+ Real-Time Interactive Segmentation with Superpixel Pre-Segmentation

Graduate Institute of Communication Engineering

National Taiwan University

Speaker: Chia-Jung Lin

Advisor: Jian-Jiun Ding

07.24.2014

Outline

- Introduction
  - Interactive Image Segmentation
  - Graph Cut Segmentation
  - Superpixel
- Proposed Method
- Experimental Results
- Discussion
- Conclusion and Future Work

Image Segmentation

- Unsupervised segmentation
- Semi-supervised segmentation (Interactive Segmentation)
- (almost completely) Manual segmentation

Image Segmentation

- Unsupervised segmentation

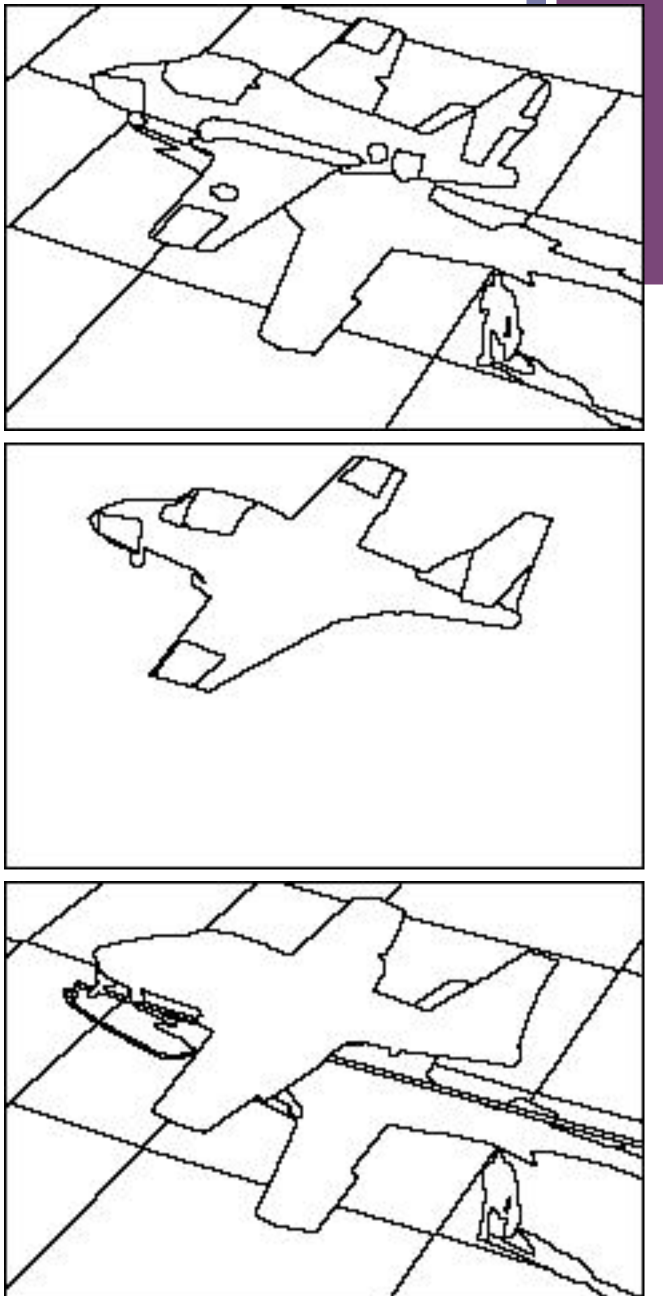
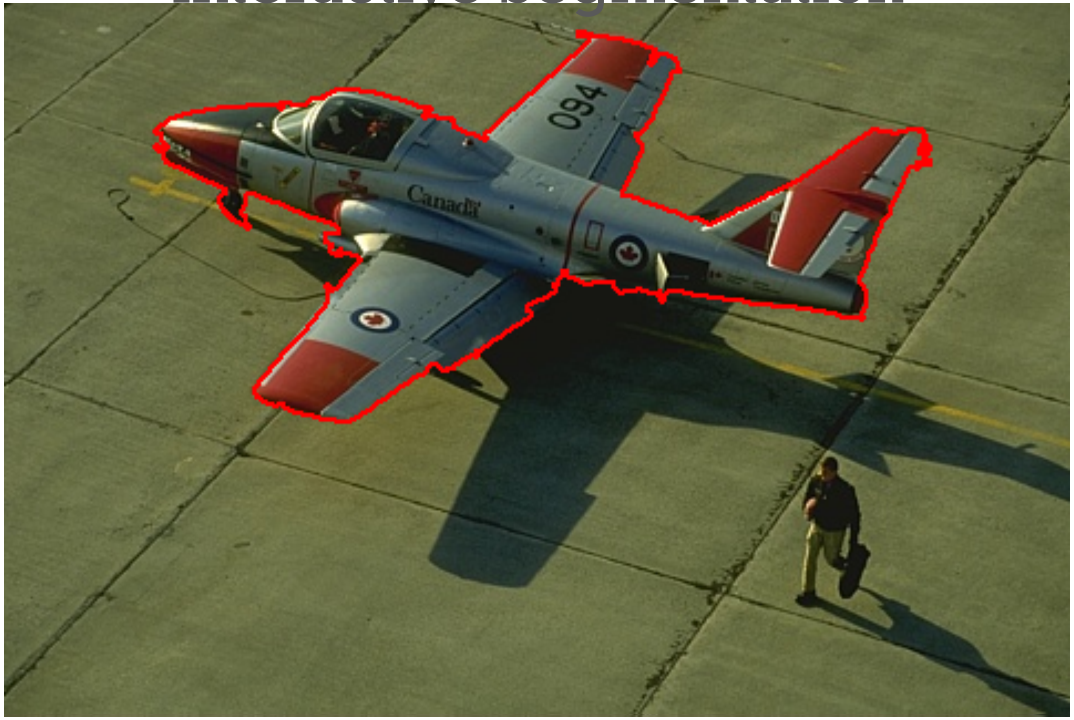


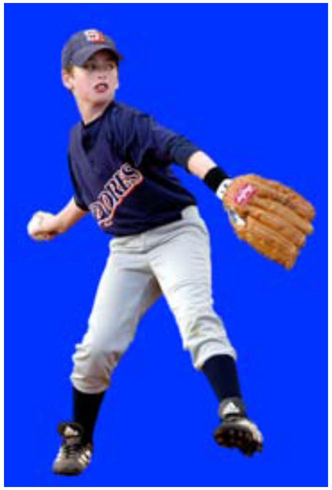
Image Segmentation

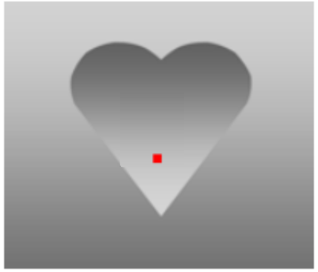
- Interactive Segmentation



Interactive Segmentation

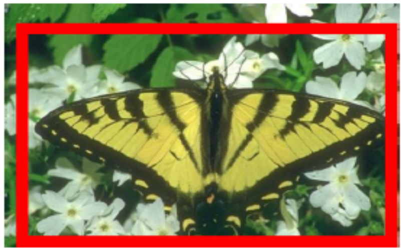
- Different Ways of User Interaction


Interactive Segmentation

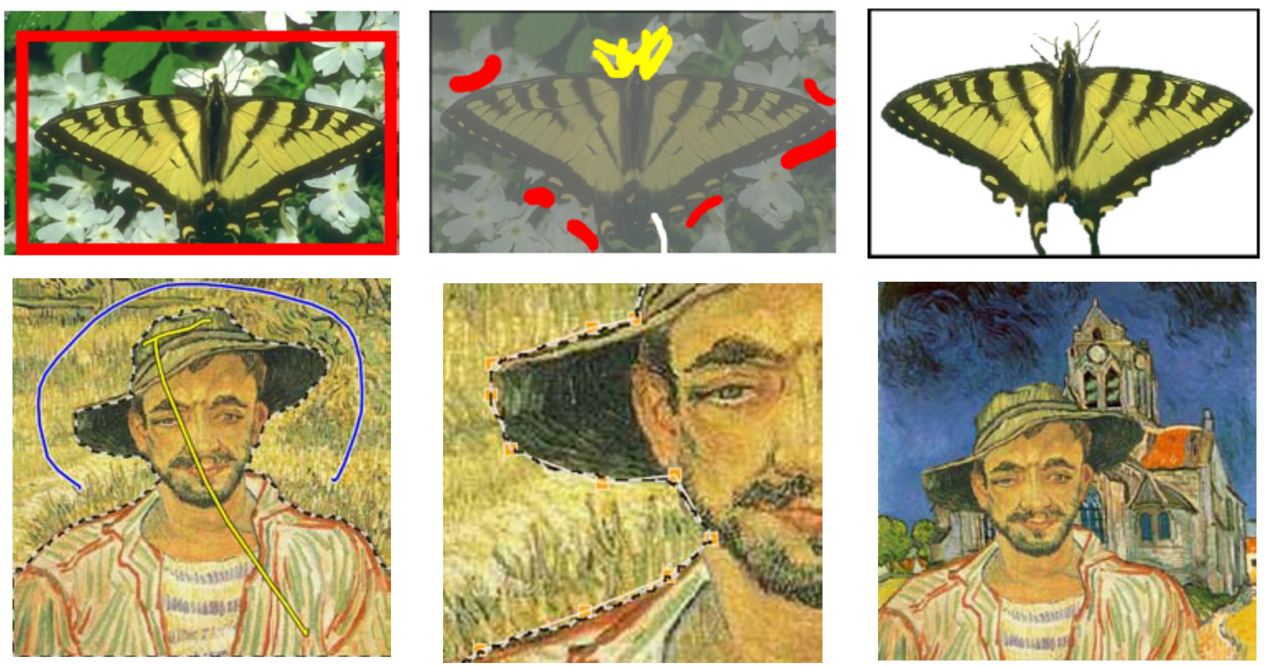
- Different Ways of User Interaction
  - provide function for correction at a finer scale
  - apply matting technique along segmentation border



Red marker: background
White marker: foreground
Yellow marker: trimap for matting

Interactive Segmentation

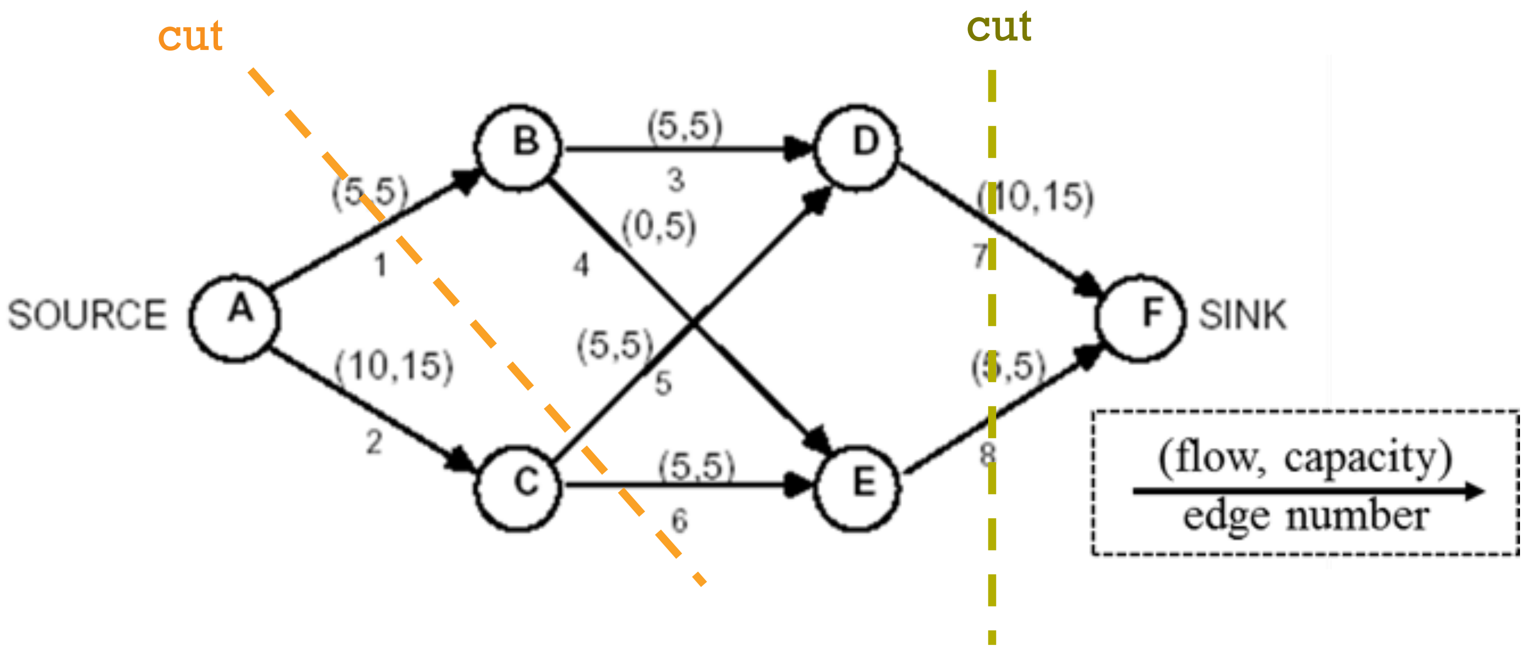
- Different Ways of User Interaction
  - provide function for correction at a finer scale
  - apply matting technique along segmentation border



