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

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

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

Sravan Danda†, Aditya Challa†, B.S.Daya Sagar†, and Laurent Najman‡

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 Γ-limit of Shortest Path Filters (SPFs) and show that it is the same as UMST filter. Consequently, TF can be viewed as an approximate Γ-limit of the SPFs, thereby providing a theoretical basis to it's 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 Γ-limit by showing that TF and one of our approximations yield similar results.

Key words. Optimization, Image Filtering, Γ-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.

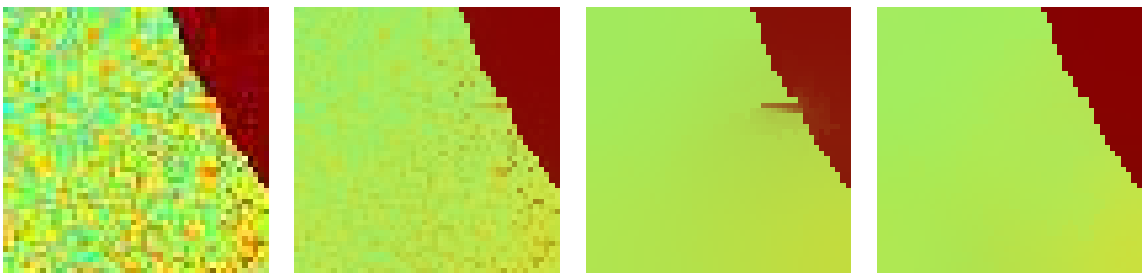
In the literature, there exist several edge-aware smoothing filters such as bilateral filter (BF) [35], guided filter (GF) [26], weighted least squares filter (WLS) [22], L_0 smoothing [37], propagated image filter [11], morphological amoebas or adaptive kernel based filters [27], tree filter (TF) [6] 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.

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

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



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

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

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 [39] and yields promising results in applications such as denoising, texture removal, stereo matching and scene simplification. However it exhibits a leak problem at some of the object boundaries. This problem occurs due to the presence of some object boundary edges in the MST and cannot be avoided as any spanning tree connects all the nodes in a connected graph. Although, the authors in [6] tried to negate the leak effect using a bilateral filter as a post-processing step, the problem still persists (see Figure 1(c)). Also, the filtering results vary with the choice of MST which is undesirable.

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

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

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

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

2. Union Minimum Spanning Tree Filter.

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

2.1. Tree Filter.

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

$$w_{ij} = ||I_i - I_j|| \tag{1}$$

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

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

where σ controls the falling rate and the summation over q is over all the pixels in the graph.

The tree filtered value at pixel i is given by

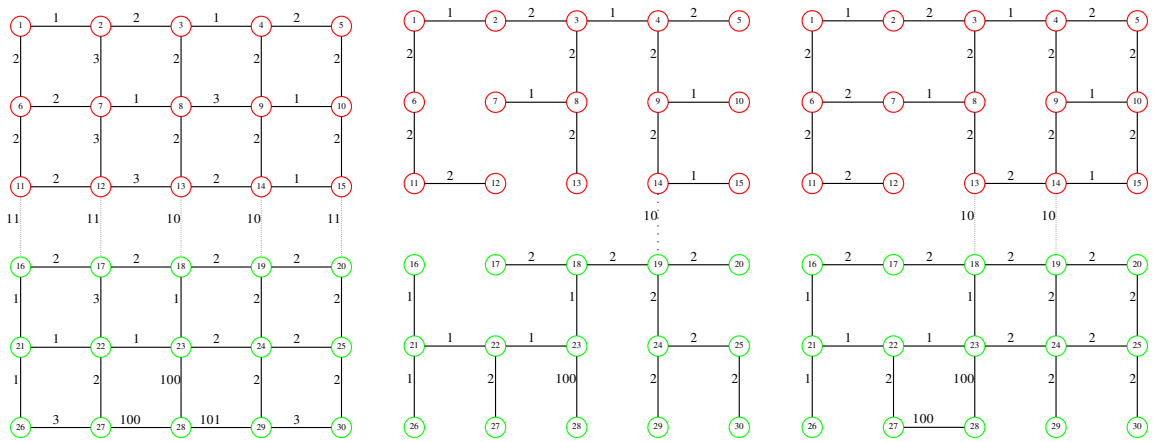
$$S_i = \sum_j t_i(j) I_j \tag{3}$$

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

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

2.2. Why UMST filter?.

The edges in the image graph that do not belong to object boundaries induce a disconnected subgraph. On the other hand, a MST of a graph is connected, hence any MST contains one or more object boundary edges. These edges cause a leak effect in the tree filtered image (see Figure 1(c)). Also, the filtering results vary with the



(a) (b) (c)

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* $13-14$*.*

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

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

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

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

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

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

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

Now, we shall recall notions of Γ-convergence and the power watershed framework before we state the main result of the paper.

