Fast Marching farthest point sampling

Carsten Moenning (cm230@cl.cam.ac.uk)
Neil A. Dodgson (nad@cl.cam.ac.uk)

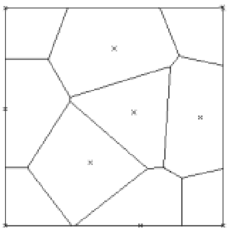
UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Introduction

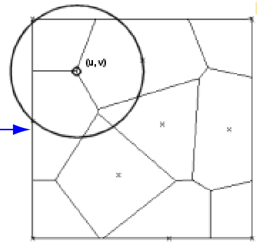
We introduce the Fast Marching farthest point sampling (FastFPS) approach for the *progressive sampling of planar domains and curved manifolds in triangulated, point cloud or implicit form*. We use Fast Marching methods [3,4,5,8] for the incremental computation of distance maps across the sampling domain and obtain a farthest point sampling technique which performs equally well in both the uniform and the adaptive case. Unlike similar previous sampling schemes [1,2], it is equally efficiently applicable to both images and higher dimensional surfaces.

The sampling principle

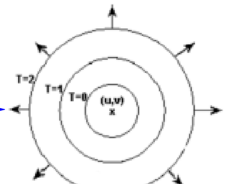
FastFPS is based on the idea of repeatedly placing the next sample point in the middle of the least-known area of the sampling domain. For simplicity, take the bounded discrete Voronoi diagram representation of a planar domain (a). The point farthest away from all other points, and thus the next FastFPS sample, is given by the centre of the largest circle empty of any other point. In the case of farthest point sample sets, this centre necessarily coincides with a vertex of the bounded Voronoi diagram (b) [1]. This new sample point is inserted into the diagram and Fast Marching is used to locally update the Voronoi diagram accordingly (c,d,e). Thus, incremental discrete Voronoi diagram construction yields FastFPS samples *progressively*. We use recent extensions [3,4,5] to the basic Fast Marching algorithm [8] to apply the farthest point principle to domains in *triangular*, *point cloud* and *implicit form* [6] without any loss in efficiency. By varying the speed of the local front propagation with, for example, local curvature or local similarity measures, this scheme inherently supports *adaptive sampling*.



(a) The bounded (discrete) Voronoi diagram (BVD) of n sample points in the plane (above) and the corresponding equal distance contours (below).



(b) Sample point n+1 is located at the centre of the largest circle empty of any other sample points. This will always be a vertex of BVD.



(c) The BVD is updated locally by propagating a front (here: with unit speed) from the new sample point outwards.

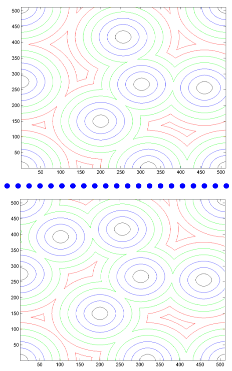
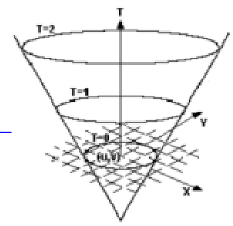
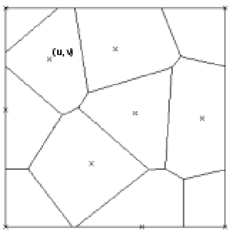


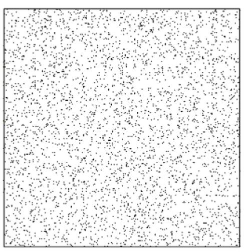
Figure 1: The Fast Marching farthest point sampling principle applied on a regular quadrilateral grid.



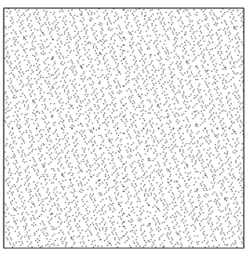
(d) Front propagation is posed as a boundary PDE, the solution to which can be approximated very efficiently using Fast Marching techniques [3,4,5,8]. This is equivalent to distance mapping directly across the domain irrespective of its representation.



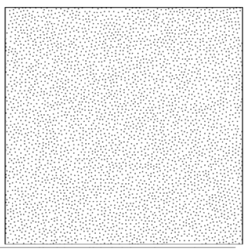
(e) The BVD (bottom) and the underlying distance map (top) have been updated. If applicable, the process continues from step (a).



(a) Randomly sampled point set: The distribution is characterised by clusters and large holes.



