COMPUTER VISION

LECTURE 7 25.9.2019

Laura Ruotsalainen, Associate Professor
Department of Computer Science

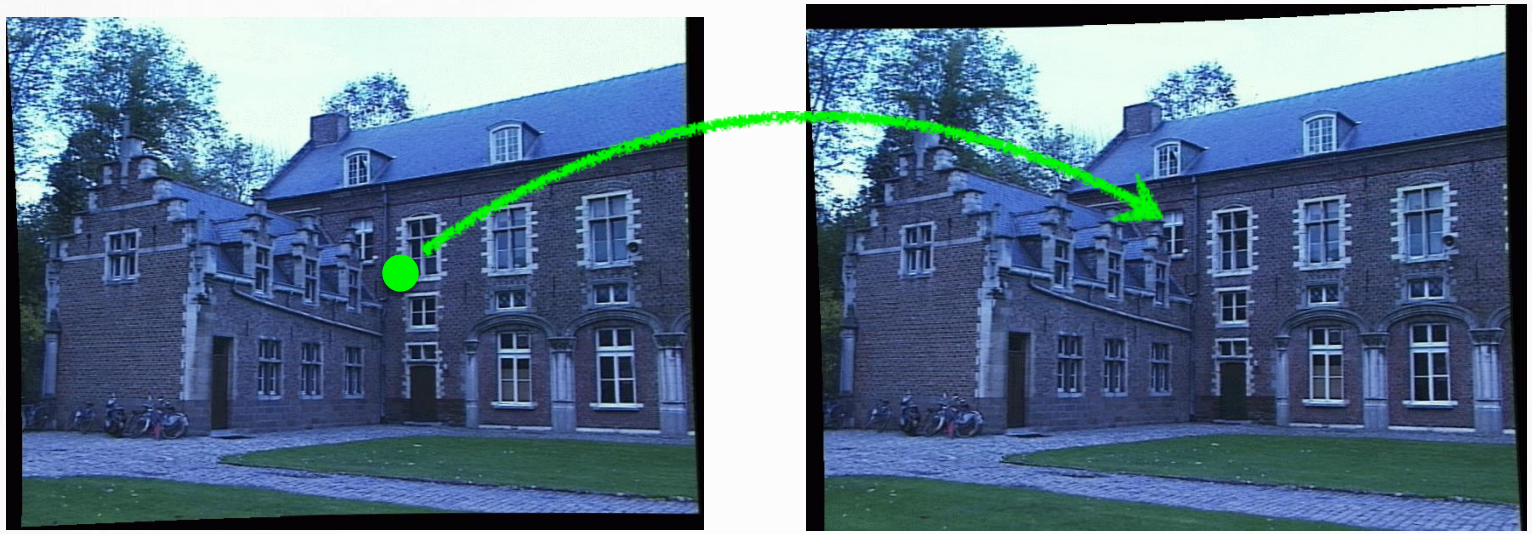
TODAY'S LECTURE

- Two-frame structure from motion (SFM) => estimatin simultaneously the 3D scene structure and camera pose from matched image features
  - Essential Matrix
  - Fundamental matrix
  - Solving ambiguous scale
  - Stereo vision

- Szeliski 7,
- Hartley, Zissermann

EPIPOLAR CONSTRAINT

Task: Match point in left image to point in right image



Left image

Right image

Epipolar constraint reduces search to a single line

The epipolar constraint is an important concept for stereo vision

How do you compute the epipolar line?

Slide credit:
Kris Kitani

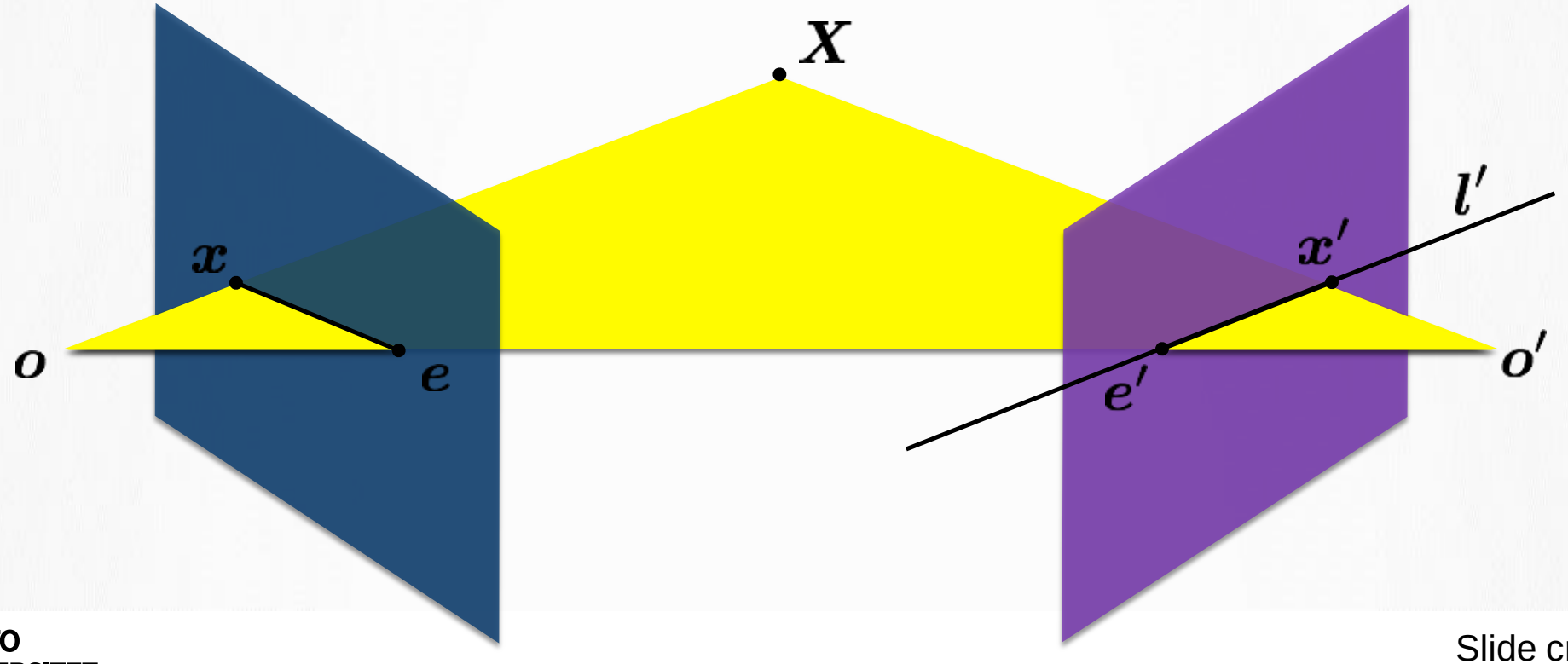
FUNDAMENTAL MATRIX SONG

https://www.youtube.com/watch?v=DgGV3l82NTk

THE ESSENTIAL MATRIX

Given a point in one image, multiplying by the **essential matrix** will tell us the **epipolar line** in the second view.

$$\mathbf{E}x = l'$$

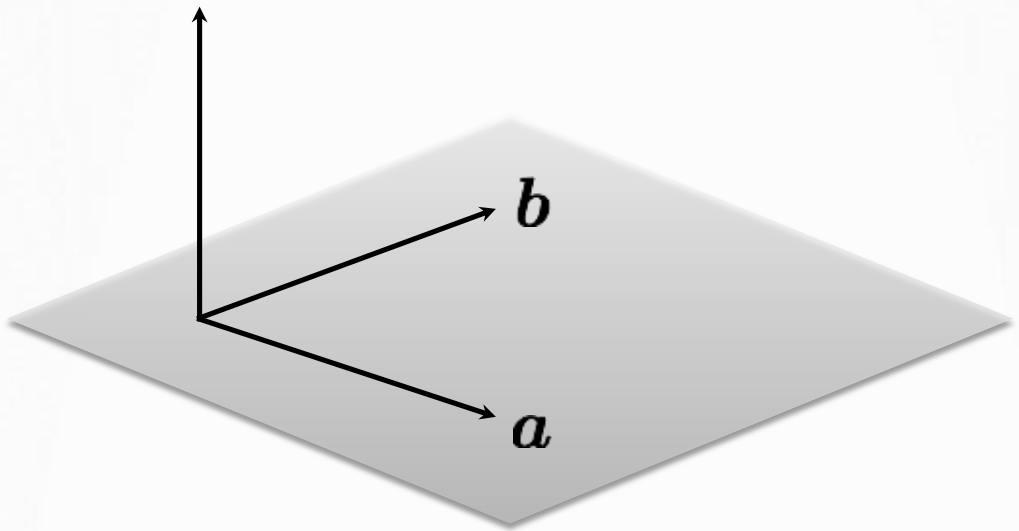


Slide credit:
Kris Kitani

RECALL: CROSS PRODUCT

Vector (cross) product
takes two vectors and returns a vector perpendicular to both

$$\boldsymbol{c} = \boldsymbol{a} \times \boldsymbol{b}$$



$$\boldsymbol{c} \cdot \boldsymbol{a} = 0 \qquad \boldsymbol{c} \cdot \boldsymbol{b} = 0$$

dot product of two orthogonal vectors is zero

Cross product

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

Can also be written as a matrix multiplication

$$\boldsymbol{a} \times \boldsymbol{b} = [\boldsymbol{a}]_\times \boldsymbol{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

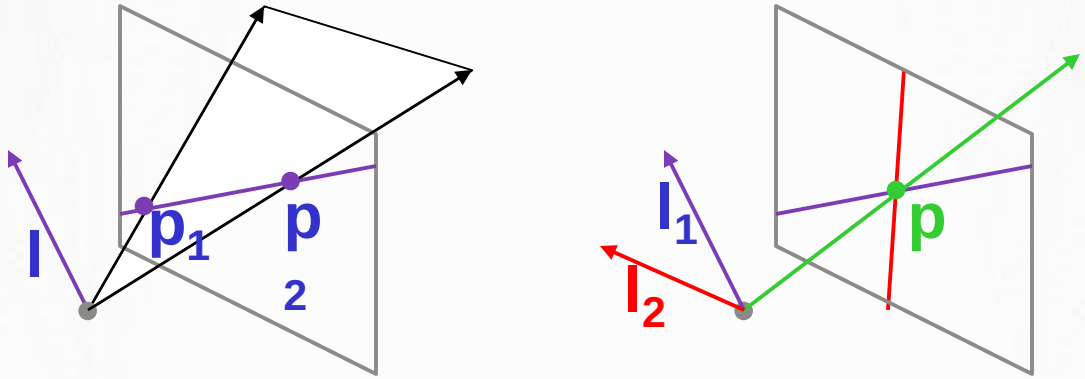
Skew symmetric

Slide credit:
Kris Kitani

POINT AND LINE DUALITY (LECTURE 2)

A line **l** is a homogeneous 3-vector

It is ⊥ to every point (ray) **p** on the line: **l**·**p**=0



What is the line **l** spanned by rays $\mathbf{p}_1$ and $\mathbf{p}_2$?

- **l** is ⊥ to $\mathbf{p}_1$ and $\mathbf{p}_2$ ⇒ $\mathbf{l} = \mathbf{p}_1 \times \mathbf{p}_2$
- **l** can be interpreted as a *plane normal*

What is the intersection of two lines $\mathbf{l}_1$ and $\mathbf{l}_2$?