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

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

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

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

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

3. Shortest Path Filters and Related Methods.

In this section, we shall review the shortest path filters in detail. In particular, we develop them as a natural edge-aware extension of Gaussian-like filters. The rest of the section is dedicated to the discussions on their links with other related geodesic methods.

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

3.1. Basic Notions.

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

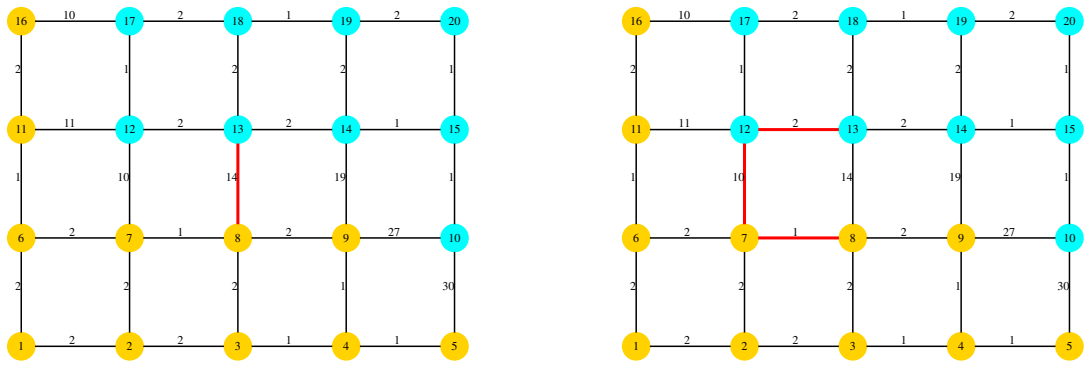
Definition 3.2. *For $p \in \mathbb{Z}^+$, we denote by $\mathcal{G}^{(p)} = (V, E, W^{(p)})$, the graph that contains the same set of nodes and edges as of $\mathcal{G}$ and $W^{(p)}(e_{ij}) = (W(e_{ij}))^p$ for each edge $e_{ij} \in E$ and we call $\mathcal{G}^{(p)}$ as an **exponentiated graph** of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

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

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

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

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



(a) (b)

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* $< 8, 7, 12, 13 >$ *highlighted in red in (b) for computing the corresponding collaborative weight. This illustrates that SPF respects the object boundaries.*

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 have* $\sum_{r=1}^{k} n_r w_r \leq \sum_{r=1}^{k} m_r w_r$ *where* $(n_1, \cdots, n_k)$ *and* $(m_1, \cdots, m_k)$ *denote the edge-weight distributions of* $P(i,j)$ *and* $Q(i,j)$ *respectively.*

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

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

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

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

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

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

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

where the edge weights are given by:

$$w_{ij} = ||I_i - I_j|| + 1 \tag{6}$$

The SPF at pixel i is defined as:

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

where σ controls the falling rate and the summations are over all pixels in the image.

Note that the weights in Eq. (6) are different from the ones in Eq. (1) to ensure that the weights are strictly positive. We use the edge weights as per (6) in the rest of the paper. 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

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

where σ 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 [25], the adaptive kernel filter such as morphological amoebas [27].

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:

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

where $P(i,j)$ is a path between pixels i and j, $< i = x_0, x_1, \cdots, x_n = j >$ and $\lambda \geq 0$ is a user input.

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

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.

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

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

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

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

where

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

Proof. It readily follows from (5), (6), (10) and (13) ∎

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

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

(a) (b)

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*

4.1. Some Definitions.

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

$$P \geq Q \Leftrightarrow A = \emptyset \text{ or } n_l > m_l \tag{14}$$

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

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

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

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

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

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

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

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

path between i and j. Let $(n_1, \cdots, n_k)$ and $(m_1, \cdots, m_k)$ denote the edge-weight distributions of paths $P(i,j)$ and $Q(i,j)$ respectively. Let $A(P,Q) = \{1 \leq r \leq k : n_r \neq m_r\}$. 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 $l = sup(A(P,Q))$. We have $m_l > n_l$ by choice of $P(i,j)$. Also, the difference of the total 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 $\exists p_{Q(i,j)} \geq 1$ 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 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 note that given any path which is not simple, one can drop the redundant edges to construct a simple path with strictly smaller total weight. It is hence enough to show that the total weight of $P(i,j)$ is lesser than or equal to every simple path between i and j in $\mathcal{G}^{(p)}$ for sufficiently large p. As V is finite, setting $p_0 = \sup\{p_{ij} : i,j \in V\}$ completes the argument.

