Thin structures retrieval using anisotropic neighborhoods of superpixels: Application to Shape-From-Focus

Christophe Ribal, S. Le Hégarat-Mascle, Nicolas Lermé

Statements and Declarations

- **Funding:** The authors did not receive support from any organization for the submitted work.

- **Conflict of interest:** The authors have no relevant financial or non-financial interests to disclose.
- **Data availability:** The datasets generated during and/or analysed during the current study are available at https://vision.middlebury.edu/stereo/data/.
- **Authors contribution:** All authors equally contributed to the writing and the reviewing of this paper. All authors approved the current version of this paper.

Thin Structures Retrieval Using Anisotropic Neighborhoods of Superpixels: Application to Shape-From-Focus

Christophe Ribal[1], Sylvie Le Hégarat-Mascle[1] and Nicolas Lermé[1]

[1*]Université Paris-Saclay, SATIE Laboratory UMR 8029, Avenue des sciences, Gif-sur-Yvette, 91190, France.

Contributing authors: christophe.ribal@mail.fr; sylvie.le-hegarat@universite-paris-saclay.fr; nicolas.lerme@universite-paris-saclay.fr;

Abstract

Shape-From-Focus (SFF) refers to the challenging inverse problem of recovering the scene depth from a given set of focused images using a static camera. Standard approaches model the interactions between neighboring pixels to get a regularized solution. Nevertheless, isotropic regularization is known to introduce undesired artifacts and to remove early thin structures. These structures have a small size in at least one dimension and are more numerous when considering superpixel preprocessing. This paper addresses the improvement of SFF regularization through the estimation of the presence of such structures and the construction of anisotropic neighborhoods sticking along image edges and proposes a flexible formulation over pixels or superpixels. A thoroughly study comparing different strategies for constructing these neighborhoods in terms of accuracy and running time for the targeted application is provided. Notably, experiments performed on a reference dataset show the overall superiority of the approach, e.g. a decrease of the RMSE value by about **20%**, and its robustness against generated superpixels.

Keywords: Shape-From-Focus, Thin structures regularization, Anisotropic neighborhood, Superpixel, Tensor Voting, RORPO

1 Introduction

For many image processing problems such as image segmentation or reconstruction, low-level information delivered by a single pixel is limited and prone to noise, corrupted data and optic phenomena altering the original image. Therefore, taking into account a statistical relationship between spatially close pixels has been introduced relatively early in image processing by Geman and Geman (1984). A classical way is to model the two-dimensional (2D) field of pixels as a Markov Random Field (MRF). This allows for introducing a prior on the expected solution. Variational approaches provide solutions by combining the prior and conditional probabilistic models into a single parametric functional to be minimized. However, due to the dimensionality of the solution space and depending on the form of the functional, finding a global minimizer often appears as a challenging task. Szeliski et al. (2008) study gives an insight by comparing several minimization algorithms, including graph cuts, on typical vision problems such as image segmentation and image reconstruction. Graph cuts are known to be very competitive both in terms of accuracy (global minimum is theoretically reached for a number of problems when the functional being minimized is convex) and running time (by avoiding stochastic iterative convergence and with an empirical linear complexity in the number of nodes). Concerning the prior term involved in the functional to minimize, standard Total Variation (TV) regularization (in image reconstruction, as in Ribal, Lermé, and Le Hégarat-Mascle (2018)) or Potts regularization (in image segmentation, as in Boykov and Jolly (2001)) using isotropic neighborhoods have been proposed, both being able to provide a fast and exact solution to their respective problem (modulo a quantization for the latter). However, it was established that such regularizations behave poorly on *thin* structures, defined as having very small size in at least one dimension compared to the other one(s). In two dimensional images, thin structures are curves, that extend in one dimension. In 3D, thin structures may also be planes, as long as their thickness can be considered as very small compared to one of their other dimensions. Although thin structures are ubiquitous in a number of applications (vessels, rivers, cracks, etc.), their detection remains very difficult because of their spatial sparsity, their small size and their potential complex geometry. Since these structures essentially consist of discontinuities, standard TV and Potts regularization tend to early remove them as regularization increases and are thus not adapted to handle them correctly, e.g. according to Favaro (2010). Thus, some authors (e.g., Merveille, Naegel, Talbot, and Passat (2019); Ulen, Strandmark, and Kahl (2015) have proposed specific regularization for thin structures depending on the application (low curvature river networks, 3D coronary arteries and vessels in retinal images). However, these approaches are specifically designed for thin structure detection, and not the segmentation of images including both on large structures and thin ones.

In parallel with algorithmic developments, the volume and the diversity of data have greatly increased over the last years. Therefore to reduce the computational burden, superpixel decomposition methods, e.g. Stutz, Hermans, and

Leibe (2018), have been developed for grouping pixels sharing similar radiometric intensities into homogeneous regions, and then drastically reducing the number of elements to process while preserving the geometrical information that is lost with multi-resolution approaches. For instance and specifically for segmentation problem, Arbeláez, Maire, Fowlkes, and Malik (2011); Gould, Fulton, and Koller (2009) grow and merge regions from an initial set of superpixels. Only a few superpixel algorithms offer formal proof of some of their properties (Tang, Fu, and Cao (2012)), while a lot of them widely used only rely on heuristics to produce satisfying superpixels (Achanta et al. (2012)). A major drawback of working with superpixels is therefore that the usual hypothesis of a regular topological lattice is lost, as well as the regularity in size and shape of every lattice element. As a result, image segmentation approaches taking advantage of superpixels must cope with these problems and introduce new methods and spatial relationships. Often, superpixels are considered as neighbors when sharing a common border, e.g. Cui, Xie, Ma, Ren, and Ma (2018); Fulkerson, Vedaldi, and Soatto (2009); Liu, Yu, Yu, and He (2016); Stawiaski and Decencière (2011). The authors of Stawiaski and Decencière (2011) propose to globally minimize a convex energy via graph cuts based on the adjacency graph obtained from the watershed segmentation, where edges connecting two regions are weighted upon the common border length between these regions. Similarly, Cui et al. (2018) proposes to ease the classification of the high-dimensional noisy hyperspectral images by building a weighted graph based on superpixels. In Pei, Chang, and Shen (2014), the authors compute saliency from MRF using the same concept of adjacency, with second-order neighborhood to ease the propagation of information between superpixels. Some other superpixel approaches, such as Giraud, Ta, Bugeau, Coupe, and Papadakis (2017), use patches to analyze the spatial content over a neighboring window and find the nearest matches in a set of reference patches. For instance, the authors of Yu, Guan, and Ji (2015) train a deep Hough forest from a set of superpixel patches in order to detect objects in aerial images.

In this paper we propose a solution for estimating an anisotropic neighborhood. This solution can be applied to several segmentation or reconstruction problems. However, to instantiate it, we focus on the application Shape From Focus (SFF). As described in Nayar and Nakagawa (1994), SFF is a popular method used for inferring the 3D shape of an object from a set of images with varying focus settings. Such an approach only requires one fixed camera with a rather short depth of field and is able to move this camera or to change the focal distance of the optical system. SFF is therefore applicable in many real world applications including industrial inspection, micro manufacturing, robotic control, 3D model reconstruction, medical imaging systems and microscopy. Specifically to superpixel-based approaches, Lai and Leou (2021) propose a multi-focus image fusion approach where superpixels of input focus images are computed, then either classified as “focused” or “defocused” and finally fused to form an all-in-focus image. However, since no regularization process is involved, such an approach is likely to fail in presence of input

images including textureless areas (see Section 2.1). Indeed, naive pixel-level estimates are hampered by the presence of homogeneous surfaces, thus requiring to propagate the information from reliable areas to uncertain ones, while preserving thin structures. Such a regularization process is all the more challenging that the number of labels (i.e., the number of discrete depth values) is important and that structures (and input data) are 3D, cf. Ali, Pruks, and Mahmood (2019). This study allows us to propose two main contributions:

- Firstly, we propose different estimations of anisotropic neighborhoods on an irregular lattice such as the ones provided by superpixel segmentation; for this, we first compute a guidance map, relying on vesselness operators used for the first time (up to our knowledge) for neighborhood orientation estimation, and then, we provide and discuss four original strategies for neighborhood estimation based on different desired neighborhood properties.
- Secondly, we propose SSF based on superpixels, that is also, to our knowledge, an original contribution. It allows us to constraint depth estimation with color information since superpixels are computed on the all-in-focus image as well as to illustrate the benefit of anisotropic neighborhood since SSF represents a sufficiently complex application so that results may depend on the type of considered neighborhood.

The rest of this paper is organized as follows. In Section 2, we specify the considered problem, namely SSF using superpixel segmentation. In Section 3, we detail the proposed path-based constructions of anisotropic neighborhoods, based on a preliminary estimation of local anisotropy and orientation either from Tensor Voting initially proposed by Medioni, Tang, and Lee (2000), or from RORPO proposed by Merveille, Talbot, Najman, and Passat (2018). Section 4 discusses the results and benefits of our approach in a comprehensive comparative study between isotropic and anisotropic neighborhoods both in terms of accuracy and time complexity. Finally, Section 5 draws main conclusions and perspectives.

2 Superpixels-based SFF

2.1 Basics of SFF

The core idea of SFF is that the closer an object is to the object focal plane (i.e., the more it is focused), the more it appears sharp, and the farther an object is from its focal plane, the more it appears blurred. Then, in the absence of regularization (*blind* estimation), SFF relies on a sharpness operator to find the distance where each pixel maximising the sharpness measure, and reconstructs a depth image. Specifically, Nayar and Nakagawa (1994) approximate the sharpness curve (that represents the sharpness values versus the focus parameter values) with a Gaussian model, and interpolates (along the optical axis) the three focus measures centered on the maximum sharpness value to allow for a better depth estimation. Alternative interpolations, either quadratic, cubic or polynomial interpolation Moeller, Benning, Schonlieb, and

Cremers (2015) or Gaussian interpolation Ribal et al. (2018), have also been investigated.

Nevertheless, *blind* depth estimation remains prone to noise and ambiguities since, in homogeneous or poorly textured areas, the measured sharpness will be quite low and unreliable. To overcome this limitation, some authors Ali and Mahmood (2021); Gaganov and Ignatenko (2009); Moeller et al. (2015); Ribal et al. (2018) have reformulated SFF in the variational framework. Indeed, thanks to regularization, the information extracted in the scene areas where SFF is reliable, such as in objects details, contours, is propagated to ambiguous areas, such as homogeneous, overexposed or underexposed regions. However, usual (isotropic) regulation terms are likely to remove thin structures, so we felt the need to propose an anisotropic regularisation, especially as we work with superpixels.

2.2 From pixel level to superpixel one

Let us consider the 3D space defined by an orthonormal basis $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ such that $\mathbf{e}_0$ and $\mathbf{e}_1$ are aligned with the image row and column dimensions and $\mathbf{e}_2$ represents the focus dimension. The set of input image pixels, denoted $\mathcal{P}$, defines a cube in $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ having dimensions $n_{\text{row}} \times n_{\text{col}} \times n_{\text{foc}}$, where n_{row}, n_{col}, and n_{foc} are positive integers. We also assume without loss of generality that these three dimensions are sampled with a unit step even if for focus it implies a simple transformation.