Yellow marker: foreground
Blue marker: background

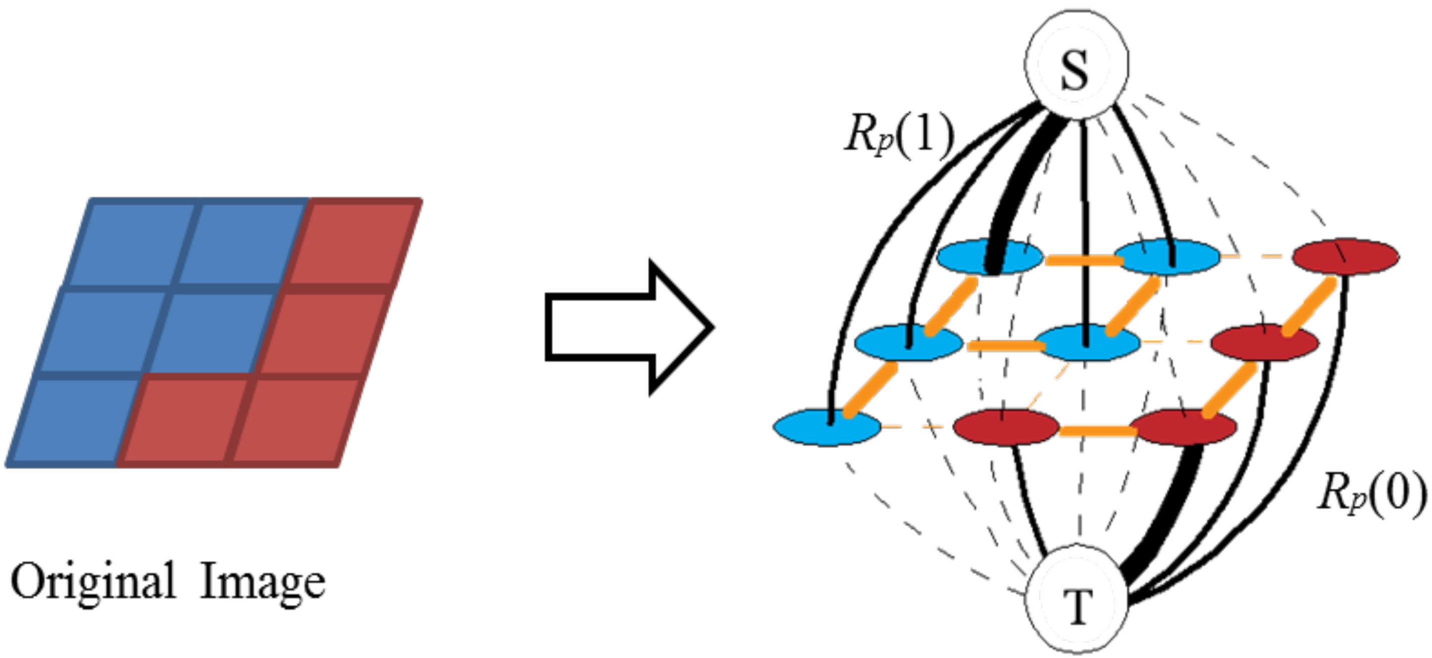
Interactive Segmentation

- Graph Cut Based Interactive Segmentation
  - Graph G = (V, E) -- V: vertex set, E: edge set
  - Flow Network : additional vertex S (source) and T (sink)
  - Max-Flow/Min-Cut Algorithm



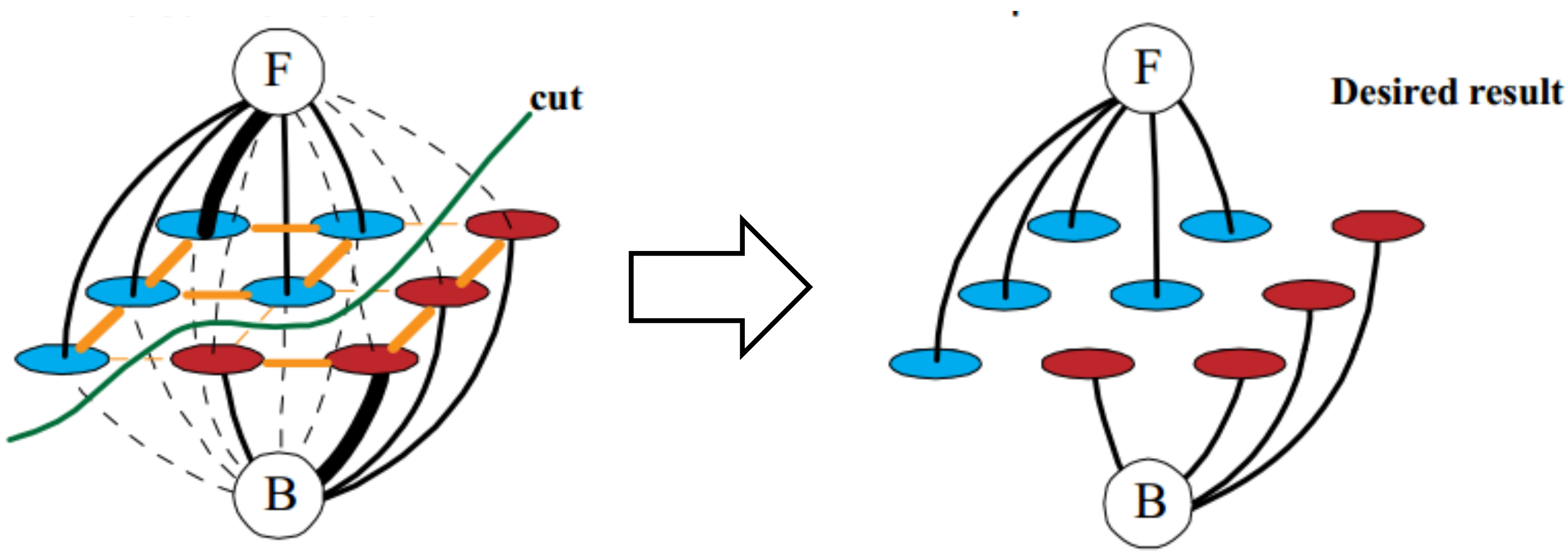
Graph Cut Based Interactive Segmentation

- Graph G = (V, E) -- V: vertex set, E: edge set
- Flow Network : additional vertex S (source) and T (sink)
- Max-Flow/Min-Cut Algorithm



Graph Cut Based Interactive Segmentation

- Graph G = (V, E) -- V: vertex set, E: edge set
- Flow Network : additional vertex S (source) and T (sink)
- Max-Flow/Min-Cut Algorithm

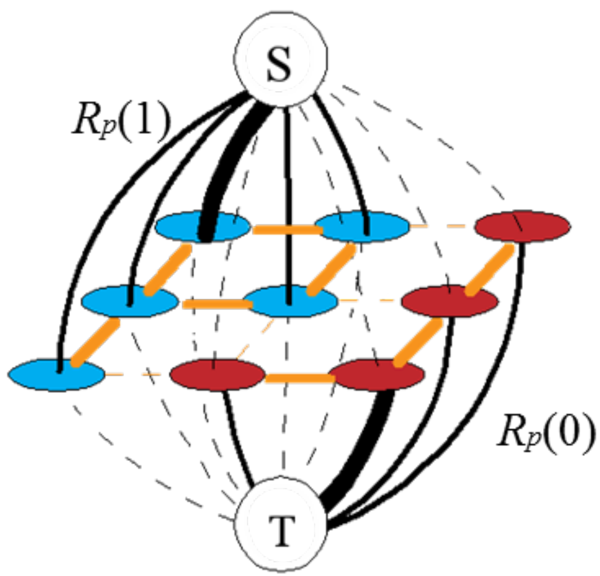


Graph Cut Based Interactive Segmentation

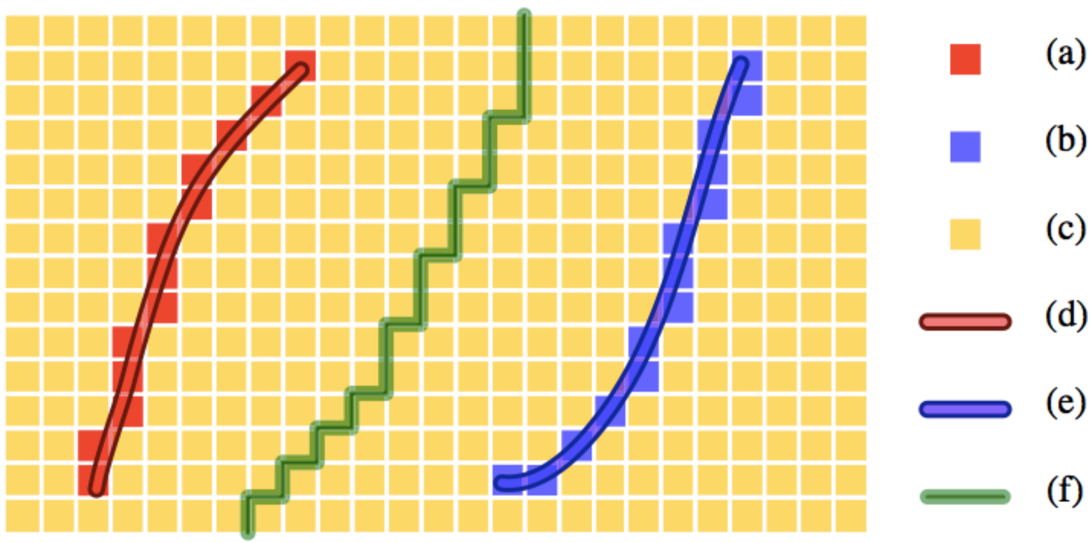
- solve the labeling problem by minimizing the energy function.

$$E(L) = \sum_{p} R_p(l_p) + \lambda \cdot \sum_{\{p,q\} \in N} B_{p,q} \cdot \delta(l_p, l_q)$$

- $R_p(l_p)$: regional term
- $B_{p,q}$: boundary term
- l_p : label of node p
- $\delta(l_p, l_q) = \begin{cases} 1 & \text{if } l_p \neq l_q \\ 0 & \text{if } l_p = l_q \end{cases}$
- N : neighborhood



Graph Cut Based Interactive Segmentation



(a) Foreground seeds $\mathcal{F}$ (b)Background seeds $\mathcal{B}$ (c)Uncertain regions $\mathcal{U}$
(d) Foreground marker (e) Background marker (f) Graph cut result

Graph Cut Based Interactive Segmentation

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

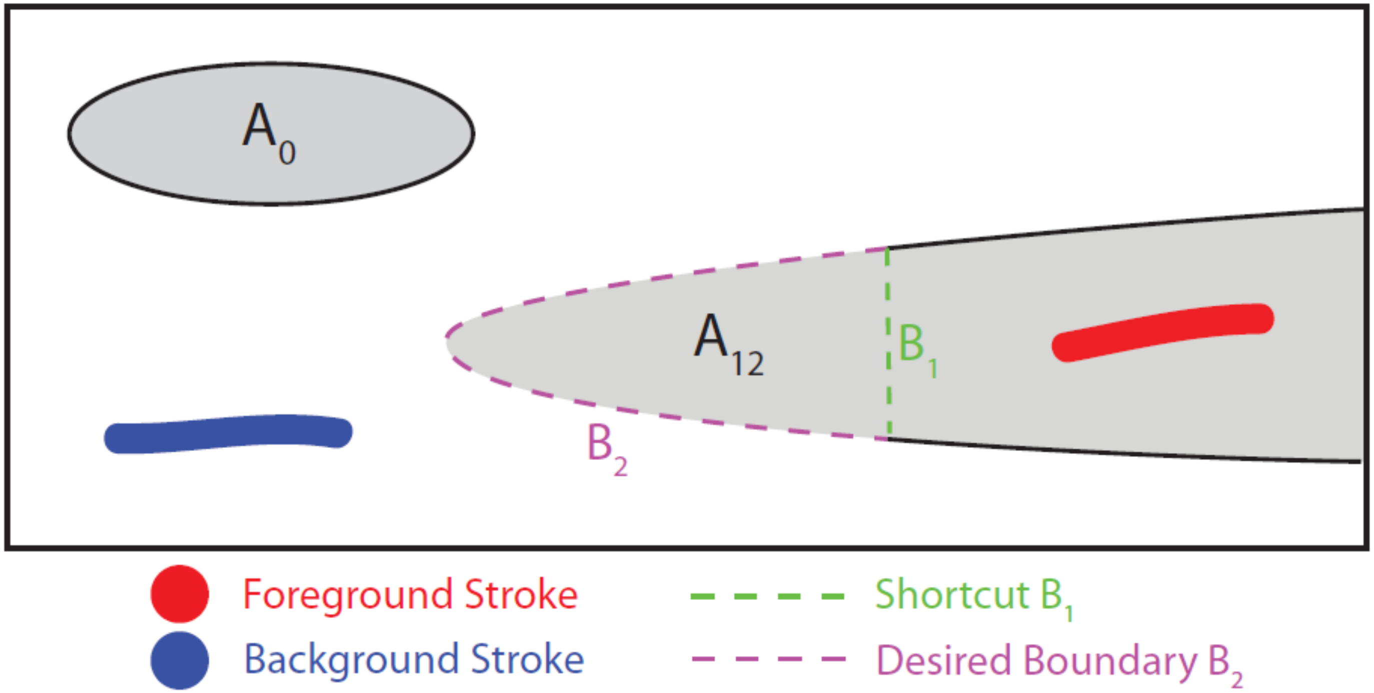
Red marker: foreground
Blue marker: background

