Computer Aided Geometric Design 43 (2016) 68–81

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Kernel-based adaptive sampling for image reconstruction and meshing

Zichun Zhong *, Jing Hua

Department of Computer Science, Wayne State University, USA

ARTICLE INFO

Article history:
Available online 17 February 2016

Keywords:
Adaptive sampling
Gaussian kernel
Image reconstruction
Image-based meshing

ABSTRACT

This paper introduces a kernel-based sampling approach for image reconstruction and meshing. Given an input image and a user-specified number of points, the proposed method can automatically generate adaptive distribution of samples globally. We formulate the problem as an optimization to reconstruct the image by summing a small number of Gaussian kernels to approximate the given target image intensity or density. Each Gaussian kernel has the fixed size and the same energy. After the optimization, the samples are well distributed according to the image intensity or density variations as well as faithfully preserved the feature edges, which can be used to reconstruct high-quality images. Finally, we generate the adaptive triangular or tetrahedral meshes based on the well-spaced samples in 2D and 3D images. Our results are compared qualitatively and quantitatively with the state-of-the-art in image sampling and reconstruction on several examples by using the standard measurement criteria.

1. Introduction

Sampling and reconstruction are widely used in computer graphics and its applications, such as halftoning and stippling, point-based rendering, and geometry processing, etc. There are several excellent surveys that describe different algorithms and their used domains (Mitchell, 1990; Glassner, 1995; Dutre et al., 2006; Pharr and Humphreys, 2004; Zwicker et al., 2015). One of the most important properties of the sampling distributions is adapting to a given image, so that the number of points in a region is proportional to the image density as well as the positions of points do well preserve the object shapes in the image without aliasing artifacts. Another desirable property is possessing spectral characteristics, i.e., blue noise property, such as Poisson disc samplings (Deussen et al., 2000), Lloyd's method for weighted Voronoi stippling (Secord, 2002), a variant of Lloyd's method by using capacity-constrained Voronoi tessellation (CCVT) (Balzer et al., 2009), an interacting particle model for electrostatic halftoning (Schmaltz et al., 2010), variational blue noise sampling (Chen et al., 2012), etc. Recently, Fattal (2011) presented a kernel density approach for generating point sets with high-quality blue noise properties that formulates the problem using a statistical mechanics interacting particle model. However, in the Fattal's method, users cannot explicitly specify the final sampling population since each kernel has a different size and the final sampling density is controlled by a statistical mechanism with local interactions. Thus if the initialization does not have a good estimation of target density, it will obtain a poor result. Besides that, all the above previous sampling methods

* Corresponding author.
E-mail address: zichunzhong@wayne.edu (Z. Zhong).

http://dx.doi.org/10.1016/j.cagd.2016.02.013
0167-8396/

mainly focus on blue noise characteristics without considering the fidelity of the sampling result to the original image, and the reconstructed image quality.

In this paper, we propose a method to minimize a total energy that is the difference between the total summation of a user-specified number of Gaussian energies and the given input image, leading to an adaptive distribution of samples, i.e., the Gaussian kernel centers, conforming to the input image intensity or density, which is determined by different applications. For instance, image intensity is used in image reconstruction; image density is used in stippling; and both of them can be used in adaptive mesh generation. Each Gaussian kernel has the fixed size and the same energy, enforcing the equal importance of each sample in the distribution. Then, the formula for the explicit gradient of the proposed energy function is derived and an efficient numeric approach is proposed to optimize the energy function by using a fast local search based on the L-BFGS method (Liu and Nocedal, 1989). After the optimization, the samples are well distributed according to the image intensity or density variations, i.e., densely in the high intensity or density regions and sparsely in the low intensity or density regions. Since each sample endowed with a fixed Gaussian energy, we can obtain the reconstructed image by summing all these sampling kernels at the optimized positions together. Finally, the adaptive triangular or tetrahedral meshes are generated based on the well-spaced samples in 2D and 3D images.

This paper makes the following contributions for generating high-fidelity adaptive samplings resulting in computing high-quality images and meshes:

- It introduces a new kernel-based approach and consists in optimizing all the samples with kernel energies globally without explicit control of sample population, so that it is sufficiently simple and efficient. The key idea is derived from reconstructing a high-quality image by using the sparse sampling points with kernel energies. When the total energy is optimized, the samples in the image domain will achieve the adaptive pattern with the desired input image intensity or density, as well as faithfully preserve the image feature edges. Since the samples have regular/adaptive patterns in the constant or linear intensity variation regions, and meanwhile well capture the sharp image features, i.e., non-linear intensity variation regions, we can generate high-quality adaptive triangular or tetrahedral meshes based on the well-spaced samples in 2D and 3D images.
- It presents a computationally feasible and efficient method for our energy optimization (Sec. 3). The computational complexity is $O(n+m)$, where n is the number of sampling points, and m is the number of pixels in the image domain. Such energy is C^{∞} smoothness and energy optimization strategy demonstrates very fast convergence speed, without any need for the explicit control of sampling population (e.g., dynamically inserting or deleting samples to meet the input given image intensity or density).

2. Related work

There are extensive studies in the literatures about the sampling distribution and its applications in image reconstruction. We mainly review the most related previous works to our study.

2.1. Point distribution and sampling

In computer graphics, sampling distribution on the image space has become an interesting research topic in the past few decades. Poisson disk samplings (Dippé and Wold, 1985) are uniformly distributed in space without overlapping between disks according to the predefined radius and have good spectrally sampling patterns. Cook (1986) proposed a dart throwing algorithm to generate point distributions. Deussen et al. (2000) generated the stipple drawings by using Poisson disc distributions. Liang et al. (2015) proposed a Poisson disk sampling algorithm based on disk packing for image stippling. Xing et al. (2014) presented highly parallel algorithms for remeshing polygonal models based on the sampling points from human visual perception. In the recent decades, Lloyd's (1982) method is a traditional method to compute centroidal Voronoi tessellation (CVT) for the uniform sampling distribution. McCool and Fiume (1992) improved the spectral properties of the results from Lloyd's method, but without explicit termination criteria. Moreover, Lloyd's method can provide a facility to generate the density sampling by specifying a target density function by Secord (2002). In Balzer et al.'s (2009) work, they proposed a variant of Lloyd's method by imposing a capacity constraint on the Voronoi tessellation. Schmaltz et al. (2010) generated halftoning using an interactive particle system inspired by the physical principles of electrostatics. The computational complexity of the above methods is very high as compared in Sec. 7.1. Chen et al. (2012) proposed an efficient variational framework based on an energy function combining the CVT energy and the CCVT energy. Zhong et al. (2013) used the Gaussian kernel to define an inter-particle energy and forces for surface meshing. Their method is formulated based on minimizing the total particle energy and our proposed method is to minimize the total reconstruction errors; besides that, two methods have different applications. Recently, Fattal (2011) presented a kernel density model for generating point sets with blue noise properties that formulates the problem using a statistical mechanics interacting particle model. However, in the Fattal's method, users cannot explicitly specify the final sampling population and the final density is controlled by a statistical mechanism with local interactions. Thus if the initialization of the sampling does not have a good estimation of target density, it will obtain a poor result. Furthermore, his method cannot well capture the image sharp features and we have compared with our method in Sec. 7.2. Generally, our kernel-based approach is different and consists in optimizing all the samples globally without explicit control of sample population, so that it makes the computation simple and efficient.

