UNIVERSITÄT BONN

Juergen Gall

Stereo

MA-INF 2201 - Computer Vision

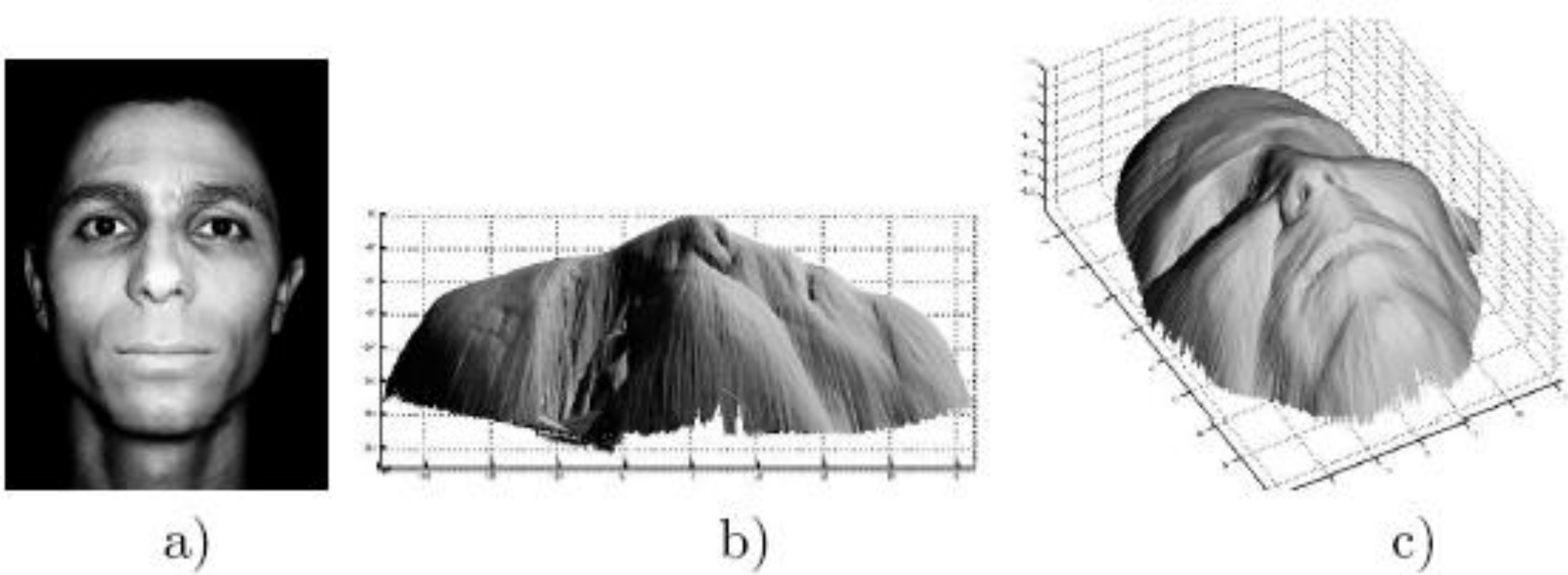
WS24/25

3D Reconstruction

What cues help us to perceive 3d shape and depth?

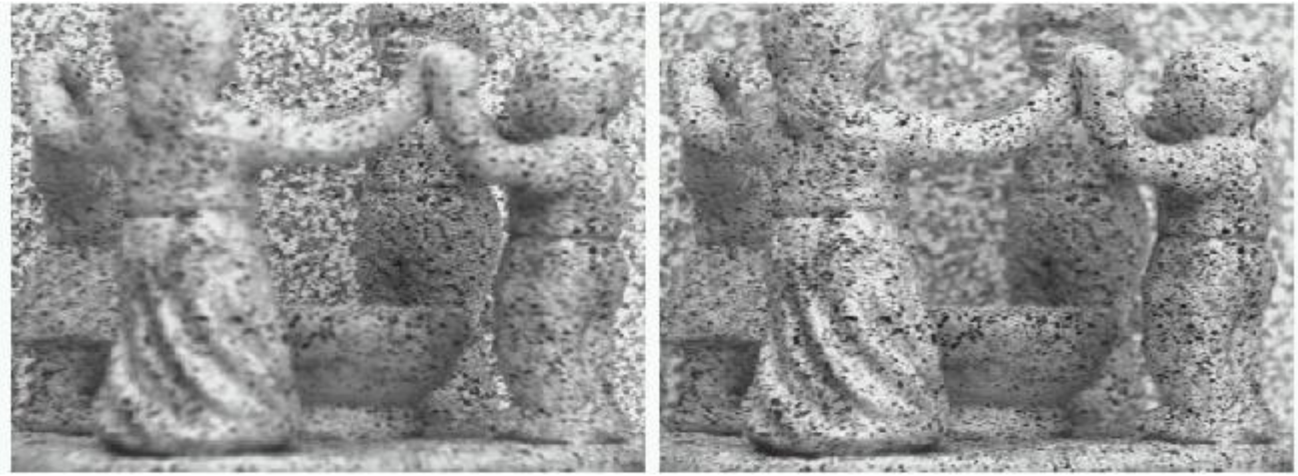
Kristen Grauman

Shading

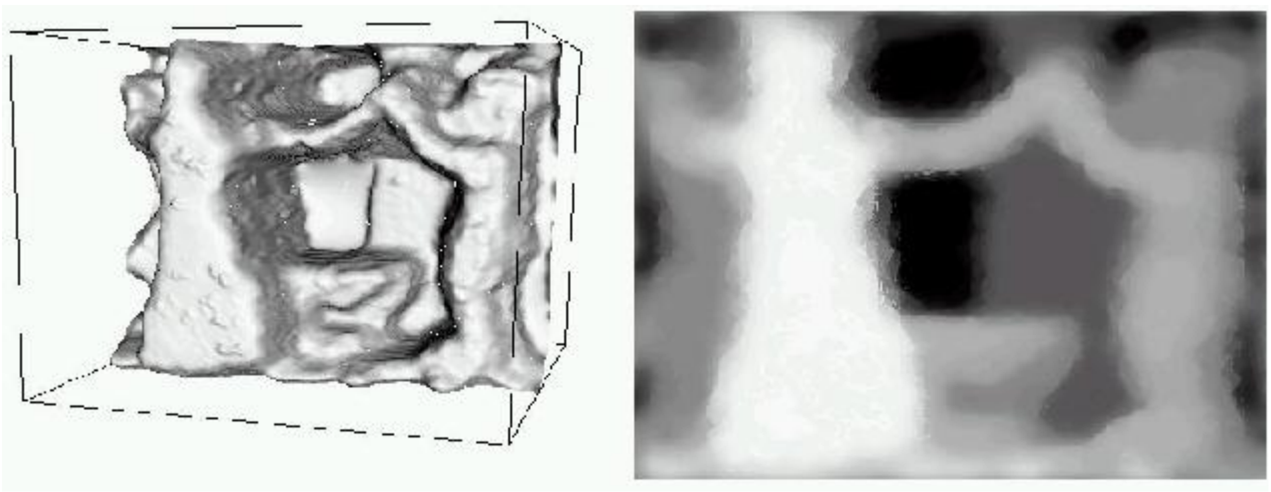


[Figure from Prados & Faugeras 2006]

Focus/defocus



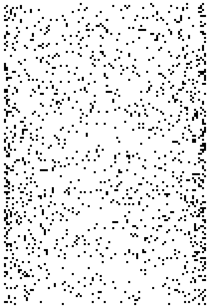
Images from same point of view, different camera parameters



3d shape / depth estimates

[figs from H. Jin and P. Favaro, 2002]

Motion



Figures from L. Zhang

http://www.brainconnection.com/teasers/?main=illusion/motion-shape

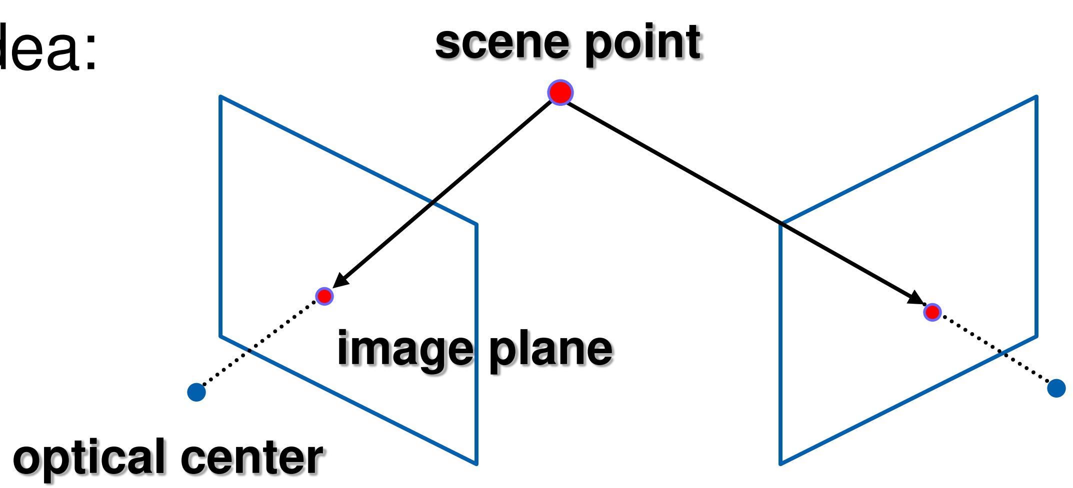
Estimating scene shape

"Shape from X": Shading, Texture, Focus, Motion…

Stereo:

- shape from "motion" between two views
- infer 3d shape of scene from two (multiple) images from different viewpoints

Main idea:

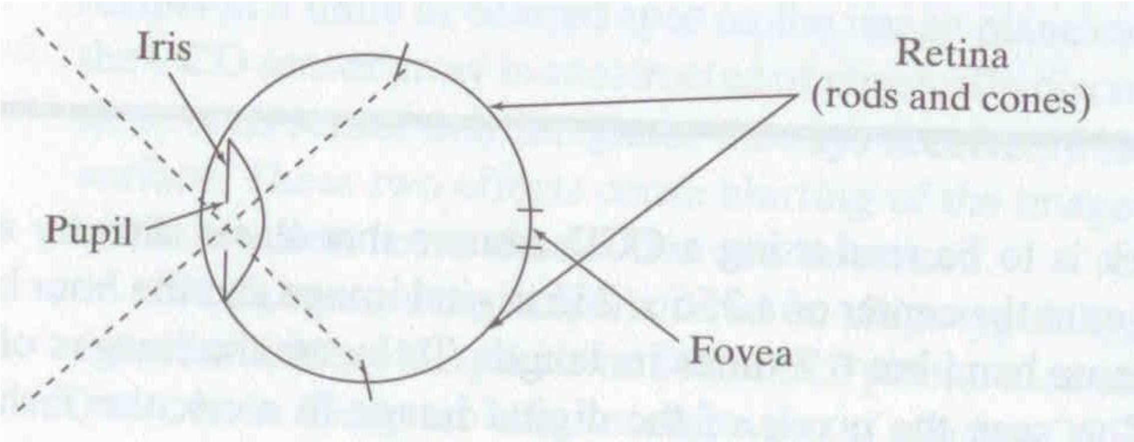


Kristen Grauman

Human eye

UNIVERSITÄT BONN

Rough analogy with human visual system:



Pupil/Iris – control amount of light passing through lens

Retina - contains sensor cells, where image is formed

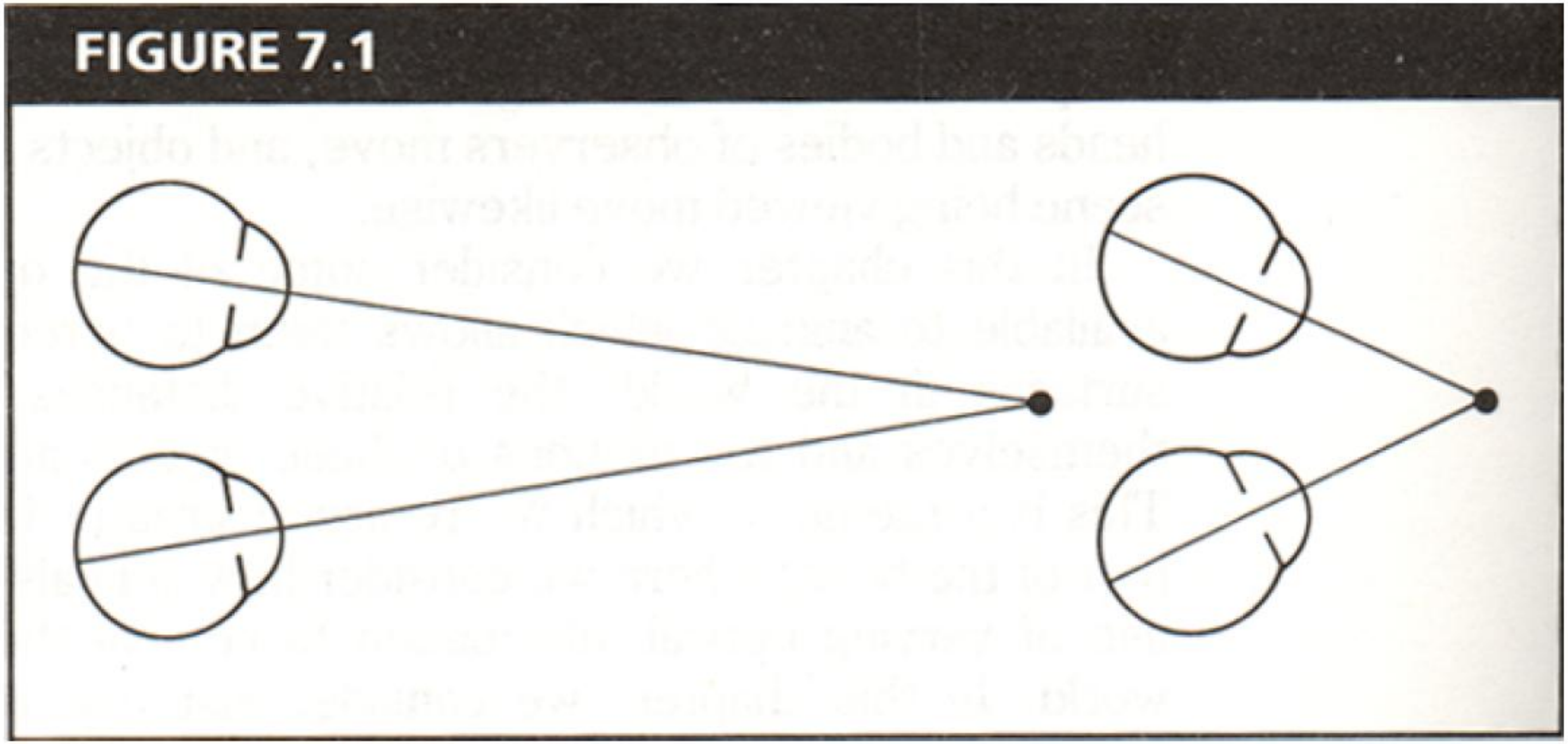
Fovea – highest concentration of cones

Fig from Shapiro and Stockman

Kristen Grauman

Human stereopsis: disparity

UNIVERSITÄT BONN



From Bruce and Green, Visual Perception, Physiology, Psychology and Ecology

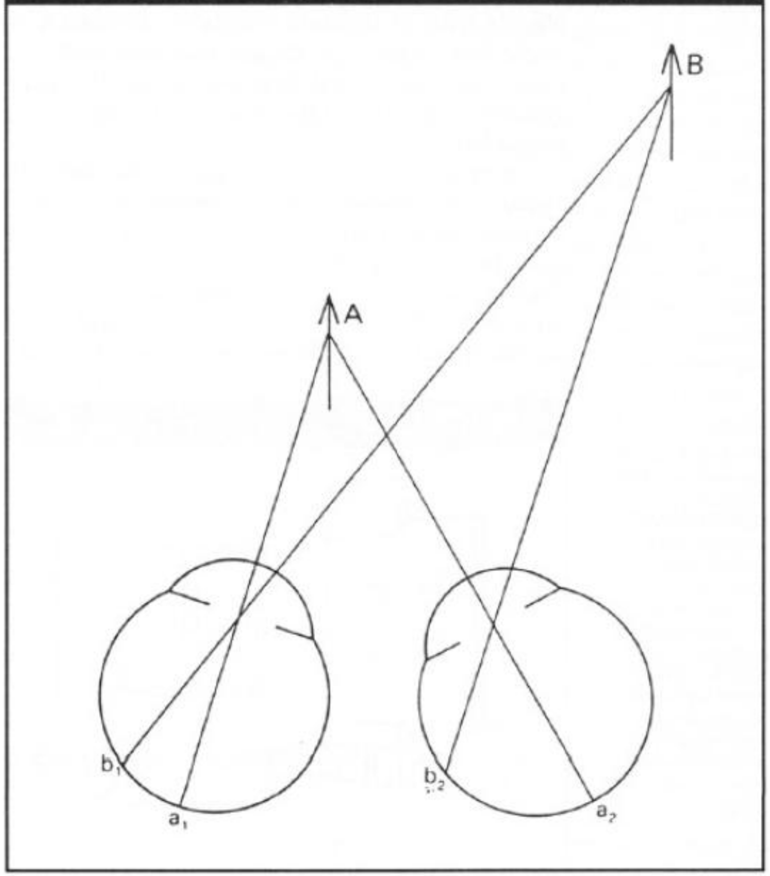
- Human eyes fixate on point in space – rotate so that corresponding images form in centers of fovea

Kristen Grauman

Human stereopsis: disparity

UNIVERSITÄT BONN

FIGURE 7.3



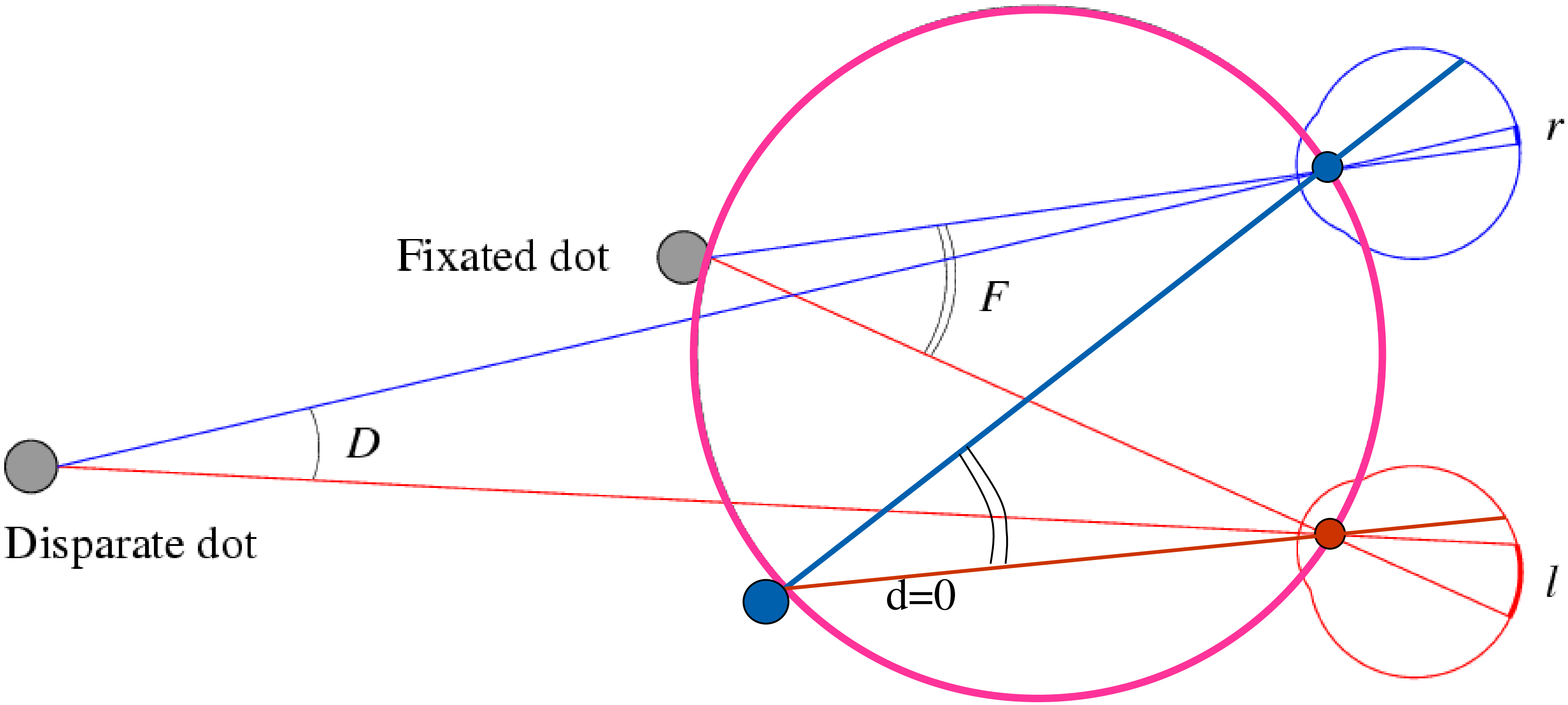
From Bruce and Green, Visual Perception, Physiology, Psychology and Ecology

