Field of Junctions: Extracting Boundary Structure at Low SNR

Dor Verbin and Todd Zickler
Harvard University
{dorverbin,zickler}@seas.harvard.edu

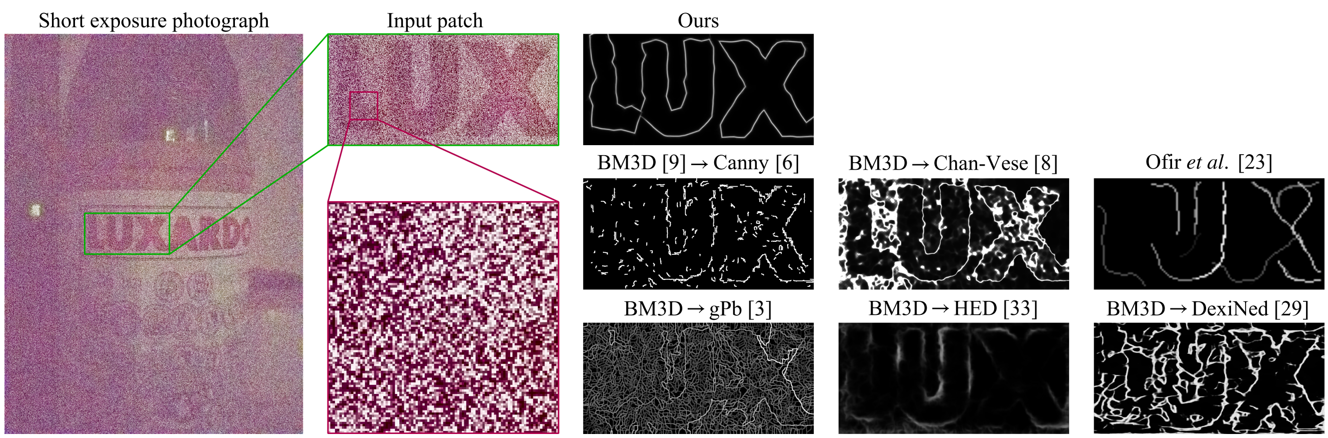

Figure 1: Detecting boundaries at short exposure (1/5000s). The field of junctions extracts boundary structure at noise levels where other methods fail, even when the others are preceded by denoising and are optimally tuned for the image. Additionally, our model interprets its boundaries into component contours, corners, junctions, and regional colors (see Figure 2).

Abstract

We introduce a bottom-up model for simultaneously finding many boundary elements in an image, including contours, corners and junctions. The model explains boundary shape in each small patch using a 'generalized M-junction' comprising M angles and a freely-moving vertex. Images are analyzed using non-convex optimization to cooperatively find $M+2$ junction values at every location, with spatial consistency being enforced by a novel regularizer that reduces curvature while preserving corners and junctions. The resulting 'field of junctions' is simultaneously a contour detector, corner/junction detector, and boundary-aware smoothing of regional appearance. Notably, its unified analysis of contours, corners, junctions and uniform regions allows it to succeed at high noise levels, where other methods for segmentation and boundary detection fail.

1. Introduction

Identifying boundaries is fundamental to vision, and being able to do it from the bottom up is helpful because vision systems are not always familiar with the objects and scenes they encounter. The essence of boundaries is easy to articulate: They are predominantly smooth and curvilinear; they include a small but important set of zero-dimensional events like corners and junctions; and in between boundaries, regional appearance is homogeneous in some sense.

Yet, despite this succinct description, extracting boundaries that include all of these elements and exploit their interdependence has proven difficult. After decades of work on various subsets of contour detection, corner detection, junction detection, and segmentation, the community is still searching for comprehensive and reliable solutions. Even deep encoder-decoder CNNs, which can be tuned to exploit many kinds of local and non-local patterns in a dataset, struggle to localize boundaries with precision, motivating an ongoing search for architectural innovations like skip connections, gated convolutions, bilateral regularization, multi-scale supervision, kernel predictors, and so on.

We introduce a bottom-up model that precisely discerns complete boundary structure—contours, corners, and junctions—all at the same time (see Figures 1 & 2). It does this by fitting a non-linear representation to each small image patch, with $M+2$ values that explain the patch as being uniform or containing an edge, thin bar, corner, or

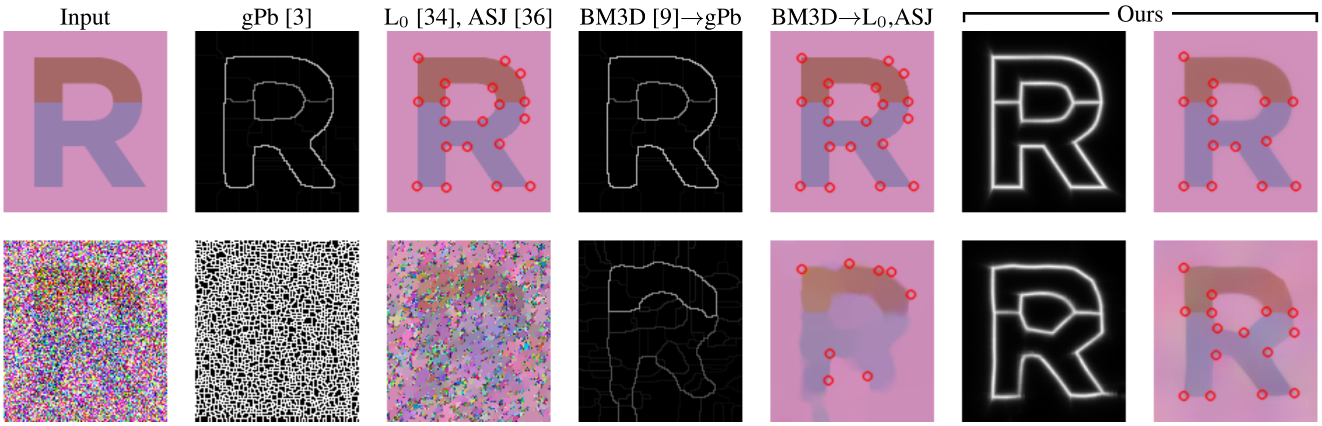

Figure 2: Interpreting boundary structure at high and low SNR (top and bottom). The field of junctions identifies contours (column 6), corners/junctions (circles, column 7) and smooth colors (column 7). It is more resilient to noise than previous methods that are specific to contours, junctions or smoothing, even when they are preceded by optimally-tuned denoising.

junction of any degree up to M (see Figure 3). The model encourages consistency between overlapping patches using a new form of spatial regularization that, instead of penalizing overall curve length or elastica, expresses preference for global boundary maps comprising isolated corners and junctions that are connected by contours with small curvature. As far as we know, this is the first time such regularization has been achieved in the presence of junctions.

