



**HAL**  
open science

## Exploring the Links Between Edge-Preserving Collaborative Filters via Power Watershed Framework

Sravan Danda, Aditya S Challa, B S Daya Sagar, Laurent Najman

► **To cite this version:**

Sravan Danda, Aditya S Challa, B S Daya Sagar, Laurent Najman. Exploring the Links Between Edge-Preserving Collaborative Filters via Power Watershed Framework. 2018. hal-01617799v4

**HAL Id: hal-01617799**

**<https://hal.science/hal-01617799v4>**

Preprint submitted on 24 Jan 2018 (v4), last revised 22 Nov 2018 (v6)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Exploring the Links Between Edge-Preserving Collaborative Filters via Power Watershed Framework\*

Sravan Danda<sup>†</sup>, Aditya Challa<sup>†</sup>, B.S.Daya Sagar<sup>†</sup>, and Laurent Najman<sup>‡</sup>

**Abstract.** Edge-aware filtering is an important pre-processing step in many computer vision applications. In literature, there exist collaborative edge-aware filters that work well in practice but are based only on heuristics and/or principles. For instance, Tree Filter (TF) which is proposed recently based on a minimum spanning tree (MST) heuristic yields promising results in many filtering applications. However the usage of an arbitrary MST for filtering is theoretically not justified. In this article, we introduce an edge-aware generalization of the TF, termed as *UMST filter* based on all MSTs. The major contribution of this paper is establishing theoretical links between filters based on MSTs and filters based on geodesics via power watershed framework. More precisely, we compute the  $\Gamma$ -limit of Shortest Path Filters (SPFs) and show that it is the same as UMST filter. Consequently, TF can be viewed as an approximate  $\Gamma$ -limit of the SPFs, thereby providing a theoretical basis to its working. Also, we propose and provide a detailed analysis of two different implementations of the UMST filter based on shortest paths. Further, we establish empirically with the help of denoising experiments that TF is an approximate  $\Gamma$ -limit by showing that TF and one of our approximations yield similar results.

**Key words.** Optimization, Image Filtering,  $\Gamma$ -convergence, MST, Shortest Paths

**AMS subject classifications.** 90C27, 94A08, 94A12

**1. Introduction.** Image filtering has been a fundamental problem in computer vision for several years. Edge-preserving filtering is a crucial step in many low-level vision problems such as image abstraction [34], texture removal [34], texture editing [34], scene simplification [34], stereo matching [34], optical flow [34] etc. Real world images often contain noise and irrelevant information such as texture along with the object boundaries (which are the major image structures). The goal of an image filtering algorithm is thus to preserve the image structures while getting rid of the redundant information. Hence for several of the applications, it is important for any filtering algorithm to preserve object boundaries.

In the literature, there exist several edge-aware smoothing filters such as bilateral filter (BF) [35], guided filter (GF) [26], weighted least squares filter (WLS) [22],  $L_0$  smoothing [37], propagated image filter [11], morphological amoebas or adaptive kernel based filters [27], tree filter (TF) [6] and relative total variation filter (RTV) [38] etc. Although these filters work well in practice, some of them are not extensively studied from a theoretical perspective. In

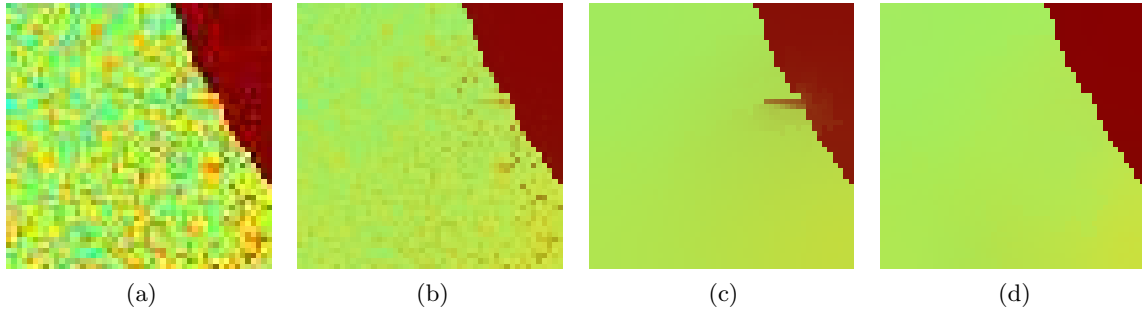
---

\*A preliminary version is published as 'Power tree filter: A theoretical framework linking shortest path filters and minimum spanning tree filters' in International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, Springer, 2017, pp 199-210.

**Funding:** This work was funded by EMR/2015/000853 SERB, ISRO/SSPO/Ch-1/2016-17 ISRO, ANR-15-CE40-0006 CoMeDiC and ANR-14-CE27-0001 GRAPHSIP research grants.

<sup>†</sup>Systems Science and Informatics Unit, Indian Statistical Institute ([sravan8809@gmail.com](mailto:sravan8809@gmail.com), [aditya.challa.20@gmail.com](mailto:aditya.challa.20@gmail.com), [bsdsagar@yahoo.co.uk](mailto:bsdsagar@yahoo.co.uk), <https://sites.google.com/site/sravandanda1988>).

<sup>‡</sup>Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Équipe A3SI, ESIEE Paris, France ([laurentnajman@esiee.fr](mailto:laurentnajman@esiee.fr), <http://www.laurentnajman.org/>).



**Figure 1.** (a) Original image (b) Bilateral filter (c) Tree filter + Bilateral filter (d) Power Tree Filter + Bilateral filter.

34 this article, we study the recent state-of-art edge-aware Tree Filter (TF) which is based on a  
 35 Minimum Spanning Tree (MST) heuristic. TF admits a linear time algorithm [39] and yields  
 36 promising results in applications such as denoising, texture removal, stereo matching and scene  
 37 simplification. However it exhibits a leak problem at some of the object boundaries. This  
 38 problem occurs due to the presence of some object boundary edges in the MST and cannot  
 39 be avoided as any spanning tree connects all the nodes in a connected graph. Although, the  
 40 authors in [6] tried to negate the leak effect using a bilateral filter as a post-processing step,  
 41 the problem still persists (see Figure 1(c)). Also, the filtering results vary with the choice of  
 42 MST which is undesirable.

43 This motivated us to explore the theoretical foundations of the TF for a deeper under-  
 44 standing on how it works. Further, the links between the TF and the other edge-aware filtering  
 45 methods might provide a possibility to design novel edge-aware filters. The main theme of this  
 46 article is to answer this question and we establish a bridge between filters based on shortest  
 47 paths and those based on minimum spanning trees. This paper is an extended version of the  
 48 conference paper [18], our contributions are the following:

- 49 1. We introduce an edge-aware filter based on the union of all MSTs of the image graph  
 50 namely UMST filter, a generalization of the TF which is an established filtering tool.
- 51 2. We compute the  $\Gamma$ -limit of the SPFs i.e. the Power Tree Filter (PTF) and show that  
 52 it is precisely given by the UMST filter (see section 4 for details). Consequently, we  
 53 provide a theoretical basis for the TF as it can be visualised as an approximate  $\Gamma$ -limit  
 54 of the SPFs.
- 55 3. We propose two different implementations of the  $\Gamma$ -limit which serve as an alternative  
 56 to the TF (see section 5 for details) with a detailed analysis on how they work.
- 57 4. We establish empirically with the help of denoising experiments that TF is an ap-  
 58 proximate  $\Gamma$ -limit by showing that TF and one of our approximations yield similar  
 59 results.

60 The rest of the paper is organized as follows: In section 2, we briefly recall the notions  
 61 of TF,  $\Gamma$ -convergence, power watershed framework and introduce UMST filter. In section 3,  
 62 we develop SPFs as edge-aware filters starting from Gaussian-like filters and discuss their  
 63 properties, links with other geodesic based methods. In section 4, we compute the  $\Gamma$ -limit of

64 the SPFs and show that it is precisely the UMST filter. In [section 5](#), we discuss approximations  
 65 of UMST filter i.e. TF and propose two approximations based on shortest paths. We provide  
 66 a detailed analysis of each of these implementations and empirically establish with the help  
 67 of denoising experiments that TF and one of our approximations yield similar results. In  
 68 [section 6](#), the conclusions follow and we speculate some possible directions to extend the ideas  
 69 in the paper.

70 **2. Union Minimum Spanning Tree Filter.** In this section, we briefly recall the Tree Filter  
 71 (TF) and provide our motivation on why one should consider using a filter based on union of  
 72 all the MSTs (UMST).

73 **2.1. Tree Filter.** Suppose  $I$  is a given image which possibly contains noise, we let  $I_i$   
 74 denote the color or intensity of the pixel  $i$  in the image  $I$ . Let  $S$  denote the tree filtered  
 75 image. The authors in [\[6\]](#) construct a 4-adjacency edge-weighted graph, with the weights  
 76 between adjacent pixels reflecting the color or intensity dissimilarity. More formally, if  $i$  and  
 77  $j$  are 4-adjacent pixels, they use  $w_{ij}$  defined by

$$78 \quad (1) \quad w_{ij} = ||I_i - I_j||$$

79 One can construct a MST on this edge-weighted graph,  $I_{MST}$ . Since a spanning tree connects  
 80 every pair of pixels and does not contain cycles, there exists a unique path between every pair  
 81 of pixels. Let  $D(i, j)$  denote the number of edges on the path between  $i$  and  $j$  on  $I_{MST}$ . For  
 82 each pair  $i$  and  $j$ , the collaborative weights  $t_i(j)$  are given by:

$$83 \quad (2) \quad t_i(j) = \frac{\exp(-\frac{D(i,j)}{\sigma})}{\sum_q \exp(-\frac{D(i,q)}{\sigma})}$$

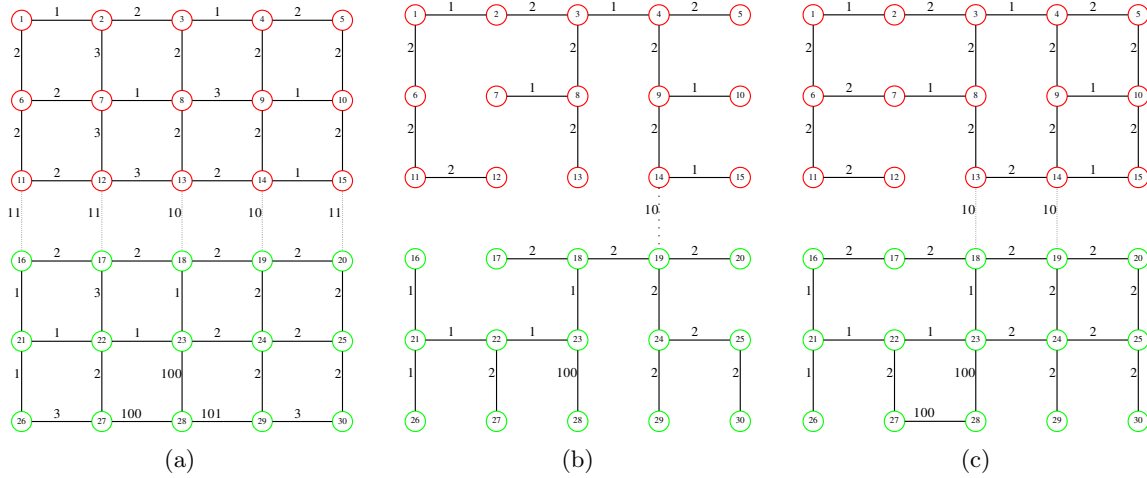
84 where  $\sigma$  controls the falling rate and the summation over  $q$  is over all the pixels in the graph.  
 85 The tree filtered value at pixel  $i$  is given by

$$86 \quad (3) \quad S_i = \sum_j t_i(j)I_j$$

87 Here the summation over  $j$  is over all the pixels in the graph.

88 It is reasonable to assume that the pixel color or intensities vary vastly across objects  
 89 and are similar within objects. In other words, the higher weight edges mostly correspond  
 90 to object boundaries and lower weight edges mostly correspond to object interiors. We shall  
 91 work under this assumption in the rest of the article. The TF works on the following intuition:  
 92 most of the higher weight edges do not appear while the lower weight edges mostly do appear  
 93 in any MST. The collaboration across object boundaries is thus low while smoothing within  
 94 objects is achieved well.