In this study, we will extend SFF variational formulation to superpixel level. However, first the sharpness profiles are computed at pixel level since sharpness evaluation requires best resolution. Specifically, in our case, we consider the sharpness operator introduced in Pertuz, Puig, and Garcia (2013), namely the Summed Modified LAPlacian (SMLAP): $f(p) = \text{SMLAP}(p), \forall p \in \mathcal{P}$. Then, a sharpness profile is a vector gathering the sharpness values varying the focus dimension for a given pair of (row, column) coordinates: In the following, denoting by $.\downarrow$ the projection on $(\mathbf{e}_0, \mathbf{e}_1)$, $\mathbf{f}(p_\downarrow)$ denotes the sharpness profile at any pixel $p_\downarrow \in \mathcal{P}_\downarrow$. Then, the maximum of sharpness is estimated at any $p_\downarrow \in \mathcal{P}_\downarrow$ as $\max_{k \in [\![1, n_{\text{foc}}]\!]} \mathbf{f}_k(p_\downarrow)$ where $\mathbf{f}_k$ denote the k^{th} component of sharpness profile $\mathbf{f}$. From maximum of sharpness, one can estimate the all-in-focus image defined on $\mathcal{P}_\downarrow$. This image allows us to compute the superpixels that have a good sensitivity to the sharp edges of the scene, since in 2D space it picks the pixel that is the "sharpest", i.e. that has the highest contrast with its neighbors. Besides, the all-in-focus image is a color image and superpixels are groups of pixels that are both spatially and feature-wise close (i.e., similar in color) while sticking to object edges ("boundary adherence" property), and usually correspond to sub-parts of objects in the scene. Then, since depth map will be computed at superpixel-level, i.e. one depth estimate per superpixel, it is constrained by the information on which superpixels are based, namely the color in our case.

Let $\mathcal{S} \in (\mathbf{e}_0, \mathbf{e}_1)$ denote the set of superpixels. From $\mathcal{S}$, the set of superpixels extended to 3D $\mathcal{S}^{\uparrow 3}$ is then derived by duplicating n_{foc} times any superpixel $s \in \mathcal{S}$ along the axis $\mathbf{e}_2$. Note that $\mathcal{S}^{\uparrow 3}$ thus defines a 3D partition of $\mathcal{P}$.

The sharpness value of any element $t \in \mathcal{S}^{\uparrow 3}$ is then derived as the mean sharpness values of the 3D pixels $p \in t$, and so, for each superpixel $s \in \mathcal{S}$, a sharpness profile is derived (gathering sharpness value of $t \in \mathcal{S}^{\uparrow 3}$ such that $t_{\downarrow} = s$. Finally, *blind* depth estimation in each superpixel $s \in \mathcal{S}$ is derived from the maximum value in s sharpness profile. In the following, the *blind* depth superpixel map is denoted $\hat{\mathbf{u}} = (\hat{u}_s)_{s \in \mathcal{S}}$ with $\hat{u}_s \in \mathbb{N}$ assuming (without loss of generality) that depth values are sampled as integer numbers. This depth map may be noisy and sensitive to the low sharpness profile of homogeneous regions of the scene, which we will cope with our anisotropic neighborhood based regularization.

2.3 Energetic formulation

Usual energetic formulations map a realisation of the random field, $\mathbf{u}$, to an energy, i.e. a real number representing $\mathbf{u}$ inadequacy to correspond to the observations and prior knowledge. In our case, since the neighborhoods are also unknown, the energy depends also on a neighborhood field that maps a local anisotropic neighborhood to any field element (superpixels in our case). In the following, $\mathbf{u} \in \mathbb{N}^{\mathcal{S}}$ is the researched depth field, V is the neighborhood field and $\mathbb{V}$ is the set of possible neighborhood fields. Then, we aim at finding a minimizer of

$$F(\mathbf{u}, V) = E_1(\mathbf{u}) + \alpha E_2(\mathbf{u}, V), \tag{1}$$

where $\alpha \in \mathbb{R}_{\geq 0}$ is an hyperparameter left to the user.

For data fidelity term, we take inspiration from previous SFF variational formulations such as Gaganov and Ignatenko (2009); Moeller et al. (2015) that however are not convex. In order to take advantage of fast and exact optimization algorithms based on graph-cuts, we rather propose $E_1(\mathbf{u})$ be instantiated with a quadratic distance to the *blind* estimate $\hat{u}_s$:

$$E_1(\mathbf{u}) = \sum_{s \in \mathcal{S}} W_s (u_s - \hat{u}_s)^2, \tag{2}$$

where W_s depends on the dynamics of the sharpness profile normalized by its averaged value:

$$W_s \propto \left(\frac{\max\limits_{k \in [\![1, n_{\text{foc}}]\!]} (\mathbf{f}_k(s)) - \min\limits_{k \in [\![1, n_{\text{foc}}]\!]} (\mathbf{f}_k(s))}{\frac{1}{n_{\text{foc}}} \left(\sum\limits_{k \in [\![1, n_{\text{foc}}]\!]} \mathbf{f}_k(s) \right) - \min\limits_{k \in [\![1, n_{\text{foc}}]\!]} (\mathbf{f}_k(s)) + \epsilon} \right), \tag{3}$$

with $\epsilon \in \mathbb{R}_{>0}$ set to avoid division by zero. With the weighting term W_s, the importance of the data fidelity term E_1 is decreased when the sharpness profile

is homogeneous or when it presents a very low dynamic. Conversely, the areas with a sharpness profile with a precisely localized high response have high values of W_s reflecting the belief that they are trustful. Note that since W_s is fixed (it does not depend on u_s), $E_1(\mathbf{u})$ is convex.

The regularization term, we propose $E_2(\mathbf{u}, V)$ be derived from the TV operator: For any $\mathbf{u} \in \mathbb{N}^{\mathcal{S}}$ and any $V \in \mathbb{V}$,

$$E_2(\mathbf{u}, V) = \sum_{s \in \mathcal{S}} \sum_{t \in V(s)} W_{st} |u_s - u_t|, \tag{4}$$

where W_{st} is a weighting function such that

$$W_{st} = \frac{1}{2} \left(\frac{1}{\sharp V(s)} + \frac{1}{\sharp V(t)} \right), \tag{5}$$

where $\sharp V(s)$ denotes the cardinality of the neighborhood at the superpixel s. The weighting term W_{st} allows us to normalize the regularization terms E_2 with respect to the size of the considered neighborhoods since this latter is no longer constant (as it was with usual 4 or 8-connectivity for instance at pixel level).

2.4 Optimization

Considering simultaneous estimation of V and $\mathbf{u}$, the resolution appears very complex if not intractable, and considering alternate estimation would require an iterative scheme ensuring the convergence in a controlled number of iterations. Therefore in this study, we rather focus on a single estimation of V as a first attempt, with obvious methodological and computational benefit, at the expense of defining an estimation sufficiently robust to the input data imperfections.

Then, for a given V (e.g., estimated as described in Section 3), we have to find the optimal label field $\mathbf{u}$. Let us recall that graph cuts optimization refers to the computation of minimum cuts/maximum-flows in a graph of appropriate topology for minimizing functionals arising in computer vision, e.g. composed of unary and pairwise terms. Compared to other combinatorial algorithms, graph cuts are very competitive both in terms of accuracy (global minimum is very well approached if not reached as for many binary problems) and running time (by avoiding stochastic iterative convergence) for a wide range of computer vision tasks Szeliski et al. (2008). Practically, graph cuts depict linear complexity in the number of nodes Boykov and Kolmogorov (2004). Moreover, compared to continuous minimization algorithms, they are able to deal with regular or irregular lattices without any difficulties.

Even if guarantying a global minimizer of the functional has only been established for some binary problems Boykov and Kolmogorov (2004), in the multi-labels case (such as in our case), efficient algorithmic schemes exist for finding minimizers of functionals. Given V, the functional $u \mapsto F(u, V)$ is

convex (since data fidelity and regularization terms are convex according to Equations (2), (3), (4) and (5)). A global minimizer of this functional can then be efficiently obtained by decomposing the problem into a set of subproblems only involving binary variables (as in the case of isotropic neighborhoods in Ribal et al. (2018)), where each one of them is solved standard graph cuts in the binary case.

3 Anisotropic neighborhood construction

The construction of anisotropic neighborhoods can be decomposed into (i) the estimation of the presence of thin structures (in our case performed by a vesselness operator) discussed in Sections 3.1 and 3.2, and (ii) the actual computation of the neighbors of each superpixel discussed in Section 3.3. Let us recall that the neighborhoods are constructed on an irregular lattice of superpixels, and that, in prevision of graph cut optimization, we formalize neighborhood relationships through a graph whose nodes correspond to the superpixels and whose edges represent the neighborhood relationships. The neighborhood is thus an application that maps the set of superpixels $\mathcal{S}$ to its powerset $2^{\mathcal{S}}$ without any specific constraint (e.g., bound on spatial distance) at this stage. We outline that it may differ from the notion of *adjacency* that refers to the existence of a common border between the superpixels and that allows for the definition of connected components.

Then, to estimate anisotropic neighborhoods, we rely on a guidance map, denoted $\mathbf{g}$, that encodes the information of anisotropy and orientation for every superpixel $s \in \mathcal{S}$. Such a map must encourage the alignment of the neighborhoods with the thin structures of the image. In the absence of knowledge of the scene objects, the estimation of $\mathbf{g}$ is not trivial at all. More specifically, we investigate two options, the *Tensor Voting* (TVo) as presented by Medioni et al. (2000) and the *Ranking the Orientation Responses of Path Operators* (RORPO) vesselness operator as introduced by Merveille et al. (2018). In what follows, $\mathbf{g}$ is a field of $\mathbb{R}^2$ vectors encoding both the direction and the saliency.

3.1 Tensor Voting-based guidance map

3.1.1 Tensor Voting basics

Tensor Voting (TVo) has been selected for its robustness to noise and efficiency for connecting thin structures like edges Medioni et al. (2000). TVo relies on the Gestalt principles of perceptual organization (such as proximity, continuity and similarity) for designing the voting operation. Its formulation involves one scale parameter, $\sigma_T \in \mathbb{R}_{>0}$, setting the spatial range in which most of the energy of the TVo will be distributed. The basic idea is that casting a vote to other site locations allows the information of each tensor to be propagated, and then thanks to the voting step, the tensors are smoothed and their orientations refined. Voting operation is performed through voting kernels that have continuous and smoothly varying orientations of eigenvectors and decreasing

eigenvalues, except at the origin of the kernel. For implementation purpose, the voting kernels are often discretized and stored into a precomputed field of tensors, which evaluates the values of the tensors cast from the voter on each point of a regular lattice. In Appendix A, we specifie all the main equations and steps useful for 3D TVo, that is much more complex than 2D one used in Zou, Cao, Li, Mao, and Wang (2012) for instance.

