

Camera Arrays and Light Field Photography

Computational Photography

CSE 273

Lecture 8

Announcements

- Assignment 4 is due Feb 7, 11:59 PM
- Initial project proposal is due Feb 18
- Revised project proposal is due Feb 25
- Initial project presentation slides are due Feb 25
- Draft project report is due Mar 10
- Final project report is due Mar 17
- Final project presentation slides are due Mar 20

Camera arrays



The Matrix
(bullet time)

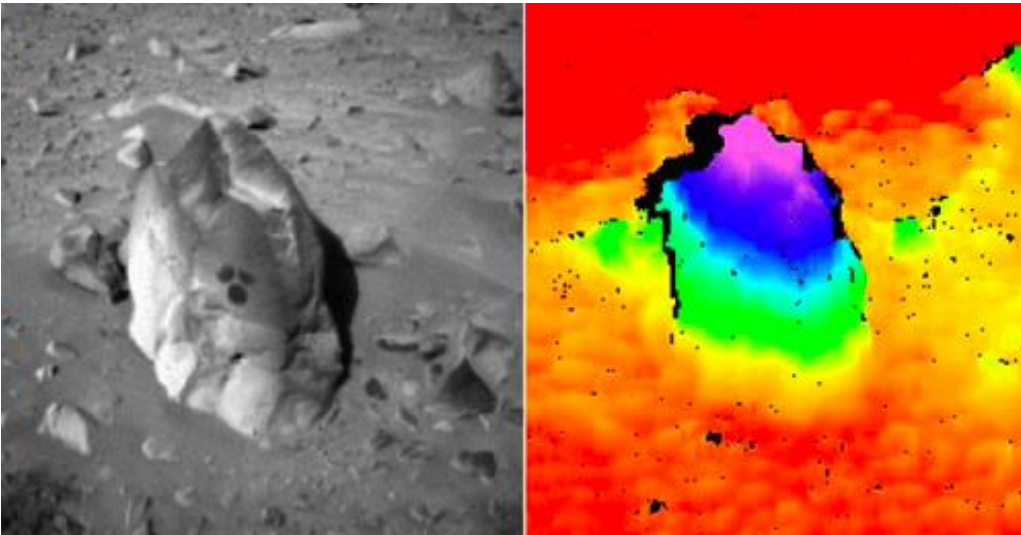
Application:
view synthesis

Camera arrays

- 2 cameras
 - 2 views, binocular stereo
- 3 cameras
 - 3 views, trinocular stereo
- n cameras
 - n views, multiview stereo

2 cameras

- Mars exploratory rovers
 - Spirit and Opportunity



Reconstructed scene

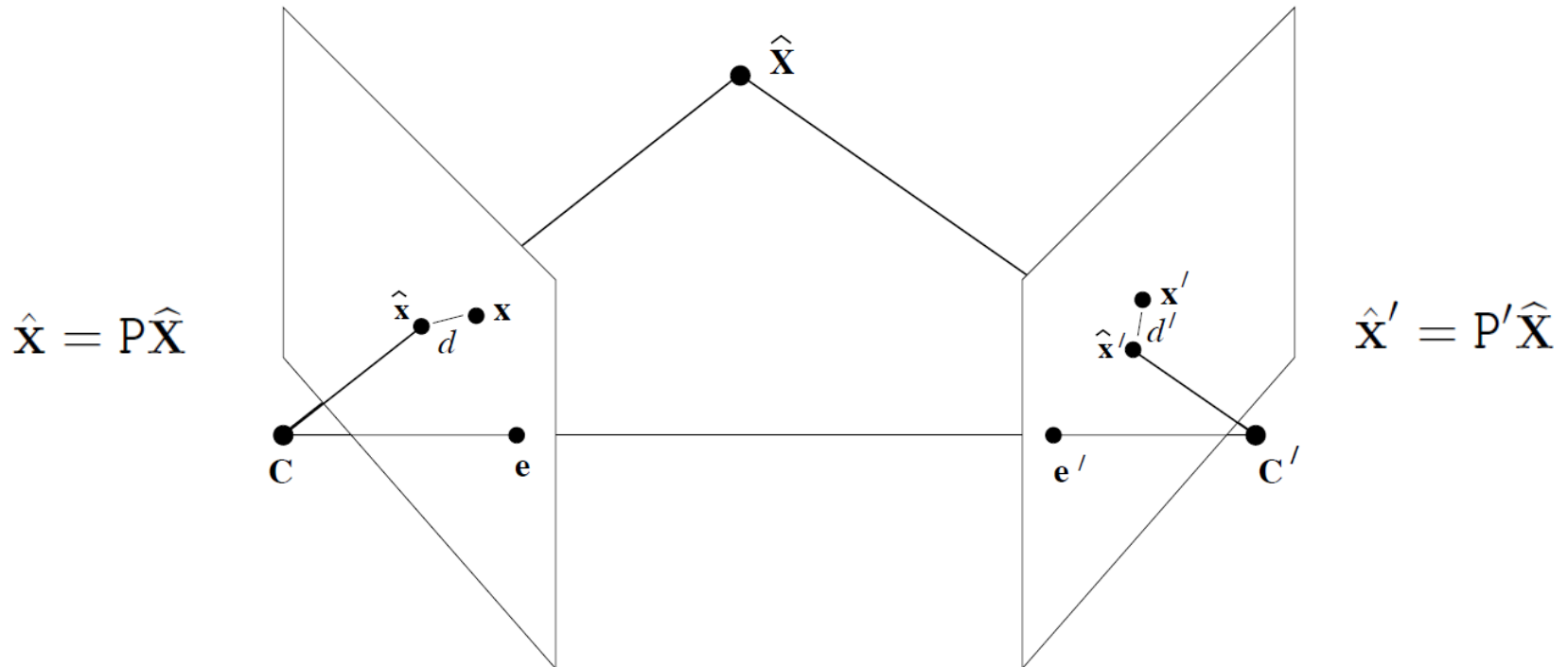
2 cameras

- Curiosity rover
 - Navigation cameras, 45 degrees field of view
 - Hazard avoidance cameras, 4 pairs, 120 degrees field of view



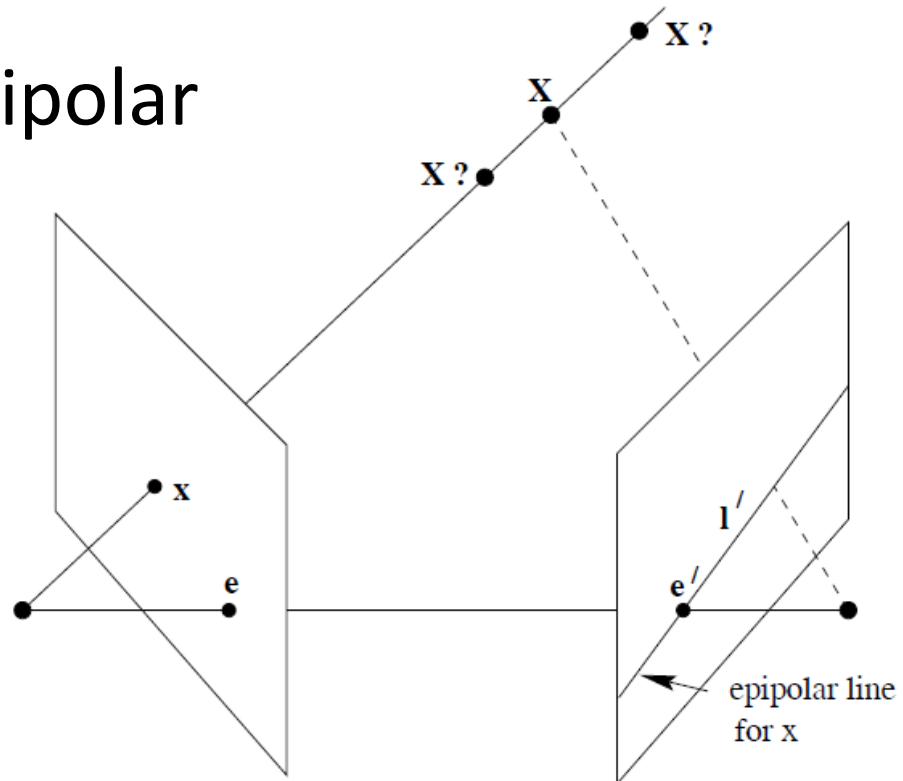
2 view triangulation

- Given two image measurements x and x' , estimate scene point $\hat{X}$



Point correspondences

- Potential matches for $\mathbf{x}$ have to lie on the corresponding epipolar line l'