2.2. Sampling for image reconstruction

Nowadays, non-uniform sampling has been used to solve aliasing problem in image reconstruction and representation. The basic idea of these kinds of techniques is to compute the error of each pixel between the reconstructed value and the target value, and then adaptively distribute more sampling points in pixels with large errors. Mitchell (1987) has proposed the pioneering work in generating antialiased image by using samples. Bala et al. (2003) described an interactive rendering approach by explicitly representing edges. However, in such method, artifacts may occur once the edge detection fails. Our approach can automatically capture the image feature edges during the optimization without any extra pre-processing work. Bolin and Meyer (1998) used Haar wavelets and developed a perceptual error metric for adaptive sampling. Farrugia and Péroche (2004) presented an adaptive perceptually based image metric for rendering algorithm. Adaptive wavelet rendering technique proposed by Overbeck et al. (2009) is based on smoothing the image and distributing new samples iteratively. Chen et al. (2011) proposed adaptive sampling for reconstruction with depth of field effects. Recently, Rousselle et al. (2011) introduced an approach for image space adaptive sampling and reconstruction in Monte Carlo rendering. Our approach in this paper handles the reconstruction problem in 2D and 3D image spaces by using kernel-based adaptive sampling approach. In contrast, our method concentrates on the quality and fidelity of the reconstructed images without considering the rendering or the depth of field effects.

In this study, we only focus on the adaptive sampling for image representation. Another kind of approaches is to use a mesh directly to represent the image, for example Terzopoulos and Vasilescu (1991) introduced an adaptive mesh approach that can reconstruct image intensity, which is beyond the scope of this work and future studies will be considered.

3. Kernel-based approach

In this section, we present the proposed kernel-based method for uniform and adaptive samplings. First, we introduce the problem and background. Then, the basic framework and computation method are provided. Finally, we give the details of the algorithm.

3.1. Basic framework

With the specified number of samples, the quality of the reconstructed image is closely related to the distribution of the samples in our framework. In Fattal's (2011) work, he proposed to place a kernel on each sample with an adaptive kernel size and use their weighted summation as an approximation to a given target function. Minimizing the error of such approximation can lead to a uniform or adaptive distribution of the samples. In his formulation, a statistical mechanics is used to interact particle model and the final number of samples cannot be set explicitly by users in advance, which need extra trial-and-error iterations. However, our proposed energy optimization strategy is based on the inter-kernel energy and such optimization is directly controlled by the image intensity or density. It shows very fast convergence speed, without any need for the explicit control of sample population (e.g., inserting or deleting samples to meet the desired density or property). In this paper, we show that in uniform or adaptive case, such kernel-based function optimization can be considered as interactions between kernels. Each Gaussian kernel has the fixed size and the same energy, so that its importance is equal. The potential energy between the sample and its covered image pixels/voxels is called *Gaussian energy*. When the kernels reach the optimal balanced state with uniform or adaptive distribution, the Gaussian energies applied on each sample become equilibrium.

Energy definition: Given n samples with their positions $\boldsymbol{\mu} = \{\boldsymbol{\mu}_i | i = 1 \dots n\}$ in the domain $\Omega \subset \mathbb{R}^d$, we place the Gaussian kernel centered at each sample position $\boldsymbol{\mu}_i$, and the Gaussian kernel energy is:

$$G(\mathbf{x}, \boldsymbol{\mu}_i) = \frac{1}{(\sqrt{2\pi}\sigma)^d} e^{-\frac{\|\mathbf{x}-\boldsymbol{\mu}_i\|^2}{2\sigma^2}}, \tag{1}$$

where $\mathbf{x}$ are the positions of pixels or voxels in the image domain Ω. σ is the standard deviation of Gaussian kernel, and d is the dimension of space Ω (in this paper, we focus on 2D and 3D cases). We call σ the "kernel width", and assume that all kernels have the same fixed width. Here the normalization factor $\frac{1}{(\sqrt{2\pi}\sigma)^d}$ ensures that the kernel's integral to be fixed. Given any target function $C(\mathbf{x})$, such as the intensity or density values of pixels/voxels, we use the summation of weighted kernels to approximate $C(\mathbf{x})$, i.e., to minimize the following total energy:

$$E(\boldsymbol{\mu}) = \int_{\Omega} |C(\mathbf{x}) - \sum_{i=1}^{n} \kappa G(\mathbf{x}, \boldsymbol{\mu}_i)|^2 \, \mathrm{d}s, \tag{2}$$

where κ is the kernel weight, which is a constant for all kernels.

Kernel weight constraint: The concept of the kernel weight is defined as follows. Suppose a set $\boldsymbol{\mu}$ of n samples that determines n kernels in the entire image space Ω, the total energy of an image can be considered as the integral of the target function values in Ω as $\int_{\Omega} C(\mathbf{x}) \mathrm{d}s$.

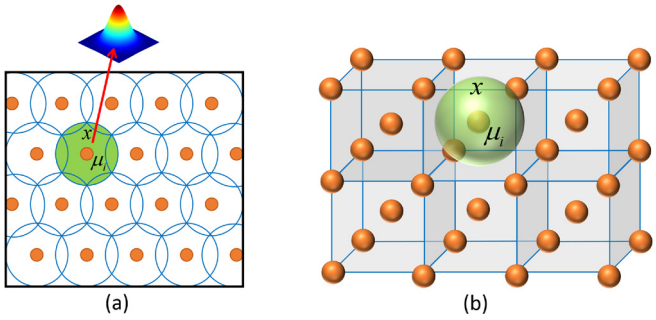

Fig. 1. Kernel-based scheme in a constant domain: (a) hexagonal pattern in 2D case; (b) body-centered-cubic (BCC) lattice in 3D case.

So that, for each sample, the kernel weight is given by:

$$\kappa = \frac{\int_\Omega C(\mathbf{x})\mathrm{d}s}{n}. \tag{3}$$

Intuitively, our kernel weight constraint enforces that each sample in a distribution is equally important. This constraint has the similar functionality to the capacity constraint in CCVT method (Balzer et al., 2009). At the same time, this constraint preserves the energy conservation: the total summation of all the kernel energies should be equal to the image total energy $\int_\Omega C(\mathbf{x})ds$.

Uniform case: If we set the target function to be a constant in the entire domain, such as $C(\mathbf{x}) = 1$, then minimizing $E(\boldsymbol{\mu})$ w.r.t. the sample positions $\boldsymbol{\mu}$ will achieve a hexagonal distribution of samples in 2D space, like Fattal (2011) as shown in Fig. 1 (a). So this kernel-based function approximation framework has the capability to achieve uniform sampling, similar to the results of CVT. In the 3D volume space, it is intuitive that the sample positions $\boldsymbol{\mu}$ will achieve a body-centered-cubic (BCC) lattice (i.e., similar to 3D CVT results Du and Wang, 2005) as Fig. 1 (b).

Adaptive case: If $C(\mathbf{x})$ is an unconstant in the domain, such as a real gray-scale image, minimizing $E(\boldsymbol{\mu})$ w.r.t. the sample positions $\boldsymbol{\mu}$ will achieve an adaptive hexagonal/BCC distribution of samples in 2D/3D space, which is similar to the results of CVT in density case. Since each kernel has equal importance to the reconstruction, if the target function $C(\mathbf{x})$ has higher value in the domain, the samples will be densely positioned there; otherwise, the samples will be sparsely positioned.