In Medioni et al. (2000), TVo involves the five following main steps. First step is the initial vote that requires the definition of the initial set of voters also called *tokens* and the set of cast locations (for vote). In the absence of orientation information, the set of tokens is usually converted into a sparse set of ball tensors that vote in every image site. Second step is a refinement step. Based on the previous sparse vote, the initial set of ball tensors can be refined into a set of stick tensors. For this, each tensor is projected on the stick tensor axis in the basis used for tensor decomposition. Third step is a dense voting in order to propagate the stick information at every point. It yields the dense tensor map. Then, fourth step projects the tensors on the three axes of the decomposition basis so that three saliency maps can be derived, encoding for surface, curve and junction saliency. From these maps, the final step of the algorithm derives the probabilities of presence of surfaces, curves and points.

3.1.2 Computation of guidance map

We adapt TVo to our SSF problem as follows. The *tokens* are the local maxima of sharpness profiles in every superpixel (in $(\mathbf{e}_0, \mathbf{e}_1)$ plane). To avoid redundancy between close maxima (inducing artificial reinforcement of these latter) of a same profile, a non-maximum suppression step is performed such that we only keep one maximum per continuous interval of focus values associated to sharpness values greater than 80% of the maximum sharpness (the *tokens* are thus all separated by a local minimum having value below 80% of the global maximum). This initialization provides a tensor map that is sparse in 3D, but dense in 2D.

Then, since the number of pairs in $(\mathcal{S}^{\uparrow 3} \times \mathcal{S}^{\uparrow 3})$ is very large, the vote for the orientation estimation is also restricted to the set of *tokens*. This allows for reducing the computational burden by removing the dense voting step, at the risk of a loss of accuracy.

Although TVo allows us to handle tensors defined in $\mathbb{R}^3$ for the vote, at the end (for decision after voting) we have to decide a single tensor for any 2D superpixel $s \in \mathcal{S}$. In our experiments, we found that the most convincing results are obtained when only considering the cumulated tensors (after voting) at the *blind* estimated depth $\hat{u}_s$. Indeed, while this leads to irregularities when $\hat{\mathbf{u}}$ is noisy, this also allows for gaps in the orientations estimated on the edges of the structures of the images, which could be beneficial. Then, for extracting the guidance map $\mathbf{g}$, for each superpixel $s \in \mathcal{S}$, we project the selected tensor (in $\hat{u}_s$) into image plane $(\mathbf{e}_0, \mathbf{e}_1)$ and derive the major eigenvector $\hat{\mathbf{e}}_{0s}$ in s and the two eigenvalues $(\lambda_{0s}, \lambda_{1s}) \in \mathbb{R}^2_{\geq 0}$ so that the saliency and orientation of

the guidance map in s is

$$\mathbf{g}_s = (\lambda_{0s} - \lambda_{1s})\hat{\mathbf{e}}_{0s}, \quad \forall s \in \mathcal{S}.$$

3.2 RORPO-based guidance map

As an alternative to TVo, we consider RORPO, a non linear operator based on mathematical morphology and used for thin structure detection Merveille et al. (2018).

3.2.1 RORPO basics

The idea of RORPO is to use a set of oriented filters with different orientations to analyze the image in terms of the response to multiple morphological operations. Indeed, since, for a thin structure, at least one dimension is substantially smaller than the other ones, determining the pixels (or sites) where only a small number of high responses are measured among the oriented filters discriminates the thin structures. In RORPO, the oriented filters, called path openings, are parameterized by structuring functions defining the set of connection relationships $\mathcal{R}$ between sites (pixels, or superpixels in our case). Let us briefly recall how these latter work, firstly on a binary image and secondly on a gray level image.

Denoting by $\mathcal{R}$ the set of connection relationships, for each connection relationship $\rightsquigarrow_\theta \in \mathcal{R}$, usually corresponding to an imprecise orientation θ, a path opening is defined for binary images in Merveille et al. (2018): Given $\rightsquigarrow_\theta$ and a length $L \in \mathbb{R}_{>0}$, the path opening $\mathbb{O}_{\rightsquigarrow_\theta, L}(X)$ is the union of all paths connected by $\rightsquigarrow_\theta$ and of length L in the set of *true* pixels (1-valued) in the considered binary image X. Each path opening filters out the structures that are not consistent (i.e., usually θ aligned) with the considered orientation. Thus, a thin structure will be deleted by at least one oriented filter, conversely to isotropic structures.

Then, to extend binary path openings to gray level images, one considers level sets, i.e. sets of sites having a gray value greater than a given real value: for any gray level image Y and $\tau \in \mathbb{R}$, the τ-level set of Y is $Y_{\geq\tau} = \{s \mid Y(s) \geq \tau\}$. Then, given $\rightsquigarrow_\theta$ and $L \in \mathbb{R}_{>0}$, the gray level path opening of Y is

$$\mathbb{O}_{\rightsquigarrow_\theta, L}(Y, s) = \max\left\{\tau \in \mathbb{R}_{>0} \mid s \in \mathbb{O}_{\rightsquigarrow_\theta, L}(Y_{\geq\tau})\right\}.$$

In Merveille et al. (2018), RORPO implementation involves the following five main steps. The first step is the dilation of the gray level input image with respect to spatial adjacency. The second step deals with direction sampling. It boils down defining a finite set of connection relationships, denoted by $\rightsquigarrow_\theta$, such that two sites s and t are connected if and only if (i) they are adjacent and (ii) $\overrightarrow{st}$ vector's direction and the sampled direction θ are considered equal been given the imprecision angle ϕ_T threshold. The third step is the computation of the path opening results for the sampled directions and the fourth step ranks their responses as follows: For each site, the responses to the $\#\mathcal{R}$ path openings

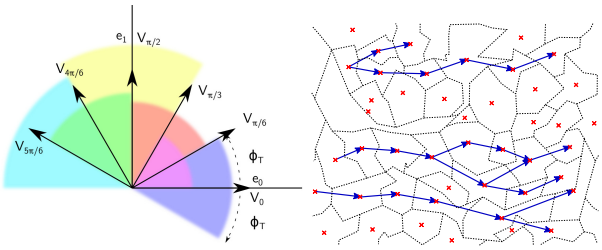


Fig. 1: Illustration of the 6 directions of $\mathcal{R}$ (left) and an example of path obtained with one structuring function $\rightsquigarrow_\theta$ (right). The connectedness $\rightsquigarrow_\theta$ is characterized by the vector $\mathbf{v}_\theta$ and the angular width ϕ_T. For this illustration, we have represented directed edges for positive displacements, but the paths are computed in both directions.

are ranked in decreasing order of magnitude, i.e. denoting RF_1 the maximum value and $RF_{\#\mathcal{R}}$ the minimum value. This ranking of the orientation responses of the path openings gave its name to the algorithm RORPO. Then, for each site, the RORPO value is the difference between maximum path opening value (RF_1) and the i^{th} largest response, (RF_i). In our case, we set $i = 4$. Finally, fifth step derives, for each site, an orientation by averaging the orientations of the three largest responses.

This formulation yields higher responses for thin structures that have a small number of high responses in path openings. Therefore, the value returned by the RORPO allows us to discriminate the saliency of thin structures.

3.2.2 Computation of guidance map

Our implementation of RORPO works with the data volume corresponding to the sharpness profiles in every superpixel. However, for numerical convenience, path openings are only performed with 2D slices, i.e. at given focus value, which boils down researching structures in image plane. For this, we simply restrict the connection relationships $\rightsquigarrow_\theta$ to be within $(\mathbf{e}_0, \mathbf{e}_1)$. Then, we consider six directions $\mathbf{v}_\theta$ such that $\mathbf{v}_\theta = \cos(\theta)\mathbf{e}_0 + \sin(\theta)\mathbf{e}_1$ (see Figure 1). Besides, extending the case of pixel lattice to superpixel one, the path length L is a real $L \in \mathbb{R}_{>0}$, computed as the sum of the distances between the superpixels' barycenters in the path.

Each of the connection relationships yields a path opening result. From this set of path openings, we firstly compute the RORPO index that is further interpreted as a saliency index and secondly the structure orientation. For the latter, we use a specific average operation such that orthogonal vectors cancel and vectors of opposite directions would not. The trick consists in considering polar coordinates, doubling the argument value of the vectors before averaging them, and dividing the argument of the averaged result by two. Mathematically, with complex notations, and omitting the normalization

coefficient useless here, it is as follows:

$$\mathbf{v}_{RO}(s) \propto \left(\epsilon + \sum_{\rightsquigarrow_\theta \in \mathcal{R}'(s)} \mathbb{O}_{\rightsquigarrow_\theta, L}(Y, s) \exp(2i\theta) \right)^{\frac{1}{2}},$$

where $\epsilon > 0$ is a very small real number used for numerical stability of the expression, and $\mathcal{R}'(s) \subset \mathcal{R}$ is the set of orientations of the three first answers (according to rank filter) at site s.

The proposed RORPO implementation provides a 3D volume of vectors with the same dimensions as the input data. The 2D guidance map $\mathbf{g}$ is then derived by performing a final average operation along depth dimension:

$$\mathbf{g}_s = \left(\sum_{t \in \mathcal{S}^{\uparrow 3}, t_\downarrow = s} |\mathbf{g}_t| \exp(2i \arg(\mathbf{g}_t)) \right)^{\frac{1}{2}}, \quad \forall s \in \mathcal{S},$$

where $t_\downarrow$ is the result of the projection of 3D site t on the 2D image plane and $\arg(c)$ denotes the argument of any complex number $c \in \mathbb{C}$. In previous equation, the angles are weighted by the norm $\mathbf{g}_t$ to to take into account the significance of the orientation.

Compared to TVo, RORPO allows for a faster computation of the guidance map and is consistent with the notion of path-based neighborhood introduced in Section 3.3 that specifies the construction of the neighborhoods from $\mathbf{g}$.

3.3 Path-based neighborhoods

This section depicts the proposed construction of anisotropic neighborhoods, $V \in \mathbb{V}$.

We propose path-based neighborhoods to fit into thin structures of the image, possibly only one superpixel width. Being based on the adjacency graph $\mathcal{A}$, our neighborhood construction ensures that the neighbors of a superpixel define a single connected component. Two superpixels being adjacent when they share a common border at pixel level, the adjacency is a symmetric relationship: $s \in \mathcal{A}(t) \iff t \in \mathcal{A}(s)$, $\forall s, t \in \mathcal{S}$. Then, a path of length $n \in \mathbb{N}$ is an ordered list $(s_0, \ldots, s_n)$ of consecutive adjacent superpixels.

More formally, let $\Pi_K(s, t)$ denote the set of paths joining any pair of superpixels $(s, t) \in \mathcal{S}^2$, without any loop, and having length K: $\Pi_K(s, t) = (s_0, \ldots, s_K) \subset \mathcal{S}^{K+1}$, such that $\forall k \in [\![0, K[\![, s_{k+1} \in \mathcal{A}(s_k)$, $s_0 = s$, $s_K = t$, and $\forall j, k \in [\![0, K]\!], s_j \neq s_k$. Similarly, we also define $\Pi(s, t)$, the set of paths joining superpixels s, t with any length:

$$\Pi(s, t) = \bigcup_{K \in \mathbb{N}} \Pi_K(s, t). \tag{6}$$

The proposed path-based neighborhoods rely on previously estimated guidance map $\mathbf{g}$ that contains the information about the orientation and saliency of the structures of the scene. In particular, when the norm $\|\mathbf{g}_s\|$ at a given superpixel s is below a fixed threshold, the neighborhood in s is $V(s) = \mathcal{A}(s)$, i.e. an isotropic neighborhood that ensures adjacency. Otherwise, the set of neighbors is given by the union of the elements of two paths that expand from s to the two opposite directions corresponding to the orientation of $\mathbf{g}_s$. In the next subsections, we present the two options investigated for constructing these path-based neighborhoods.

