Power Tree Filter: A Theoretical Framework Linking Shortest Path Filters and Minimum Spanning Tree Filters

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

Power Tree Filter: A Theoretical Framework Linking Shortest Path Filters and Minimum Spanning Tree Filters

Sravan Danda[1], Aditya Challa[1], B.S.Daya Sagar[1], Laurent Najman[2]

[1] Systems Science and Informatics Unit, Indian Statistical Institute, 8th Mile, Mysore Road, Bangalore, India.
[2] Université Paris-Est, LIGM, Equipe A3SI, ESIEE, France.

Abstract. Edge-preserving image filtering is an important pre-processing step in many filtering applications. In this article, we analyse the basis of edge-preserving filters and also provide theoretical links between the MST filter [1], which is a recent state-of-art edge-preserving filter, and filters based on geodesics. We define *shortest path filters*, which are closely related to adaptive kernel based filters [11] [15], and show that MST filter is an approximation to the $\Gamma-$limit of the shortest path filters. We also propose a different approximation for the $\Gamma-$limit that is based on union of all MSTs and show that it yields better results than that of MST approximation by reducing the leaks across object boundaries. We demonstrate the effectiveness of the proposed filter in edge-preserving smoothing by comparing it with the tree filter.

1 Introduction

Image filtering is a prominent problem in computer vision which has been studied for several years. Along with the relevant information, images often contain noise as well as irrelevant information such as texture. Image filtering is usually the first step of several applications such as image abstraction, texture removal, texture editing, stereo matching, optical flow etc. One needs to make sure that relevant information does not get 'smoothed' out during the filtering step, and thus edge preserving filters were developed. There exists several filters for edge-preserving smoothing such as bilateral filter (BF) [16], guided filter (GF) [10], weighted least squares filter (WLS) [7] and L_0 smoothing [17]. Among the other noted techniques are propagated image filter [2] and relative total variation (RTV) filter [18].

Minimum Spanning Tree (MST) filters are one of the edge preserving filters which have been proved to be effective in applications such as scene simplification, texture editing and stereo matching. To the best of our knowledge, Stawiaski et. al [15] first used a MST for filtering. Later, tree filter (TF) [1] was developed as a non-local weighted-average filter on similar idea. The TF can be implemented in linear time [19] (linear in number of pixels) and yields promising results. Although the TF yields good results, it admits a drawback (termed

as leak problem) due to the fact that any MST has some higher weight edges leading to high collaboration across the object boundaries. As a result, the edges across these boundaries are not well preserved in the filtered image and cause a leakage. Also, these filters are based on heuristics and exploring theoretical links with other existing filters sheds more light on why these work well.

In this article, we aim to provide a theoretical justification for the MST filtering as described in [1] using the concept of $\Gamma-$limit (defined in Sec. 3). Loosely speaking, we show that the MST filtering is an approximate solution to the $\Gamma-$limit of the shortest path filters which are morphological amoeba-like filters. We also describe another approximation, which reduces the 'leak' problem.

2 Background

2.1 Tree Filter

In this section we describe the MST filtering as in [1]. Let I be a given image and let S denote the filtered image. Let I_i denote the intensity or color of pixel i in the image. Note that given an image I, one can construct a 4-adjacency graph, with weights between adjacent pixels reflecting the absolute difference of intensities or color. More formally, for adjacent pixels i and j, with intensities I_i and I_j respectively, we can define

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

Note that adding a positive constant is for our convenience and the MSTs of the graph are invariant. With slight abuse of notation, we refer to this graph by I as well. One can then construct an MST on this graph, say I_{MST}. Note that on any given MST, there exists a unique path between any two pixels. Let $D(i,j)$ denote the number of edges on the path between the pixels i and j on I_{MST}. Define $t_i(j)$ by

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

Then tree filter is defined as, for each pixel i

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

where $t_i(j)$ denotes the collaborative weights between pixels i and pixel j, and the summation is over all the pixels.

The main advantage of the tree filter is the edge preserving property. Intuitively, any MST includes only a few high weighted edges and thus reduces the collaborations across most of the object boundaries. However, as any MST is connected, the existence of at least one edge on the boundary of every object is guaranteed. These edges lead to high collaborative weights and the 'leaks' arise.