(b) Quasi-randomly (Halton) sampled point set: Points are still placed arbitrarily closely to each other.



(c) Farthest point-sampled point set: The sampling domain is filled irregularly uniformly without any clusters or holes of significant size.

Figure 2: Distributions of randomly (a), quasi-randomly (b) and farthest point-sampled (c) point sets.

Properties

(a) Applicability

The algorithm can be applied to sample domains in regular grid, triangular, point cloud or implicit form. This allows for numerous applications such as surface reconstruction from unorganised point clouds, progressive transmission & rendering, remeshing, resampling, point cloud & mesh simplification [7], multiresolution representations, implicit surface sampling.

(b) Efficiency

The algorithm is $O(N \log N)$, N representing the number of grid points or triangle vertices, irrespective of the domain representation.

(c) Anti-aliasing

Farthest point sequences feature excellent anti-aliasing properties documented by a "blue noise" power spectral density, i.e. aliasing effects are traded for high frequency noise [1].

(d) Irregular uniform point set distribution

Farthest point sequences are deterministic and minimise the maximum distance between sample points. This yields cluster- and hole-free point distributions contributing to property (c) [2].

References

[1] Y. Eldar, M. Lindenbaum, M. Porat and Y. Y. Zeevi. The Farthest Point Strategy for Progressive Image Sampling. *IEEE Transactions on Image Processing*, **6**(9):1305–1315, 1997.
[2] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, **38**:293–306, 1985.
[3] R. Kimmel and J. A. Sethian. Computing Geodesic Paths on Manifolds. *Proceedings of the National Academy of Sciences*, **95**(15):8431–8435, 1998.
[4] F. Memoli and G. Sapiro. Fast Computation of Weighted Distance Functions and Geodesics on Implicit Hyper-Surfaces. *Journal of Computational Physics*, **173**(1):764–795, 2001.
[5] F. Memoli and G. Sapiro. Distance Functions and Geodesics on Point Clouds. Technical Report 1902, IMA, University of Minnesota, USA, 2002.
[6] C. Moenning and N. A. Dodgson. Fast Marching farthest point sampling for implicit surfaces and point clouds. Computer Laboratory Tech. Rep't No. 565, University of Cambridge, UK, 2003.
[7] C. Moenning and N. A. Dodgson. A new point cloud simplification algorithm. *Proc. 3rd IASTED International Conference on Visualization, Imaging and Image Processing*, to appear, 2003.
[8] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. 2nd ed. Cambridge University Press, Cambridge, UK, 1999.

RAINBOW RESEARCH GROUP

Application to planar domains

(a) The original "Mandrill" and "Peppers" test images. (http://www.bragzone.com).

(b) Point set produced by FastFPS using a similarity measure based on the CLAB colour space to sample the image *adaptively* at 3.1% (8k).

(c) As in (b) but for a sample budget of 6.2% (16k).

(d) Four nearest neighbour reconstruction of the *uniformly* sampled test image using FastFPS for planar domains with a 3.1% (8k) sample budget.

(e) Four nearest neighbour reconstruction using the point set in (b).

(f) Four nearest neighbour reconstruction using the point set in (c).

Figure 3: FastFPS for planar domains — example.

