging criterion is the number of regions. This merging criterion does not modify the order (proposed by the merging order) in which links are processed, but simply acts as a termination criterion. Instead, more complex merging criteria may be used to control more precisely the way regions are merged, acting as a sieve among the set of mergings proposed by the merging order. In section IV, depth information will be used to prevent the merging of regions that belong to different depth planes.

In [21], the regions are modeled deterministically by their mean color value and the order in which regions are merged is determined by a similarity measure based on color difference between the region models. Recently, Calderero and Marques [22], have presented a new region model based on color histogram and a new family of statistical similarity measures between the region models based on information theory. The authors proposed two different statistical merging orders: the Kullback-Leibler (KL) merging order and the Batthacharyya (BHAT) merging order. The KL merging order is based on measuring the similarity between the probability distributions of the regions and the probability distribution of their merging, weighted (or not) by the size of the regions.

KL area-weighted merging order:

$$f(R_a, R_b) = -n_i \cdot D_{KL}(P_a \parallel P_{a\cup b}) - n_j \cdot D_{KL}(P_{\parallel} P_{i\cup j}), \quad (1)$$

where R_a and R_b are two adjacent regions (written as (R_a, R_b), with size n_a and n_b and empirical distribution P_a and P_b respectively, whose union would generate a new region, $a \cup b$, with empirical distribution

$$P_{a\cup b} = \frac{n_a}{n_a + n_b} P_a + \frac{n_a}{n_a + n_b} P_b. \quad (2)$$

and

$$D_{KL}(P_a \parallel P_{a\cup b}) = P_a log \frac{P_a}{P_{a\cup b}} \quad (3)$$

is the Kullback-Leibler divergence operator [23] between the statistical distributions P_a and $P_{a\cup b}$.

The Batthacharyya (BATH) merging order is based on a size-weighted direct statistical measure of the probability distributions. This criterion leads to merge pairs of adjacent regions with the maximum probability of fusion.

BHAT area-weighted merging order:

$$f(R_a, R_b) = \arg \max_{R_a \backsim R_b} -min(n_a, n_b) \cdot B(P_a, P_b), \quad (4)$$

where

$$B(P_a, P_b) = -log(\sum_x P_a^{1/2}(x) P_b^{1/2}(x)) \quad (5)$$

is the Bhattacharyya coefficient [24].

In both cases, the merging cost depends on the size of the regions. The size term assures that the resulting partitions are size consistent, meaning that the area of the regions tends to increase as the number of regions into the partition decreases. However, this dependence favors the fusion of small regions, delaying the fusion of larger regions. Indeed, small regions cause less significant errors since the error contribution of the union of two small regions is small compared to the contribution resulting from the merging with a large region. As the fusion of large regions is delayed, area weighted merging orders suffer generally from over-segmentation. To provide a trade-off between under-segmentation and over-segmentation, the corresponding area unweighted version of the KL and BHAT merging orders have also been proposed [22]. They are as follows:

KL area-unweighted merging order:

$$f(R_a, R_b) = -D_{KL}(P_a \parallel P_{a\cup b}) - D_{KL}(P_a \parallel P_{a\cup b}), \quad (6)$$

BHAT area-unweighted merging order:

$$f(R_a, R_b) = \arg \max_{R_a \backsim R_b} \cdot B(P_a, P_b). \quad (7)$$

The region modeling based on empirical distribution has demonstrated a noticeable improvement with respect to first order statistical models where mean or median color values are used as region model since they do not assume that regions are homogeneous in color nor texture. However, the merging process starts by considering that each pixel is a single region, which is modeled still deterministically by its color value and therefore the effect of the statistical modeling become really important only in the late stages of the merging process. In the following sections, we discuss the limitations of this kind of modeling for segmentation purpose and we propose a solution.

III. Detecting occlusion

This section focuses on the occlusion detection. As discussed in the previous section, geometric signatures of occlusion are TJs. They are the projections of points where the contours of two objects in occlusion meet. The piece of each contour that emanates from the junction point is defined as a *branch*.

The detection of TJs in real images has been object of research for over thirty years but it still represents a valid challenge. The difficulty arises from the fact that junctions correspond to the crossing of different object contours, and, therefore, at junction locations the pixel intensities of different objects mix because of the blur introduced by the image acquisition system. Early junction detectors rely on a convolution-based approach, which includes gradient-based and edge-based methods. Gradient-based methods hypothesize junction locations by analyzing local gradient and level line curvature [25], [26], while edge-based methods detect junctions as

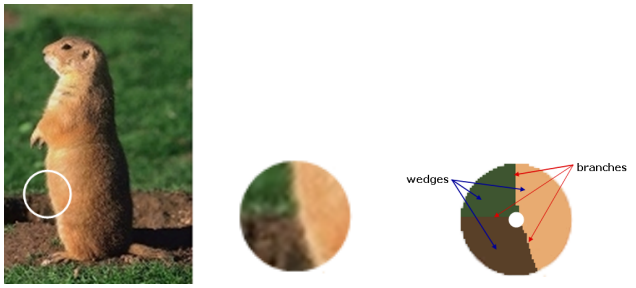


Fig. 3: Examples of T-junction template: (a) Original image, where a T-junction is marked by a white circle. (b) Zoom on the T-junction. (c) T-junction template: the T-junction is modeled as a piecewise constant wedges, delimited by the branches and a small neighborhood around the junction center is omitted because considered unreliable.

intersection of edges [27]. In all these methods, the use of a Gaussian impulse response leads to an inaccurate detection of the junction centers. A different strategy to hypothesize junction centers is adopted by model-based template matching techniques [28]–[31]. In this approach, the characterization of the junction is added to the localization criteria. These methods assume that a suitably small local neighborhood is sufficient to detect a junction. During the last decade, hybrid methods retaining both the edge-based and the template-based approaches have received much attention. Typically, hybrid methods model junctions as piecewise constant regions called *wedges* emanating from a central point, omitting a smaller disk centered at this point (see Fig. 3). The radial partition of the template is generally obtained by minimizing an energy function which measures the distance of the junction-model from the input signal. Candidate radial partitions are found by detecting edges and grouping them around the central point. The task is to find the minimum number of wedges that best describes the junction. The center identification is generally based on a local operator, while many different minimization strategies and many different criteria to characterize edges have been proposed [32]–[37].

A common limitation to state of art algorithms is that all methods used to find wedges rely on the assumptions of color or texture homogeneity that are generally not hold in a neighborhood of a junction. The basic idea underlying the region-merging process we propose is of avoiding the classical assumptions of color and texture homogeneity by taking a more general assumption.

The algorithm we have developed involves three main steps. A first selection step provides candidate points that represent potential TJs to be characterized and possibly validated in a second step. The characterization, namely the branch extraction, is performed on a close surrounding of candidate points, omitting a small neighborhood centered at them. The obtained branches are then propagated inside the omitted domain according to the "good continuation principle" [38] and constrained to meet at the candidate point. This procedure is also supported by psychophysical experiments [39] suggesting that junctions are detected even when the center is occluded. To each validated T-junction, a graduate measure of *junction likelihood* is assigned relying on the regularity of its branches. This measure is used in the third step, devoted to the reduction of clusters of validated points.

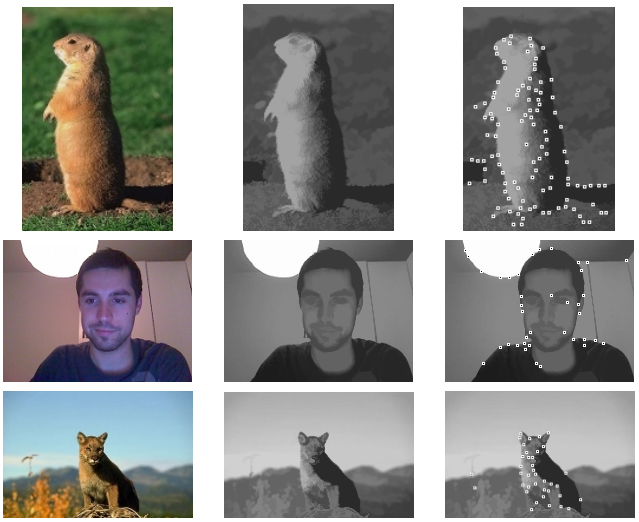


Fig. 4: Examples of candidate point selection: (a) Original images. (b) Result of applying a hierarchy of leveling. (c) Result of applying the SUSAN filter to the images in (b): candidate points are marked in black and are surrounded by a white square.