Apart from the 'leak' problem, the above method does not have a theoretical justification for its working. Another drawback of this method is: since MST of a graph is not unique, the filtering results depends on the choice of MST which is not desired.

2.2 Gamma Convergence

The main idea of $\Gamma-$convergence [12] is to calculate a limit of minimizers of a family of minimum problems. This limit is referred to as a $\Gamma-$limit. The main advantages of such a calculation are - 1) $\Gamma-$limits share a few properties with the minimizers and are sometimes easier to calculate 2) $\Gamma-$limits also exhibit interesting 'new' properties and usually result in novel methods. For instance, $\Gamma-$limit of methods described in [4] [5] [13] [9] was calculated in [3] and is shown to have interesting properties. In Sec. 3, we show that the MST filter described above can be seen as an approximation to the $\Gamma-$limit of shortest path filters. This then would provide theoretical links to justify the edge-preserving capability of the MST filter.

3 Power Tree Filter

Let $\mathcal{G} = (V, E, W)$ be an edge-weighted graph, where V denotes the set of pixels, E denotes the set of edges given by the 4-adjacency relation and w_{ij} indicates the weight on the edge between pixels i and j. Let $W^{(p)}$ denote the exponentiated weights, $\{w_{ij}^p\}$. The graph $\mathcal{G}^{(p)} = (V, E, W^{(p)})$ is called an *exponentiated graph*. We also assume that there are k distinct weights $0 < w_1 < w_2 < \cdots < w_k$.

Note that to every path between pixels i and j, $P(i, j)$, one can assign a tuple $(p_1, p_2, \cdots, p_k)$, where p_i denotes the number of edges of weight w_i. We denote this tuple as the *edge-weight distribution* of the path. Let $(p_1, p_2, \cdots, p_k)$ and $(q_1, q_2, \cdots, q_k)$ be edge-weight distributions of two paths P and Q respectively. Let $A = \{1 \leq i \leq k : p_i \neq q_i\}$, We say that

$$P \geq Q \Leftrightarrow \text{if } A = \emptyset \text{ or } p_{sup(A)} > q_{sup(A)} \tag{4}$$

This is referred to as the *dictionary ordering*. Note that this induces a complete ordering on the set of paths. We call $P(i, j) = (p_1, p_2, \cdots, p_k)$ a *shortest path* if $\sum w_l p_l$ is smallest for P compared to any other path $Q(i, j)$. Let $\Pi(P(i, j))$ denote the number of edges on the path $P(i, j)$.

The overview of the rest of the section is - We first define *shortest path filters*, and give a heuristic explanation as to their advantage. We then explore their relation to morphological amoeba filters and show that these filters can be seen as minimizers to a cost function. We then calculate the $\Gamma-$limit of these minimizers, and show that it is the shortest path filter on the union of minimum spanning trees (UMST).

3.1 Shortest Path Filters

Let i and j be any two pixels. Define

$$\Theta(i,j) = \inf\{\Pi(P(i,j)) \text{ where } P(i,j) \text{ is a shortest path.}\} \tag{5}$$

Note that $\Theta(i,j)$ is well defined. Now, consider the cost function

$$Q_i(x) = \sum_j exp\left(-\frac{\Theta(i,j)}{\sigma}\right)(x - I_j)^2 \tag{6}$$

Then the filtered value at pixel i is given by the minimizer of $Q_i(x)$. From Eq. (6), it can easily be seen that

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

where S denotes the filtered image. Shortest path filters can be seen as an edge aware extension to the gaussian convolution. Note that gaussian filter considers exponential decay over spatial distances without considering the edge weights. To adapt the gaussian filter to be edge aware, one might consider taking number of edges on a shortest path instead of spatial distances. In the special case of all pixel values being equal in the image, shortest path filter is exactly gaussian filter with l_1 metric. Shortest path filters are also related to the adaptive kernel filter such as morphological amoebas [11], as discussed below.