An image is analyzed by solving a non-convex optimization problem that cooperatively determines $M+2$ junction values at every location. This produces a *field of junctions*: a distilled representation of the contours, corners, junctions and homogeneous regions of an image. It is an intermediate representation that is useful for a variety of tasks, including contour detection, junction/keypoint detection, and boundary-aware smoothing.

Experimentally, the field of junctions provides unprecedented resilience to noise. It is repeatable over a wide range of noise levels, including very high noise regimes where other approaches—whether based on denoising, segmentation, contour detection, or junction detection—all tend to fail (see Figures 1 & 2). We attribute this to the form of its regularization and to its unified representation of contours, corners, junctions and uniformity, which allows all of these signals to mutually excite and inhibit during analysis.

We introduce the field of junctions model in Section 3, where we formulate analysis as a non-convex optimization problem. We describe how the model can be used for both single-channel and multi-channel images, and how it includes a parameter controlling the scale of its output. The following Section 4 is the heart of the paper: It introduces the optimization techniques that allow analysis to succeed. In particular, we present a greedy algorithm for initializing each patch's junction parameters that has convergence guarantees under certain conditions, and is very effective in practice even when they do not hold. In Section 5 we apply the field of junctions to contour, corner, and junction detection, showing that it provides novel regularization capabilities and repeatable performance across many noise levels. Extended versions of our figures, generalizations of the model, additional results, and a video summary of our paper, are all available in the supplement.

2. Related Work

Contour, corner and junction detection. These have been studied for decades, often separately, using halved receptive fields to localize contours [6, 16, 21] and wedges or other patch-based models for corners and junctions [12, 26, 10, 20, 7, 32, 36]. The drawback of separating these processes is that, unlike our model, it does not exploit concurrency between contours, corners and junctions at detection time.

Contour detection at low SNR. The naive way to detect contours at low SNR is to precede a contour detector by a strong generic denoiser. Ofir *et al*. [24, 23] were perhaps the first to convincingly show that better results can be achieved by designing optimization strategies that specifically exploit the regularity of contours (also see Figure 1). We build on this idea by developing different optimization schemes that handle a broader set of boundary structures and that improve upon [24, 23] in both accuracy and scalability.

Curvature regularization. Boundaries extracted at low SNR are strongly influenced by the choice of regularization. Prior work has shown that minimizing curvature—either alone or in combination with length (Euler's elastica)—generally does better at preserving elongated structures and fine details than minimizing length alone; and there have been many attempts to invent good numerical schemes for minimizing boundary curvature [28, 22, 39, 37, 30, 13]. All of these methods lead to rounded corners, and more critically, they only apply to boundaries between two regions so provide no means for preserving junctions (see Figure 4). In contrast, our model preserves sharp corners and junctions while also reducing curvature along contours.

Segmentation. Our patch model is inspired by the level-set method of Chan and Vese [8] and in particular its multi-phase generalizations [31, 14]. In fact, our descent strategy in Section 4.2 can be interpreted as pursuing optimal level-set functions in each patch, with each patch's functions constrained to a continuous $(M + 2)$-parameter family. Our experiments show that our regularized patch-wise approach obviates the needs for manual initialization and re-initializing during optimization, both of which have been frequent requirements in practice [8, 31, 14, 19].

Boundary-aware smoothing. When locating boundaries, our model infers the regional colors adjacent to each boundary point and so provides boundary-aware smoothing as a by-product. It is not competitive with the efficiency of dedicated smoothers [34, 25, 11] but is more resilient to noise.

Deep encoder/decoder networks. Our approach is very different from relying on deep CNNs to infer the locations of boundaries (*e.g.*, [33, 29]) or lines and junctions [15, 38, 35]. CNNs have an advantage of being trainable over large datasets, allowing both local and non-local patterns to be internalized and exploited for prediction; but there are ongoing challenges related to overcoming their internal spatial subsampling (which makes boundaries hard to localize) and their limited interpretability (which makes it hard to adapt to radically new situations). Unlike CNNs, the field of junctions model does not have capacity to maximally exploit the intricacies of a particular dataset or imaging modality. But it has the advantages of: not being subsampled; interpreting boundary structure into component contours, corners and junctions; applying to many noise levels and many single-channel or multi-channel 2D imaging modalities; and being controlled by just a few intuitive parameters.

3. Field of Junctions

From a K-channel image $I : \Omega \to \mathbb{R}^K$ with 2D support Ω, we extract dense, overlapping $R \times R$ spatial patches, denoted $\mathcal{I}_R = \{I_i(\mathbf{x})\}_{i=1}^N$. We also define a continuous family of patch-types, $\mathcal{P}_R = \{\mathbf{u}_{\boldsymbol{\theta}}(\mathbf{x})\}$, parametrized by $\boldsymbol{\theta}$, describing the boundary structure in an $R \times R$ patch. For $\mathcal{P}_R$ we use the family of *generalized M-junctions*, comprising M angular wedges around a vertex. The parameters $\boldsymbol{\theta} = (\boldsymbol{\phi}, \mathbf{x}^{(0)}) \in \mathbb{R}^{M+2}$ are M angles $\boldsymbol{\phi} = (\phi^{(1)}, ..., \phi^{(M)})$ and vertex position $\mathbf{x}^{(0)} = (x^{(0)}, y^{(0)})$. Importantly, the vertex can be inside or outside of the patch, and wedges may have size 0. Figure 3 shows examples for $M = 3$.

Assume all image patches $\mathcal{I}_R$ are described by patches from $\mathcal{P}_R$ with additive white Gaussian noise. This means that for every $i \in \{1, ..., N\}$ there exist parameters $\boldsymbol{\theta}_i$, and

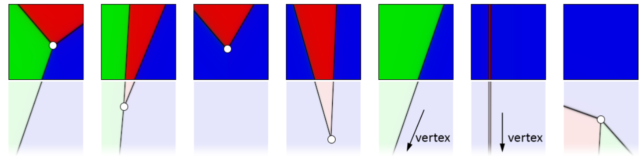

Figure 3: A generalized M-junction comprises a vertex and M angles, partitioning each patch into at most M uniform regions (here, $M = 3$). By freeing the vertex to be variously inside or outside of patches as needed, the model simultaneously accommodates contours, lines, corners, junctions, and uniform regions, thereby allowing concurrencies between all of them to be exploited during analysis.