In the following, the working principle and the algorithmic implementation of each step are discussed and detailed.

A. Candidate point selection

In this section, we shall use for the discrete image representation a grid with integer coordinates. Since TJs are structural features, the search for candidate points is performed on the structural part of the image, also called *cartoon component*, obtained by a simplification of the original image with a hierarchy of leveling [40], based on Gaussian scale-space markers. More precisely, at each scale k the "cartoon component" U_k of the discrete image $U : \Gamma \to \mathcal{Z}$ is obtained as $U_k = \Lambda(M_k|U_{k-1})$, where $M_k = U * G_{\delta k}$ is the marker obtained by convolution of U with a Gaussian kernel G of standard deviation δ, U_{k-1} is the reference image with $U_0 = U$ and Λ is a leveling [41]. In all experiments, we used $\delta = 3$ and $k \in \{1, 2, 3\}$.

The selection of candidate points relies on the observation that TJs can be thought as a superposition of two adjacent corners. As a consequence, corners are good candidates points for TJs. Corners are localized by the SUSAN [42] filter. To take into account the localization inaccuracy of the SUSAN filter, coordinates of candidate points are allowed to vary on a small neighborhood. In practice, we apply a dilation with a square structuring element of size 5×5 on the mask of candidate points obtained by applying SUSAN. The mask resulting from the dilation defines the set of candidate points and the branch extraction is performed for each candidate point.

Although the proposed strategy selects corners as candidate points, it allows a fast and significant reduction of the number

of image pixels to be processed while keeping all real TJs. Fig. 4 shows some results of the candidate point selection step. As can be observed, the ratio between the number of selected candidate points and the number of image pixels is very small in all shown examples. However, in the case of very textured regions such as near the paw of the tiger or in the image on the first row, this ratio increases.

The branches of a T-junction correspond to lines passing between image pixels and therefore the T-junction center does not have integer coordinates. For each candidate point x with coordinates (i, j), we consider a squared neighborhood W of size $w \times w$ centered at the right down point x_{dc} with coordinates $(i + \frac{1}{2}, j + \frac{1}{2})$, where the branch extraction is restricted to and a 4×4 squared neighborhood Ω of x_{dc}, where the photometric profile is considered unreliable (see Fig. 5(a)). Taking into account that the photometric profile is considered reliable only outside Ω , our strategy consists in first extracting the branches in $(W - \Omega)$, by relying on the intensity profile, and then in extending them in Ω until the candidate point is reached, by relying on the continuity of the branches. The next section details how to perform the branch extraction in $(W - \Omega)$.

B. Branch extraction in $(W - \Omega)$

Contrary to classical hybrid approaches founded on region-merging, the branch extraction in $(W - \Omega)$ is performed by a BPT-based statistical region-merging algorithm. As merging order we use a statistical similarity measure between statistical region models proposed by Calderero et al. [22]. In section II, it has been pointed out that the advantages of the statistical region model became visible only in the late stages of the merging process, since a single pixel is modeled still deterministically by its color value instead of statistically. In the context of T-junction detection, since the region to be segmented is a small neighborhood of a given candidate point, the importance of a good modeling in early stages of the merging process becomes crucial. In [43] we have proposed to solve this problem by modeling each pixel statistically by a probability distribution, instead of deterministically by its color value. The probability distribution of a given pixel is obtained by exploiting self-similar structures, which can be detected by patch comparisons. By *self-similarity*, we mean that every small patch in a natural image has many similar patches in the same image. As can be appreciated in Fig. 6, this assumption is intuitively true for natural images since most objects in the real world have a self-similar or periodic structure: different parts of the same object show the same statistical properties at many different locations, as for instance in correspondence of objects contours. The fact that natural images have such a self-similarity property is a kind of stationarity assumption, actually more general and more accurate than any existing image statistics since it does not rely on an underlying model but directly on the data itself. This assumption has been proved to be sound by the works of Efros and Leung [44], and Levinia [45] and it has been successfully used in the seminal work of Efros [44] for texture synthesis and then in [46] and in [47] for image and video denoising. To the best of our knowledge, it is the first time that self-similarity is exploited for segmentation purpose.

Under the stationarity assumption, the probability distribution of a single pixel can be computed as follows. Let U be an image, x a pixel of the image domain and let $\mathcal{N}(x)$ be a square image patch centered at x which does not include x. Let us assume that the probability distribution of $U(x)$ depends only on the values of the pixels in $\mathcal{N}(x)$ and it is independent of the rest of the image (Markovian model). Then, the probability distribution of $U(x)$ given the pixel values of its neighborhood $\mathcal{N}(x)$, can be estimated by computing the set:

$\Gamma(x) = \{y : \frac{d(\mathcal{N}(x),\mathcal{N}(y))}{d(\mathcal{N}(x),\mathcal{N}_{best})\}} < (1 + \epsilon)\}$,

where $d(\mathcal{N}(x), \mathcal{N}(y))$ is a distance between a patch centered at x and a patch centered at another pixel location y of the image domain, $\mathcal{N}_{best}$ is the patch that gives the best patch match and ϵ is a small constant. The histogram of all pixel values in $\Gamma(x)$ gives an estimation of the probability distribution of the value of x given the values of its neighborhood $\mathcal{N}(x)$ [44]. More precisely, assuming that the pixel values range from 1 to L, the histogram is constructed by adding one to the value $U(y)$ for each $y \in \Gamma(x)$ and then normalizing so that the histogram integral is equal to one. However, setting a hard threshold $(1 + \epsilon)$ to defines the set $\Gamma(x)$ leads to be able to estimate the probability distribution only of pixels for which similar patches can be found. Indeed, in the absence of similar patches, the hard threshold strategy would leave the set $\Gamma(x)$ empty. To overcome such a problem, Buades et al. [48] proposed to use an exponential function, which allows a more continuous distribution. More precisely, the probability distribution of a pixel x conditioned to its neighborhood $\mathcal{N}(x)$, can be computed by computing for each pixel y the quantity:

$$w(x, y) = \frac{1}{Z(x)} e^{\frac{-d(\mathcal{N}(x),\mathcal{N}(y))}{h}}, \tag{8}$$

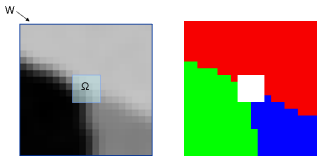
where $Z(x)$ is the normalizing factor:

$$Z(x) = \sum_{y\in U} e^{\frac{-d(\mathcal{N}(x),\mathcal{N}(y))}{h}}, \tag{9}$$

and

$$d(\mathcal{N}(x), \mathcal{N}(y)) = \sum_{z\in\{-1,1,...\}} \frac{(U(x - z) - U(y - z))^2}{K(z)}, \tag{10}$$

is the similarity between pixel values of a patch centered at x and a patch centered at y. The variable z indicates the displacement on the patch with respect to x, and K is a Gaussian-like function decaying from the center of the patch to its boundary. More precisely, $K(z) = (2 * \|z\| + 1)^2$ acts as a weight function of the euclidean distance between two patches. The goal of the function K is to give more importance on the patch to pixels closer to the reference pixel. Indeed, since we would like to compare local structure as accurately as possible, the error for nearby pixels in the patch should be greater than that of distant pixels. The parameter h controls the decay of the exponential function, and therefore of the function w. Due to the fast decay of the exponential term, large euclidean distances lead to nearly zero weights acting as an automatic threshold. To reduce the computation cost, the



(a) (b)

Fig. 5: (a) A window W centered at a candidate point: Ω is the neighborhood we consider unreliable, (b) Branch extraction in $W-\Omega$.

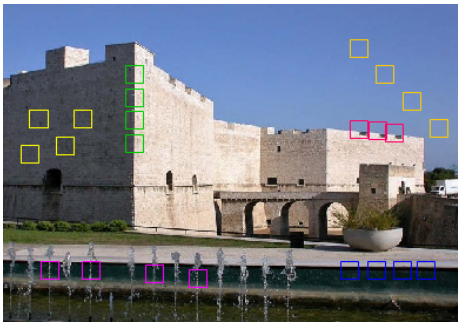
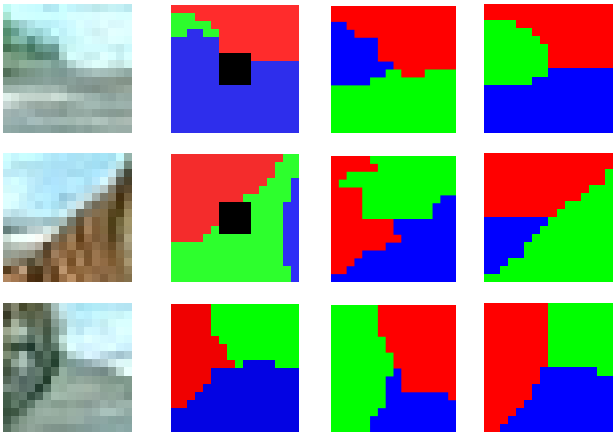


Fig. 6: Figure illustrating the phenomenon of non-local self-similarity. Similar patches are delimited by windows of the same color.



