An Improved Content-Adaptive Mesh-Generation Method for Image Representation

Michael D. Adams

Department of Electrical and Computer Engineering
University of Victoria
Victoria, BC, V8W 3P6, Canada
E-mail: mdadams@ece.uvic.ca

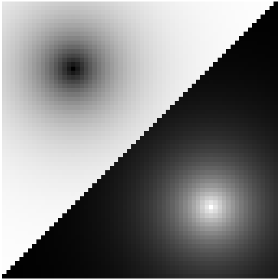
IEEE ICIP
September 26–29, 2010
Hong Kong

Outline

Introduction

- growing interest in geometric representations of images, especially those based on triangle meshes
- triangle-mesh representations of images have many advantages, including:
  1. trivialize application of affine transformations (e.g., rotations, scaling, shears, translations) to images
  2. greatly simplify image interpolation
  3. facilitate easier handling of image domains with arbitrary polygonal shape (i.e., not necessarily rectangular)
- such representations useful in many diverse areas, including:
  - filtering, restoration
  - tomographic reconstruction
  - pattern recognition, feature detection
  - computer vision
  - image/video compression
- constructing triangle-mesh representation of image is challenging task
- want mesh-generation method that produces mesh of high quality (i.e., low approximation error) while requiring minimal computation and memory

Conceptual Model for Image



Image

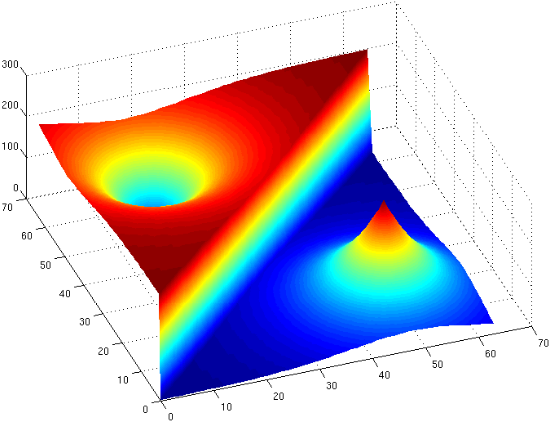
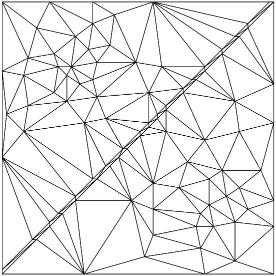
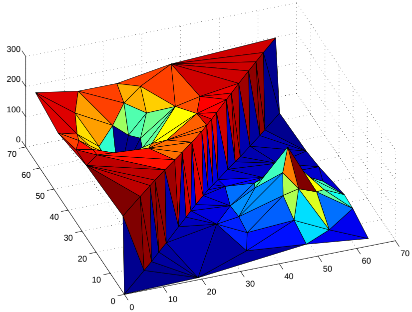


Image Modelled as Surface

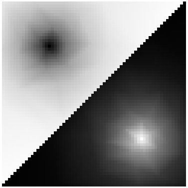
Mesh Approximation of Image (Sampling Density 2.5%)



Triangulation of Image Domain



Resulting Triangle Mesh



Reconstructed Image

Error Diffusion (ED) Method (Yang, Wernick, and Brankov)

- for image f sampled at points in set I, uses Floyd-Steinberg error diffusion to generate set S of N sample points distributed such that local density of sample points at each point $(x,y) \in I$ proportional to largest magnitude second-order directional derivative of f at (x,y)
- in more detail, algorithm consists of following steps:
  1. From f, compute the sample-point density function d defined on I given by $d(x,y) = \tilde{d}(x,y)/\tilde{d}_{\max}$, where $\tilde{d}_{\max} = \max_{(x,y) \in I} \tilde{d}(x,y)$, and $\tilde{d}(x,y)$ is the maximum magnitude second-order directional derivative of f at (x,y).
  2. Initially, set the threshold τ to use for Floyd-Steinberg error diffusion to be $\tau_0 = \frac{1}{2N} \sum_{(x,y) \in I} d(x,y)$.
  3. Convert d to a binary-valued function b using nonleaky Floyd-Steinberg error diffusion with the threshold τ and a serpentine scan order.
  4. Set S to the set of all points (x,y) for which $b(x,y) \neq 0$. Then, let $S := S \cup H$, where H is the set of the (four) extreme convex hull points of I.
  5. If $|S|$ is close enough to N, stop; otherwise, adjust τ appropriately (i.e., if $|S| > N$, increase τ; if $|S| < N$, decrease τ) and go to step 3.
- explicit construction of mesh not formally part of algorithm
- extremely fast and requires minimal memory
- derivatives computed by convolution, raising issue of noise suppression (via lowpass filtering) and how to handle image boundaries

