Fast Marching farthest point sampling

Carsten Moenning and Neil A. Dodgson
University of Cambridge
Computer Laboratory
15 JJ Thomson Avenue,
Cambridge CB3 0FD - UK
{cm230,nad}@cl.cam.ac.uk

Abstract

We introduce the Fast Marching farthest point sampling (FastFPS) approach for the progressive sampling of planar domains and curved manifolds in triangulated, point cloud or implicit form. By using Fast Marching methods [10, 11, 20] for the incremental computation of distance maps across the sampling domain, we obtain a farthest point sampling technique superior to earlier point sampling principles in two important respects. Firstly, our method performs equally well in both the uniform and the adaptive case. Secondly, the algorithm is applicable to both images and higher dimensional surfaces in triangulated, point cloud or implicit form. This paper presents the methods underlying the algorithm and gives examples for the processing of images and triangulated surfaces. A companion report [12] provides details regarding the application of the FastFPS algorithm to point clouds and implicit surfaces.

1 Introduction

We consider the problem of sampling progressively from planar domains or curved manifolds in triangulated, point cloud or implicit form. An efficient solution to this problem is of interest for a large number of applications including progressive transmission of image or 3D surface data in a limited bandwidth client/server environment, progressive rendering and radiance computing [13], meshless surface reconstruction, point-based multiresolution representation and implicit surface rendering. The method may further be used for the efficient uniform or feature-sensitive simplification of both images and 3D surfaces in triangulated, implicit or point cloud form. In the case of surfaces in point cloud or implicit form, this is achieved without the need for any prior surface reconstruction.

Eldar et al. [6, 7] introduce an efficient uniform irregular farthest point sampling strategy featuring a high data acquisition rate, excellent anti-aliasing properties documented by the "blue noise" power spectrum of the generated sampling distribution and an elegant relationship to the Voronoi diagram concept. Overall, their method compares favourably to other established point sampling techniques such as Poisson disk or jittering [3].

To improve a sample distribution's efficiency its underlying uniform sampling strategy should be made adaptive by taking into account variations in local frequency. Within the farthest point context, this is most naturally achieved by incrementally computing a Voronoi diagram in a non-uniform distance metric with the distance between two points in a highly variable region being greater than the distance between two points in a relatively smooth region. It is well-known that a Voronoi diagram's desirable properties such as connected and convex regions are no longer guaranteed when dealing with a non-uniform distance metric [14]. This leads Eldar et al. [6, 7] to conclude that finding the farthest point in such a Voronoi diagram is impractical. They opt for using the Euclidean Voronoi diagram with its vertices weighted by the bandwidth estimated in the local vertex neighbourhood instead, with the individual weighting approach varying with the particular application.

We propose an alternative (farthest) point sampling approach. This technique incrementally constructs discrete Voronoi diagrams in uniform or non-uniform distance metrics modelled in the form of (weighted) distance maps computed with the help of Fast Marching methods [19, 20, 21]. This novel approach yields an elegant and highly efficient implementation with the resulting Voronoi diagram remaining tractable even when modelling non-uniform metrics for feature-sensitive point selection.

In the following, we briefly review both the Voronoi diagram and the farthest point sampling as well as the Fast Marching level concept. We then put forward our farthest point sampling algorithm, followed by a set of worked examples. We conclude with a brief summary and discussion.

2 Previous Work

2.1 Voronoi diagrams

Given a finite number n of distinct data sites $P := \{p_1, p_2, \ldots, p_n\}$ in the plane, for p_i, $p_j \in P$, $p_i \neq p_j$, let

$$B(p_i, p_j) = \{t \in \mathbb{R}^2 | d(p_i - t) = d(p_j - t)\} \quad (1)$$

where d may be an arbitrary distance metric provided the bisectors with regard to d remain curves bisecting the plane. $B(p_i, p_j)$ is the perpendicular bisector of the line segment $\overline{p_i p_j}$. Let $h(p_i, p_j)$ represent the half-plane containing p_i bounded by $B(p_i, p_j)$. The Voronoi cell of p_i with respect to point set P, $V(p_i, P)$, is given by

$$V(p_i, P) = \bigcap_{p_j \in P, p_j \neq p_i} h(p_i, p_j) \quad (2)$$

That is, the Voronoi cell of p_i with respect to P is given by the intersection of the half-planes of p_i with respect to p_j, $p_j \in P$, $p_j \neq p_i$.