95 **2.2. Why UMST filter?.** The edges in the image graph that do not belong to object  
 96 boundaries induce a disconnected subgraph. On the other hand, a MST of a graph is con-  
 97 nected, hence any MST contains one or more object boundary edges. These edges cause a  
 98 leak effect in the tree filtered image (see [Figure 1\(c\)](#)). Also, the filtering results vary with the



**Figure 2.** (a) 4-adjacency graph of a synthetic image containing two objects coloured in red and green. The pixels in this image are indexed from 1 to 30 and the weights on the edges denote the intensity dissimilarities. The edges corresponding to object boundaries are represented by dotted lines, (b) A MST obtained from (a), and (c) UMST obtained from (a). In order to illustrate that UMST filter yields better results, it should perform at least as good as TF for - removing noise at pixel numbered 28 and reducing the leak at object boundaries say at pixel numbered 13 and 14. Consider pixel numbered 28. One can see that both the edges of weights 100 incident on this pixel are present in the UMST, the noise removal is enhanced due to higher collaboration with the neighbouring pixels when compared to that of tree filter where MST had only one of the edges with weight 100. Now consider the pixel numbered 13. We see that although an extra boundary edge (edge 13 – 18) appears in the UMST, the presence of an additional interior edge incident on 13 in the UMST nullifies the effect of the boundary edge collaboration. At pixel numbered 14, the UMST filter performs better than tree filter due to the presence of the additional interior edge 13 – 14.

99 MST used, making the choice of an arbitrary MST debatable. On the other hand, a filter  
 100 based on UMST would ensure the following:

- 101 1. The filtering results would not depend on arbitrary MST computations.
- 102 2. There would be a significant reduction in the leak effect when compared to the TF  
 103 (see Figure 1(d)).

104 The first property is a direct consequence of the fact that UMST filter uses all the MSTs  
 105 of the image graph. The second property can be explained intuitively as follows: The edges  
 106 in the UMST is a superset of the edges of an arbitrary MST. Now, among the edges that  
 107 belong to UMST but not the MST are mostly object interior edges. These object interior  
 108 edges dominate the collaborative effect of the object boundary edges to ensure a reduction in  
 109 the leakage. Figure 2 illustrates the above properties on a synthetic image.

110 Extending the idea of TF, we use an exponential falling weight similar to (2) for computing  
 111 collaborative weights. However, we observe that there are possibly multiple paths between  
 112 a given pair of pixels  $i$  and  $j$  in the UMST. In order to define the collaborative weights of  
 113 the UMST filter, we need a criterion to choose a path among all the paths between  $i$  and  $j$ .  
 114 We consider  $\eta(i, j)$ , the number of edges on a path with smallest dictionary or lexicographic  
 115 order of edge weights (see Definition 3.4 and Definition 4.1) replacing  $D(i, j)$  in (2). This is  
 116 a natural way to generalize the TF since: (a) the lesser the lexicographic order of a path, the

117 lesser the chance of the path crossing an object boundary, (b) in the special case of the graph  
 118 having a unique MST, this filter is exactly the same as TF.

119 **2.3. Lexicographic Ordering and Pass Values.** Lexicographic order of a path is related  
 120 very closely to the notion of pass values used in segmentation. Pass value [16] or the minimax  
 121 distance [21] between a pair of nodes is the minimum of the  $l^\infty$  norm over all the paths  
 122 between them. To the best of our knowledge, this feature was first used in image filtering  
 123 in [33]. Pass values between different minima of a gradient image is a measure of contrast  
 124 difference between objects in watershed segmentation [30]. One can observe that any image  
 125 transformation on the gradient image that keeps the object boundaries intact preserves the  
 126 contrast difference between objects. It is hence a desired condition for a segmentation method  
 127 to preserve the contrast difference (topological watersheds [14] for instance).

128 Observe that given pixels  $i$  and  $j$ , a path with smallest lexicographic order of the edge-  
 129 weights would be a special case of a path with smallest  $l^\infty$  norm. In simpler words, the smallest  
 130 lexicographic order path is a critical path that determines the contrast difference between a  
 131 pair of pixels. In practice, a path with smallest lexicographic order would be unique. This  
 132 serves as a tie-breaker on choosing a critical path among the smallest  $l^\infty$  norm paths thus  
 133 reducing the ambiguity.

134 Now, we shall recall notions of  $\Gamma$ -convergence and the power watershed framework before  
 135 we state the main result of the paper.

136 **2.4.  $\Gamma$ -convergence, Power Watershed Framework and UMST Filter.**  $\Gamma$ -convergence  
 137 [8] is the study of asymptotic behaviour of a sequence of minimization problems. Suppose  
 138 for each  $n \in \mathbb{N}$ ,  $F_n : \mathbb{R}^l \rightarrow \mathbb{R}$  is a cost function such that  $\arg \min_{x \in \mathbb{R}^l} (F_n(x)) \neq \emptyset$ , what  
 139 does  $\lim_{n \rightarrow \infty} x_n$  minimize (assuming the limit exists) where  $x_n \in \arg \min_{x \in \mathbb{R}^l} (F_n(x))$ ? In  
 140 other words, in what sense do  $F_n$  converge to  $F$  where  $F : \mathbb{R}^l \rightarrow \mathbb{R}$  so that  $\lim_{n \rightarrow \infty} x_n \in$   
 141  $\arg \min_{x \in \mathbb{R}^l} (F(x))$ . In general, the usual notions of pointwise and uniform convergence do  
 142 not convey information about the limiting behaviour of minimizers of functionals and  $\Gamma$ -  
 143 convergence is a more appropriate mode of convergence. In simple words, it provides sufficient  
 144 conditions under which one can approximate a minimizer of  $F$  using the minimizers of  $F_n$ .  
 145 We refer the interested reader to [8] for a comprehensive study of  $\Gamma$ -convergence.

146  $\Gamma$ -convergence has been proved to be very useful in many computer vision applications  
 147 especially the ones based on variational formulations. Classically,  $\Gamma$ -convergence is used in the  
 148 cases where it is known that a sequence of functions  $F_n$  gamma converge to  $F$  and it is difficult  
 149 to compute the minimizer of  $F$  directly. For example, the Ambrosio-Tortorelli approximation  
 150 [2] of the Mumford-Shah functional by a sequence of smooth phase field functionals is one  
 151 such instance. In such cases, one typically picks a large  $n$  and investigates if the minimizers  
 152 of  $F_n$  exhibit the desired properties of the minimizer of  $F$ .

153 However, studying the  $\Gamma$ -limit of minimizers on finite graphs [19] have been useful in recent  
 154 times. In some cases, an explicit computation of a  $\Gamma$ -limit is possible and provides useful  
 155 insights. The following are a few such instances : In [13], the power watershed framework  
 156 unifies and extends a common framework of semi-supervised or seeded graph-based image  
 157 segmentation methods namely graph cuts [7], random walker [24], geodesics [1, 5, 17, 21]  
 158 and watershed cuts [15, 16]; More recently, [29] generalizes the power watershed framework  
 159 for a larger class of cost functions (Eqns. 1 and 2 in [29]) and formalizes the notion of

160 scale for investigating minimizers. In simple words, [29] develops theory and algorithms to  
 161 combine data reduction and optimization techniques; In [31, 3], the elementary mathematical  
 162 morphological (MM) operators have been formulated as limits of variational problems; and  
 163 in [36], the authors view the local min-max filters as a limit of normalized power-weighted  
 164 averaging filter; In [10], the authors propose a fast alternative to spectral clustering methods  
 165 by considering their  $\Gamma$ -limit.

166 As edge-preserving image filtering and image segmentation are closely related problems,  
 167 one can anticipate to establish links between existing filtering methods using the power wa-  
 168 tershed framework [13, 29]. In fact, we prove the following theorem which is the main result  
 169 of the paper: *UMST filter is the  $\Gamma$ -limit of shortest path edge-aware filters.* Although both  
 170 [29] and this article involve the computation of  $\Gamma$ -limit of parametrized cost functions, it is  
 171 important to note that the cost functions used in this article do not fall in the framework  
 172 of Eqns. 1 and 2 in [29]. See section 3 for a formal definition of shortest path filters and  
 173 section 4 for a proof. This result implies that one can view the UMST filter and the shortest  
 174 path filters in an optimization framework.

175 **3. Shortest Path Filters and Related Methods.** In this section, we shall review the  
 176 shortest path filters in detail. In particular, we develop them as a natural edge-aware extension  
 177 of Gaussian-like filters. The rest of the section is dedicated to the discussions on their links  
 178 with other related geodesic methods.

179 Before formally defining the SPF, we need some notions of graphs that we define below.

### 180 3.1. Basic Notions.

181 **Definition 3.1.** An *edge-weighted graph*  $\mathcal{G} = (V, E, W)$  consists of a finite set  $V$  of nodes,  
 182 and set of unordered pairs of elements of  $V$  i.e.  $\{\{x, y\} \subset V : x \neq y\}$ , called the edge set  $E$ ,  
 183 a positive real-valued function  $W$  on the set  $E$ . We denote  $w_{ij}$  or  $W(e_{ij})$  as the weight of the  
 184 edge joining pixels  $i$  and  $j$ .

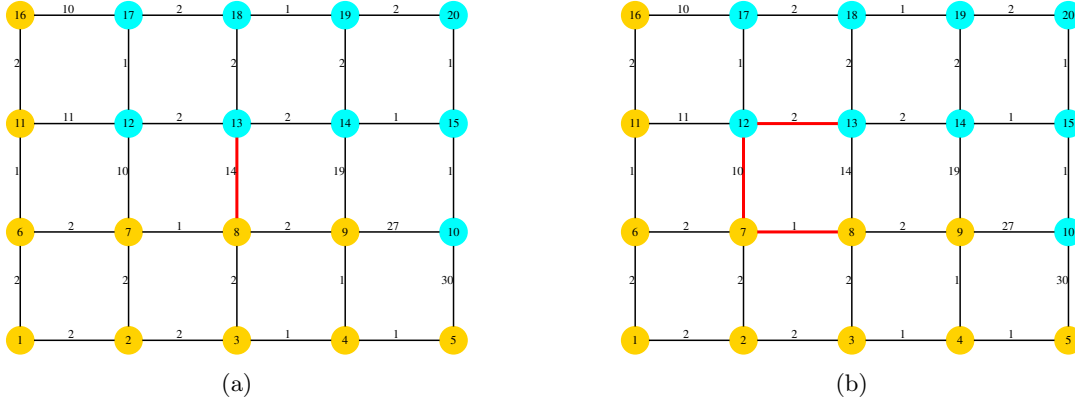
185 **Definition 3.2.** For  $p \in \mathbb{Z}^+$ , we denote by  $\mathcal{G}^{(p)} = (V, E, W^{(p)})$ , the graph that contains the  
 186 same set of nodes and edges as of  $\mathcal{G}$  and  $W^{(p)}(e_{ij}) = (W(e_{ij}))^p$  for each edge  $e_{ij} \in E$  and we  
 187 call  $\mathcal{G}^{(p)}$  as an **exponentiated graph** of  $\mathcal{G}$ .

188 **Definition 3.3.** A **path**  $P(i, j)$  between nodes  $i$  and  $j$  is a finite ordered sequence of nodes  
 189 of  $\mathcal{G}$  such that there is an edge incident on every adjacent pair of nodes in the sequence. We  
 190 say that a path from  $i$  to  $j$  is a **simple path** if all the nodes in the sequence are distinct.

191 **Definition 3.4.** Assume that the distinct weights in  $\mathcal{G}$  are given by  $0 < w_1 < w_2 < \dots < w_k$ .  
 192 Given a path  $P(i, j)$  in  $\mathcal{G}$ , one can assign a  $k$ -tuple  $(n_1, \dots, n_k)$  to the path, where  $n_r$  denotes  
 193 the number of edges of weight  $w_r$  on the path  $P(i, j)$ . This  $k$ -tuple is referred to as the **edge-**  
 194 **weight distribution** of the path  $P(i, j)$ .