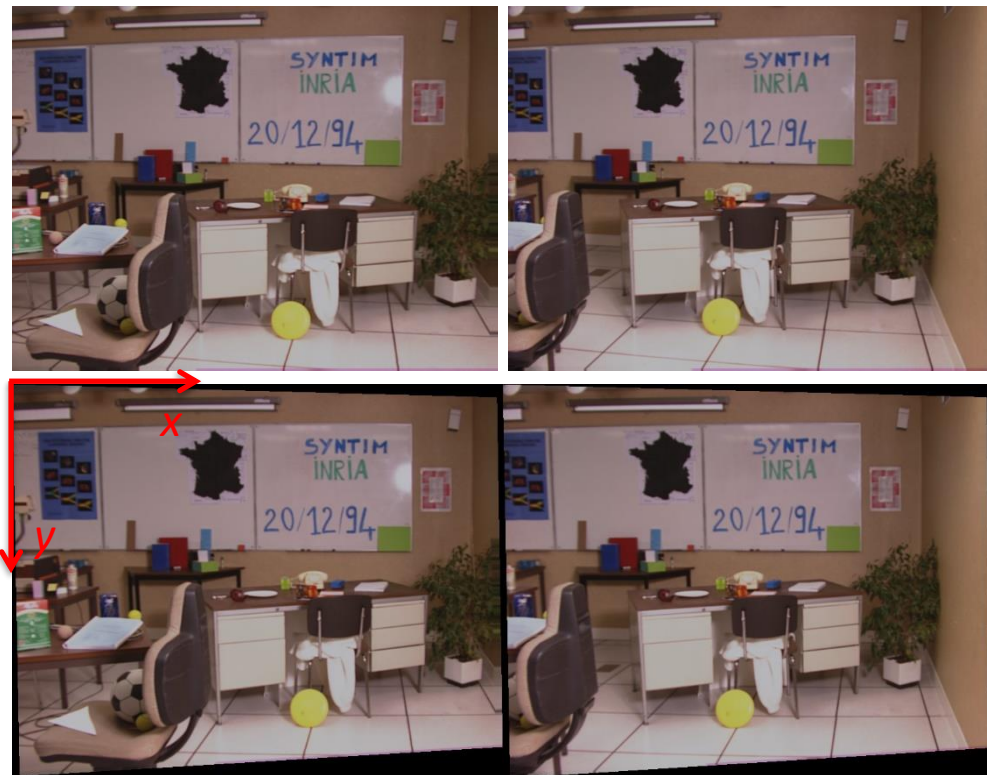


Definitions

- Rectify: to make straight, make right
- Rectification: the action or process of rectifying
- Image rectification: the process of projecting an image to a specific plane such that the image is “made straight” or “made right”

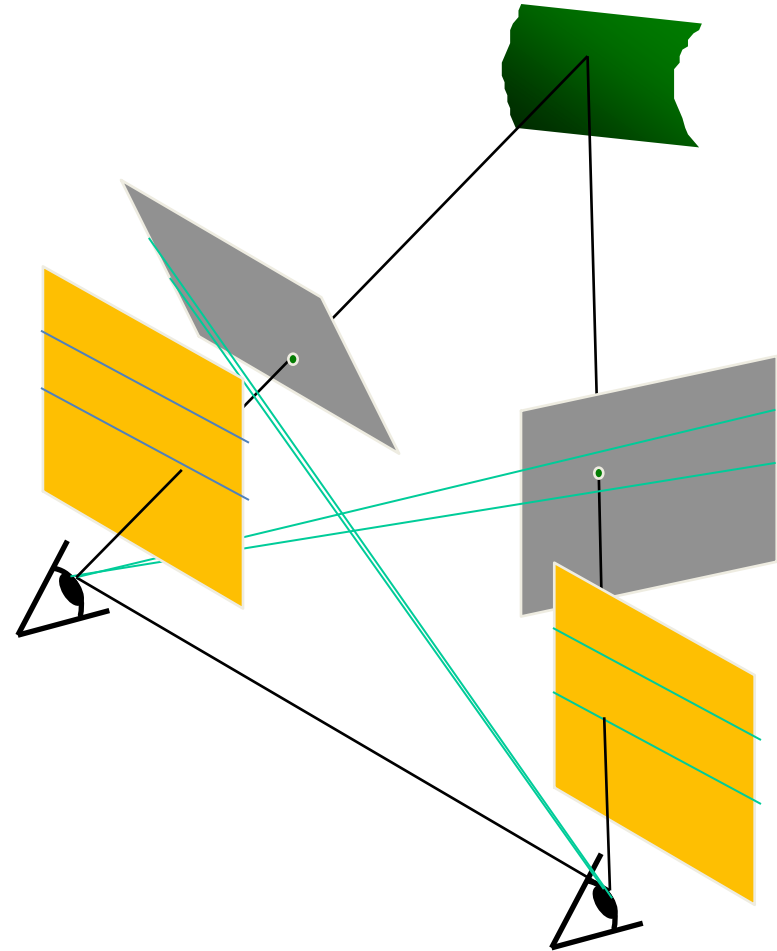
Two views, epipolar rectification

- Objective: epipolar lines run parallel with the x-axis and are aligned between two views (no y disparity)



Epipolar rectification

- Given a pair of images, transform both images so that epipolar lines are image rows
- Create pair of virtual cameras
 - The virtual cameras have the same camera centers as real cameras
 - Both virtual cameras have the same:
 - Camera rotation matrix $\mathbf{R}$
 - Camera calibration matrix $\mathbf{K}$



Epipolar rectification

- Given calibrated stereo cameras (i.e., $\mathbf{K}_1$, $\mathbf{R}_1$, $\mathbf{t}_1$, $\mathbf{K}_2$, $\mathbf{R}_2$, $\mathbf{t}_2$) determine the (same) rotation matrix $\mathbf{R}$ and (same) calibration matrix $\mathbf{K}$ of the virtual cameras
- To minimize image distortion:
 - For the calibration matrix
 - For principal point, set x_0 and y_0 to average of input x_0 and y_0 values, respectively
 - Set two focal length parameters to average of all four input focal length parameters (results in square pixels)
 - Set skew to 0
 - For the rotation matrix $\mathbf{R}$, interpolate halfway between the two 3D rotations embodied by $\mathbf{R}_1$ and $\mathbf{R}_2$

Epipolar rectification

- For the rotation matrix $\mathbf{R}$, interpolate halfway between the two 3D rotations embodied by $\mathbf{R}_1$ and $\mathbf{R}_2$
 - Rotating a camera does not change its center $\mathbf{C}$, but does change its translation $\mathbf{t}$

$$\tilde{\mathbf{0}}_{\text{cam}} = \mathbf{R}\tilde{\mathbf{C}} + \mathbf{t}$$

$$-\mathbf{R}\tilde{\mathbf{C}} = \mathbf{t}$$

$$\tilde{\mathbf{C}} = -\mathbf{R}^\top \mathbf{t}$$

$$\mathbf{x} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{X}$$

$$\mathbf{x} = \mathbf{K}[\mathbf{R} \mid -\mathbf{R}\tilde{\mathbf{C}}]\mathbf{X}$$

$$\mathbf{x} = \mathbf{K}\mathbf{R}[\mathbf{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}$$

$$\tilde{\mathbf{C}} = -\mathbf{R}_{\text{virtual}}^\top \mathbf{t}_{\text{virtual}}$$

$$\tilde{\mathbf{C}} = -\mathbf{R}_{\text{real}}^\top \mathbf{t}_{\text{real}}$$

$$-\mathbf{R}_{\text{virtual}}^\top \mathbf{t}_{\text{virtual}} = -\mathbf{R}_{\text{real}}^\top \mathbf{t}_{\text{real}}$$

$$\mathbf{t}_{\text{virtual}} = \mathbf{R}_{\text{virtual}}\mathbf{R}_{\text{real}}^\top \mathbf{t}_{\text{real}}$$

Rectification transformation matrices

- Transformation from image acquired by real camera to image acquired by virtual camera

$$\mathbf{x}_{\text{real}} = \mathbf{K}_{\text{real}} \mathbf{R}_{\text{real}} [\mathbf{I} \mid -\tilde{\mathbf{C}}] \mathbf{X}$$

$$\mathbf{R}_{\text{real}}^{\top} \mathbf{K}_{\text{real}}^{-1} \mathbf{x}_{\text{real}} = [\mathbf{I} \mid -\tilde{\mathbf{C}}] \mathbf{X}$$

$$\mathbf{x}_{\text{virtual}} = \mathbf{K}_{\text{virtual}} \mathbf{R}_{\text{virtual}} [\mathbf{I} \mid -\tilde{\mathbf{C}}] \mathbf{X}$$

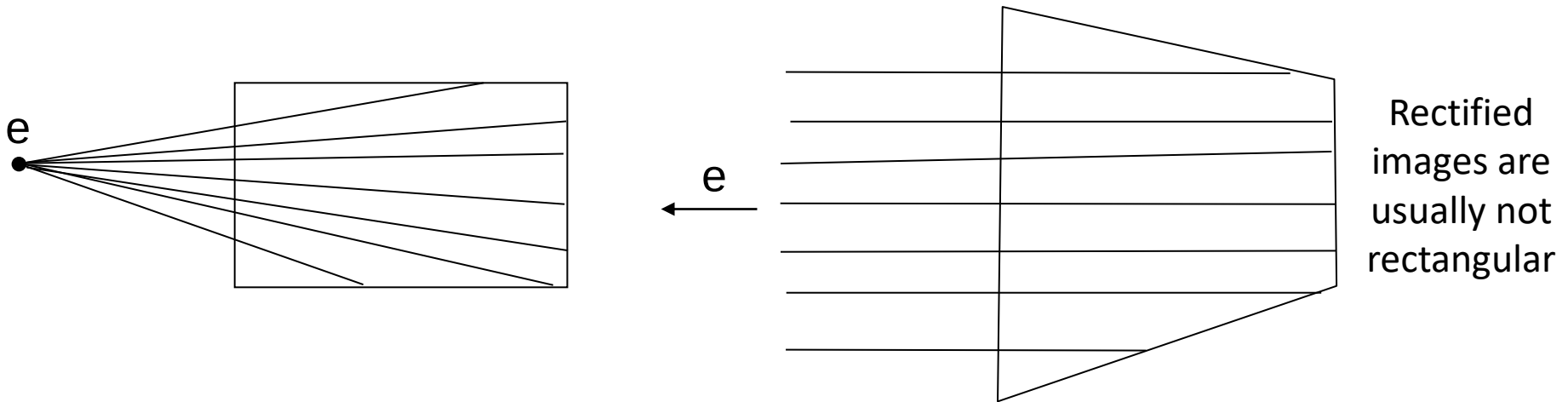
$$\mathbf{x}_{\text{virtual}} = \mathbf{K}_{\text{virtual}} \mathbf{R}_{\text{virtual}} \mathbf{R}_{\text{real}}^{\top} \mathbf{K}_{\text{real}}^{-1} \mathbf{x}_{\text{real}}$$