If p_i represents an element on the convex hull of P, $V(p_i, P)$ is unbounded. For a finite domain, the bounded Voronoi cell, $BV(p_i, P)$, is defined as the conjunction of the cell $V(p_i, P)$ with the domain.

The boundary shared by a pair of Voronoi cells is called a Voronoi edge. Voronoi edges meet at Voronoi vertices.

The Voronoi diagram of P is given by

$$VD(P) = \bigcup_{p_i \in P} V(p_i, P) \tag{3}$$

The bounded Voronoi diagram, $BVD(P)$, follows correspondingly as:

$$BVD(P) = \bigcup_{p_i \in P} BV(p_i, P) \tag{4}$$

Figure 1 shows an example of a bounded Voronoi diagram.
Note that the Voronoi diagram concept extends to higher dimensions. For more detail, see the comprehensive treatment by Okabe et al. [14] or the survey article by Aurenhammer [1].

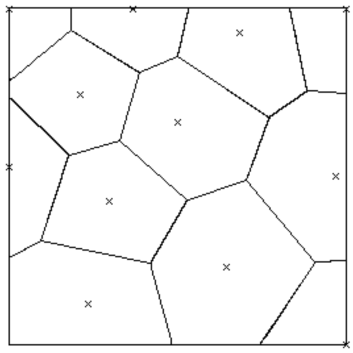

Figure 1: Bounded Voronoi diagram of 12 sites in the plane.

2.2 Farthest point sampling

Farthest point sampling is based on the idea of repeatedly placing the next sample point in the middle of the least-known area of the sampling domain. In the following, we summarise the reasoning underlying this approach for both the uniform and non-uniform case presented in [6, 7].

Starting with the uniform case, Eldar et al. [6, 7] consider the case of an image representing a continuous stochastic process featuring constant first and second order central moments with the third central moment, i.e., the covariance, decreasing exponentially with spatial distance. That is, given a pair of sample points $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$, the points' correlation, $E(p_i, p_j)$, is assumed to decrease with the Euclidean distance, d_{ij}, between the points

$$E(p_i, p_j) = \sigma^2 e^{-\lambda d_{ij}} \tag{5}$$

with $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

Based on their linear estimator, the authors subsequently put forward the following representation for the mean square error, i.e., the deviation from the "ideal" image resulting from estimation error, after the Nth sample

$$\varepsilon^2(p_0, \ldots, p_{N-1}) = \iint \sigma^2 - U^T R^{-1} U \, dx \, dy \tag{6}$$

where

$$R_{ij} = \sigma^2 e^{-\lambda\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}$$

and

$$U_i = \sigma^2 e^{-\lambda\sqrt{(x_i-x)^2+(y_i-y)^2}}$$

for all $0 \leq i, j \leq N-1$. The assumption of stationary first and second order central moments has therefore yielded the result that the expected mean square (reconstruction) error depends on the location of the $N+1$th sample only. Since stationarity implies that the image's statistical properties are spatially invariant and given that point correlations decrease with distance, uniformly choosing the $N+1$th sample point to be that point which is farthest away from the current set of sample points therefore represents the optimal sampling approach within this framework.

This sampling approach is intimately linked with the incremental construction of a Voronoi diagram over the image domain. To see this, note that the point farthest away from the current set of sample sites, S, is represented by the centre of the largest circle empty of any site $s_i \in S$. Eldar et al. [7] show that in the case of farthest point sequences, the centre of such a circle is given by a vertex of the bounded Voronoi diagram of S, $BVD(S)$. Thus, as indicated in figure 2, incremental (bounded) Voronoi diagram construction provides sample points progressively.

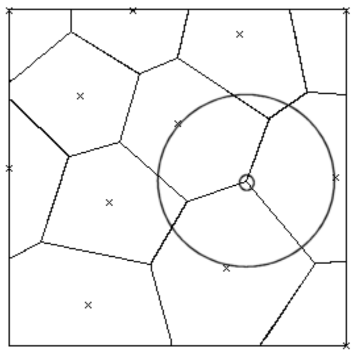

Figure 2: The next farthest point sample (here: sample point 13) is located at the centre of the largest circle empty of any other sample site.

From visual inspection of images it is clear that usually not only the sample covariances but also the sample means and variances vary spatially across an image. When allowing for this more general variability and thus turning to the design of a non-uniform, adaptive sampling strategy, the assumption of sample point covariances decreasing, exponentially or otherwise, with point distance remains valid. However, since Voronoi diagrams in non-uniform metrics may lose favourable properties such as cell connectedness, Eldar et al. [6, 7] opt for the non-optimal choice of augmenting their model by an application-dependent weighting scheme for the vertices in the Euclidean Voronoi diagram.

