Random Walks for Interactive Alpha-Matting

Leo Grady, Thomas Schiwietz, Shmuel Aharon
Department of Imaging and Visualization
Siemens Corporate Research

Rüdiger Westermann
Technische Universität München

Introduction

Problem: Determine the alpha-matte for a foreground/background blend

$$I = \alpha F + (1 - \alpha) B$$

Leading approaches are either complicated, with many free parameters (e.g., GrabCuts) or require significant additional user input (e.g., Poisson matting)

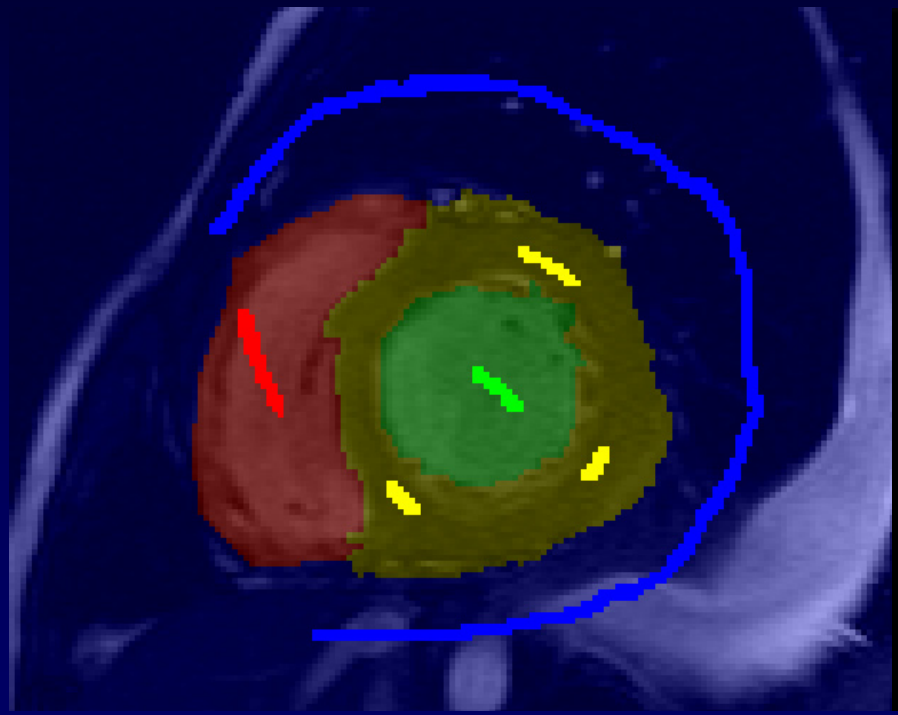
Proposal: Given a trimap of known foreground, known background and unknown pixels, treat the alpha values in the unknown region as the *probability that a random walker leaving those pixels first reaches a known foreground pixel before striking a background pixel*

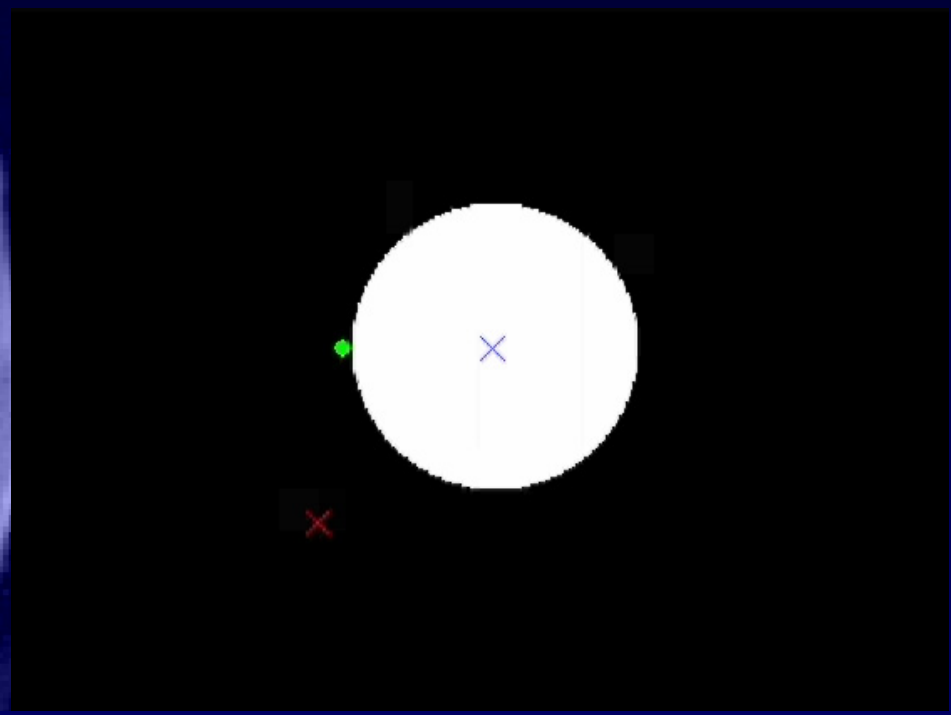
Outline

- Random walker algorithm
- Application to color images - LPP
- GPU Implementation
- Results
- Conclusion

Random Walker Segmentation

Given labeled voxels, for each voxel ask: What is the probability that a random walker starting from this voxel first reaches each set of labels?