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

Fig. 7: Example of comparison: (a) Neighborhood of a candidate point to be segmented. (b) Local segmentation by using a region model of order zero. (c) Local segmentation by using the region model proposed in [22]. (d) Local segmentation by using the statistical region modeling proposed in III-B.

search for similar patches is restricted to a search window of size $S \times S$. The histogram corresponding to the probability distribution of x is obtained by adding, for each pixel y the value of the function $w(x,y)$ to $U(y)$.

Summarizing, the function $w(x,y)$ associated to the pixel x, depends on the similarity $d(\mathcal{N}(x),\mathcal{N}(y))$ between a patch centered at x and a patch centered at $y \in S \times S$ and satisfies the conditions $0 < w(x,y) < 1$ and $\sum_{y \in \S} w(x,y) = 1$. As the size of the image grows, for every pixel x, $w(x,y)$ converges to the conditional expectation of x given its neighborhood $\mathcal{N}(x)$. Indeed, as demonstrated in [48], $w(x,y)$ can be seen as an instance for the exponential operator of the Naradaya-Watson estimator [49], [50], which estimates the conditional expectation of a random variable.

Once pixels, which are considered as initial regions have been modeled by their pdf, the region merging algorithm starts to iteratively merge pairs of neighboring regions in $W - \Omega$ following the statistical *Kullback-Leibler merging criterion* (KL) [22], until three regions are obtained (Fig. 5(b)). The KL merging criterion for color images is defined as follows. If P_a and $P_{a \cup b}$ are the color histograms in the color space (YUV), then $D_{KL}(P_a \parallel P_{a \cup b})$ is given by

$D_{KL}(P_a \parallel P_{a \cup b}) = \alpha \cdot D_{KL}(P_{Ya} \parallel P_{Ya \cup Yb}) + (1 - \alpha) \cdot (D_{KL}(P_{Ua} \parallel P_{Ua \cup Ub}) + D_{KL}(P_{Va} \parallel P_{Va \cup Vb}))$. We set $\alpha = \frac{1}{2}$.

In Fig.7, we illustrate the advantage of the proposed pixel modeling in the context of the branch extraction. In Fig.7 (a) are shown the neighborhoods of candidate points those branches have to be extracted. In Fig.7 (b), 7 (c), and 7 (d) are shown the results of the branch extraction when using respectively a region model of order zero, the region model proposed in [22], and the statistical region modeling proposed in III-E. As it can be observed, the proposed pixel modeling yields a significant improvement in terms of contour definition with respect a deterministic pixel modeling. Points in 7 (b) are discarded before the branch propagation in since the segmented neighborhood does not fit the validation conditions described in III-D.

C. Setting the parameters for the local segmentation

The above described region merging algorithm involves a set of parameters that need to be fixed:

- *Size w of the local neighborhood W*: this parameter addresses the scale issue. The right scale depends on both the image resolution and the viewing distance.
- *Size n of the similarity window (patch size)*: this parameter has to be as small as possible to take care of the image details and fine structure, being at the same time robust to noise. Therefore, its value should increase with the amount of noise in the image.
- *Size S of the search window*: theoretically this parameter should be the same as the image size, but in practice a search window of five times the size of the similarity window guarantees good results with a reduced computational cost.
- *Value h of the filtering parameter*: this parameter controls the decay of the exponential functions. When the standard deviation of the noise is known, the value of h should depend on it [46]. For a small h, the similarity function would not be robust to noise since very similar neighborhood could give small value of the similarity. Nevertheless, by increasing the value of h, very different neighborhood could give large value of the similarity.
- *Number of bins used to construct the histogram*: the number of bins determines how accurately the probability distribution is represented. The highest the number of

used bins is, the more accurate would be the representation. However, by increasing the number of bins, the estimation of the divergence between histograms becomes inaccurate since there might be a very small number of samples per bin. In all experiments, we have fixed the ratio between the number of bins used for the luminance component (Y) and the number of bins used for the chroma components (U and V) to 3:1:1 in the YUV color space.

In order to fix these parameters we proceeded as follows. We considered two values of the similarity window: 3×3 (see Fig.s 8, 10 and 12) and 5×5 (see Fig.s 9, 11 and 13). For each value of the similarity window, we ranged the test values of the filtering parameter h between 30 and 110 with intervals of 20 and the values of the number of bins of the luminance component between 5 and 250 with intervals of 50. As can be observed, a smaller similarity window gives in general more accurate segmentation results. For a similarity window of 3×3 and a search window of size 15×15, good values for the filtering parameter and the number of bins for the luminance component are respectively 70 and 150.

D. Candidate point validation in $W - \Omega$ before branch extraction in Ω

After the local segmentation, we have to check if the segmented neighborhood fits the T-junction model. The validation of candidate points is based on the topology of the branches, as well as on the geometrical and photometric profile of wedges. The region merging strategy does not guarantee neither that the three final regions will reach Ω (see Fig. 14 (a)) nor that all the regions will intersect the boundary of W (see Fig. 14 (b)) with at least a minimum number of pixels equal to half the window length. In both cases the candidate point is discarded. To guarantee the visibility of each wedge, we impose a threshold on the minimum gray level difference between the mean gray level of each pair of wedges and on the minimum color difference between the mean color of each pair of wedges. If the minimum gray level or color difference is below a given threshold (t_{gray} and t_{color} respectively), the point is discarded (see Fig. 14 (c)). In most cases corresponding to object boundaries or textured regions, one wedge is composed of a very small number of pixels or looks like a narrow band. We then use a "size criterion" that is as follows: if at least one region completely disappears after an erosion (binary) with a square structuring element, then the candidate point is discarded (see Fig. 14 (d)). The size s of the square structuring element is related to the length w of W. To keep TJs, whose contours converging at the junction center form a small angle, s is taken as: $\frac{w}{6}$.

E. Branch extraction in Ω

As explained above, the photometric profile is not reliable in Ω, and thus the use of a region merging algorithm would be misleading. Instead, the extrapolation of the branches inside Ω is made according to the "good continuation principle" [38] and it is achieved in two steps. Let $\Omega = \Omega_1 \cup \Omega_2$ (see Fig. 15(b)).

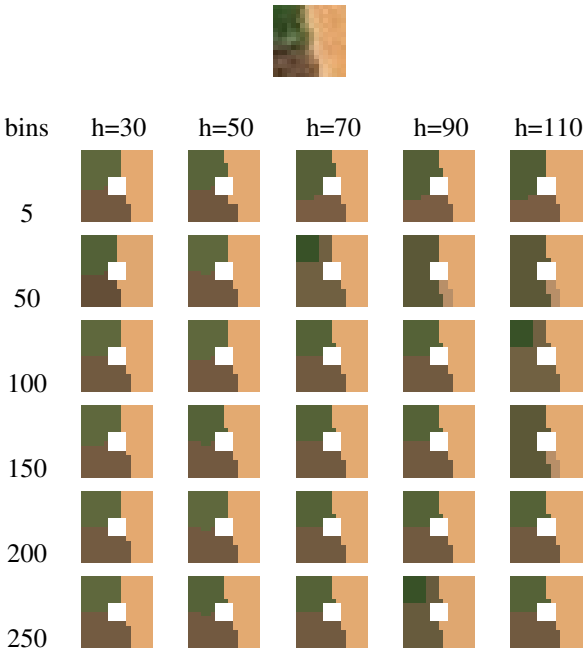


Fig. 8: Size of the similarity window of 3×3.