2.3 Fast Marching

Fast Marching represents a very efficient technique for the solution of front propagation problems which can be formulated as boundary value partial differential equations. We show that the problem of computing the distance map across a sampling domain can be

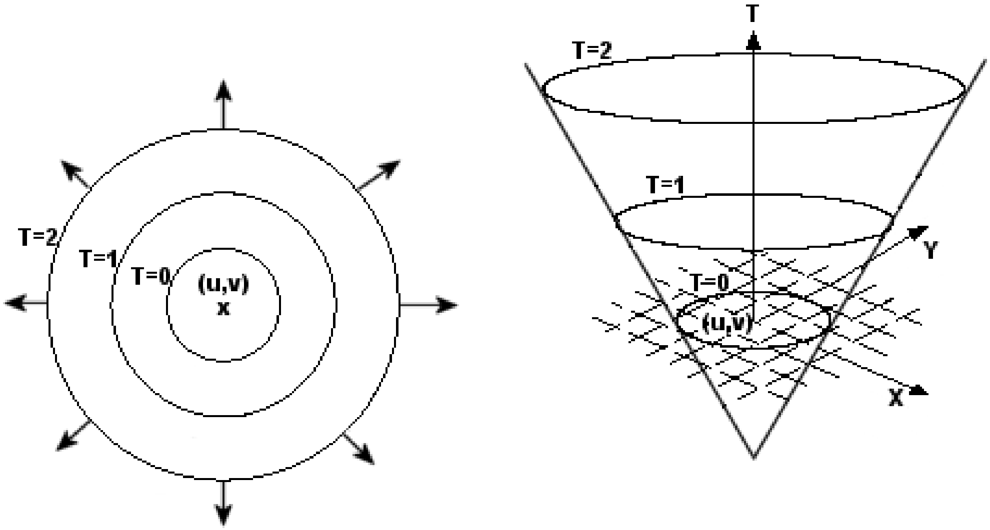

Figure 3: The problem of determining the distance map originating from a point (u,v) (left) formulated as a boundary value problem (right) solvable using Fast Marching.

posed in the form of such a partial differential equation and outline the Fast Marching approach towards approximating its solution.

For simplicity, take the case of an interface propagating with speed function $F(x, y)$ away from a source (boundary) point (u, v) across a planar Euclidean domain. When interested in the time of arrival, $T(x, y)$, of the interface at grid point (x, y), i.e., the distance map T given source point (u, v), the relationship between the magnitude of the distance map's gradient and the given weight $F(x, y)$ at each point can be expressed as the following boundary value formulation

$$|\nabla T(x, y)| = F(x, y) \tag{7}$$

with boundary condition $T(u, v) = 0$.
That is, the distance map gradient is proportional to the weight function. The problem of determining a weighted distance map has therefore been transformed into the problem of solving a particular type of Hamilton-Jacobi partial differential equation, the Eikonal equation [9, 15] (figure 3). For $F(x, y) > 0$, this type of equation can be solved for $T(x, y)$ using Fast Marching.

Since the Eikonal equation is well-known to become non-differentiable through the development of corners and cusps during propagation, the Fast Marching method considers only upwind, entropy-satisfying finite difference approximations to the equation thereby consistently producing weak solutions. As an example for a first order appromixation to the gradient operator, consider [17]

$$\begin{aligned}&[max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 + \\ &max(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0)^2]^{1/2} \quad = F_{ij}\end{aligned} \tag{8}$$

where $F_{ij} \equiv F(i\Delta x, j\Delta y)$. $D_{ij}^{-x}T \equiv \frac{T_{ij} - T_{i-1j}}{h}$ and $D_{ij}^{+x}T \equiv \frac{T_{i+1j} - T_{ij}}{h}$ are the standard backward and forward derivative approximation with h representing the grid spacing;

equivalently for $D_{ij}^{-y}T$ and $D_{ij}^{+y}T$. T_{ij} is the discrete approximation to $T(i\Delta x, j\Delta y)$ on a regular quadrilateral grid.

This upwind difference approximation implies that information propagates from smaller to larger values of T only, i.e., a grid point's arrival time gets updated by neighbouring points with smaller T values only. This monotonicity property allows for the maintenance of a narrow band of candidate points around the front representing its outward motion. The property can further be exploited for the design of a simple and efficient algorithm by freezing the T values of existing points and subsequently inserting neighbouring ones into the narrow band thereby marching the band forward. The basic Fast Marching algorithm can thus be summarised as follows [20, 21]:

0) Mark an initial set of grid points as ALIVE. Mark as CLOSE, all points neighbouring ALIVE points. Mark all other grid points as FAR.