Graph Cut Based Interactive Segmentation

- Issues in Graph Cut Algorithm
  - Shrinking Bias

$$E(L)=\sum_p R_p(l_p)+\lambda\cdot\sum_{\{p,q\}\in N} B_{p,q}\cdot\delta(l_p,l_q)$$

λ is a magical constant



Graph Cut Based Interactive Segmentation

- Issues in Graph Cut Algorithm
  - Shrinking Bias

Graph Cut Based Interactive Segmentation

- Issues in Graph Cut Algorithm
  - Shrinking Bias
  - Vulnerability to Noise
  - Efficiency

Graph Cut Based Interactive Segmentation

- Extensions of Graph Cut Algorithm
  - Interactive based
  - Shape prior based
  - Speed-up based

Superpixel

- Definition: a perpetually uniform region in the image

+ Superpixel

- Evaluated by :
  - Boundary Recall: measures the percentage of the natural boundaries recovered by the superpixel boundaries
  - Undersegmentation Error: compares segment areas to measure to what extend superpixels flood over the ground truth segment borders.
  - Running Time

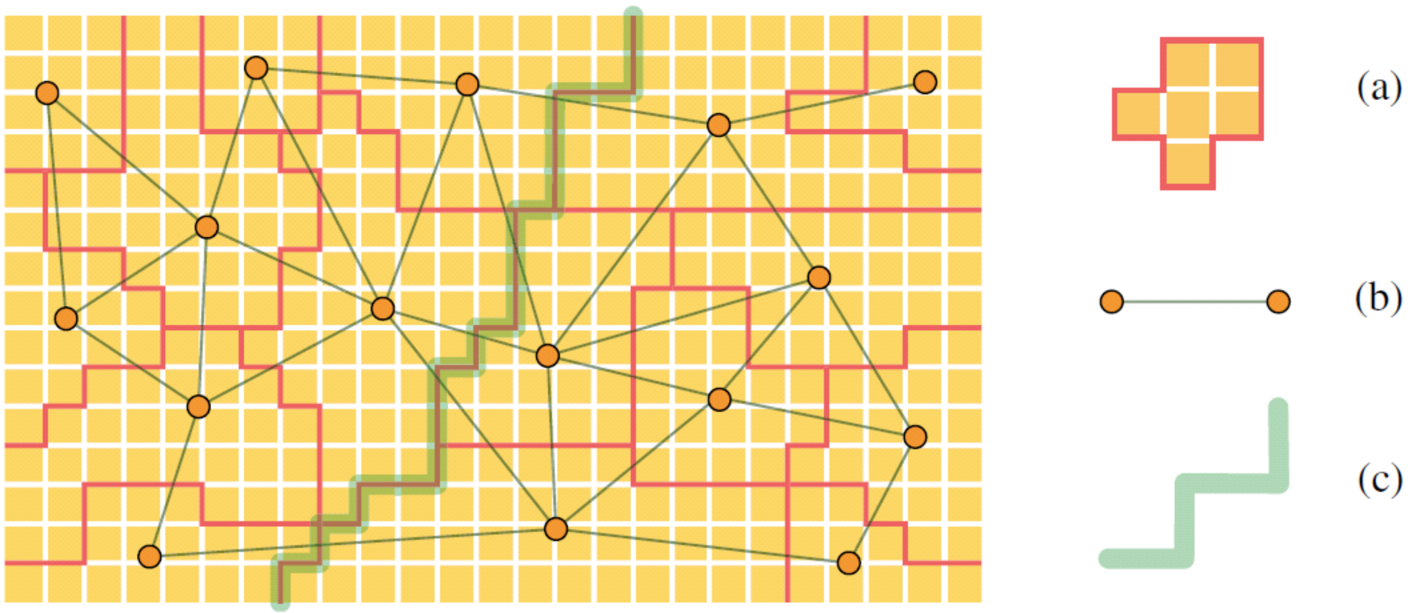
Interactive Segmentation

- Lazy Snapping [Li et al. 2004]

(a) Input image (b) Object Marking (c) Boundary editing (d) Output composition

Interactive Segmentation

- Lazy Snapping [Li et al. 2004]
  - Use watershed to do pre-segmentation

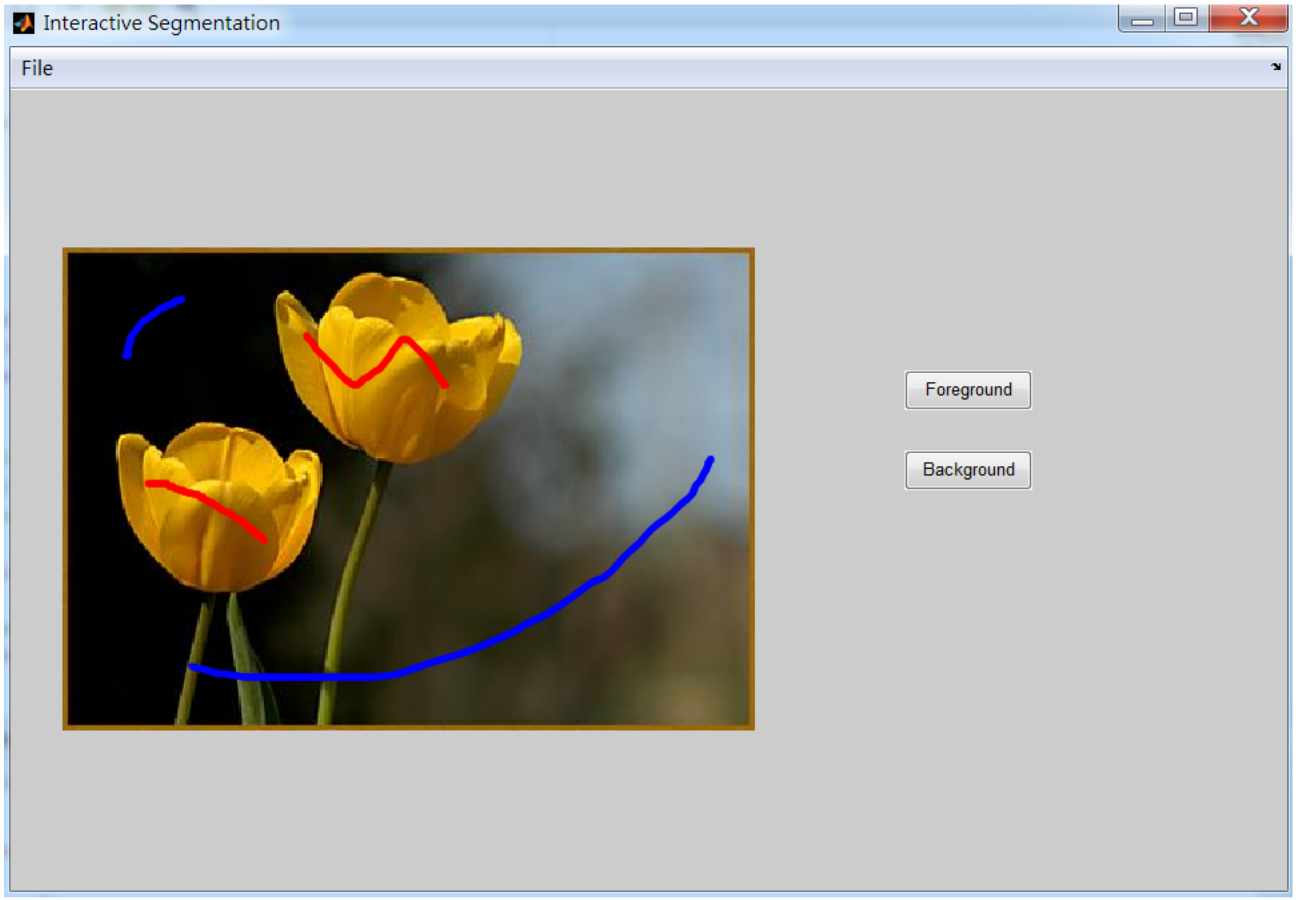


Proposed Method

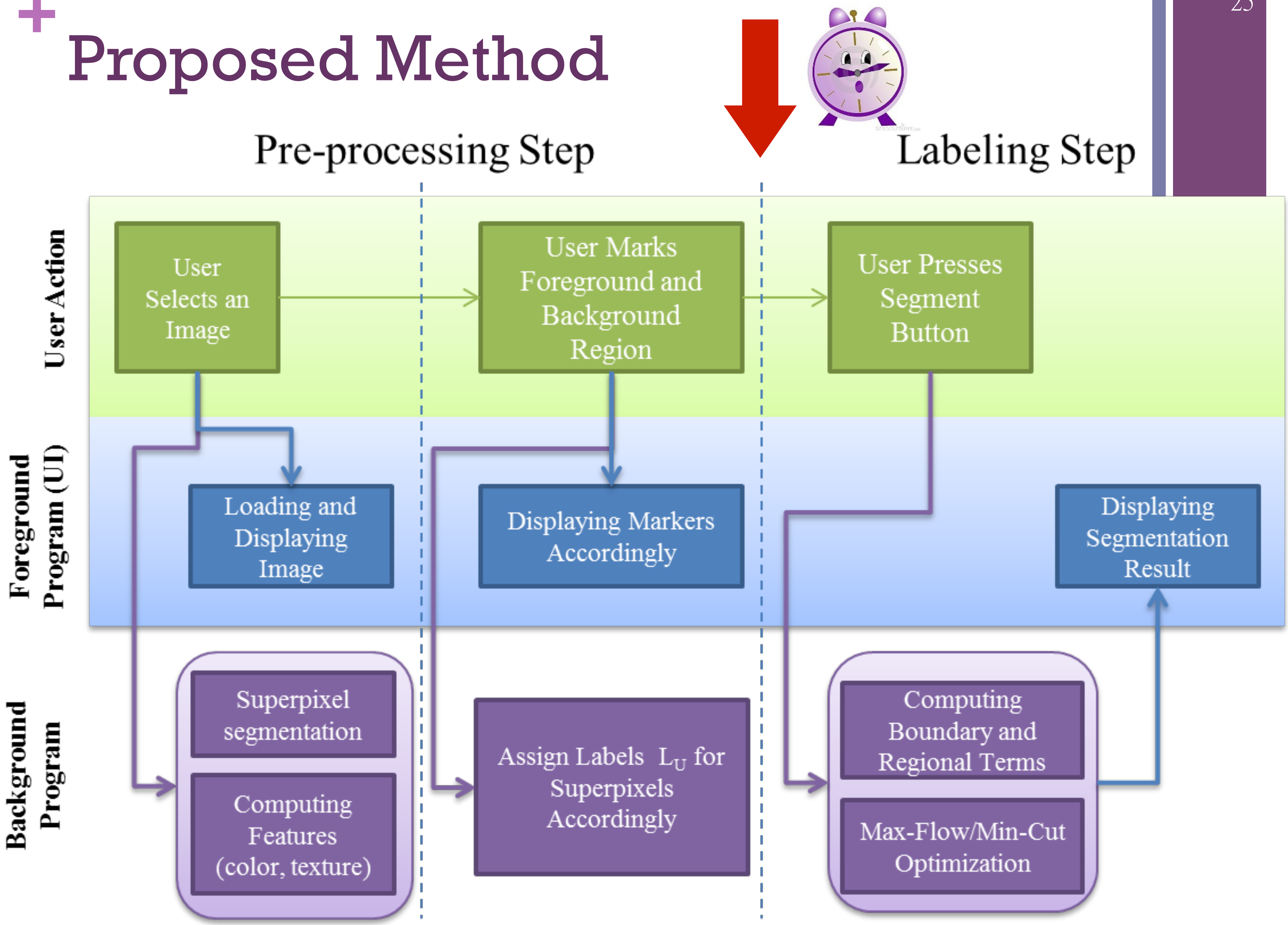
- Observation
  - User experience values short response time
  - Adopt superpixel pre-segmentation :
    - should use superpixel method with higher quality

