CEE598 - Visual Sensing for Civil Infrastructure Eng. & Mgmt.

Session 13 – Stereo Systems

Mani Golparvar-Fard
Department of Civil and Environmental Engineering
3129D, Newmark Civil Engineering Lab
e-mail: mgolpar@illinois.edu

Outline

- Stereo Systems

- Stereo vision
- Rectification
- Correspondence problem
- Active stereo vision systems

- Reading: [HZ] Chapter: 11
  [FP] Chapter: 11

Some slides of this lectures are courtesy of prof. S. Savarese, and various other lecturers

Large-Scale Stereo Matching

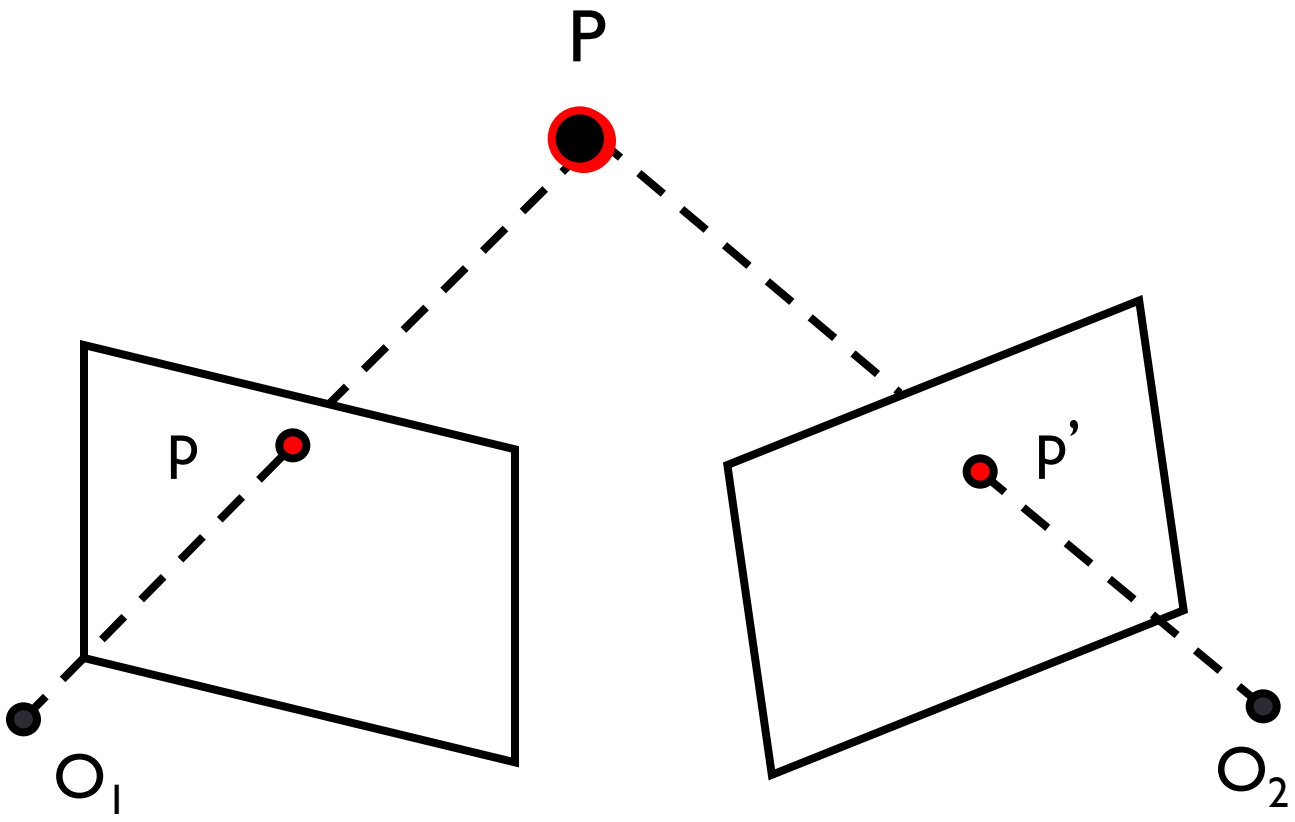
- http://www.youtube.com/watch?v=3rWycBEHn3s
- https://sites.google.com/site/yoichimorales/projects/yamazumiproject

- http://www.youtube.com/watch?v=2squ9HDuBeI&feature=related

- http://www.youtube.com/watch?v=pFsV8UY4sT4

- http://www.youtube.com/watch?v=XiTZVi2A7Cc&feature=related

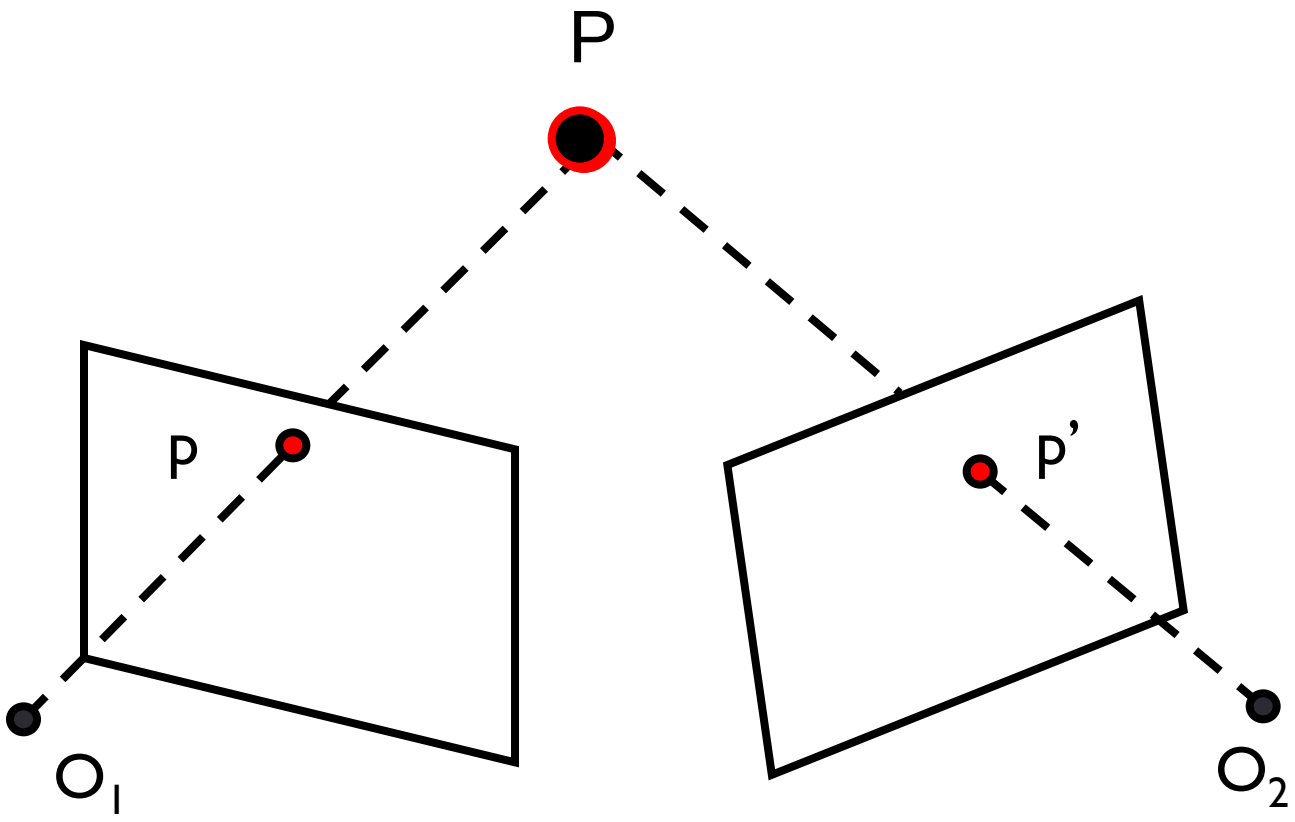
Stereo vision



Goal: estimate the position of P given the observation of P from two view points

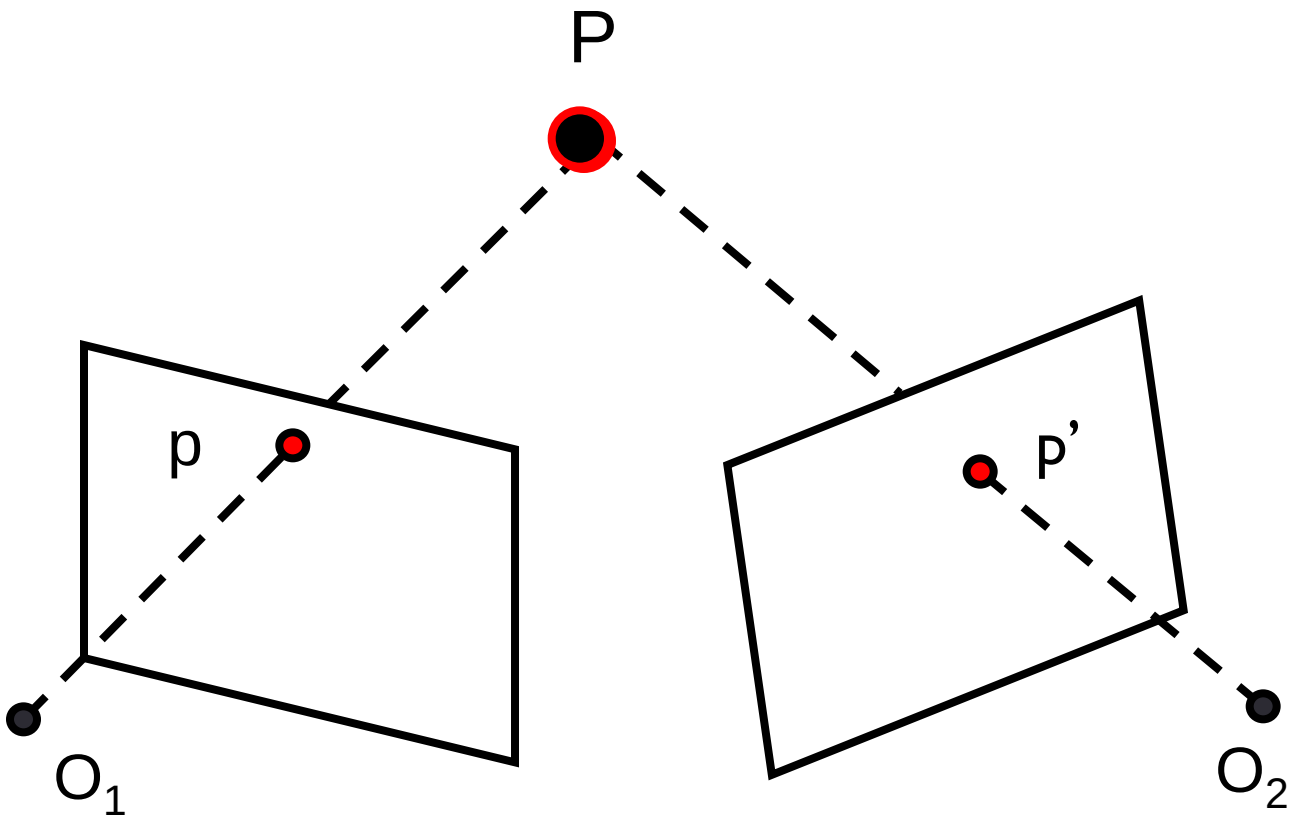
Assumptions: known camera parameters and position (K, R, T)

Stereo vision



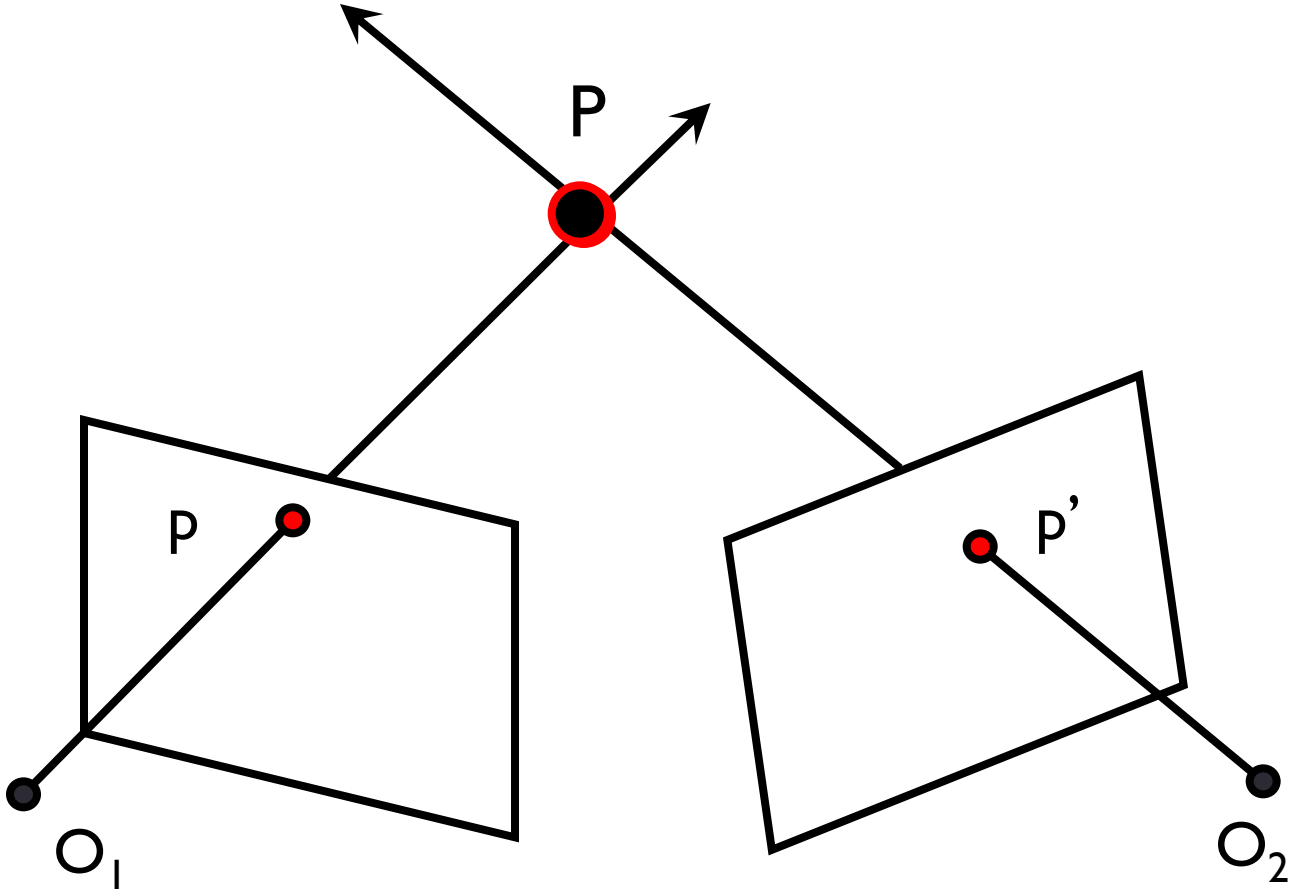
Subgoals:
- Solve the correspondence problem
- Use corresponding observations to triangulate

Correspondence problem



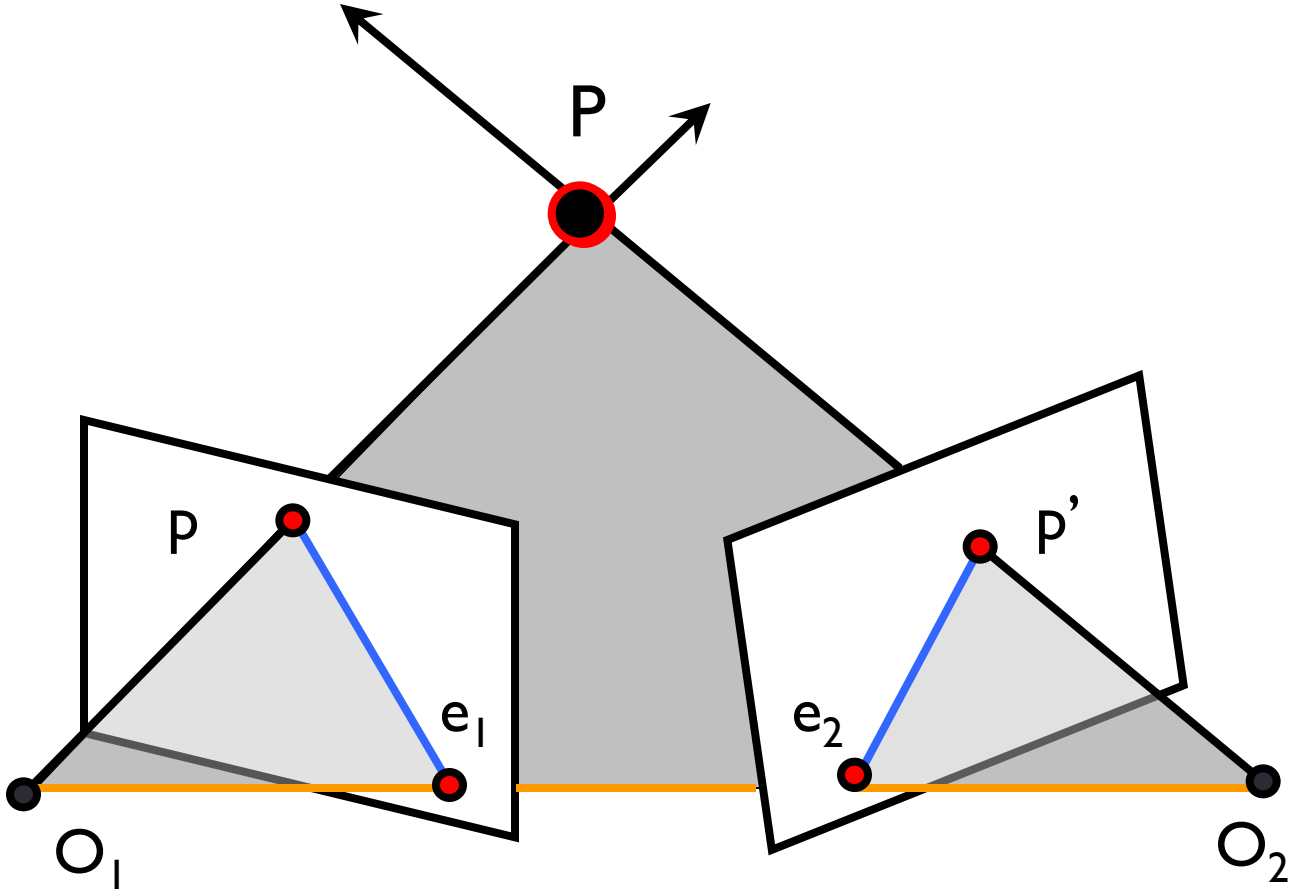
Given a point in 3d, discover corresponding observations in left and right images [also called binocular fusion problem]

Triangulation



Intersecting the two lines of sight gives rise to P

Epipolar geometry

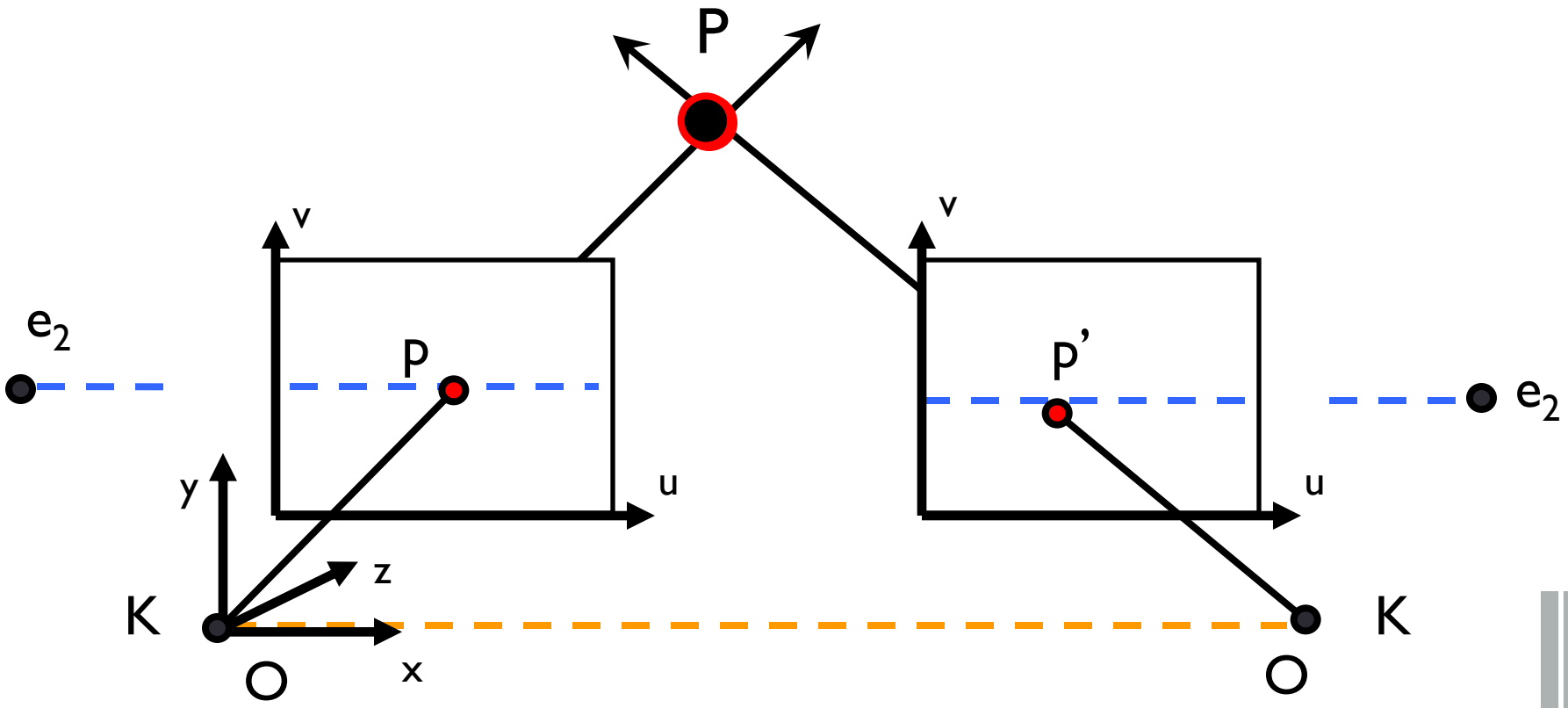


- Epipolar Plane
- Baseline
- Epipolar Lines

- Epipoles e_1, e_2

  = intersections of baseline with image planes
  = projections of the other camera center
  = vanishing points of camera motion direction

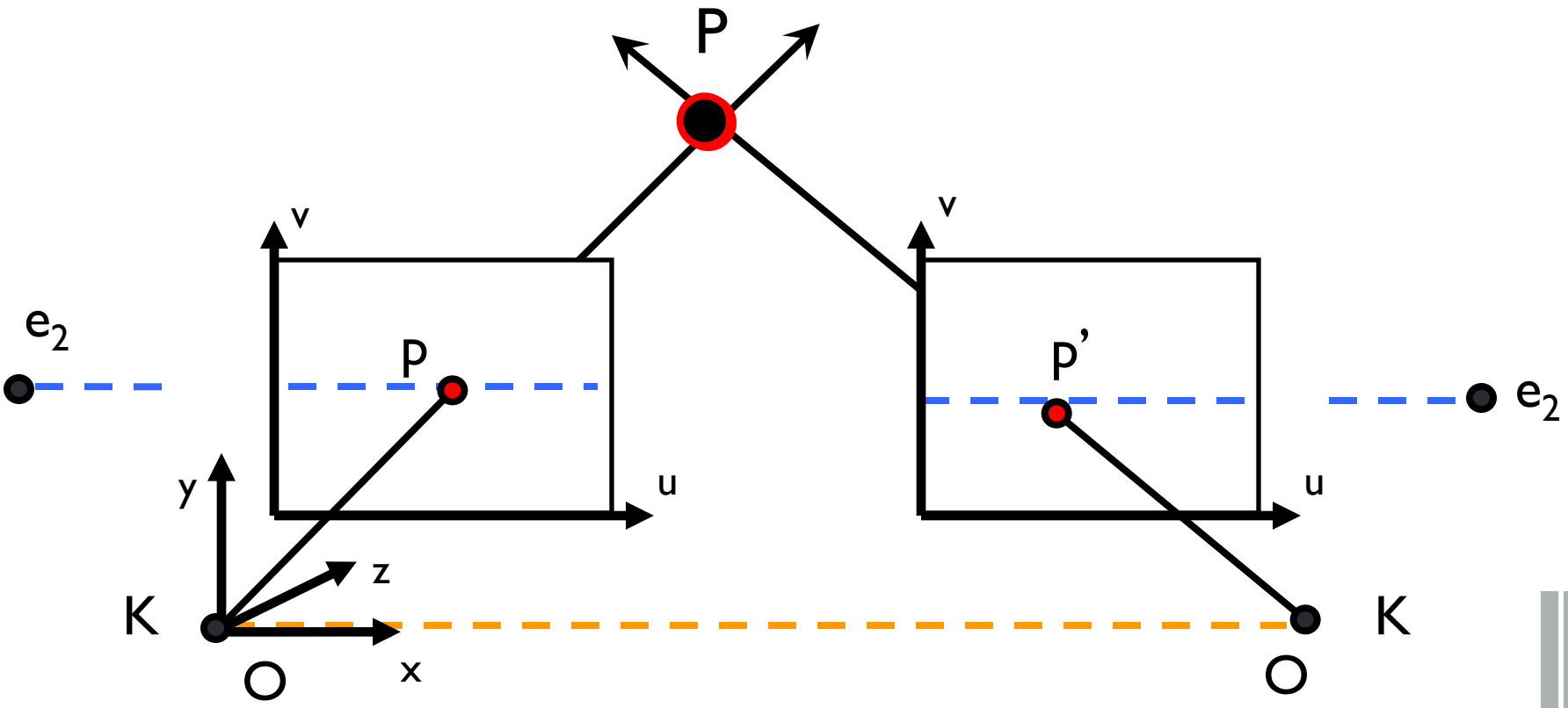
Parallel image planes



- Parallel epipolar lines
- Epipoles at infinity
- v = v'

Rectification: making two images "parallel"

Parallel image planes

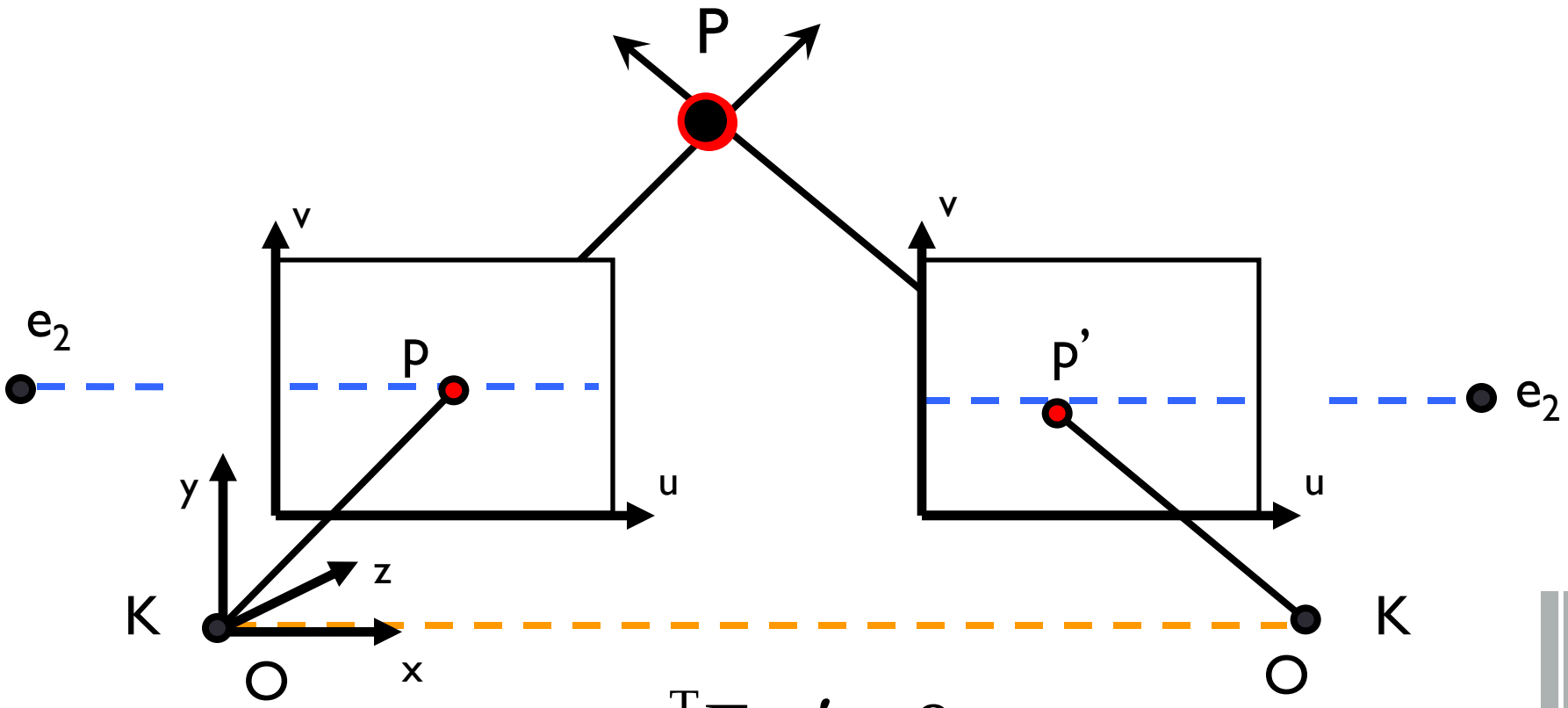


$K_1 = K_2$ = known

x parallel to O_1O_2

$$E = [t_\times]R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \rightarrow v = v'?$$

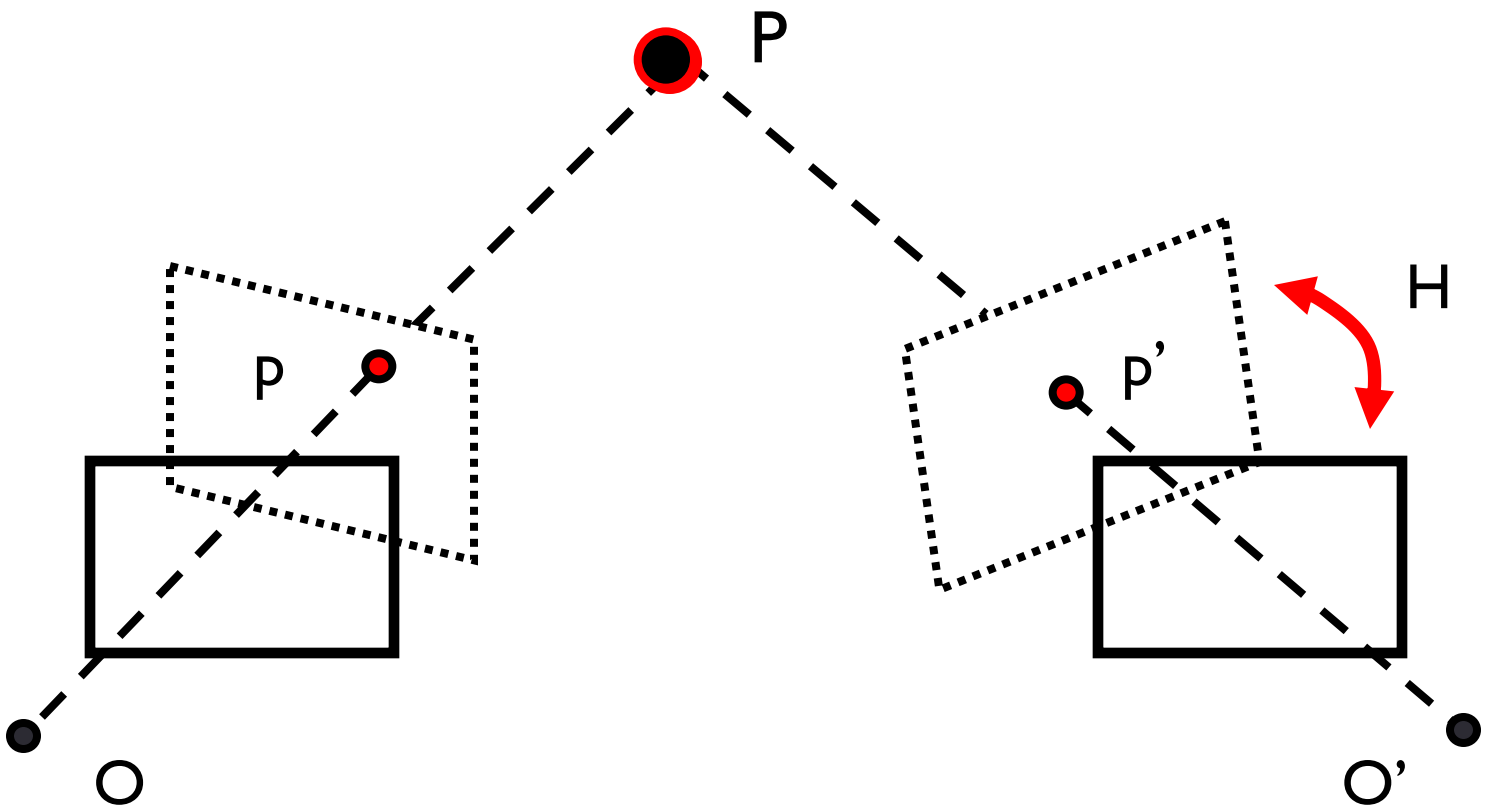
Parallel image planes



$$p^T E\, p' = 0$$

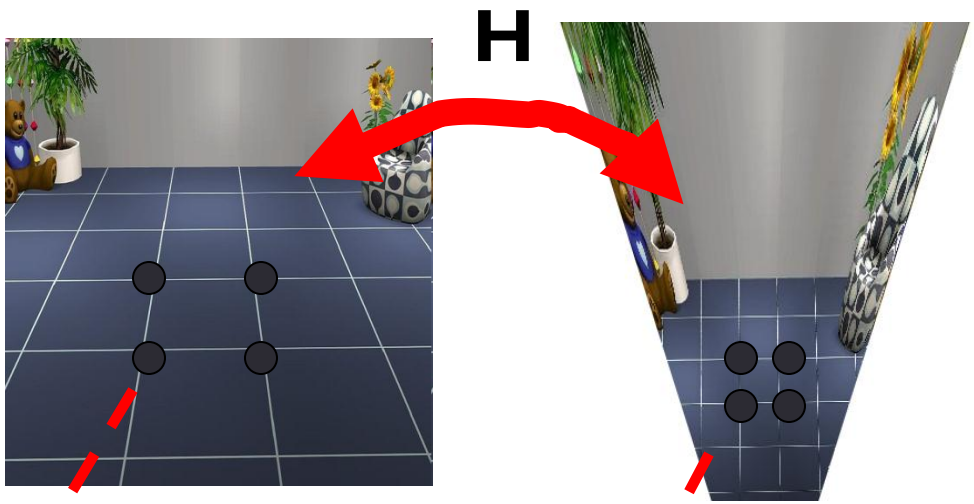
$$\begin{pmatrix} u & v & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \qquad \begin{pmatrix} u & v & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -T \\ Tv' \end{pmatrix} = 0 \qquad Tv = Tv'$$

Making image planes parallel



GOAL: Estimate the perspective transformation H that makes the images parallel

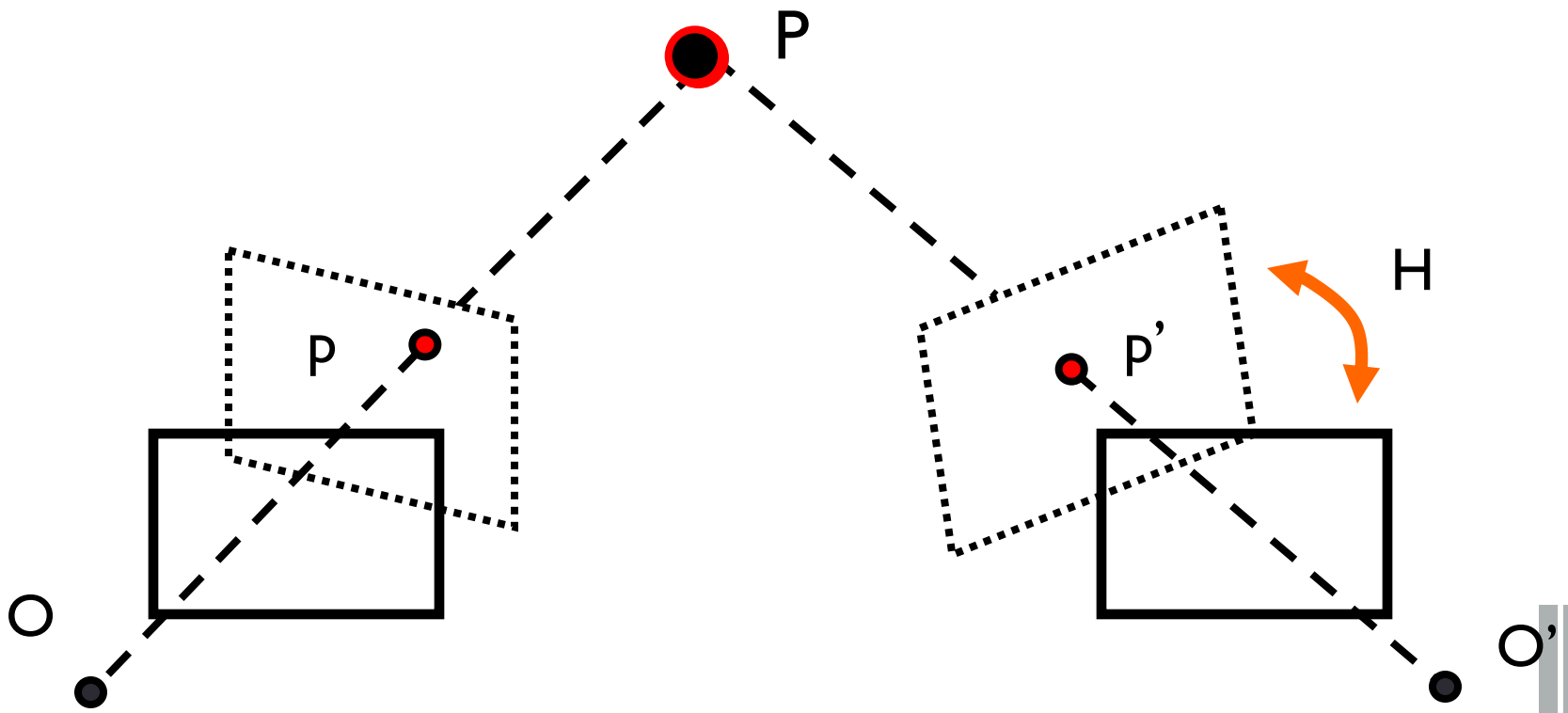
Projective transformation



$$x_i' = H\, x_i$$

Now we don't have the destination image ☹

Making image planes parallel



GOAL: Estimate the perspective transformation H that makes images parallel

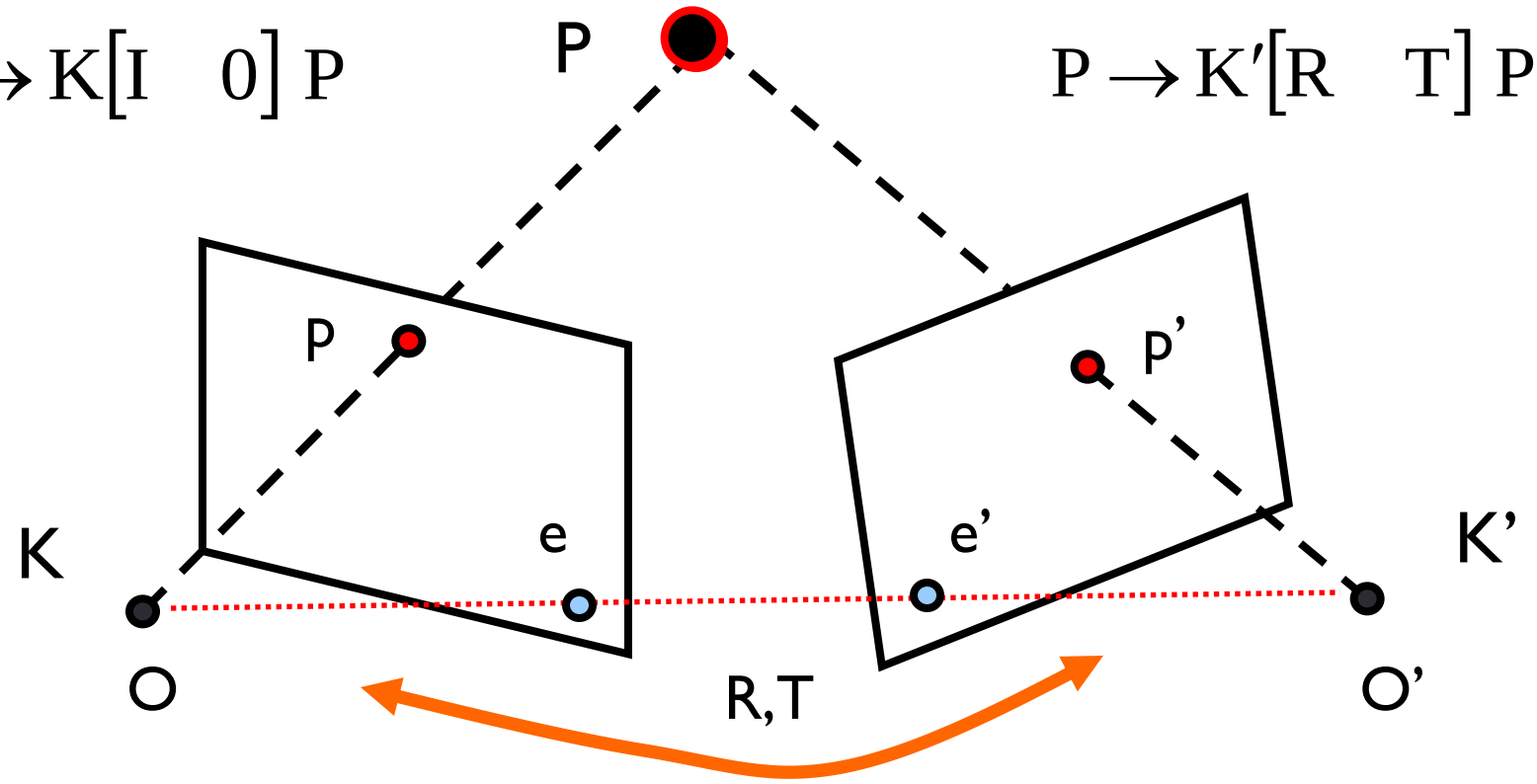
Impose v'=v → How? → Map e to infinity
Make epipolar lines parallel

- This leaves (4) degrees of freedom for determining H
- If the appropriate H is not chosen, severe projective distortions on image take place
- We impose a number of restriction while computing H

Making image planes parallel

$$P \rightarrow K[I \quad 0]\, P$$

$$P \rightarrow K'[R \quad T]\, P$$



0. Compute epipoles

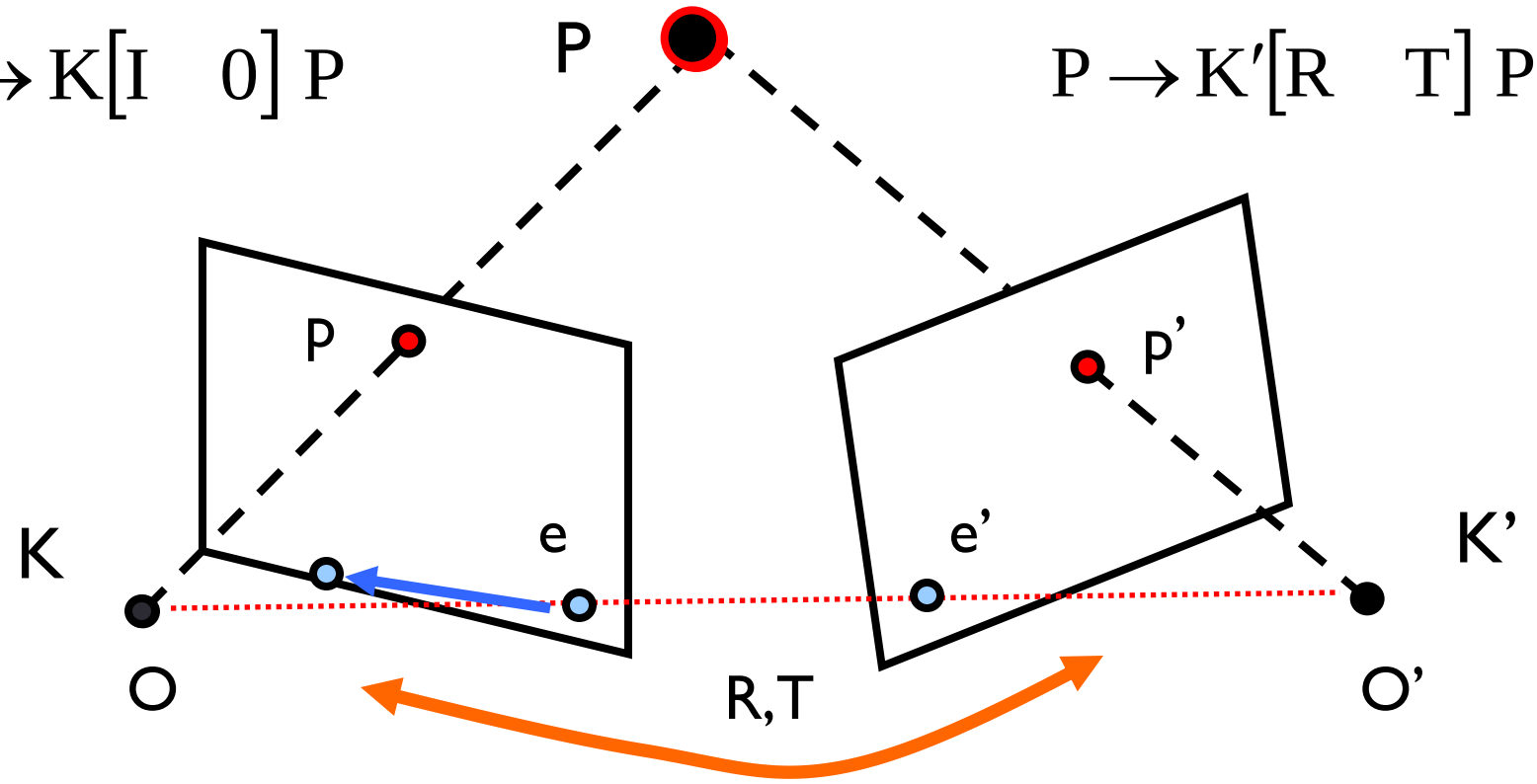
$$e = K\, R^T\, T = [e_1 \quad e_2 \quad 1]^T$$

$$e' = K'\, T$$

Making image planes parallel

$$P \rightarrow K[I \quad 0]\,P$$

$$P \rightarrow K'[R \quad T]\,P$$

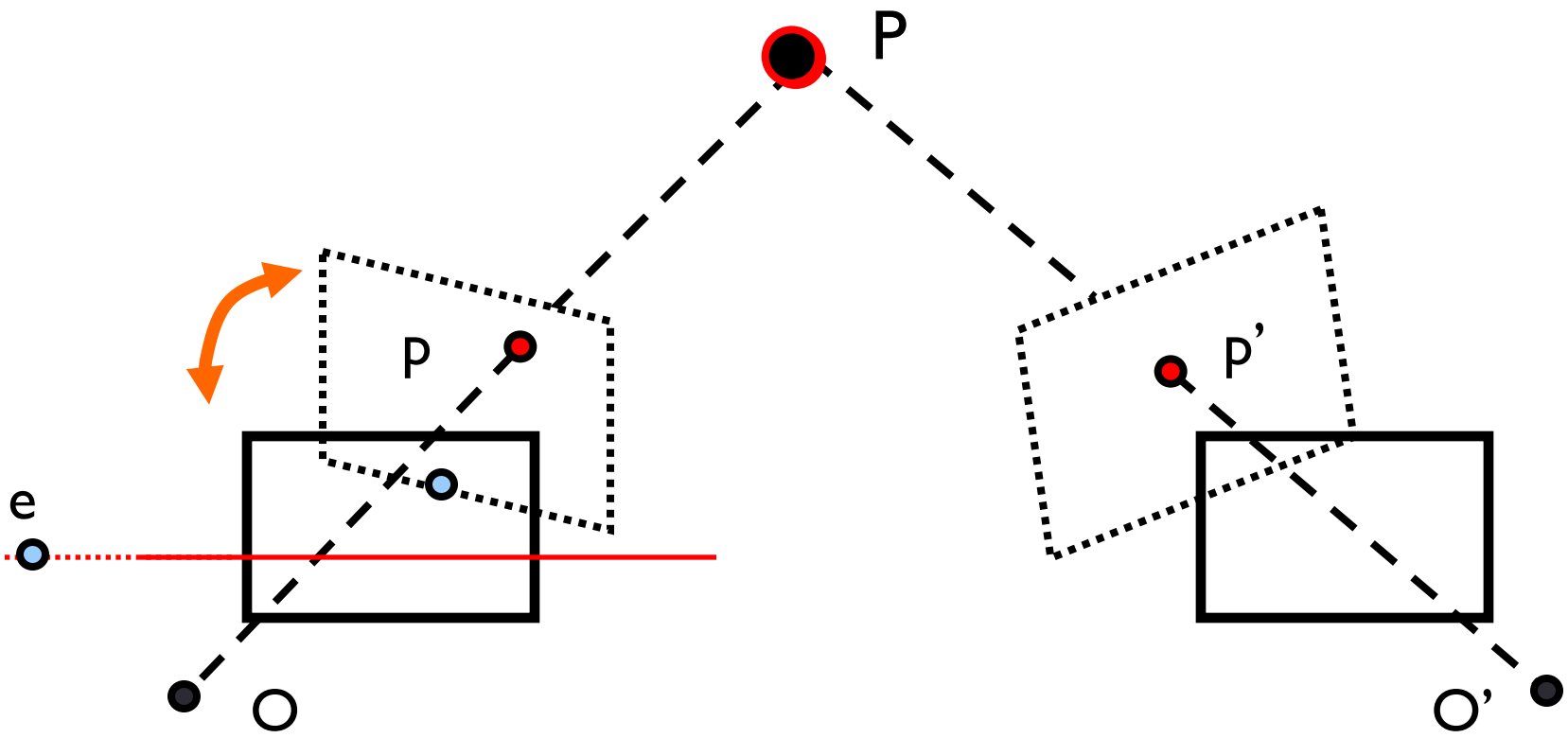


1. Map e to the x-axis at location $[1,0,1]^T$ (normalization)

$$e = [e_1 \quad e_2 \quad 1]^T \rightarrow [1 \quad 0 \quad 1]^T$$

$$H_1 = T_H \qquad T_H = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Making image planes parallel