1) Let TRIAL denote the point in CLOSE featuring the smallest arrival time. Remove TRIAL from CLOSE and insert it in ALIVE.

2) Mark all neighbours of TRIAL which are not ALIVE as CLOSE. If applicable, remove the neighbour under consideration from FAR.

3) Using the gradient approximation, update the T values of all neighbours of TRIAL using only ALIVE points in the computation.

4) Loop from 1).

Arrangement of the elements in CLOSE in a min-heap [18] leads to an $O(N \log N)$ implementation, with N representing the number of grid points. Note that a single min-heap structure may be used to track multiple propagation fronts originating from different points in the domain.

Unlike other front propagation algorithms [4], each grid point is only touched once, namely when it is assigned its final value. Furthermore, the distance map $T(x, y)$ is computed with "sub-pixel" accuracy, the degree of which varies with the order of the approximation scheme and the grid resolution. In addition, the distance map is computed directly across the domain, a separate binary image providing the source points is not required. Finally, since the arrival time information of a grid point is only propagated in the direction of increasing distance, the size of the narrow band remains small. Therefore, the algorithm's complexity is closer to the theoretical optimum of $O(N)$ than $O(N \log N)$ [20].

Although the algorithm was presented in the context of a planar orthogonal grid, it can easily be extended to the case of triangulated domains in 2D or 3D by modifying the gradient approximation. Suitable upwind approximations are provided by Barth and Sethian [2] and Kimmel and Sethian [8]. As shown by Kimmel and Sethian [9] this property can be exploited for the $O(N \log N)$ computation of (geodesic) Voronoi diagrams either in the plane or directly on the surface of a curved manifold.

3 Fast Marching farthest point sampling

For simplicity, we first consider the formulation of our Fast Marching farthest point sampling algorithm for a uniform metric and a planar domain.

Starting with an initial sample point set S, we compute $BVD(S)$ by simultaneously propagating fronts from each of the initial sample points outwards. This process is equivalent to the computation of the Euclidean distance map across the domain given S and is achieved by solving the Eikonal equation with $F(x, y) = 1$, for all x, y, and using a single min-heap.

The vertices of $BVD(S)$ are given by those grid points entered by three or more propagation waves (or two for points on the domain boundary) and are therefore obtained as a by-product of the propagation process. The Voronoi vertices' arrival times are inserted into a max-heap data structure. The algorithm then proceeds by extracting the root from the max-heap, the grid location of which represents the location of the next farthest point sample. The sample is inserted into $BVD(S)$ by resetting its arrival time to zero and propagating a front away from it. The front will continue propagating until it hits grid points featuring lower arrival times and thus belonging to a neighbouring Voronoi cell. The T values of updated grid points are updated correspondingly in the max-heap using back pointers. New and obsolete Voronoi vertices are inserted or removed from the max-heap respectively. The algorithm continues extracting the root from the max-heap until it is empty or the sample point budget has been exhausted.
By allowing $F(x, y)$ to vary with any weights associated with points in the domain, this algorithm is easily extended to the case of non-uniform, adaptive sampling.

FastFPS for planar domains
FastFPS for planar domains can be summarised as follows

0) Given an initial sample set S, $n = |S| \geq 1$, compute $BVD(S)$ by propagating fronts with speed F_{ij} from the sample points outwards. Store the Voronoi vertices' arrival times in a max-heap.

1) Extract the root from the max-heap to obtain s_{n+1}. $S' = S \cup \{s_{n+1}\}$. Compute $BVD(S')$ by propagating a front locally from s_{n+1} outwards using Fast Marching and a finite difference approximation for planar domains (such as (8)).

2) Correct the arrival times of updated grid points in the max-heap. Insert the vertices of $BV(s_{n+1}, S')$ in the max-heap. Remove obsolete Voronoi vertices of the neighbours of $BV(s_{n+1}, S')$ from the max-heap.

3) If neither the max-heap is empty nor the point budget has been exhausted, loop from 1).

This algorithm is conceptually similar to Eldar et al. [7] but features a more natural and consistent augmentation to the more interesting case of adaptive progressive sampling.