3.2. Optimization with L-BFGS method

3.2.1. Gradient of energy function and its smoothness

The gradient of $E(\boldsymbol{\mu})$ w.r.t. $\boldsymbol{\mu}_i$ can be considered as:

$$\frac{\partial E(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}_i} = -2\int_\Omega [C(\mathbf{x}) - \sum_{i=1}^{n} \kappa \frac{1}{(\sqrt{2\pi}\sigma)^d} e^{-\frac{\|\mathbf{x}-\boldsymbol{\mu}_i\|^2}{2\sigma^2}}] \frac{\kappa(\mathbf{x}-\boldsymbol{\mu}_i)}{(\sqrt{2\pi})^d \sigma^{d+2}} e^{-\frac{\|\mathbf{x}-\boldsymbol{\mu}_i\|^2}{2\sigma^2}} \mathrm{d}s. \tag{4}$$

From Eq. (4), it is clear to see that the kernel-based energy $E(\boldsymbol{\mu})$ is C^∞, i.e., high order smoothness, since we have used the Gaussian kernel, which is a nonnegative radially-symmetric exponential function. This is a significant good property during the optimization: the C^∞ continuity can ensure to obtain a good optimal result, instead of trapping into some local minima. For instance, the previous CVT-based energies (Secord, 2002; Balzer et al., 2009; Chen et al., 2012) are at most C^2 continuity (Liu et al., 2009); if the global optimization strategy is not used, it is easy to compute a local optimal result. Another good property of Gaussian kernel is that its domain is radially-symmetric: in 2D case, it is a circle, and in 3D case, it is a sphere, which leads to easily simulate the interactions between kernels and form the ideal patterns of the sampling as mentioned in the previous section.

For Gaussian kernels, about 99.99% of the integral values are within five standard deviations (5σ). So for each sample, we only need to compute the kernel effects from the samples within such neighboring pixels/voxels region and this truncated Gaussian kernel is applied to speed up the optimization process.

3.2.2. L-BFGS method

Due to the high order smoothness (C^∞ continuity) of the energy function with the explicit gradient formula, it is therefore possible to minimize the total Gaussian kernel energy function using Newton-like optimization methods and expect fast convergence. Specifically, we use the L-BFGS algorithm (Liu and Nocedal, 1989), a quasi-Newton method, to optimize the sample positions, which can quickly find the minimum of the energy for our kernel-based sampling. It is much more efficient than the traditional Lloyd's (1982) method for CVT computation and more effective than the L-BFGS optimization solver for CVT with a C^2 continuity (Liu et al., 2009). For each iteration of L-BFGS optimization, we update the energy E by summing the differences between the target values and the kernels over the domain, and update its gradients $\frac{\partial E}{\partial \boldsymbol{\mu}_i}$ on each sample. The L-BFGS method computes the approximated inverse Hessian matrix by accumulating the gradients

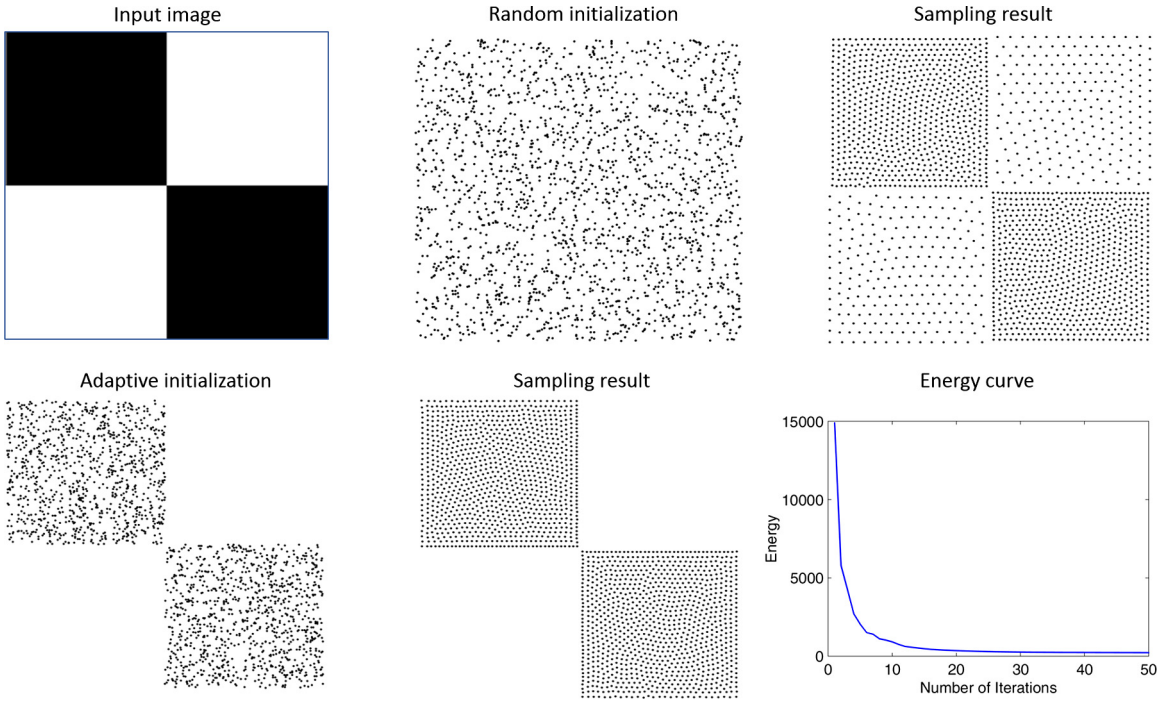

Fig. 2. Sampling results of a 256×256 Chessboard image with 2000 samples in a random initialization and an adaptive initialization cases (the energy curve shows that the proposed energy function with L-BFGS optimization method can converge very quickly).

over a small fixed number of preceding iterations. Both the memory complexity and the computational cost of each iteration of L-BFGS are $O(n)$, where n is the number of variables, here it is the number of samples. The third image of Row 2 in Fig. 2 shows the optimization energy E's curve of a Chessboard image with 2000 samples in an adaptive initialization. The energy optimization by L-BFGS method can converge very quickly in 50 iterations for all our experimental images.

3.3. Kernel width

The Gaussian kernel energy as defined in Eq. (1) depends on the choice of the fixed kernel width σ, which will determine the final sampling. The slope of this energy peaks at distance of σ and it is near zero at much smaller or much greater distances. If σ is chosen too small then kernels will nearly stop spreading when their separation is about 5σ, because there is almost no overlapping between kernels, which may lead to aliasing and artifacts in the computed image. If σ is chosen too large then nearby kernels cannot repel each other and the resulting sampling pattern will be poor. In this work, σ is set to be proportional to the average "radius" of each kernel when they are uniformly or adaptively distributed on Ω: $\sigma = c_\sigma \sqrt[d]{|\Omega|/n}$, where $|\Omega|$ denotes the area/volume of the image Ω; n is the number of samples; d is the dimension; and c_σ is a constant coefficient. It is noted that our goal is to let the samples be uniformly or adaptively located in Ω. We conduct the experiments in both 2D and 3D cases, with different image sizes and different numbers of samples. From our extensive experiments, we find out that the best value of c_σ is around 0.3.