Disparity occurs when eyes fixate on one object; others appear at different visual angles

Adapted from David Forsyth, UC Berkeley

Kristen Grauman

Human stereopsis: disparity



Disparity: $d = r\text{-}l = D\text{-}F.$

Forsyth & Ponce

http://www.well.com/~jimg/stereo/stereo_list.html

Kristen Grauman

Stereo vision

Two cameras, simultaneous views

Single moving camera and static scene

Kristen Grauman

Binocular stereo

Given a calibrated binocular stereo pair, fuse it to produce a depth image

image 1

image 2

Dense depth map

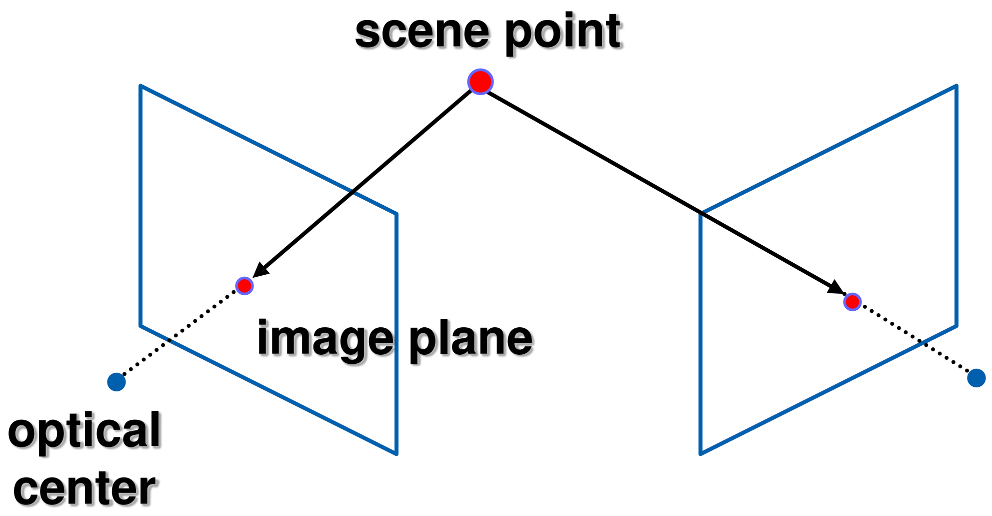
Lazebnik

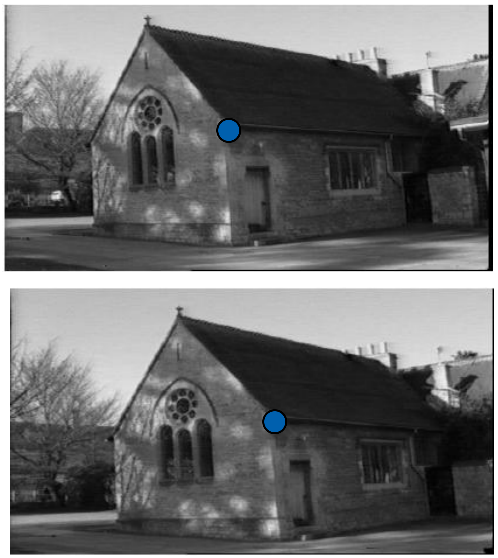
Estimating depth with stereo

Stereo: shape from "motion" between two views

We'll need to consider:

- Info on camera pose ("calibration")
- Image point correspondences

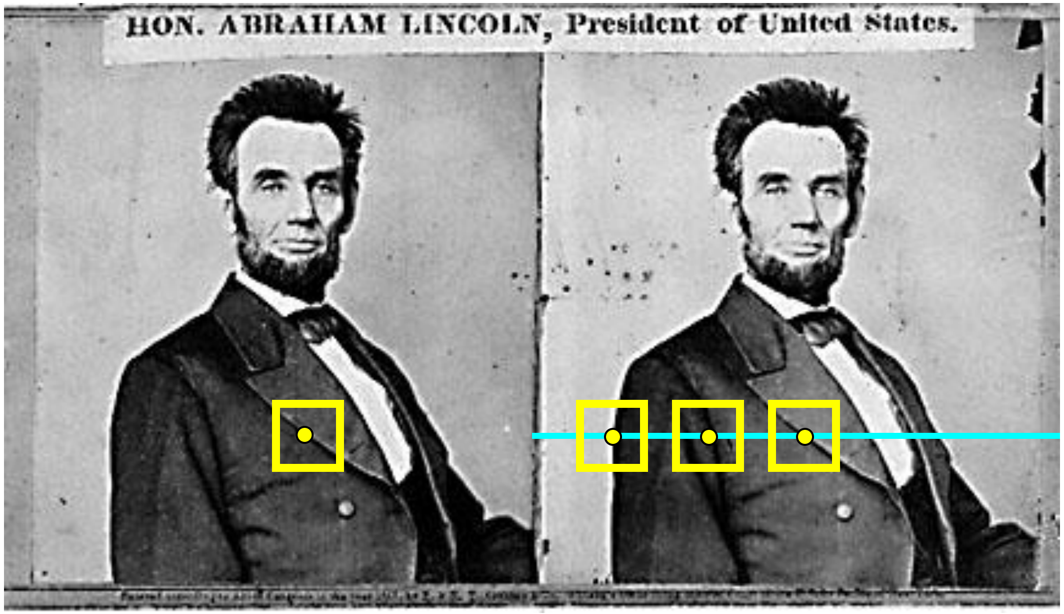




Kristen Grauman

Basic stereo matching algorithm

UNIVERSITÄT BONN

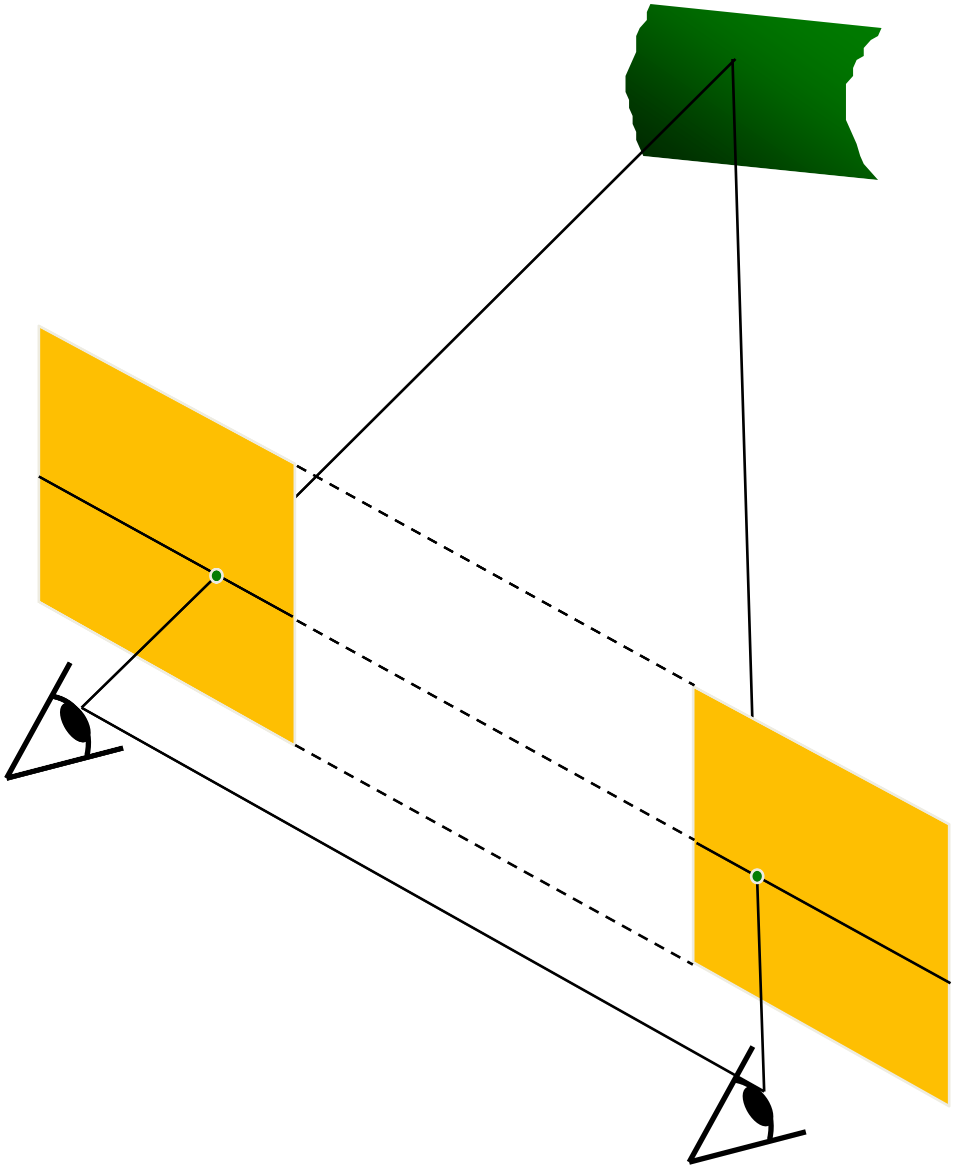


- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match
  - Triangulate the matches to get depth information
- Simplest case: epipolar lines are scanlines

Lazebnik

Simplest Case: Parallel images

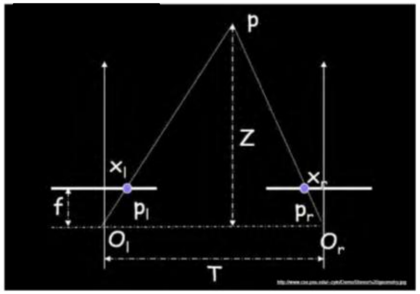
UNIVERSITÄT BONN



- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same
- Then, epipolar lines fall along the horizontal scan lines of the images

Lazebnik

Recall: Essential matrix



$$\mathbf{R} = \mathbf{I}$$

$$\mathbf{T} = [-d, 0, 0]^{\mathrm{T}}$$

$$\mathbf{E} = [\mathbf{T}_{\mathbf{x}}]\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix}$$

$$\mathbf{p} = [x, y, f]$$

$$\mathbf{p'} = [x', y', f]$$

$$\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0$$

$$\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

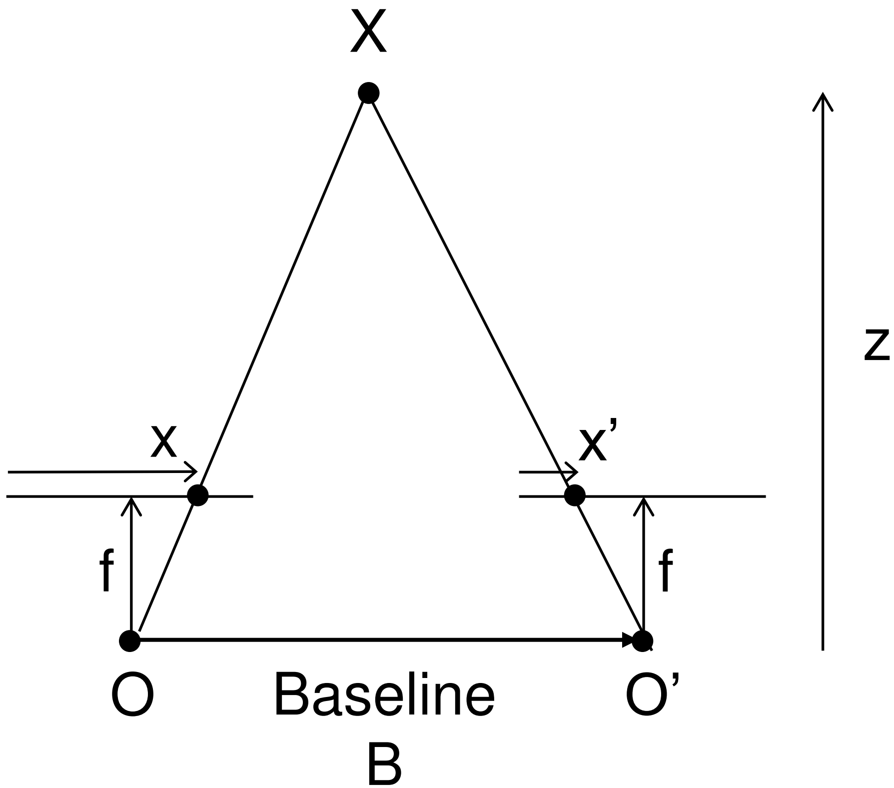
$$\Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ df \\ -dy \end{bmatrix} = 0$$