M color functions $c_i^{(1)}, ..., c_i^{(M)} : \Omega_i \to \mathbb{R}^K$ (to be defined momentarily), such that for all $\mathbf{x} \in \Omega_i$:

$$I_i(\mathbf{x}) = \sum_{j=1}^{M} u_{\boldsymbol{\theta}_i}^{(j)}(\mathbf{x}) c_i^{(j)}(\mathbf{x}) + n_i(\mathbf{x}), \tag{1}$$

where $n_i(\mathbf{x}) \sim \mathcal{N}(0, \sigma^2)$ is noise, and $u_{\boldsymbol{\theta}_i}^{(j)} : \Omega_i \to \{0, 1\}$ is an indicator function that returns 1 if $\mathbf{x}$ is inside the jth wedge defined by $\boldsymbol{\theta}_i$ and 0 otherwise.

Each color function $c_i^{(j)}$ is defined over the support of the ith patch Ω_i and explains the continuous field of K-channel values within the jth wedge of that patch. These functions are constrained to a pre-chosen family of functions $\mathcal{C}$, such as constant functions $\mathcal{C} = \{c(\mathbf{x}) \equiv c : c \in \mathbb{R}^K\}$ or linear functions $\mathcal{C} = \{c(\mathbf{x}) = A\mathbf{x} + b : A \in \mathbb{R}^{K \times 2}, b \in \mathbb{R}^K\}$.

We write the process of analyzing an image into its field of junctions as solving the optimization problem:

$$\max_{\Theta, \mathrm{C}} \log p(\Theta) + \log p(\mathrm{C}) + \sum_{i=1}^{N} \log p(I_i | \boldsymbol{\theta}_i, \mathbf{c}_i), \tag{2}$$

where $p(\Theta)$ and $p(\mathrm{C})$ are spatial consistency terms over all junction parameters $\Theta = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N)$ and color functions $\mathrm{C} = (\mathbf{c}_1, ..., \mathbf{c}_N)$ respectively, and $p(I_i | \boldsymbol{\theta}_i, \mathbf{c}_i)$ is the likelihood of a patch I_i given the junction parameters $\boldsymbol{\theta}_i$ and color functions $\mathbf{c}_i = (c_i^{(1)}, ..., c_i^{(M)})$. If the consistency terms $p(\Theta)$ and $p(\mathrm{C})$ are 0 whenever overlapping patches disagree within their overlap, this objective is precisely the MAP estimate of the field of junctions, where the consistency terms are interpreted as priors over junction parameters and color functions, which we model as independent.

In the remainder of this section we provide more information about the three terms in Equation 2. For simplicity we use $M = 3$ and a constant color model $c_i^{(j)}(\mathbf{x}) \equiv c_i^{(j)}$, but expansions to higher-order color models and to $M > 3$ are trivial and described in the supplement. The supplement also shows how the model performs when noise is not spatially-independent as is assumed in Equation 1.

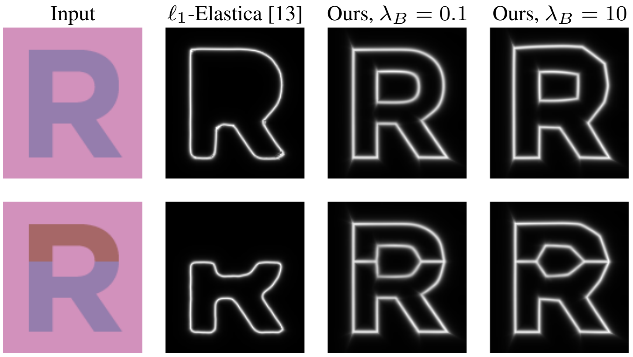

Figure 4: Our boundary consistency term, governed by λ_B, favors isolated corners and junctions connected by contours with low curvature. Unlike other regularizers, it: is agnostic to contour length and convexity; preserves sharp corners; and preserves junctions between three or more regions.

3.1. Patch Likelihood

For a single patch, Equation 1 directly shows that the log-likelihood term is negatively proportional to the mean squared error in that patch:

$$\log p(I_i|\boldsymbol{\theta}_i, \mathbf{c}_i) = -\alpha \sum_{j=1}^{M} \int u_{\boldsymbol{\theta}_i}^{(j)}(\mathbf{x}) \left\| c_i^{(j)} - I_i(\mathbf{x}) \right\|^2 d\mathbf{x}, \quad (3)$$

where $\alpha > 0$ is a constant determined by the noise level σ.

The likelihood term in Equation 3 can be treated as a function of the junction parameters at a single location, $\boldsymbol{\theta}_i$, because finding the optimal colors $\mathbf{c}_i$ is trivial for a given $\boldsymbol{\theta}_i$ (see Equation 10 in Section 3.2). However, despite the low dimensionality of the problem, which requires estimating an $(M+2)$-dimensional junction parameter per patch, solving it efficiently is a substantial challenge. We present an efficient solution to this problem in Section 4.1.

3.2. Spatial Consistency

Our spatial consistency terms $p(\Theta)$ and $p(\mathbf{C})$ require that all junction models agree within their overlap. The boundary consistency can be succinctly written as a constraint on the boundaries defined by each junction:

$$\log p(\Theta) = \begin{cases} 0 & \text{if } B_i(\mathbf{x}) = \hat{B}(\mathbf{x}) \text{ for all i} \\ -\infty & \text{otherwise} \end{cases}, \quad (4)$$

where $B_i(\mathbf{x})$ is the *boundary map* at the ith patch that returns 1 if $\mathbf{x}$ is a boundary location according to $\boldsymbol{\theta}_i$ and 0 otherwise, and $\hat{B}(\mathbf{x}) = \max_{i \in \{1,\ldots,N\}} B_i(\mathbf{x})$ is the *global boundary map* defined by the field of junctions.