3.4. Adaptive initialization

In order to obtain a faster convergence rate and better sampling result, we use Algorithm 1 as follows to demonstrate the adaptive initialization strategy according to the image intensity or density values.

Fig. 2 shows the sampling results of a 256×256 Chessboard image with 2000 samples in a random initialization and an adaptive initialization cases. Chessboard image has sharp edges between the white and black grids. Without a good initialization, it is very challenging to optimize the samples only located in the black regions. It is clear to see that in the random initialization case, there are still some samples sparsely located in the white grids after optimization in Sec. 3.5, although the number of them is much less than that located in the black grids. Through the designed adaptive initialization strategy, we can obtain a good initialization as all the samples are located in the black grids. After the optimization, we can have the expected results as we desired as shown in Fig. 2. The running time (on a desktop with Intel(R) Xeon E5-2650 CPU 2.30 GHz and 32G DDR4 RAM) of using the adaptive initialization is 1.2043 seconds and it converges at 50 iterations; however, the random initialization needs more iterations to converge, i.e., 75 iterations and it takes 1.7911 seconds. From the above example, this pre-processing step for adaptive initialization is proved to be very important to affect the efficiency and accuracy of the following sampling optimization step.

Data: an image Ω and the desired number of samples n
Result: an adaptive initialization $\boldsymbol{\mu}$ of Ω
Normalize the image intensity or density values from 0 to 1;
while *current sampling number i less than n* **do**
  Randomly locate sample i in the image domain Ω;
  Get sample i's density value ρ (use bilinear interpolation, if sample i is not located exactly at pixel's position);
  Generate a random number p as a probability from 0 to 1;
  if $\rho >= p$ **then**
    Reserve this sample i as the adaptive initialization location for $\boldsymbol{\mu}_i$;
    Update $i = i + 1$;
  end
  else
    Discard this sample i;
  end
end

Algorithm 1. Adaptive initialization for an input image.

Data: an image Ω and the desired number of samples n
Result: an adaptive sampling $\boldsymbol{\mu}$ of Ω
Initialize sampling locations $\boldsymbol{\mu}$;
while *stopping condition not satisfied* **do**
  for *each sample i* **do**
    Get sample i's kernel domain with the width 5σ in Ω, i.e., $N(i)$;
    for *each pixel $j \in N(i)$* **do**
      Compute the weighted Gaussian kernels $\kappa G(\mathbf{x}_j, \boldsymbol{\mu}_i)$;
    end
  end
  Sum the total kernels $\sum_{i=1}^{n} \kappa G(\mathbf{x}, \boldsymbol{\mu}_i)$;
  for *each pixel $j \in \Omega$* **do**
    Compute the energy at pixel j, i.e., the difference between the target value of the image and the summed weighted Gaussian value: $C(\mathbf{x}_j) - \sum_{i=1}^{n} \kappa G(\mathbf{x}_j, \boldsymbol{\mu}_i)$;
  end
  Sum the total energy E in Eq. (2);
  for *each sample i* **do**
    Compute the gradients $\frac{\partial E(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}_i}$ using Eq. (4);
  end
  Run L-BFGS with E and $\frac{\partial E(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}_i}$, to get updated locations $\boldsymbol{\mu}$;
  Project $\boldsymbol{\mu}$ into the image if $\boldsymbol{\mu}$ is out of image domain;
end

Algorithm 2. Adaptive sampling optimization for an input image.

3.5. Sampling optimization algorithm

Our kernel-based image sampling method is summarized in Algorithm 2 below. To help reproduce our results, we further detail each component of the algorithm and the implementation issues. For any input image Ω, we use the adaptive initialization strategy in Sec. 3.4 according to the image density.

During L-BFGS optimization, the samples need to be constrained in the image Ω. In each iteration, the updated samples $\boldsymbol{\mu}_i$ need to be projected to their nearest locations inside Ω, if they are out of the image domain. This optimization process is iterated until convergence by satisfying a specified stopping condition, e.g., the maximal number of iterations, the magnitude of the gradient, the magnitude of the total energy, or the maximal displacement of samples is smaller than a threshold. Algorithm 2 shows the details of our adaptive sampling optimization for an image. Our formulation is based on interactions between kernels and the computational complexity is $O(n + m)$, where n is the number of sampling points, and m is the number of discrete spatial points, i.e., pixels or voxels in the image domain.

4. Applications

In this section, we introduce the applications by using our proposed sampling method, such as stipple drawing, image reconstruction, and mesh generation.

4.1. Stipple drawing

One of the applications for the proposed sampling framework is to generate stipple drawings to approximate the gray-scale images, such as in printing, when $C(\mathbf{x})$ in the energy function (Eq. (2)) is equal to image ink density values, i.e., the inverse pixel intensity values of the gray-scale image. Visually, stipples are positioned closer together to form dark regions and further apart to form bright regions. Note that the significant advantage of the proposed method over all previous approaches based on CVT is that they need to compute a Voronoi diagram (Du et al., 1999) in each iteration of the optimization process, which is much more time-consuming. Our proposed energy function is C^{∞} continuity, so it is easy to

obtain an optimal result; however, the CVT-based methods are C^2 continuity. Compared with Fattal's method, it needs the statistical mechanism to interact the sample model and the population of the samples is dynamically controlled during the computation, which is not easy for users to manipulate. We compare both the computational complexity and the quality of sampling with existing sampling approaches in Sec. 7.

4.2. Image reconstruction

We can use such optimized samplings to reconstruct the original images. After the optimization of the sampling positions, the final reconstructed image is computed as the summation of all the weighted kernels, which is similar to the idea of "splat" or "footprint" (Westover, 1990). We evaluate the quality of the reconstructed images in Sec. 6.

4.3. Adaptive mesh generation

Besides the image reconstruction, we can compute adaptive triangular or tetrahedral meshes according to the optimized samples. The final output mesh is generated as the 2D or 3D Delaunay triangulation based on the images. In 2D Delaunay triangulation, we compute the triangulation for the entire image domain at first and then remove the triangles which is outside the image boundaries, such as in the "M" of cap in Mario image, the guitar boundaries in Mark image, and bones, tumor and bubbles in lung regions of CT images in Fig. 5. As for 3D Delaunay triangulation, there are two steps in the mesh generation: (1) Compute the surface mesh: after the sample optimization, we can find the boundary samples at first, and then the triangular mesh is generated as the dual of the Voronoi diagram restricted on the surface (Yan et al., 2009). The dual of the connected components of restricted Voronoi diagram and topology control ensure that the components are discs (possibly by inserting points) (Yan et al., 2009; Lévy and Liu, 2010). (2) Compute the volume mesh: after the surface mesh generation, we can insert the interior samples inside the closed surface mesh and compute the 3D Delaunay triangulation (Si, 2015). Finally, in Sec. 6, we evaluate the quality of the generated meshes by using the criteria provided in Sec. 5.2.

5. Evaluations

Since we are considering the image reconstruction and meshing in the experimental applications, in this section, the image and mesh quality measurements are provided.