Relation to Morphological Amoebas In [11] the authors introduced morphological amoebas (adaptive kernels). The kernel is dictated by

$$\kappa(i,j) = \min_{P(i,j)} L_\lambda(P(i,j)) \ , \tag{8}$$

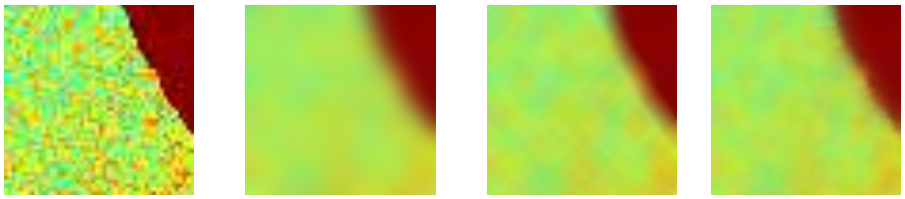
where $P(i,j)$ is a path between pixels i and j, $< i = x_0, x_1, \cdots, x_n = j >$ and

$$L_\lambda(P(i,j)) = \sum_{i=0}^{n-1} \left(1 + \lambda \| I_{x_{i+1}} - I_{x_i} \|\right) \tag{9}$$

Then the kernel at pixel i is given by $\{j, \kappa(i,j) < r\}$, where r denotes the radius of the kernel chosen according to the level of smoothing required. An alternative to the the $\kappa(.,.)$ above is the smallest number of edges on a shortest path, denoted by $\Theta(.,.)$ as in Eq. (5). It is easy to see that

$$\Theta(i,j) = \min L_0(P(i,j)) \text{ subject to } P(i,j) \in \arg\min L_1(P(i,j)) \tag{10}$$

The results obtained by the shortest path filters are shown in Fig. 1. Observe that, even though edges are better preserved in 1(c) and 1(d), compared to gaussian filter Fig. 1(b), there is still a halo effect on the boundaries. Taking the gamma limit would reduce these effects, as shown later. See Fig. 2 (b) and (c).



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

Fig. 1. (a) Original Image, (b) Gaussian Convolution on (a), (c) Mean filter with Shortest Path kernel on (a), and (d) Shortest path filter on (a)

Dictionary Ordering, Pass Values and Watershed Cuts Recall that we are interested in calculating collaborative weights for every pair of pixels, and have concluded that spatial distances alone are not sufficient. Thus for a given pair of pixels, one has to consider the number of edges on a path between them. However, there exists several paths and the choice dictates the filtering method. The minimum of the l^∞ norm over all the paths between a given pair of pixels is called *pass value* of the pair. Thus, one of the choices is to consider a path with its l^∞ norm equal to the pass value. To the best of our knowledge, this measure was first suggested for adaptive filtering in [15]. The number of edges on one such path might differ from the other creating an ambiguity on which among them to choose. On the other hand, a path with smallest dictionary ordering between a given pair of pixels would also have its l^∞ norm equal to pass value. Moreover, any such path has the same number of edges. The dictionary ordering thus reduces the ambiguity compared to the filters which choose a path with respect to any other norm.

Pass values and shortest paths have also been used in the context of seeded segmentation [5][6]. The authors in [4][5] have proved that given a set of seeds, the segments generated by any watershed cut are also the segments generated by minimum spanning forest cuts relative to the seeds and vice versa. Further, they have proved that segments corresponding to watershed cuts have the optimality property that they are connected components in some shortest path forest (SPF) relative to the seeds. The converse does not hold true and the authors thus have provided one way of choosing 'good' SPF cuts.

Now, if a non-seed pixel admits multiple shortest paths to different seeds, to which seed would the pixel be associated? An obvious answer would be the path which has lowest number of l^∞ norm weighted edges. In case of a tie on the number of such edges, the next choice would be to consider the second highest weight, and so on. This exactly reflects the definition of the dictionary ordering in Sec 3. Thus, in the segmentation aspect, the dictionary ordering provides another way of selecting a SPF cut among all the SPF cuts.

3.2 Gamma Limit of Shortest Path Filters

Recall that $\mathcal{G}^{(p)} = (V, E, W^{(p)})$ denotes the exponentiated graph. Let i and j denote two pixels in the image. Let $\Theta^{(p)}(i,j)$ denote the smallest number of edges among all the shortest paths between i and j in the graph $\mathcal{G}^{(p)}$. We say that a path $P(i,j)$ is a *shortest path with respect to dictionary ordering* if for any other path $Q(i,j)$ we have $P \leq Q$, where the comparison is based on the dictionary ordering defined in eqn. 4. Accordingly we define $\Delta(i,j)$ as the number of edges on a shortest path with respect to the dictionary ordering. Note that $\Delta(i,j)$ is well defined since if for any two paths, P and Q if $P \leq Q$ and $Q \leq P$, then $p_i = q_i$ for all i and hence both the paths have same number of edges. The following is the main theorem of this paper.