The boundary consistency term in Equation 4 provides a hard constraint on the junction parameters, which is difficult to use in practice. We instead replace it with a relaxed, finite version having width δ and strength β_B:

$$\log p(\Theta) = -\beta_B \sum_{i=1}^{N} \int \left[B_i^{(\delta)}(\mathbf{x}) - \hat{B}_i^{(\delta)}(\mathbf{x}) \right]^2 d\mathbf{x}, \quad (5)$$

where $B_i^{(\delta)}(\mathbf{x})$ is a smooth boundary map with dropoff width δ from the exact boundary position, to be defined precisely in Section 4.2. The relaxed global boundary map $\hat{B}^{(\delta)}(\mathbf{x})$ is now computed by taking the mean (rather than maximum) of the smooth local boundary map at each position $\mathbf{x}$ over all patches containing it:

$$\hat{B}^{(\delta)}(\mathbf{x}) = \frac{1}{|N_{\mathbf{x}}|} \sum_{i \in N_{\mathbf{x}}} B_i^{(\delta)}(\mathbf{x}), \quad (6)$$

where $N_{\mathbf{x}} = \{i \colon \mathbf{x} \in \Omega_i\}$ is the set of indices of patches that contain $\mathbf{x}$. We denote by $\hat{B}_i^{(\delta)}(\mathbf{x})$ the ith patch of the relaxed global boundary map in Equation 6. Note that the relaxed consistency in Equation 5 approaches the strict one from Equation 4 when $\delta \to 0$ and $\beta_B \to \infty$.

Similar to the boundary spatial consistency term, we define the color spatial consistency term as:

$$\log p(\mathbf{C}) = -\beta_C \sum_{i=1}^{N} \sum_{j=1}^{M} \int u_{\boldsymbol{\theta}_i}^{(j)}(\mathbf{x}) \left\| c_i^{(j)} - \hat{I}_i(\mathbf{x}) \right\|^2 d\mathbf{x}, \quad (7)$$

where $\hat{I}_i(\mathbf{x})$ is the ith patch of the *global color map*:

$$\hat{I}(\mathbf{x}) = \frac{1}{|N_{\mathbf{x}}|} \sum_{i \in N_{\mathbf{x}}} \sum_{j=1}^{M} u_{\boldsymbol{\theta}_i}^{(j)}(\mathbf{x}) c_i^{(j)}. \quad (8)$$

Using the expressions for the log-likelihood and the relaxed consistency in Equations 3, 5, and 7, analyzing an image into its field of junctions can now be written as the solution to the following minimization problem:

$$\begin{aligned} \min_{\Theta, \mathbf{C}} & \sum_{i=1}^{N} \sum_{j=1}^{M} \int u_{\boldsymbol{\theta}_i}^{(j)}(\mathbf{x}) \left\| c_i^{(j)} - I_i(\mathbf{x}) \right\|^2 d\mathbf{x} \\ & + \lambda_B \sum_{i=1}^{N} \int \left[B_i^{(\delta)}(\mathbf{x}) - \hat{B}_i^{(\delta)}(\mathbf{x}) \right]^2 d\mathbf{x}, \\ & + \lambda_C \sum_{i=1}^{N} \sum_{j=1}^{M} \int u_{\boldsymbol{\theta}_i}^{(j)}(\mathbf{x}) \left\| c_i^{(j)} - \hat{I}_i(\mathbf{x}) \right\|^2 d\mathbf{x}, \end{aligned} \quad (9)$$

where $\lambda_B = \beta_B / \alpha$ and $\lambda_C = \beta_C / \alpha$ are parameters controlling the strength of the boundary and color consistency.

We solve Problem (9) by alternation, updating junction parameters and colors $(\Theta, \mathbf{C})$ while global maps $(\hat{B}^{(\delta)}, \hat{I})$ are fixed, and then updating the global maps. This takes advantage of closed-form expressions for the optimal colors. For the constant color model the expression is

$$c_i^{(j)} = \frac{\int u_{\boldsymbol{\theta}_i}^{(j)}(\mathbf{x}) \left[I_i(\mathbf{x}) + \lambda_C \hat{I}_i(\mathbf{x}) \right] d\mathbf{x}}{(1 + \lambda_C) \int u_{\boldsymbol{\theta}_i}^{(j)}(\mathbf{x}) d\mathbf{x}}, \quad (10)$$

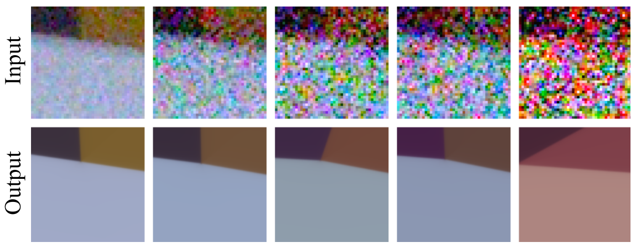

Figure 5: Output of Algorithm 2 for a patch in SIDD [2] captured at decreasing light levels. Algorithm convergence is only guaranteed when noise is absent, but output is quite accurate in practice even when noise is high.

and for piecewise-linear colors, *i.e.*, $c_i^{(j)}(\mathbf{x}) = A_i^{(j)}\mathbf{x} + b_i^{(j)}$, there is a similar expression that replaces each of the K divisions with a 3×3 matrix inversion and multiplication.

Our formulation of boundary consistency encourages each patch i to agree with its overlapping neighbors, by inhibiting its own boundariness $B_i^{(\delta)}(\mathbf{x})$ at pixels $\mathbf{x}$ that are assigned a low score by their neighbors (as quantified by $\hat{B}^{(\delta)}(\mathbf{x})$) and exciting its boundariness at pixels assigned a high score. This means only salient junctions, corners, and contours end up contributing to the final global boundary map $\hat{B}^{(\delta)}(\mathbf{x})$. Junction values in uniform patches and other less salient patches tend to disagree with other patches, so spurious boundaries within them are suppressed.

At the same time, our use of a smoothed version of consistency instead of a strict one allows for contours having nonzero curvature to be well approximated by local collections of corners that have slightly different vertices, while incurring a penalty. This has the effect of a curvature regularizer, because the only way for all junctions in the field to exactly agree is when the global boundary has zero curvature everywhere except at a finite number of vertices spaced at least ℓ_∞-distance R apart (*e.g.* a polygon).

The color consistency term of our objective promotes agreement on color between overlapping patches. It improves the results of the field of junctions under high noise by enforcing long-range consistency between the colors of sets of pixels not separated by a boundary.