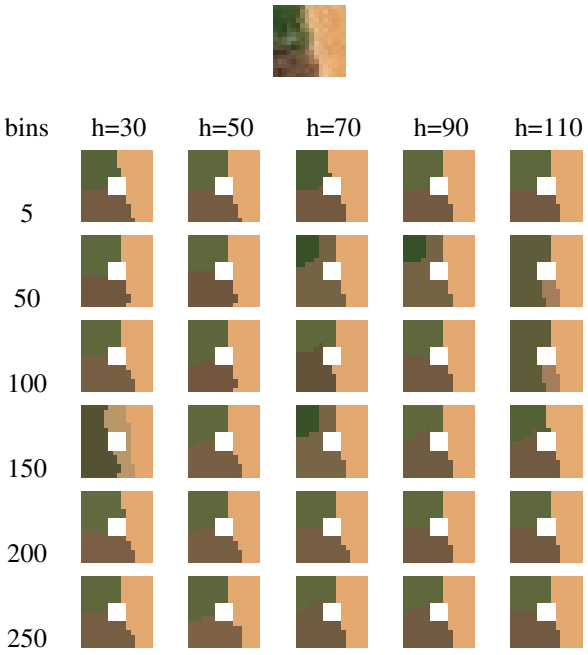


Fig. 9: Size of the similarity window of 5×5.

The first step consists in assigning to each pixel in Ω_2 the value of its adjacent pixel (in connectivity 4) outside Ω with the constraint that all three labels (red, green, blue) have to be assigned to at least one of the pixels N_i (see Fig. 15 (c)). This guarantees the propagation of contours as straight lines (see Fig. 15 (e)). In the second step, the branch extrapolation in Ω_1 is achieved using a geometric criterion that minimizes the sum of the absolute curvature at the new branch points created by the hypothetical labeling, with the constraint that branches meet at the candidate point. The curvature at the candidate

point is computed eliminating the stem of the hypothetical T-junction.

Let P be the set of pixels to be labeled, N the set of neighbors, and L the set of labels to be assigned (see Fig. 15 (c)).

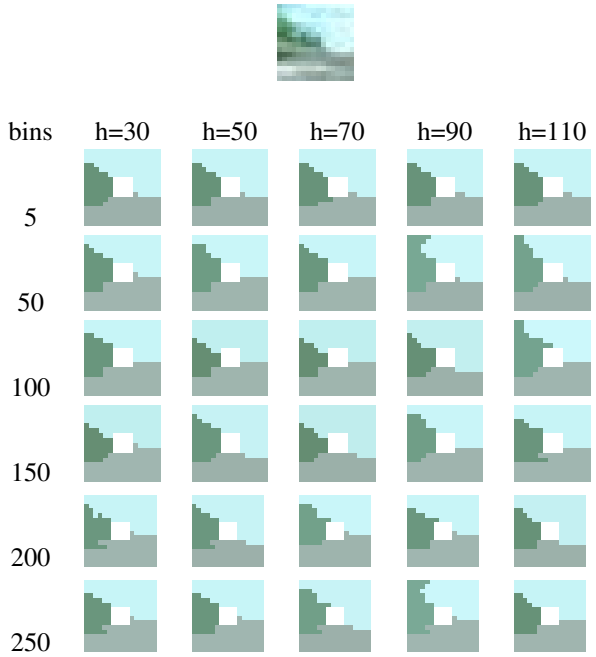


Fig. 10: Size of the similarity window of 3×3.

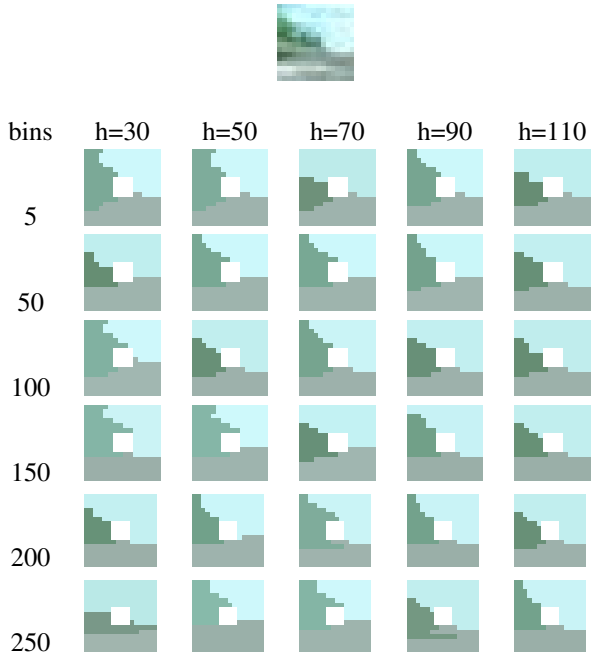


Fig. 11: Size of the similarity window of 5×5.

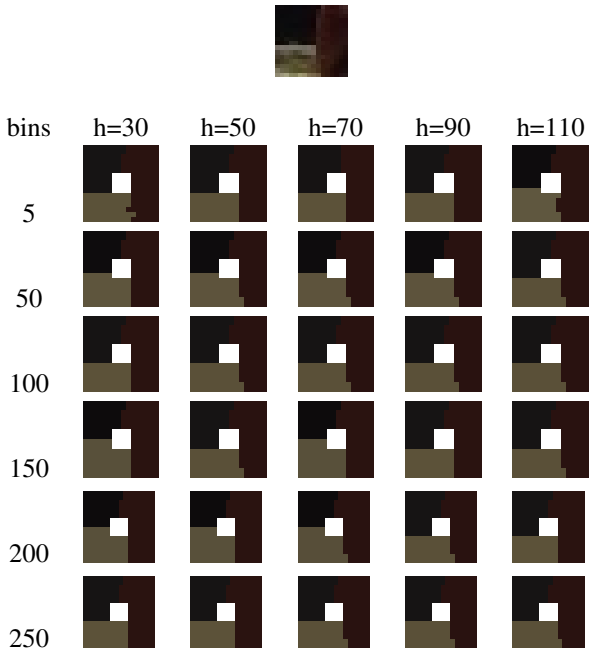


Fig. 12: Size of the similarity window of 3×3.

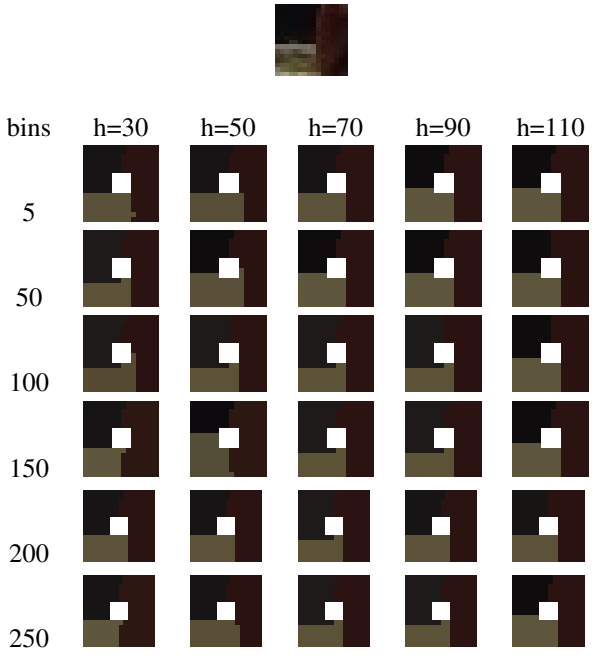


Fig. 13: Size of the similarity window of 5×5.

$P = \{A, B, C, D\}$
$N = \{N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8\}$
$L = \{red, green, blue\}$ In 4-connectivity, each pixel in P has two neighbors in N, whereas any pixel in N has only one neighbor in P.

To fulfill the constraint that branches meet at the candidate point, each label has to be assigned at least once at pixels in P. This goal is pursued in two stages: the first one is devoted to perform all label assignments that are mandatory (to fulfill the constraint). The second one, is devoted to label the remaining pixels.