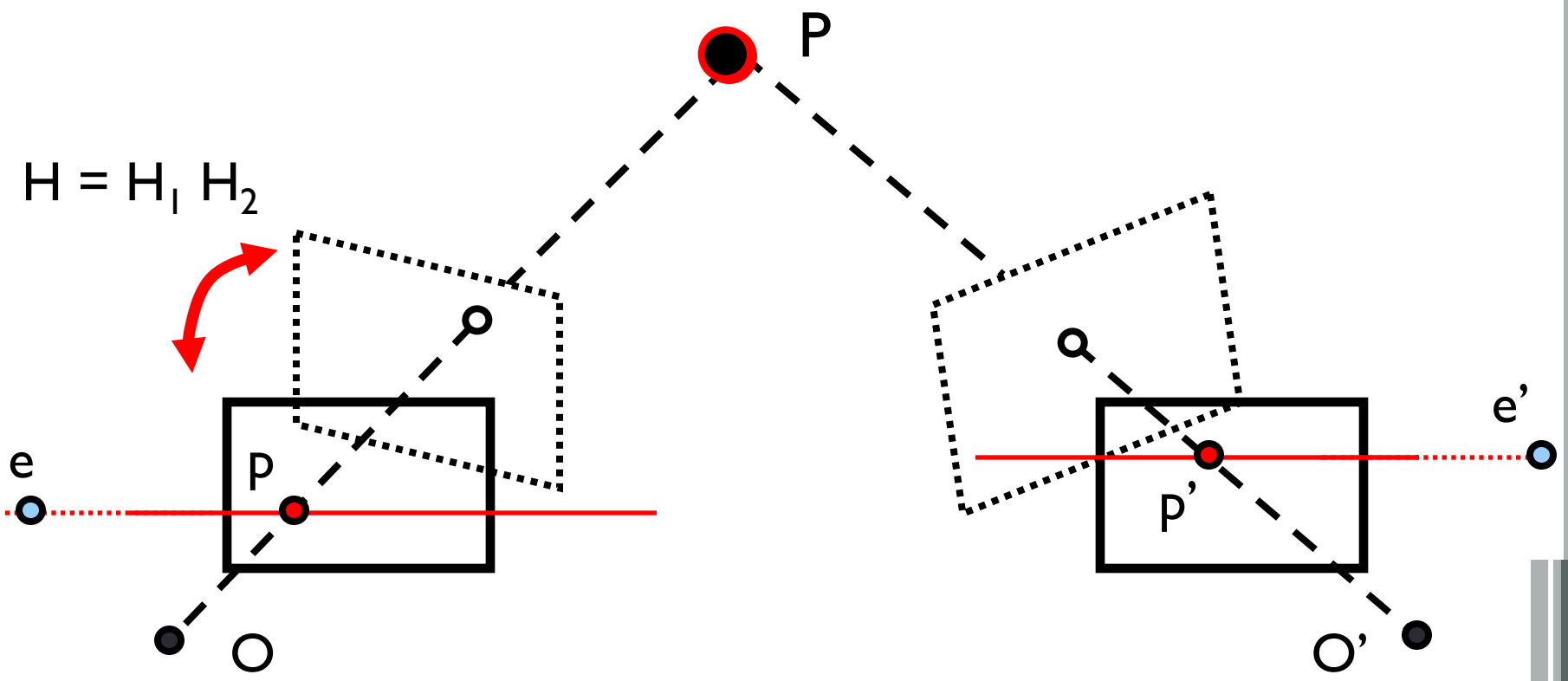
2. Send epipole to infinity:

$$e = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^T \rightarrow \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

$$H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Minimizes the distortion in a neighborhood (approximates id. mapping)

Making image planes parallel

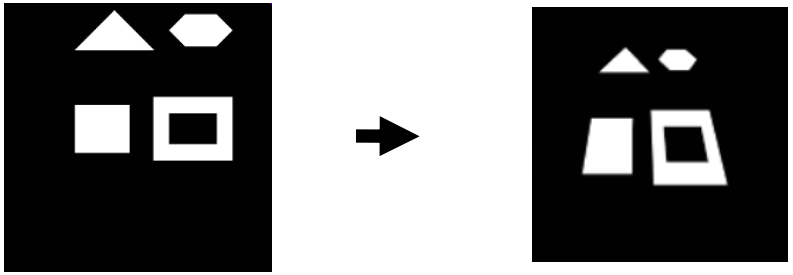


3. Define: $H = H_1 H_2$

4. Align epipolar lines

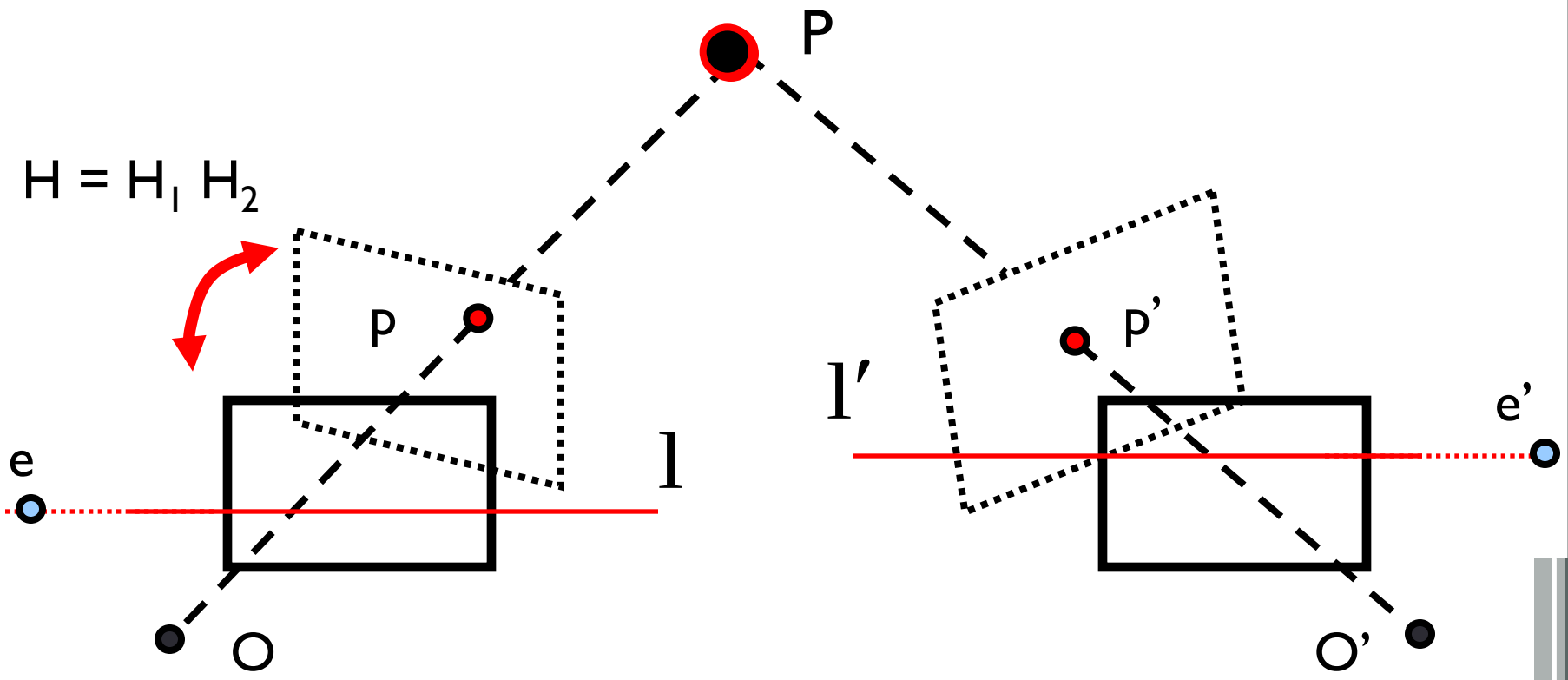
Projective transformation of a line (in 2D)

$$H = \begin{bmatrix} A & t \\ v & b \end{bmatrix}$$



$$l' = H^{-T} l$$

Making image planes parallel



3. Define: $H = H_1 H_2$

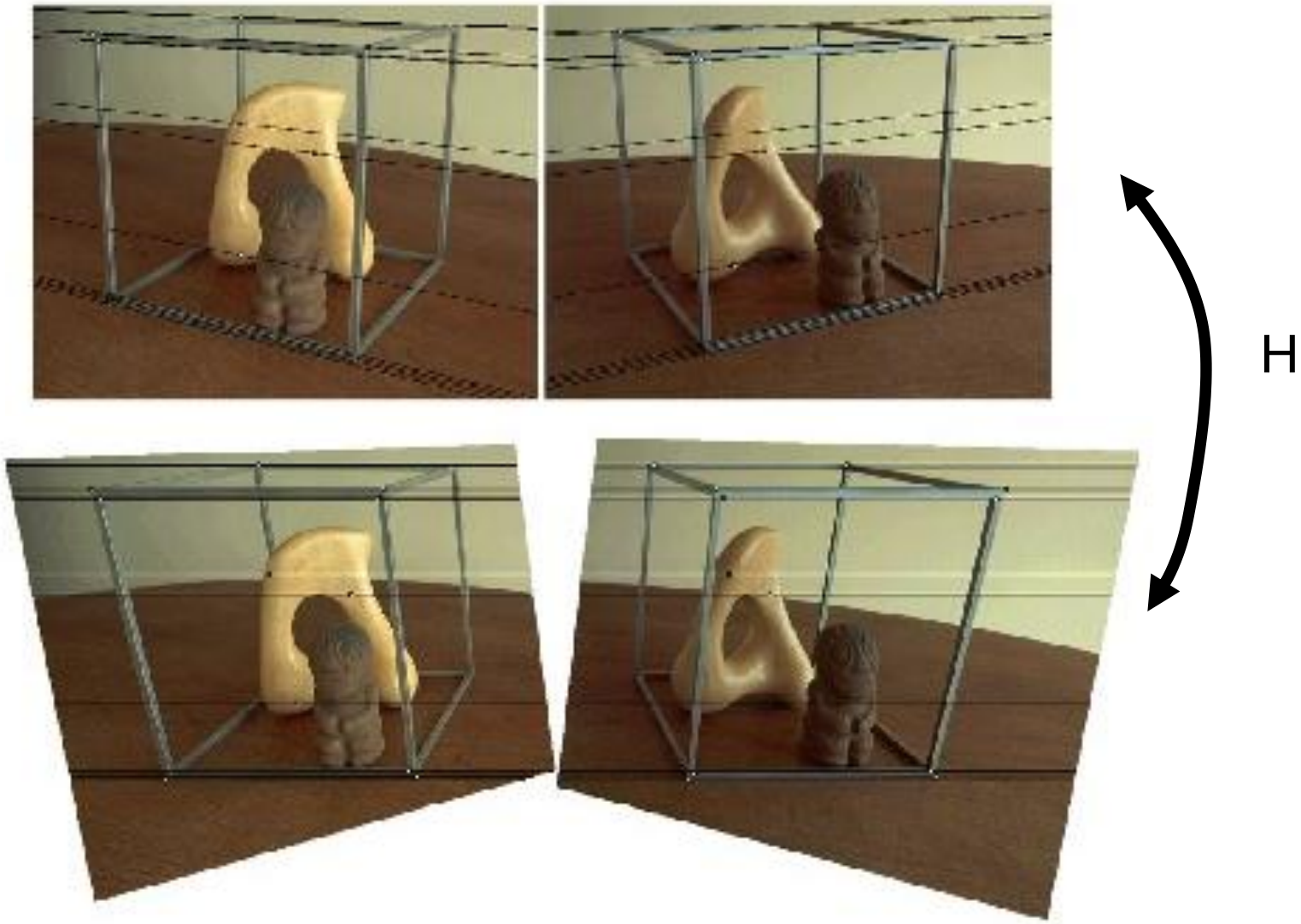
4. Align epipolar lines

$$H'^{-T} l' = H^{-T} l$$

H, H' called matched pair of transformation

[HZ] Chapters: 11 (sec. 11.12)

Making image planes parallel

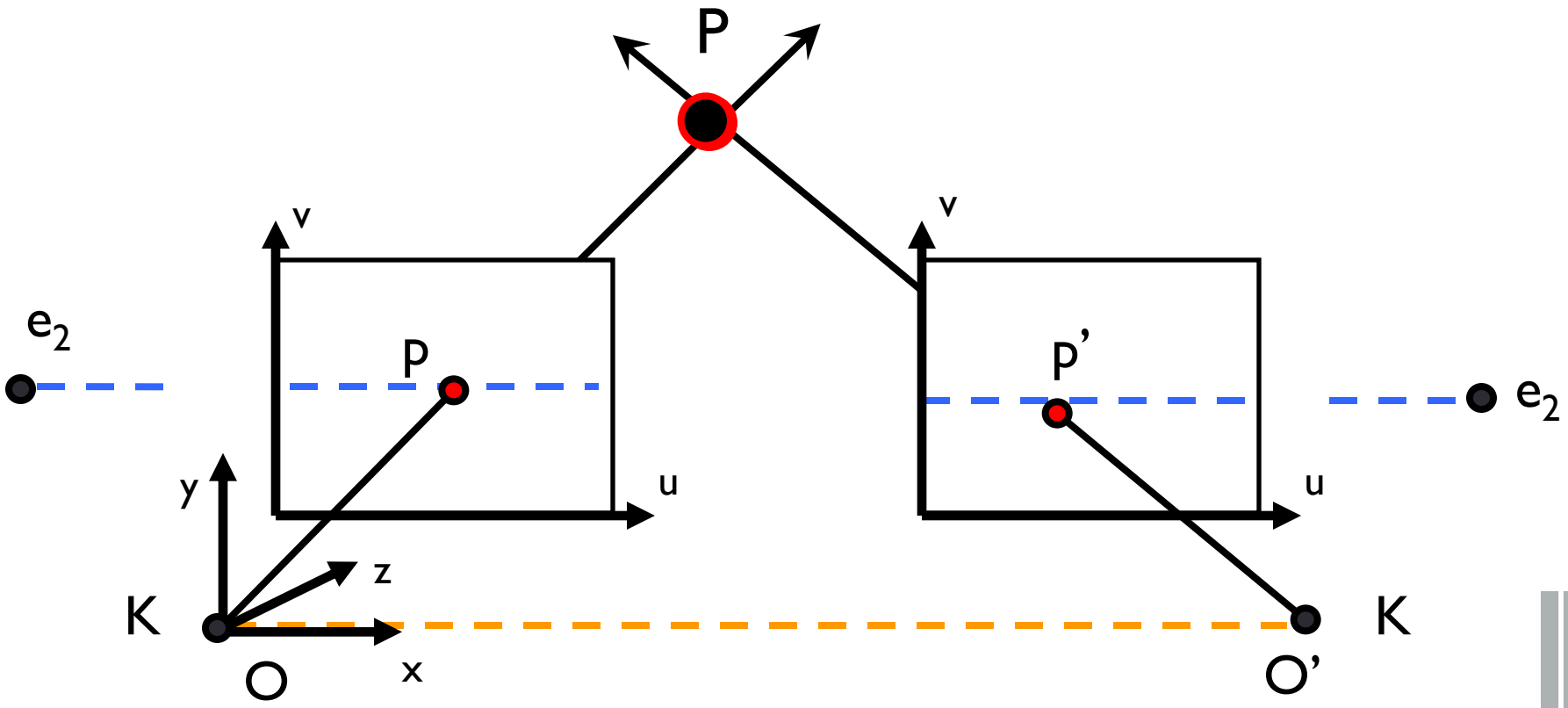


Courtesy figure S. Lazebnik

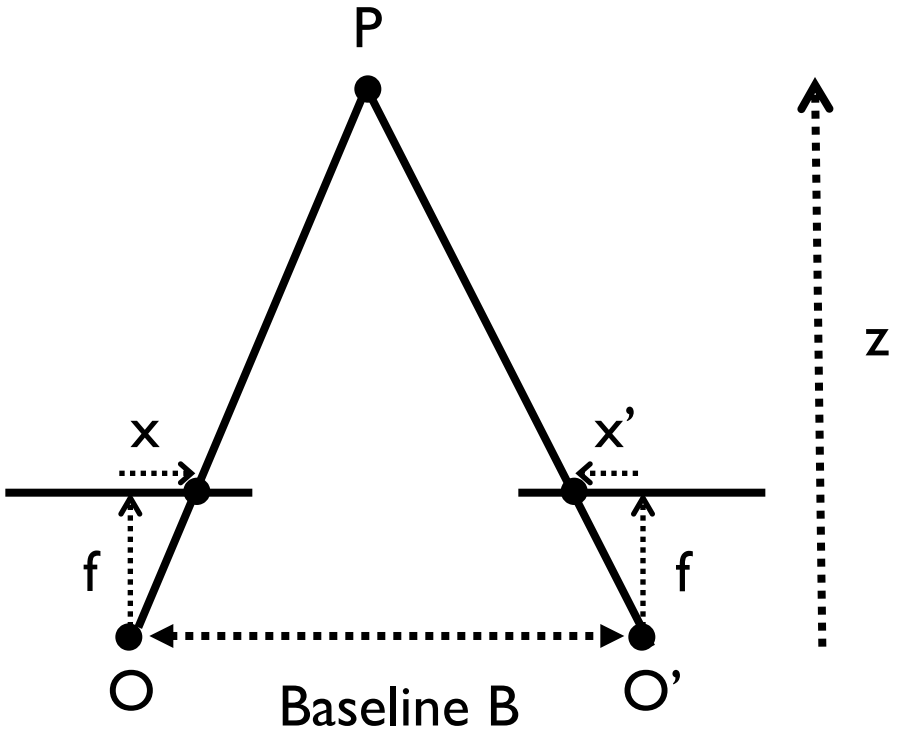
Why rectification is useful?

- Makes the correspondence problem easier
- Makes triangulation easy

Computing depth



Computing depth



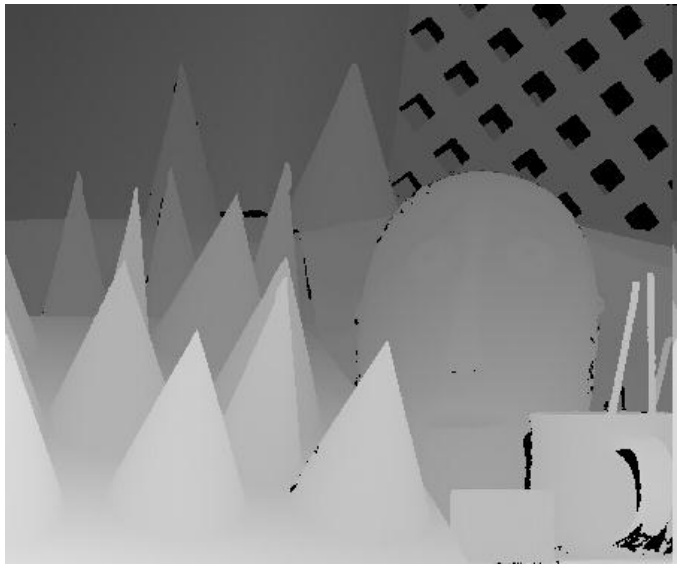
$$x - x' = \frac{B \cdot f}{z}$$ = disparity

Note: Disparity is inversely proportional to depth

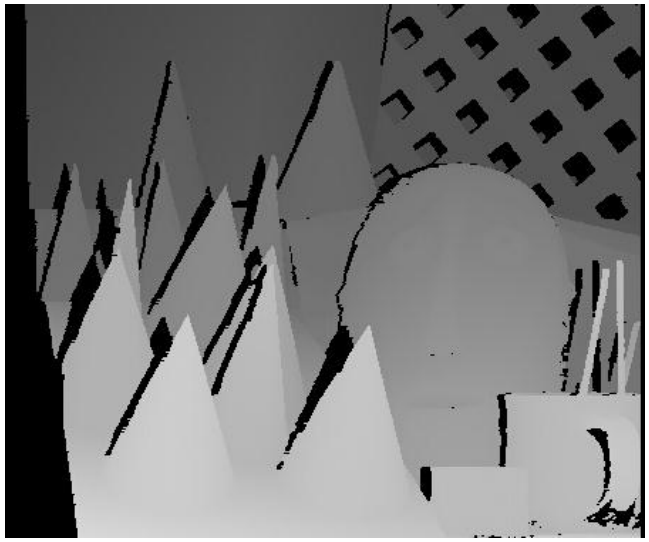
Disparity maps

$$x - x' = \frac{B \cdot f}{z}$$

Stereo pair



Disparity map / depth map



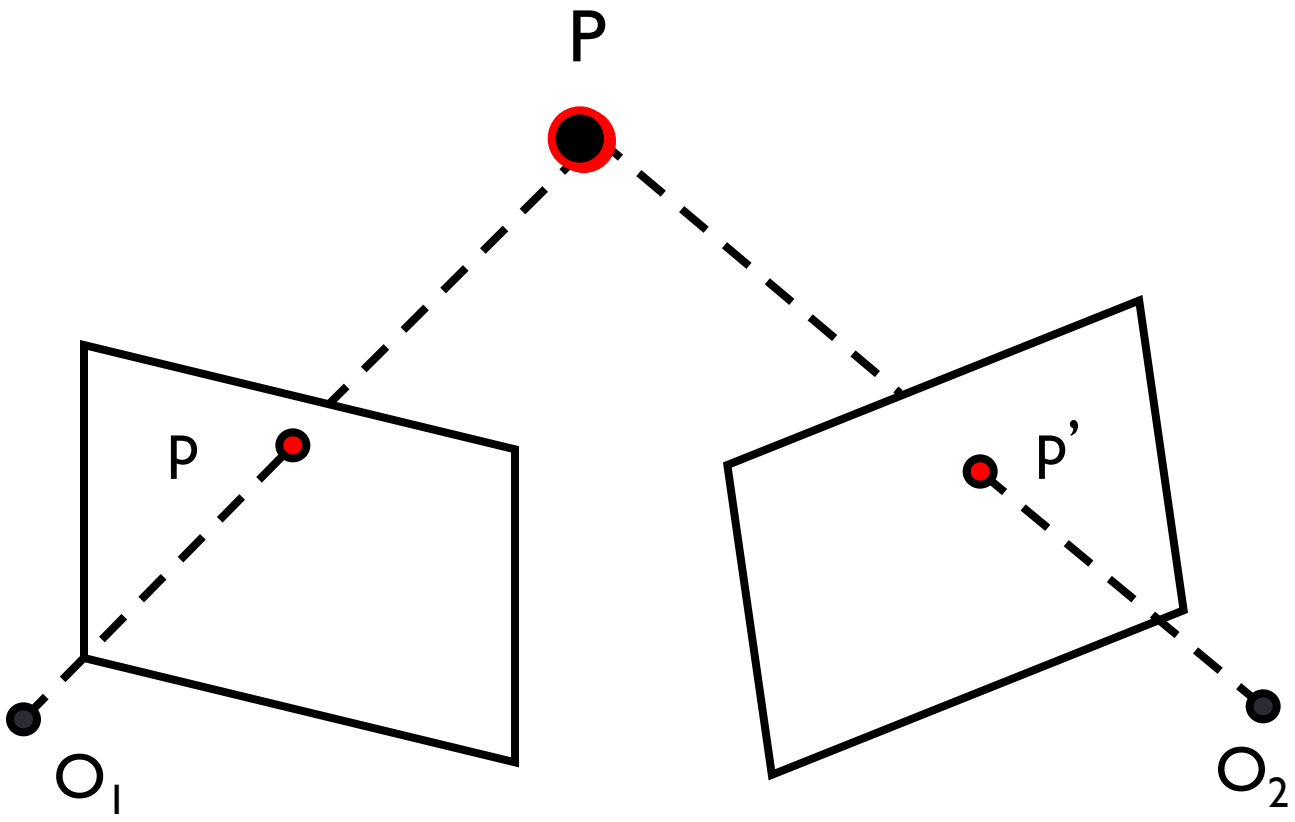
Disparity map with occlusions

Outline

- Stereo Systems

- Stereo vision
- Rectification
- Correspondence problem
- Active stereo vision systems

Correspondence problem



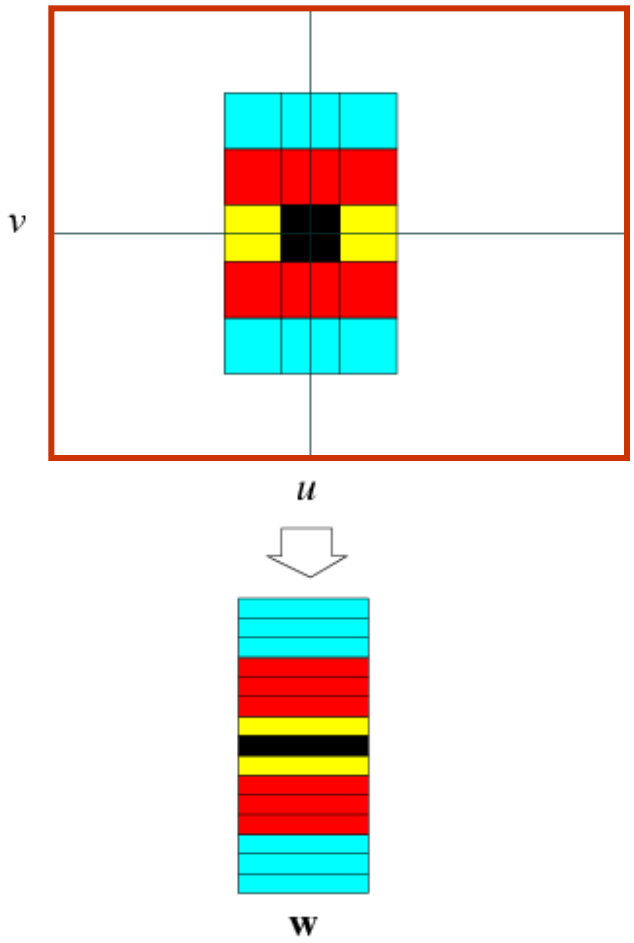
Given a point in 3d, discover corresponding observations in left and right images [also called binocular fusion problem]

Correspondence problem

- A Cooperative Model (Marr and Poggio, 1976)
- Correlation Methods (1970--)
- Multi-Scale Edge Matching (Marr, Poggio and Grimson, 1979-81)

[FP] Chapters: 11

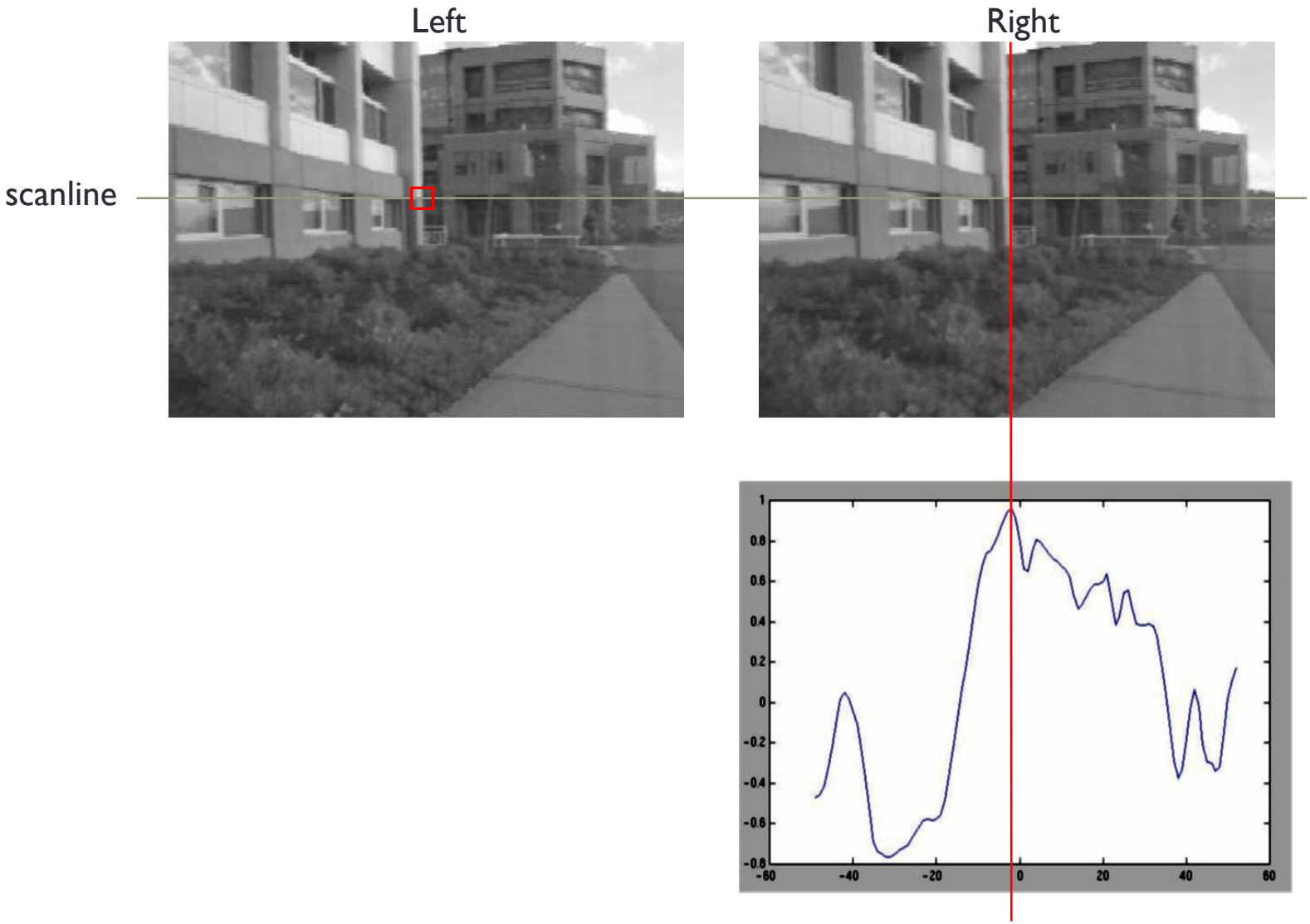
Correlation Methods (1970--)



Slide the window along the epipolar line until w·w' is maximized.

courtesy slide to J. Ponce

Correlation methods



Credit slide S. Lazebnik

Image Normalization

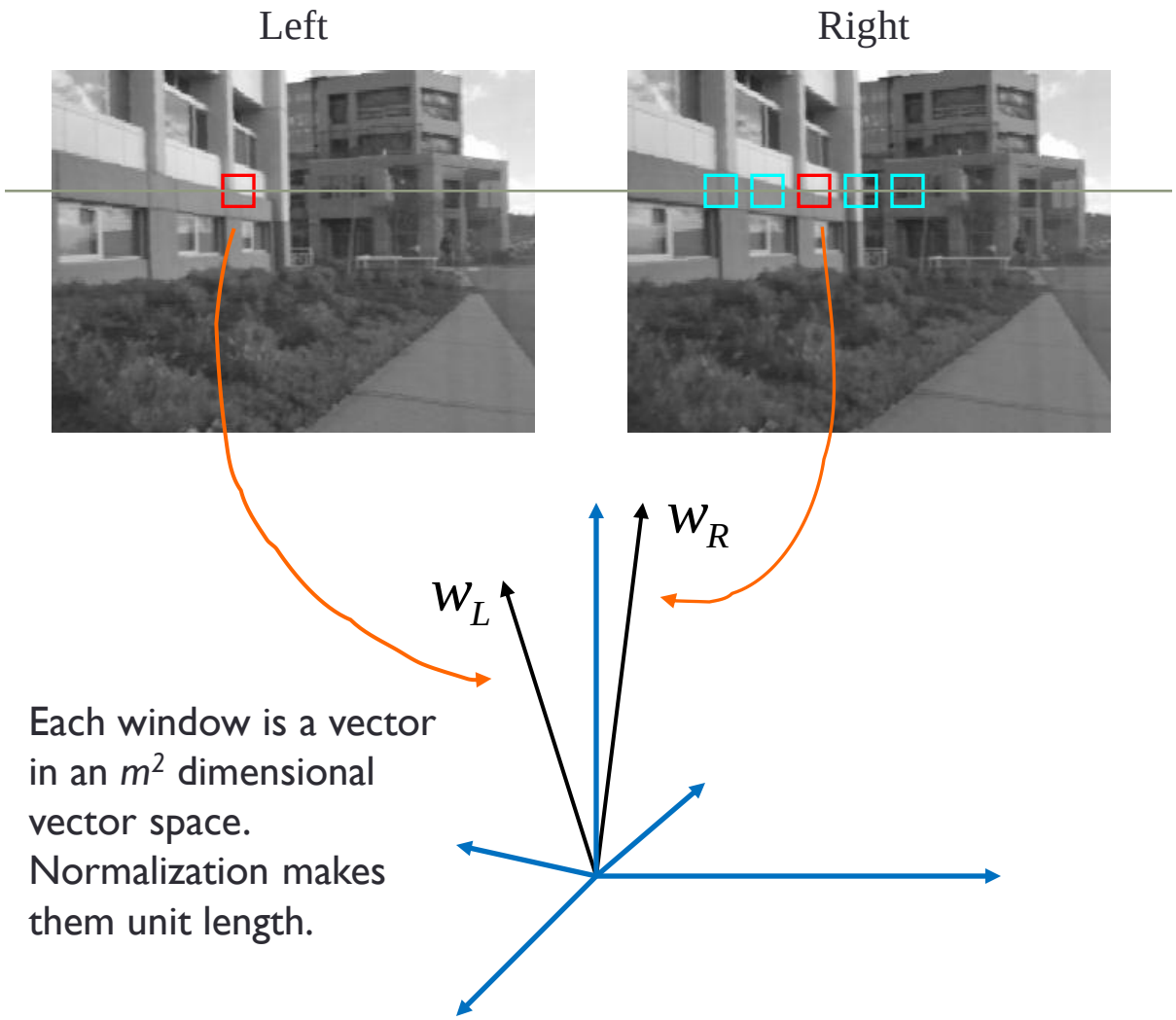
- Even when the cameras are identical models, there can be differences in gain and sensitivity.
- The cameras do not see exactly the same surfaces, so their overall light levels can differ.
- For these reasons and more, it is a good idea to normalize the pixels in each window:

$$\bar{I} = \frac{1}{|W_m(x,y)|} \sum_{(u,v)\in W_m(x,y)} I(u,v) \qquad \text{Average pixel}$$

$$\|I\|_{W_m(x,y)} = \sqrt{\sum_{(u,v)\in W_m(x,y)} [I(u,v)]^2} \qquad \text{Window magnitude}$$

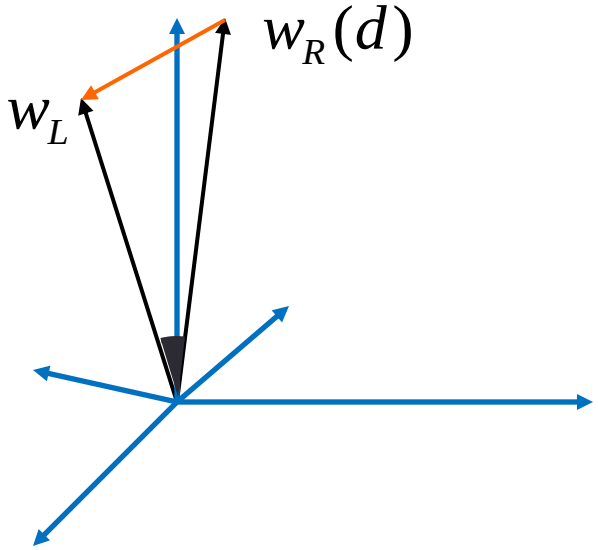
$$\hat{I}(x,y) = \frac{I(x,y) - \bar{I}}{\|I - \bar{I}\|_{W_m(x,y)}} \qquad \text{Normalized pixel}$$

Images as Vectors



Each window is a vector in an m^2 dimensional vector space. Normalization makes them unit length.

Image Metrics



(Normalized) Sum of Squared Differences

$$C_{\text{SSD}}(d) = \sum_{(u,v)\in W_m(x,y)} [\hat{I}_L(u,v) - \hat{I}_R(u-d,v)]^2$$

$$= \left\| w_L - w_R(d) \right\|^2$$

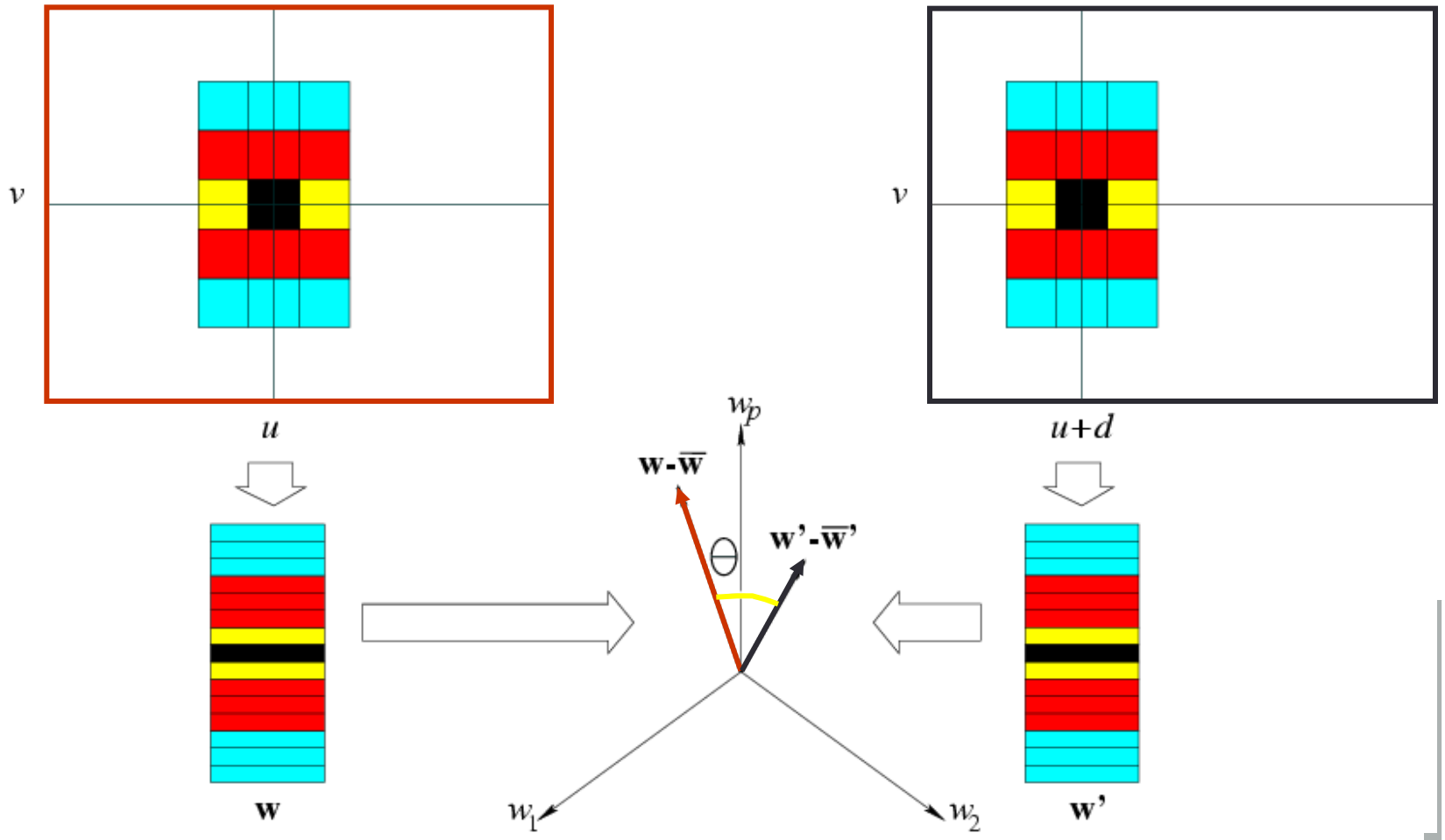
Normalized Correlation

$$C_{\text{NC}}(d) = \sum_{(u,v)\in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v)$$

$$= w_L \cdot w_R(d) = \cos\theta$$

$$d^* = \arg\min_d \left\| w_L - w_R(d) \right\|^2 = \arg\max_d w_L \cdot w_R(d)$$

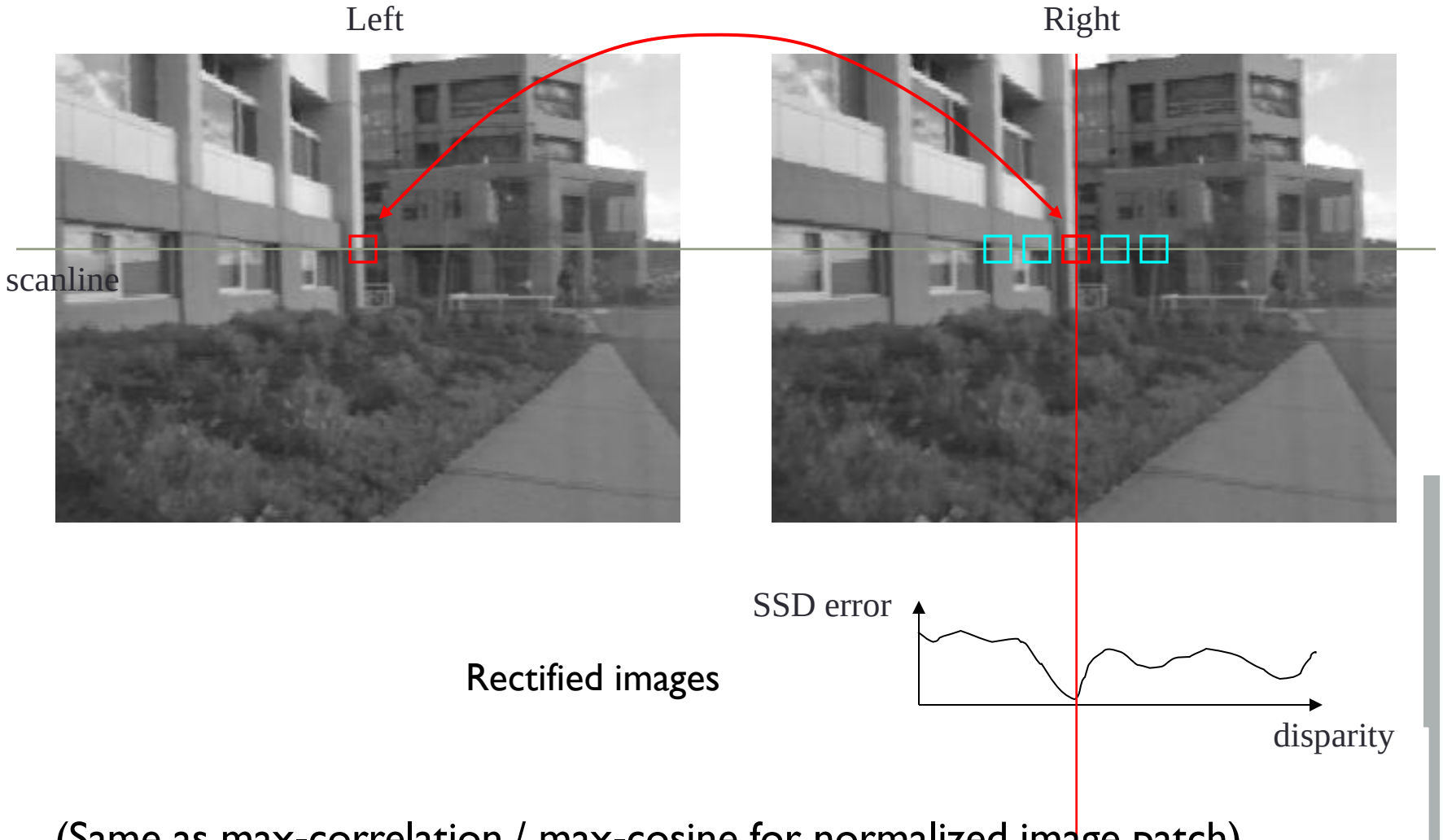
Correlation Methods (1970--)



Normalized Correlation; minimize:

$$\frac{(w-\overline{w})(w'-\overline{w}')}{\left\|(w-\overline{w})(w'-\overline{w}')\right\|}$$

Correspondence via Correlation

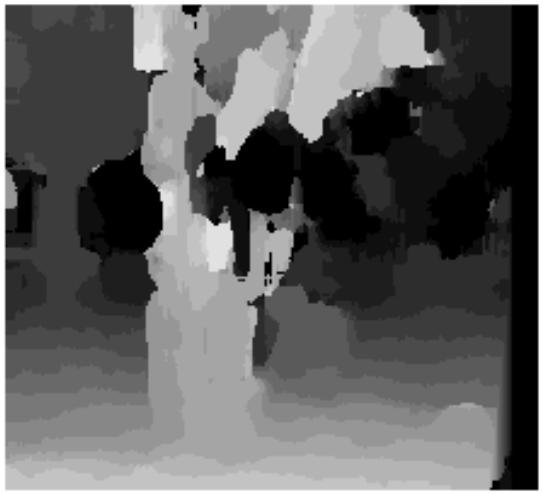


Rectified images

(Same as max-correlation / max-cosine for normalized image patch)

Correlation methods

Window size = 3



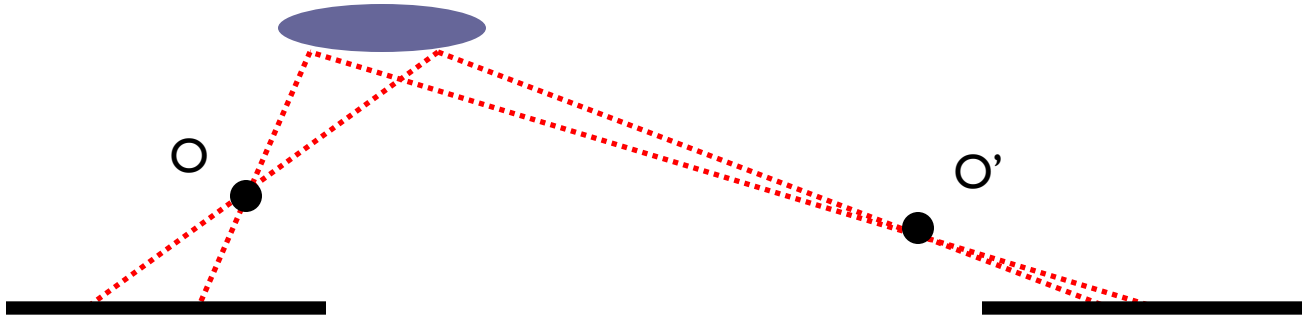
Window size = 20

- Smaller window
  - More detail
  - More noise
- Larger window
  - Smoother disparity maps
  - Less prone to noise

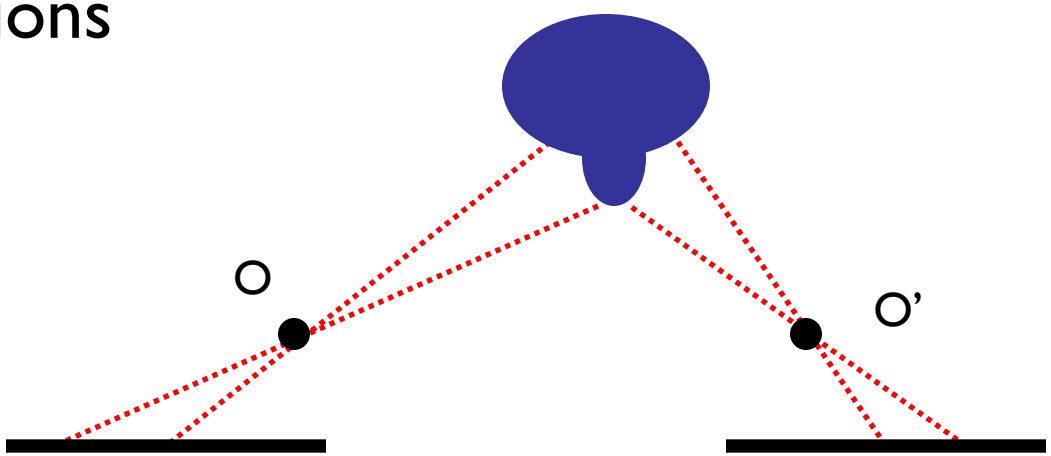
Credit slide S. Lazebnik

Issues

- Foreshortening effect



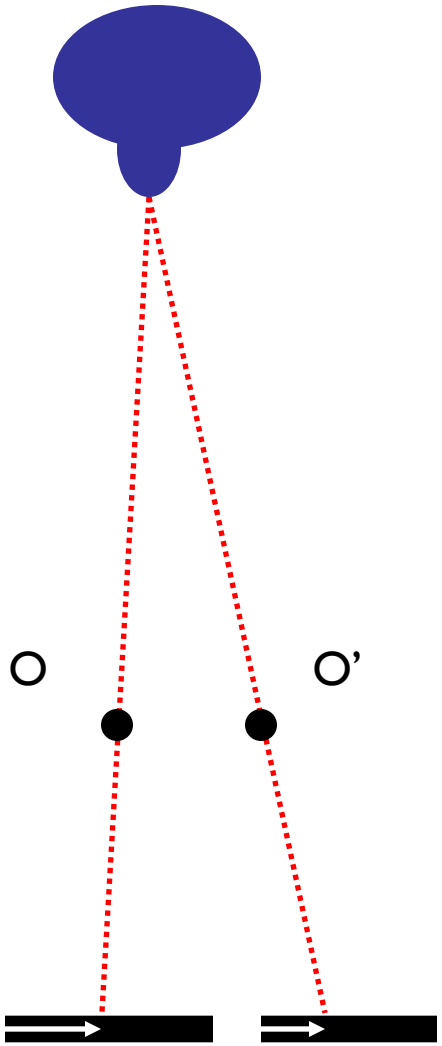
- Occlusions



It is desirable to have small B/z ratio!