ED Method: Smoothing

- considered: 1) Gaussian smoothing (with various choices of standard deviation parameter σ) as well as 2) no smoothing
- Gaussian smoothing with $\sigma = 1$ found to perform better than no smoothing, typically by about 0.75 to 2.75 dB
- no smoothing tends to result in more uniform distribution of sample points, due to spurious large-magnitude derivatives caused by noise
- effect of smoothing on choice of sample points illustrated below for `lena` image at sampling density of 8%

No Smoothing

Gaussian Smoothing

ED Method: Boundary Handling

- for boundary handling, considered: 1) zero extension, 2) constant extension, and 3) symmetric extension
- zero extension found to perform best, regardless of whether smoothing employed, typically by margin of 0.25 to 1.05 dB
- constant and symmetric extension tend not to place sufficient number of points along boundary of image domain
- effect of boundary handling strategy on choice of sample points shown below for `lena` image at sampling density of 8%

Zero Extension

Symmetric Extension

Greedy Point-Removal (GPR) Method (Demaret and Iske)

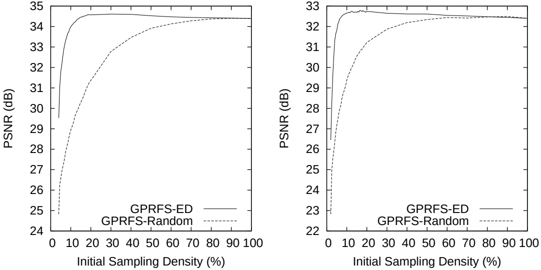
- for image of width W and height H and sampled at points in I, selects set S of N sample points, by first constructing mesh containing all WH sampling points
- each iteration removes point minimizing error increase
- in more detail, algorithm consists of following steps:
  1. Let $S := I$ (hence, $|S| = WH$).
  2. Construct the Delaunay triangulation of S.
  3. If $|S| \leqslant N$, output S and stop.
  4. For each point $p \in S$, compute the increase Δe_p in the squared error of the mesh approximation that is incurred if p is removed from the triangulation.
  5. For the point $p \in S$ that minimizes Δe_p, delete p from the triangulation, and let $S := S \setminus \{p\}$.
  6. Go to step 3.
- step 4 can be implemented efficiently since vertex deletion only has local effect in Delaunay triangulation
- step 5 can be implemented efficiently via heap-based priority queue
- for images of reasonable size, initial mesh size very large, leading to very high computational/memory requirements
- greedy approach unlikely to yield globally optimal solution

Greedy Point-Removal From Subset (GPRFS) Framework

- N: number of sample points to select; S: set of selected sample points; W: image width; H: image height; I: set of sample points of original image
- only differs from GPR algorithm in step 1
- starts with intelligently chosen subset of sample points
- in more detail, algorithm consists of following steps:
  1. Select a subset S_0 of I such that $|S_0| = N_0$ (where $N_0 \in [N, WH]$), and let $S := S_0$.
  2. Construct the Delaunay triangulation of S.
  3. If $|S| \leqslant N$, output S and stop.
  4. For each point $p \in S$, compute the increase Δe_p in the squared error of the mesh approximation that is incurred if p is removed from the triangulation.
  5. For the point $p \in S$ that minimizes Δe_p, delete p from the triangulation, and let $S := S \setminus \{p\}$.
  6. Go to step 3.
- using ED scheme in GPRFS framework to select S_0 yields proposed GPRFS-ED method
- GPRFS-ED method includes GPR and ED schemes as special cases (i.e., $N_0 = N$ and $N_0 = WH$, respectively)