Proposed Method

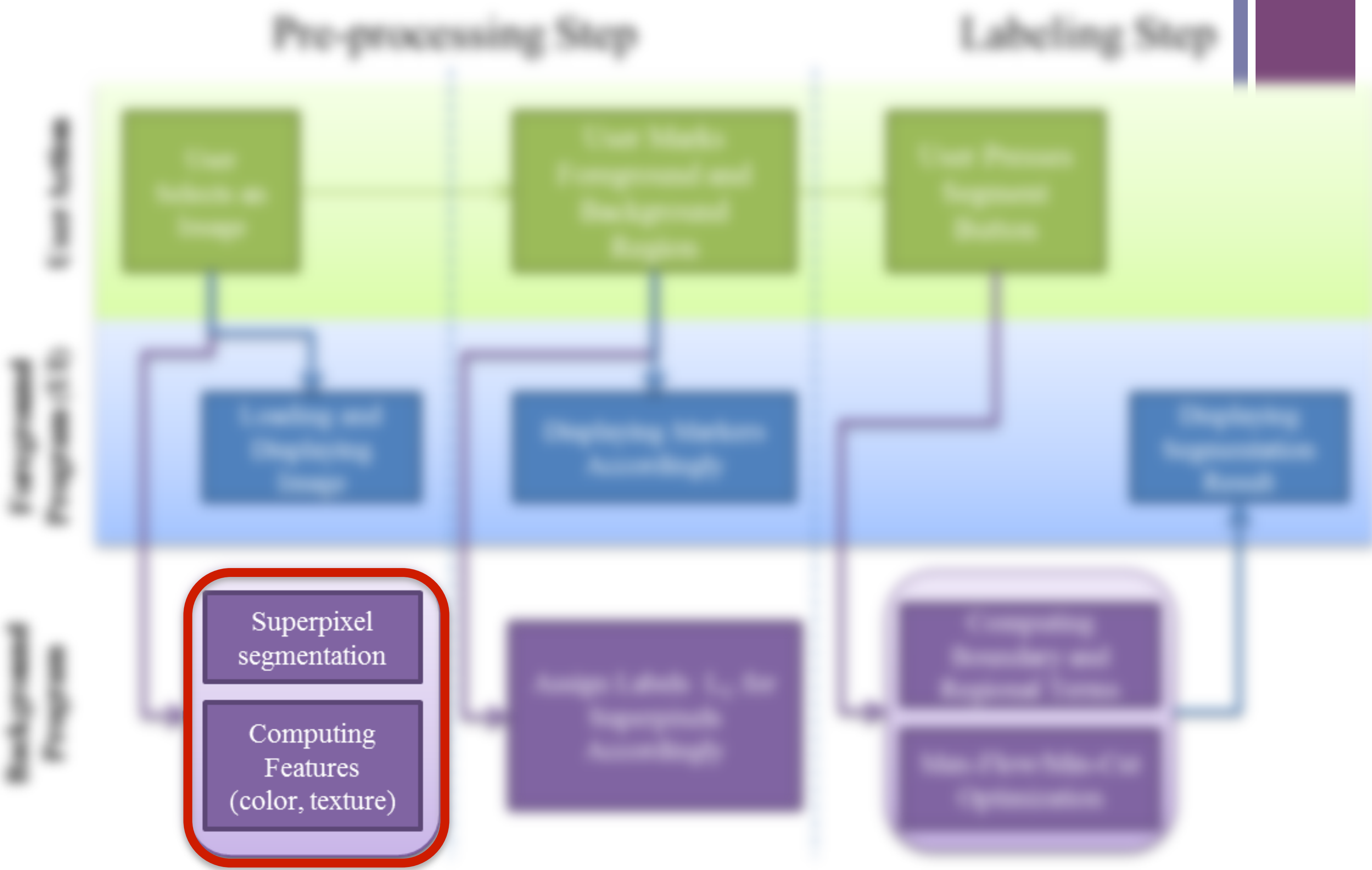
- Prototype UI



Proposed Method



Proposed Method



Proposed Method

Superpixel segmentation

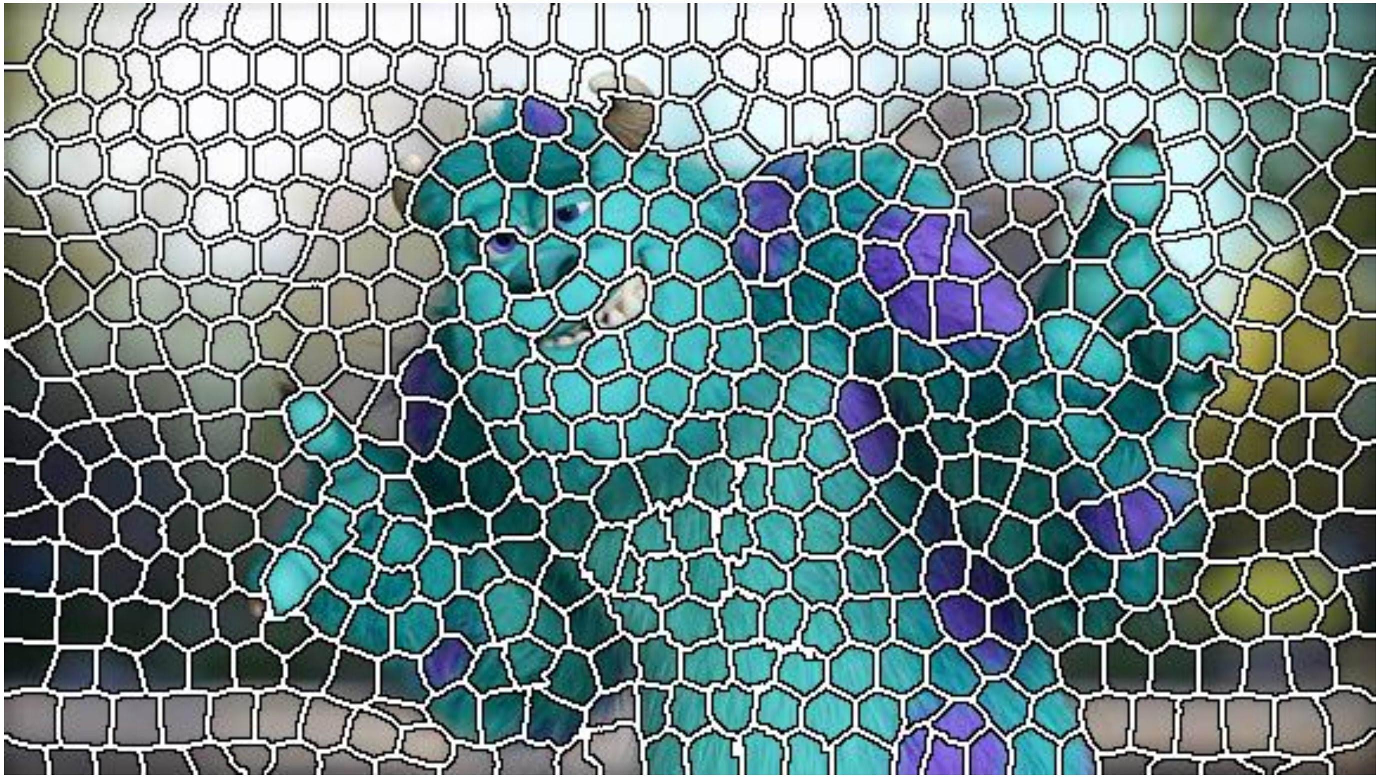
- Desired property
  - Given superpixel number control
  - Produce superpixels of similar size
  - Short processing time

- Consider all the factors above, SLIC (SLICO) and ERS would be our best candidates.

+ Proposed Method

Superpixel segmentation

- Candidates
  - Simple Linear Iterative Clustering – Zero (SLICO)

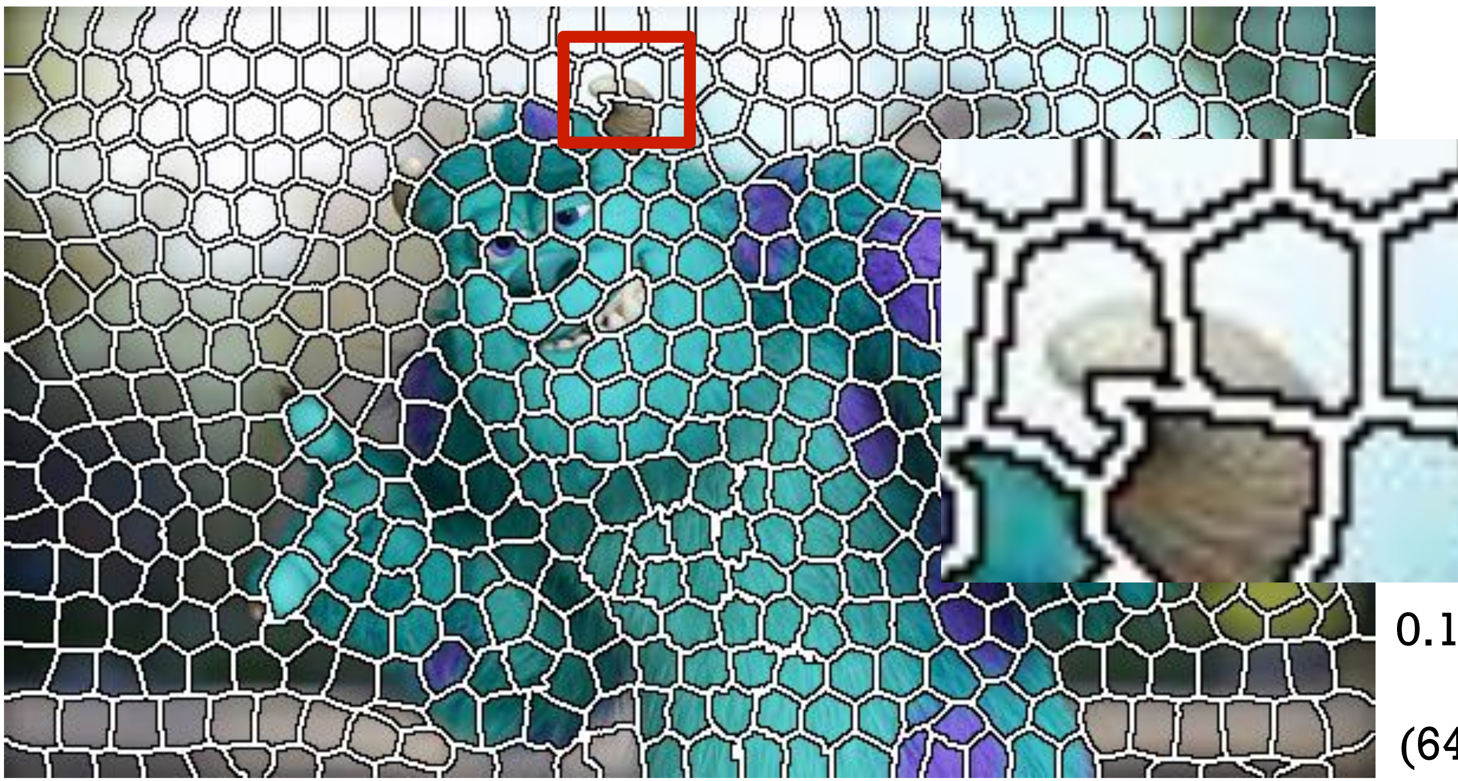


(645 x 363)

+ Proposed Method

Superpixel segmentation

- Candidates
  - Simple Linear Iterative Clustering – Zero (SLIC0)



