



**HAL**  
open science

## Exploring the Links Between Edge-Preserving Collaborative Filters via Gamma Convergence

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 Gamma Convergence. 2017. hal-01617799v1

**HAL Id: hal-01617799**

**<https://hal.science/hal-01617799v1>**

Preprint submitted on 17 Oct 2017 (v1), 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 Gamma Convergence\*

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. 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 the notion of  $\Gamma$ -convergence. 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. Further, we propose and provide a detailed analysis of two different implementations of the UMST filter based on shortest paths.

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

**AMS subject classifications.** ?, ?, ?

**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 [32], texture removal [32], texture editing [32], scene simplification [32], stereo matching [32], optical flow [32] 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) [33], guided filter (GF) [24], weighted least squares filter (WLS) [20],  $L_0$  smoothing [36], propagated image filter [10], morphological amoebas or adaptive kernel based filters [25], tree filter (TF) [5] and relative total variation filter (RTV) [37] etc. Although these filters work well in practice, some of them are not extensively studied from a theoretical perspective. In this article, we study the recent state-of-art edge-aware Tree Filter (TF) which is based on a Minimum Spanning Tree (MST) heuristic. TF admits a linear time algorithm [38] and yields promising results in applications such as texture removal, stereo matching and scene simplification. However it exhibits a leak problem at some of the object boundaries. This

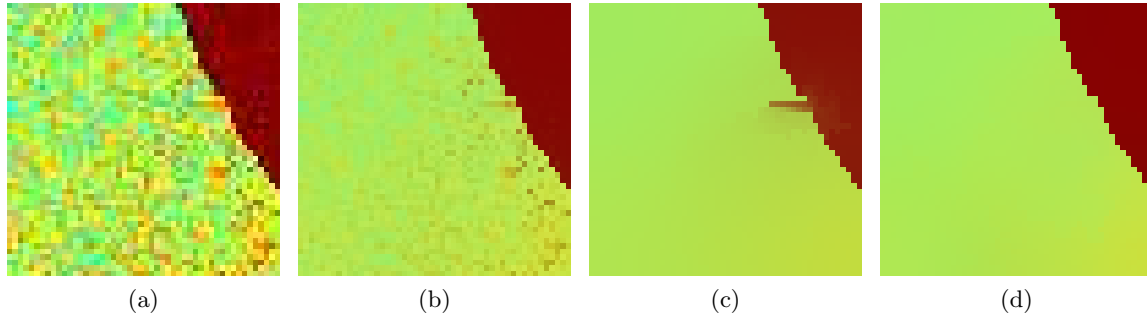
---

\*Submitted to the editors DATE.

**Funding:** This work was funded by .

<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.

36 problem occurs due to the presence of some object boundary edges in the MST and cannot  
 37 be avoided as any spanning tree connects all the nodes in a connected graph. Although, the  
 38 authors in [5] tried to negate the leak effect using a bilateral filter as a post-processing step,  
 39 the problem still persists (see Figure 1(c)). Also, the filtering results vary with the choice of  
 40 MST which is undesirable.

41 This motivated us to explore the theoretical foundations of the TF for a deeper under-  
 42 standing on how it works. Further, the links between the TF and the other edge-aware filtering  
 43 methods might provide a possibility to design novel edge-aware filters. This article aims to  
 44 answer this question and is an extended version of the conference paper [17], our contributions  
 45 are the following:

- 46 1. We introduce an edge-aware filter based on the union of all MSTs of the image graph  
 47 namely UMST filter, a generalization of the TF .
- 48 2. We compute the  $\Gamma$ -limit of the SPFs i.e. the Power Tree Filter (PTF) and show that  
 49 it is precisely given by the UMST filter (see section 4 for details). Consequently, we  
 50 provide a theoretical basis for the TF as it can be seen as an approximation of the  
 51  $\Gamma$ -limit of the SPFs.
- 52 3. We propose two different implementations of the  $\Gamma$ -limit which serve as an alternative  
 53 to the TF (see section 5 for details) with a detailed analysis on how they work.

54 The rest of the paper is organized as follows: In section 2, we briefly recall the notions  
 55 of TF,  $\Gamma$ -convergence and introduce UMST filter. In section 3, we develop SPFs as edge-  
 56 aware filters starting from Gaussian-like filters and discuss their properties, links with other  
 57 geodesic based methods. In section 4, we compute the  $\Gamma$ -limit of the SPFs and show that it  
 58 is precisely the UMST filter. In section 5, we discuss approximations of UMST filter i.e. TF  
 59 and propose two approximations based on shortest paths. We provide a detailed analysis of  
 60 each of these implementations. In section 6, the conclusions follow and we speculate some  
 61 possible directions to extend the ideas in the paper.

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

65 **2.1. Tree Filter.** Suppose  $I$  is a given image which possibly contains noise, we let  $I_i$   
 66 denote the color or intensity of the pixel  $i$  in the image  $I$ . let  $S$  denote the tree filtered image.  
 67 The authors in [5] construct a 4-adjacency edge-weighted graph, with the weights between  
 68 adjacent pixels reflecting the color or intensity dissimilarity. More formally, if  $i$  and  $j$  are  
 69 4-adjacent pixels, they use  $w_{ij}$  defined by

$$70 \quad (1) \quad w_{ij} = ||I_i - I_j||$$

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

$$75 \quad (2) \quad t_i(j) = \frac{\exp\left(\frac{-D(i,j)}{\sigma}\right)}{\sum_q \exp\left(\frac{-D(i,q)}{\sigma}\right)}$$

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

$$78 \quad (3) \quad S_i = \sum_j t_i(j) I_j$$

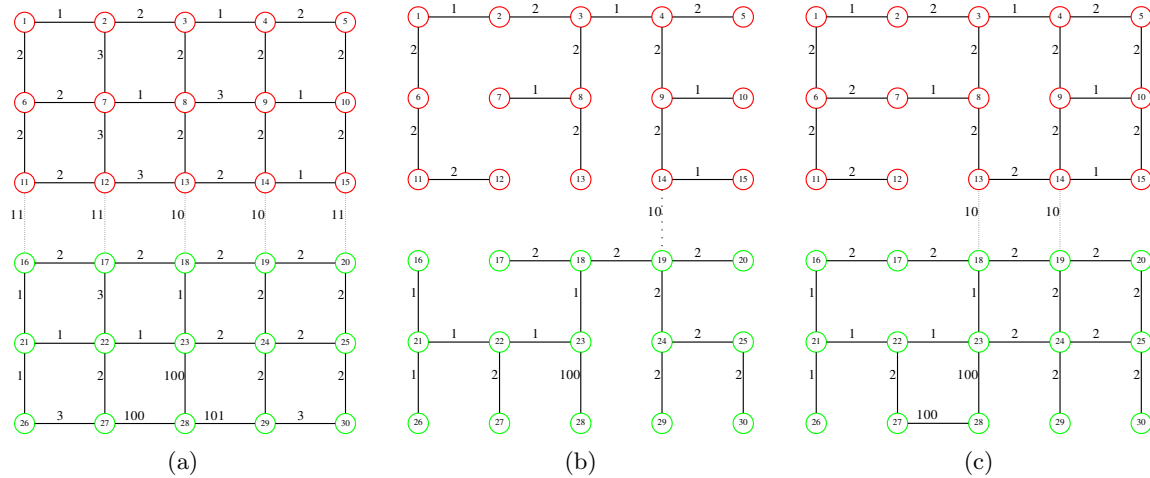
79 Here the summation over  $j$  is over all the pixels in the graph.

