Shape from Shading: Recognize the Mountains through a Global View

Qihui Zhu

Oct 26, 2005

In preparation for CVPR 2006

Penn

Outline

- Introduction
- Fast marching algorithm
- Local uncertainties
- Exploiting global constraints
- Proposed approach
- Preliminary results
- Conclusion and future work

Outline

- Introduction
- Fast marching algorithm
- Local uncertainties
- Exploiting global constraints
- Proposed approach
- Preliminary results
- Conclusion and future work

Introduction

- A very old problem(dating back to 70's)
- Problem definition
  - Shape recovery from a single image
  - Many other assumptions

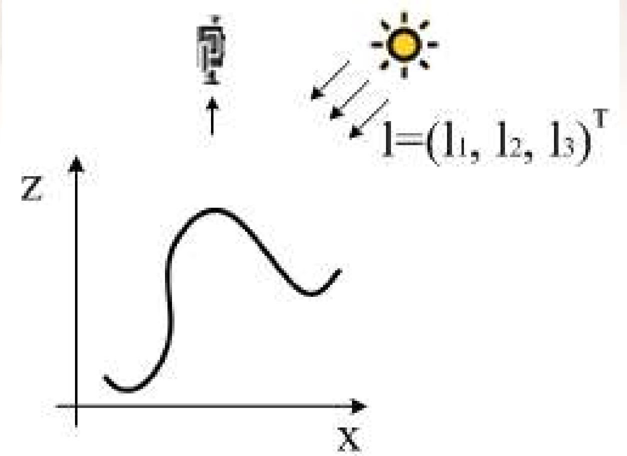
Shading image formulation

- Shading image formulation

$$I(p) = \rho\, \mathbf{l} \cdot \mathbf{n}(p)$$

$$I(p) = \frac{\rho(l_1 z_x + l_2 z_y + l_3)}{\sqrt{z_x^2 + z_y^2 + 1}}$$

- $I(p)$ intensity
- ρ albedo
- $\mathbf{l}$ light source direction
- $\mathbf{n}(p)$ surface normals



Different assumptions

- Classical assumptions
  - Lambertian
  - Point light source at infinity, known
  - Orthogonal view
  - Smooth surface
  - No shadows

- Recent concerns
  - Perspective view
  - $1/r^2$ Effect
  - With shadows

- Real image conditions(difficulties!)
  - Multiple light sources, diffuse
  - Light directions unknown
  - Albedo unknown
  - Shadows
  - Occluding contours

Penn

Previous methods

- Minimization
  - A whole family of methods...
- Propagation
  - Characteristic strip
  - Fast marching
  - Viscosity solutions for PDE
- Others
  - Spectral graph
  - Belief propagation...

Outline

- Introduction
- Fast marching algorithm
- Local uncertainties
- Exploiting global constraints
- Proposed approach
- Preliminary results
- Conclusion and future work

Fast marching algorithm

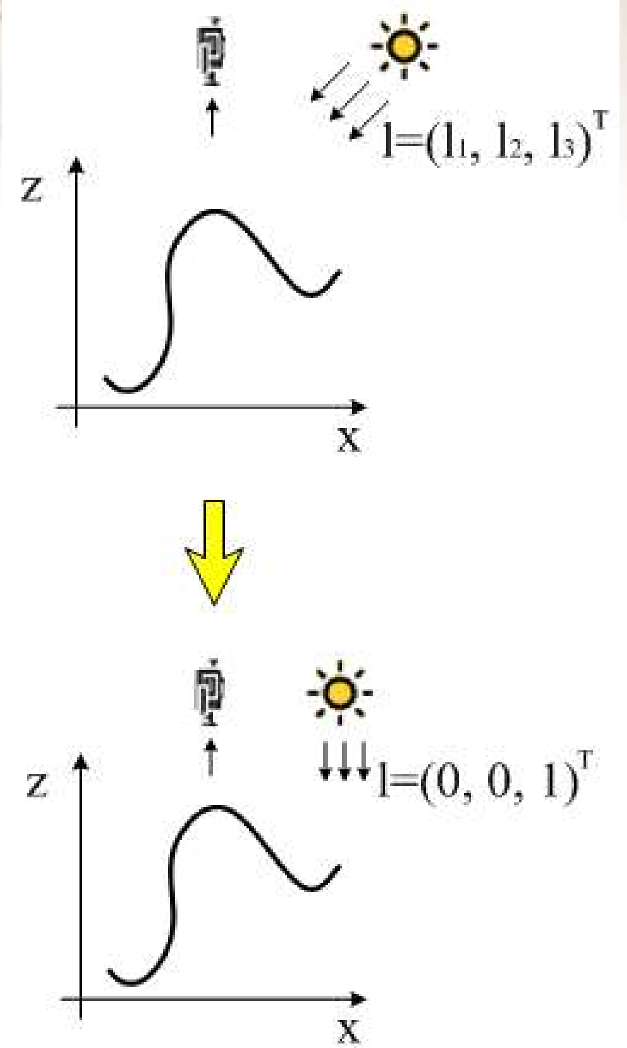
- Shading image formulation

$$I(p) = \rho\, \mathbf{l} \cdot \mathbf{n}(p)$$

$$I(p) = \frac{\rho(l_1 z_x + l_2 z_y + l_3)}{\sqrt{z_x^2 + z_y^2 + 1}}$$

- If $\mathbf{l} = (0,0,1)^T$, this reduces to

$$||\nabla z|| = \sqrt{z_x^2 + z_y^2} = \sqrt{\frac{1}{I^2} - 1}$$



Fast marching algorithm

- How to solve this Partial Derivative Equation (PDE)?