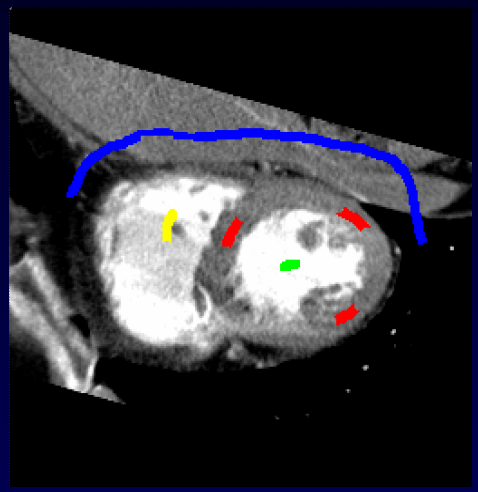




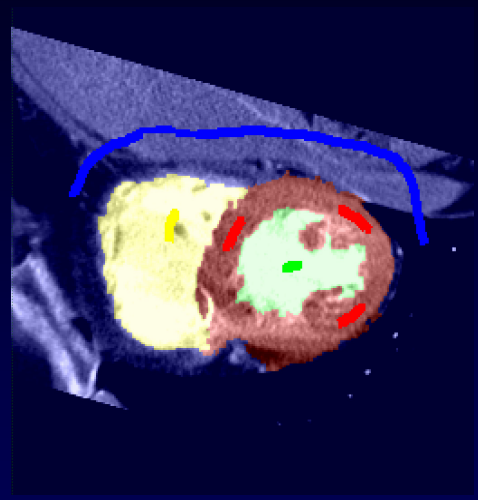
Can be computed analytically!