In order to illustrate the benefit of the two proposed path-based neighborhoods, Figure 2 provides a toy example with, on the first row, two kinds of usual neighbourhoods and, on the second row, the two variants of path-based neighborhoods. Considering Disc neighborhood, a superpixel is a neighbor if its barycenter is included in a disc centered on the reference site s. The fact that the union of the neighborhood with the site $V(s) \cup \{s\}$ would constitute a single connected component is not ensured, as shown with the green site in the upper left corner. Considering Stawiasky's isotropic neighborhood Stawiaski and Decencière (2011), a superpixel is a neighbor if it is adjacent to the reference one. Therefore, the union of the neighborhood with the site is always a single connected component, but the actual shape of the neighborhood is not controlled since some sites may extend spatially far away from their actual neighbor.

3.3.1 Target-based neighborhood

Target-Based Neighborhood (TBN) gather the elements of two paths that join, starting from a source superpixel $s \in \mathcal{S}$, two "targets" corresponding to distant superpixels $(t_0^*, t_1^*) \in (\mathcal{S} \setminus \{s\})^2$. Specifically, for $j \in \{0, 1\}$, the targets are selected with

$$t_j^* \in \underset{t \in \mathcal{S} \text{ s.t. } (-1)^j \langle \mathbf{g}_s, \overrightarrow{st} \rangle > 0}{\operatorname{argmin}} \|I(s) - I(t)\|_2^2 - \eta \|\overrightarrow{st}\|_2 \times \left| \leqslant \mathbf{g}_s, \overrightarrow{st} \geqslant \right|, \qquad (7)$$

where $\langle ., . \rangle$ denotes the dot product between two vectors so that the constraint $(-1)^j \langle \mathbf{g}_s, \overrightarrow{st} \rangle > 0$ refers to an half-space domain, $\leqslant ., . \geqslant$ stands for the cosine similarity (also called normalized dot product) between two vectors, $\|.\|_2$ is the Euclidean norm of a vector, $|.|$ is the absolute value of a real number, and $\eta \in \mathbb{R}_{>0}$ an hyperparameter set to $\eta = 30$ in our experiments.

In Equation (7), the first term favors the superpixels s and t to share similar image intensities while the second one favors far targets being aligned with $\mathbf{g}_s$. Note that, for computational convenience, the range of search (of those target superpixels) is restricted to an ellipse centered at s with major axis aligned with $\mathbf{g}_s$ and that solutions are derived by Dynamic Programming (DP).

Then, the paths are selected among the two sets $\Pi(s, t_0^*)$ and $\Pi(s, t_1^*)$ (cf. Equation 6). Path selection itself relies on a cost function that the optimal

path (denoted by p_j^*, $j \in \{0,1\}$), has to minimize:

$$p_j^* \in \underset{p \in \Pi(s,t_j^*)}{\operatorname{argmin}} \sum_{k=0}^{|p|-1} \|I(p(k)) - I(p(k+1))\|_2^2, \tag{8}$$

where $|p|$ stands for the length of the path p, and $p(k)$ denotes the k^{th} element of it. The term to minimize in Equation (8) is large when the gray levels of successive superpixels along a path are dissimilar and small otherwise. We also use DP to derive a minimizer of previous equation, and the neighborhood $V(s)$ is finally constructed as the set of the superpixels in p_0^* or in p_1^*, but s: $V(s) = (p_0^* \cup p_1^*) \setminus \{s\}$. The adjacency along these paths being ensured by construction, the derived neighborhood forms a single connected component.

Figure 2c provides an illustration of TBN. While this neighborhood ensures that the set of neighbors of a superpixel s forms a single connected component and favors paths oriented in the estimated direction of thin structures, there is no guarantee concerning the cardinality of these paths. Depending on the image content that influences the location of the target sites, one site may have a very small amount of neighbors in a very contrasted location, or conversely could possibly have a large number of neighbors if there exists an arbitrary long path with constant radiometry.

3.3.2 Cardinal-based neighborhood

Cardinal-Based Neighborhood (CBN) is also a path-based neighborhood. However, instead of constraining the path extremities like TBN, it constraints path cardinality (and thus neighborhood cardinality): $\forall s \in \mathcal{S}$, $V(s)$ is the union (excepting element s) of two length-fixed paths $p_0^*, p_1^* \in \Pi_K(s,\cdot)$, with $\Pi_K(s,\cdot)$ denoting the set of paths of length $K \in \mathbb{N}_{>1}$ starting from s. Additionally, these paths are encouraged to expand in opposite directions.

As previously, we define a cost function presenting a tradeoff between fidelity to the thin structure orientation and fidelity to the gray level of originating superpixel s. For any $j \in \{0,1\}$,

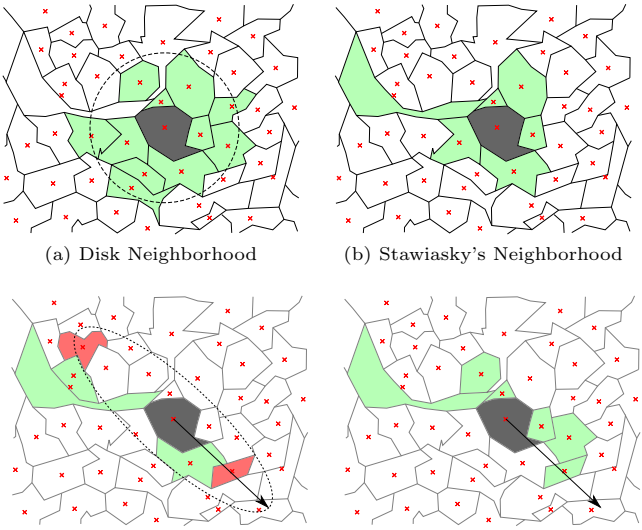
$$p_j^* \in \underset{p \in \Pi_K(s,\cdot)}{\operatorname{argmin}} \sum_{t \in p} \|I(s) - I(t)\|_2^2 + \eta' \psi_j(\overrightarrow{st}, \mathbf{g}_s),$$

where $\eta' \in \mathbb{R}_{>0}$ is an hyperparameter set to $\eta' = 100$ along with $K = 3$, in our experiments, and

$$\psi_j(\vec{u}, \vec{v}) = \begin{cases} \arccos\left(|\leqslant \vec{u}, \vec{v} \geqslant|\right) & \text{if } (-1)^j \langle \vec{u}, \vec{v} \rangle > 0, \\ +\infty & \text{otherwise,} \end{cases}$$

measures the angle between the vectors $\vec{u}$ and $\vec{v}$ and discriminates whether the dot product is positive or not.

Note that, in CBN, the cost function compares gray level and positions of each site of the path versus the site s instead of computing these differences



(a) Disk Neighborhood (b) Stawiasky's Neighborhood

(c) Target-Based Neighborhood (d) Cardinal-Based Neighborhood

Fig. 2: Toy example of usual and path-based neighborhoods. The neighbors of the site s (in grey) are pictured in green. When considered, the arrow indicates the orientation of $\mathbf{g}_s$ and superpixels of neighborhood are shown in color. For "Disk" and Stawiasky's neighborhoods, the orientation is not considered but distance or connectivity to superpixel s. For TBN, two "targets" (in red) are selected in an ellipse centered on the source superpixel s (in grey). For CBN, two paths with $K = 3$ elements are built to be aligned with $\mathbf{g}_s$, taking into account radiometric similarity with the site s.

between adjacent sites on the path (like TBN), to allow for local deviations while ensuring global neighborhood orientation and gray level value. Figure 2d shows an example of CBN. Ideally, the two paths are aligned with $\mathbf{g}_s$, but the energy term allows for deviations from this axis for following thin structures in the image.

4 Experiments and results

4.1 Data

We tested our approach on some scenes extracted from the public[1] Middlebury College dataset Scharstein and Pal (2007), used in stereo matching and also in SFF (Kumar G. & Sahay, 2017; Ribal et al., 2018). For each scene, a ground truth depth map and an all-in-focus RGB image are provided. Both images have 360×360 pixels. These images come along with the defocusing algorithm Pertuz et al. (2013), that is currently available as a Matlab source

[1] https://vision.middlebury.edu/stereo/data/

on MathWorks file exchange, that enable us to simulate the desired set of blurred images, corresponding to different focal object plane depths. For simplicity and readability, focus values were regularly sampled with step equal to the unit. The maximum depth, denoted by $\Delta_h \in \mathbb{N}$, is therefore equal to n_{foc} that we set equal to 50. However, some images taken at irregular steps could be considered as well without loss of generality.

Then, the set of defocused images was assumed to be the only input data available, and we reconstructed depth values based on the following steps. Firstly, we computed the sharpness profiles in each pixel independently and from maximum of these profiles, we derived the *blind* estimate of all-in-focus image. Secondly, we computed the superpixels from this *blind* all-in-focus image. The number of superpixel algorithms proposed in the literature is rather important, including different kinds of superpixels that embed different properties, such as the adherence to the boundaries of the objects, the compactness or convexity of the resulting superpixels, their regularity, or the smoothness of their boundaries. We refer the reader to Stutz et al. (2018) to have an overview of the variety of superpixel algorithms. In practice, after a few comparisons, we focused on the superpixels provided by an algorithm called ETPS Yao, Boben, Fidler, and Urtasun (2015), since it was energy based (as the general framework adopted for our work) and offered smooth and regular superpixels. Thirdly, as described in Section 2.2, the sharpness profile in each superpixel as well as the *blind* superpixel depth map $\hat{\mathbf{u}}$ were derived. Fourthly, we computed the guidance map $\mathbf{g}$ and constructed the neighborhood field $\{V(s), \forall s \in \mathcal{S}\}$, based on the chosen method as described in Sections 3.1.2, 3.2.2, 3.3.1 and 3.3.2. Fifthly, V and $\hat{\mathbf{u}}$ allowed us to instantiate our anisotropic regularization and to derive the regularized depth map results presented in the following next sections.

4.2 Evaluation criteria

The Ground Truth (GT) provided in Middlebury College dataset Scharstein and Pal (2007) is at pixel level. To perform evaluation, we duplicated the depth estimated for a given superpixel to each of its pixels. Then, we also denote by $\mathbf{u}$ the estimated depth map at pixel level (the element lattice $\mathcal{P}$ or $\mathcal{S}$ removing ambiguity if any) and by $\tilde{\mathbf{u}}$ the GT.