$$\mathbf{x}_{\text{virtual}} = \mathbf{H} \mathbf{x}_{\text{real}}, \text{ where } \mathbf{H} = \mathbf{K}_{\text{virtual}} \mathbf{R}_{\text{virtual}} \mathbf{R}_{\text{real}}^{\top} \mathbf{K}_{\text{real}}^{-1}$$

$$\begin{bmatrix} x_{\text{virtual}} \\ y_{\text{virtual}} \\ w_{\text{virtual}} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_{\text{real}} \\ y_{\text{real}} \\ w_{\text{real}} \end{bmatrix}$$

Epipolar rectification

- Maps epipole to homogeneous 2D point $(1, 0, 0)^T$, a point at infinity on the x -axis
- Rectified image pair has the same scale in y direction

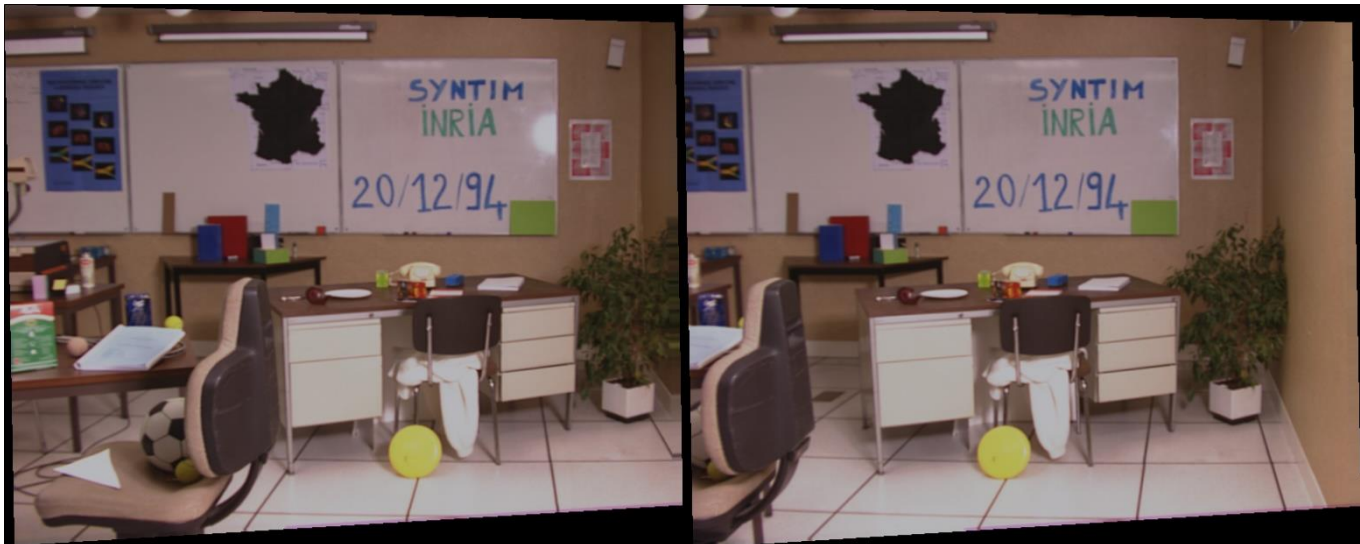


Rectification

Original

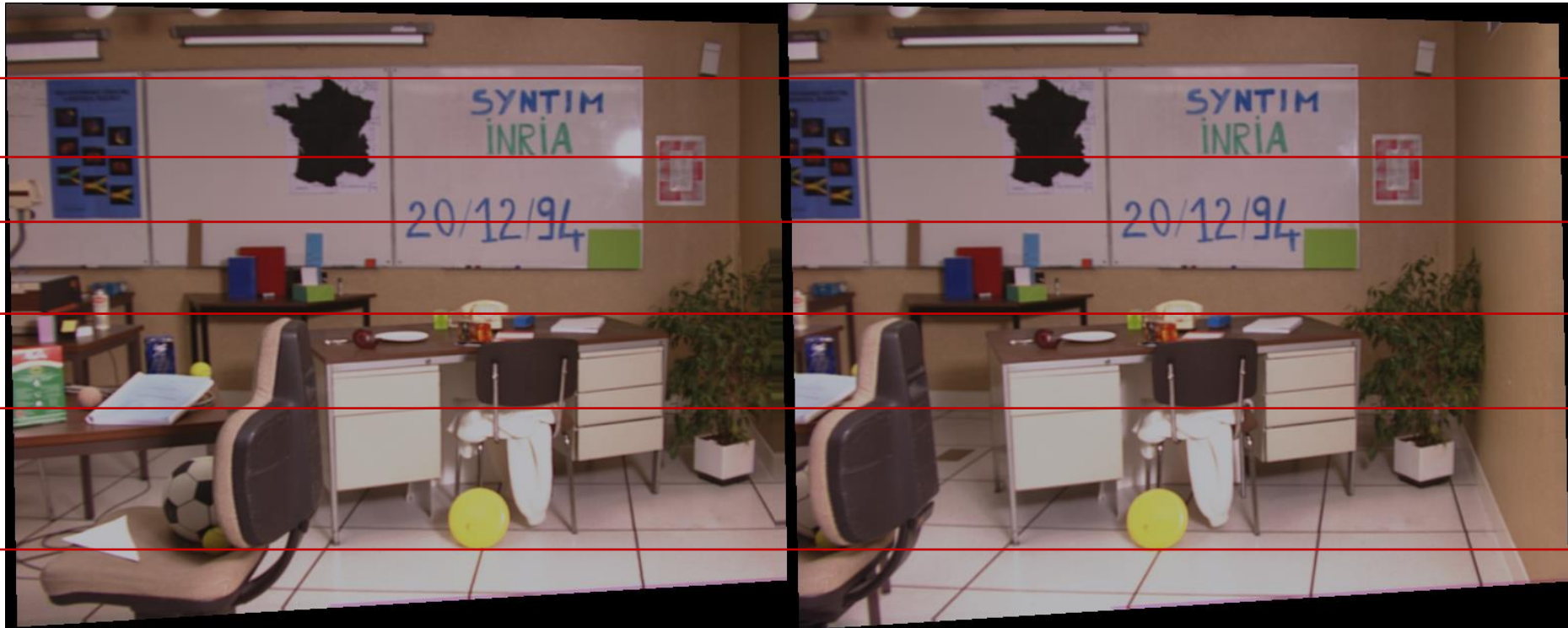


Rectified



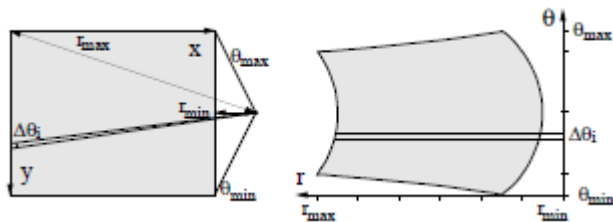
Rectification

- Epipolar lines



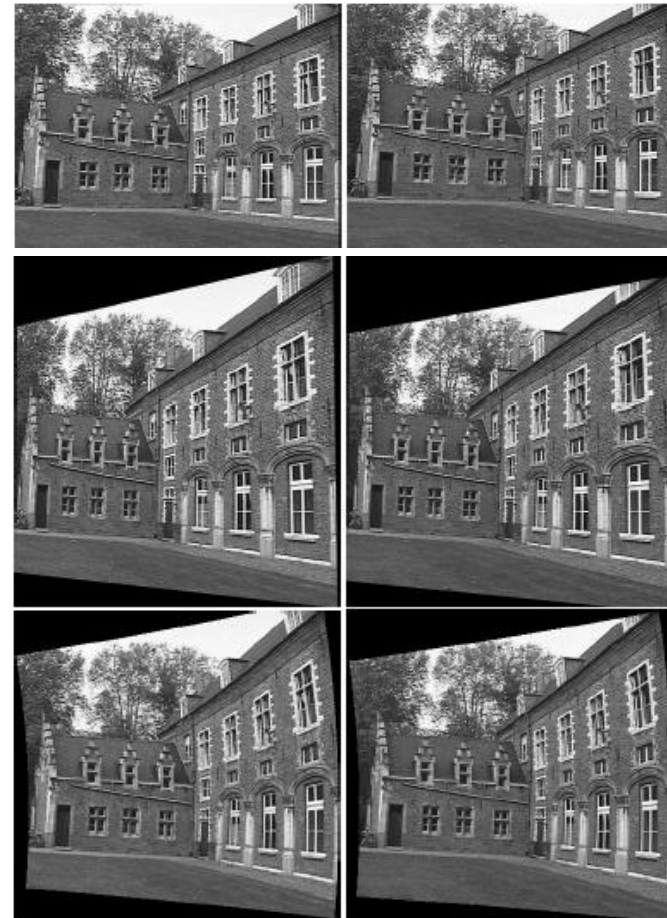