Initial Subset Selection

- use ED scheme to choose initial subset of sample points, but need to choose initial sampling density D_0 (where $D_0 = \frac{N_0}{WH} \in [D, 1]$)
- effect of varying initial sampling density D_0 on mesh quality shown below



`lena` image, $D = 4\%$ `peppers` image, $D = 2\%$

- best mesh quality not obtained when $D = 100\%$ (where GPRFS-ED method becomes equivalent to GPR scheme)
- for sampling density D of practical interest (i.e., $D < 10\%$) GPRFS-ED method usually achieves PSNR very close to GPR scheme if D_0 about $4D$
- choose $D_0 = 4D$

Objective Mesh-Quality Comparison

lena image

Samp. Density (%)	PSNR (dB)		
	GPRFS-ED	GPR	ED
1.0	28.85	**29.11**	22.24
1.5	**30.68**	30.68	24.75
2.0	**31.95**	31.78	26.32
4.0	**34.50**	34.40	29.43
8.0	**37.11**	37.00	32.35

peppers image

Samp. Density (%)	PSNR (dB)		
	GPRFS-ED	GPR	ED
1.0	29.85	**30.05**	22.23
1.5	**31.57**	31.55	24.84
2.0	**32.55**	32.40	26.33
4.0	**34.43**	34.20	29.78
8.0	**36.11**	35.76	32.04

- at sampling densities above 1% (typically required for good quality image reconstructions), GPRFS-ED method fairly consistently outperforms GPR scheme
- at sampling density of 1%, GPRFS-ED method typically produces meshes of slightly lower quality than GPR, but requires about 17 times less computation and 25 times less memory
- GPRFS-ED method vastly superior to ED scheme, so benefit of GPRFS-ED method not solely from its use of ED scheme

Subjective Mesh-Quality Comparison (lena image, sampling density of 2%)

GPRFS-ED (31.95 dB)

GPR (31.78 dB)

- in terms of subjective image quality, GPRFS-ED method produces meshes of quality comparable to (or slightly better than) GPR scheme

Computational-Complexity Comparison

- computational complexity measured in terms of execution time

lena image

Samp. Density (%)	Time (s) GPRFS-ED	Time (s) GPR	Ratio
1	3.47	58.41	16.8
2	5.39	57.39	10.6
4	9.26	56.30	6.0
8	17.37	54.02	3.1

- GPRFS-ED method requires 3 to 17 times less computation than GPR scheme, with difference most pronounced at lower sampling densities
- GPRFS-ED method yields significant computational savings in spite of producing higher quality meshes in most cases

Memory-Complexity Comparison

Comparison of the peak mesh size for the various methods

Method	Peak Mesh Size	Relative Peak Mesh Size				
		General	$D = 1\%$	$D = 2\%$	$D = 4\%$	$D = 8\%$
GPRFS-ED	$4DWH$	1	1	1	1	1
GPR	WH	$\frac{1}{4D}$	25	12.5	6.25	3.125

sampling density D, image width W, image height H

- same data structures used for GPR and GPRFS-ED methods
- memory usage dominated by mesh data structure and priority queue with one entry per mesh vertex
- peak memory usage approximately proportional to peak mesh size (in vertices)
- for sampling densities from 1% to 8%, GPRFS-ED method requires from 25 to 3.125 times less memory than GPR, with difference most pronounced at lower sampling densities
- GPRFS-ED method offers very substantial memory savings

Conclusions

- proposed new content-adaptive mesh-generation method for image representation, known as GPRFS-ED
- our GPRFS-ED method shown to yield better (or comparable) quality meshes in terms of squared error and subjective quality than state-of-the-art GPR method, at only very small fraction of computational and memory costs
- our GPRFS-ED method can easily tradeoff between mesh quality and computational/memory complexity (through choice of initial sampling density D_0)