# Accelerating the Distribution Estimation for the Weighted Median/Mode Filters

Lu Sheng, King Ngi Ngan, Tak-Wai Hui

Department of Electronic Engineering, the Chinese University of Hong Kong

**Abstract.** Various image filters for applications in the area of computer vision require the properties of the local statistics of the input image, which are always defined by the local distribution or histogram. But the huge expense of computing the distribution hampers the popularity of these filters in real-time or interactive-rate systems. In this paper, we present an efficient and practical method to estimate the local weighted distribution for the weighted median/mode filters based on the kernel density estimation with a new separable kernel defined by a weighted combinations of a series of probabilistic generative models. It reduces the large number of filtering operations in previous constant time algorithms [1, 2] to a small amount, which is also adaptive to the structure of the input image. The proposed accelerated weighted median/mode filters are effective and efficient for a variety of applications, which have comparable performance against the current state-of-the-art counterparts and cost only a fraction of their execution time.

## 1 Introduction

A variety of popular image filters in computer vision are related to the local statistics of the input image. For example, the median filter outputs the point that reaches half of the local cumulative distribution [3, 4, 1]. The weighted mode filter [5–7] tries to find the global mode of the local distribution. Not only that, the widely popular bilateral filter [8], can be expressed as the mean of the local distribution that is estimated by a Gaussian kernel density estimator [9]. Provided a *guidance* feature map (*e.g.*, image intensity, patch and *etc.*), the weighted local distribution can be modified to jointly reflect the statistics of both the input image and the feature map, which in addition contributes to several kinds of structure- or style-transfer applications, like depth or disparity refinement in the stereo matching [5, 1] and joint filtering [10].

Not explicitly estimating the local distribution, there are a certain number of approaches that are designed for accelerating the bilateral filter or similar weighted average filter, such as the domain transform filter [11], adaptive manifolds filter [12] and the guided filter [13]. However, efficient methods for immediate estimation of the local distributions need further attention because many applications require direct operations on these distributions. Although the brute-force implementation is still adopted in many computer vision systems, its high complexity limits its popularity and hampers real-time systems and applications.

Constant time algorithms for the estimation of the local distributions (or histograms) have been proposed in the literature. For instance, the constant time weighted median filter [1] and the smoothed local histogram filters [2]. The complexity of these methods relies on the *number of bins* to generate the histograms as well as the complexity of the *filtering operation* that calculates the value of each bin. Even though the complexity of filtering operations have been reported as $\mathcal{O}(1)$ in the literature, an 8-bit single channel gray-scale image usually needs 256 bins to produce a sufficiently accurate result, not to mention continuous or high-precision images.

Related to but different from these methods, in this paper we proposed a novel distribution estimation method for the sake of efficiency to accelerate various image filters. It is based on the kernel density estimation with a new separable kernel defined by a weighted combination of a series of probabilistic generative models. The resultant distribution has a much reduced number of filtering operations which are also independent of the values of the bins. The number of filtering operations is exactly the number of models used, and is usually smaller than the number of bins so as to abate the computational complexity. The required models can be the uniform quantization of the domain of the input image, or locally adaptive to the structures of the inputs. Since it is always the case that a local patch of an image can be decomposed into a limited number of distinct local structures, only a small amount of the locally adaptive models are necessary, thus the complexity is further reduced. We also accelerated the weighted mode filter and the weighted median filter by leveraging the proposed distribution estimation method. They own comparable performance in various applications but a faster speed in comparison with current state-of-art algorithms.

## 2 Related Work

Weighted average filters, like the bilateral filter [8, 10], implicitly reflect properties of the local distribution. The brute-force implementation generally suffers the issue of inefficiency. In [14] an approximated solution was proposed by formulating the bilateral filtering as a high-dimensional low-pass filter, and can be accelerated by downsampling the space-range domain. Following this idea, different data structures have been proposed afterwards to further speedup the filters [15–17, 12], in which the adaptive manifolds [12] caught our attention and inspired our research to construct the locally adaptive models. Guided filter [13] is a popular and efficient constant-time alternative. It can imitate a similar filter response as that of bilateral filter, but enforces local linear relationship between the filtering output and the guidance image. Domain transform filter [11] also produces a similar constant-time edge-preserving filter and earns real-time performance without quantization or coarsening.

Median filter might be the first image filter that explicitly applies the local histogram (a discretized distribution). Unlike the weighted median filter, which has no abundant work focusing on its acceleration, the unweighted counterpart receives several constant time solutions. One kind of these algorithms was present in the literature to lessen the histogram update complexity [3, 4]. Another version introduced by Kass and Solomon [2] drawn the isotropic filtering into the construction of a so-called smoothed local histogram, which is a special case of the kernel density estimation, and the median and mode of this histogram are thus estimated by a look-up table.

The weighted median filter as well as the weighted mode filter, however, cannot directly duplicate the success in the previous discussion, since the weights are spatially varying for each local window. Min *et. al.* [5] proposed a weighted mode filtering that adopts bilateral weights for the depth video enhancement, but it lacks an efficient implementations. The constant time weighted median filter [1] for disparity refinement is one of the most recent works that try to accelerate the local distribution construction. This method performs edge-preserving filtering to produce the probability of each bin in the local histogram. The number of bins determines the number of filtering operations applied. Thus it is less effective when hundreds of intensity levels are required, especially for the processing of the natural images.

## 3 Motivation

Estimating the probability distribution of each pixel is an essential element in various kinds of image filters like the weighted median filter [1], the weighted mode filter [5] and the bilateral filter [8] as a special case. A conventional way is to construct a weighted histogram of the target pixel in its local window, but a more flexible treatment is to exploit the kernel density estimation [7] so as to favor a smooth approximation [2].

Denote the probability distribution at pixel $\mathbf{x}$ as $h(\mathbf{x},\cdot)$ and is specifically defined as

$$h(\mathbf{x},g)=\frac{1}{\mathsf{Z}(\mathbf{x})}\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} w(\mathbf{x},\mathbf{y})\phi_{\mathbf{x}}\left(g,f_{\mathbf{y}}\right), \tag{1}$$

where $f_{\mathbf{y}}$ is the input data of pixel $\mathbf{y}$. $\mathcal{N}(\mathbf{x})$ is a local window centered at $\mathbf{x}$, and the normalized factor $\mathsf{Z}(\mathbf{x})=\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} w(\mathbf{x},\mathbf{y})$. The weight $w(\mathbf{x},\mathbf{y})$ depends on the spatial relationship and the similarity of the guidance features between $\mathbf{x}$ and $\mathbf{y}$. The kernel $\phi_{\mathbf{x}}(g,f_{\mathbf{y}})$ varies in different applications. For discrete signals, a common kernel is the Kronecker delta function $\delta(\cdot)$, thus $h(\mathbf{x},\cdot)$ becomes a weighted histogram [1]. An alternative common choice is the Gaussian kernel.