Random Walker Segmentation

Partially labeled image



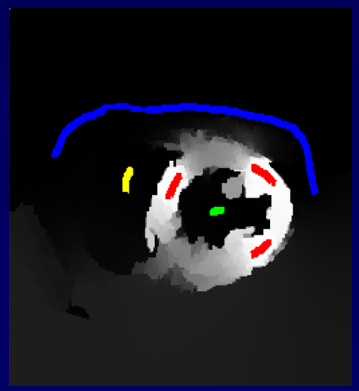
Segmented image



Probabilities

Green

Red

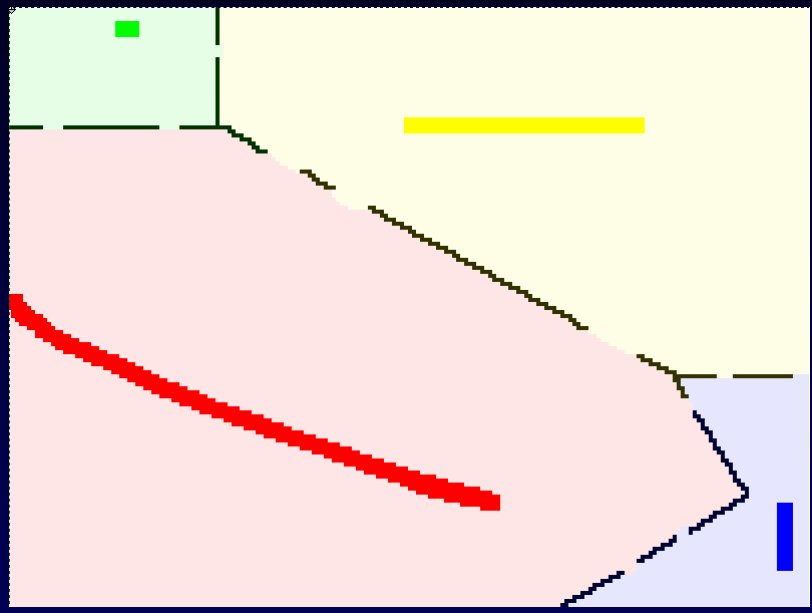


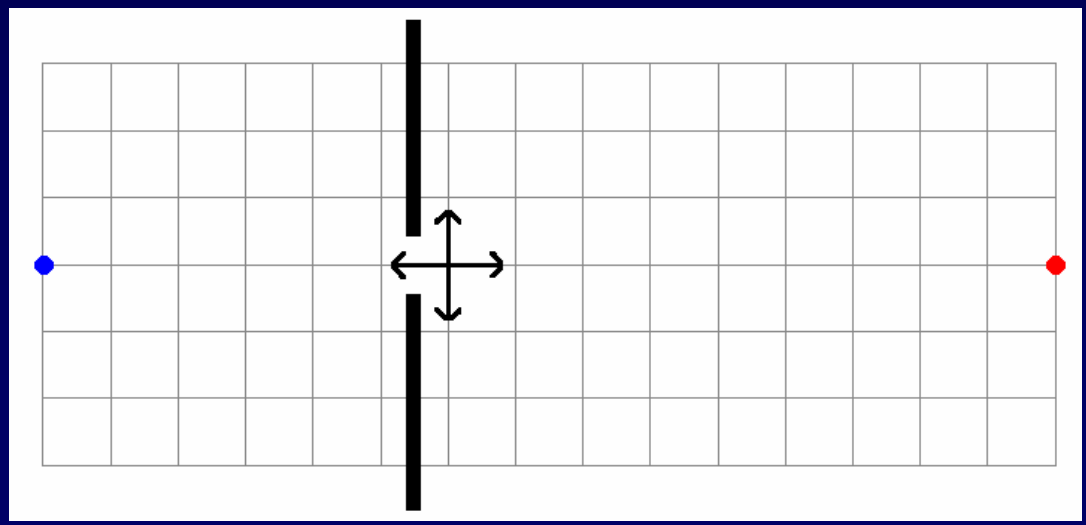
Yellow

Blue

Random Walker Segmentation

Naturally respects weak object boundaries





Random Walker Segmentation

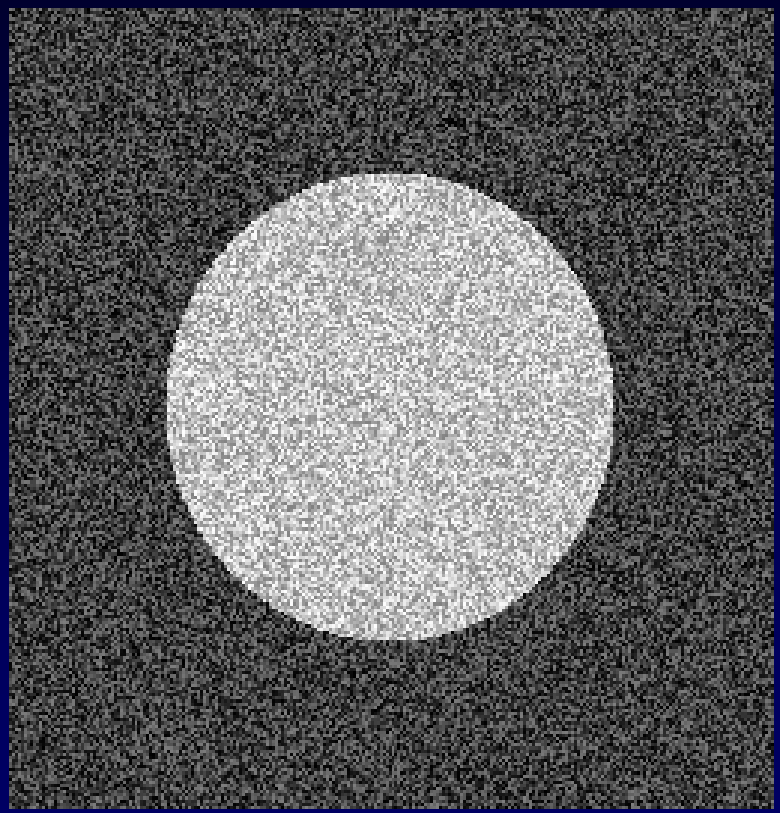
Naturally respects weak object boundaries

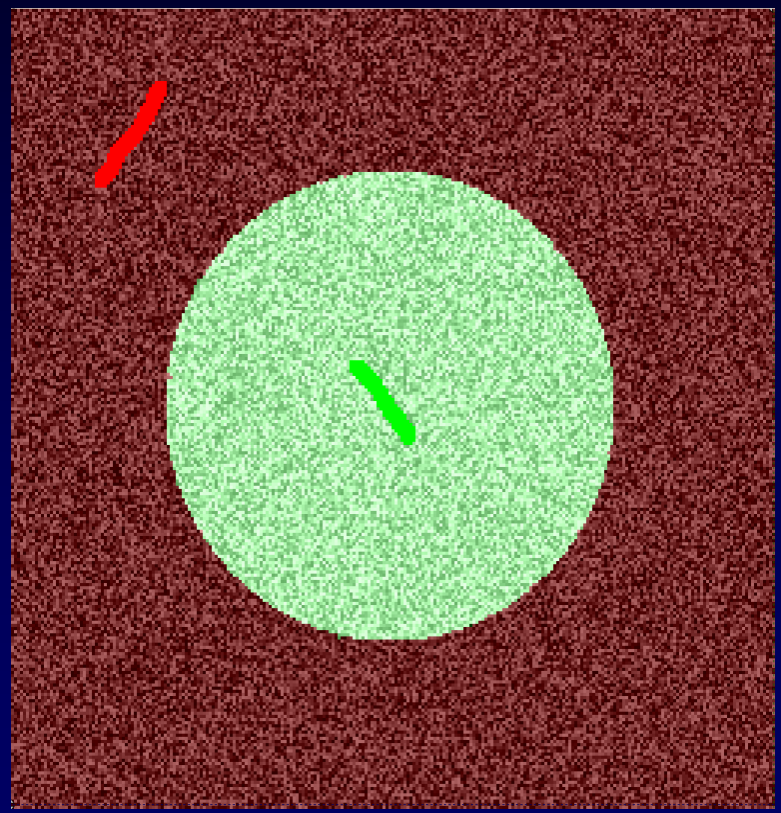
Solid border

Weak border

Random Walker Segmentation

Provably robust to identically distributed noise (need not be independent)