Epipolar rectification

- Polar rectification
 - Alternative method that minimizes pixel distortion

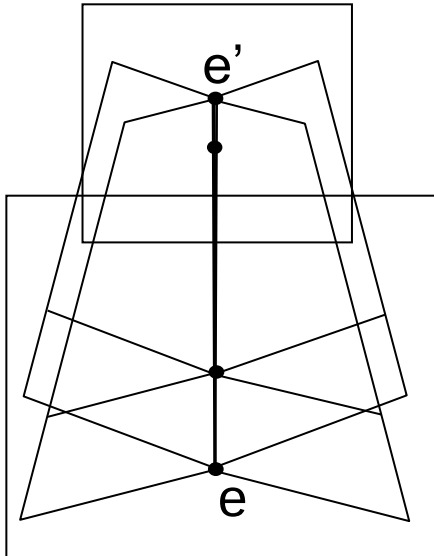


Planar transformation-based rectification

Polar rectification



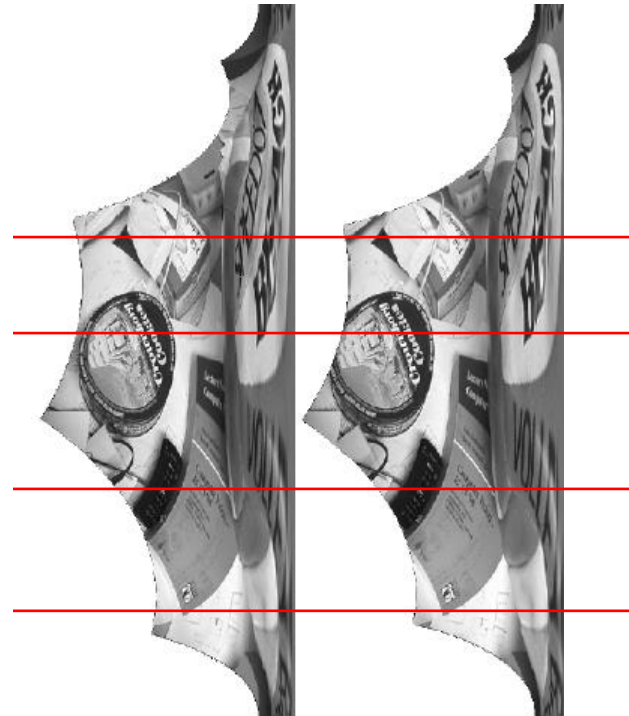
Polar rectification



Epipoles are in images
(white dot on ball)

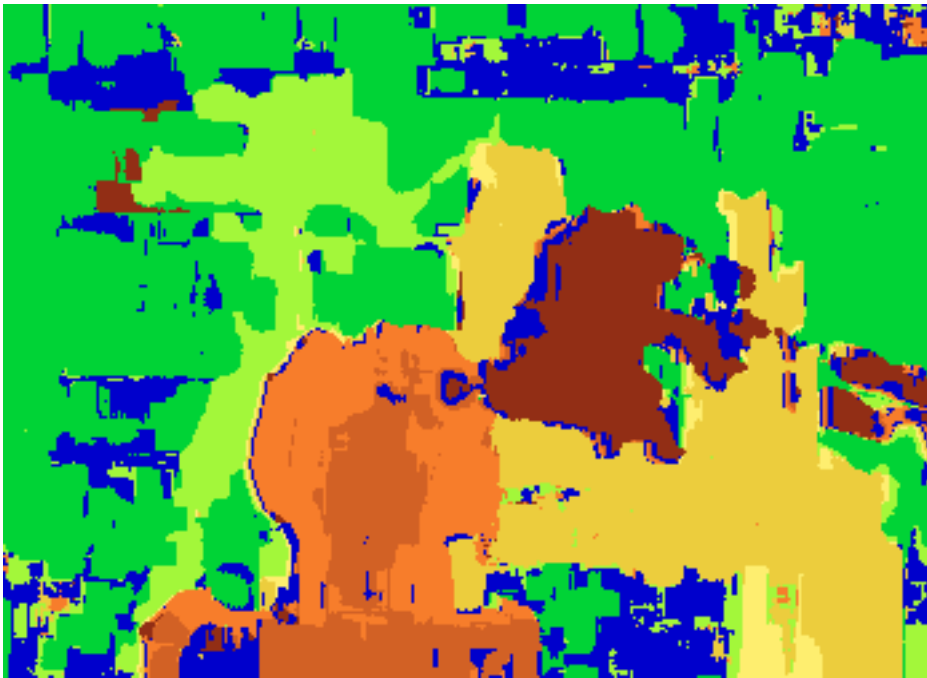


Planar transformation-based
rectification is not possible



Epipolar lines

Reconstruction results with window correlation



Window-based matching
(best window size)



Ground truth

(Seitz)

Reconstruction results with better method



Using global optimization

Boykov et al., [Fast Approximate Energy Minimization via Graph Cuts](#),
International Conference on Computer Vision, September 1999.

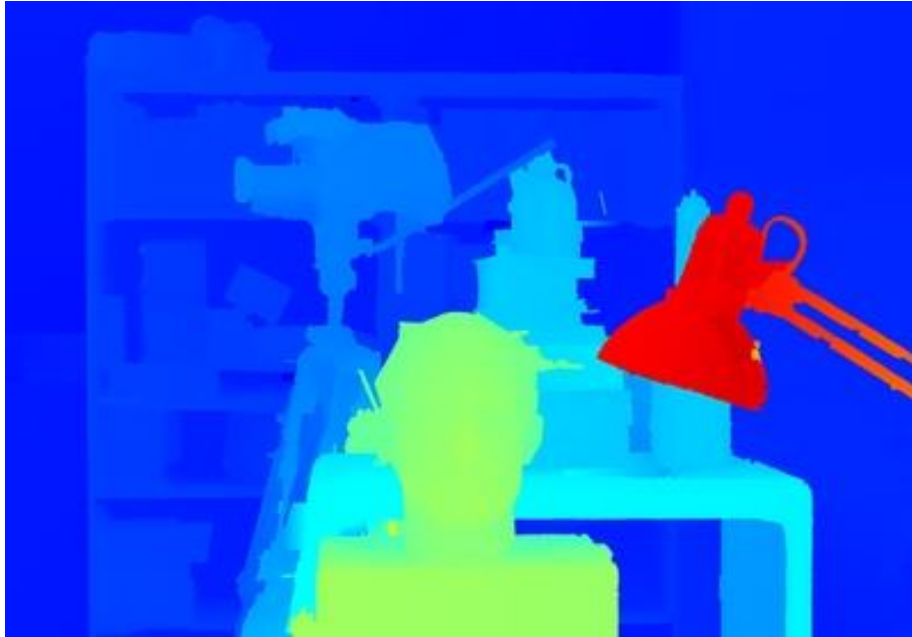
(Seitz)