The assignment of the label L_m of N_j to its neighbor in P, say P_i, is said mandatory if:

- N_j is the only pixel of N labeled with L_m such that its neighbor in P is still to be labeled
- there is no pixel in P labeled with L_m

Let us suppose that the label L_m of N_j is assigned to P_i. Then the other neighbor in N of P_i, say N_k has become useless since its unique neighbor in P has already been labeled. As a consequence, a new mandatory assignment may

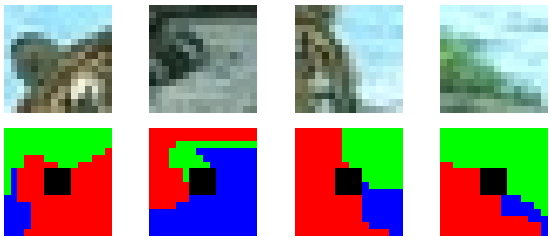


Fig. 14: Examples of candidate points discarded because: (a) All three regions do not join the boundary of Ω. (b) Not all three regions join the boundary of W with a minimum number of pixels. (c) The medium color difference between two regions is too small. (d) One of the three regions is too small.

have been generated if the occurrence of the label of N_k, say L_n, is 2 before the mandatory assignment. Since there are only three labels for eight pixels, this "cascade effect" may only occur once. According to the above considerations, the algorithmic structure of the first stage is as follows:

1) For each label L_m , compute how many neighbors belonging to N are labeled with label L_m, that is C_{red}, C_{green}, C_{blue}, where C_m is the number of neighbors N_i having label L_m.
2) If C_m is equal to 1:
   (a) search the N_i having value L_m
   (b) assign L_m to P_j such that P_j and N_i are neighbors
   (c) consider the neighbor of P_j: one is N_i , the other one is N_k. Decrement C_k by one.
3) go back to 2

The second step is a propagation (possibly with constraints) that minimizes a cost function. The possible constraints correspond to labels that still have not been assigned. The optimization cost is defined as the sum of the absolute curvature at the new branch points created by the hypothetical labeling. The points of the branches are points with half-integer coordinates. The computation of the curvature at these points is based on an interpolation at the center of the 2×2 window made of pixels (x, y), $(x + 1, y)$, $(x, y + 1)$ and $(x + 1, y + 1)$. More precisely, the gradient of each interpolated point is computed as:

$$Du_x(i,j) = \tfrac{1}{2}(-[u(i,j+1) + u(i+1,j+1)] + [u(i,j) + u(i+1,j)]) \quad (11)$$

$$Du_y(i,j) = \tfrac{1}{2}([u(i+1,j) + u(i+1,j+1)] - [u(i,j) + u(i,j+1)]) \quad (12)$$

and the curvature by the finite difference scheme proposed in [51]. The algorithmic structure of the second stage is as follows:

- If there is a P_j whose two neighbors in N have the same label L_m, then P_j has to be labeled with L_m.
- If there are P_j that still have to be labeled, then compute all possible assignments and their costs, and among all assignments that satisfy the constraints, choose the one having the minimum cost.

The final result is shown in Fig. 15 (f).

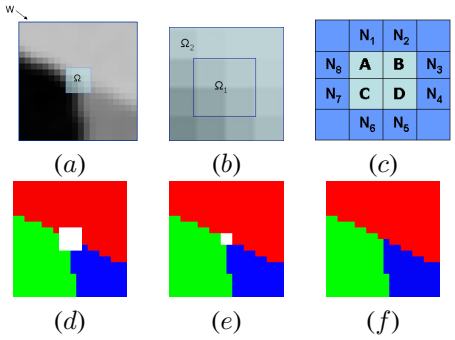


Fig. 15: (a) Image to be segmented. (b) Partitioning of $\Omega = \Omega_1 \cup \Omega_2$. (c) Labeling of Ω_1 and Ω_2. (d) Branch extraction in $W - \Omega$. (e) Branch propagation in Ω_2. (f) Branch propagation in Ω_1.

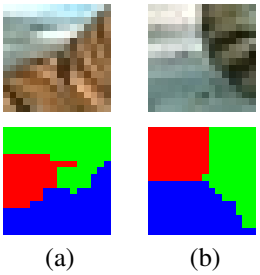


Fig. 16: Examples of candidate points discarded because: (a) The sum the curvature of each point of a branch is too large. (b) There is no pair of branches with an angle of $\phi + -\phi/4$.

F. Candidate points validation after branch extraction in Ω

After the branch extraction is completed, two validation criteria are checked. The first validation criterion consists in measuring the "smoothness" of the branches. Since object contours are smooth, T-junction branches are expected to be smooth. A single branch is considered to be smooth if the value of the integral of the absolute curvature on the three branches is below to a given threshold t. The value of this threshold depends on the window size w and it is computed as follows: $k = cw/2$, where c is a small value (see Fig. 16 (a)).

The second validation criterion deals with the branches orientation and it is necessary in order to distinguish from other junction types. For each branch, we first compute the vector that represents its medium orientation in W by averaging the orientation of each point of the branch. Then, we compute the angles between each pair of vectors. We say that a candidate point represents a T-junction if there is a pair of vectors such that the angle between them is equal to π with precision $\frac{1}{n}$ (see Fig. 16 (b)). In all experiments we fixed $n = 4$.

G. Cluster reduction

As result of the validation step, we obtain a set of clusters (see Fig. 18 (a)). Clusters are due to the locality of the branch extraction strategy: the shape of wedges of adjacent candidate points varies slightly and if a candidate point is validated, its neighbors have a high probability of being validated too.

Fig. 17: Examples of cluster: (a) Original image. (b) Before removing isolated points: there are some spurious TJs. (c) After removing isolated points. (d) After the cluster reduction step: the vectors point to the region closer to the viewpoint. As can be observed, the roofs of the TJs have been correctly computed.

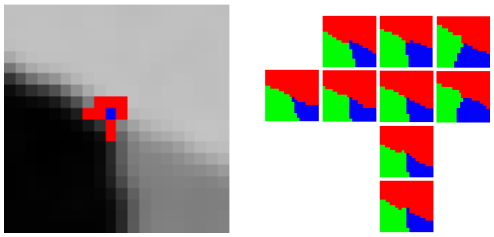


Fig. 18: Example of cluster reduction: (a) Validated points are marked as color pixels. The point having the smallest value of the sum of the absolute curvature on each branch is marked in blue. (b) Local segmentations corresponding to each validated point: as can be observed, the pixel marked in blue in (a) is the one having the most regular branches.

As a consequence, isolated points have an high probability of being spurious detection and therefore they are removed (see Fig. 17). For each cluster, the choice of the point that best represents the cluster is based on a graduate measure of T-junction likelihood related to the smoothness of the branches. More precisely, the point that best represents the cluster is the one which has the minimum value of the sum of the absolute curvature of each point of the branches (see Fig. 17 (d) and Fig. 18).

Fig. 19 summarizes the complete T-junction detection algorithm through a block diagram.

H. Parameter setting

The validation process described in section III-D and section III-F involves a set of parameters to be adjusted: the maximum value of the sum of the absolute curvature of each point of a branch, the minimum color difference between two wedges, and the minimum gray level difference between two wedges. In order to fix them, we have studied the performance of the T-junction detection algorithm as a function of the parameters. More precisely, we have plotted the Precision/Recall (P/R) curves corresponding to 4 natural images, for which the ground truth has been fixed beforehand manually. The image set forming the benchmark for parameters optimization has been chosen to minimize the subjectivity of any hand-marked ground truth and to maximize the variety of natural scene. Each P/R curve is associated to a single parameter p and it has been obtained by ranging the value of p in an interval around the initial optimum value, keeping fixed the other parameter values. As initial set of optimal parameter values, we have taken 0.8 as maximum sum of the absolute curvature for each branch, 10 and 5 respectively as minimum color difference and minimum gray level difference between wedges. By looking at the P/R curves, we have update the initial set of optimal parameter values and we have computed again the P/R curves. We have iterated this procedure until the set of initial parameter values has remained fix during two consecutive iterations.