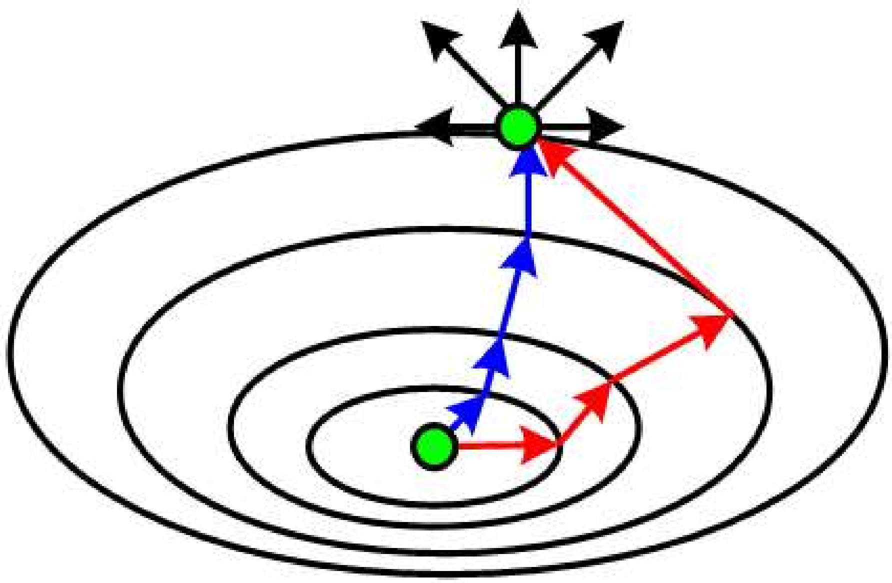
$$||\nabla z|| = \sqrt{z_x^2 + z_y^2} = \sqrt{\frac{1}{I^2} - 1}$$

- Propagate from a singular point
- This is equivalent to computing the shortest path from the singular point, with weight $\sqrt{\frac{1}{I^2} - 1}$ on every node.

The shortest path

$$||\nabla z|| = \sqrt{z_x^2 + z_y^2} = \sqrt{\frac{1}{I^2} - 1}$$

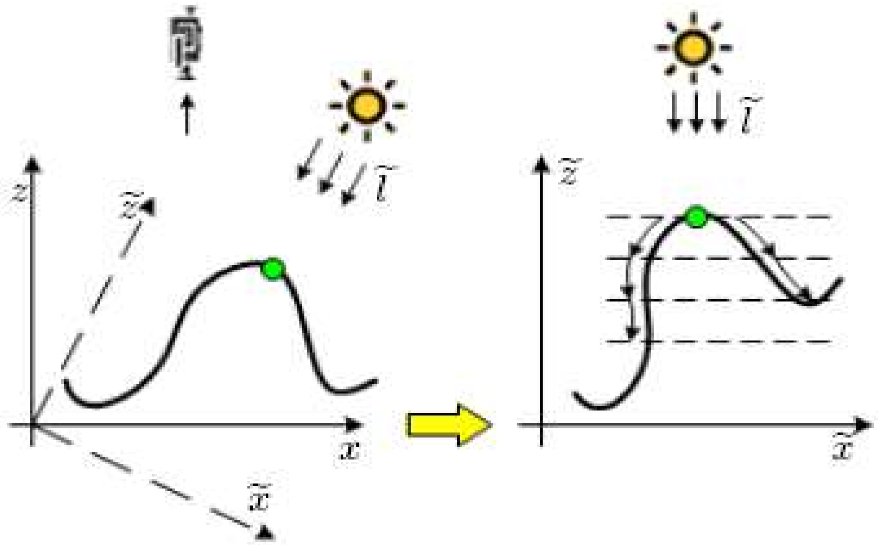
$$z(r) - z(r + \Delta s) \leq \Delta s ||\nabla z(r)||$$



Fast marching algorithm

- What if the light source is not vertial?
  - Assume $l_2 = 0$
  - Propagate in the new coordinate system

$$\tilde{l} = (0,0,1)^T, \tilde{p} = (\tilde{x}, \tilde{y}) = (-l_3\tilde{x} + l_1\tilde{z},\ y), \tilde{z} = l_1 x + l_3 z$$

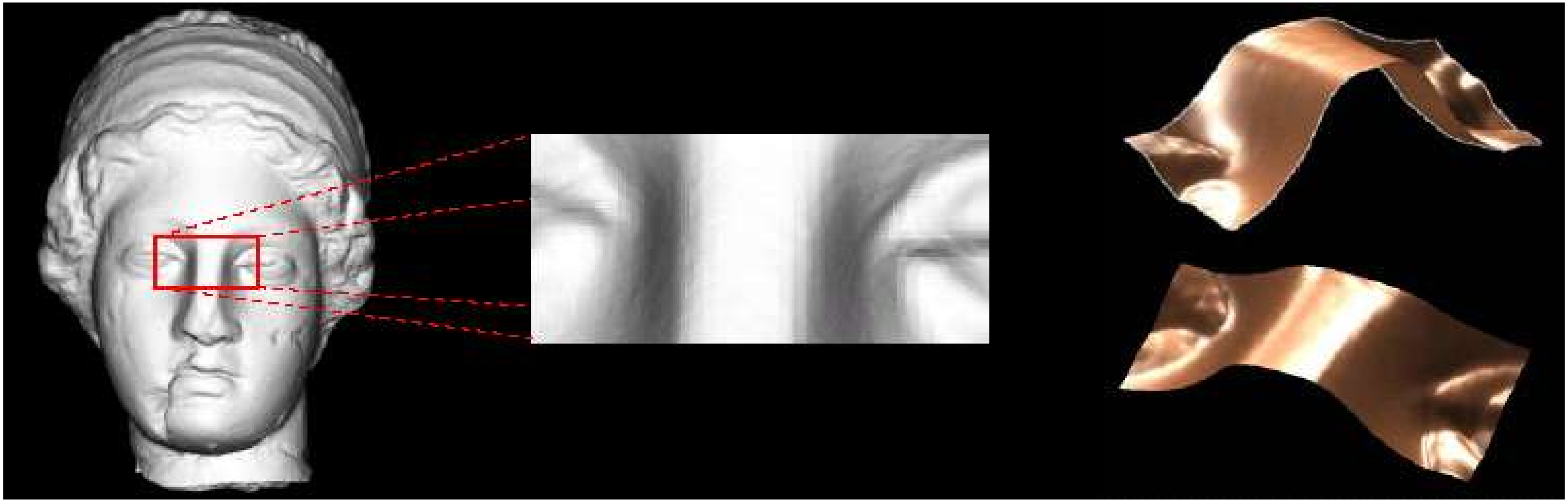


Outline

- Introduction
- Fast marching algorithm
- Local uncertainties
- Exploiting global constraints
- Proposed approach
- Preliminary results
- Conclusion and future work