Extracting the root from, inserting into and removing from the max-heap with subsequent re-heapifying are $O(\log M)$ operations, where M represents the number of elements in the heap. M is $O(N)$, N representing the number of grid points. The updating of existing max-heap entries is $O(1)$ due to the use of back pointers from the grid to the heap. The detection of a (bounded) Voronoi cell's vertices and boundary is a by-product of the $O(N \log N)$ front propagation[1]. Thus, the algorithm's asymptotic efficiency is $O(N \log N)$.

FastFPS for triangulated domains

Certain applications such as the sampling of dense range maps or 3D object surfaces require the extension of the FastFPS principle to curved manifolds. Since triangulated domains may be more readily available on these surfaces than orthogonal rectilinear coordinate systems, we use Fast Marching for triangulated domains [8, 9] to compute geodesic Voronoi diagrams across the surface thereby generalising the FastFPS principle to the case of curved manifolds.

Consequently, the algorithm no longer considers points in an orthogonal grid but vertices in a triangulated domain. Front propagation occurs directly on the surface with F being a positive constant (uniform) or varying with any cost associated with the surface points (non-uniform). This means gradient approximations such as (8) for the planar case are generally no longer applicable and a suitable monotone and consistent finite difference approximation to the Eikonal equation converging to a weak solution needs to be used. Suitable approximations can be found in [2, 8].
The outline of FastFPS for triangulated domains is as follows

0) Given an initial sample set S, $n = |S| \geq 1$, compute $BVD(S)$ by propagating fronts with speed F_{ijk}, $F_{ijk} \equiv F(i\Delta x, j\Delta y, k\Delta z)$, from the sample points outwards using Fast Marching and a finite difference approximation for triangulated domains [2, 8].[2] March along the triangles and linearly interpolate the intersection curve between pairs of distance maps of different origin across each triangle [9]. Store the Voronoi vertices' arrival times in a max-heap.

1) Extract the root from the max-heap to obtain s_{n+1}. $S' = S \cup \{s_{n+1}\}$. Compute $BVD(S')$ by propagating a front locally from s_{n+1} outwards. March the triangles touched by this local update procedure and interpolate the intersection curves.

2) Correct the arrival times of updated grid points in the max-heap. Insert the vertices of $BV(s_{n+1}, S')$ in the max-heap. Remove obsolete Voronoi vertices of the neighbours of $BV(s_{n+1}, S')$ from the max-heap.

3) If neither the max-heap is empty nor the point budget has been exhausted, loop from 1).

Triangle marching and the linear interpolation of the intersection curves are $O(N)$ processes, N representing the number of triangle vertices. The remaining operations are as in FastFPS for planar domains so the overall complexity of FastFPS for curved manifolds is $O(N \log N)$.

[1] As shown by Eldar et al. [7], in the case of farthest point sequences, the number of a Voronoi polygon's neighbours has a low constant upper bound making boundary construction extremely fast.

[2] For approximation consistency reasons, Fast Marching for triangulated domains requires any obtuse triangles in the given surface triangulation to be split into acute triangles during a preprocessing step. Although this preprocessing step does not affect the algorithm's asymptotic efficiency, it affects its accuracy [8].

4 Worked examples

Starting with the application of FastFPS for planar domains to progressive image sampling, we use adaptive FastFPS for planar domains to generate sparse and effective representations of the (512x512) Lena and Mandrill test images. Points are weighted by an estimate of the local intensity gradient thereby favouring the sampling of points in regions of relatively higher frequency. Since FastFPS provides us with a Voronoi diagram representation across the sampling domain, the FastFPS point sets can immediately be used for the k nearest neighbour reconstruction of the images. Figure 4 shows the Voronoi diagram and sample point sets generated by adaptive FastFPS for planar domains. The image intensity gradient is approximated using the derivative of a Gaussian filter. Points are clearly more densely sampled in regions of relatively high variation of frequency. Figure 4 also presents the corresponding 4 nearest neighbour reconstructions.

We apply uniform FastFPS for triangulated domains to the problem of sampling the "David" object surface for mesh decimation and/or progressive transmission. The FastFPS point sets, corresponding meshes and their renderings for different sample point budgets are presented in figure 5. The cluster-free point sets fill the space uniformly and irregularly thereby suppressing any noticeable aliasing effects and allowing for both high-quality renderings early on into the sequence and significantly decimated model sizes.

A thorough experimental analysis of the algorithm's execution and memory efficiency will be reported elsewhere. Table 1, however, gives an indication of the algorithm's speed on a AMD Athlon 1.3 Ghz, 256MB, Windows 2000 machine.