Issues

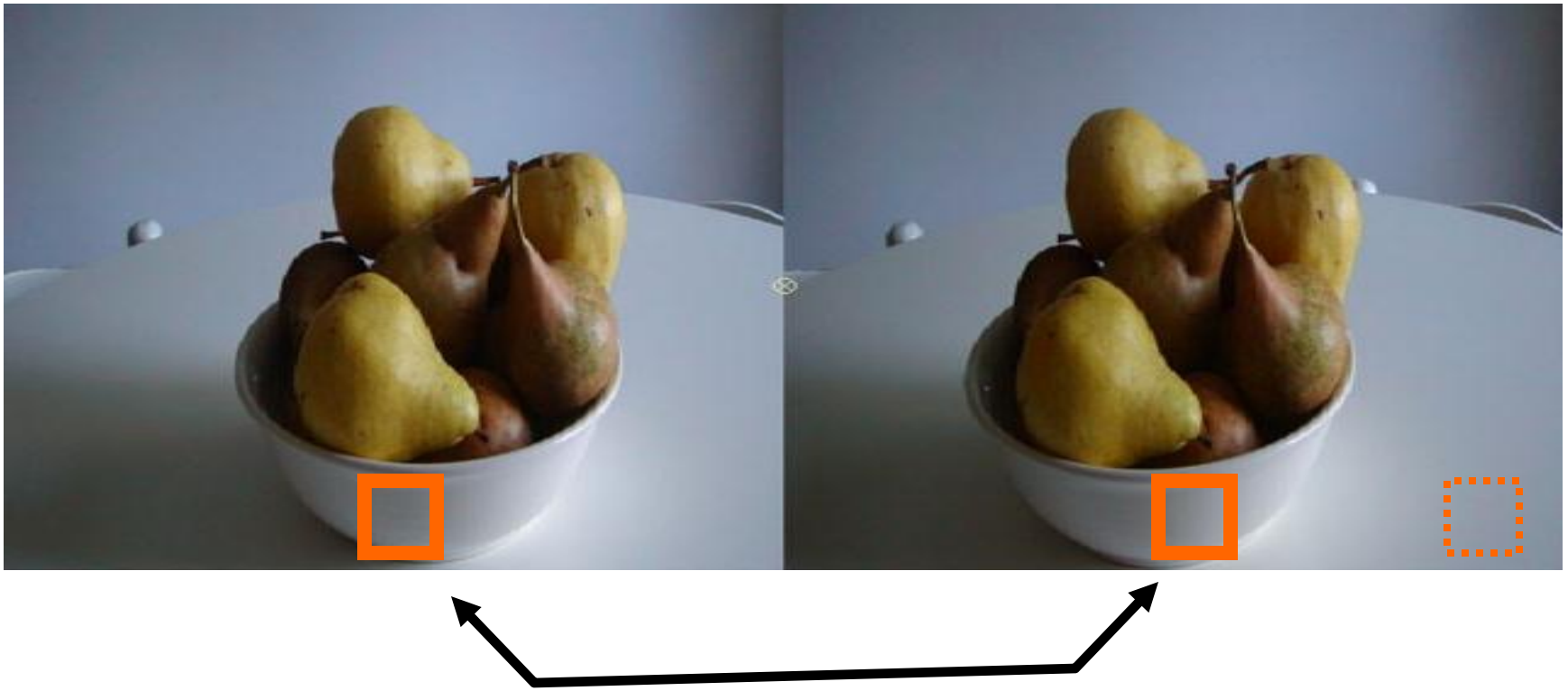
- small B/z ratio



Small error in measurements implies large error in estimating depth

Issues

- Homogeneous regions



Hard to match pixels in these regions

Issues

- Repetitive patterns

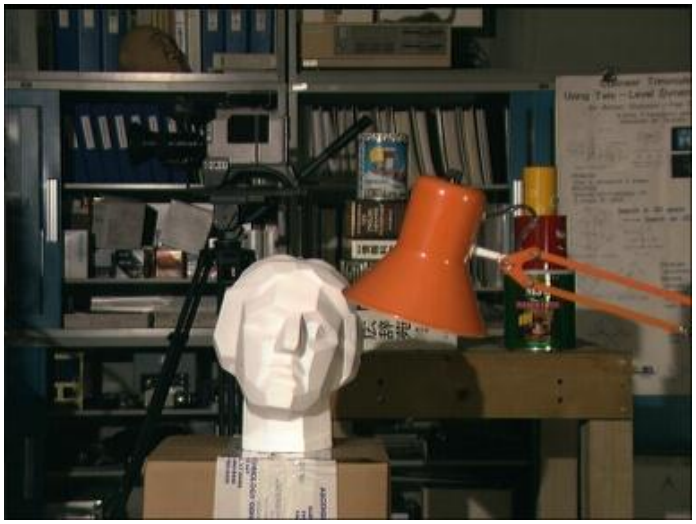
Correspondence problem is difficult!

- Occlusions
- Fore shortening
- Baseline trade-off
- Homogeneous regions
- Repetitive patterns

Apply non-local constraints to help enforce the correspondences

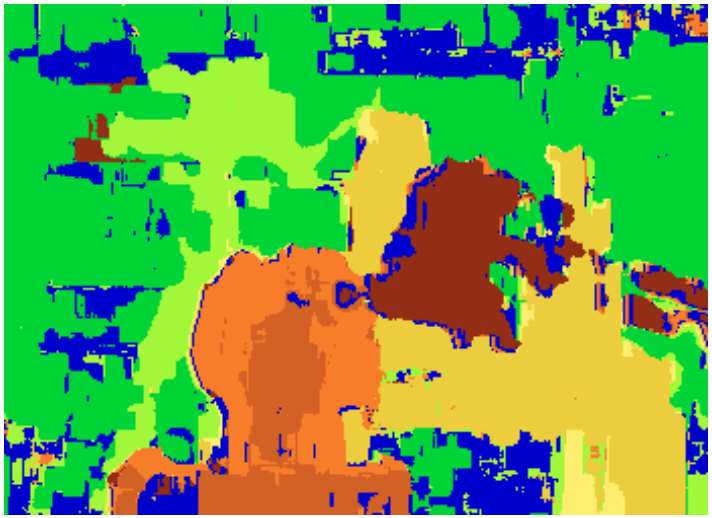
Results with window search

Data



Ground truth

Window-based matching



Credit slide S. Lazebnik

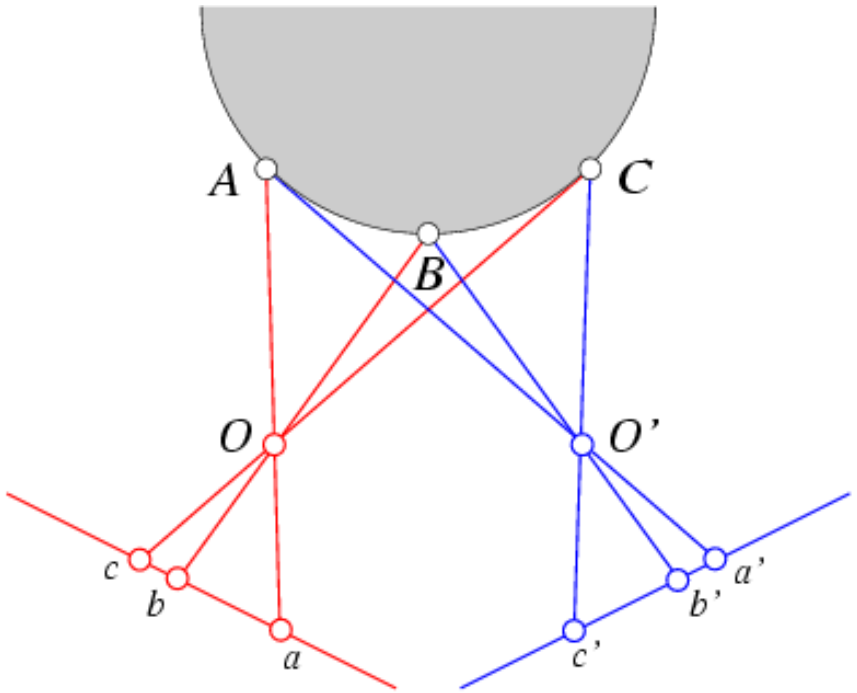
Improving correspondence: Non-local constraints

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image

Improving correspondence: Non-local constraints

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image

- Ordering
  - Corresponding points should be in the same order in both views

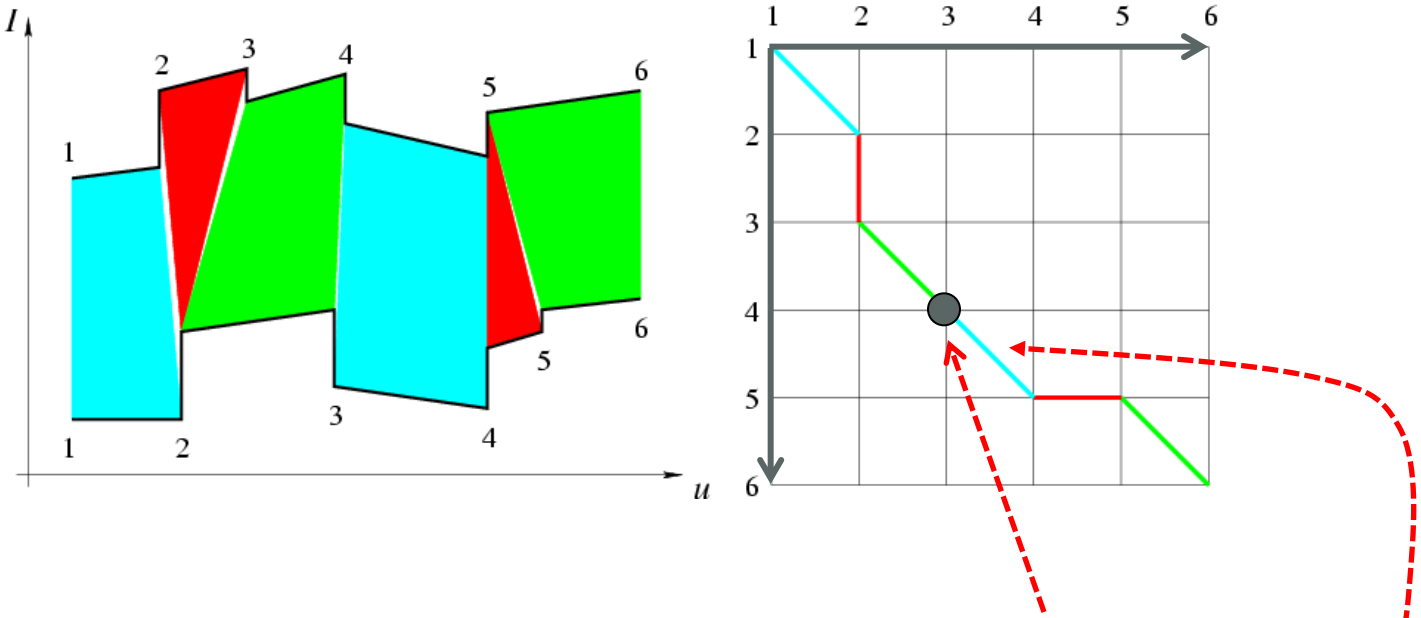


Not always in presence of occlusions!

courtesy slide to J. Ponce

Dynamic Programming (Baker and Binford, 1981)

[Uses ordering constraint]

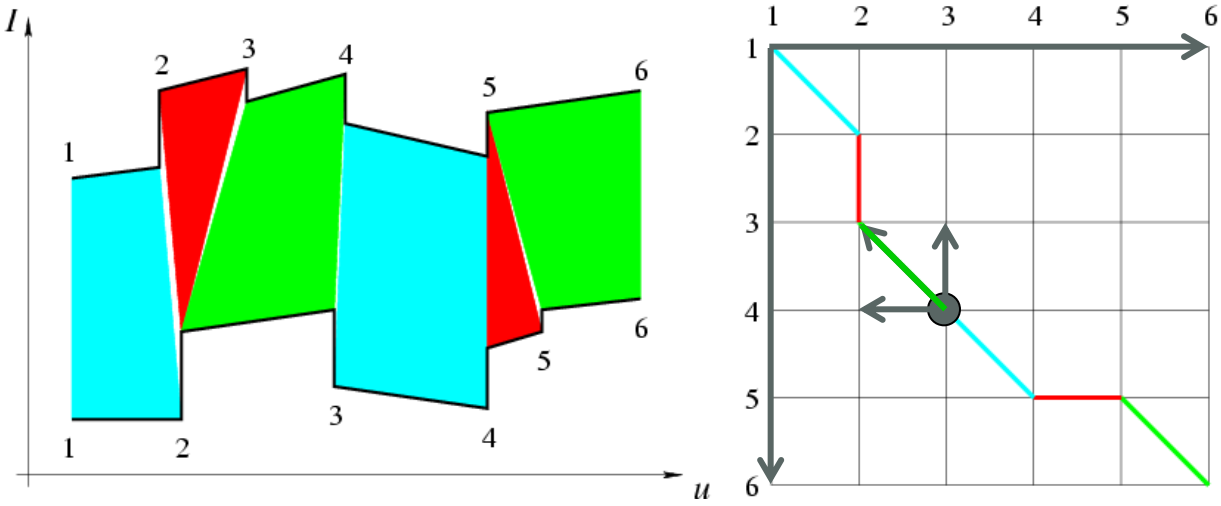


- Nodes = matched feature points (e.g., edge points).
- Arcs = matched intervals along the epipolar lines.
- Arc cost = discrepancy between intervals.

Find the minimum-cost path going monotonically down and right from the top-left corner of the graph to its bottom-right corner.

courtesy slide to J. Ponce

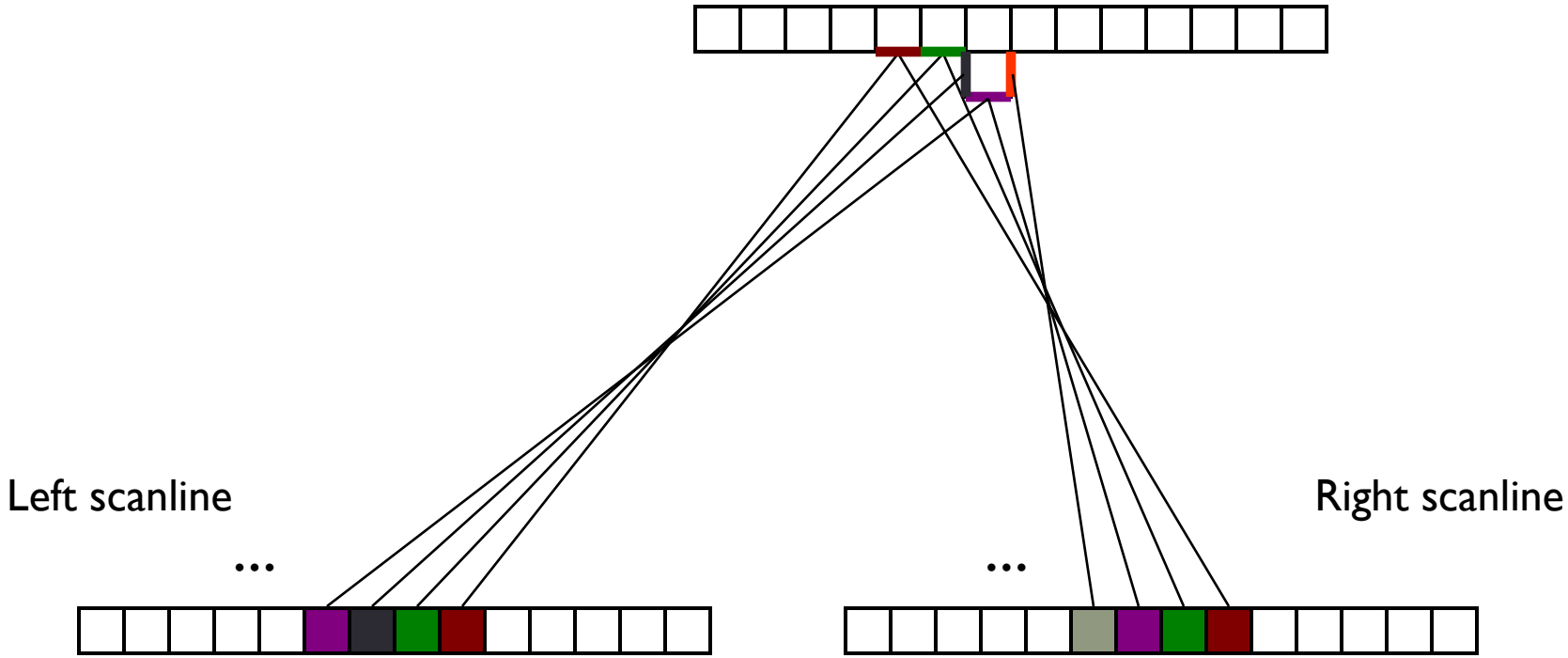
Dynamic Programming (Baker and Binford, 1981)



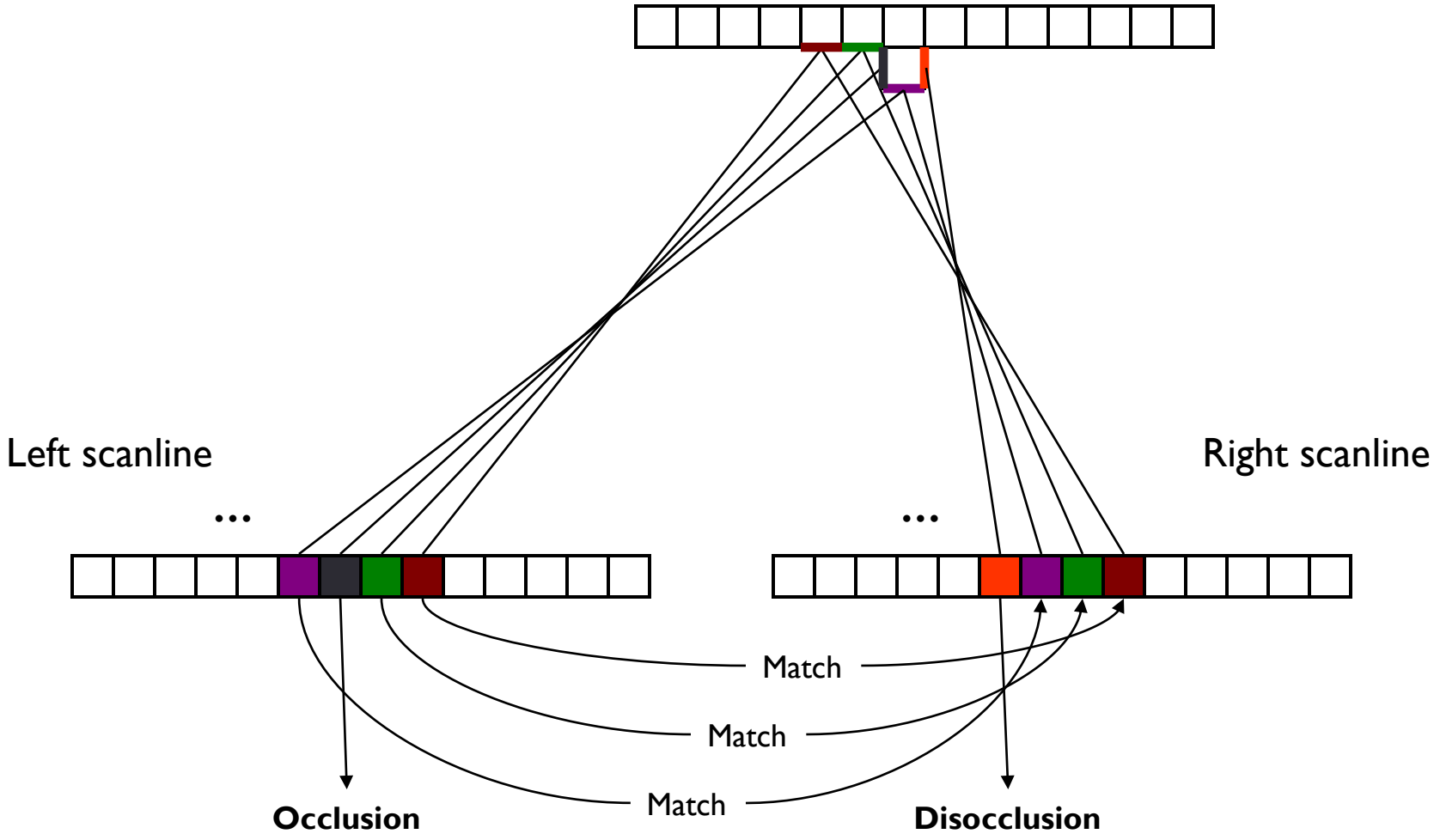

```
% Loop over all nodes (k, l) in ascending order.
for k = 1 to m do
  for l = 1 to n do
    % Initialize optimal cost C(k, l) and backward pointer B(k, l).
    C(k, l) ← +∞; B(k, l) ← nil;
    % Loop over all inferior neighbors (i, j) of (k, l).
    for (i, j) ∈ Inferior − Neighbors(k, l) do
      % Compute new path cost and update backward pointer if necessary.
      d ← C(i, j) + Arc − Cost(i, j, k, l);
      if d < C(k, l) then C(k, l) ← d; B(k, l) ← (i, j) endif;
      endfor;
    endfor;
  endfor;
% Construct optimal path by following backward pointers from (m, n).
P ← {(m, n)}; (i, j) ← (m, n);
while B(i, j) ≠ nil do (i, j) ← B(i, j); P ← {(i, j)} ∪ P endwhile.
```

courtesy slide to J. Ponce

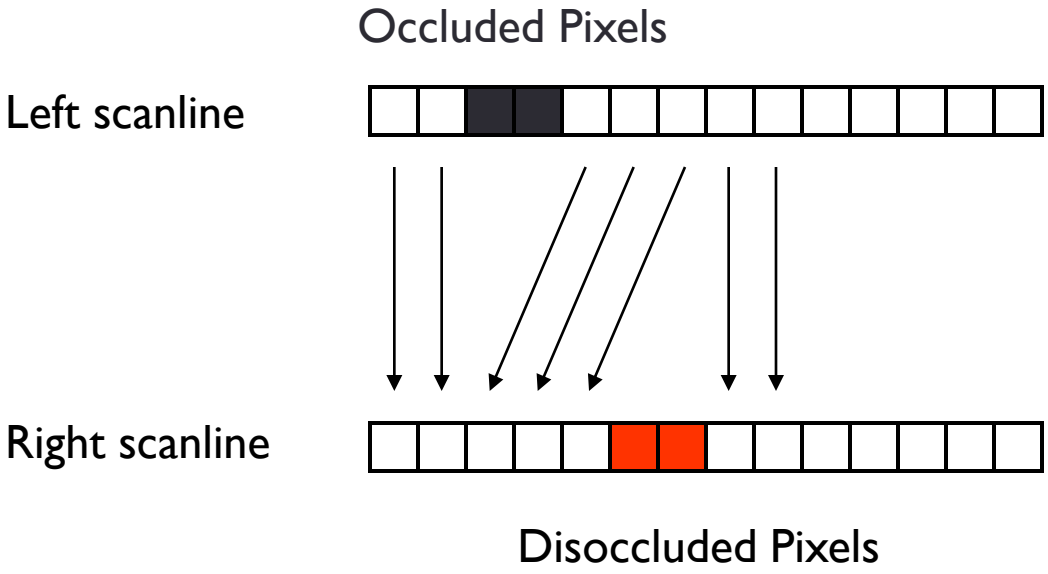
Stereo Correspondences



Stereo Correspondences



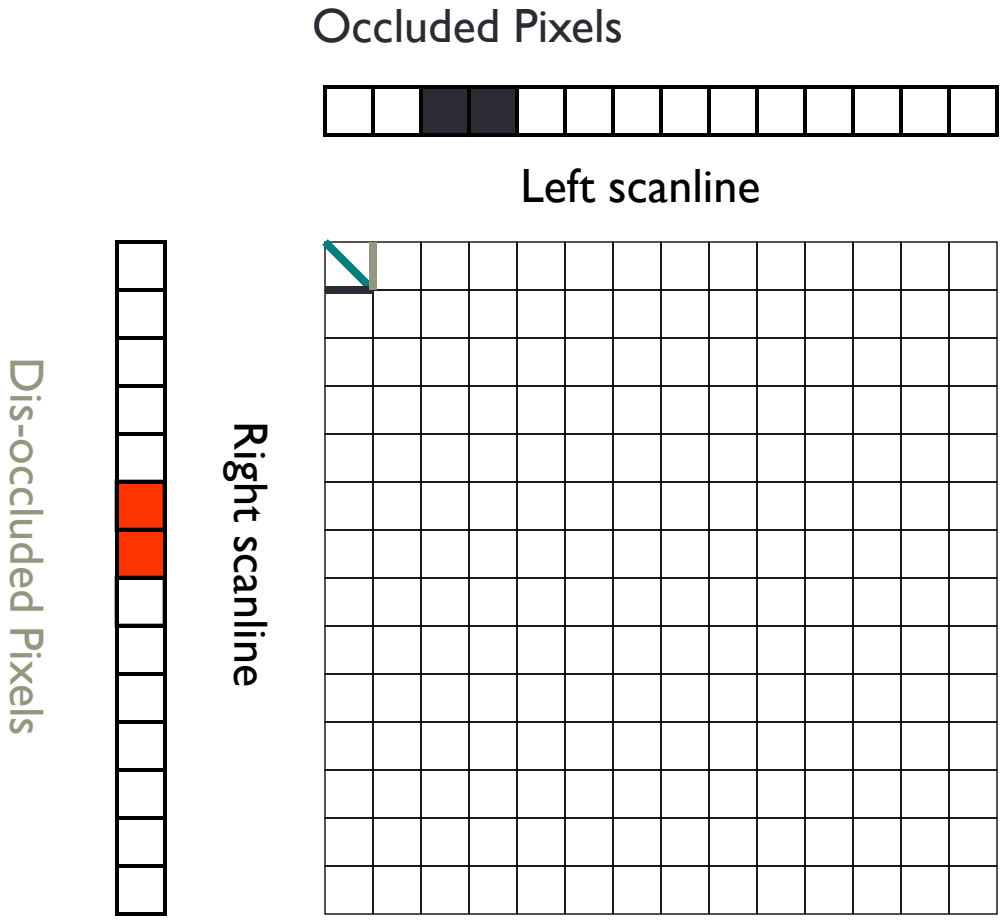
Search Over Correspondences



Three cases:

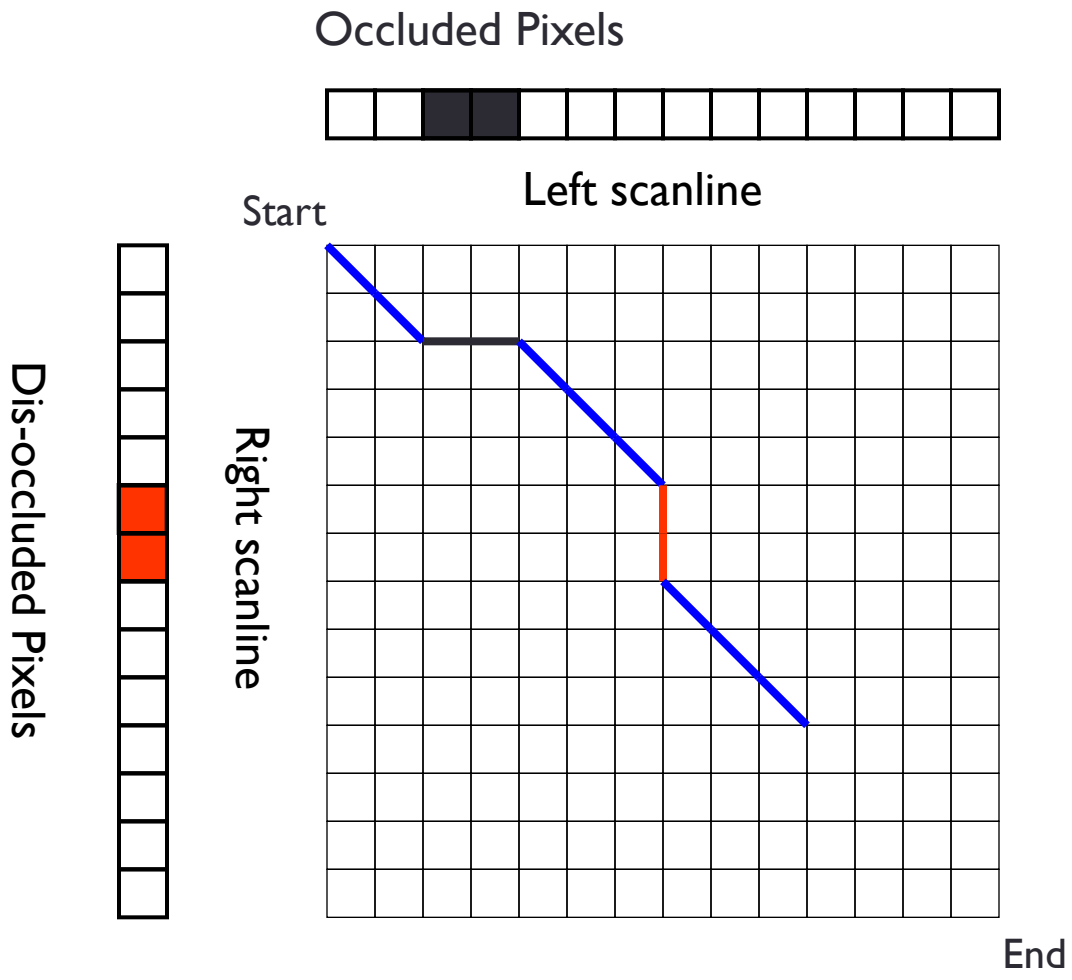
- Sequential – cost of match
- Occluded – cost of no match
- Disoccluded – cost of no match

Stereo Matching with Dynamic Programming



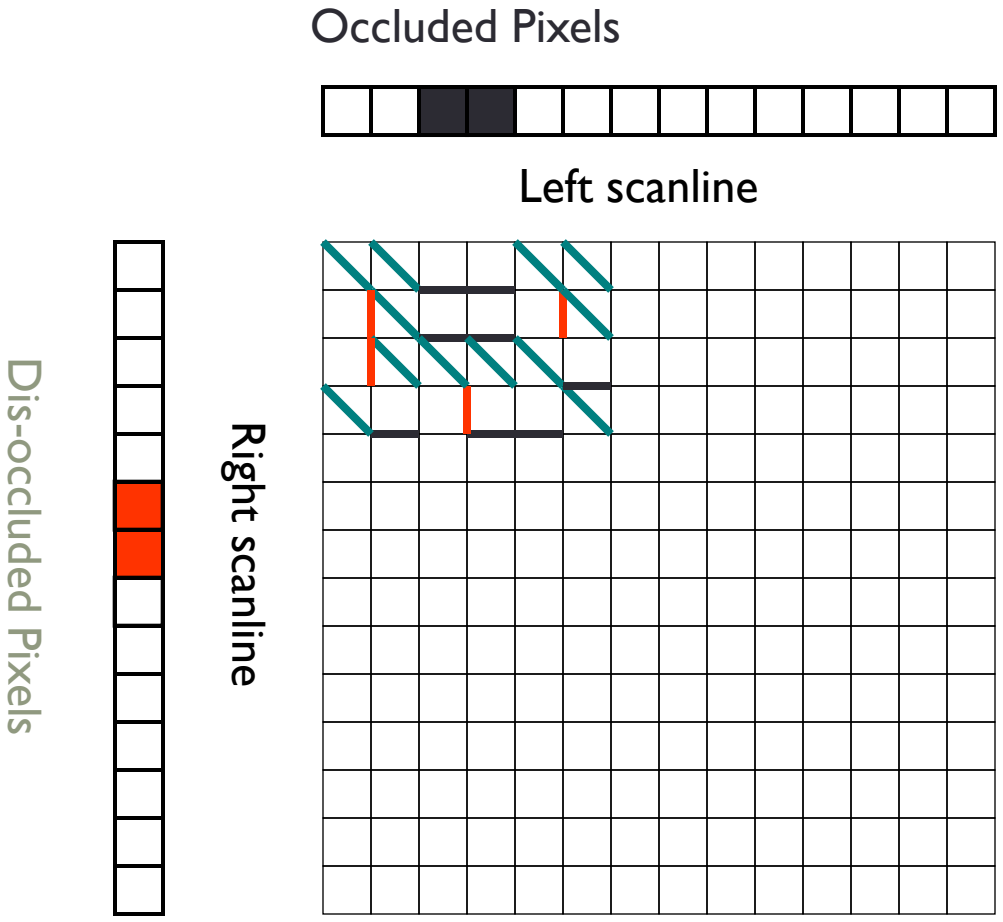
Scan across grid computing optimal cost for each node given its upper-left neighbors. Backtrack from the terminal to get the optimal path.

Stereo Matching with Dynamic Programming



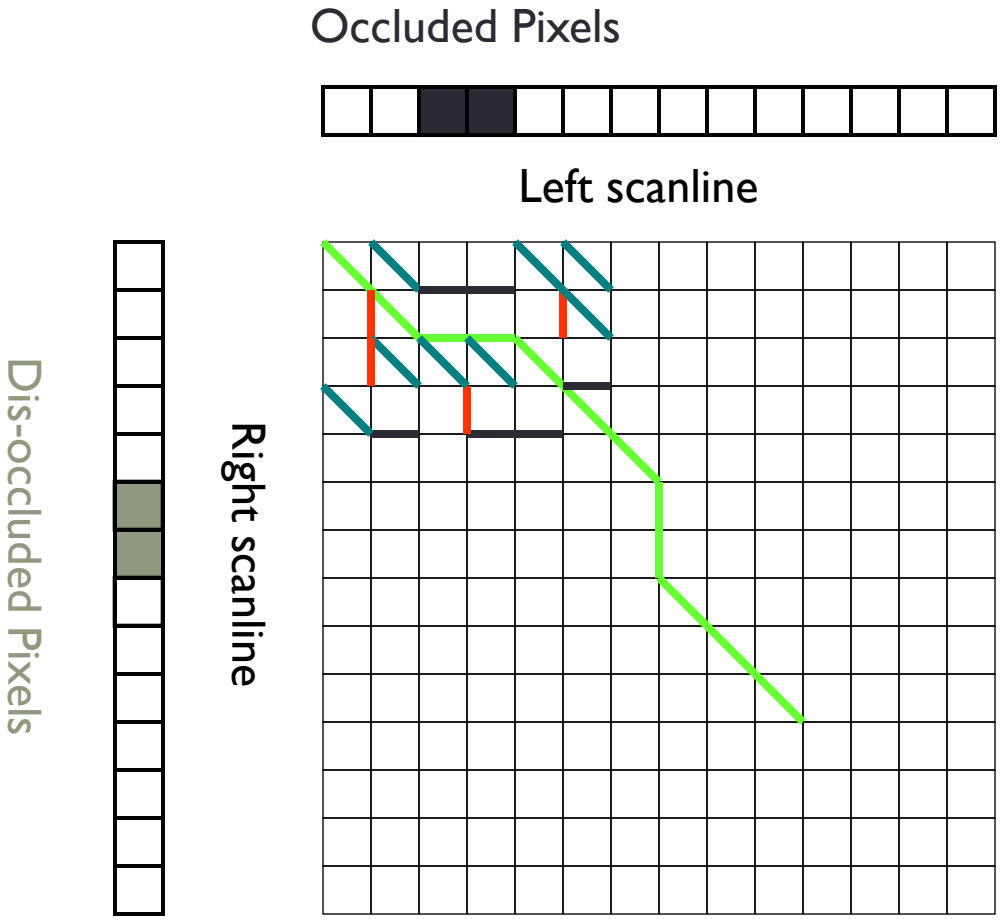
Dynamic programming yields the optimal path through grid. This is the best set of matches that satisfy the ordering constraint

Stereo Matching with Dynamic Programming



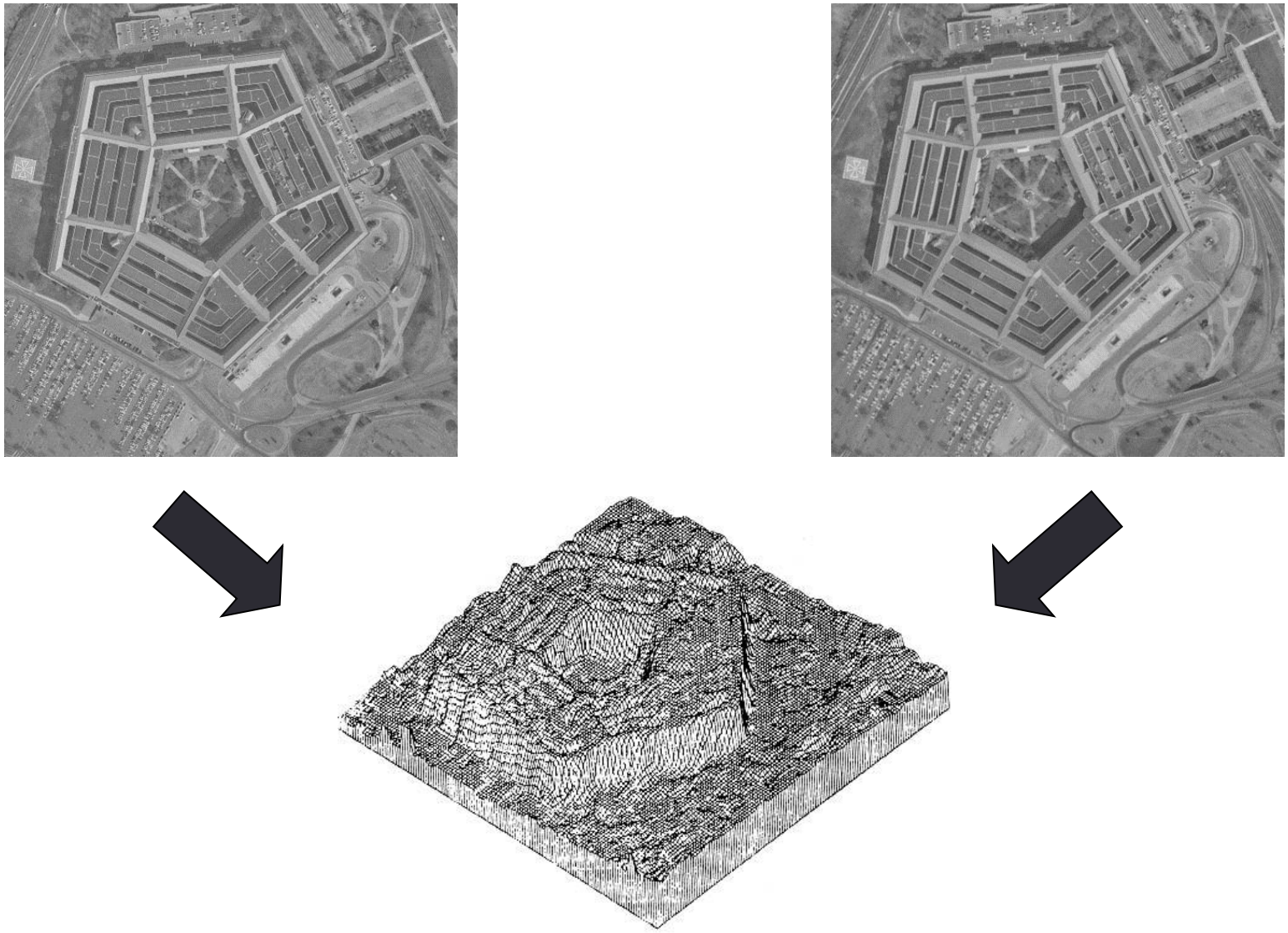
Scan across grid computing optimal cost for each node given its upper-left neighbors. Backtrack from the terminal to get the optimal path.

Stereo Matching with Dynamic Programming



Scan across grid computing optimal cost for each node given its upper-left neighbors. Backtrack from the terminal to get the optimal path.

Dynamic Programming (Ohta and Kanade, 1985)

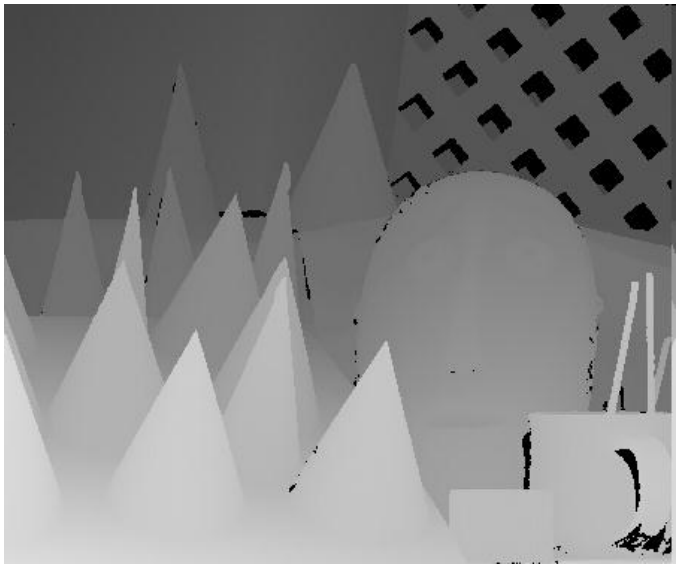


Reprinted from "Stereo by Intra- and Intet-Scanline Search," by Y. Ohta and T. Kanade, IEEE Trans. on Pattern Analysis and Machine Intelligence, 7(2):139-154 (1985). © 1985 IEEE.

Improving correspondence: Non-local constraints

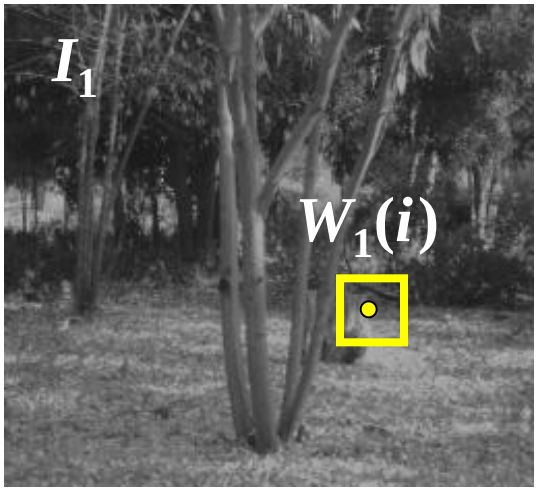
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views
- Smoothness
  - Disparity is typically a smooth function of x (expect in occluding boundaries)

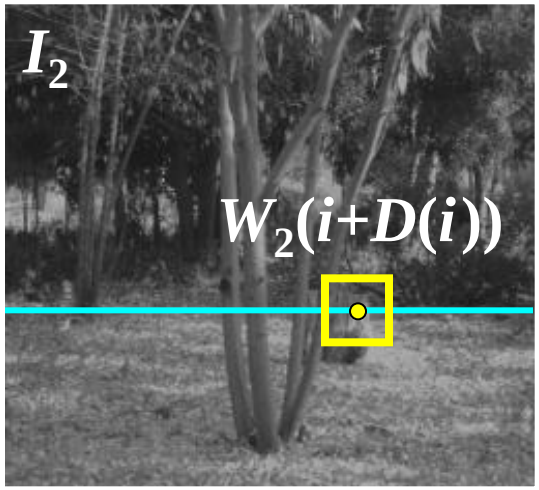
Smoothness

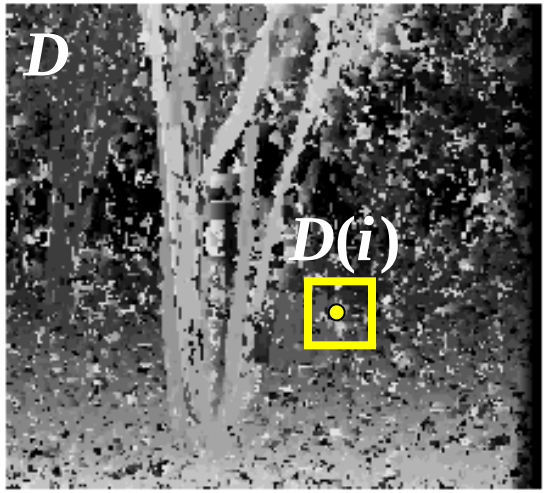


Stereo matching as energy minimization

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 01







$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

$$E_{\text{data}} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2$$

$$E_{\text{smooth}} = \sum_{\text{neighbors}\, i,j} \rho\left(D(i) - D(j)\right)$$

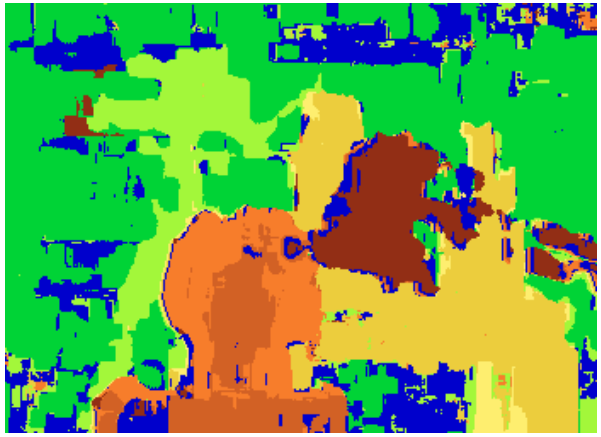
- Energy functions of this form can be minimized using *graph cuts*

Credit slide S. Lazebnik

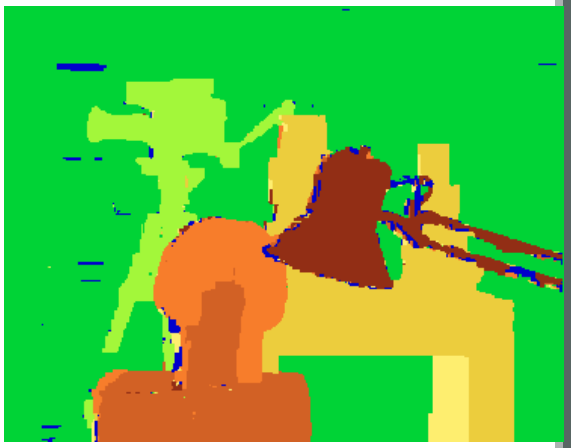
Stereo matching as energy minimization

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 01

Ground truth



Window-based matching



Graph cuts

Two-frame stereo correspondence algorithms

StereoMatcher

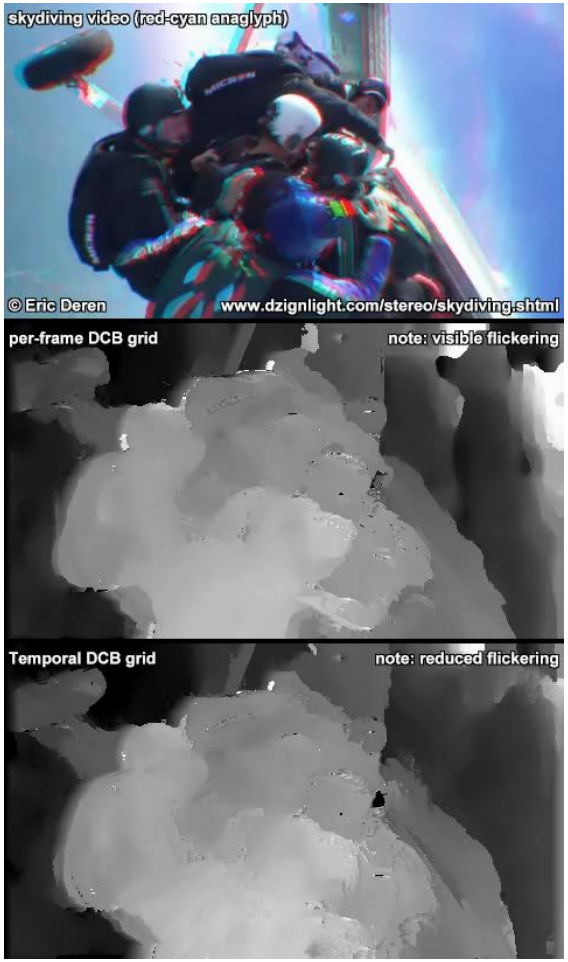
StereoMatcher is an implementation of some commonly used two-frame stereo matching algorithms. It also contains code to evaluate the quality of a computed depth map relative to a ground truth image.