$$\boxed{\Leftrightarrow y = y'}$$

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

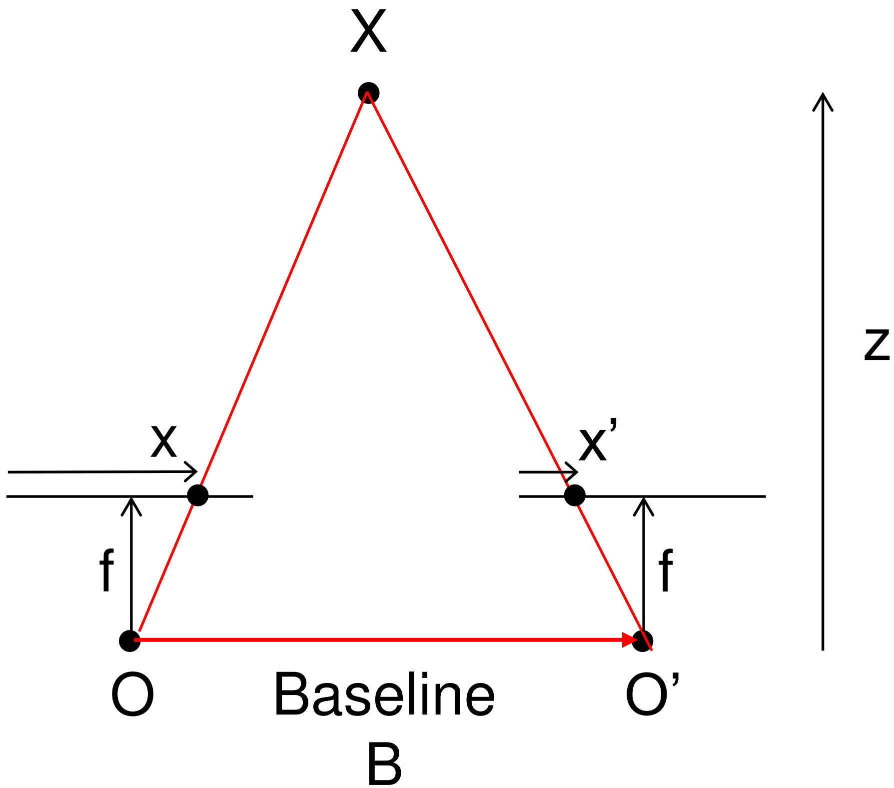
Kristen Grauman

Depth from disparity



Lazebnik

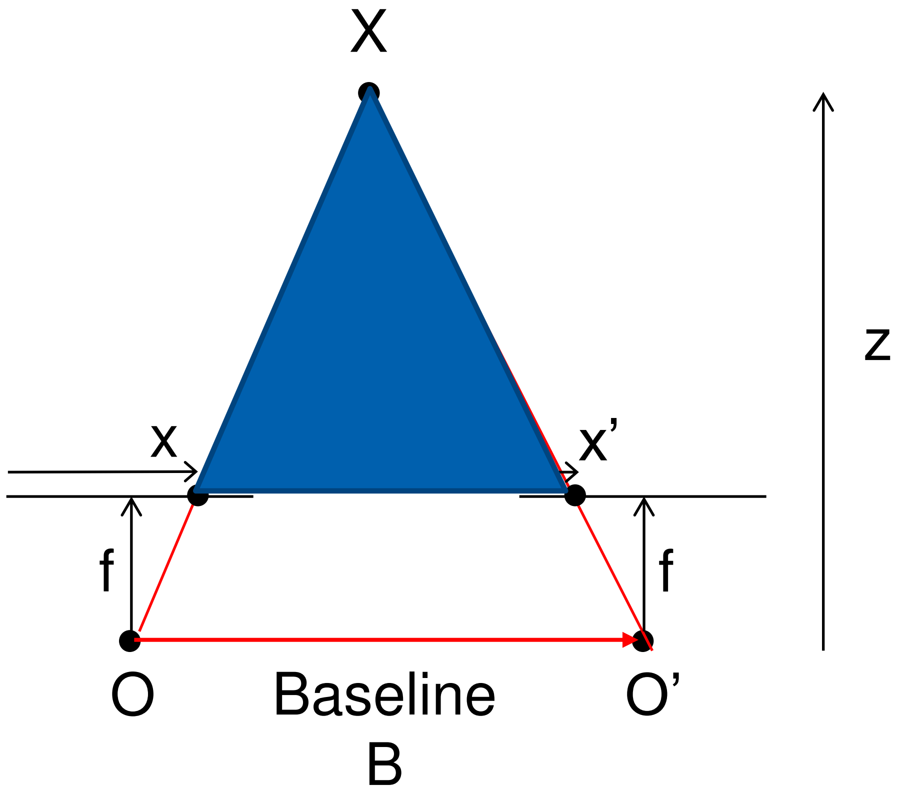
Depth from disparity



$$\frac{Z}{B}$$

Lazebnik

Depth from disparity

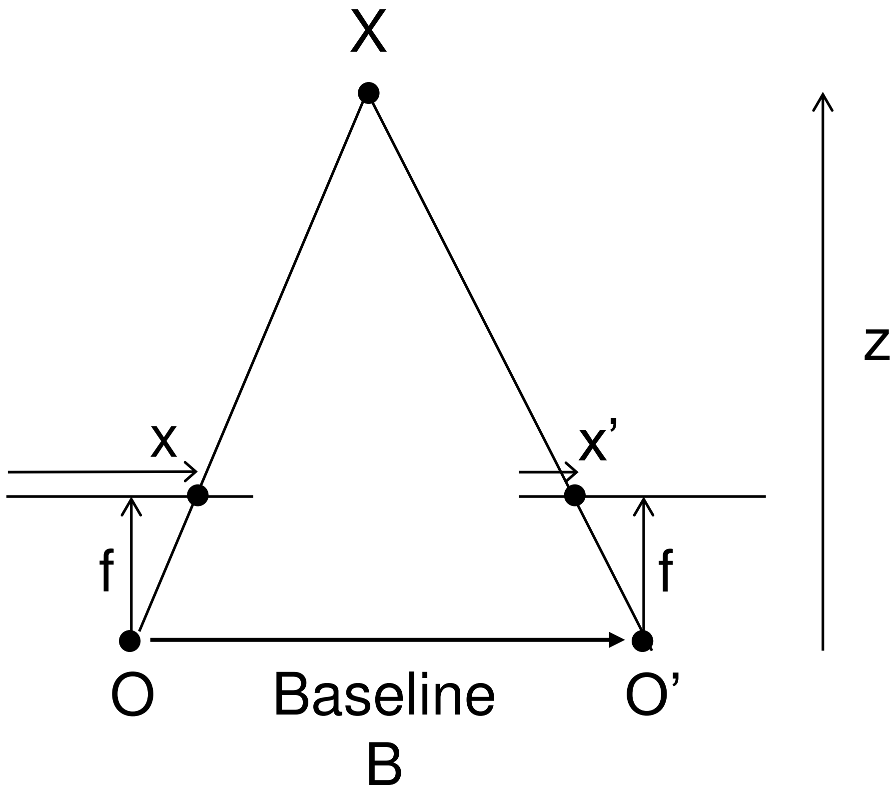


$$\frac{Z}{B} = \frac{Z - f}{B + x' - x}$$

Lazebnik

Depth from disparity

UNIVERSITÄT BONN

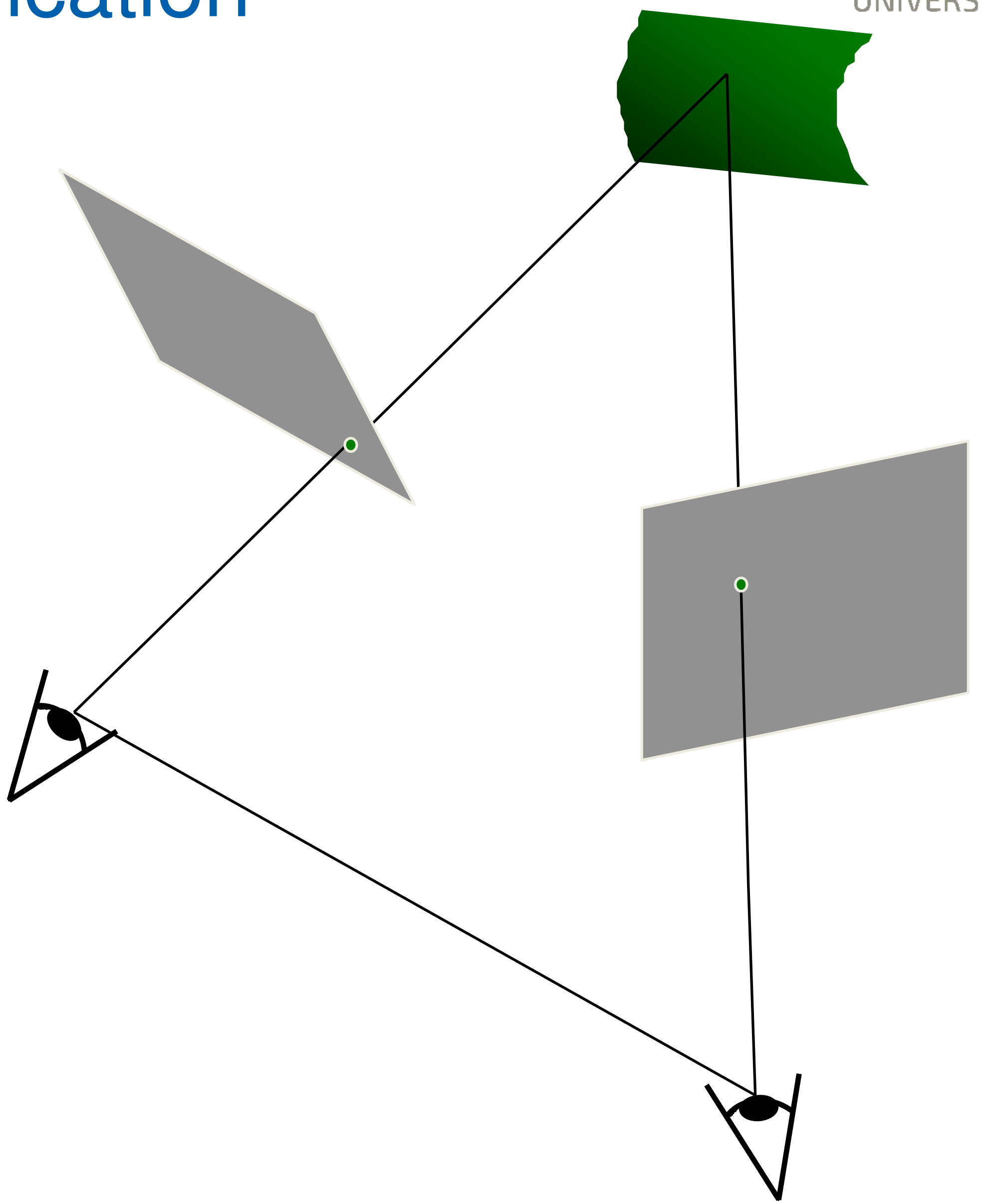


$$disparity = x - x' = \frac{B \cdot f}{z}$$

Disparity is inversely proportional to depth!

Lazebnik

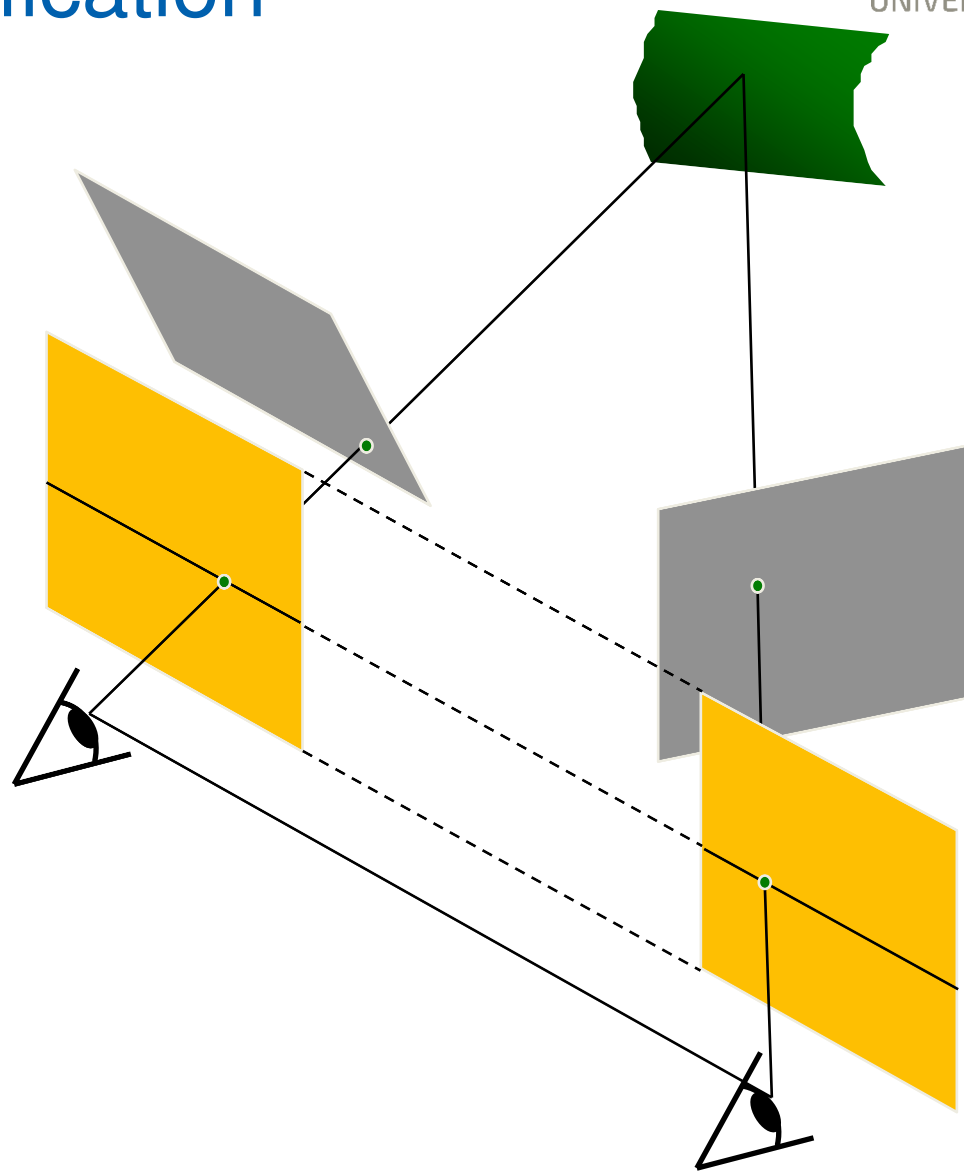
Stereo image rectification



Lazebnik

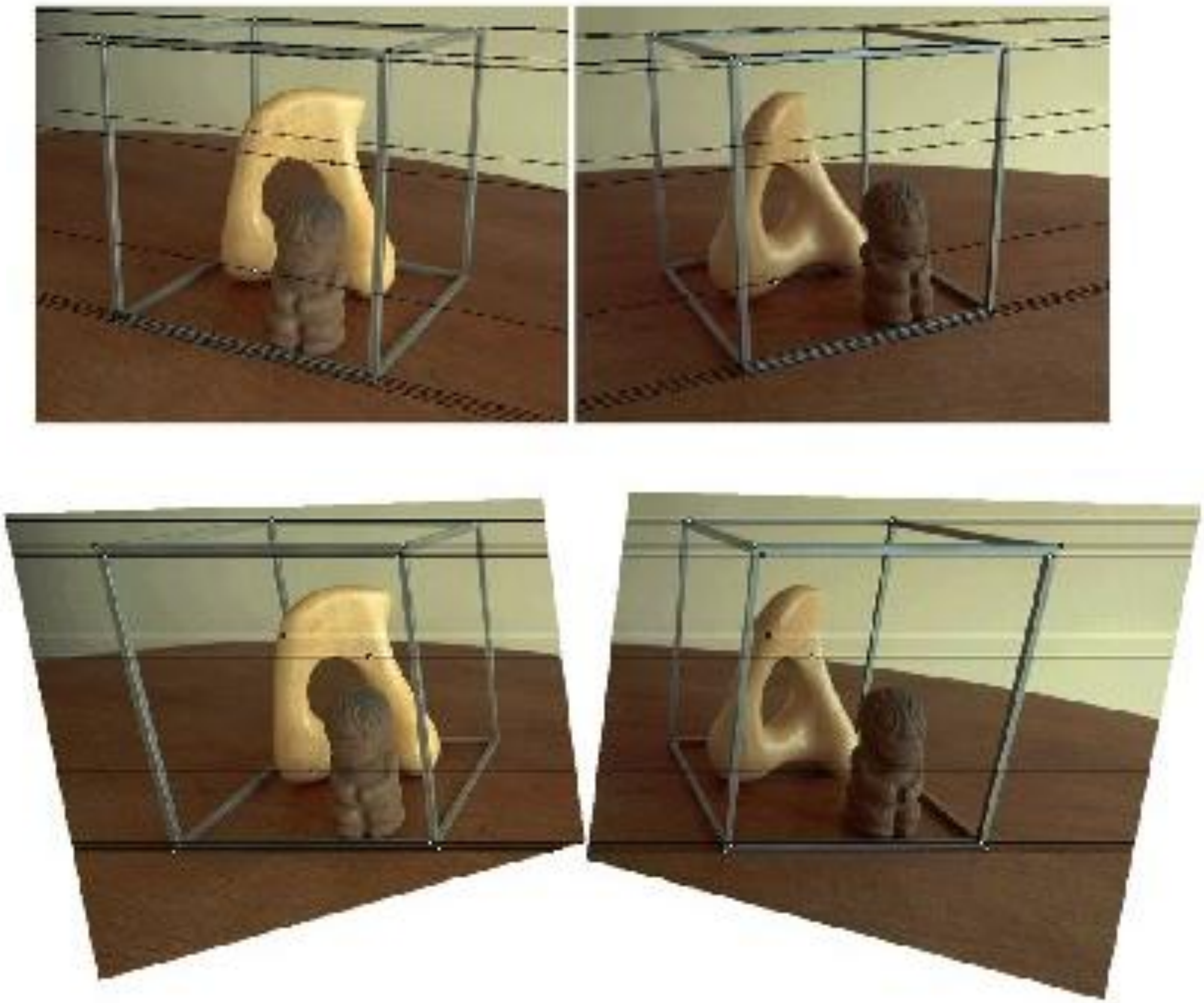
Stereo image rectification

- Reproject image planes onto a common plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each input image reprojection
- C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.



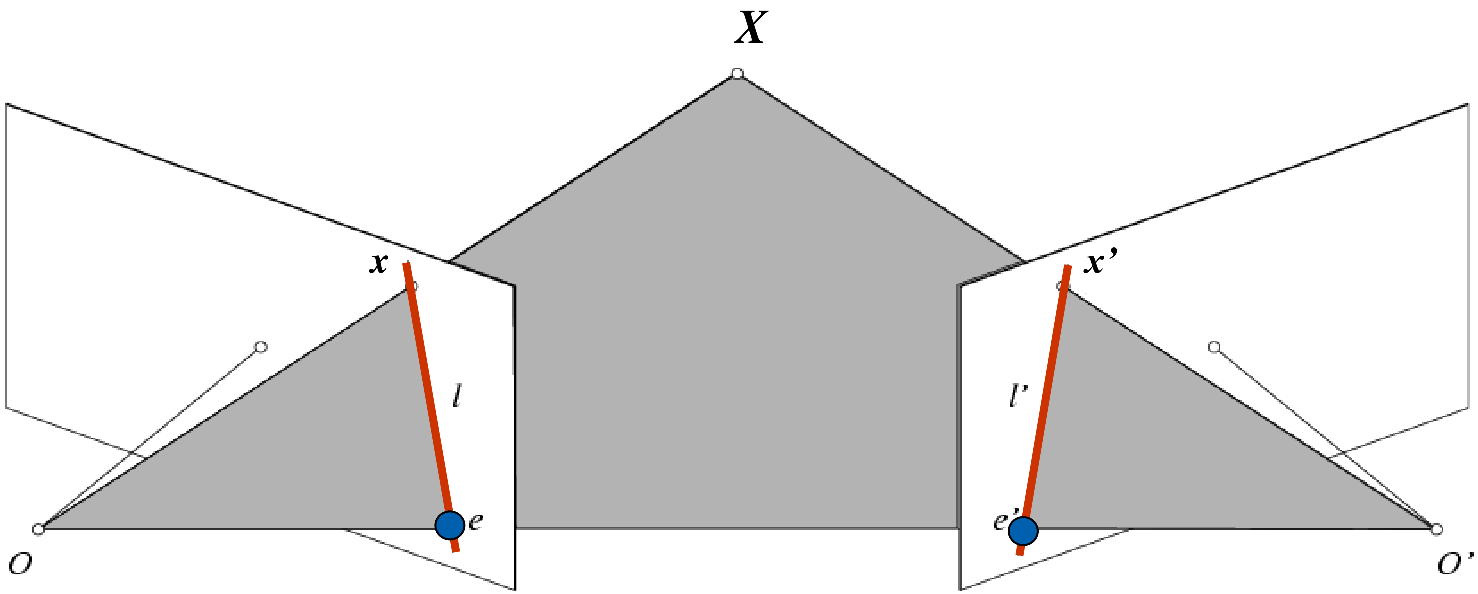
Lazebnik

Rectification example



Lazebnik

Recall: Epipolar constraint



$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- $F x'$ is the epipolar line associated with x' ($l = F x'$)
- $F^T x$ is the epipolar line associated with x ($l' = F^T x$)
- $F e' = 0$ and $F^T e = 0$
- F is singular (rank two)
- F has seven degrees of freedom

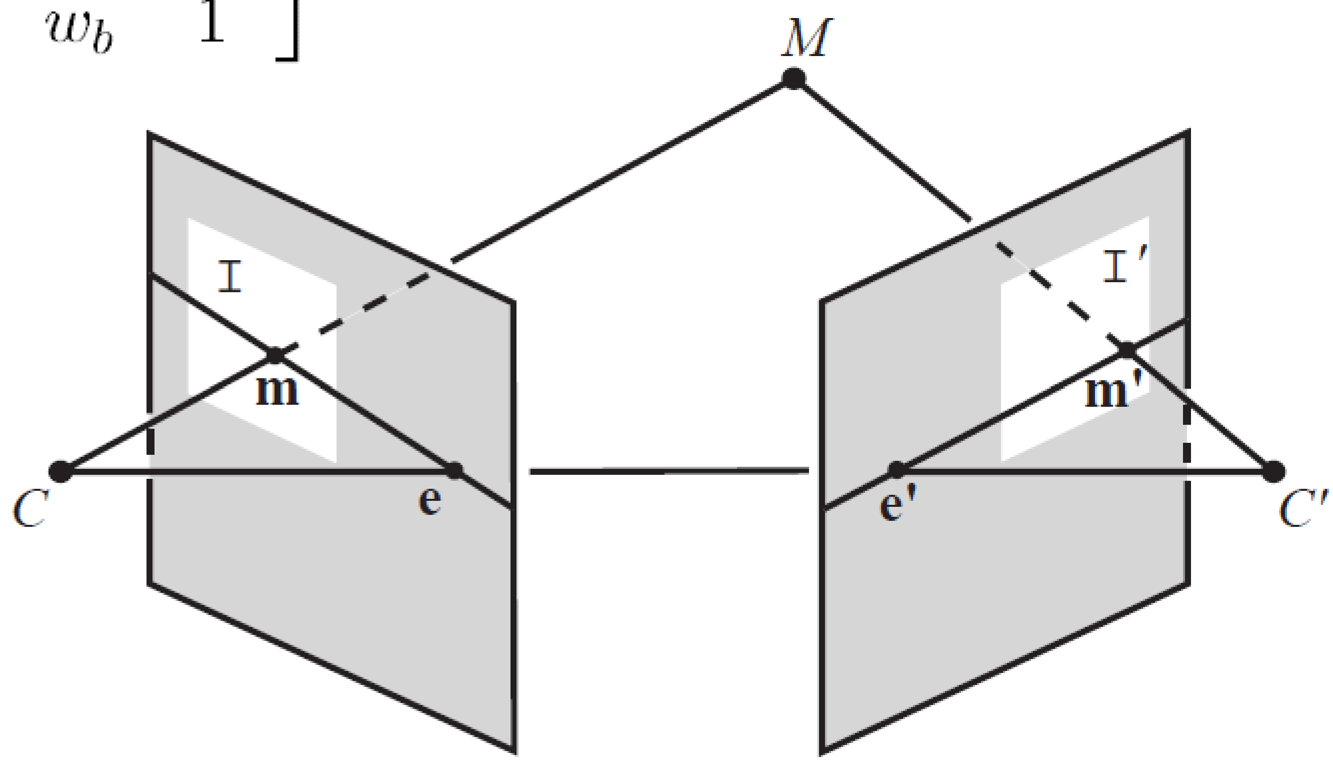
Lazebnik

Rectification

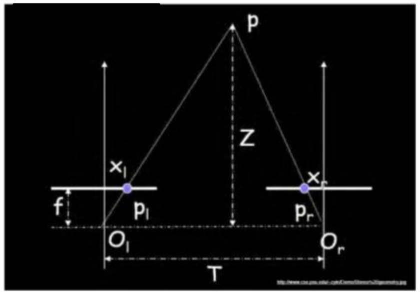
UNIVERSITÄT BONN

Estimate two homographies **H** and **H’**

$$\mathbf{H} = \begin{bmatrix} u_a & u_b & u_c \\ v_a & v_b & v_c \\ w_a & w_b & 1 \end{bmatrix}$$