Conversely, suppose $P(i,j)$ is NOT a smallest path w.r.t. dictionary ordering between i and j, we shall construct a sequence $(p_n)_{n\geq 1}$ converging to ∞ such that $P(i,j)$ is not 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 path w.r.t. dictionary ordering between i and j, $\exists$ a path $T(i,j)$ between i and j such that $P(i,j) \geq T(i,j)$ holds but $T(i,j) \geq P(i,j)$ does not hold. Equivalently, if the edge weight distributions of $P(i,j)$ and $T(i,j)$ are given by $(n_1, \cdots, n_k)$ and $(t_1, \cdots, t_k)$ respectively then $A(P,T) \neq \emptyset$ and $t_l < n_l$ where $l = \sup(A(P,T))$ and $A(P,T) = \{1 \leq r \leq k : n_r \neq t_r\}$. 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 leading coefficient. Thus, $\exists$ a constant $\rho_{ij} \geq 1$ such that for each $p \geq \rho_{ij}$, $P(i,j)$ is not a shortest path between i and j. We set $(p_n)_{n\geq 1}$ by $p_n = \rho_{ij} + n - 1$ to complete the proof. ■

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

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

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

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

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

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

where U denotes the UMST filtered image.

Firstly, we need the following:

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

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

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

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

Proof. The proof directly follows from Lemma 4.5 and Lemma 4.6. ∎

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

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

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

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

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

The main result is formally stated as follows:

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

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

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

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

Proof. The proof follows readily from Proposition 4.8 and Lemma 4.3. ∎

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

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

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

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

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

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

For any pixel t, there is an optimum path π ending at t which is either trivial or is of the form $\tau\cdot < s,t >$, where

- $C(\tau) \leq C(\pi)$,
- τ is an optimum path ending at s,
- $C(\tau'\cdot < s,t >) = C(\pi)$ if τ' is an optimum path ending at s

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**

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

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

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

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

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

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

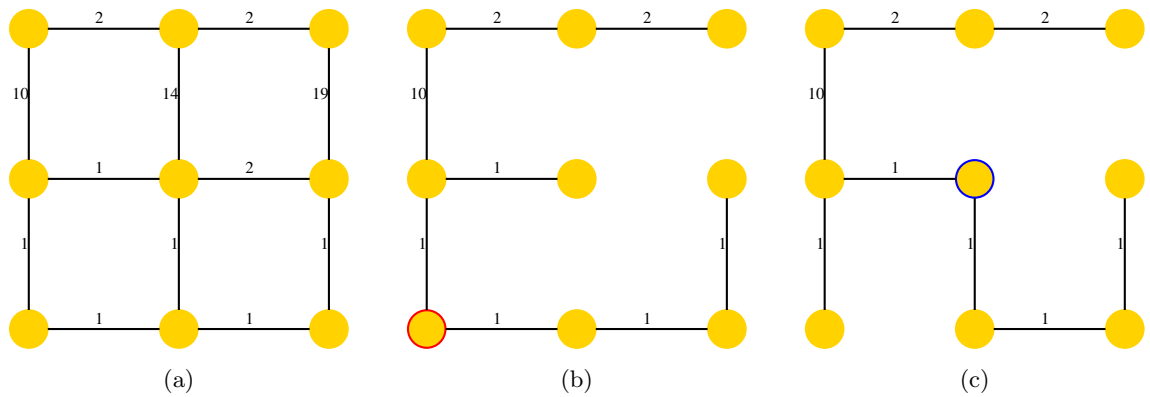


Figure 5. *(a) A synthetic image with the edge-weights reflecting the intensity dissimilarities, (b) and (c) Adaptive spanning trees of the pixels circled in red and blue respectively.*

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 $T_{i,d}$

Set $X = \{i\}$ and $T_{i,d} = (i, \emptyset)$
while True **do**
  break = **true**
  for e in shortest edges from X to X^c **do**
    if $dist(e, i, T_{i,d}) < d$ **then**
      $T_{i,d} \cup e$
      break = **false**
    end if
  end for
  if break = **true then**
    return $T_{i,d}$.
  end if
end while

filter.

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

We rewrite (16) as:

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

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

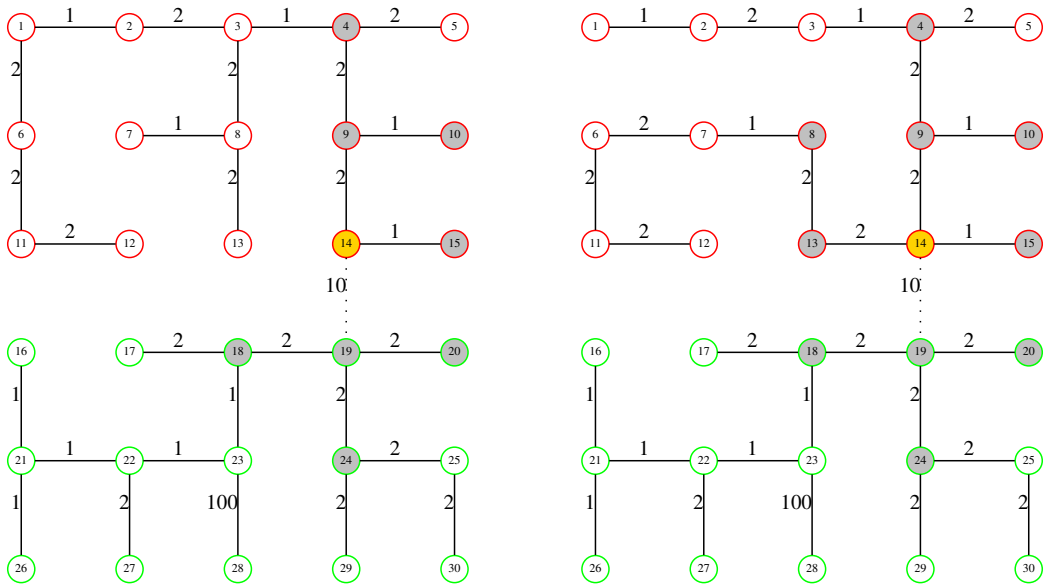


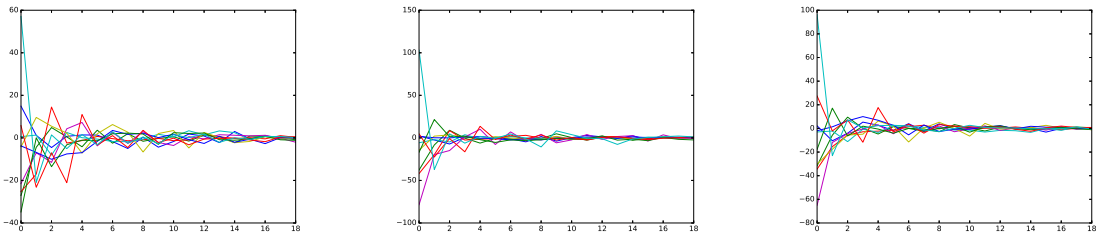
Figure 6. *(a) TF of pixel numbered* 14 *in Figure* 2*, here the pixels with significant collaboration (distance* ≤ 2*) are highlighted in grey (b) Depth-based truncation of UMST filter at pixel numbered* 14 *in Figure* 2*, here the pixels with significant collaboration (depth* ≤ 2*) are highlighted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

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

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

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

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



(a) (b) (c)

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.*

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

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

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

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

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

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

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

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

The order-based truncation of the adaptive spanning tree would precisely compute the Γ-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 $\hat{T}_{i,N}$

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

following result:

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

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

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

Proof. The proof follows directly from Lemma 4.3. ∎

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

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

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

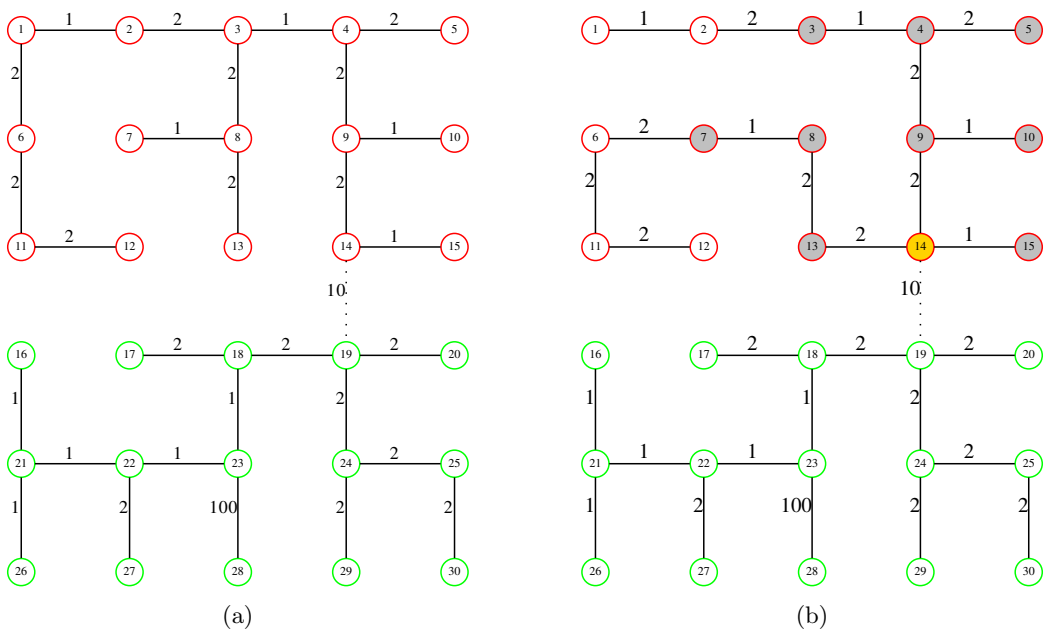


Figure 8. *(a) TF of pixel numbered* 14 *in Figure* 2 *(b) Order-based truncation of PTF at pixel numbered* 14 *in Figure* 2*, here the pixels that are closest w.r.t. lexicographic order from* 14 *(top* 10 *including itself) are highlighted in grey. Observe that the collaboration in the PTF within the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