195 We remark that the  $k$ -tuples associated with a path  $P(i, j)$  in graph  $\mathcal{G}$  and its exponentiated  
 196 graph  $\mathcal{G}^{(p)}$  are identical by the virtue of it's definition. However it is important to note that:  
 197 corresponding to each of the coordinates, the weights in the edge-weight distributions are  
 198 different.

199 **Definition 3.5.** Assume that the distinct weights in  $\mathcal{G}$  are given by  $0 < w_1 < w_2 < \dots <$   
 200  $w_k$ . Suppose  $P(i, j)$  is a path between pixels  $i$  and  $j$  in  $\mathcal{G}$ .  $P(i, j)$  is said to be a **shortest**



**Figure 3.** (a) and (b) 4-adjacency graph of a synthetic image containing two objects coloured in blue and yellow. The pixels in this image are indexed from 1 to 20 and the weights on the edges denote the intensity dissimilarities. The collaborative weight for the pair of pixels 8 and 13 (which crosses an object boundary) given by Gaussian-like filter would consider the edge highlighted in red in (a) and yields a high value. On the other hand, the SPF considers the path  $\langle 8, 7, 12, 13 \rangle$  highlighted in red in (b) for computing the corresponding collaborative weight. This illustrates that SPF respects the object boundaries.

201 **path** between the pixels  $i$  and  $j$  in  $\mathcal{G}$  if for every path  $Q(i, j)$  between  $i$  and  $j$  in  $\mathcal{G}$ , we  
 202 have  $\sum_{r=1}^k n_r w_r \leq \sum_{r=1}^k m_r w_r$  where  $(n_1, \dots, n_k)$  and  $(m_1, \dots, m_k)$  denote the edge-weight  
 203 distributions of  $P(i, j)$  and  $Q(i, j)$  respectively.

204 We remark that a shortest path is always a simple path since all the weights in the edge-  
 205 weighted graphs are strictly positive.

206 We shall now build an edge-aware filter from scratch: Let  $i$  and  $j$  be two pixels in  $\mathcal{G}^{(p)}$ .  
 207 Consider the simplest weighted-average filter whose collaborative weights are given by

$$208 \quad (4) \quad g_i(j) = \frac{\exp\left(-\frac{\|i-j\|}{\sigma}\right)}{\sum_k \exp\left(-\frac{\|i-k\|}{\sigma}\right)}$$

209 where  $\|i-j\|$ :  $l_1$  norm between the pixels  $i$  and  $j$  and  $\sigma$  is the parameter controlling the  
 210 level of smoothing.

211 We observe that this is a Gaussian-like filter and collaborative weights purely depend on  
 212 the spatial distance between pixels  $i$  and  $j$ . More specifically, collaborative weights between  
 213 pixels separated by same distance is indifferent w.r.t. existence of an object boundary between  
 214 them. Hence, one has to find a way to ensure that the collaborative weights are lesser across  
 215 boundaries. A natural way to extend the idea of a Gaussian-like filter is: given a pair of pixels  
 216  $i$  and  $j$ , consider the number of edges on a path with smallest sum of weights between them.

217 Let  $\Pi(P(i, j))$  denote the number of edges on a path  $P(i, j)$ . Define

$$218 \quad (5) \quad \Theta^{(p)}(i, j) = \inf\{\Pi(P(i, j)) \text{ where } P(i, j) \text{ is a shortest path in } \mathcal{G}^{(p)}\}$$



219 where the edge weights are given by:

$$220 \quad (6) \quad w_{ij} = \|I_i - I_j\| + 1$$

221 The SPF at pixel  $i$  is defined as:

$$222 \quad (7) \quad S_i^{(p)} = \sum_j \frac{\exp\left(-\frac{\Theta^{(p)}(i,j)}{\sigma}\right)}{\sum_k \exp\left(-\frac{\Theta^{(p)}(i,k)}{\sigma}\right)} I_j,$$

223 where  $\sigma$  controls the falling rate and the summations are over all pixels in the image.

224 Note that the weights in Eq. (6) are different from the ones in Eq. (1) to ensure that  
 225 the weights are strictly positive. We use the edge weights as per (6) in the rest of the paper.  
 226 Since, we use an increasing transformation on edge weights, shortest paths and MSTs are  
 227 invariant to the modification. This assumption is needed to ensure that the converse part in  
 228 [Lemma 4.3](#) holds.

229 We remark that in the special case of all pixel values being equal in the image, SPF is  
 230 exactly Gaussian-like filter. Also, in practice, the shortest path distances between a pair of  
 231 pixels within the objects are close to the spatial distances and are larger than the spatial  
 232 distances across object boundaries. [Figure 3](#) illustrates on a synthetic image that the SPF is  
 233 a natural edge-preserving extension of the Gaussian-like filter.

234

235 Further, one can see that the SPF value at pixel  $i$  is a solution of the following optimization  
 236 problem:

237 Consider the cost function

$$238 \quad (8) \quad Q_i^{(p)}(x) = \sum_j \exp\left(-\frac{\Theta^{(p)}(i,j)}{\sigma}\right) (x - I_j)^2$$

239 where  $\sigma$  controls the falling rate and the summation is over all pixels. The shortest path  
 240 filtered value at pixel  $i$  is given by the minimizer of  $Q_i^{(p)}(x)$  i.e.

$$241 \quad S_i^{(p)} = \arg \min_x Q_i^{(p)}(x)$$

242 SPFs are not completely new and there exist in literature, several edge-preserving filters  
 243 using geodesics such as the ones discussed in [\[25\]](#), the adaptive kernel filter such as morpho-  
 244 logical amoebas [\[27\]](#).

245 **3.2. Relation to Morphological Amoebas.** SPFs are also closely related to Morphological  
 246 Amoebas. Morphological Amoebas are adaptive structuring elements based on shortest path  
 247 distances used to build edge-aware filters. These kernels work on the assumption that the  
 248 gradients are low within the objects and high across the object boundaries. In order to ensure

249 that the kernels do not cross the object boundaries, the amoeba distance defined below is used  
 250 to compute them:

$$251 \quad (9) \quad \kappa(i, j) = \min_{P(i, j)} L_\lambda(P(i, j))$$

252 where  $P(i, j)$  is a path between pixels  $i$  and  $j$ ,  $\langle i = x_0, x_1, \dots, x_n = j \rangle$  and  $\lambda \geq 0$  is a  
 253 user input.

$$254 \quad (10) \quad L_\lambda(P(i, j)) = \sum_{t=0}^{n-1} (1 + \lambda \|I_{x_{t+1}} - I_{x_t}\|)$$

255 The closed ball at pixel  $i$  given by  $\{j : \kappa(i, j) \leq r\}$  is the kernel used for edge-aware  
 256 smoothing. The cardinality or the size of the kernel depends on  $r$  and is chosen as per the  
 257 level of smoothing desired.

258 **Proposition 3.6.**  $\Theta^{(1)}(i, j)$  is a constrained minima of the amoeba kernel path length given  
 259 by

$$260 \quad (11) \quad \Theta^{(1)}(i, j) = \min L_0(P(i, j)) \text{ subject to } P(i, j) \in \arg \min L_1(P(i, j))$$

261 Further, the family of parameters  $\Theta^{(p)}(i, j)$  are given by:

$$262 \quad (12) \quad \Theta^{(p)}(i, j) = \min L_0^p(P(i, j)) \text{ subject to } P(i, j) \in \arg \min L_1^p(P(i, j))$$

263 where

$$264 \quad (13) \quad L_\lambda^p(P(i, j)) = \sum_{t=0}^{n-1} (1 + \lambda \|I_{x_{t+1}} - I_{x_t}\|)^p$$

265

266 *Proof.* It readily follows from (5), (6), (10) and (13) ■

267 Note that the morphological amoeba lengths are a special case of the lengths given by (13).  
 268 We can hence view the SPF as a generalization of the notion of morphological amoeba lengths.

269 **4. UMST Filter: Gamma Limit of Shortest Path Filters.** In this section, we shall prove  
 270 that the UMST filter is the  $\Gamma$ -limit of SPFs. As the weights of the graphs of the shortest path  
 271 filters are powers of natural numbers (see section 3), we use the term Power Tree Filter to  
 272 denote the  $\Gamma$ -limit of Shortest Path Filters. We shall need some definitions before we prove  
 273 this result.



**Figure 4.** (a) and (b) synthetic images to illustrating the lexicographic ordering of paths. The lexicographic order of path in blue is lesser than that of the one in green

#### 274 4.1. Some Definitions.

275 **Definition 4.1.** Assume that graph  $\mathcal{G}$  has  $k$  distinct weights given by  $w_1 < \dots < w_k$ . Let  
 276  $(n_1, \dots, n_k)$  and  $(m_1, \dots, m_k)$  denote the edge-weight distributions of paths  $P$  and  $Q$  in  $\mathcal{G}$   
 277 respectively. Let  $l = \sup(A)$  where  $A = \{r : 1 \leq r \leq k, n_r \neq m_r\}$  We define **dictionary**  
 278 **ordering** or **lexicographic ordering** on the set of paths in  $\mathcal{G}$  as follows:

$$279 \quad (14) \quad P \geq Q \Leftrightarrow A = \emptyset \text{ or } n_l > m_l$$

280 See Figure 4 for an illustration on dictionary ordering. Note that dictionary ordering  
 281 yields a complete ordering on the set of paths in  $\mathcal{G}$  and the ordering remains same in each of  
 282 the exponentiated graphs  $\mathcal{G}^{(p)}$ .

283 **Definition 4.2.** Suppose  $P(i, j)$  is a path between pixels  $i$  and  $j$  in  $\mathcal{G}$ .  $P(i, j)$  is said to be  
 284 a **smallest path w.r.t. dictionary order** between the pixels  $i$  and  $j$  in  $\mathcal{G}$  if for every path  
 285  $Q(i, j)$  between  $i$  and  $j$  in  $\mathcal{G}$ , we have  $Q(i, j) \geq P(i, j)$ .

286 Note that every smallest path w.r.t. dictionary order between pixels  $i$  and  $j$  in  $\mathcal{G}$  has the  
 287 same edge-weight distribution. In particular, the number of edges on a smallest path w.r.t.  
 288 dictionary order between  $i$  and  $j$  denoted by  $\Pi(i, j)$  is well-defined.

289 Any MST in  $\mathcal{G}^{(p)}$  is a MST in  $\mathcal{G}$  and vice-versa. This follows directly from the fact that  
 290 MST is invariant to any strictly increasing transformation on the weights of a connected  
 291 graph. The notions of smallest paths w.r.t. dictionary order and that of MSTs in  $\mathcal{G}^{(p)}$  are  
 292 hence independent of  $p$ .

293 **4.2. Gamma Limit of Shortest Path Filters.** In this subsection, we characterize the  
 294 Power Tree Filter or  $\Gamma$ -limit of Shortest Path Filters. Firstly, we have the following result:

295 **Lemma 4.3.** Let  $\mathcal{G} = (V, E, W)$ . For every pair of pixels  $i$  and  $j$  in  $V$ , there exists  $p_0 \geq 1$   
 296 such that, a path  $P(i, j)$  is a shortest path between  $i$  and  $j$  in  $\mathcal{G}^{(p)}$  for all  $p \geq p_0$  if and only  
 297 if  $P(i, j)$  is a smallest path w.r.t. dictionary order between  $i$  and  $j$  in  $\mathcal{G}$ . Further,  $p_0$  is  
 298 independent of  $i$  and  $j$ .

299 *Proof.* Let  $\mathcal{G} = (V, E, W)$  and let the distinct weights in  $\mathcal{G}$  be given by  $w_1 < \dots < w_k$ .

