CIS Paper Presentation

Alexander Liu, Project 7

Semi-Automated Segmentation of MRI

Mentors:
Alfredo Quiñones-Hinojosa, M.D.
Hadie Adams, M.D.

Hernandez, S. E., K. E. Barner, et al. (2005).

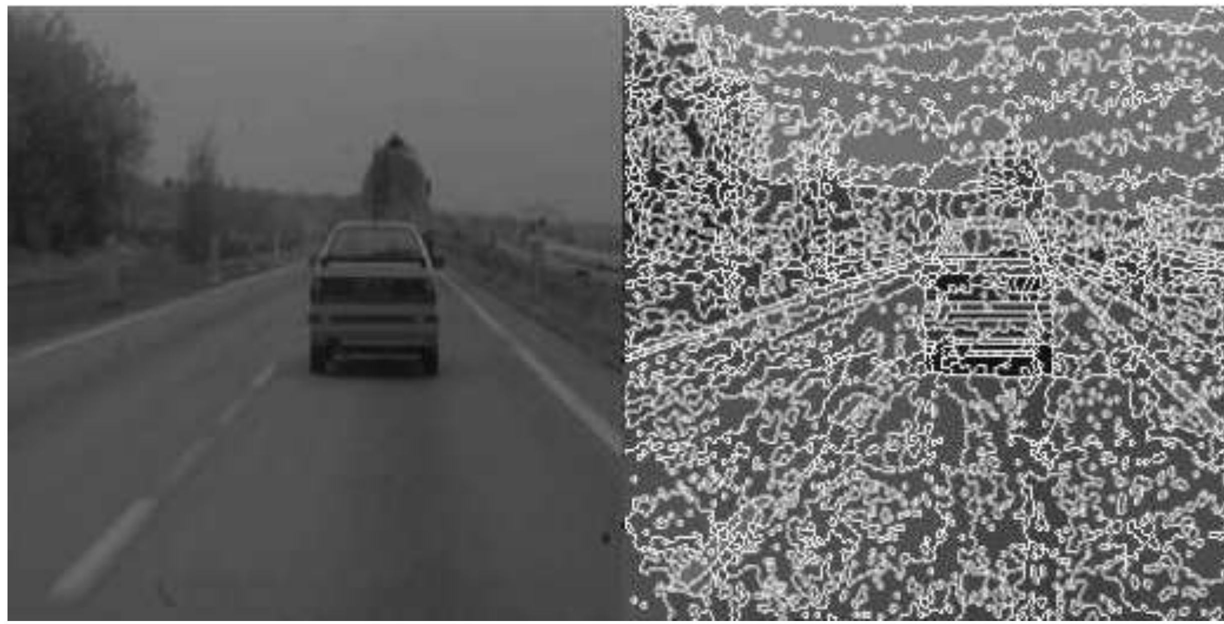
"Region merging using homogeneity and edge integrity for watershed-based image segmentation."

<u>Optical Engineering **44(1): 017004-017014.**</u>

Anything un-cited is from this paper.

Problems Addressed

- Over-segmentation
- Edge Distortion
- False Boundaries

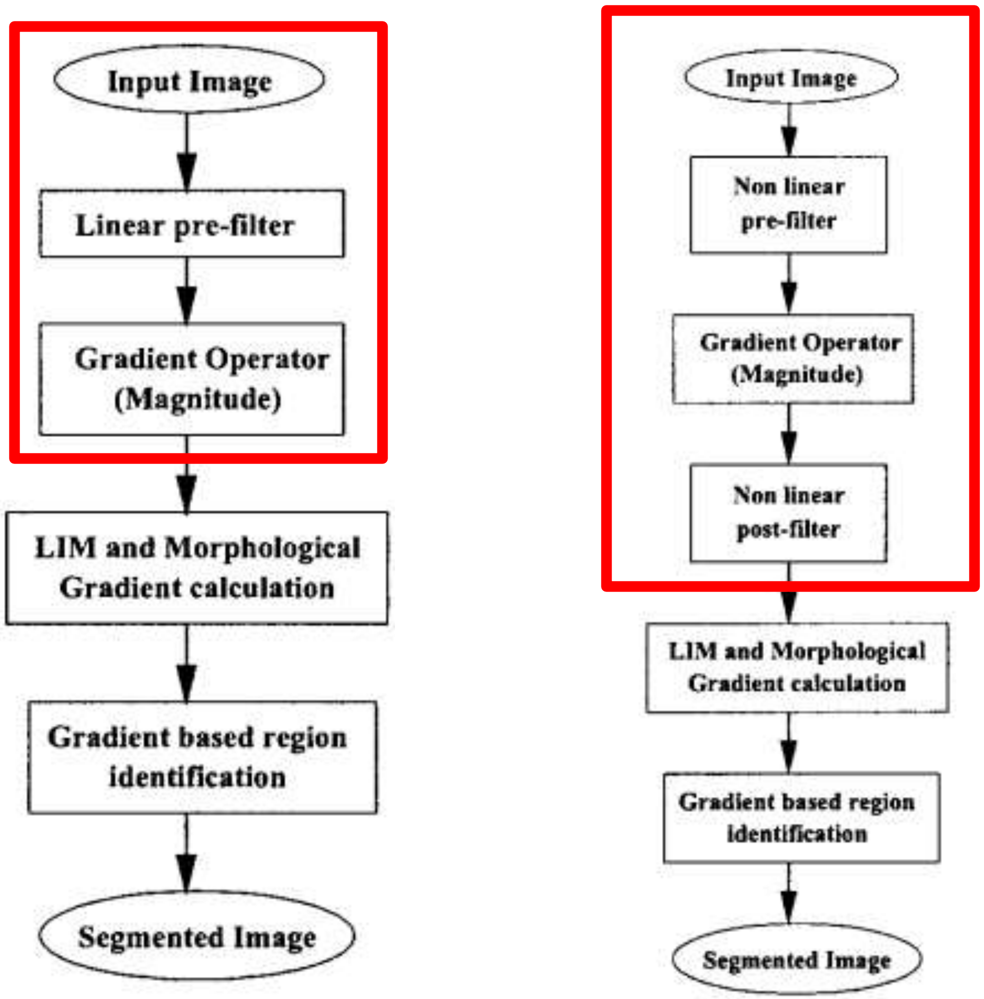


Watershed, hierarchical segmentation and waterfall algorithm. Proc. Mathematical morphology and its applications to image processing, fontainebleau, sept. 1994, jean serra and pierre soille (eds.), Kluwer ac. Publ., Nld, 1994, pp. 69-76.

Main Changes

1. Modified watershed merging criterion – based on homogeneity and edge height
    - Existing methods use them separately
2. Nonlinear pre-filtering and post-filtering to address over-segmentation
    - Existing methods use Gaussian Blur
3. Dynamic weighting algorithm that adjusts emphasis on these two criterions
    - Targets false boundaries specifically

Pre-Processing Steps



J.M. Gauch, "Image segmentation and analysis via multiscale gradient watershed hierarchies", presented at IEEE Transactions on Image Processing, 1999, pp.69-79.

Conventional Watershed

- Typically begins with linear Gaussian pre-filtering
- This removes plateaus (regions of uniform pixel value) via Floating-Point Conversion
- Followed by a gradient operator, yielding the Gradient Magnitude Image
- Overall Result can be viewed as surface where largest gradient magnitude values are boundaries of features in the image

$$\chi_{GMI} = \left\| \nabla\left(\mathcal{G}\left(\chi_{\mathrm{Orig}}\right)\right) \right\|$$

Sobel Operator

- Convolution with a 3x3 kernel in either x or y direction yields approximate derivatives
- These gradients are combined to yield magnitude

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G = \sqrt[2]{G_x^2 + G_y^2}$$

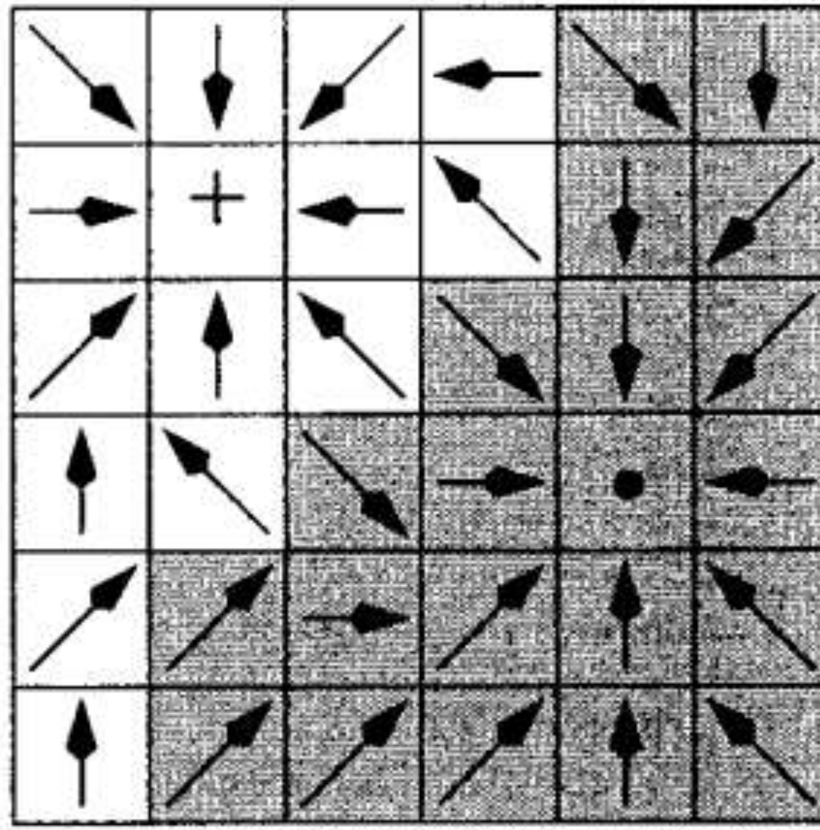
Irwin Edward Sobel. 1970. *Camera Models and Machine Perception*. Ph.D. Dissertation. Stanford University, Stanford, CA, USA. AAI7102831.