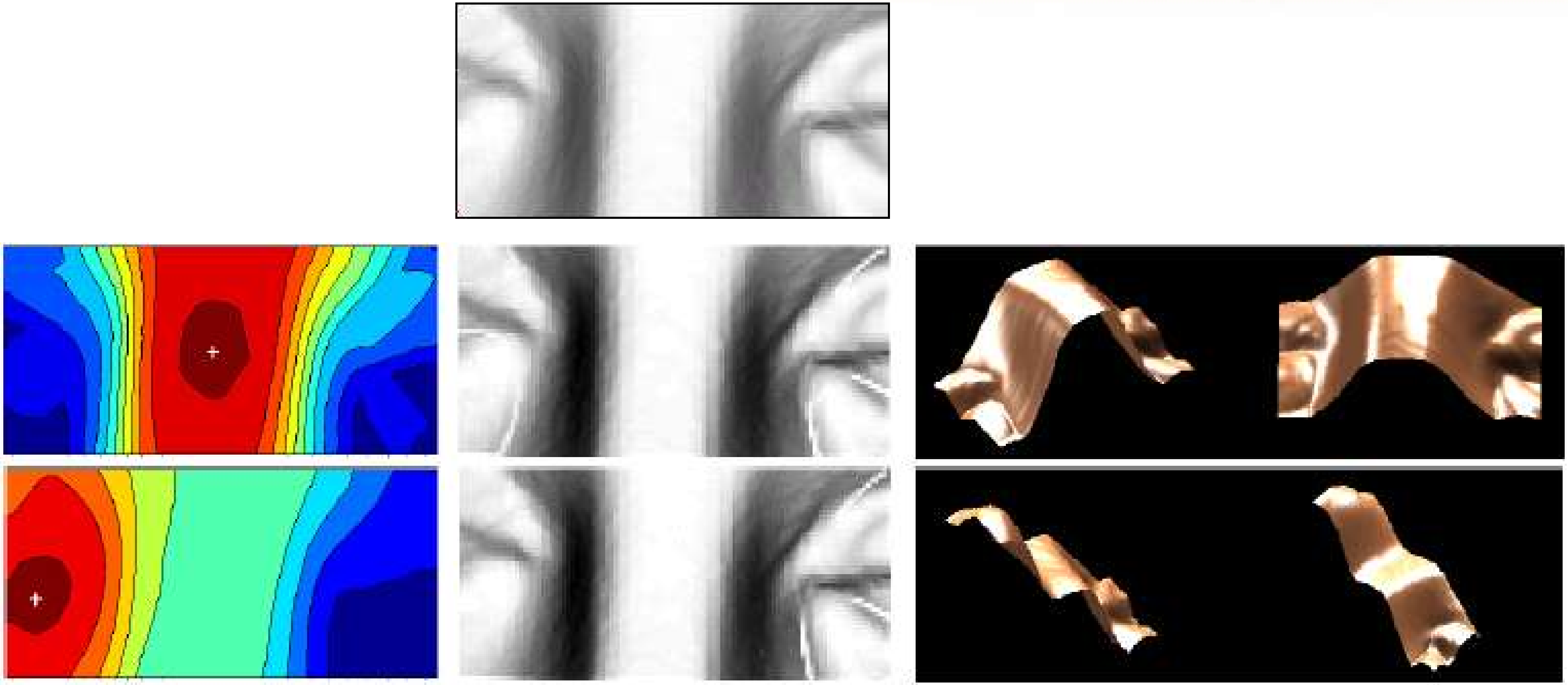
Local uncertainties

- Fast marching is good, but not solving everything
- Venus' nose



Local uncertainties

- Different results...

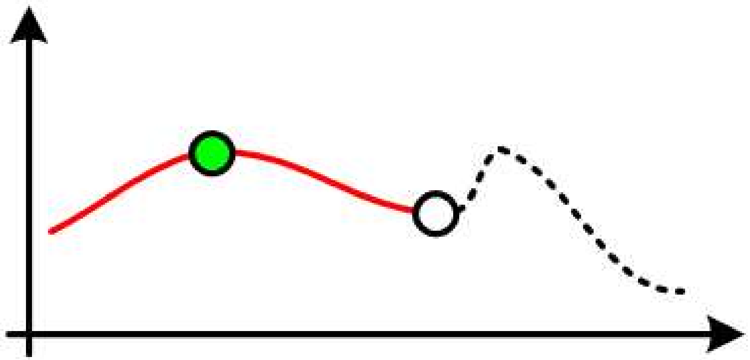


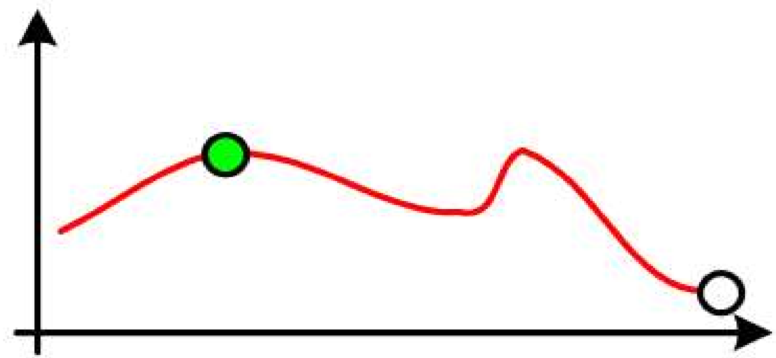
Local uncertainties

- What remains unknown after shortest path?
  - How far you can travel?
  - Are you going up or down?
  - Convex or concave?
- Common problems for propagation methods, not just for fast marching
- Let's see some simple cases...

Local uncertainties

- How far you can travel?