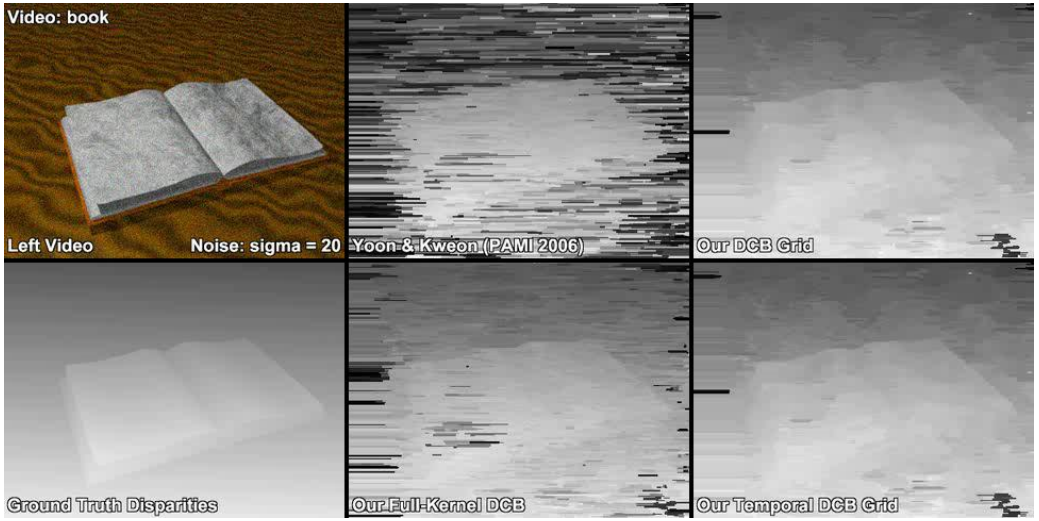
http://research.microsoft.com/en-us/downloads/9bc7fd74-5953-4064-9732-76405573aaef/default.aspx

http://vision.middlebury.edu/stereo/

Real-time Spatiotemporal Stereo Matching Using the Dual-Cross-Bilateral Grid

Christian Richardt, Douglas Orr, Ian Davies, Antonio Criminisi and Neil A. Dodgson





http://www.cl.cam.ac.uk/research/rainbow/projects/dcbgrid/

Application – 3D Urban Scene Modeling

http://www.cs.unc.edu/Research/urbanscape/,
N. Cornelis, B. Leibe, K. Cornelis, L. Van Gool, IJCV 08.

http://www.cs.unc.edu/Research/urbanscape/

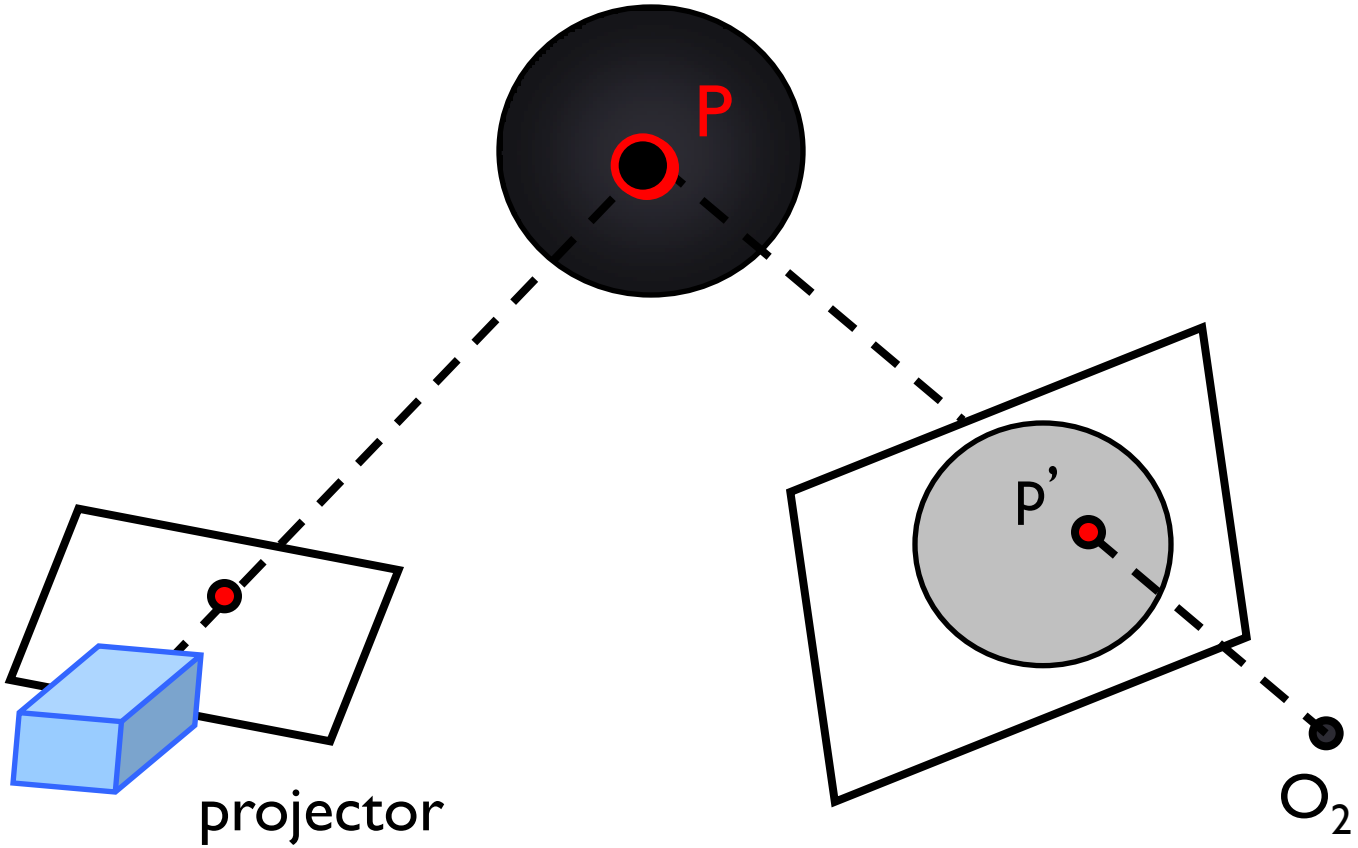
http://www.vision.ee.ethz.ch/showroom/index.en.html#

Outline

- Stereo Systems

- Stereo vision
- Rectification
- Correspondence problem
- Active stereo vision systems

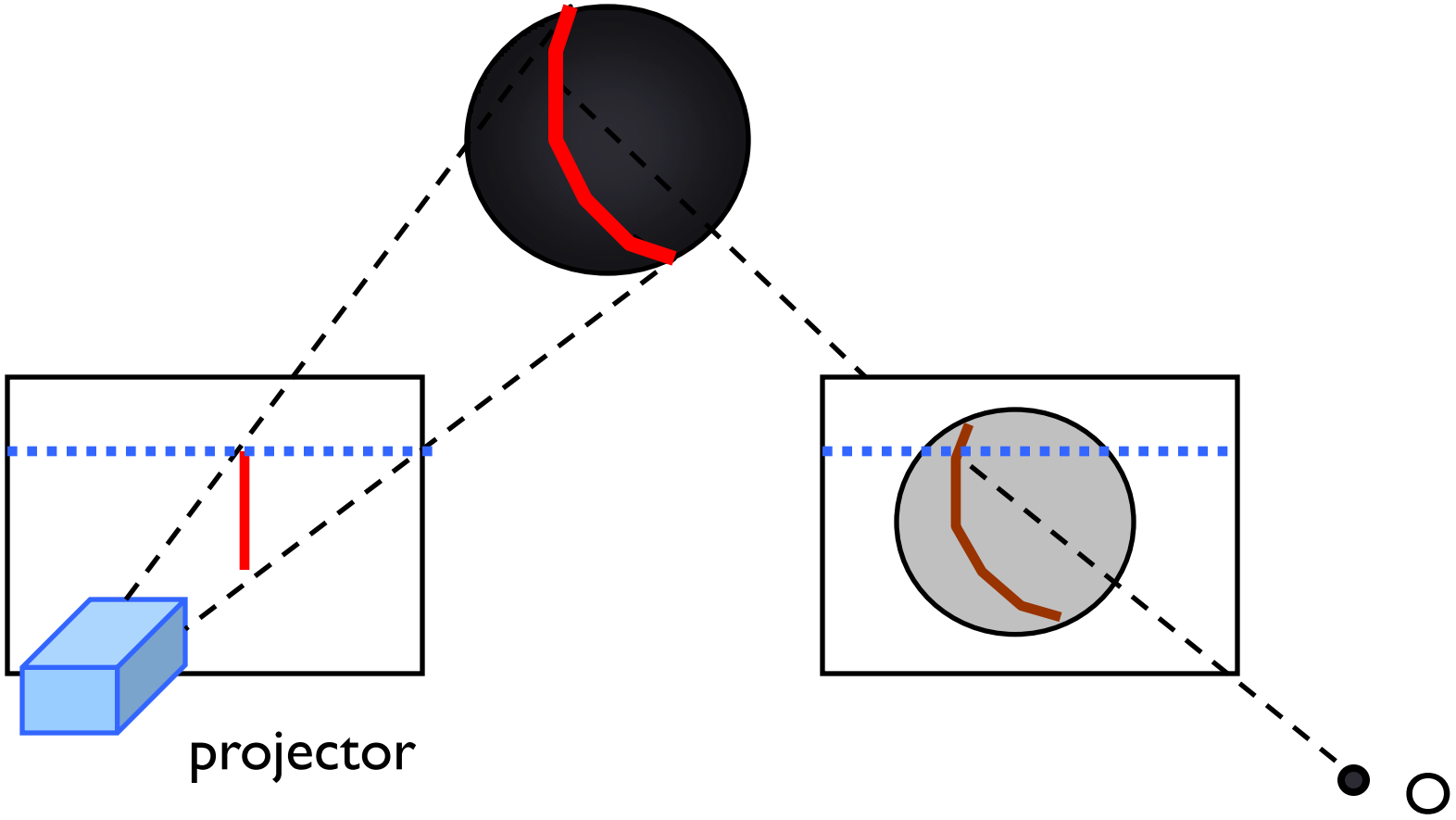
Active stereo (point)



Replace one of the two cameras by a projector

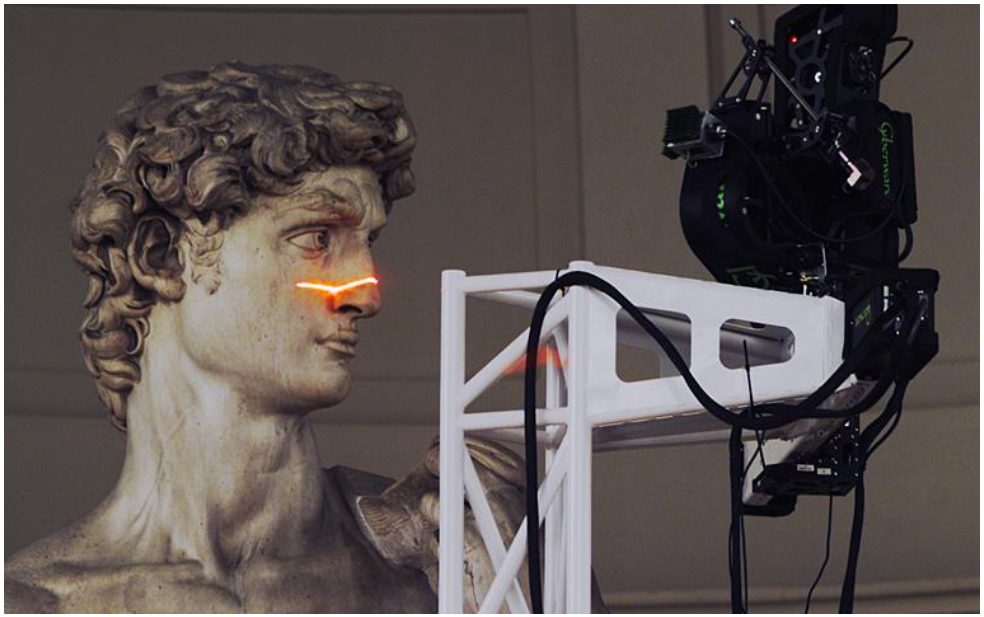
- Single camera
- Projector geometry calibrated
- Correspondence problem solved!

Active stereo (stripe)



-Projector and camera are parallel
- Correspondence problem solved!

Laser scanning



Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Source: S. Seitz

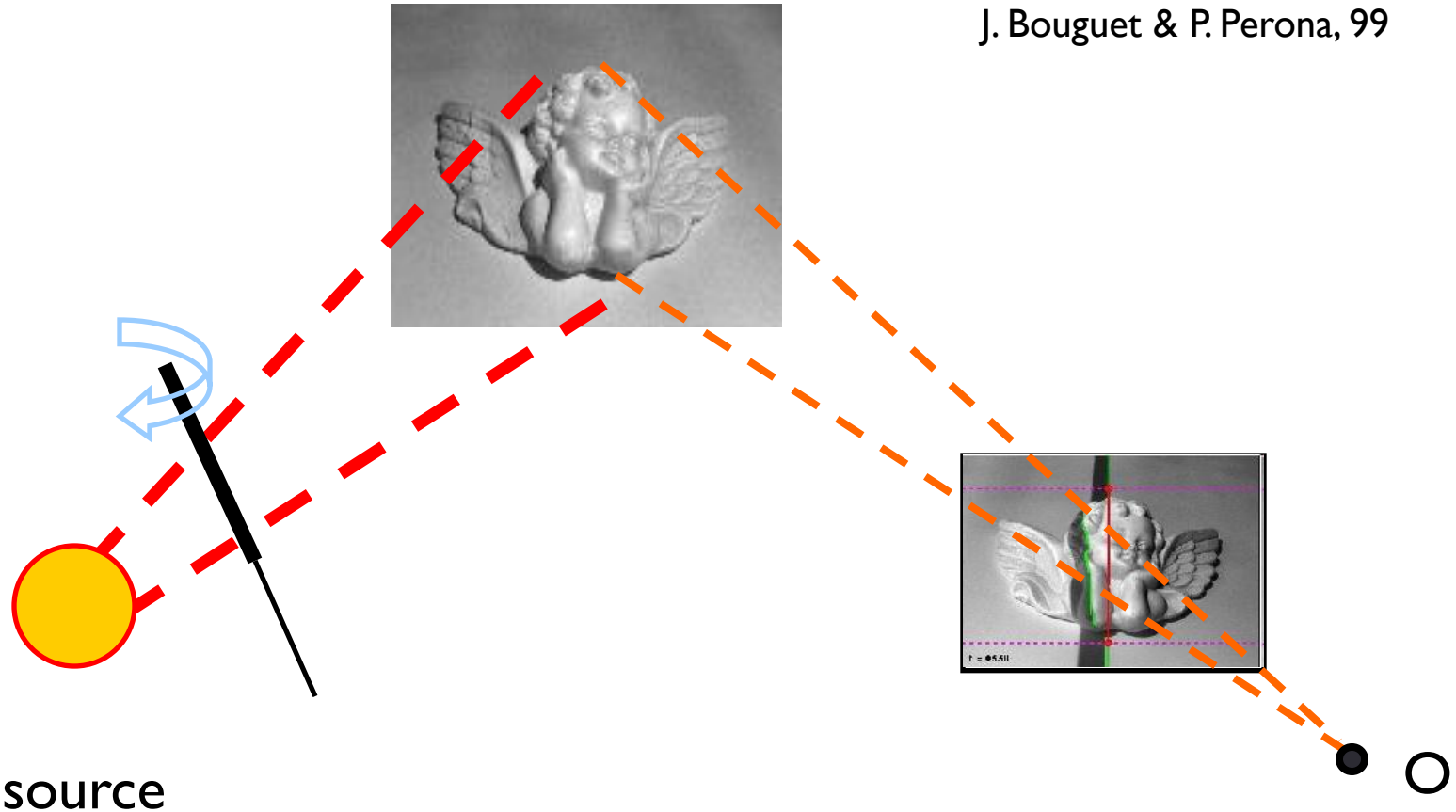
Laser scanning

The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

Active stereo (shadows)

J. Bouguet & P. Perona, 99



Light source

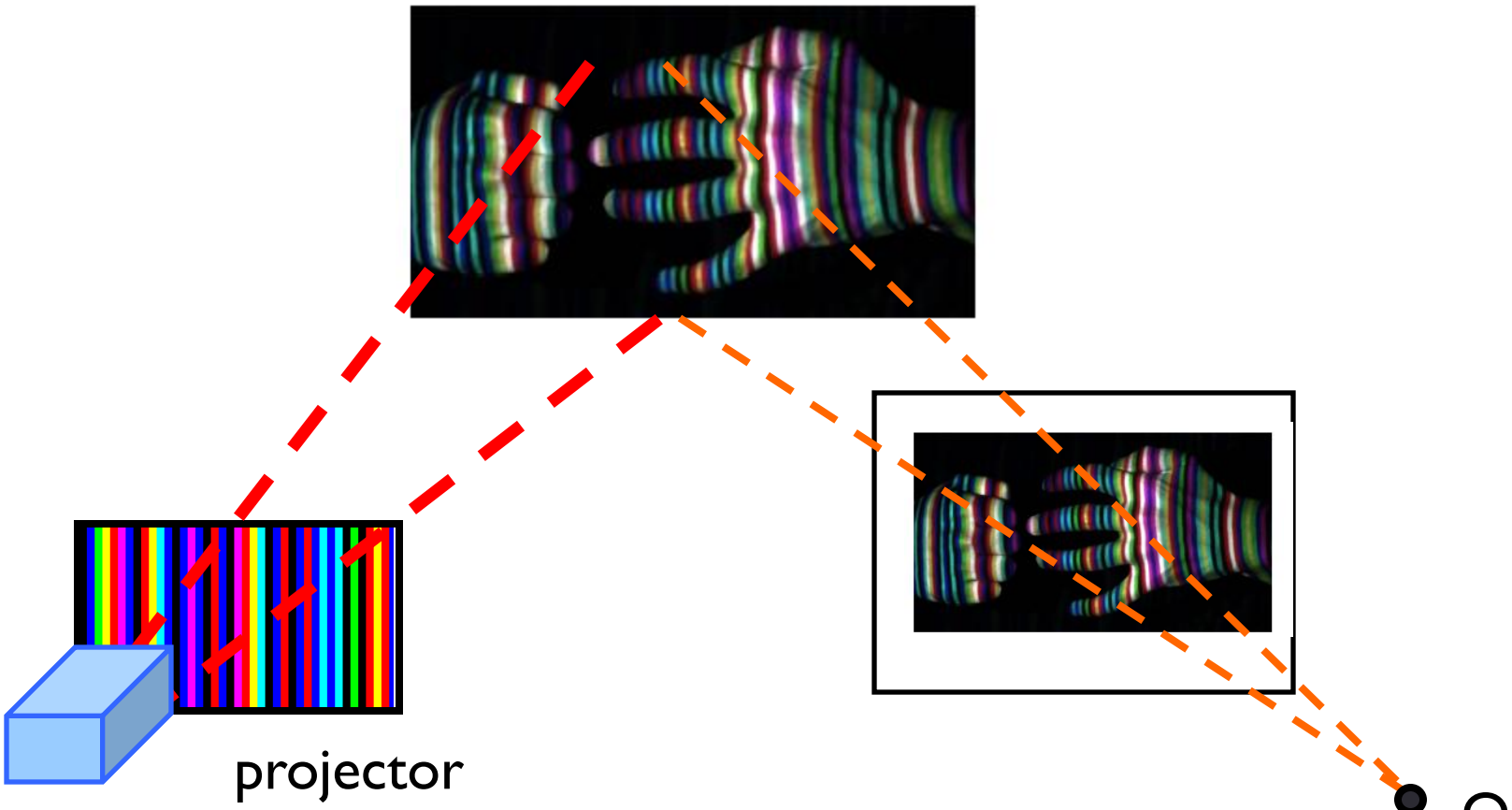
- 1 camera, 1 light source
- very cheap setup
- calibrated the light source

Source: S. Seitz

Active stereo (shadows)

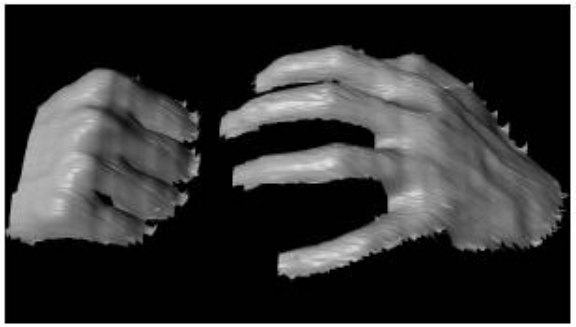
Click here

Active stereo (dense)



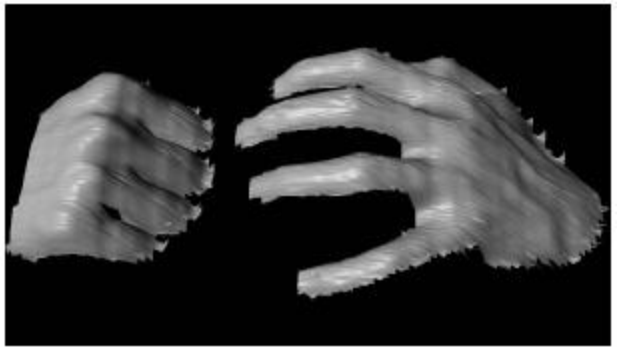
- Dense reconstruction
- Correspondence problem again
- Get around it by using color codes





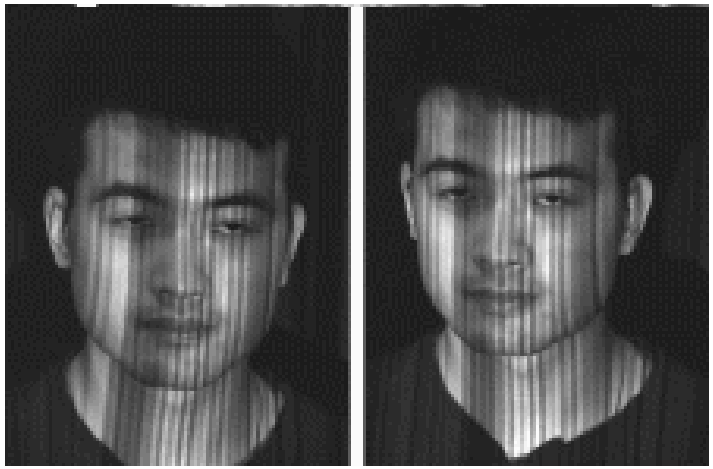
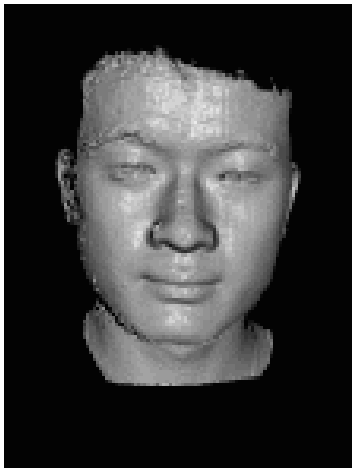
L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002





L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Rapid shape acquisition: Projector + stereo cameras

Next lecture…

Volumetric stereo

CEE598 - Visual Sensing for Civil Infrastructure Eng. & Mgmt.

Session 15b – Volumetric Reconstruction

Mani Golparvar-Fard
Department of Civil and Environmental Engineering
3129D, Newmark Civil Engineering Lab
e-mail: mgolpar@illinois.edu

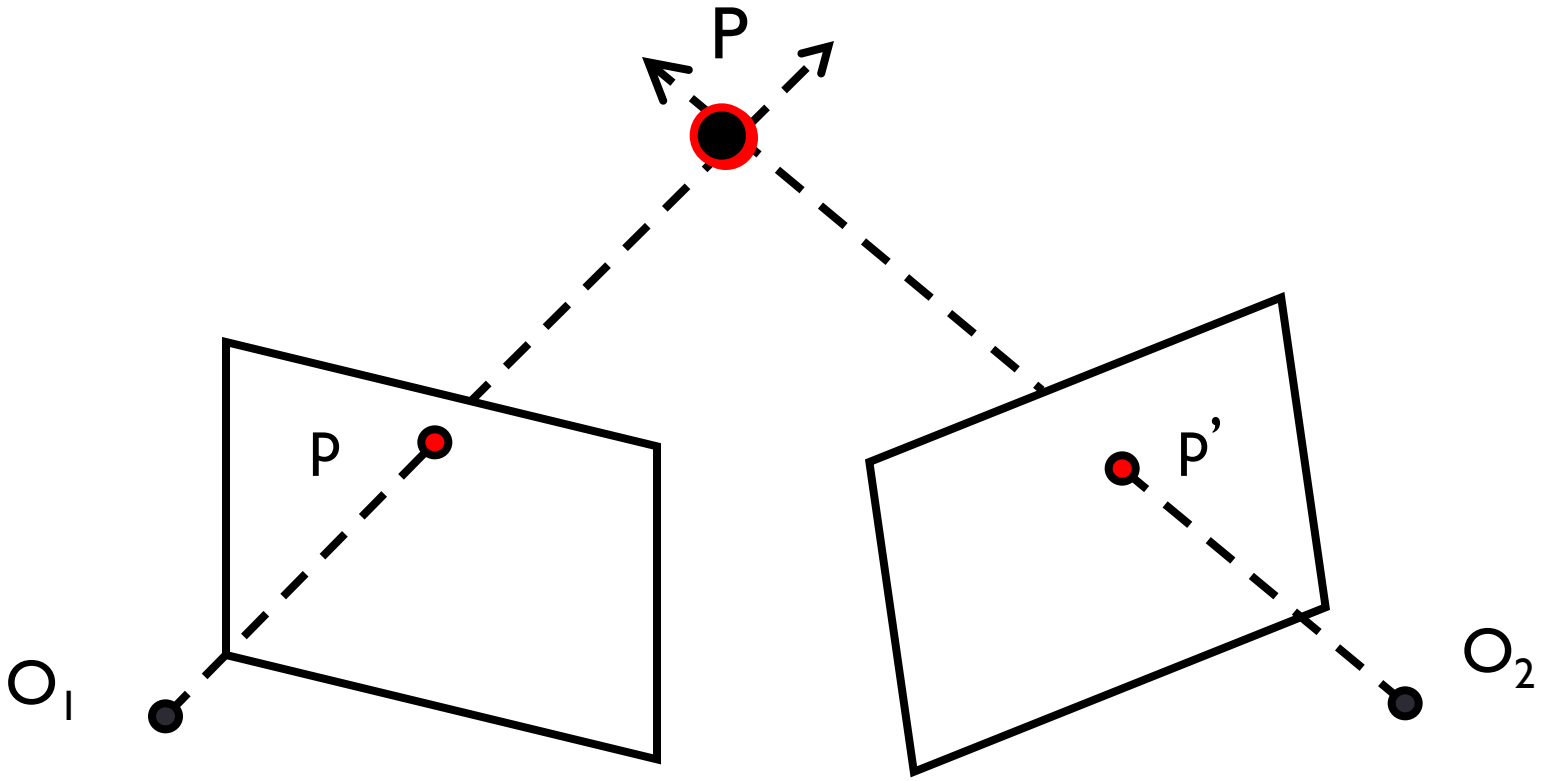
Outline

- Volumetric Reconstruction

- Definition
- Shape from Contours
- Shape from Shadows
- Voxel Coloring

Some slides of this lectures are courtesy of prof. S. Savarese, and various other lecturers

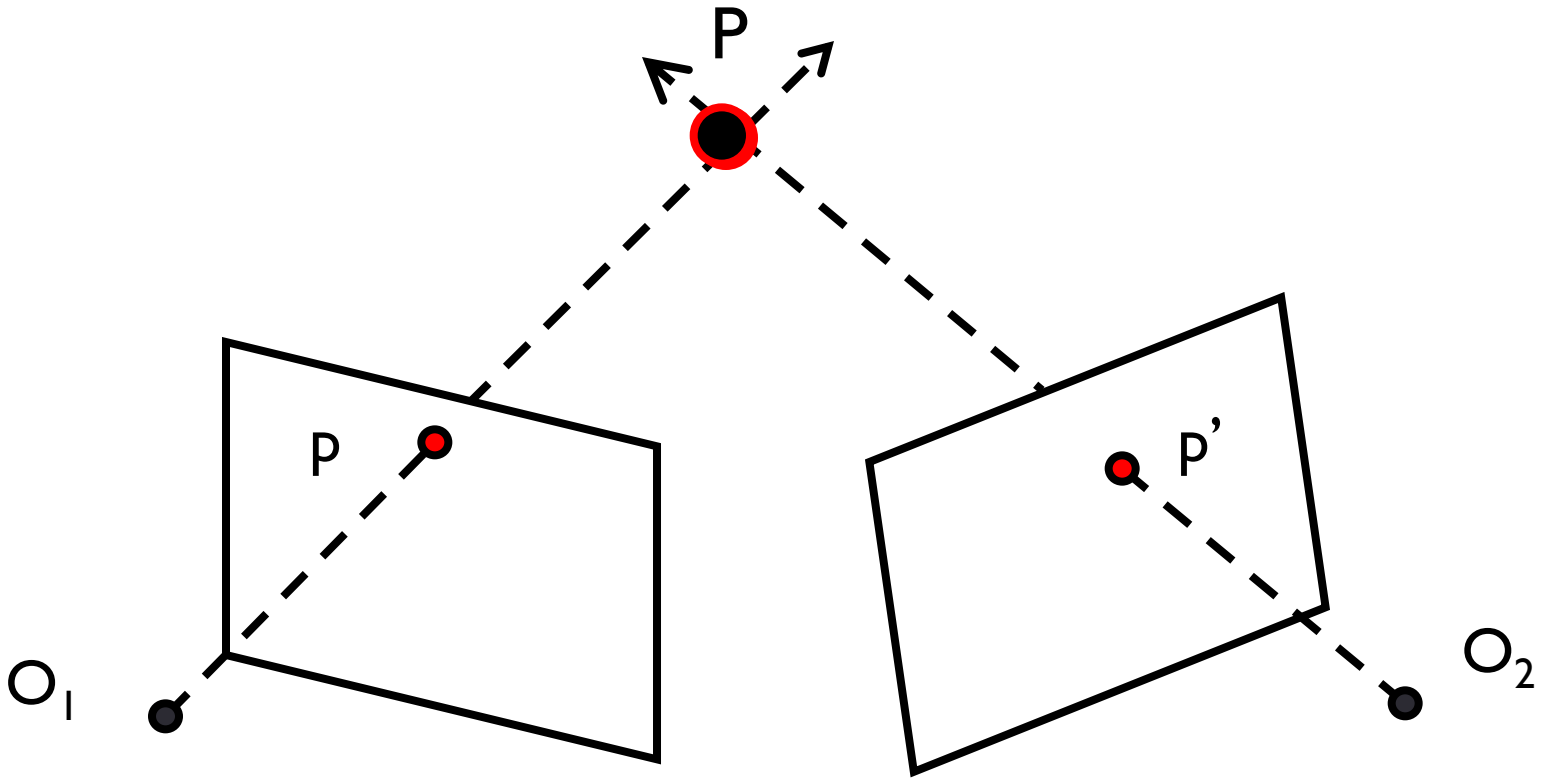
“Traditional” Stereo



Goal: estimate the position of P given the observation of P from two view points

Assumptions: known camera parameters and position (K, R, T)

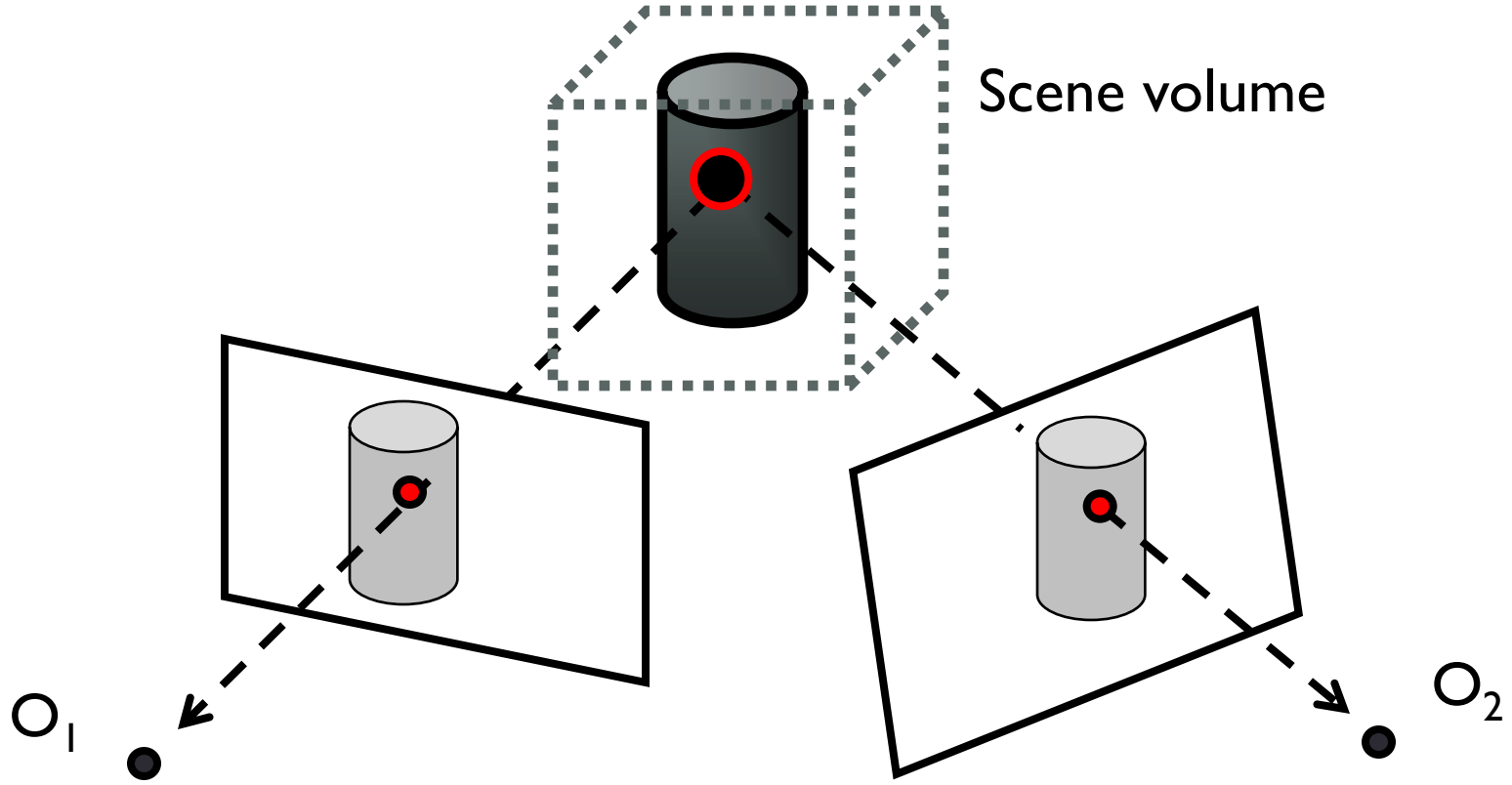
“Traditional” Stereo



Subgoals:

1. Solve the correspondence problem
2. Use corresponding observations to triangulate

Volumetric Reconstruction



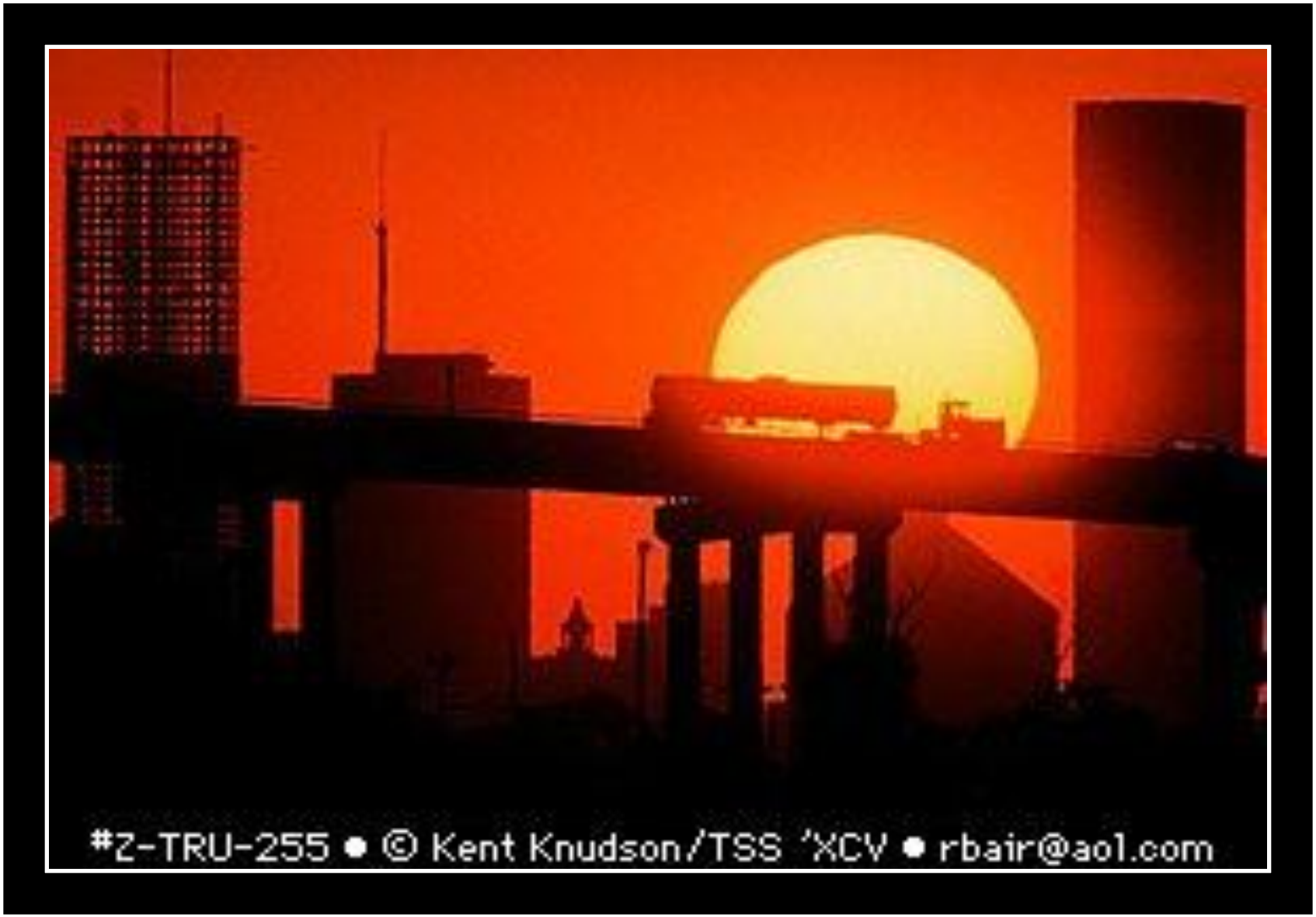
1. Hypothesis: pick up a point within the volume
2. Project this point into 2 (or more) images
3. Validation: are the observations consistent?

Assumptions: known camera parameters and position (K, R, T)

Consistency based on cues such as:

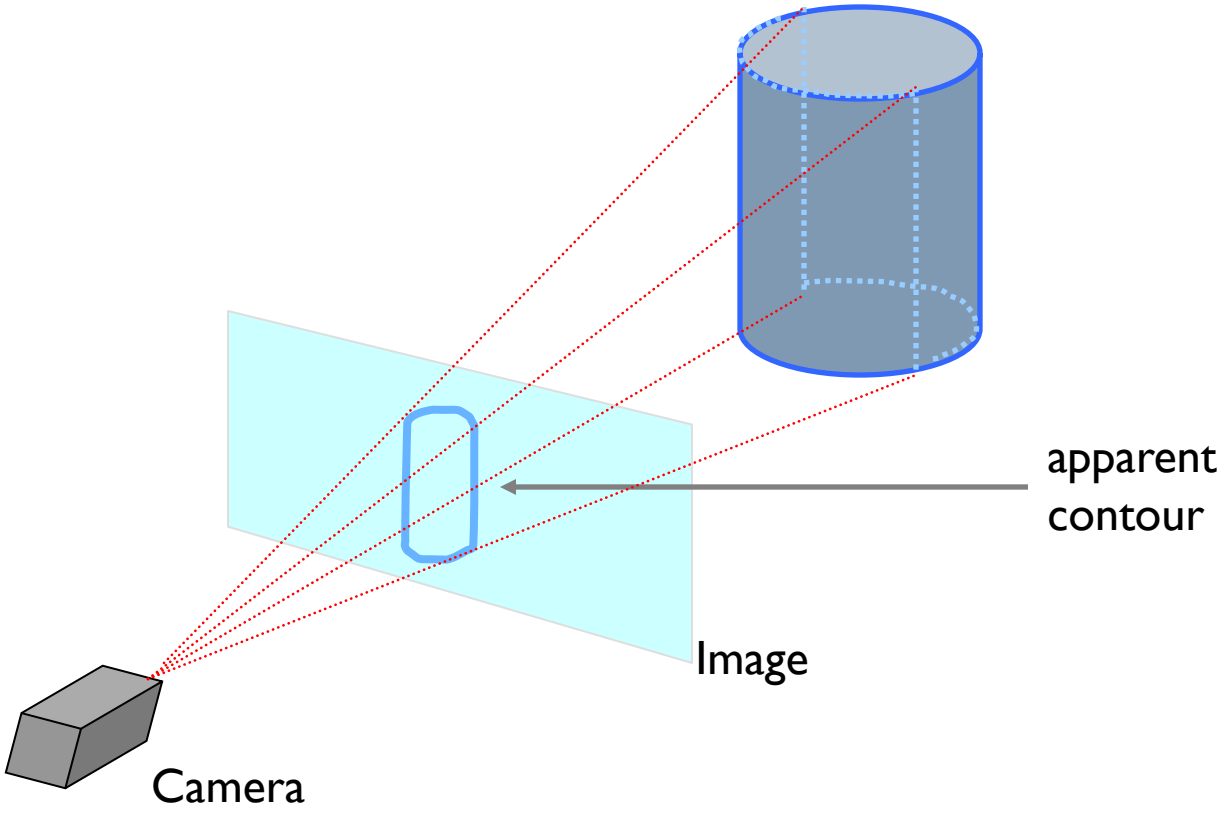
- Contours/silhouettes
- Shadows
- Colors

Contours are a rich source of geometric information

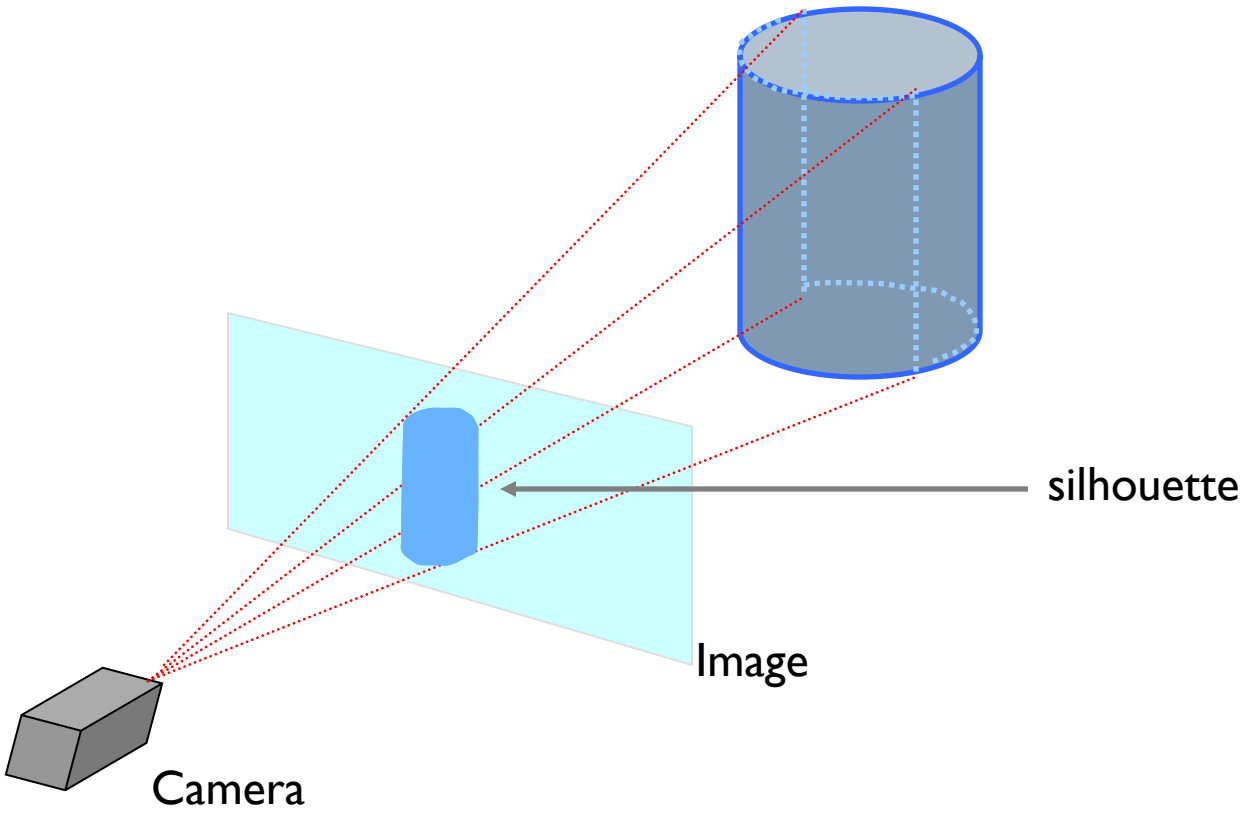


Apparent contour

- DEFINITION: projection of the locus of points on the surface which separate the visible and occluded parts on the surface *[sato & cipolla]*



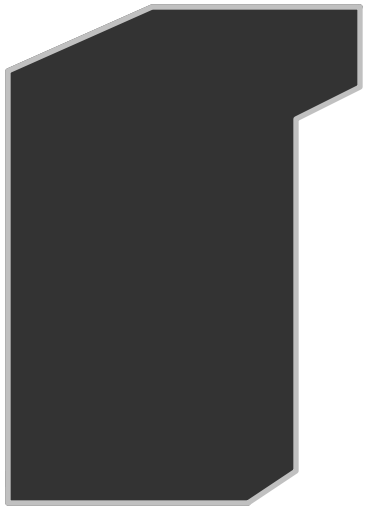
Silhouettes



Why contours are interesting visual cues?

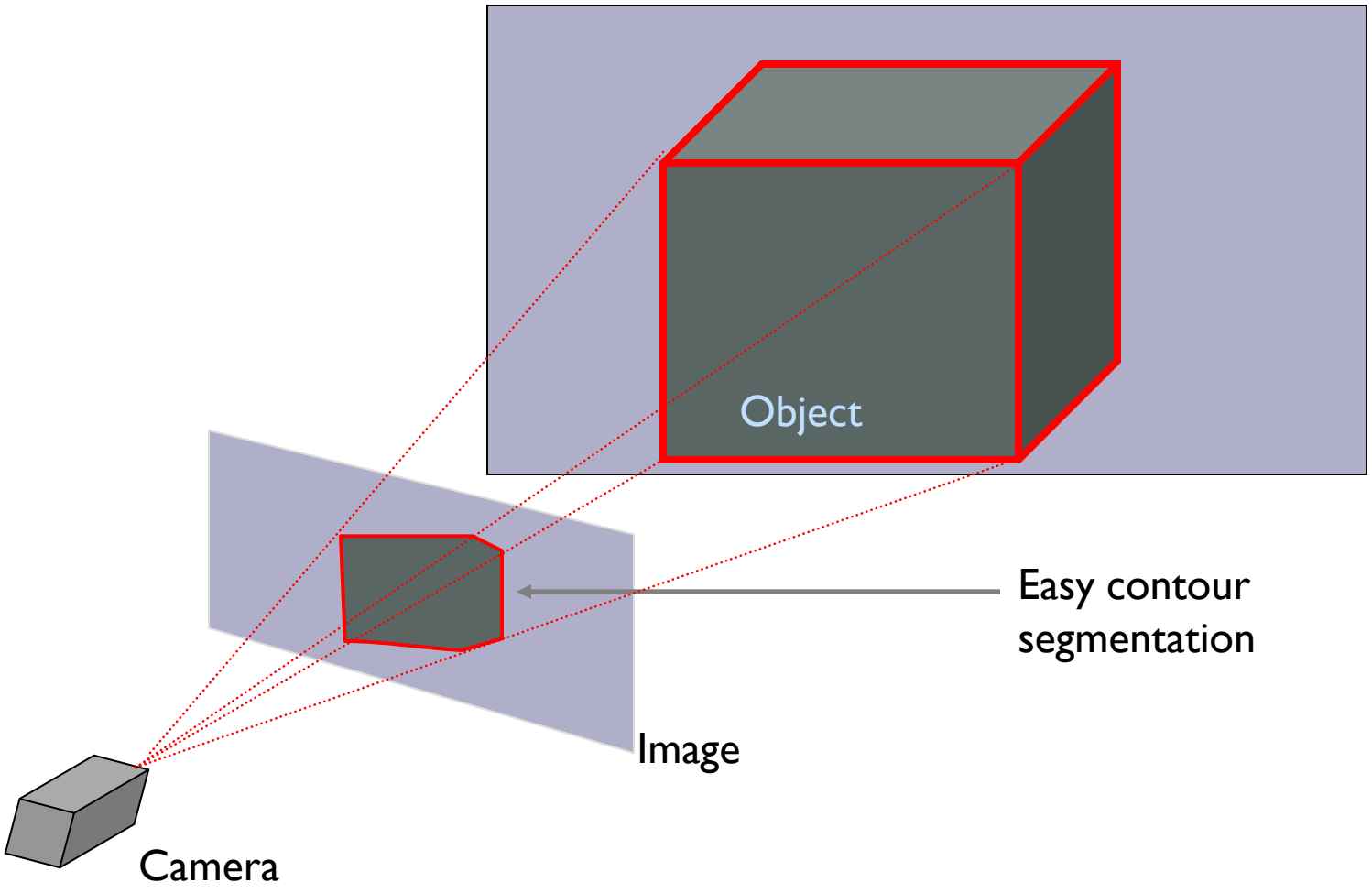
- Provide information in absence of other visual cues

- No texture
- No shading

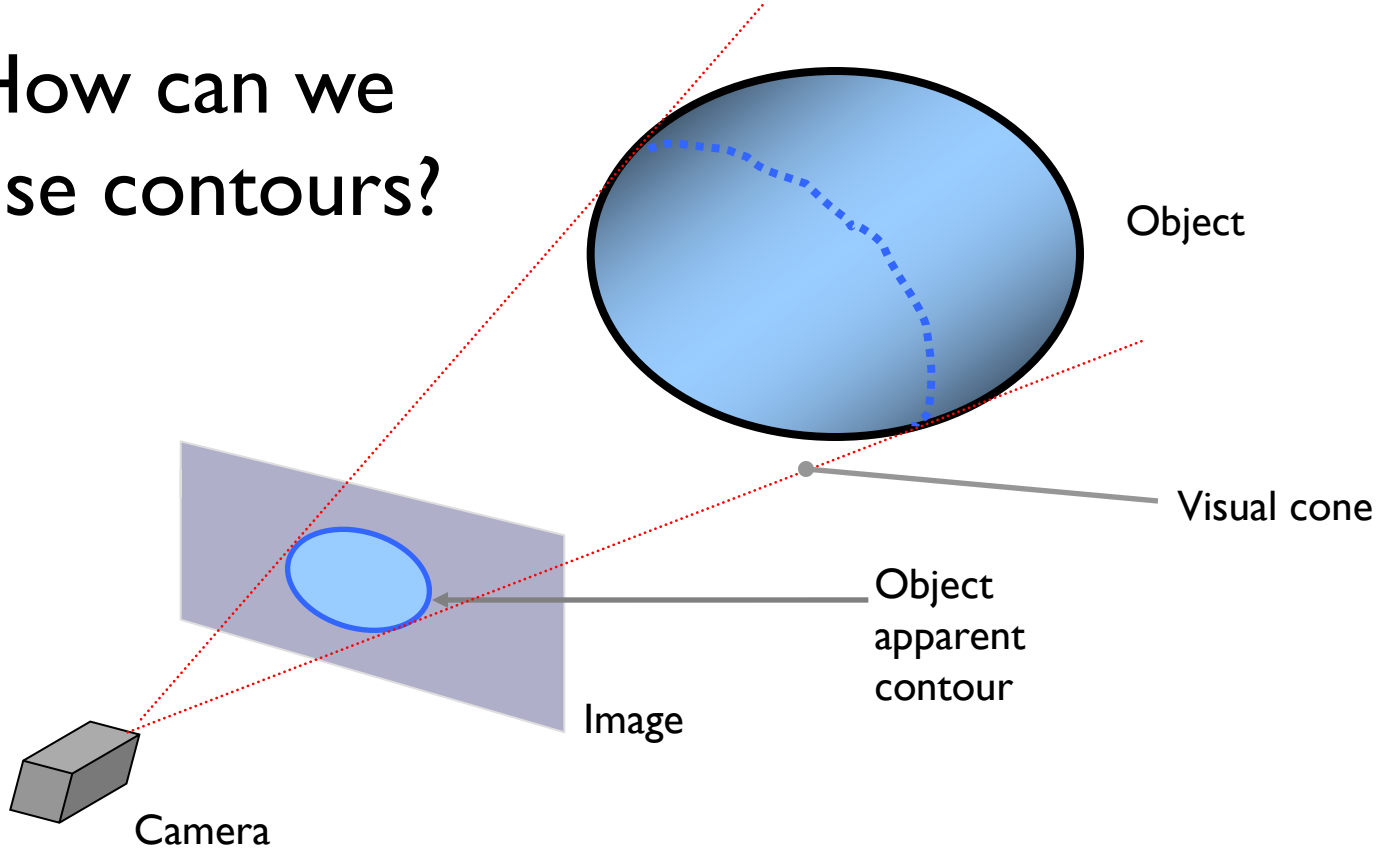


Why contours are interesting visual cues?