Theorem 1. *As $p \to \infty$, we have $\Theta^{(p)}(i,j) \to \Delta(i,j)$ for all pairs i and j. Moreover, any shortest path with respect to the dictionary ordering lies on a MST.*

The above theorem states that the shortest paths converge to the shortest paths with respect to dictionary ordering in the limit. An important consequence of the above theorem is that one can now calculate the $\Gamma-$limit of the shortest path filters as described in the following proposition. The proof of the proposition follows from theorem 1.

Proposition 1. *As $p \to \infty$ the shortest path filter converges to*

$$S_i = \sum_j \frac{exp\left(-\frac{\Delta(i,j)}{\sigma}\right)}{\sum_k exp\left(-\frac{\Delta(i,k)}{\sigma}\right)} I_j \tag{11}$$

We refer to the above limit as *Power Tree Filter* (PTF). As we shall soon see, calculation of the above filter is computationally expensive and an approximation is desired.

Recall that from Eqs. (2) and (3) we have $D(i,j)$ that denotes the path length on a minimum spanning tree, which can be seen as an approximation for $\Delta(i,j)$. This provides a justification for the MST filter - *MST filter is nothing but an approximation to the gamma limit of the shortest path filters.*

3.3 Calculation of the Gamma limit

In this part, we provide an algorithm to calculate the $\Gamma-$limit and give another approximation. Union minimum spanning tree (UMST) is defined as the subgraph generated by the edges of all the minimum spanning trees. Theorem 1 states that any shortest path with respect to dictionary ordering lies on an MST, and hence on UMST. Thus we need to only concern ourselves with UMST instead of the whole graph. Thanks to the characterization of UMST in [12] which is stated below, we develop an efficient algorithm based on the result.

Theorem 2. *Let $\mathcal{G} = (V, E, W)$ be an edge weighted graph. Let $\mathcal{G}_{\leq w}$ denote the graph with the vertex set V and all the edges e_{ij} whose weight $w_{ij} \leq w$, where $1 \leq j \leq k$. Let $\mathcal{G}_{umst}$ denote the UMST of $\mathcal{G}$. Then an edge e with weight $w(e)$ belongs to the UMST iff either $w(e) = w_1$ or if the edge e joins two connected components in $\mathcal{G}_{\leq w(e)}$.*

Theorem 2 is equivalent to the *cut property* of the MST[14]. With a suitable choice of the data structure, the above theorem gives an $\mathcal{O}(|E|)$ (which in our case is $\mathcal{O}(|V|)$) algorithm to calculate the UMST.

Note that the calculation of $\Delta(i, j)$ for each pair (i, j) takes $\mathcal{O}(|V|)$ and hence the crudest algorithm takes $\mathcal{O}(|V|^3)$ time. Using a slightly different version of Floyd-Warshall [8] allows us to calculate the $\Delta(i, j)$ for all pairs in $\mathcal{O}(|V|^2)$, which is still expensive. Thus, one needs an approximation to calculate the Power Tree Filter.

Instead of considering UMST, if one considers a single MST as in MST filter described above, the calculation can be made efficient using top-down and bottom-up calculations in $\mathcal{O}(|V|)$ time [19]. The following steps give an overview of the bottom-up algorithm as described in [19]. 1) Pick a root node i. 2) Starting from the root nodes, update the values recursively using

$$S_p = I_p + \sum_{q \in \text{children of p}} exp(\frac{-1}{\sigma}) S_q$$

At the end of these calculations we are ensured to get the filtered value at the root node i. The top-down procedure is similar. For further details see [19]. This justifies taking an MST as an approximation to UMST. A different method is described below, based on a few observations.