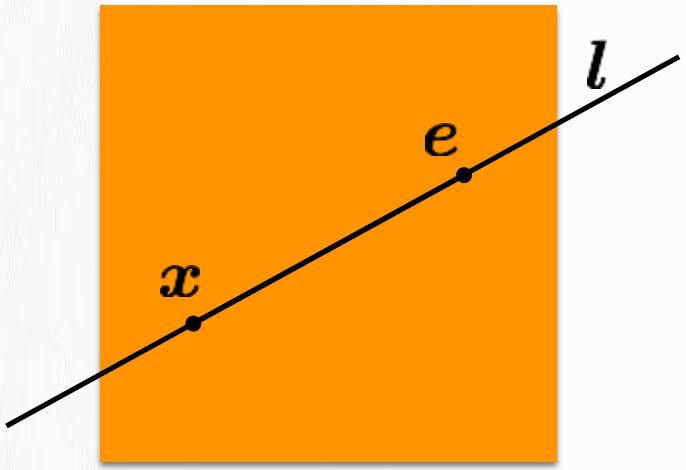
- **p** is ⊥ to $\mathbf{l}_1$ and $\mathbf{l}_2$ ⇒ $\mathbf{p} = \mathbf{l}_1 \times \mathbf{l}_2$

Points and lines are *dual* in projective space

Slide credit:
Noah Snavely

EPIPOLAR LINE

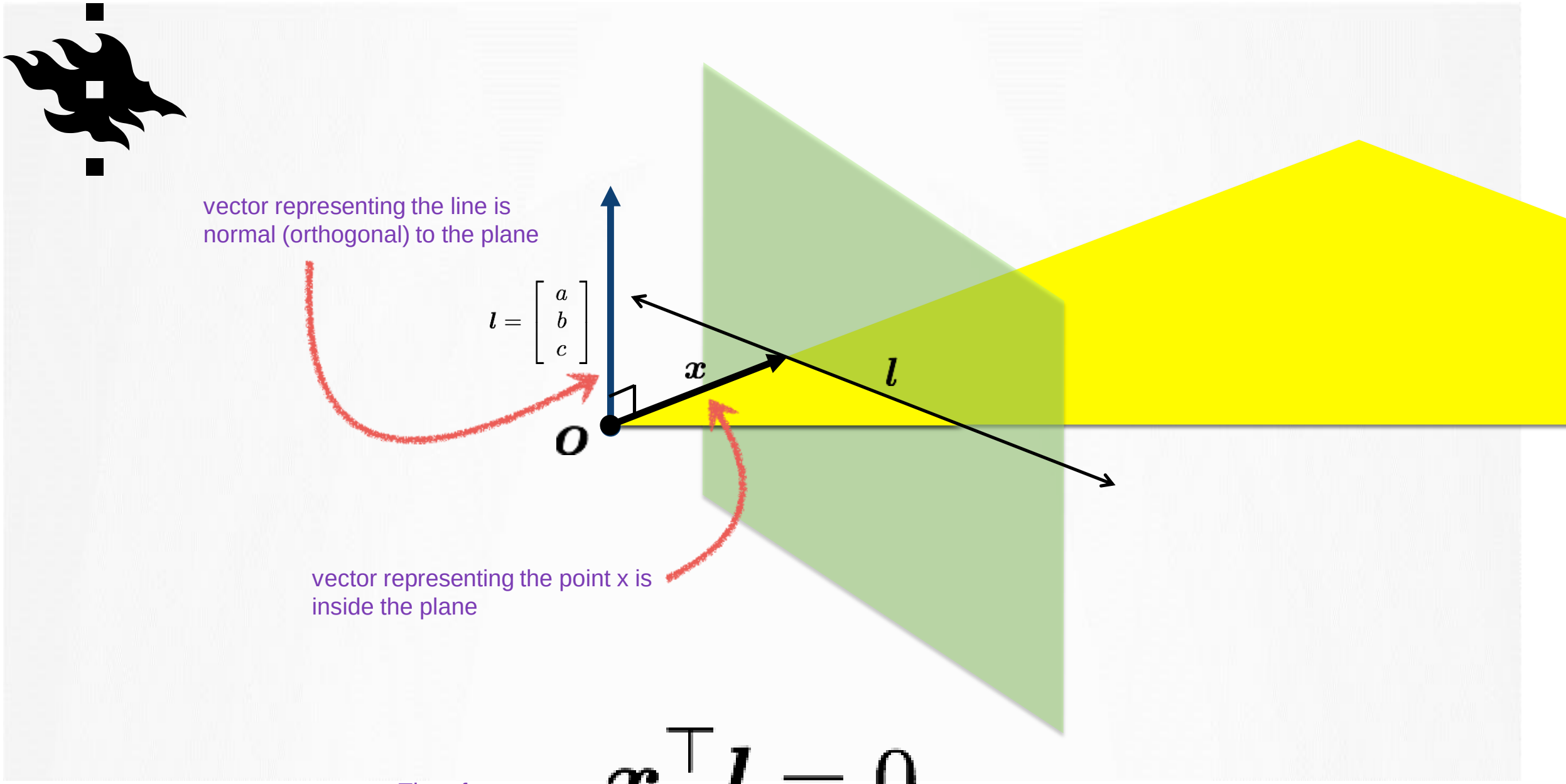
$$ax + by + c = 0$$ in vector form $$\boldsymbol{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



If the point $\boldsymbol{x}$ is on the epipolar line $\boldsymbol{l}$ then

$$\boldsymbol{x}^\top \boldsymbol{l} = 0$$

Slide credit:
Kris Kitani

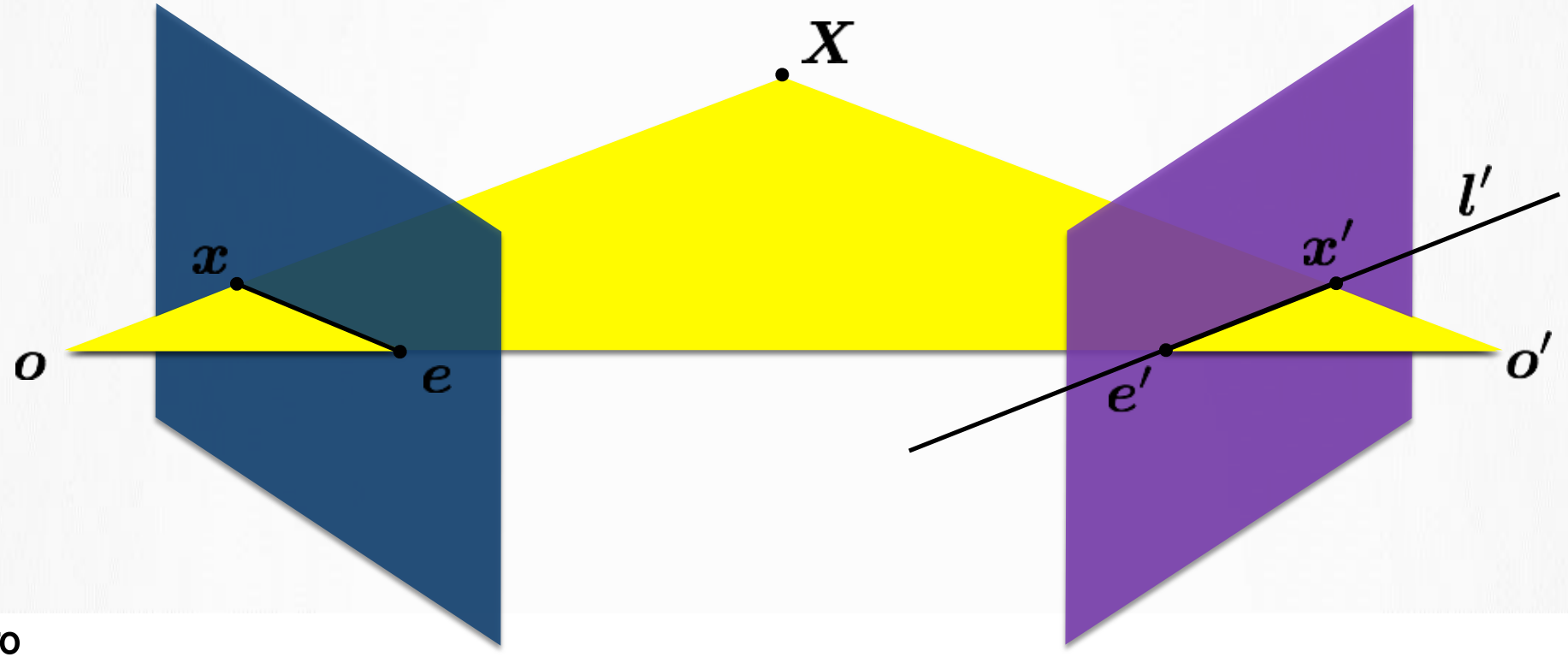


Therefore: $\boldsymbol{x}^\top \boldsymbol{l} = 0$

Slide credit:
Kris Kitani

So if $\boldsymbol{x}^\top \boldsymbol{l} = 0$ and $\mathbf{E}\boldsymbol{x} = \boldsymbol{l}'$ then

$$\boxed{\boldsymbol{x}'^\top \mathbf{E}\boldsymbol{x} = 0}$$



PUTTING IT TOGETHER

rigid motion

$$x' = \mathbf{R}(x - t)$$

coplanarity

$$(x - t)^\top (t \times x) = 0$$

$$(x'^\top \mathbf{R})(t \times x) = 0$$

$$(x'^\top \mathbf{R})([\mathbf{t}_\times] x) = 0$$

$$x'^\top (\mathbf{R}[\mathbf{t}_\times]) x = 0$$

$$\boxed{x'^\top \mathbf{E} x = 0}$$

Essential Matrix
[Longuet-Higgins 1981]

Slide credit:
Kris Kitani

PROPERTIES OF THE E MATRIX

Longuet-Higgins equation

$$\boldsymbol{x}'^{\top}\mathbf{E}\boldsymbol{x} = 0$$

(points in normalized coordinates)

Slide credit:
Kris Kitani

PROPERTIES OF THE E MATRIX

Longuet-Higgins equation

$$\boldsymbol{x}'^{\top}\mathbf{E}\boldsymbol{x} = 0$$

Epipolar lines

$$\boldsymbol{x}^{\top}\boldsymbol{l} = 0 \qquad \boldsymbol{x}'^{\top}\boldsymbol{l}' = 0$$

$$\boldsymbol{l}' = \mathbf{E}\boldsymbol{x} \qquad \boldsymbol{l} = \mathbf{E}^{T}\boldsymbol{x}'$$

(points in normalized coordinates)

Slide credit:
Kris Kitani

PROPERTIES OF THE E MATRIX

Longuet-Higgins equation $\boldsymbol{x}'^{\top}\mathbf{E}\boldsymbol{x} = 0$

Epipolar lines $\boldsymbol{x}^{\top}\boldsymbol{l} = 0$ $\quad \boldsymbol{x}'^{\top}\boldsymbol{l}' = 0$

$\boldsymbol{l}' = \mathbf{E}\boldsymbol{x}$ $\quad \boldsymbol{l} = \mathbf{E}^{T}\boldsymbol{x}'$

Epipoles $\boldsymbol{e}'^{\top}\mathbf{E} = \mathbf{0}$ $\quad \mathbf{E}\boldsymbol{e} = \mathbf{0}$

(points in normalized camera coordinates)

Slide credit:
Kris Kitani

THE FUNDAMENTAL MATRIX

- Essential matrix requires that the points are aligned to camera coordinate axis (calibrated camera)
- When the task needs to be generalized, a Fundamental Matrix is used
- We can set camera pose for image 1 as t=(0,0,0)', R identity

$$\hat{x}'^{\top} \mathbf{E} \hat{x} = 0$$

The Essential matrix operates on image points expressed in **normalized coordinates**
(points have been aligned (normalized) to camera coordinates)

$$\hat{x}' = \mathbf{K}^{-1} x' \qquad \hat{x} = \mathbf{K}^{-1} x$$

camera point — image point

Writing out the epipolar constraint in terms of image coordinates

$$x'^{\top} \mathbf{K}'^{-\top} \mathbf{E} \mathbf{K}^{-1} x = 0$$
$$x'^{\top} (\mathbf{K}'^{-\top} \mathbf{E} \mathbf{K}^{-1}) x = 0$$
$$x'^{\top} \mathbf{F} x = 0$$

maps pixels to epipolar lines

Slide credit:
Kris Kitani

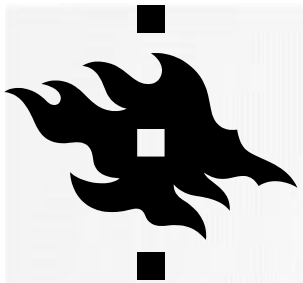
Breaking down the fundamental matrix

$$\mathbf{F} = \mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}$$

$$\mathbf{F} = \mathbf{K}'^{-\top}[\mathbf{t}_\times]\mathbf{R}\mathbf{K}^{-1}$$

Depends on both intrinsic and extrinsic parameters

Slide credit:
Kris Kitani



Breaking down the fundamental matrix

$$\mathbf{F} = \mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}$$

$$\mathbf{F} = \mathbf{K}'^{-\top}[\mathbf{t}_\times]\mathbf{R}\mathbf{K}^{-1}$$

Depends on both intrinsic and extrinsic parameters

How would you solve for F?

$$\boldsymbol{x}_m'^{\top}\mathbf{F}\boldsymbol{x}_m = 0$$

Slide credit:
Kris Kitani

ESTIMATING F

If we don't know $\mathbf{K}_1$, $\mathbf{K}_2$, **R**, or **t**, can we estimate **F** for two images?

