An Improved Incremental/Decremental Delaunay Mesh-Generation Strategy for Image Representation

Badr Eddine El Marzouki and Michael D. Adams

marzouki@uvic.ca and mdadams@ece.uvic.ca

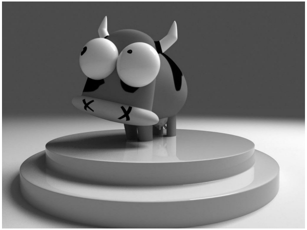
Department of Electrical & Computer Engineering
University of Victoria, Victoria, BC, Canada

August 2017

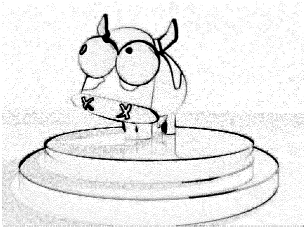
Outline

Motivation

- most images nonstationary
- uniform sampling leads to too few samples in regions of image with high detail (e.g., edges) and too many samples elsewhere
- motivates use of adaptive (nonuniform) sampling
- by making sampling density adaptive to image content, better approximation can be achieved for given number of samples
- one popular approach to adaptive sampling is based on triangle meshes

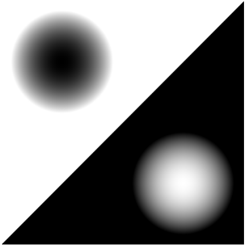


Image



Adaptively-Chosen Sample Points

Triangle-Mesh Models of Images



Original Image

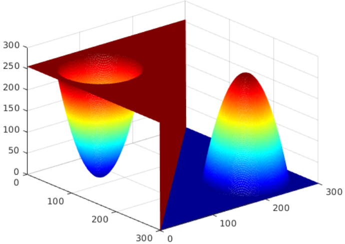
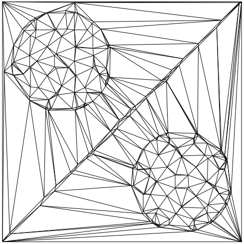
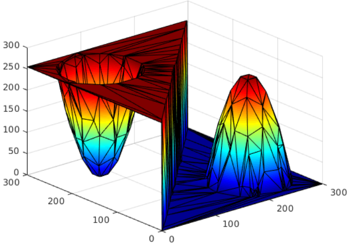


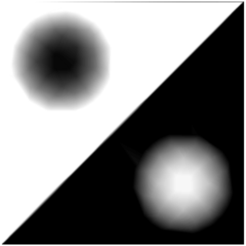
Image Viewed as Surface



Triangulation of Image Domain



Mesh Model



Reconstructed Image

Mesh-Generation Problem

- sampling density: ratio of number of sample points to number of pixels in original image
- in context of our work, mesh model employs:
  - Delaunay triangulation (which is completely determined by sample points)
  - linear interpolant over each face of triangulation
- mesh model completely characterized by:
  - set of sample points
  - set of function values at sample points
- mesh-generation problem: for given image and mesh size (i.e., sampling density), find mesh that best approximates image in terms of mean-squared error (MSE)
- essentially, need to find set of sample points
- want to keep computational cost to minimum

- based on Floyd-Steinberg error diffusion
- adaptively distributes sample points in image domain in proportion to density function
- density function is maximum-magnitude second-order directional derivative (MMSODD)