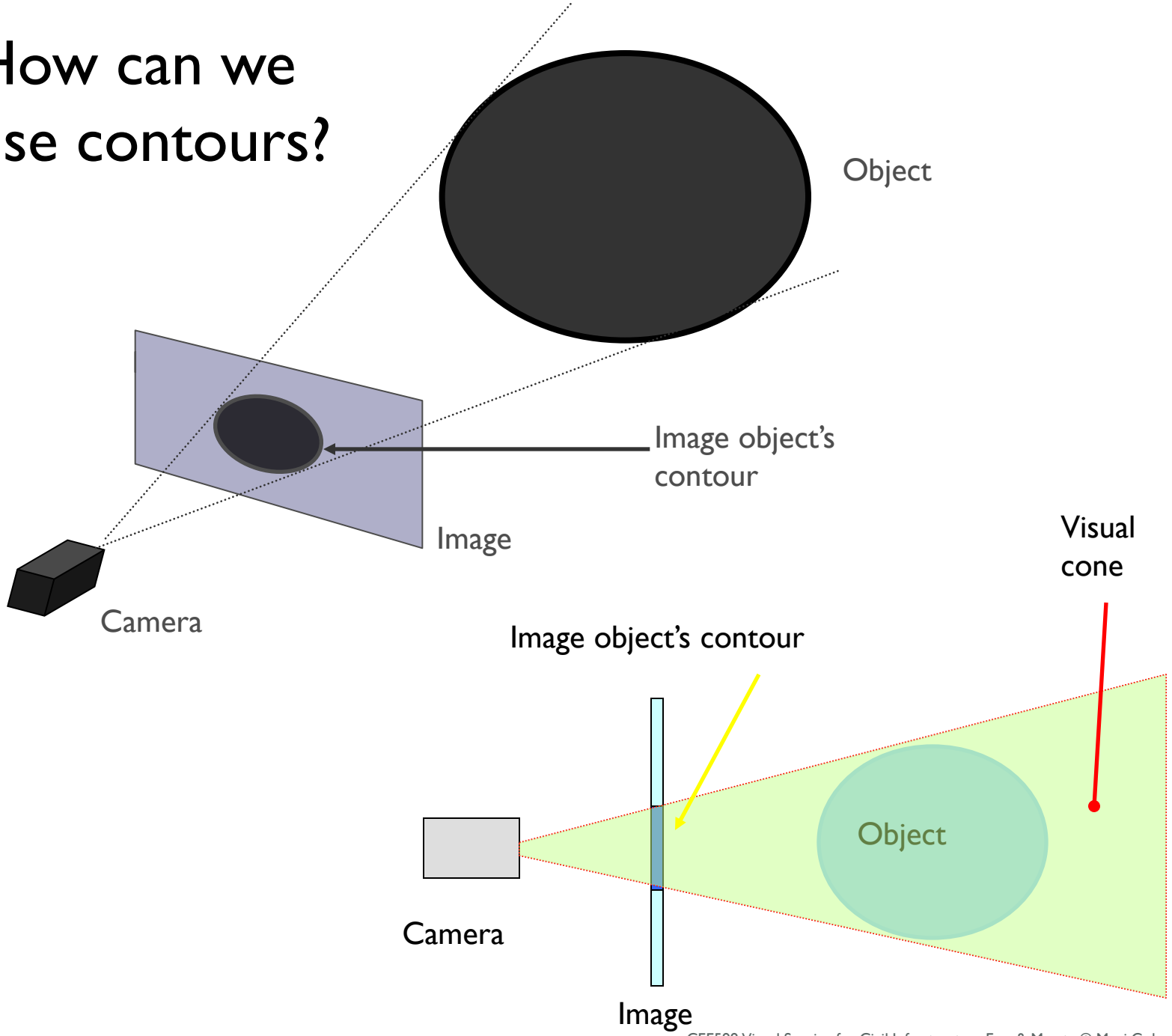
- Relatively easy to detect

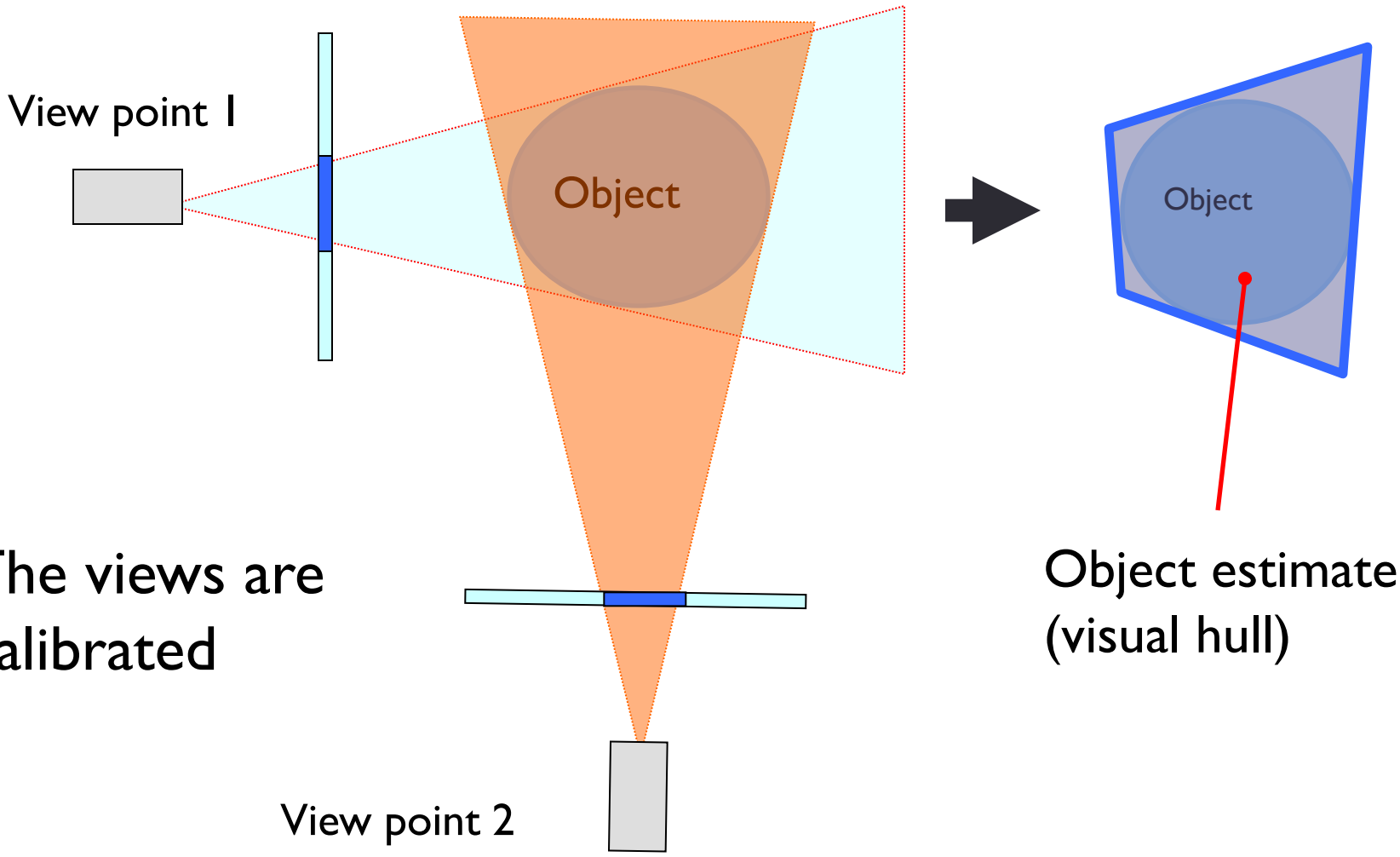


How can we use contours?



How can we use contours?





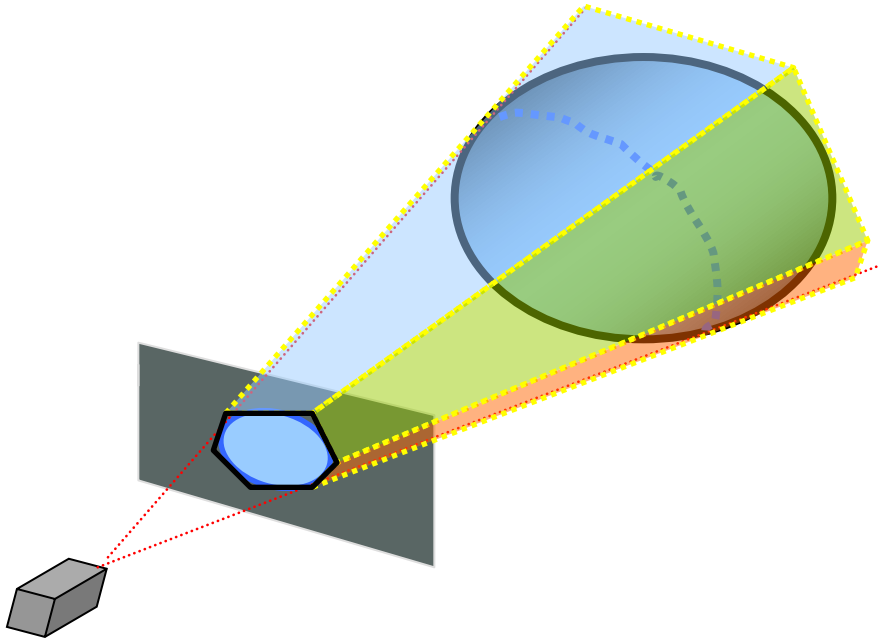
The views are calibrated

Object estimate (visual hull)

How to perform visual cones intersection?

- Decompose visual cone in polygonal surfaces

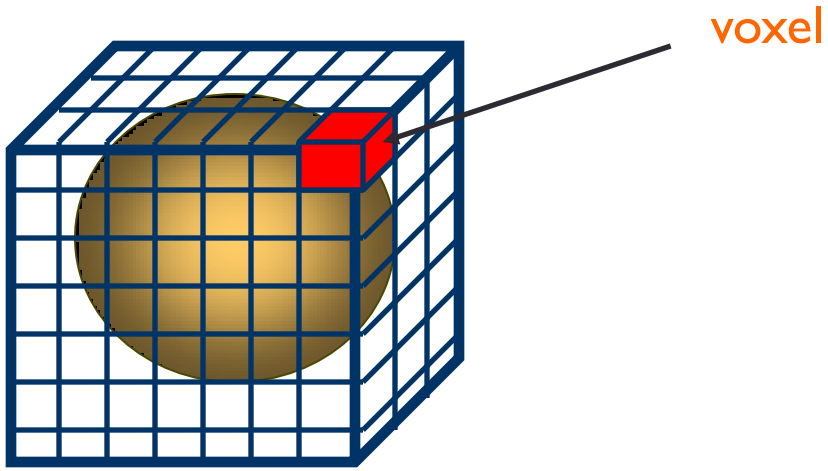
  (among others: Reed and Allen '99)



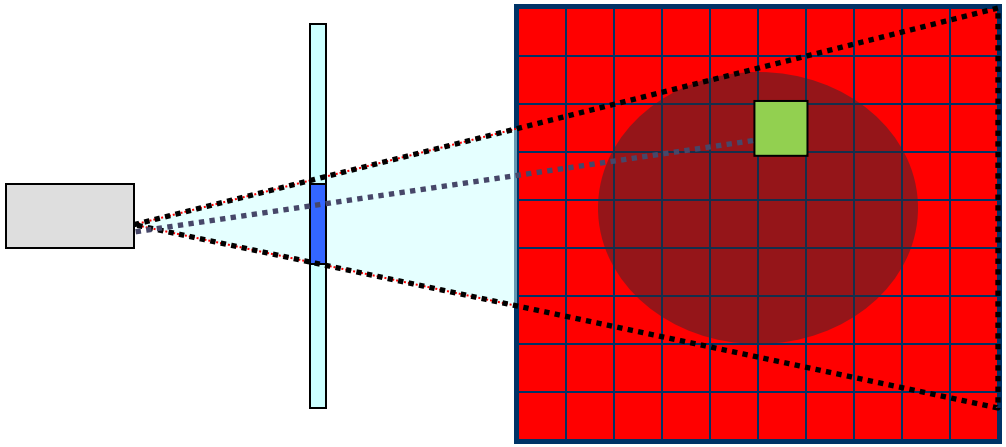
Using contours/silhouettes in volumetric stereo

also called Space carving

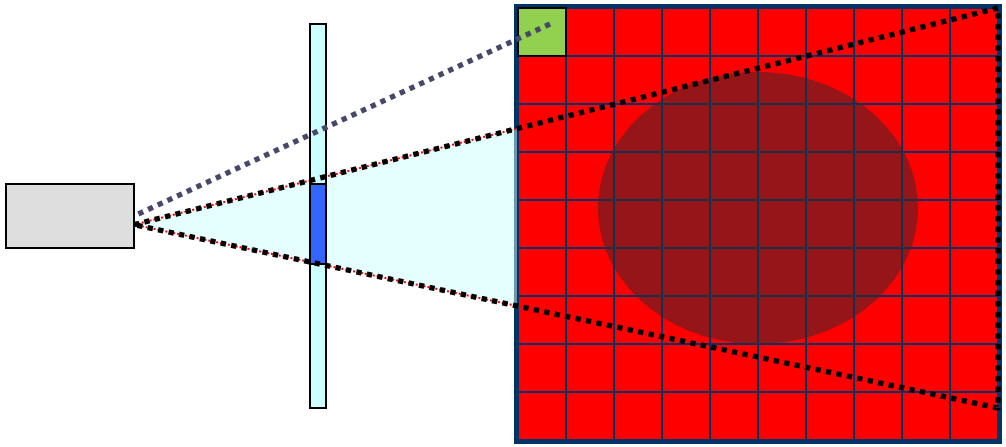
[Martin and Aggarwal (1983)]



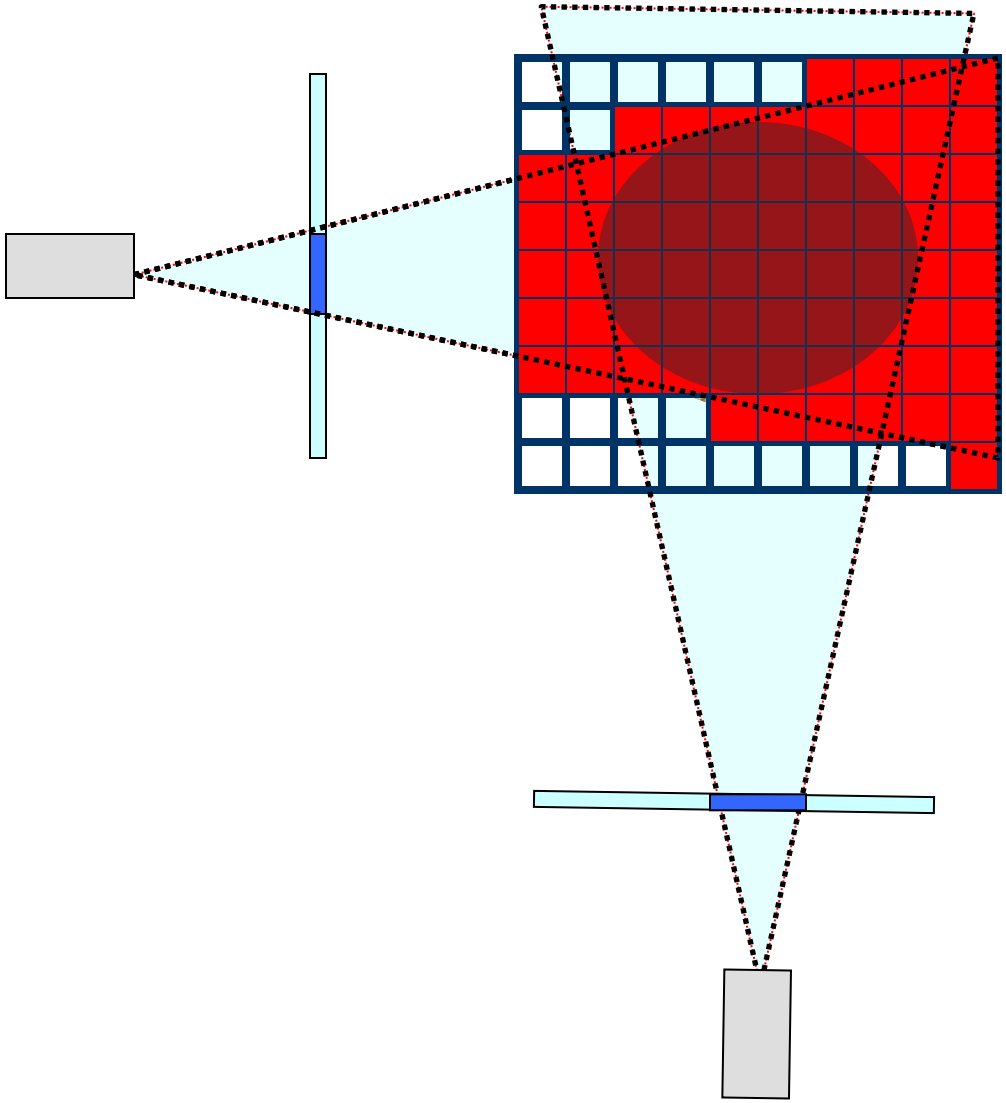
Computing visual hull in 2D



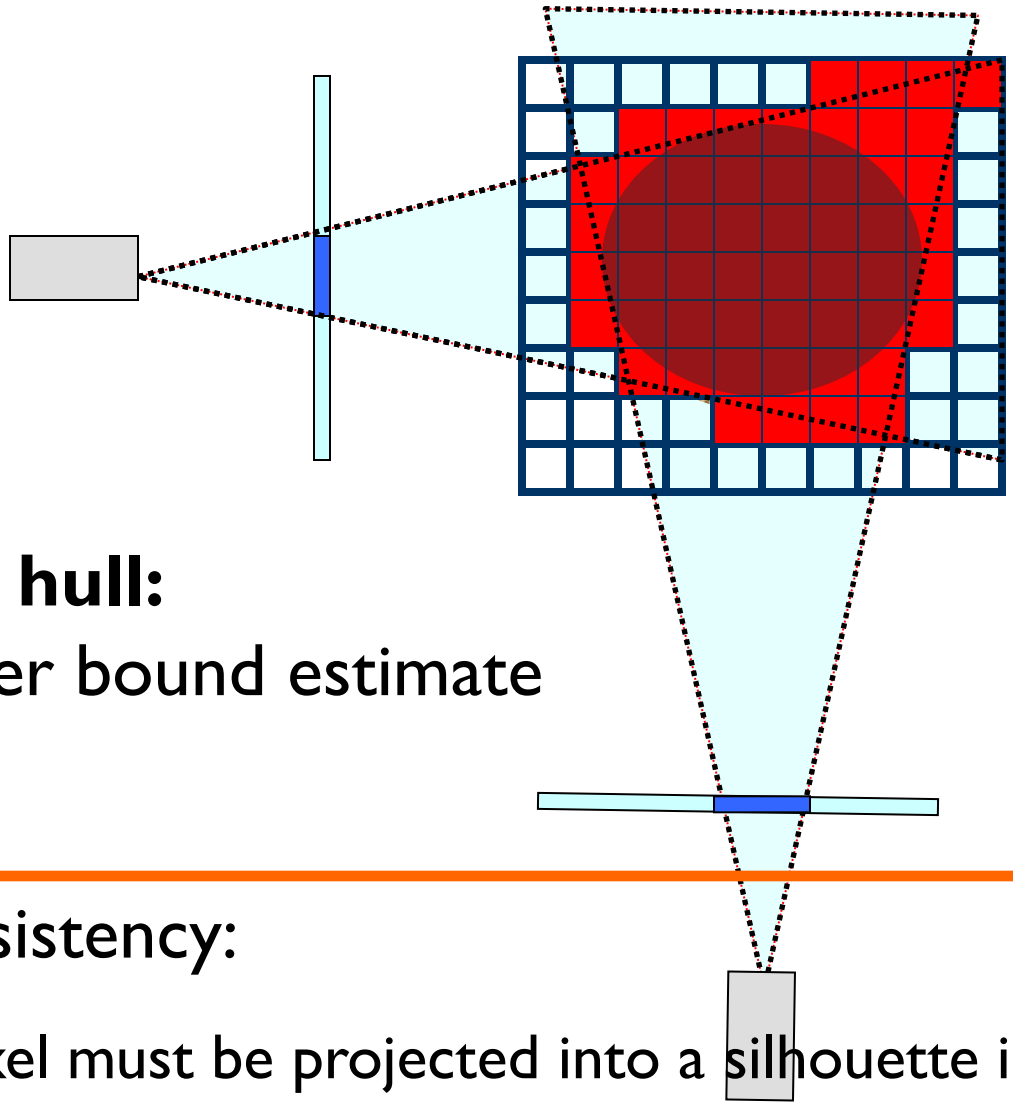
Computing visual hull in 2D



Computing visual hull in 2D



Computing visual hull in 2D



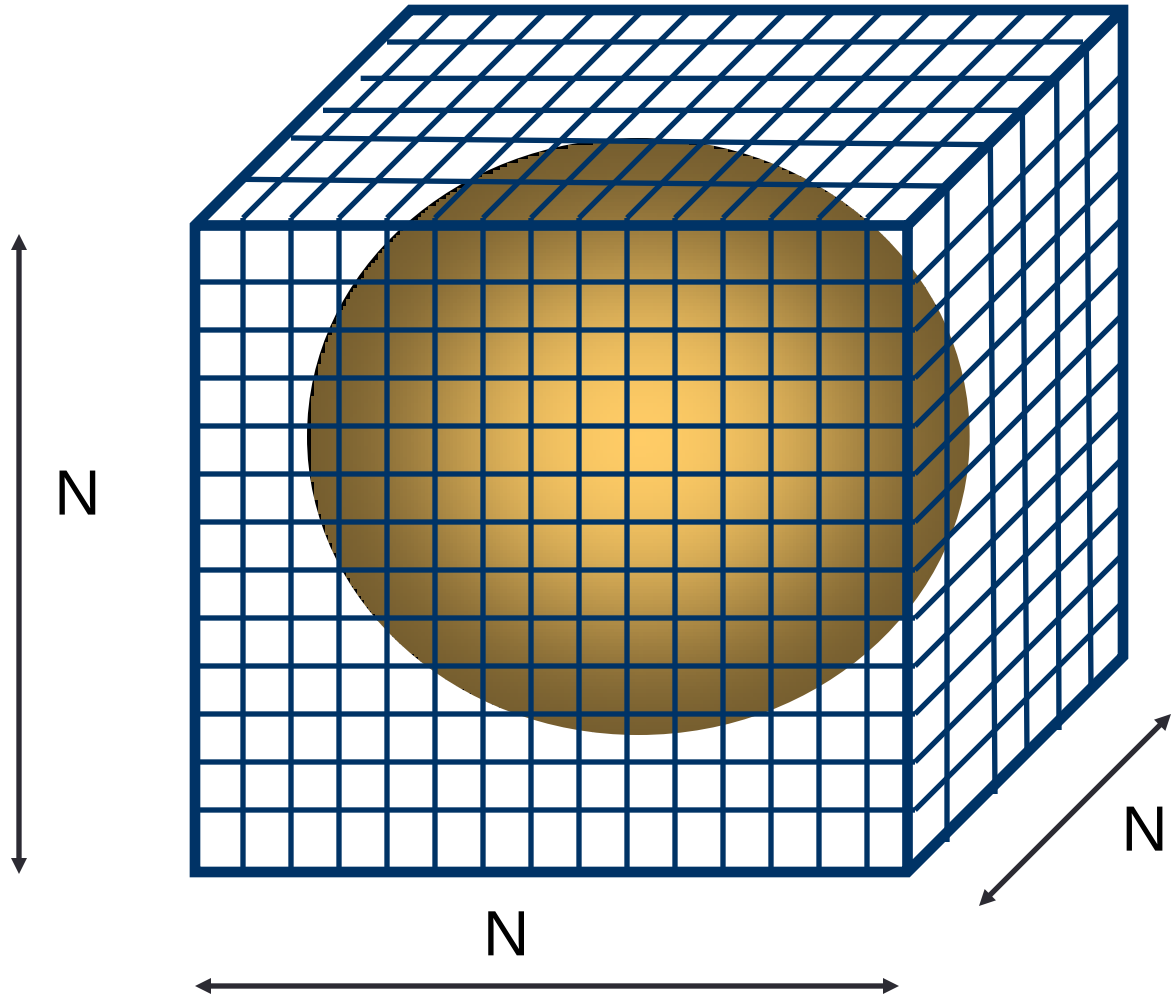
Visual hull:
an upper bound estimate

Consistency:

A voxel must be projected into a silhouette in each image

Space Carving has complexity …

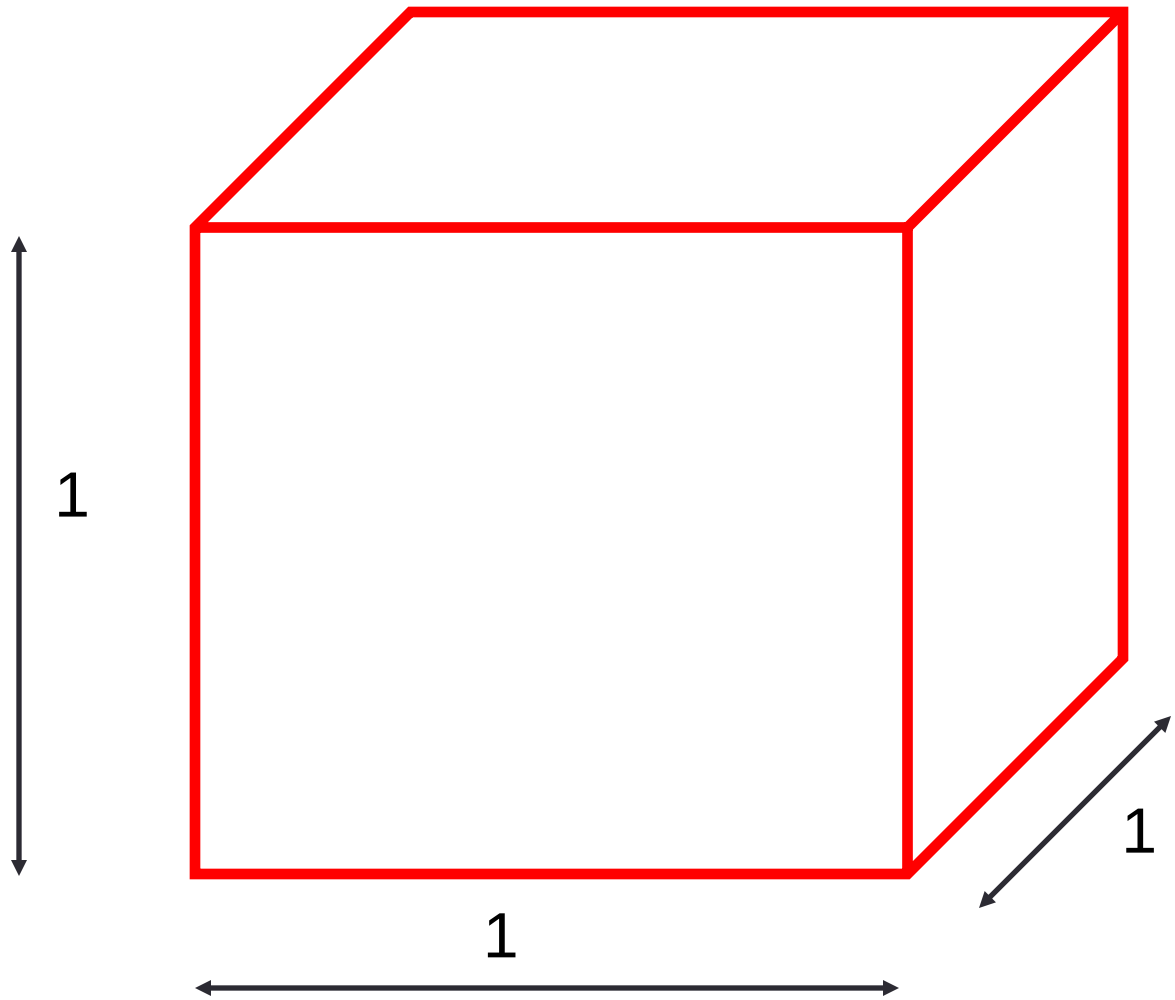
$O(N^3)$



Octrees

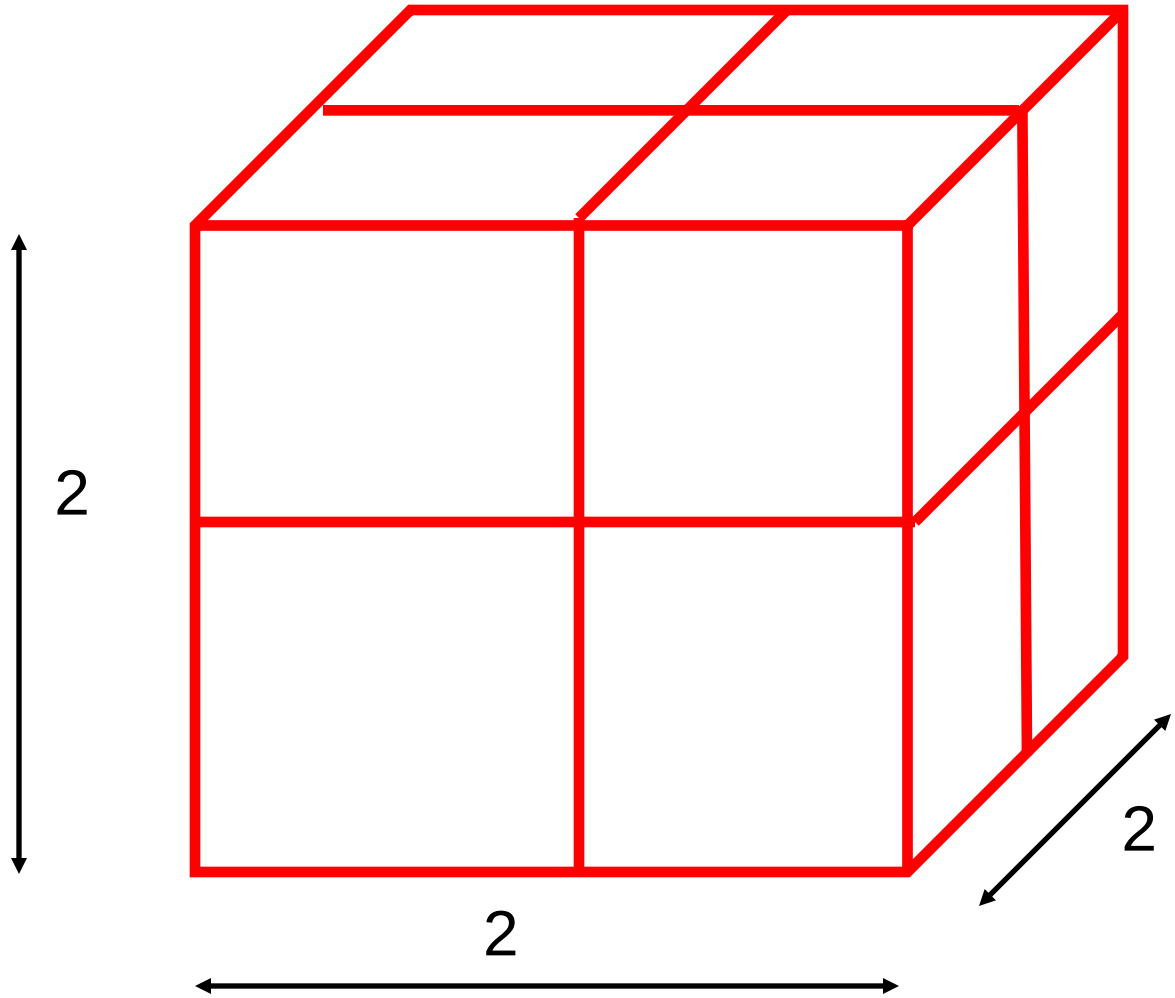
(Szeliski '93)

Complexity reduction: octrees

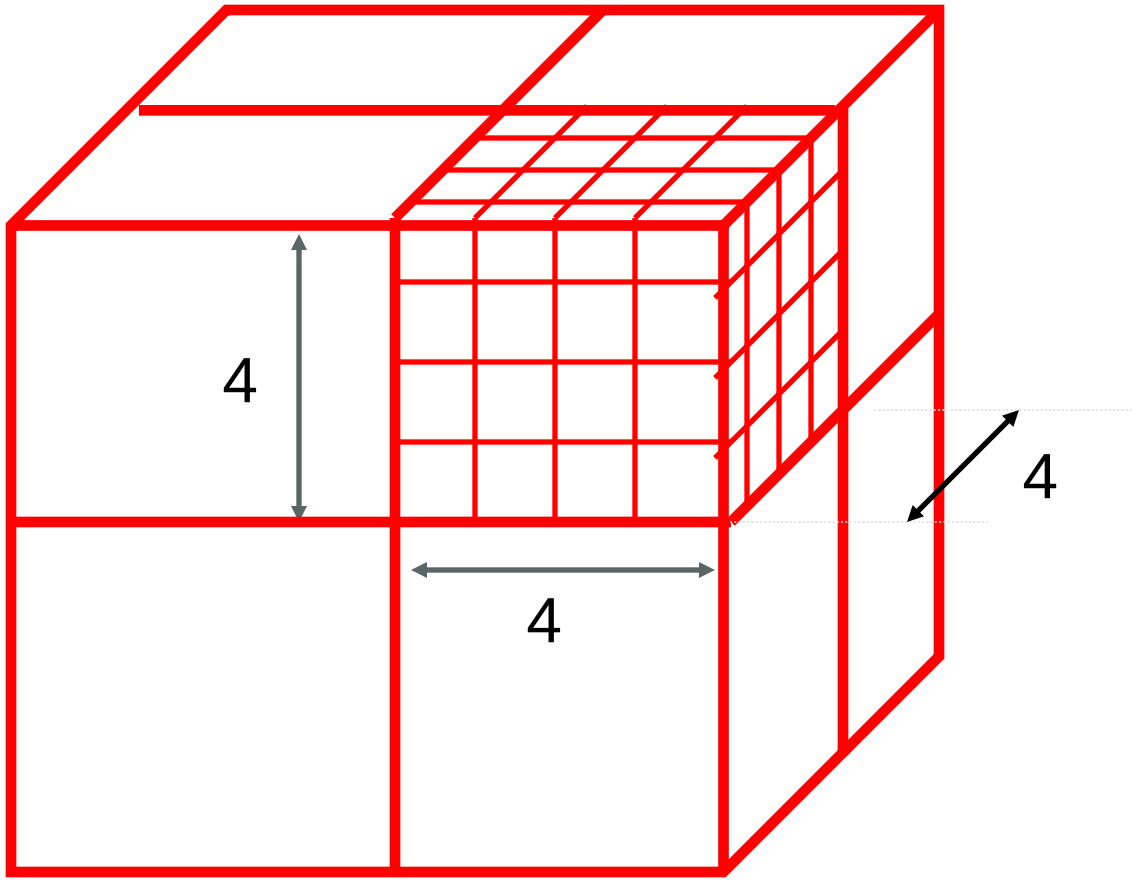


Complexity reduction: octrees

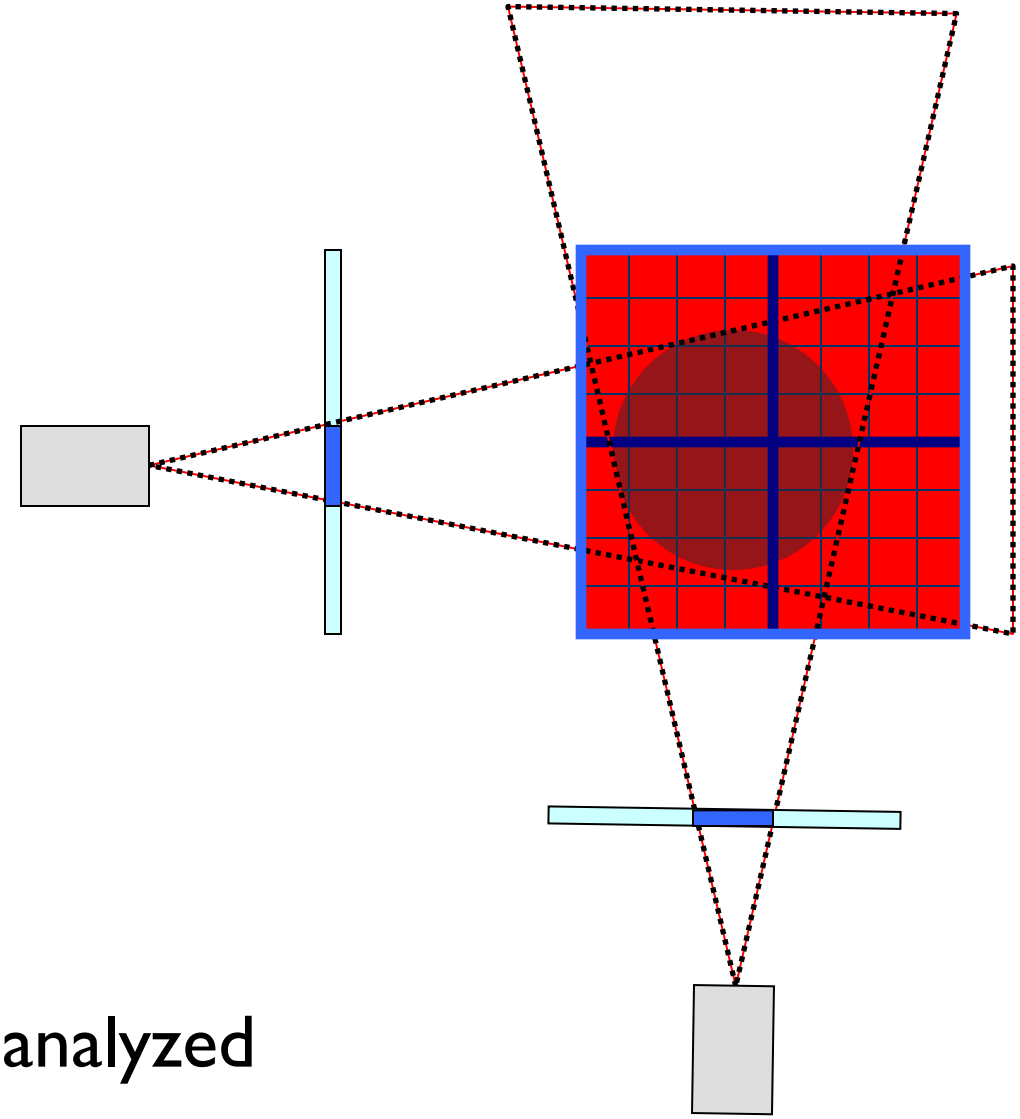
- Subdiving volume in voxels of progressive smaller size



Complexity reduction: octrees

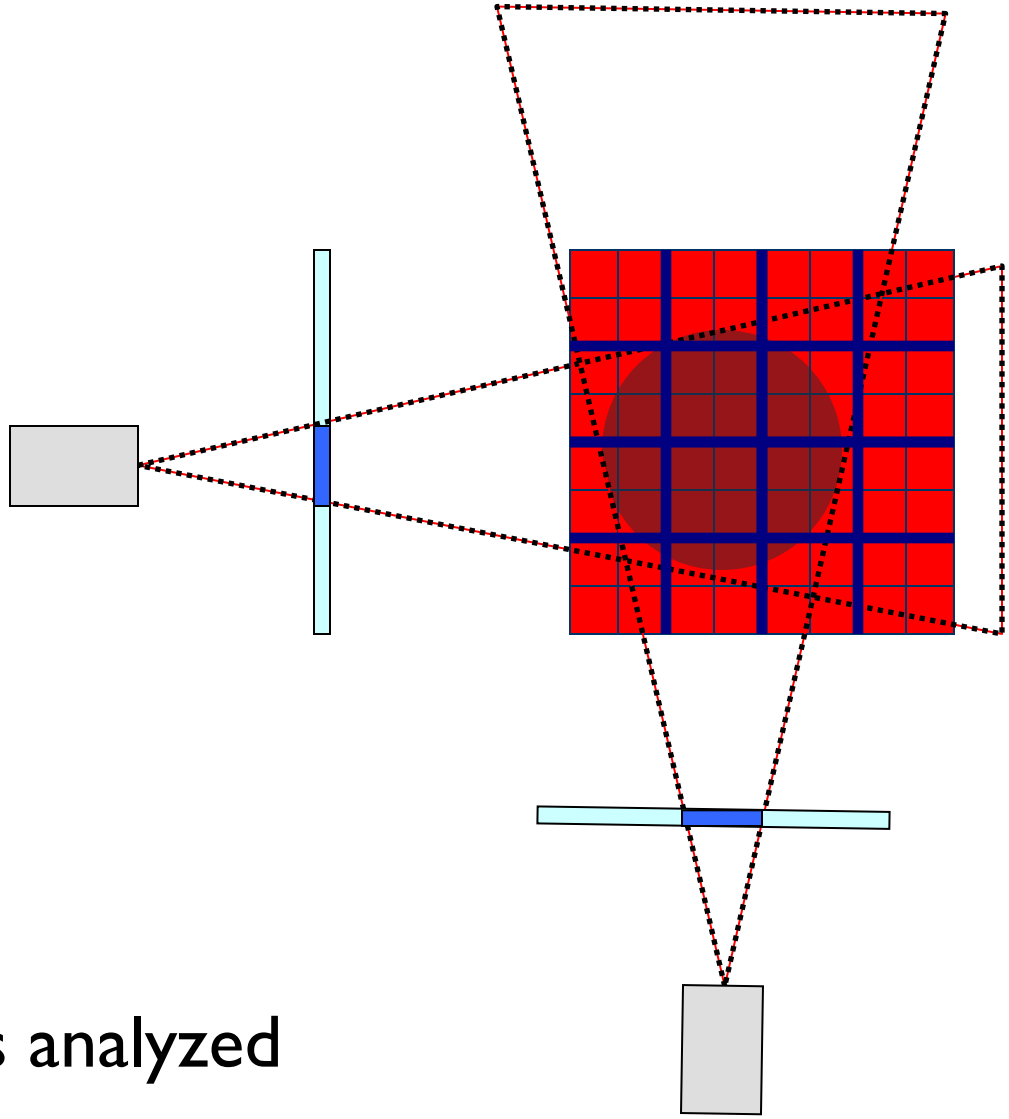


Complexity reduction: 2D Example



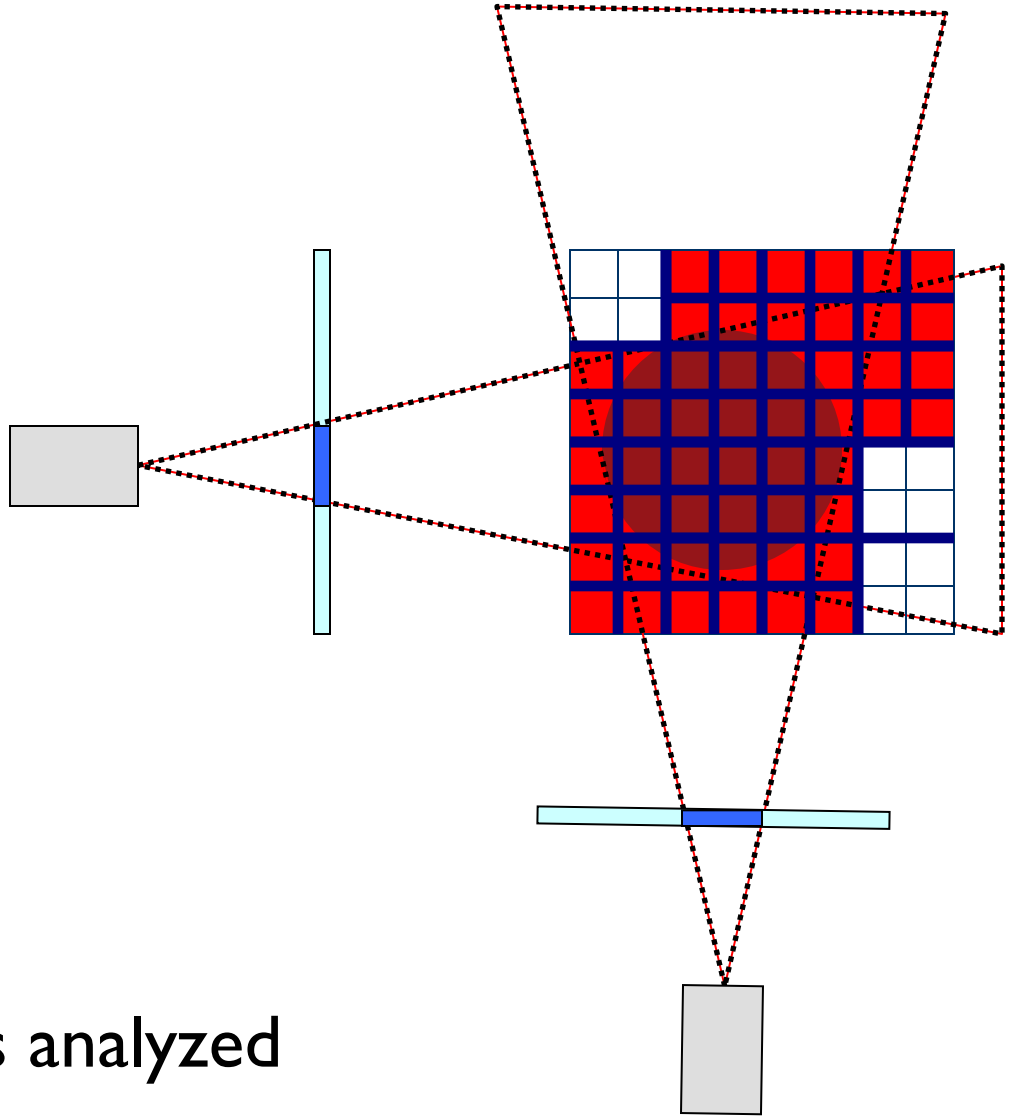
4 voxels analyzed

Complexity reduction: 2D Example



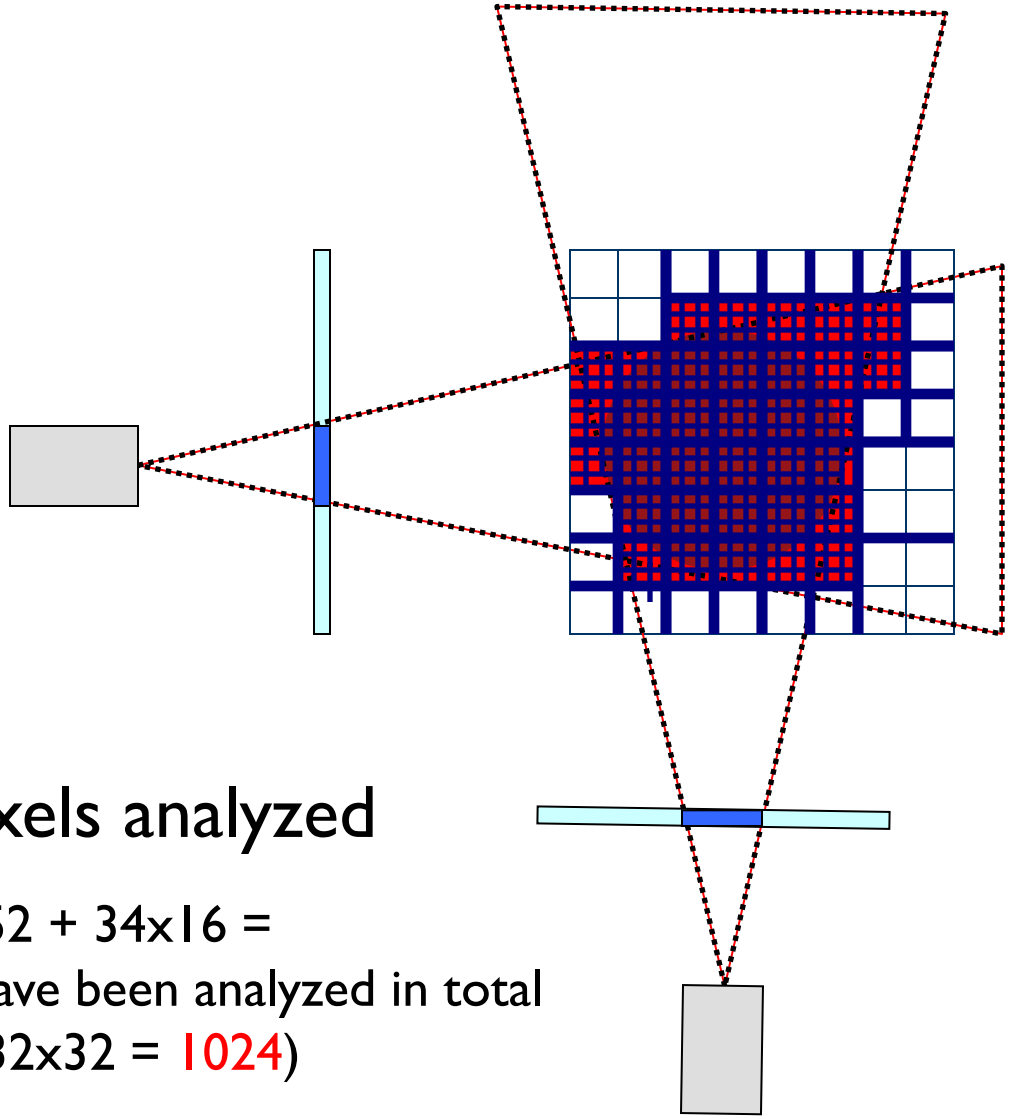
16 voxels analyzed

Complexity reduction: 2D Example



52 voxels analyzed

Complexity reduction: 2D Example



16x34 voxels analyzed

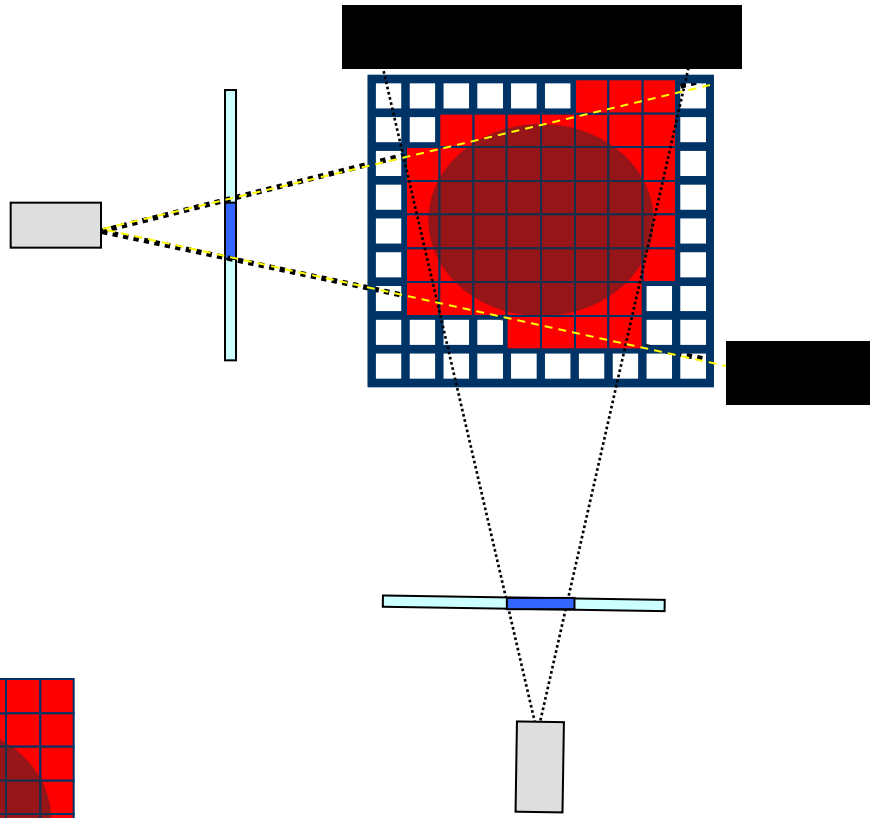
1+ 4 + 16 + 52 + 34x16 =
617 voxels have been analyzed in total
(rather than 32x32 = 1024)

Advantage of Space Carving

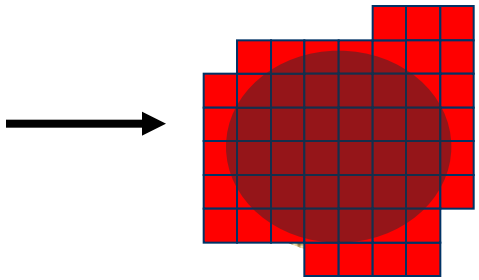
- Robust and Simple
- No need to solve for correspondences

Limitation of Space Carving

- Accuracy function of number of views



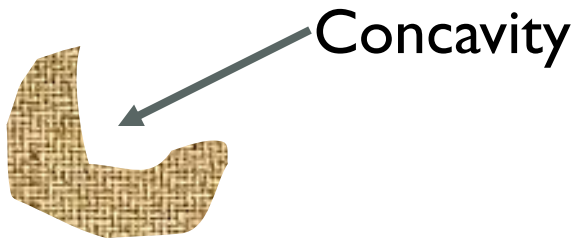
Not a good estimate

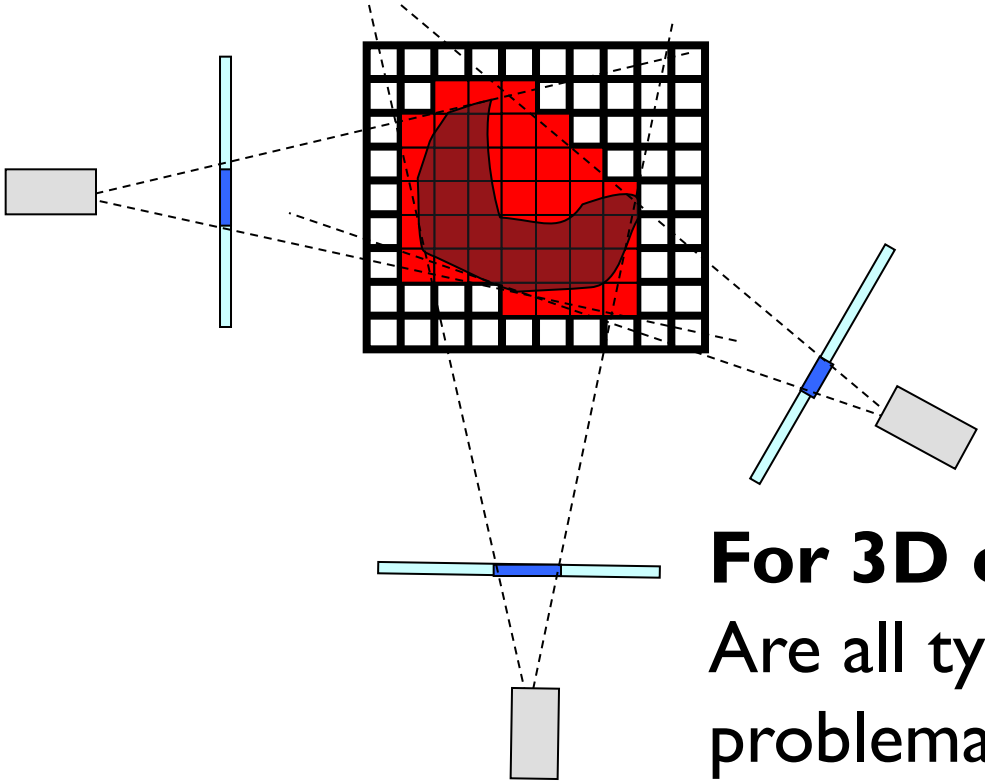


What else?