Yes, given enough correspondences

Slide credit:
Kris Kitani

ESTIMATING F – 8-POINT ALGORITHM

The fundamental matrix F is defined by

$$\mathbf{x'}^{\mathrm{T}}\mathbf{F}\mathbf{x} = 0$$

for any pair of matches x and x' in two images.

- Let x=$(u,v,1)^{\mathrm{T}}$ and x'=$(u',v',1)^{\mathrm{T}}$, $\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$

  each match gives a linear equation

$$uu'f_{11} + vu'f_{12} + u'f_{13} + uv'f_{21} + vv'f_{22} + v'f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

Slide credit:
Kris Kitani

8-POINT ALGORITHM

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Like with homographies, instead of solving $\mathbf{Af} = 0$, we seek $\mathbf{f}$ to minimize $\|\mathbf{Af}\|$, least eigenvector of $\mathbf{A}^T\mathbf{A}$.

Slide credit:
Kris Kitani

8-POINT ALGORITHM – PROBLEM?

F should have rank 2

To enforce that **F** is of rank 2, F is replaced by F' that minimizes subject to the rank constraint.

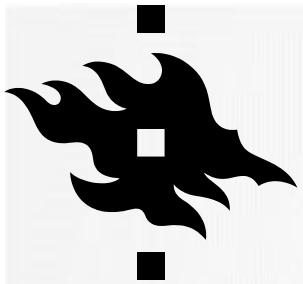
$$\|\mathbf{F} - \mathbf{F}'\|$$

- This is achieved by SVD. Let $\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, where

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \text{ let } \quad \Sigma' = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

then $\mathbf{F}' = \mathbf{U}\Sigma'\mathbf{V}^{\mathrm{T}}$ is the solution.

Slide credit:
Kris Kitani



8-POINT ALGORITHM

```
% Build the constraint matrix
A = [x2(1,:)'.*x1(1,:)'  x2(1,:)'.*x1(2,:)'  x2(1,:)' ...
     x2(2,:)'.*x1(1,:)'  x2(2,:)'.*x1(2,:)'  x2(2,:)' ...
     x1(1,:)'            x1(2,:)'            ones(npts,1) ];

[U,D,V] = svd(A);

% Extract fundamental matrix from the column of V
% corresponding to the smallest singular value.
F = reshape(V(:,9),3,3)';

% Enforce rank2 constraint
[U,D,V] = svd(F);
F = U*diag([D(1,1) D(2,2) 0])*V';
```


Slide credit:
Kris Kitani

MOTION FROM E

$$E = [\hat{t}]_x R = U\Sigma V^T$$

$$= \begin{bmatrix} u_0 & u_1 & \hat{t} \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \begin{bmatrix} v_0^T \\ v_1^T \\ v_2^T \end{bmatrix}$$

t = singular value associated with smallest singular vector

R = four possible matrices $R = \pm U R_{\pm 90^\circ}^T V^T$

1. Keep the ones with det |R|=1
2. Pair the remaining ones with both possible signs of translation direction
3. Select the combination for which the largest number of points is seen in front of both cameras

Slide credit:
Kris Kitani

PROBLEM WITH 8-POINT ALGORITHM

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

~10000 ~10000 ~100 ~10000 ~10000 ~100 ~100 ~100 1

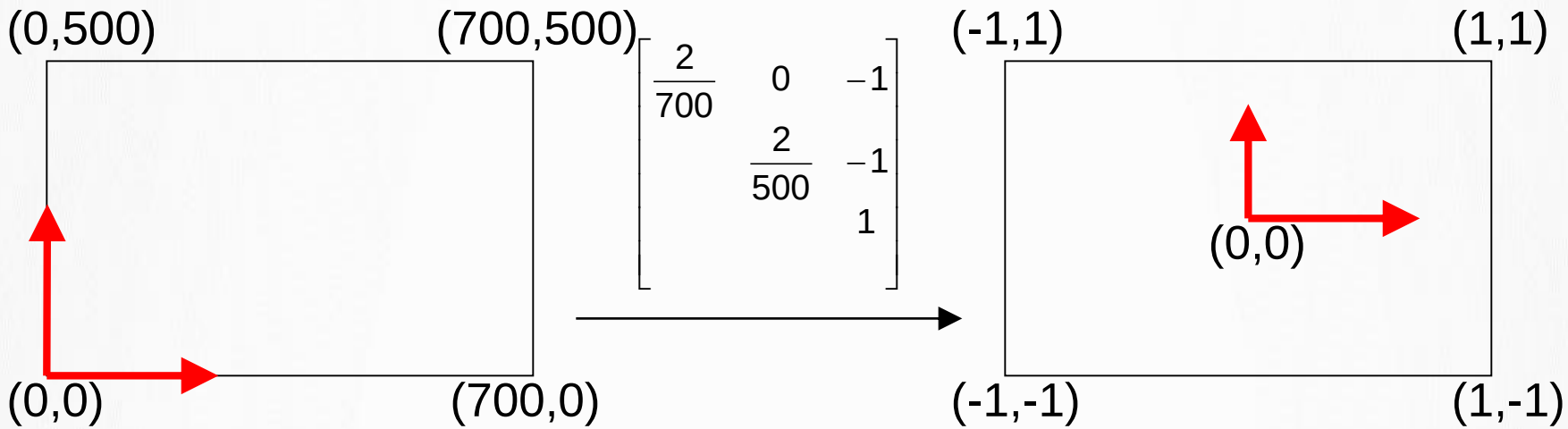
Orders of magnitude difference between column of data matrix
→ least-squares yields poor results

Slide credit:
Kris Kitani

NORMALIZED 8-POINT ALGORITHM

normalized least squares yields good results

Transform image to ~[-1,1]x[-1,1]



Slide credit:
Kris Kitani

NORMALIZED 8-POINT ALGORITHM

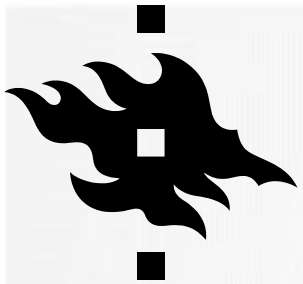
1. Transform input by $\hat{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$, $\hat{\mathbf{x}}'_i = \mathbf{T}\mathbf{x}'_i$
2. Call 8-point on $\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i$ to obtain $\hat{\mathbf{F}}$
3. $\mathbf{F} = \mathbf{T}'^{\mathrm{T}}\hat{\mathbf{F}}\mathbf{T}$

$$\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x} = 0$$

$$\hat{\mathbf{x}}'^{\mathrm{T}}\mathbf{T}'^{-\mathrm{T}}\mathbf{F}\mathbf{T}^{-1}\hat{\mathbf{x}} = 0$$

$$\hat{\mathbf{F}}$$

Slide credit:
Kris Kitani

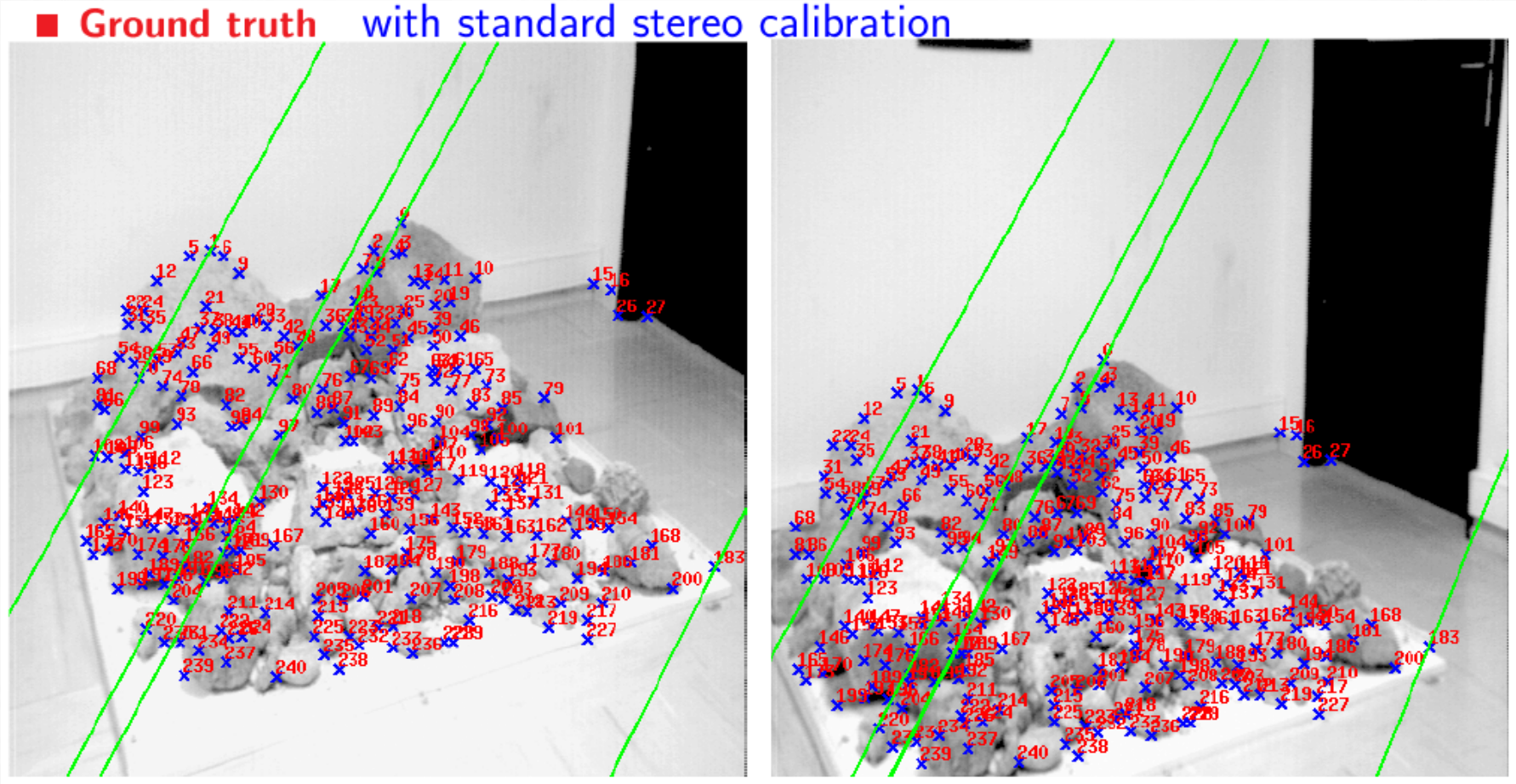


NORMALIZED 8-POINT ALGORITHM

```
[x1, T1] = normalise2dpts(x1);
[x2, T2] = normalise2dpts(x2);

A = [x2(1,:)'.*x1(1,:)'   x2(1,:)'.*x1(2,:)'   x2(1,:)' ...
     x2(2,:)'.*x1(1,:)'   x2(2,:)'.*x1(2,:)'   x2(2,:)' ...
     x1(1,:)'             x1(2,:)'             ones(npts,1) ];

[U,D,V] = svd(A);
F = reshape(V(:,9),3,3)';
[U,D,V] = svd(F);
F = U*diag([D(1,1) D(2,2) 0])*V';

% Denormalise
F = T2'*F*T1;
```