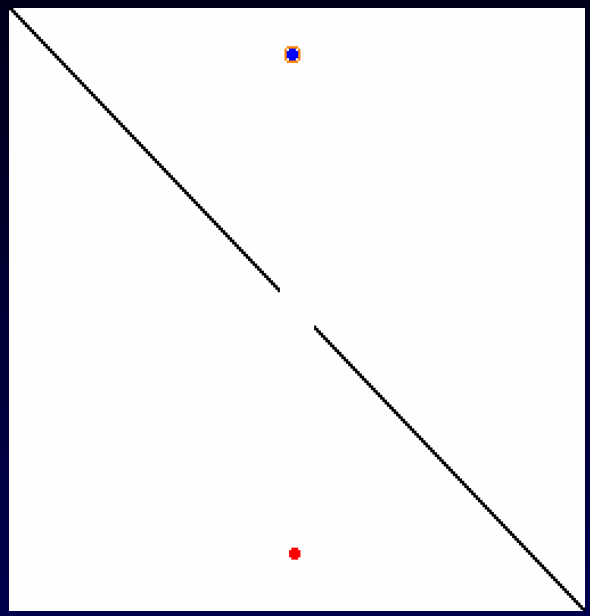


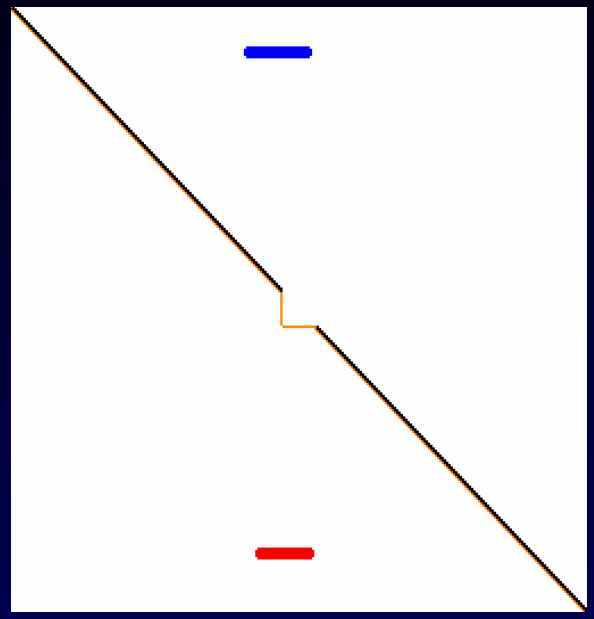


No texture or filtering used – Based purely on intensity weighting

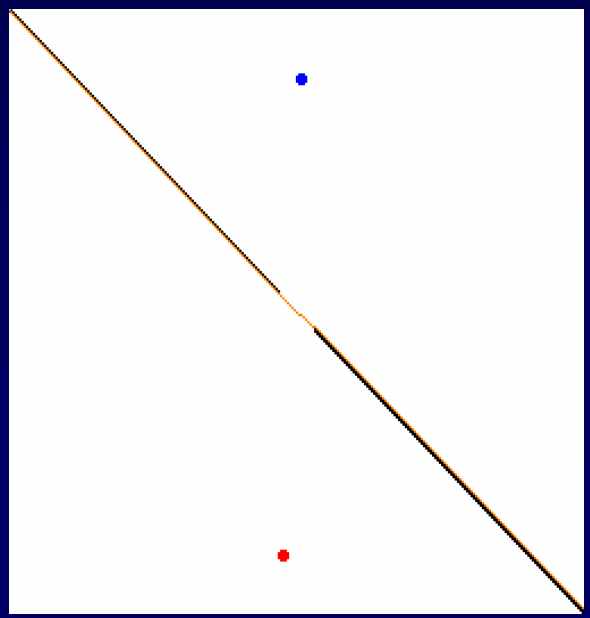
Random Walker Segmentation

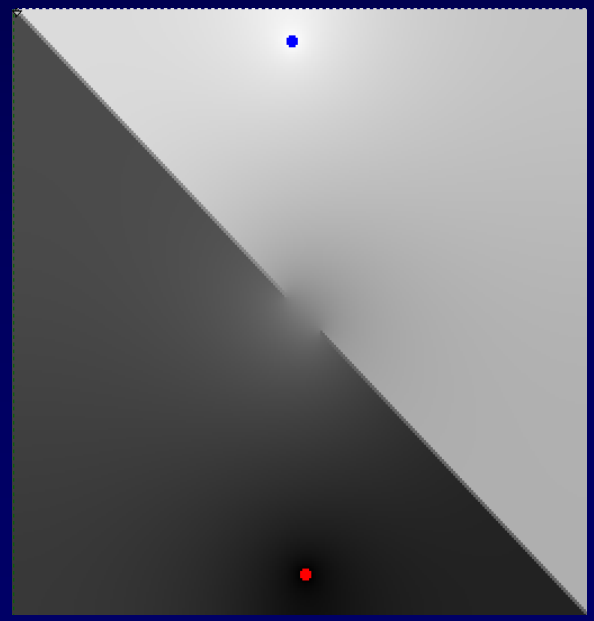
Graph cuts





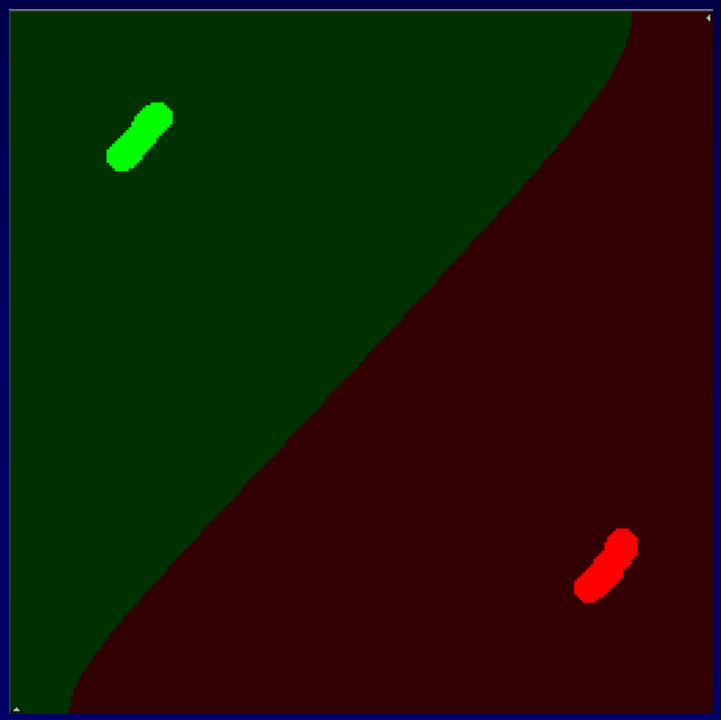
Random walker

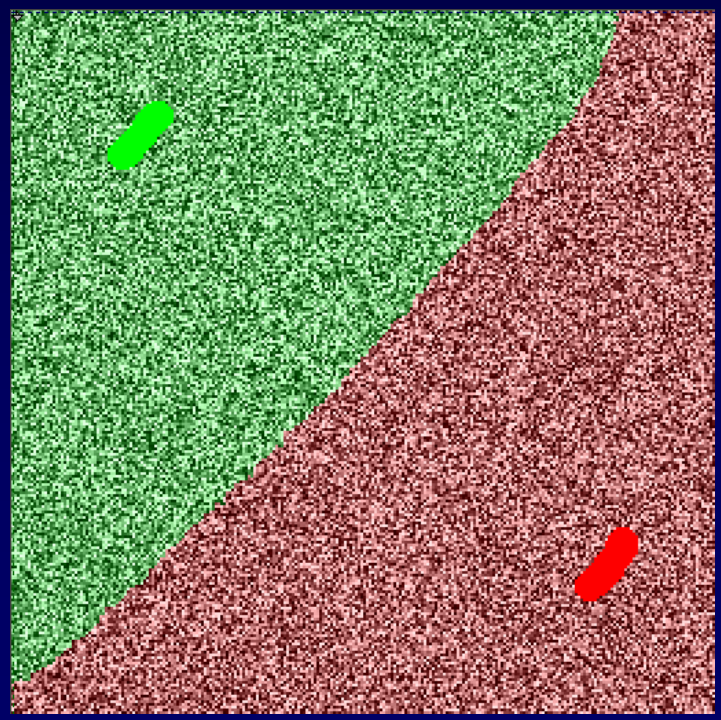




Random Walker Segmentation

1. Segmented regions are connected to a seed