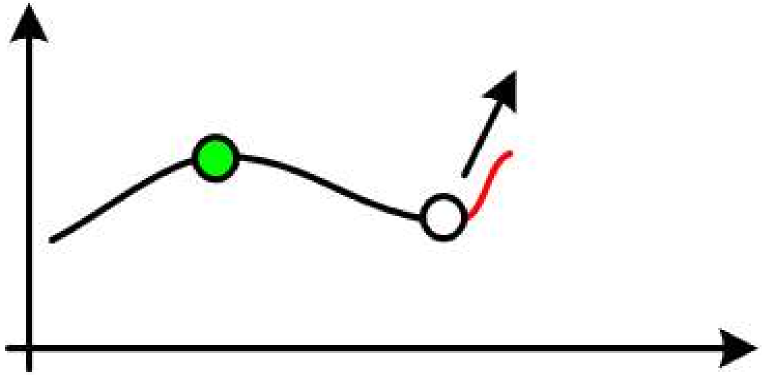


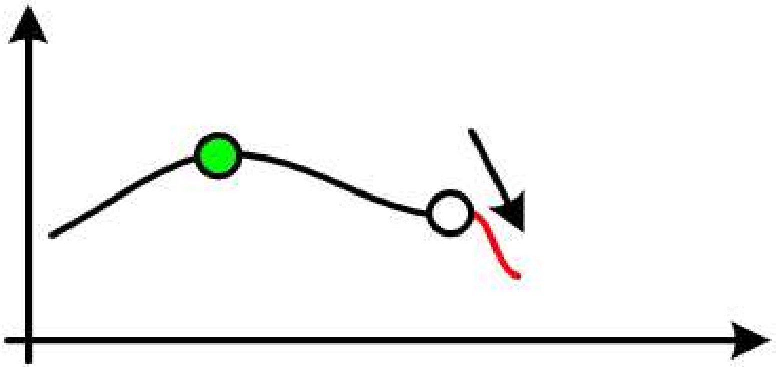


- Left or right?

Local uncertainties

- Are you going up or down?

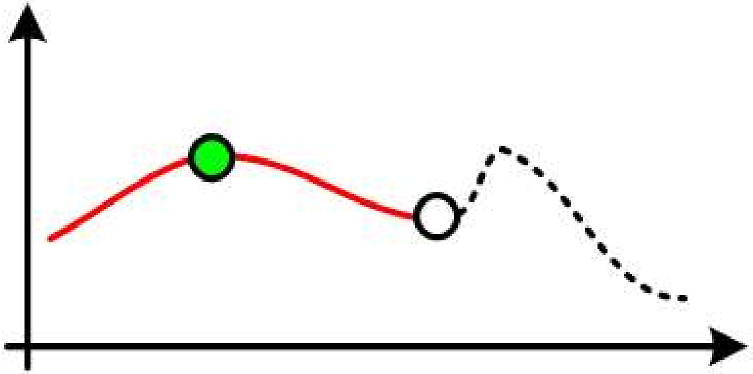


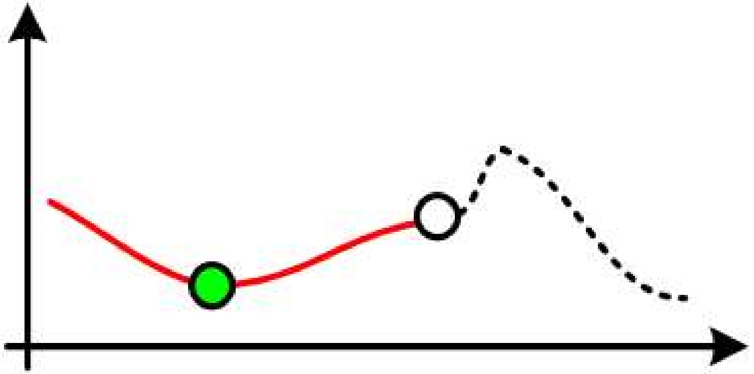


- Left or right?

Local uncertainties

- Convex or concave?





- Left or right?

Local uncertainties

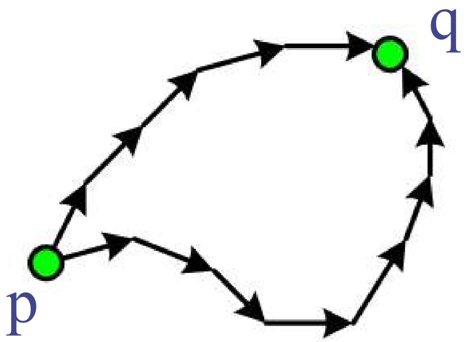
- What remains unknown after shortest path?
  - How far you can travel?
  - Are you going up or down?
  - Convex or concave?
- Unsolvable locally!

Outline

- Introduction
- Fast marching algorithm
- Local uncertainties
- Exploiting global constraints
- Proposed approach
- Preliminary results
- Conclusion and future work

Exploiting global constraints

- Global integrability constraints
  - Continuous surface, no sudden 'jumps'
  - Local estimation of height differences
  - Integration along a loop must be 0, or different paths should have the same height difference.



- Smoothness constraints
  - Use propagation to generate local patches
  - Boundaries between patches must be smooth

Outline

- Introduction
- Fast marching algorithm
- Local uncertainties
- Exploiting global constraints
- Proposed approach
- Preliminary results
- Conclusion and future work

Proposed approach

- Configuration graph G=(V, E, W)
  - V Singular points $n = |V|$
  - E Edges connecting neighboring vertices $m = |E|$
  - W Height difference estimation by fast marching

$$W = \text{diag}(w_1, w_2, ..., w_m)$$

- Representing configurations
  - d +1/-1 defined on edges

$$d = (d_1, d_2, ..., d_m)^T \text{ with } d_i = \pm 1 (i = 1, 2, ..., m)$$

Configuration graph

- How do d solve the local uncertainties?
  - Are you going up or down?
    - easy, simply +1 for up, -1 for down
  - Convex or concave?
    - peaks: all edges going out +1, convex
    - valleys: all edges going out -1, concave
  - How far you can travel?
    - only start from peaks
    - always go down as far as you can

Constraints on the graph

- A little more definition
  - A Adjacency matrix $A \in \mathbb{R}^{m \times n}$

$$A_{ij} = \begin{cases} +1 & e_i = (v_j, v_k) \text{ for some } k \\ -1 & e_i = (v_k, v_j) \text{ for some } k \\ 0 & \text{otherwise} \end{cases}$$