CSE 273, Winter 2024

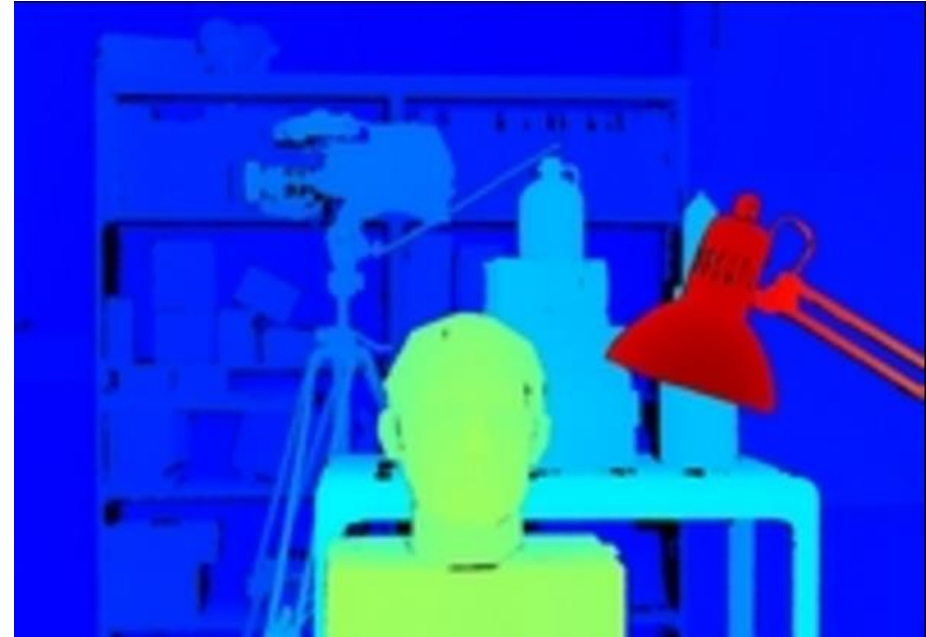


Ground truth

Reconstruction results with state of the art method



Using neural networks



Ground truth

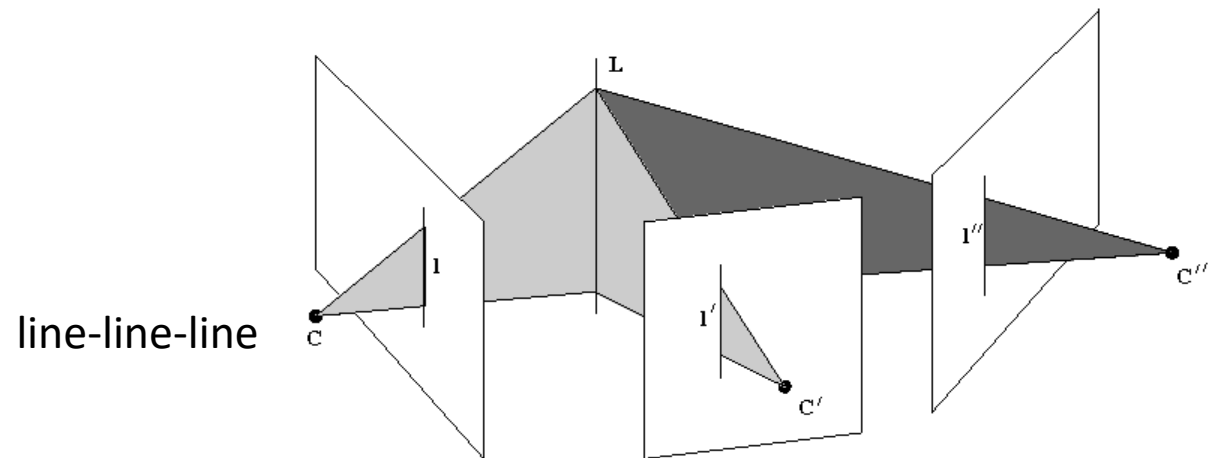
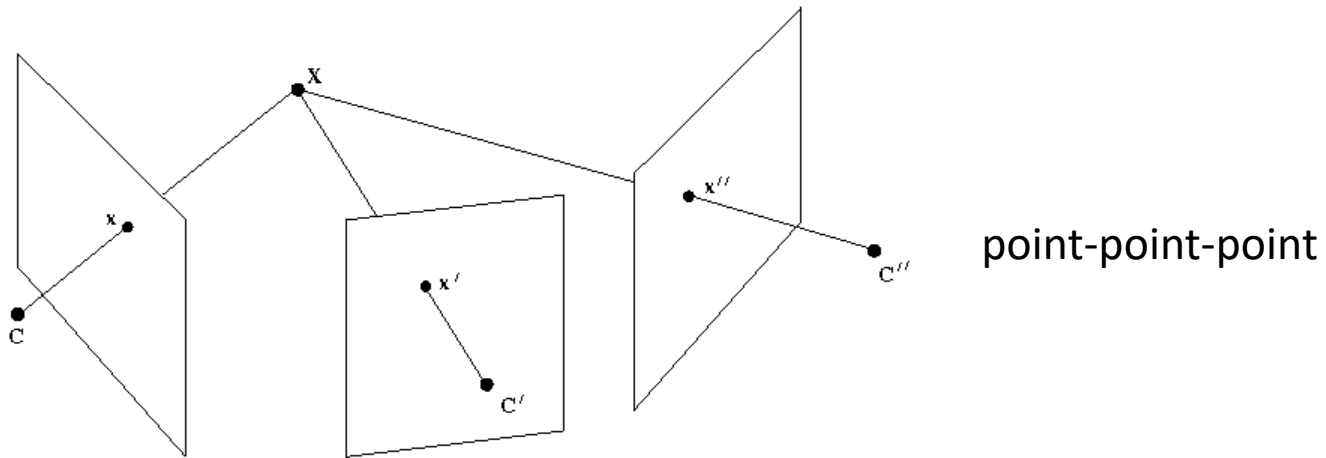
S. Drouyer, S. Beucher, M. Bilodeau, M. Moreaud, and L. Sorbier.
[Sparse stereo disparity map densification using hierarchical image segmentation](#). 13th International Symposium on Mathematical Morphology.

3 cameras

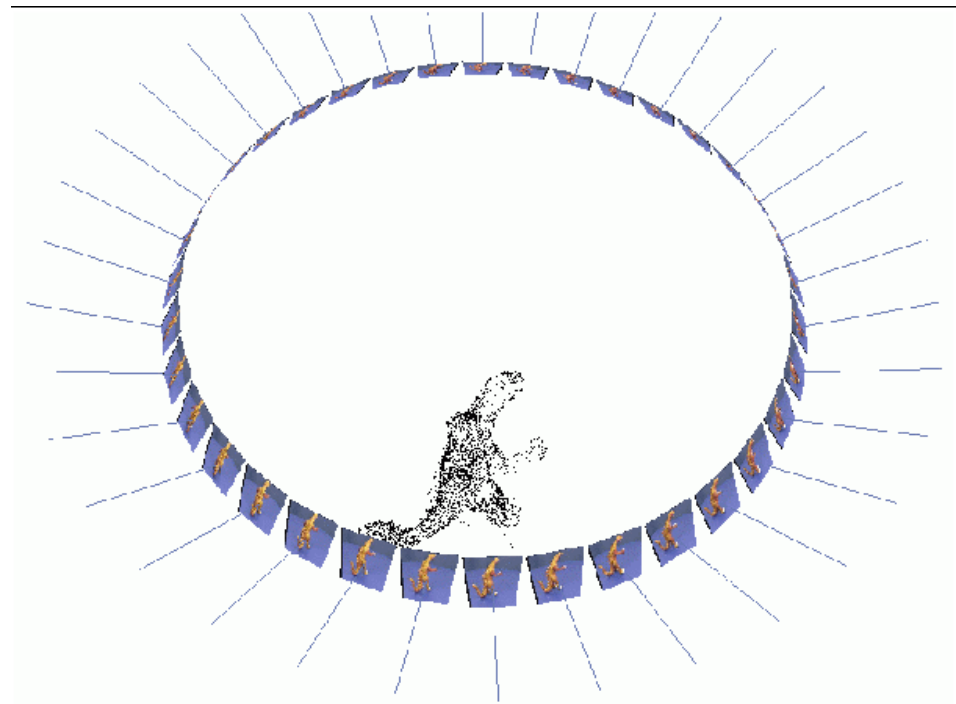
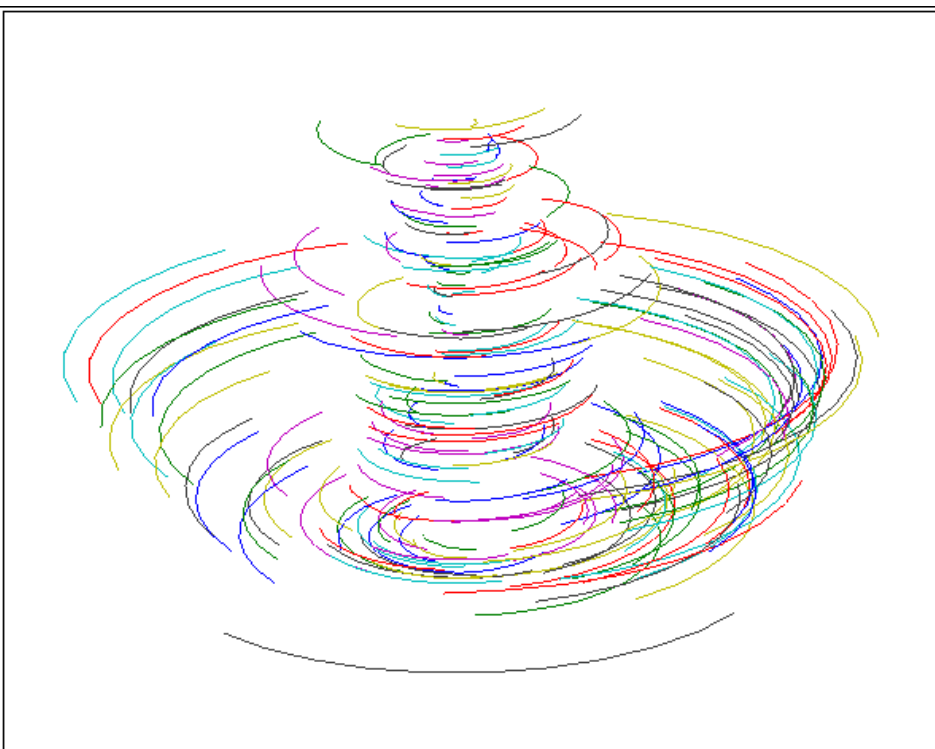
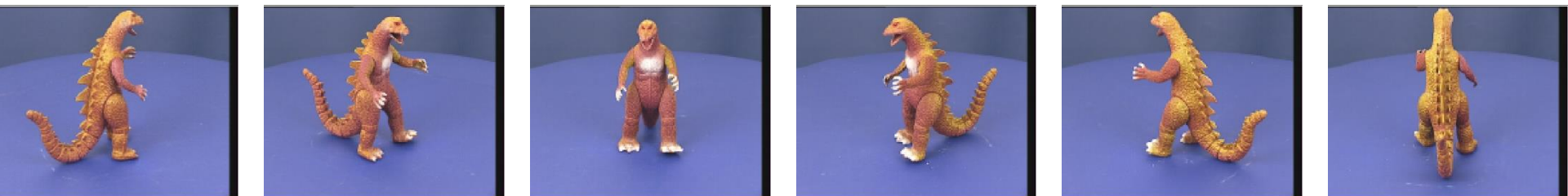
- 3 views, trinocular stereo