2. The probabilities for a blank image (e.g., all black) yield a Voronoi-like segmentation
3. The expected segmentation for an image of pure noise (identical, not necessarily independent r.v.s) is equal to the Voronoi-like segmentation obtained from a blank image





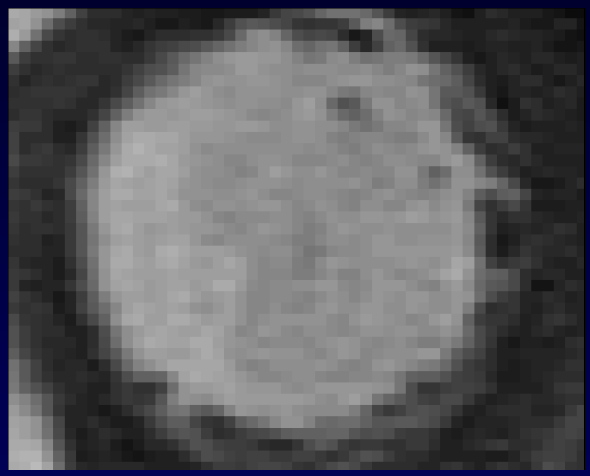
Outline

- Random walker algorithm
- Application to color images - LPP
- GPU Implementation
- Results
- Conclusion