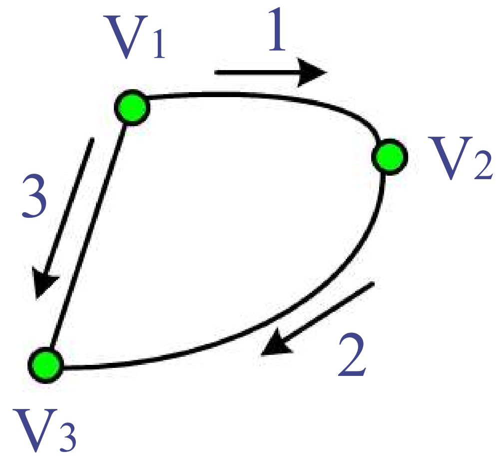
  - H Heights at vertices $h = (h_1, h_2, ..., h_n)^T$

Constraints on the graph

- Height difference constraints

$$Ah = Wd$$

- What are the constraints doing?
  - Check triangles & loops
  - Assume edge monotonous
  - V_2 should not be a peak or a valley
  - Why?
  - Global integrability constraints!
  - Check for every loop



Optimal configuration

- Optimize $||Ah - Wd||_2$

$$d_{opt} = \arg\min_{d,\, h} ||Ah - Wd||_2$$

- For a fixed d $h = A^+ W d$ $\quad A^+ = [A^T A]^{-1} A^T$
- Finally

$$\begin{aligned} d_{opt} &= \arg\min_{d,h} ||Ah - Wd||_2 \\ &= \arg\min_{d} d^T E d \end{aligned}$$

$$E = W^T (AA^+ - I)^T (AA^+ - I) W$$

Penn

Max-cut problem

- Optimizing $d^T E d$ is simply a Max-cut!

$$d^T E d = \sum_{d_i d_j = 1} E_{ij} d_i d_j + \sum_{d_i d_j = -1} E_{ij} d_i d_j = 2 \sum E_{ij} - \sum_{d_i d_j = -1} E_{ij}$$

$$\arg\min_d d^T E d = \arg\max_d \sum_{d_i d_j = -1} E_{ij}$$

- Min-cut, N-cut is polynomial 
- Max-cut is NP-hard 
- But the graph is small...

Numerical approach

- Semi-Definite Programming(SDP)

$$\begin{aligned} \text{minimize} \quad & tr(CX) \\ \text{subject to} \quad & tr(A_i X) = b_i, \quad i = 1, 2, ..., p \\ & X \in S_+^n \end{aligned}$$

- Our problem

$$X = dd^T$$

$$\begin{aligned} \text{minimize} \quad & d^T E d = tr(EX) \\ \text{subject to} \quad & X_{ii} = tr(A_i X) = 1, \quad i = 1, 2, ..., m \\ & X \in S_+^m, A_i = e_i e_i^T \end{aligned}$$

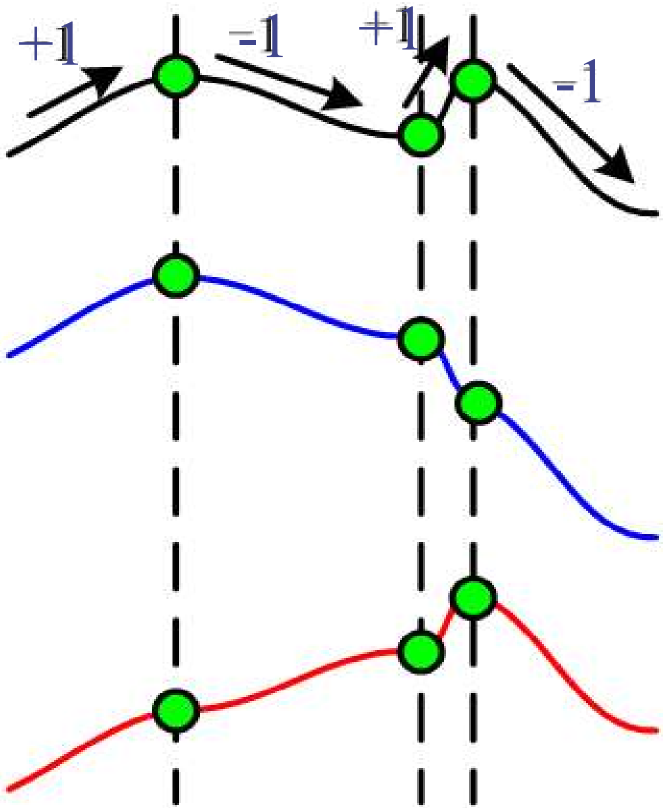
Shape recovery

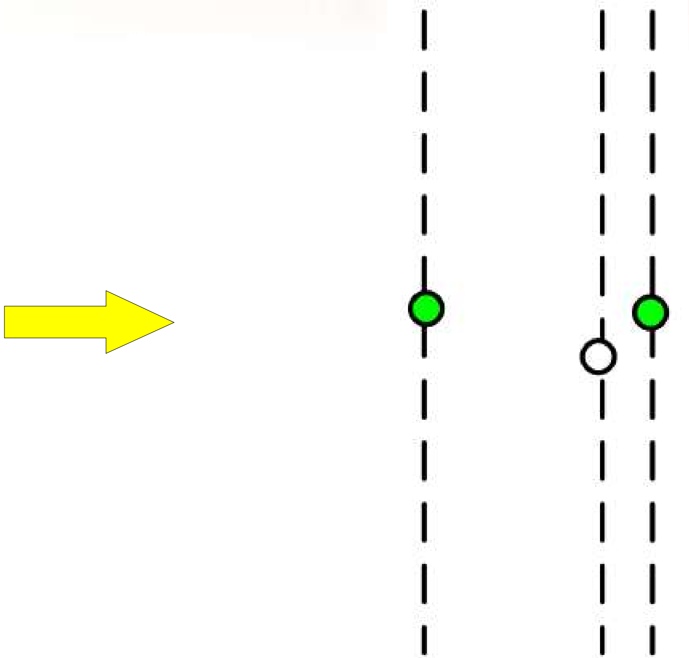
- We know d_{opt} then $h = A^{+} W d_{opt}$
- Also know which vertices are peaks $\mathcal{P}$
- Build patches around peaks
  - Fast marching
- Stitch the patches together