Recall: Essential matrix



$$\mathbf{R} = \mathbf{I}$$

$$\mathbf{T} = [-d, 0, 0]^{\mathrm{T}}$$

$$\mathbf{E} = [\mathrm{T}_{\mathbf{x}}]\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix}$$

$$\mathbf{p} = [x, y, f]$$

$$\mathbf{p}' = [x', y', f]$$

$$\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0$$

$$\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

$$\Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ df \\ -dy \end{bmatrix} = 0$$

$$\boxed{\Leftrightarrow y = y'}$$

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

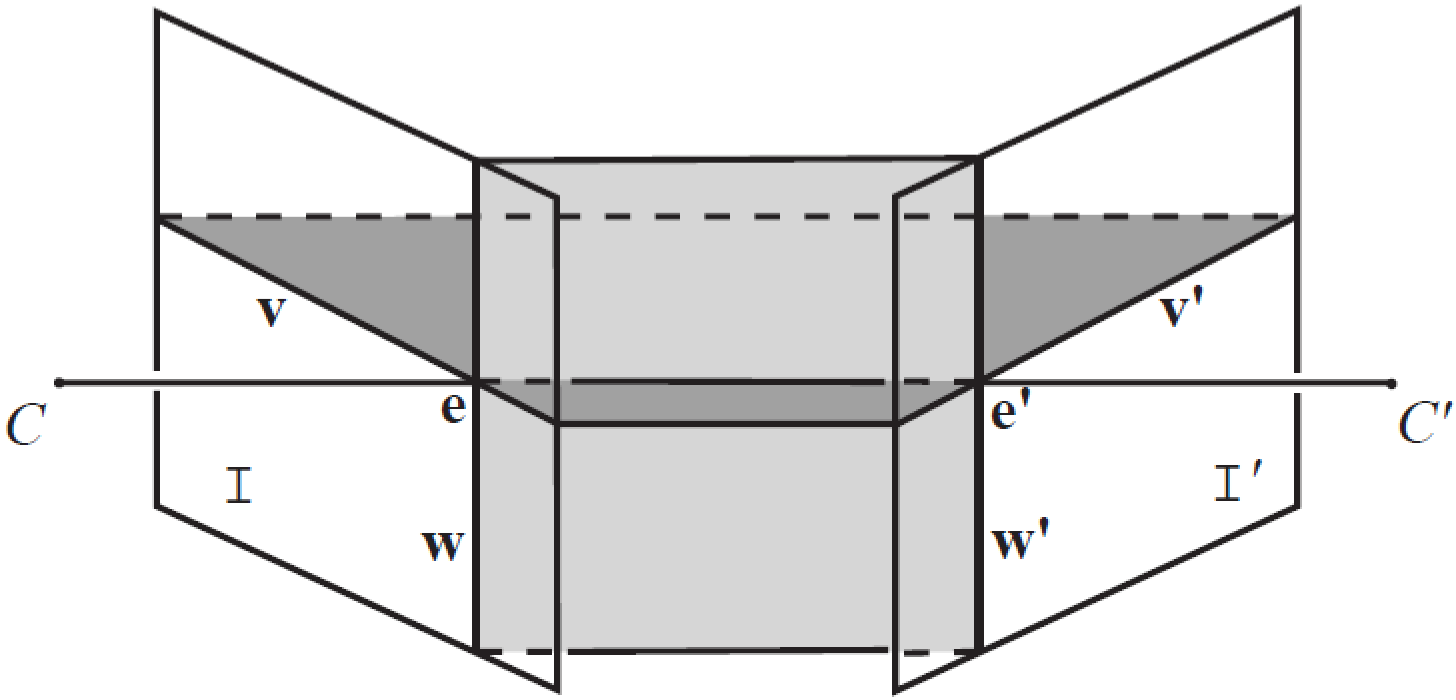
Kristen Grauman

Rectification

- Map epipoles to infinity (1,0,0) (canonical form)
- Fundamental matrix after rectification:

$$\bar{\mathbf{F}} = [\,\mathbf{i}\,]_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Rectification



$$\mathbf{H} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} u_a & u_b & u_c \\ v_a & v_b & v_c \\ w_a & w_b & w_c \end{bmatrix}$$

$$\mathbf{He} = \begin{bmatrix} \mathbf{u}^T\mathbf{e} & \mathbf{v}^T\mathbf{e} & \mathbf{w}^T\mathbf{e} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

Decompose homography

- Decompose **H** into affine transformation $\mathbf{H}_a$ and projective transformation $\mathbf{H}_p$
- Decompose affine trans. $\mathbf{H}_a$ into similarity trans. $\mathbf{H}_r$ and shearing trans. $\mathbf{H}_s$
- $\mathbf{H} = \mathbf{H}_a\mathbf{H}_p = \mathbf{H}_s\mathbf{H}_r\mathbf{H}_p$

$$\mathbf{H} = \begin{bmatrix} u_a & u_b & u_c \\ v_a & v_b & v_c \\ w_a & w_b & 1 \end{bmatrix}$$

$$\mathbf{H}_s = \begin{bmatrix} s_a & s_b & s_c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_r = \begin{bmatrix} v_b - v_c w_b & v_c w_a - v_a & 0 \\ v_a - v_c w_a & v_b - v_c w_b & v_c \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{H}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{bmatrix}$$

Rectification example

$\mathbf{H}_p$

Lazebnik

Rectification example

$\mathbf{H}_p$

$\mathbf{H}_r\mathbf{H}_p$

Lazebnik

Rectification example

$H_r H_p$

$H_s H_r H_p$

Lazebnik

Estimate projective transformation

UNIVERSITÄT BONN

- Estimate $\mathbf{H}_p$: $\mathbf{H}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{bmatrix}$
- All pixels from images $\mathbf{p}_i = [p_{i,u}\ p_{i,v}\ 1]^T$ are mapped by $\mathbf{H}_p$ with homogenous component: $w_i = \mathbf{w}^T \mathbf{p}_i$
- After rectification, the homogenous component should be the same for all points including the mean:

$$\mathbf{p}_c = \frac{1}{n} \sum_{i=1}^{n} \mathbf{p}_i$$

- Minimize non-affine distortion: (is zero for affine transformation) $\sum_{i=1}^{n} \left[\frac{\mathbf{w}^T(\mathbf{p}_i - \mathbf{p}_c)}{\mathbf{w}^T \mathbf{p}_c} \right]^2$

Estimate projective transformation

- Estimate $\mathbf{H}_p$:
- All pixels from images

$$\mathbf{H}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} p_{1,u} - p_{c,u} & p_{2,u} - p_{c,u} & \cdots & p_{n,u} - p_{c,u} \\ p_{1,v} - p_{c,v} & p_{2,v} - p_{c,v} & \cdots & p_{n,v} - p_{c,v} \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad \mathbf{p}_c = \frac{1}{n} \sum_{i=1}^{n} \mathbf{p}_i$$

- Minimize: $$\frac{\mathbf{w}^T \mathbf{P} \mathbf{P}^T \mathbf{w}}{\mathbf{w}^T \mathbf{p}_c \mathbf{p}_c^T \mathbf{w}}$$

- Relation: $$\mathbf{w} = [\,\mathbf{e}\,]_\times \mathbf{z}$$ $$\mathbf{w}' = \mathbf{F} \mathbf{z}$$

Estimate projective transformation

- Estimate $\mathbf{H}_p$:
- All pixels from images

$$\mathbf{H}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} p_{1,u} - p_{c,u} & p_{2,u} - p_{c,u} & \cdots & p_{n,u} - p_{c,u} \\ p_{1,v} - p_{c,v} & p_{2,v} - p_{c,v} & \cdots & p_{n,v} - p_{c,v} \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad \mathbf{p}_c = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i$$

- Minimize:

$$\frac{\mathbf{z}^T \overbrace{[\mathbf{e}]_\times^T \mathbf{P}\mathbf{P}^T [\mathbf{e}]_\times}^{\mathbf{A}} \mathbf{z}}{\mathbf{z}^T \underbrace{[\mathbf{e}]_\times^T \mathbf{p}_c\mathbf{p}_c^T [\mathbf{e}]_\times}_{\mathbf{B}} \mathbf{z}} + \frac{\mathbf{z}^T \overbrace{\mathbf{F}^T \mathbf{P}'\mathbf{P}'^T \mathbf{F}}^{\mathbf{A}'} \mathbf{z}}{\mathbf{z}^T \underbrace{\mathbf{F}^T \mathbf{p}'_c\mathbf{p}'^T_c \mathbf{F}}_{\mathbf{B}'} \mathbf{z}}$$

$$\mathbf{w} = [\mathbf{e}]_\times \mathbf{z}$$

$$\mathbf{w}' = \mathbf{F}\mathbf{z}$$

- Cholesky decomposition: $\mathbf{A} = \mathbf{D}^T\mathbf{D}$
- **y** is eigenvector with highest eigenvalue of $\mathbf{D}^{-T}\mathbf{B}\mathbf{D}^{-1}$
- **w and w'** is given by $\mathbf{w} = [\mathbf{e}]_\times \mathbf{z}$ $\quad \mathbf{w}' = \mathbf{F}\mathbf{z}$ $\quad \mathbf{z} = \mathbf{D}^{-1}\mathbf{y}$

Rectification example

$\mathbf{H}_p$

Lazebnik

Estimate similarity transformation

- We know fundamental matrix **F** and

$$\mathbf{H}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{bmatrix}$$

- Similarity matrix is then given by

$$\mathbf{H}_r = \begin{bmatrix} F_{32} - w_b F_{33} & w_a F_{33} - F_{31} & 0 \\ F_{31} - w_a F_{33} & F_{32} - w_b F_{33} & F_{33} + v'_c \\ 0 & 0 & 1 \end{bmatrix}$$

- v'_c is set such that pixels (v) start with zero

Rectification example

$\mathbf{H}_p$

$\mathbf{H}_r\mathbf{H}_p$

Lazebnik

Map image back to image size

- Homographies $\mathbf{H}_r\mathbf{H}_p$ rectify images, but might introduce unnecessary distortion, e.g., images are very small or very large.
- Map images back to original size (shearing):

$$\mathbf{S} = \begin{bmatrix} a & b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

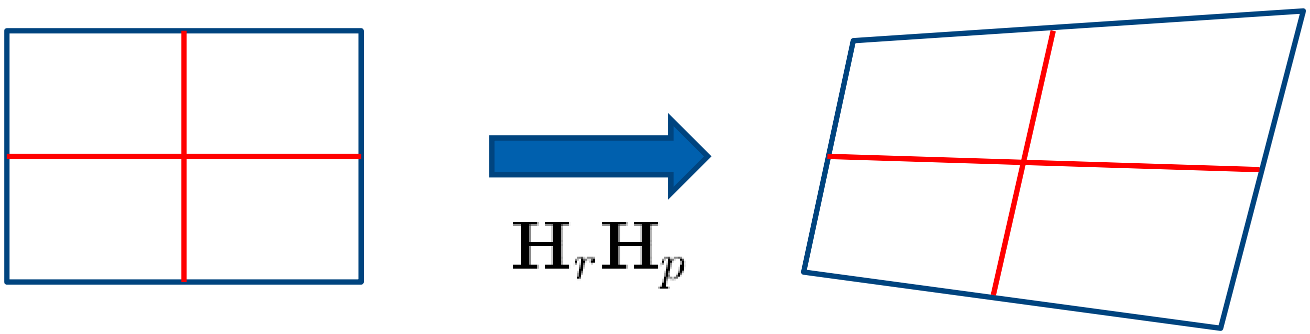
Map image back to image size

- Define reference lines:

$\mathbf{a} = [\,\frac{w-1}{2}\;\; 0\;\; 1\,]^T$, $\mathbf{b} = [\,w-1\;\; \frac{h-1}{2}\;\; 1\,]^T$, $\mathbf{c} = [\,\frac{w-1}{2}\;\; h-1\;\; 1\,]^T$, and $\mathbf{d} = [\,0\;\; \frac{h-1}{2}\;\; 1\,]^T$

$$\hat{\mathbf{a}} = \mathbf{H}_r\mathbf{H}_p\mathbf{a}$$

$$\begin{aligned} \mathbf{x} &= \hat{\mathbf{b}} - \hat{\mathbf{d}}, \\ \mathbf{y} &= \hat{\mathbf{c}} - \hat{\mathbf{a}}. \end{aligned}$$



- Make lines perpendicular and keep width and height ratio:

$$(\mathbf{S}\mathbf{x})^T(\mathbf{S}\mathbf{y}) = 0, \quad \frac{(\mathbf{S}\mathbf{x})^T(\mathbf{S}\mathbf{x})}{(\mathbf{S}\mathbf{y})^T(\mathbf{S}\mathbf{y})} = \frac{w^2}{h^2}.$$

Map image back to image size

- Map images back to original size (shearing):

$$\mathbf{S} = \begin{bmatrix} a & b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$a = \frac{h^2 x_v^2 + w^2 y_v^2}{hw(x_v y_u - x_u y_v)} \qquad b = \frac{h^2 x_u x_v + w^2 y_u y_v}{hw(x_u y_v - x_v y_u)}$$