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

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

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

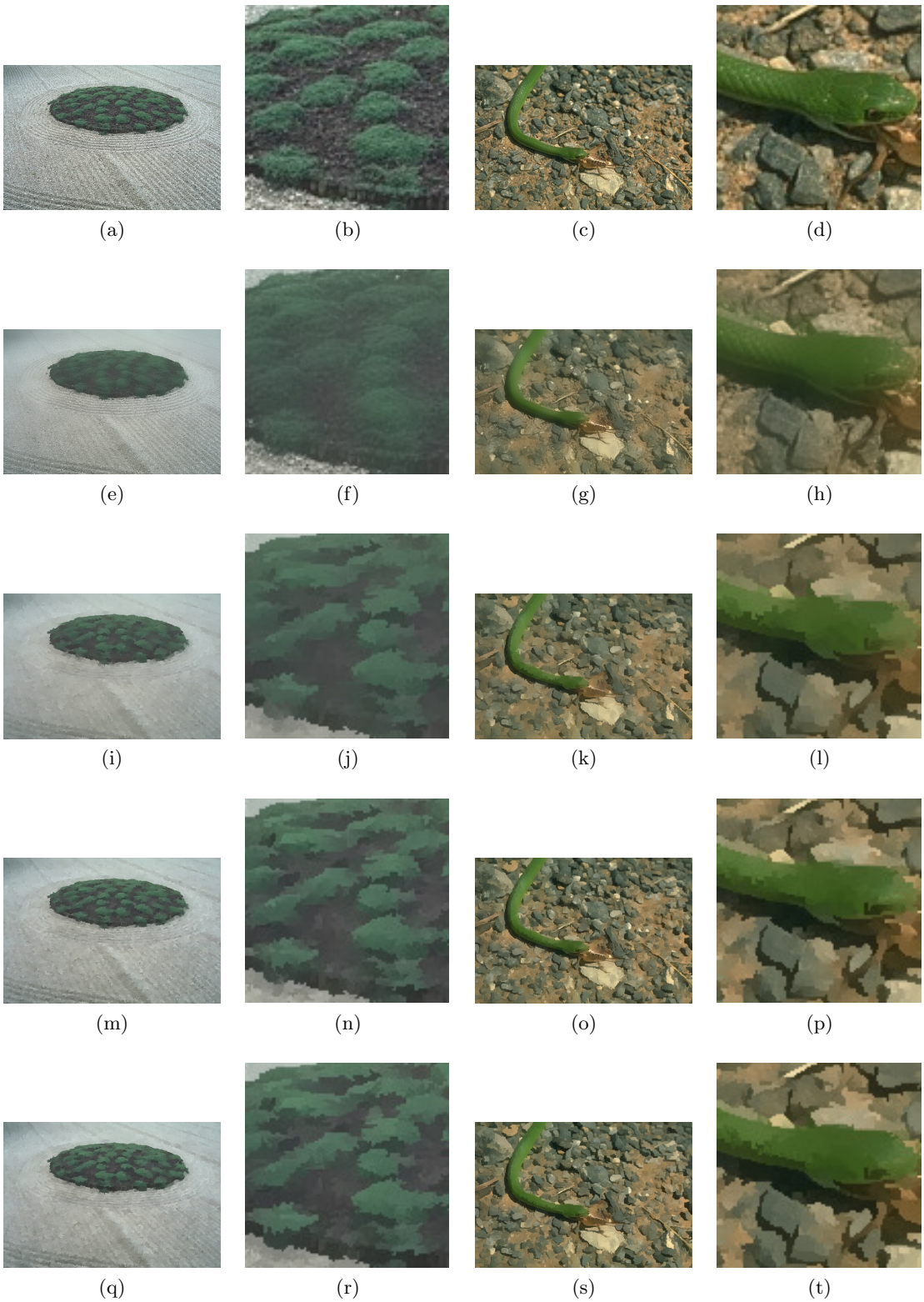


Figure 9. *A visual comparison of the subtle differences in the performance of TF and our multi-tree approximations of UMSTF on BSDS500 [4] images are illustrated. (a), (b), (c), (d) original images (e), (f), (g), (h) Bilateral Filter (σ color $= 100$, σ space $= 10$) (i), (j), (k), (l) Tree Filter ($\sigma = 10$) (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.*

