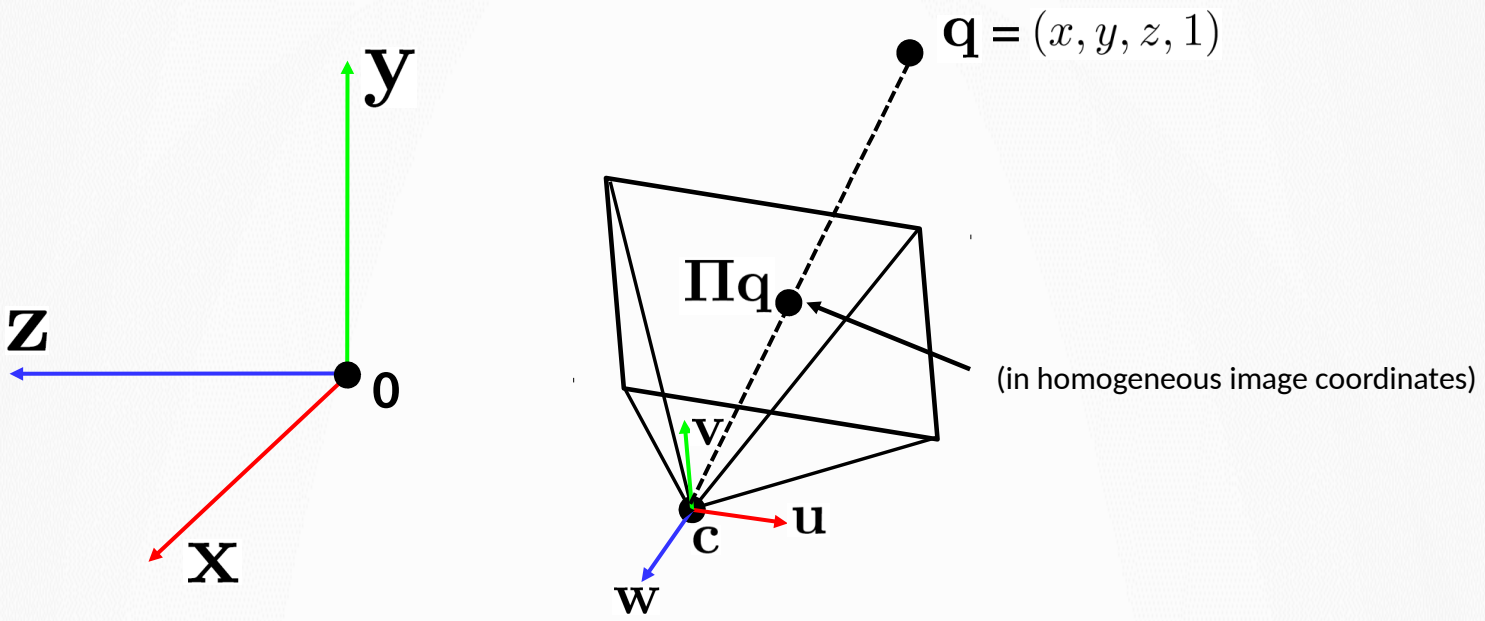
COMPUTER VISION

LECTURE 14 18.10.2019

Laura Ruotsalainen, Associate Professor
Department of Computer Science

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Camera projection matrix: recap



(in homogeneous image coordinates)

slide credit Noah Snavely

Projection matrix

$$\Pi = \mathbf{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{c} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

intrinsics

projection

rotation

translation

The **K** matrix converts 3D rays in the camera's coordinate system to 2D image points in image (pixel) coordinates.