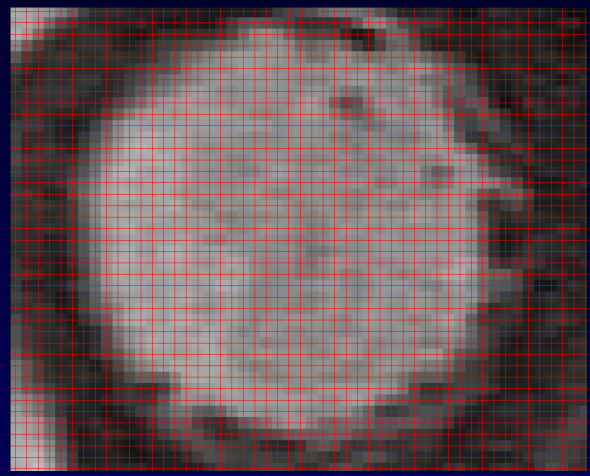
Image Weighting - Color

Random walk formulated on a lattice (graph) that represents the image

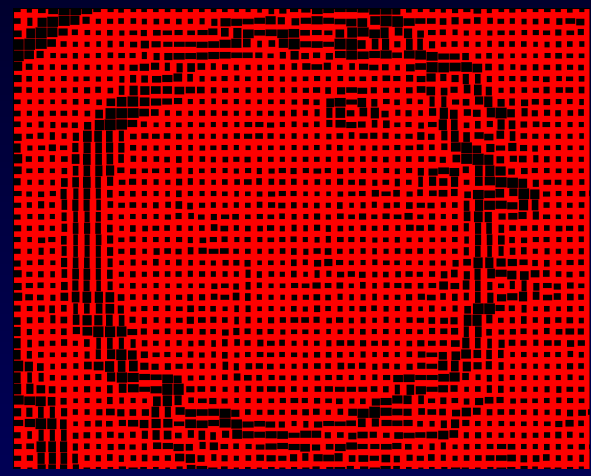
Input image



Overlaid graph (lattice)



Edge strength (line width) encodes image gradient

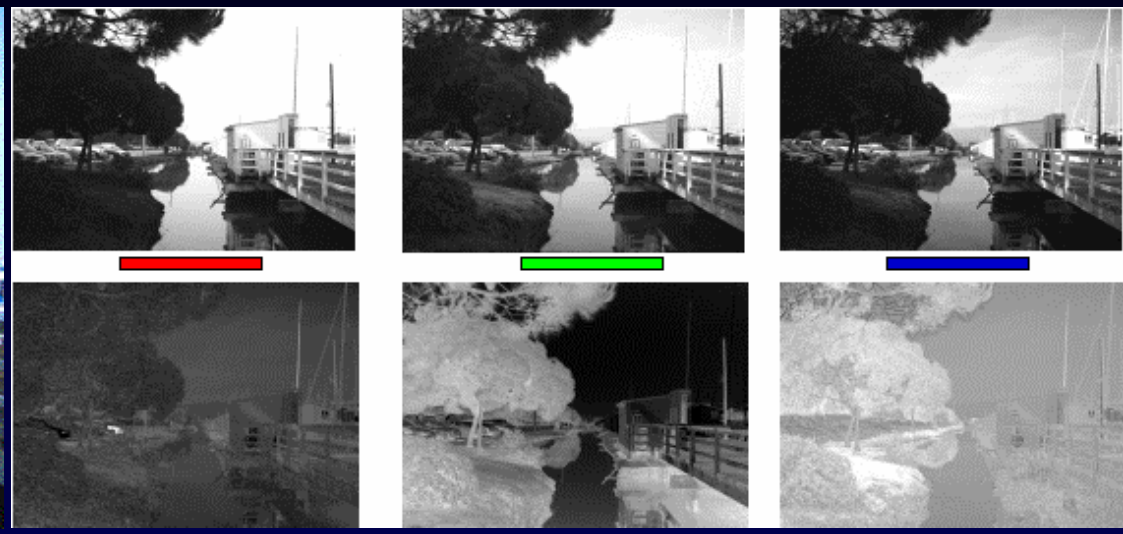


$$w_{ij} = \exp\left(\frac{-(I_i - I_j)^2}{\sigma^2}\right)$$

How to handle color images?