300 Firstly, we shall show that for a given pair of pixels  $i$  and  $j$ , if  $P(i, j)$  is a smallest path  
 301 w.r.t. dictionary order between  $i$  and  $j$  in  $\mathcal{G}$  then there exists a constant  $p_0$  such that for  
 302 each  $p \geq p_0$ ,  $P(i, j)$  is a shortest path between  $i$  and  $j$  in  $\mathcal{G}^{(p)}$ . Let  $P(i, j)$  be a small-  
 303 est path w.r.t. dictionary order between  $i$  and  $j$  in  $\mathcal{G}$ . Let  $Q(i, j)$  be an arbitrary simple

304 path between  $i$  and  $j$ . Let  $(n_1, \dots, n_k)$  and  $(m_1, \dots, m_k)$  denote the edge-weight distribu-  
 305 tions of paths  $P(i, j)$  and  $Q(i, j)$  respectively. Let  $A(P, Q) = \{1 \leq r \leq k : n_r \neq m_r\}$ .  
 306 Suppose  $A(P, Q) = \emptyset$  then  $\sum_{r=1}^k n_r w_r^p \leq \sum_{r=1}^k m_r w_r^p \forall p \geq 1$ . If  $A(P, Q) \neq \emptyset$  then let  
 307  $l = \sup(A(P, Q))$ . We have  $m_l > n_l$  by choice of  $P(i, j)$ . Also, the difference of the total  
 308 weights i.e.  $\sum_{r=1}^k m_r w_r^p - \sum_{r=1}^k n_r w_r^p = \Theta(w_l^p)$  with a positive leading coefficient. Hence  
 309  $\exists p_{Q(i,j)} \geq 1$  such that  $\sum_{r=1}^k n_r w_r^p \leq \sum_{r=1}^k m_r w_r^p \forall p \geq p_{Q(i,j)}$ . Now, let  $\mathcal{S}_{ij}$  denote the set of  
 310 all simple paths from  $i$  to  $j$ . Then  $|\mathcal{S}_{ij}| < \infty$ . Set  $p_{ij} = \sup\{p_{Q(i,j)} : Q(i, j) \in \mathcal{S}_{ij}\} < \infty$ . We  
 311 note that given any path which is not simple, one can drop the redundant edges to construct a  
 312 simple path with strictly smaller total weight. It is hence enough to show that the total weight  
 313 of  $P(i, j)$  is lesser than or equal to every simple path between  $i$  and  $j$  in  $\mathcal{G}^{(p)}$  for sufficiently  
 314 large  $p$ . As  $V$  is finite, setting  $p_0 = \sup\{p_{ij} : i, j \in V\}$  completes the argument.

315 Conversely, suppose  $P(i, j)$  is NOT a smallest path w.r.t. dictionary ordering between  
 316  $i$  and  $j$ , we shall construct a sequence  $(p_n)_{n \geq 1}$  converging to  $\infty$  such that  $P(i, j)$  is not  
 317 a shortest path between  $i$  and  $j$  in  $\mathcal{G}^{(p_n)}$  for each  $n \geq 1$ . Since  $P(i, j)$  is not a smallest  
 318 path w.r.t. dictionary ordering between  $i$  and  $j$ ,  $\exists$  a path  $T(i, j)$  between  $i$  and  $j$  such that  
 319  $P(i, j) \geq T(i, j)$  holds but  $T(i, j) \geq P(i, j)$  does not hold. Equivalently, if the edge weight  
 320 distributions of  $P(i, j)$  and  $T(i, j)$  are given by  $(n_1, \dots, n_k)$  and  $(t_1, \dots, t_k)$  respectively then  
 321  $A(P, T) \neq \emptyset$  and  $t_l < n_l$  where  $l = \sup(A(P, T))$  and  $A(P, T) = \{1 \leq r \leq k : n_r \neq t_r\}$ .  
 322 The difference of the total weights i.e.  $\sum_{r=1}^k t_r w_r^p - \sum_{r=1}^k n_r w_r^p = \Theta(w_l^p)$  and has a negative  
 323 leading coefficient. Thus,  $\exists$  a constant  $\rho_{ij} \geq 1$  such that for each  $p \geq \rho_{ij}$ ,  $P(i, j)$  is not a  
 324 shortest path between  $i$  and  $j$ . We set  $(p_n)_{n \geq 1}$  by  $p_n = \rho_{ij} + n - 1$  to complete the proof.  $\blacksquare$

325 Loosely speaking, for a large enough power  $p$ , on  $\mathcal{G}^{(p)}$ , between every pair of pixels, a  
 326 shortest path is a smallest path w.r.t. dictionary order and vice-versa. In short, we have the  
 327 following corollary:

328 **Corollary 4.4.** *Let  $\Delta(i, j)$  denote the number of edges on a smallest path w.r.t. dictionary*  
 329 *order between pixels  $i$  and  $j$  in  $\mathcal{G}$ . As  $p \rightarrow \infty$ , we have  $\Theta^{(p)}(i, j) \rightarrow \Delta(i, j)$  for each pair  $i$  and*  
 330  *$j$ .*

331 **4.3. Optimization Framework for UMST Filter.** Recall the definition of UMST filter  
 332 from section 2. Let  $\mathcal{G}$  denote the given image, let  $\mathcal{G}_{UMST}$  denote the UMST on the edge-  
 333 weighted graph constructed from  $\mathcal{G}$ . Let  $\eta(i, j)$  denote the number of edges on a smallest path  
 334 w.r.t. dictionary order between pixels  $i$  and  $j$  in  $\mathcal{G}_{UMST}$ . Now consider the cost function given  
 335 by:

$$336 \quad (15) \quad \hat{Q}_i(x) = \sum_j \exp\left(-\frac{\eta(i, j)}{\sigma}\right) (x - I_j)^2$$

337 where  $\sigma$  controls the falling rate and the summation is over all pixels, the UMST filtered  
 338 value at pixel  $i$  is given by the minimizer of  $\hat{Q}_i(x)$ .

$$339 \quad (16) \quad U_i = \arg \min \hat{Q}_i(x) = \sum_j \frac{\exp\left(-\frac{\eta(i, j)}{\sigma}\right)}{\sum_k \exp\left(-\frac{\eta(i, k)}{\sigma}\right)} I_j,$$

340 where  $U$  denotes the UMST filtered image.

341 Firstly, we need the following:

342 **Lemma 4.5.** (*Cut Property*) For any cut  $C$  of a connected graph  $\mathcal{G} = (V, E, W)$ , if the  
343 weight of an edge  $e$  in the cut-set  $C$  is not larger than the weights of all other edges in  $C$ , then  
344 this edge belongs to a MST of the graph  $\mathcal{G} = (V, E, W)$ .

345 **Lemma 4.6.** (*Cycle Property*) For any cycle  $C$  in the graph  $\mathcal{G} = (V, E, W)$ , if the weight  
346 of an edge  $e$  of  $C$  is larger than the individual weights of all other edges of  $C$ , then this edge  
347 cannot belong to a MST.

348 Before we prove the main result of the paper, we need **Lemma 4.7** which is a modified  
349 version of a result stated in [29].

350 **Lemma 4.7.** Let  $\mathcal{G} = (V, E, W)$  be an edge-weighted graph. Let  $\mathcal{G}_{<w}$  denote the induced  
351 subgraph of  $\mathcal{G}$  with the vertex set  $V$  and all the edges  $e_{ij} \in E$  whose weight  $w_{ij} < w$ . Let  
352  $\mathcal{G}_{UMST}$  denote the UMST of  $\mathcal{G}$ . Then an edge  $e$  with weight  $w(e)$  belongs to the  $\mathcal{G}_{UMST}$  if and  
353 only if the edge  $e$  joins two connected components in  $\mathcal{G}_{<w(e)}$ .

354 *Proof.* The proof directly follows from **Lemma 4.5** and **Lemma 4.6**. ■

355 In simple words, an edge  $e$  is in some MST if and only if it connects two different compo-  
356 nents of the induced subgraph generated by edges of weights lower than that of  $e$ .

357 **Proposition 4.8.** Every smallest path w.r.t. dictionary order between any two arbitrary  
358 nodes in  $\mathcal{G} = (V, E, W)$  lies on a MST of  $\mathcal{G}$  and hence on the union of MSTs of  $\mathcal{G}$ .

359 *Proof.* Let  $i$  and  $j$  be two arbitrary nodes in  $\mathcal{G}$ . Let  $P(i, j)$  be a smallest path w.r.t.  
360 dictionary order between  $i$  and  $j$ . It is now enough to show that every edge in the path  $P(i, j)$   
361 satisfies the characterization given in **Lemma 4.7**. Suppose if possible, let  $e \in P(i, j)$  be of  
362 smallest possible weight such that  $e$  is incident on nodes in a same connected component of  
363  $\mathcal{G}_{<w(e)}$ . Adding  $e$  thus forms a cycle  $C$  and the other edges in  $C$  have weights strictly less  
364 than  $w(e)$ .

365 Now, consider the subgraph generated by edges in  $P(i, j) \cup C \setminus \{e\}$ . This subgraph is  
366 connected and hence there exists a path  $Q(i, j)$  (say) between  $i$  and  $j$ . It is easy to see that  
367  $Q(i, j)$  has smaller dictionary order compared to  $P(i, j)$ : number of edges of weight greater  
368 than  $w(e)$  in  $Q(i, j)$  cannot exceed to that in  $P(i, j)$  since  $C$  has edges of weight strictly less  
369 than  $w(e)$ ; number of edges of weight  $w(e)$  in  $Q(i, j)$  is at least one less than that of  $P(i, j)$ .  
370 This contradicts the fact that  $P(i, j)$  is a smallest path w.r.t. dictionary order between  $i$  and  
371  $j$ . ■

372 **Corollary 4.9.** For every pair  $i$  and  $j$  in  $\mathcal{G}$ , we have  $\eta(i, j) = \Delta(i, j)$

373 The main result is formally stated as follows:

374 **Theorem 4.10.** As  $p \rightarrow \infty$ , we have the following:

$$375 \quad (17) \quad Q_i^{(p)}(x) \xrightarrow{\Gamma} \widehat{Q}_i(x)$$

376 In other words, the shortest path filters converge to the UMST filter as  $p \rightarrow \infty$  i.e.

$$377 \quad (18) \quad S_i^{(p)} \longrightarrow U_i$$

378

379 *Proof.* The proof follows readily from Proposition 4.8 and Lemma 4.3. ■

380 The fact that UMST filter is a  $\Gamma$ -limit of shortest path filters is useful: The state-of-art  
 381 algorithms in shortest paths and MSTs can be jointly exploited to obtain novel algorithms to  
 382 compute UMST filter.

383 **5. Implementation.** In this section, we discuss several possible implementations of UMST  
 384 filter. We utilize ideas from shortest paths and spanning trees to obtain two novel approxima-  
 385 tion algorithms to compute UMST filter. We provide detailed analyses of our implementations  
 386 along with the TF which is yet another approximation of the UMST filter.

387 **5.1. Exact Algorithms.** Thanks to UMST characterization given by Lemma 4.7, we can  
 388 compute the UMST of a graph in  $\mathcal{O}(|E|)$  which in the case of 4-adjacency graphs would be  
 389  $\mathcal{O}(|V|)$ . Also one can hope to reduce at least significant number of edges when compared to  
 390  $\mathcal{G}$  in practice.

391 A naive approach is to adapt the Floyd-Warshall [23] algorithm to calculate  $\eta(i, j)$  for each  
 392 pair and the computation of UMST filter takes  $\mathcal{O}(|V|^3)$  time and  $\mathcal{O}(|V|^2)$  space. This is very  
 393 expensive in terms of both memory and space and is hence not practical. In order to reduce  
 394 the complexity, one needs to exploit the properties of lexicographic ordering and that of the  
 395 UMST graph structure. One approach is by borrowing ideas from Image Foresting Transform  
 396 (IFT) [21].