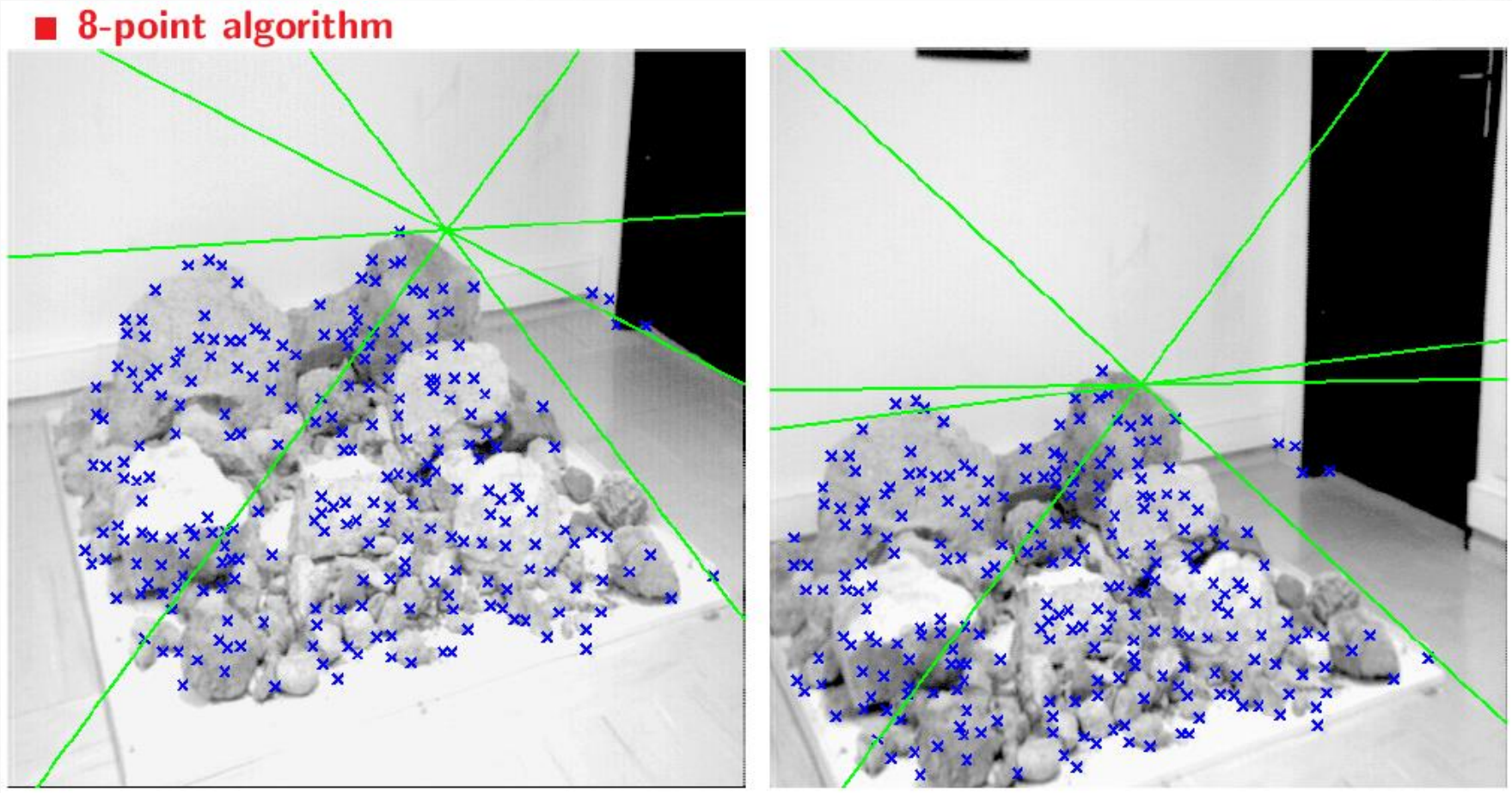

Slide credit:
Kris Kitani

RESULTS (GROUND TRUTH)



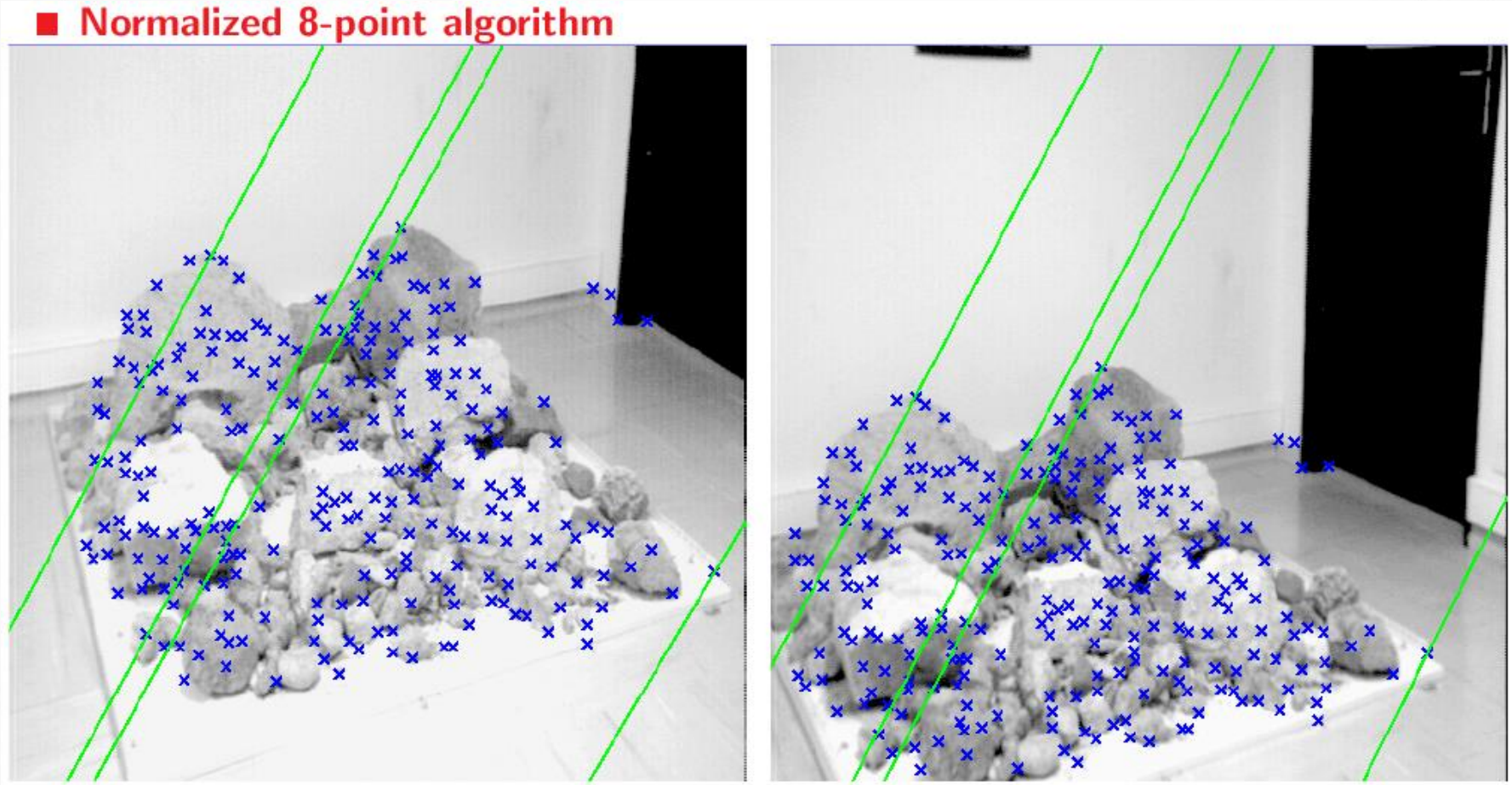
Slide credit:
Kris Kitani

RESULTS (8-POINT ALGORITHM)



Slide credit:
Kris Kitani

RESULTS (NORMALIZED 8-POINT ALGORITHM)



Slide credit:
Kris Kitani

WHAT ABOUT MORE THAN TWO VIEWS?

The geometry of three views is described by a 3 x 3 x 3 tensor called the *trifocal tensor*

The geometry of four views is described by a 3 x 3 x 3 x 3 tensor called the *quadrifocal tensor*

After this it starts to get complicated…

Slide credit:
Kris Kitani

SOLVING THE SCALE PROBLEM

Solutions for finding scale (Z) :

LiDars, RGB-D cameras => include Z

Stereo camera => Known pose of the cameras give Z by triangulation : length of baseline important

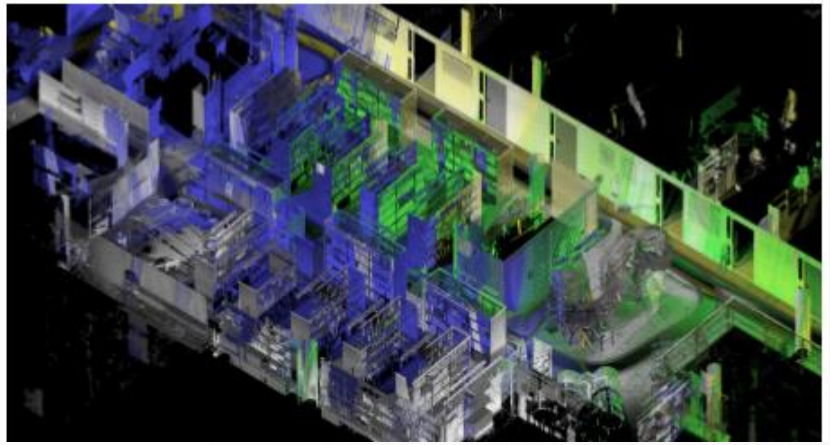
Monocular camera

- Objects with known size: needs a priori preparation and is restricted to one area
- Camera facing down with known height : problematic indoors due to uniform texture of the floor
- Estimation, this we'll look at when talking about SLAM

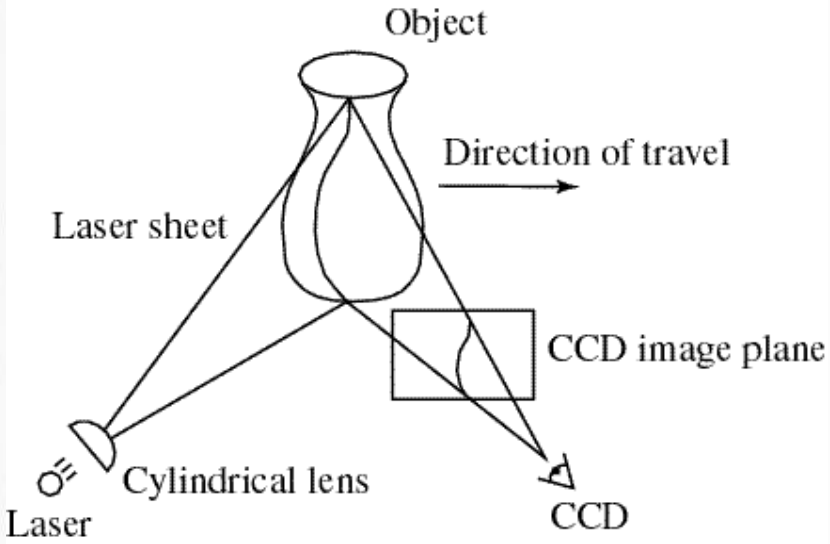
Santos et al(2009). An analysis of navigation algorithms for smartphones using J2ME. *MOBILWARE*.