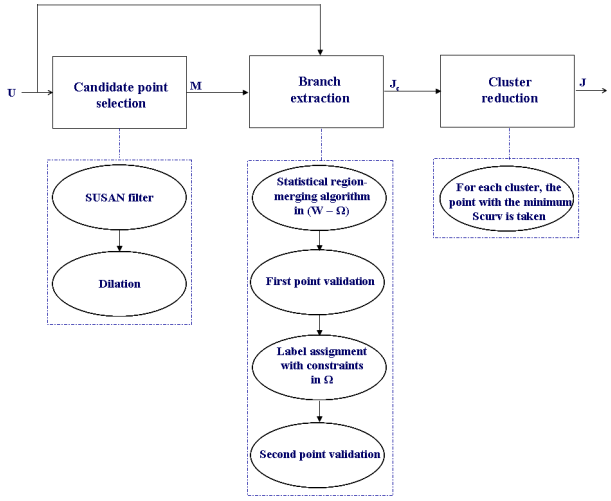
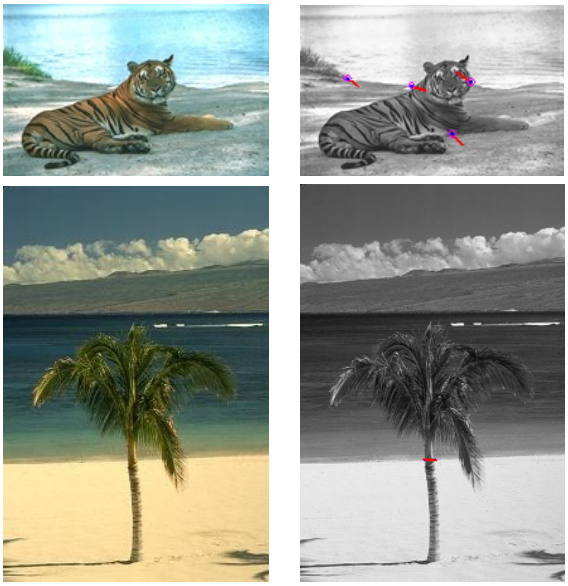


Fig. 19: Block diagram of the T-junction detection algorithm by region merging. U is the original image, M is the mask marking the candidate points to be analyzed, J_c is the image marking the points that have been validated, J is the image marking the center of the detected TJs, S_{curv} is the sum of the absolute curvature at each point of each branch.

The recommendable set of values for the three above mentioned parameters are 0.7, 20, and 20. Fig. 20 shows some examples obtained by using this set of parameter values. Each detected T-junction has been visualized as a vector whose origin (in blue) represents the T-junction center, whose direction is given by the direction of the stem and whose orientation points to the top. The overall performances are in general convincing.

IV. Segmenting the image preserving TJs

This section details how to obtain a BPT-based segmentation of the image, which preserves the previously detected TJs. We start from the initial partition of all image pixels and we model each pixel statistically by a probability distribution obtained as described in section . From our practical experience, we have found that, to segment the whole image, for a similarity



(a) (b)

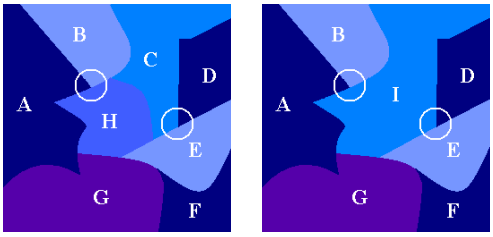
Fig. 20: (a) Original image. (b) Result of the junction detection on the original image.

window of 3×3 and a search window of size 15×15, good values for the filtering parameter h and the number of bins for the luminance component are respectively 50 and 50 and the merging order which allows to define more precisely self-similar regions is the BATH area unweighted criterion.

In Fig.23 we show the results of the segmentation obtained by modeling each pixel deterministically (see Fig.23 (b)) and by modeling each pixel statistically (see Fig.23 (c)). As it can be observed, the proposed pixel modeling gives much accurate results.

In order to preserve TJs, we introduce the concept of *incompatibility*. Two regions are said *incompatible* if they are involved in an occlusion relation and therefore are supposed to belong to different levels of depth. When two regions are incompatible, they cannot be merged. Hence, the concept of incompatibility is used as merging criterion: if the pair of neighboring regions proposed by the merging order are incompatible, the proposed merging is skipped. Incompatibility is an inheritable property. In Fig. 21, the regions A, B and H are incompatible with each other as well as the regions C, D, E. When the regions C and H are merged to form the region I, all incompatible relations in which C and H are involved, are inherited by I. As a consequence, the region I becomes incompatible with the regions A, B, D, and E. The region-merging process terminates when all regions became incompatible, and therefore no more mergings are allowed.

In Fig. 22 the results of applying the region merging algorithm described in this section without preserving TJs and using the number of regions as merging criterion are shown. Both, the KL and the BHAT merging orders, in their weighted and unweighted versions have been used. As can be observed in all four cases (see Fig. 22 (c), (d), (e), (f)), the 10 regions of the final partition do not correspond to the 10 most perceptually meaningful regions.



(a) (b)

Fig. 21: (a) Example illustrating the concept of incompatibility in the merging process. (a) TJs are marked by white circles. The regions A, B, and C are incompatible with each other as well as the regions D, E, and C.(b) The regions C and H have been merged to form the region I. The region I is incompatible with A, B, D, and E.

Instead, when incompatibility is used as merging criterion, TJs are preserved and the result of the segmentation is correct (see Fig. 22 (b)). This improvement is attributable to the fact that the process of grouping by statistical-similarity through the merging order is somehow corroborated by the process of separating by depth-dissimilarity through the merging criterion, being both processes treated under an unified region-merging framework. Indeed, the merging order proposes a merging based on a statistical similarity between the region models and the merging criterion validates or not the proposed merging depending on if the pair of neighboring regions involved belong to the same level of depth or not, that is if they are or not compatible.

The following section presents a method to infer a global, *consistent* depth ordering between regions of the final partition.

V. Graph formalization and reasoning

As stated in the previous section, the regions of the final partition are incompatible. For each triplet of incompatible regions arisen from an occlusion relation, a depth assessment based on the depth interpretation of TJs can be done: locally, the region delimited by the roof of the T appears to be in front of the ones delimited by the stem. However, it needs to be taken into account that the depth interpretation of pairs of TJs that share an edge may give rise to an inconsistency. In Fig. 24 (a) there is an example of first order *inconsistency* (involving a pair of TJs). Region D is in front of region A for one T-junction, while the converse is true for the other. Higher order inconsistencies involve more than two TJs. We formalize the problem of finding a global, consistent depth interpretation through a Directed Graph (DG). A DG is specified by $DG = (V, E_W, W)$, where V is a set of nodes, E_W is a set of edges and W is the matrix of weights attached to the edges. In our formalization, each node represents a region of the final partition and each directed edge represents the relative depth relation between two regions. Edges are specified as ordered pairs: an edge $e = (X, Y) \in E$ is considered to be directed from X to Y meaning that the region X is in front of the region Y. The weight attached to each edge corresponds to the number of occurrences the

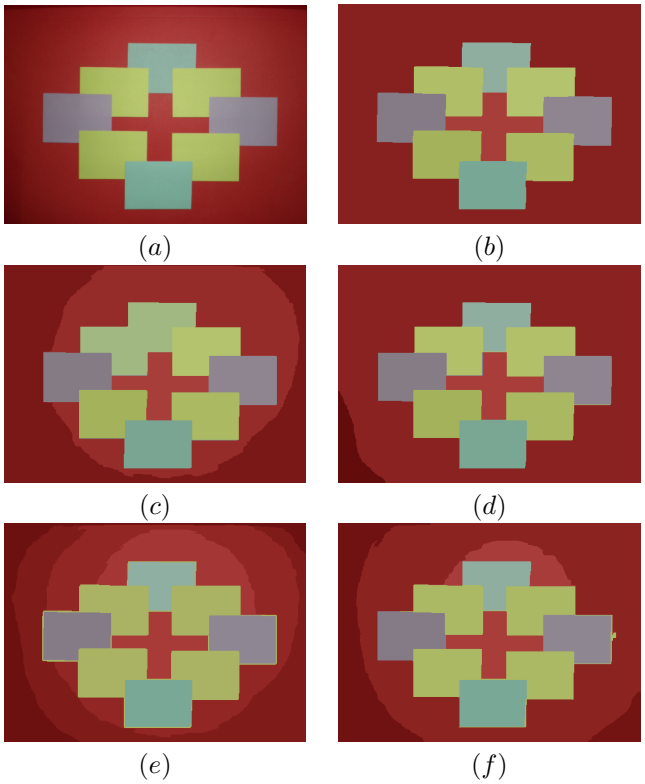


Fig. 22: (a) Original image to be segmented. (b) Segmentation obtained preserving TJs, using the BHAT unweighted merging order. Segmentation results obtained by using the statistical region merging algorithm without preserving TJs, by using the: (c) Area weighted BHAT merging order. (d) Area unweighted BHAT merging order. (e) Area weighted KL merging order. (f) Area unweighted KL merging order.

depth relation represented by the edge has been inferred from different occlusion relationships.