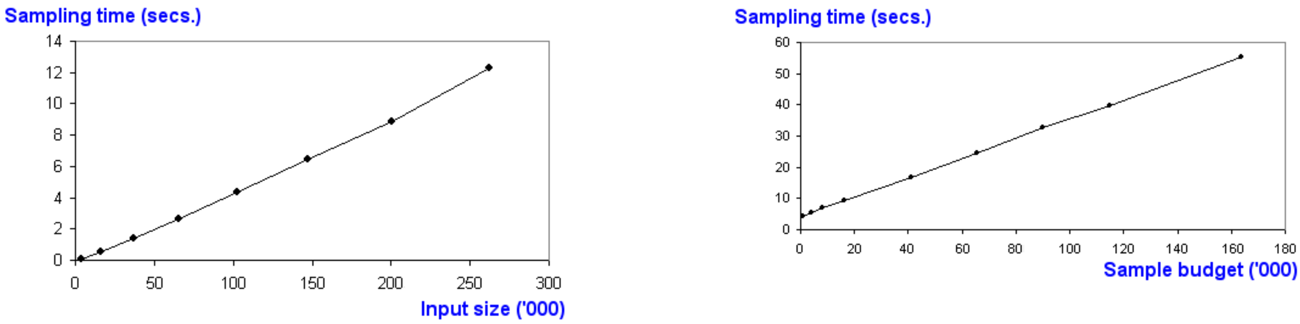
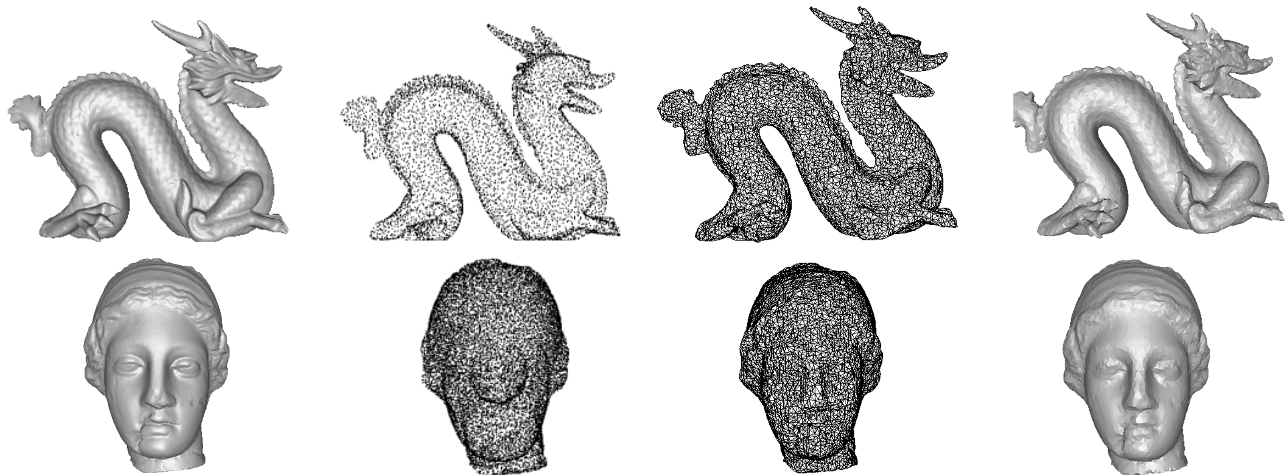


Figure 4: FastFPS for planar domains — execution efficiency.
The algorithm's execution performance on a moderately specified PC (Pentium III 700 MHz, 512 MB, Windows machine) as a function of the size of the input image with a constant 10% sample budget (left) and as a function of the sample budget for a constant (512x512) input image (right).

Application to triangular meshes



(a) The original "Dragon" and "Venus" objects (http://www.cyberware.com).

(b) 3D point sets produced by uniform FastFPS for triangular domains with a 4.3% (8k, Dragon) and 10.0% (14k, Venus) sample budget.

(c) Triangular meshes corresponding to (b).

(d) Shaded renderings of meshes in (c).

Figure 5: FastFPS for triangular domains — example.

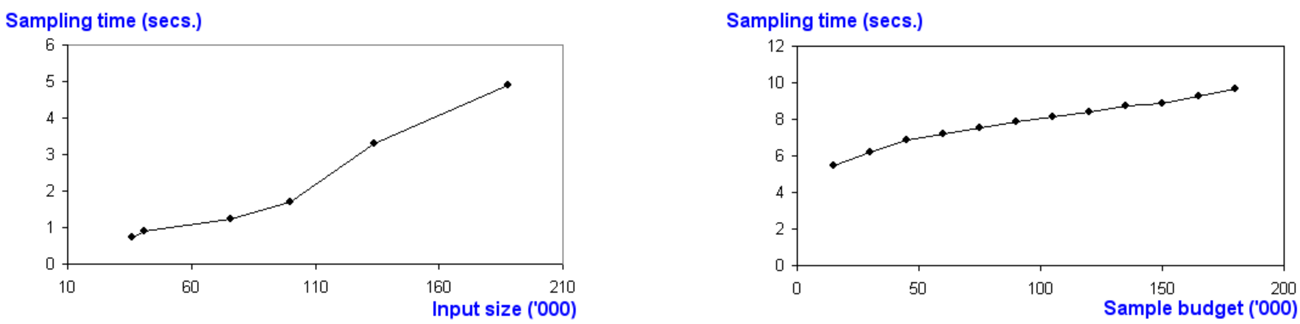


Figure 6: FastFPS for triangular domains — execution efficiency.
The algorithm's execution performance on a moderately specified PC (Pentium III 700 MHz, 512 MB, Windows machine) as a function of the size of the input model with a constant 10% sample budget (left) and as a function of the sample budget for a constant input model (right).