4. Analysis

Analyzing an image into its field of junctions is a challenge, with Problem (9) consisting of N junction-fitting problems that are coupled by spatial consistency terms. Even without consistency, finding the optimal junction for a single patch i requires minimizing a non-smooth and non-convex function in $\boldsymbol{\theta}_i$.

We solve the problem in two parts: initialization and refinement. Both of these are key to our model's robustness to noise. The initialization procedure independently optimizes each patch, using a handful of coordinate updates to find discrete values for its angles and vertex location. Then, the refinement procedure performs gradient descent on a relaxation of Problem (9), cooperatively adjusting all junction parameters to find continuous angles and sub-pixel vertex locations that improve spatial consistency while maintaining fidelity to local appearance. We next describe each step.

4.1. Initialization

Many previous methods for junction estimation, such as [10, 7], use gradient descent to optimize the vertex and angles of a single wedge model. These methods rely on having a good initialization from a human or a corner detector, and they fail when such initializations are unavailable. Indeed, even in the noiseless case, there always exists an initialization of a patch's junction parameters around which the negative log-likelihood is locally constant.

In the present case, we need an initialization strategy that is automatic and reliable for *every* patch, or at least the vast majority of them. We first describe an initialization algorithm for the simpler problem in which the vertex of a patch is known, where our algorithm guarantees optimality in the absence of noise; and then we expand it to solve for the vertex and angles together.

When the vertex is known, optimizing the parameters of one patch reduces to finding a piecewise-constant, one-dimensional angular function. There are algorithms for this based on dynamic programming [4, 17] and heuristic particle swarm optimization [5]. We instead propose Algorithm 1, which is guaranteed to find the true junction angles $\boldsymbol{\phi} = (\phi^{(1)}, ..., \phi^{(M)})$ that minimize the negative log-likelihood $\ell(\boldsymbol{\phi}, \mathbf{x}^{(0)}) = -\log p(I_i|\boldsymbol{\theta}, \mathbf{c}_i)$ in the noiseless case. The algorithm consists of a single coordinate-descent update over the M junction angles, that is, it minimizes $\ell_j(\phi) \triangleq \ell(\phi^{(1)}, ..., \phi^{(j-1)}, \phi, \phi^{(j+1)}, ..., \phi^{(M)}, x^{(0)}, y^{(0)})$ for $j = 1, ..., M$.

Algorithm 1: Optimization of angles

```
Initialize φ^(1), ..., φ^(M) ← 0.
for j = 1, ..., M do
    φ^(j) ← argmin_φ ℓ_j(φ)
end
```

Theorem 1. *For a junction image $I_i(\mathbf{x})$ with no noise (i.e., $n_i \equiv 0$ in Eq. 1) and with vertex $\mathbf{x}^{(0)}$ known, Algorithm 1 is guaranteed to find the globally optimal angles $\boldsymbol{\phi}$.*

Proof Sketch. (See full proof in supplement.) First, note that $\ell_j(\phi)$ is continuous and smooth for all ϕ other than possibly a discontinuity in the derivative at any of the true junction angles. If the optimal ϕ is not one of the true junction angles then it must lie in the *open* interval between two

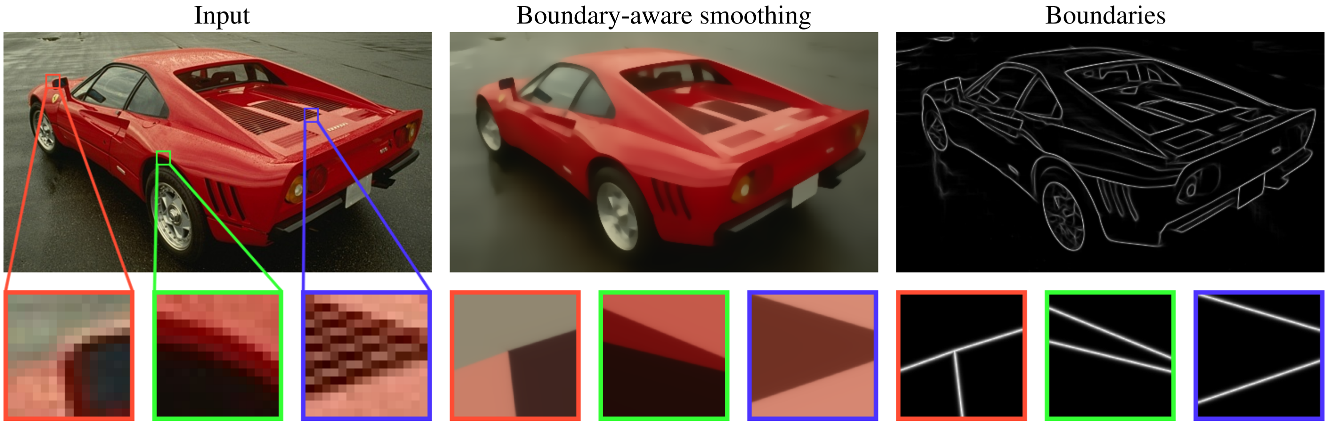

Figure 6: Field of junctions from a photograph. It can extract boundary-aware smoothing and boundary structure from natural images because it is robust to texture and other natural deviations from the ideal generalized M-junction model.

such angles, *i.e.* $\phi \in (\phi^-, \phi^+)$. It can be shown that $\ell_j(\phi)$ does not have any local minima in (ϕ^-, ϕ^+), and therefore for each angular interval between two true junction angles the cost function must be minimized at one of the endpoints. Therefore repeatedly minimizing $\ell_j(\phi)$ for $j = 1, ..., M$ is guaranteed to provide a globally optimal set of angles. □

In practice, we find that Algorithm 1 provides an excellent estimate of the true junction angles even when the input patch is noisy. It also has a significant efficiency advantage. Each coordinate update can be done to an arbitrarily small error ε with complexity $O(1/\varepsilon)$, by exhaustively searching over all angles in increments of ε. The complexity for a single junction is therefore $O(M/\varepsilon)$, in contrast with the $O(1/\varepsilon^2)$ dynamic programming solution of [17] and the $O(1/\varepsilon^M)$ of naive exhaustive search over all possible M-angle sets. Moreover, each step of the algorithm can be run in parallel over all angles (and over all patches) by computing the value of $\ell_j(\phi^{(j)})$ for each of the $O(1/\varepsilon)$ values and choosing the minimizing angle. Thus, runtime can be accelerated significantly using a GPU or multiple processors.