GMI Interpretation

- The largest gradient magnitude values are *Watershed Lines*, or lines that represent areas of greatest contrast
- These lines separate *Catchment Basins*, or regions with lesser gradient than their surrounding Watershed Lines
- Each Catchment Basin also has a *Local Intensity Minimum*, a place where gradient is lowest

16.8	19.2	13.5	20.5	31.2	30.1
18.7	11.9	15.4	18.5	22.1	18.4
20.1	21.9	26.6	20.8	17.3	18.1
25.3	22.8	20.9	19.8	15.1	15.9
30.7	35.5	29.9	18.7	17.6	39.9
34.8	38.6	33.4	32.7	33.5	36.7

(a)



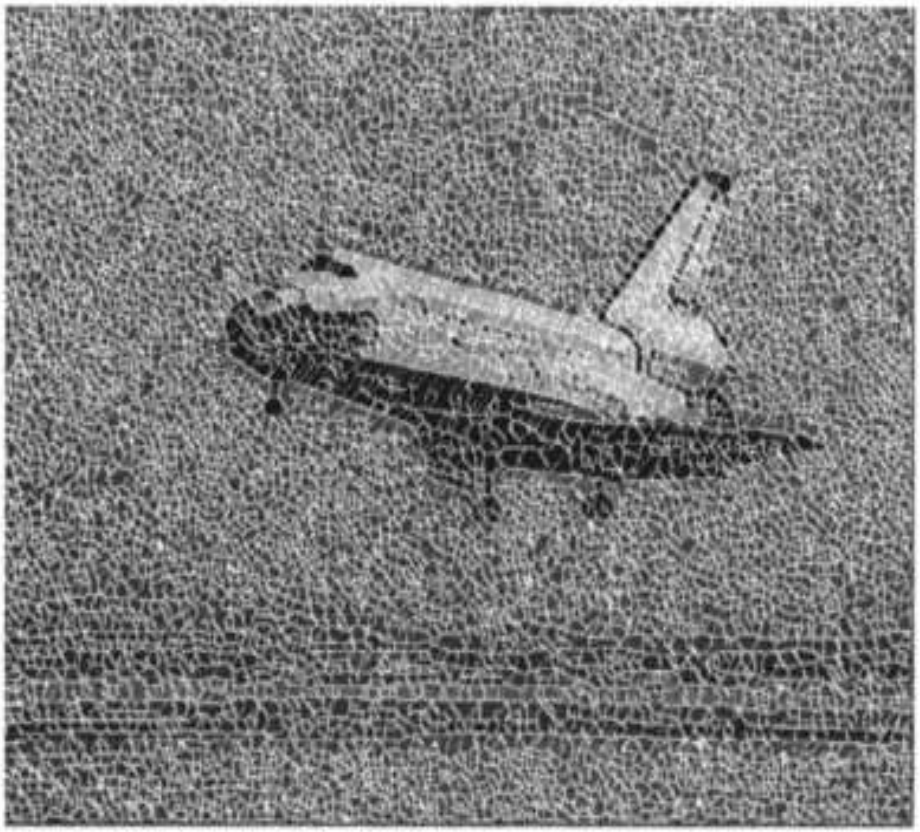
(b)

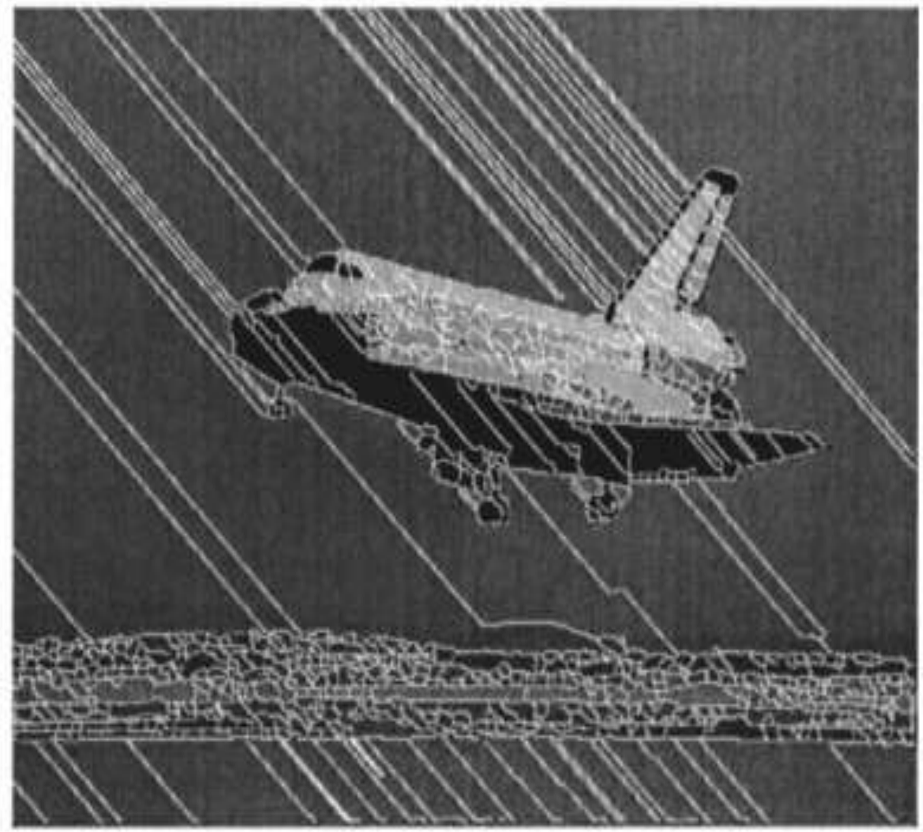
Fig. 2 (a) Matrix showing typical gradient magnitude values. The pixels inside the circles correspond to LIMs. (b) Morphological gradient directions, and corresponding regions, for the matrix values of part (a).

New Method (Pre-Merge)

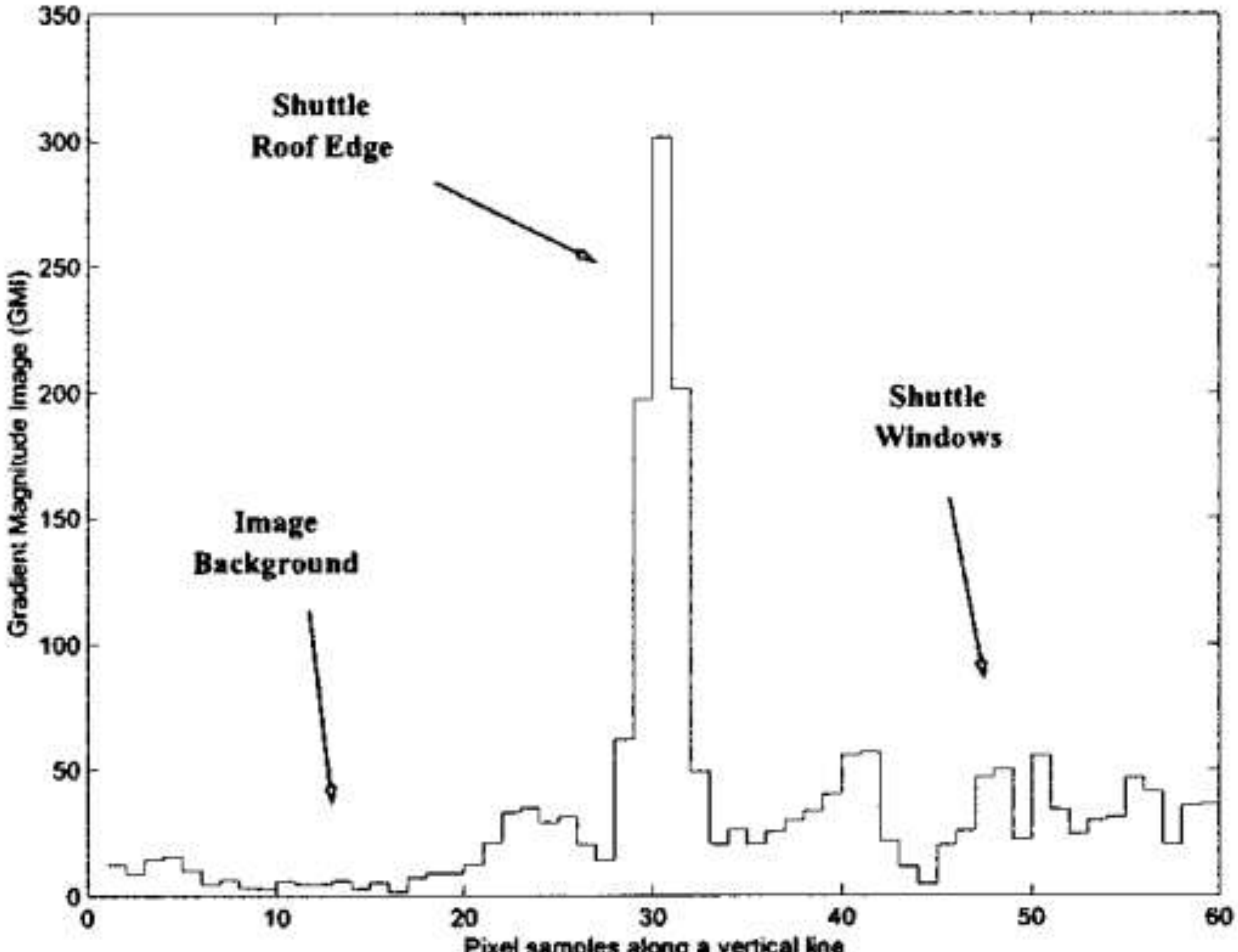
- Median pre-filtering
  - Does not require floating-point conversion
  - Eliminates false LIMs which were found in the previous method
- Post-processing via Thresholding
  - Max(Pixel, Threshold) eliminates isolated low-intensity regions
  - These are often background regions