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

87 **2.2. Why UMST filter?.** The edges in the image graph that do not belong to object  
 88 boundaries induce a disconnected subgraph. On the other hand, a MST of a graph is con-  
 89 nected, hence any MST contains one or more object boundary edges. These edges cause a  
 90 leak effect in the tree filtered image (see [Figure 1\(c\)](#)). Also, the filtering results vary with the  
 91 MST used, making the choice of an arbitrary MST debatable. On the other hand, a filter  
 92 based on UMST would ensure the following:

- 93 1. The filtering results would not depend on arbitrary MST computations.
- 94 2. There would be a significant reduction in the leak effect when compared to the TF  
 95 (see [Figure 1\(d\)](#)).

96 The first property is a direct consequence of the fact that UMST filter uses all the MSTs  
 97 of the image graph. The second property can be explained intuitively as follows: The edges  
 98 in the UMST is a superset of the edges of an arbitrary MST. Now, among the edges that  
 99 belong to UMST but not the MST are mostly object interior edges. These object interior



**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 the presence of the additional interior edge 13 – 14.

100 edges dominate the collaborative effect of the object boundary edges to ensure a reduction in  
 101 the leakage. **Figure 2** illustrates the above properties on a synthetic image.

102 Extending the idea of TF, we use an exponential falling weight similar to (2) for computing  
 103 collaborative weights. However, we observe that there are possibly multiple paths between  
 104 a given pair of pixels  $i$  and  $j$  in the UMST. In order to define the collaborative weights of  
 105 the UMST filter, we need a criterion to choose a path among all the paths between  $i$  and  $j$ .  
 106 We consider  $\eta(i, j)$ , the number of edges on a path with smallest dictionary or lexicographic  
 107 order of edge weights (see **Definition 3.4** and **Definition 4.1**) replacing  $D(i, j)$  in (2). This is  
 108 a natural way to generalize the TF since: (a) the lesser the lexicographic order of a path, the  
 109 lesser the chance of the path crossing an object boundary, (b) in the special case of the graph  
 110 having a unique MST, this filter is exactly the same as TF.

111 **2.3. Lexicographic Ordering and Pass Values.** Lexicographic order of a path is related  
 112 very closely to the notion of pass values used in segmentation. Pass value [15] or the minimax  
 113 distance [19] between a pair of nodes is the minimum of the  $l^\infty$  norm over all the paths  
 114 between them. To the best of our knowledge, this feature was first used in image filtering  
 115 in [31]. Pass values between different minima of a gradient image is a measure of contrast  
 116 difference between objects in watershed segmentation [28]. One can observe that any image

117 transformation on the gradient image that keeps the object boundaries intact preserves the  
 118 contrast difference between objects. It is hence a desired condition for a segmentation method  
 119 to preserve the contrast difference (topological watersheds [13] for instance).

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

126 Now, we shall recall notions of  $\Gamma$ -convergence before we state the main result of the paper.

127 **2.4. UMST filter and  $\Gamma$ -convergence.**  $\Gamma$ -convergence [8] is the study of asymptotic be-  
 128 haviour of a sequence of minimization problems. Suppose for each  $n \in \mathbb{N}$ ,  $F_n : \mathbb{R}^l \rightarrow \mathbb{R}$  is a  
 129 cost function such that  $\arg \min_{x \in \mathbb{R}^l} (F_n(x)) \neq \emptyset$ , what does  $\lim_{n \rightarrow \infty} x_n$  minimize (assuming the  
 130 limit exists) where  $x_n \in \arg \min_{x \in \mathbb{R}^l} (F_n(x))$ ? In other words, in what sense do  $F_n$  converge  
 131 to  $F$  where  $F : \mathbb{R}^l \rightarrow \mathbb{R}$  so that  $\lim_{n \rightarrow \infty} x_n \in \arg \min_{x \in \mathbb{R}^l} (F(x))$ .

132 The usual notions of point wise and uniform convergence do not make sense when working  
 133 with functionals and we need the following:

134 **Definition 2.1.** We say that  $F_n \xrightarrow{\Gamma} F$  (read as  $F_n$  gamma converges to  $F$ ) if: (1) for every  
 135  $x \in \mathbb{R}^l$  and every sequence  $(x_n)_{n \in \mathbb{N}}$ , such that  $x_n \rightarrow x$ , we have  $F(x) \leq \liminf_{n \rightarrow \infty} F_n(x_n)$ ,  
 136 and (2) for every  $x \in \mathbb{R}^l$  there exists a sequence  $(x_n)_{n \in \mathbb{N}}$ , such that  $x_n \rightarrow x$ , and  $F(x) \geq$   
 137  $\limsup_{n \rightarrow \infty} F_n(x_n)$ .

138 **Theorem 2.2.** (Fundamental Theorem of  $\Gamma$ -Convergence) If  $F_n \xrightarrow{\Gamma} F$  and  $x_n$  minimizes  $F_n$   
 139 for each  $n \in \mathbb{N}$ , then every limit point of the sequence  $(x_n)_{n \in \mathbb{N}}$  is a minimizer of  $F$ .

140 *Proof.* Refer to [8] ■

141 In simple words, one can approximate a minimizer of  $F$  using the minimizers of  $F_n$ . **Defini-**  
 142 **tion 2.1** is a simplified version of the general definition and suffices for our purposes. For a  
 143 comprehensive study of  $\Gamma$ -convergence, we refer the interested reader to [8].

144  $\Gamma$ -convergence has been proved to be very useful in many computer vision applications  
 145 especially the ones based on variational formulations. The following are a few instances:  
 146 In [12], the authors unified and extended a common framework of semi-supervised or seeded  
 147 graph-based image segmentation methods namely graph cuts [7], random walker [22], geodesics  
 148 [1, 4, 16, 19] and watershed cuts [14, 15]; In [29, 2], the elementary mathematical morphological  
 149 (MM) operators have been formulated as limits of variational problems; and in [34], the authors  
 150 view the local min-max filters as a limit of normalized power-weighted averaging filter.

151 As edge-preserving image filtering and image segmentation are closely related problems,  
 152 one can anticipate to establish links between existing filtering methods using  $\Gamma$ -convergence  
 153 (in the similar lines as the unified seeded-segmentation framework in [12]). In fact we prove  
 154 the following theorem which is the main result of the paper: *UMST filter is the  $\Gamma$ -limit of*  
 155 *shortest path edge-aware filters*. See [section 3](#) for a formal definition of shortest path filters  
 156 and [section 4](#) for a proof. This result implies that one can view the UMST filter and the  
 157 shortest path filters in an optimization framework.