The approximated probability distribution immediately gets involved in the weighted median filter or the weighted mode filter since it replaces the value of a pixel by the median or the global mode of $h(\mathbf{x},\cdot)$. The median is usually estimated by tracing the cumulative distribution [2]:

$$\mathcal{C}(\mathbf{x},\hat{g})=\int_{-\infty}^{\hat{g}} h(\mathbf{x},g)dg=\frac{1}{\mathsf{Z}(\mathbf{x})}\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} w(\mathbf{x},\mathbf{y})\cdot\int_{-\infty}^{\hat{g}}\phi_{\mathbf{x}}\left(g,f_{\mathbf{y}}\right)dg \tag{2}$$

until it meets 0.5. Because it involves a high dimensional filtering operation in estimating $\mathcal{C}(\mathbf{x},\hat{g})$ at each $\hat{g}$, too many samples of $\hat{g}$ will bring about heavy computational cost. On the other hand, typical ways to find the mode are the fixed-point iteration [6] or sampling by a look-up table and interpolation [2]. The key element in either method is the gradient of $h(\mathbf{x}, g)$ as

$$\frac{\partial h(\mathbf{x}, g)}{\partial g}\Big|_{g=\hat{g}} = \frac{1}{\mathsf{Z}(\mathbf{x})} \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} w(\mathbf{x}, \mathbf{y}) \cdot \frac{\partial \phi_{\mathbf{x}}\left(g, f_{\mathbf{y}}\right)}{\partial g}\Big|_{g=\hat{g}}, \tag{3}$$

which is also the output after filtering. Similar problem occurs since the number of filtering operations depends on the number of iterations to converge or the sampling density of the look-up table.

To eliminate this issue, in the following sections we define a novel separable kernel as a weighted combination of a series of probabilistic generative models to decrease the number of filtering operations required to represent the distribution, and exploit the constant time filters [13, 11] to reduce the complexity of the filtering operation.

## 4 Accelerating the Distribution Estimation

In this paper, we propose a novel approach to approximate the probability distribution by defining a new kernel based on a series of probabilistic generative models, which can be factorized explicitly so as to extract the filtering operations in advance before the distribution construction. With the proposed kernel, we introduce the accelerated versions of the weighted mode filter and the weighted median filter. We will show it later that they have excellent performance in terms of both quality and efficiency in various applications.

### 4.1 Kernel Definition

Assume the input image is modeled by several (*e.g.*, $L$) models throughout the whole pixel domain, each of which is governed by a distribution as $p(\eta_{\mathbf{x}}|l), l \in \mathcal{L} = \{1, 2 \ldots, L\}$ at each pixel $\mathbf{x}$. These models actually act as prior knowledge to represent *distinct local structures* in the input image. Two pixels $\mathbf{x}$ and $\mathbf{y}$ are similar if they both have high probabilities to agree with the $l^{\text{th}}$ model (see Fig. 1) as the following kernel:

$$\begin{aligned}\kappa^l(f_{\mathbf{x}}, f_{\mathbf{y}}) &= p_{\mathbf{x}}(f_{\mathbf{x}}|l) p_{\mathbf{y}}(f_{\mathbf{y}}|l) & (4)\\ &= \int_{\eta_{\mathbf{x}}\in\mathcal{H}_{\mathbf{x}}} p(f_{\mathbf{x}}|\eta_{\mathbf{x}}) p(\eta_{\mathbf{x}}|l) d\eta_{\mathbf{x}} \cdot \int_{\eta_{\mathbf{y}}\in\mathcal{H}_{\mathbf{y}}} p(f_{\mathbf{y}}|\eta_{\mathbf{y}}) p(\eta_{\mathbf{y}}|l) d\eta_{\mathbf{y}}, & (5)\end{aligned}$$

where $p_{\mathbf{x}}(f_{\mathbf{x}}|\eta_{\mathbf{x}})$ is the data likelihood. $\mathcal{H}_{\mathbf{x}}$ and $\mathcal{H}_{\mathbf{y}}$ are the domains of $\eta_{\mathbf{x}}$ and $\eta_{\mathbf{y}}$, respectively. When all the $L$ models are available, the overall kernel can be further defined as their weighted combination:

$$\kappa(f_{\mathbf{x}}, f_{\mathbf{y}}) = \sum_{l=1}^{L} \kappa^l(f_{\mathbf{x}}, f_{\mathbf{y}}) p_{\mathbf{x},\mathbf{y}}(l) = \sum_{l=1}^{L} p_{\mathbf{x}}(f_{\mathbf{x}}|l) p_{\mathbf{y}}(f_{\mathbf{y}}|l) p_{\mathbf{x},\mathbf{y}}(l), \tag{6}$$

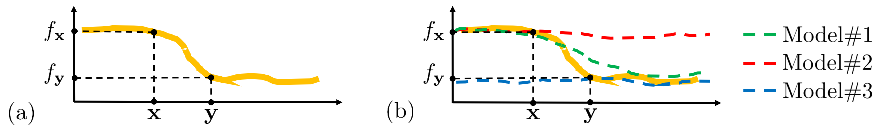

**Fig. 1.** Illustration of the proposed kernel. (a) shows a 1D signal and two pixels $\mathbf{x}$ and $\mathbf{y}$. (b) represents the construction of $\kappa(f_\mathbf{x}, f_\mathbf{y})$, where the mean values of three models are shown in three different colors. It measures the similarity of $f_\mathbf{x}$ and $f_\mathbf{y}$ by evaluating the summation of the joint likelihood of them *w.r.t.* each model.

where $p_{\mathbf{x},\mathbf{y}}(l)$ is the compatibility prior that measures the similarity between $\mathbf{x}$ and $\mathbf{y}$ on the $l^{\text{th}}$ model. This kernel is valid since it is an inner product in the $L$-dimensional feature space. What's more, it is able to reliably approximate some popular kernels like Gaussian kernel [12] or Kronecker delta kernel [1].