This part converts 3D points in world coordinates to 3D rays in the camera's coordinate system. There are 6 parameters represented (3 for position/translation, 3 for rotation).

slide credit Noah Snavely

Projection matrix

$$\Pi = \mathbf{K} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{c} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}}_{\text{translation}}$$

intrinsics

$$\underbrace{\begin{bmatrix} \mathbf{R} & -\mathbf{Rc} \end{bmatrix}}_{\mathbf{t}}$$

(**t** in book's notation)

$$\Pi = \mathbf{K} \begin{bmatrix} \mathbf{R} & -\mathbf{Rc} \end{bmatrix}$$

slide credit Noah Snavely

Typical intrinsics matrix

$$\mathbf{K} = \begin{bmatrix} -f & 0 & x_c \\ 0 & -f & y_c \\ 0 & 0 & 1 \end{bmatrix}$$

- **2D affine transform** corresponding to a scale by *f* (focal length) and a translation by (x_c, y_c) (principal point)

slide credit Noah Snavely

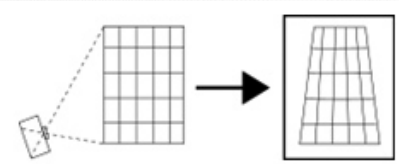
Perspective distortion

- Problem for architectural photography: converging verticals

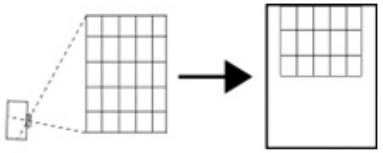
Source: F. Durand

Perspective distortion

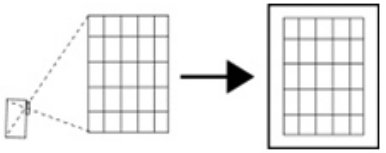
- Problem for architectural photography: converging verticals



Tilting the camera upwards results in converging verticals

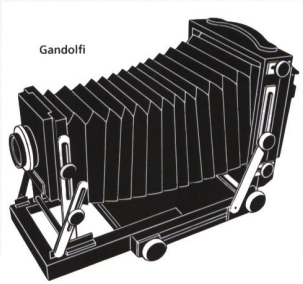


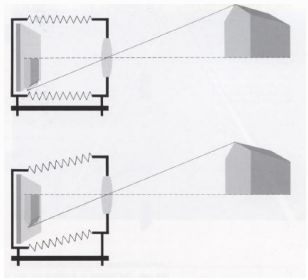
Keeping the camera level, with an ordinary lens, captures only the bottom portion of the building



Shifting the lens upwards results in a picture of the entire subject

- Solution: view camera (lens shifted w.r.t





(Corresponds to shifting the *principal point*)