Evaluation metrics

We focus on three complementary global metrics, namely RMSE (Root Mean Square Error) that has good additive properties, PSNR (Peak Signal to Noise ratio) derived from RMSE and SSIM (Structural Similarity Index Measure Wang and Sheikh (2004)) that is based on perception-model to measure the similarity between two images:

$$\text{RMSE}(\mathbf{u}, \tilde{\mathbf{u}}) = \sqrt{\frac{1}{\#\mathcal{P}} \sum_{p \in \mathcal{P}} (u_p - \tilde{u}_p)^2},$$

$$\mathrm{PSNR}(\mathbf{u},\tilde{\mathbf{u}}) = 20\log_{10}\left(\frac{\Delta_h}{\mathrm{RMSE}(\mathbf{u},\tilde{\mathbf{u}})}\right),$$

$$\mathrm{SSIM}_\Omega(\mathbf{u},\tilde{\mathbf{u}}) = \frac{1}{\sharp\mathcal{P}}\sum_{p\in\mathcal{P}}\frac{(2\mu_{u,p}\mu_{\tilde{u},p}+C_1)\,(2\sigma_{u,\tilde{u},p}+C_2)}{\left(\mu_{u,p}^2+\mu_{\tilde{u},p}^2+C_1\right)\left(\sigma_{u,p}^2+\sigma_{\tilde{u},p}^2+C_2\right)},$$

where $\sharp\mathcal{P}$ stands for the cardinality of $\mathcal{P}$, Ω is a window centered at any pixel p and of size 7×7 in our case, $\mu_{u,p}$, $\mu_{\tilde{u},p}$ are the means over Ω centered at p of $\mathbf{u}$ and $\tilde{\mathbf{u}}$ values respectively, $\sigma_{u,p}^2$, $\sigma_{\tilde{u},p}^2$, and $\sigma_{u,\tilde{u},p}$ are the variances and covariance, respectively. Finally, the constants C_1 and C_2 are computed from Δ_h as $C_1 = (0.01\Delta_h)^2$ and $C_2 = (0.03\Delta_h)^2$ for numerical stability. This is the version of SSIM specified in Wang and Sheikh (2004) with (according to author's notations) $\alpha = \beta = \gamma = 1$. By computing the variances, covariance and mean values on a set of windows covering the whole image, SSIM incorporates comparison measurements of luminance, contrast and structure of images that allows to take into account important perceptual phenomena in its evaluation.

For result comparison, we remind that the lower the RMSE values are (in $[0,\Delta_h]$), the better the results are while for PSNR and SSIM criteria, higher values (in $\mathbb{R}_{\geq 0}$ and $[0,1]$ respectively) reflect better performance.

Evaluation maps

Three complementary kinds of maps allow us to visualize the difficult areas. Firstly, depth error map, called E, will stress the image areas with poorest reconstruction. Secondly, neighborhood orientation map will represent saliency and direction information extracted from the guidance map, that allows us to evaluate qualitatively this latter. Thirdly, depth dynamic within neighborhoods, called Q_V, provides a measure of the neighborhood consistency in terms of depth. Pixel values of E and Q_V maps are computed as follows:

$$E(p) = |u_p - \tilde{u}_p|, \quad \forall p\in\mathcal{P},$$

$$Q_V(p) = \max_{q\in V(p)} |\tilde{u}_q - \tilde{u}_p|, \quad \forall p\in\mathcal{P},$$

where $V(p)$ at pixel level is simply the set of pixels that belong to any superpixel neighbors of the superpixel including p.

Concerning the interpretation of these maps, the lower the E values (in $[0,\Delta_h]$), the better the depth estimation at considered pixel. The orientation map is expected to be relatively smooth while following the sharp edges of the objects and aligning with the thin structures. Finally, in Q_V, low values (in $[0,\Delta_h]$) reflect a consistent neighborhood (without implying uniqueness of the solution). Note that a major benefit of Q_V criterion is that it does not require any neighborhood ground truth (which we obviously do not have).

4.3 Alternative approaches considered for comparison

To evaluate the benefits of our approach compared against isotropic neighborhoods or simplest anisotropic ones, we focused on the following alternative approaches.

Stawiaski's isotropic neighborhood

In Stawiaski and Decencière (2011), an isotropic neighborhood is computed such that the superpixels that share a common border are neighbors and their interactions are weighted by the length of this common border. This neighborhood corresponds to the adjacency relationship, with a weighting function. This formulation ensures that the set constituted by a superpixel and its neighbors is a single connected component, but it does not ensure that the barycenters of neighboring superpixels are close from each other.

Shape-based neighborhood inspired from Giraud et al. (2017)

Shape-based neighborhood is an intuitive method for building anisotropic neighborhoods. The "shape" refers to the approximation of the neighborhood as a parametric shape, namely ellipse in our case. The neighbors of a superpixel s are then the superpixels whose barycenter is included in the "shape" centered in s. We considered in our case parametric ellipses whose major semi axis directions are given by the guidance map in s. Practically, when saliency in superpixel s is very low, i.e. $\|\mathbf{g}_s\|$ is lower a given threshold (0.05), the direction is not reliable so that we rather define neighborhood as a disc, which boils down to the isotropic superpatch neighborhood of Giraud et al. (2017). Note that we do not exploit further saliency information that appears noisier than direction, and set the eccentricity as a constant parameter of the model.

Perfect neighborhood

For having an estimation of the possibly best performance brought by an anisotropic approach, we propose a so-called *perfect* anisotropic neighborhood. The latter is computed with respect to GT depth map $\tilde{\mathbf{u}}$ as follows. *Perfect* neighborhood is implemented as a shape-based neighborhood with a disc of given radius centered in $s \in \mathcal{S}$, where we remove the neighbors presenting a depth difference between the depths of GT and s higher than a fixed threshold $D_V = \frac{\Delta_h}{10} + 1$. Additionally, elements that do not belong to the s connected component are removed from the neighbors. Thus, *perfect* neighborhood refers to a neighborhood having good properties in terms of homogeneity, connectivity and shape, even if it is not unique.

4.4 Results

4.4.1 Global performance analysis

Let us first consider global performance obtained considering the whole Middlebury college dataset. Figure 3 shows the results achieved using 5000 superpixels (ETPS Yao et al. (2015)), in terms of RMSE (allowing summing individual image performance), varying the regularization parameter α. We notice that the *perfect* neighborhood and the CBN, either from RORPO or TVo, yield the lowest RMSE values meaning they outperform all the other approaches for a wide range of regularization coefficients. Since *perfect* neighborhood was designed to evaluate the performance gain specifically related to anisotropic

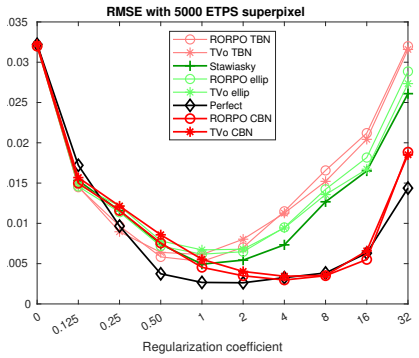


Fig. 3: Comparison of neighborhood anisotropy benefit measured through RMSE (y-axis) on the whole dataset. The results are achieved using 5000 ETPS superpixels Yao et al. (2015) and different neighborhood estimations.

neighborhood (leaving apart the question of its estimation) with respect to isotropic one (represented by Stawiaski's approach), the results clearly underline the benefit of anisotropic neighborhood for regularization. A satisfactory result is that CBN provides almost as good results as *perfect* neighborhood (which we remind is unrealistic since it requires GT), stressing the performance of neighborhood estimation itself. Comparing with "ellip" that refers to the "Shape-based neighborhood inspired from Giraud et al. (2017)", we notice that these latest results are much worse, underlining the importance of a fine (not too simplistic) estimation. About TBN estimation, we notice it only leads to interesting results for low regularization ($\alpha < 1$). Finally, we also note that RORPO or TVo use for $\mathbf{g}$ estimation does not really impact the results, but a very slight advantage for RORPO. In conclusion, according to Figure 3, best performance is achieved by RORPO CBN, with a very noticeable robustness of the results with respect to regularization coefficient, $\alpha \in [2, 16]$. This robustness of CBN to the regularization parameter that is also confirmed by visual inspection of error maps, is one of the strengths of this approach against its alternatives.

We now check the result dependency to superpixel segmentation. However, to investigate how results are dependent on ETPS superpixels, we consider, as an alternative to ETPS superpixels Yao et al. (2015), the WaterPixels (WP) proposed in Machairas et al. (2015). Figure 4 shows curves analogous to those in Figure 3 considering either 5000 (like with ETPS superpixels) or 2000 WP, respectively. With respect to Figure 3, we notice the curves and conclusions are remarkably similar, but a slight loss of performance when the number of superpixels is lower (it can be seen looking at the lowest value achieved considering *perfect* neighborhood) and a more distinct advantage for RORPO with respect to TVo (when looking at the CBN curves).

To further investigate the performance variability with respect to scene and/or superpixels, Figure 5 and Figure 6 respectively show the PSNR and the

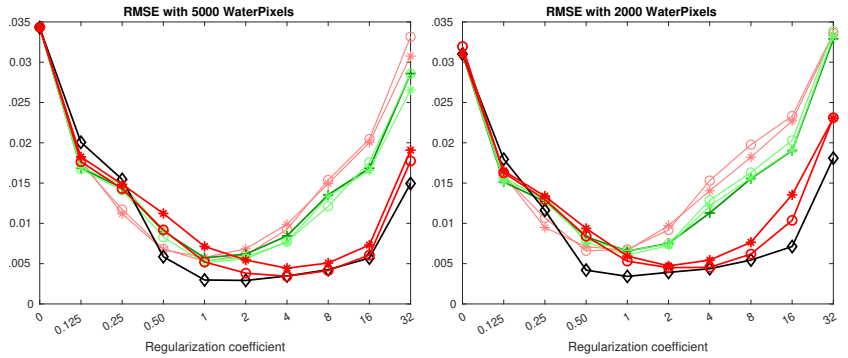


Fig. 4: Comparison of neighborhood anisotropy benefit measured through RMSE (y-axis) on the whole dataset. The results are achieved using either 5000 WP (left) or 2000 WP (right) using different neighborhood estimations. The legend is the same as in Figure 3.

SSIM obtained on each scene, for the best result obtained with a varying α. First of all, the remarks concerning the robustness to the two kinds of considered superpixels (ETPS and WP) or their number (5000 and 2000) still hold: Difficult scenes are the same and CBN achieves very interesting performance in most cases. Indeed, on some scenes such as *Aloe1*, *Books1*, *Wood11*, all approaches yield equivalent results, whereas in other scenes such as *Lampshade*, *Plastic1* or *Reindeer*, achieved results appear more sensible to neighborhood estimation. We also note that the two criteria PSNR and SSIM are complementary since differences of performance can be visible in only one of them, such as with scene *Midd11* or *Moebius1*. However, let us underline that in most scenes, the top trio is RORPO-CBN, TVo-CBN and quite obviously *perfect* neighborhood. These approaches outperform both isotropic neighborhood (represented by Stawiaski's approach) and naive anisotropic one (ellipse-based). Nevertheless we also confirm the fact that an isotropic neighborhood assumption is preferable to too naive anisotropic neighborhood estimation. In conclusion, despite the scene disparity inducing variable performance, the main conclusions concerning the benefit of anisotropic neighborhood fine estimation can also be drawn at scene level.