Image

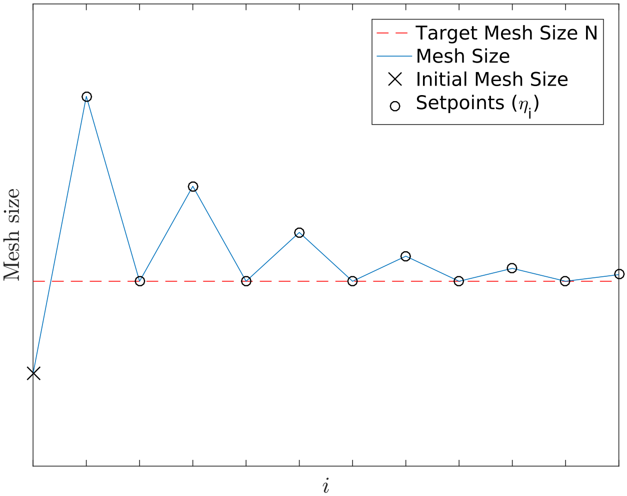
MMSODD

Sample Points

Mesh-Generation Framework

- our work based on incremental/decremental Delaunay mesh-generation framework previously proposed by Adams
- combines advantages of mesh-refinement and mesh-simplification methods
- mesh generation alternates between insertion and deletion of points
- framework has several free parameters
- mesh size evolves in accordance with growth schedule
- growth schedule: sequence of setpoints $\{\eta_i\}_{i=0}^{L-1}$

Growth Schedule: Example



Growth Schedule Example (A')

Mesh-Generation Framework Algorithm

1. select initial mesh points
2. while setpoints remain in growth schedule:
    1. get current setpoint
    2. if mesh size $<$ current setpoint, add points to mesh until mesh size equals setpoint; each point added as follows:
        - select face f^* into which point is to be inserted
        - select point p^* in face f^* to insert
        - insert selected point p^* into the mesh
    3. else if mesh size $>$ current setpoint, remove points from mesh until mesh size equals setpoint; each point deleted as follows:
        - delete point that causes least increase in error
    4. proceed to next setpoint
3. optionally postprocess mesh

Mesh-Generation Framework: Free Parameters

- initial mesh
  - method of selecting sample points
  - size
- growth schedule
- face selection policy for point insertion
- point-in-face selection policy for point insertion
- postprocessing method

Proposed Methods

- propose two methods based on preceding framework which make different trade-offs:
  1. IID1: favors lower computational cost over higher mesh quality
  2. IID2: favors higher mesh quality over lower computational cost
- initial mesh: error diffusion with sampling density of 1%
- growth schedule: A' (mesh size oscillates between target mesh size and exponentially decaying values above it), which is characterized by amplitude A and length L:
  - IID1: A=3 and L=4
  - IID2: A=3 and L=6
- face-selection policy: highest squared error
- point-in-face selection policy:
  - IID1: point with largest MMSODD
  - IID2: approximate local squared-error minimizer (ALSEM)
- postprocessing: bad point replacement (BPR)

Evaluation

- implementation in C++ developed by authors
- compared performance of proposed schemes to four state of the art methods (based on Delaunay mesh and linear interpolant):
  - GPR, IDDT, ID1, and ID2
- for evaluation purposes, used 350 mesh-generation test cases:
  - 50 images (photographic, medical, and computer-generated)
  - 7 sampling densities (0.125% to 4%)
- mesh model generated then approximation error of reconstruction computed (PSNR)
- for each test case, PSNRs obtained with various methods ranked from 1 (best) to 6 (worst)
- computed average ranking (and standard deviation) per sampling density and overall
- for each test case, also measured computation times for various methods