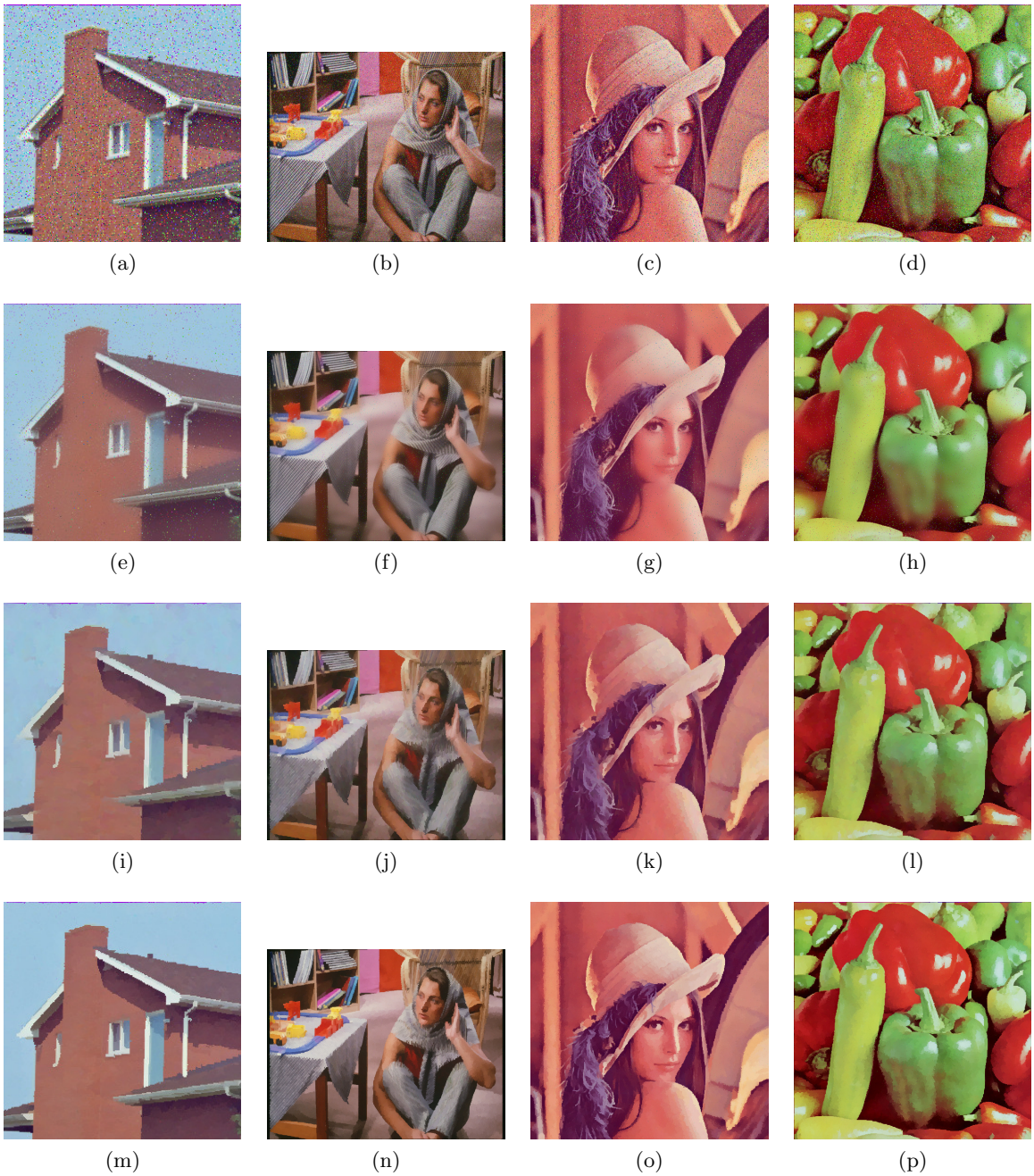


Figure 10. *Visual illustration of edge-preserving filters on images contaminated with salt and pepper noise. BF erases some object boundaries and does not eliminate the noise completely while TF and order-based approximation eliminate noise and yield similar results. (a), (b), (c), (d) Salt and pepper noisy images (e), (f), (g), (h) Bilateral Filter (σ color $= 100$, σ 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		
	Bilateral Filter	**Tree Filter**	**PTF Order-based**
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
Mean	**23.27**	**22.95**	**22.59**

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		
	Bilateral Filter	**Tree Filter**	**PTF Order-based**
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
Mean	**0.68**	**0.74**	**0.77**