### 4.2 Probability Distribution Approximation

The approximated distribution can be written similarly as Eq. (1) by replacing $\phi_\mathbf{x}(g, f_\mathbf{x})$ with the proposed kernel as

$$\tilde{h}(\mathbf{x}, g) \propto \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} w(\mathbf{x},\mathbf{y}) \sum_{l=1}^{L} p_\mathbf{x}(g|l) p_\mathbf{y}(f_\mathbf{y}|l) p_{\mathbf{x},\mathbf{y}}(l) = \sum_{l=1}^{L} p_\mathbf{x}(g|l) \cdot \psi_\mathbf{x}(l). \quad (7)$$

The filtering operation $\psi_\mathbf{x}(l) = \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} w(\mathbf{x},\mathbf{y}) p_\mathbf{y}(f_\mathbf{y}|l) p_{\mathbf{x},\mathbf{y}}(l)$ is independent of $g$, and thus the approximated distribution becomes a mixture of $L$ densities. Instead of immediately filtering $\phi_\mathbf{x}(g, f_\mathbf{y})$ for each $g$ (*cf.*, Eq. (1)) to obtain $h(\mathbf{x}, g)$, the proposed method can precompute $\psi_\mathbf{x}(l)$ by merely $L$ filtering operations in total and then estimate $\tilde{h}(\mathbf{x}, g)$ provided the priors $p(g|l)$. The proposed kernel approximates the distribution by extracting the filtering operations independent of $g$ and therefore reduces the complexity of the distribution construction.

The cumulative distribution is hence $\tilde{\mathcal{C}}(\mathbf{x}, \hat{g}) \propto \sum_{l=1}^{L} \psi_\mathbf{x}(l) \int_{-\infty}^{\hat{g}} p(g|l) dg$ and the gradient $\frac{\partial \tilde{h}(\mathbf{x},g)}{\partial g}|_{g=\hat{g}} \propto \sum_{l=1}^{L} \psi_\mathbf{x}(l) \frac{\partial p(g|l)}{\partial g}|_{g=\hat{g}}$, both of which do not contain additional filtering operations except those for $\psi_\mathbf{x}(l)$, and thus have the potential to accelerate the weighted median/mode filters.

**Relationship with the Constant Time Weighted Median Filter [1] (CT-Median)** Let the $L$ models be equally quantized levels $\mu^l, l \in \mathcal{L}$ of the intensity space, and denote $p(\eta_\mathbf{x}|l) = \delta(\eta_\mathbf{x} - \mu^l), p(f_\mathbf{x}|\eta_\mathbf{x}) = \delta(f_\mathbf{x} - \eta_\mathbf{x}), p_{\mathbf{x},\mathbf{y}}(l) = 1/L$. We have the distribution as $\tilde{h}(\mathbf{x}, g) \propto \sum_{l=1}^{L} \delta(g - \mu^l) \cdot \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} w(\mathbf{x},\mathbf{y}) \delta(f_\mathbf{y} - \mu^l)$, which is exact the form introduced in CT-median.

### 4.3 Accelerated Weighted Median/Mode Filters

In this section, we propose the accelerated versions of the weighted median/mode filters based on the kernel discussed previously. In particular, we apply the Gaus-

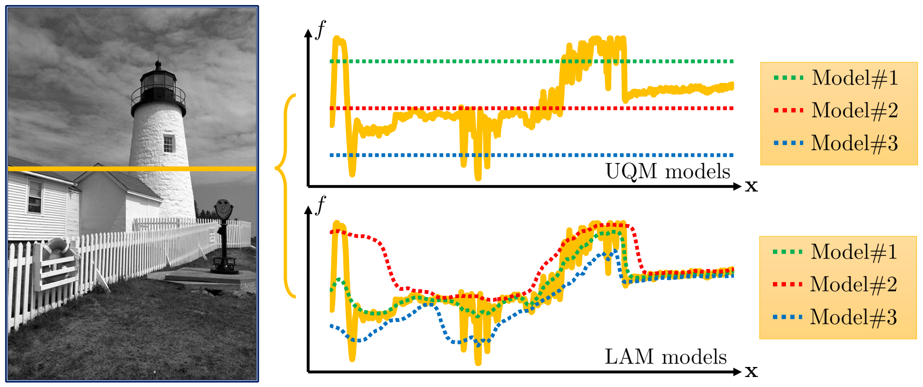

**Fig. 2.** Locally adaptive models (LAM) *v.s.* uniformly quantized models (UQM). A 1D signal is extracted from a gray-scale image shown in the left column and marked by orange. Both the LAM and UQM models ($L = 3$) are exploited to represent the signal, which are shown in the right column. The top row is by UQM models, the bottom one is by LAM models. The LAM models are adaptive to the local structures and own superior performance on representing the signal with limited number of models.

sian model to define the probabilities for its efficiency in various image processing applications.

**Kernel Definition** The *first* task is to define the $L$ models that are suitable as the priors to represent the input image.

*Case*-I: *Uniformly Quantized Models* (UQM). A simple strategy is just to equally quantize the domain $f$, the mean of each model represents a quantization level $\mu^l$ and the diagonal elements in $\mathbf{\Sigma}^l$ is set as the square of half of the quantization interval. For a multi-dimensional image, each channel shares the same process. Specifically, $\mu^l_{\mathbf{x}} = \mu^l, \mathbf{\Sigma}^l_{\mathbf{x}} = \mathbf{\Sigma}^l$ at the $l^{\text{th}}$ model for all $\mathbf{x}$. It can well represent cartoon style images and disparity maps from frontal parallel stereos. However, more quantization levels are required to present a local complex structure under a sufficient accuracy, as shown in Fig. 2.

*Case*-II: *Locally Adaptive Models* (LAM). Locally adaptive models ought to be a superior idea since they tend to describe the local structures by fewer models. The idea behind it is that we assume a Gaussian mixture model in any local patch. Each model actually represents a local mean estimator. Therefore, we just need the number of *models* is a few more than the number of *modes* in the local distribution. For example, a natural image shown in Fig. 2 can be well represented by the LAM models. On the contrary, the UQM models cannot fit the local distribution if its number is insufficient.