Limitations of space carving

Concavities are not modeled

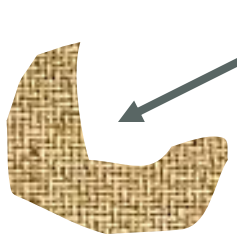


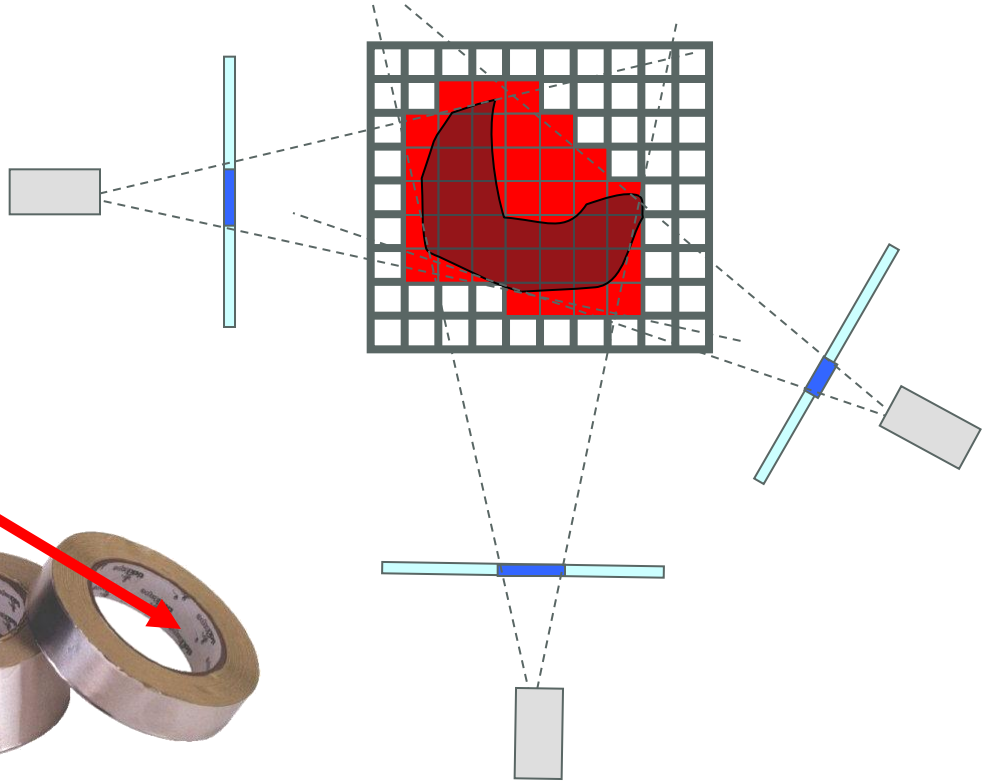


For 3D objects:
Are all types of concavities problematic?

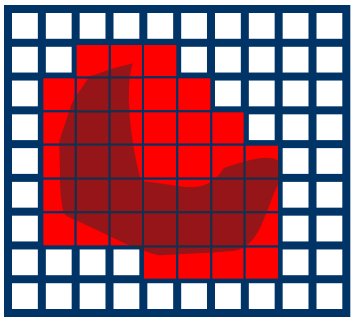
Limitations of space carving

Concavities are not modeled





Visual hull



(hyperbolic regions are ok)

- Closest approximation

  Conservative

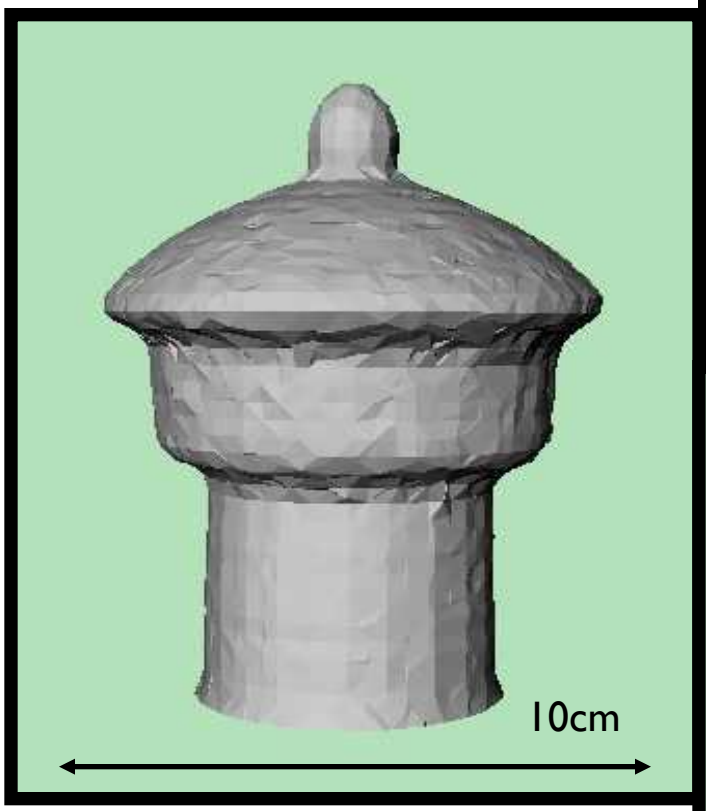
Space carving: a classic setup

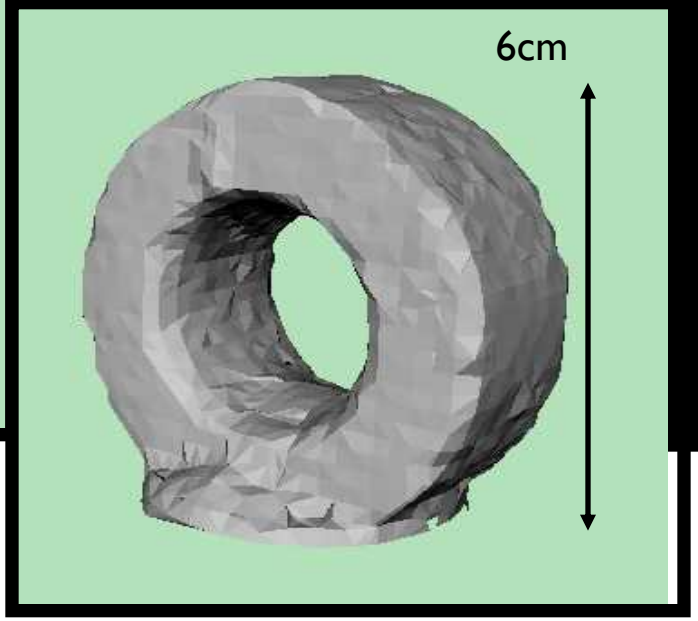
Courtesy of seitz & dyer

Space carving: a classic setup

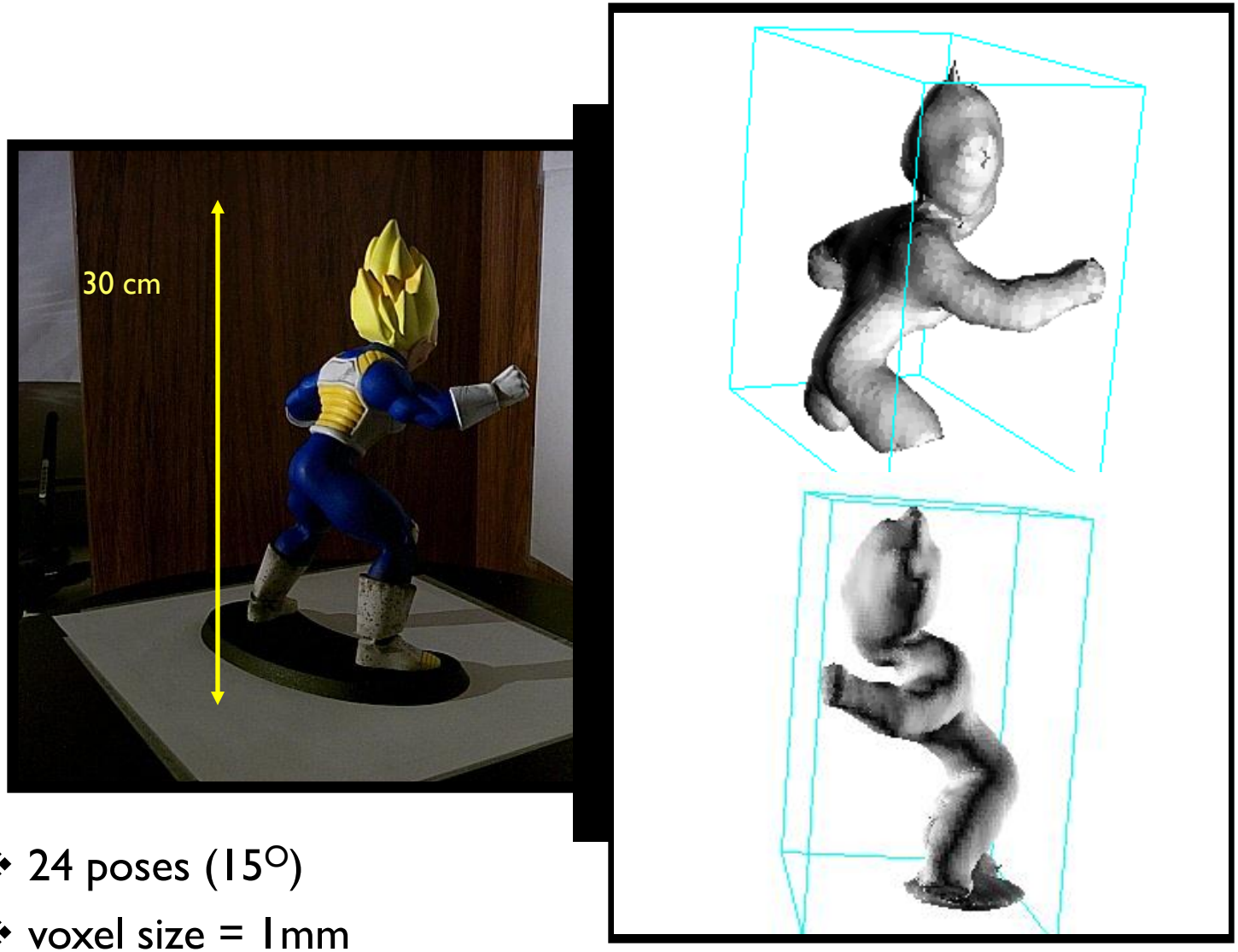
Space carving: Experiments

- 24 poses (15°)
- Voxel size = 2mm





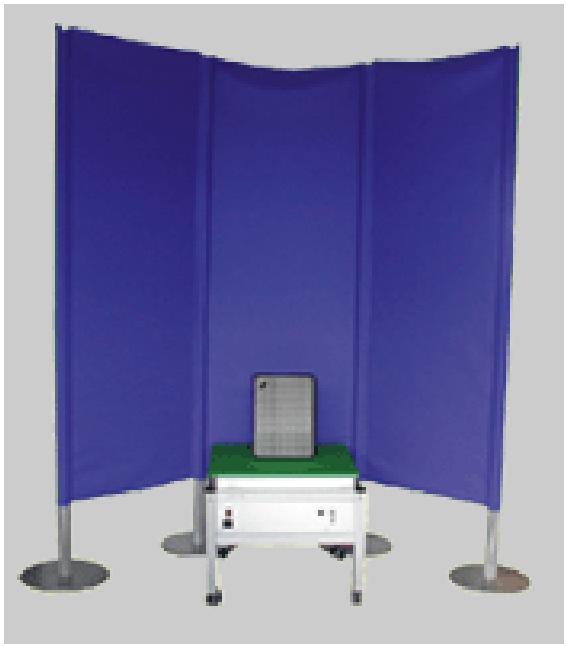
Space carving: Experiments

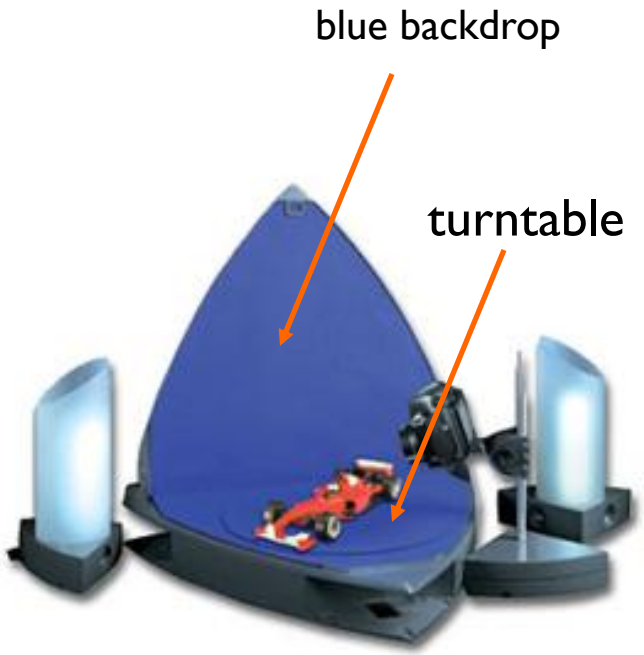


- 24 poses (15°)
- voxel size = 1mm

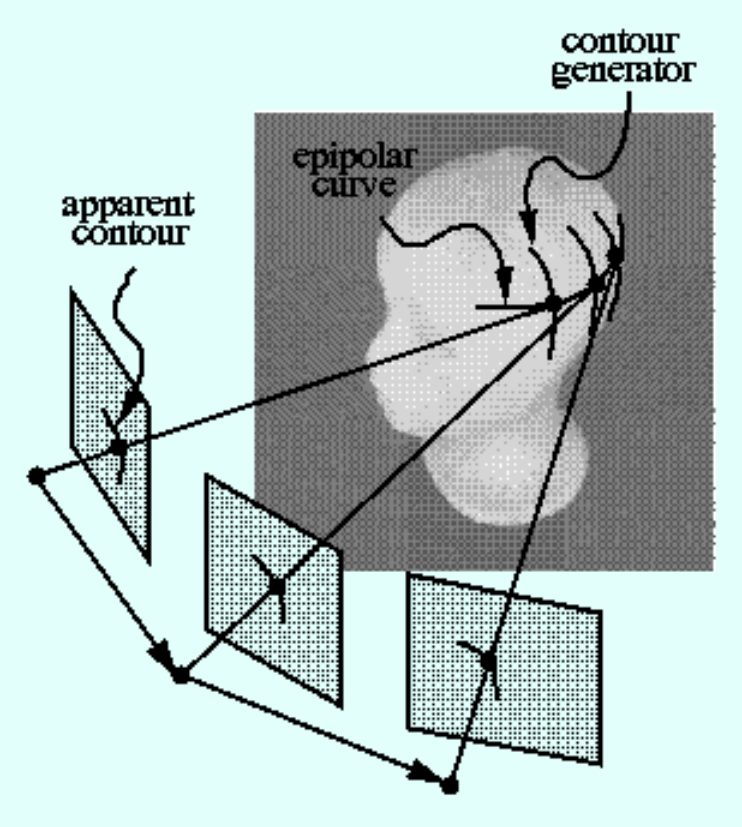
Space carving: Conclusions

- Robust
- Produce conservative estimates
- Concavities can be a problem
- Low-end commercial 3D scanners





Contours in the computer vision literature



Picture from of Sato & Cipolla

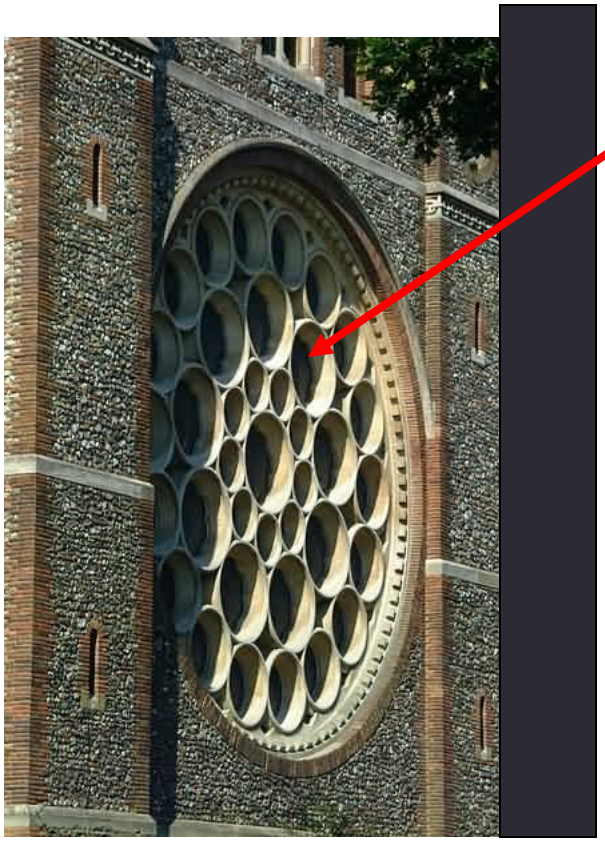
- Giblin and Weiss (1987)
- Cipolla and Blake (1992)
- Vaillant and Faugeras (1992)
- Ponce ('92), Zheng('94)
- Furukawa et al. ('05…)

Outline

- Volumetric Reconstruction

- Definition
- Shape from Contours
- Shape from Shadows
- Voxel Coloring

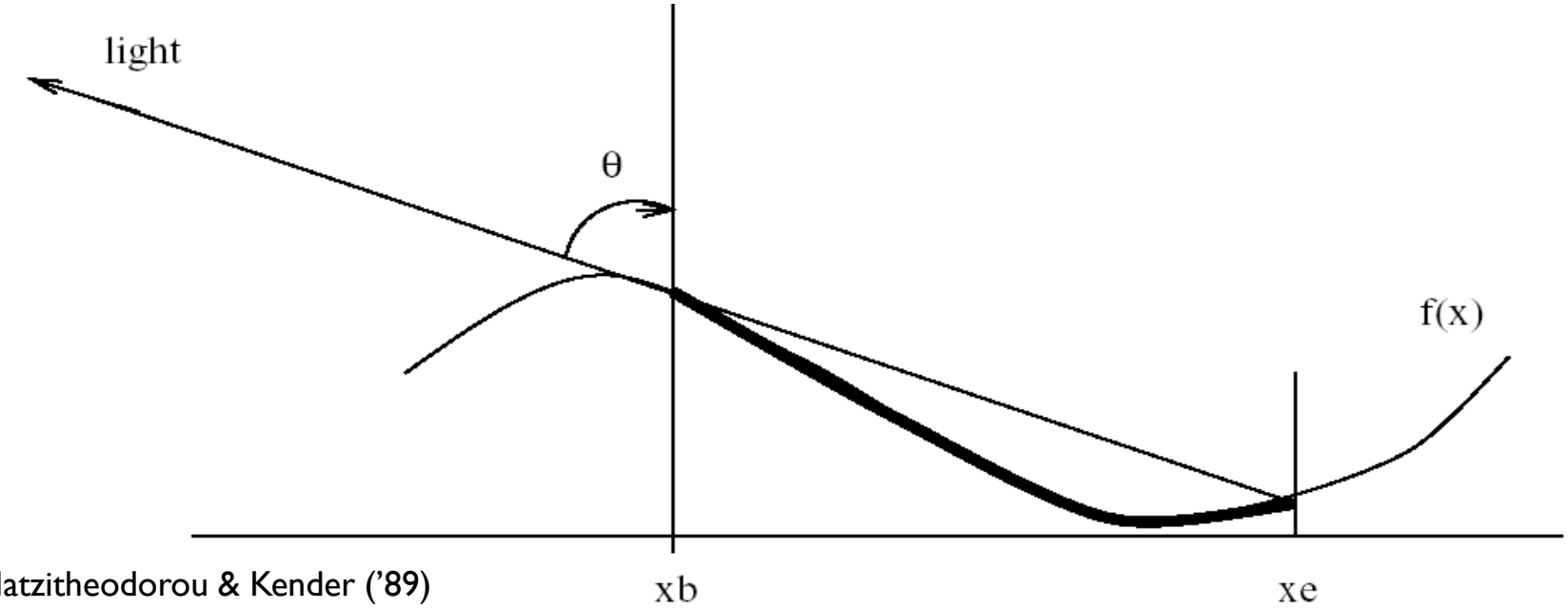
Self-shadows are visual cues for shape recovery



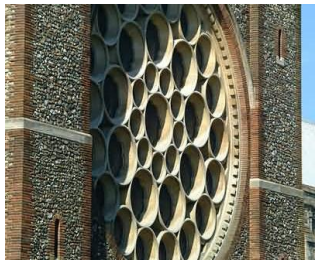
Self-shadows indicate concavities
(not modeled by contours)

Shape from shadow in the literature

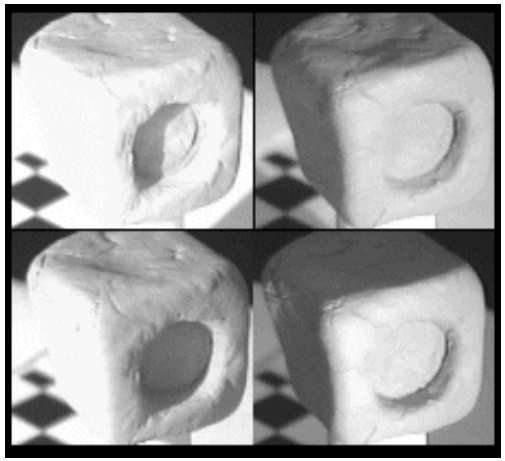
- Shafer & Kanade ('83)
- Hatzitheodorou & Kender ('89)
- et al…



Hatzitheodorou & Kender ('89)

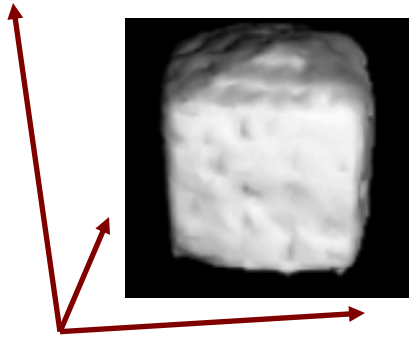


Shadow Carving

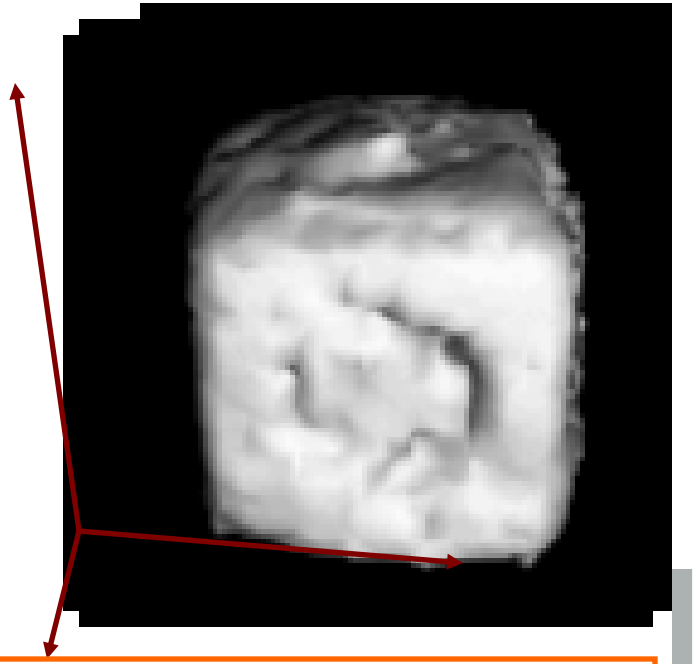


Self-shadows

+



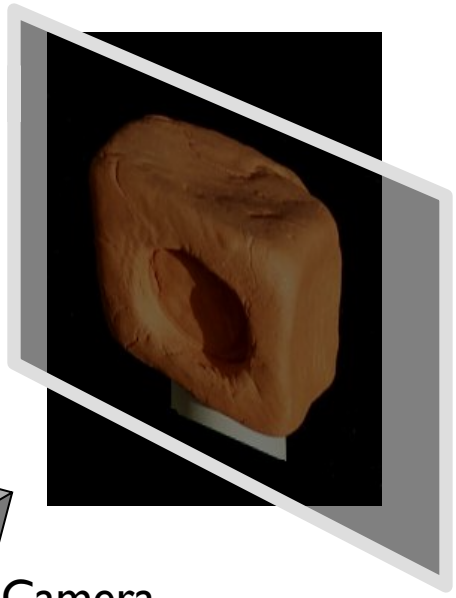
Object's upper bound



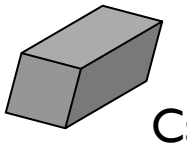
- Robust with respect to shadow estimates
- Object with arbitrary topology (no 2.5D terrains)