5.1. Image evaluations

The proposed method is evaluated thoroughly by using both 2D and 3D images. A conventional normalized cross correlation (NCC) is used to evaluate the similarity (i.e., the linear correlation) between the reconstructed images and the target images:

$$NCC = \frac{\sum_{i=1}^{m}[C_{recon}(i) - \overline{C}_{recon}][C(i) - \overline{C}]}{\sqrt{\sum_{i=1}^{m}[C_{recon}(i) - \overline{C}_{recon}]^2}\sqrt{\sum_{i=1}^{m}[C(i) - \overline{C}]^2}}, \tag{5}$$

where $C_{recon}(i)$ and $C(i)$ are the reconstructed and target values, respectively, over m pixels/voxels. $\overline{C}_{recon}$ and $\overline{C}$ are the average values of the reconstructed and target values. The range of the NCC is $[-1, 1]$. If NCC is 1, it means two values are exactly the same. The larger the NCC is, the more similar the values are.

The normalized root mean square error (NRMSE) between the reconstructed values $C_{recon}(i)$ and the target values $C(i)$ is also used for comparison of images and it denotes the related error:

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{m}[C_{recon}(i) - C(i)]^2}{\sum_{i=1}^{m}[C(i)]^2}}. \tag{6}$$

The range of the NRMSE is $[0, +\infty)$. If NRMSE is 0, it means two values are exactly the same. The smaller the NRMSE is, the more similar the values are.

5.2. Mesh evaluations

To measure the triangular mesh quality, we use the criteria suggested by Frey and Borouchaki (1997). The quality of a triangle is measured by $G = 2\sqrt{3}\frac{S}{ph}$, where S is the triangle area, p is its half-perimeter, and h is the length of its longest edge. G_{min} is the minimal quality of all triangles, and G_{avg} is the average quality. θ_{min} is the smallest angle of the minimal angles of all triangles, and θ_{avg} is the average of the minimal angles of all triangles. $\%_{<30^\circ}$ is the percentage of triangles with their minimal angles smaller than 30°. The angle histogram is also provided to show the angles' distribution of all generated triangles. G is between 0 and 1, where 0 denotes a very skinny triangle and 1 denotes a regular triangle.

Table 1
Statistics of all 2D images and their computational times by the proposed adaptive sampling method.

Image	Trui	Lena	Mario	Plant	Mark	CT
Resolution	256 × 256	256 × 256	256 × 256	256 × 256	256 × 256	256 × 150
# Samples	20,000	20,000	10,000	10,000	20,000	5000
Time (seconds)	11.7681	11.7816	5.9069	5.0897	11.7572	3.6920

Table 2
Evaluation of reconstruction accuracy on all 2D images.

Image	Trui	Lena	Mario	Plant	Mark	CT
NCC	0.9963	0.9871	0.9697	0.9962	0.9935	0.9915
NRMSE	0.0374	0.0613	0.2279	0.0740	0.0713	0.0853

To measure the tetrahedral mesh quality, the criteria in Dardenne et al. (2009), Yan et al. (2013) are employed. The quality of a tetrahedron is measured by $G = \frac{12\sqrt[3]{9V^2}}{\sum l_{i,j}^2}$, where V is the volume of the tetrahedron, and $l_{i,j}$ is the length of the edge which connects vertices v_i and v_j. G_{min} is the minimal quality of all tetrahedrons, and G_{avg} is the average quality of all tetrahedrons. θ_{min} is the smallest angle of the minimal dihedral angles of all tetrahedrons, and θ_{avg} is the average of the minimal dihedral angles of all tetrahedrons. $\%_{10^\circ}$ is the percentage of tetrahedrons with their dihedral angles smaller than 10°, which are considered as slivers. The angle histogram is also provided to show the dihedral angles' distribution of all generated tetrahedrons. G is between 0 and 1, where 0 denotes a sliver and 1 denotes a regular tetrahedron.

6. Results

We implement the algorithms using both Microsoft Visual C++ 2013 and Matlab R2015a. For the hardware platform, the experiments are implemented on a desktop computer with Intel(R) Xeon E5-2650 CPU with 2.30 GHz, and 32G DDR4 RAM.

6.1. 2D gray-scale images

Firstly, we have tested our proposed method on some 2D gray-scale images, such as Trui, Lena, Mario, Plant, Mark Knopfler, and CT images. The resolutions of these images are all 256 × 256, except the CT image is 256 × 150.

6.1.1. Stippling

To demonstrate the quality of the sample distributions computed by our method, we present non-photorealistic stippling with small black dots in such a way that their distribution gives the impression of tone. Rows 1 and 3 in Fig. 3 show the input images, and Rows 2 and 4 are the stippling results on 2D gray-scale images with samples from 5000 to 20,000. We can see that the stippling results well preserve the gray-scale density and meanwhile the image feature edges are well captured. Table 1 gives the statistics of the 2D images and their computational times by the proposed adaptive sampling method.

6.1.2. Image reconstruction

According to the sampling results from the previous section, we can compute the reconstructed images by summing the Gaussian kernels at the optimized sampling positions. The reconstructed images computed by our proposed method in Fig. 4 can not only well preserve the image features and fine structures, but also suppress the aliasing artifacts. Table 2 reports the quantitative evaluations on all 2D reconstructed images by using NCC and NRMSE measurements.

6.1.3. Meshing

Since our framework can generate adaptive regular sampling patterns, it is possible to compute the Delaunay triangulations based on the image. Fig. 5 shows the adaptive triangular meshing results of Trui, Mario, Mark and CT images. Visually and quantitatively, most of the triangles are adaptive regular, except those triangles around the regions of image sharp edges (i.e., highly non-linear intensity variations) that is because the sampling densities/image intensities are varying dramatically in such regions.

6.2. 3D gray-scale images

Fig. 6 shows the sampling, reconstruction and tetrahedral meshing results with 100,000 points of a 256 × 256 × 152 3D CT volume image. The sampling density is according to the image intensity (noted that is different from previous stippling) and we use adaptive initialization in this 3D volume image, so that there are no points outside the body surface. The middle small image in Row 1 of Fig. 6 shows the samplings of some cross section slices. The samples are densely located in the high intensity regions, such as organs, tissues, and bones; while sparsely located in the low intensity regions, such as lung, and airs between different organs. The computational time of the 3D CT volume image is 603.7533 seconds. The reconstructed

Fig. 3. Stippling results of Trui with 20,000 samples, Lena with 20,000 samples, Mario with 10,000 samples, Plant with 10,000 samples, Mark with 20,000 samples, and CT with 5000 samples.

image can preserve the general image features and shapes of organs, since we only use $\frac{1}{100}$ of the number of the original high resolution points with kernel energies to represent the images. From the 3D reconstructed image, it shows that the image quality is good: NCC is 0.9265, NRMSE is 0.3126. Finally, the adaptive tetrahedral mesh can well capture the desired density as well, such as the lung regions are represented by larger tetrahedrons, and the other parts are using smaller tetrahedrons. The mesh quality is given in Fig. 6. This example demonstrates that the proposed method provides a good approach of adaptive sampling for efficiently computing image reconstruction and high-quality meshes for 3D volume image.

7. Comparisons

In this section, we show our comparative analysis and experiments with other sampling approaches, including CVT-based methods (Sec. 7.1) and Fattal's kernel density model approach (Sec. 7.2). To compare with other methods, we use the same number of output samples.