For instance, in Fig. 24, the weight of the edge $e = (C, A)$ is 2, whereas the weight of the edge $e = (A, C)$ is 1. With this formalization, local constraints are allowed to propagate along the graph and the search for inconsistent pairs of TJs is reduced to the search of cycles on the DG (dashed thick red arrows in Fig. 24(b)). The search for directed cycles is performed by a Depth-First Search (DFS) algorithm [52]. This algorithm may be computationally expensive when the number of nodes involved is high. However, being usually the number of regions of the final partition small, the corresponding computational load is moderate. Inconsistencies are solved by suppressing the edge(s) on the cycle with lowest cost. Since the depth relation associated to the edge with the lowest cost is considered unreliable, the other edge (dashed thin blue arrow in Fig. 24(b)) associated with the T-junction from which the unreliable depth relation arises is also removed. As a result, a DAG is obtained (see Fig. 24 (c)). Each DAG gives rise to a *partial order* $\leqslant$ on the set of it nodes V.

Many different DAGs may give rise to the same *reachability relation*. The reachability relation of a DAG is the set of all ordered pairs (X, Y) of nodes in V for which there exist nodes $v_0 = X, v_1, \ldots, v_d = Y$ such that $e = (v_{i-1}, v_i) \in E, \forall 1 \leqslant i \leqslant d$. The reachability relation of a DAG is also called

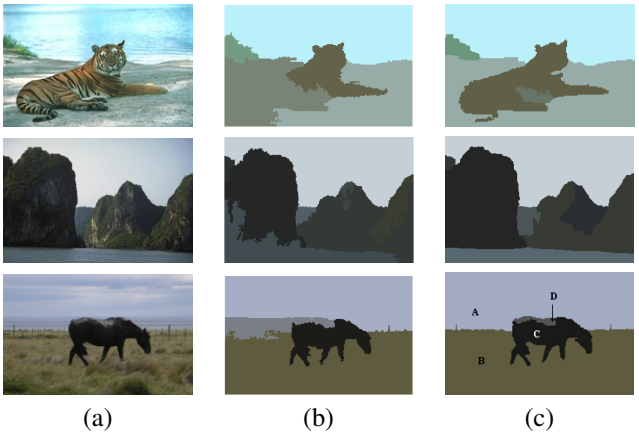


Fig. 23: Example of comparison: (a) Original image to be segmented. (b) Segmentation which preserves TJs by using the region model proposed in [22]. (d) Segmentation which preserves TJs by using the statistical region modeling proposed in III-B.

transitive closure and corresponds, among all DAGs which give rise to the same reachability relation, to the one with the maximum number of edges. Instead, the DAG, among all the DAGs which give rise to the same reachability relation, with the minimum number of edges is called *transitive reduction*. The transitive reduction of a finite DAG is obtained by removing redundant edges while maintaining identical reachability properties. An edge $e = (X, Y)$ is said *redundant* if there exists a path from X to Y that does not contain the edge. For example (see Fig. 24 (c)), the edge $e = (G, A)$ is redundant since it is possible to go from the node G to the node A passing through the node H. The graphical rendering of a transitive reduction is called Hasse diagram. Each element of the DAG is drawn on the Hasse diagram as a node and line segments are drawn between these nodes according to the following two rules:

- If $X \leqslant Y$, then the node corresponding to X appears lower in the Hasse diagram than the point corresponding to Y.
- The line segment between the points corresponding to any two nodes X and Y of the set V, is included in the Hasse diagram as a line segment that goes upward from x to y if and only if, $X \leqslant Y$ and there is no Z such that $X \leqslant Z \leqslant Y$ (the edge $e = (X, Y)$ is not redundant).

Any Hasse diagram uniquely determines a partial order, and any finite partial order has a unique transitive reduction.

In our formalization, the Hasse diagram corresponding to the transitive reduction of the DAG is exactly the depth map (see Fig. 24 (d)). Since there is no depth order between the regions forming the stem of a T-junction, they appear on the Hasse diagram as as leaves (A and B), without any information about their respective depth, unless of course, an order between them can be inferred by other TJs.

VI. Experimental results

We tested our algorithm on a set of real images. For each experiment we show four images: the original image (see

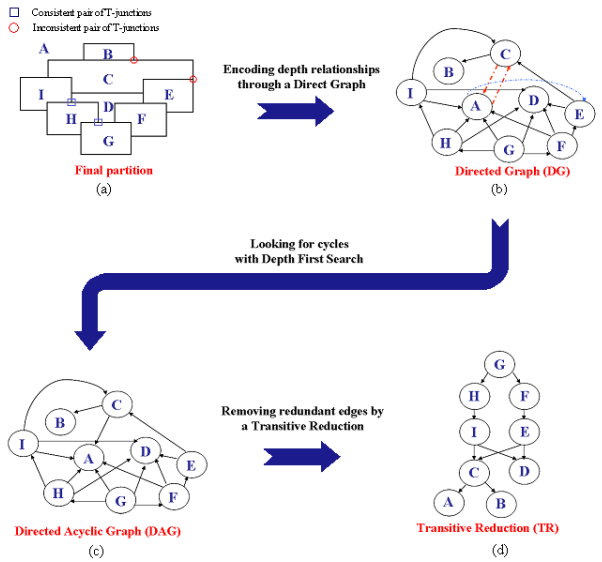


Fig. 24: (a) Partitioned image. (b) Associated DG. (c) Associated DAG. (d) Hasse diagram resulting from the transitive reduction of the DAG.

Figure 25(a)); a gray level version of the original image, where the TJs detected by using the region-merging algorithm detailed in section III-B are represented through a vector pointing to the region closer to the viewpoint (see Figure 25(b)); the segmented image (see Figure 25(c)); the map of relative depths, which is rendered as a gray level image (high values indicate regions closer to the viewpoint) (see Figure 25(d)). As can be observed in the example on the first row and on the second rows, the last level of depth includes two regions, corresponding to leave nodes of the Hasse diagram. In the example on the third row, there is a case of conflict between the regions A and C: while region A is interpreted as foreground and region C as background for one TJs, the contrary is true for two TJs. The solution of this conflict leads to a correct depth interpretation. A similar case is shown in the example on the fourth row, which involves more depth levels. The following two examples are more complex scenes, for which a correct depth interpretation is obtained.

The example on the last two rows illustrates the limitations of the proposed method. In the first example, the regions corresponding to the sky visible through the tree branches have been merged with the region corresponding to the tree branches because there is simply no occlusion relationship allowing to separate them. In general, this happens when the complementary of the foreground region is not a single regions, that is when the foreground regions is not simply connected. In the second example, a case of self-occlusion is involved. The nodes corresponding to the regions A, B, C, and D form a cycle on the DG, characterized by the fact that all edges connecting the nodes of the cycle have the same weight. In this case, all regions corresponding to the nodes of the cycle are considered to belong to the same level of depth and, therefore, they appear as a single region on the map of relative depths.

VII. Conclusions

This paper has addressed the problem of estimating relative depth information from a single still image by relying only on the depth cue of occlusion.

The proposed strategy consists in first detecting TJs, then in segmenting the image preserving the TJs previously detected, and finally in depth ordering the regions of the final partition by relying on the depth information provided by TJs. The global depth ordering is achieved through a graph formalization, which allows to easily detect and solve possible conflicting interpretations. Contrary to the state of the art, our method is fully automatic and does not make any assumption on the image structure. A new region based approach for the detection of TJs has been proposed The unified framework is provided by a region-merging algorithm, which iteratively merges pairs of neighboring regions following a statistical similarity criterion. Under this framework, the process of grouping by feature similarity relies on a statistical pixel modeling, which exploits the image self-similarity, while the process of separating by depth dissimilarity corresponds to a merging criterion that acts as a sieve on the mergings proposed by the merging criterion. The mechanism allowing to solve possible conflicting local depth interpretations relies on a DG, which encodes the depth relationships between the regions of the final partition and allows to detect and solve possible conflicts as cycles on the graph. The final depth ordering is then obtained as transitive reduction of the DAG. Experimental results on real images have demonstrate

References

[1] L.R. Williams, "Perceptual organization of occluding contours," in *In Proc. of International Conference on Computer Vision (ICCV)*, 1990, pp. 133–137.

[2] E. Saund, "Perceptual organization of occluding contours of opaque surfaces," *Computer Vision and Image Understanding*, vol. 76, no. 1, pp. 70–82, 1999.

[3] D. Geiger and Parida L., "Visual organization for figure-ground separation," in *International Conference on Computer Vision and Pattern Recognition, CVPR*, 1996, pp. 155–160.

[4] S. Madarasmi and D. Ting-Chuen Pong Kersten, "Illusory contour detection using MRF models," in *World Congress on Computational Intelligence*, 1994, pp. 4343 – 4348.