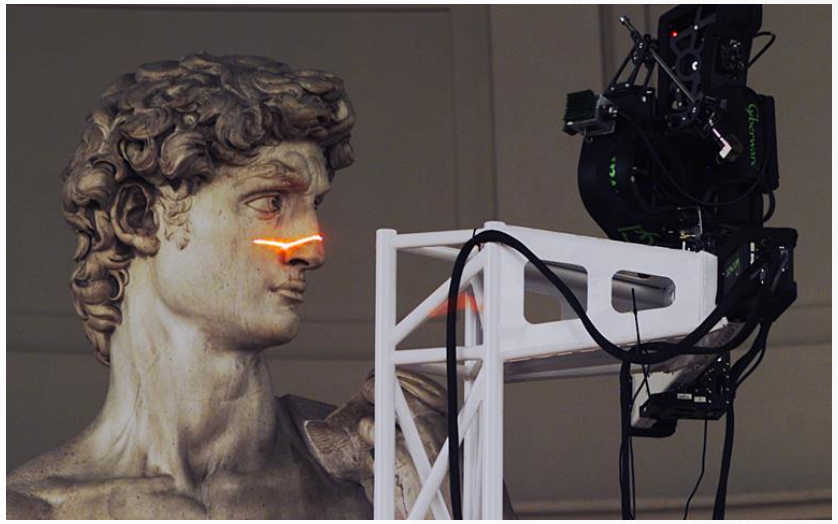
LIGHT DETECTION AND RANGING (LIDAR)

- Controlled steering of laser beams followed by a distance measurement (time it took for the signal to bounce and get back to the system at every pointing direction => 3D model of objects
- Doesn’t require light
- Traditionally expensive and large => technology evolving fast
- Materials of surroundings matter, e.g. glass lets the rays through
- Quite slow to process
- Motion estimation very accurate, cm level accuracy



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

Slide credit:
Kris Kitani

LASER SCANNED MODELS

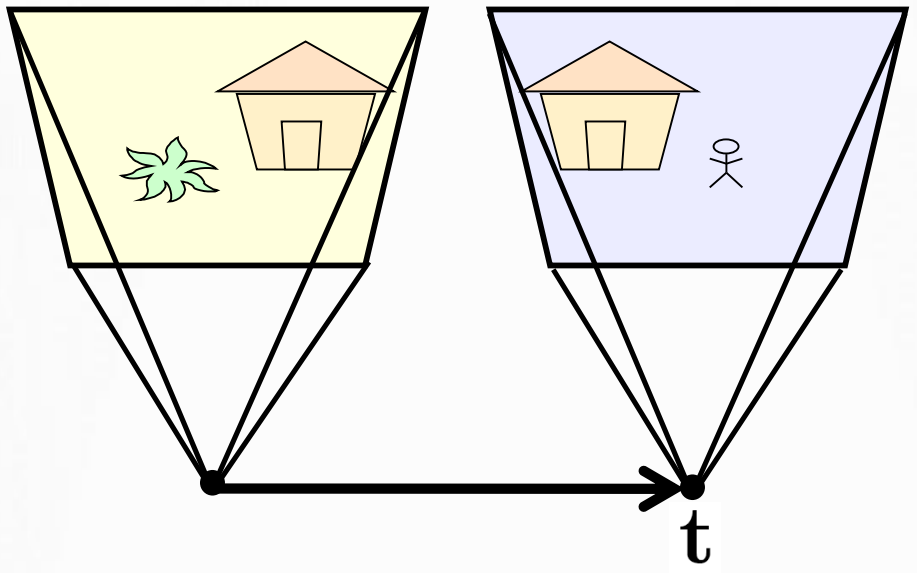
The Digital Michelangelo Project, Levoy et al.

RGB-D CAMERAS

- Combine color information with per pixel depth infromation using infrared sensors
- Limitations due to field of view, max object distance from sensor, huge amount of data
- Inexpensive, small and light cameras have entered the market

Microsoft Kinect

RECTIFIED CASE



$$\mathbf{R} = \mathbf{I}_{3\times 3}$$

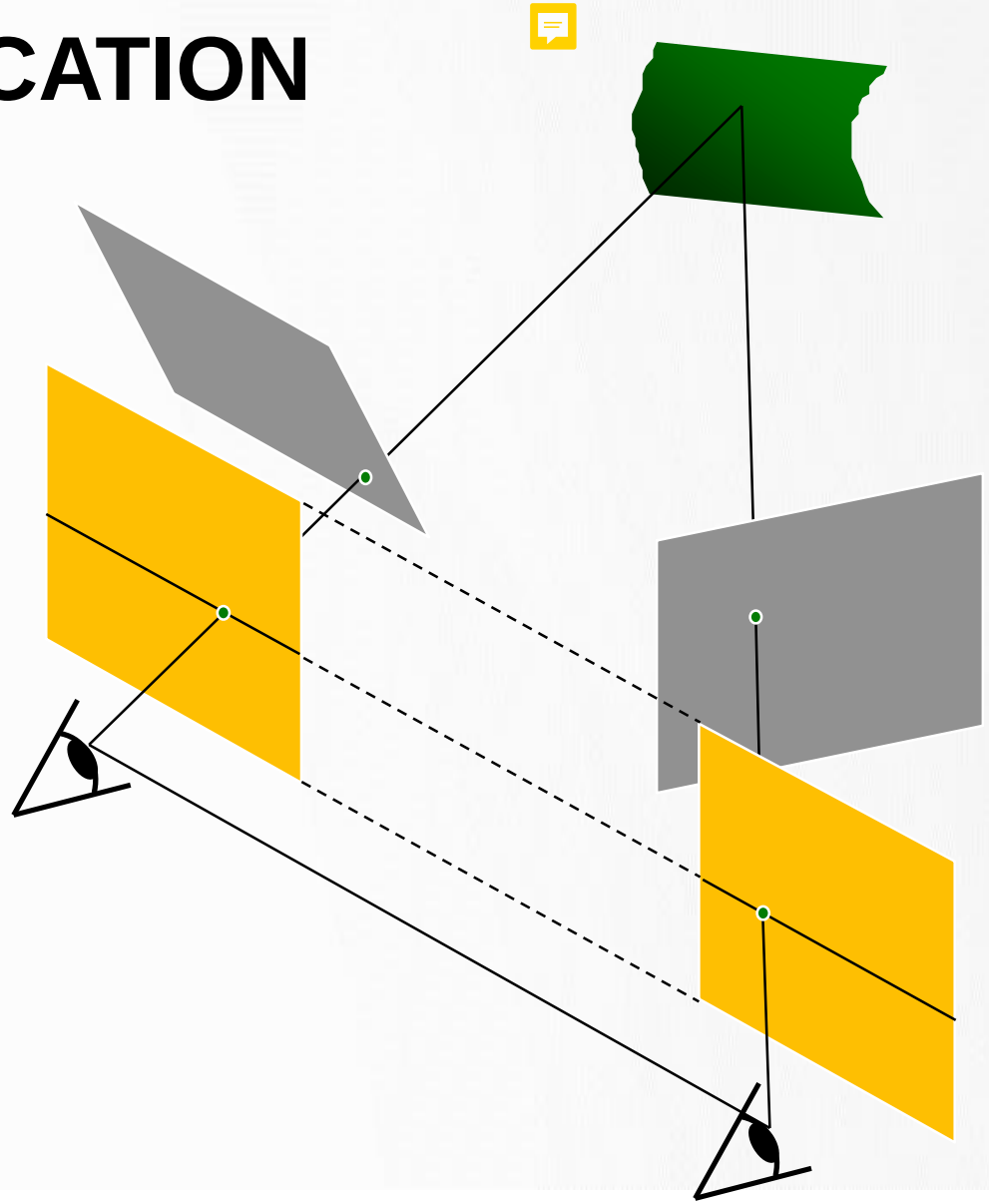
$$\mathbf{t} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

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

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.