These efficiency advantages become especially important when we expand the problem to optimize the vertex in addition to the angles. We simply do this by initializing the vertex at the center of the patch and updating it along with the angles using a coordinate descent procedure. See Algorithm 2. Figure 5 shows a typical example, where the algorithm results in a good estimate of the true vertex position and angles despite a substantial amount of noise.

Algorithm 2: Optimization of angles and vertex

Initialize $x^{(0)}, y^{(0)}$ at the center of the patch.
for $i = 1, ..., N_{init}$ **do**
  Find angles $\boldsymbol{\phi}$ using Algorithm 1.
  $x^{(0)} \leftarrow \operatorname{argmin}_x \ell(\boldsymbol{\phi}, x, y^{(0)})$
  $y^{(0)} \leftarrow \operatorname{argmin}_y \ell(\boldsymbol{\phi}, x^{(0)}, y)$
end

4.2. Refinement

After initializing each patch separately, we refine the field of junctions using continuous, gradient-based optimization. In order to compute the gradient of the objective in Problem (9) with respect to Θ we relax the indicator functions $\{\mathbf{u}_{\boldsymbol{\theta}}(\mathbf{x})\}$, making them smooth in $\mathbf{x}$ and in $\boldsymbol{\theta}$, similar to level-set methods [8, 31]. We do this by describing each 3-junction using two distance functions (a similar parametrization exists using $M-1$ functions for M-junctions). Given the vertex position $(x^{(0)}, y^{(0)})$ and angles $\phi^{(1)}, \phi^{(2)}, \phi^{(3)}$, and assuming without loss of generality that $0 \leq \phi^{(1)} \leq \phi^{(2)} \leq \phi^{(3)} < 2\pi$, we define a junction using two signed distance functions d_{12} and d_{13} defined by:

$$d_{kl}(\mathbf{x}) = \begin{cases} \min\{d_k(\mathbf{x}), -d_l(\mathbf{x})\} & \text{if } \phi^{(l)} - \phi^{(k)} < \pi \\ \max\{d_k(\mathbf{x}), -d_l(\mathbf{x})\} & \text{otherwise} \end{cases} \tag{11}$$

where $d_l(x, y) = -(x - x^{(0)})\sin(\phi^{(l)}) + (y - y^{(0)})\cos(\phi^{(l)})$ is the signed distance function from a line with angle $\phi^{(l)}$ passing through $(x^{(0)}, y^{(0)})$.

Our relaxed indicator functions are defined as:

$$\begin{aligned} u_{\boldsymbol{\theta}}^{(1)}(\mathbf{x}) &= 1 - H_\eta(d_{13}(\mathbf{x})), \\ u_{\boldsymbol{\theta}}^{(2)}(\mathbf{x}) &= H_\eta(d_{13}(\mathbf{x}))\left[1 - H_\eta(d_{12}(\mathbf{x}))\right], \\ u_{\boldsymbol{\theta}}^{(3)}(\mathbf{x}) &= H_\eta(d_{13}(\mathbf{x}))H_\eta(d_{12}(\mathbf{x})), \end{aligned} \tag{12}$$

where H_η is the regularized Heaviside function, as in [8]:

$$H_\eta(d) = \frac{1}{2}\left[1 + \frac{2}{\pi}\arctan\left(\frac{d}{\eta}\right)\right]. \tag{13}$$

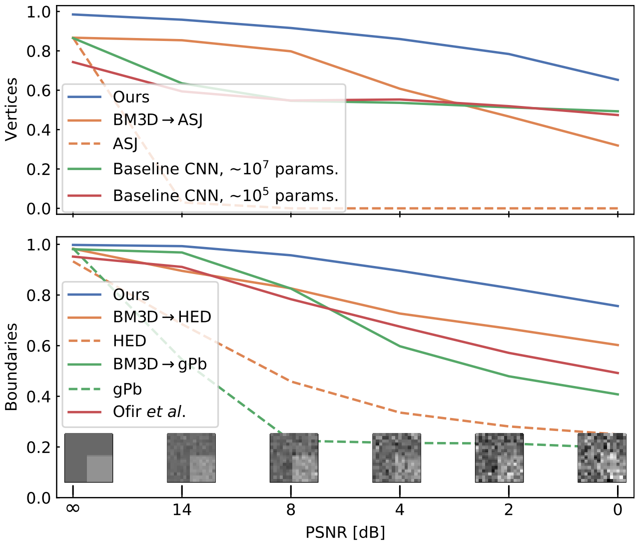

Figure 7: Vertex and boundary detection F-score for increasing noise on our dataset. At low noise our model is comparable to existing edge and junction detectors and a baseline CNN, but it significantly outperforms them at high noise, even when preprocessed by BM3D. Insets: sample patch at different noise levels.

The smooth boundary maps for the consistency term are:

$$B_i^{(\delta)}(\mathbf{x}) = \pi\delta \cdot H'_\delta\left(\min\{|d_{12}(\mathbf{x})|, |d_{13}(\mathbf{x})|\}\right), \qquad (14)$$

where $H'_\delta(d)$ is the derivative of $H_\delta(d)$ with respect to d, and the scaling factor ensures that $0 \leq B_i^{(\delta)}(\mathbf{x}) \leq 1$.

Our experiments use $\eta = 0.01$ and $\delta = 0.1$. We find that the algorithm is fairly insensitive to these values, and that varying them does not provide useful control of the model's behavior. This is in contrast to the other parameters—patch size R and consistency weights λ_B, λ_C—that control scale and level of boundary and color detail.

4.3. Optimization Details

We analyze an image into its field of junctions by first initializing with Algorithm 2 for $N_{\text{init}} = 30$ iterations, followed by refinement to minimize Problem (9) using the Adam optimizer [18] for $N_{\text{iter}} = 1000$ iterations. Initialization is performed by evaluating the restricted negative log-likelihood functions in Algorithms 1 and 2 at 100 evenly-spaced values. Because the vertex of a junction can be outside its patch (see Figure 3), each of its two coordinates is searched over an interval of length $3R$ around the center of each patch. The accuracy of our initialization is thus $3.6°$ in the junction angles, and $0.03R$ in the vertex position.