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

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

### 163 3.1. Basic Notions.

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

168 **Definition 3.2.** For  $p \in \mathbb{Z}^+$ , we denote by  $\mathcal{G}^{(p)} = (V, E, W^{(p)})$ , the graph that contains the  
 169 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  
 170 call  $\mathcal{G}^{(p)}$  as an **exponentiated graph** of  $\mathcal{G}$ .

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

174 **Definition 3.4.** Assume that the distinct weights in  $\mathcal{G}$  are given by  $0 < w_1 < w_2 < \dots < w_k$ .  
 175 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  
 176 the number of edges of weight  $w_r$  on the path  $P(i, j)$ . This  $k$ -tuple is referred to as the **edge-**  
 177 **weight distribution** of the path  $P(i, j)$ .

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

182 **Definition 3.5.** Assume that the distinct weights in  $\mathcal{G}$  are given by  $0 < w_1 < w_2 < \dots <$   
 183  $w_k$ . Suppose  $P(i, j)$  is a path between pixels  $i$  and  $j$  in  $\mathcal{G}$ .  $P(i, j)$  is said to be a **shortest**  
 184 **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  
 185 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  
 186 distributions of  $P(i, j)$  and  $Q(i, j)$  respectively.

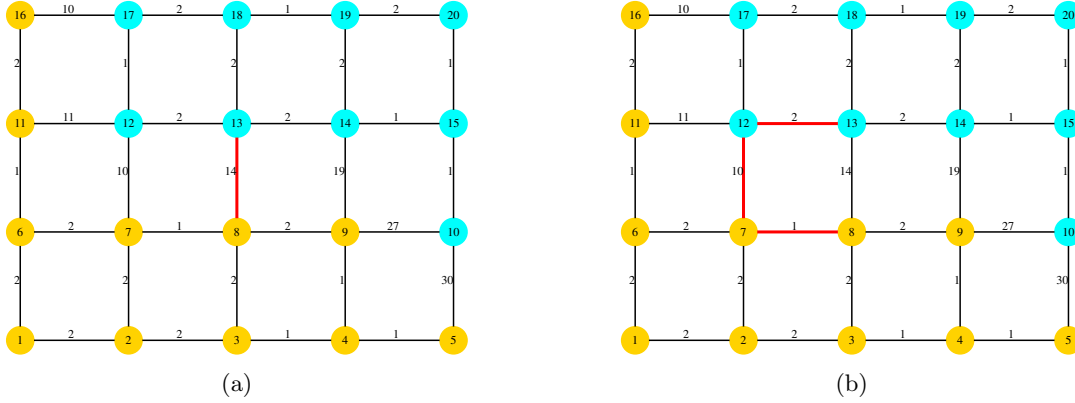
187 We remark that a shortest path is always a simple path since all the weights in the edge-  
 188 weighted graphs are strictly positive.

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

$$191 \quad (4) \quad g_i(j) = \frac{\exp\left(-\frac{\|i-j\|}{\sigma}\right)}{\sum_k \exp\left(-\frac{\|i-k\|}{\sigma}\right)}$$

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





**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.

194 We observe that this is a Gaussian-like filter and collaborative weights purely depend on  
 195 the spatial distance between pixels  $i$  and  $j$ . More specifically, collaborative weights between  
 196 pixels separated by same distance is indifferent w.r.t. existence of an object boundary between  
 197 them. Hence, one has to find a way to ensure that the collaborative weights are lesser across  
 198 boundaries. A natural way to extend the idea of a Gaussian-like filter is: given a pair of pixels  
 199  $i$  and  $j$ , consider the number of edges on a path with smallest sum of weights between them.  
 200 Let  $\Pi(P(i, j))$  denote the number of edges on a path  $P(i, j)$ . Define

$$201 \quad (5) \quad \Theta^{(p)}(i, j) = \inf\{\Pi(P(i, j)) \text{ where } P(i, j) \text{ is a shortest path in } \mathcal{G}^{(p)}\}$$

202 where the edge weights are given by:

$$203 \quad (6) \quad w_{ij} = ||I_i - I_j|| + 1$$

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

$$205 \quad (7) \quad S_i^{(p)} = \sum_j \frac{\exp\left(-\frac{\Theta^{(p)}(i, j)}{\sigma}\right)}{\sum_k \exp\left(-\frac{\Theta^{(p)}(i, k)}{\sigma}\right)} I_j,$$

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

207 Note that the weights in Eq. (6) are different from the ones in Eq. (1) to ensure that  
 208 the weights are strictly positive. We use the edge weights as per (6) in the rest of the paper.  
 209 Since, we use an increasing transformation on edge weights, shortest paths and MSTs are



invariant to the modification. This assumption is needed to ensure that the converse part in Lemma 4.3 holds.

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

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

Consider the cost function

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

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

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

SPFs are not completely new and there exist in literature, several edge-preserving filters using geodesics such as the ones discussed in [23], the adaptive kernel filter such as morphological amoebas [25].

**3.2. Relation to Morphological Amoebas.** SPFs are also closely related to Morphological Amoebas. Morphological Amoebas are adaptive structuring elements based on shortest path distances used to build edge-aware filters. These kernels work on the assumption that the gradients are low within the objects and high across the object boundaries. In order to ensure that the kernels do not cross the object boundaries, the amoeba distance defined below is used to compute them:

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

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 user input.

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

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



**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

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

$$243 \quad (11) \quad \Theta^{(1)}(i, j) = \min L_0(P(i, j)) \text{ subject to } P(i, j) \in \arg \min L_1(P(i, j))$$

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

$$245 \quad (12) \quad \Theta^{(p)}(i, j) = \min L_0^p(P(i, j)) \text{ subject to } P(i, j) \in \arg \min L_1^p(P(i, j))$$

246 where

$$247 \quad (13) \quad L_\lambda^p(P(i, j)) = \sum_{t=0}^{n-1} (1 + \lambda \|I_{x_{t+1}} - I_{x_t}\|)^p$$

248

249 *Proof.* The proofs readily follow from (5), (6), (10) and (13) ■

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

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

#### 257 4.1. Some Definitions.

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

$$262 \quad (14) \quad P \geq Q \Leftrightarrow A = \emptyset \text{ or } n_l > m_l$$

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

266 **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  
 267 a **smallest path w.r.t. dictionary order** between the pixels  $i$  and  $j$  in  $\mathcal{G}$  if for every path  
 268  $Q(i, j)$  between  $i$  and  $j$  in  $\mathcal{G}$ , we have  $Q(i, j) \geq P(i, j)$ .

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

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

276 **4.2. Gamma Limit of Shortest Path Filters.** In this subsection, we characterize the  
 277 Power Tree Filter or  $\Gamma$ -limit of Shortest Path Filters. Firstly, we have the following result:

278 **Lemma 4.3.** Let  $\mathcal{G} = (V, E, W)$ . For every pair of pixels  $i$  and  $j$  in  $V$ , there exists  $p_0 \geq 1$   
 279 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  
 280 if  $P(i, j)$  is a smallest path w.r.t. dictionary order between  $i$  and  $j$  in  $\mathcal{G}$ . Further,  $p_0$  is  
 281 independent of  $i$  and  $j$ .