Proposition 2. *Given a pixel i, there exists an MST, M, such that the shortest path with respect to dictionary ordering between pixels i and any other pixel j is in M.*

The MST in proposition 2 is referred to as *Power MST*. If one can calculate the MST as in Proposition 2 for each pixel i, the bottom-up procedure as in [19] can be followed for each pixel separately and we get the PTF filtered value at that pixel. This however is still $\mathcal{O}(|V|^2)$. The algorithm to calculate Power MST is given in Alg. 1. $dist(e, i, M)$ denotes the distance of the edge e from the vertex i on the MST M, and is defined as the number of edges on the path between i and the farthest node in e (from i) on the MST M.

Note that we can rewrite Eq. (11) as

$$S_i = \frac{1}{NC} \sum_l exp(-\frac{l}{\sigma}) \sum_{j, \Delta(i,j)=l} I_j \ , \tag{12}$$

where NC is the normalizing constant. In the above equation observe that the exponential term quickly converges to 0 and hence one can approximate the

Algorithm 1 Approximate Algorithm to calculate MST at pixel

Input: An UMST of the graph, depth d and pixel i
Output: Power MST

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

Algorithm 2 Approximate Algorithm to calculate PTF

Input: A 4-adjancecy graph of the image I, $\mathcal{G}$
Output: Filtered image S.

Construct the UMST of $\mathcal{G}$
for all pixels i **do**
  Construct Power MST M_i, using algorithm 1.
  Use *bottom-up* to get the filtered value S_i.
end for

above expression by

$$S_i \approx \frac{1}{NC} \sum_{l=1}^{D} exp(-\frac{l}{\sigma}) \sum_{j, \Delta(i,j)=l} I_j \ , \tag{13}$$

where D is a parameter indicating a fixed depth. This approximation implies that, theoretically the calculation of the PTF filtered value at each pixel reduces to constant time and hence we get an approximate algorithm which is $\mathcal{O}(|V|)$. The pseudocode is given in algorithm 2.

4 Analysis and Experiments

4.1 Comparison with MST filter

Before proceeding to the analysis of the Power Tree filter, we provide the results of the Power Tree filter and compare with the MST filter. Note that in [1] the MST filter results were post-processed with the bilateral filter as well. We provide a visual comparison of the Power Tree filter and MST filter in Fig. 2 with and without the usage of bilateral filter as a post-processing step. Note that the 'leak' issue is reduced in the Power Tree filter compared to the MST filter, as in

Figs. 2(b) and (d). Also in Fig 2 (q), (r), (s) and (t) the black border is better preserved in the Power Tree filter compared to the MST filter, irrespective of the usage of bilateral filtering. We further discuss the heuristic reason for this behaviour in the next section.