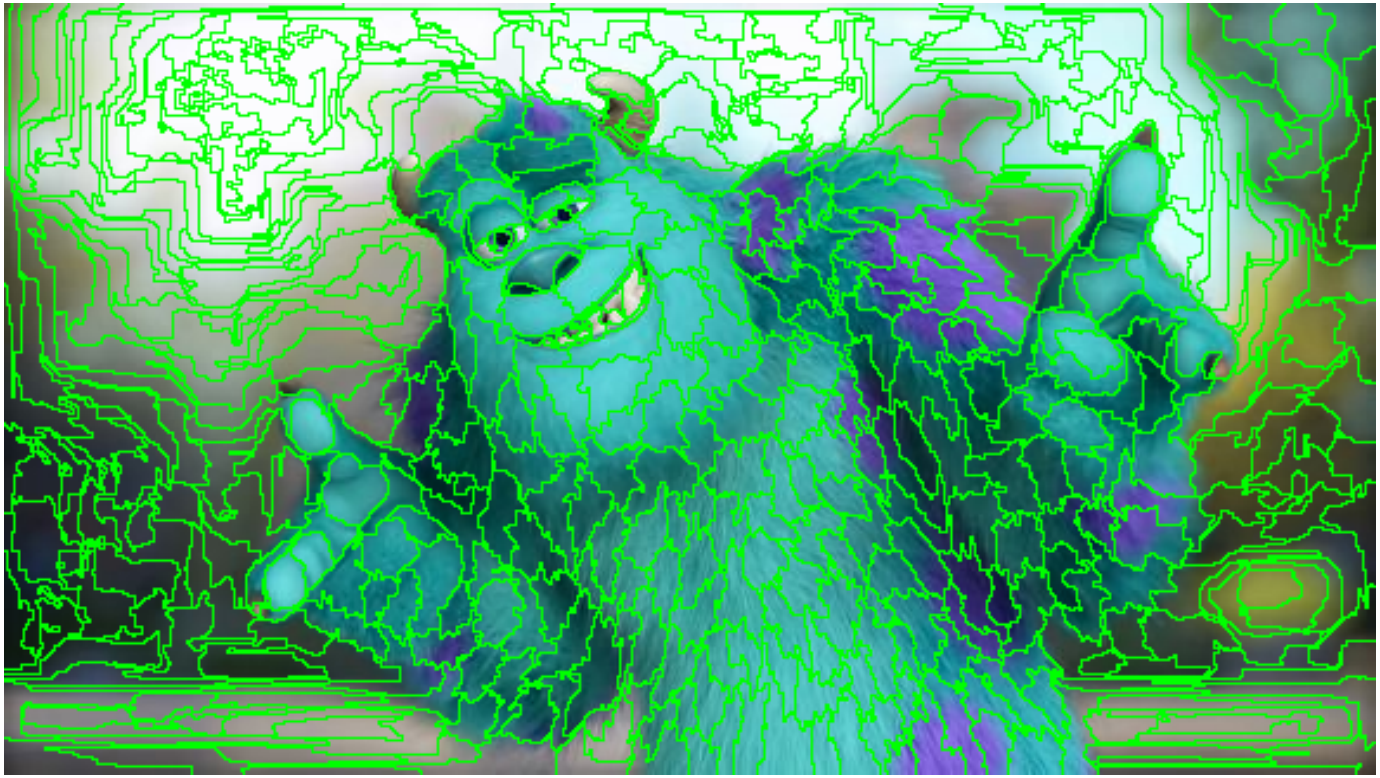
0.18s

(645 x 363)

+ Proposed Method

Superpixel segmentation

- Candidates
  - Entropy Rate Superpixel (ERS)

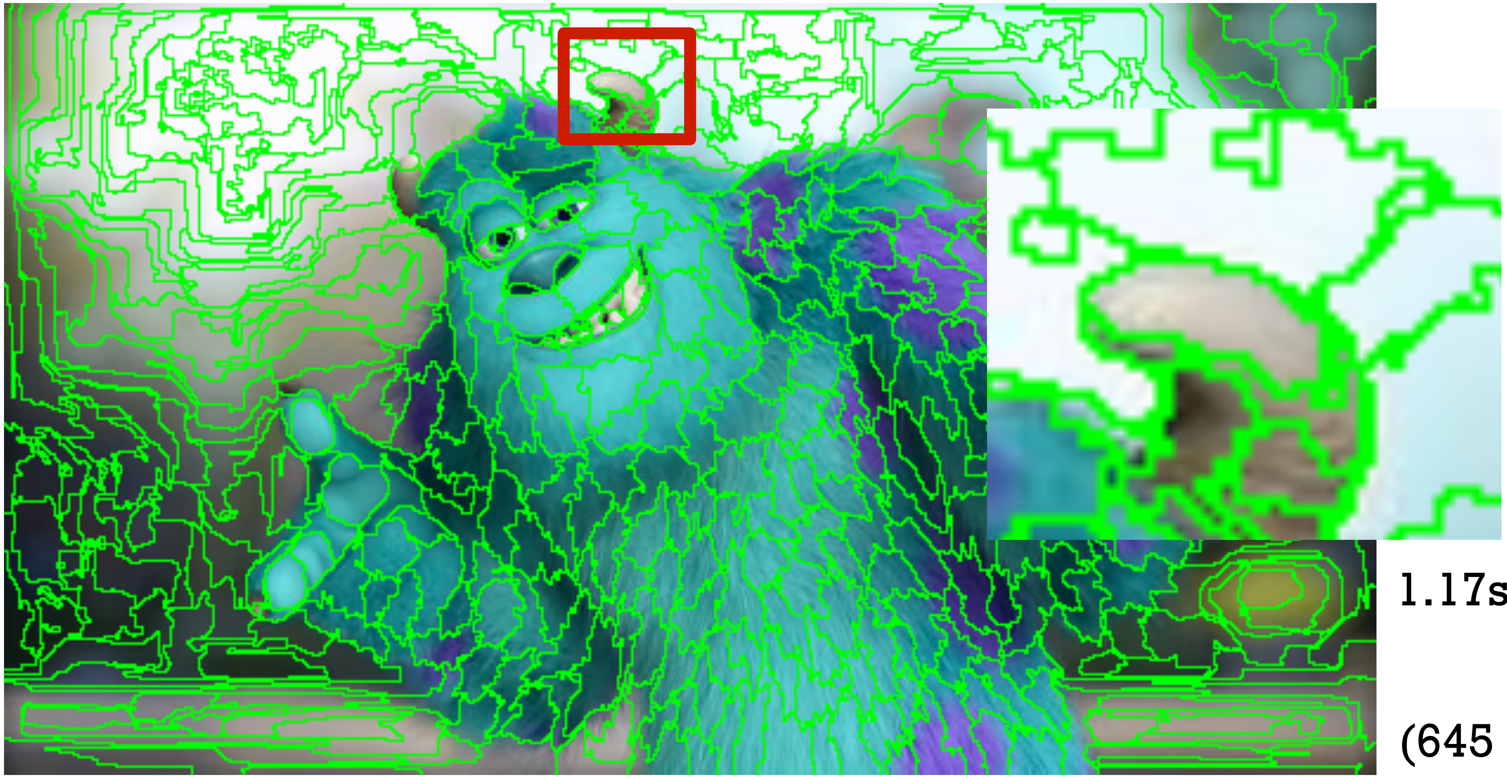


(645 x 363)

Proposed Method

Superpixel segmentation

- Candidates
  - Entropy Rate Superpixel (ERS)



1.17s

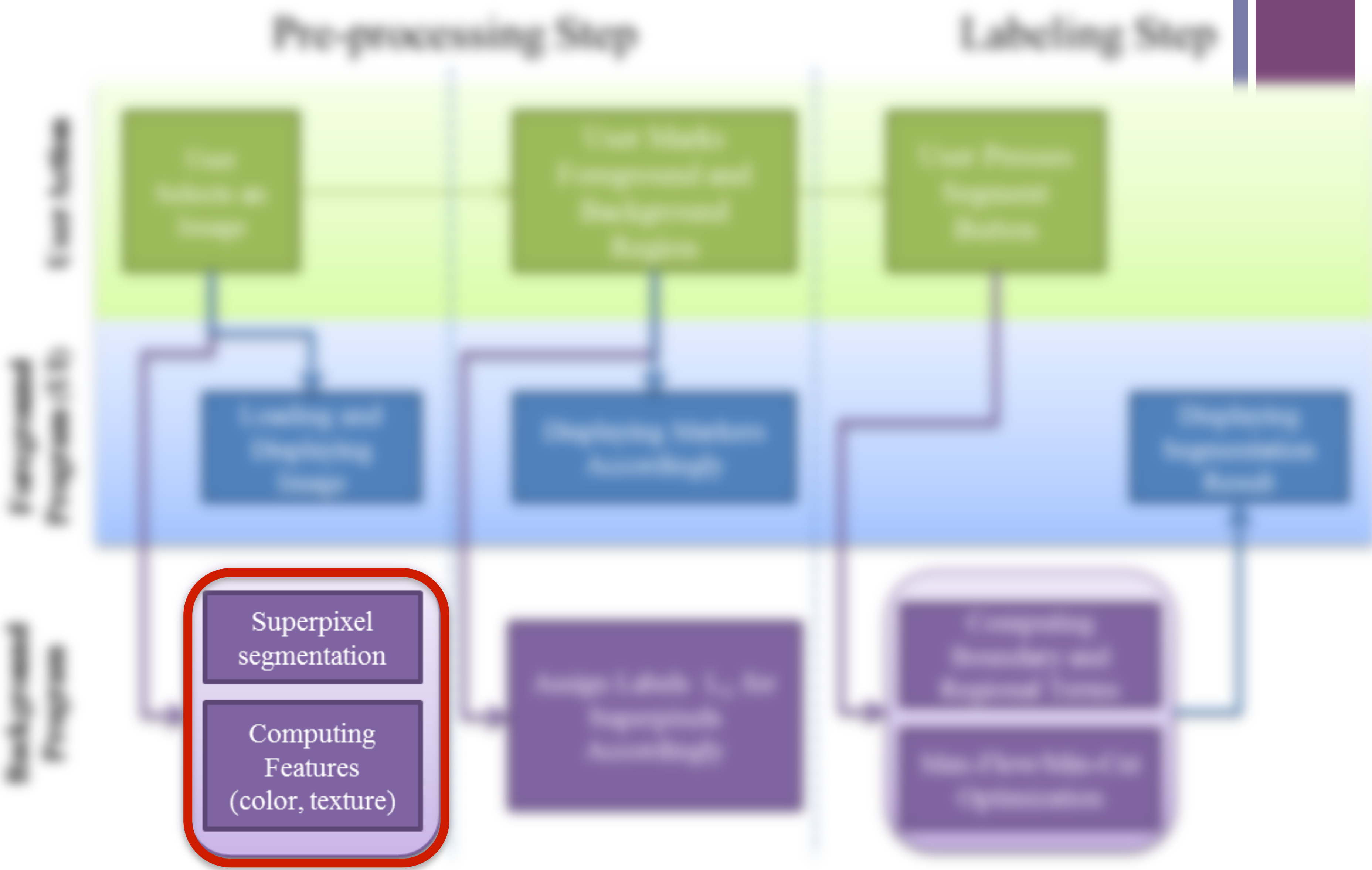
(645 x 363)

+ Proposed Method

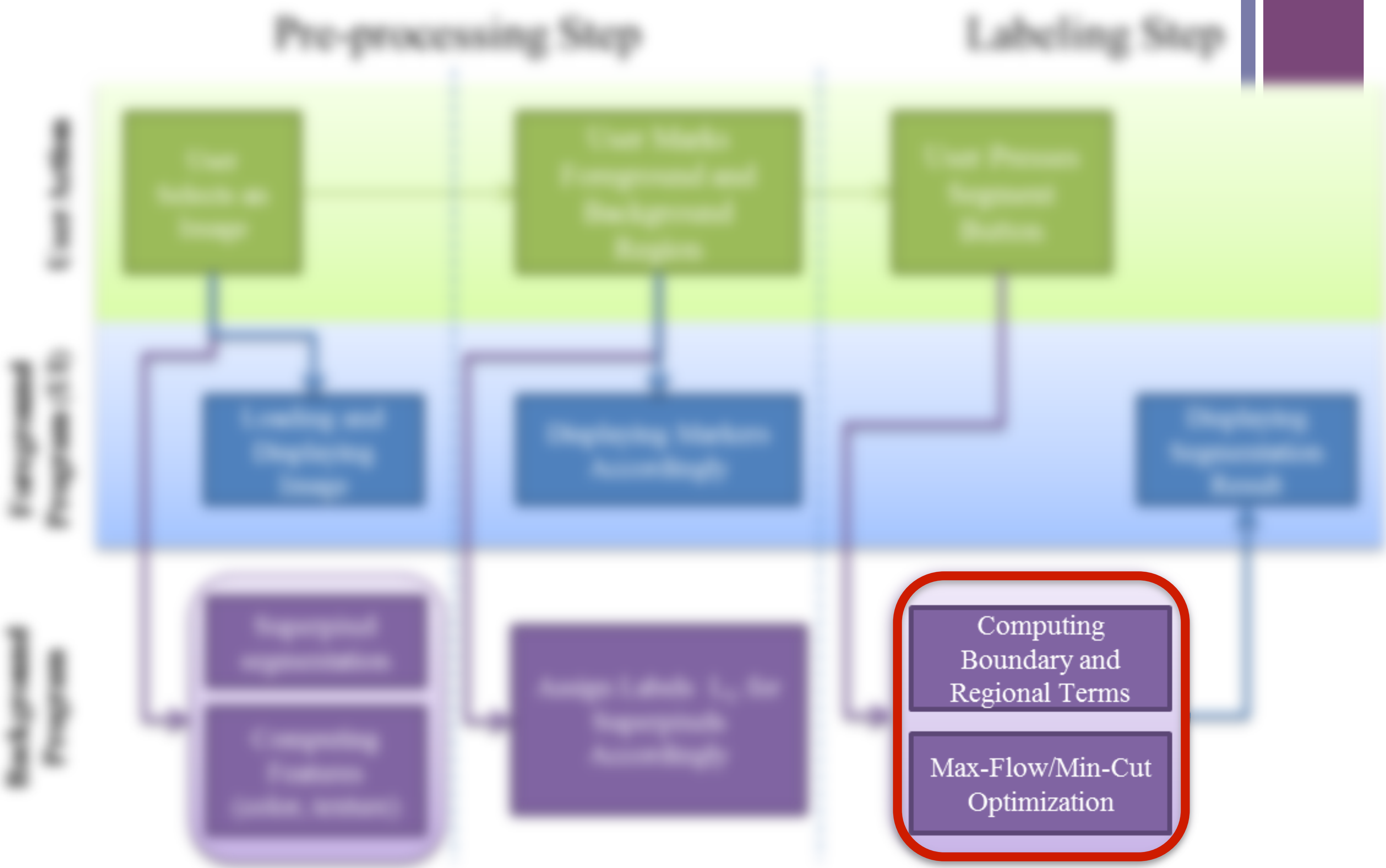
Compute Features (color, texture)