The popular EM algorithm [18] is abandoned for the training of the LAM models due to its high complexity and instability to ensure a good estimation. In this paper, we exploit an alternative and more efficient way to train the required models. Similarly as [12], we also use a *hierarchical segmentation* approach to iteratively separate pixels from distinct structures, which act as local clusters, into different models. We set the segments as $\mathcal{S}_l, l \in \mathcal{L}$. This method involves simple low-pass filtering and fast PCA operations [12], thus is efficient in implementation. The mean and variance of the pixel $\mathbf{x}$ for the $l^{\text{th}}$ model are

$$\mu_{\mathbf{x}}^l = \frac{1}{\mathsf{W}_{\mathbf{x}}^l} \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \theta_{\mathbf{y}}^l f_{\mathbf{y}},\ \mathbf{\Sigma}_{\mathbf{x}}^l = \frac{1}{\mathsf{W}_{\mathbf{x}}^l} \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \theta_{\mathbf{y}}^l f_{\mathbf{y}} f_{\mathbf{y}}^\top - \mu_{\mathbf{x}}^l {\mu_{\mathbf{x}}^l}^\top, \tag{8}$$

where $\theta_{\mathbf{y}} = 1_{[\mathbf{y}\in\mathcal{S}_l]}$ means the mask indicating pixels inside $\mathcal{S}_l$. $1_{[\cdot]}$ is the indicator function that equals to 1 when the input argument is true. The neighborhood $\mathcal{N}(\mathbf{x})$ is set as the same local window as Eq. (7). $\mathsf{W}_{\mathbf{x}}^l = \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \theta_{\mathbf{y}}^l$ is the normalization factor.

Therefore, the prior probability for the $l^{\text{th}}$ model is $p(\eta_{\mathbf{x}}|l) = N(\eta_{\mathbf{x}}|\mu_{\mathbf{x}}^l, \mathbf{\Sigma}_{\mathbf{x}}^l)$. Assume the data likelihood $p(f_{\mathbf{x}}|\eta_{\mathbf{x}}) = N(f_{\mathbf{x}}|\eta_{\mathbf{x}}, \mathbf{\Sigma}_n)$, where $\mathbf{\Sigma}_n$ denotes the noise variance. Thus we have

$$\kappa^l(f_{\mathbf{x}}, f_{\mathbf{y}}) = N(f_{\mathbf{x}}|\mu_{\mathbf{x}}^l, \mathbf{\Sigma}_n + \mathbf{\Sigma}_{\mathbf{x}}^l) \cdot N(f_{\mathbf{y}}|\mu_{\mathbf{y}}^l, \mathbf{\Sigma}_n + \mathbf{\Sigma}_{\mathbf{y}}^l). \tag{9}$$

Given a prior tells the compatibility between pixel $\mathbf{x}$ and $\mathbf{y}$ for the $l^{\text{th}}$ model as $p_{\mathbf{x},\mathbf{y}}(l) = \exp(-\frac{1}{2}(\mu_{\mathbf{x}}^l - \mu_{\mathbf{y}}^l)^\top \mathbf{\Sigma}_n^{-1}(\mu_{\mathbf{x}}^l - \mu_{\mathbf{y}}^l))$, the kernel $\kappa(f_{\mathbf{x}}, f_{\mathbf{y}})$ is therefore defined accordingly.

The proposed kernel otherwise needs two parameters $\mathbf{\Sigma}_n$ and $L$. For a highly complex image, $L$ should be increased to fit the local structure to a large extent. Large $\mathbf{\Sigma}_n$ brings about smoother results but on the contrary, the necessary number of models can be reduced as well.

**Probability Distribution Approximation** Based on the proposed kernel, the approximated probability distribution to each pixel $\mathbf{x}$ is

$$\tilde{h}(\mathbf{x}, g) = \frac{1}{\tilde{\mathsf{Z}}(\mathbf{x})} \sum_{l=1}^{L} N(g|\mu_{\mathbf{x}}^l, \mathbf{\Sigma}_n + \mathbf{\Sigma}_{\mathbf{x}}^l)\psi_{\mathbf{x}}(l), \tag{10}$$

where $\psi_{\mathbf{x}}(l) = \sum_{\mathbf{y}\in N(\mathbf{x})} w(\mathbf{x},\mathbf{y})\mathcal{N}(f_{\mathbf{y}}|\mu_{\mathbf{y}}^l, \mathbf{\Sigma}_n + \mathbf{\Sigma}_{\mathbf{y}}^l) p_{\mathbf{x},\mathbf{y}}(l)$ and $\tilde{\mathsf{Z}}(\mathbf{x}) = \sum_{l=1}^{L} \psi_{\mathbf{x}}(l)$.

The *second* step of the weighted median/mode filters is to estimate $\psi_{\mathbf{x}}(l), l \in \mathcal{L}$ by filtering $N(f_{\mathbf{y}}|\mu_{\mathbf{y}}^l, \mathbf{\Sigma}_n + \mathbf{\Sigma}_{\mathbf{y}}^l)$ characterized by the properties of $w(\mathbf{x},\mathbf{y}) \times p_{\mathbf{x},\mathbf{y}}(l)$. This weight defines a joint filtering with the guidance of the guided feature map and the estimated models. Here we denote its parameters as $\boldsymbol{\omega}$. In this paper, we choose two kinds of filters: Guided filter (GF) [13] and Domain-transform filter (DF) [11]. They both have $\mathcal{O}(1)$ complexity and approximate the bilateral weight. GF has better performance on transferring local structures from the guided feature map to the target image while DF is natural to process higher dimensional images. Different applications exploit different weights.

The overall algorithm about the distribution approximation acceleration is summarized in Algorithm 1. The parameter setup refers to Sec. 5.1.