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

283 Firstly, we shall show that for a given pair of pixels  $i$  and  $j$ , if  $P(i, j)$  is a smallest path  
 284 w.r.t. dictionary order between  $i$  and  $j$  in  $\mathcal{G}$  then there exists a constant  $p_0$  such that for  
 285 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-  
 286 est path w.r.t. dictionary order between  $i$  and  $j$  in  $\mathcal{G}$ . Let  $Q(i, j)$  be an arbitrary simple  
 287 path between  $i$  and  $j$ . Let  $(n_1, \dots, n_k)$  and  $(m_1, \dots, m_k)$  denote the edge-weight distribu-  
 288 tions of paths  $P(i, j)$  and  $Q(i, j)$  respectively. Let  $A(P, Q) = \{1 \leq r \leq k : n_r \neq m_r\}$ .  
 289 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  
 290  $l = \sup(A(P, Q))$ . We have  $m_l > n_l$  by choice of  $P(i, j)$ . Also, the difference of the total  
 291 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  
 292  $\exists p_{Q(i,j)} \geq 1$  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  
 293 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  
 294 note that given any path which is not simple, one can drop the redundant edges to construct a  
 295 simple path with strictly smaller total weight. It is hence enough to show that the total weight  
 296 of  $P(i, j)$  is lesser than or equal to every simple path between  $i$  and  $j$  in  $\mathcal{G}^{(p)}$  for sufficiently  
 297 large  $p$ . As  $V$  is finite, setting  $p_0 = \sup\{p_{ij} : i, j \in V\}$  completes the argument.

298 Conversely, suppose  $P(i, j)$  is NOT a smallest path w.r.t. dictionary ordering between  
 299  $i$  and  $j$ , we shall construct a sequence  $(p_n)_{n \geq 1}$  converging to  $\infty$  such that  $P(i, j)$  is not  
 300 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  
 301 path w.r.t. dictionary ordering between  $i$  and  $j$ ,  $\exists$  a path  $T(i, j)$  between  $i$  and  $j$  such that  
 302  $P(i, j) \geq T(i, j)$  holds but  $T(i, j) \geq P(i, j)$  does not hold. Equivalently, if the edge weight  
 303 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  
 304  $A(P, T) \neq \emptyset$  and  $t_l < n_l$  where  $l = \sup(A(P, T))$  and  $A(P, T) = \{1 \leq r \leq k : n_r \neq t_r\}$ .

305 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  
 306 leading coefficient. Thus,  $\exists$  a constant  $\rho_{ij} \geq 1$  such that for each  $p \geq \rho_{ij}$ ,  $P(i, j)$  is not a  
 307 shortest path between  $i$  and  $j$ . We set  $(p_n)_{n \geq 1}$  by  $p_n = \rho_{ij} + n - 1$  to complete the proof. ■

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

311 **Corollary 4.4.** *Let  $\Delta(i, j)$  denote the number of edges on a smallest path w.r.t. dictionary*  
 312 *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*  
 313  *$j$ .*

314 **4.3. Optimization Framework for UMST Filter.** Recall the definition of UMST filter  
 315 from section 2. Let  $\mathcal{G}$  denote the given image, let  $\mathcal{G}_{UMST}$  denote the UMST on the edge-  
 316 weighted graph constructed from  $\mathcal{G}$ . Let  $\eta(i, j)$  denote the number of edges on a smallest path  
 317 w.r.t. dictionary order between pixels  $i$  and  $j$  in  $\mathcal{G}_{UMST}$ . Now consider the cost function given  
 318 by:

$$319 \quad (15) \quad \widehat{Q}_i(x) = \sum_j \exp\left(-\frac{\eta(i, j)}{\sigma}\right) (x - I_j)^2$$

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

$$322 \quad (16) \quad U_i = \arg \min \widehat{Q}_i(x) = \sum_j \frac{\exp\left(-\frac{\eta(i, j)}{\sigma}\right)}{\sum_k \exp\left(-\frac{\eta(i, k)}{\sigma}\right)} I_j,$$

323 where  $U$  denotes the UMST filtered image.

324 Firstly, we need the following standard results:

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

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

331 Before we prove the main result of the paper, we need Lemma 4.7 which is a modified  
 332 version of a result stated in [27] (need to update this reference).

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

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

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

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

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

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

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

356 The main result is formally stated as follows:

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

$$358 \quad (17) \quad Q_i^{(p)}(x) \xrightarrow{\Gamma} \widehat{Q}_i(x)$$

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

$$360 \quad (18) \quad S_i^{(p)} \longrightarrow U_i$$

361

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

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

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

370 **5.1. Exact Algorithms.** Thanks to UMST characterization given by Lemma 4.7, we can  
 371 compute the UMST of a graph in  $\mathcal{O}(|E|)$  which in the case of 4-adjacency graphs would be  
 372  $\mathcal{O}(|V|)$ . Also one can hope to reduce at least significant number of edges when compared to  
 373  $\mathcal{G}$  in practice.

374 A naive approach is to adapt the Floyd-Warshall [21] algorithm to calculate  $\eta(i, j)$  for  
 375 each pair and the computation of UMST filter takes  $\mathcal{O}(|V|^3)$  time and  $\mathcal{O}(|V|^2)$  space. This  
 376 is very expensive in terms of both memory and space even on a  $500 \times 500$  image and is  
 377 hence not practical. In order to reduce the complexity, one needs to exploit the properties of  
 378 lexicographic ordering and that of the UMST graph structure. One approach is by borrowing  
 379 ideas from Image Foresting Transform (IFT) [19].

380 Image Foresting Transform is a unified framework for several image processing operators  
 381 that are based on shortest paths. Some of these operators include watersheds [35, 6], fuzzy-  
 382 connected segmentation [11, 30] and distance transforms [26, 18]. In simple words, IFT is a  
 383 generalization of Dijkstra’s algorithm where an image (with a specified adjacency relation), a  
 384 set of seeds and a path cost function are specified and one needs to assign to every non-seeded  
 385 pixel, a seed label to which it admits a path with smallest cost. The path costs are usually  
 386 application-specific and are not necessarily given by sum of the weights of the edges on the  
 387 path. Hence, a modified Dijkstra’s algorithm is used to handle a general class of path cost  
 388 functions that arise in computer vision applications. We briefly describe the IFT framework  
 389 below as in [19] and then discuss the relations with the SPF’s:

390 The IFT takes as an input, an image  $I$ , an adjacency relation  $\mathcal{A}$  (usually given by 4-  
 391 adjacency in case of 2D images), a cost function  $\mathcal{C}$  for all paths and outputs an optimum  
 392 spanning forest. Note that the seeds can be specified implicitly by the cost function by  
 393 assigning a fixed cost for every path that starts at a certain pixel (finite for seed pixels and  
 394 infinite for non-seed pixels). Although there are no restrictions on the dimension of the  
 395 image and the adjacency relation, the path costs are restricted and the following are sufficient  
 396 conditions for the optimal IFT algorithm [19] to be applicable:

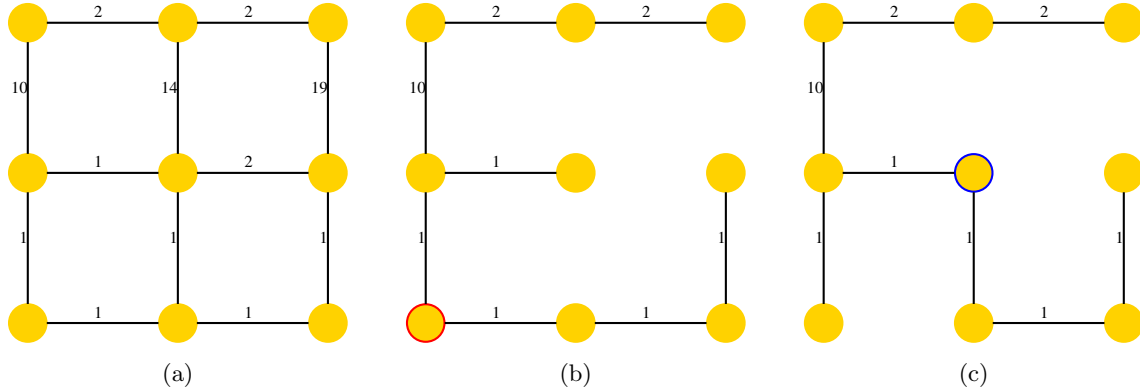
397 For any pixel  $t$ , there is an optimum path  $\pi$  ending at  $t$  which is either trivial or is of the  
 398 form  $\tau \cdot \langle s, t \rangle$ , where

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

402 Using a single seed, the optimum-path forest obtained (which is a tree rooted at the seed)  
 403 can be used to compute the shortest path filtering collaborative weights (see (5)). By varying  
 404 the seed  $s$ , the SPF can be computed for the whole image. However, such an implementation  
 405 would still take  $\mathcal{O}(|V|^2)$  time. To the best of our knowledge, we do not have any linear-time  
 406 exact algorithms for computing the UMST filter. Thus, it calls a need to develop at least a  
 407 quasi-linear algorithm approximation algorithm.

408 **5.2. Approximation Algorithms.** *Single tree-based approximation:* As we have developed  
 409 the UMST filter by generalizing the notion of TF in section 2, we can view TF as a heuristic  
 410 approximation of the UMST filter. Note that TF uses only one MST and hence can be  
 411 computed dynamically in linear time by doing an upward aggregation followed by a downward  
 412 aggregation on the tree (see [38] for details). However, the usage of an arbitrary MST makes  
 413 it difficult to analyse the degree of approximation quantitatively.

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



**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 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**
- 

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

420 *Proof.* Let  $i$  be an arbitrary pixel in  $I$ . We shall construct an adaptive spanning tree  $T_i$ ,  
 421 such that  $T_i$  contains a smallest path with respect to dictionary ordering between pixels  $i$  and  
 422 any other pixel  $j$  in  $I$ . The construction is a special case of IFT (see [19]) with the following  
 423 as the inputs:  $I$  is the image with 4-adjacency. The path costs are given by:  $f(\pi) = \infty$  for  
 424 every path  $\pi$  that does not start at  $i$ . All paths  $\pi_l$  that start at  $i$  are completely ordered in  
 425 the order of decreasing lexicographic ordering using their edge-weight distributions. The path  
 426 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  
 427 paths starting from  $i$  ordered w.r.t. lexicographic ordering then the path cost  $f$  is given by  
 428  $f(\pi_t) = l - t + 1$  where  $1 \leq t \leq l$ . We remark that the this path cost is *monotonic incremental*  
 429 (see [19]). The IFT algorithm applied thus yields  $T_i$ , an adaptive spanning tree of  $i$  with the  
 430 required properties. ■

431 **Proposition 5.1** essentially implies that one can decompose the UMST into possibly differ-  
 432 ent spanning trees  $T_i$  for each  $i \in V$ . Using each of the trees independently (see **Figure 5** for  
 433 an illustration on a synthetic image), one can obtain the exact UMST filter (see **Algorithm 1**).  
 434 The exact computation takes  $\mathcal{O}(|V|^2)$  and  $\mathcal{O}(|V|)$  time and space complexities respectively.  
 435 However, by truncating each of these trees in **Algorithm 1**, one can obtain fast approximate



---

**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**

---

436 solutions. We present two ways to truncate the adaptive trees to obtain approximate UMST  
437 filter.

438

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

442 We rewrite (16) as:

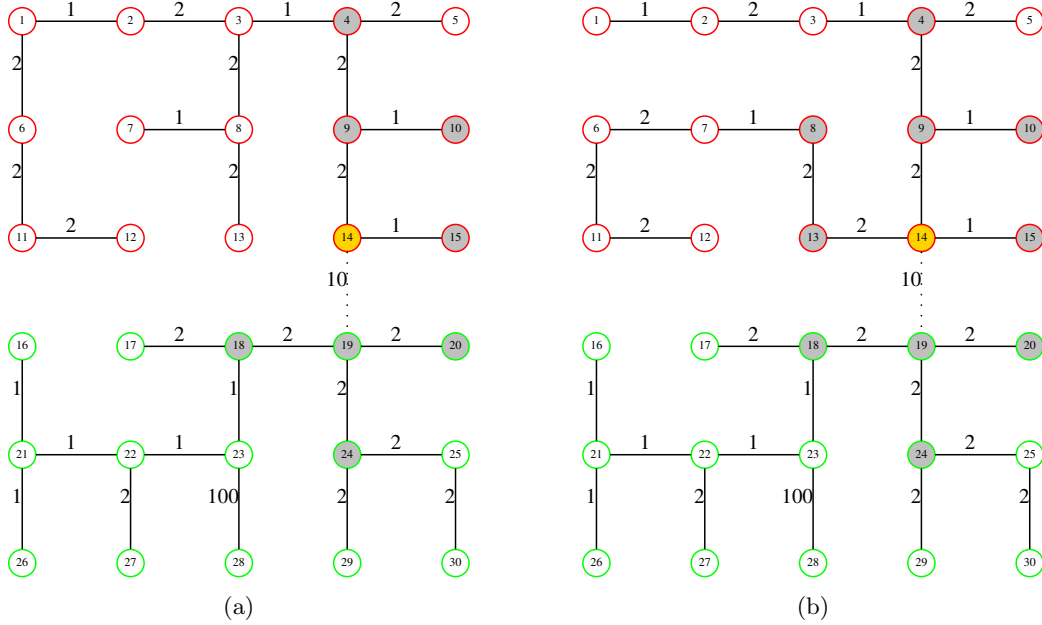
443 (19) 
$$U_i = \frac{1}{C} \sum_l \exp\left(-\frac{l}{\sigma}\right) \sum_{j:\eta(i,j)=l} I_j ,$$

444 where  $C$  is the normalizing constant. In (19), we observe that the exponential term rapidly  
445 converges to 0 and hence one can approximate the above expression by

446 (20) 
$$U_i \approx U_{i,d} = \frac{1}{C} \sum_{l=1}^d \exp\left(-\frac{l}{\sigma}\right) \sum_{j:\eta(i,j)=l} I_j ,$$