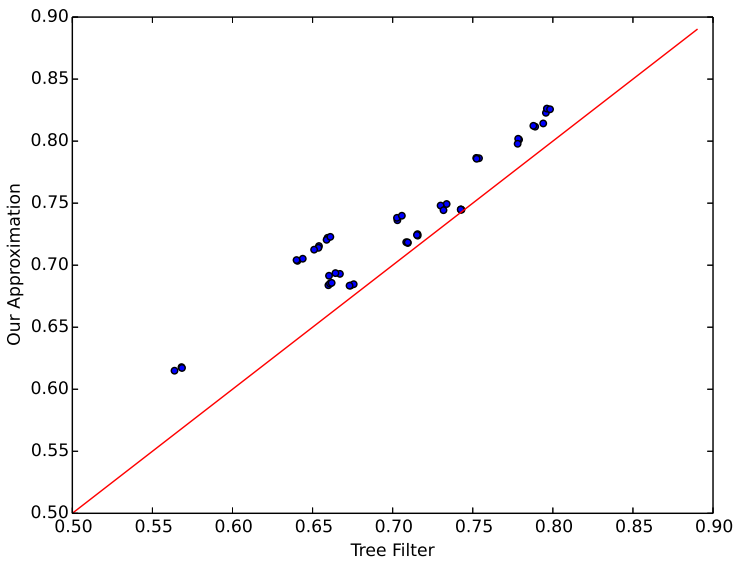


Figure 11. *Scatter plot of SSIM values of TF versus Order-based approximation of UMST filter on three iterations of random Poisson, Salt and Pepper, Gaussian and Speckle noises on House, Barbara, Lena and Pepper images.*

computationally expensive. However, it is important to note that our approximations can be implemented in parallel as each pixel is processed independently of the other. Also we note that one has to choose an appropriate filter depending on the type of noise. For instance, BF outperforms the tree-based filters in presence of Gaussian noise (see Table 3 and Table 4). To summarize, we have demonstrated that TF is a fast approximation of the Γ-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		
	Bilateral Filter	**Tree Filter**	**PTF Order-based**
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
Mean	**23.52**	**21.88**	**22.07**

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		
	Bilateral Filter	**Tree Filter**	**PTF Order-based**
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
Mean	**0.77**	**0.66**	**0.69**

6. Conclusions.

In this paper, we have analysed the edge-aware filters from scratch by developing shortest path filters as a natural extension of Gaussian-like filters. We have provided a common optimization framework for the well-known filters based on shortest paths and the ones based on minimum spanning trees in the power watershed framework. We have thus established a theoretical justification of the MST heuristic based tree filter by proving that the tree filter is an approximate Γ-limit of shortest path filters. Further, we have proposed two different approximation algorithms of the Γ-limit by leveraging ideas from shortest paths and minimum spanning trees.

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

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

REFERENCES

[1] C. Alvino, G. Unal, G. Slabaugh, B. Peny, and T. Fang, *Efficient segmentation based on eikonal and diffusion equations*, International Journal of Computer Mathematics, 84 (2007), pp. 1309–1324.
[2] L. Ambrosio and V. M. Tortorelli, *Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence*, Comm. Pure Appl. Math., 43 (1990), pp. 999–1036, https://doi.org/10.1002/cpa.3160430805.
[3] J. Angulo, *Pseudo-morphological image diffusion using the counter-harmonic paradigm*, in Advanced Concepts for Intelligent Vision Systems, Springer, 2010, pp. 426–437.
[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, *Contour detection and hierarchical image segmentation*, IEEE transactions on pattern analysis and machine intelligence, 33 (2011), pp. 898–916.
[5] X. Bai and G. Sapiro, *A geodesic framework for fast interactive image and video segmentation and matting*, in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE, 2007, pp. 1–8.
[6] L. Bao, Y. Song, Q. Yang, H. Yuan, and G. Wang, *Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree*, IEEE TIP, 23 (2014), pp. 555–569.
[7] Y. Y. Boykov and M.-P. Jolly, *Interactive graph cuts for optimal boundary & region segmentation of objects in nd images*, in Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, vol. 1, IEEE, 2001, pp. 105–112.
[8] A. Braides, *Gamma-convergence for Beginners*, vol. 22, Clarendon Press, 2002.
[9] D. Brunet, E. R. Vrscay, and Z. Wang, *On the mathematical properties of the structural similarity index*, IEEE Transactions on Image Processing, 21 (2012), pp. 1488–1499.
[10] A. Challa, S. Danda, B. S. Daya Sagar, and L. Najman, *An Introduction to Gamma-Convergence for Spectral Clustering*, in Discrete Geometry for Computer Imagery, vol. 10502 of Lecture Note In Computer Sciences, Vienna, Austria, Sept. 2017, Kropatsch, Walter G. and Artner, Nicole M. and Janusch, Ines, Springer, pp. 185–196, https://hal.archives-ouvertes.fr/hal-01427957.
[11] J.-H. R. Chang and Y.-C. F. Wang, *Propagated image filtering*, in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 10–18.
[12] K. C. Ciesielski, J. K. Udupa, A. X. Falcão, and P. A. Miranda, *Fuzzy connectedness image segmentation in graph cut formulation: A linear-time algorithm and a comparative analysis*, Journal of Mathematical Imaging and Vision, 44 (2012), pp. 375–398.
[13] C. Couprie, L. Grady, L. Najman, and H. Talbot, *Power watershed: A unifying graph-based optimization framework*, IEEE PAMI, 33 (2011), pp. 1384–1399.
[14] M. Couprie, G. Bertrand, et al., *Topological grayscale watershed transformation*, in SPIE vision geometry V proceedings, vol. 3168, 1997, pp. 136–146.
[15] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, *Watershed cuts: Minimum spanning forests and the drop of water principle*, IEEE PAMI, 31 (2009), pp. 1362–1374.
[16] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, *Watershed cuts: Thinnings, shortest path forests, and topological watersheds*, IEEE PAMI, 32 (2010), pp. 925–939.
[17] A. Criminisi, T. Sharp, and A. Blake, *Geos: Geodesic image segmentation*, Computer Vision–ECCV 2008, (2008), pp. 99–112.
[18] S. Danda, A. Challa, B. D. Sagar, and L. Najman, *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.
[19] Y. C. de Verdiere, Y. Pan, and B. Ycart, *Singular limits of schrodinger operators and markov*