Mesh Quality – Averaged Rankings for 50 Images

Sampling density (%)	Mean Rank (standard deviation in parentheses)					
	IID1	IID2	ID1	ID2	IDDT	GPR
0.125	4.92 (0.57)	**1.36** (0.94)	3.72 (0.86)	2.08 (0.57)	5.64 (1.16)	3.28 (0.83)
0.250	4.74 (0.69)	**1.38** (0.81)	3.68 (1.04)	2.06 (0.68)	5.70 (1.04)	3.44 (0.93)
0.500	4.02 (0.84)	**1.42** (0.84)	3.70 (1.36)	2.04 (0.70)	5.78 (0.91)	4.02 (0.94)
1.000	3.58 (0.95)	**1.38** (0.67)	3.46 (1.28)	2.02 (0.74)	5.70 (1.20)	4.46 (0.99)
2.000	3.12 (0.77)	**1.32** (0.65)	3.38 (1.18)	1.96 (0.73)	5.68 (1.20)	4.64 (1.05)
3.000	3.10 (0.81)	**1.40** (0.83)	3.22 (1.23)	1.98 (0.65)	5.68 (1.20)	4.70 (0.99)
4.000	3.18 (0.80)	**1.38** (0.67)	3.20 (1.26)	1.96 (0.67)	5.68 (1.20)	4.70 (0.99)
Overall	3.81 (1.06)	**1.38** (0.77)	3.48 (1.19)	2.01 (0.67)	5.69 (1.13)	4.18 (1.11)

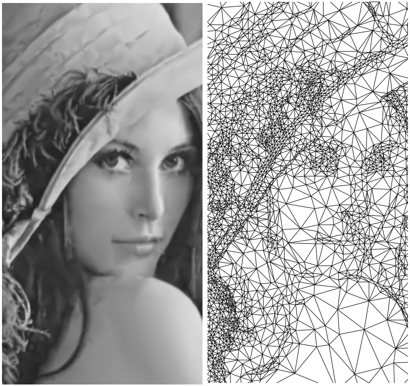
- proposed IID2 consistently performs best compared to other methods
- proposed IID1 better than IDDT and on par with ID1

Mesh Quality – Representative Results

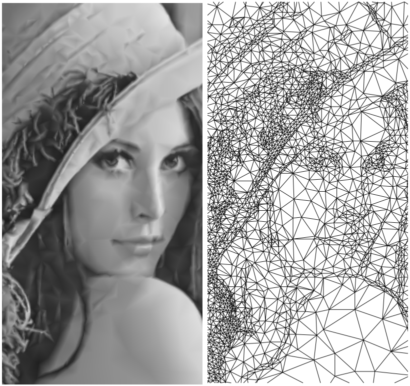
Image	Sampling density (%)	PSNR (dB)					
		IID1	IID2	ID1	ID2	IDDT	GPR
bull	0.125	31.14	34.18	**34.90**	*34.56*	33.85	33.51
	0.250	37.63	**39.15**	*38.87*	38.76	37.51	38.18
	0.500	41.45	**42.24**	41.84	**42.24**	40.42	41.89
	1.000	43.81	*44.22*	43.91	**44.27**	42.50	43.97
	2.000	45.73	*46.09*	45.79	**46.13**	44.46	45.83
	3.000	47.08	**47.37**	47.15	*47.37*	45.78	47.14
	4.000	48.23	**48.44**	48.26	*48.44*	46.97	48.24
ct	0.125	27.71	**28.88**	*28.62*	28.60	27.52	28.22
	0.250	32.30	*33.09*	**33.27**	32.99	32.43	32.38
	0.500	37.77	37.87	**38.20**	*37.88*	37.44	37.44
	1.000	**42.07**	41.79	*41.97*	41.74	41.37	41.45
	2.000	45.62	45.59	**45.83**	*45.69*	45.25	45.32
	3.000	47.96	48.10	**48.30**	*48.17*	47.74	47.88
	4.000	49.91	49.99	**50.16**	*50.07*	49.63	49.80
lena	0.125	20.43	**22.76**	22.03	*22.50*	20.39	22.08
	0.250	23.73	**24.90**	24.68	*24.84*	23.18	24.38
	0.500	26.75	**27.19**	26.93	*27.10*	25.82	26.59
	1.000	29.40	*29.58*	29.44	**29.62**	28.46	29.09
	2.000	32.10	**32.22**	32.15	*32.17*	31.05	31.78
	3.000	33.63	**33.73**	33.59	*33.64*	32.50	33.37
	4.000	34.66	34.71	34.62	**34.72**	33.49	34.42