7.1. Comparison with CVT-based methods

Our proposed method is based on interactions between kernels and the computational complexity is $O(n+m)$, where n is the number of sampling points, and m is the number of discrete spatial points, i.e., pixels in the image domain. In

Fig. 4. Image reconstruction results of Trui with 20,000 samples, Lena with 20,000 samples, Mario with 10,000 samples, Plant with 10,000 samples, Mark with 20,000 samples, and CT with 5000 samples.

contrast, the computational complexity of the Lloyd's weighted CVT method by Secord (2002) is $O(m \log n)$ and Balzer et al.'s (2009) CCVT method by imposing a capacity constraint over the Voronoi tessellation is $O(n^2 + nm \log \frac{m}{n})$. It is clear to see that our proposed method is much efficient than other CVT-based methods, since they need to compute a Voronoi diagram in each iteration of the optimization process, which is much more time-consuming.

Fig. 7 shows the sampling results on a Plant image with 20,000 points of Lloyd's CVT method by Secord (2002), Balzer et al.'s (2009) CCVT method, and our kernel-based method. Image data for Secord's weighted CVT method and Balzer et al.'s CCVT method is provided in Balzer et al. (2009) and the number of iterations is reported on the project website of Balzer et al. (2009) as: CCVT method needs about 130 iterations, and CVT method needs about 1350 iterations, which is much slower since the Lloyd's method is applied. However, our proposed kernel-based adaptive sampling method only needs 50 iterations to achieve the result as shown in Fig. 7, since the proposed Gaussian kernel energy function has high order smoothness (C^∞ continuity) and the L-BFGS algorithm is used to minimize the energy function with a fast convergence speed.

Visually, our sampling result can well capture the image features as well as the local contrasts, such as the "zoom-in" region, i.e., there are two sharp dark edges; while the samples in the other two CVT-based methods scatter in such area to meet the local density requirement. The key reason is that our energy function is defined based on the image reconstruction, so that the samples are easy to capture the intensity variations. For instance, the kernels for the reconstruction, especially in such feature edges areas, will have smallest errors when the samples (i.e., kernel centers) locate exactly on the sharp strip edges.

7.2. Comparison with Fattal's method

In this subsection, we compare the results of Fattal's (2011) approach in two aspects as follows.

(1) Initialization and population control: We use L-BFGS optimizer to directly minimize Fattal's energy in Eq. (3) of Fattal (2011) and our energy in Eq. (2) with numerical integration in a 2D square domain with certain density functions. Fig. 8 shows such comparative results with 1000 sample points in a random initialization. We can see that by using Fattal's energy function, all the samples are attracted to the high density area, which does not match the input density field. The main reason is that Fattal's method is based on the local kernel interactions to update the sampling numbers and positions, so that it is important to have a good initialization as well as it cannot determine the final number of the samples explicitly in advance. In contrast, optimizing our kernel-based energy can generate the correct sampling result conforming to the input density.

(2) Quality of sampling and reconstructed images: Fig. 9 shows sampling and kernel results of Child image with 13,000 samples, which is the same number of output samples as shown in Fattal (2011). Visually, our results can better capture the image feature edges, such as wrinkles under the eyes, white spots in the eyes, nose, and mouth regions, etc. It seems that Fattal's method can generate samples fitting the density field with the good blue noise property in linear time as mentioned in Fattal (2011), but it is not good for capturing image features and fine structures as well as reconstructing images. To further compare the quantitative measurements, Fattal's reconstructed image quality is worse than ours: Fattal's NCC is: 0.9612, NRMSE is 0.1875; our NCC is 0.9813, and NRMSE is 0.0930.

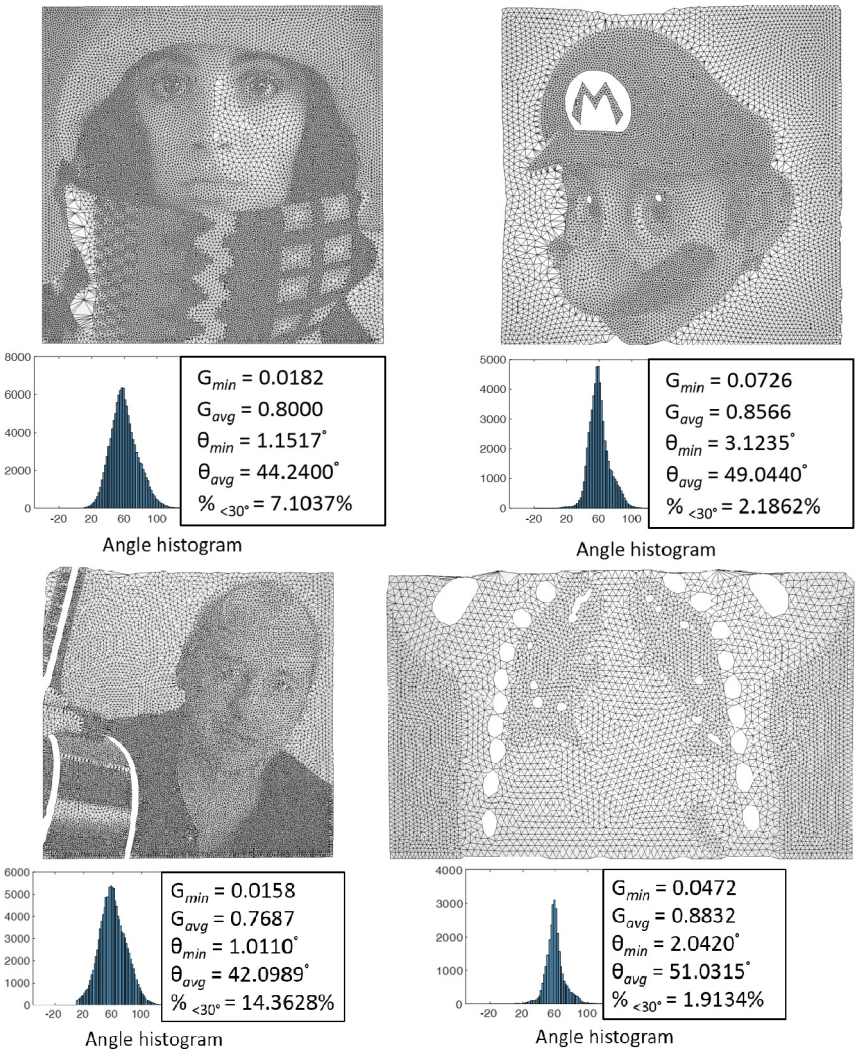

Fig. 5. Adaptive meshing results of Trui, Mario, Mark and CT images.