processes, Journal of Operator Theory, 41 (1999), pp. 151–173.
[20] A. X. Falcão, L. da Fontoura Costa, and B. Da Cunha, *Multiscale skeletons by image foresting transform and its application to neuromorphometry*, Pattern recognition, 35 (2002), pp. 1571–1582.
[21] A. X. Falcao, J. Stolfi, and R. de Alencar Lotufo, *The image foresting transform: Theory, algorithms, and applications*, IEEE PAMI, 26 (2004), p. 19.
[22] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, *Edge-preserving decompositions for multi-scale tone and detail manipulation*, in ACM Transactions on Graphics (TOG), vol. 27, ACM, 2008, p. 67.
[23] R. W. Floyd, *Algorithm 97: shortest path*, Communications of the ACM, 5 (1962), p. 345.
[24] L. Grady, *Random walks for image segmentation*, IEEE PAMI, 28 (2006), pp. 1768–1783.
[25] J. Grazzini and P. Soille, *Edge-preserving smoothing using a similarity measure in adaptive geodesic neighbourhoods*, Pattern Recognition, 42 (2009), pp. 2306–2316.
[26] K. He, J. Sun, and X. Tang, *Guided image filtering*, in Computer Vision–ECCV 2010, Springer, 2010, pp. 1–14.
[27] R. Lerallut, É. Decencière, and F. Meyer, *Image filtering using morphological amoebas*, Image and Vision Computing, 25 (2007), pp. 395–404.
[28] R. d. A. Lotufo, A. A. Falcão, and F. A. Zampirolli, *Fast euclidean distance transform using a graph-search algorithm*, in Computer Graphics and Image Processing, 2000. Proceedings XIII Brazilian Symposium on, IEEE, 2000, pp. 269–275.
[29] L. Najman, *Extending the PowerWatershed framework thanks to Γ-convergence*, SIAM Journal on Imaging Sciences, 10 (2017), pp. 2275–2292, https://doi.org/10.1137/17M1118580, https://hal.archives-ouvertes.fr/hal-01428875.
[30] L. Najman, M. Couprie, and G. Bertrand, *Watersheds, mosaics, and the emergence paradigm*, Discrete Applied Mathematics, 147 (2005), pp. 301–324.
[31] L. Najman, J.-C. Pesquet, and H. Talbot, *When convex analysis meets mathematical morphology on graphs*, in International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, Springer, 2015, pp. 473–484.
[32] P. K. Saha and J. K. Udupa, *Relative fuzzy connectedness among multiple objects: theory, algorithms, and applications in image segmentation*, Computer Vision and Image Understanding, 82 (2001), pp. 42–56.
[33] J. Stawiaski and F. Meyer, *Minimum spanning tree adaptive image filtering*, in 2009 16th IEEE ICIP, IEEE, 2009, pp. 2245–2248.
[34] R. Szeliski, *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.
[35] C. Tomasi and R. Manduchi, *Bilateral filtering for gray and color images*, in Sixth International Conference on Computer Vision, 1998. ICCV 1998, IEEE, 1998, pp. 839–846.
[36] L. J. Van Vliet, *Robust local max-min filters by normalized power-weighted filtering*, in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 1, IEEE, 2004, pp. 696–699.
[37] L. Xu, C. Lu, Y. Xu, and J. Jia, *Image smoothing via L_0 gradient minimization*, in ACM Transactions on Graphics (TOG), vol. 30, ACM, 2011, p. 174.
[38] L. Xu, Q. Yan, Y. Xia, and J. Jia, *Structure extraction from texture via relative total variation*, ACM Transactions on Graphics (TOG), 31 (2012), p. 139.
[39] Q. Yang, *Stereo matching using tree filtering*, IEEE PAMI, 37 (2015), pp. 834–846.