- Color Feature
  - mean value in LAB space (dim = 3)
- Texture Feature
  - mean value of Log-Gabor response
  - scale = 2, orientation = 4 (dim = 8)

Proposed Method



Proposed Method



+ Proposed Method

Compute
B and *R*

- Define Reginal Term in Graph Cut Algorithm

$$E(L) = \sum_{p} R_p(l_p) + \lambda \cdot \sum_{\{p,q\}\in N} B_{p,q} \cdot \delta(l_p, l_q)$$

  - Color distance

$$dC_j^F = \min_n \| C(j) - C_n^F \|$$

$$dC_j^B = \min_n \| C(j) - C_n^B \|$$

- **C(j): mean [lab] value of superpixel j**
- C_n^F **: mean [lab] value of foreground superpixel j**

  - Texture distance

$$dT_j^F = \min_n \| T(j) - T_n^F \|$$

$$dT_j^B = \min_n \| T(j) - T_n^B \|$$

- **C(j): mean Log-Gabor value of superpixel j**
- T_n^F **: mean Log-Gabor value of foreground superpixel j**

Proposed Method

Compute
B and *R*

Define Regional Term in Graph Cut Algorithm

$$E(L)=\sum_{p} R_p(l_p)+\lambda\cdot\sum_{\{p,q\}\in N} B_{p,q}\cdot\delta(l_p,l_q)$$

Label 1 : foreground
Label 0 : background

	$l_i = 1$	$l_i = 0$
$i\in F$	0	∞
$i\in B$	∞	0
$i\in U$	$\frac{dC_i^F}{dC_i^F+dC_i^B}+\frac{dT_i^F}{dT_i^F+dT_i^B}\cdot\alpha$	$\frac{dC_i^B}{dC_i^F+dC_i^B}+\frac{dT_i^B}{dT_i^F+dT_i^B}\cdot\alpha$

- B : background
- F : foreground
- U=V\{F∪B}

- dC : color distance
- dT : texture distance

+ Proposed Method

Compute
B **and** ***R***

- Define Boundary Term in Graph Cut Algorithm

$$E(L) = \sum_p R_p(l_p) + \lambda \cdot \sum_{\{p,q\} \in N} B_{p,q} \cdot \delta(l_p, l_q)$$

$$\mathrm{B}_{p,q} = \frac{1}{\left\| C(p) - \mathrm{C}(q) \right\|^2 + 1}$$ [Li et al. 2004]

  - $R_p(l_p)$: regional term
  - $B_{p,q}$: boundary term
  - l_p : label of node p
  - $\delta(\mathrm{l}_p, \mathrm{l}_q) = \begin{cases} 1 & if\ l_p = l_q \\ 0 & if\ l_p \neq l_q \end{cases}$

Proposed Method

Max-Flow/Min-Cut Optimization

- Max-Flow/Min-Cut Optimization
  - [Boykov & Kolmogorov 2004]

Experimental Results

- Motivation Revisited

Node Ratio = *#pixels/ #superpixels*

Ideal

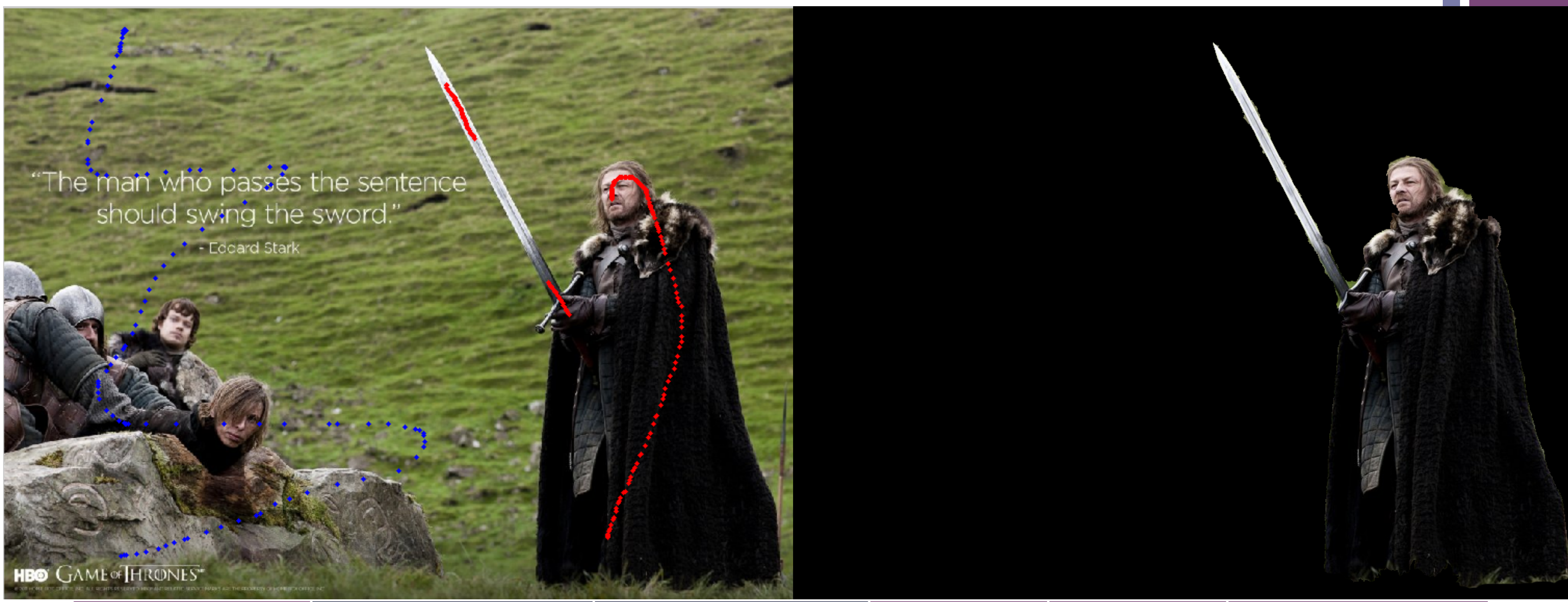
Ours

Experimental Results

- Motivation Revisited

step / image size	Superpixel (Node Ratio = 100)	Computing Features	B & R	Max-Flow/ Min-Cut	Total Delay
645x363	0.182s (1.17)	1.217s	0.044s	0.018s	0.062s

+ Experimental Results



step / image size	Superpixel (Node Ratio = 200)	Feature extraction	B & R	Graph Cut	Total Delay
1024x768	0.896s (4.96)	1.021s	0.120s	0.053s	0.173s

Experimental Results

Experimental Results

- Time Delay Comparison with Lazy Snapping

Li et al. [2004]

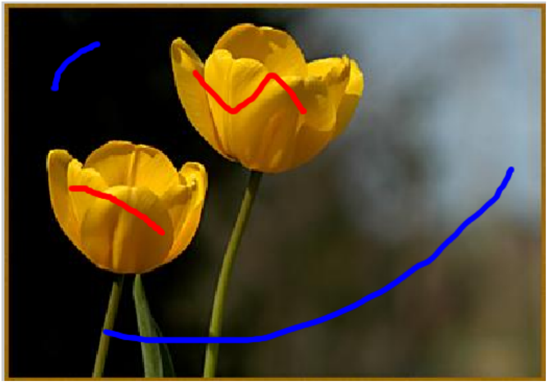
Image Size	Node Ratio	Response Time	Response Time without Pre-segmentation
408x600	11	0.12	0.57s
440x800	11	0.21	1.39s
1024x768	21	0.25	1.82s
1147x768	19	0.22	3.56s

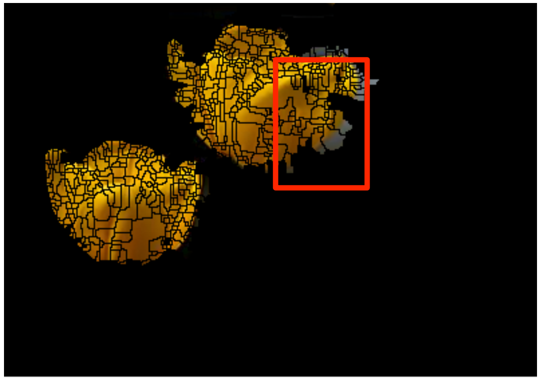
Ours

Image Size	Node Ratio	Response Time
645x363	100	0.062s
1024x768	200	0.173s

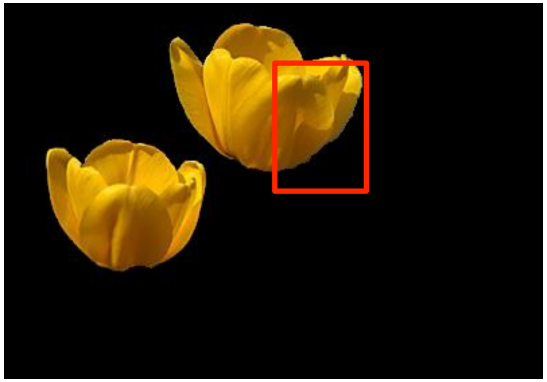
Experimental Results

- Visual Comparison with Lazy Snapping
  - Object has distinct color to background

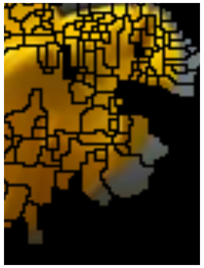




[Li et al. 2004]



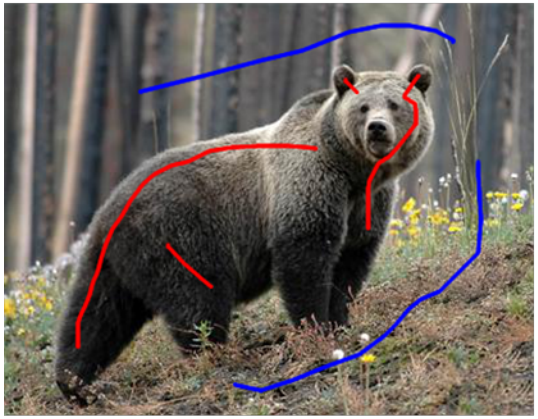
Ours



(SLICO, Node Ratio = 300)

Experimental Results

- Visual Comparison with Lazy Snapping
  - Object has similar color to background