Source: F. Durand

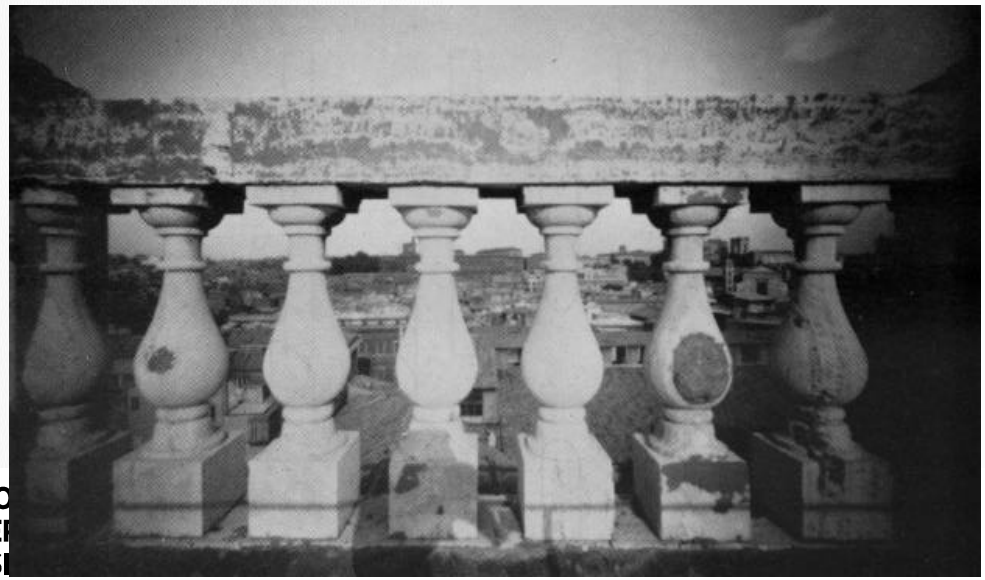
Perspective distortion

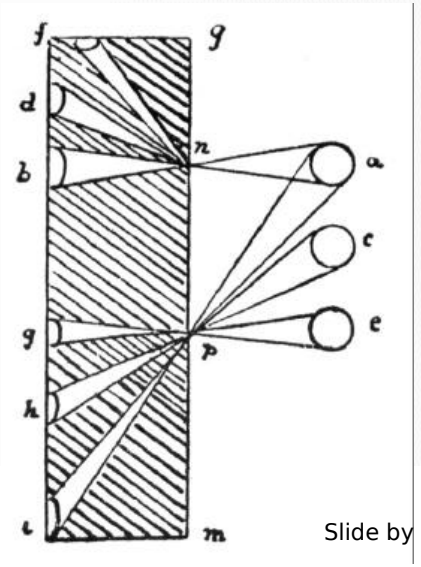
- Problem for architectural photography: converging verticals
- Result:

Source: F. Durand

Perspective distortion

- The exterior columns appear bigger
- The distortion is not due to lens flaws
- Problem pointed out by Da Vinci