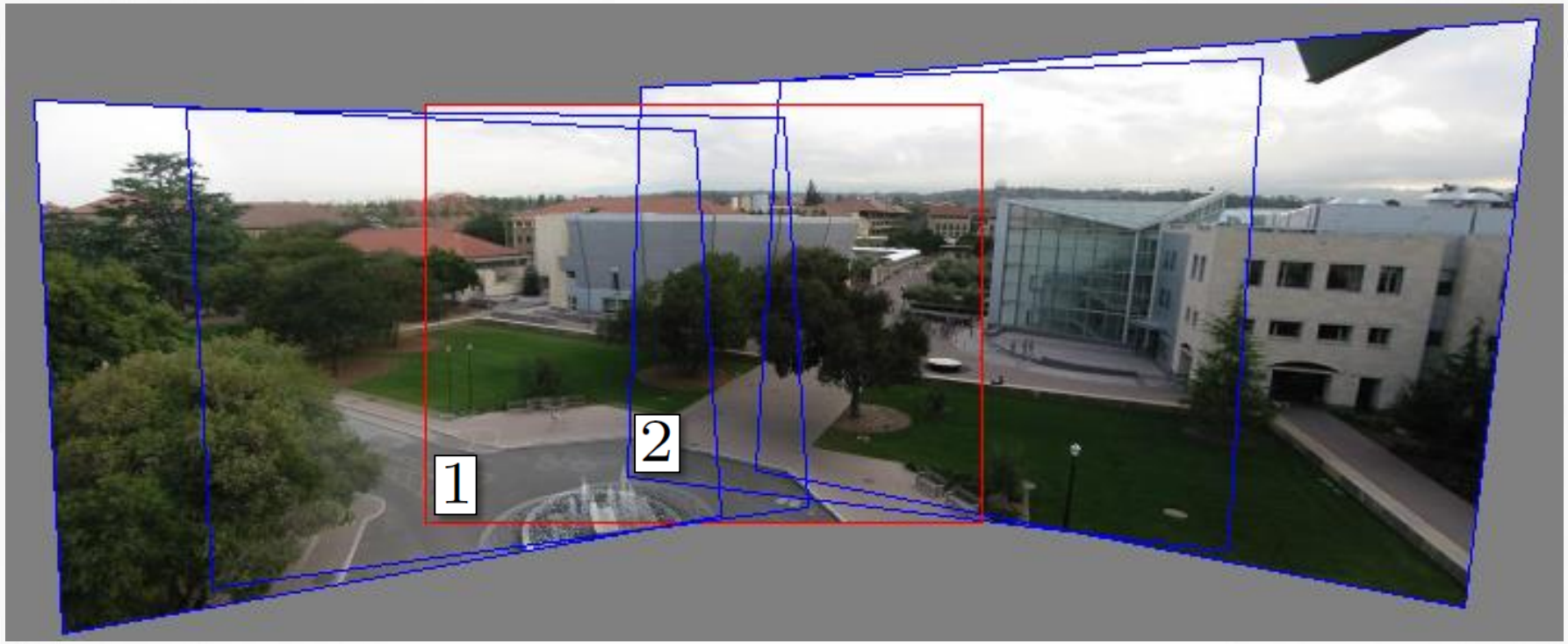


Slide credit:
Kris Kitani

Original stereo pair

After rectification

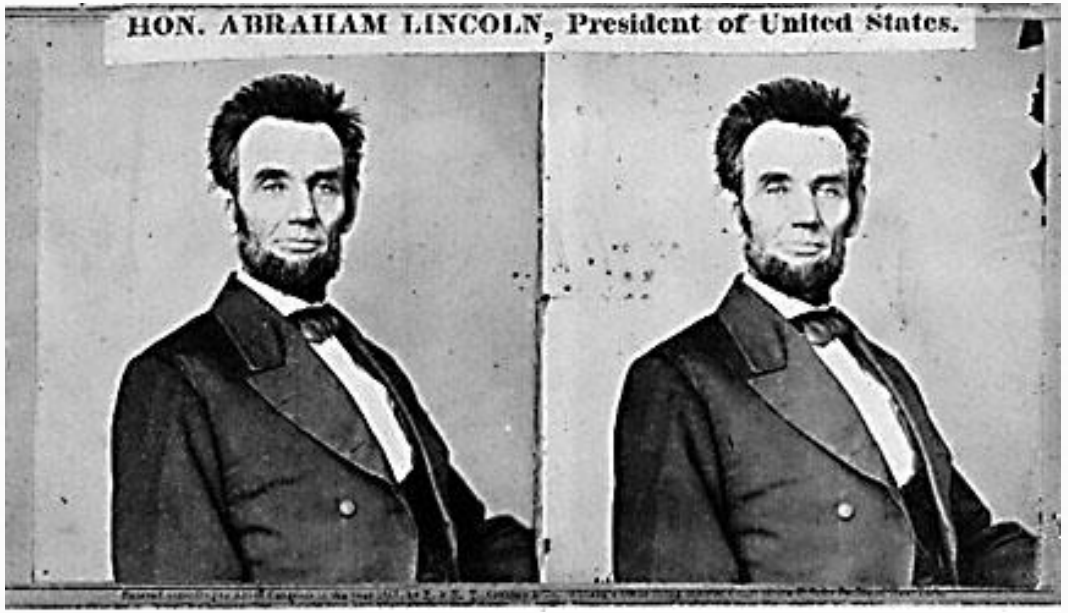
RELATIONSHIP WITH HOMOGRAPHY?



Images taken from the same center of projection? Use a homography!

Slide credit:
Kris Kitani

STEREO



Given two images from different viewpoints

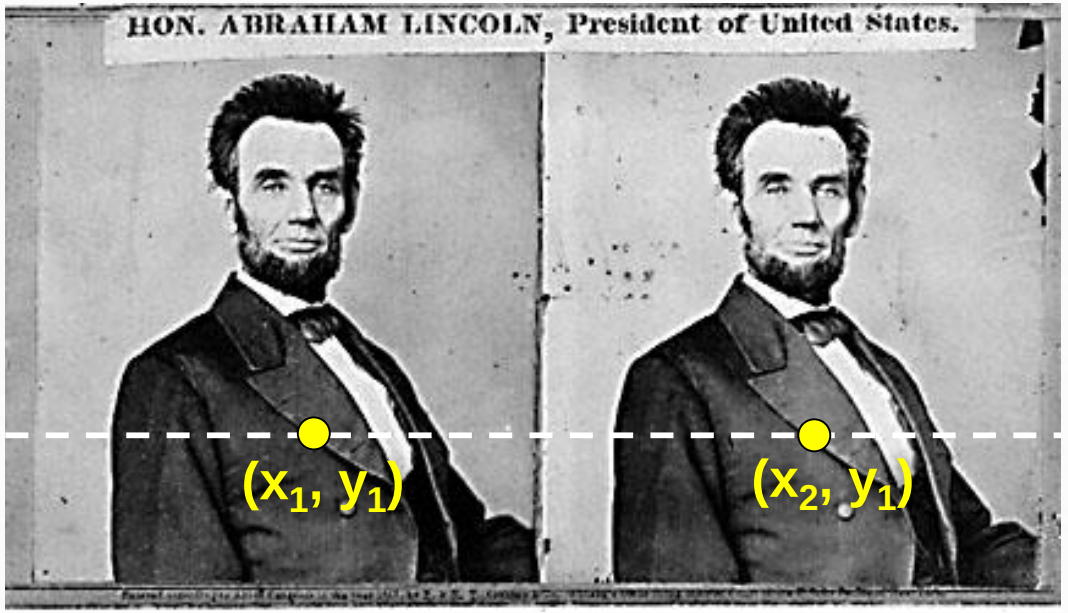
How can we compute the depth of each point in the image?

Based on *how much each pixel moves* between the two images

Slide credit:
Kris Kitani

EPIPOLAR GEOMETRY

epipolar lines

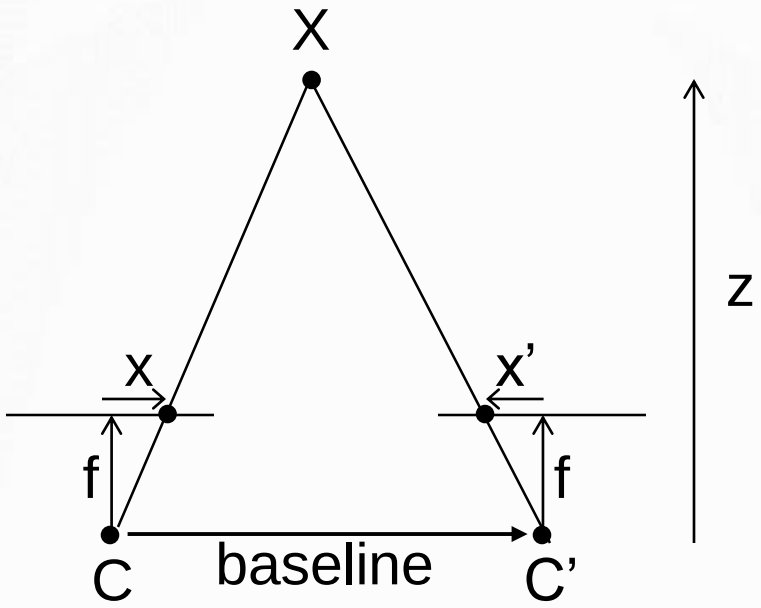


Two images captured by a purely horizontal translating camera
(*rectified* stereo pair)

$x_2 - x_1$ = the *disparity* of pixel (x_1, y_1)

Slide credit:
Kris Kitani

DEPTH FROM DISPARITY



$$disparity = x - x' = \frac{baseline * f}{z}$$

Slide credit:
Kris Kitani

BASIC STEREO MATCHING ALGORITHM

Match Pixels in Conjugate Epipolar Lines

Assume brightness constancy

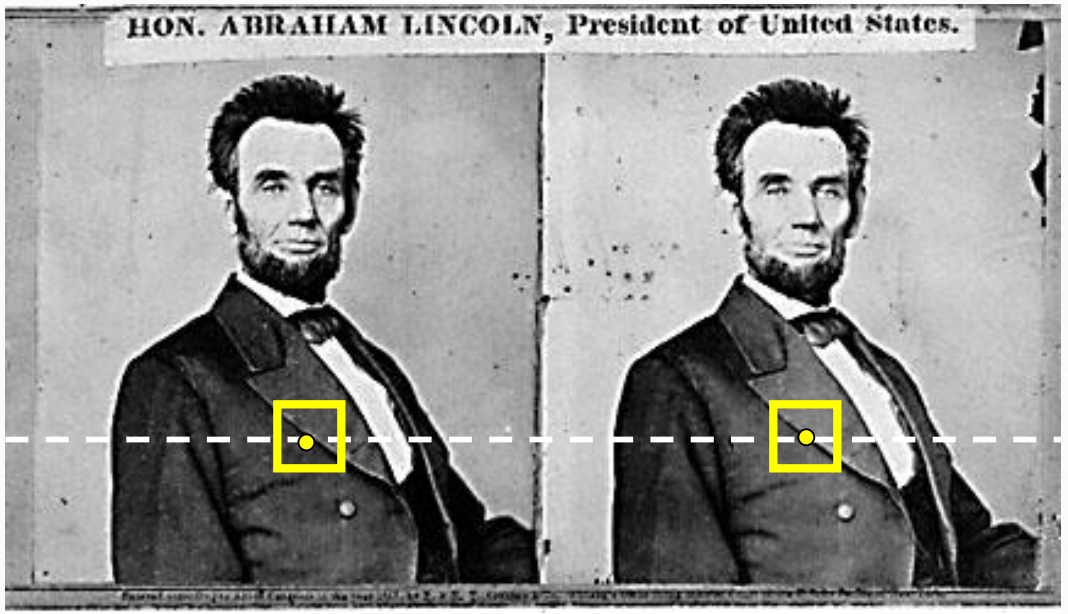
This is a challenging problem

Hundreds of approaches

– A good survey and evaluation: http://www.middlebury.edu/stereo/

Slide credit:
Kris Kitani

BASIC STEREO ALGORITHM



For each epipolar line

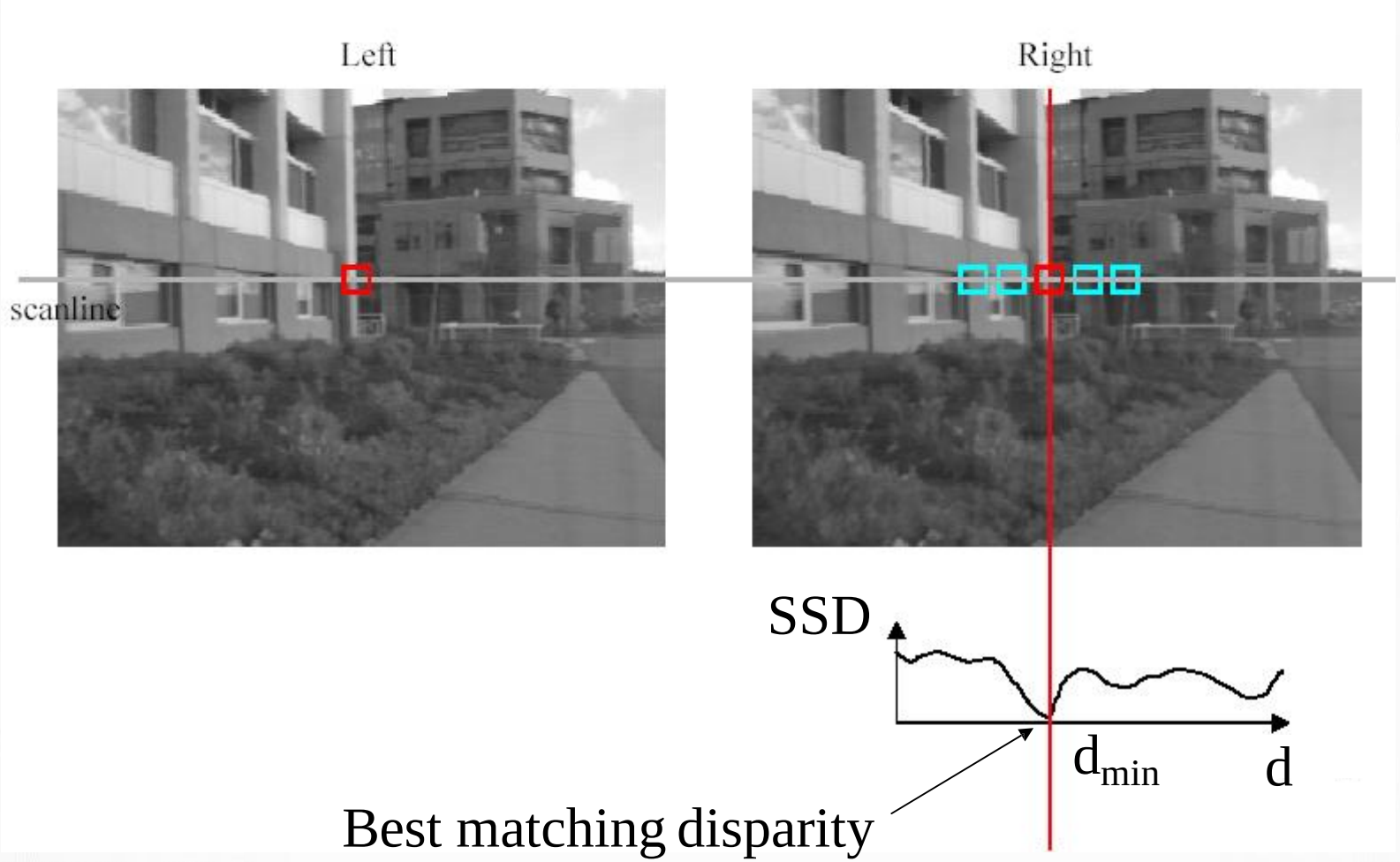
For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match ***windows***