Example Result

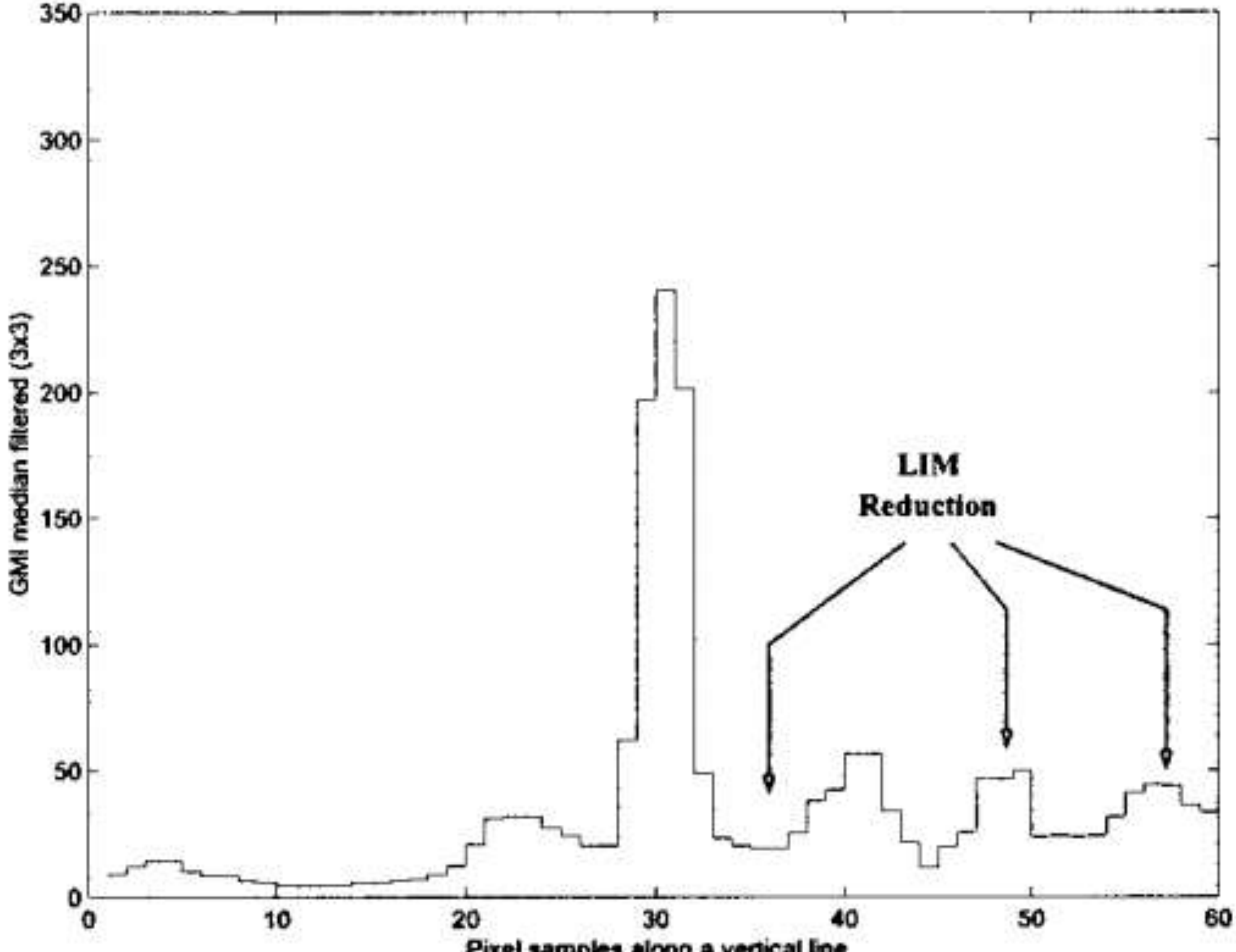




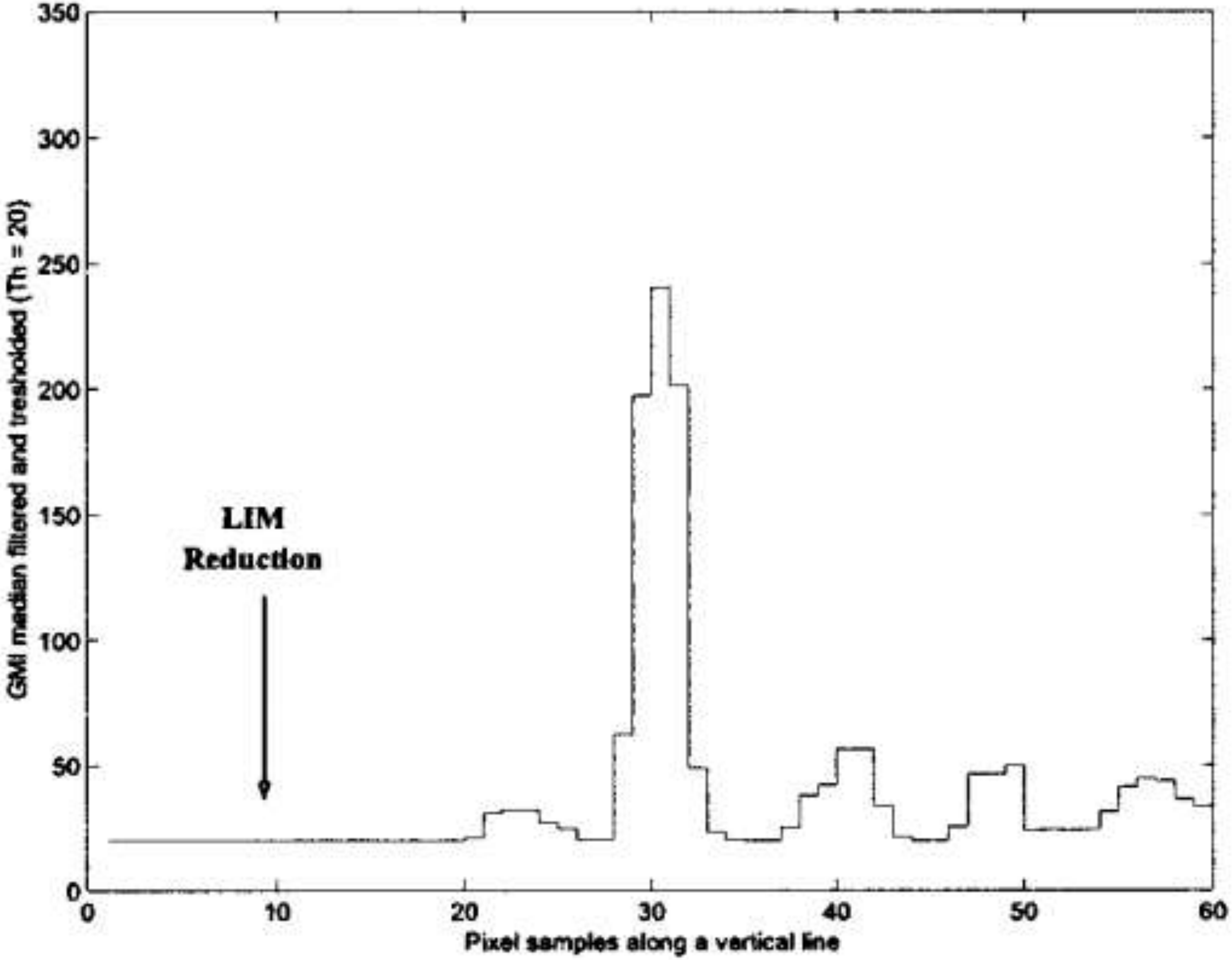
Unprocessed GMI Section



Filtered (Median 3x3)



Filtered and Thresholded (Th = 20)



Pre-Merge Comparison

Old Method

- Gaussian Filtering followed by Gradient Magnitude
- No Post-filtering
- False LIMs
- No plateaus (FP conversion)

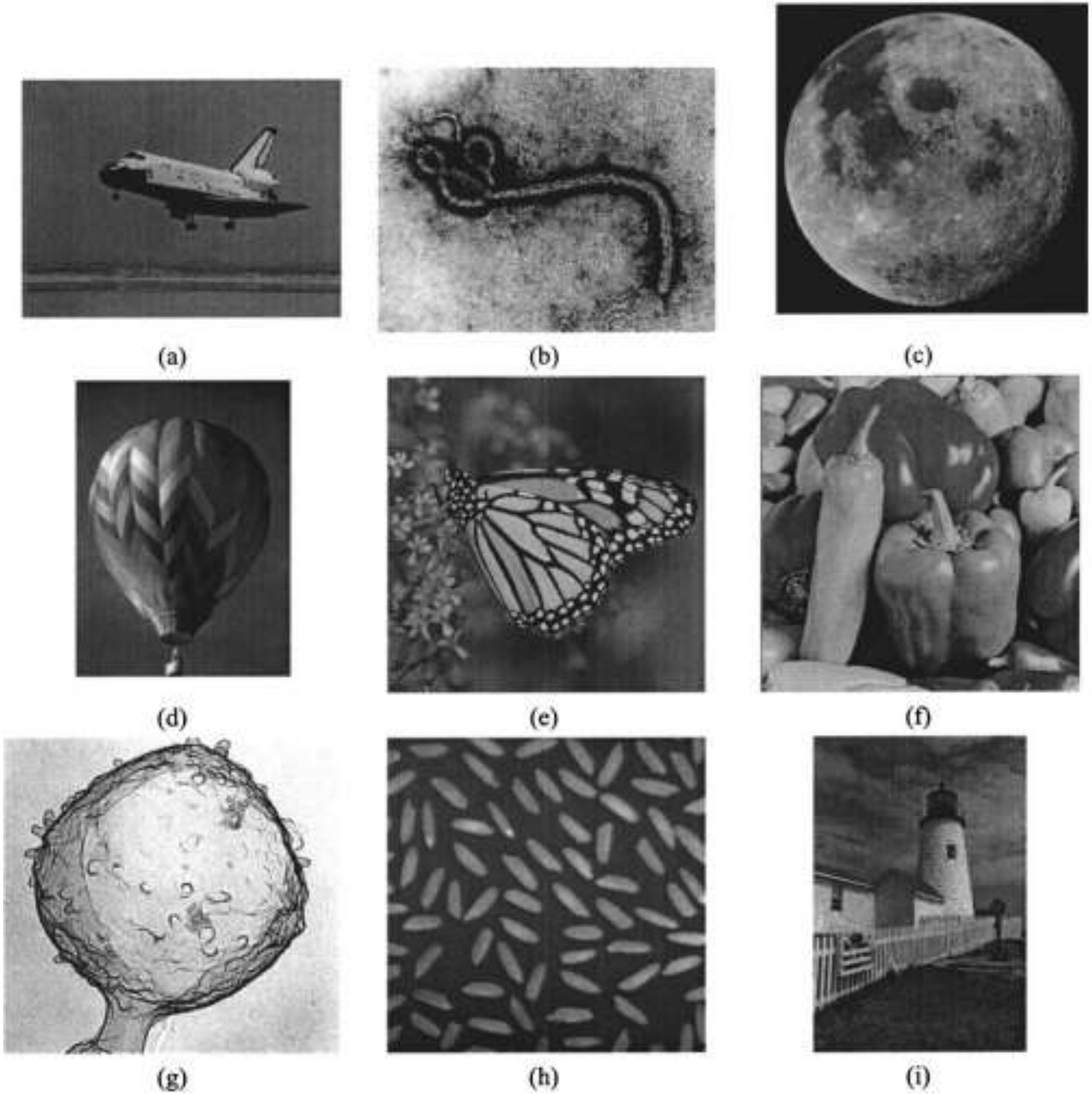
$$\chi_{GMI} = \left\|\nabla(\mathcal{G}(\chi_{\text{Orig}}))\right\|$$

Proposed Method

- Median Filtering, then Gradient Magnitude, then Median Filtering again
- Thresholding Post-filter to process background
- No False LIMs
- Can have plateaus (no FP conversion)

$$\chi_{GPP} = MAX\left[MED\left[\nabla\left(MED\left[\chi_{\text{Orig}}\right]\right)\right], \text{Threshold}\right]$$