Model	*Model size*	*Sample size*	*Secs.*
LENA image	(256x256)	30k	1.21
MANDRILL image	(348x348)	40k	2.45
PEPPERS image	(512x512)	50k	4.24
BUNNY mesh	35947	30k	0.72
DAVID mesh	99455	40k	1.43
DRAGON mesh	184018	50k	3.89

Table 1: Uniform FastFPS sampling times.

5 Conclusions

We presented a novel, efficient and easily implementable progressive farthest point sampling technique based on the incremental construction of discrete Voronoi diagrams across planar domains or triangulated surfaces using Fast Marching level set methods. The main benefits of this technique are, firstly, that it works equally well in the uniform and adaptive case. Secondly, FastFPS is applicable to planar domains and surfaces in triangulated, point cloud or implicit form. We demonstrated the quality and speed of computation of FastFPS point sets for the generation of effective sparse image and 3D surface representations. Possible applications are numerous and include image encoding and compression, image registration, progressive transmission and rendering of 2D and 3D surfaces, progressive acquisition of 3D surfaces, point-based multiresolution representation, etc. The

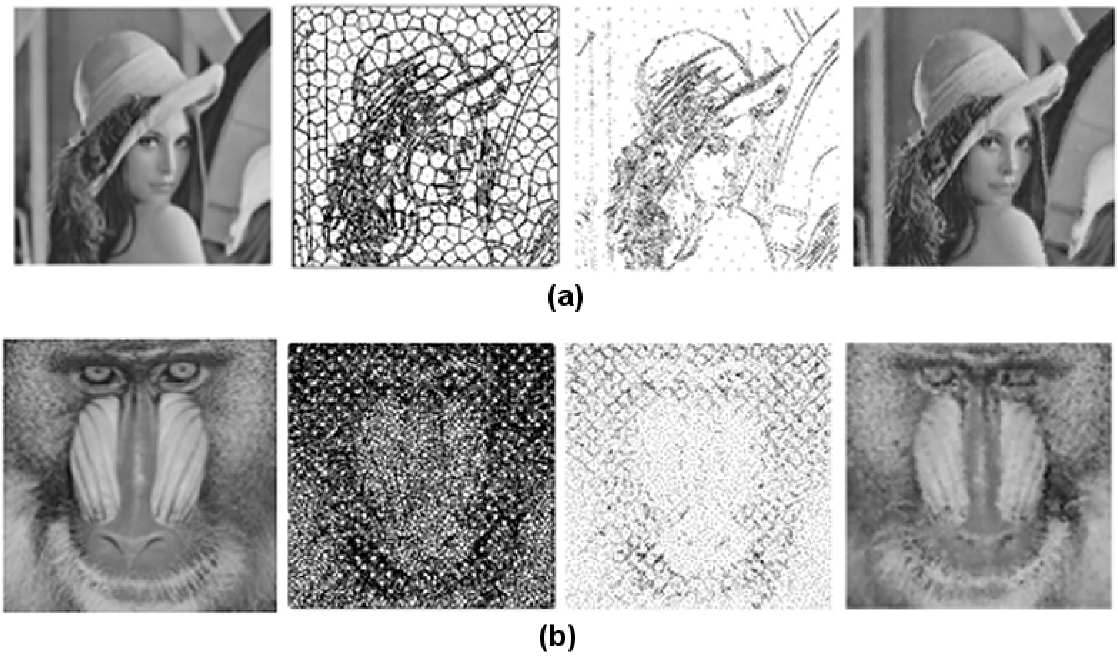

Figure 4: (512x512) Lena (a) and Mandrill (b) test images, the Voronoi diagrams and point sets produced by adaptive FastFPS for planar domains and the 4-nearest neighbour reconstructions (from left to right). Sample size: 3.1% (8k)

availability of the Voronoi diagram of the sample set further facilitates certain image and surface processing (natural neighbour [22, 23, 24], nearest neighbour, Delaunay triangulation reconstructions), encoding/compression [16] and registration [4] applications.

Surfaces in 3D may not be readily available in triangulated form or the computation of a triangulation may be undesirable since it may otherwise not be required by the application. This is, for example, the case when dealing with implicit surfaces. Apart from this, the need for the splitting of any obtuse into acute triangles as part of the FastFPS algorithm for triangulated domains is undesirable. Although this preprocessing step does not add significantly to the algorithm's computational complexity, it undermines its accuracy. This comes in addition to the uncertainty generally associated with numerical analysis over polygonal surfaces [5]. Numerical analysis on Cartesian grids, by contrast, represents a more robust technique.