For the refinement step we use a learning rate of 0.03 for the vertex positions and 0.003 for the junction angles, and the global maps $\hat{B}^{(\delta)}(\mathbf{x})$ and $\hat{I}(\mathbf{x})$ are treated as constants computed using the values of the previous iteration when computing gradients. In order to allow the parameters to first improve their estimates locally and only then use the consistency term to improve the field of junctions, we linearly increase the consistency weights from 0 to their final values λ_B and λ_C over the 1000 refinement iterations. We additionally apply Algorithm 2 (without reinitializing the junction parameters) once every 50 refinement iterations, which we find helps our method avoid getting trapped in local minima. The runtime of our algorithm on an NVIDIA Tesla V100 GPU is 110 seconds for a 192×192 image with patch size $R = 21$, but both runtime and space usage can be significantly reduced by only considering every sth patch in both spatial dimensions for some constant stride s (see supplement for the effect of s on runtime and performance). We implemented our algorithm in PyTorch, and our code and datasets are available on our project page [1].

5. Experiments

Once an image is analyzed, its field of junctions provides a distributional representation of boundary structure and smooth regional appearance. Each pixel in the field provides a "vote" for a nearby (sub-pixel) vertex location with associated wedge angles and color values around that location. Simple pixel-wise averages derived from the field are useful for extracting contours, corners and junctions, and boundary-aware smoothing. We demonstrate these uses here, and we compare our model's regularization to previous methods for curvature minimization.

We evaluate performance using three types of data. First, we show qualitative results on captured photographs. Second, we quantify repeatability using the Smartphone Image Denoising Dataset (SIDD) [2], evaluating the consistency of extracted boundaries when the same scene is photographed at decreasing light levels (and thus increasing noise levels). Finally, to precisely quantify the accuracy of extracted contours, corners and junctions, we generate a dataset of 300 synthetic grayscale images (shown in the supplement) with boundary elements known to sub-pixel precision, and with carefully controlled noise levels. In this section we provide results using uncorrelated noise, and our supplement contains results on images corrupted by other noise models.

Boundary-aware smoothing. A field of junctions readily provides a boundary-aware smoothing using Equation 8. An example for a photograph is shown in Figure 6, and a comparison of its resilience to noise with that of [34] is shown in Figure 2.

Boundary Detection. A field of junctions also immediately provides a boundary map via Equation 6. Figure 6 shows the resulting boundaries extracted from a photograph, and Figure 1 shows a qualitative comparison of our boundaries to previous edge detection and segmentation methods on a patch extracted from a noisy short-exposure photograph.

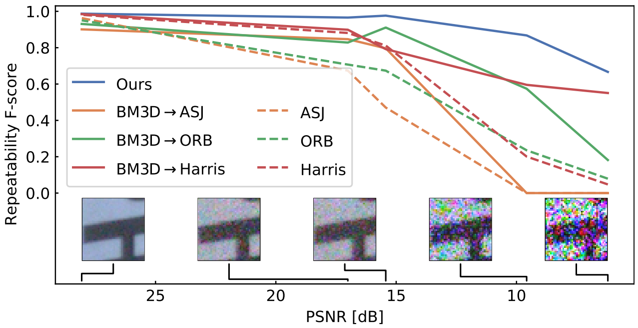

Figure 8: Vertex detection repeatability over increasing noise on patches from SIDD, compared to repeatability of other detectors with and without denoising. The number of points detected by each method on the clean ground truth is 126 (ours), 49 (ASJ), 57 (Harris), and 70 (ORB).

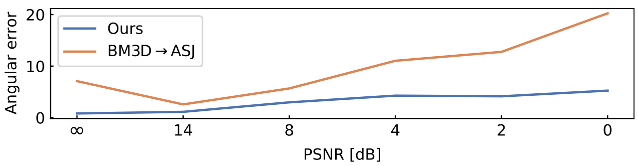

Figure 9: Error of angles (in degrees) at detected junctions on our dataset, for our method and ASJ preprocessed by BM3D. Our method degrades slowly. We reported accuracy for ASJ on correctly-detected junctions only. See Figure 7 for sample patches.

We quantitatively compare our results to existing contour and boundary detection methods: gPb [3], HED [33], Ofir *et al*. [24], and gPb and HED when denoised by BM3D [9] supplied with true noise level σ. (Since [24] is designed for low SNR, so we do not combine it with BM3D.) In Figure 7 we show the F-scores of results obtained by each method on our synthetic dataset. The F-score is computed by matching the boundaries output by each detector with the ground truth and taking the harmonic mean of its precision and recall.

Curvature Regularization. Figure 4 compares our boundary regularization to ℓ_1-elastica [13] with optimally-tuned parameters. This represents the strongest possible comparison across a large family of existing regularizers, because elastica includes pure-length and pure-curvature minimization as special cases, and because minimizing the ℓ_1-norm outperforms the ℓ_2-norm in these images. Unlike existing regularizers, the field of junctions preserves sharp corners; favors linear contours over curved ones; is agnostic to length and convexity of boundaries; and is, as far as we know, the first to do all of this while preserving junctions.

Vertex Detection. A field of junctions also provides a map of vertex locations that can be used like a traditional corner, junction, or interest point detector. To create a vertex map, we use weighted voting from each junction in the field. The likelihood that a vertex exists at location $\mathbf{x}$ is:

$$V(\mathbf{x}) \propto \sum_{i=1}^{N} w_i \kappa\left(\mathbf{x} - \mathbf{x}_i^{(0)}\right), \tag{15}$$

with Gaussian kernel $\kappa(\Delta\mathbf{x}) = \exp\left(-\frac{\|\Delta\mathbf{x}\|^2}{2\gamma^2}\right)$ of width γ, and weights w_i that suppress votes from patches having wedge-angles close to 0° or 180° (*i.e.* with no unique vertex) and from patches with vertex $\mathbf{x}^{(0)}$ very far from the patch center. (See supplement for full expression.)

Figure 2 shows the qualitative results of our vertex detector in the low- and high-noise regime, compared with ASJ [36]. A quantitative study of the robustness of our detector to noise on our synthetic dataset is shown in Figure 7. We again use F-score to compare to ASJ [36] and to BM3D followed by ASJ, and to baseline CNNs that we trained on our dataset specifically for vertex detection. In this experiment, a separate CNN was trained for each PSNR. Figure 8 shows the repeatability of our vertex detector over different noise levels using patches extracted from SIDD, compared to ASJ [36], Harris [12], and ORB [27]. The repeatability F-scores are computed by comparing the points obtained by each method on the noisy images with its output on the noiseless ground truth images. In all cases we find that our model provides superior resilience to noise. Our detector also provides repeatability over change in viewpoint angle similar to other interest point detectors (see supplement).