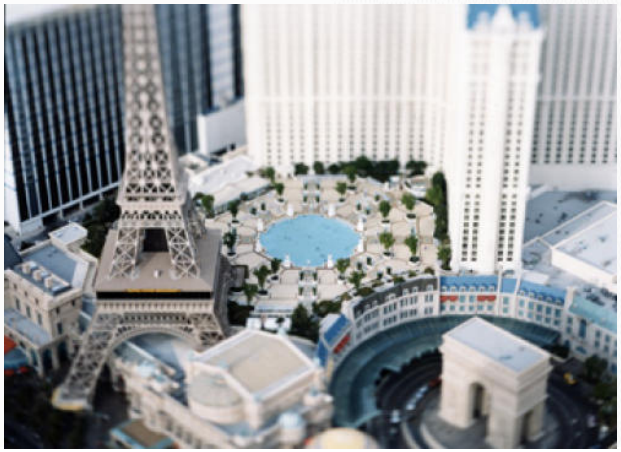




Slide by F. Durand

Tilt-shift

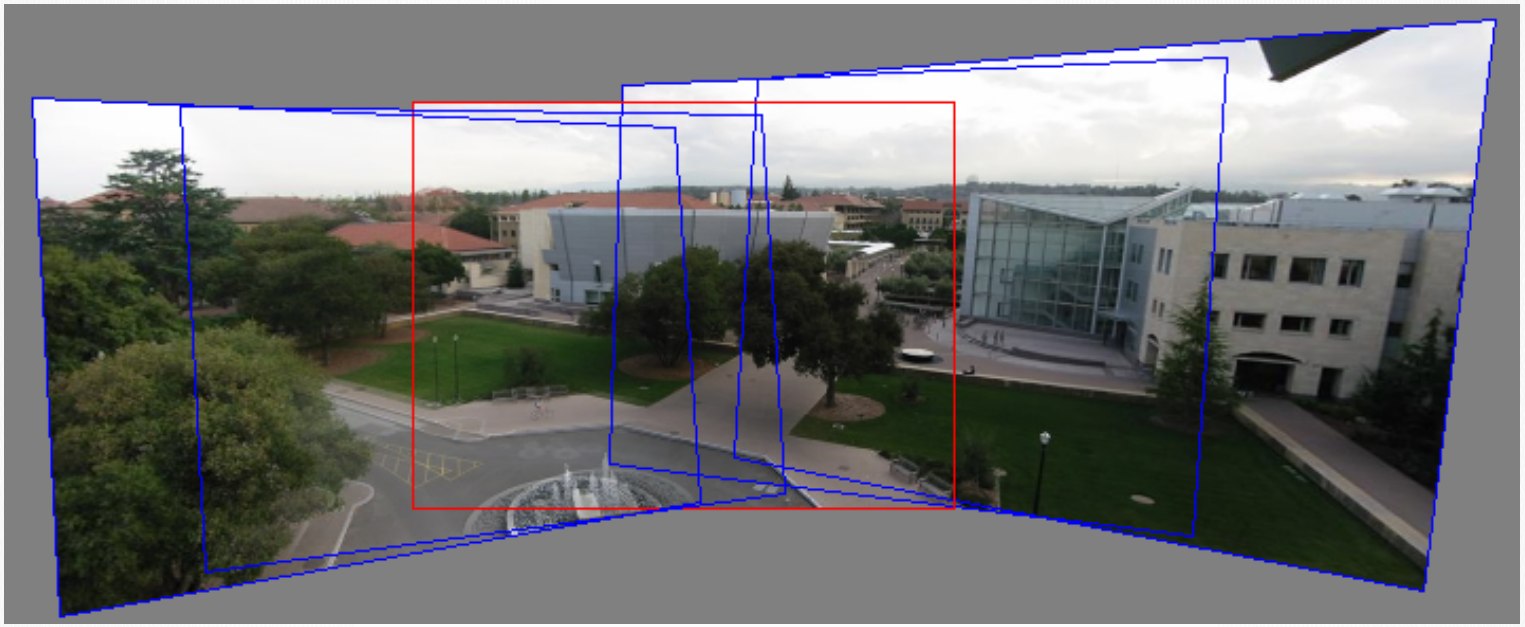
http://www.northlight-images.co.uk/article_pages/tilt_and_shift_ts-e.html