Image Weighting - Color

Answer: Project into LPP space and use a *conjugate* norm



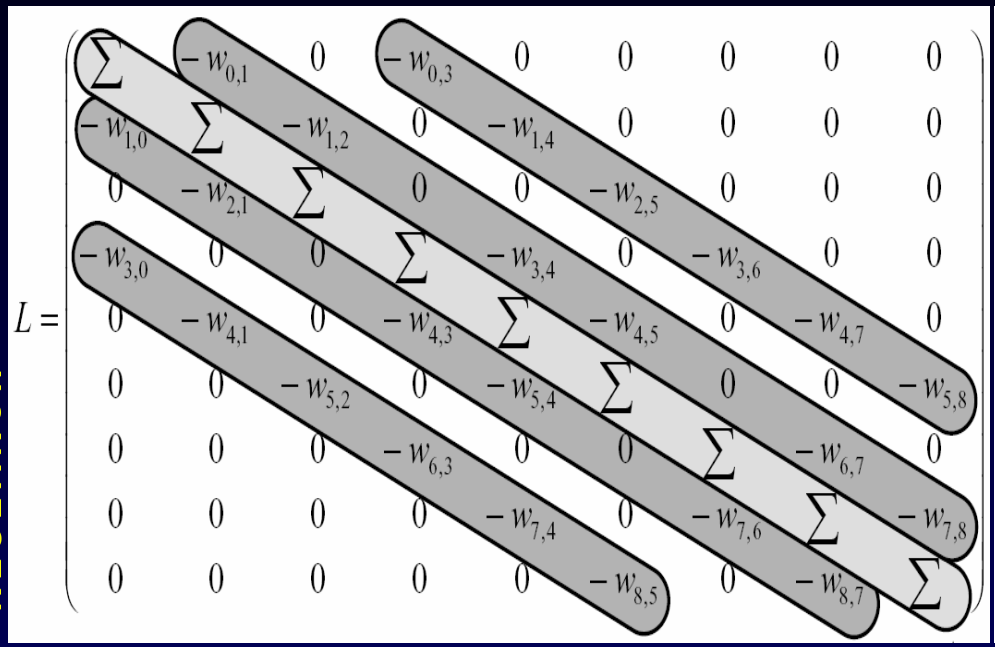
$$w_{ij} = \exp\left(\frac{(z_i - z_j)^T Q^T Q (z_i - z_j)}{\sigma^2}\right)$$

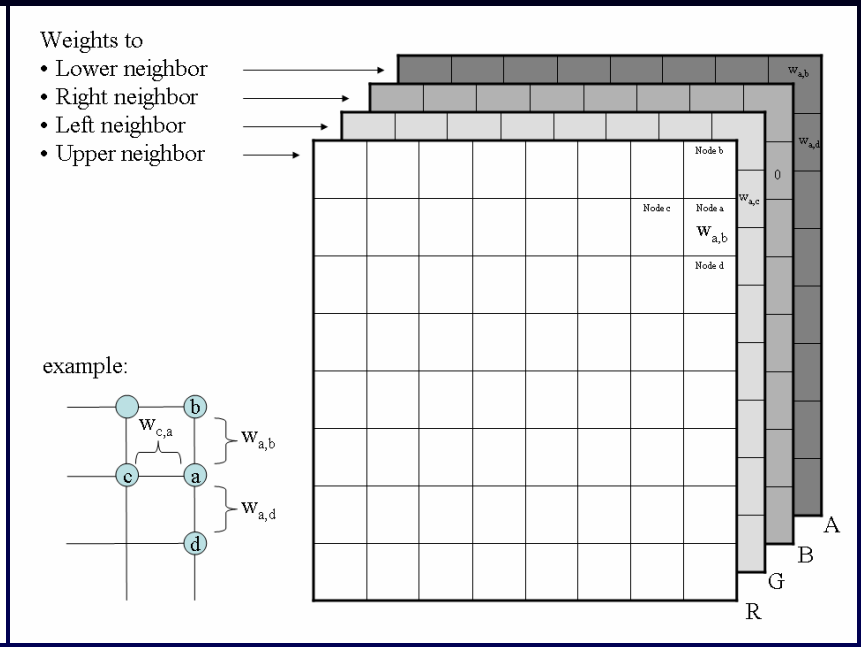
In other words: Use square of gradients *projected onto the eigenvectors (LPP space)*

Outline

- Random walker algorithm
- Application to color images - LPP
- GPU Implementation
- Results
- Conclusion