**Algorithm 1:** Distribution Approximation Acceleration

```
   Input  : Input image $\mathbf{F}_i$, guided image $\mathbf{F}_g$, parameter set $\{L^{\text{th}}, r, \sigma_n, \boldsymbol{\omega}\}$;
   Output: Approximated distribution $\tilde{h}(\mathbf{x}, g)$;
   // 1. model generation
1  if model_type is LAM then
2      $\{\mathcal{S}_l |\ l \in \mathcal{L}\} \leftarrow$ hierarchical segmentation [12] of $\mathbf{F}_i$ given $L^{\text{th}}$ and $r, \sigma_n$;
3      for $l \leftarrow 1$ to $L$ do
4          $\theta^l_{\mathbf{y}} = 1_{[\mathbf{y} \in \mathcal{S}_l]},\ \mathsf{W}^l_{\mathbf{x}} = \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \theta^l_{\mathbf{y}}$;
5          $\mu^l_{\mathbf{x}} \leftarrow \frac{1}{\mathsf{W}^l_{\mathbf{x}}} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \theta^l_{\mathbf{y}} f_{\mathbf{y}},\ \boldsymbol{\Sigma}^l_{\mathbf{x}} = \frac{1}{\mathsf{W}^l_{\mathbf{x}}} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \theta^l_{\mathbf{y}} f_{\mathbf{y}} f_{\mathbf{y}}^\top - \mu^l_{\mathbf{x}} {\mu^l_{\mathbf{x}}}^\top$ ;
6      $\mathcal{M}^l \leftarrow \{\mu^l_{\mathbf{x}}, \boldsymbol{\Sigma}^l_{\mathbf{x}} |\ \forall \mathbf{x}\}, l \in \mathcal{L}$ // model parameters
7  else
8      $\{\mu^l, \boldsymbol{\Sigma}^l |\ l \in \mathcal{L}\} \leftarrow$ quantize the image domain of $\mathbf{F}_i$ uniformly, given $L^{\text{th}}$ ;
9      $\mathcal{M}^l \leftarrow \{\mu^l_{\mathbf{x}} = \mu^l, \boldsymbol{\Sigma}^l_{\mathbf{x}} = \boldsymbol{\Sigma}^l |\ \forall \mathbf{x}\}, l \in \mathcal{L}$ // model parameters
   // 2. distribution approximation
10 $\psi_{\mathbf{x}}(l) \leftarrow \sum_{\mathbf{y} \in N(\mathbf{x})} w(\mathbf{x}, \mathbf{y}) \mathcal{N}(f_{\mathbf{y}} | \mu^l_{\mathbf{y}}, \boldsymbol{\Sigma}_n + \boldsymbol{\Sigma}^l_{\mathbf{y}}) p_{\mathbf{x},\mathbf{y}}(l),\ \psi_{\mathbf{x}}(l) \leftarrow \psi_{\mathbf{x}}(l) / \sum_{l=1}^L \psi_{\mathbf{x}}(l)$;
11 $\tilde{h}(\mathbf{x}, g) \leftarrow \sum_{l=1}^L N\left(g | \mu^l_{\mathbf{x}}, \sigma_n^2 \mathbf{I}^d + \boldsymbol{\Sigma}^l_{\mathbf{x}}\right) \psi_{\mathbf{x}}(l)$;
```

**Weighted Median Filter** The weighted median filter wants to find the median value throughout the given probability distribution. Since the resultant distribution is actually a mixture of Gaussian models, an accelerated method is proposed by estimating the cumulative probability $\tilde{\mathcal{C}}(\mathbf{x}, \mu^l_{\mathbf{x}})$ at the mean value $\mu^l_{\mathbf{x}}$ of each model. The median value is approximated by interpolating two adjacent cumulative probabilities $\tilde{\mathcal{C}}(\mathbf{x}, \mu^k_{\mathbf{x}})$ and $\tilde{\mathcal{C}}(\mathbf{x}, \mu^{k+1}_{\mathbf{x}})$, where $\tilde{\mathcal{C}}(\mathbf{x}, \mu^k_{\mathbf{x}}) \leq 0.5$ and $\tilde{\mathcal{C}}(\mathbf{x}, \mu^{k+1}_{\mathbf{x}}) \geq 0.5$. In detail,

$$g^{\text{med}}_{\mathbf{x}} \approx \frac{0.5 - \tilde{\mathcal{C}}(\mathbf{x}, \mu^k_{\mathbf{x}})}{\tilde{\mathcal{C}}(\mathbf{x}, \mu^{k+1}_{\mathbf{x}}) - \tilde{\mathcal{C}}(\mathbf{x}, \mu^k_{\mathbf{x}})} (\mu^{k+1}_{\mathbf{x}} - \mu^k_{\mathbf{x}}) + \mu^k_{\mathbf{x}}. \tag{11}$$

In practice we find the proposed method is simple and effective after all. However, please notice that the median should be tracked per-channel for UQM models.

**Weighted Mode Filter** The weighted mode filter is to find the global mode of $\tilde{h}(\mathbf{x}, g)$. Simple fixed-point iteration is sufficient for the proposed Gaussian models. Let the gradient $\partial \tilde{h}(\mathbf{x}, g) / \partial g = 0$, we have the fixed-point iteration as

$$g^{n+1}_{\mathbf{x}} = \left( \sum_{l=1}^L \mathcal{B}^l_{\mathbf{x}}(g^n_{\mathbf{x}}) \left(\boldsymbol{\Sigma}_n + \boldsymbol{\Sigma}^l_{\mathbf{x}}\right)^{-1} \right)^{-1} \left( \sum_{l=1}^L \mathcal{B}^l_{\mathbf{x}}(g^n_{\mathbf{x}}) \left(\boldsymbol{\Sigma}_n + \boldsymbol{\Sigma}^l_{\mathbf{x}}\right)^{-1} \mu^l_{\mathbf{x}} \right), \tag{12}$$

where $\mathcal{B}^l_{\mathbf{x}}(g^n_{\mathbf{x}}) = N(g^n_{\mathbf{x}} | \mu^l_{\mathbf{x}}, \boldsymbol{\Sigma}_n + \boldsymbol{\Sigma}^l_{\mathbf{x}}) \psi_{\mathbf{x}}(l)$. Eq. (12) recursively goes to the closest mode and thus a good initialization $g^0_{\mathbf{x}}$ is necessary to avoid being stuck in wrong local mode. In practice, let $g^0_{\mathbf{x}} = \mu^{m^\star}_{\mathbf{x}}$ where $m^\star = \arg\max_m \sum_{l=1}^L \mathcal{B}^l_{\mathbf{x}}(\mu^m_{\mathbf{x}})$ is both effective and reasonable.