Titlt-shift images from Olivo Barbieri and Photoshop imitations

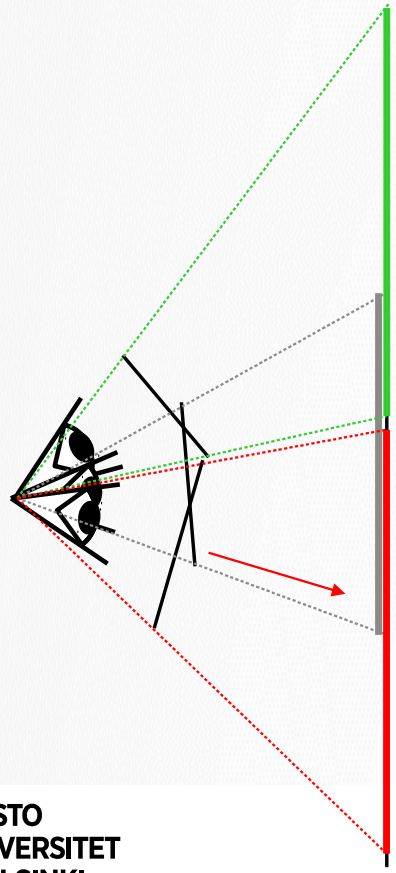
slide credit Noah Snavely

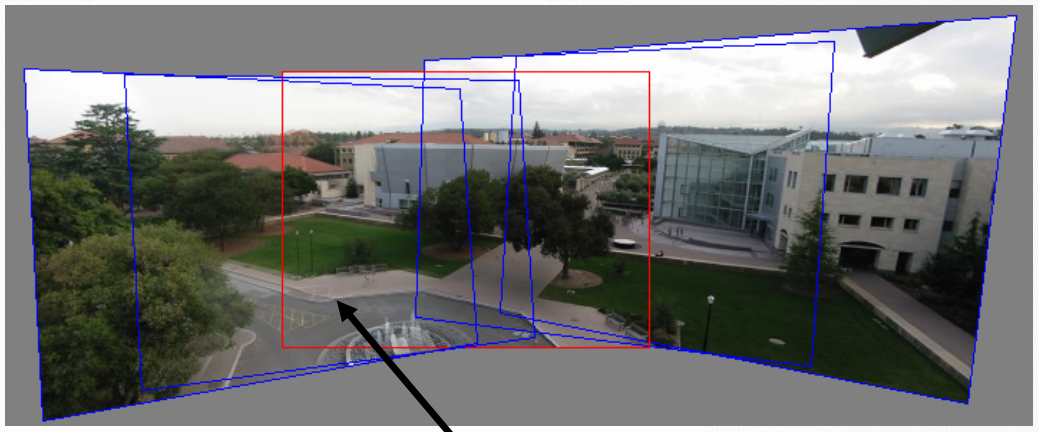
Panoramas



Can we use homographies to create a 360 panorama?

Idea: projecting images onto a common plane





each image is warped with a homography $\mathbf{H}$

Can’t create a 360 panorama this way… we’ll fix this shortly

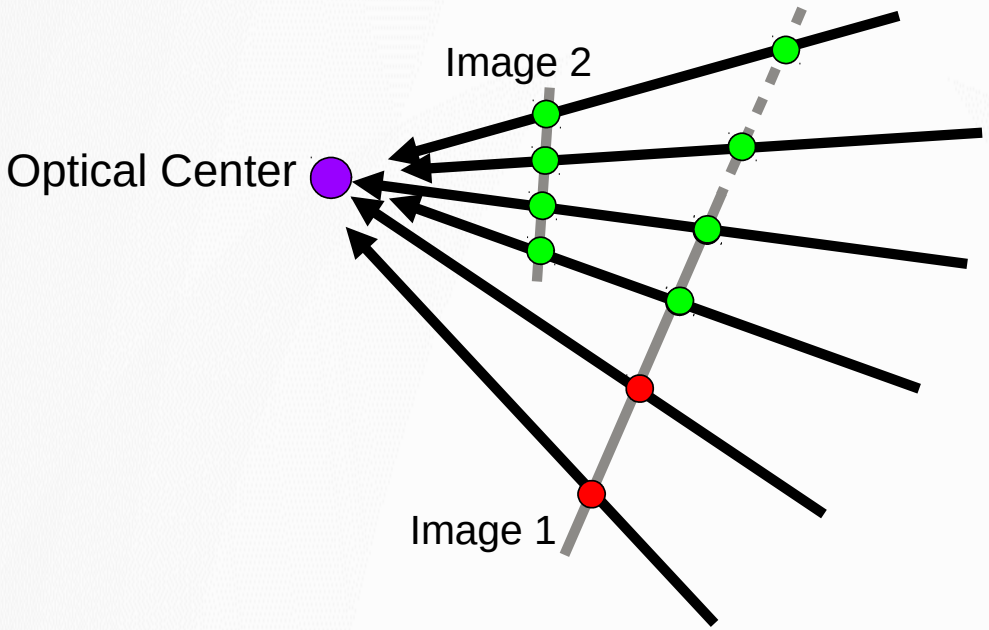
mosaic projection plane

slide credit Noah Snavely

Creating a panorama

- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation between second image and first
  - Transform the second image to overlap with the first
  - Blend the two together to create a mosaic
  - If there are more images, repeat

Geometric Interpretation of Mosaics



- If we capture all 360° of rays, we can create a 360° panorama
- The basic operation is *projecting* an image from one plane to another
- The projective transformation is scene-INDEPENDENT
  - This depends on all the images having the same optical center