Shadow carving: setup

Object



Image

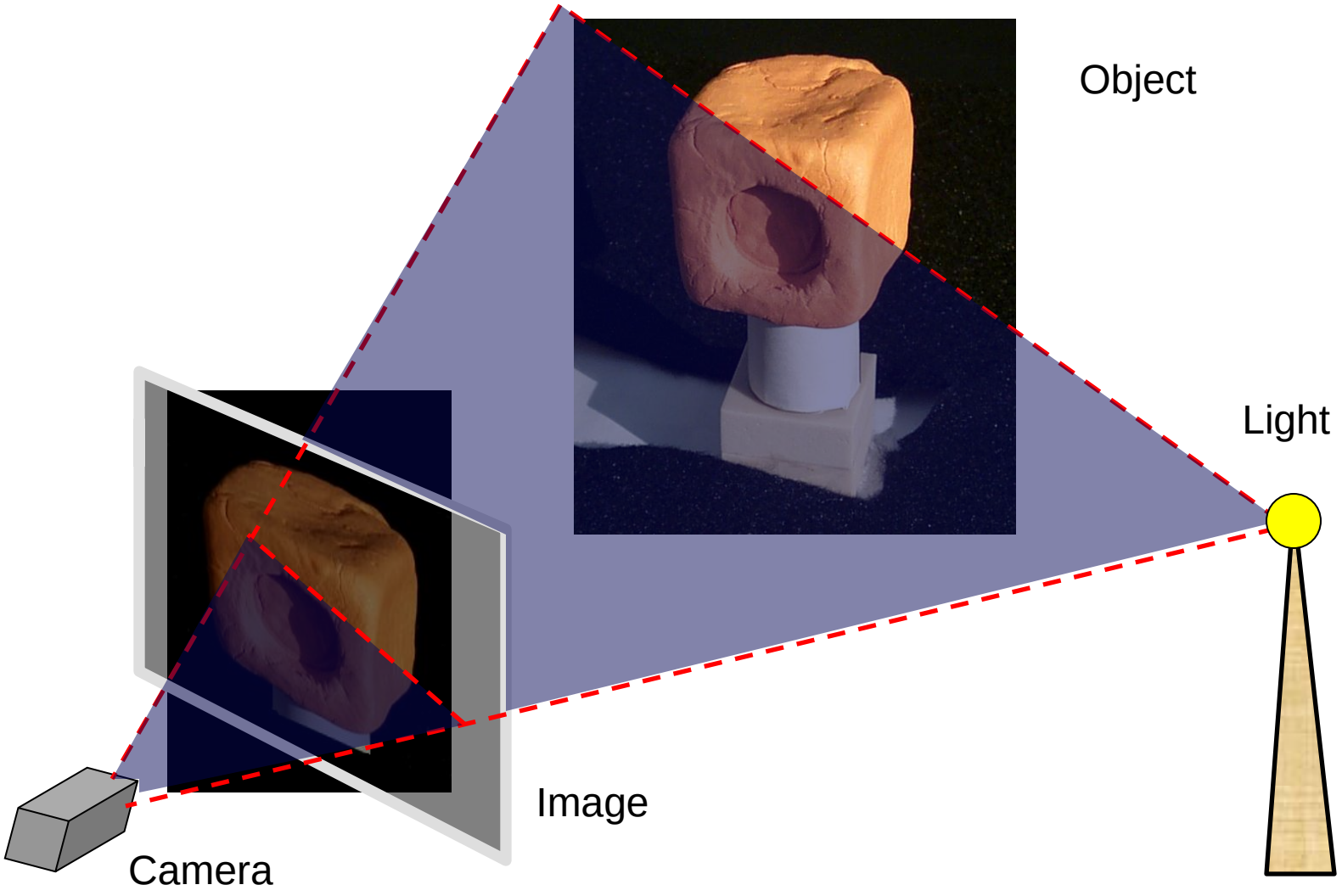


Camera

Light

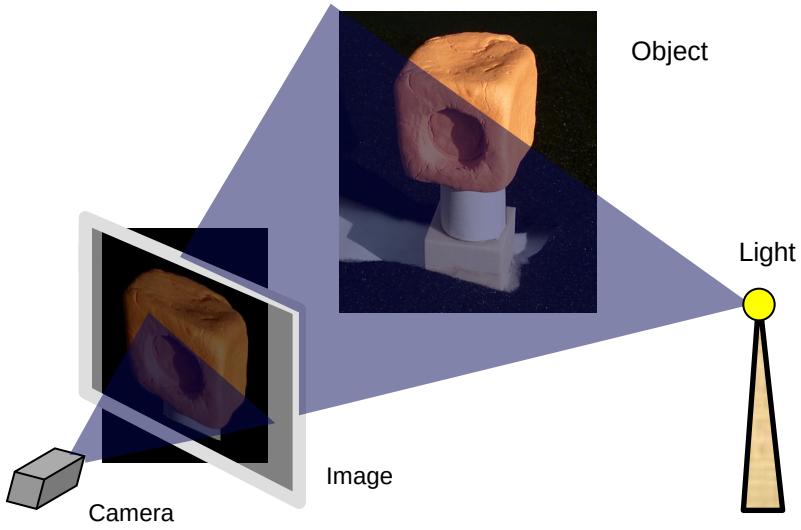
Let's define a 2D model…

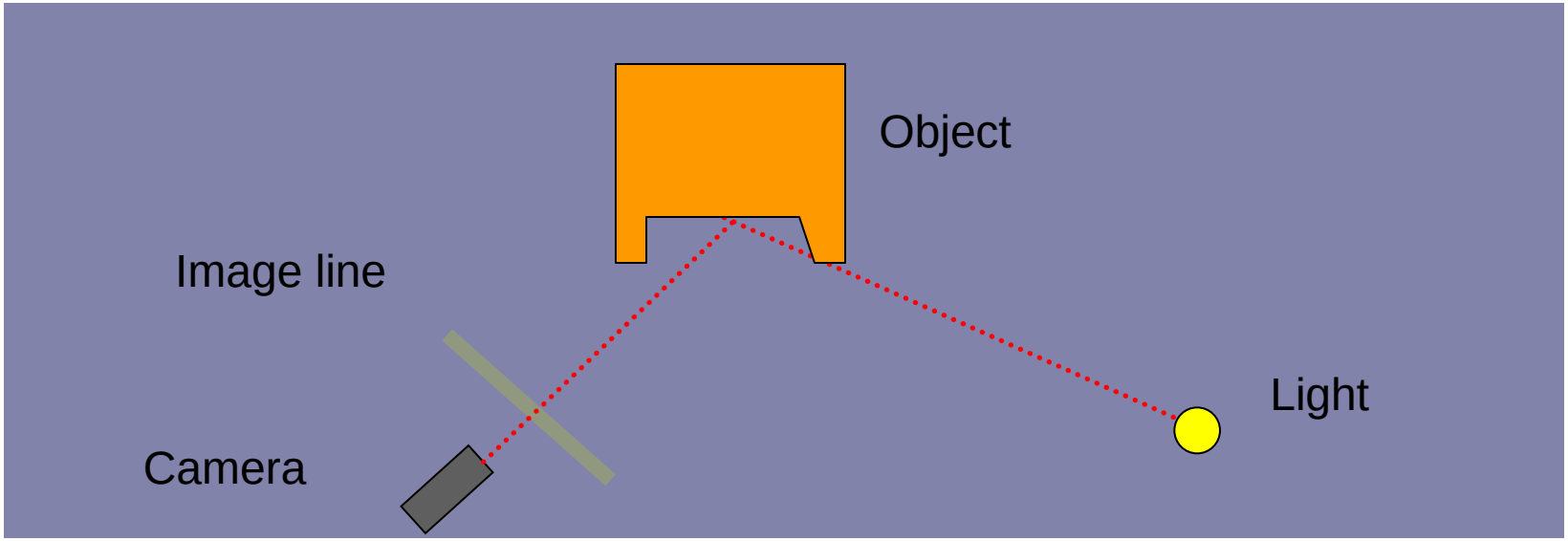
2D slice model



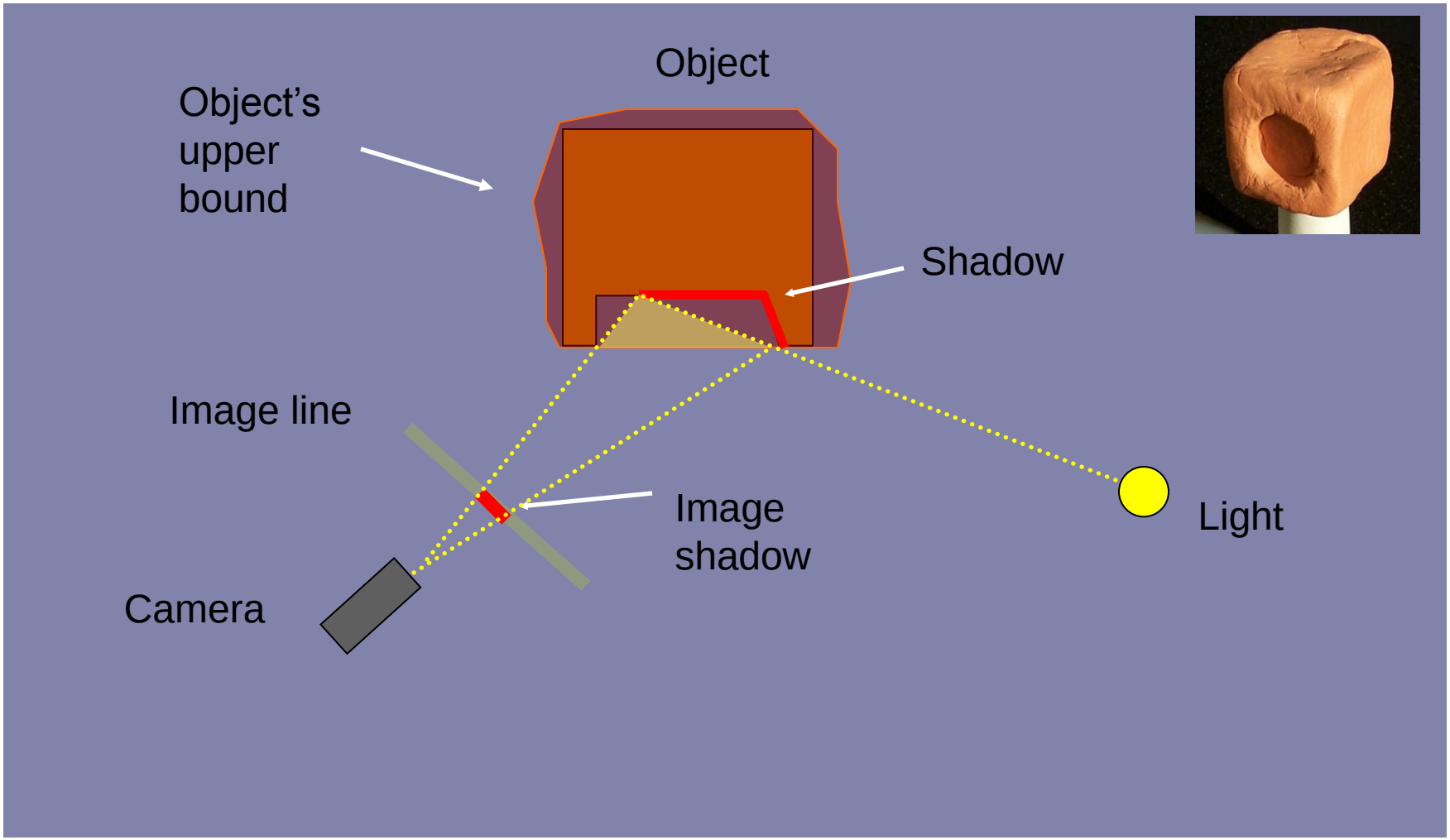
2D slice model

From 3D to 2D

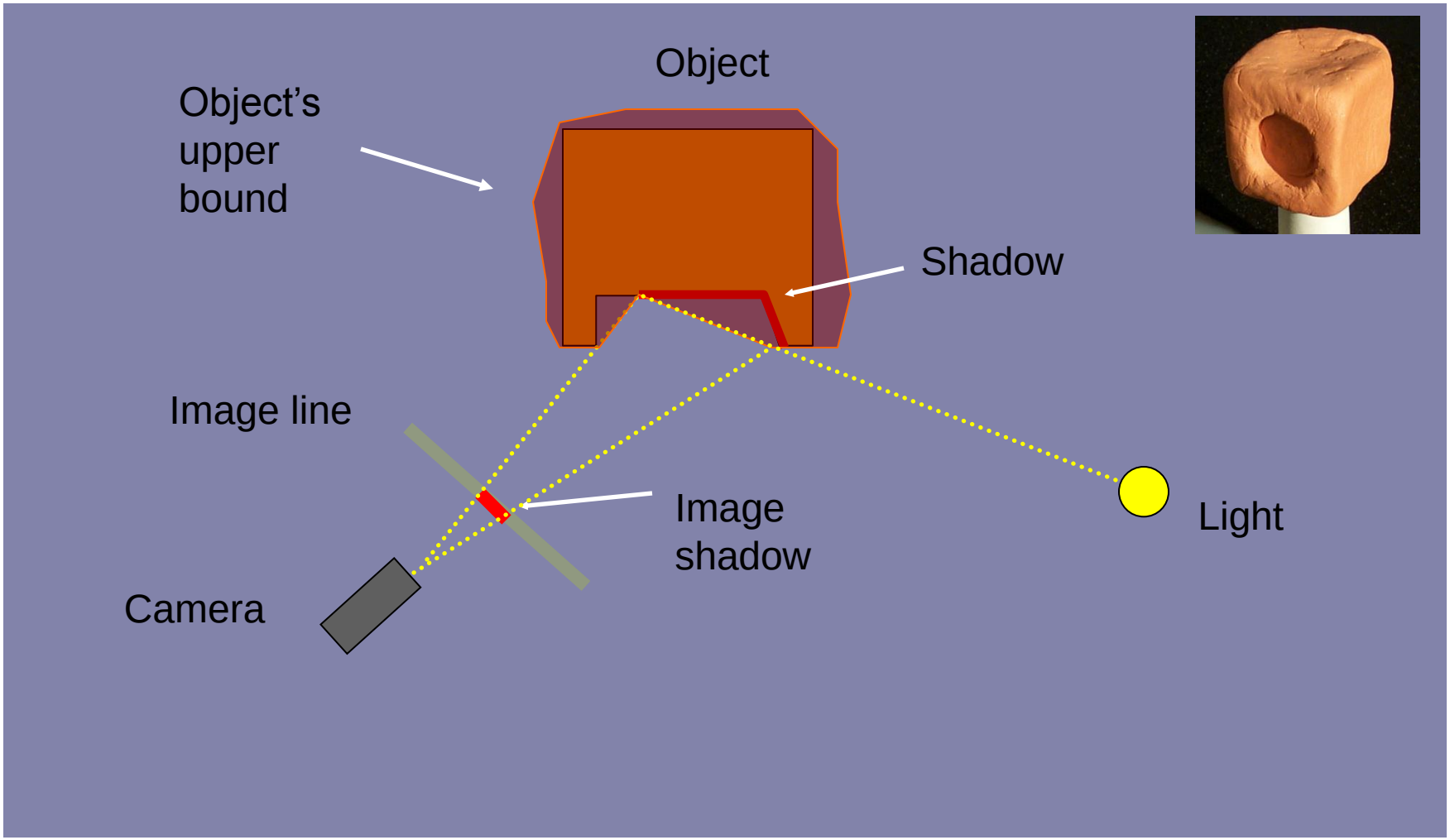




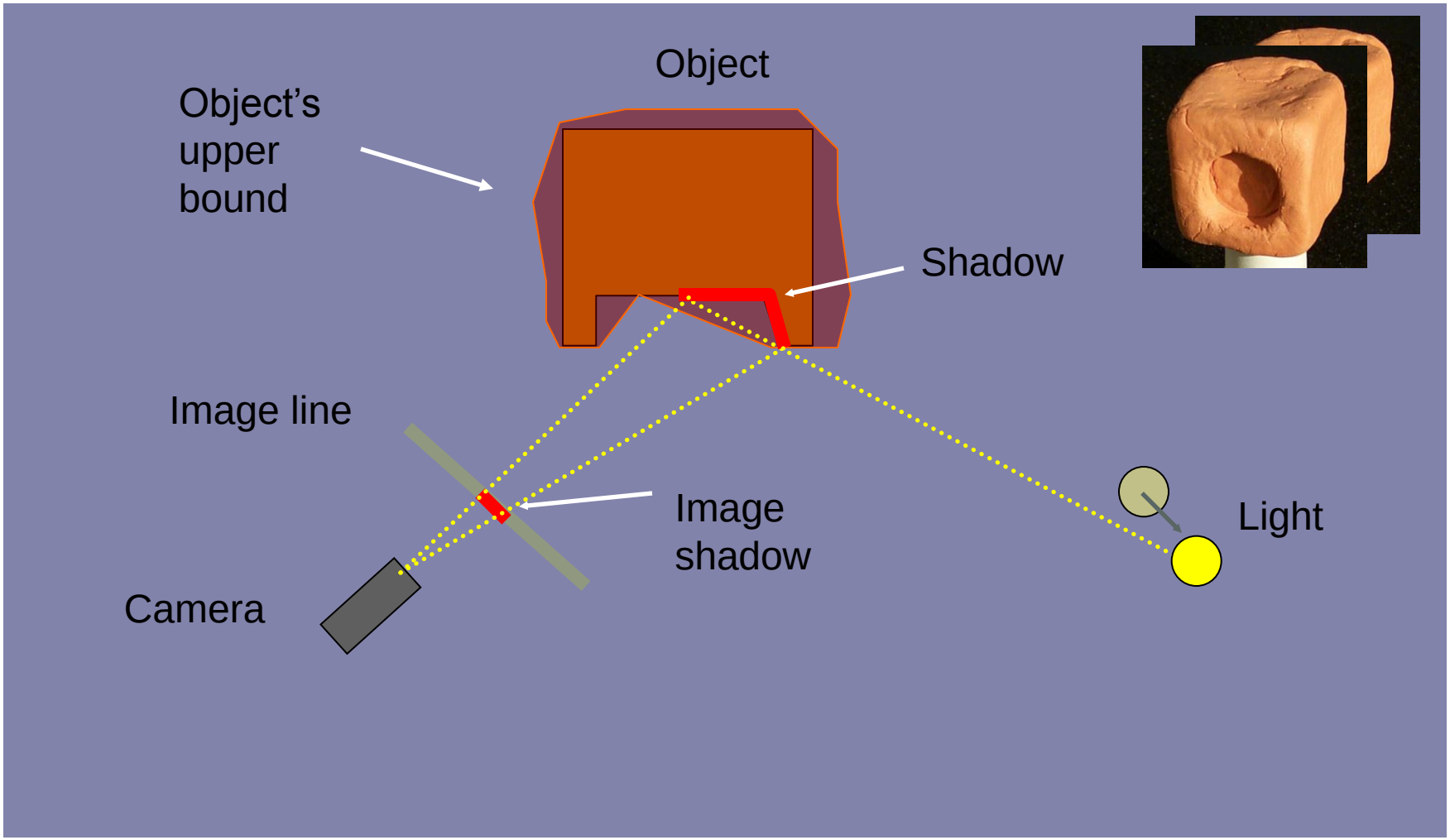
The idea


Object
Object's upper bound
Shadow
Image line
Image shadow
Light
Camera

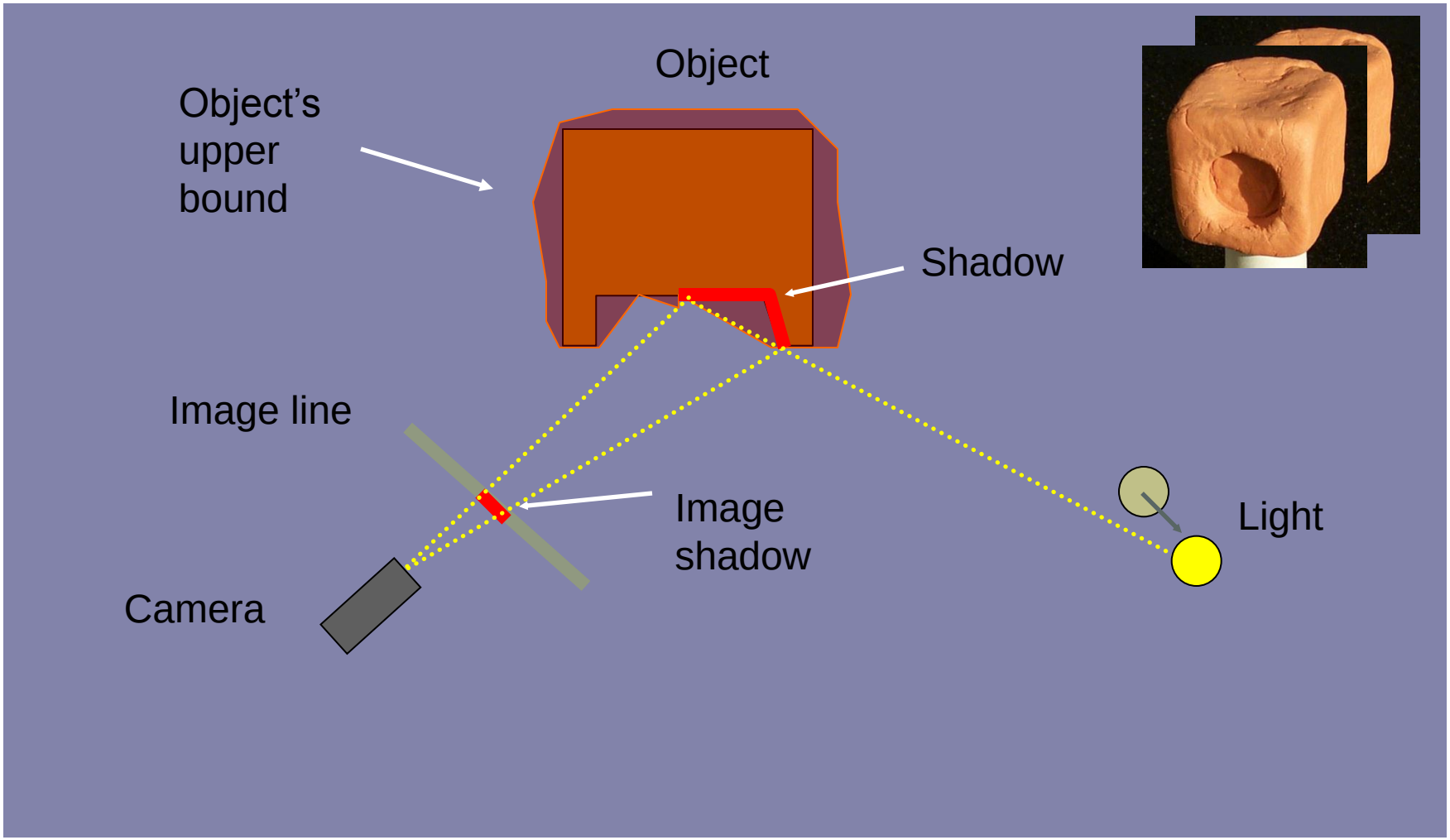
The idea


Object
Object's upper bound
Shadow
Image line
Image shadow
Light
Camera

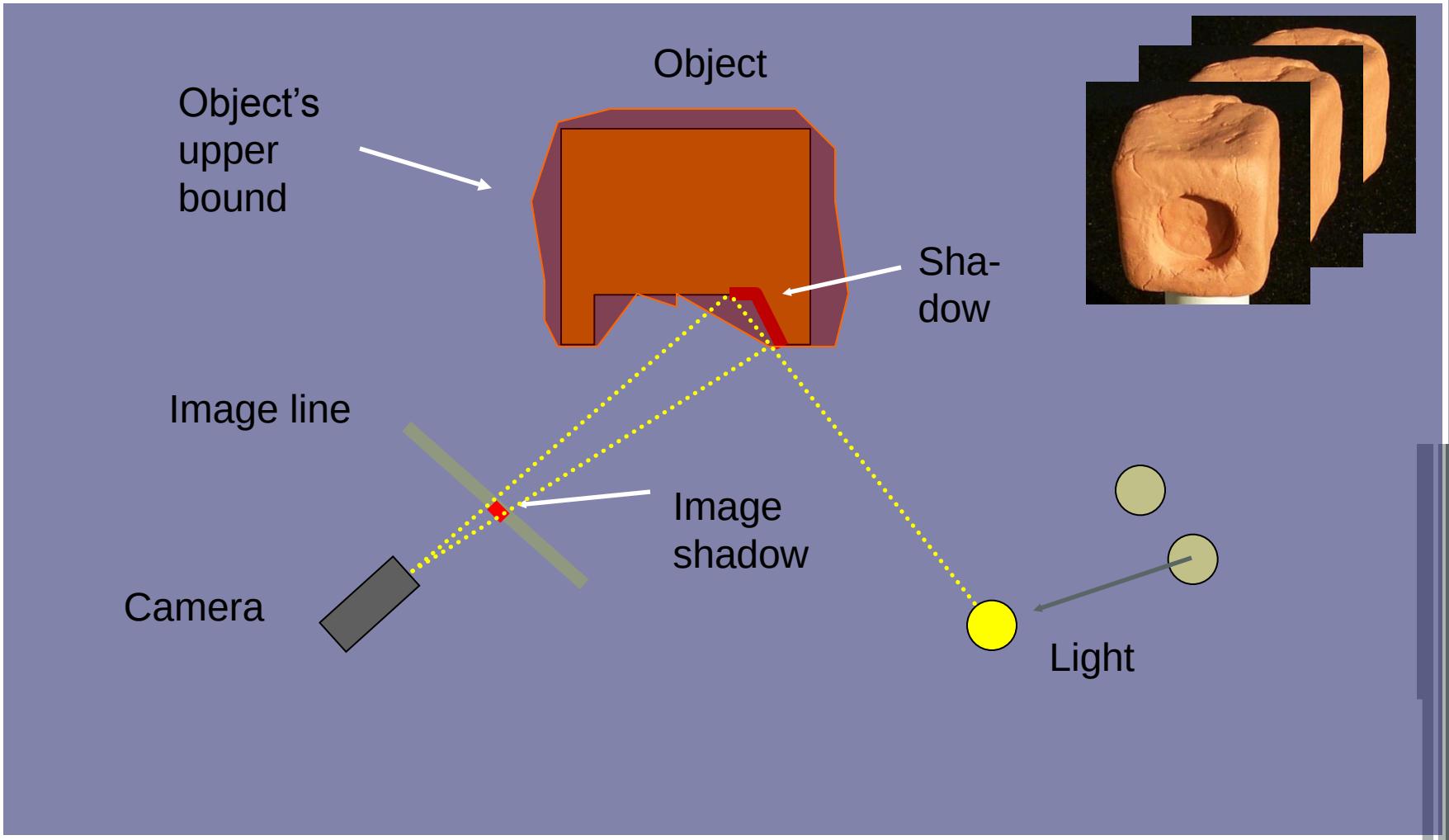
The idea


Object
Object's upper bound
Shadow
Image line
Image shadow
Light
Camera

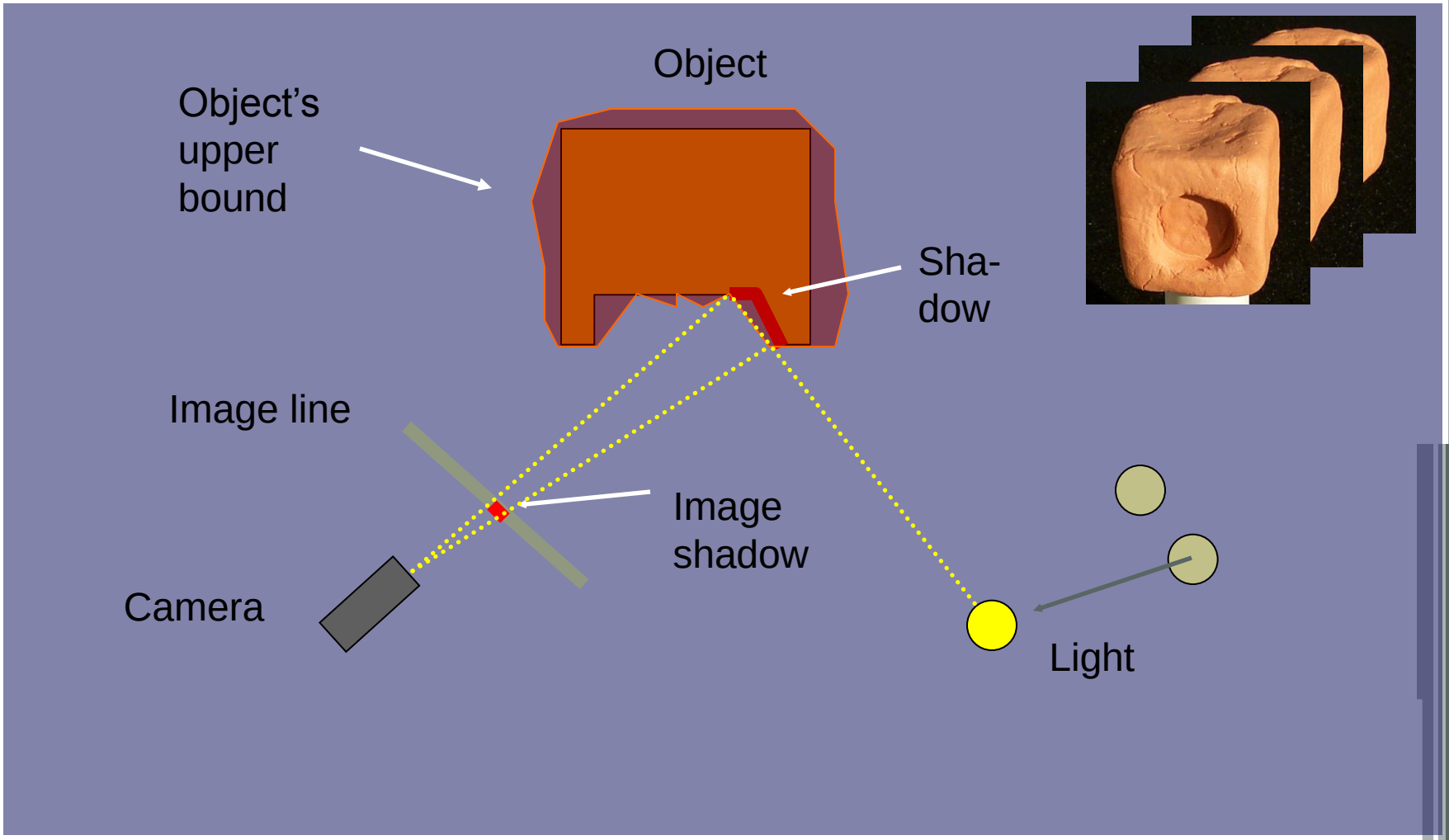
The idea



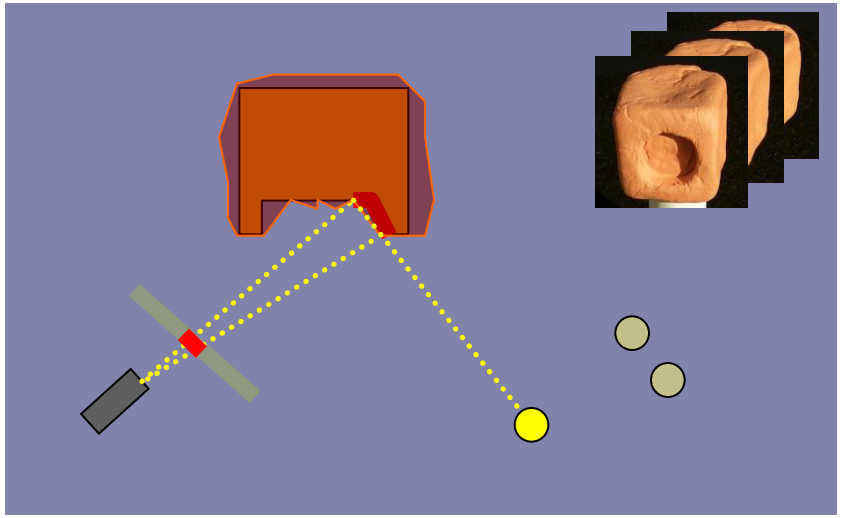
The idea

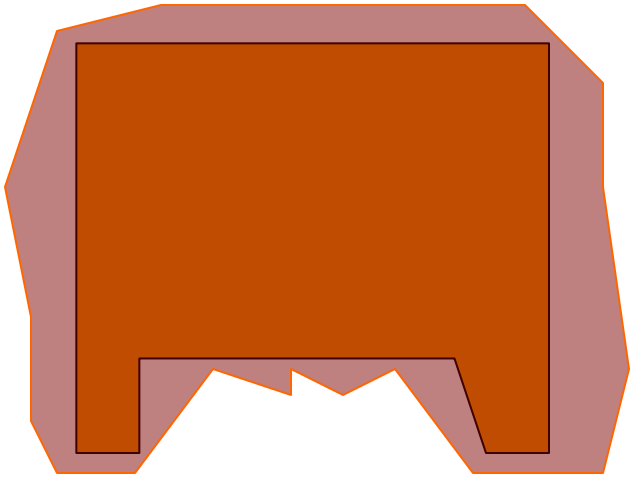


The idea



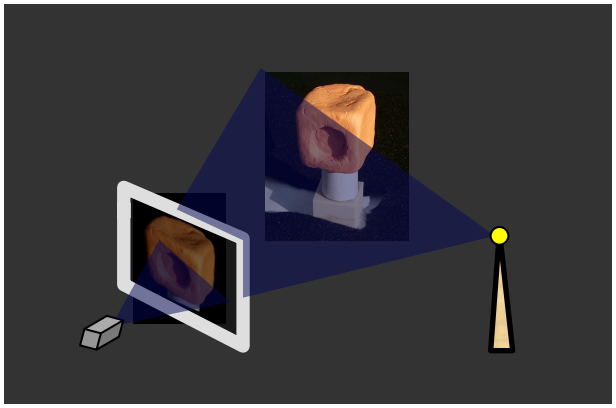
The idea

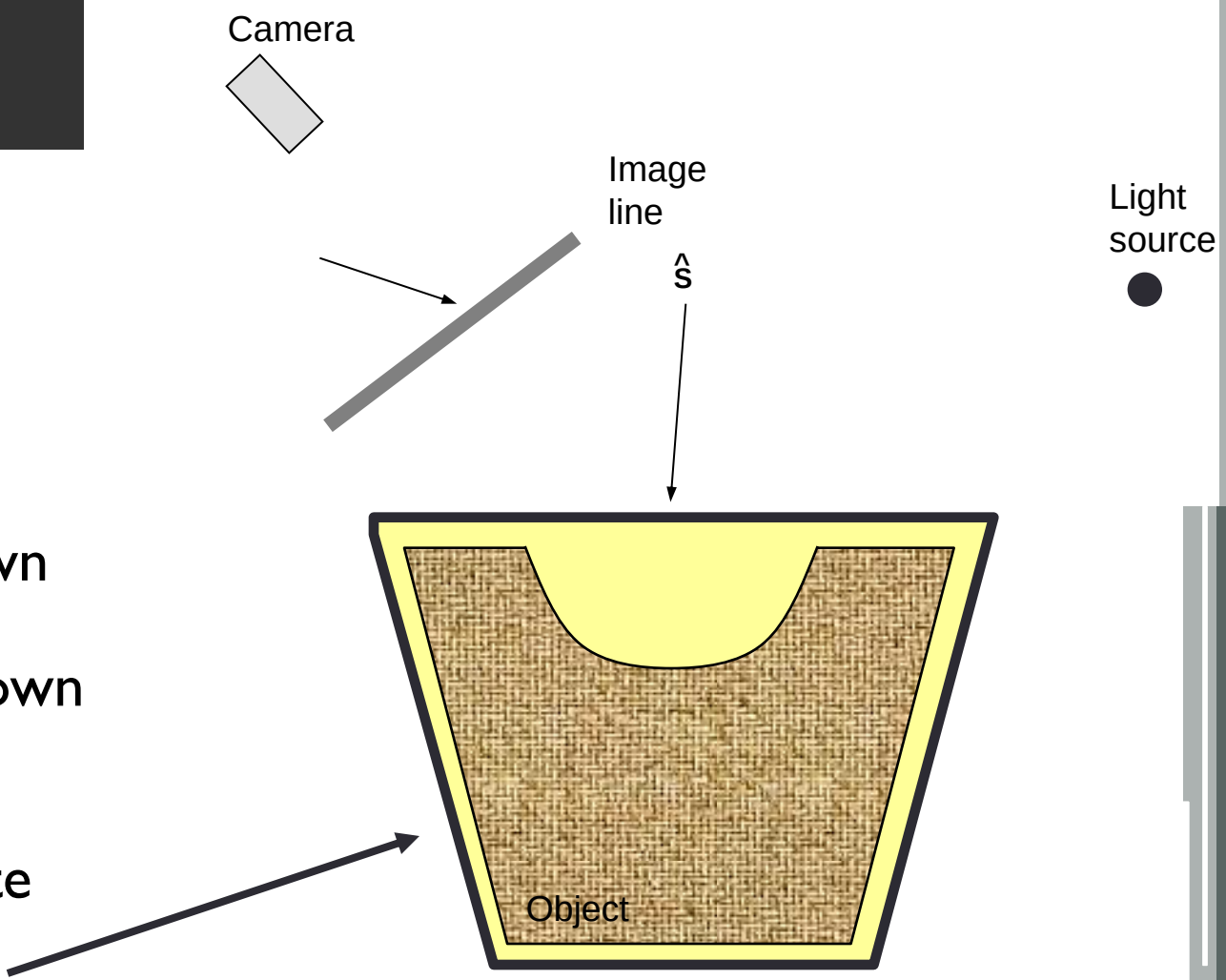




Reconstruction after 3 iterations

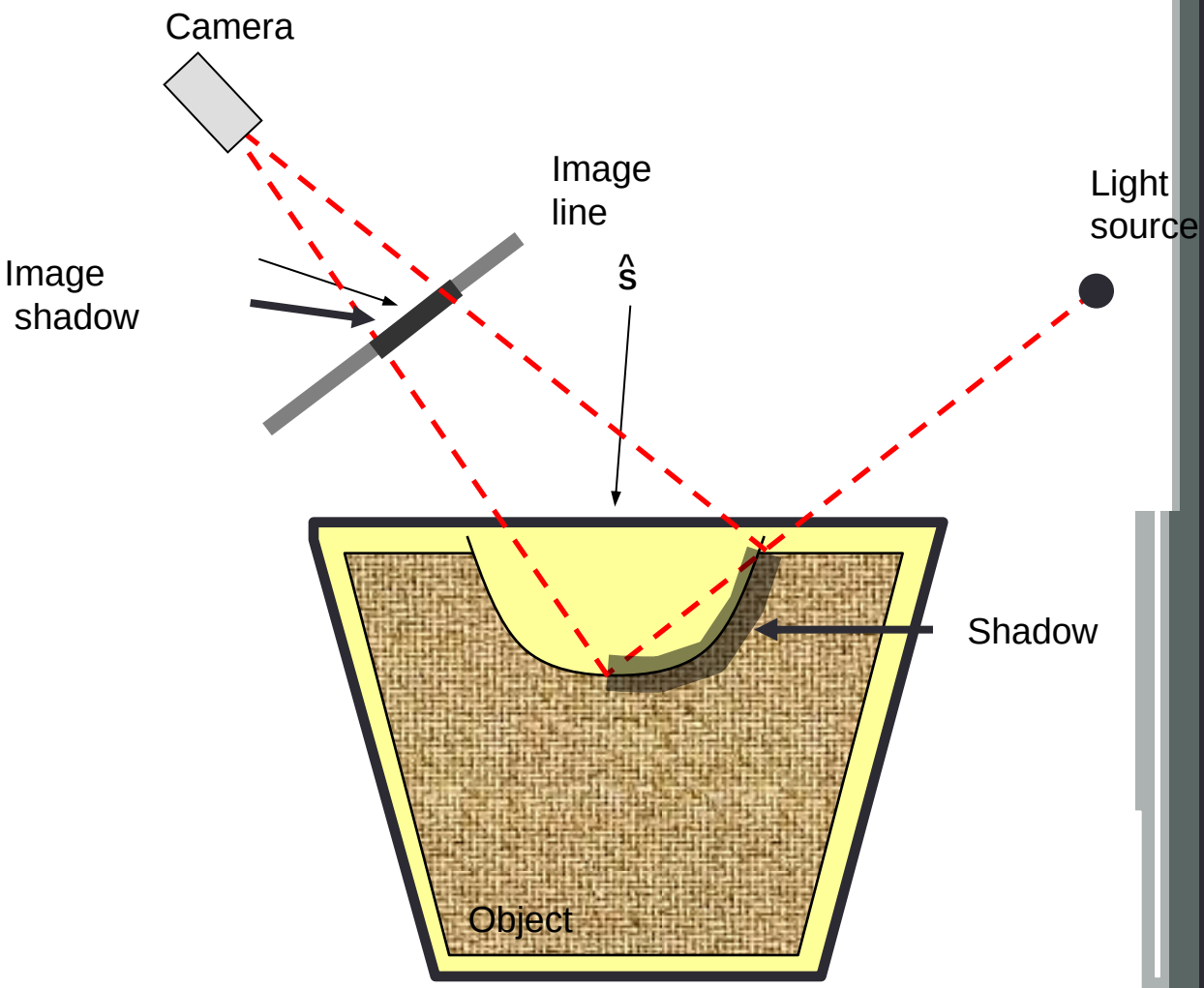
What is known?





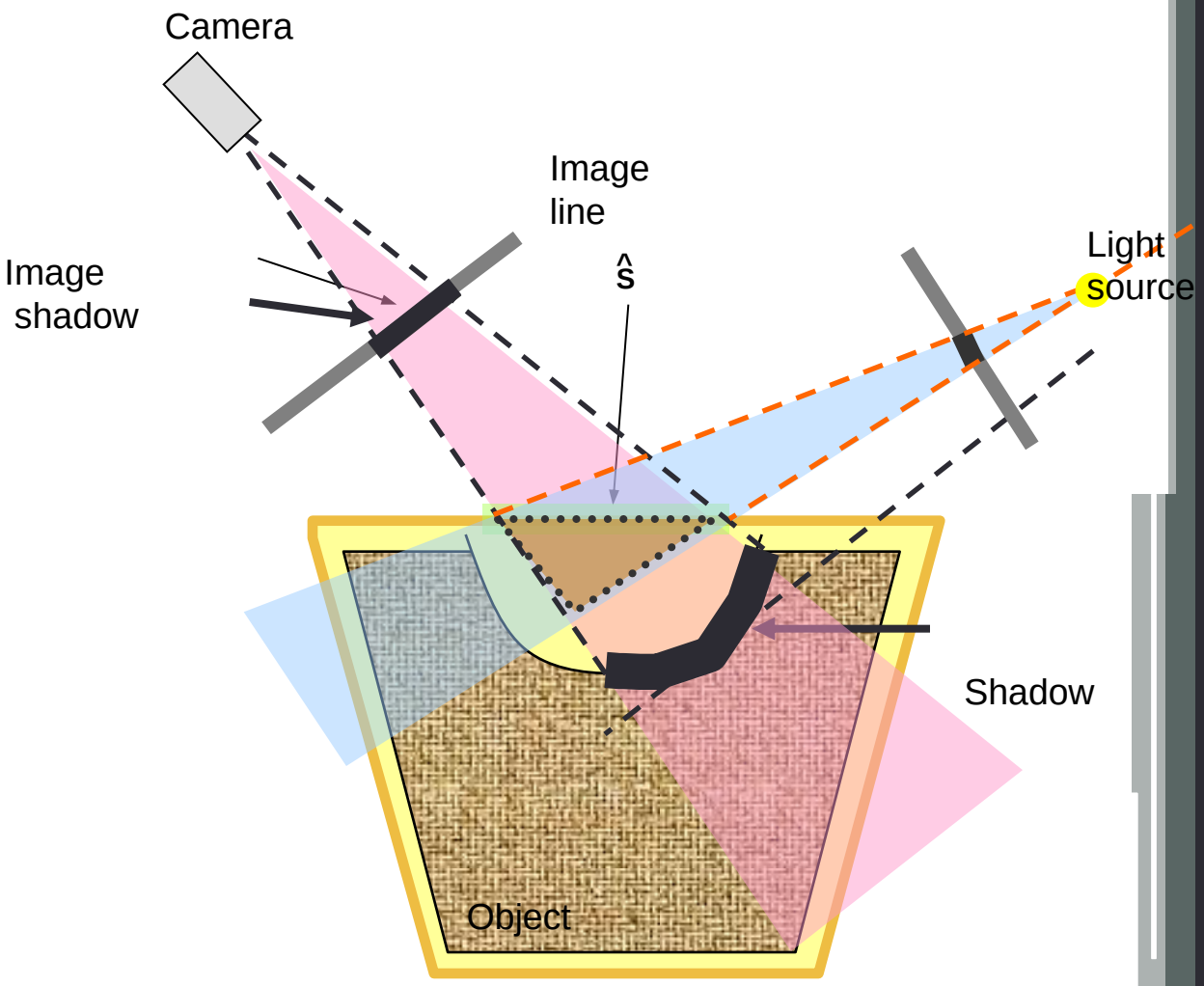
- Light known
- Camera known
- Object unknown
- Upper bound object estimate

What can we carve?



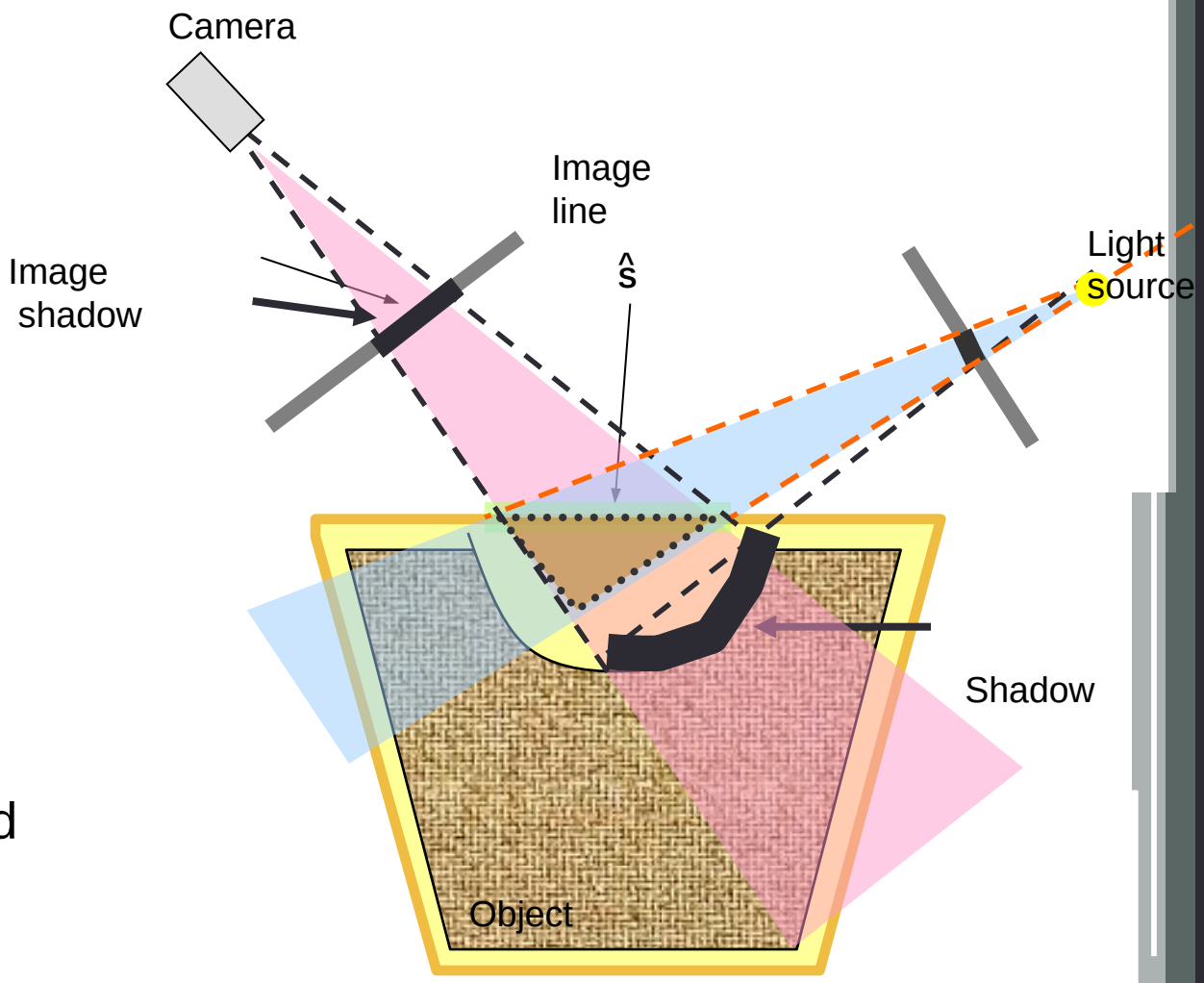
What can we carve?

- shadow cone
- upper-bound shadow
- light cone
- carvable area



What can we carve?

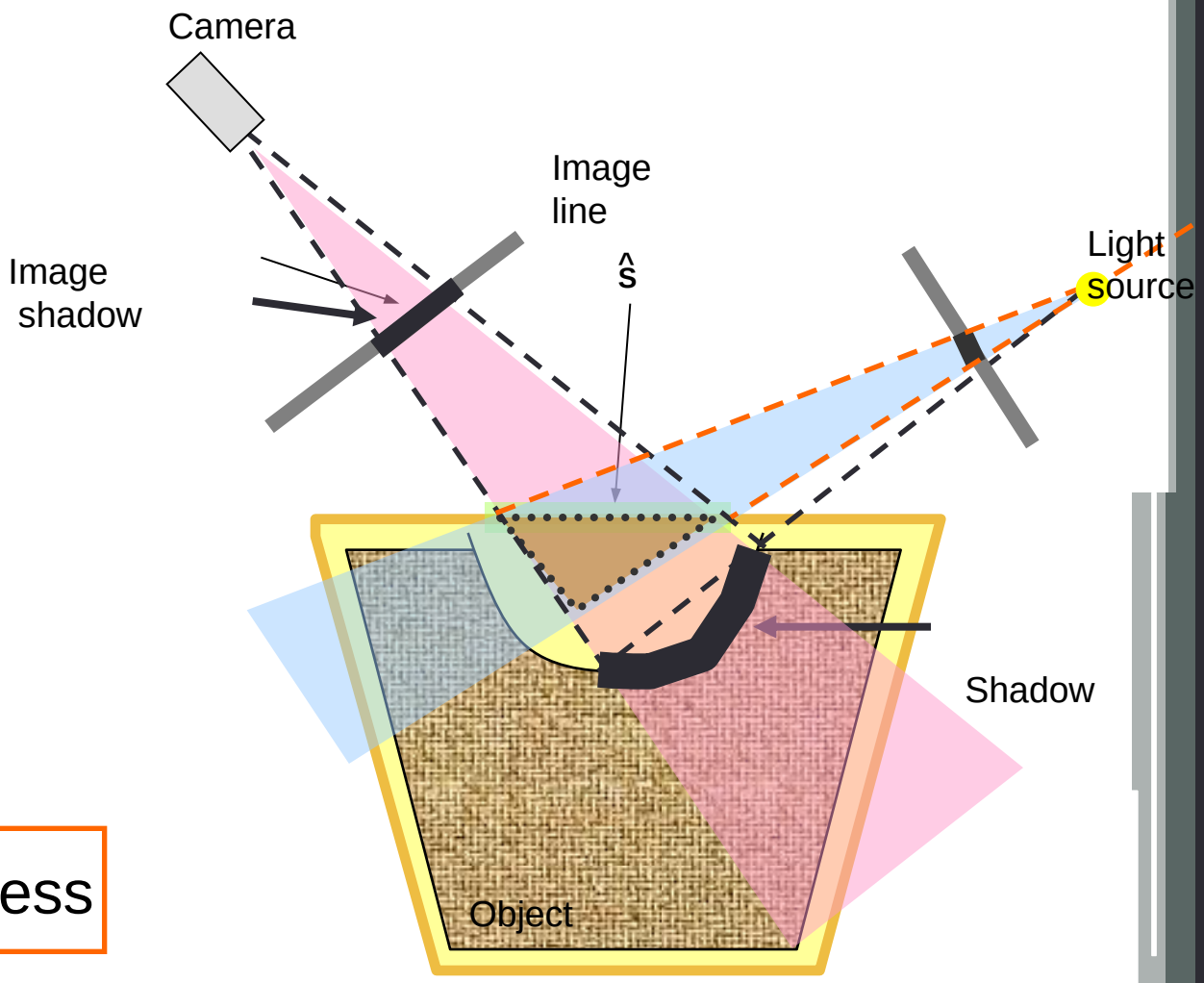
- shadow cone
- upper-bound shadow
- light cone
- carvable area



The upper-bound estimate is refined

What can we carve?

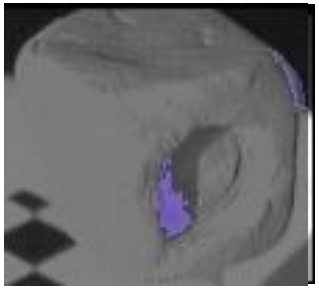
- shadow cone
- upper-bound shadow
- light cone
- carvable area



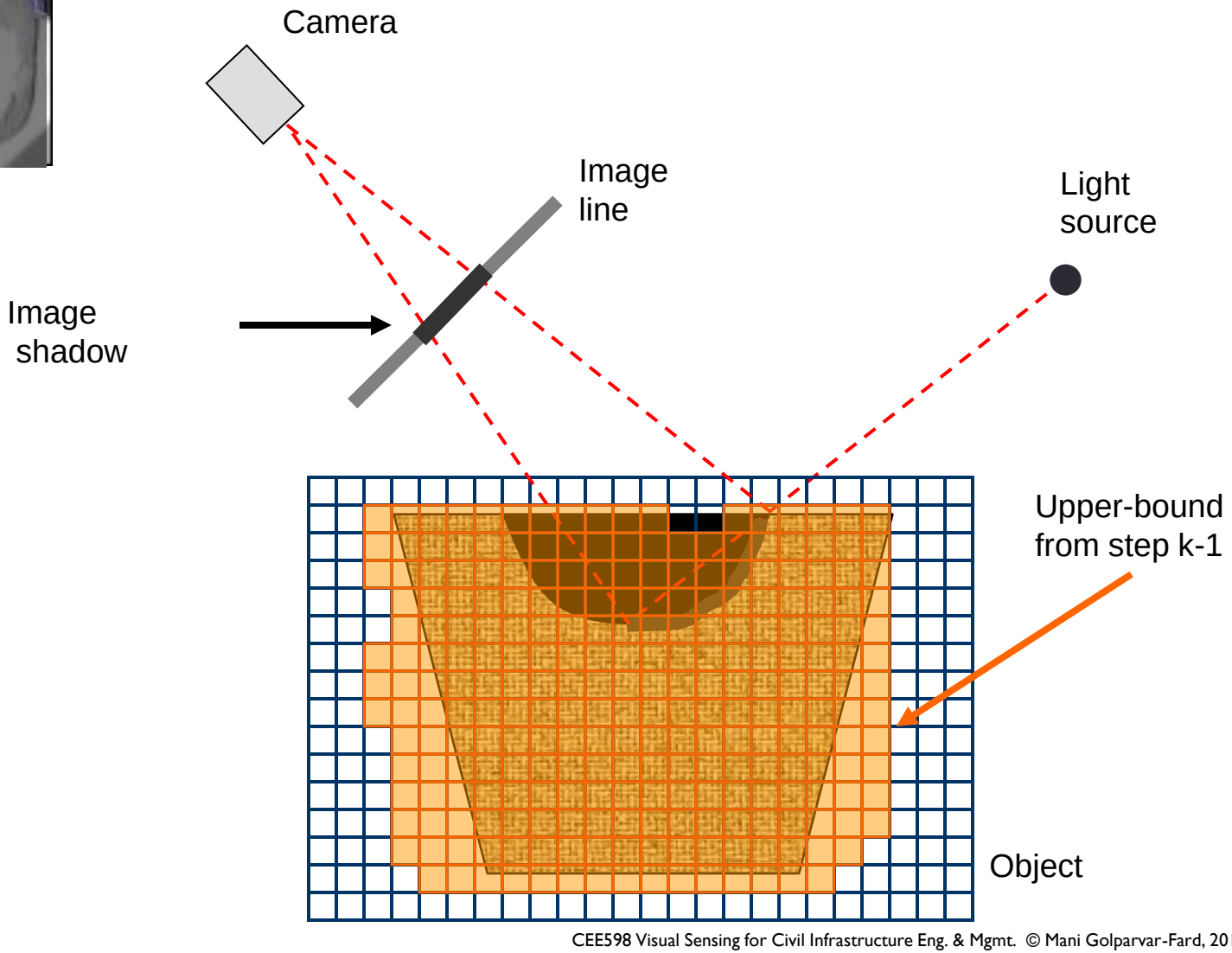
- Carving process always conservative

Proof of correctness

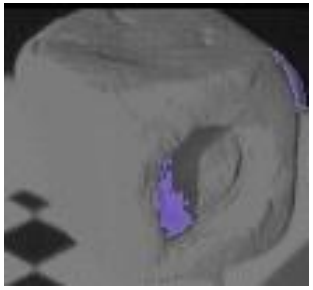
Algorithm: step k



Image

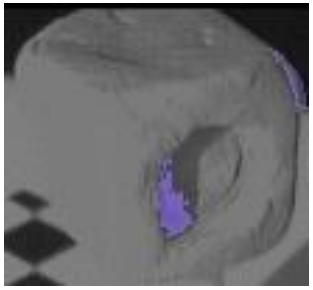


Algorithm: step k



Image

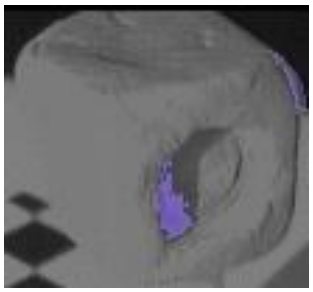
Algorithm: step k



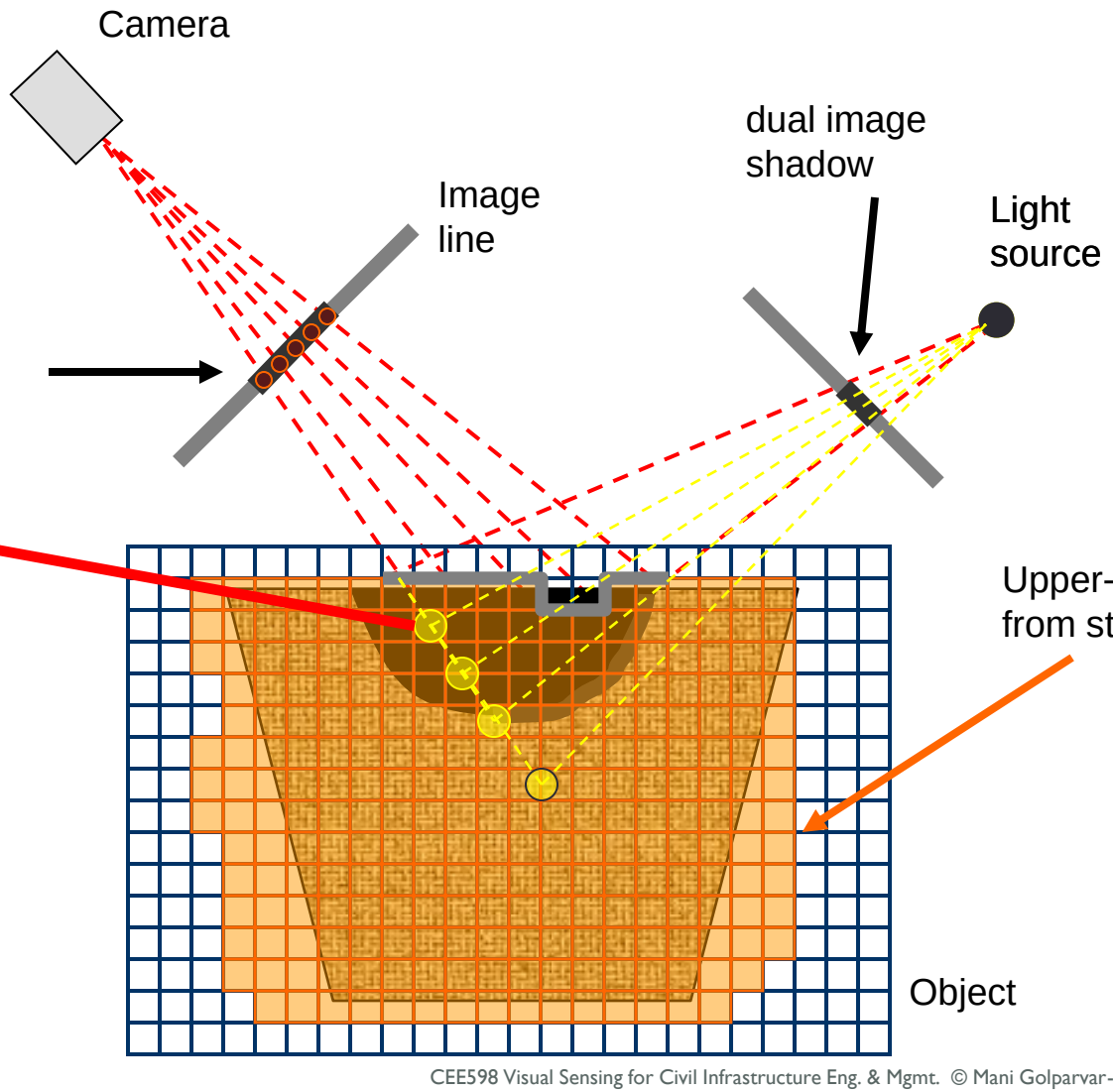
Image

Consistency:
A voxel must be projected into both image shadow and dual image shadow

Algorithm: step k



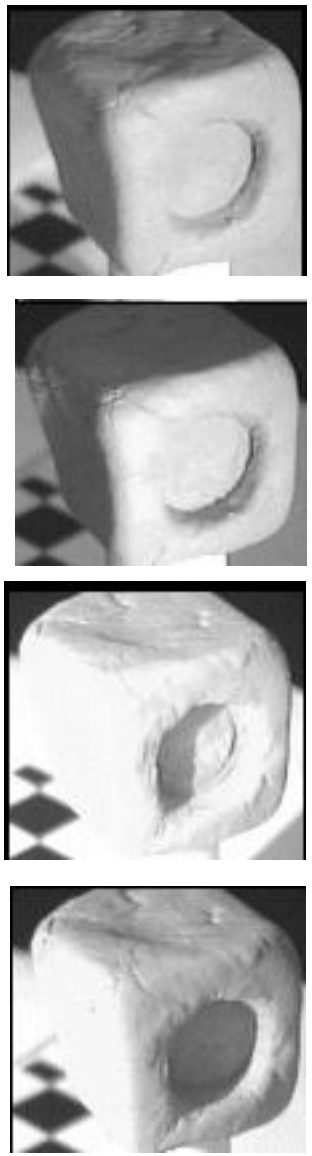
Image

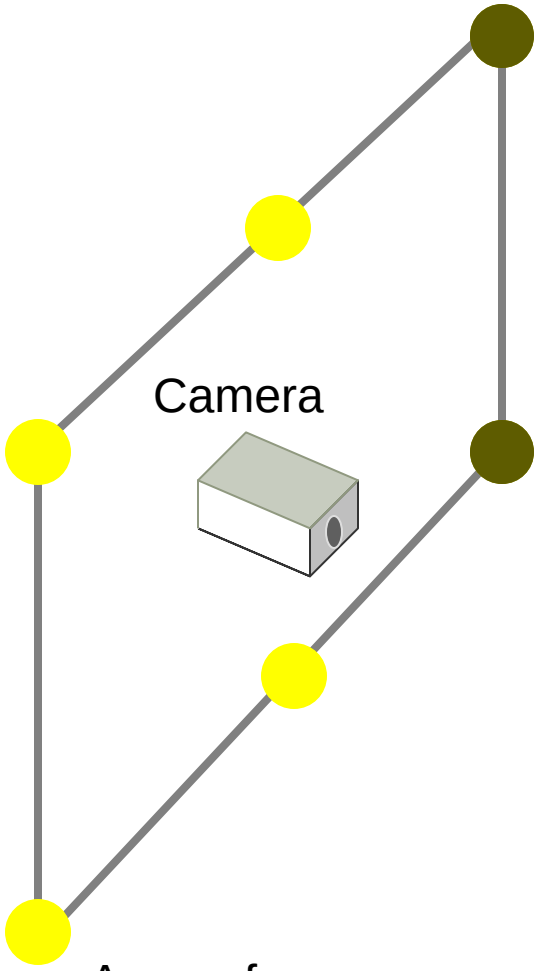


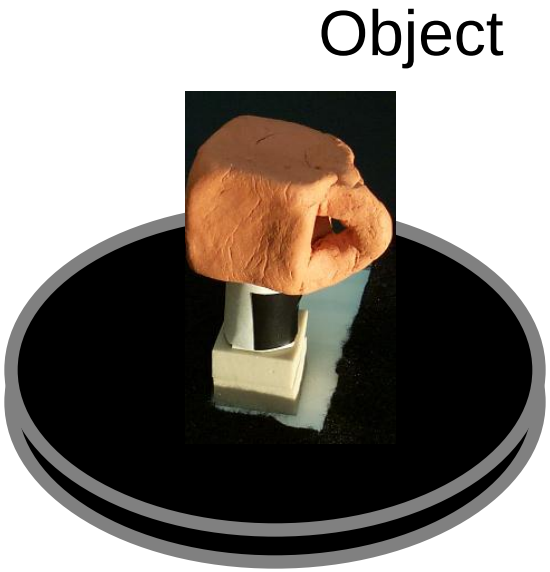
Complexity?

$O(2N^3)$

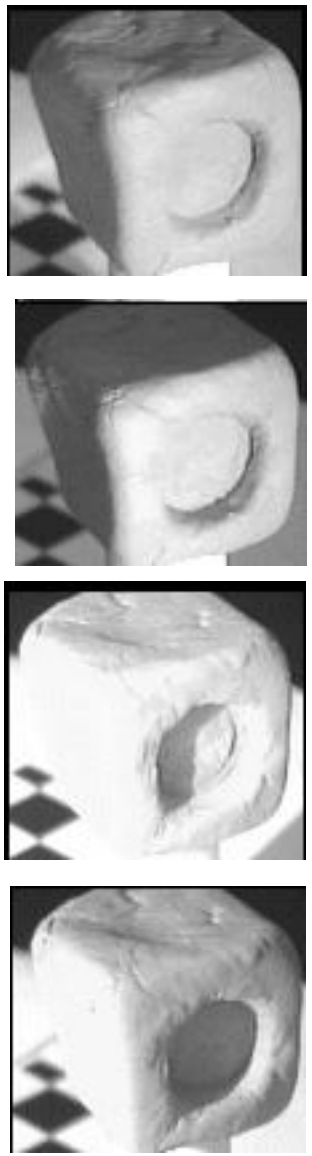
Shadow carving: the setup

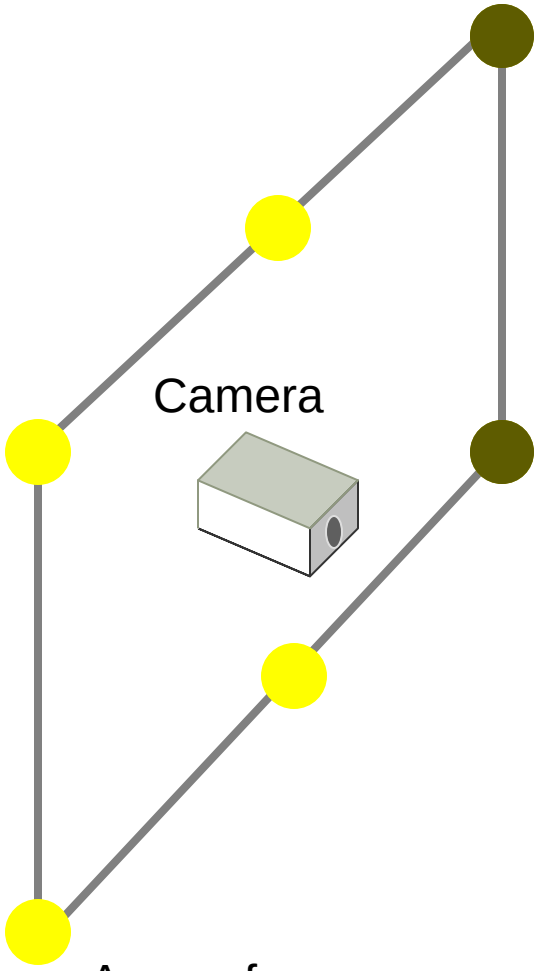


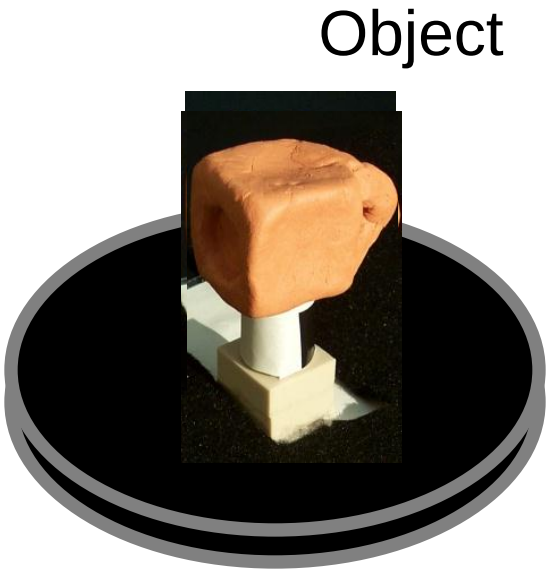




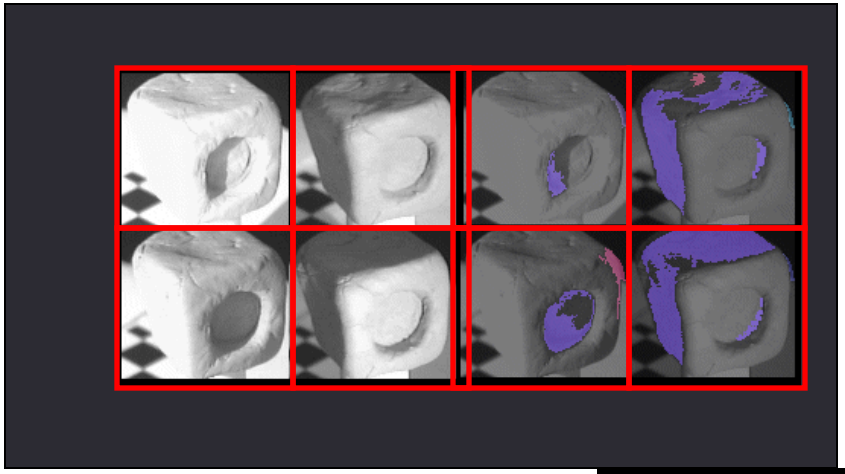
Shadow carving: the setup

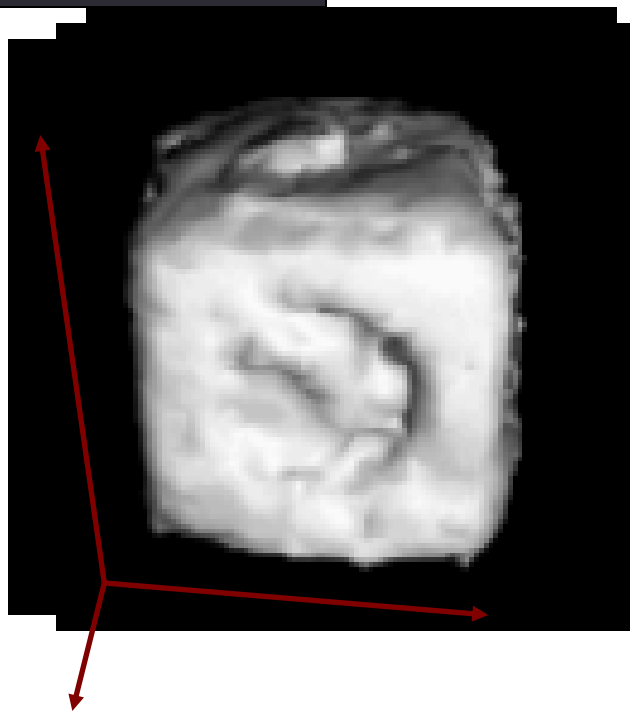






Experimental results

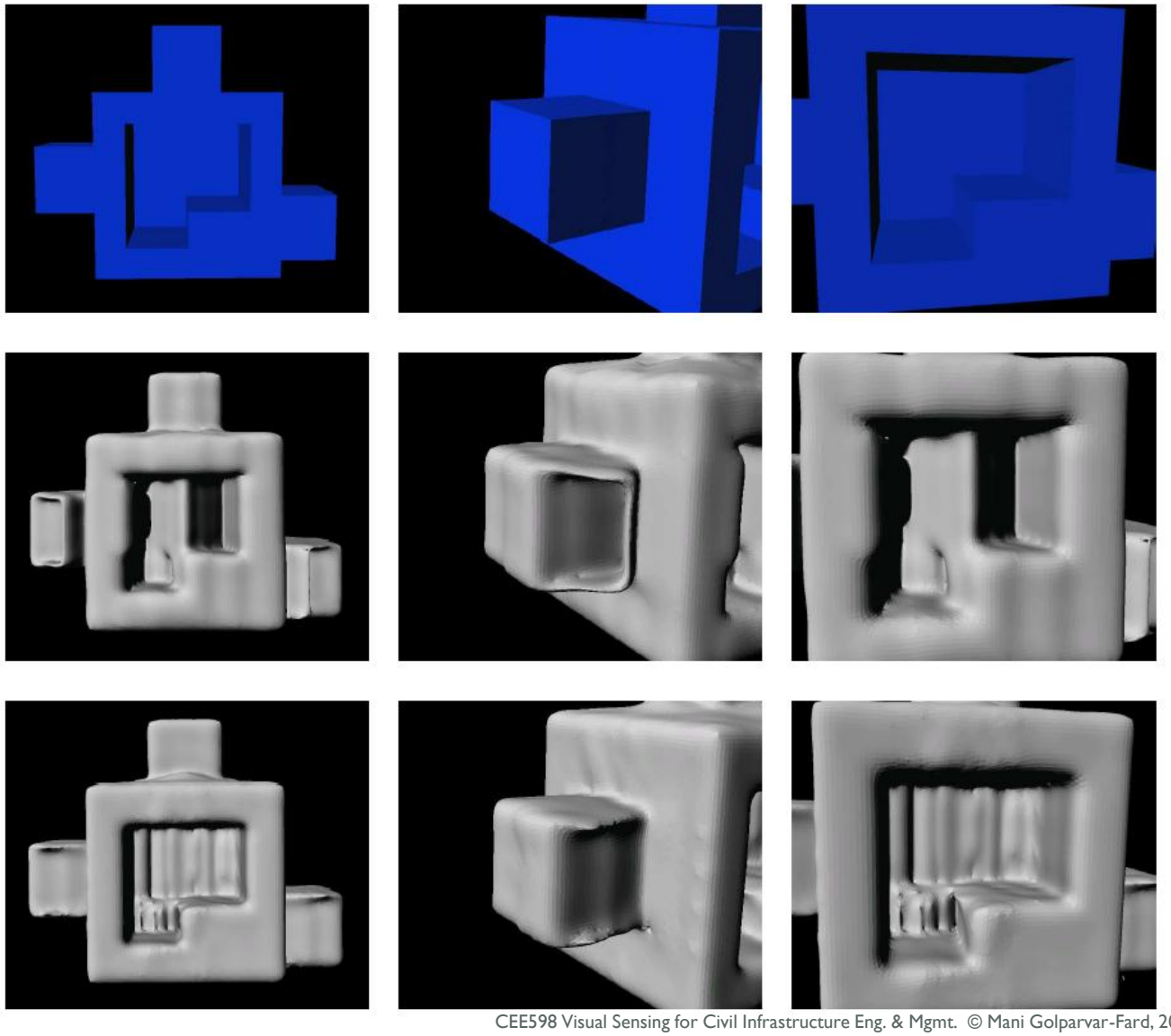




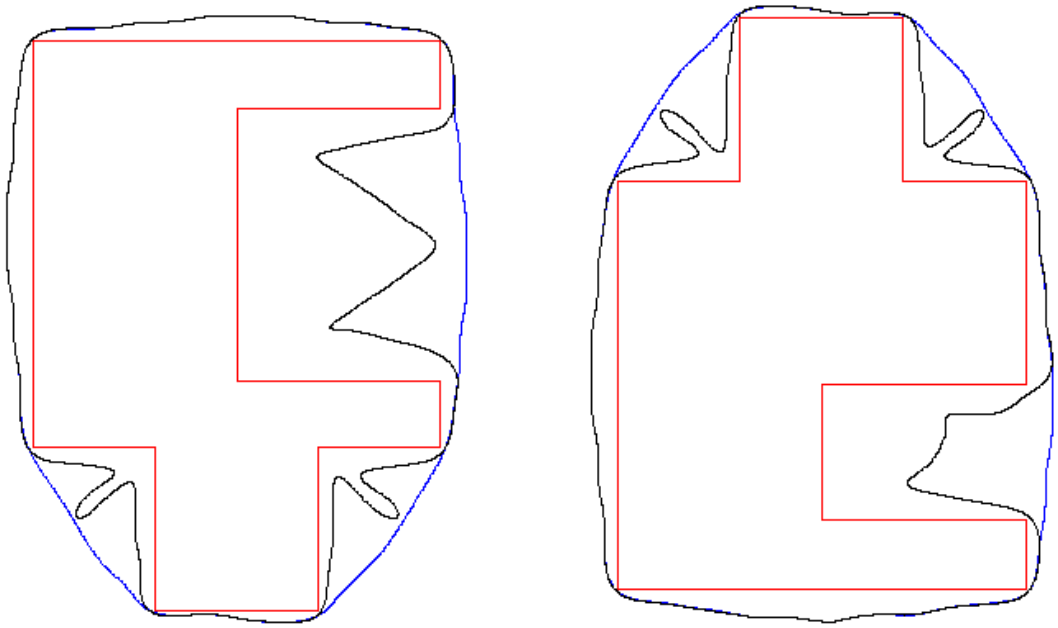
Simulating the system with 3D studio Max