GPU Implementation





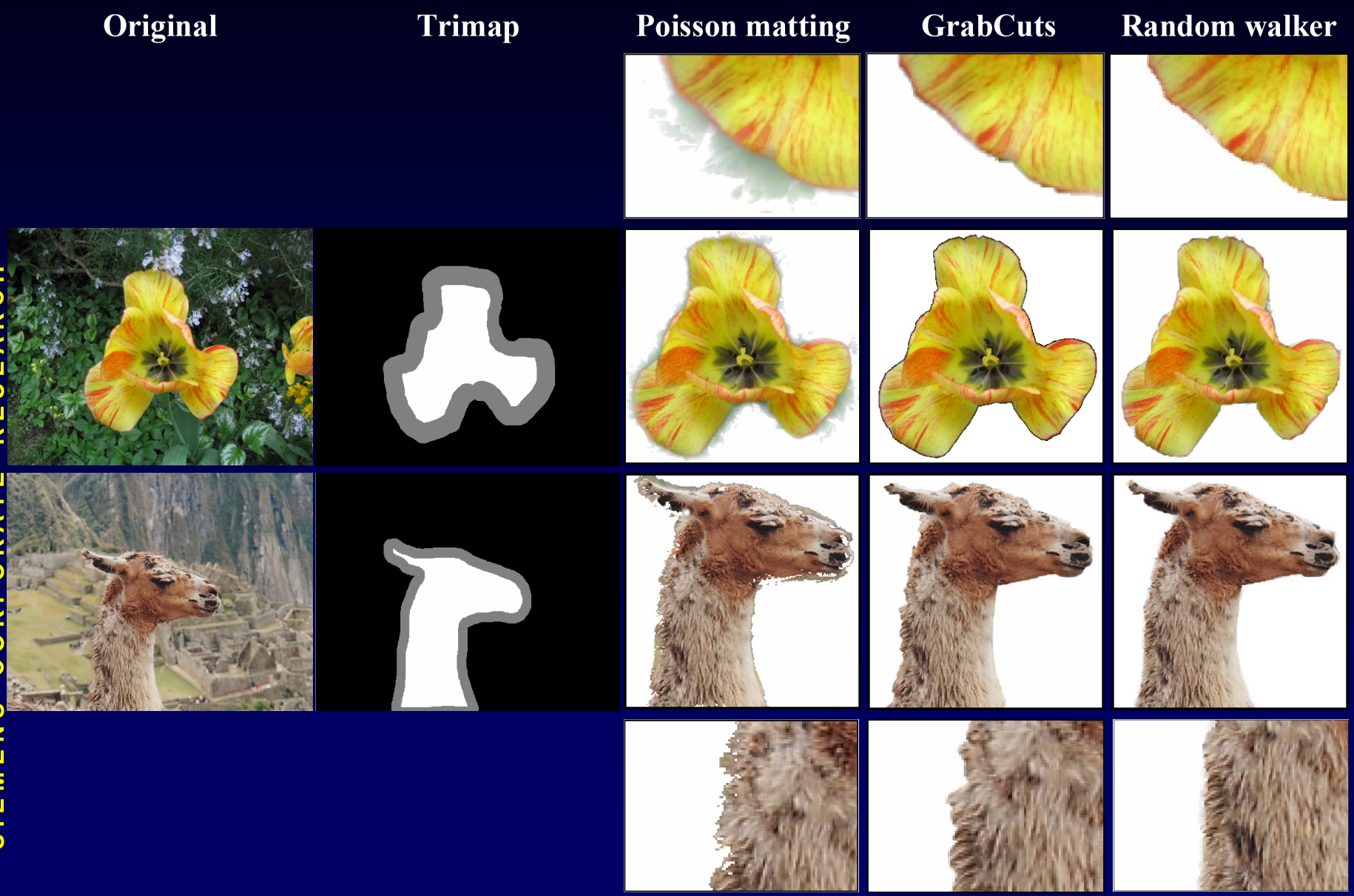
Advantages of a GPU implementation

- Structure of the Laplacian matrix allows for efficient storage and operations – Off diagonals may be packed into RGBA
- Progressive visualization of solution possible
- Z-buffer allows masking out of seeds (trimap)

Outline

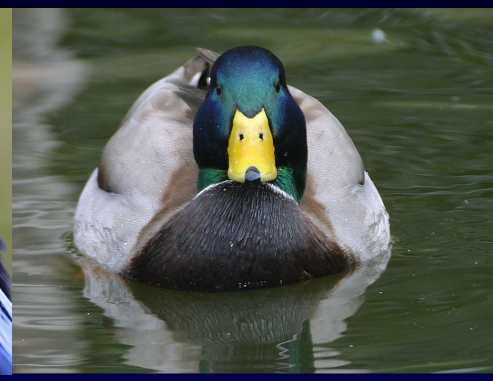
- Random walker algorithm
- Application to color images - LPP
- GPU Implementation
- Results
- Conclusion

Results

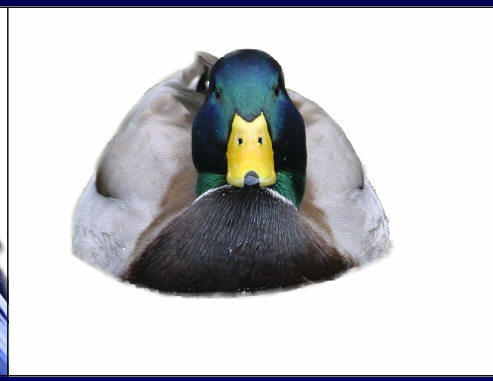


Results







Outline

- Random walker algorithm
- Application to color images - LPP
- GPU Implementation
- Results
- Conclusion

Conclusion

Advantages of random walker approach

- Segmentation/alpha-matting in a single step
- Straightforward to implement
- A single free parameter (which was not adjusted for all the results presented here)
- Fast, efficient
- Natural GPU implementation

Conclusion – More Information

Writings and code

My webpage:
http://cns.bu.edu/~lgrady

Random walkers segmentation paper:
http://cns.bu.edu/~lgrady/grady2004multilabel.pdf

GPU methods for linear systems:
http://wwwcg.in.tum.de/Research/Projects/GPUSim

MATLAB toolbox for graph theoretic image processing at:
http://eslab.bu.edu/software/graphanalysis/