4.4.2 Detailed analysis of two cases

For further analysis, we present the corresponding error maps and neighborhood quality maps, focusing on some cases where the performance highly depends on the type of neighborhood, such as with the *Lampshade* scene and the *Reindeer* one.

Let us first consider the neighborhood estimation quality. As specified in Section 4.2, the values of the depth dynamic within a neighborhood are in $[0, \Delta_h]$, with low values reflecting a consistent estimation of neighborhood. From Figure 7, we clearly see that most of the heterogeneous neighborhoods

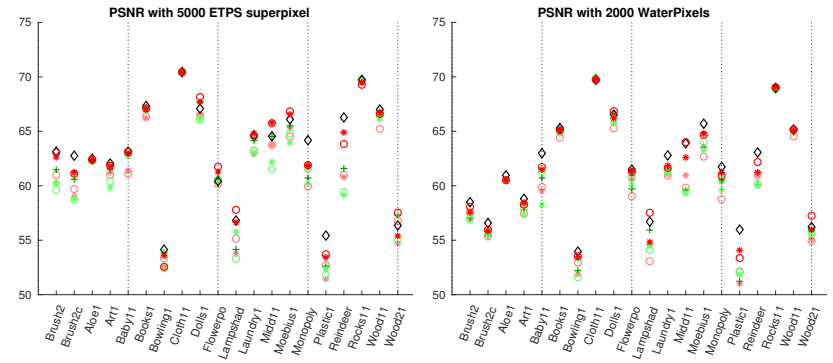


Fig. 5: Per scene best results in terms of PSNR measure (y-axis) for each neighborhood construction using either 5000 ETPS superpixels (top) or 2000 WP (bottom). The higher the value is, the better the result is.

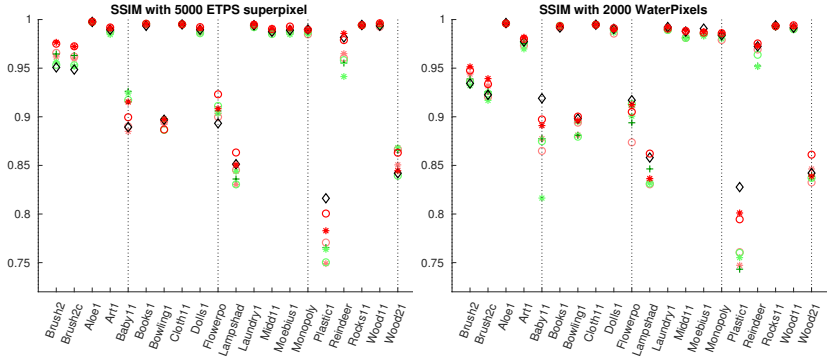


Fig. 6: Per scene best results in terms of SSIM (y-axis) for each neighborhood construction; 5000 ETPS superpixels (top) or 2000 WP (low). The higher the value is, the better the result is.

are located at the borders of thin structures such as the lampshade rod or the reindeer antlers. We also note that lowest values are achieved for the *perfect* neighborhood (by construction) and then by CBN (either from TVo or RORPO guidance map) whereas both TBN and Stawiaski's neighborhood are much worse in terms of homogeneity.

Secondly, we compare the guidance maps provided by TVo and by RORPO. Figure 8 shows these maps in the cases of the two considered scenes, *Lampshade* and *Reindeer*. In both cases, we notice that the direction of the structures is rather well estimated although we also observe some noise. Comparing the two estimators, we note that while TVo looks smoother in terms of orientation (especially on *Reindeer* scene), RORPO both better detects the isotropic areas (in white) and highlights well the sharp areas of the scene. However, these

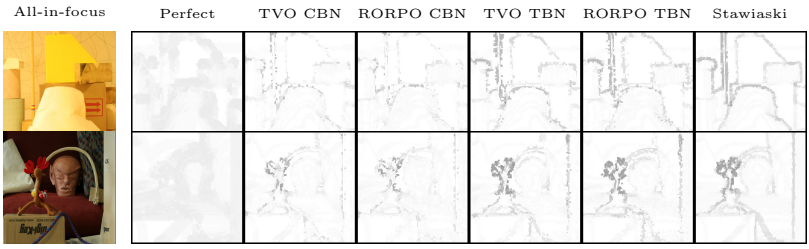


Fig. 7: Comparison of neighborhood quality Q_V for *Lampshade* (top row) and *Reindeer* (bottom row) scenes, from left to right: All-In-Focus image, Q_V maps for *perfect* neighborhood, TVo CBN, RORPO CBN, TVo TBN, RORPO TBN and Stawiaski's neighborhood. Dynamics has been reversed and spread in the interval $[0, 255]$ so that dark areas represent bad performance.

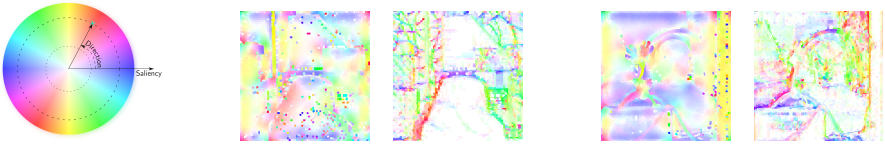


Fig. 8: Comparison of guidance maps **g** for *Lampshade* and *Reindeer* scenes, using use a color representation, such that the saturation and the hue encode respectively the saliency and the orientation; from left to right: Color wheel, TVo *Lampshade*, RORPO *Lampshade*, TVo *Reindeer*, RORPO *Reindeer*.

observed differences seems to have only little impact on the neighborhood consistency as depicted in Figure 7 or on depth map reconstruction. In what follows, we now focus on RORPO algorithm.

Finally, let us observe the error maps versus regularization parameter α for our two scenes and the three methods of neighborhood estimation: *Perfect* (reference for benefit of anisotropic neighborhood), RORPO CBN and Stawiaski (reference for isotropic neighborhood). For *Lampshade* scene, we notice the very high noise level in the absence of regularization ($\alpha = 0$) that is progressively corrected by increasing α before new errors this time due to the removal of thin structures appear. This phenomena can be clearly seen in the case of Stawiaski's neighborhood with apparition of errors located on the vertical thin bar or rod for $\alpha > 1$. From this scene, we also notice that the optimal α values vary with the considered neighborhood; as expected, anisotropic neighborhoods allow for higher α values without reconstruction degradation (in particular for the thin structures). Specifically, in *Lampshade* scene, α values providing best results are equal to 4, 8 and 2 for the *Perfect*, RORPO CBN and Stawiaski' neighborhoods, respectively. *Reindeer* scene is much less noisy than *Lampshade* scene. However, regularization is again required to remove the *blind* estimation errors in the vertical right strip and in the bottom triangle, both been part or subparts of objects presenting a very homogeneous

Superpixels segmentation	RORPO	TVo	Neighbohood construction	Depth optimization
13.2	28.1	94.6	19.0	1.3

Table 1: Mean running times (in seconds) of the main steps of the proposed depth reconstruction of a scene at superpixel level.

radiometry. Due to this lower initial level of noise, α "optimal" values are lower than in *Lampshade* scene, namely they are equal to 1, 2 and 0.5 for the *Perfect*, RORPO CBN and Stawiaski' neighborhoods, respectively. We notice that for higher values, regularization introduce depth errors on the antlers of the reindeer figure, all the more quickly as the neighborhood is isotropic (indeed with Stawiaski, bottom triangle errors cannot be corrected without degrading reindeer antlers). Using anisotropic neighborhoods, either *Perfect* or RORPO CBN, the degradation of thin structures is delayed so that we observe the existence of α values allowing for the correction of *blind* errors without introduction of new errors.

4.4.3 Superpixel versus pixel level

Finally, let us investigate the benefit of considering the superpixel level rather than the pixel one. For doing so, we consider again global performance statistics, namely the RMSE computed on the whole dataset.

In terms of complexity, the number of pixels is 360×360 versus 5000 superpixels (ETPS) in the considered experiments. Table 1 gives the mean running times in seconds computed over all the scenes of the Middlebury college dataset, for the four main steps of our approach: (i) superpixel segmentation, (ii) guidance map estimation (either based on RORPO or on TVo), (iii) neighborhood construction, and (iv) depth map optimization. These running times have been obtained on an Intel core i9-10900X @ 4.7 GHz, with 64 Go of RAM. Table 1 firstly confirms that RORPO is much faster (3 times) than TVo. Secondly, considering RORPO instead of TVo, the average running time for the global algorithm is 61.6 secs, i.e. about 1 minute per image. We consider this time as very encouraging since it was achieved with standard programming code, i.e. without optimization using GPU for instance. Thirdly, the running time for depth map optimization (using graph cuts) is very low thanks to the complexity reduction working with superpixels instead of pixels. For comparison, running the isotropic neighborhood depth map optimization at pixel level, the average running time is $36.8s$, i.e. about 30 times slower. Thus, even without code optimization, the additional running time for anisotropic neighborhood estimation steps is compensated by the running time decrease for depth map optimization step.

In terms of performance, Figure 10 allows for comparison of the RMSE curves for three kinds of neighborhoods, namely *Stawiaski* (i.e., isotropic), RORPO CBN or RORPO TBN (representing best candidate for anisotropic neighborhoods) and *Perfect*, either at superpixel level (using 5000 ETPS superpixels) or at pixel level. First of all, from Figure 10, we notice an improvement

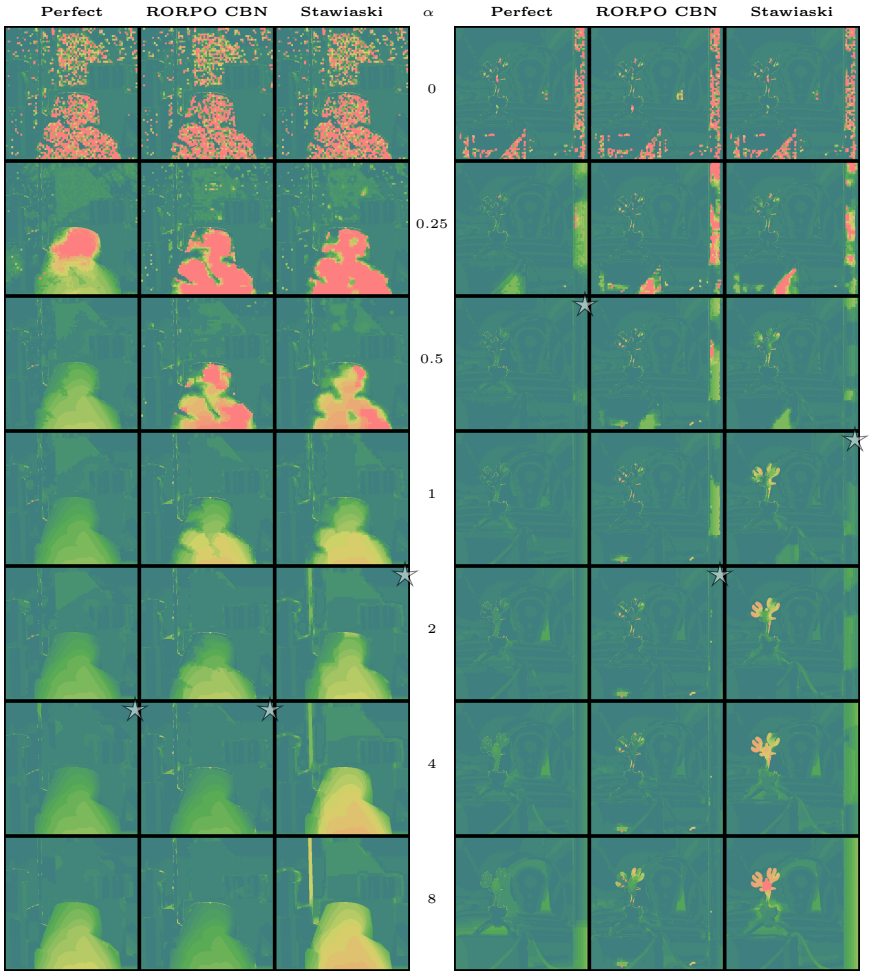


Fig. 9: Maps of depth error obtained for the scenes *Lampshade* (left) and *Reindeer* (right), for three neighborhood construction strategies (*Perfect*, RORPO CBN and Stawiaski) and different values of regularization parameter $\alpha \in \{0., 0.25, 0.5, 1, 2, 4, 8\}$. For better visualization, error value dynamic has been bounded to $\frac{2\Delta_h}{5}$.