397 Image Foresting Transform is a unified framework for several image processing operators  
 398 that are based on shortest paths. Some of these operators include fuzzy-connected segmen-  
 399 tation [12, 32] and distance transforms [28, 20]. In simple words, IFT is a generalization of  
 400 Dijkstra’s algorithm where an image (with a specified adjacency relation), a set of seeds and  
 401 a path cost function are specified and one needs to assign to every non-seeded pixel, a seed  
 402 label to which it admits a path with smallest cost. The path costs are usually application-  
 403 specific and are not necessarily given by sum of the weights of the edges on the path. Hence,  
 404 a modified Dijkstra’s algorithm is used to handle a general class of path cost functions that  
 405 arise in computer vision applications. We briefly describe the IFT framework below as in [21]  
 406 and then discuss the relations with the SPF’s:

407 The IFT takes as an input, an image  $I$ , an adjacency relation  $\mathcal{A}$  (usually given by 4-  
 408 adjacency in case of 2D images), a cost function  $\mathcal{C}$  for all paths and outputs an optimum  
 409 spanning forest. Note that the seeds can be specified implicitly by the cost function by  
 410 assigning a fixed cost for every path that starts at a certain pixel (finite for seed pixels and  
 411 infinite for non-seed pixels). Although there are no restrictions on the dimension of the  
 412 image and the adjacency relation, the path costs are restricted and the following are sufficient  
 413 conditions for the optimal IFT algorithm [21] to be applicable:

414 For any pixel  $t$ , there is an optimum path  $\pi$  ending at  $t$  which is either trivial or is of the  
 415 form  $\tau \cdot \langle s, t \rangle$ , where

- 416 •  $C(\tau) \leq C(\pi)$ ,
- 417 •  $\tau$  is an optimum path ending at  $s$ ,
- 418 •  $C(\tau' \cdot \langle s, t \rangle) = C(\pi)$  if  $\tau'$  is an optimum path ending at  $s$

419 Using a single seed, the optimum-path forest obtained (which is a tree rooted at the seed)

**Algorithm 1** Generic Algorithm to Compute UMST Filter**Input:** A 4-adjacency graph  $\mathcal{G}$  of an image  $I$ , Adaptive Spanning Trees  $T_i$  for each  $i \in V$ **Output:** Filtered image  $S$ .

- 1: **for all** pixels  $i \in V$  **do**
- 2:   Starting from  $i$  on  $T_i$ , use  $S_p = I_p + \sum_{q \in \text{children of } p} \exp(\frac{-1}{\sigma}) S_q$  recursively to compute  $S_i$
- 3: **end for**

420 can be used to compute the shortest path filtering collaborative weights (see (5)). By varying  
 421 the seed  $s$ , the SPF can be computed for the whole image. However, such an implementation  
 422 would still take  $\mathcal{O}(|V|^2)$  time. To the best of our knowledge, we do not have any linear-time  
 423 exact algorithms for computing the UMST filter. Thus, it calls a need to develop at least a  
 424 quasi-linear algorithm approximation algorithm.

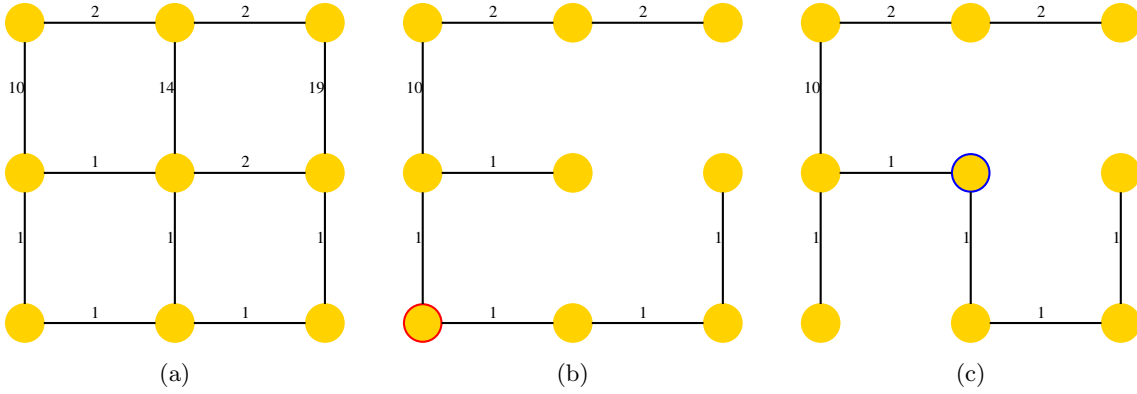
425 **5.2. Approximation Algorithms.** *Single tree-based approximation:* As we have developed  
 426 the UMST filter by generalizing the notion of TF in section 2, we can view TF as a heuristic  
 427 approximation of the UMST filter. Note that TF uses only one MST and hence can be  
 428 computed dynamically in linear time by doing an upward aggregation followed by a downward  
 429 aggregation on the tree (see [39] for details). However, the usage of an arbitrary MST makes  
 430 it difficult to analyse the degree of approximation quantitatively.

431 *Multiple tree-based approximations:* One can also uses multiple spanning trees adaptively  
 432 for filtering different pixels. In fact, one can find upper bounds on the approximation factors  
 433 as a consequence of Proposition 5.1.

434 **Proposition 5.1.** *For every pixel  $i$  in the image  $I$ , there exists a spanning tree  $T_i$  (termed  
 435 as adaptive spanning tree), such that  $T_i$  contains a smallest path with respect to dictionary  
 436 ordering between pixels  $i$  and any other pixel  $j$  in  $I$ .*

437 *Proof.* Let  $i$  be an arbitrary pixel in  $I$ . We shall construct an adaptive spanning tree  $T_i$ ,  
 438 such that  $T_i$  contains a smallest path with respect to dictionary ordering between pixels  $i$  and  
 439 any other pixel  $j$  in  $I$ . The construction is a special case of IFT (see [21]) with the following  
 440 as the inputs:  $I$  is the image with 4-adjacency. The path costs are given by:  $f(\pi) = \infty$  for  
 441 every path  $\pi$  that does not start at  $i$ . All paths  $\pi_l$  that start at  $i$  are completely ordered in  
 442 the order of decreasing lexicographic ordering using their edge-weight distributions. The path  
 443 costs are determined by the order statistics of the path i.e. if  $\pi_1 \geq \pi_2 \geq \dots > \pi_l$  are all the  
 444 paths starting from  $i$  ordered w.r.t. lexicographic ordering then the path cost  $f$  is given by  
 445  $f(\pi_t) = l - t + 1$  where  $1 \leq t \leq l$ . We remark that the this path cost is *monotonic incremental*  
 446 (see [21]). The IFT algorithm applied thus yields  $T_i$ , an adaptive spanning tree of  $i$  with the  
 447 required properties. ■

448 Proposition 5.1 essentially implies that one can decompose the UMST into possibly differ-  
 449 ent spanning trees  $T_i$  for each  $i \in V$ . Using each of the trees independently (see Figure 5 for  
 450 an illustration on a synthetic image), one can obtain the exact UMST filter (see Algorithm 1).  
 451 The exact computation takes  $\mathcal{O}(|V|^2)$  and  $\mathcal{O}(|V|)$  time and space complexities respectively.  
 452 However, by truncating each of these trees in Algorithm 1, one can obtain fast approximate  
 453 solutions. We present two ways to truncate the adaptive trees to obtain approximate UMST



**Figure 5.** (a) A synthetic image with the edge-weights reflecting the intensity dissimilarities, (b) and (c) Adaptive spanning trees of the pixels circled in red and blue respectively.

---

**Algorithm 2** To compute a depth-truncated adaptive spanning tree

---

**Input:** UMST of the graph, depth  $d$  and pixel  $i$

**Output:** Depth-Truncated Adaptive Spanning Tree  $T_{i,d}$

- 1: Set  $X = \{i\}$  and  $T_{i,d} = (i, \emptyset)$
  - 2: **while** True **do**
  - 3:     break = **true**
  - 4:     **for**  $e$  in shortest edges from  $X$  to  $X^c$  **do**
  - 5:         **if**  $\text{dist}(e, i, T_{i,d}) < d$  **then**
  - 6:              $T_{i,d} \cup e$
  - 7:             break = **false**
  - 8:         **end if**
  - 9:     **end for**
  - 10:     **if** break = **true** **then**
  - 11:         return  $T_{i,d}$ .
  - 12:     **end if**
  - 13: **end while**
- 

454 filter.

455

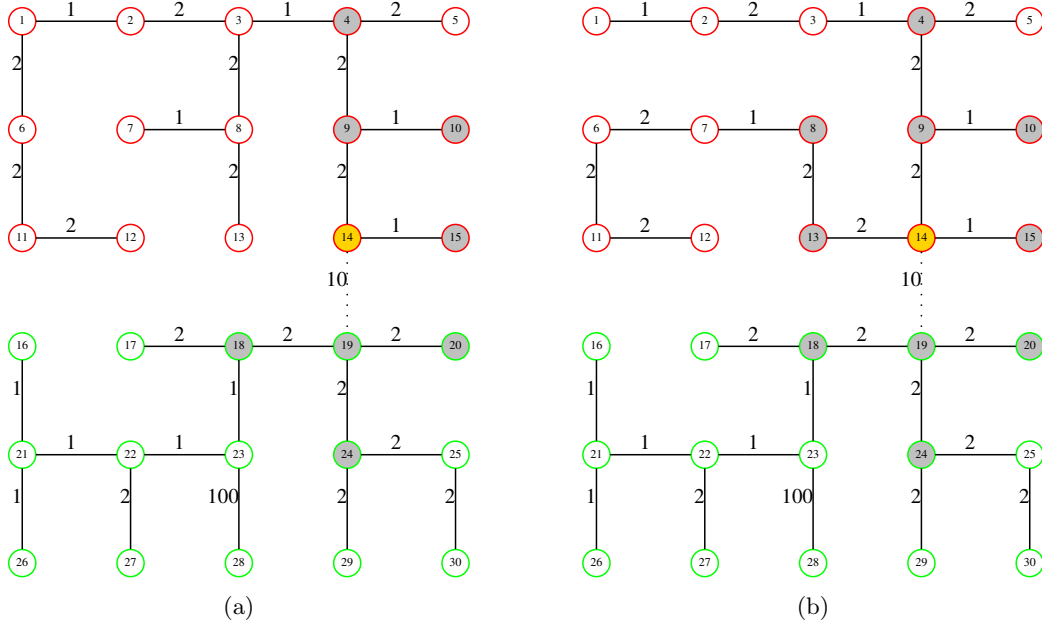
456 *Depth-based truncation:* For each  $i \in V$ , we truncate the adaptive spanning tree  $T_i$  to  $T_{i,d}$   
 457 such that it contains only the pixels  $j$  that are at most  $d$  (user-defined parameter) edges away  
 458 from  $i$  on  $T_i$  (see Figure 6(b) for an illustration and Algorithm 2 for computing it)

459 We rewrite (16) as:

460 (19) 
$$U_i = \frac{1}{C} \sum_l \exp\left(-\frac{l}{\sigma}\right) \sum_{j:\eta(i,j)=l} I_j,$$

461 where  $C$  is the normalizing constant. In (19), we observe that the exponential term rapidly





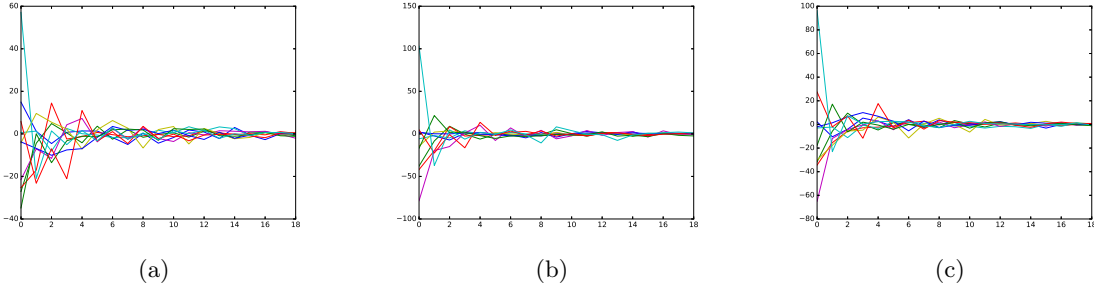
**Figure 6.** (a) TF of pixel numbered 14 in Figure 2, here the pixels with significant collaboration (distance  $\leq 2$ ) are highlighted in grey (b) Depth-based truncation of UMST filter at pixel numbered 14 in Figure 2, here the pixels with significant collaboration (depth  $\leq 2$ ) are highlighted in grey. Observe that the collaboration within the object is higher due to the usage of adaptive spanning tree instead of an arbitrary MST.