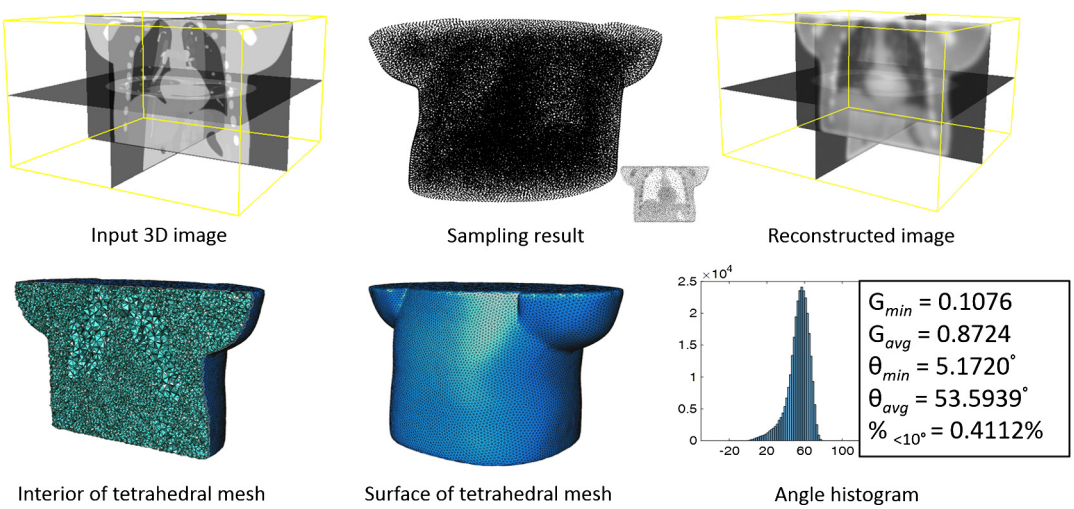

Fig. 6. Adaptive sampling, reconstruction and meshing results with 100,000 points of a 3D CT volume image.

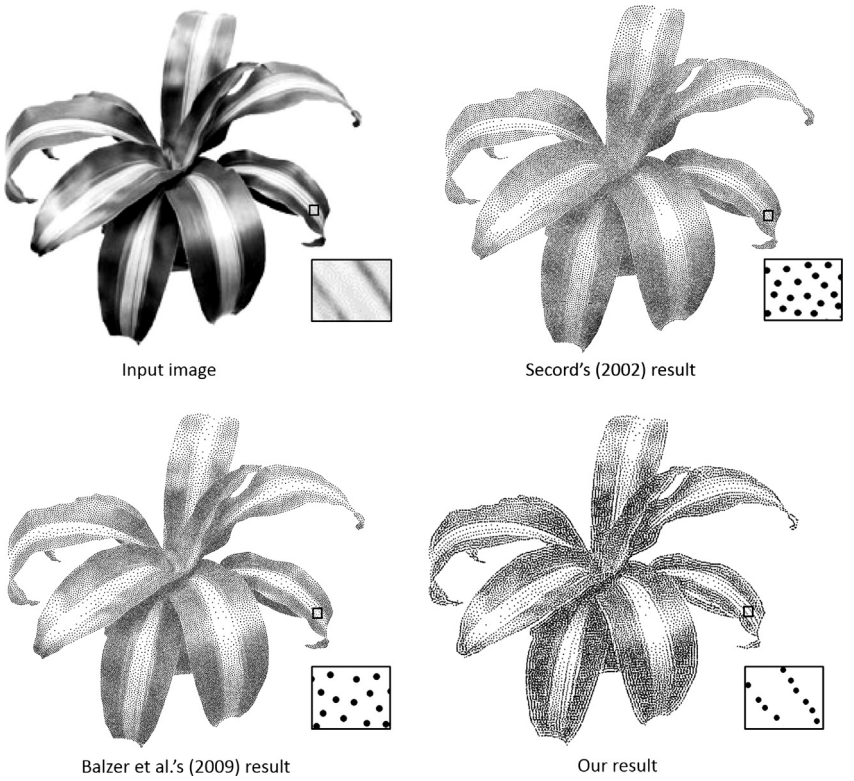

Fig. 7. Comparison with CVT-based methods with 20,000 samples on a Plant image.

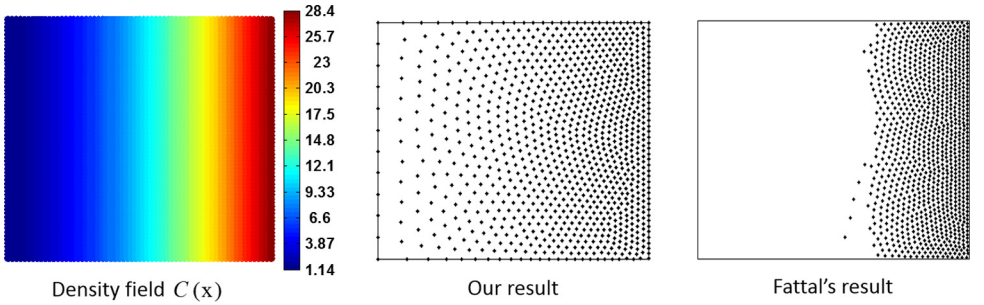

Fig. 8. The sampling results by optimizing our energy function and Fattal's energy function with 1000 samples in a 2D square density field domain.

8. Conclusion

We have presented a new kernel-based adaptive sampling method, which is to optimize the positions of Gaussian kernels globally without explicit control of sampling population. The proposed method can be used in stipple drawing, image reconstruction and adaptive mesh generation in both 2D and 3D images. Our approach has not only improved the computational complexity in linear time $O(n+m)$ compared with previous CVT-based methods, but also better captured the image features and reconstructed high-quality images, such as compared with CVT-based and Fattal's methods. In the future, we will investigate the anisotropic sampling for image reconstruction and meshing, especially images with many sharp features. It is also possible to extend this framework to the surfaces. In order to further speed up the computation, the GPU-based parallel algorithm and implementation will be investigated.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the paper. We also would like to thank Dr. Oliver Deussen and Dr. Raanan Fattal to provide their images and sampling results. The research is supported in part by grants NSFC LZ16F020002, IIS-0915933 and IIS-0937586.