Slide credit:
Kris Kitani

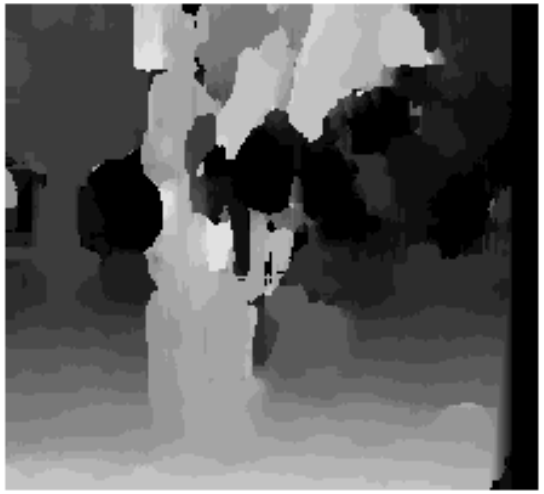
STEREO MATCHING BASED ON SSD



Slide credit:
Kris Kitani

WINDOW SIZE

W = 3



W = 20

Effect of window size

Smaller window

\+

–

Larger window

\+

Better results with *adaptive window*

- T. Kanade and M. Okutomi, *A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment*,, Proc. International Conference on Robotics and Automation, 1991.
- D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. International Journal of Computer Vision, 28(2):155-174, July 1998

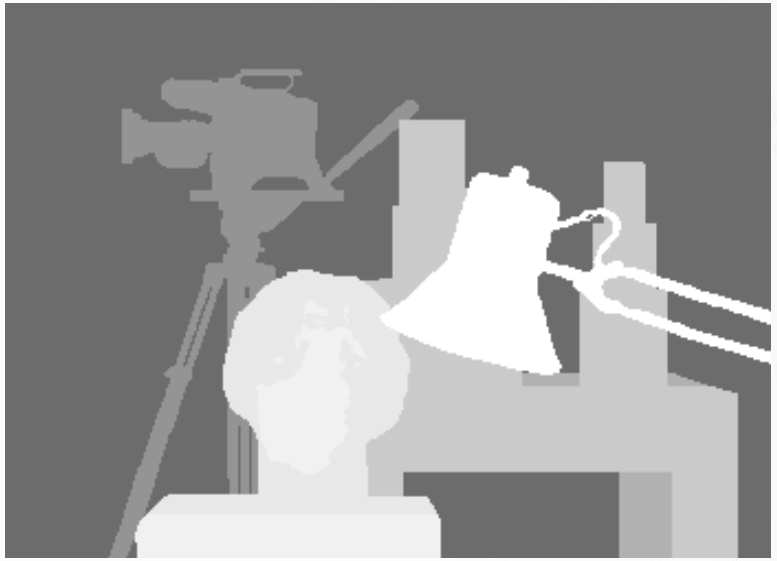
Slide credit: Kris Kitani

STEREO RESULTS

Data from University of Tsukuba

Similar results on other images without ground truth

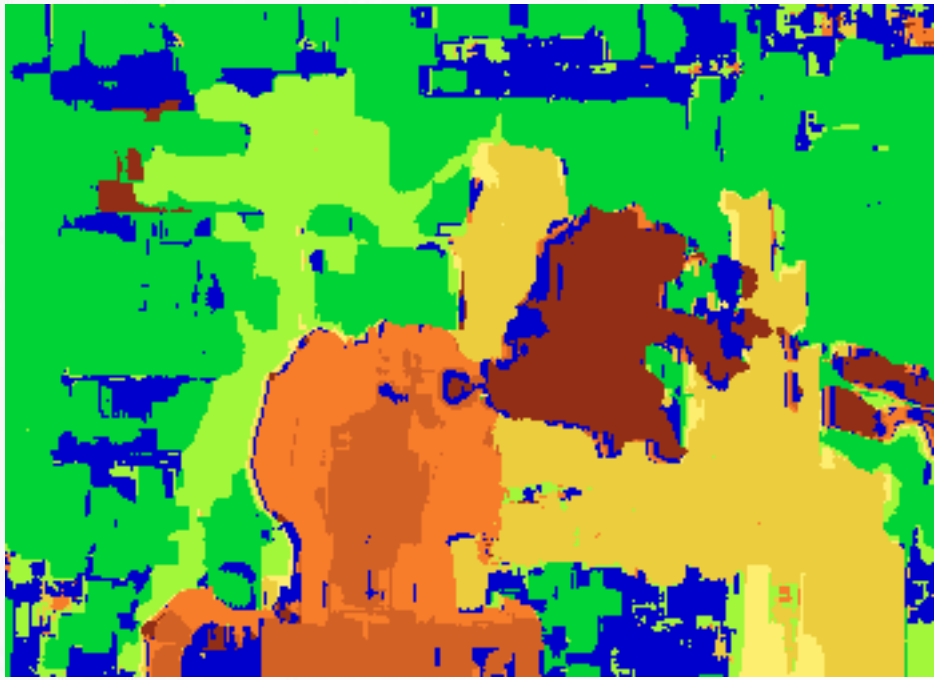
Scene



Ground truth

Slide credit:
Kris Kitani

RESULTS WITH WINDOW SEARCH



Window-based matching
(best window size)

Ground truth

Slide credit:
Kris Kitani

BETTER METHODS EXIST...

State of the art method

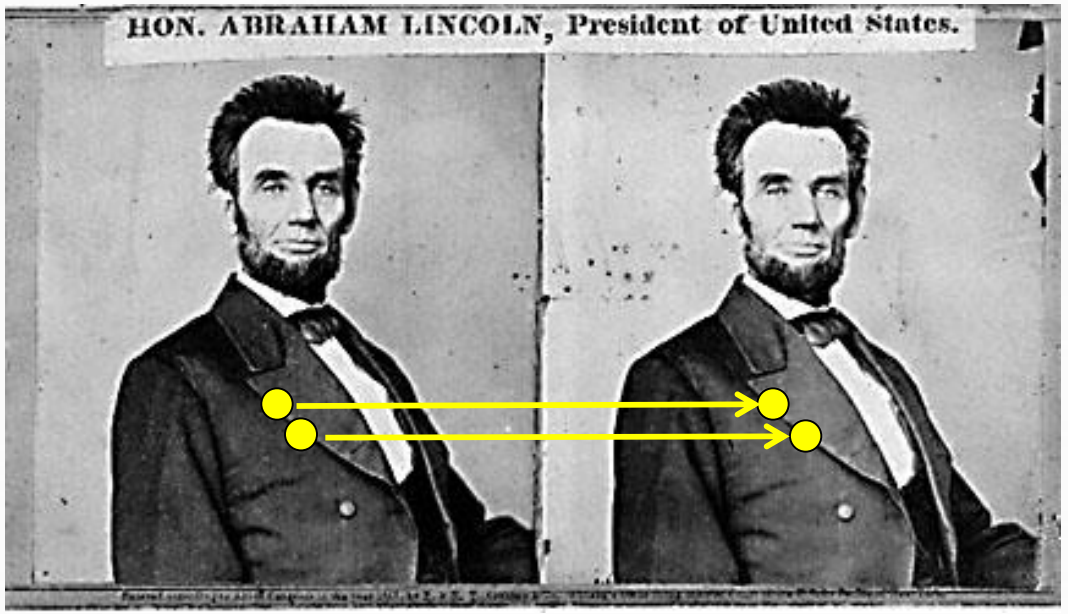
Ground truth

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

For the latest and greatest: http://www.middlebury.edu/stereo/

Slide credit: Kris Kitani

STEREO AS ENERGY MINIMIZATION



What defines a good stereo correspondence?

1. Match quality
   - Want each pixel to find a good match in the other image
2. Smoothness
   - If two pixels are adjacent, they should (usually) move about the same amount

Slide credit:
Kris Kitani

STEREO AS ENERGY MINIMIZATION

- Find disparity map *d* that minimizes an energy function

$$E(d)$$

- Simple pixel / window matching

$$E(d) = \sum_{(x,y)\in I} C(x, y, d(x, y))$$

x’ = x+d(x,y)
y’ = y

$C(x, y, d(x, y))$ = SSD distance between windows *I*(*x*, *y*) and *J*(*x* + *d*(*x*,*y*), *y*)

Slide credit:
Kris Kitani

STEREO AS ENERGY MINIMIZATION

$C(x, y, d)$; the *disparity space image* (DSI)

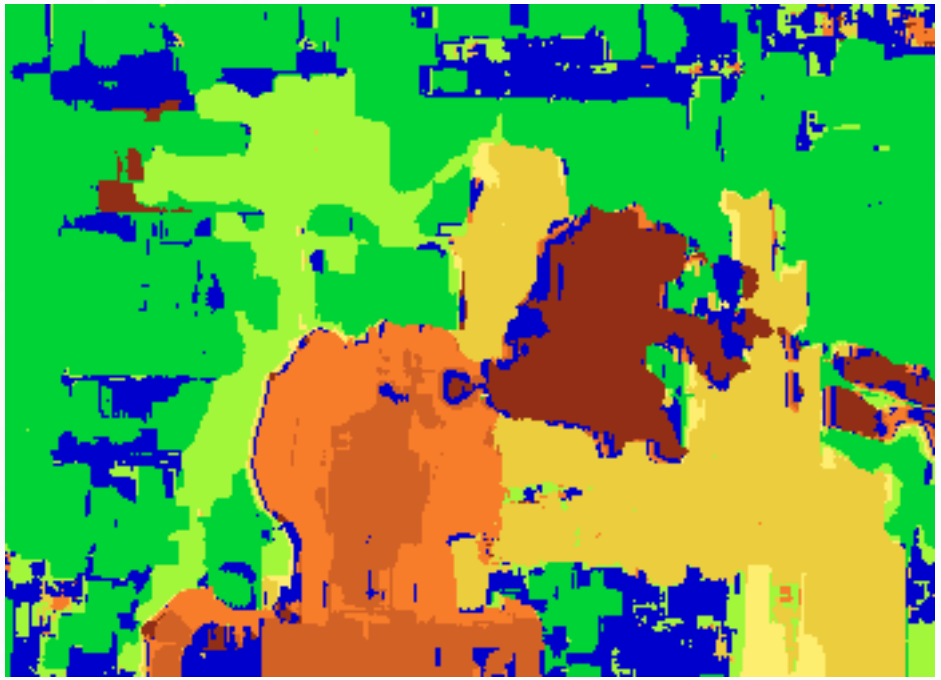
STEREO AS ENERGY MINIMIZATION

Simple pixel / window matching: choose the minimum of each column in the DSI independently:

$$d(x,y) = \arg\min_{d'} C(x,y,d')$$

Slide credit:
Kris Kitani

GREEDY SELECTION OF BEST MATCH



STEREO AS ENERGY MINIMIZATION

Better objective function

$$E(d) = \underbrace{E_d(d)}_{\text{match cost}} + \lambda \underbrace{E_s(d)}_{\text{smoothness cost}}$$