462 converges to 0 and hence one can approximate the above expression by

463 (20) 
$$U_i \approx U_{i,d} = \frac{1}{C} \sum_{l=1}^d \exp\left(-\frac{l}{\sigma}\right) \sum_{j:\eta(i,j)=l} I_j,$$

464 where  $d$  is a parameter indicating a fixed depth. This simplification reduces the calculation  
 465 for each pixel drastically and hence Algorithm 2 is practically  $\mathcal{O}(|V|)$ .

466 We shall now analyse (20) in more detail. Also as a consequence of Proposition 5.1, any  
 467 two pixels are separated by at most  $|V| - 1$  edges on a spanning tree  $T_i$  i.e.  $\eta(i, j) \leq |V| - 1$ .  
 468 Also, if  $\eta(i, j) = l > 0$  then for each  $0 \leq l' \leq l$ , there exists at least one pixel  $j'$  such that  
 469  $\eta(i, j') = l'$ . Assume that the intensities satisfy  $1 \leq I_j \leq 255$  then we have the following:



**Figure 7.** From a color image, many pixels have been chosen randomly and each curve in a sub figure represents a pixel. The RGB bands are separately processed and plotted in three sub figures. In each of the sub figures, a curve denotes the first difference of the depth-truncated approximate UMST filtered values as a function of depth. Note that the differences stabilize to 0 at a depth of 15 indicating that (20) yields good approximation to UMST filter.

$$470 \quad (21) \quad \frac{U_i - U_{i,d}}{U_i} = \frac{\sum_{l=d+1}^{|V|-1} \exp(-\frac{l}{\sigma}) \sum_{j:\eta(i,j)=l} I_j}{\sum_{l=0}^{|V|-1} \exp(-\frac{l}{\sigma}) \sum_{j:\eta(i,j)=l} I_j}$$

$$471 \quad (22) \quad \leq \frac{\exp(-\frac{d+1}{\sigma}) \sum_{j:\eta(i,j) \geq d+1} I_j}{I_i + \sum_{l=1}^d \exp(-\frac{l}{\sigma}) \sum_{j:\eta(i,j)=l} I_j}$$

$$472 \quad (23) \quad \leq \frac{\exp(-\frac{d+1}{\sigma}) \sum_{j:\eta(i,j) \geq d+1} I_j}{1 + \sum_{l=1}^d \exp(-\frac{l}{\sigma})}$$

$$473 \quad (24) \quad = \frac{\exp(-\frac{d+1}{\sigma})}{1 + \sum_{l=1}^d \exp(-\frac{l}{\sigma})} \sum_{j:\eta(i,j) \geq d+1} I_j$$

$$474 \quad (25) \quad = \frac{\exp(-\frac{d+1}{\sigma})}{1 - \exp(-\frac{d+1}{\sigma})} (1 - \exp(-\frac{1}{\sigma})) \sum_{j:\eta(i,j) \geq d+1} I_j$$

475 For an image with  $10^6$  pixels, setting  $\sigma = 0.1$ , the expression in (25) is bounded above by  
 476  $\frac{1}{100}$  whenever  $d \geq 220$ . However, the empirical results illustrated by Figure 7 indicate that  
 477 the filtered value of the pixel as a function of depth,  $d$ , stabilizes beyond a depth of 15 for  
 478  $\sigma = 0.1$ .

479 In what follows, we view a rooted spanning tree  $T$  as a directed spanning tree: for every  
 480 pixel  $j$ , a path  $P^*(j)$  recursively as  $\langle j \rangle$  if parent of  $j$  i.e.  $Parent(j) = nil$ , and  $P^*(j) =$   
 481  $P^*(s) \cdot \langle s, j \rangle$  if  $Parent(j) = s \neq null$  (notations are borrowed from [21]).

482 *Order-based truncation:* For each  $i \in V$ , we truncate the adaptive spanning tree  $T_i$  to  $\hat{T}_{i,N}$   
 483 such that it contains only the pixels  $j$  among the closest  $N$  (user-defined parameter) pixels  
 484 w.r.t. the lexicographic ordering from  $i$  on  $T_i$ . (see Figure 8(b) for an illustration Algorithm 3  
 485 for an computing it)

486 The order-based truncation of the adaptive spanning tree would precisely compute the  
 487  $\Gamma$ -limit of the morphological amoeba filters with  $\lambda = 1$  in (9). More formally, we have the

---

**Algorithm 3** To compute an order-truncated adaptive spanning tree

---

**Input:** UMST of the graph with vertex set  $I$  and edge set  $E$ , kernel size  $N$  and pixel  $i$ , path cost function  $f$  as defined in the proof of [Proposition 5.1](#)

**Output:** Order-Truncated Adaptive Spanning Tree  $\hat{T}_{i,N}$

```

1: Set  $\hat{T}_{i,N} = \emptyset$ ,  $\mathcal{Q} = I$ ,  $Parent(j) = null$  for each  $j \in I$  and  $count = 0$ 
2: while  $\mathcal{Q} \neq \emptyset$  and  $count < N$  do
3:   Remove from  $\mathcal{Q}$  a pixel  $j$  such that  $f(P^*(j))$  is minimum and add it to  $\hat{T}_{i,N}$ 
4:    $count+ = 1$ 
5:   for each pixel  $k$  such that  $(j, k) \in E$  do
6:     if  $f(P^*(j) \cdot \langle j, k \rangle) < f(P^*(k))$  then
7:       set  $Parent(k) = j$ 
8:     end if
9:   end for
10: end while
11: Return  $\hat{T}_{i,N}$ 

```

---

488 following result:

489 [Proposition 5.2](#). *As  $p \rightarrow \infty$ , we have*

490 (26) 
$$S_{i,N}^{(p)} \longrightarrow \hat{U}_{i,N}$$

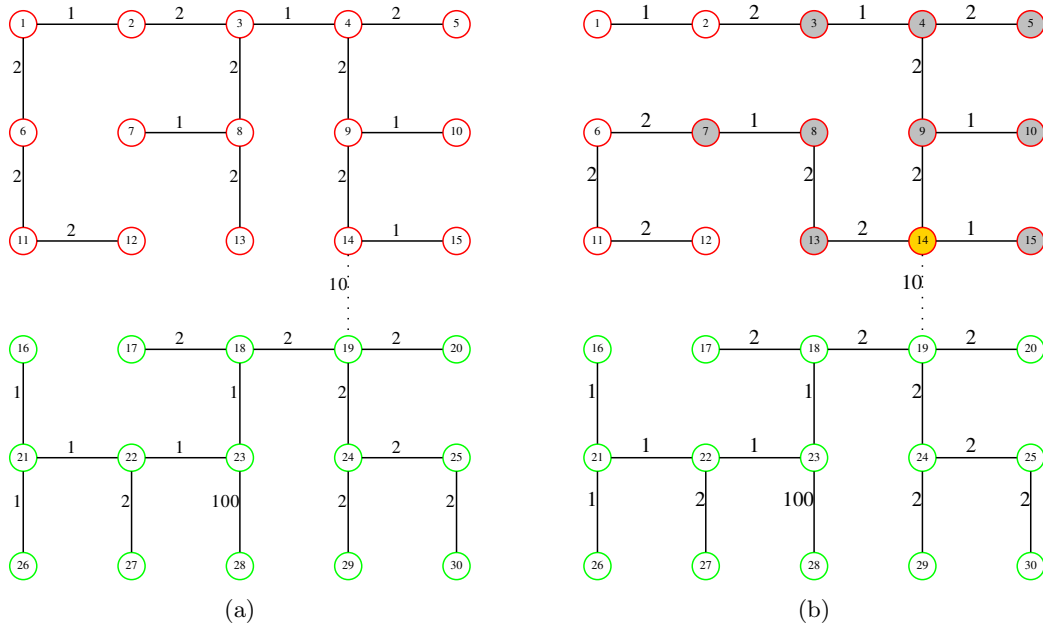
491 where  $S_{i,N}^{(p)}$  is the morphological amoeba filter at pixel  $i$  (with  $\lambda = 1$  and kernel size  $N$ ) and  
492  $\hat{U}_{i,N}$  is the collaborative filter using the closest  $N$  pixels to  $i$  w.r.t. the lexicographic ordering  
493 on the adaptive spanning tree  $T_i$ .

494 *Proof.* The proof follows directly from [Lemma 4.3](#). ■

495 We remark that the choice of the kernel size  $N$  determines the trade-off between the level of  
496 smoothing and the computational cost. In practice, using  $N \approx 100$ , one can obtain a good  
497 edge-aware filter. As the kernel size is fixed and small, [Algorithm 3](#) runs practically in  $\mathcal{O}(|V|)$   
498 time. We shall see the comparison of the performance of our approximations with that of tree  
499 filter in the experiments section.

500 **5.3. Experiments.** In this section, we demonstrate that our approximations of the  $\Gamma$ -  
501 limit perform similar to the TF (in fact marginally better) at an additional computational  
502 cost. To process 1000 pixels, it takes about 0.29 seconds, 37 seconds and 738 seconds for  
503 TF, order-based and depth-based approximations respectively. Consequently, we provide an  
504 empirical evidence that TF is a fast approximation to the  $\Gamma$ -limit of the SPFs. For all our  
505 experiments, we have used identical  $\sigma (= 10)$  parameter for computing the TF, depth-based  
506 (with  $depth = 15$ ) and order-based (with  $N = 100$ ) multi-tree approximations of UMSTF.  
507 The experiments are performed on Intel (R) Xeon(R) CPU W3565 at 3.20GHz with RAM  
508 size of 6 GigaBytes.

509 Firstly, [Figure 9](#) shows a qualitative comparison of BF, TF, depth-based and order-based  
510 multi-tree approximations on some natural images. We observe that BF yields blurry images

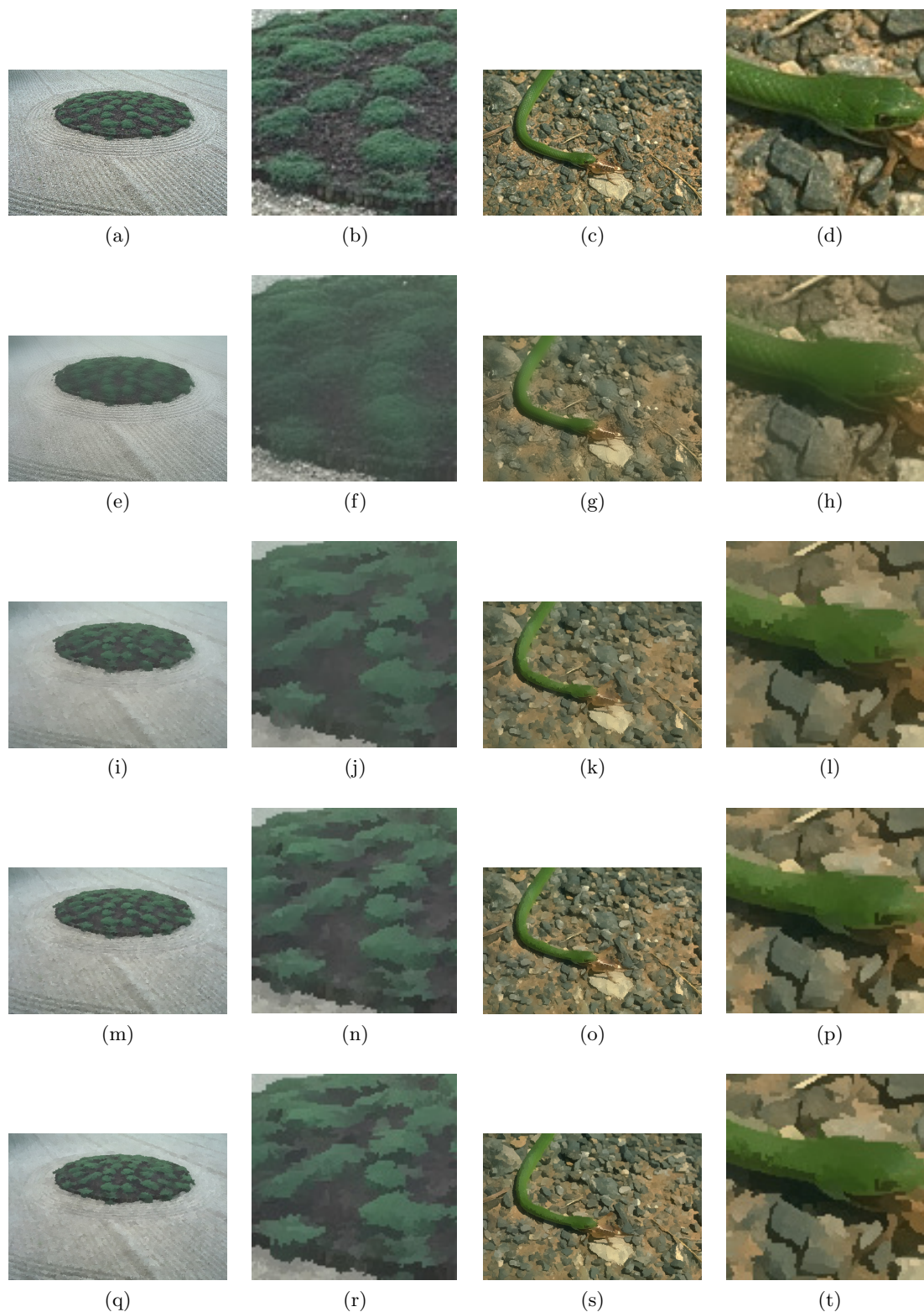


**Figure 8.** (a) TF of pixel numbered 14 in Figure 2 (b) Order-based truncation of PTF at pixel numbered 14 in Figure 2, here the pixels that are closest w.r.t. lexicographic order from 14 (top 10 including itself) are highlighted in grey. Observe that the collaboration in the PTF within the object is very high due to the usage of power spanning tree instead of an arbitrary MST.