PERFORMANCE EVALUATION



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

No Median Filtering with Thresholding of GMI

Image		$T_h = 0$		$T_h = 10$		$T_h = 20$	
Name	Size	Regions	% Red.	Regions	% Red.	Regions	% Red.
(a) Number of initial regions without median filtering of the GMI							
Ebola	285×362	10,143	0	9,740	4	9,400	7
Moon	400×400	15,502	0	11,292	27	9,450	39
Balloon	512×366	15,918	0	4,629	71	1,965	88
Monarch	512×512	21,303	0	13,942	35	6,641	69
Peppers	512×512	22,847	0	17,694	23	9,717	57
Space Shuttle	423×559	18,497	0	9,802	47	3,829	79
Cell	482×502	11,051	0	11,051	0	11,051	0
Rice	256×256	1,920	0	759	60	553	71
Lighthouse	768×512	26,764	0	17,062	23	11,175	57

3x3 Median Filtering with Thresholding of GMI

Image		$T_h = 0$		$T_h = 10$		$T_h = 20$	
Name	Size	Regions	% Red.	Regions	% Red.	Regions	% Red.
(b) Number of initial regions with median filtering of the GMI (3×3)							
Ebola	285×362	4,675	54	4,526	55	4,457	56
Moon	400×400	6,911	55	4,875	69	4,392	72
Balloon	512×366	7,312	54	2,261	86	1,191	93
Monarch	512×512	9,381	56	6,875	68	2,791	87
Peppers	512×512	10,171	55	8,594	62	4,874	79
Space Shuttle	423×559	7,728	58	4,707	75	1,798	90
Cell	482×502	5,430	51	5,117	54	5,110	54
Rice	256×256	1,240	35	473	75	412	79
Lighthouse	768×512	11,615	57	8,012	70	5,644	79

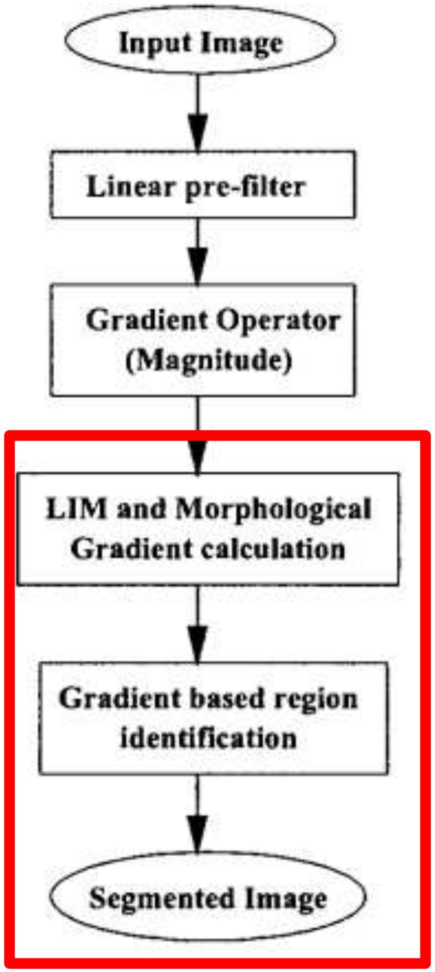
5x5 Median Filtering with Thresholding of GMI

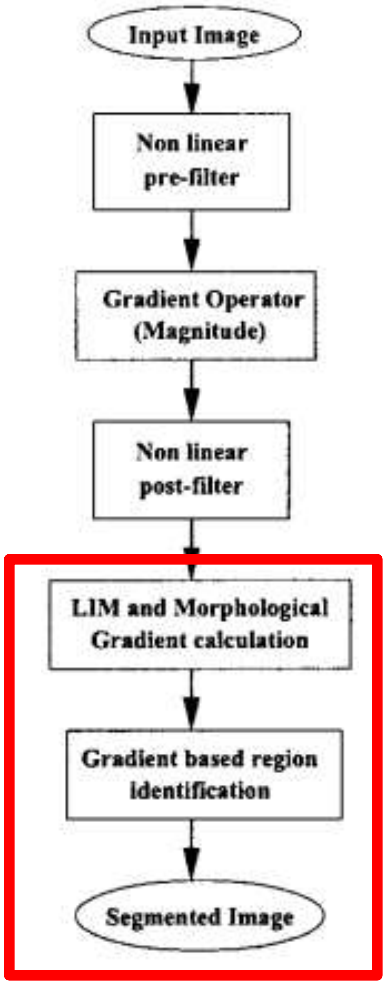
Image		$T_h = 0$		$T_h = 10$		$T_h = 20$	
Name	Size	Regions	% Red.	Regions	% Red.	Regions	% Red.
(c) Number of initial regions with median filtering of the GMI (5×5)							
Ebola	285×362	3,733	63	3,643	64	3,570	65
Moon	400×400	5,468	65	3,666	76	3,440	78
Balloon	512×366	5,849	63	1,646	90	858	95
Monarch	512×512	9,381	56	6,875	68	2,791	87
Peppers	512×512	10,171	55	8,594	62	4,874	79
Space Shuttle	423×559	5,631	70	3,495	81	1,240	93
Cell	482×502	2,714	75	2,586	77	2,586	77
Rice	256×256	775	60	364	81	328	83
Lighthouse	768×512	8,012	70	5,018	81	2,732	90

3x3 Median Pre- and Post- Filtering with Thresholding

Image		$T_h=0$		$T_h=10$		$T_h=20$	
Name	Size	Regions	% Red.	Regions	% Red.	Regions	% Red.
Ebola	285×362	3,272	68	3,087	70	2,716	73
Moon	400×400	4,538	71	3,352	78	2,764	82
Balloon	512×366	5,365	66	1,346	92	977	94
Monarch	512×512	7,076	67	3665	83	2,044	90
Peppers	512×512	7,235	68	4,553	80	2541	89
Space Shuttle	423×559	6,734	64	1,914	90	934	95
Cell	482×502	4,407	60	4,245	62	4,165	62
Rice	256×256	1,076	44	463	76	355	82
Lighthouse	768×512	9,216	66	6,976	74	4,306	84

Post-Processing Steps

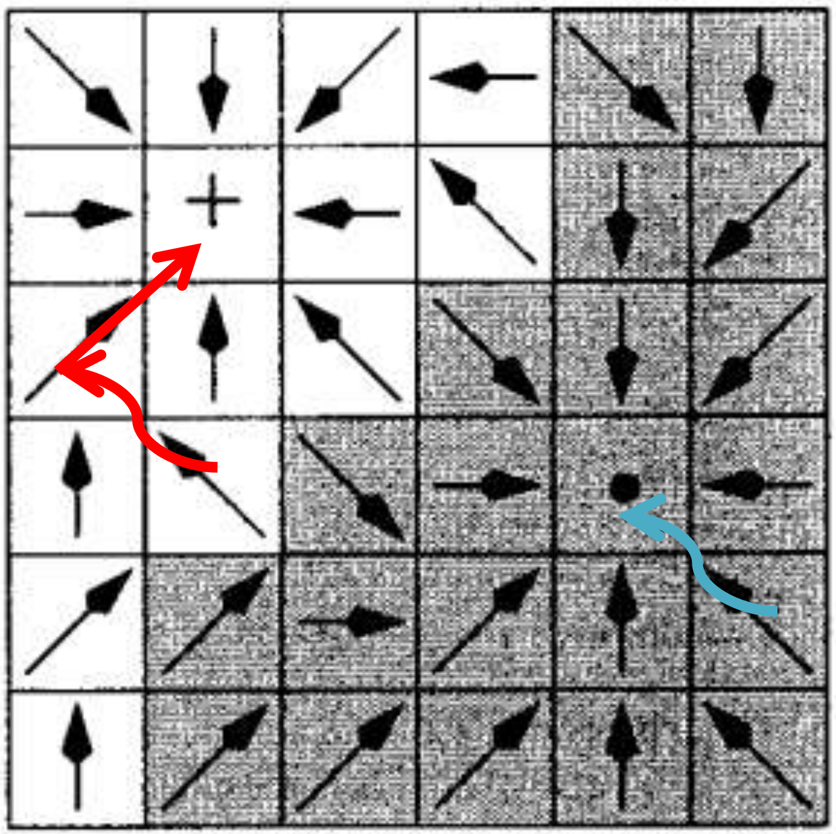




J.M. Gauch, "Image segmentation and analysis via multiscale gradient watershed hierarchies", presented at IEEE Transactions on Image Processing, 1999, pp.69-79.

Region Identification

16.8	19.2	13.5	20.5	31.2	30.1
18.7	11.9	15.4	18.5	22.1	18.4
20.1	21.9	26.6	20.8	17.3	18.1
25.3	22.8	20.9	19.8	15.1	15.9
30.7	35.5	29.9	18.7	17.6	39.9
34.8	38.6	33.4	32.7	33.5	36.7