$$z(q) = \max_{p \in \mathcal{P}} \{ z(p) - D(p,q) \}$$

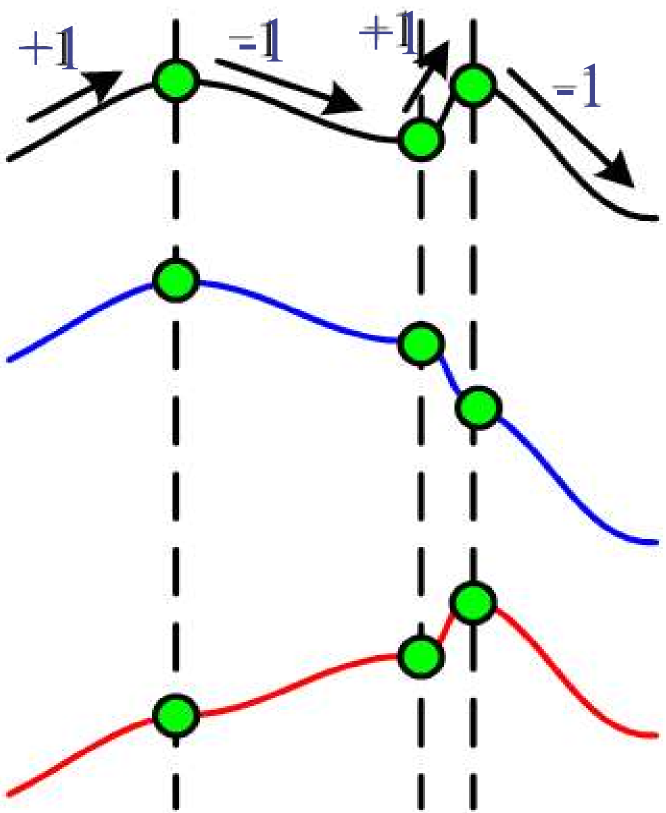
- Why does this work?