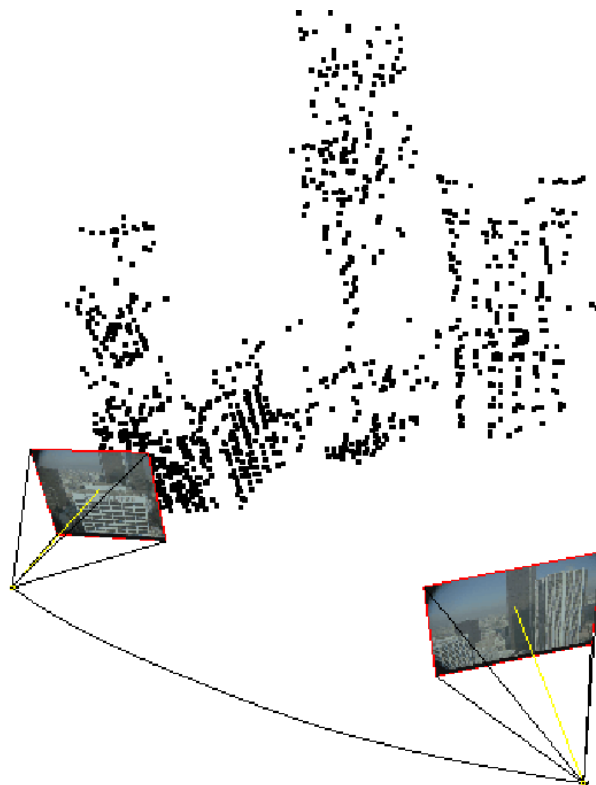
3 view triangulation



1 camera, n views



1 camera, n views



1 camera, n views

- Example results



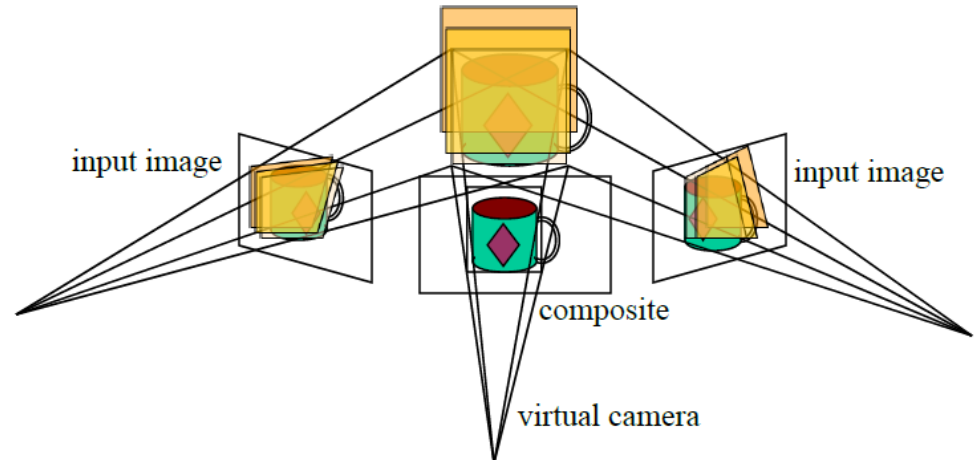
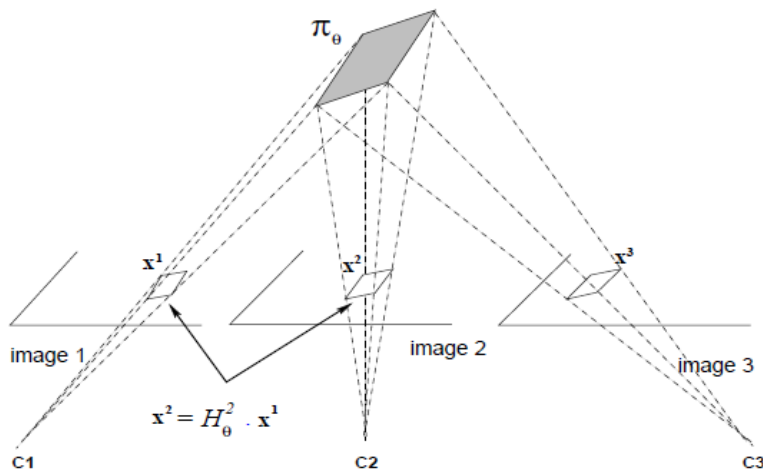
Camera arrays

- The Matrix (bullet time)

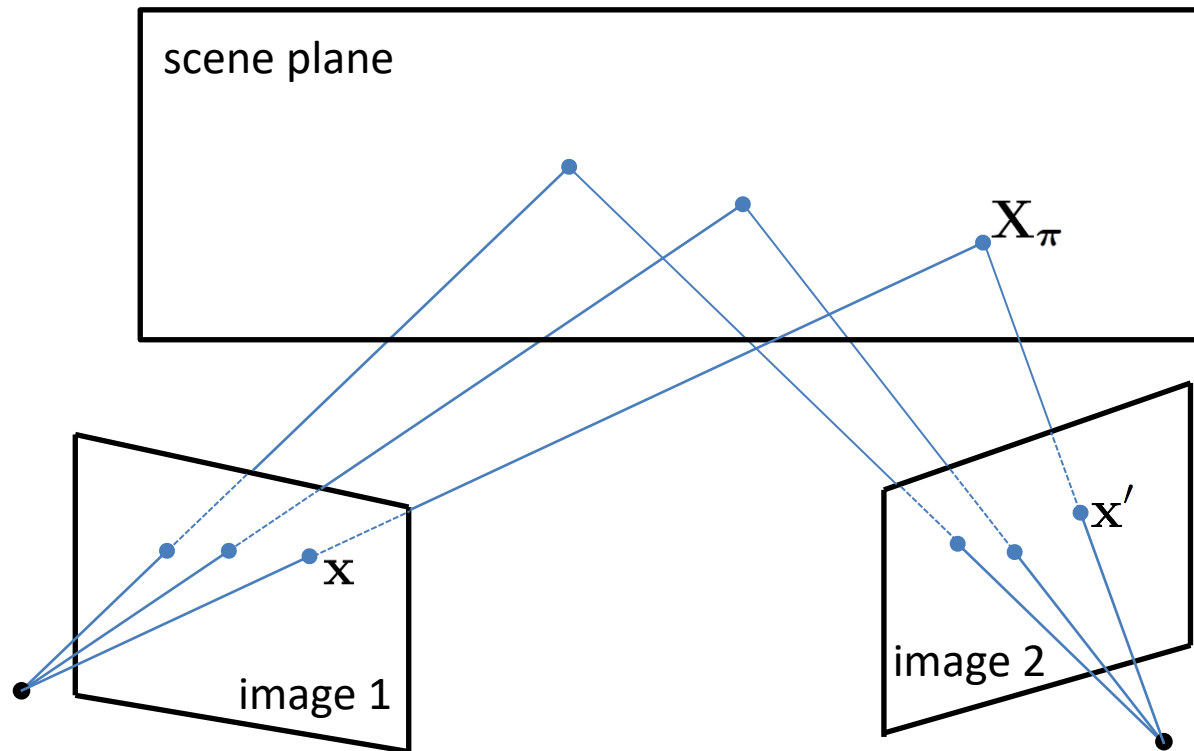


Camera arrays

- Many images, mapped to a virtual plane
 - Plane sweep methods



Imaging a plane



Imaging a plane

- Calibrated cameras

The cameras $\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$ and $\mathbf{P}' = \mathbf{K}'[\mathbf{R}' \mid \mathbf{t}']$ image the points $\mathbf{X}_{\pi,i}$ on the plane $\boldsymbol{\pi} = (\mathbf{n}^\top, d)^\top$ as

$$\mathbf{x}_i = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{X}_{\pi,i} \forall i \text{ and } \mathbf{x}'_i = \mathbf{K}'[\mathbf{R}' \mid \mathbf{t}']\mathbf{X}_{\pi,i} \forall i$$

2D projective transformation

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i \forall i, \text{ where } \mathbf{H} = \mathbf{K}'\mathbf{R}' \left(\mathbf{I} - (\mathbf{R}'^\top \mathbf{t}' - \mathbf{R}^\top \mathbf{t}) \left(\frac{\mathbf{n}}{d - \mathbf{t}^\top \mathbf{R}\mathbf{n}} \right)^\top \right) \mathbf{R}^\top \mathbf{K}^{-1}$$

Imaging a plane

- Uncalibrated cameras

The points $\mathbf{X}_{\pi,i}$ on the plane π may be written as

$$\mathbf{X}_{\pi,i} = \mathbf{M}\mathbf{x}_{\pi,i} \forall i, \text{ where } \mathbf{M} \text{ is the null space of } \pi^\top$$