Region Adjacency Graph

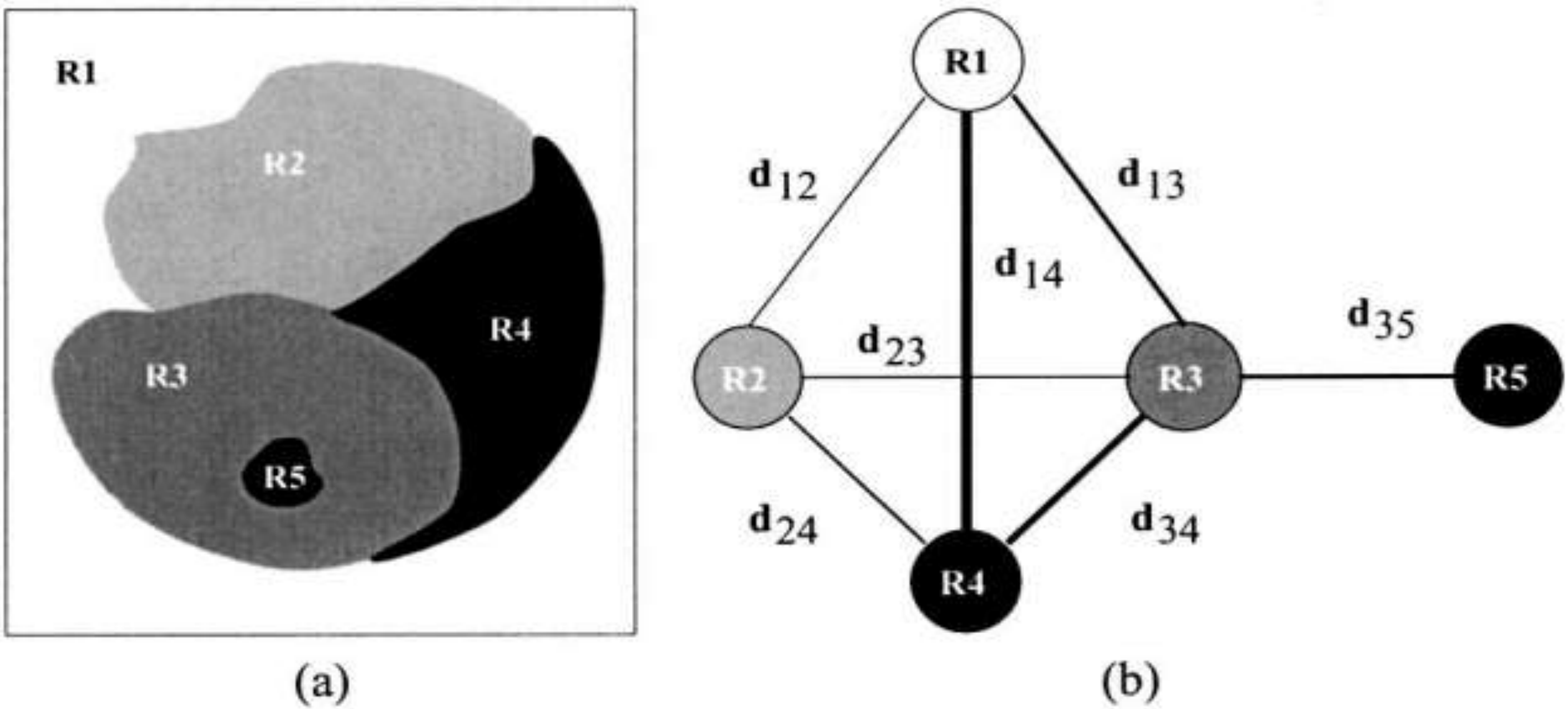


Fig. 6 (a) Partition of an image into five regions and (b) its corresponding RAG.

Graph Cost Functions

1. Edge Orientation
2. Region Homogeneity
3. Edge Integrity

Graph Cost Functions

1. Edge Orientation
2. Region Homogeneity
3. Edge Integrity

Edge Orientation Cost Function

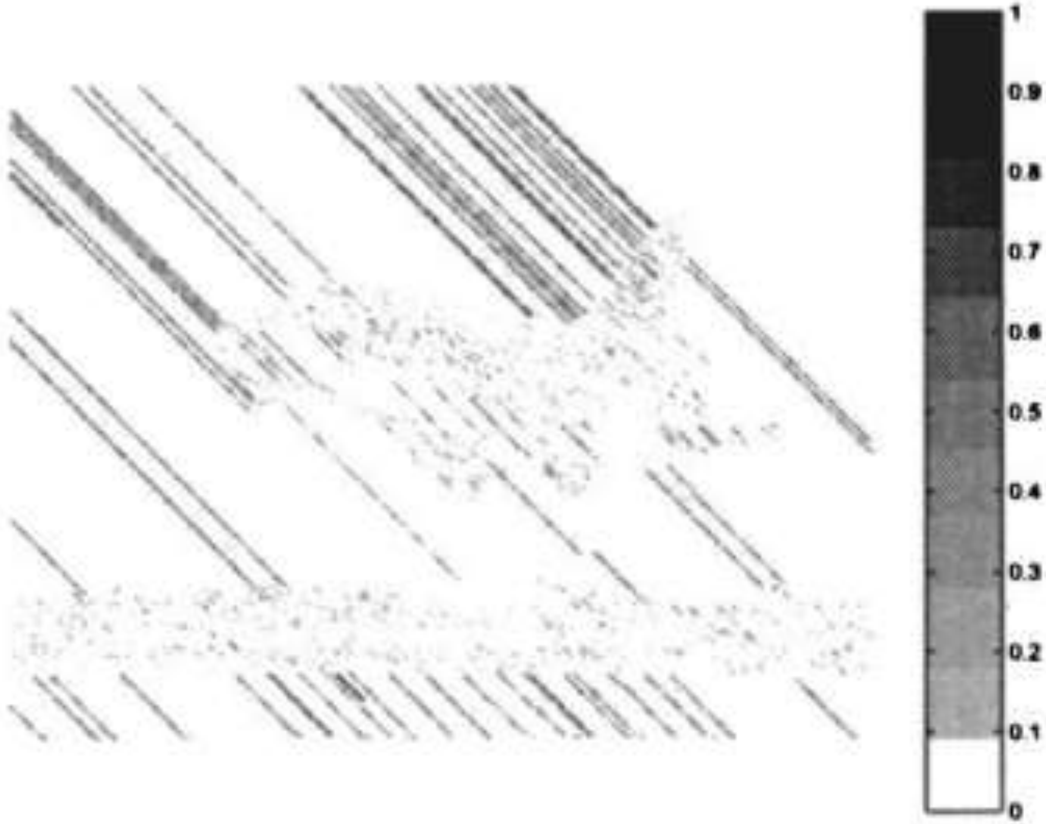
- Penalizes 135 degree edges
- Merges regions with the highest proportions of diagonal pixels (regions with greatest delta) first
- Stopping criterion based on lowest delta value (i.e. 0.8)

$$\mathcal{E}_{\mathcal{B}}^{i,j} = \{(\mathbf{n}^1, \mathbf{n}^2) : |\mathbf{n}^1 - \mathbf{n}^2| = 1,\ i(\mathbf{n}^1) \in R_K^i, i(\mathbf{n}^2) \in R_K^j\}$$

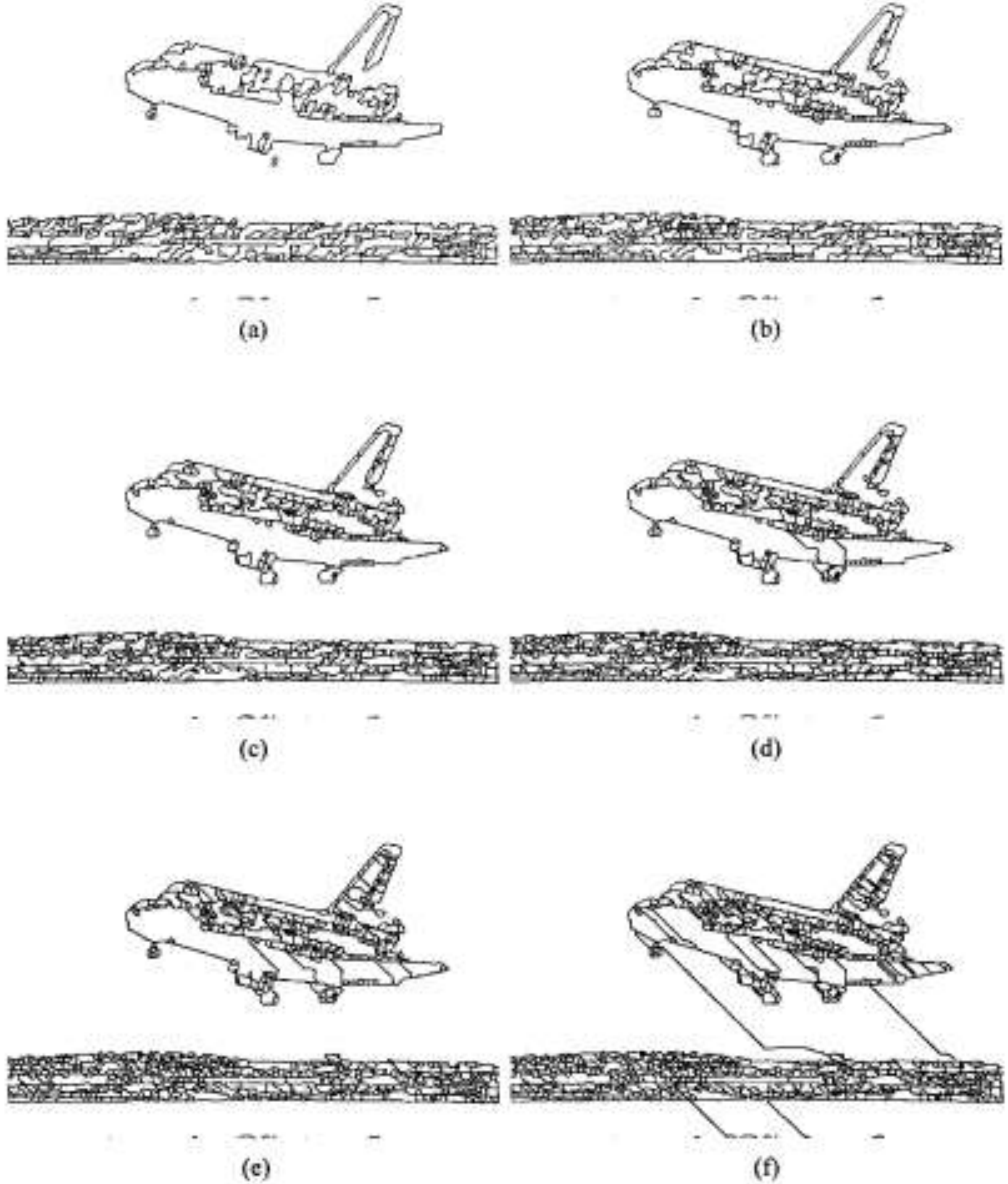
$$\mathcal{E}_{\mathcal{D}}^{i,j} = \{(\mathbf{n}^1, \mathbf{n}^2) : n_1^2 - n_1^1 = n_2^2 - n_2^1 = 1,\ (\mathbf{n}^1, \mathbf{n}^2) \in \mathcal{E}_{\mathcal{B}}^{i,j}\}$$

$$\delta^{\mathcal{O}}(R_K^{*i}, R_K^{*j}) = \frac{\|\mathcal{E}_{\mathcal{D}}^{i,j}\|}{\|\mathcal{E}_{\mathcal{B}}^{i,j}\|}$$

Cost Function Results



(c)


(a)
(b)
(c)
(d)
(e)
(f)

Graph Cost Functions

1. Edge Orientation
2. Region Homogeneity
3. Edge Integrity

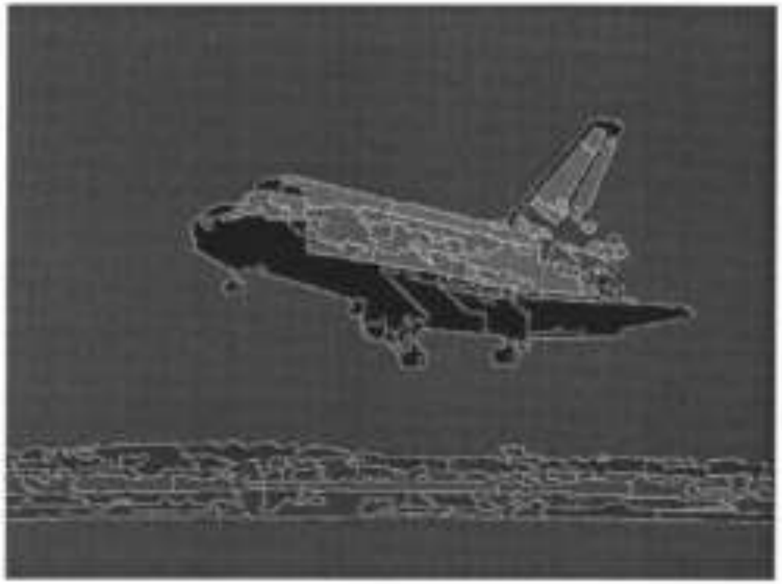
Region Homogeneity Cost Function

- Regions with greatest similarity have the lowest edge weight
- Merging order starts with regions with lowest edge weight (more homogeneous regions merged first). Region size was also tested.
- Stops when a certain number of regions is reached