For future research, we therefore intend to incorporate Mémoli and Sapiro's [10] extension of the Fast Marching technique to implicit surfaces into the FastFPS concept. This extension works on Cartesian grids, retains the efficiency of the conventional Fast Marching approach and has the additional benefit of being applicable to both triangulated or unorganised point sets. Mémoli and Sapiro [11] use their extended Fast Marching technique to produce Voronoi diagrams directly across point clouds without the need for any prior reconstruction of the underlying manifold. By incorporating this technique into the FastFPS concept, surfaces in 3D would no longer necessarily need to be processed in their triangulated form, acute or otherwise.

Acknowledgements

We would like to thank Michael Blain, Marc Cardle, Mark Grundland and Malcolm Sabin for numerous helpful discussions. Michael Lindenbaum kindly provided us with a copy of Eldar's M.Sc. thesis [6]. The David model was obtained from the Stanford Michelangelo project website.

References

[1] F. Aurenhammer. Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys*, **23**(3):345–405, 1991.

[2] T. J. Barth and J. A. Sethian. Numerical Schemes for the Hamilton-Jacobi and Level Set Equations on Triangulated Domains. *Journal of Computational Physics*, **145**(1):1–40, 1998.

[3] R. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, **5**(1):51–72, 1986.

[4] O. Cuisenaire. Distance Transformations: Fast Algorithms and Applications to Medical Image Processing. *Ph.D. thesis, Université catholique de Louvain, Laboratoire de Telecommunications et Teledetection*, Belgium, 1999.

[5] M. Desbrun, M. Meyer, P. Schröder and A. Barr. Discrete-differential operators in nD. *California Institute of Technology/USC Report*, 2000.

[6] Y. Eldar. Irregular Image Sampling Using the Voronoi Diagram. *M.Sc. thesis, Technion - IIT*, Israel, 1992.

[7] Y. Eldar, M. Lindenbaum, M. Porat and Y. Y. Zeevi. The Farthest Point Strategy for Progressive Image Sampling. *IEEE Trans. on Image Processing*, **6**(9):1305–1315, 1997.

[8] R. Kimmel and J. A. Sethian. Computing Geodesic Paths on Manifolds. *Proc. of the Nat. Acad. of Sciences*, **95**(15):8431–8435, 1998.

[9] R. Kimmel and J. A. Sethian. Fast Voronoi Diagrams and Offsets on Triangulated Surfaces. *Proc. of AFA Conf. on Curves and Surfaces*, Saint-Malo, France, 1999.

[10] F. Mémoli and G. Sapiro. Fast Computation of Weighted Distance Functions and Geodesics on Implicit Hyper-Surfaces. *Journal of Computational Physics*, **173**(1):764–795, 2001.

[11] F. Mémoli and G. Sapiro. Distance Functions and Geodesics on Point Clouds. *Technical Report 1902, Institute for Mathematics and its Applications, University of Minnesota*, USA, 2002.

[12] C. Moenning and N. A. Dodgson. Fast Marching farthest point sampling for point clouds and implicit surfaces. *Technical Report 565, Computer Laboratory, University of Cambridge*, UK, 2003.

[13] I. Notkin and C. Gotsman. Parallel progressive ray-tracing. *Computer Graphics Forum*, **16**(1):43–55, 1997.

[14] A. Okabe, B. Boots and K. Sugihara. *Spatial Tessellations - Concepts and Applications of Voronoi Diagrams*. 2nd ed. John Wiley & Sons, Chicester, 2000.

[15] S. J. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, **79**(1):12–49, 1988.

[16] H. Rom and S. Peleg. Image Representation Using Voronoi Tessellation: Adaptive And Secure. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, pages 282–285, 1988.

[17] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal of Numerical Analysis*, **29**(3):867–884, 1992.

[18] R. Sedgewick. *Algorithms in C*. 3rd ed. Addison-Wesley, Reading, USA, 1998.

[19] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. of the Nat. Acad. of Sciences*, **93**(4):1591–1595, 1996.

[20] J. A. Sethian. *Level Set Methods and Fast Marching Methods - Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. 2nd ed. Cambridge University Press, Cambridge, UK, 1999.

[21] J. A. Sethian. Fast Marching Methods. *SIAM Review*, **41**(2):199–235, 1999.

[22] R. Sibson. A vector identity for the Dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society*, **87**(1):151–155, 1980.