## 5 Experimental Results and Discussions

### 5.1 Implementation Notes

We have implemented the proposed weighted mode filter and the weighted median filter on a `MATLAB` platform. The results reported were measured on a 3.4 GHz Intel Core i7 processor with 16 GB RAM.

**Parameter Definition** All input images and guidance images were normalized into $[0, 1]$ for the convenience of parameter definition. The data variance $\mathbf{\Sigma}_n = \sigma_n^2 \mathbf{I}^d$, where $\sigma_n$ is the standard variance of the noise, $\mathbf{I}^d$ is an identity matrix and $d$ is the dimension of the input image. The guided filter (GF) and the domain transform filter (DF) share the same parameter setting, *i.e.*, $r = \sigma_s$ and $\varepsilon = \sigma_r^2$ ($\boldsymbol{\omega} = \{r, \varepsilon\}$ for GF, $\boldsymbol{\omega} = \{\sigma_s, \sigma_r\}$ for DF). $r$ and $\sigma_s$ was measured in pixels. For fair comparisons, the number of iterations in the weighted mode filter was set as 10 for all the experiments.

**Number of Models** An automatic criterion [12] stops generating the LAM models when a high percentage of pixels are close to at least one model. In detail, the criterion of closeness is set as $\|f_{\mathbf{x}} - \mu_{\mathbf{x}}^l\|_{\mathbf{\Sigma}_n} \leq 1$. Together with a user-given threshold $L^{\text{th}}$, the LAM models generation will be stopped when either the criterion or $L^{\text{th}}$ is reached. In addition, the number of the UQM models shared the same threshold $L^{\text{th}}$, and no automatic stopping criterion was applied.

**Compared Methods** In this paper, we compared our proposed filters with two popular filters: the constant time weighted median filter (CT-median) [1] and the bilateral weighted mode filter (BF-mode) [5]. The parameters of CT-median were given by the authors [1] and those of BF-mode were optimized by exhaustive search. The number of bins in the reference methods was fixed to 256 per-channel [1, 2, 5].

### 5.2 Performance Evaluation

**Runtime Comparison** Fig. 3 shows the execution time comparison between our method and the brute-force constant time algorithm (*cf.* Eq. (1)) with GF weights to construct the distribution. Both LAM and UQM models were under evaluation. Related parameters were fairly configured. The $y$-axis is the ratio of runtime of the proposed method *w.r.t.* the reference method, which assumed 256 discretized bins. $L$ was defined manually without automatic stopping criterion. Both the two proposed methods only possess a fraction of the runtime against the reference one and are nearly proportion to the number of models. But the LAM spends a little bit more time because of additional filtering operations at the model generation step. Notice that when $L$ is around 50, the execution time of the proposed methods becomes half of that of the reference one.

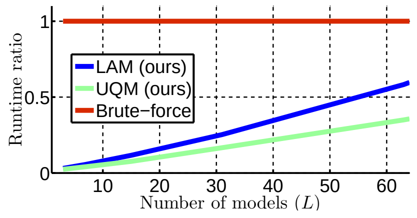

**Fig. 3.** Execution time comparison on the distribution construction *w.r.t.* the number of models. The input is a 8-bit single-channel image and the guidance is a 3-channel image. The reference method is brute-force and traverses 256 discretized bins.

**The Number of Necessary LAM models** In fact, natural images, no matter color images or disparity/depth maps, are always locally smooth. There is little necessity to generate so many LAM models (*e.g.*, more than 60) to fit the local distribution. To validate this observation, we estimated the LAM models for all the color images in a published image dataset `BSDS300` [19] with the threshold of $L^{\text{th}} = 64$ and examined the distribution of necessary number of models. The automatic stopping criterion was triggered when no less than 99.9% pixels were fulfilled the constraint in Sec. 5.1.

Results are illustrated in Fig. 4, where the left one was obtained by a window size $21\times21$ (*i.e.*, $r = 10$) and the right one was $11\times11$ (*i.e.*, $r = 5$). $\mathbf{\Sigma}_n = 0.01\times\mathbf{I}^3$ for both cases. The majority of images generally required at most 50 models to meet the criterion. What's more, the smaller the window size is, the fewer number of necessary models are required, which verifies the discussions in Sec. 4.3. Based on these results, we conclude that for the general case, the number of LAM models required for a natural image merely exceeds a certain value under a given window size. As a typical case, let the window size be $21 \times 21$ or smaller, we can safely constrain the threshold to $L^{\text{th}} = 64$, and the runtime on the probability distribution construction is always fewer than half of the brute-force implementation, as shown in Fig. 3.

As a conclusion, the gain of the proposed method is generally $2 \sim 3\times$ faster than the brute-force one for the gray-scale images. And it can be increased to $6 \sim$

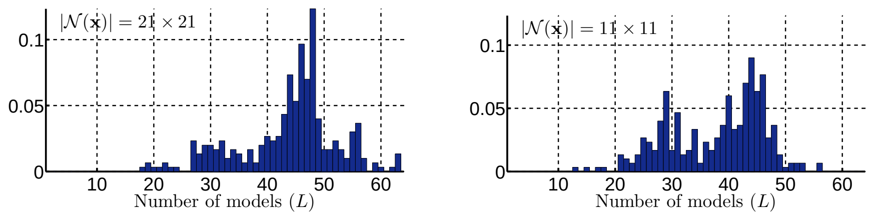

**Fig. 4.** The distribution of the number of necessary local adaptive models in `BSDS300` dataset. *Left*: the window size is $21 \times 21$. *Right*: the window size is $11 \times 11$. The smaller the window size, the fewer number of locally adaptive models is necessary.