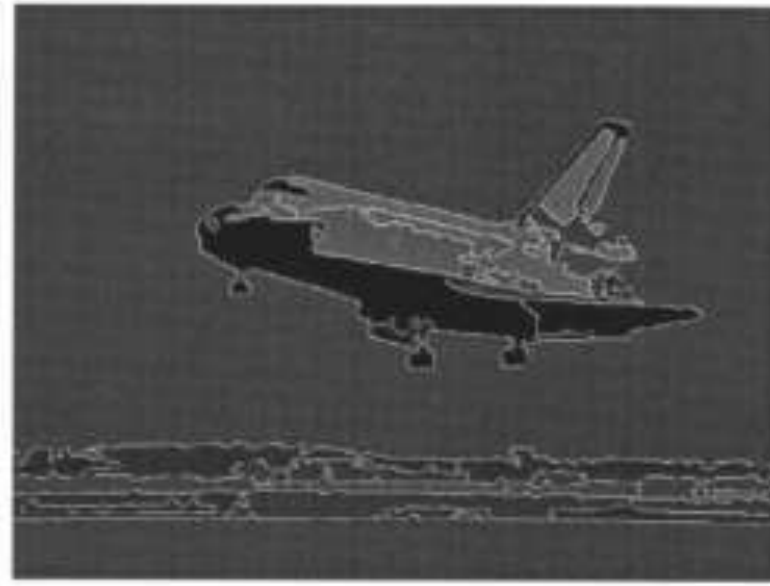
$$\mu(R_k) = \frac{1}{\|R_k\|} \sum_{i(m)\in R_k} i(m)$$

$$E(R_K) = \sum_{i(m)\in R_k} [i(m) - \mu(R_k)]^2$$

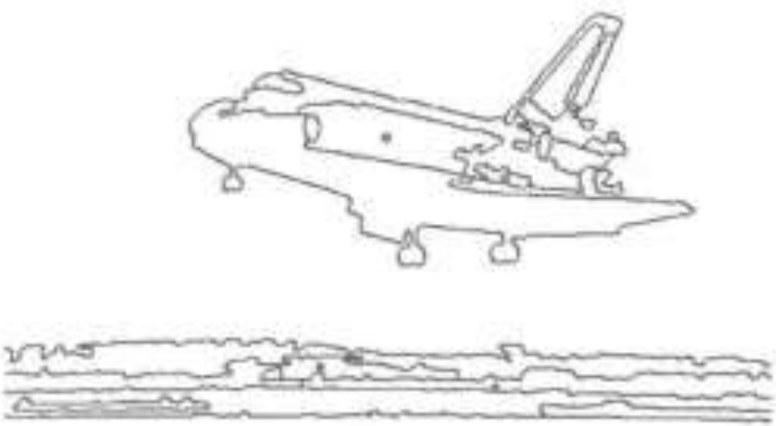
$$\delta^H(R_i, R_j) = \frac{\|R_i\| \cdot \|R_j\|}{\|R_i\| + \|R_j\|} [\mu(R_i) - \mu(R_j)]^2$$



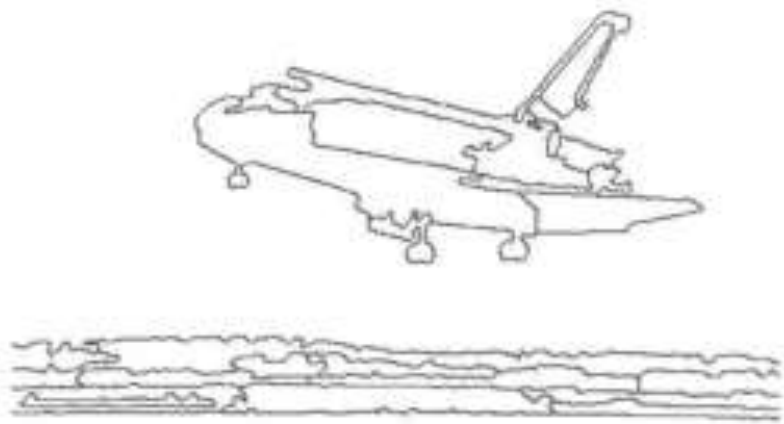
(a)



(b)

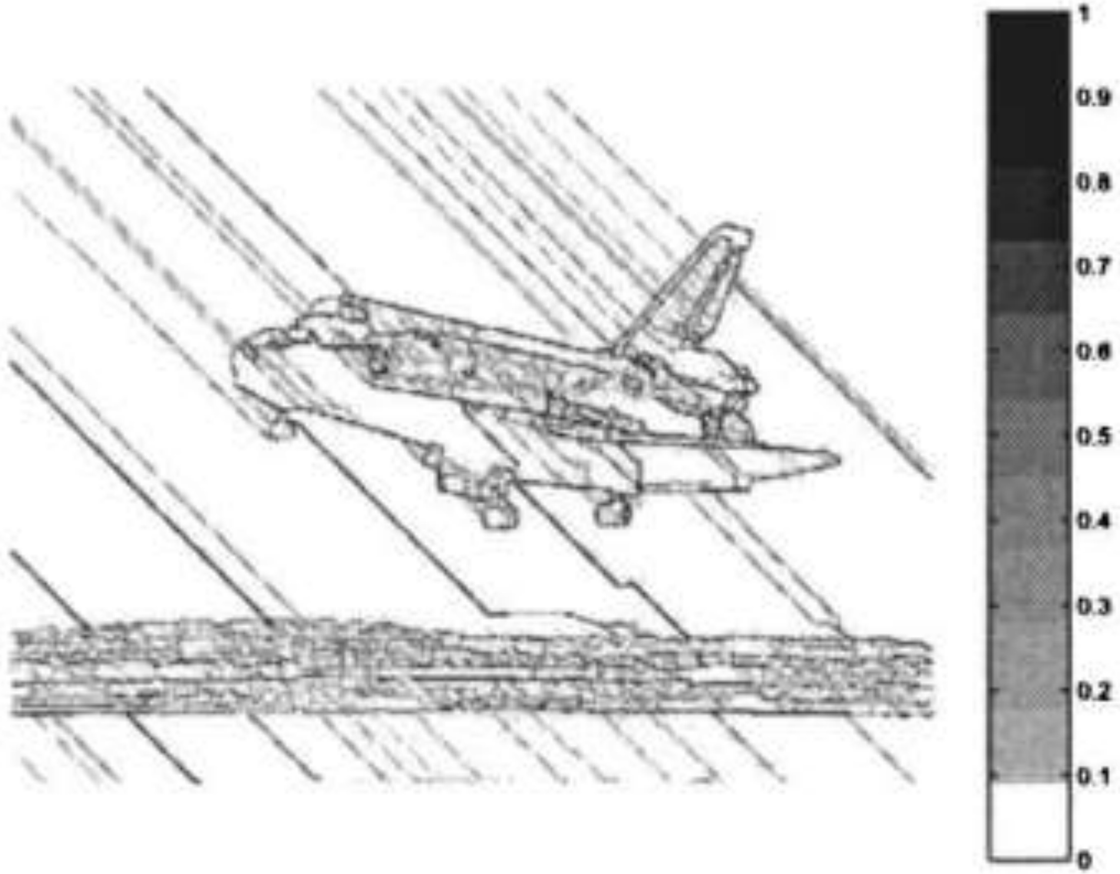


(c)



(d)

Cost Function Results



(a)

Graph Cost Functions

1. Edge Orientation
2. Region Homogeneity
3. Edge Integrity

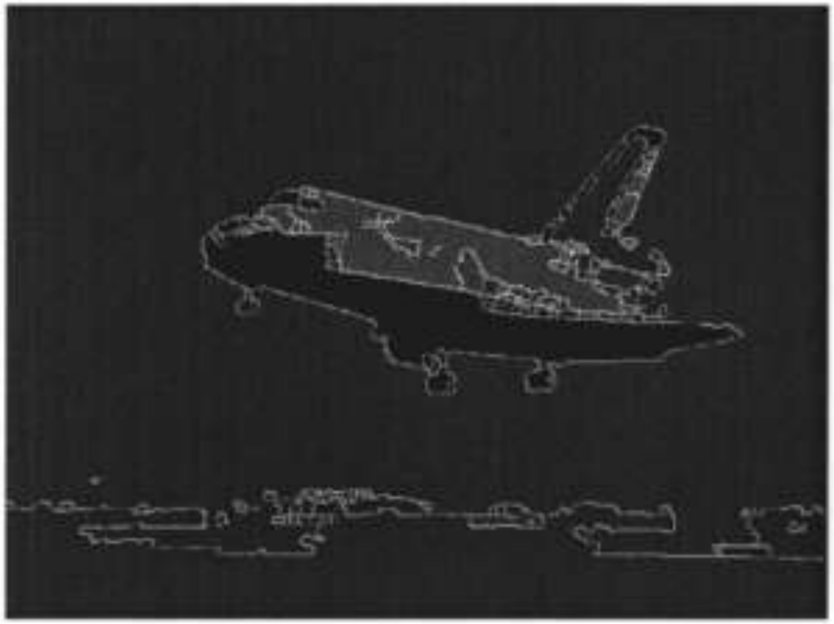
Edge Integrity Cost Function

- The max of two contour pixels from the GMI is found
- Over Threshold is 'strong edge'
- Threshold chosen with median of all height values
- Proportion is again used to define graph edge weight

$$e_h(n^1, n^2) = \max\left[i_g(n^1), i_g(n^2)\right]$$

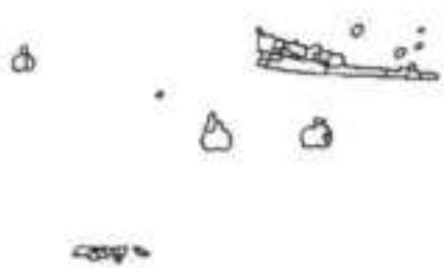
$$\varepsilon_S(n^1, n^2) = \{(\mathrm{n}^1, \mathrm{n}^2) \in \varepsilon_B, \mathrm{e_h}(\mathrm{n}^1, \mathrm{n}^2) > \mathrm{T_h}\}$$

$$\delta^{\varepsilon}\left(R^i, R^j\right) = \frac{\|\varepsilon_s\|}{\|\varepsilon_B\|}$$