of performance at superpixel level with respect to pixel one. This improvement is a very strong point since one could have expected that superpixels would introduce some spatial imprecision (at the benefit of complexity decrease), especially since the RMSE is measured at pixel level. Nevertheless, at least on the considered dataset, this preprocessing step is beneficial for the precise image reconstruction. This comment is confirmed in most cases when we examine individual scenes. For instance, for the two detailed cases *Lampshade*

	Lampshade		Reindeer	
	PSNR	SSIM	PSNR	SSIM
RORPO-CBN ETPS	**57.78**	**86.33**	63.83	**97.89**
4-connectivity pix. lev.	54.61	83.81	63.94	96.40
RORPO-CBN pix. lev.	53.32	83.00	**64.62**	97.18

Table 2: Results obtained for the scenes *Reindeer* and *Lampshade* with our proposed anisotropic neighborhood at superpixel level (first row), compared to isotropic neighborhood at pixel level (second row) and RORPO-CBN neighborhood at pixel level (last row). SSIM values are indicated in percentage. For each scene, best result is in bold and second best is underlined.

and *Reindeer*, the two first lines of Table 2 show the performance indicators PSNR and SSIM achieved by RORPO-CBN on ETPS superpixels and isotropic (4-connectivity) and we see that RORPO-CBN yields to significantly better result except in terms of PSNR on *Reindeer* scene where nevertheless the performance values are very close.

Then, we notice the potential benefit of anisotropic neighborhood with respect to isotropic one (at pixel level, Stawiaski's neighborhood boils down to 4-connectivity neighborhood) since *Perfect* neighborhood yields the best results. However, we also notice that, at pixel level, the difference of performance is very small, and that isotropic neighborhood yields slightly better result than RORPO TBN or RORPO CBN. A possible explanation is that the requirement to take into account anisotropic neighborhood is less pregnant at pixel level (due to the size of neighborhood with respect to objects in pixel numbers as well as the regularity of the lattice) and that neighborhood estimation is less efficient. Indeed it is based on *blind* depth estimation that may be much noisier at pixel level that at superpixel one. Besides, the performance may depend on the considered scene. For instance, Table 2 shows that on the *Reindeer* scene, RORPO-CBN at pixel level slightly outperforms isotropic pixel level both in terms of PSNR and SSIM indicators. These observations also open perspectives to understand the relationship between scene feature and scale of analysis (from pixel level to superpixel ones).

In conclusion, the benefit of presegmenting the scene in superpixels and then handling anisotropic neighborhood appears both in terms of global performance and in terms of robustness with respect to regularization parameter α. Besides the additional complexity introduced by neighborhood estimation (RORPO-CBN according to this study) is compensated by the complexity decrease when handling much less superpixels than pixels.

5 Conclusion and perspectives

In this paper, we propose some new anisotropic neighborhoods that offer a flexible and generic formulation with respect to the site lattice. In particular, we show that it allows us to handle irregular lattices such as those associated

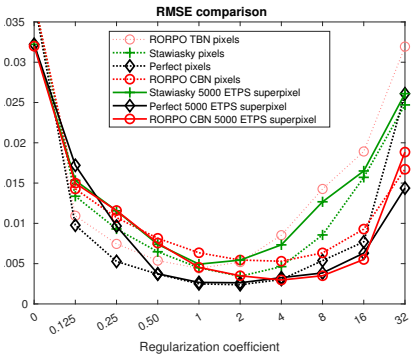


Fig. 10: Superpixel versus pixel level: Comparison in terms of RMSE (y-axis) computed on the whole dataset, for three kinds of neighborhoods.

to superpixel segmentation. For doing so, we select and customize two vesselness operators and we show their efficiency thanks to their properties of noise robustness or adaptability to thin structures. Finally, we evaluate and study the benefit of the constructed anisotropic neighborhoods in particular for thin structure preservation. Specifically, we considered SFF application and we evaluated our results on a reference dataset both according to quantitative criterion but also based on qualitative observation of evaluation maps. We showed that our approach, based on superpixel segmentation allowing to constraint depth estimation with color information and on anisotropic neighborhoods, provides an interesting alternative to classic SFF methods.

Future works will involve the following perspectives. Firstly, we aim at studying the relationships between the hyperparameters characterizing the neighborhoods and the superpixel ones (regularity, number), also relating these parameters to the scale of scene main features and objects. This study will help in the setting of these hyperparameters. Secondly, focusing on RORPO-CBN approach that appears to provide best performance and based on the evaluation of the running times per process, we will focus on the code optimization of the RORPO module. We will then provide an optimized open source code. Thirdly, since the proposed anisotropic neighborhood construction can be useful for many energetic formulations of discrete inverse problems as confirmed by preliminary tests on binary segmentation, e.g. Ribal, Lermé, and Le Hégarat-Mascle (2020), our work can be applied to the segmentation of thin structures such as frequently encountered in medical imaging (e.g., vessels) or remote sensing imagery (e.g., roads, rivers). Fourthly, in the proposed approach, neighborhood construction relies on guidance map itself estimated from a first (blind) evaluation of the solution. Now, this blind estimation can be hampered by the presence of noise on the images in the cases where acquisition conditions would be more tricky than in our dataset. Note that classic low-pass filtering will not help since it can either add blur which will prevent

accurate sharpness profile estimation or remove small objects and thin structures. Thus, the definition either of a specific filter for SFF image stack or the use of metaheuristic techniques such as alternate minimization has to be the subject of a forthcoming study.

Appendix A 3D Tensor Voting

Let $\mathbb{R}^{3\times3}$ with an origin coordinate O in $\mathbb{R}^3$ be the considered vector space, endowed with a voting function $VF : \mathbb{R}^{3\times3} \times \mathbb{R}^3 \mapsto \mathbb{R}^{3\times3}$. A tensor can be represented by a matrix $\mathbb{T} \in \mathbb{R}^{3\times3}$. The voting operation VF builds a new tensor $\mathbb{T}'$ to the cast location $P \in \mathbb{R}^3$ and adds it to the tensor at this location, since tensors have good summation properties. The tensor $\mathbb{T}'$ is a combination of rotation and scaling of the source tensor $\mathbb{T}$, combinations that are all derived from the *stick kernel*. Indeed, tensors can be decomposed in a basis of tensors, in which the stick tensor is the simplest element. Then, the stick kernel refers to the voting operation of this stick tensor.

In tensor voting, a tensor is a second order symmetric tensor that can be represented by a positive semidefinite diagonalizable matrix $\mathbb{T} \in \mathbb{R}^{3\times3}$, whose eigenvectors are orthogonal. In addition to its coordinates, one tensor can be characterized either from six scalar values corresponding to the coefficients of the symmetric matrix or, from three eigenvalues and a rotation. This rotation defines the transformation of the orthonormal basis $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ to align with $(\hat{\mathbf{e}}_0, \hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2) \in \mathbb{R}^{3\times3}$ the set of eigenvectors sorted by decreasing eigenvalue. The *decomposition* of the matrix into a set of diagonal matrices is a key point introduced by Medioni et al. (2000). By definition, the tensor is a diagonal matrix in the system $(\hat{\mathbf{e}}_0, \hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2)$, so that:

$$\begin{pmatrix} \lambda_0 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{pmatrix} = (\lambda_0 - \lambda_1)\mathbb{T}_{stick} + (\lambda_1 - \lambda_2)\mathbb{T}_{plate} + \lambda_2\mathbb{T}_{ball}, \tag{A1}$$

where $\mathbb{T}_{stick}$, $\mathbb{T}_{plate}$ and $\mathbb{T}_{ball}$ are respectively the stick tensor, the plane one and the ball one, named according to their representations as ellipsoids (see figure in Medioni, Mordohai, and Nicolescu (2005)), and each of them represents a different type of structure: The stick component encodes the saliency of surfaces that are normal to $\hat{\mathbf{e}}_0$, the plate component is encoding some curves with tangent direction $\hat{\mathbf{e}}_2$, and the ball component is encoding points, e.g. corresponding to thin structure junctions.

The stick kernel that allows for the vote cast by a stick tensor, $\mathbb{T}_{stick} \in \mathbb{R}^{3\times3}$, involves a multiplication of $\mathbb{T}_{stick}$ by a decay function DF, and a rotation by a vector $\mathbf{\Omega}$. Specifically, DF is as follows:

$$DF(r, \phi, \sigma_T) = \exp\left(-\frac{r^2 + v\phi^2}{\sigma_T^2}\right),$$

where σ_T is the scale parameter, v is a constant that controls the decay with curvature, $r \in \mathbb{R}_{>0}$ is the length of the circle arc between O and P on the osculating circle joining O and P with normal $\hat{\mathbf{e}}_0$ at point O and $\phi \in]-\pi, \pi]$ the angle between the tangent to the same osculating circle in O and $\overrightarrow{OP}$. The decay function allows for a smooth voting kernel whose support can be bounded to a sphere of radius $3\sigma_T$. Along with the term $v\phi^2$ used for increasing the decay with curvature, Medioni et al. (2000) proposes also to restrict vote to the area where $\phi < \frac{\pi}{4}$ and consider that the term $DF(r, \phi, \sigma_T)$ is null otherwise.

The rotation $\mathbf{R}(\mathbf{\Omega}) \in \mathbb{R}^{3\times3}$ is defined by the rotation vector $\mathbf{\Omega} \in \mathbb{R}^3$, that transforms the vector $\hat{\mathbf{e}}_0$ into the vector $\hat{\mathbf{e}}'_0$ with $\hat{\mathbf{e}}'_0$ and $\hat{\mathbf{e}}_0$ symmetrical with respect to the mediator of the segment OP. This allows for computing the cast tensor $\mathbb{T}'_{stick} \in \mathbb{R}^{3\times3}$ as follows:

$$\mathbb{T}'_{stick} = DF(r, \phi, \sigma_T)\mathbf{R}(\mathbf{\Omega})\mathbb{T}_{stick}\mathbf{R}^T(\mathbf{\Omega}).$$

where $\cdot^T$ is the transposition operation.