[5] N. Kogo, C. Strecha, R. Fransen, G. Caenen, J. Wagemans, and L.J. Van Gool, "Reconstruction of subjective surfaces from occlusion cues," in *Biologically Motivated Computer Vision: second workshop of BMVC*, 2002, pp. 311–312.

[6] P. Mordohai and G. Medioni, "Junction Inference and Classification for Figure Completion using Tensor Voting," in *Computer Vision and Pattern Recognition Workshop, CVPRW*, 2004, vol. 4, pp. 56–64.

[7] S.R. Thiruvenkadam, F. Chan, T, and B.W. Hong, "Segmentation under occlusion using selective shape prior," *Scale Space and Variational Methods in Computer Vision*, vol. 4485, pp. 191–202, 2007.

[8] X.Y. Stella, T.S. Lee, and T. Kanade, "A hierarchical markov random field model for figure-ground segregation," in *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition, EMM CVPR*, 2001, pp. 110–133.

[9] R.-X. Gao, T.F. Wu, S. C. Zhu, and N. Sang, "Bayesian inference for layer representation with mixed markov random field.," in *Energy Minimization methods in Computer Vision and Pattern Recognition*, 2007, vol. 4679, pp. 213–224.

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

[11] E. Delage, H. Lee, and Y. Ng, "A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image," in *International Conference on Computer Vision and Pattern Recognition, CVPR*, 2006, pp. 1–8.

Fig. 25: Examples of segmentation with depth: (a) Original image. (b) T-junction detection. (c) Segmentation. (d) Depth ordering.

[12] A. Saxena, Min Sun, and A.Y. Ng, "Learning 3-d scene structure from a single still image," in *In Proc. of International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[13] D. Hoiem, A.N. Stein, A.A. Efros, and M. Hebert, "Recovering Occlusion Boundaries from a Single Image," in *In Proc. of International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[14] D. Rother and G. Sapiro, "3d reconstruction from a single image," *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence. IMA PreprInternational*, 2009.

[15] W. Metzger, *Gesetze des sehens.*, Waldemar Kramer, 1975.

[16] A. Baumberg, "Reliable Feature Matching Across Widely Separated Views," in *Proc. of Computer Vision and Pattern Recognition*, 2000, pp. 774–781.

[17] P. Favaro, A. Duci, Y. Ma, and S. Soatto, "On exploiting Occlusions in Multiple-view Geometry," in *Proc. of International Conference on Computer Vision*, 2003, pp. 479–486.

[18] N. Apostoloff and A. Fitzgibbon, "Automatic video segmentation using spatiotemporal T-junctions," in *Proc. of British Machine Video Conference*, 2006, pp. 1–10.

[19] P.J. Kellman and T.F. Shipley, "Visual interpolation in object perception," *Current Directions in Psychological Science*, vol. 1, no. 6, pp. 193–199, 1991.

[20] M. Dimiccoli and P. Salembier, "Exploiting t-junctions for depth segregation in single images.," in *In proc. of International Conference in Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei (Taiwan), April 2009.

[21] M. Dimiccoli, *Monocular depth estimation for image segmentation and filtering.*, Dept. Signal Theory Commun. Ph.D. dissertation, Univ.

Politecnica de Catalunya, Barcelona, Spain., 2009.
[22] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation and information retrieval," *IEEE Trans. on IP*, vol. 7(4), pp. 561–576, 2000.
[23] F. Calderero and F. Marques, "General region merging approaches based on information theory statistical measures.," in *Proc. of International Conference on Image Processing*, San Diego (CA), October 2008.
[24] S. Kullback and R.A Leibler, "On Information and Sufficiency.," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
[25] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions.," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99 –109, 1943.
[26] J.J. Koenderink and W. Richards, "Two-dimensional curvature operators.," *Journal of Optical society of America*, vol. A5, pp. 1136–1141, 1988.
[27] B.M. Romeny, L.M. Florack, J.J. Koenderink, and M.A. Viergever, "Invariant third order properties of isophotes: T-junction detection," in *Proc. of 7th Scand. Conf. on Image Analysis*, 1991, pp. 346–353.
[28] C.G. Harris and M. Stephens, "A combined corner and edge detection," in *Proc. of 4th Alvey Vision Conference*, 1988, pp. 147–151.
[29] D. Beymer, *Junctions: Their Detection and Use for Grouping Images*, Master Thesis, Massachusetts Institute of Technology, Departament of Electrical Engineering, 1989.
[30] J. Bigun, "A Structure Feature for Some Image Processing Appications Based on Spiral Functions," in *Proc. of European Conference on Computer Vision*, 1994, pp. 383–394.
[31] W. Forstner, "A framework for Low Level Features Extraction," in *Proc. of European Conference on Computer Vision*, 1994, pp. 383–394.
[32] M.F. Hueckel, "An Operator which Locates Edges in Digitized Pictures," *Journal of the ACM*, vol. 18, pp. 113–125, 1971.
[33] L. Parida, D. Geiger, and R. Hummel, "Junctions: Detection, Classification, and Reconstruction," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 687–698, 1998.
[34] M.A. Ruzon and C. Tomasi, "Edge, junction, and corner detection using color distributions.," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2001.
[35] M.A. Cazorla and F. Ercolano, "Two Bayesian Methods for Junctions Classification," *IEEE Trans. on Image Processing*, vol. 12, no. 3, pp. 317–327, 2003.
[36] R. Laganiere and R. Elias, "The detection of junction features in images.," in *In Proc. of International Conference in Acoustics, Speech, and Signal Processing (ICASSP)*, 2004, vol. 3, pp. 573–576.
[37] R. Bergevin and A. Bubel, "Detection and Characterization of junctions," *Computer Vision and Image Understanding*, vol. 93, no. 3, pp. 288–309, 2004.
[38] E.D. Sinzinger, "A model-based approach to junction detection using radial energy.," *Pattern Recognition*, vol. 41, pp. 494–505, 2008.
[39] N. Montanari, "On the optimal detection of curves in noisy pictures," *Communications of the ACM archive*, vol. 14, no. 5, pp. 335 – 345, 1971.
[40] S. Wuerger, R. Shapley, and N. Rubin, "On the visual perceived direction of motion, Hans Wallach: 60 years later," *Vision Research*, vol. 25, pp. 317–367, 1996.
[41] P. Maragos and G. Evangelopoulos, "Leveling cartoons, texture energy markers, and image decomposition," in *Proc. 8th International Symp. on Mathematical Morphology*, Rio de Janeiro (Brazil), October 2007.
[42] F. Meyer, "The levelings," in *Proc. of International Symposium on Mathematical Morphology (ISMM).*, 1997, vol. 2, pp. 211–214.
[43] S.M. Smith and M. Brady, "Susan - a new approach to low level image processing," *International Journal of Computer Vision (IJCV)*, vol. 23, no. 1, pp. 45–78, 1997.
[44] M. Dimiccoli and P. Salembier, "Hierarchical region-based representation for segmentation and filtering with depth in single images.," in *In proc. of International Conference on Image Processing (ICIP)*, Cairo (Egypt), November 2009.
[45] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," *In Proc. of International Conference on Computer Vision (ICCV)*, October 1999.
[46] E. Levinia, *Statistical issues in texture analysis.*, PhD Thesis, Berkley, CA, 2002.
[47] A. Buades, B. Coll, and J.M. Morel, "The staircasing effect in neighborhood filters and its solution," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1499–1505, 2006.
[48] A. Buades, B. Coll, and J.M. Morel, "Nonlocal image and movie denoising," *International Journal of Computer Vision*, vol. 76, no. 2, pp. 123 – 139, 2008.
[49] A. Buades, B. Coll, and J.M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale modeling and Simulation, Society for Industrial and Applied Mathematics (SIAM)*, vol. 4, no. 2, pp. 490–530, 2005.
[50] G.S. Watson, "Smooth regression analysis.," *Sankhya: The Indian Journal of Statistics.*, vol. 26, pp. 359–372, 1964.
[51] E.A. Naradaya, "On estimating regression.," *Theory of Probability and its Applications.*, vol. 10, pp. 186–190, 1964.
[52] F. Guichard, J.M. Morel, and R. Ryan, *Contrast invariant image analysis and PDE's.*, Preprint CMLA. Web Address: www.cmla.ens-cachan.fr/Utilisateurs/morel/JMMBookOct04.pdf, 2004.
[53] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, chapter Depth-first-search, pp. 540–549, MIT Press and McGraw-Hill, 2001.