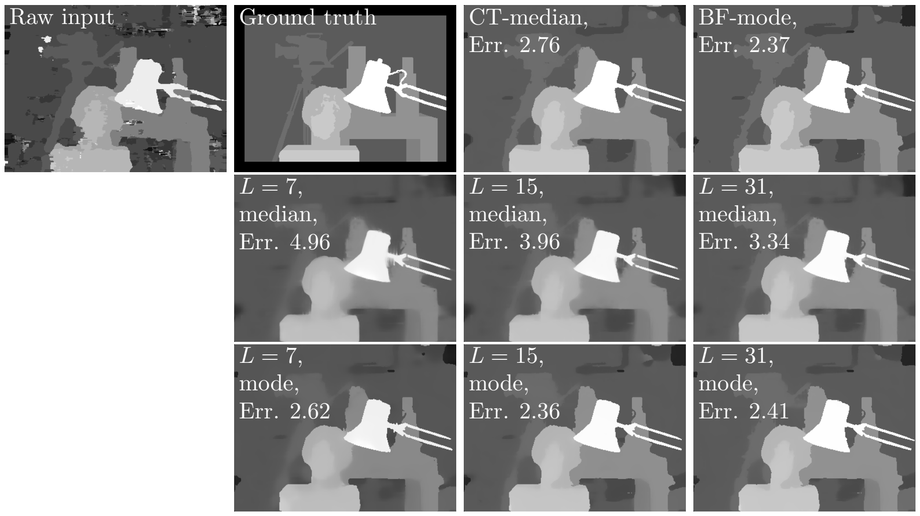
Raw input
Ground truth
CT-median, Err. 2.76
BF-mode, Err. 2.37
$L = 7$, median, Err. 4.96
$L = 15$, median, Err. 3.96
$L = 31$, median, Err. 3.34
$L = 7$, mode, Err. 2.62
$L = 15$, mode, Err. 2.36
$L = 31$, mode, Err. 2.41

**Fig. 5.** Depth map enhancement on `tsukuba`. The first row shows the raw input disparity map, the ground truth, results by CT-median [1] and BF-mode [5] respectively, from left to right. Disparity maps in the 2nd and 3rd rows were obtained by the proposed weighted median filter and weighted mode filter, under different number of models. The models were generated by the LAM models. The error was evaluated on bad pixel ratio with the threshold 1. GF weights were chosen and related parameters were fairly configured.

$9\times$ for color images as the number of channels is increased. For disparity/depth maps and cartoon images, the number of necessary models can be reduced even further because of their high structure homogeneity.

### 5.3 Applications

**Depth Map Enhancement** Depth maps with low resolution and poor quality, *e.g.*, structural outliers, depth holes, noise and *etc.*, can be enhanced with the guidance of the registered high resolution texture images [5, 1]. It is a popular and practical post-processing for acquiring visual plausible and high accurate depth map from various depth acquisition techniques, like stereo, ToF-camera or Kinect. Two state-of-the art approaches that take advantage of the statistics information of the depth map are BF-mode [5] and CT-median [1]. Our methods, both the weighted mode filter and the weighted median filter, gain similar performance against them and require much less cost.

Fig. 5 shows the results of a disparity map named `tsukuba`. The raw input was generated by a simple box-filter aggregation [20] followed by left-right check and hole-filling. LAM models were adopted for all these results and we fixed the number of models utilized. Small $L$ (*e.g.*, $L = 7$) limits the LAM to define enough models to cover all the local structures, thus tended to output slightly blurred results or assign incorrect values in comparison with the referenced methods.

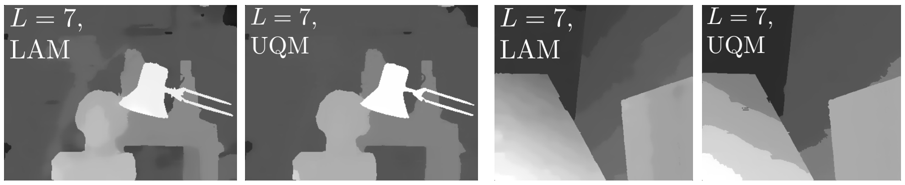

**Fig. 6.** Results of the weighted mode filter with 7 models.

Fortunately, by adopting a few more models, the results become stable and similar to the reference results. For instance, the BF-mode in our implementation required 15.09 sec to process the `tsukuba` image, but the proposed weighted mode filter with 31 LAM models only cost 5.23 sec. What's more, the bad pixel ratio of the proposed method (Err. 2.41) is similar as that (Err. 2.37) of BF-mode, but the PSNR is otherwise higher (25.28dB) against that of the BF-mode (25.09dB).

Although a small $L$ of the LAM models cannot cover all the details of the input image, it still has a superior performance against the UQM models with the same $L$. As shown in Fig. 6, when $L = 7$, the LAM models captured more details of the two test disparity maps and produced smoother outputs than the UQM models The staircase artifact of the UQM models also occurs at BF-mode and CT-median, since both of them are based on a discretized weighted histogram. When the bin number is not sufficient, the quantization artifact will happen around the smooth and slanted surfaces.

**JPEG Artifact Removal** JPEG compression is a lossy compression scheme that usually brings about quantization noise and block artifact. CT-median has been proven effective in eliminating this compression artifact in clip-art cartoon images [1]. However, since CT-median encourages piecewise constant intensities/colors, its drawback is apparent when processing natural images.

As shown in Fig. 7(b) and its zoomed-in patch, CT-median forced the image `eyes` into several distinct layers, pixels inside one layer seemed constant everywhere. Contrary to it, exploiting the LAM models, our method represented a piecewise smooth result, as shown in Fig. 7(c). Not only the compression artifact was removal, but the structure of the input image was still preserved. The UQM models, unfortunately, had a slightly worse performance than that of LAM. The reason is straightforward as it also tried to recover piecewise constant colors. In terms of runtime comparison, both the LAM and UQM models only spent a small fraction of the runtime owned by CT-median (*i.e.*, 88.134 sec) to obtain Fig. 7(b). The LAM models required $L = 15, \mathbf{\Sigma}_n = 0.07^2 \times \mathbf{I}^3$ and $|\mathcal{N}(\mathbf{x})| = 11 \times 11$, it cost 16.74 sec in total. The UQM models also owned $L = 15$, and the runtime was a little faster as 15.54 sec.