Projection

$$\mathbf{x}_i = \mathbf{P}\mathbf{X}_{\pi,i} \forall i$$

$$\mathbf{x}_i = \mathbf{P}\mathbf{M}\mathbf{x}_{\pi,i} \forall i$$

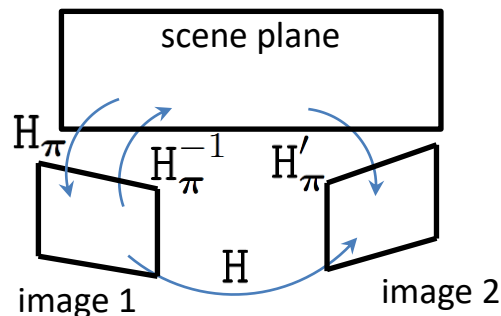
$$\mathbf{x}_i = \mathbf{H}_\pi \mathbf{x}_{\pi,i} \forall i, \text{ where } \mathbf{H}_\pi = \mathbf{P}\mathbf{M}$$

$$\mathbf{x}'_i = \mathbf{P}'\mathbf{X}_{\pi,i} \forall i$$

$$\mathbf{x}'_i = \mathbf{P}'\mathbf{M}\mathbf{x}_{\pi,i} \forall i$$

$$\mathbf{x}'_i = \mathbf{H}'_\pi \mathbf{x}_{\pi,i} \forall i, \text{ where } \mathbf{H}'_\pi = \mathbf{P}'\mathbf{M}$$

2D projective transformation



$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i \forall i, \text{ where } \mathbf{H} = \mathbf{H}'_\pi \mathbf{H}_\pi^{-1}$$

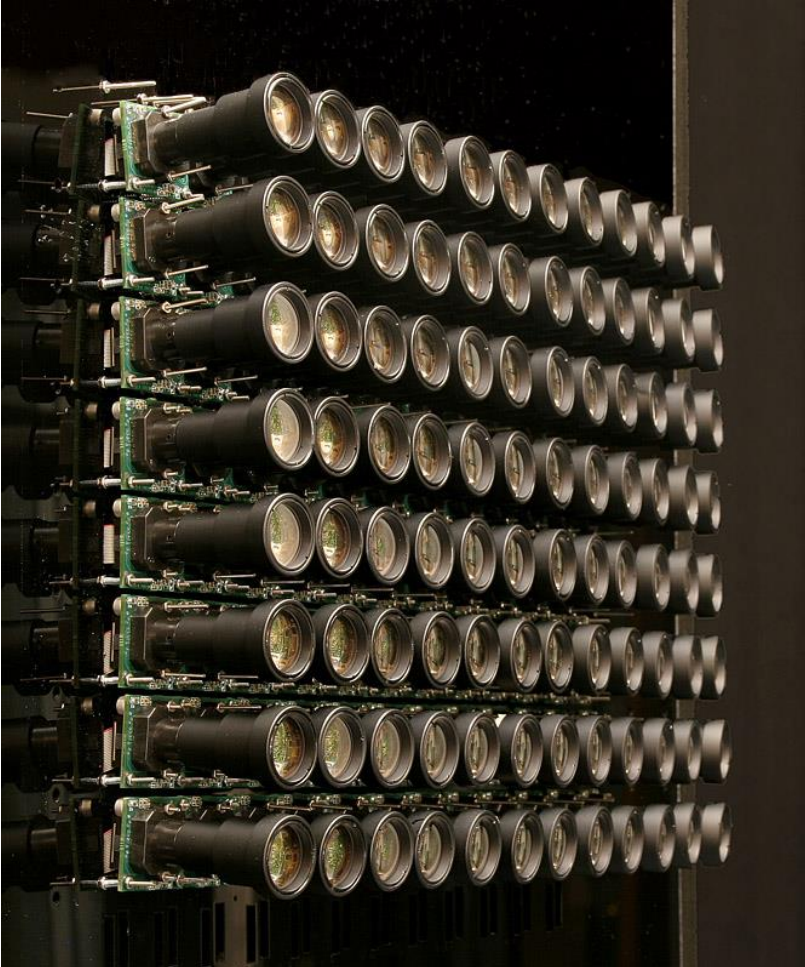
$$\mathbf{H} = \mathbf{P}'\mathbf{M}(\mathbf{P}\mathbf{M})^{-1}$$

Camera arrays

- Many images, mapped to a virtual plane
 - Plane sweep methods

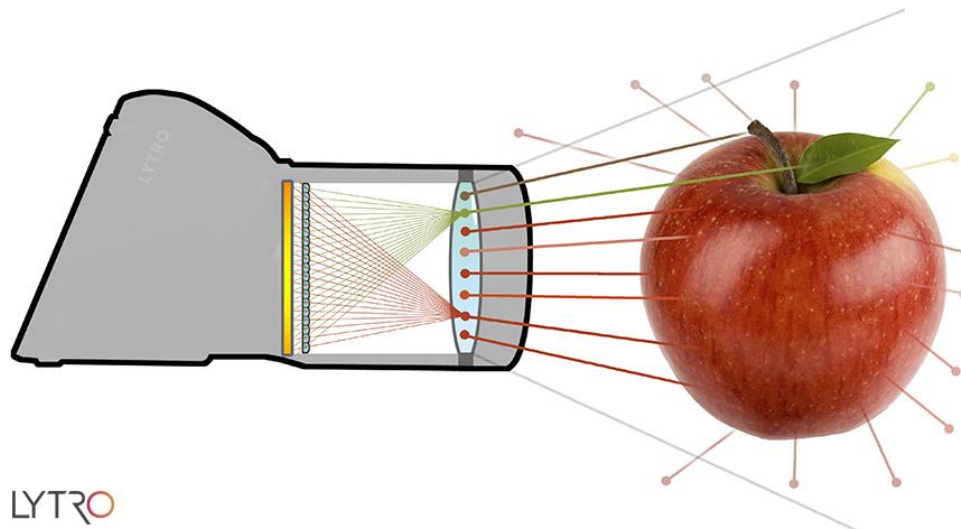


Camera arrays



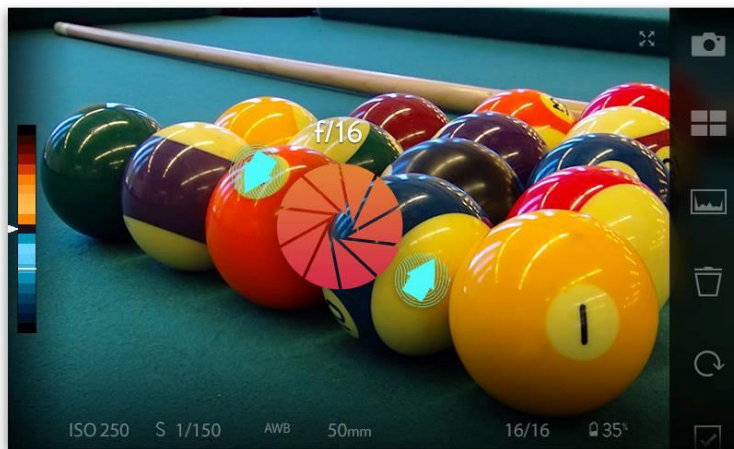
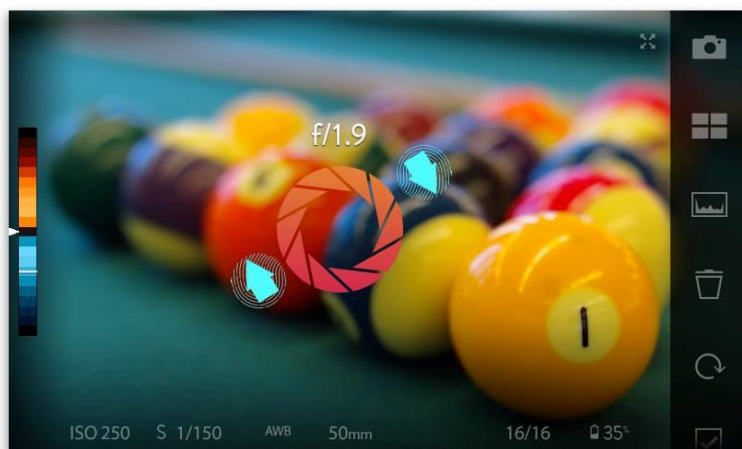
Application: synthetic focus

Light field photography



LYTRO

The Microlens Array in the Lytro ILLUM gathers bundles of light rays and distributes those bundles to specific pixels on the sensor to capture the rays' angularity, color and intensity. Proportions not to scale.