511 and erases some object boundaries as expected. On the other hand, TF and our approxima-  
 512 tions yield similar results. However, on a closer look, one can observe that the boundaries are  
 513 marginally better preserved in our approximations (compare green patches in Figure 9 second  
 514 column, snake’s eye in Figure 9 fourth column).

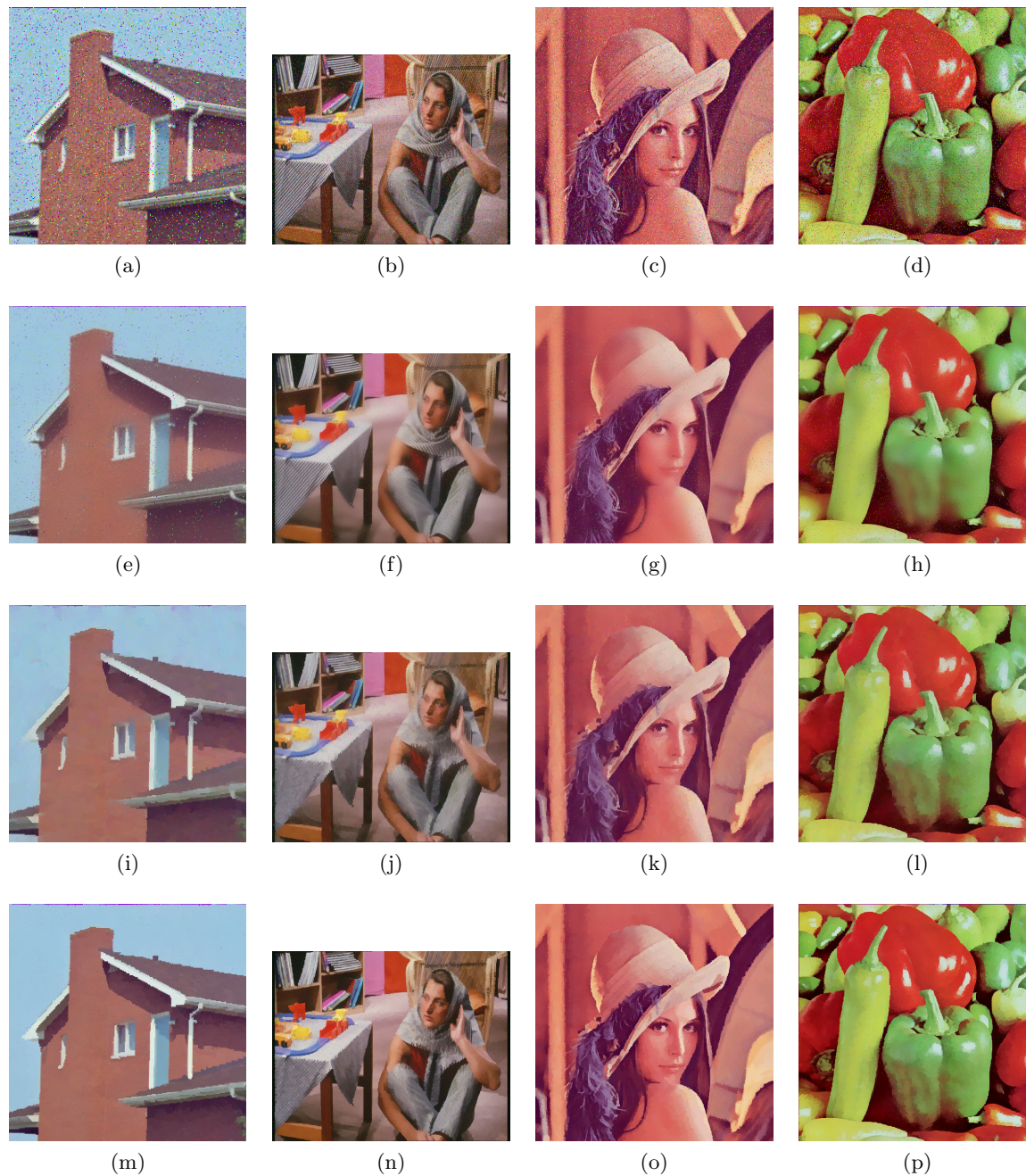
515 For a quantitative comparison of these filters, we have computed the PSNR and structural  
 516 similarity index (SSIM) [9] on images corrupted with synthetic noises. In general, higher PSNR  
 517 values and higher SSIM (SSIM equal to 1 implies identical structures) indicate that the image  
 518 structures are better preserved. However, SSIM is a superior measure when compared to  
 519 PSNR as the latter estimates absolute errors while the former takes structural information  
 520 into consideration. To see this, observe that the mean PSNR values of BF in Table 1 is  
 521 higher than that of other two filters over three iterations of random salt and pepper noise.  
 522 However, a visual comparison of these filters (see Figure 10) suggests that noise is better  
 523 eliminated by TF and our approximation. The mean SSIM values however (see Table 2) are  
 524 in-line with the visual results and indicate that order-based approximation of UMST filter  
 525 performs better than TF in presence of salt and pepper noise. Further, the scatter plot (see  
 526 Figure 11) of the SSIM values of TF and order-based approximation of UMST filter on three  
 527 iterations of random Poisson, salt and pepper, Gaussian and speckle noises on these images  
 528 (House, Barbara, Lena and Pepper) indicate that our approximation is slightly better than  
 529 TF irrespective of the type of noise.

530 Although our approximations yield marginally better results than that of TF, they are



**Figure 9.** A visual comparison of the subtle differences in the performance of TF and our multi-tree approximations of UMSTF on BSDS500 [4] images are illustrated. (a), (b), (c), (d) original images (e), (f), (g), (h) Bilateral Filter ( $\sigma$  color = 100,  $\sigma$  space = 10) (i), (j), (k), (l) Tree Filter ( $\sigma = 10$ ) (m), (n), (o), (p) Depth-based multi-tree truncation ( $\sigma = 10$ , depth = 15) (q), (r), (s), (t) Order-based multi-tree truncation ( $\sigma = 10$ ,  $N = 100$ ). In the second and fourth columns, observe the green patches and the snake's eye respectively. The leaks are more prominent in TF when compared to our approximations. These comparisons are meant for a qualitative comparison only.

*This manuscript is for review purposes only.*



**Figure 10.** Visual illustration of edge-preserving filters on images contaminated with salt and pepper noise. BF erases some object boundaries and does not eliminate the noise completely while TF and order-based approximation eliminate noise and yield similar results. (a), (b), (c), (d) Salt and pepper noisy images (e), (f), (g), (h) Bilateral Filter ( $\sigma_{\text{color}} = 100$ ,  $\sigma_{\text{space}} = 10$ ) (i), (j), (k), (l) Tree Filter ( $\sigma = 10$ ) (m), (n), (o), (p) Order-based multi-tree truncation ( $\sigma = 10$ ,  $N = 100$ )

**Table 1**

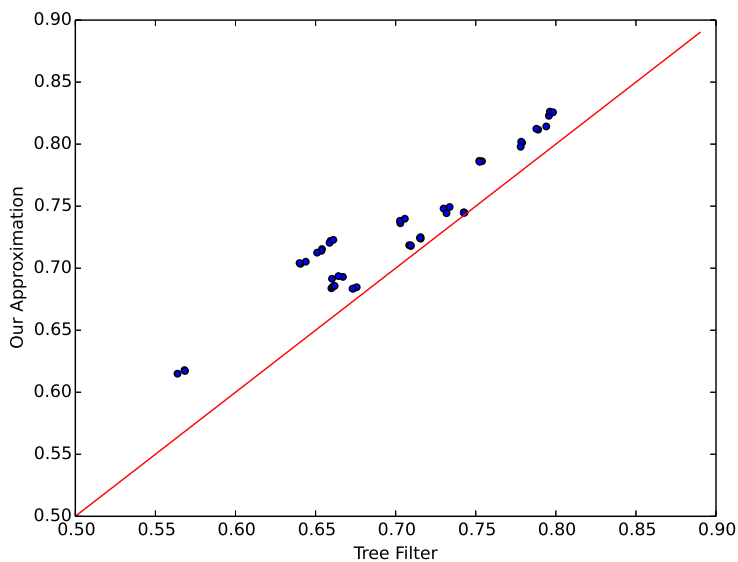
Peak Signal to Noise Ratios (PSNR) measured in db on filtered images corrupted by salt and pepper noise. A higher value indicates a better filter

	Mean PSNR values obtained on Salt Pepper Noise		
	<b>Bilateral Filter</b>	<b>Tree Filter</b>	<b>PTF Order-based</b>
House	24.71	25.04	24.67
Barbara	24.08	22.19	22.47
Lena	22.58	22.59	22.64
Pepper	21.70	21.96	20.59
<b>Mean</b>	<b>23.27</b>	<b>22.95</b>	<b>22.59</b>

**Table 2**

Structural Similarity Indices (SSIM) on filtered images corrupted by salt and pepper noise. A higher value indicates a better filter and a value close to 1 indicates an ideal filter

	Mean SSIM values obtained on Salt Pepper Noise		
	<b>Bilateral Filter</b>	<b>Tree Filter</b>	<b>PTF Order-based</b>
House	0.69	0.80	0.83
Barbara	0.72	0.66	0.72
Lena	0.69	0.75	0.79
Pepper	0.62	0.74	0.74
<b>Mean</b>	<b>0.68</b>	<b>0.74</b>	<b>0.77</b>



**Figure 11.** Scatter plot of SSIM values of TF versus Order-based approximation of UMST filter on three iterations of random Poisson, Salt and Pepper, Gaussian and Speckle noises on House, Barbara, Lena and Pepper images.

531 computationally expensive. However, it is important to note that our approximations can be  
 532 implemented in parallel as each pixel is processed independently of the other. Also we note  
 533 that one has to choose an appropriate filter depending on the type of noise. For instance, BF  
 534 outperforms the tree-based filters in presence of Gaussian noise (see Table 3 and Table 4). To  
 535 summarize, we have demonstrated that TF is a fast approximation of the  $\Gamma$ -limit of SPFs.