In addition to the vertex locations, a field of junctions provides an estimate of the angles of each detected vertex. We treat ϕ_i as an estimate for the angles at a pixel i. Figure 9 shows a comparison of this angle estimation accuracy over multiple noise levels with ASJ preprocessed by BM3D. Because ASJ alone fails at moderate noise levels (see Figure 7), we only plot the results of BM3D followed by ASJ.

6. Limitations

The field of junctions is governed by just a few parameters, so compared to deep CNNs it has much less capacity to specialize to non-local patterns of boundary shape and appearance that exist in a particular dataset or imaging modality. Also, as currently designed, it analyzes images at only one scale at a time, with R determining the minimum separation between vertices in the output at that scale. Finally, while the analysis algorithm scales well with image size ($O(N)$, compared to the $O(N^{1.5})$ and $O(N \log N)$ algorithms of [24, 23]) and has runtime comparable to some other analyzers like gPb, it is slower than feedforward CNNs and dedicated smoothers and contour/corner detectors that are engineered for speed on high-SNR images.

Acknowledgements. This work is supported by the National Science Foundation under Cooperative Agreement PHY-2019786 (an NSF AI Institute, http://iaifi.org).

References

[1] Project page: Field of junctions: extracting boundary structure at low SNR. http://vision.seas.harvard.edu/foj/. 7

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018. 5, 7

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 8

[4] Ivan E Auger and Charles E Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of mathematical biology*, 51(1):39–54, 1989. 5

[5] Leif Bergerhoff, Joachim Weickert, and Yehuda Dar. Algorithms for piecewise constant signal approximations. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019. 5

[6] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 2

[7] Miguel A Cazorla and Francisco Escolano. Two Bayesian methods for junction classification. *IEEE Transactions on Image Processing*, 12(3):317–327, 2003. 2, 5

[8] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001. 3, 6

[9] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 8

[10] Rachid Deriche and Thierry Blaszka. Recovering and characterizing image features using an efficient model based approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 530–535. IEEE, 1993. 2, 5

[11] Eduardo SL Gastal and Manuel M Oliveira. Domain transform for edge-aware image and video processing. *ACM Transactions on Graphics (TOG)*, 2011. 3

[12] Christopher G Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 2, 8

[13] Xuan He, Wei Zhu, and Xue-Cheng Tai. Segmentation by elastica energy with L1 and L2 curvatures: a performance comparison. *Numerical Mathematics: Theory, Methods and Applications*, 12, 06 2019. 2, 8

[14] Erlend Hodneland, Xue-Cheng Tai, and Hans-Hermann Gerdes. Four-color theorem and level set methods for watershed segmentation. *International Journal of Computer Vision*, 82(3):264–283, 2009. 3

[15] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 626–635, 2018. 3

[16] Lee A. Iverson and Steven W. Zucker. Logical/linear operators for image curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):982–996, 1995. 2

[17] Brad Jackson, Jeffrey D Scargle, David Barnes, Sundararajan Arabhi, Alina Alt, Peter Gioumousis, Elyus Gwin, Paungkaew Sangtrakulcharoen, Linda Tan, and Tun Tao Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2):105–108, 2005. 5, 6

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[19] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D Fox. Level set evolution without re-initialization: a new variational formulation. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 430–436. IEEE, 2005. 3

[20] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[21] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549, 2004. 2

[22] C. Nieuwenhuis, E. Toeppe, L. Gorelick, O. Veksler, and Y. Boykov. Efficient squared curvature. In *Proc. CVPR*, 2014. 2

[23] Nati Ofir, Meirav Galun, Sharon Alpert, Achi Brandt, Boaz Nadler, and Ronen Basri. On detection of faint edges in noisy images. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):894–908, 2019. 2, 8

[24] Nati Ofir, Meirav Galun, Boaz Nadler, and Ronen Basri. Fast detection of curved edges at low SNR. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 213–221, 2016. 2, 8

[25] Sylvain Paris and Frédo Durand. A fast approximation of the bilateral filter using a signal processing approach. *International journal of computer vision*, 81(1):24–52, 2009. 3

[26] Karl Rohr. Recognizing corners by fitting parametric models. *International journal of computer vision*, 9(3):213–230, 1992. 2

[27] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 8

[28] Thomas Schoenemann, Fredrik Kahl, Simon Masnou, and Daniel Cremers. A linear framework for region-based image segmentation and inpainting involving curvature penalization. *International Journal of Computer Vision*, 99(1):53–68, 2012. 2

[29] Xavier Soria, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust CNN model for edge detection. In *The IEEE Winter Conference on Applications of Computer Vision (WACV '20)*, 2020. 3

[30] Xue-Cheng Tai, Jooyoung Hahn, and Ginmo Jason Chung. A fast algorithm for Euler's elastica model using augmented

lagrangian method. *SIAM Journal on Imaging Sciences*, 4(1):313–344, 2011. 2

[31] Luminita A Vese and Tony F Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International journal of computer vision*, 50(3):271–293, 2002. 3, 6

[32] Gui-Song Xia, Julie Delon, and Yann Gousseau. Accurate junction detection and characterization in natural images. *International journal of computer vision*, 106(1):31–56, 2014. 2

[33] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. ICCV*, 2015. 3, 8

[34] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via L_0 gradient minimization. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–12, 2011. 3, 7

[35] Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu, and Liangpei Zhang. Learning attraction field representation for robust line segment detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1595–1603, 2019. 3

[36] Nan Xue, Gui-Song Xia, Xiang Bai, Liangpei Zhang, and Weiming Shen. Anisotropic-scale junction detection and matching for indoor images. *IEEE Transactions on Image Processing*, 27(1):78–91, 2017. 2, 8

[37] Qiuxiang Zhong, Yutong Li, Yijie Yang, and Yuping Duan. Minimizing discrete total curvature for image processing. In *Proc. CVPR*, 2020. 2

[38] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 962–971, 2019. 3

[39] Wei Zhu and Tony Chan. Image denoising using mean curvature of image surface. *SIAM Journal on Imaging Sciences*, 5(1):1–32, 2012. 2