Application:
refocus

Plenoptic function

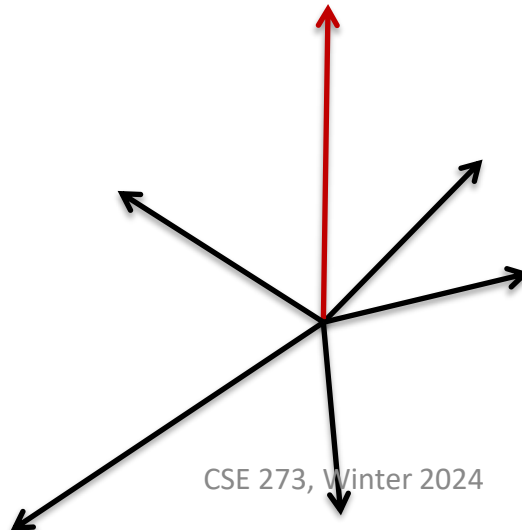
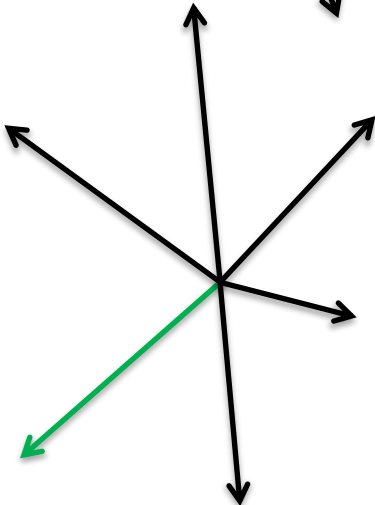
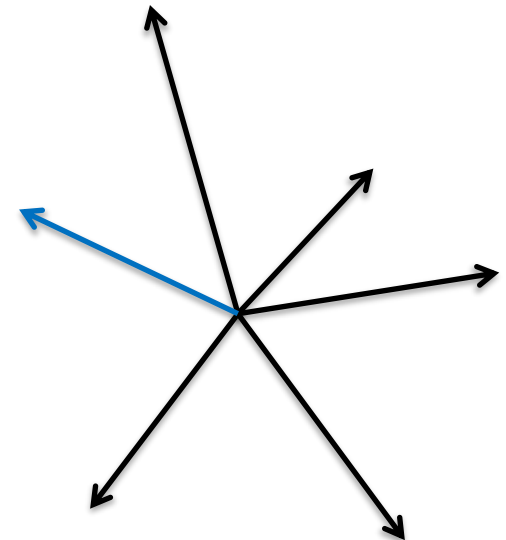
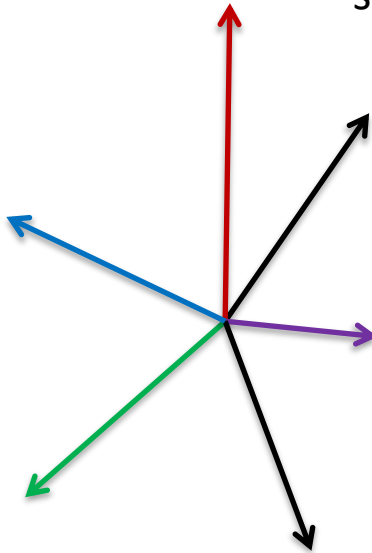
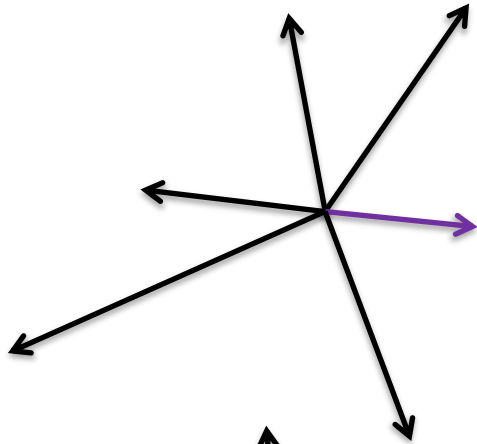
- Recall that
 - Light is carried on a ray
 - The amount of light traveling along a ray is radiance
- Plenoptic function: the radiant energy along a ray passing through a point in space at a given time
 - The radiance at all 3D points in all ray directions at a given time

Light field

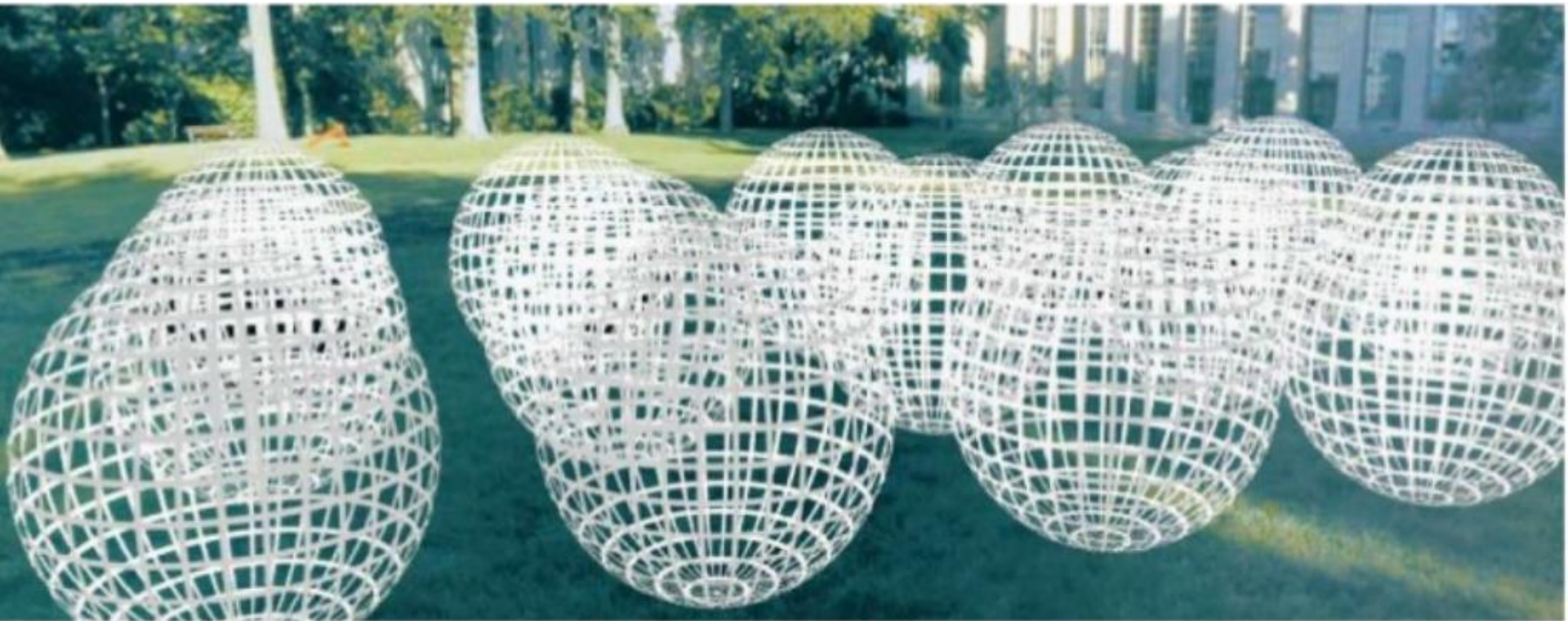
- Light field: a snapshot of the plenoptic function for light traveling in a vacuum in the absence of obstacles
 - Independent of time
 - Independent of the position of the point along a ray
 - Radiance is constant along straight lines in a vacuum

2D light field

same color = same ray = same radiance
(except black rays)



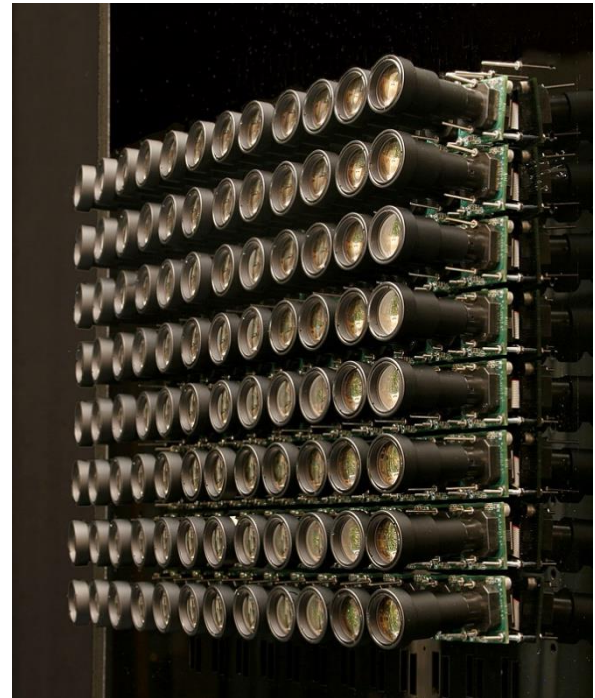
3D light field



Spherical mosaics

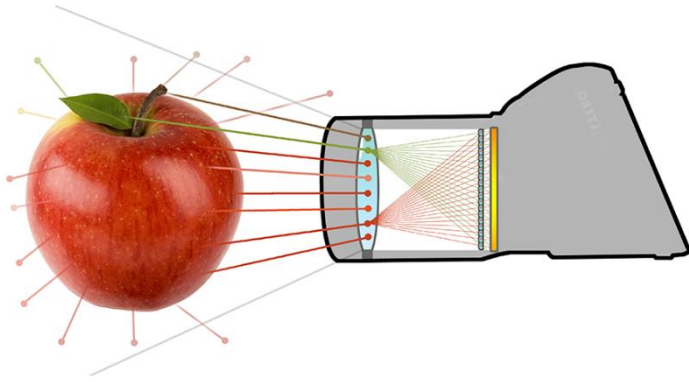
Capturing light fields

- A set of spherical mosaics
 - Each at a different 3D point
- An array of cameras



Capturing and processing light fields

- A dense array of micro-lenses



Rectified, raw light field



Refocused



Depth map



Summary

- Camera arrays
 - Single moving camera
 - Multiple cameras
 - Dense array of micro-lenses
- Applications
 - Scene reconstruction
 - View synthesis
 - Matchmoving
 - Synthetic (re)focus

Next lecture

- Computational illumination