Want each pixel to find a good match in the other image

Adjacent pixels should (usually) move about the same amount

Slide credit:
Kris Kitani

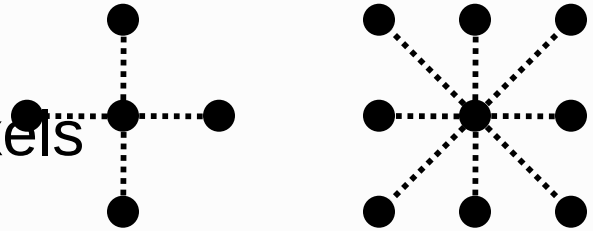
STEREO AS ENERGY MINIMIZATION

$$E(d) = E_d(d) + \lambda E_s(d)$$

match cost: $$E_d(d) = \sum_{(x,y)\in I} C(x, y, d(x, y))$$

smoothness cost: $$E_s(d) = \sum_{(p,q)\in\mathcal{E}} V(d_p, d_q)$$

$\mathcal{E}$: set of neighboring pixels



4-connected neighborhood

8-connected neighborhood

Slide credit: Kris Kitani

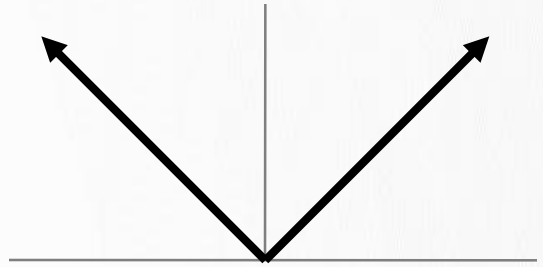
SMOOTHNESS COST

$$E_s(d) = \sum_{(p,q)\in\mathcal{E}} V(d_p, d_q)$$

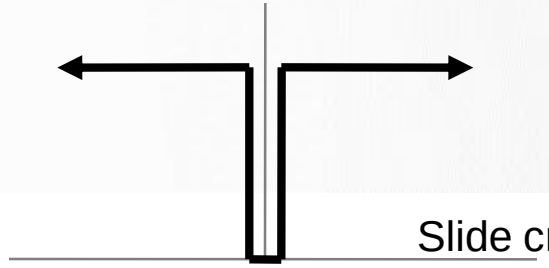
How do we choose *V*?

$$V(d_p, d_q) = |d_p - d_q|$$

L_1 distance



$$V(d_p, d_q) = \begin{cases} 0 & \text{if } d_p = d_q \\ 1 & \text{if } d_p \neq d_q \end{cases}$$



"Potts model"

Slide credit:
Kris Kitani

STEREO AS A MINIMIZATION PROBLEM

$$E(d) = E_d(d) + \lambda E_s(d)$$

The 2D problem has many local minima

Gradient descent doesn't work well

And a large search space

n x *m* image w/ *k* disparities has k^{nm} possible solutions

Finding the global minimum is NP-hard in general

Good approximations exist…

Slide credit:
Kris Kitani

DYNAMIC PROGRAMMING

$$E(d) = E_d(d) + \lambda E_s(d)$$

Can minimize this independently per scanline using dynamic programming (DP)

Basic idea: incrementally build a table of costs *D* one column at a time

$D(x, y, i)$: minimum cost of solution such that $d(x,y) = i$

Base cas $D(0, y, i) = C(0, y, i), i = 0, \ldots, L$ (L = max disparity)

Recurrence: $D(x, y, i) = C(x, y, i) + \min_{j \in \{0,1,\ldots,L\}} D(x-1, y, j) + \lambda|i - j|$

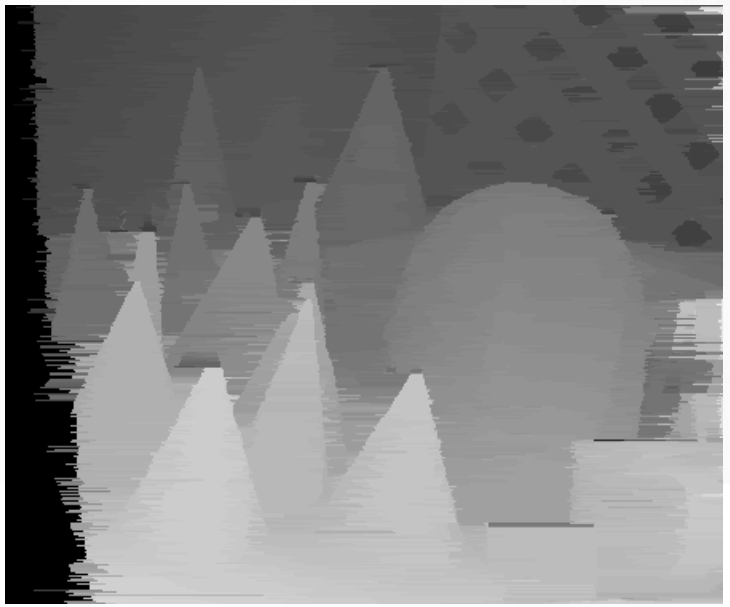
Slide credit:
Kris Kitani

DYNAMIC PROGRAMMING

Finds “smooth”, low-cost path through DPI from left to right

Slide credit:
Kris Kitani

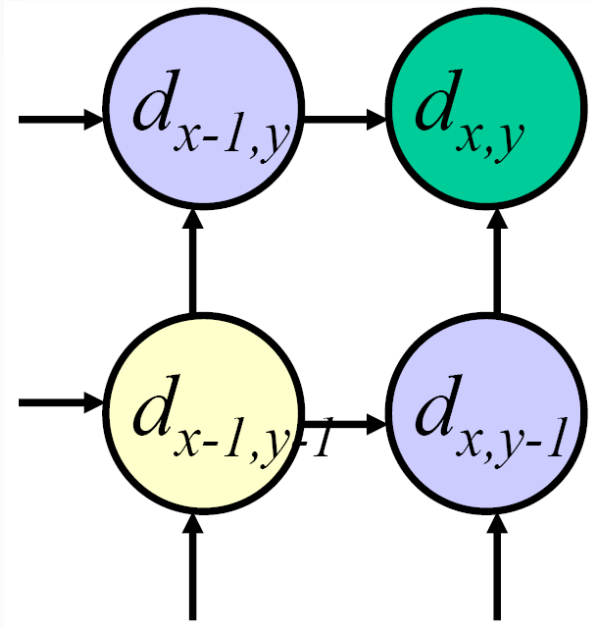
DYNAMIC PROGRAMMING



Slide credit:
Kris Kitani

DYNAMIC PROGRAMMING

Can we apply this trick in 2D as well?



No: $d_{x,y-1}$ and $d_{x-1,y}$ may depend on different values of $d_{x-1,y-1}$

Slide credit: D. Huttenlocher

REAL-TIME STEREO

Nomad robot searches for meteorites in Antartica
http://www.frc.ri.cmu.edu/projects/meteorobot/index.html

Used for robot navigation (and other tasks)

Several real-time stereo techniques have been developed (most based on simple discrete search)

Slide credit:
Kris Kitani

STEREO RECONSTRUCTION PIPELINE

Steps

Calibrate cameras

Rectify images

Compute disparity

Estimate depth

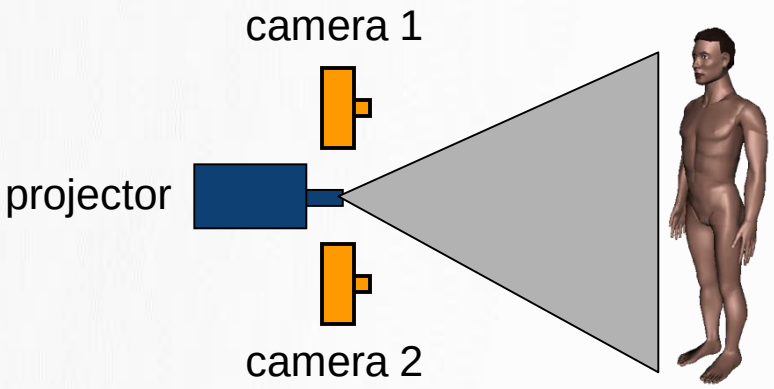
What will cause errors?

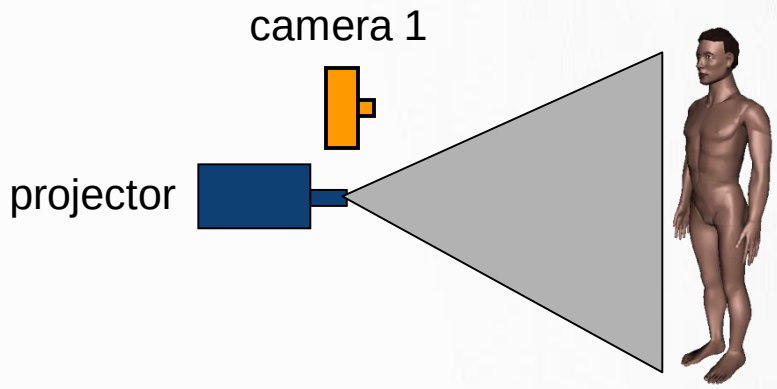
- Camera calibration errors
- Poor image resolution
- Occlusions
- Violations of brightness constancy (specular reflections)
- Large motions
- **Low-contrast image regions**

Slide credit:
Kris Kitani

ACTIVE STEREO WITH STRUCTURED LIGHT

Li Zhang's one-shot stereo





Project "structured" light patterns onto the object

simplifies the correspondence problem

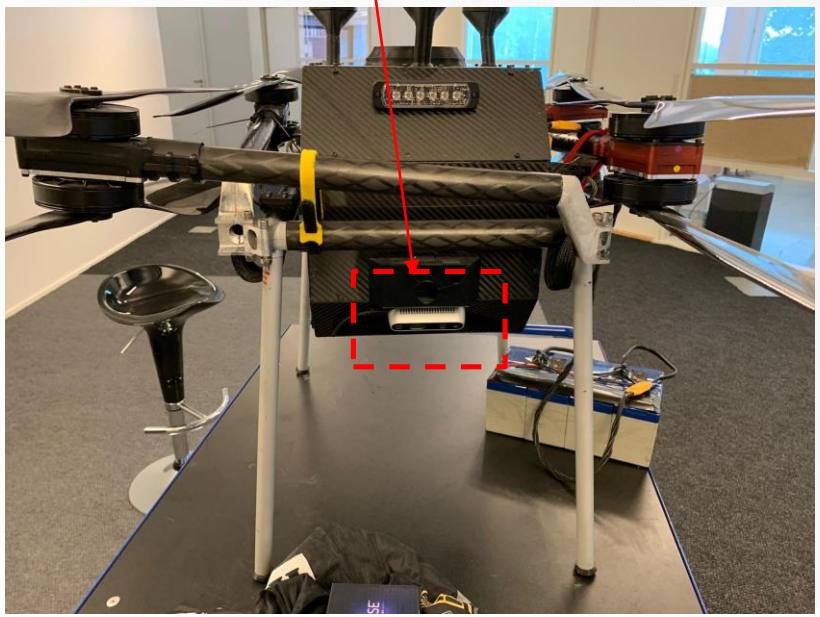
basis for active depth sensors, such as Kinect and iPhone X (using IR)

ACTIVE STEREO WITH STRUCTURED LIGHT

https://ios.gadgethacks.com/news/watch-iphone-xs-30k-ir-dots-scan-your-face-0180944/

INTEL REALSENSE

- RealSense cameras
  - Stereoscopic, subpixeld disparity accuracy
  - Function well **outdoors**
  - Low power consumption, 100x10x4 mm in size
  - Includes 3 cameras
  - Parameters can’t be modified, depth solution not optimal




60˚ 10 1.2 N, 24˚ 57 18 E