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

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



**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.

$$453 \quad (21) \quad \frac{U_i - U_{i,d}}{U_i} = \frac{\sum_{l=d+1}^{|V|-1} \exp(-\frac{l}{\sigma}) \sum_{j:\eta(i,j)=l} I_j}{\sum_{l=0}^{|V|-1} \exp(-\frac{l}{\sigma}) \sum_{j:\eta(i,j)=l} I_j}$$

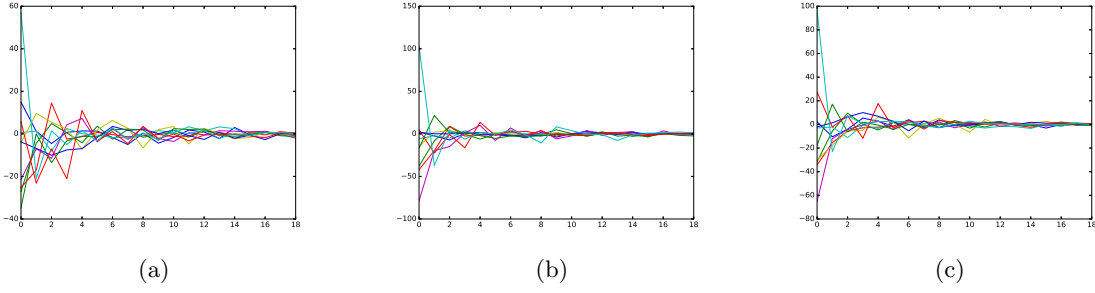
$$454 \quad (22) \quad \leq \frac{\exp(-\frac{d+1}{\sigma}) \sum_{j:\eta(i,j) \geq d+1} I_j}{I_i + \sum_{l=1}^d \exp(-\frac{l}{\sigma}) \sum_{j:\eta(i,j)=l} I_j}$$

$$455 \quad (23) \quad \leq \frac{\exp(-\frac{d+1}{\sigma}) \sum_{j:\eta(i,j) \geq d+1} I_j}{1 + \sum_{l=1}^d \exp(-\frac{l}{\sigma})}$$

$$456 \quad (24) \quad = \frac{\exp(-\frac{d+1}{\sigma})}{1 + \sum_{l=1}^d \exp(-\frac{l}{\sigma})} \sum_{j:\eta(i,j) \geq d+1} I_j$$

$$457 \quad (25) \quad = \frac{\exp(-\frac{d+1}{\sigma})}{1 - \exp(-\frac{d+1}{\sigma})} (1 - \exp(-\frac{1}{\sigma})) \sum_{j:\eta(i,j) \geq d+1} I_j$$

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



**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.

---

**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](#)

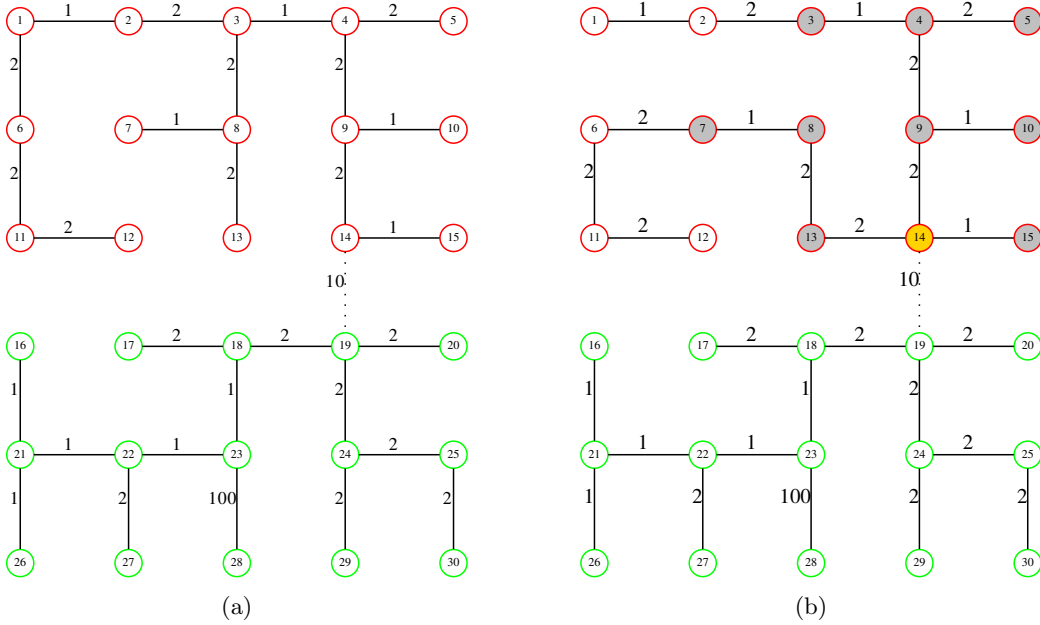
**Output:** Order-Truncated Adaptive Spanning Tree  $\hat{T}_{i,N}$

- 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  $\hat{T}_{i,N}$
- 

462     In what follows, we view a rooted spanning tree  $T$  as a directed spanning tree: for every  
 463 pixel  $j$ , a path  $P^*(j)$  recursively as  $\langle j \rangle$  if parent of  $j$  i.e.  $Parent(j) = nil$ , and  $P^*(j) =$   
 464  $P^*(s) \cdot \langle s, j \rangle$  if  $Parent(j) = s \neq nil$  (notations are borrowed from [19]).