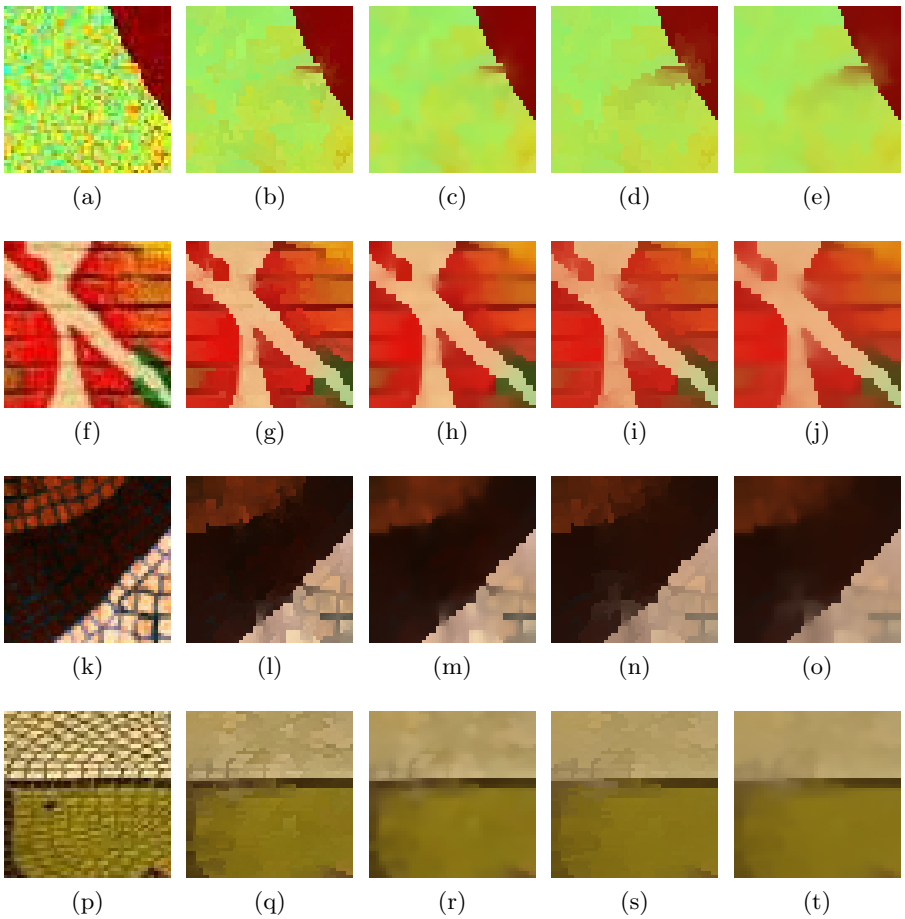
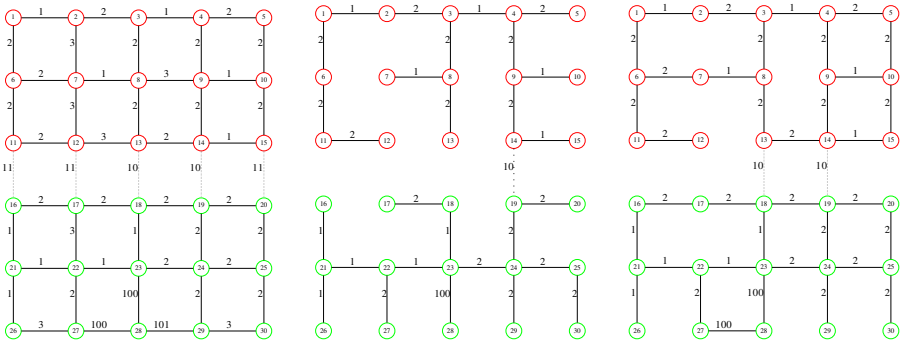


Fig. 2. (a), (f), (k), (p) are the original noisy images. (b), (g), (l), (q) are obtained by Power Tree filtering. (c), (h), (m), (r) are obtained by Power Tree filtering followed by bilateral filtering. (d), (i), (n), (s) are obtained by MST filtering. (e), (j), (o), (t) are obtained by MST filtering followed by bilateral filtering.

In general there would be a trade-off between the amount of smoothing and the level of edge-preserving in an image. This also can be seen from various examples in Fig. 2. The parameters used to generate Fig. 2 are $\sigma = 10$ for both Power Tree filter and MST filter, and $depth = 15$ for the Power Tree filter.

4.2 Analysis of the Leak Problem



(a) (b) (c)

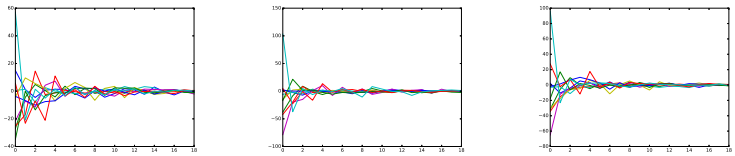
Fig. 3. (a) 4-adjacency graph of a synthetic image, (b) An MST obtained from (a), and (c) UMST obtained from (a)

Consider a synthetic grey scale image which has two objects (whose pixels colored in red and green for easy identification, see figure 3(a)). Assume that the pixel values within each of the objects are similar and different across objects. The pixels are numbered 1 to 30 and the edge weights are displayed on the edges. The edges corresponding to object boundaries are represented as dotted edges. In order to illustrate that UMST filter yields better results, it should perform at least as good as tree filter for - removing noise at pixel numbered 28 and reducing the leak at object boundaries say at pixel numbered 13 and 14.

Figures 3(b) and 3(c) denote a MST and the UMST respectively. 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 $13-14$.