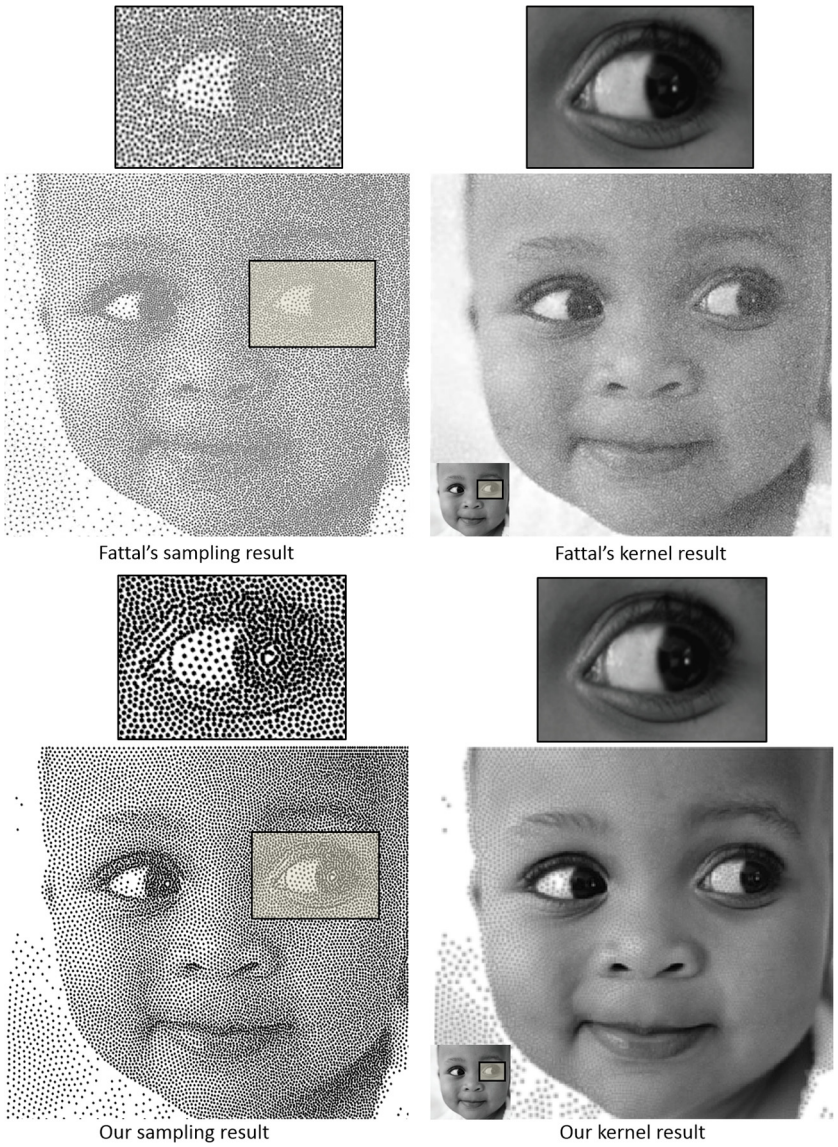

Fig. 9. Comparison with Fattal's method with 13,000 samples on a Child image.

References

Bala, K., Walter, B., Greenberg, D.P., 2003. Combining edges and points for interactive high-quality rendering. ACM Trans. Graph. 22 (3), 631–640.
Balzer, M., Schlömer, T., Deussen, O., 2009. Capacity-constrained point distributions: a variant of Lloyd's method. ACM Trans. Graph. 28 (3), 86.
Bolin, M.R., Meyer, G.W., 1998. A perceptually based adaptive sampling algorithm. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH 1998, pp. 299–309.
Chen, J., Wang, B., Wang, Y., Overbeck, R.S., Yong, J.-H., Wang, W., 2011. Efficient depth-of-field rendering with adaptive sampling and multiscale reconstruction. Comput. Graph. Forum 30 (6), 1667–1680.
Chen, Z., Yuan, Z., Choi, Y.-K., Liu, L., Wang, W., 2012. Variational blue noise sampling. IEEE Trans. Vis. Comput. Graph. 18 (10), 1784–1796.
Cook, R.L., 1986. Stochastic sampling in computer graphics. ACM Trans. Graph. 5 (1), 51–72.
Dardenne, J., Valette, S., Siauve, N., Burais, N., Prost, R., 2009. Variational tetrahedral mesh generation from discrete volume data. Vis. Comput. 25 (5–7), 401–410.
Deussen, O., Hiller, S., van Overveld, C., Strothotte, T., 2000. Floating points: a method for computing stipple drawings. Comput. Graph. Forum 19, 40–51.
Dippé, M.A.Z., Wold, E.H., 1985. Antialiasing through stochastic sampling. In: Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH 1985, pp. 69–78.
Du, Q., Faber, V., Gunzburger, M., 1999. Centroidal Voronoi tessellations: applications and algorithms. SIAM Rev. 41 (4), 637–676.
Du, Q., Wang, D., 2005. The optimal centroidal Voronoi tessellations and the Gersho's conjecture in the three-dimensional space. Comput. Math. Appl. 49, 1355–1373.
Dutre, P., Bala, K., Bekaert, P., Shirley, P., 2006. Advanced Global Illumination. AK Peters Ltd.
Farrugia, J.-P., Péroche, B., 2004. A progressive rendering algorithm using an adaptive perceptually based image metric. Comput. Graph. Forum 23, 605–614.
Fattal, R., 2011. Blue-noise point sampling using kernel density model. ACM Trans. Graph. 30 (4), 48.

Frey, P.J., Borouchaki, H., 1997. Surface mesh evaluation. In: 6th International Meshing Roundtable, pp. 363–373.
Glassner, A., 1995. Principles of Digital Image Synthesis. Morgan Kaufmann.
Lévy, B., Liu, Y., 2010. L_p centroidal Voronoi tessellation and its applications. ACM Trans. Graph. 29 (4), 119.
Liang, G., Lu, L., Chen, Z., Yang, C., 2015. Poisson disk sampling through disk packing. In: Computational Visual Media (Proc. CVM), pp. 17–26.
Liu, D.C., Nocedal, J., 1989. On the limited memory BFGS method for large scale optimization. Math. Program. 45 (3), 503–528.
Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D., Lu, L., Yang, C., 2009. On centroidal Voronoi tessellation – energy smoothness and fast computation. ACM Trans. Graph. 28 (4), 101.
Lloyd, S., 1982. Least squares quantization in PCM. IEEE Trans. Inf. Theory 28 (2), 129–137.
McCool, M., Fiume, E., 1992. Hierarchical Poisson disk sampling distributions. In: Proceedings of the Conference on Graphics Interface '92, pp. 94–105.
Mitchell, D., 1987. Generating antialiased images at low sampling densities. ACM Trans. Graph. 21 (4), 65–72.
Mitchell, D., 1990. The antialiasing problem in ray tracing. In: SIGGRAPH 1990 Course Notes.
Overbeck, R.S., Donner, C., Ramamoorthi, R., 2009. Adaptive wavelet rendering. ACM Trans. Graph. 28 (5), 140.
Pharr, M., Humphreys, G., 2004. Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann.
Rousselle, F., Knaus, C., Zwicker, M., 2011. Adaptive sampling and reconstruction using greedy error minimization. ACM Trans. Graph. 30 (6), 159.
Schmaltz, C., Gwosdek, P., Bruhn, A., Weickert, J., 2010. Electrostatic halftoning. Comput. Graph. Forum 29 (8), 2313–2327.
Secord, A., 2002. Weighted Voronoi stippling. In: Proceedings of the Second International Symposium on Non-photorealistic Animation and Rendering, pp. 37–43.
Si, H., 2015. TetGen, a Delaunay-based quality tetrahedral mesh generator. ACM Trans. Math. Softw. 41 (2), 11.
Terzopoulos, D., Vasilescu, M., 1991. Sampling and reconstruction with adaptive meshes. In: Proceedings Computer Vision and Pattern Recognition (CVPR). 1991, pp. 70–75.
Westover, L., 1990. Footprint evaluation for volume rendering. In: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH 1990, pp. 367–376.
Xing, L., Zhang, X., Wang, C.C.L., Hui, K.-C., 2014. Highly parallel algorithms for visual-perception-guided surface remeshing. IEEE Comput. Graph. Appl. 34 (1), 52–64.
Yan, D., Lévy, B., Liu, Y., Sun, F., Wang, W., 2009. Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. Comput. Graph. Forum 28 (5), 1445–1454.
Yan, D., Wang, W., Lévy, B., Liu, Y., 2013. Efficient computation of clipped Voronoi diagram for mesh generation. Comput. Aided Des. 45 (4), 843–852.
Zhong, Z., Guo, X., Wang, W., Lévy, B., Sun, F., Liu, Y., Mao, W., 2013. Particle-based anisotropic surface meshing. ACM Trans. Graph. 32 (4), 99.
Zwicker, M., Jarosz, W., Lehtinen, J., Moon, B., Ramamoorthi, R., Rousselle, F., Sen, P., Soler, C., Yoon, S.-E., 2015. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. Comput. Graph. Forum 34 (2), 667–681.