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

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



**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.

472 **Proposition 5.2.** As  $p \rightarrow \infty$ , we have

473 (26) 
$$S_{i,N}^{(p)} \longrightarrow \hat{U}_{i,N}$$

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

477 *Proof.* The proof follows directly from Lemma 4.3. ■

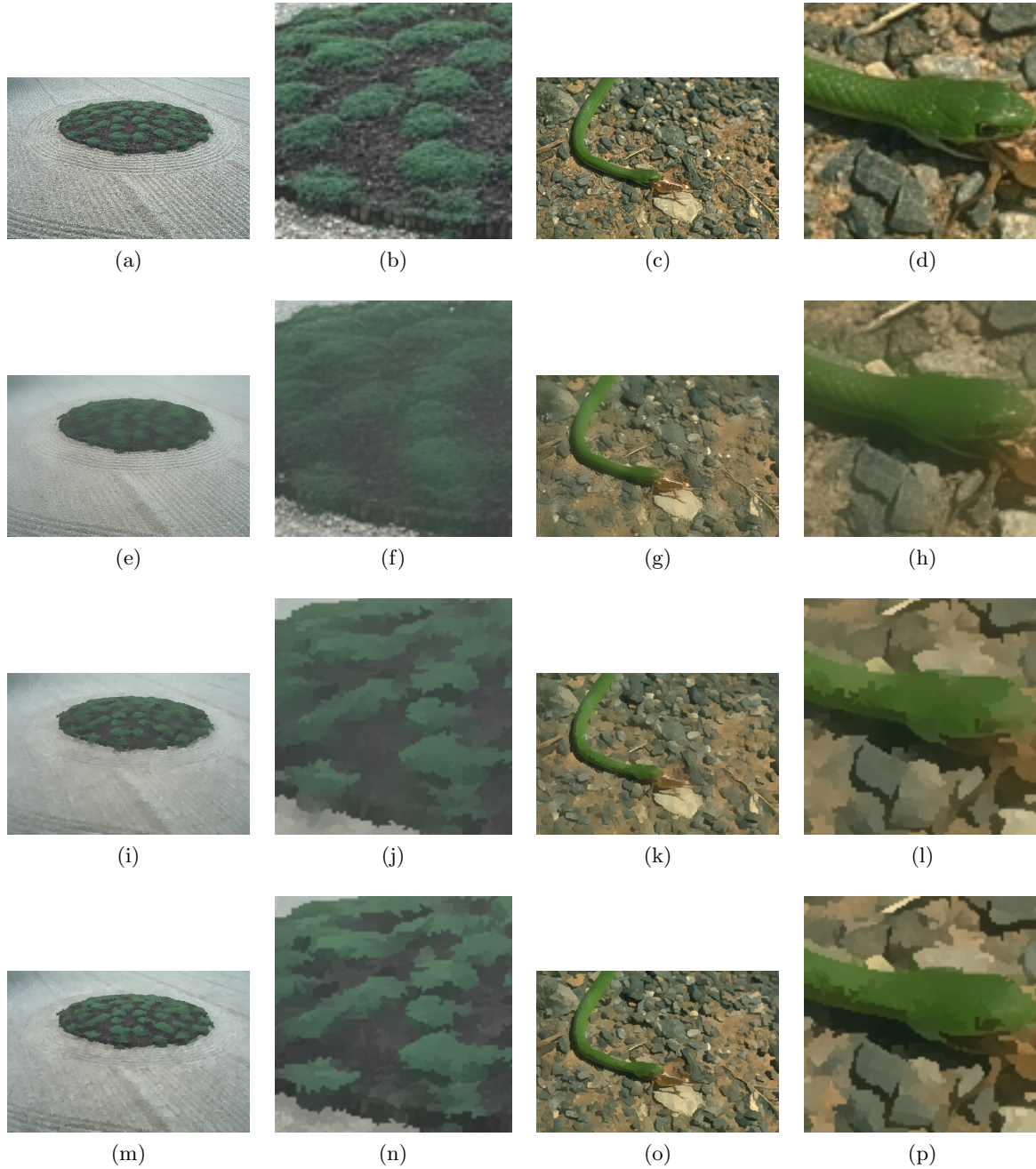
478 We remark that the choice of the kernel size  $N$  determines the trade-off between the level of  
 479 smoothing and the computational cost. In practice, using  $N \approx 100$ , one can obtain a good  
 480 edge-aware filter. As the kernel size is fixed and small, Algorithm 3 runs practically in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

483 **5.3. Experiments.** In this section, we shall demonstrate some qualitative and quantitative  
 484 comparisons of various edge-preserving filters. For all our experiments, we have used images  
 485 from BSDS500 dataset [3]. Also, we have used identical  $\sigma$  ( $= 10$ ) parameter for computing  
 486 the TF and order-based multi-tree approximation (with  $N = 100$ ) of UMSTF. Figure 9 shows  
 487 the comparison of BF, TF and order-based multi-tree approximation on some natural images.  
 488 We observe that BF yields blurry images and erases some object boundaries as expected.



**Figure 9.** Visual illustration of edge-preserving filters. *BF* erases some object boundaries while *TF* and our approximations yield similar results. (a), (b), (c) Original image (d), (e), (f) Bilateral Filter ( $\sigma$  color = 100,  $\sigma$  space = 10) (g), (h), (i) Tree Filter ( $\sigma = 10$ ) (j), (k), (l) Depth-based multi-tree truncation ( $\sigma = 10$ , depth = 15) (m), (n), (o) Order-based multi-tree truncation ( $\sigma = 10$ ,  $N = 100$ )





**Figure 10.** Subtle differences of the performance of TF vs order-based multi-tree approximation of UMSTF are illustrated. In the second and fourth rows, observe the green patches and the snake's eye respectively. The leaks are more prominent in TF when compared to our approximation. (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) Order-based multi-tree truncation ( $\sigma = 10$ ,  $N = 100$ )

489 On the other hand, TF and our approximations yield similar results. However, on a closer  
 490 look, one can observe in Figure 10 that the boundaries are marginally better preserved in our  
 491 approximation (compare green patches in Figure 10 (k), (o) and (s), snake’s eye in Figure 10  
 492 (l), (p) and (t)).

493 **6. Conclusions.** In this paper, we have analysed the edge-aware filters from scratch by de-  
 494 veloping shortest path filters as a natural extension of Gaussian-like filters. We have provided  
 495 a common optimization framework for the filters based on shortest paths and the ones based  
 496 on minimum spanning trees using the notion of  $\Gamma$ -convergence. We have thus established a  
 497 theoretical justification of the MST heuristic based tree filter by proving that the tree filter  
 498 is an approximate  $\Gamma$ -limit of shortest path filters. Further, we have proposed two different  
 499 approximation algorithms of the  $\Gamma$ -limit by leveraging ideas from shortest paths and minimum  
 500 spanning trees.

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

507 **Acknowledgments.** The authors SD and AC would like to thank Indian Statistical Insti-  
 508 tute for providing fellowship to pursue the above research. *BSD and LN acknowledgments to*  
 509 *be added*

## 510 REFERENCES

- 511 [1] C. ALVINO, G. UNAL, G. SLABAUGH, B. PENY, AND T. FANG, *Efficient segmentation based on eikonal*  
 512 *and diffusion equations*, International Journal of Computer Mathematics, 84 (2007), pp. 1309–1324.  
 513 [2] J. ANGULO, *Pseudo-morphological image diffusion using the counter-harmonic paradigm*, in Advanced  
 514 Concepts for Intelligent Vision Systems, Springer, 2010, pp. 426–437.  
 515 [3] P. ARBELAEZ, M. MAIRE, C. FOWLKES, AND J. MALIK, *Contour detection and hierarchical image seg-*  
 516 *mentation*, IEEE transactions on pattern analysis and machine intelligence, 33 (2011), pp. 898–916.  
 517 [4] X. BAI AND G. SAPIRO, *A geodesic framework for fast interactive image and video segmentation and*  
 518 *matting*, in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE, 2007,  
 519 pp. 1–8.  
 520 [5] L. BAO, Y. SONG, Q. YANG, H. YUAN, AND G. WANG, *Tree filtering: Efficient structure-preserving*  
 521 *smoothing with a minimum spanning tree*, IEEE TIP, 23 (2014), pp. 555–569.  
 522 [6] S. BEUCHER AND F. MEYER, *The morphological approach to segmentation: the watershed transformation*,  
 523 Optical Engineering-New York-Marcel Dekker Incorporated-, 34 (1992), pp. 433–433.  
 524 [7] Y. Y. BOYKOV AND M.-P. JOLLY, *Interactive graph cuts for optimal boundary & region segmentation of*  
 525 *objects in nd images*, in Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International  
 526 Conference on, vol. 1, IEEE, 2001, pp. 105–112.  
 527 [8] A. BRAIDES, *Gamma-convergence for Beginners*, vol. 22, Clarendon Press, 2002.  
 528 [9] D. BRUNET, E. R. VRSCAY, AND Z. WANG, *On the mathematical properties of the structural similarity*  
 529 *index*, IEEE Transactions on Image Processing, 21 (2012), pp. 1488–1499.  
 530 [10] J.-H. R. CHANG AND Y.-C. F. WANG, *Propagated image filtering*, in 2015 IEEE Conference on Computer  
 531 Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 10–18.  
 532 [11] K. C. CIESIELSKI, J. K. UDUPA, A. X. FALCÃO, AND P. A. MIRANDA, *Fuzzy connectedness image*  
 533 *segmentation in graph cut formulation: A linear-time algorithm and a comparative analysis*, Journal