**Table 3**

*Peak Signal to Noise Ratios (PSNR) measured in db on filtered images corrupted by Gaussian noise. A higher value indicates a better filter*

	Mean PSNR values obtained on Gaussian Noise		
	<b>Bilateral Filter</b>	<b>Tree Filter</b>	<b>PTF Order-based</b>
House	25.12	23.76	23.97
Barbara	24.14	21.18	21.58
Lena	22.67	21.47	21.78
Pepper	22.15	21.09	20.93
<b>Mean</b>	<b>23.52</b>	<b>21.88</b>	<b>22.07</b>

**Table 4**

*Structural Similarity Indices (SSIM) on filtered images corrupted by Gaussian noise. A higher value indicates a better filter and a value close to 1 indicates an ideal filter*

	Mean SSIM values obtained on Gaussian Noise		
	<b>Bilateral Filter</b>	<b>Tree Filter</b>	<b>PTF Order-based</b>
House	0.79	0.73	0.75
Barbara	0.77	0.57	0.62
Lena	0.76	0.66	0.69
Pepper	0.75	0.67	0.68
<b>Mean</b>	<b>0.77</b>	<b>0.66</b>	<b>0.69</b>

536 **6. Conclusions.** In this paper, we have analysed the edge-aware filters from scratch by de-  
 537 veloping shortest path filters as a natural extension of Gaussian-like filters. We have provided  
 538 a common optimization framework for the well-known filters based on shortest paths and the  
 539 ones based on minimum spanning trees in the power watershed framework. We have thus  
 540 established a theoretical justification of the MST heuristic based tree filter by proving that  
 541 the tree filter is an approximate  $\Gamma$ -limit of shortest path filters. Further, we have proposed  
 542 two different approximation algorithms of the  $\Gamma$ -limit by leveraging ideas from shortest paths  
 543 and minimum spanning trees.

544 Establishing methods based on principles and/or heuristics as limits of solutions of opti-  
 545 mization problems would enable us to design efficient algorithms. We believe that extending  
 546 the ideas from our paper, one can obtain efficient parallel algorithms for practical applica-  
 547 tions. Further, we believe that one could design novel edge-aware image filters based on these  
 548 theoretical foundations. In this line of research, ultimately we aim to show that  $\Gamma$ -convergence  
 549 serves as a powerful tool in applications beyond image segmentation and filtering.



550 **Acknowledgments.** Sravan Danda and Aditya Challa would like to thank Indian Statisti-  
 551 cal Institute for providing fellowship to pursue the research. B.S.Daya Sagar would like to ac-  
 552 knowledge the partial support received from EMR/2015/000853 SERB and ISRO/SSPO/Ch-  
 553 1/2016-17 ISRO research grants. Laurent Najman would like acknowledge the funding re-  
 554 ceived from ANR-15-CE40-0006 CoMeDiC, ANR-14-CE27-0001 GRAPHSIP and Programme  
 555 d’Investissements d’Avenir (LabEx BEZOUT ANR-10-LABX-58) research grants.

556

## REFERENCES

- 557 [1] C. ALVINO, G. UNAL, G. SLABAUGH, B. PENY, AND T. FANG, *Efficient segmentation based on eikonal*  
 558 *and diffusion equations*, International Journal of Computer Mathematics, 84 (2007), pp. 1309–1324.
- 559 [2] L. AMBROSIO AND V. M. TORTORELLI, *Approximation of functionals depending on jumps by elliptic*  
 560 *functionals via  $\Gamma$ -convergence*, Comm. Pure Appl. Math., 43 (1990), pp. 999–1036, [https://doi.org/](https://doi.org/10.1002/cpa.3160430805)  
 561 [10.1002/cpa.3160430805](https://doi.org/10.1002/cpa.3160430805).
- 562 [3] J. ANGULO, *Pseudo-morphological image diffusion using the counter-harmonic paradigm*, in Advanced  
 563 Concepts for Intelligent Vision Systems, Springer, 2010, pp. 426–437.
- 564 [4] P. ARBELAEZ, M. MAIRE, C. FOWLKES, AND J. MALIK, *Contour detection and hierarchical image seg-*  
 565 *mentation*, IEEE transactions on pattern analysis and machine intelligence, 33 (2011), pp. 898–916.
- 566 [5] X. BAI AND G. SAPIRO, *A geodesic framework for fast interactive image and video segmentation and*  
 567 *matting*, in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE, 2007,  
 568 pp. 1–8.
- 569 [6] L. BAO, Y. SONG, Q. YANG, H. YUAN, AND G. WANG, *Tree filtering: Efficient structure-preserving*  
 570 *smoothing with a minimum spanning tree*, IEEE TIP, 23 (2014), pp. 555–569.
- 571 [7] Y. Y. BOYKOV AND M.-P. JOLLY, *Interactive graph cuts for optimal boundary  $\mathcal{E}$  region segmentation of*  
 572 *objects in nd images*, in Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International  
 573 Conference on, vol. 1, IEEE, 2001, pp. 105–112.
- 574 [8] A. BRAIDES, *Gamma-convergence for Beginners*, vol. 22, Clarendon Press, 2002.
- 575 [9] D. BRUNET, E. R. VRSCAY, AND Z. WANG, *On the mathematical properties of the structural similarity*  
 576 *index*, IEEE Transactions on Image Processing, 21 (2012), pp. 1488–1499.
- 577 [10] A. CHALLA, S. DANDA, B. S. DAYA SAGAR, AND L. NAJMAN, *An Introduction to Gamma-Convergence*  
 578 *for Spectral Clustering*, in Discrete Geometry for Computer Imagery, vol. 10502 of Lecture Note In  
 579 Computer Sciences, Vienna, Austria, Sept. 2017, Kropatsch, Walter G. and Artner, Nicole M. and  
 580 Janusch, Ines, Springer, pp. 185–196, <https://hal.archives-ouvertes.fr/hal-01427957>.
- 581 [11] J.-H. R. CHANG AND Y.-C. F. WANG, *Propagated image filtering*, in 2015 IEEE Conference on Computer  
 582 Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 10–18.
- 583 [12] K. C. CIESIELSKI, J. K. UDUPA, A. X. FALCÃO, AND P. A. MIRANDA, *Fuzzy connectedness image*  
 584 *segmentation in graph cut formulation: A linear-time algorithm and a comparative analysis*, Journal  
 585 of Mathematical Imaging and Vision, 44 (2012), pp. 375–398.
- 586 [13] C. COUPRIE, L. GRADY, L. NAJMAN, AND H. TALBOT, *Power watershed: A unifying graph-based opti-*  
 587 *mization framework*, IEEE PAMI, 33 (2011), pp. 1384–1399.
- 588 [14] M. COUPRIE, G. BERTRAND, ET AL., *Topological grayscale watershed transformation*, in SPIE vision  
 589 geometry V proceedings, vol. 3168, 1997, pp. 136–146.
- 590 [15] J. COUSTY, G. BERTRAND, L. NAJMAN, AND M. COUPRIE, *Watershed cuts: Minimum spanning forests*  
 591 *and the drop of water principle*, IEEE PAMI, 31 (2009), pp. 1362–1374.
- 592 [16] J. COUSTY, G. BERTRAND, L. NAJMAN, AND M. COUPRIE, *Watershed cuts: Thinnings, shortest path*  
 593 *forests, and topological watersheds*, IEEE PAMI, 32 (2010), pp. 925–939.
- 594 [17] A. CRIMINISI, T. SHARP, AND A. BLAKE, *Geos: Geodesic image segmentation*, Computer Vision–ECCV  
 595 2008, (2008), pp. 99–112.
- 596 [18] S. DANDA, A. CHALLA, B. D. SAGAR, AND L. NAJMAN, *Power tree filter: A theoretical framework linking*  
 597 *shortest path filters and minimum spanning tree filters*, in International Symposium on Mathematical  
 598 Morphology and Its Applications to Signal and Image Processing, Springer, 2017, pp. 199–210.
- 599 [19] Y. C. DE VERDIERE, Y. PAN, AND B. YCART, *Singular limits of schrodinger operators and markov*

- 600 *processes*, Journal of Operator Theory, 41 (1999), pp. 151–173.
- 601 [20] A. X. FALCÃO, L. DA FONTOURA COSTA, AND B. DA CUNHA, *Multiscale skeletons by image foresting*  
602 *transform and its application to neuromorphometry*, Pattern recognition, 35 (2002), pp. 1571–1582.
- 603 [21] A. X. FALCAO, J. STOLFI, AND R. DE ALENCAR LOTUFO, *The image foresting transform: Theory,*  
604 *algorithms, and applications*, IEEE PAMI, 26 (2004), p. 19.
- 605 [22] Z. FARBMAN, R. FATTAL, D. LISCHINSKI, AND R. SZELISKI, *Edge-preserving decompositions for multi-*  
606 *scale tone and detail manipulation*, in ACM Transactions on Graphics (TOG), vol. 27, ACM, 2008,  
607 p. 67.
- 608 [23] R. W. FLOYD, *Algorithm 97: shortest path*, Communications of the ACM, 5 (1962), p. 345.
- 609 [24] L. GRADY, *Random walks for image segmentation*, IEEE PAMI, 28 (2006), pp. 1768–1783.
- 610 [25] J. GRAZZINI AND P. SOILLE, *Edge-preserving smoothing using a similarity measure in adaptive geodesic*  
611 *neighbourhoods*, Pattern Recognition, 42 (2009), pp. 2306–2316.
- 612 [26] K. HE, J. SUN, AND X. TANG, *Guided image filtering*, in Computer Vision–ECCV 2010, Springer, 2010,  
613 pp. 1–14.
- 614 [27] R. LERALLUT, É. DECENCIÈRE, AND F. MEYER, *Image filtering using morphological amoebas*, Image and  
615 Vision Computing, 25 (2007), pp. 395–404.
- 616 [28] R. D. A. LOTUFO, A. A. FALCÃO, AND F. A. ZAMPIROLI, *Fast euclidean distance transform using a*  
617 *graph-search algorithm*, in Computer Graphics and Image Processing, 2000. Proceedings XIII Brazilian  
618 Symposium on, IEEE, 2000, pp. 269–275.
- 619 [29] L. NAJMAN, *Extending the PowerWatershed framework thanks to  $\Gamma$ -convergence*, SIAM Journal on  
620 Imaging Sciences, 10 (2017), pp. 2275–2292, <https://doi.org/10.1137/17M1118580>, [https://hal.](https://hal.archives-ouvertes.fr/hal-01428875)  
621 [archives-ouvertes.fr/hal-01428875](https://hal.archives-ouvertes.fr/hal-01428875).
- 622 [30] L. NAJMAN, M. COUPRIE, AND G. BERTRAND, *Watersheds, mosaics, and the emergence paradigm*, Dis-  
623 crete Applied Mathematics, 147 (2005), pp. 301–324.
- 624 [31] L. NAJMAN, J.-C. PESQUET, AND H. TALBOT, *When convex analysis meets mathematical morphology on*  
625 *graphs*, in International Symposium on Mathematical Morphology and Its Applications to Signal and  
626 Image Processing, Springer, 2015, pp. 473–484.
- 627 [32] P. K. SAHA AND J. K. UDUPA, *Relative fuzzy connectedness among multiple objects: theory, algorithms,*  
628 *and applications in image segmentation*, Computer Vision and Image Understanding, 82 (2001),  
629 pp. 42–56.
- 630 [33] J. STAWIASKI AND F. MEYER, *Minimum spanning tree adaptive image filtering*, in 2009 16th IEEE ICIP,  
631 IEEE, 2009, pp. 2245–2248.
- 632 [34] R. SZELISKI, *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.
- 633 [35] C. TOMASI AND R. MANDUCHI, *Bilateral filtering for gray and color images*, in Sixth International  
634 Conference on Computer Vision, 1998. ICCV 1998, IEEE, 1998, pp. 839–846.
- 635 [36] L. J. VAN VLIET, *Robust local max-min filters by normalized power-weighted filtering*, in Pattern Recog-  
636 nition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 1, IEEE, 2004,  
637 pp. 696–699.
- 638 [37] L. XU, C. LU, Y. XU, AND J. JIA, *Image smoothing via  $L_0$  gradient minimization*, in ACM Transactions  
639 on Graphics (TOG), vol. 30, ACM, 2011, p. 174.
- 640 [38] L. XU, Q. YAN, Y. XIA, AND J. JIA, *Structure extraction from texture via relative total variation*, ACM  
641 Transactions on Graphics (TOG), 31 (2012), p. 139.
- 642 [39] Q. YANG, *Stereo matching using tree filtering*, IEEE PAMI, 37 (2015), pp. 834–846.