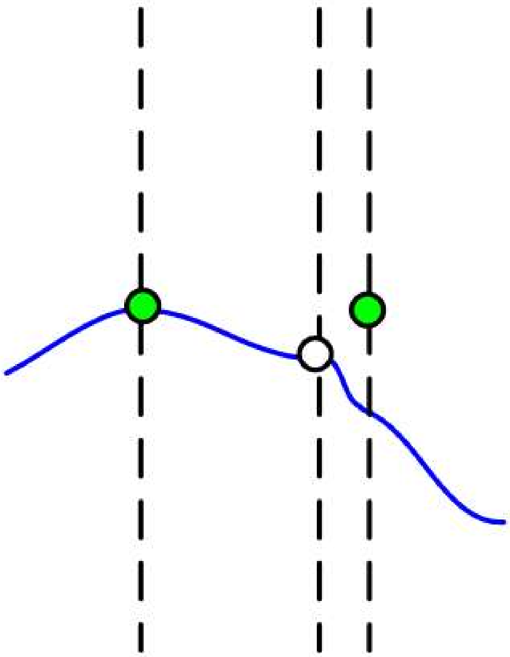
Shape recovery



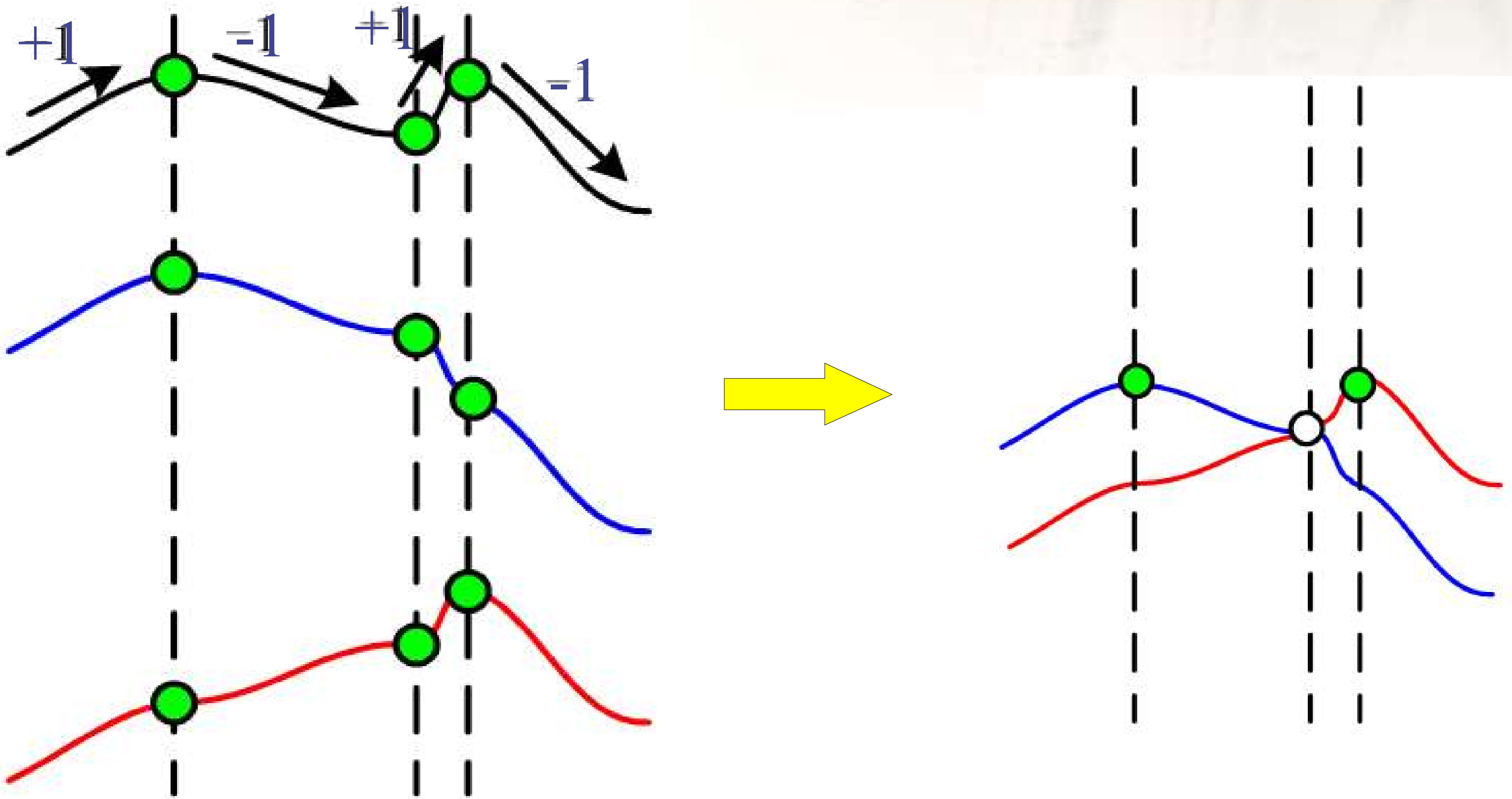


Shape recovery

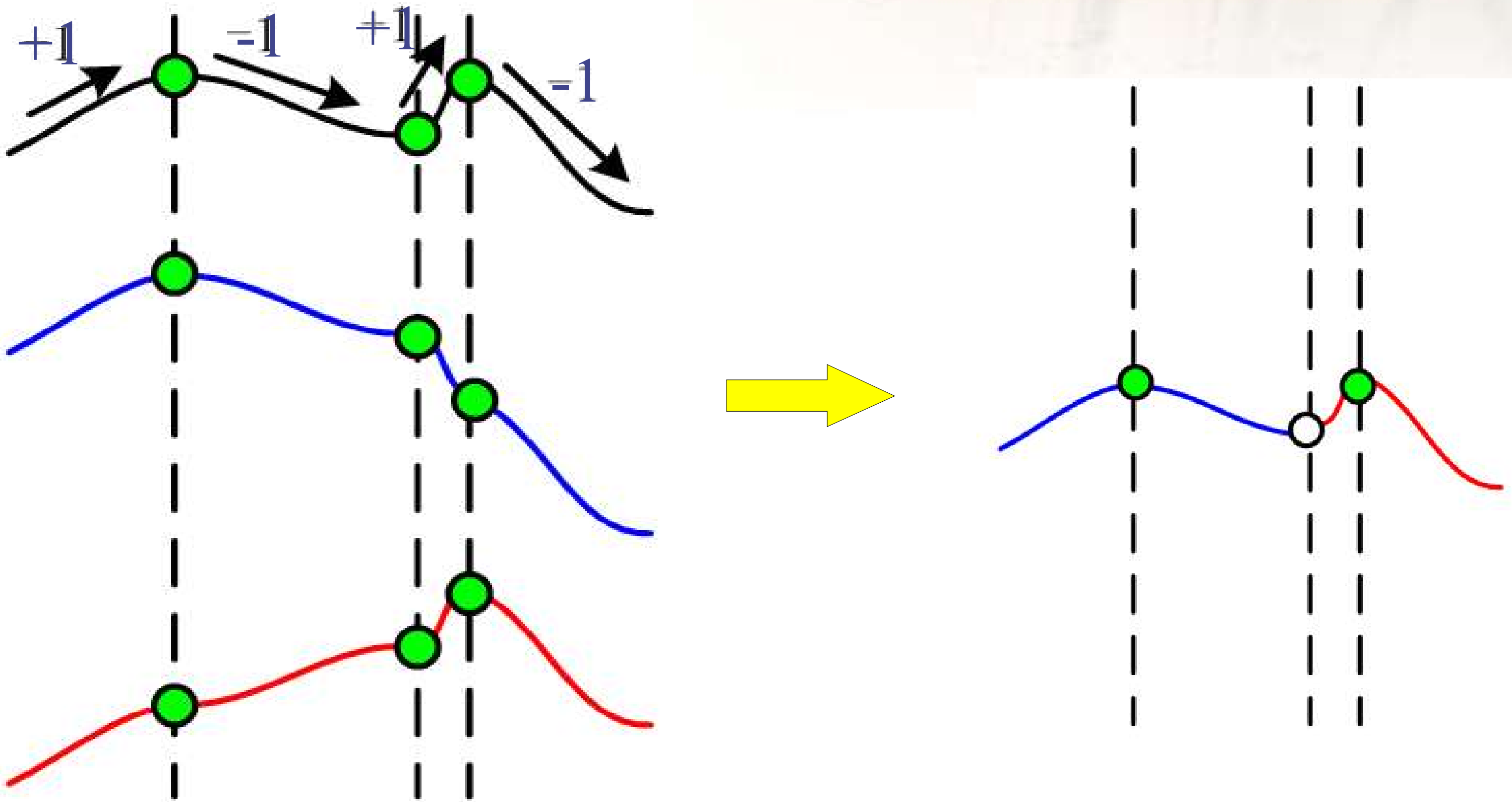




Shape recovery



Shape recovery



Algorithm overview

- Singular point detection
- Fast marching
- Graph formulation
  - Delaunay triangulation
  - Remove invalid edges
- Optimize $d^T E d$ by SDP
- Postprocessing
  - Identify peaks
- Shape recovery
  - Combine patches

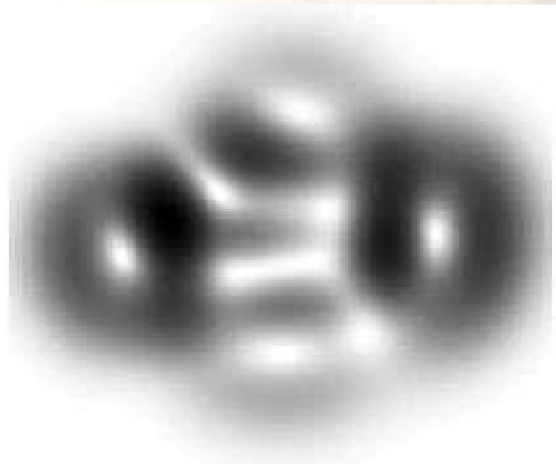
Penn

Outline

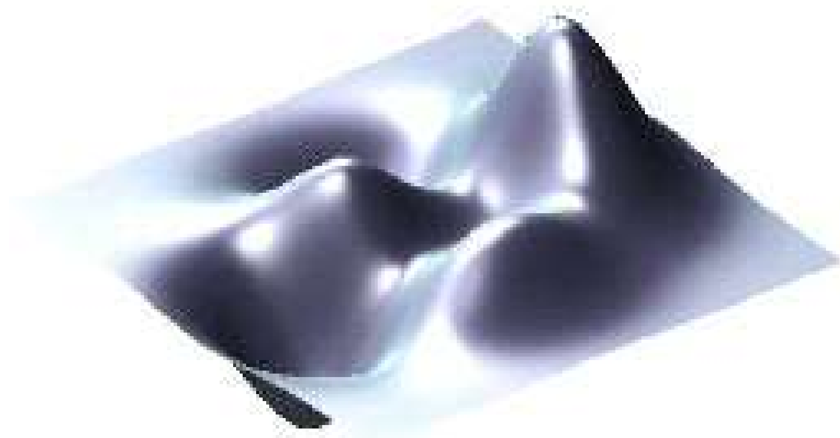
- Introduction
- Fast marching algorithm
- Local uncertainties
- Exploiting global constraints
- Proposed approach
- Preliminary results
- Conclusion and future work