**More Applications** We show two additional applications to indicate the potential of the proposed weighted median filter and the weighted mode filter. Fig. 8 shows the detail enhancement for a natural `rock` image by the proposed weighted

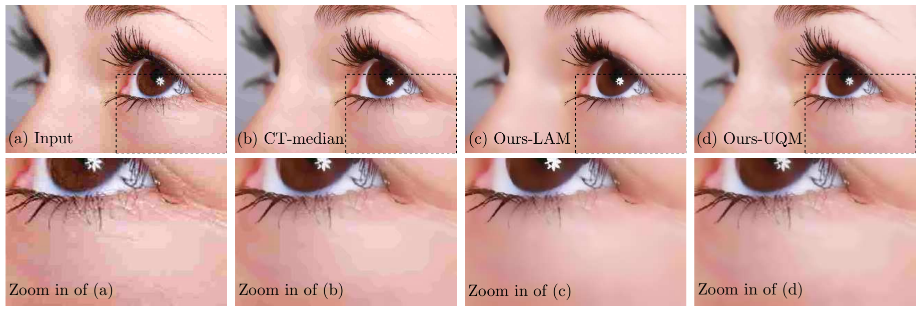

**Fig. 7.** JPEG compression artifact removal results by the weighted median filter. (a) The input degraded `eyes` image. (b) CT-median [1]. (c) The proposal weighted median filter with the LAM models and (d) is with the UQM models. The second row shows the corresponding zoomed-in patches. The DF weights were chosen and all the related parameters were fairly configured. Best viewed in electronic version.

median filter under the LAM models. The result is plausible for naked eyes without apparent artifact. Fig. 9 presents the joint upsampling of a low-resolution and noisy disparity map with the guidance of a registered high-resolution image. Both of the proposed filters generated satisfactory results but the result by the weighted median filter tended to be smoother and introduced a little blurring artifact, while that by the weighted mode filter was sharper and contained a slight of staircase artifact.

## 6 Conclusion and Future Work

In this paper, we propose a novel distribution construction method for accelerating the weighted median/mode filters by defining a new separable kernel based on the probabilistic generative models. Different from traditional methods that need quite a number of filtering operations to estimate a sufficiently accurate distribution, the proposed approach only requires a finite and a small amount of filtering operations based on the structure of the input image. The accelerated

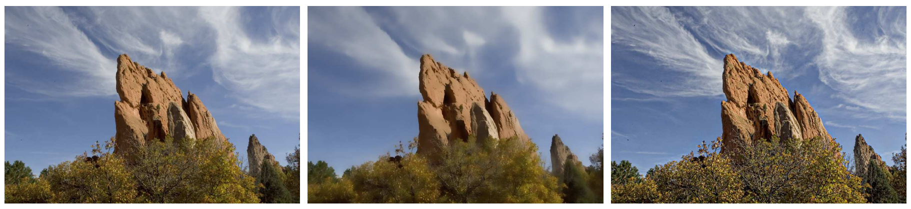

**Fig. 8.** Detail enhancement by the proposed weighted median filter under the LAM models. From left to right, the original `rock` image, after edge-preserving smoothing, and the detail enhanced image. GF weights were chosen.

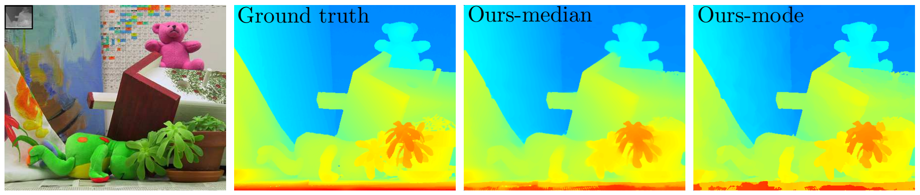

**Fig. 9.** Joint depth map upsampling. The input disparity map was 8× upsampled by the proposed weighted median filter and the weighted mode filter under the LAM models. The raw input diparity map is shown in the top-left corner of the leftmost image. GF weights were chosen.

weighted median filter and weighted mode filter are thus introduced and utilized into various applications from depth map enhancement, joint depth upsampling, outlier removal, detail enhancement and so on.

As a part of the future work, the extension for video processing is interesting and meaningful. A more robust and efficient way to estimate the locally adaptive models shall be a great benefit. Moreover, increasing the efficiency on the median tracking and mode seeking can further accelerate the proposed filters.

## References

1. Ma, Z., He, K., Wei, Y., Sun, J., Wu, E.: Constant time weighted median filtering for stereo matching and beyond. In: Proc. IEEE Int. Conf. Comput. Vis. (2013)
2. Kass, M., Solomon, J.: Smoothed local histogram filters. ACM Trans. Graph. **29** (2010) 100
3. Perreault, S., Hébert, P.: Median filtering in constant time. IEEE Trans. Image Process. **16** (2007) 2389–2394
4. Cline, D., White, K., Egbert, P.: Fast 8-bit median filtering based on separability. In: Proc. IEEE Int. Conf. Image Process. Volume 5. (2007) V – 281–V – 284
5. Min, D., Lu, J., Do, M.: Depth video enhancement based on weighted mode filtering. IEEE Trans. Image Process. **21** (2012) 1176–1190
6. Van de Weijer, J., Van den Boomgaard, R.: Local mode filtering. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Volume 2. (2001) II–428
7. Parzen, E.: On estimation of a probability density function and mode. Annals of Mathematical Statistics **33** (1962) 1065–1076
8. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proc. IEEE Int. Conf. Comput. Vis. (1998) 839–846
9. Barash, D., Comaniciu, D.: A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. Image and Vision Computing **22** (2004) 73–81
10. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. In: ACM Trans. Graph. Volume 26. (2007) 96
11. Gastal, E.S.L., Oliveira, M.M.: Domain transform for edge-aware image and video processing. ACM Trans. Graph. **30** (2011) 69:1–69:12
12. Gastal, E.S., Oliveira, M.M.: Adaptive manifolds for real-time high-dimensional filtering. ACM Trans. Graph. **31** (2012) 33

13. He, K., Sun, J., Tang, X.: Guided image filtering. In: Proc. Eur. Conf. Comput. Vis. Springer (2010) 1–14
14. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. In: Proc. Eur. Conf. Comput. Vis. Springer (2006) 568–580
15. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. In: ACM Trans. Graph. Volume 26. (2007) 103
16. Adams, A., Gelfand, N., Dolson, J., Levoy, M.: Gaussian kd-trees for fast high-dimensional filtering. In: ACM Trans. Graph. Volume 28. (2009) 21
17. Adams, A., Baek, J., Davis, M.A.: Fast high-dimensional filtering using the permutohedral lattice. In: Computer Graphics Forum. Volume 29. (2010) 753–762
18. Bishop, C.M., Nasrabadi, N.M.: Pattern recognition and machine learning. Volume 1. springer New York (2006)
19. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. IEEE Int. Conf. Comput. Vis. Volume 2. (2001) 416–423
20. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. Journal of Comput. Vis. **47** (2002) 7–42