[23] R. Sibson. A brief description of the natural neighbor interpolant. In: *Interpreting Multivariate Data*, John Wiley & Sons, Chicester, pages 21-36, 1981.

[24] K. Sugihara. Surface interpolation based on new local coordinates. *Computer-Aided Design*, **31**:51–58, 1999.

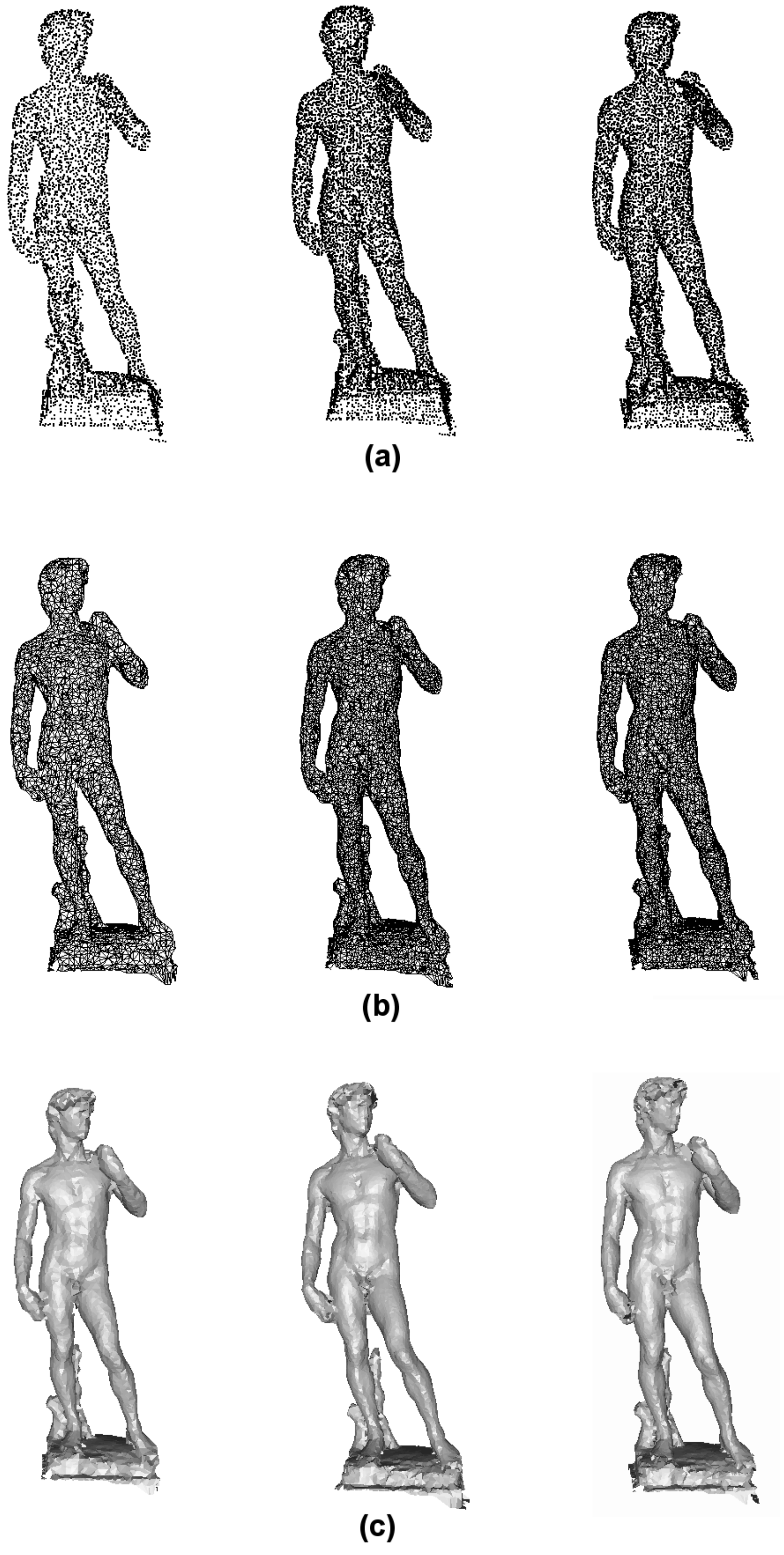

Figure 5: Points sets (a) produced by uniform FastFPS for triangulated domains, the corresponding meshes (b) and their renderings (c).
Sample size: (left) 8.2% (8k), (middle) 20.6% (20k), (right) 41.2% (40k)