- Add translation and scale to preserve image area and pixels start at zero

Rectification example

$\mathbf{H}_p$

Lazebnik

Rectification example

$\mathbf{H}_p$

$\mathbf{H}_r\mathbf{H}_p$

Lazebnik

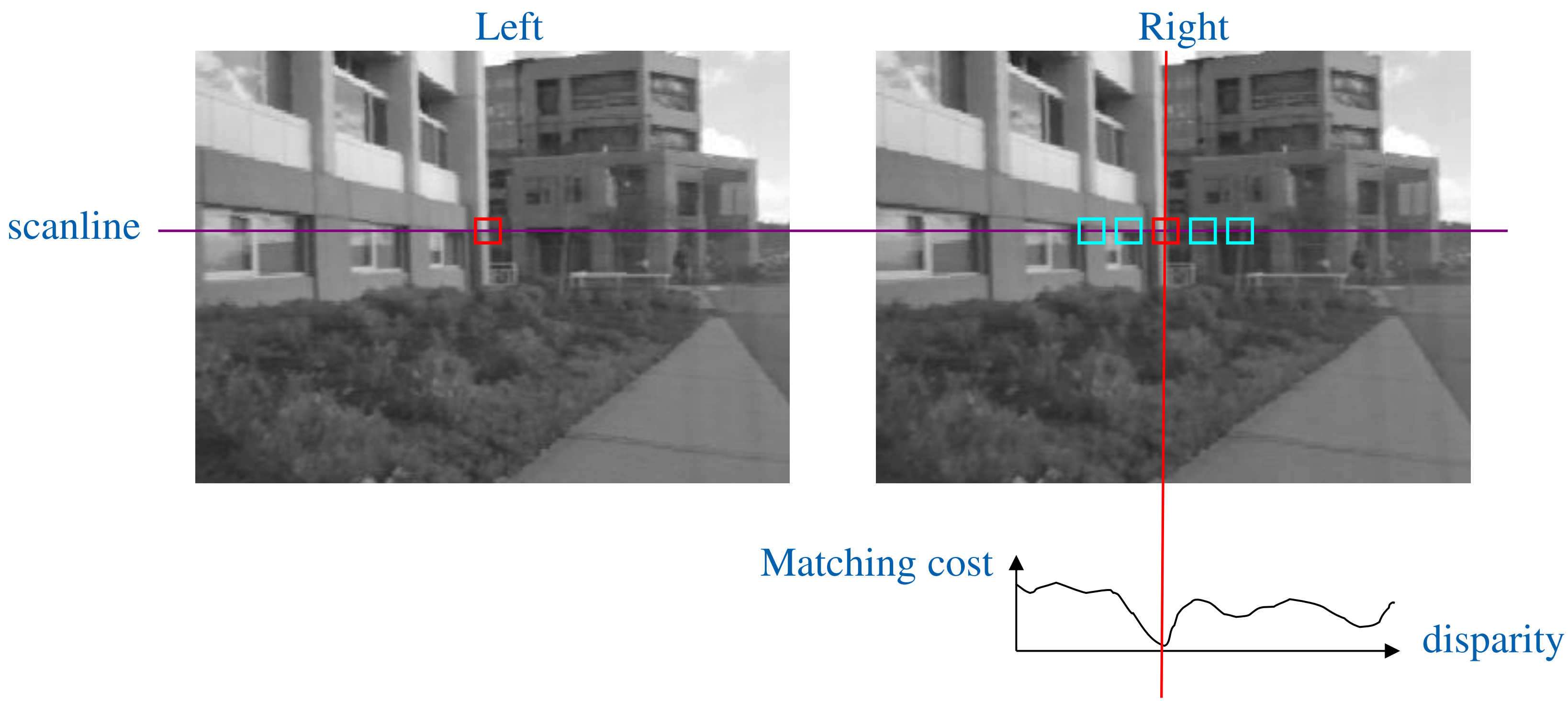
Rectification example

$H_r H_p$

$H_s H_r H_p$

Lazebnik

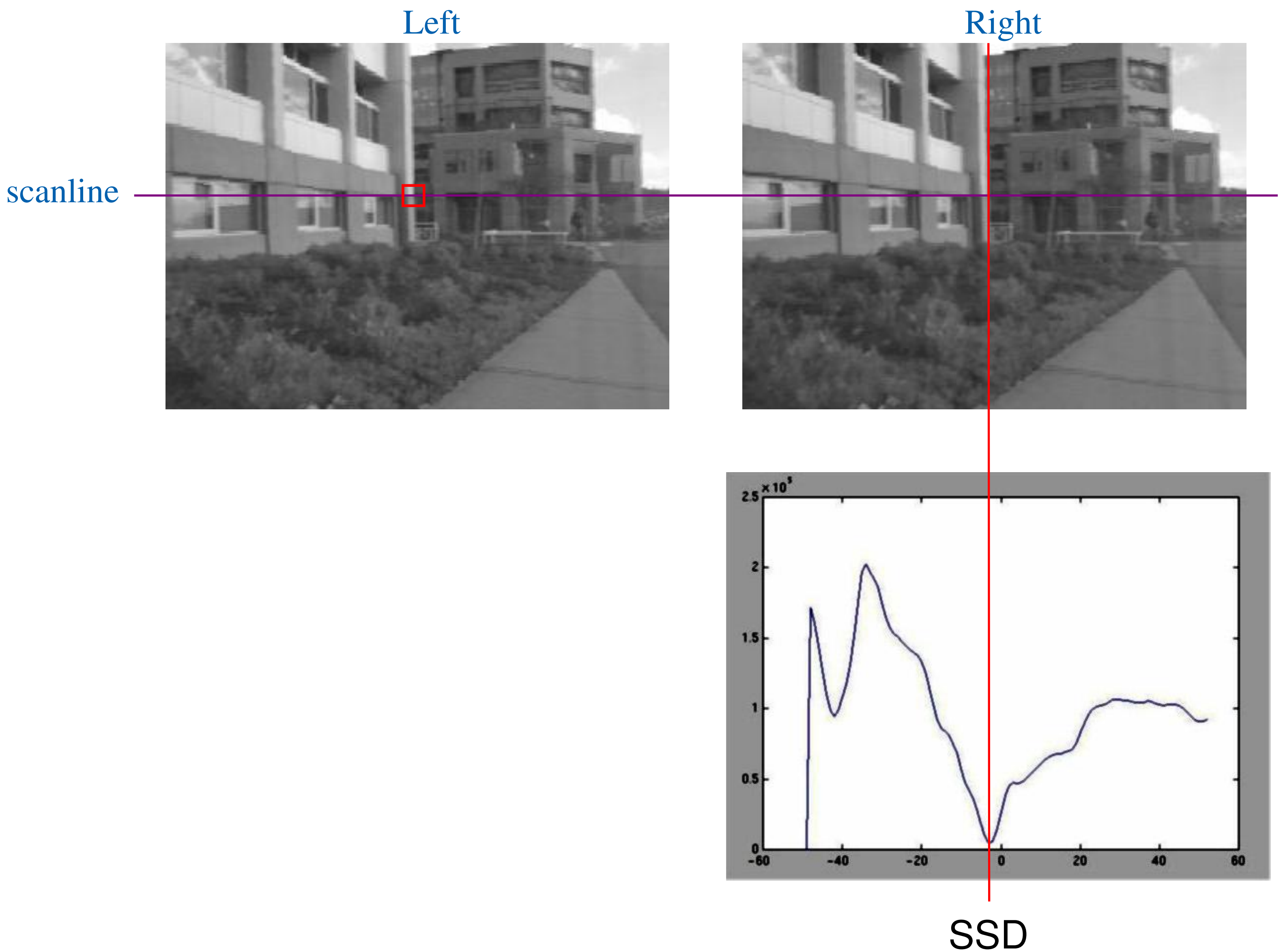
Correspondence search



- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

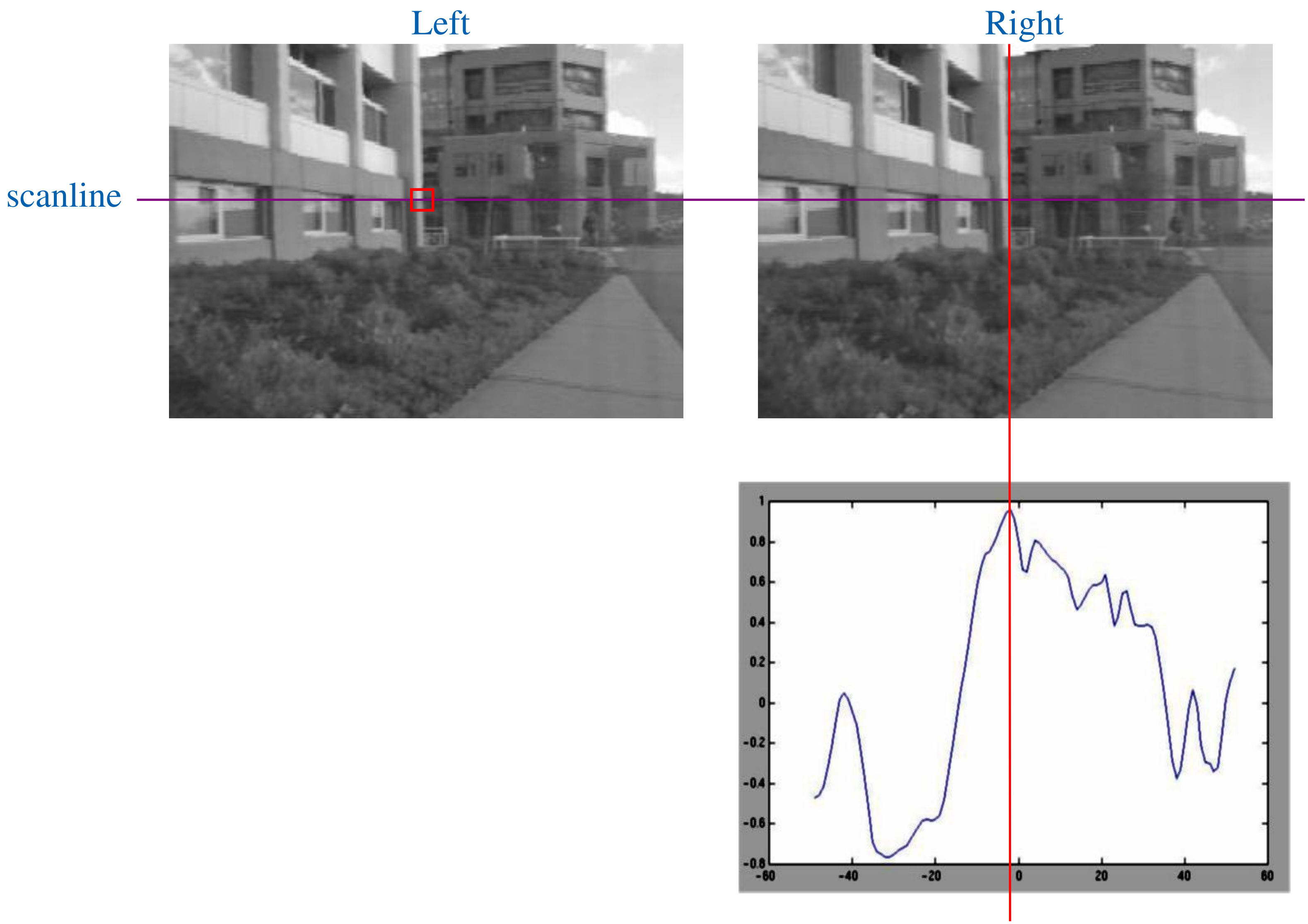
Lazebnik

Correspondence search



Lazebnik

Correspondence search

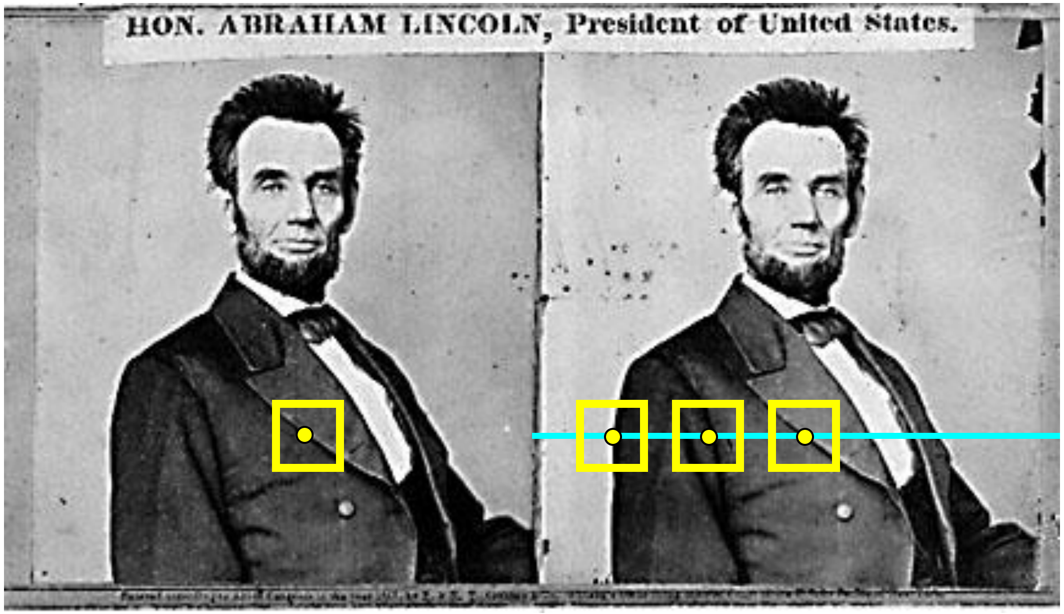


Norm. corr

Lazebnik

Basic stereo matching algorithm

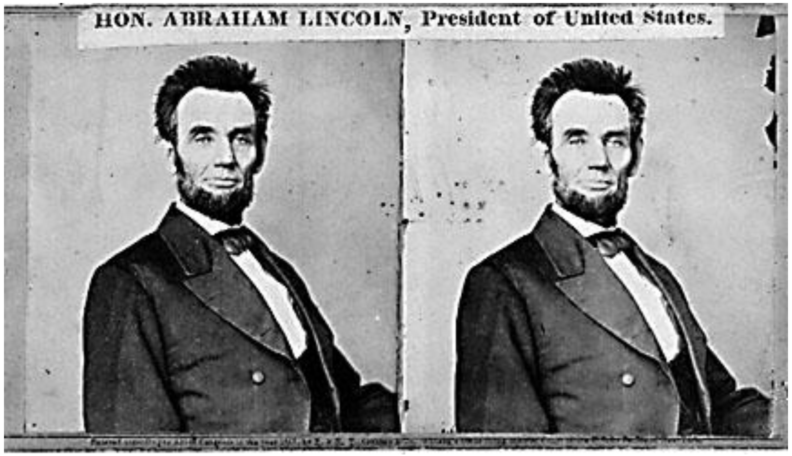
UNIVERSITÄT BONN



- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
  - Find corresponding epipolar scanline in the right image
  - Examine all pixels on the scanline and pick the best match x'
  - Compute disparity x-x' and set depth(x) = B*f/(x-x')

Lazebnik

Failures of correspondence search



Textureless surfaces

Occlusions, repetition

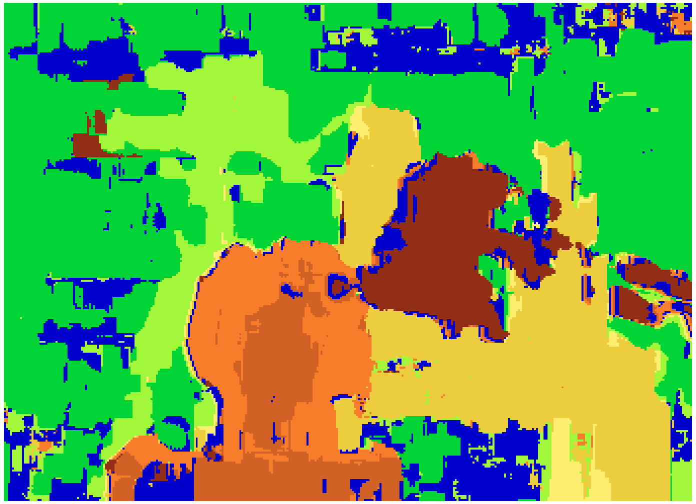
Non-Lambertian surfaces, specularities

Lazebnik

Results with window search

Data

Window-based matching



Ground truth

Lazebnik

How can we improve window-based matching?

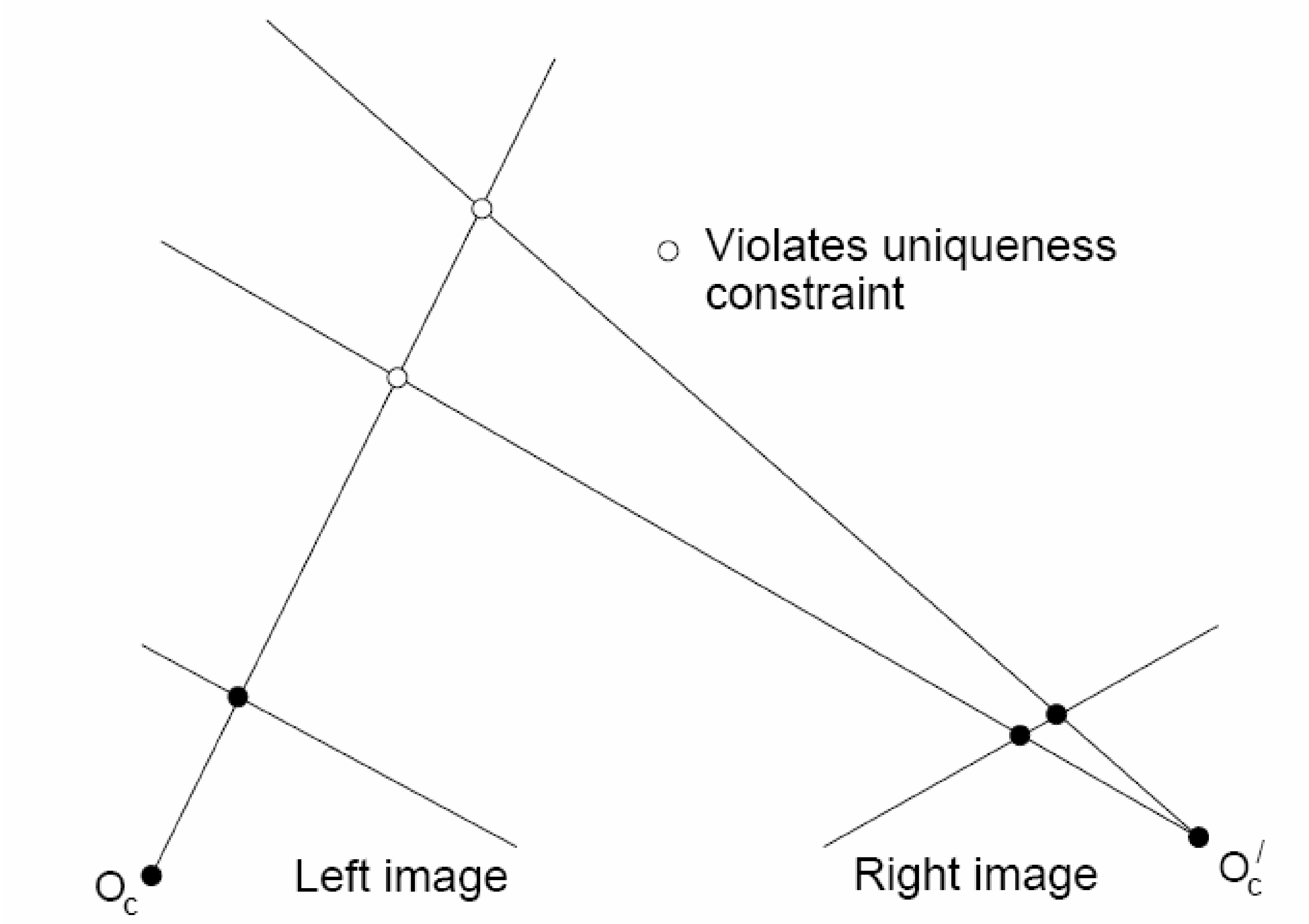
- The similarity constraint is local (each reference window is matched independently)
- Need to enforce some correspondence constraints

Lazebnik

Non-local constraints

UNIVERSITÄT BONN

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

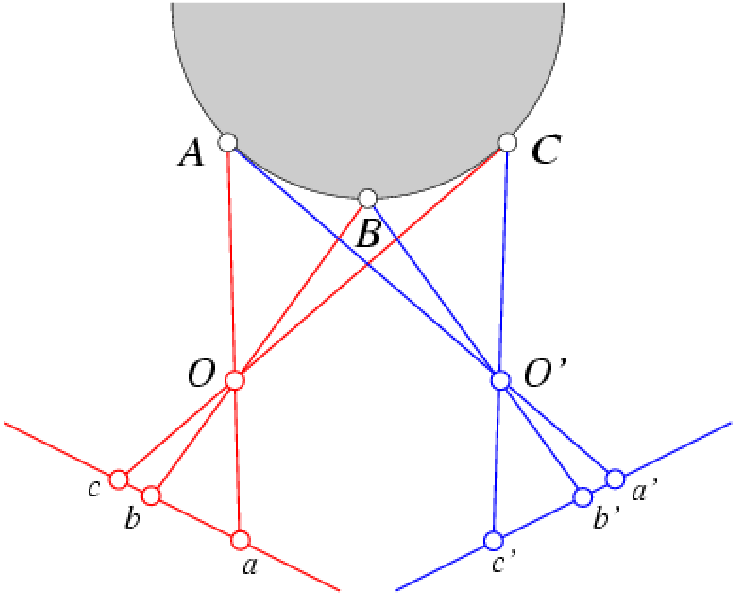


Lazebnik

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

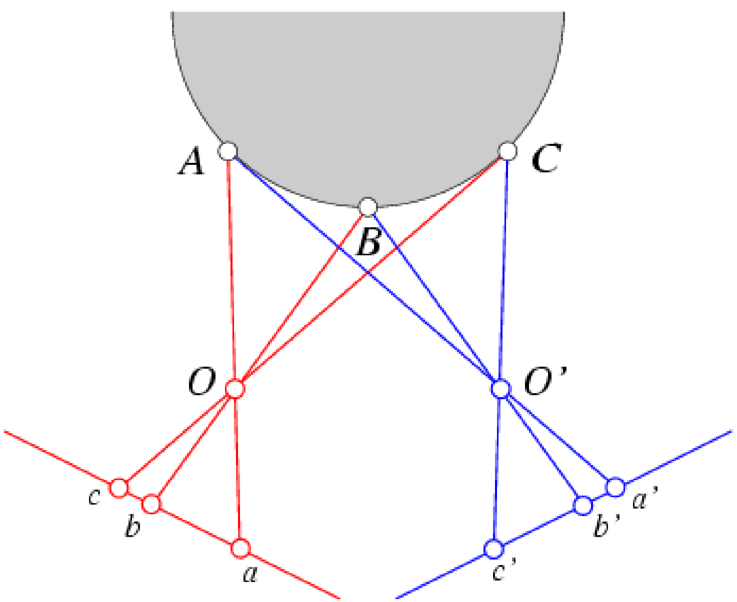


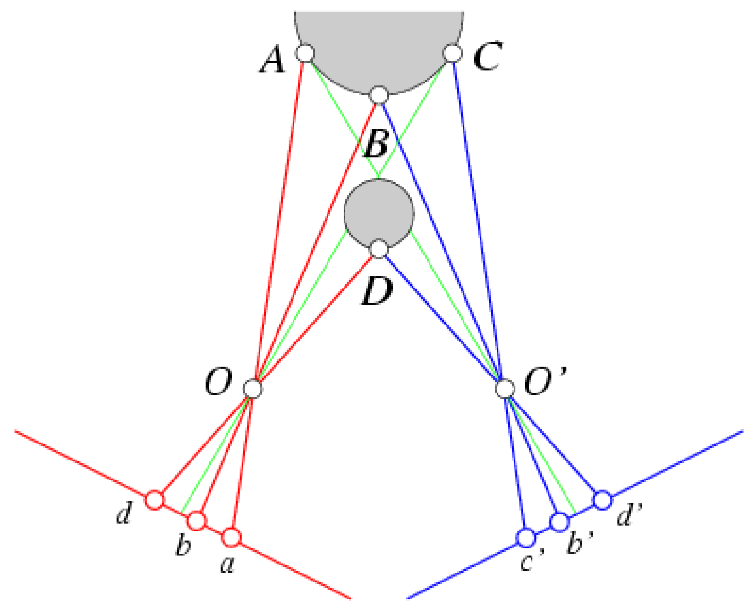
Lazebnik

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





Ordering constraint doesn’t hold

Lazebnik

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

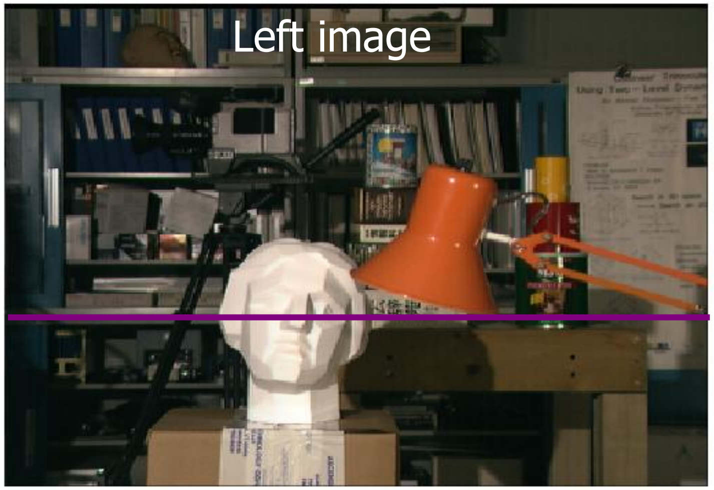
– We expect disparity values to change slowly (for the most part)