4.3 Error Analysis of the approximation

In Eq. 13, we have considered an approximation of the PTF. In this section, we analyze the appropriateness of the approximation. In particular we empirically show that as the depth, D, increases the filtered value of the pixel remains constant. In Fig. 4, several pixels are randomly chosen from an image and the filtered values with varying depths are calculated. Then for each pixel, the first



(a) (b) (c)

Fig. 4. First difference of the Power Tree filtered value as a function of depth for the 3 bands of RGB image. Each curve corresponds to a randomly chosen pixel in the image. Note that the pixel values stabilize at a depth of 15.

differences of the filtered values with respect to the depth are plotted in Fig. 4, for all the 3 bands. Observe that the first differences tend to 0, and hence the filtered values are stabilized. From the figure, at a depth of 15, the filtered values are more or less stable and hence this is taken for all the experiments in Fig. 2.

5 Conclusion and Future Work

We have shown that MST filter can be seen as an approximation to the $\Gamma-$limit of shortest path filters, referred to as Power Tree filter, which provides theoretical links between MST filter and geodesic based filters. Also, we have provided an alternate approximation to Power Tree filter and have shown that it yields better results. We further validated this approximation empirically. The Power Tree filter being closely related to MST filter is expected to yield good results in applications such as stereo-matching, optical-flow, image-abstraction, texture removal and editing, depth up sampling etc.

Note that although algorithm 2 is theoretically linear time, the constant is quite high and thus one might be able to find faster/better approximations to the PTF. For example, algorithm 2 can be parallelized since each pixel is processed independently of the other. This is a topic of further research.

6 Acknowledgments

Sravan Danda and Aditya Challa are thankful for the financial support provided by the Indian Statistical Institute. 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 partial support received from ANR-15-CE40-0006 CoMeDiC and ANR-14-CE27-0001 GRAPHSIP research grants.

References

1. Bao, L., Song, Y., Yang, Q., Yuan, H., Wang, G.: Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. IEEE TIP 23(2), 555–569 (2014)

2. Chang, J.H.R., Wang, Y.C.F.: Propagated image filtering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10–18. IEEE (2015)
3. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watershed: A unifying graph-based optimization framework. IEEE PAMI 33(7), 1384–1399 (2011)
4. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Minimum spanning forests and the drop of water principle. IEEE PAMI 31(8), 1362–1374 (2009)
5. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Thinnings, shortest path forests, and topological watersheds. IEEE PAMI 32(5), 925–939 (2010)
6. Falcao, A.X., Stolfi, J., de Alencar Lotufo, R.: The image foresting transform: Theory, algorithms, and applications. IEEE PAMI 26(1), 19 (2004)
7. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. In: ACM Transactions on Graphics (TOG). vol. 27, p. 67. ACM (2008)
8. Floyd, R.W.: Algorithm 97: shortest path. Communications of the ACM 5(6), 345 (1962)
9. Grady, L.: Random walks for image segmentation. IEEE PAMI 28(11), 1768–1783 (2006)
10. He, K., Sun, J., Tang, X.: Guided image filtering. In: Computer Vision–ECCV 2010, pp. 1–14. Springer (2010)
11. Lerallut, R., Decencière, É., Meyer, F.: Image filtering using morphological amoebas. Image and Vision Computing 25(4), 395–404 (2007)
12. Najman, L.: Extending the PowerWatershed framework thanks to Γ-convergence. Tech. rep., Université Paris-Est, LIGM, ESIEE Paris (2017), https://hal-upec-upem.archives-ouvertes.fr/hal-01428875
13. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007. pp. 1–8. IEEE (2007)
14. StackOverFlow: Cut property. http://stackoverflow.com/questions/3327708/minimum-spanning-tree-what-exactly-is-the-cut-property (Accessed: 2017-01-05)
15. Stawiaski, J., Meyer, F.: Minimum spanning tree adaptive image filtering. In: 2009 16th IEEE ICIP. pp. 2245–2248. IEEE (2009)
16. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision, 1998. ICCV 1998. pp. 839–846. IEEE (1998)
17. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via l 0 gradient minimization. In: ACM Transactions on Graphics (TOG). vol. 30, p. 174. ACM (2011)
18. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM Transactions on Graphics (TOG) 31(6), 139 (2012)
19. Yang, Q.: Stereo matching using tree filtering. IEEE PAMI 37(4), 834–846 (2015)