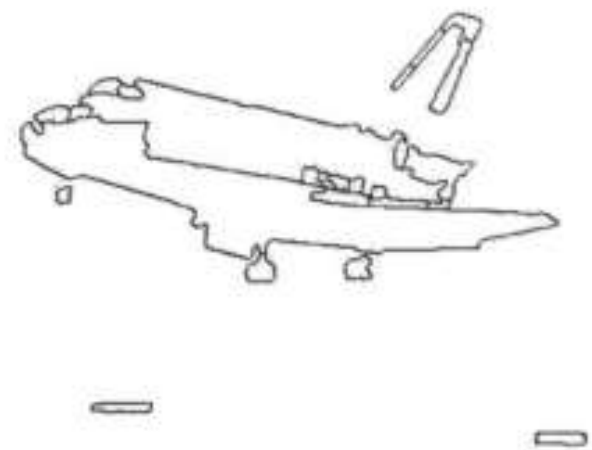


(a)

(b)

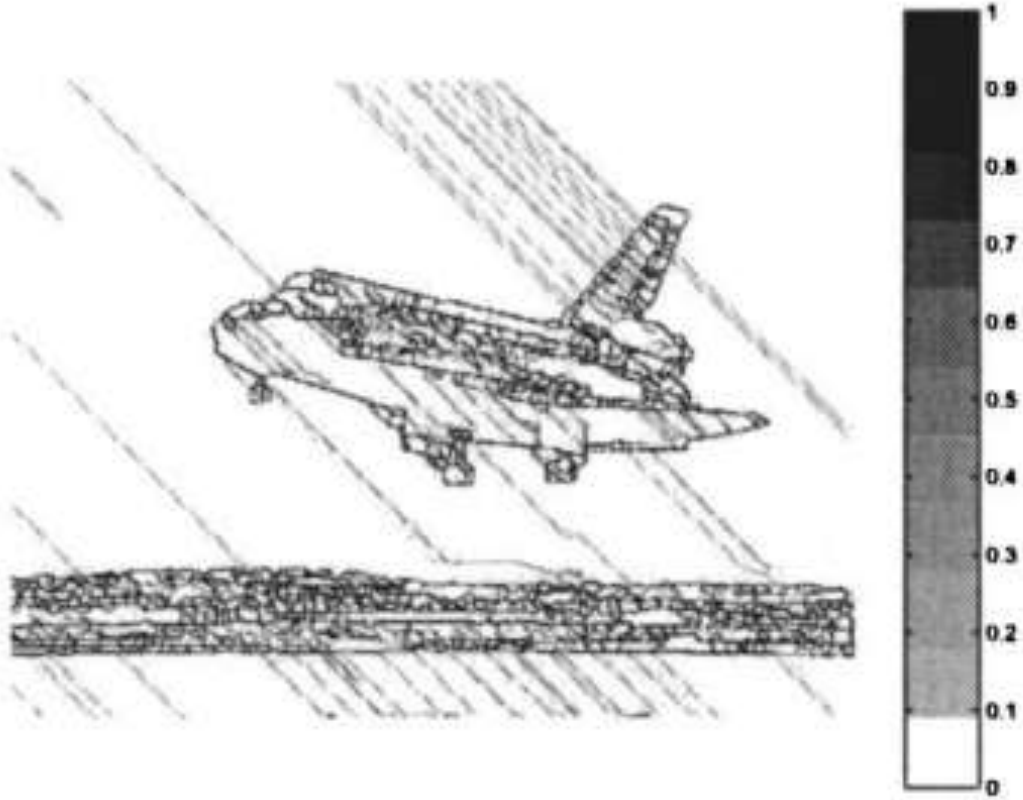


(c)



(d)

Cost Function Results



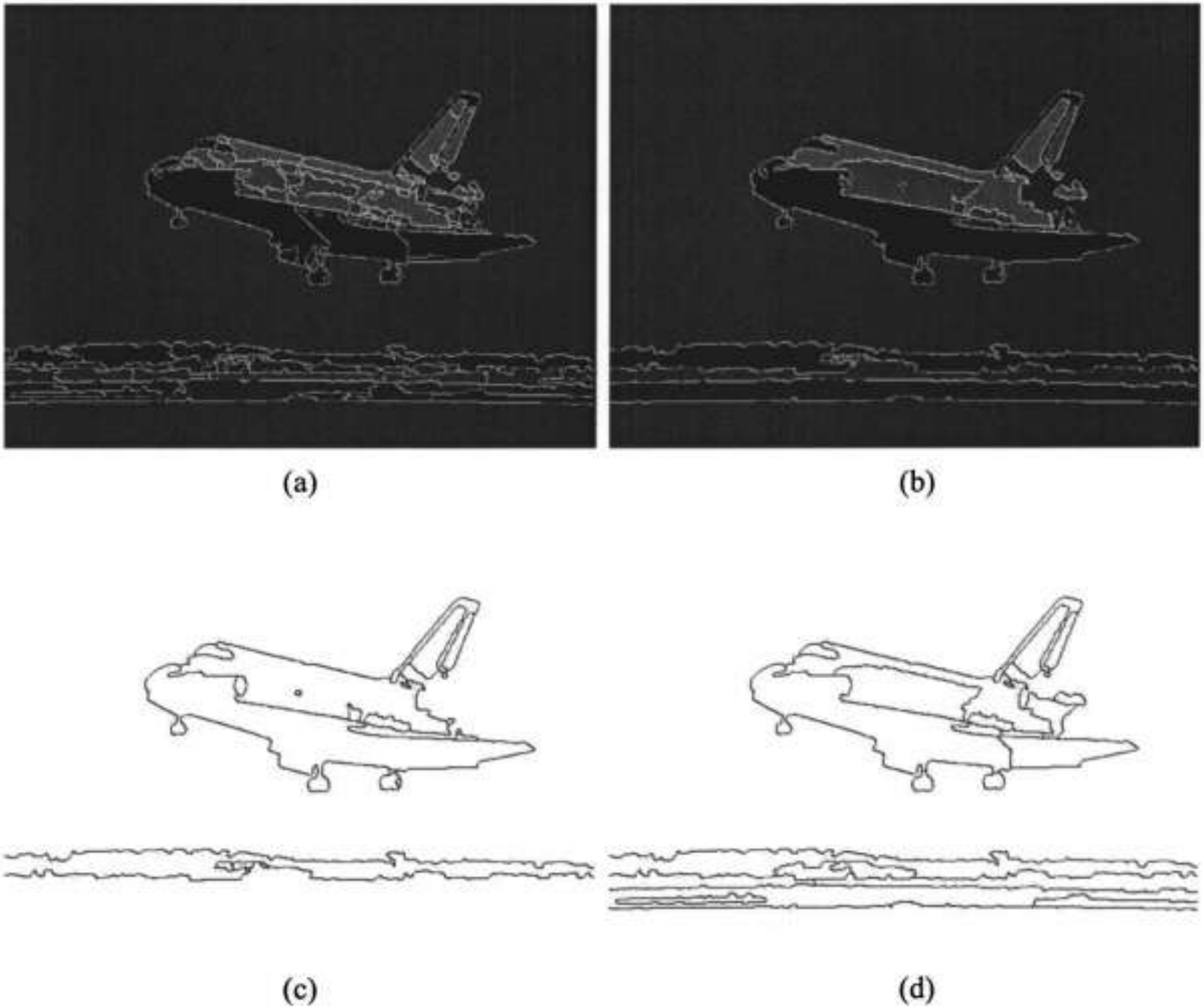
(b)

Hybrid Method

- Weighting function based on Edge Integrity and Region Homogeneity
- Represents a measure of dissimilarity
- Merge order based on smallest W

$$W = \alpha R^{H} + (1-\alpha) R^{\varepsilon}$$

$$0 \leq \alpha \leq 1$$



(a) (b)

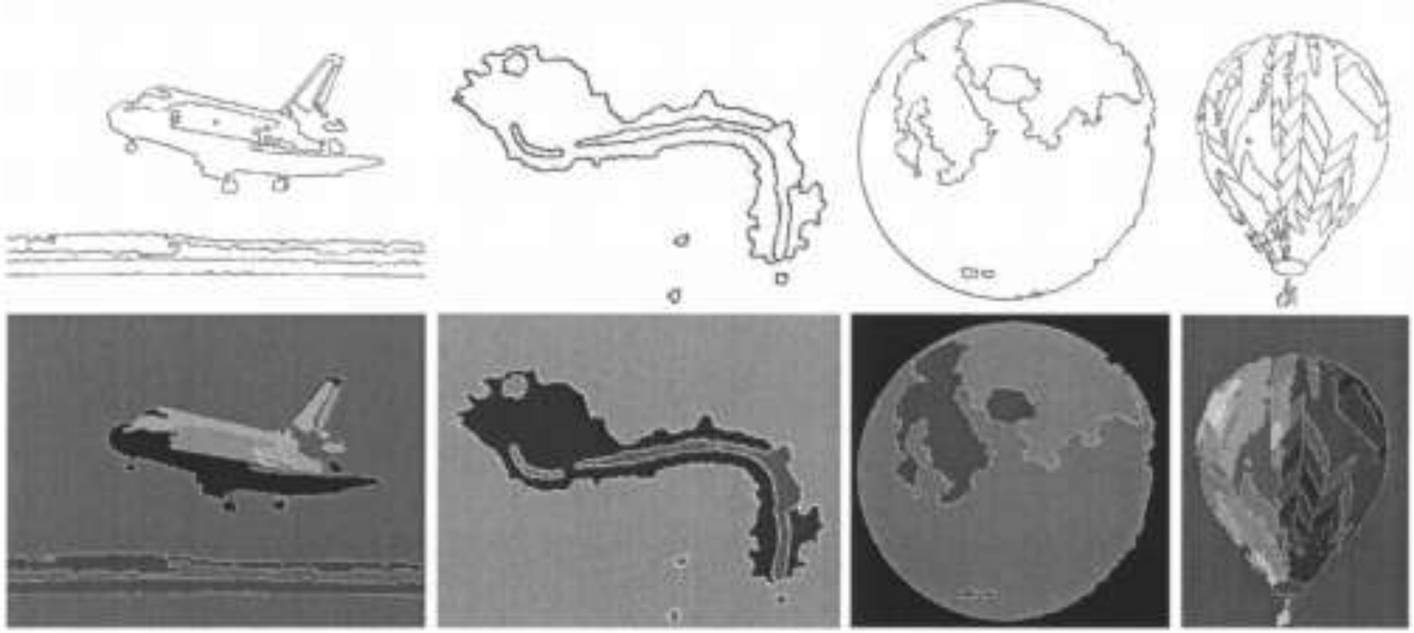
(c) (d)