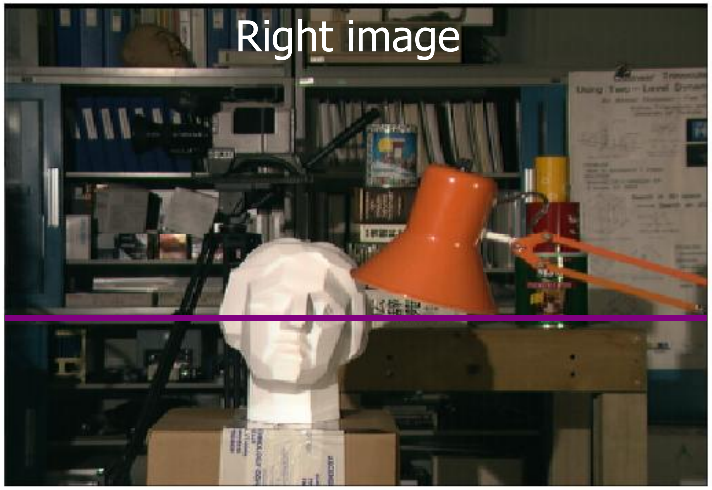
Lazebnik

Scanline stereo

Try to coherently match pixels on the entire scanline

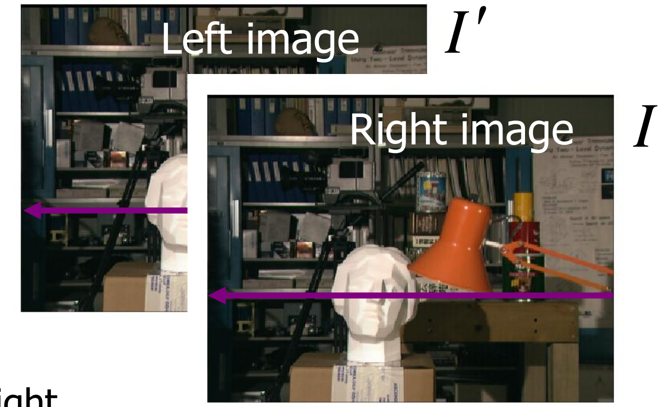
Different scanlines are still optimized independently

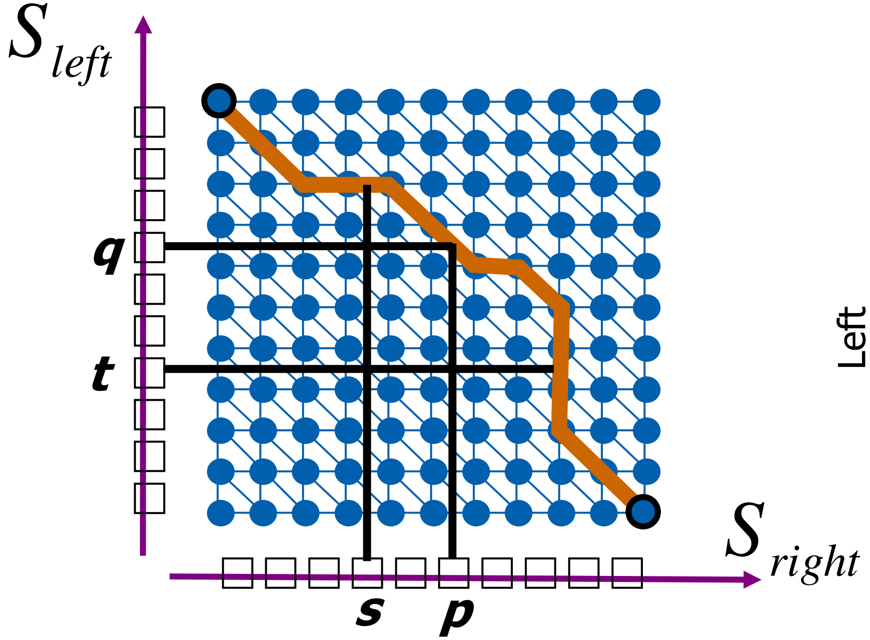


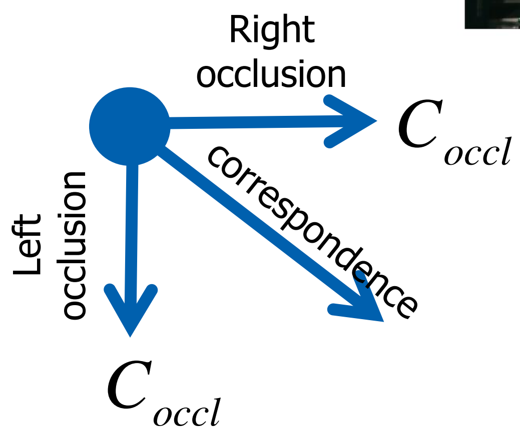


Lazebnik

“Shortest paths” for scan-line stereo







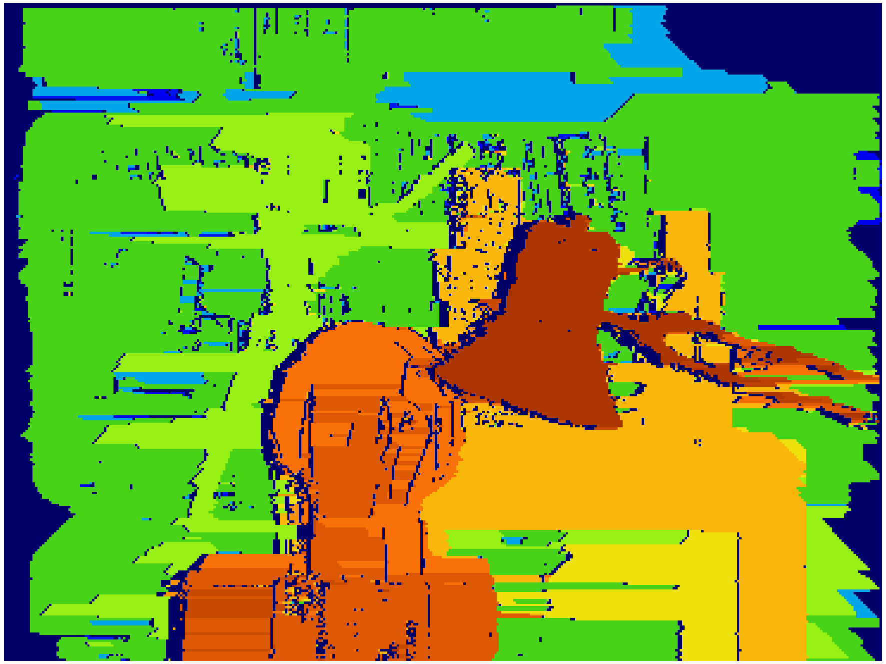
Can be implemented with dynamic programming

Ohta & Kanade ’85, Cox et al. ‘96

Y. Boykov

Coherent stereo on 2D grid

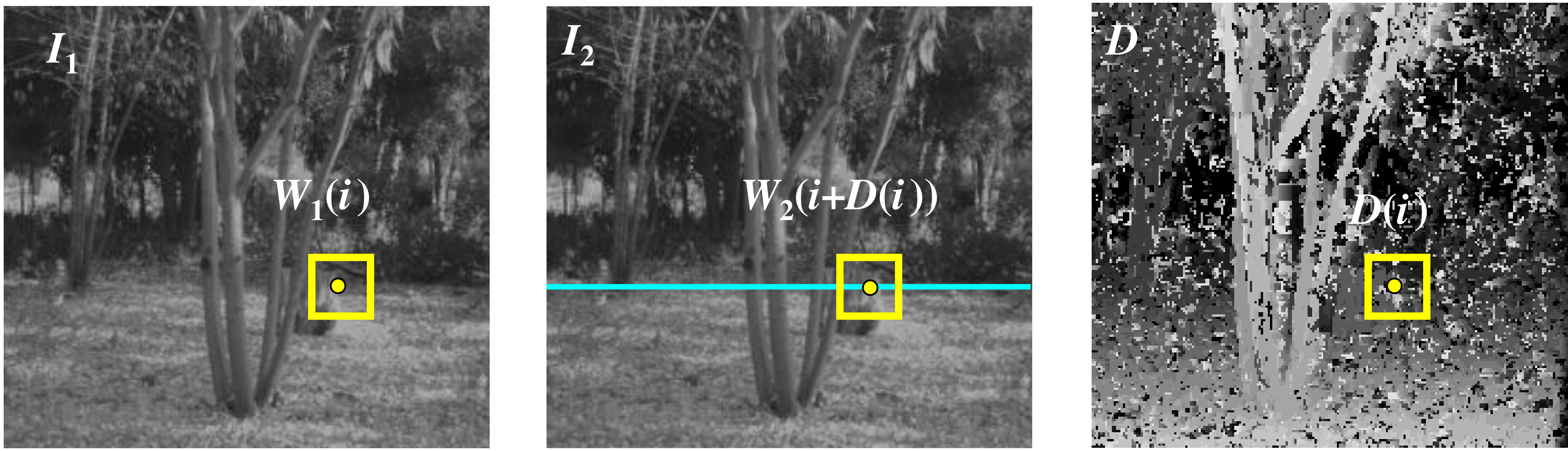
Scanline stereo generates streaking artifacts



Can’t use dynamic programming to find spatially coherent disparities/ correspondences on a 2D grid

Lazebnik

Stereo matching as energy minimization



$$E(D) = \underbrace{\sum_i \left(W_1(i) - W_2(i + D(i))\right)^2}_{\textit{data term}} + \lambda \underbrace{\sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)}_{\textit{smoothness term}}$$

Energy functions of this form can be minimized using graph cuts

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

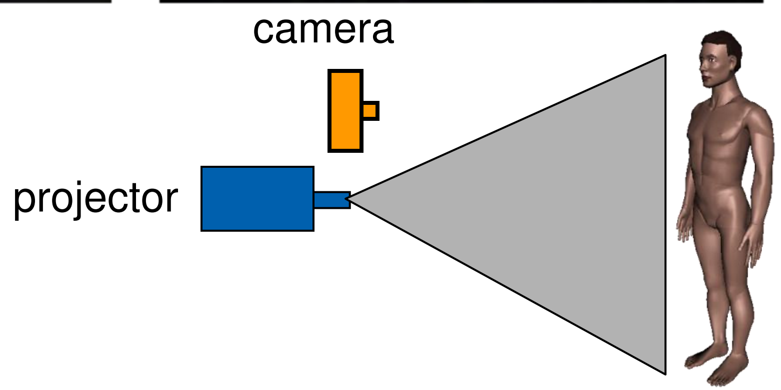
Lazebnik

Active stereo with structured light

Project “structured” light patterns onto the object

– Simplifies the correspondence problem

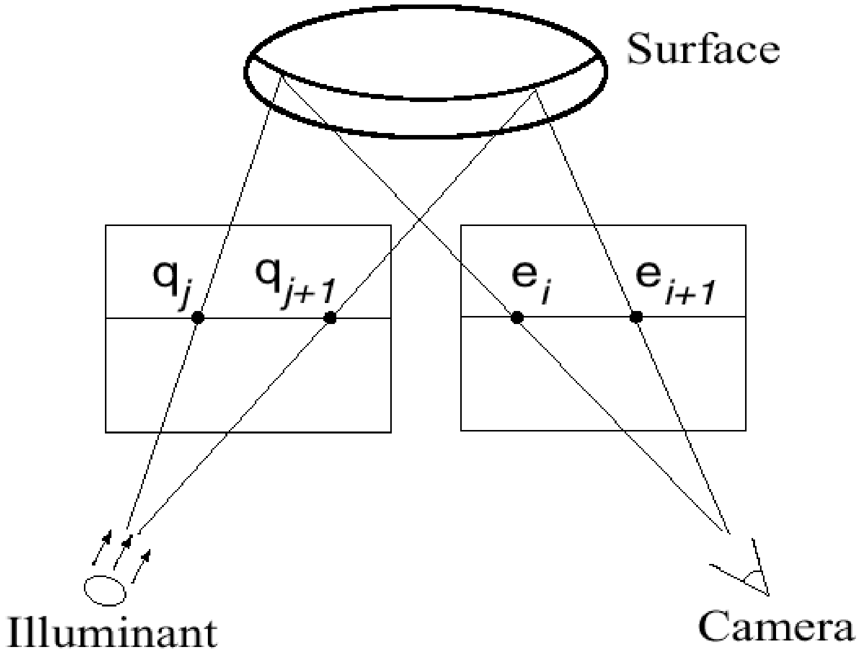
– Allows us to use only one camera

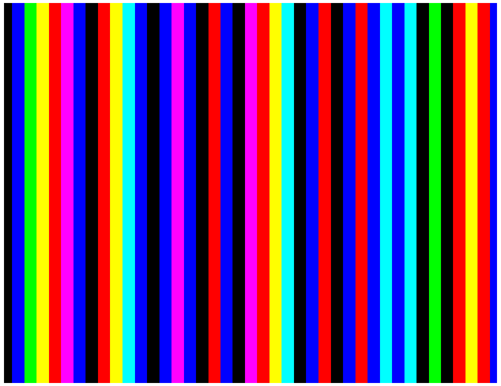


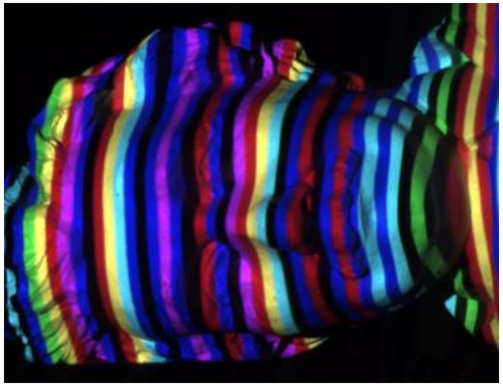
L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Lazebnik

Active stereo with structured light







Lazebnik

Aligning range images

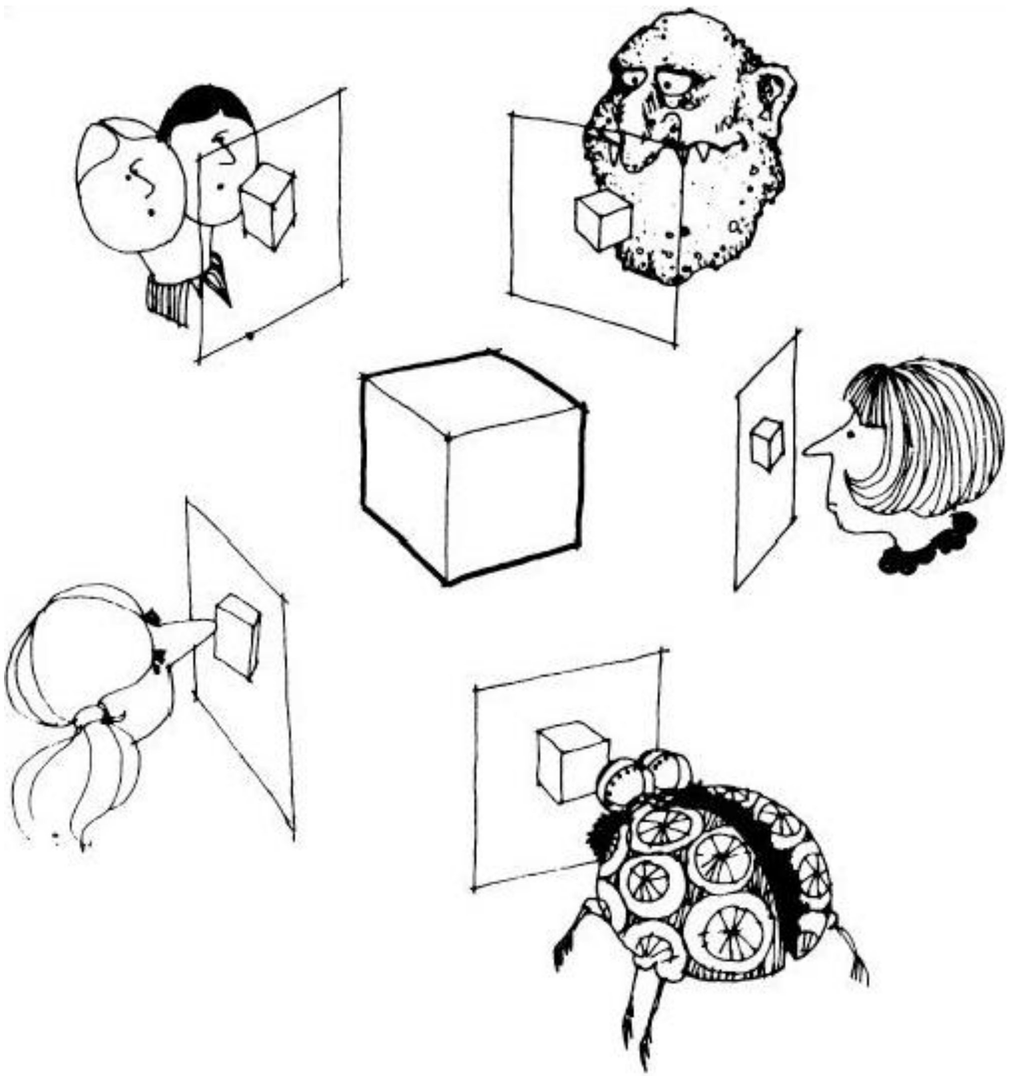
A single range scan is not sufficient to describe a complex surface

Need techniques to register multiple range images

B. Curless and M. Levoy, A Volumetric Method for Building Complex Models from Range Images, SIGGRAPH 1996

Lazebnik

Multi-view stereo



Lazebnik

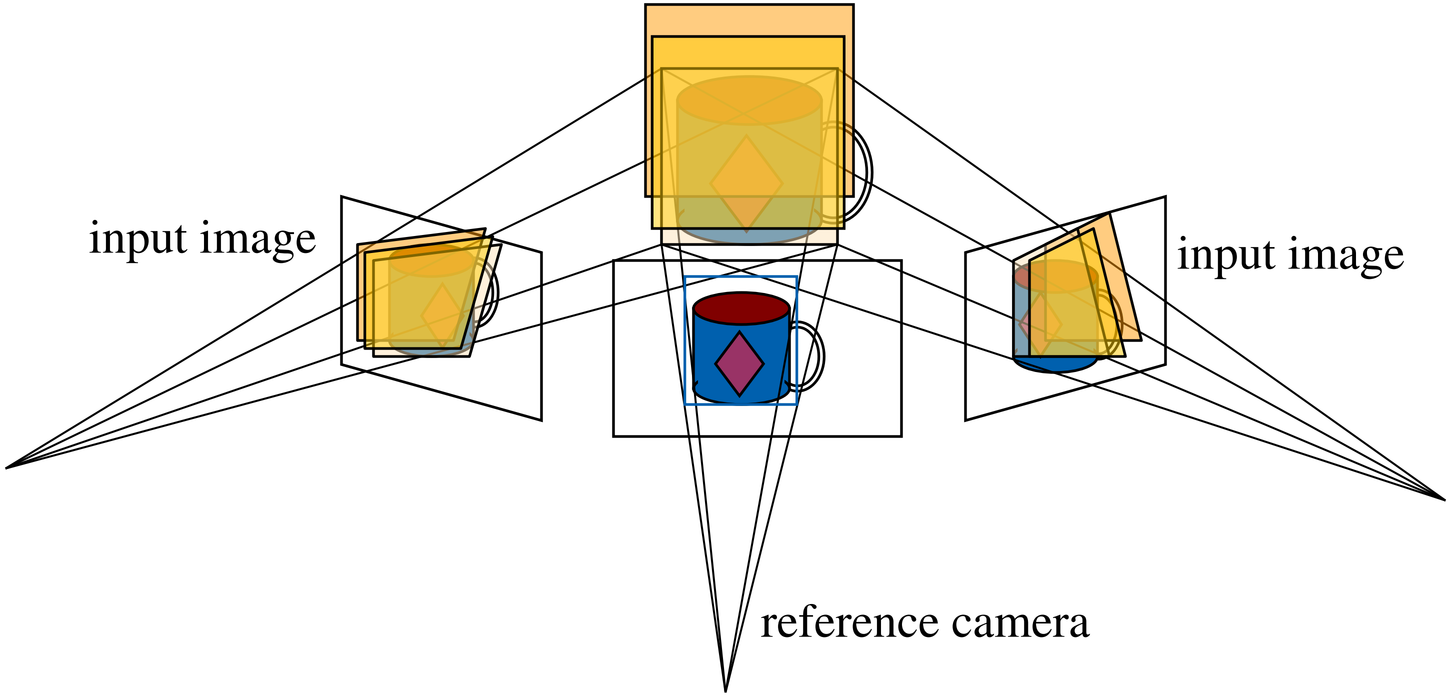
What is stereo vision?

Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape

Lazebnik

Plane Sweep Stereo

- Choose a reference view
- Sweep family of planes at different depths with respect to the reference camera



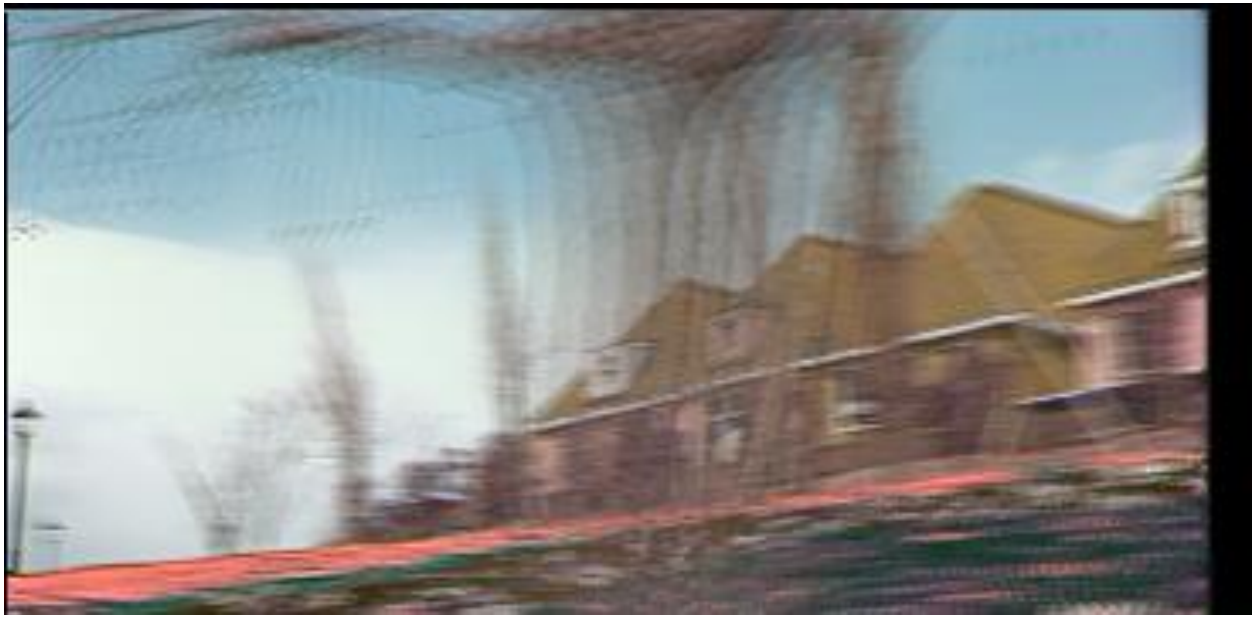
Each plane defines a homography warping each input image into the reference view

R. Collins. A space-sweep approach to true multi-image matching. CVPR 1996.

Lazebnik

Plane Sweep Stereo

- For each depth plane
  - For each pixel in the composite image stack, compute the variance

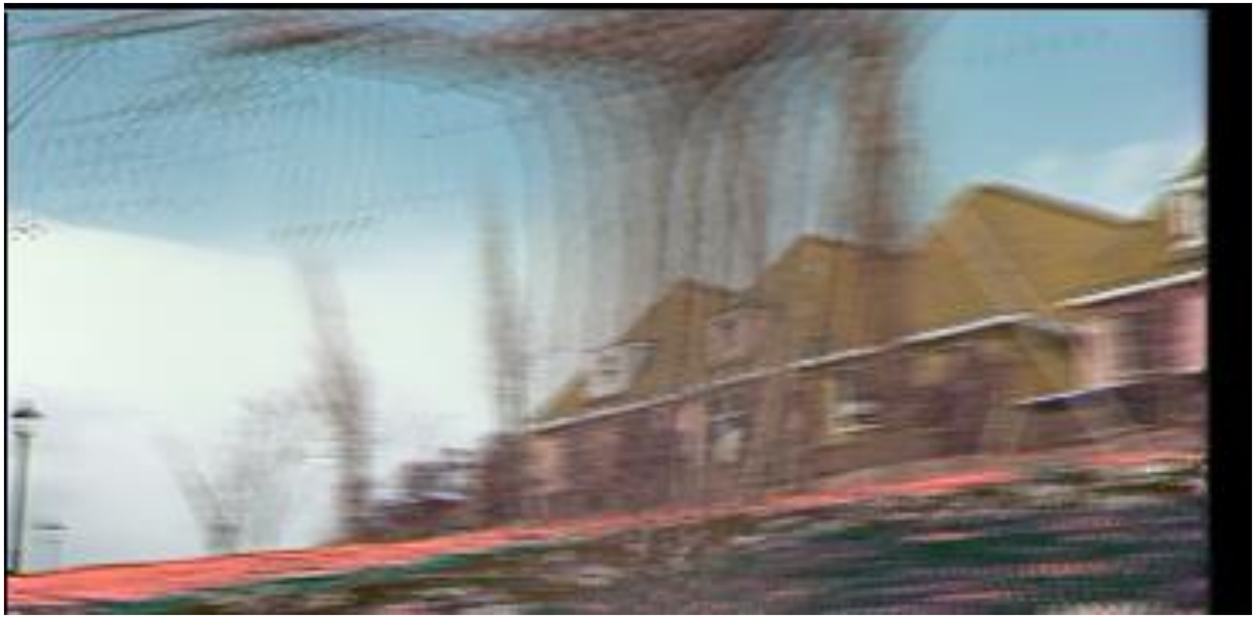


- For each pixel, select the depth that gives the lowest variance

Lazebnik

Plane Sweep Stereo

- For each depth plane
  - For each pixel in the composite image stack, compute the variance



- For each pixel, select the depth that gives the lowest variance

- Can be accelerated using graphics hardware

R. Yang and M. Pollefeys. *Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware*, CVPR 2003

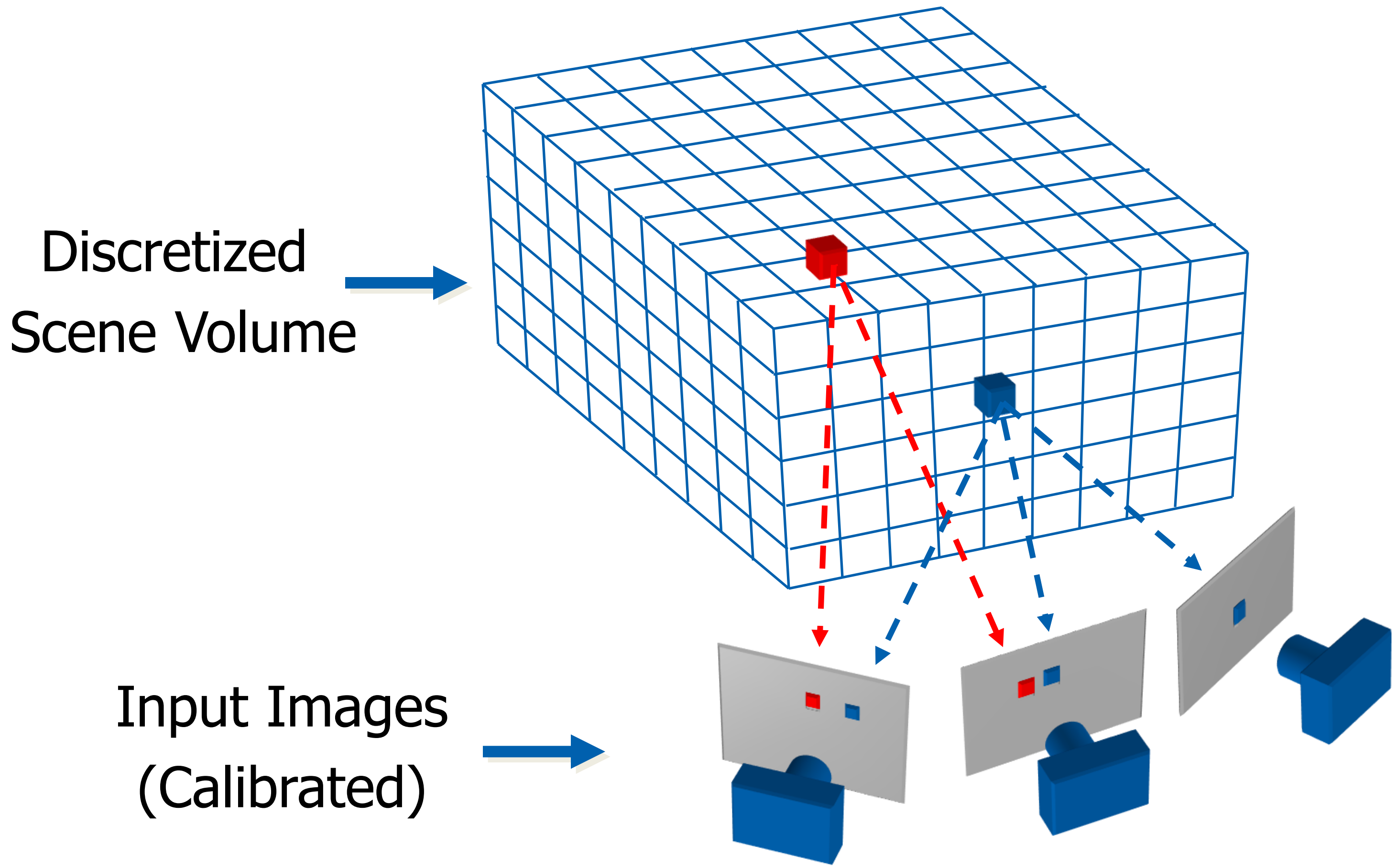
Lazebnik

Volumetric Stereo

- In plane sweep stereo, the sampling of the scene depends on the reference view
- We can use a voxel volume to get a view-independent representation

Lazebnik

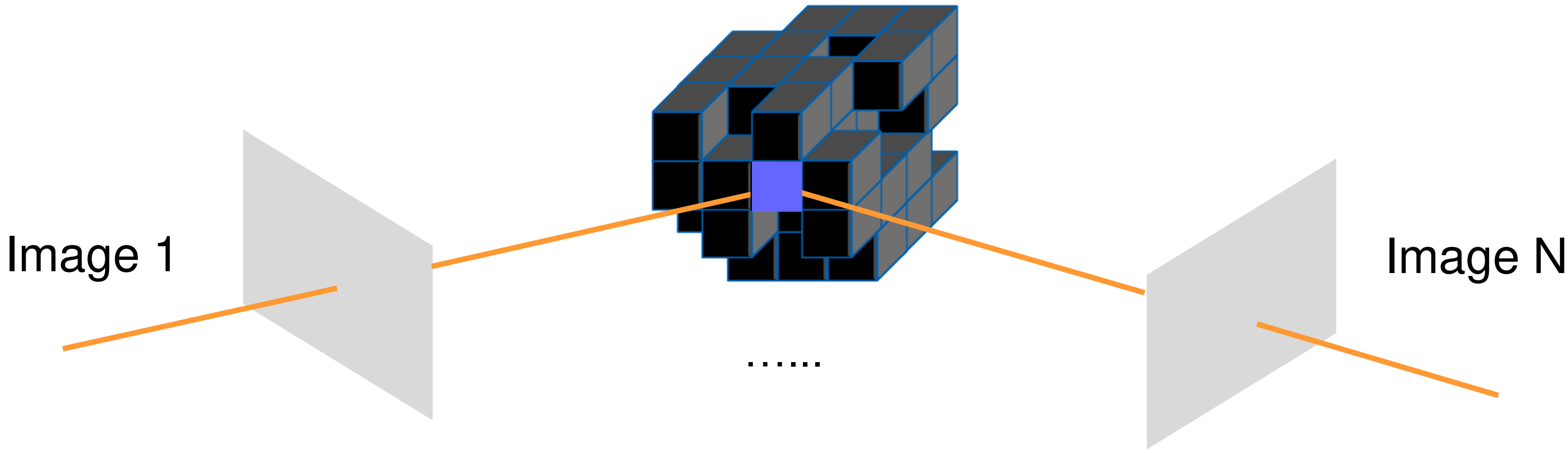
Volumetric Stereo / Voxel Coloring



Goal: Assign RGB values to voxels in V *photo-consistent* with images

Lazebnik

Space Carving



Space Carving Algorithm

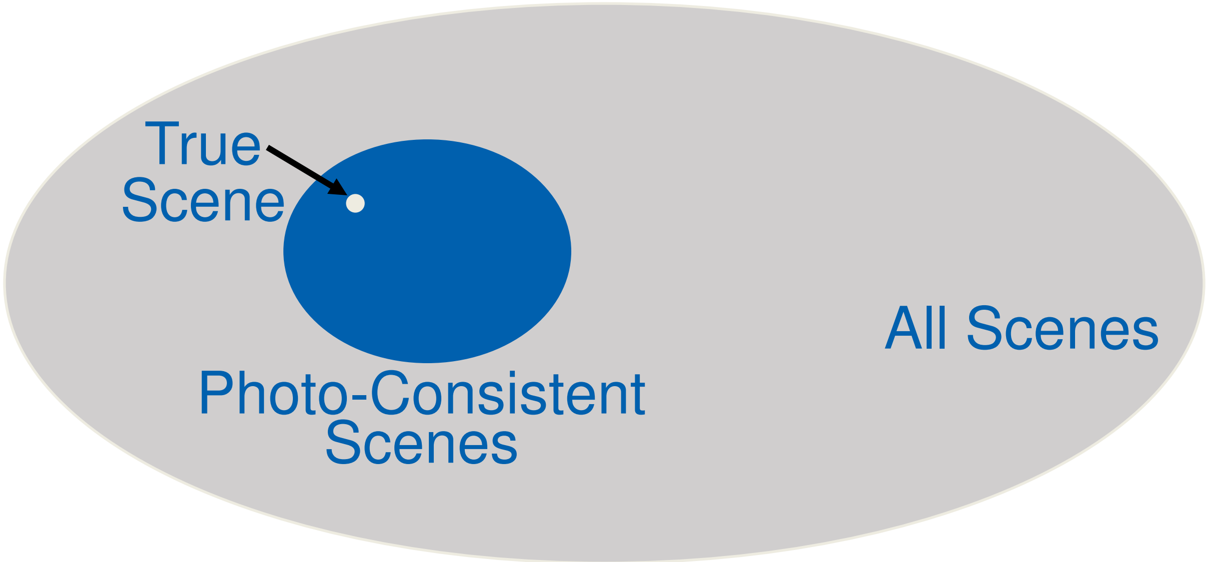
- Initialize to a volume V containing the true scene
- Choose a voxel on the outside of the volume
- Project to visible input images
- Carve if not photo-consistent
- Repeat until convergence

K. N. Kutulakos and S. M. Seitz, **A Theory of Shape by Space Carving**, *ICCV* 1999

Lazebnik

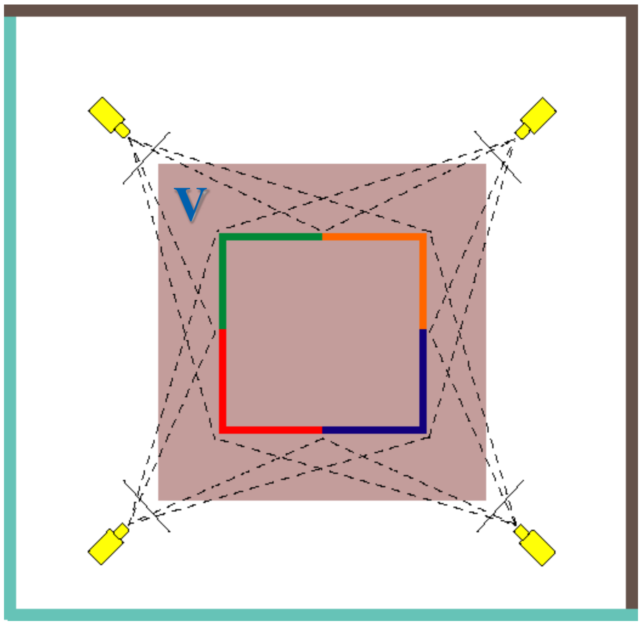
Photo-consistency

- A photo-consistent scene is a scene that exactly reproduces your input images from the same camera viewpoints
- You can't use your input cameras and images to tell the difference between a photo-consistent scene and the true scene



Lazebnik

Which shape do you get?



True Scene

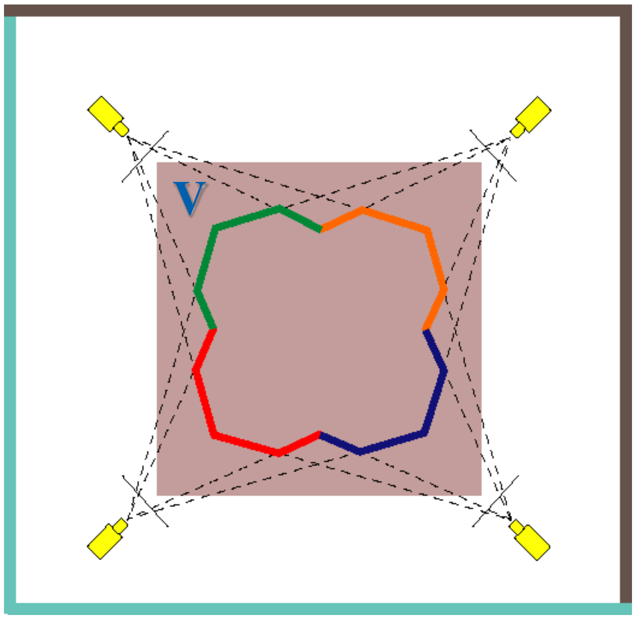


Photo Hull

- The Photo Hull *is the UNION of all photo-consistent scenes in V*
  - It is a photo-consistent scene reconstruction
  - Tightest possible bound on the true scene

Source: S. Seitz

Space Carving Results: African Violet

Input Image (1 of 45)

Reconstruction

Reconstruction

Reconstruction

Source: S. Seitz

Space Carving Results: Hand

Input Image
(1 of 100)

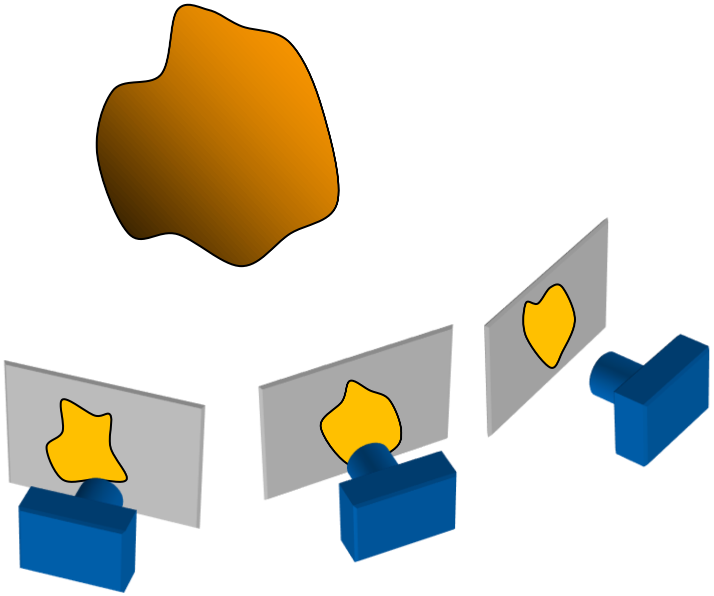
Views of Reconstruction

Lazebnik

Reconstruction from Silhouettes

- The case of binary images: a voxel is photo-consistent if it lies inside the object’s silhouette in all views