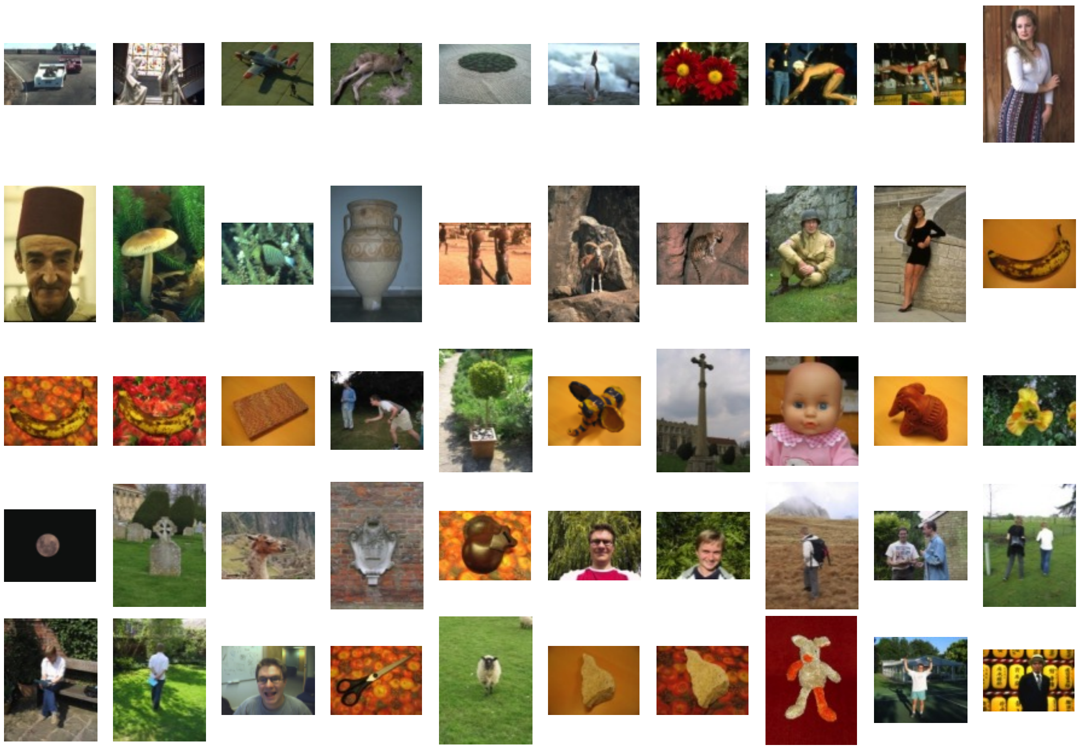
[Li et al. 2004]

Ours

(SLICO,
Node Ratio = 300)

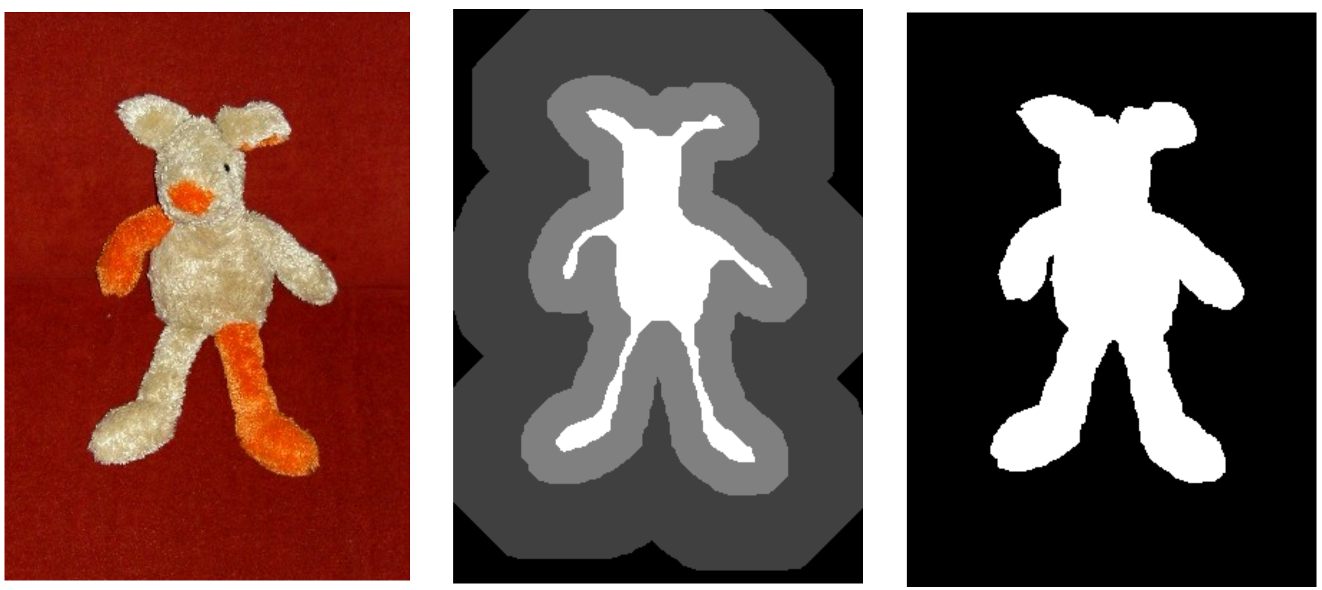
Experimental Results

- GrabCut Database [Rother et al. 2004]



Experimental Results

- GrabCut Database [Rother et al. 2004]
  - $Error\ Rate = \#misclassified\ pixels / \#unknown\ area\ pixels$



+ Experimental Results

- **GrabCut Database** [Rother et al. 2004]
  - *Error Rate* = *#misclassified pixels/#unknown area pixels*
  - *Achievable Segmentation Error(ASE)*

  =the lowest error rate we can achieve based on superpixel segmentation result
  - *Node Ratio* = *#pixels/#superpixels*

+ Experimental Results

Segmentation Model	Error Rate(%)	Response Time
Li et al. (Lazy Snapping) [2004]	15.1	*0.005s-1.129s, mean 0.339
Grady (Random Walker) [2006]	10.9	0.15s - 0.83s, mean 0.47s (C++ with MATLAB wrapper)
Couprie et al. (Powerful Watershed) [2009]	10.4	0.12s - 0.36s, mean 0.25s (C++)
Our Method without using texture information (ERS, Node Ratio = 100)	8.2	-
Proposed Method (ERS, Node Ratio = 100)	**7.7**	**0.020s – 0.101s, mean 0.058s (C++/MATLAB)**
Graph Cut (Vicente et al.) [2008]	6.7	-
Duchenne et al. (Segmentation by Transduction) [2008]	5.4	2s – 3mins (C++)
Price et al. (Geodesic Graph Cut) [2010]	4.8	0.2s – 2.6s (C++)

Experimental Results

Trimap

Li et al. [2004]

Duchenne et al. [2008]

Grady [2008]

Couprie et al. [2009]

Proposed Method

Experimental Results

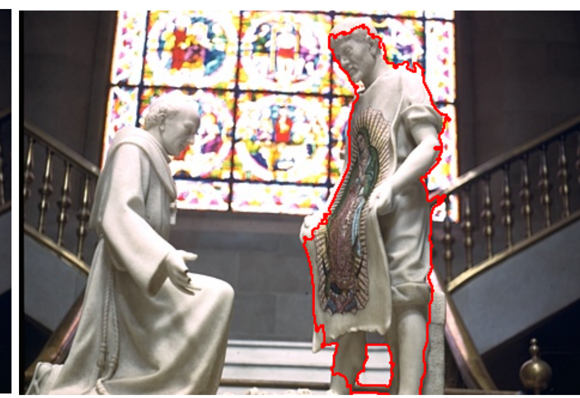
Trimap

Li et al. [2004]

Duchenne et al. [2008]

Couprie et al. [2009]

Grady [2006]



Proposed Method

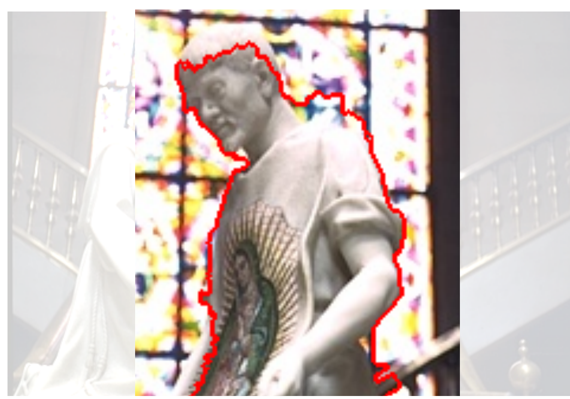
Experimental Results

Trimap

Li et al. [2004]

Duchenne et al. [2008]

Couprie et al. [2009]



Grady [2006]

Proposed Method

Experimental Results

Trimap

Li et al. [2004]

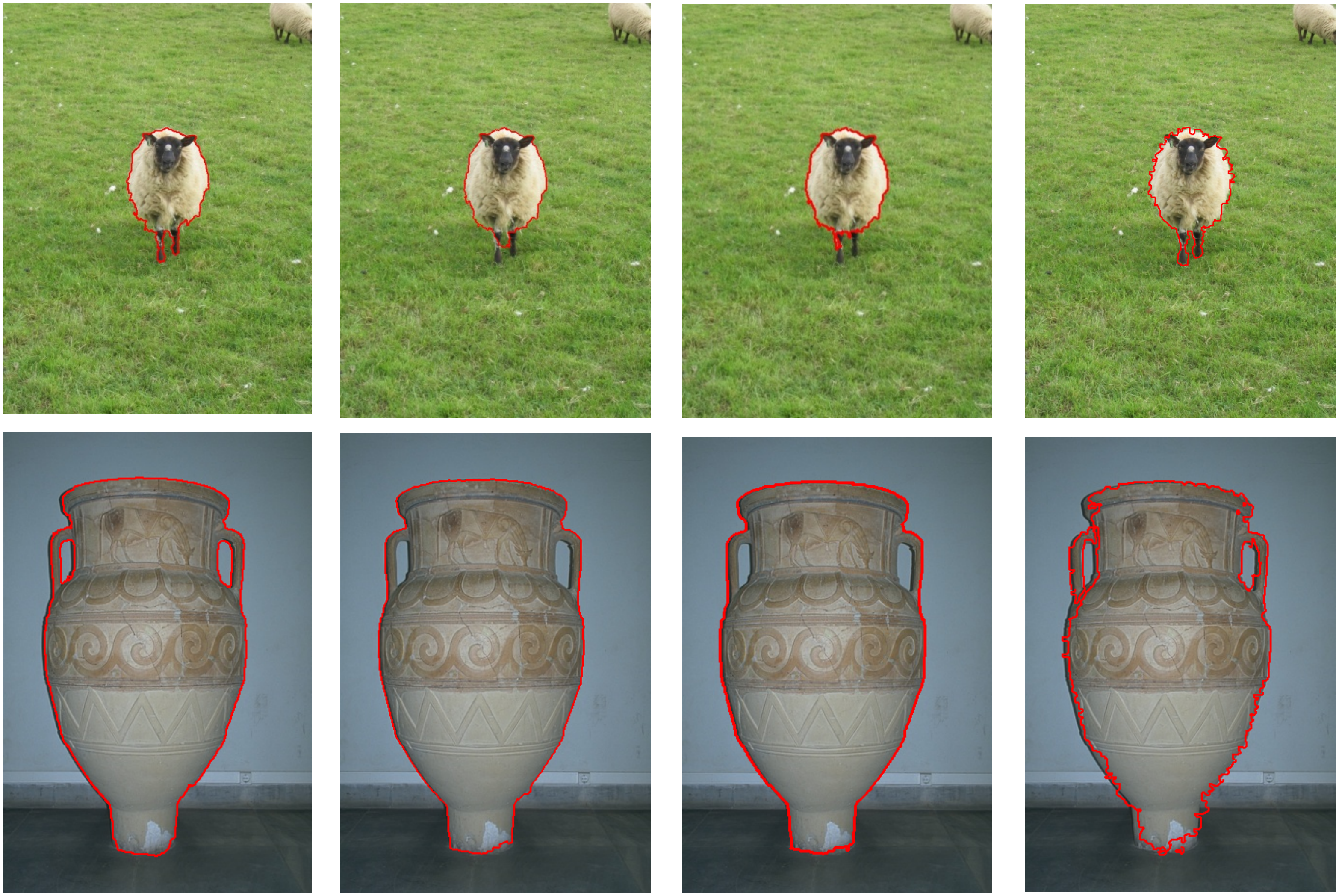
Duchenne et al. [2008]

Couprie et al. [2009]

Grady [2006]