Dynamic Weighting

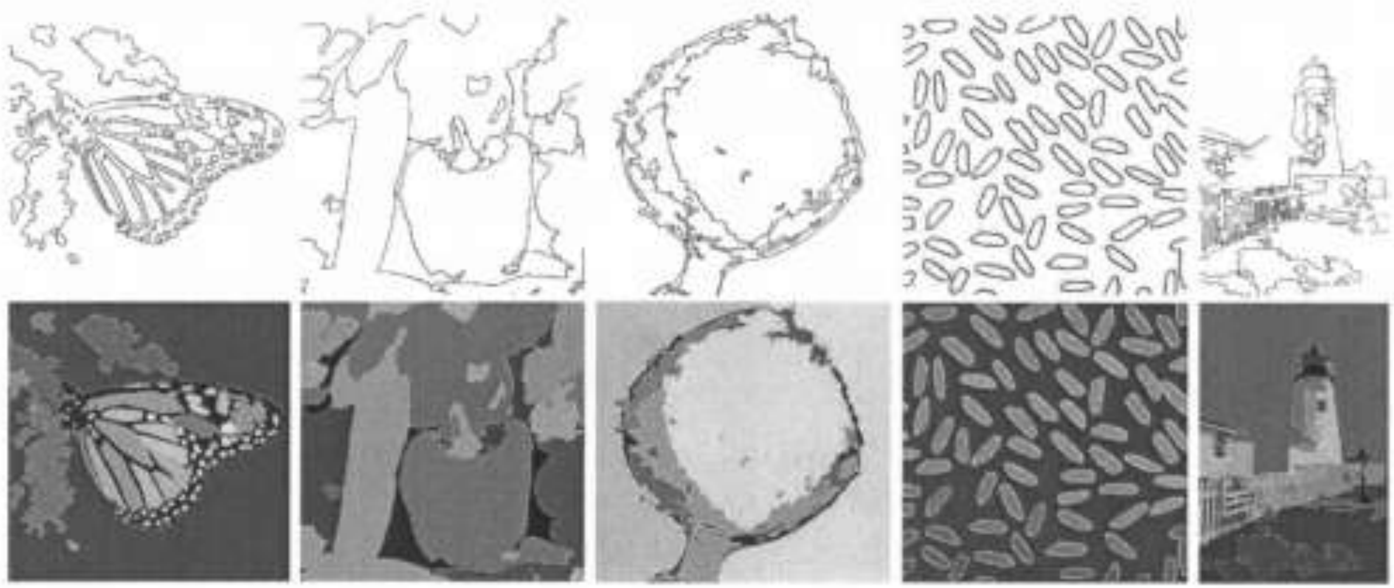
- Alpha parameter changed with proportion of small regions
- A region is considered 'small' if its size fraction is less than a threshold

$$S_R = \left\{ R^j : \frac{\|R^j\|}{\|\chi\|} < P_s \right\}$$

$$\alpha = \frac{\|S_R\|}{K}$$

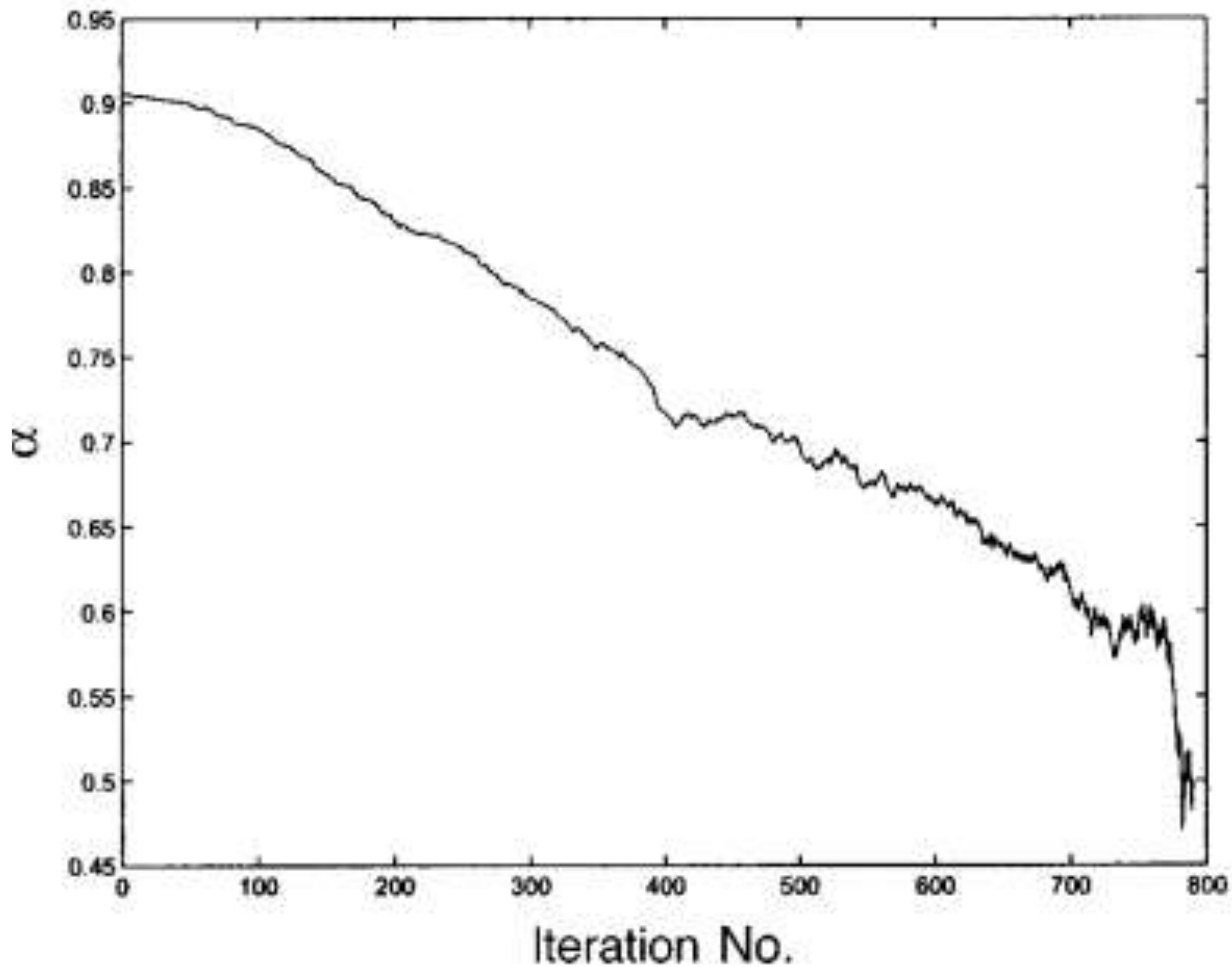


(a) (b) (c) (d)



(e) (f) (g) (h) (i)

Trace of alpha (Shuttle)



Regions

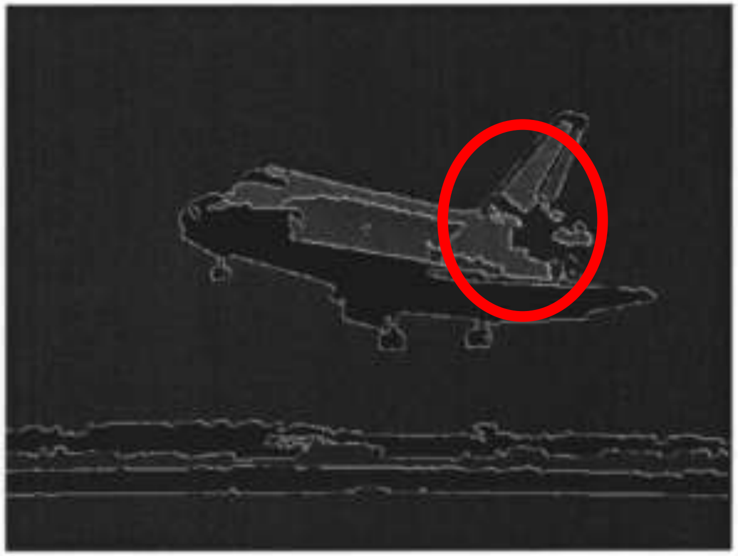
Image		Simulation parameters						
		Watershed procedure				Region-merging procedure		
			Postfilter					
Name	Size	Pre-filter	T_h	Size	Initial regions	β	P_s (%)	Final regions
Space Shuttle	423×559	3×3	20	3×3	934	0.7	0.5	25
Ebola	207×262	3×3	30	3×3	1413	0.7	0.5	9
Moon	400×400	3×3	30	3×3	1893	0.7	0.5	10
Balloon	512×366	3×3	30	3×3	754	0.7	0.5	100
Monarch	512×512	3×3	25	3×3	1747	0.7	0.5	100
Peppers	512×512	3×3	30	3×3	1697	0.7	0.5	50
Cell	482×502	3×3	60	3×3	2474	0.7	0.5	50
Rice	256×256	3×3	20	3×3	412	0.7	0.5	75
Lighthouse	768×512	3×3	40	3×3	2881	0.7	0.5	150

Summary

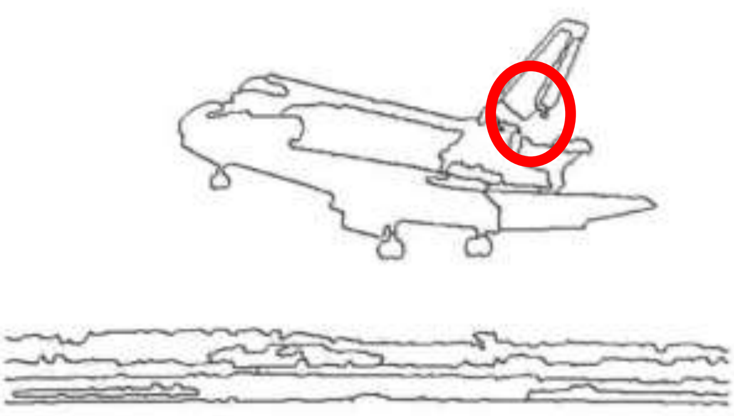
- The combined method appeared to retain the benefits of both methods while leaving behind the pitfalls of each
- The described Pre- and Post- processing could possibly be used in conjunction with the Dynamic Weighted Method for a better result

Critiques

- Only showed results from one case
- Spurious region elimination specific to noise
- “Leaks” still occur, causing some regions to be contiguous with background



(b)



(d)

Acknowledgements

- Dr. Hadie Adams
- Dr. Alfredo Quiñones-Hinojosa
- Dr. Russell Taylor
- Nathaniel Tippens

QUESTIONS?