- 24 positions
- 4 lights

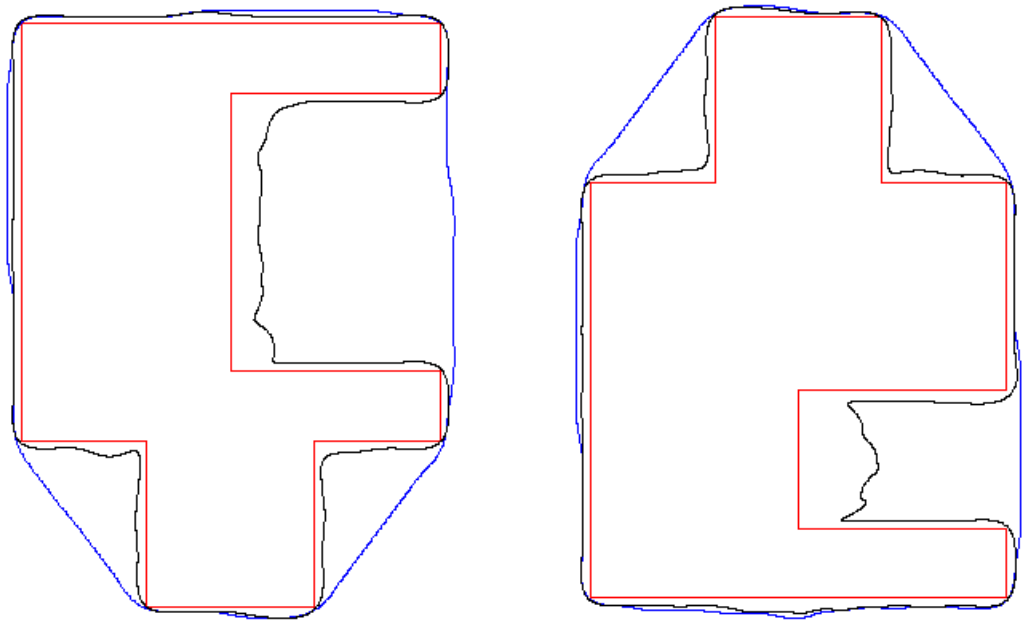
- 72 positions
- 8 lights



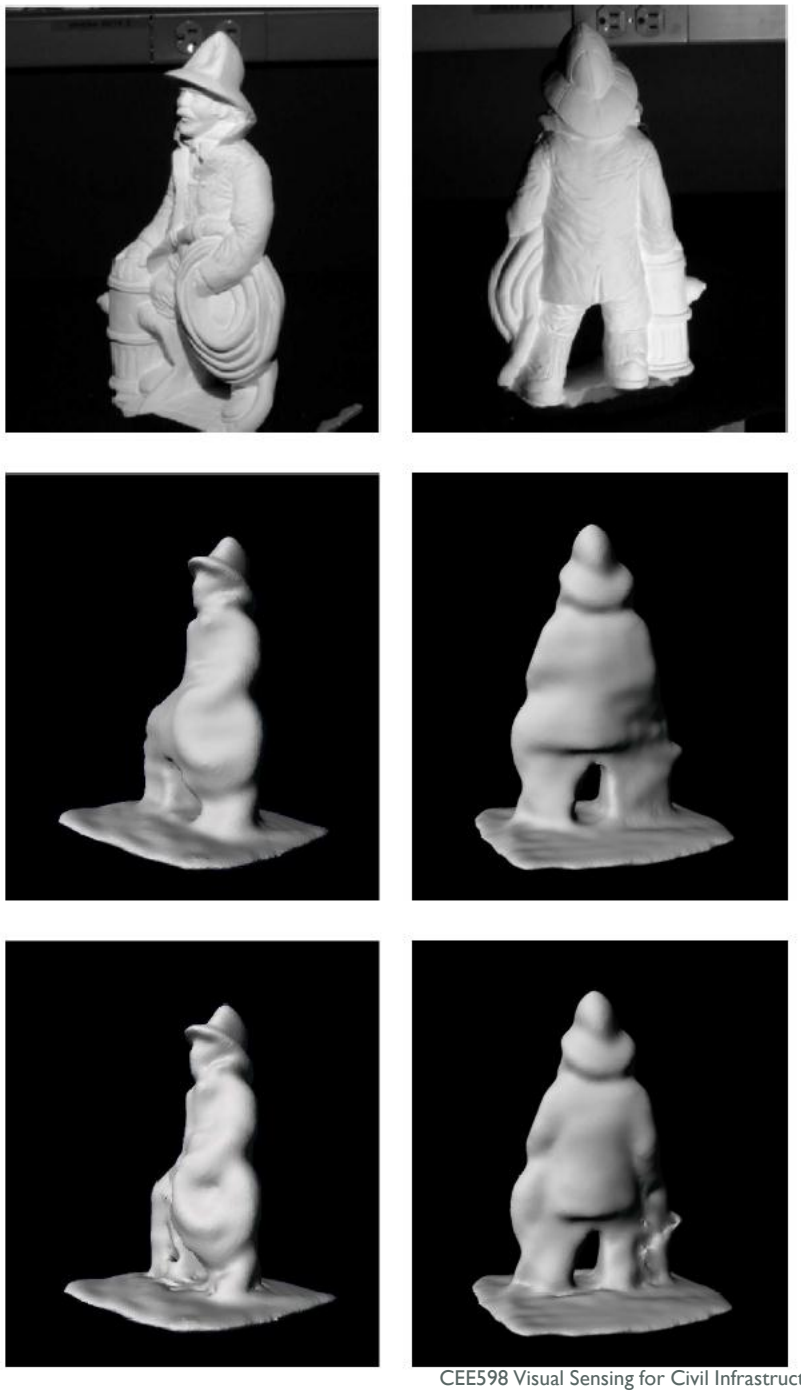
- 24 positions
- 4 lights

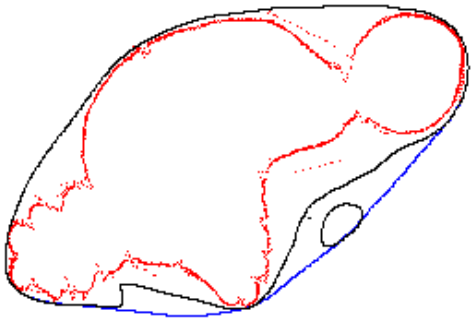


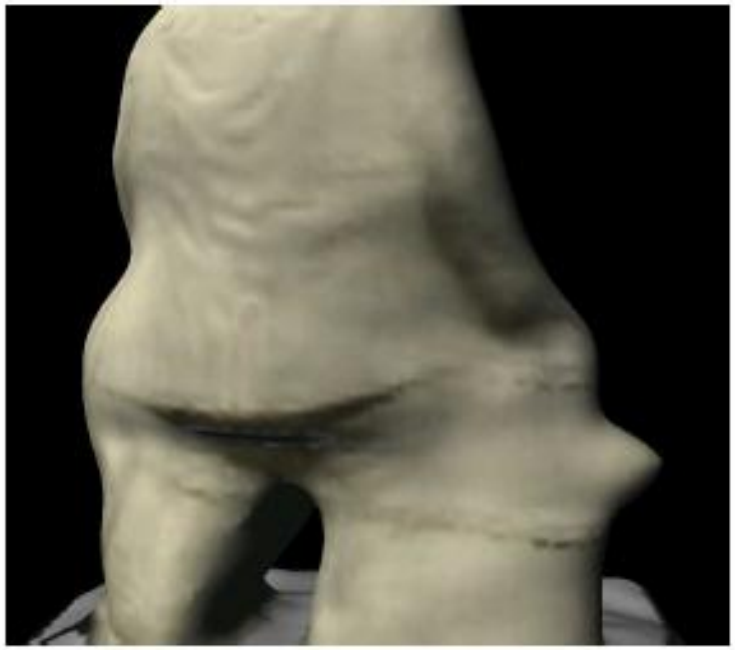
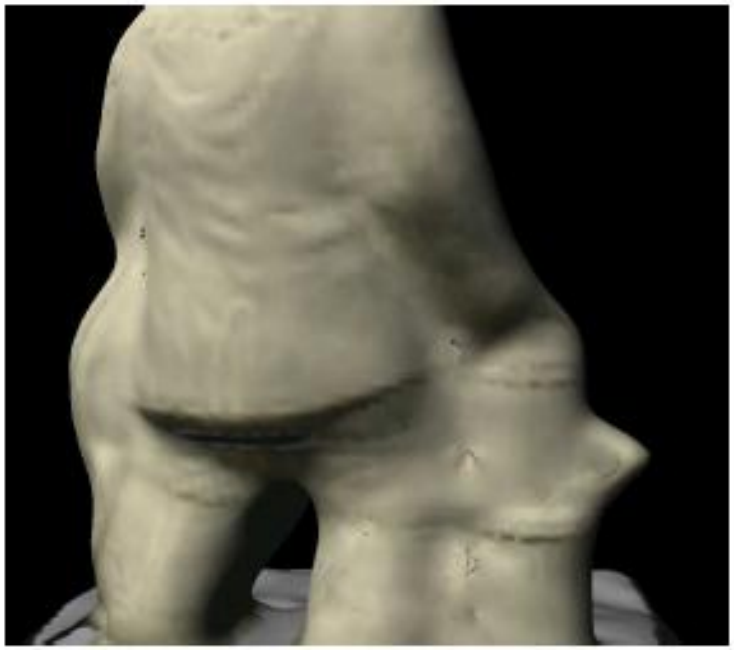
- 72 positions
- 8 lights



- 16 positions
- 4 lights





Shadow Carving: Summary

- Produces a conservative volume estimate
- Accuracy depending on view point and light source number
- Limitations with specular & low albedo regions

Outline

- Volumetric Reconstruction

- Definition
- Shape from Contours
- Shape from Shadows
- Voxel Coloring

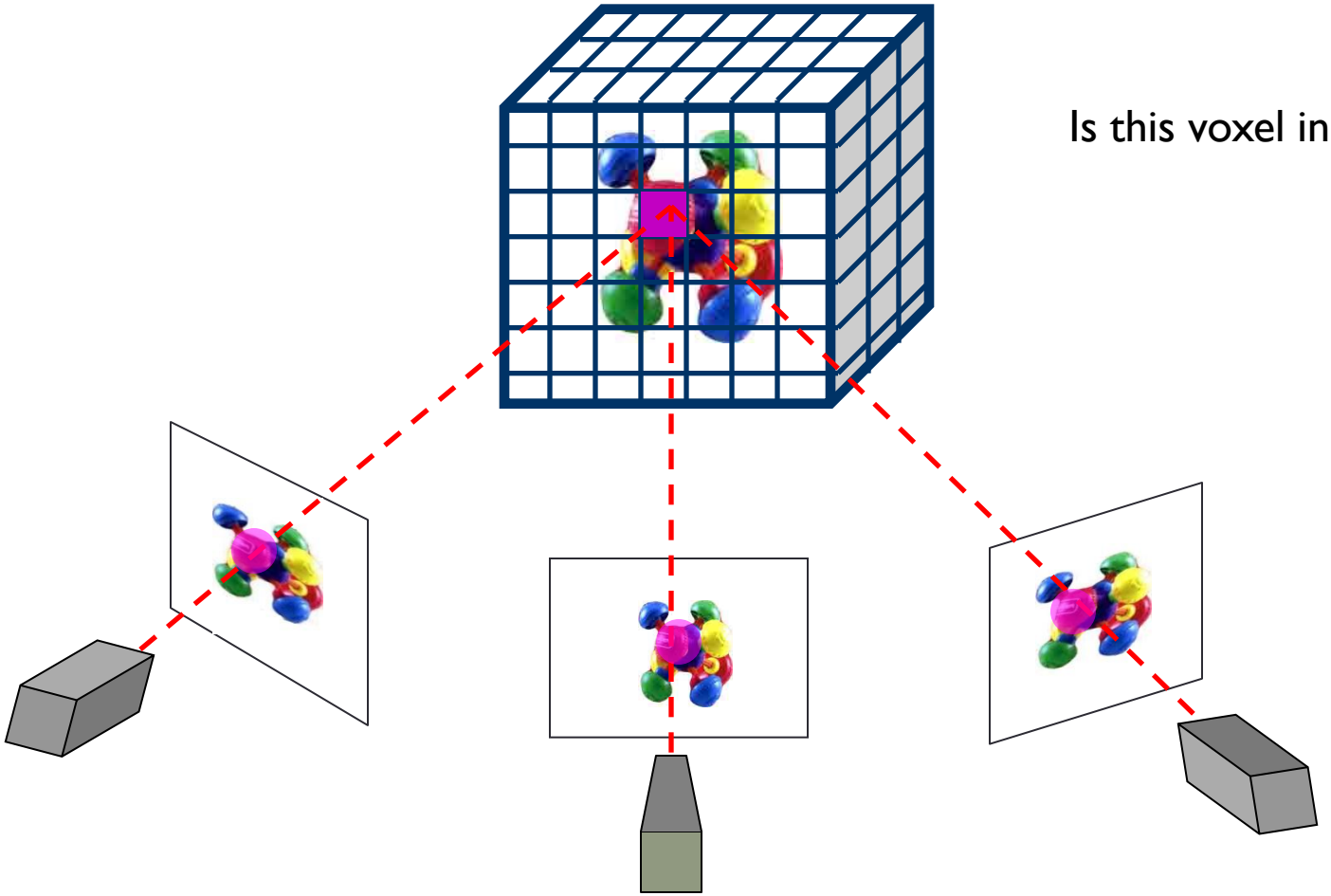
Some slides of this lectures are courtesy of prof. S. Savarese, and various other lecturers

Voxel Coloring

- Seitz & Dyer ('97)
- R. Collins (Space Sweep, '96)

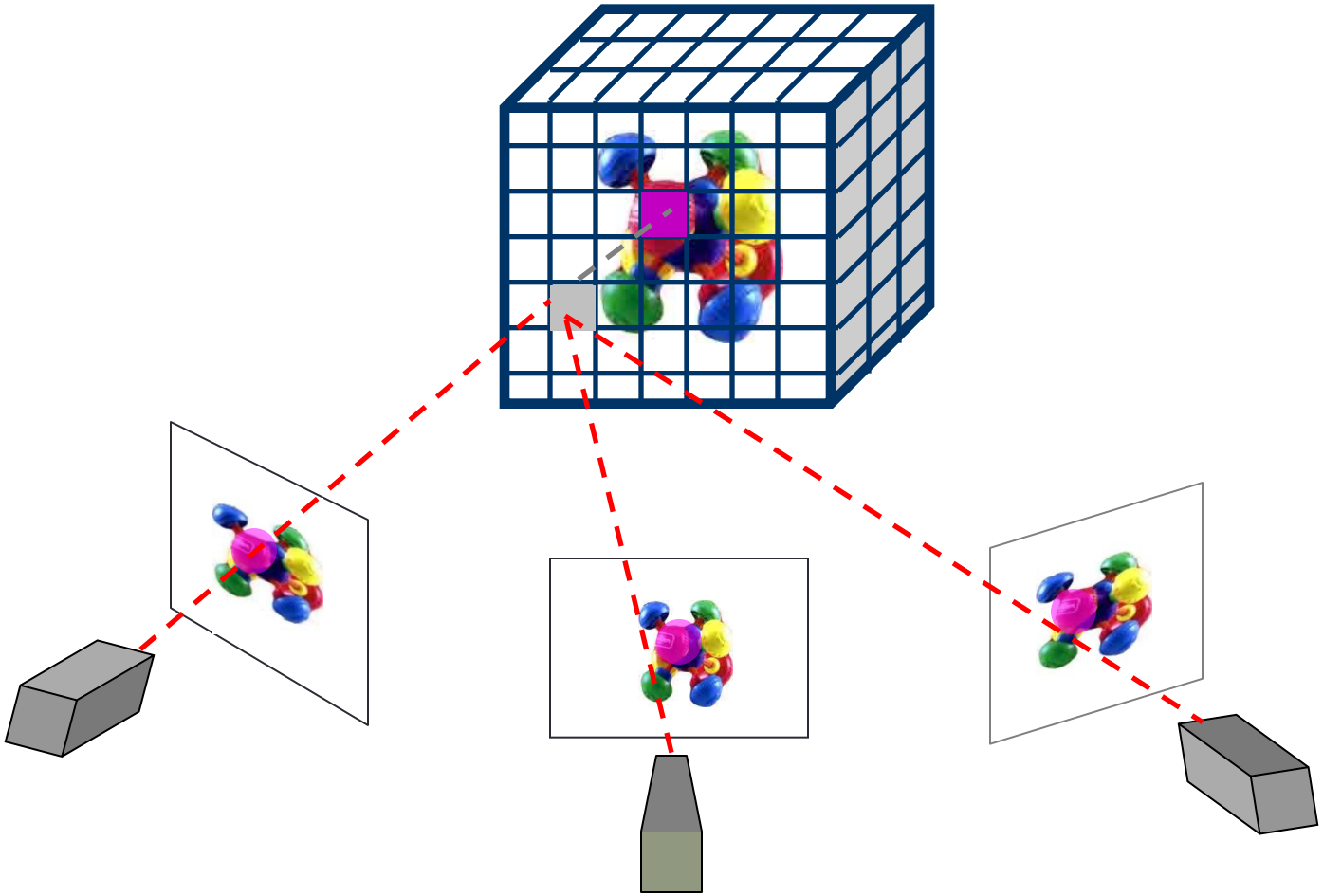
- color/photo-consistency
- Jointly model structure and appearance

Basic idea



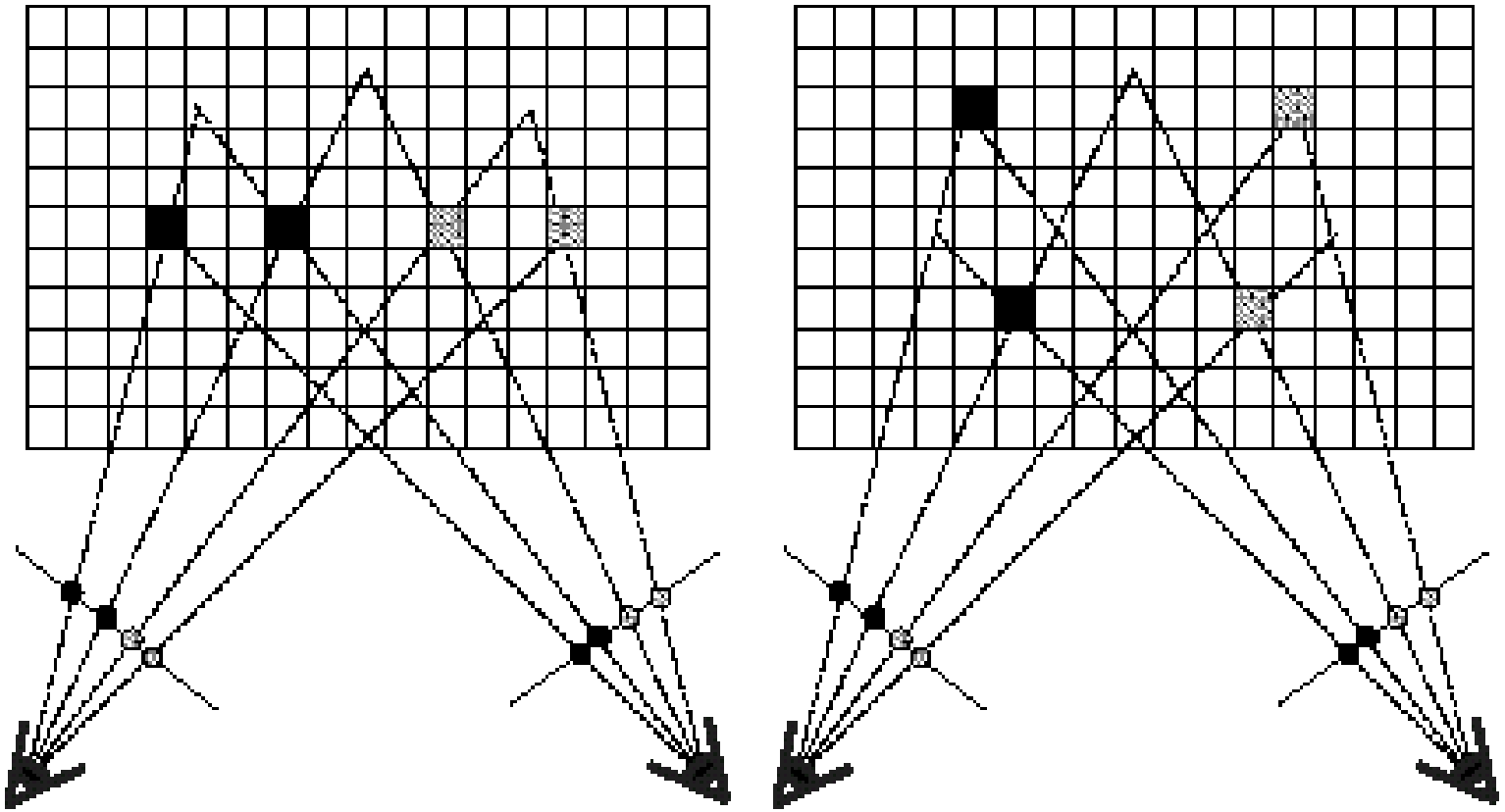
Is this voxel in or out?

Basic idea



Uniqueness

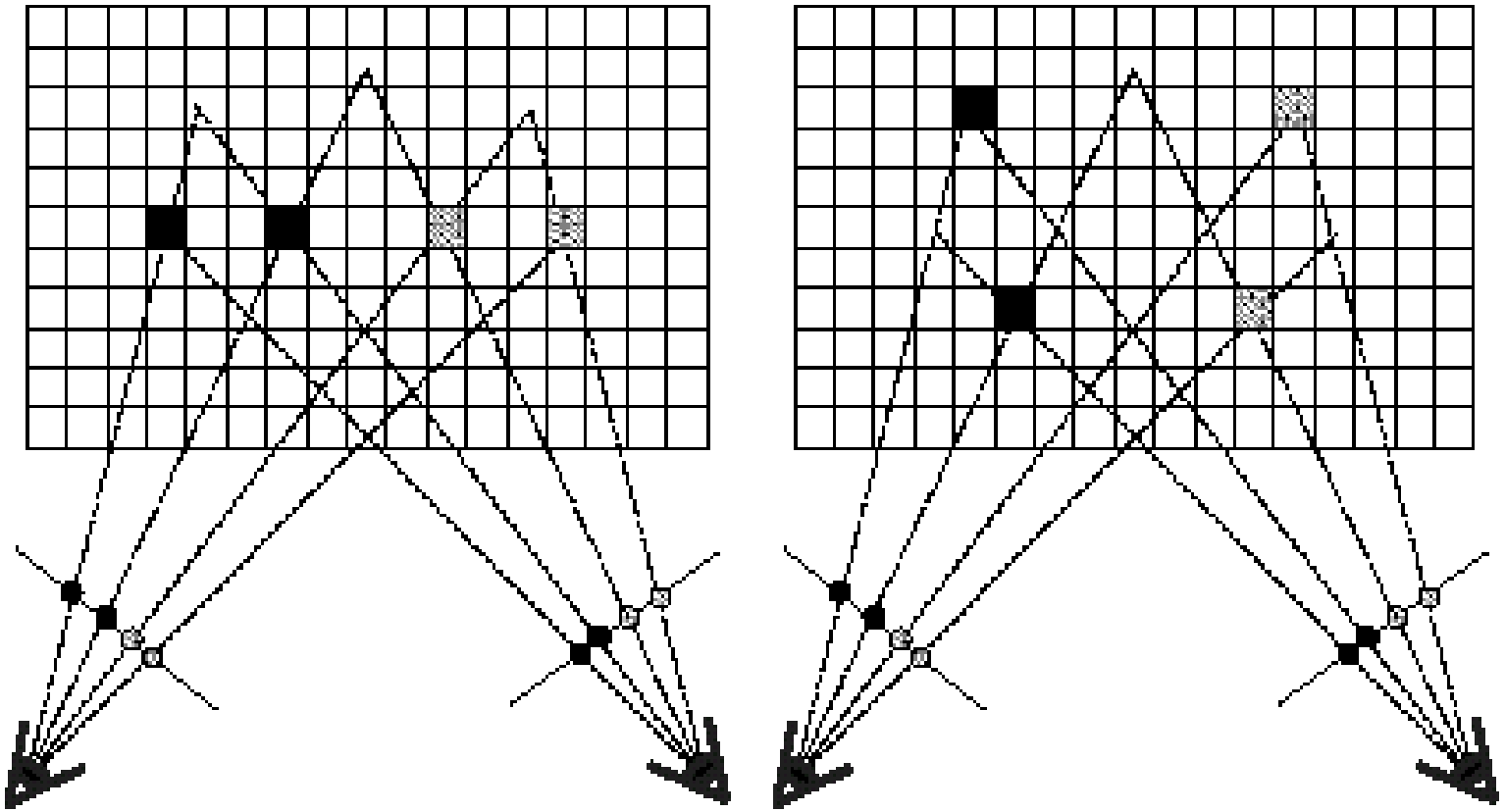
Multiple consistent scenes



Ijcv – seitz & dyer

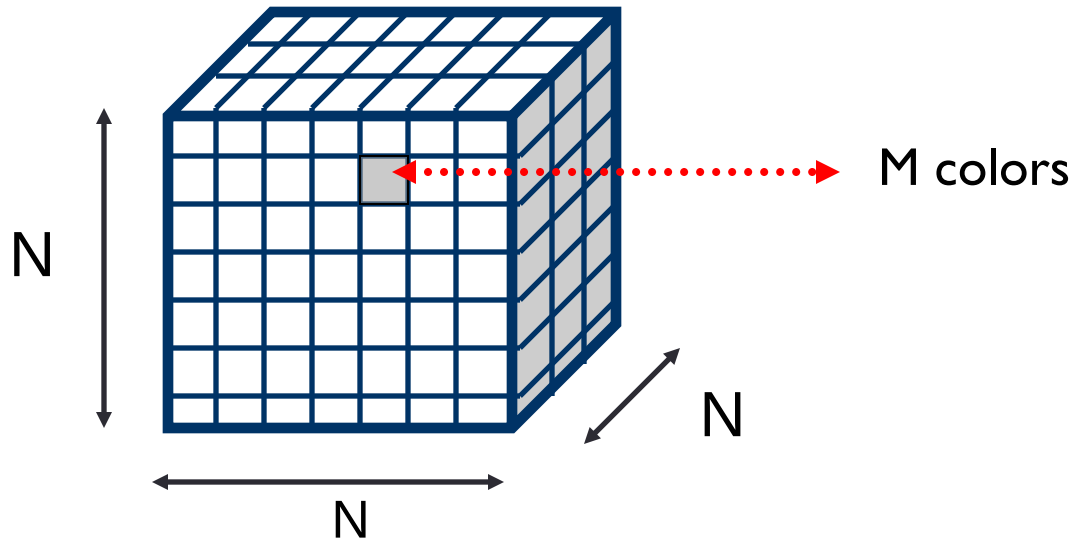
Uniqueness

Multiple consistent scenes



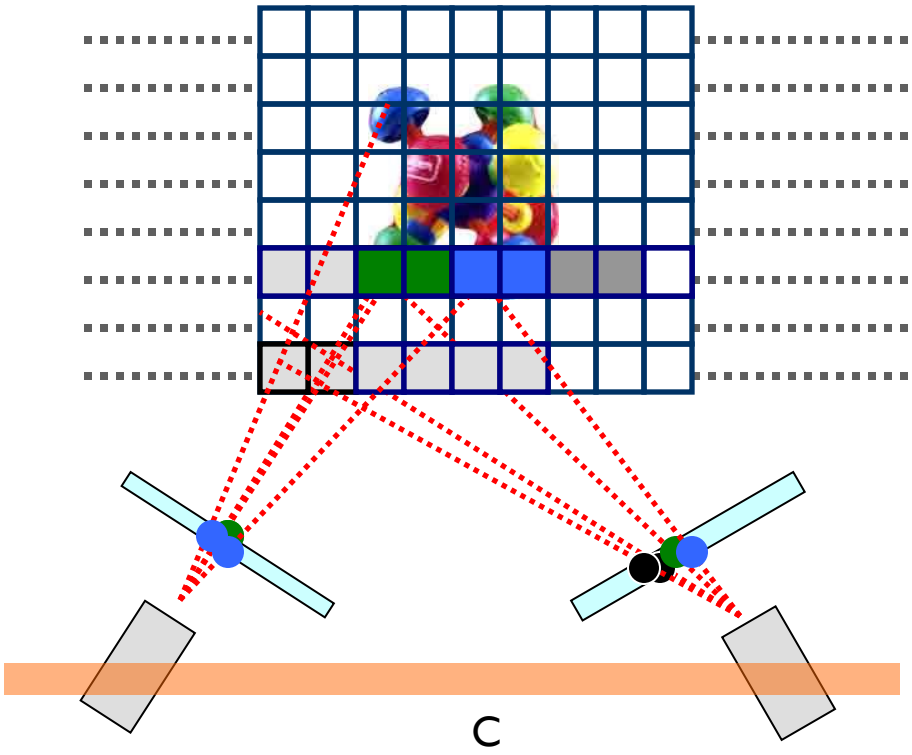
Ijcv – seitz & dyer

Tractability



- Combinatorial number possible assignments!
- Exhaustive search not feasible
- Model occlusions!
- Use *visibility constraint*

The algorithm



Algorithm complexity

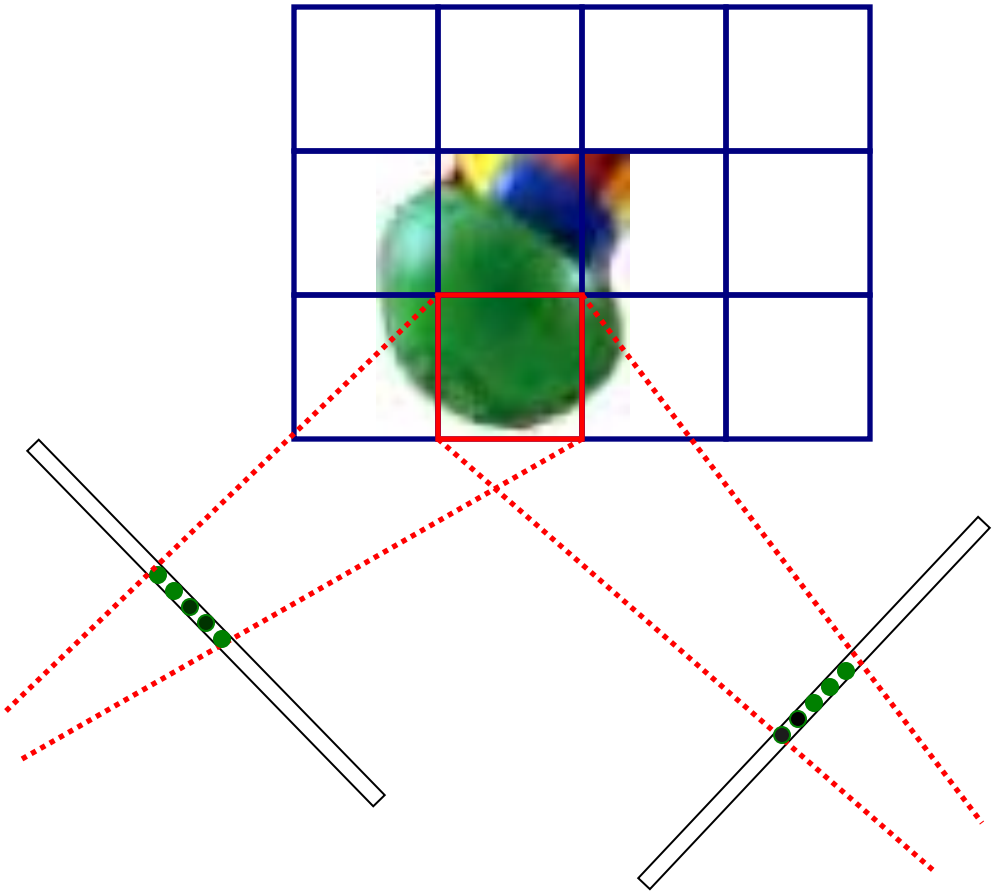
- Voxel coloring visits each N^3 voxels only once

- Project each voxel into *L* images

→ $O(L\ N^3)$

NOTE: not function of the number of colors

Photo-Consistency test

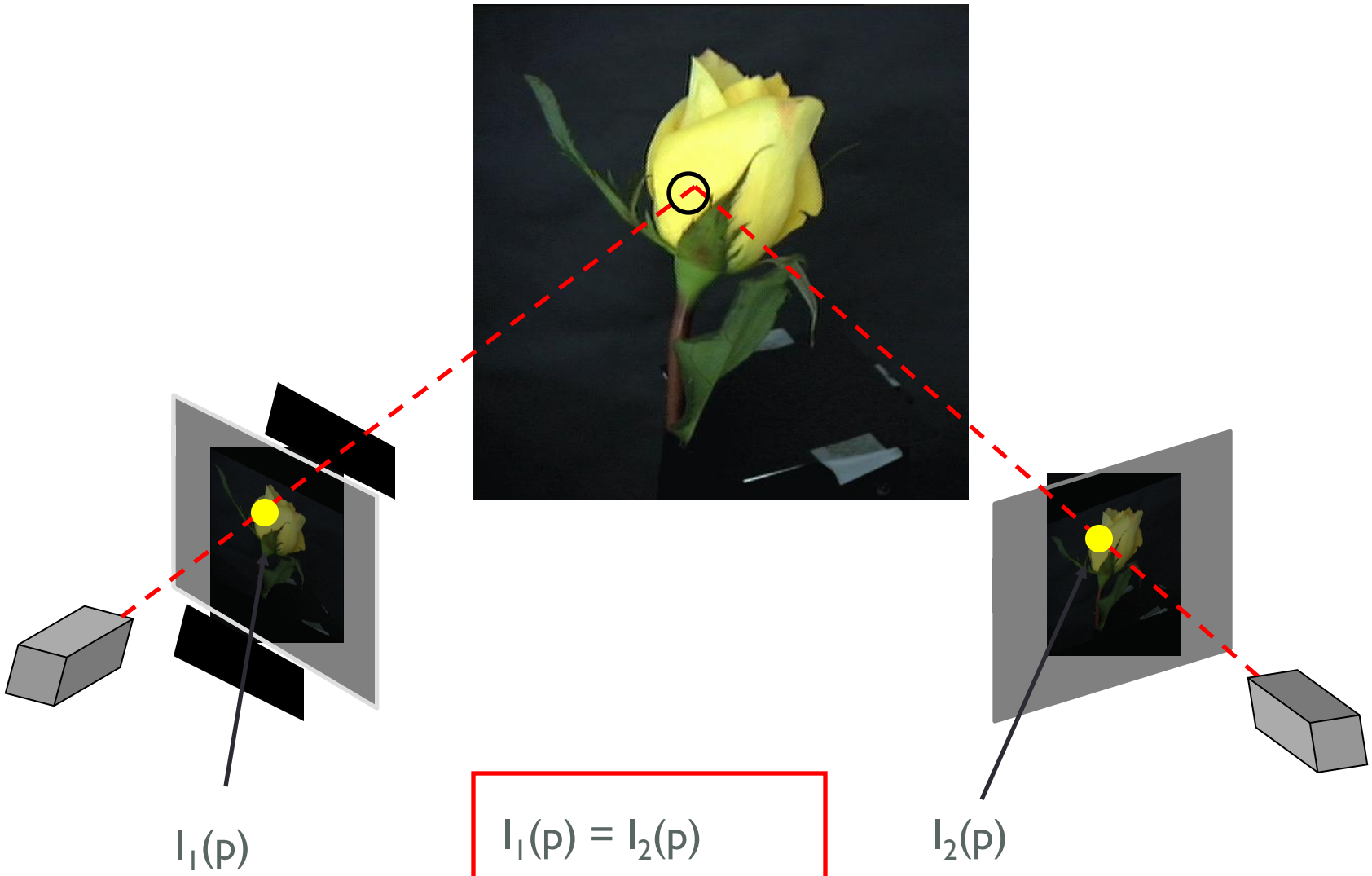


λ = corr (,)

Image 1 Image 2

If λ < Thresh ➔ voxel consistent

A critical assumption: Lambertian surfaces



Non Lambertian surfaces

Experimental results

Dinosaur

- 72 k voxels colored
- 7.6 M voxels tested
- 7 min to compute on a 250MHz

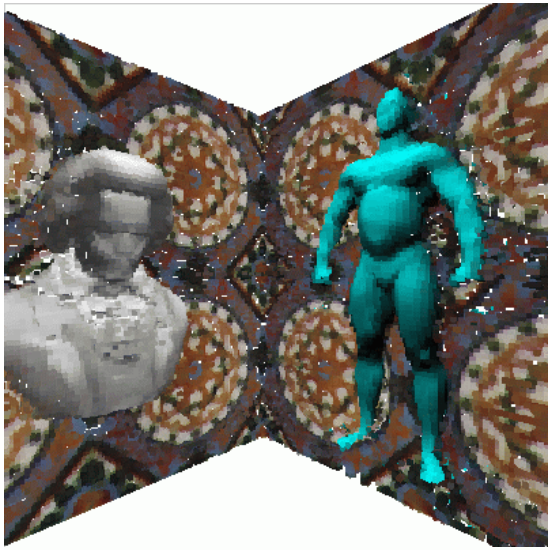
Experimental results

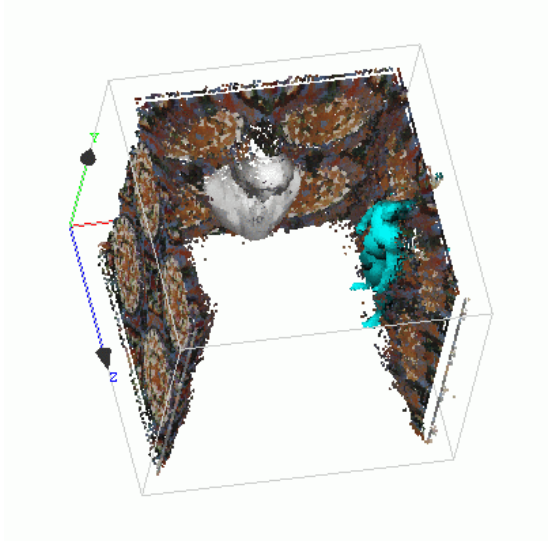
Flower

- 70 k voxels colored
- 7.6 M voxels tested
- 7 min to compute on a 250MHz

Experimental results

Room + weird people





Voxel coloring: conclusions

- Model intrinsic scene colors and texture
- No assumptions on scene topology
- Challenge with Varying Camera Locations and Viewpoints

Voxel coloring: conclusions

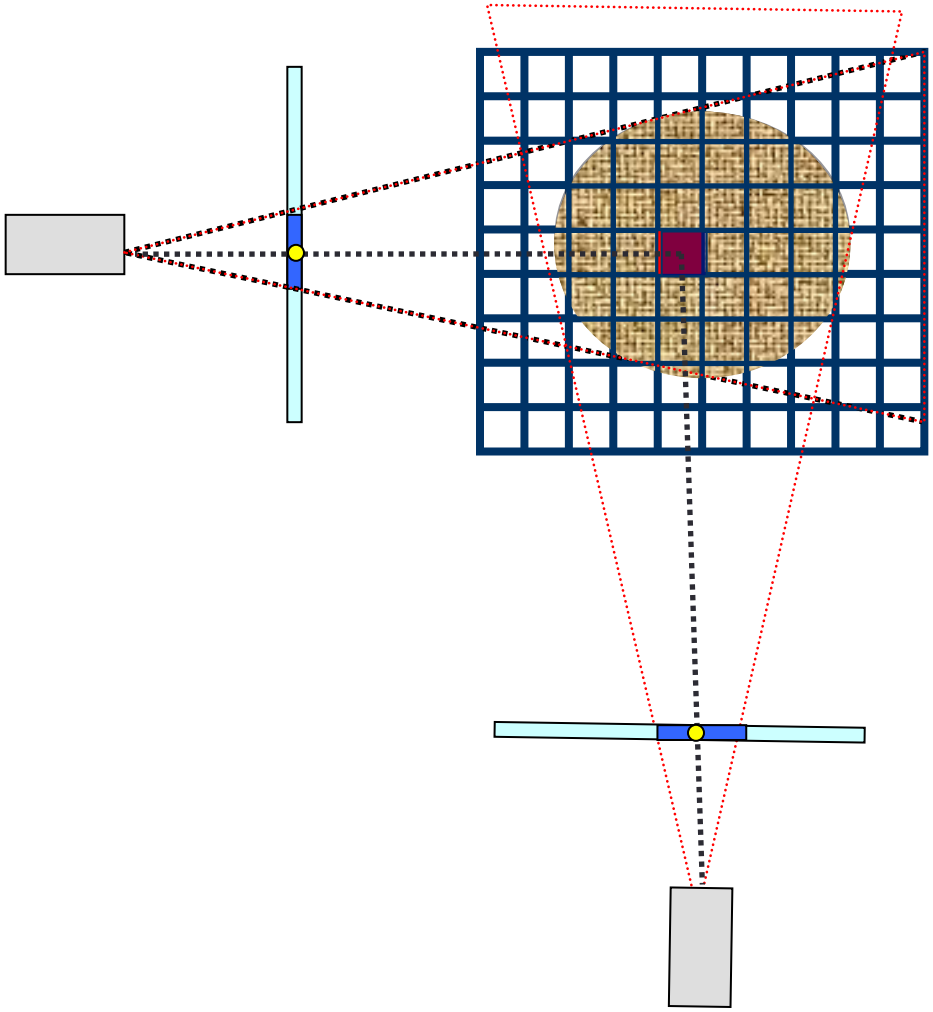
Good things:

- model intrinsic scene colors and texture
- no assumptions on scene topology

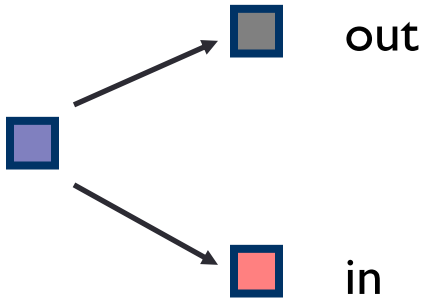
Limitations:

- Constrained camera positions
- Lambertian assumption

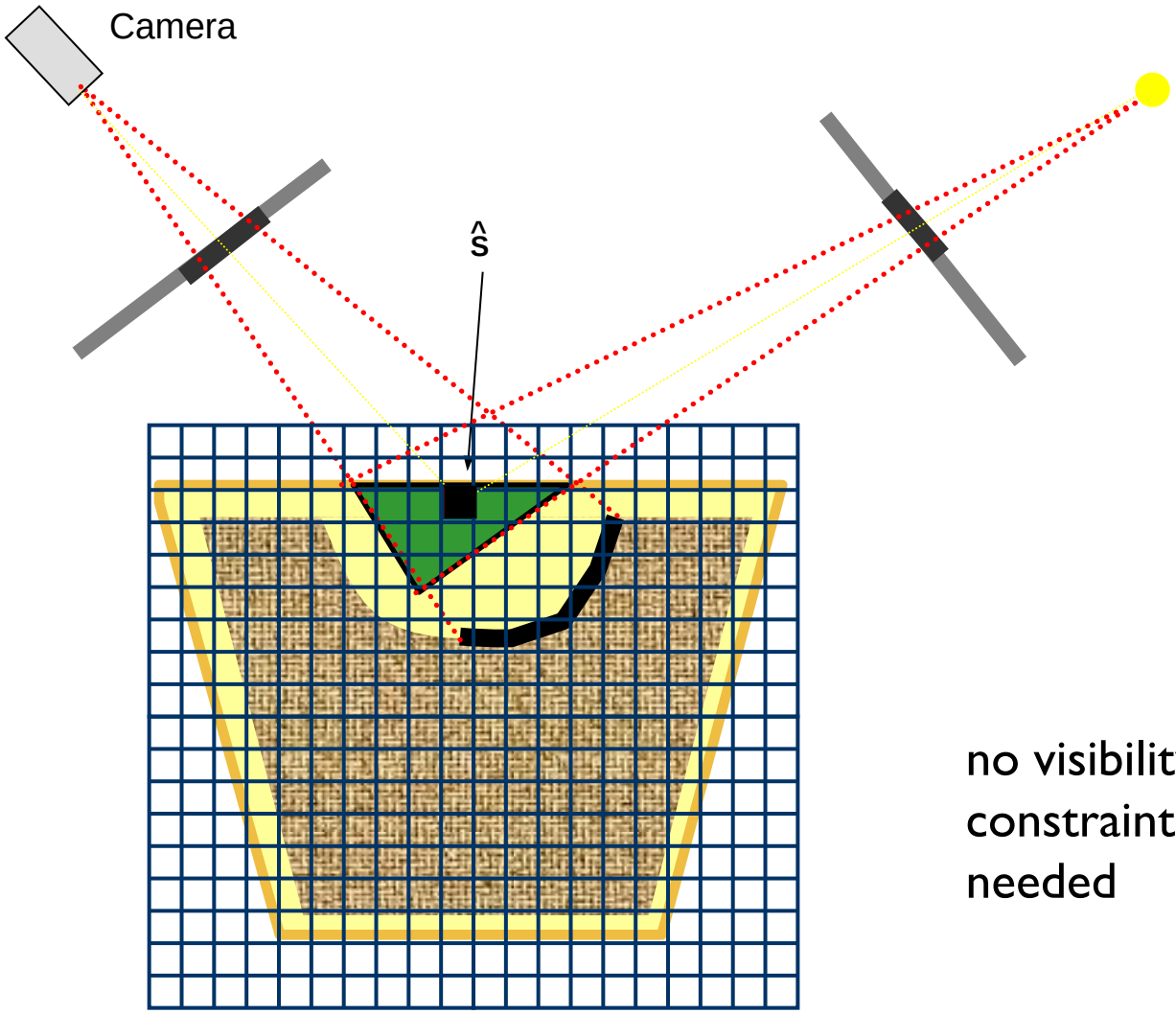
Space Carving



❖ Space carving is a binary voxel coloring



Shadow Carving



no visibility constraint needed

Further contributions

A Theory of Space Carving

(Kutulakos & Seitz '99)

Voxel coloring in more general framework

No restrictions on camera position

Probabilistic Space Carving

(Broadhurst & Cipolla, ICCV 2001)

(Bhotika, Kutulakos et. al, ECCV 2002)