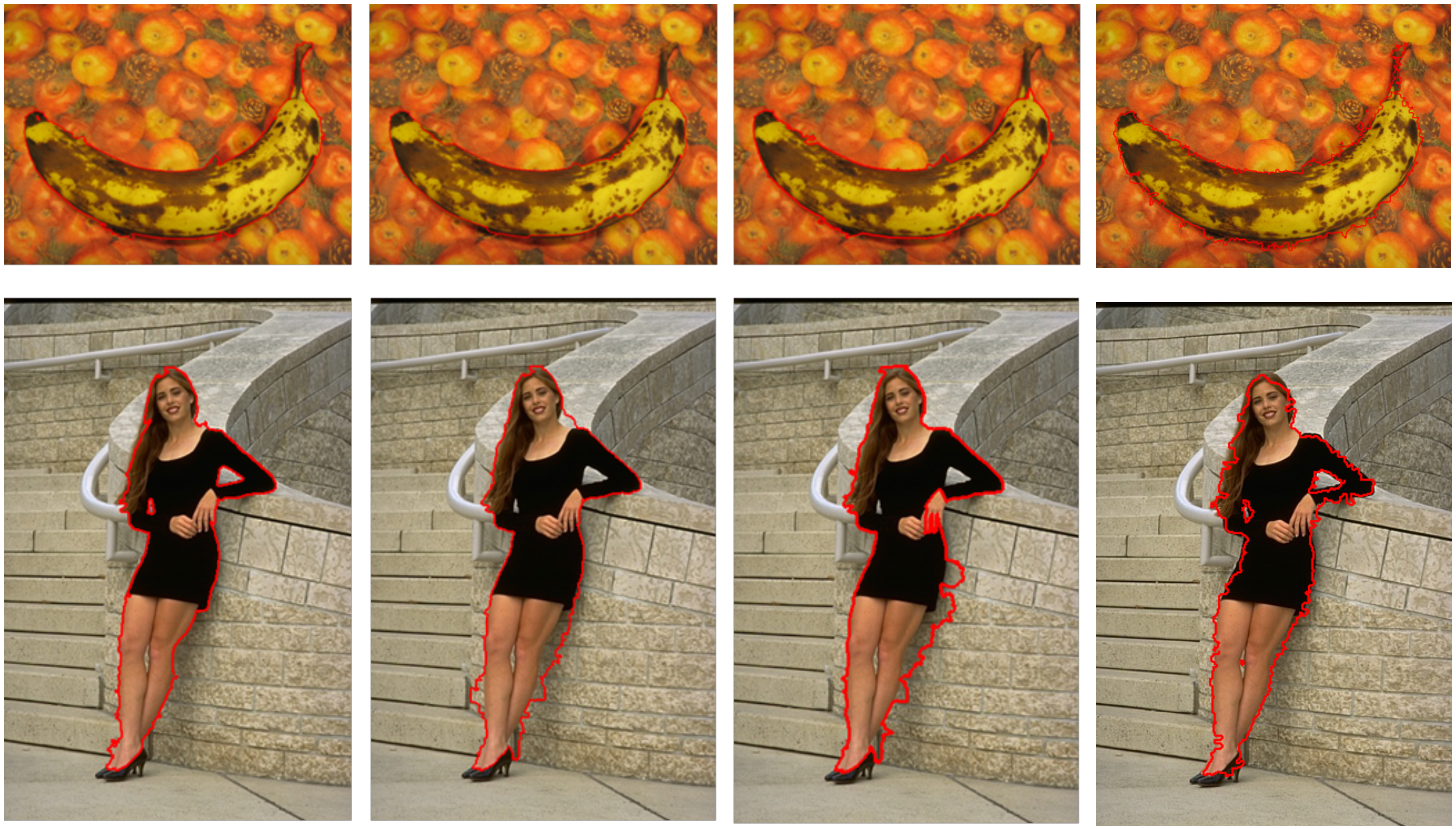
Proposed Method

Experimental Results



Proposed Method | Grady [2008] | Couprie et al. [2009] | Li et al. [2004]

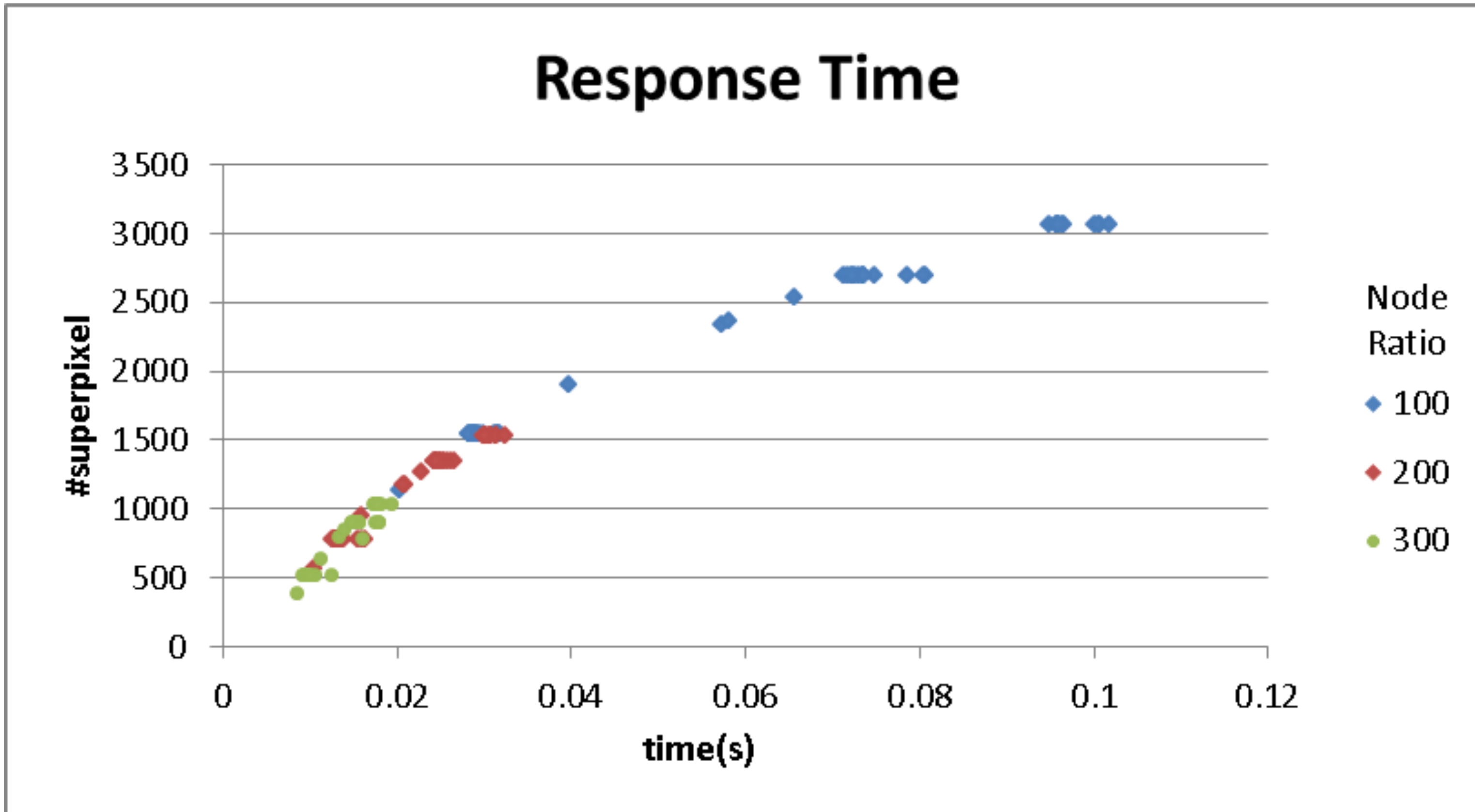
Experimental Results



Proposed Method | Grady [2008] | Couprie et al. [2009] | Li et al. [2004]

Discussion

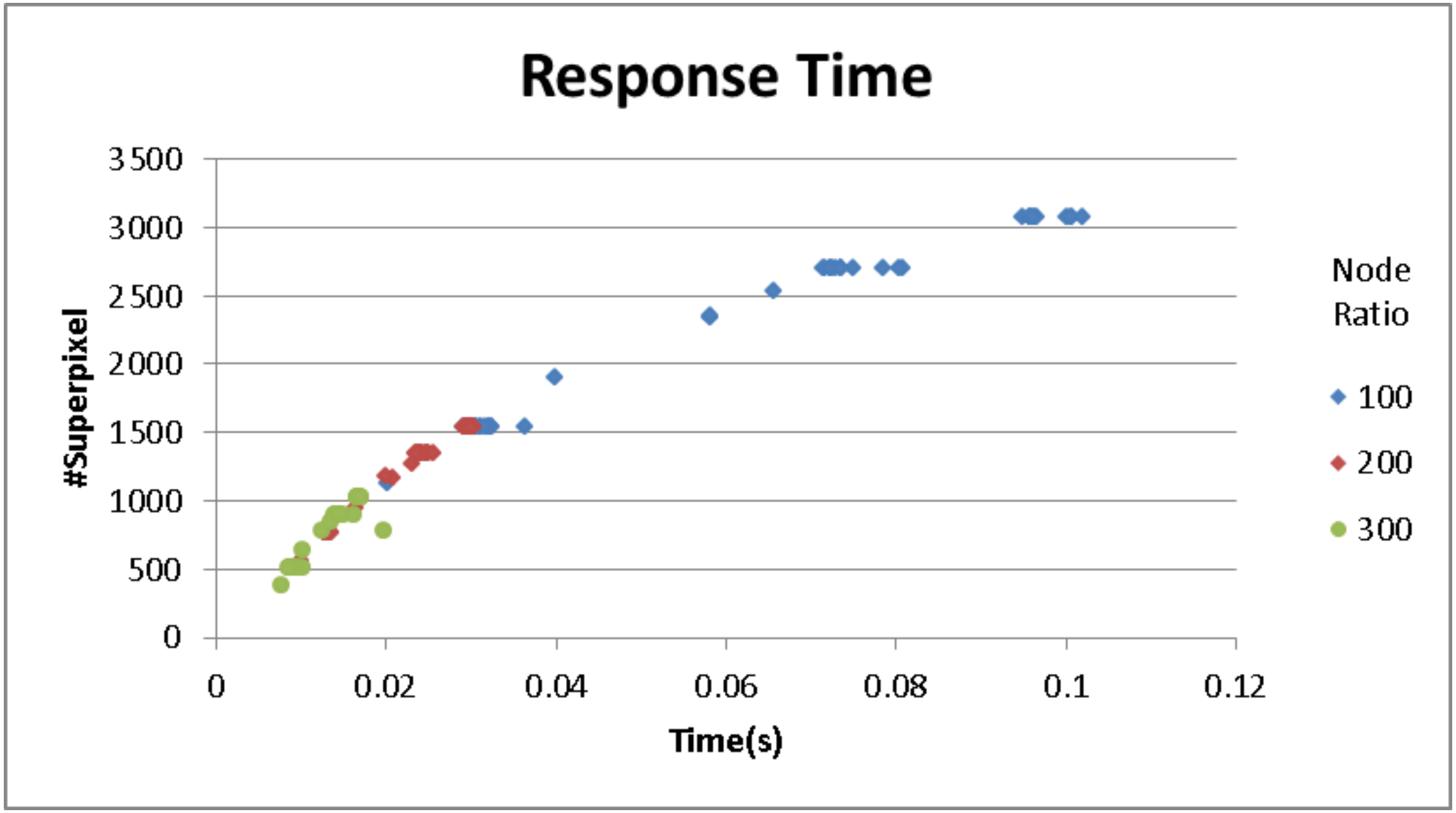
- Processing Time



with ERS pre-segmentation

Discussion

- Processing Time



with SLICO pre-segmentation

+ Discussion

- Processing Time
  - Pre-processing time (in seconds)

Process	Node Ratio	100	200	300
SLICO	Pre-segmentation	0.291	0.375	0.457
	Computing Features	1.249	1.043	0.856
	Preprocessing	1.540	1.418	1.313
ERS	Pre-segmentation	1.392	1.414	1.512
	Computing Features	1.117	0.817	0.717
	Preprocessing	2.509	2.231	1.289

Discussion

- Achievable Segmentation Error (ASE)

Algorithm	Node Ratio	100	200	300
ERS	ASE	4.9387	6.0141	6.6641
	Proposed Method	7.7543	8.1311	8.3464
SLICO	ASE	6.0416	6.8367	7.8556
	Proposed Method	8.4763	9.1628	9.9610

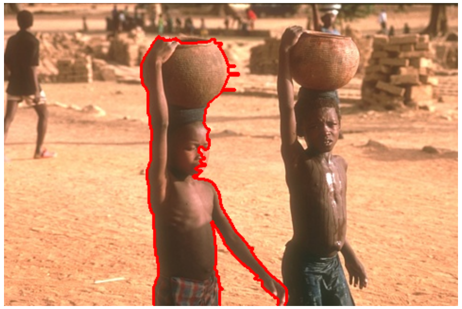
Discussion

- Graph Cut Segmentation Issues Revisited
  - Shrinking Bias

Proposed Method

Trimap

Grady [2008]



Couprie et al. [2009]

+ Discussion

- Graph Cut Segmentation Issues Revisited
  - Vulnerability to Noise

	Original	Add Gaussian noise with mean=0, variance = 0.001	Add Gaussian noise with mean=0, variance = 0. 01
Proposed Method	7.7	7.9	8.6
Grady	10.9	11.3	12.3
Couprie et al.	10.4	11.5	12.5

  - Efficiency

Conclusion

- We propose a framework that could further shorten interactive segmentation response time while maintaining satisfying quality.
- We propose using superpixel methods with better segmentation quality such as SLICO or ERS instead of watershed.
- We propose a region term for max-flow/min-cut optimization which uses the Log-Gabor texture feature to stabilize the quality of segmentation.

Future Work

- Efficiency:
  - Improve our implementation
- Solve Shrinking Bias Issue:
  - Adaptively adjust weight between regional and boundary term
  - Combine other interactive segmentation theory such as geodesic segmentation

A

Thank You

Q