Preliminary results

Matlab PEAKS



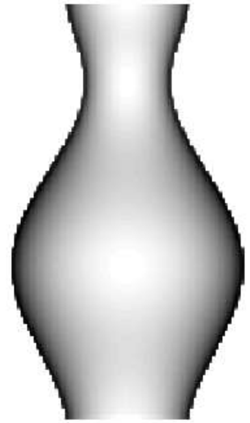
Ground truth



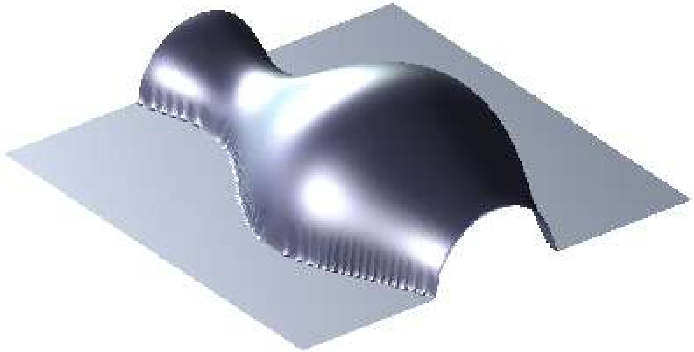
Reconstruction

Preliminary results

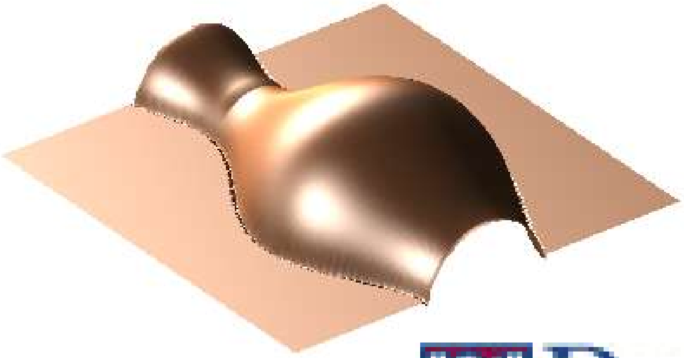
Vase



Ground truth



Reconstruction



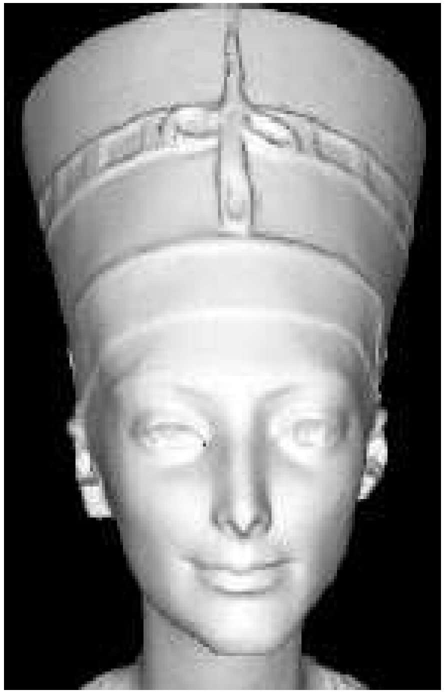
Preliminary results

Venus

Reconstructions

Preliminary results

Ancient woman



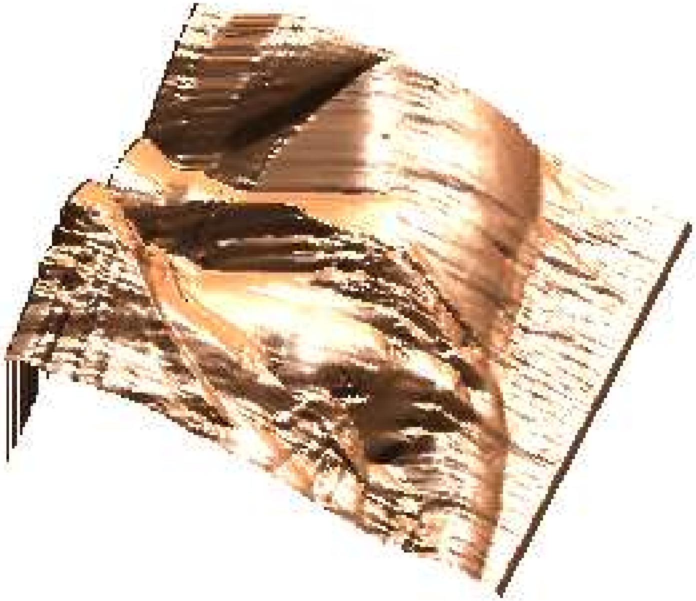
Reconstructions

Penn

Preliminary results

Relief of Athena

Reconstruction



Outline

- Introduction
- Fast marching algorithm
- Local uncertainties
- Exploiting global constraints
- Proposed approach
- Preliminary results
- Conclusion and future work

Conclusion

- Global constraints are important and powerful
- Pros & Cons of our approach
  - \+ Address ambiguities directly
  - \+ Make decisions on structures, not pixels
  - \+ Also solve the self-shadow problem
  - \+ Simple and fast
  - \- Smoothness not in the framework
  - \- Mixing little peaks with global, big peaks
  - \- Relying on singular points

Future work

- Work on real images
- Consider multiple light sources
- Combine with shadows and occluding contours
- Combine with object models
- ...

Shape from Shading

Comments...