Binary Images

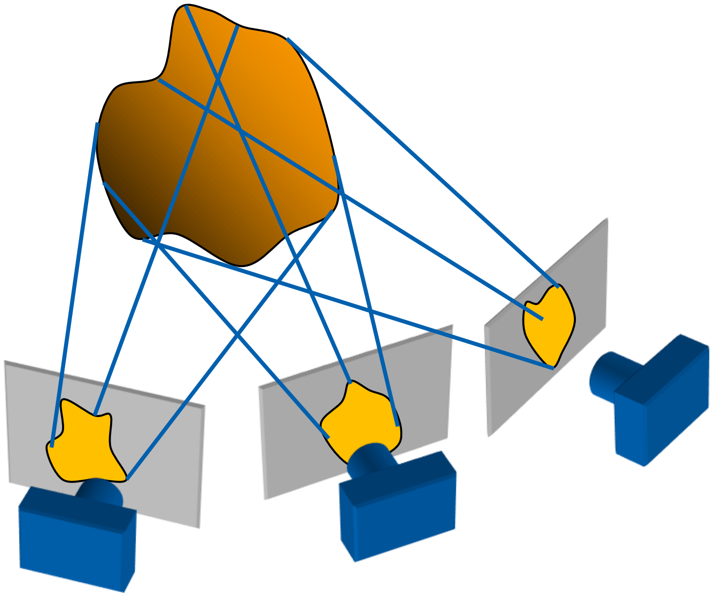


Lazebnik

Reconstruction from Silhouettes

- The case of binary images: a voxel is photo-consistent if it lies inside the object's silhouette in all views

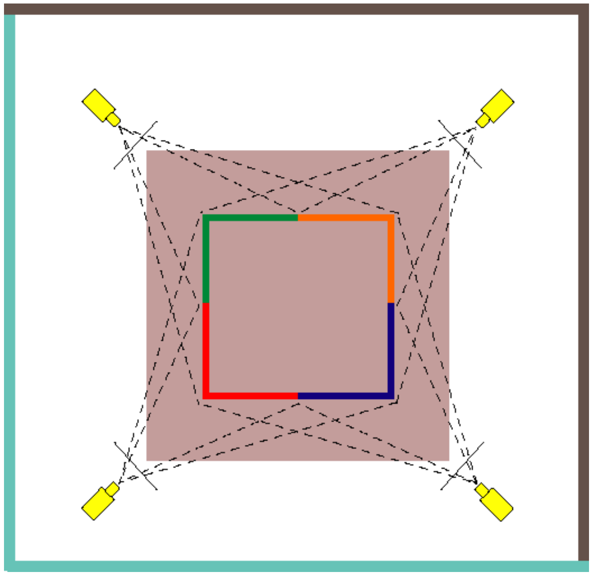
Binary Images



Finding the silhouette-consistent shape (*visual hull*):
- *Backproject* each silhouette
- Intersect backprojected volumes

Lazebnik

Photo-consistency vs. silhouette-consistency



True Scene

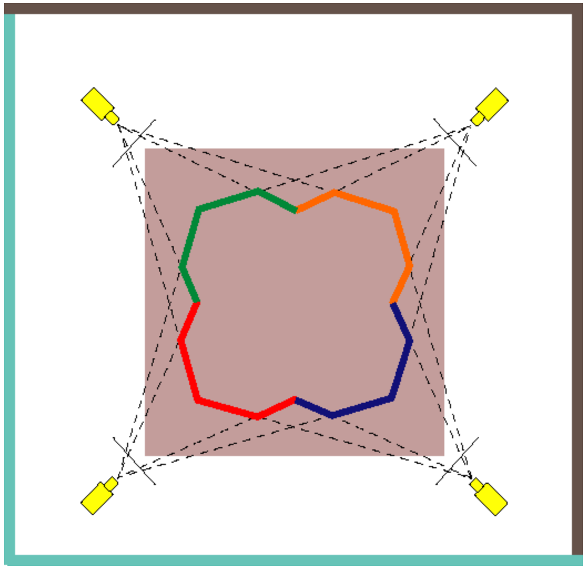
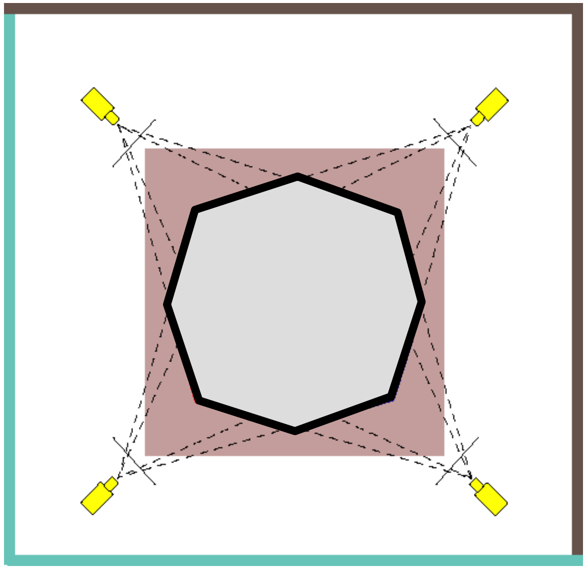


Photo Hull



Visual Hull

Lazebnik

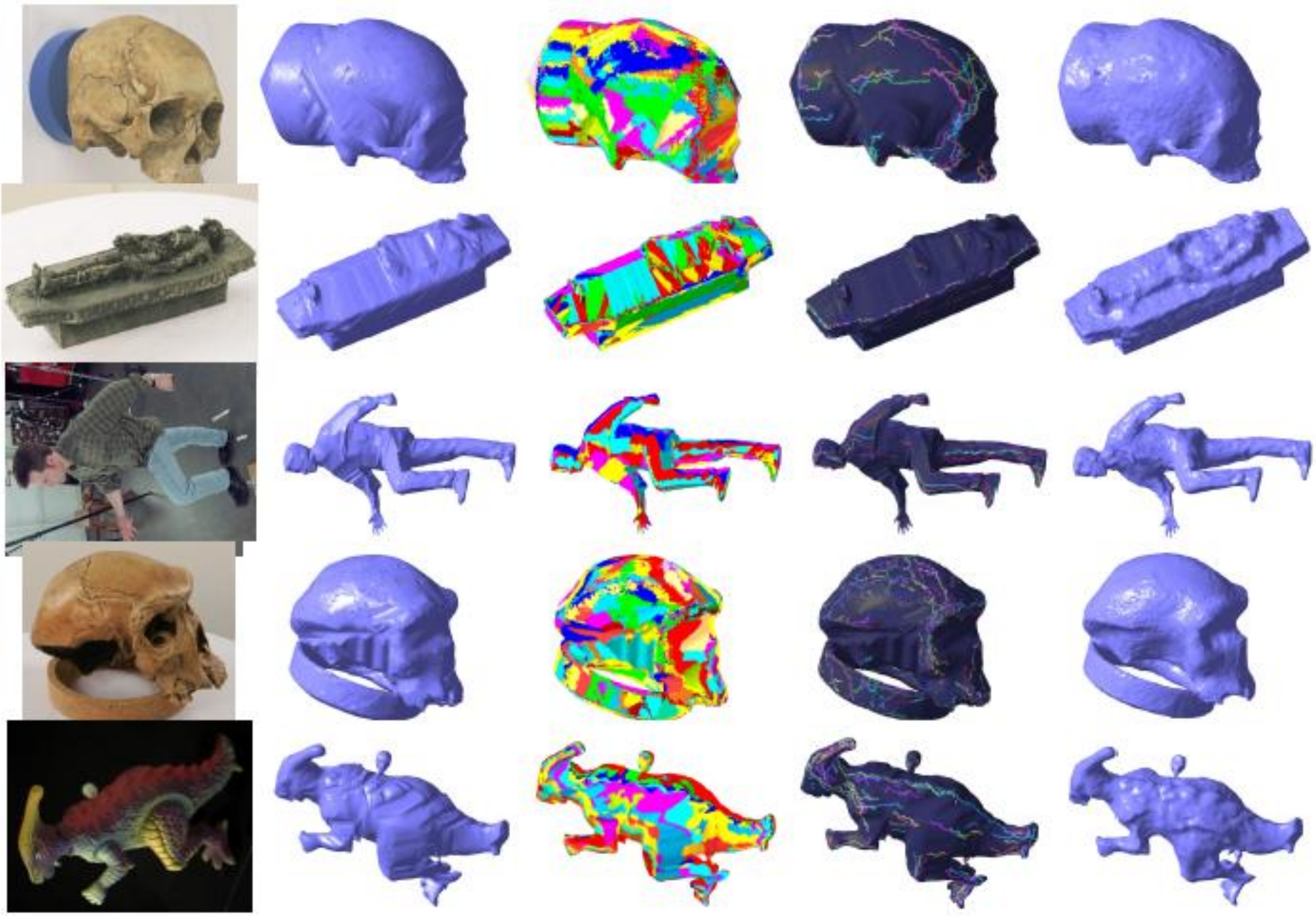
Carved visual hulls

1. Compute visual hull
2. Use dynamic programming to find rims and constrain them to be fixed
3. Carve the visual hull to optimize photo-consistency

Yasutaka Furukawa and Jean Ponce, **Carved Visual Hulls for Image-Based Modeling**, ECCV 2006.

Lazebnik

Carved visual hulls



Yasutaka Furukawa and Jean Ponce, **Carved Visual Hulls for Image-Based Modeling**, ECCV 2006.

Lazebnik

Carved visual hulls: Pros and cons

- Pros
  - Visual hull gives a reasonable initial mesh that can be iteratively deformed
- Cons
  - Need silhouette extraction
  - Have to compute a lot of points that don't lie on the object
  - Finding rims is difficult
  - The carving step can get caught in local minima
- Possible solution: use sparse feature correspondences as initialization

Lazebnik

From feature matching to dense stereo

1. Extract features
2. Get a sparse set of initial matches
3. Iteratively expand matches to nearby locations
4. Use visibility constraints to filter out false matches
5. Perform surface reconstruction

Yasutaka Furukawa and Jean Ponce, **Accurate, Dense, and Robust Multi-View Stereopsis**, CVPR 2007.

Lazebnik

From feature matching to dense stereo

Yasutaka Furukawa and Jean Ponce, **Accurate, Dense, and Robust Multi-View Stereopsis**, CVPR 2007.

Lazebnik

Towards Internet-Scale Multi-View Stereo

Yasutaka Furukawa, Brian Curless, Steven M. Seitz and Richard Szeliski, Towards Internet-scale Multi-view Stereo, CVPR 2010.

Lazebnik

Fast stereo for Internet photo collections

- Start with a cluster of registered views
- Obtain a depth map for every view using plane sweeping stereo with normalized cross-correlation

Frahm et al. Building Rome on a Cloudless Day, ECCV 2010.

Lazebnik

Plane sweeping stereo

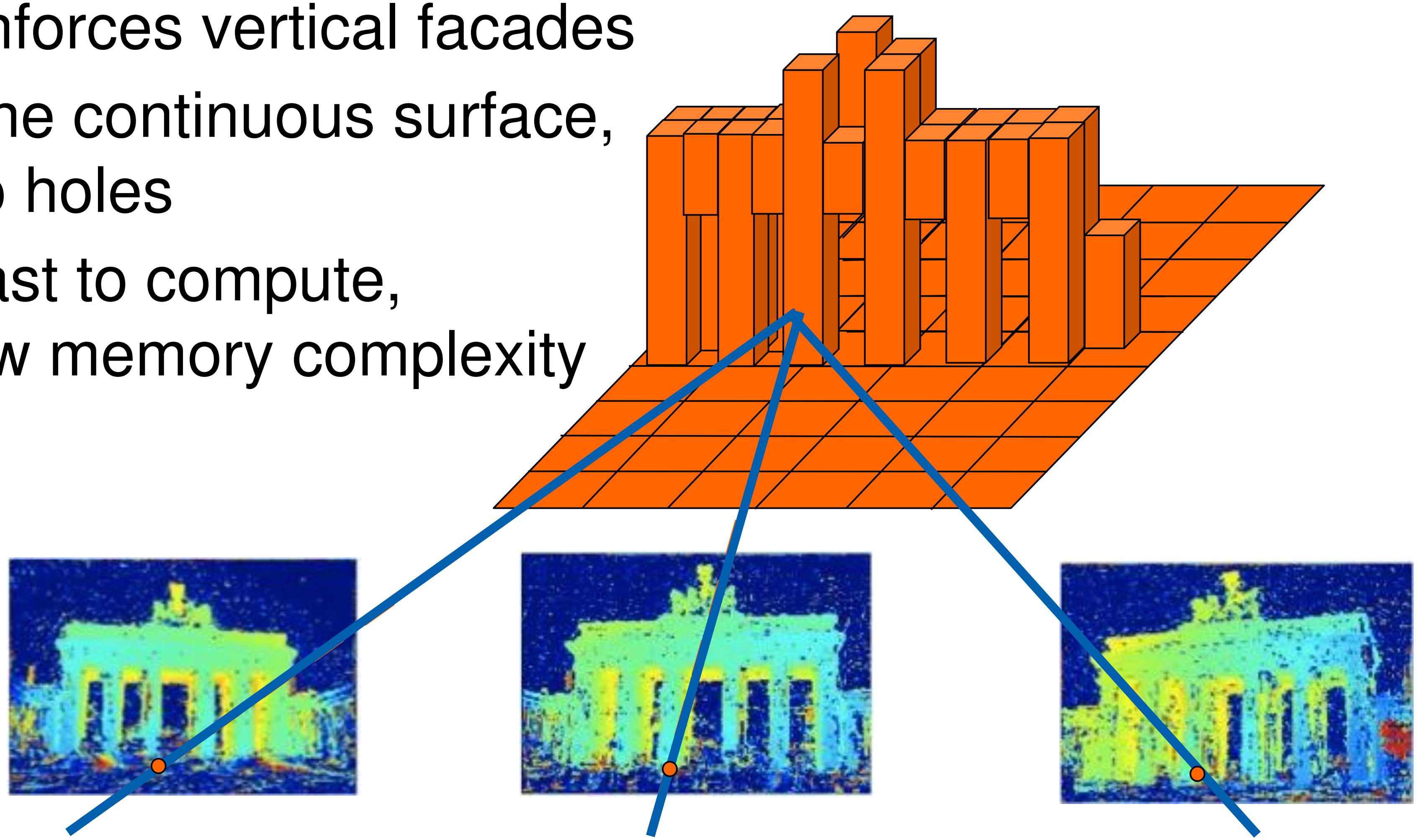
- Need to register individual depth maps into a single 3D model
- Problem: depth maps are very noisy

Frahm et al. Building Rome on a Cloudless Day, ECCV 2010.

Lazebnik

Robust stereo fusion using a heightmap

- Enforces vertical facades
- One continuous surface, no holes
- Fast to compute, low memory complexity



David Gallup, Marc Pollefeys, Jan-Michael Frahm, “3D Reconstruction using an n-Layer Heightmap”, DAGM 2010

Lazebnik

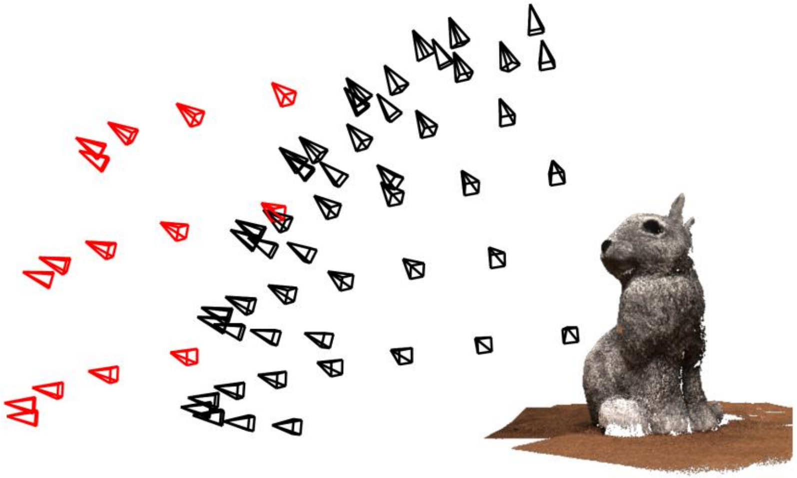
Results

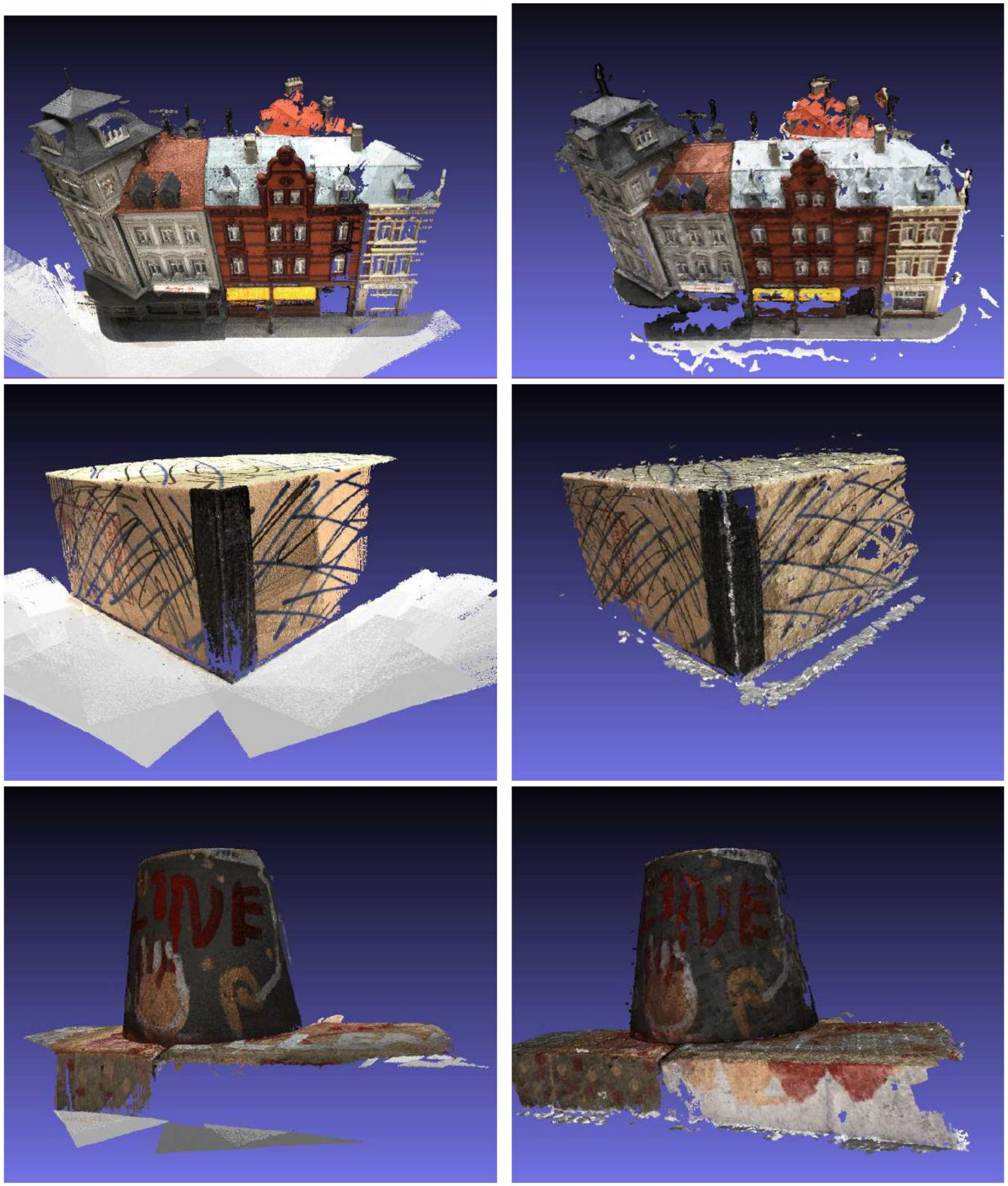
Frahm et al., “Building Rome on a Cloudless Day,” ECCV 2010.

Lazebnik

Results

Reconstruction with CNNs (CV-II)





Mengqi et al., “SurfaceNet: An End-to-end 3D Neural Network for Multiview Stereopsis” ICCV 2017.

UNIVERSITÄT BONN