- 534 of Mathematical Imaging and Vision, 44 (2012), pp. 375–398.
- 535 [12] C. COUPRIE, L. GRADY, L. NAJMAN, AND H. TALBOT, *Power watershed: A unifying graph-based opti-*  
536 *mization framework*, IEEE PAMI, 33 (2011), pp. 1384–1399.
- 537 [13] M. COUPRIE, G. BERTRAND, ET AL., *Topological grayscale watershed transformation*, in SPIE vision  
538 geometry V proceedings, vol. 3168, 1997, pp. 136–146.
- 539 [14] J. COUSTY, G. BERTRAND, L. NAJMAN, AND M. COUPRIE, *Watershed cuts: Minimum spanning forests*  
540 *and the drop of water principle*, IEEE PAMI, 31 (2009), pp. 1362–1374.
- 541 [15] J. COUSTY, G. BERTRAND, L. NAJMAN, AND M. COUPRIE, *Watershed cuts: Thinnings, shortest path*  
542 *forests, and topological watersheds*, IEEE PAMI, 32 (2010), pp. 925–939.
- 543 [16] A. CRIMINISI, T. SHARP, AND A. BLAKE, *Geos: Geodesic image segmentation*, Computer Vision–ECCV  
544 2008, (2008), pp. 99–112.
- 545 [17] S. DANDA, A. CHALLA, B. D. SAGAR, AND L. NAJMAN, *Power tree filter: A theoretical framework linking*  
546 *shortest path filters and minimum spanning tree filters*, in International Symposium on Mathematical  
547 Morphology and Its Applications to Signal and Image Processing, Springer, 2017, pp. 199–210.
- 548 [18] A. X. FALCÃO, L. DA FONTOURA COSTA, AND B. DA CUNHA, *Multiscale skeletons by image foresting*  
549 *transform and its application to neuromorphometry*, Pattern recognition, 35 (2002), pp. 1571–1582.
- 550 [19] A. X. FALCAO, J. STOLFI, AND R. DE ALENCAR LOTUFO, *The image foresting transform: Theory,*  
551 *algorithms, and applications*, IEEE PAMI, 26 (2004), p. 19.
- 552 [20] Z. FARBMAN, R. FATTAL, D. LISCHINSKI, AND R. SZELISKI, *Edge-preserving decompositions for multi-*  
553 *scale tone and detail manipulation*, in ACM Transactions on Graphics (TOG), vol. 27, ACM, 2008,  
554 p. 67.
- 555 [21] R. W. FLOYD, *Algorithm 97: shortest path*, Communications of the ACM, 5 (1962), p. 345.
- 556 [22] L. GRADY, *Random walks for image segmentation*, IEEE PAMI, 28 (2006), pp. 1768–1783.
- 557 [23] J. GRAZZINI AND P. SOILLE, *Edge-preserving smoothing using a similarity measure in adaptive geodesic*  
558 *neighbourhoods*, Pattern Recognition, 42 (2009), pp. 2306–2316.
- 559 [24] K. HE, J. SUN, AND X. TANG, *Guided image filtering*, in Computer Vision–ECCV 2010, Springer, 2010,  
560 pp. 1–14.
- 561 [25] R. LERALLUT, É. DECENCIÈRE, AND F. MEYER, *Image filtering using morphological amoebas*, Image and  
562 Vision Computing, 25 (2007), pp. 395–404.
- 563 [26] R. D. A. LOTUFO, A. A. FALCÃO, AND F. A. ZAMPIROLI, *Fast euclidean distance transform using a*  
564 *graph-search algorithm*, in Computer Graphics and Image Processing, 2000. Proceedings XIII Brazilian  
565 Symposium on, IEEE, 2000, pp. 269–275.
- 566 [27] L. NAJMAN, *Extending the Power Watershed framework thanks to  $\Gamma$ -convergence*, tech. report, Université  
567 Paris-Est, LIGM, ESIEE Paris, 2017, <https://hal-upec-upem.archives-ouvertes.fr/hal-01428875>.
- 568 [28] L. NAJMAN AND M. COUPRIE, *Watershed algorithms and contrast preservation*, in DGCI, vol. 3, Springer,  
569 2003, pp. 62–71.
- 570 [29] L. NAJMAN, J.-C. PESQUET, AND H. TALBOT, *When convex analysis meets mathematical morphology on*  
571 *graphs*, in International Symposium on Mathematical Morphology and Its Applications to Signal and  
572 Image Processing, Springer, 2015, pp. 473–484.
- 573 [30] P. K. SAHA AND J. K. UDUPA, *Relative fuzzy connectedness among multiple objects: theory, algorithms,*  
574 *and applications in image segmentation*, Computer Vision and Image Understanding, 82 (2001),  
575 pp. 42–56.
- 576 [31] J. STAWIASKI AND F. MEYER, *Minimum spanning tree adaptive image filtering*, in 2009 16th IEEE ICIP,  
577 IEEE, 2009, pp. 2245–2248.
- 578 [32] R. SZELISKI, *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.
- 579 [33] C. TOMASI AND R. MANDUCHI, *Bilateral filtering for gray and color images*, in Sixth International  
580 Conference on Computer Vision, 1998. ICCV 1998, IEEE, 1998, pp. 839–846.
- 581 [34] L. J. VAN VLIET, *Robust local max-min filters by normalized power-weighted filtering*, in Pattern Recog-  
582 nition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 1, IEEE, 2004,  
583 pp. 696–699.
- 584 [35] L. VINCENT AND P. SOILLE, *Watersheds in digital spaces: an efficient algorithm based on immersion*  
585 *simulations*, IEEE Transactions on Pattern Analysis & Machine Intelligence, (1991), pp. 583–598.
- 586 [36] L. XU, C. LU, Y. XU, AND J. JIA, *Image smoothing via  $l_0$  gradient minimization*, in ACM Transactions  
587 on Graphics (TOG), vol. 30, ACM, 2011, p. 174.

- 588 [37] L. XU, Q. YAN, Y. XIA, AND J. JIA, *Structure extraction from texture via relative total variation*, ACM  
589 Transactions on Graphics (TOG), 31 (2012), p. 139.  
590 [38] Q. YANG, *Stereo matching using tree filtering*, IEEE PAMI, 37 (2015), pp. 834–846.