Plate tensor can be written $\mathbb{T}_{plate} = \hat{\mathbf{e}}_0\hat{\mathbf{e}}_0^T + \hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T$, while ball tensor is written $\mathbb{T}_{ball} = \hat{\mathbf{e}}_0\hat{\mathbf{e}}_0^T + \hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T$. The plate and ball kernels are derived from the stick kernel by integration of stick tensors. Approximating these integrals as sums of tensors,

$$\mathbb{T}'_{plate} \approx \sum_{i=0}^{I} DF(r, \phi, \sigma_T)\mathbf{R}(\mathbf{\Omega})\mathbb{T}_{stick}(i\Delta_\rho)\mathbf{R}^T(\mathbf{\Omega})\Delta_\rho,$$

$$\mathbb{T}'_{ball} \approx \sum_{i=0}^{I}\sum_{j=-J/2}^{J/2} DF(r, \phi, \sigma_T)\mathbf{R}(\mathbf{\Omega})\mathbb{T}_{stick}(i\Delta_\rho, j\Delta_\psi) \mathbf{R}^T(\mathbf{\Omega})\sin(j\Delta_\psi)\Delta_\psi\Delta_\rho,$$

where $\Delta_\rho = \frac{\Pi}{I}$ and $\Delta_\psi = \frac{\Pi}{J}$, and $I, J \in \mathbb{N}$ are arbitrary constants. Note that these kernels are usually precomputed for computational efficiency.

Then, any tensor $\mathbb{T}_s$ at location $s \in \mathbb{R}^3$ can be decomposed from Equation (A1) in a basis $(\hat{\mathbf{e}}_0, \hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2)$ as $\mathbb{T}(s) = (\lambda_0 - \lambda_1)\hat{\mathbf{e}}_0\hat{\mathbf{e}}_0^T + (\lambda_1 - \lambda_2)\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \lambda_2\hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T$, and the vote cast at location $t \in \mathbb{R}^3$ is written:

$$\begin{aligned} VF(\mathbb{T}, \overrightarrow{st}) = &(\lambda_0 - \lambda_1)VF(\mathbb{T}_{stick}(t), \overrightarrow{st}) \\ + &(\lambda_1 - \lambda_2)VF(\mathbb{T}_{plate}(t), \overrightarrow{st}) \\ + &\lambda_2 VF(\mathbb{T}_{ball}(t), \overrightarrow{st}) \end{aligned}$$

Having introduced the voting operation for one tensor, let us specify the global voting process.

From $\mathcal{S}_0, \mathcal{S}_1 \subset \mathcal{S}$ the sets of voters and the cast locations respectively, $\forall s \in \mathcal{S}$,

$$\begin{cases} \forall p \notin \mathcal{S}_1,\ \mathbb{T}'(p) = \mathbb{T}(p), \\ \forall p \in \mathcal{S}_1,\ \mathbb{T}'(p) = \mathbb{T}(p) + \sum_{s\in\mathcal{S}_0} VF(\mathbb{T}(s), \overrightarrow{sp}), \end{cases}$$

where $\mathbb{T}'(s)$ is the tensor at location s after vote and $\mathbb{T}(s)$ before.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S. (2012, November). SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(11), 2274–2282. Retrieved 2017-05-17, from http://ieeexplore.ieee.org/document/6205760/

10.1109/TPAMI.2012.120

Ali, U., & Mahmood, M. (2021). Robust focus volume regularization in shape from focus. *IEEE Transactions on Image Processing*, *30*, 7215-7227.

https://doi.org/10.1109/TIP.2021.3100268

Ali, U., Pruks, V., Mahmood, M.T. (2019). Image focus volume regularization for shape from focus through 3D weighted least squares. *Information Sciences*, *489*, 155–166.

https://doi.org/10.1016/j.ins.2019.03.056

Arbeláez, P., Maire, M., Fowlkes, C., Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 898–916.

https://doi.org/10.1109/TPAMI.2010.161

Boykov, Y., & Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. *Proceedings of international conference on computer vision* (Vol. 1, pp. 105–112). https://doi.org/10.1109/ICCV.2001.937505

Boykov, Y., & Kolmogorov, V. (2004, September). An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(9), 1124–1137.

https://doi.org/10.1109/TPAMI.2004.60

Cui, B., Xie, X., Ma, X., Ren, G., Ma, Y. (2018). Superpixel-based extended random walker for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *56*(6), 1–11.

https://doi.org/10.1109/TGRS.2018.2796069

Favaro, P. (2010). Recovering thin structures via nonlocal-means regularization with application to depth from defocus. *Proceedings of international*

conference on computer vision and pattern recognition (pp. 1133–1140). https://doi.org/10.1109/CVPR.2010.5540089

Fulkerson, B., Vedaldi, A., Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. *Proceedings of international conference on computer vision* (pp. 670–677). https://doi.org/10.1109/ICCV.2009.5459175

Gaganov, V., & Ignatenko, A. (2009). Robust shape from focus via Markov random fields. *Conference "GraphiCon'2009"*, 74–80.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*(6), 721–741.

https://doi.org/10.1109/TPAMI.1984.4767596

Giraud, R., Ta, V.-T., Bugeau, A., Coupe, P., Papadakis, N. (2017). SuperPatchMatch: An algorithm for robust correspondences using superpixel patches. *IEEE Transactions on Image Processing*, *26*(8), 4068–4078.

http://dx.doi.org/10.1109/TIP.2017.2708504

Gould, S., Fulton, R., Koller, D. (2009). Decomposing a scene into geometric and semantically consistent regions. *Proceedings of international conference on computer vision* (pp. 1–8). https://doi.org/10.1109/ICCV.2009.5459211

Kumar G., P., & Sahay, R.R. (2017, October). Accurate Structure Recovery via Weighted Nuclear Norm: A Low Rank Approach to Shape-from-Focus. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)* (pp. 563–574). IEEE. https://doi.org/10.1109/ICCVW.2017.73

Lai, K.-N., & Leou, J.-J. (2021). Superpixel-based multi-focus image fusion. *Advances in computer vision and computational biology* (pp. 221–233). https://doi.org/10.1007/978-3-030-71051-4_17

Liu, Y.-J., Yu, C.-C., Yu, M.-J., He, Y. (2016). Manifold SLIC: A fast method to compute content-sensitive superpixels. *Proceedings of international conference on computer vision and pattern recognition* (pp. 651–659). https://doi.org/10.1109/CVPR.2016.77

Machairas, V., Faessel, M., Cárdenas-Peña, S., Chabardes, T., Walter, T., Decencière, E. (2015). Waterpixels. *IEEE Transactions on Image Processing*, *24*(11), 3707–3716.

https://doi.org/10.1109/TIP.2015.2451011

Medioni, G., Mordohai, P., Nicolescu, M. (2005). The Tensor Voting Framework. *Handbook of Geometric Computing* (pp. 535–568). Berlin/Heidelberg: Springer-Verlag. http://dx.doi.org/10.1007/3-540-28247-5_16

Medioni, G., Tang, C.-K., Lee, M.-S. (2000). Tensor Voting: Theory and applications. *Proceedings of rfia.*

Merveille, O., Naegel, B., Talbot, H., Passat, N. (2019). n d variational restoration of curvilinear structures with prior-based directional regularization. *IEEE Transactions on Image Processing*, *28*(8), 3848–3859.

Merveille, O., Talbot, H., Najman, L., Passat, N. (2018). Curvilinear structure analysis by ranking the orientation responses of path operators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(2), 304–317.

https://doi.org/10.1109/TPAMI.2017.2672972

Moeller, M., Benning, M., Schonlieb, C., Cremers, D. (2015). Variational depth from focus reconstruction. *IEEE Transactions on Image Processing*, *24*(12), 5369–5378.

https://doi.org/10.1109/TIP.2015.2479469

Nayar, S., & Nakagawa, Y. (1994). Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(8), 824–831.

https://doi.org/10.1109/34.308479

Pei, S.-C., Chang, W.-W., Shen, C.-T. (2014). Saliency detection using superpixel belief propagation. *Proceedings of international conference on image processing* (pp. 1135–1139). http://dx.doi.org/10.1109/ICIP.2014.7025226

Pertuz, S., Puig, D., Garcia, M. (2013). Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, *46*(5), 1415–1432.

https://doi.org/10.1016/j.patcog.2012.11.011

Ribal, C., Lermé, N., Le Hégarat-Mascle, S. (2018). Efficient graph cut optimization for shape from focus. *Journal of Visual Communication and Image Representation*, *55*, 529–539.

https://doi.org/10.1016/j.jvcir.2018.06.029

Ribal, C., Lermé, N., Le Hégarat-Mascle, S. (2020). Thin structures segmentation using anisotropic neighborhoods. *Information processing and management of uncertainty in knowledge-based systems* (Vol. 1237, pp. 601–612). https://doi.org/10.1007/978-3-030-50146-4_44

Scharstein, D., & Pal, C. (2007). Learning conditional random fields for stereo. *Proceedings of international conference on computer vision and pattern recognition* (pp. 1–8). https://doi.org/10.1109/CVPR.2007.383191

Stawiaski, J., & Decencière, E. (2011). Region merging via graph-cuts. *Image Analysis & Stereology*, *27*(1), 39.

Stutz, D., Hermans, A., Leibe, B. (2018). Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, *166*, 1–27.

https://doi.org/10.1016/j.cviu.2017.03.007

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., ... Rother, C. (2008). A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(6), 1068–1080.

https://doi.org/10.1109/TPAMI.2007.70844

Tang, D., Fu, H., Cao, X. (2012, July). Topology Preserved Regular Superpixel. *2012 IEEE International Conference on Multimedia and Expo* (pp. 765–768). IEEE. Retrieved 2018-05-03, from http://ieeexplore.ieee.org/document/6298495/ 10.1109/ICME.2012.184

Ulen, J., Strandmark, P., Kahl, F. (2015). Shortest paths with higher-order regularization. *IEEE transactions on pattern analysis and machine intelligence*, *37*(12), 2588–2600.

Wang, Z., & Sheikh, H.R. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 14.

https://doi.org/10.1109/TIP.2003.819861

Yao, J., Boben, M., Fidler, S., Urtasun, R. (2015). Real-time coarse-to-fine topologically preserving segmentation. *Proceedings of international conference on computer vision and pattern recognition* (pp. 2947–2955). https://doi.org/10.1109/CVPR.2015.7298913

Yu, Y., Guan, H., Ji, Z. (2015). Rotation-invariant object detection in high-resolution satellite imagery using superpixel-based deep Hough forests. *IEEE Geoscience and Remote Sensing Letters*, *12*(11), 2183–2187.

https://doi.org/10.1109/LGRS.2015.2432135

Zou, Q., Cao, Y., Li, Q., Mao, Q., Wang, S. (2012). Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, *33*(3), 227–238.

https://doi.org/10.1016/j.patrec.2011.11.004