- representative results typically consistent with statistical results

Mesh Quality: Proposed IID1 vs. ID1 Example (Lena image @2%)

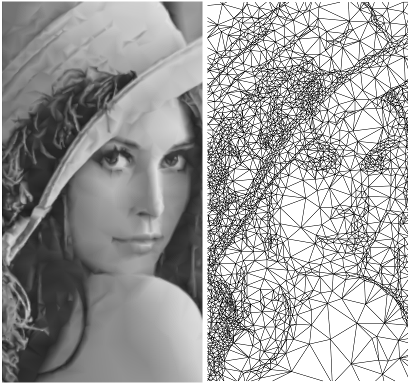


Proposed IID1 (32.10 dB)

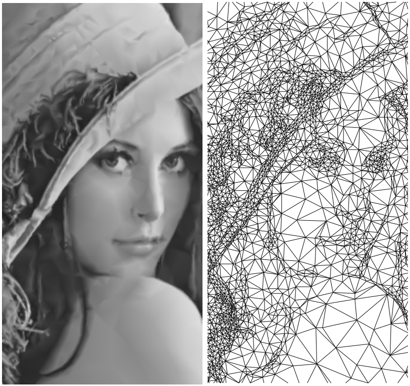


ID1 (32.15 dB)

- comparable overall subjective quality for IID1 and ID1
- IID1 faster than ID1, as will be seen later



Proposed IID2 (32.22 dB)



ID2 (32.17 dB)

- slightly more accurate representation for IID2 compared to ID2 (e.g. nose)
- IID2 faster than ID2, as will be seen later

Proposed IID1 (32.10 dB) | ID1 (32.15 dB) | Proposed IID2 (32.22 dB) | ID2 (32.17 dB) | GPR (31.78 dB)

- subjective quality generally consistent with PSNR
- IID2 has fewer visible artifacts

Computational Cost Comparison

Image	Sampling density(%)	Execution Time (s)					
		IID1	IID2	ID1	ID2	IDDT	GPR
bull	0.125	4.0	8.8	5.0	11.5	3.2	125.4
	0.250	5.4	11.1	8.0	16.3	4.3	117.5
	0.500	7.2	16.0	16.1	26.7	7.5	116.1
	1.000	9.9	22.6	24.0	38.5	11.4	115.2
	2.000	14.8	31.4	30.3	45.4	16.1	112.3
	4.000	24.4	41.2	29.6	45.2	22.0	109.7
ct	0.125	1.1	2.7	1.4	3.4	0.9	38.9
	0.250	1.2	3.2	1.8	4.0	1.1	39.2
	0.500	1.8	4.0	2.4	5.2	1.5	36.6
	1.000	2.7	5.4	3.0	6.0	1.9	37.4
	2.000	4.2	7.2	4.1	7.6	2.6	36.8
	4.000	7.1	12.2	6.7	11.1	4.0	35.3
lena	0.125	1.1	3.0	1.6	3.6	1.2	39.1
	0.250	1.2	2.9	1.7	4.1	1.2	39.1
	0.500	1.6	3.9	2.5	4.9	1.4	38.7
	1.000	2.5	5.3	3.3	6.3	2.0	37.5
	2.000	4.1	7.8	4.7	8.6	2.7	37.3
	4.000	7.1	12.7	7.6	11.9	4.3	36.8

- IID1 and IID2 typically faster than ID1 and ID2, respectively
- IID1 and IID2 both much faster than GPR

Conclusions

- proposed two new methods for generating mesh models of images based on incremental/decremental approach
- proposed methods shown to outperform state-of-the-art mesh-generation methods when mesh quality and computational cost considered together
- proposed methods more efficient: higher quality approximations for a given computational cost
- higher complexity IID2 method: better quality approximations than ID2 and GPR methods
- lower complexity IID1 method: (typically) small penalty in image approximation traded for lower computational cost
- two proposed methods give choice of method based on application and priorities: quality or computational cost

Questions?