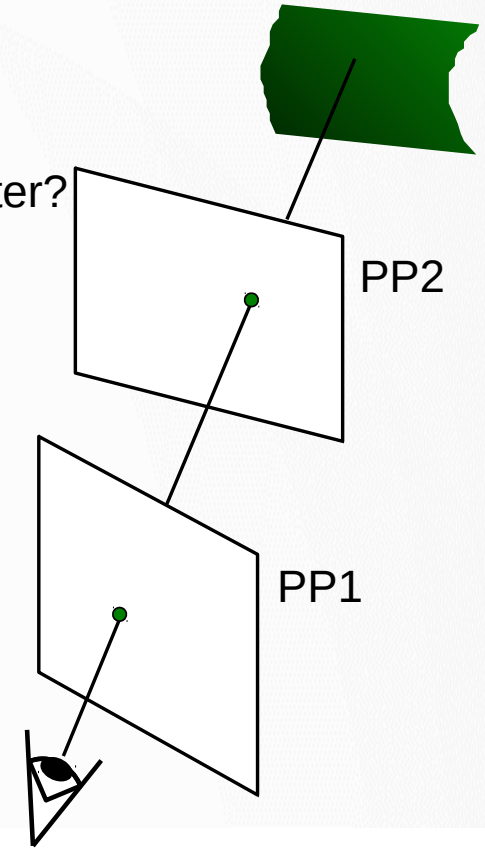
slide credit Noah Snavely

Image reprojection

- Basic question
  - How to relate two images from the same camera center?
    - how to map a pixel from PP1 to PP2

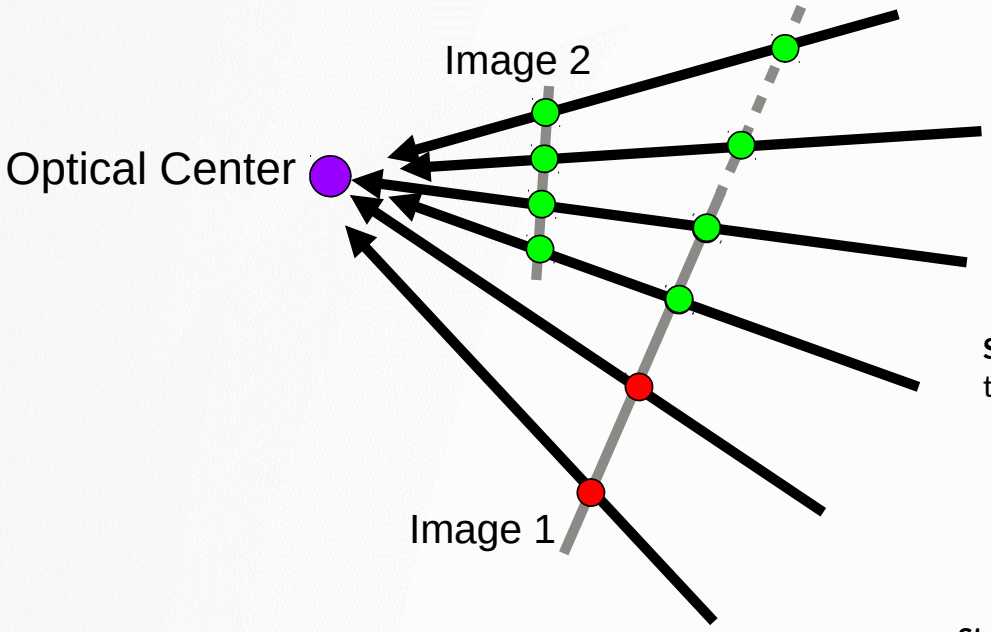
Answer

- Cast a ray through each pixel in PP1
- Draw the pixel where that ray intersects PP2



slide credit Noah Snavely

What is the transformation?



Step 1: Convert pixels in image 2 to rays in camera 2's coordinate system.

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \mathbf{K}_2^{-1} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

Step 2: Convert rays in camera 2's coordinates to rays in camera 1's coordinates.

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \mathbf{R}_2^T \mathbf{K}_2^{-1} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

Step 3: Convert rays in camera 1's coordinates to pixels in image 1's coordinates.

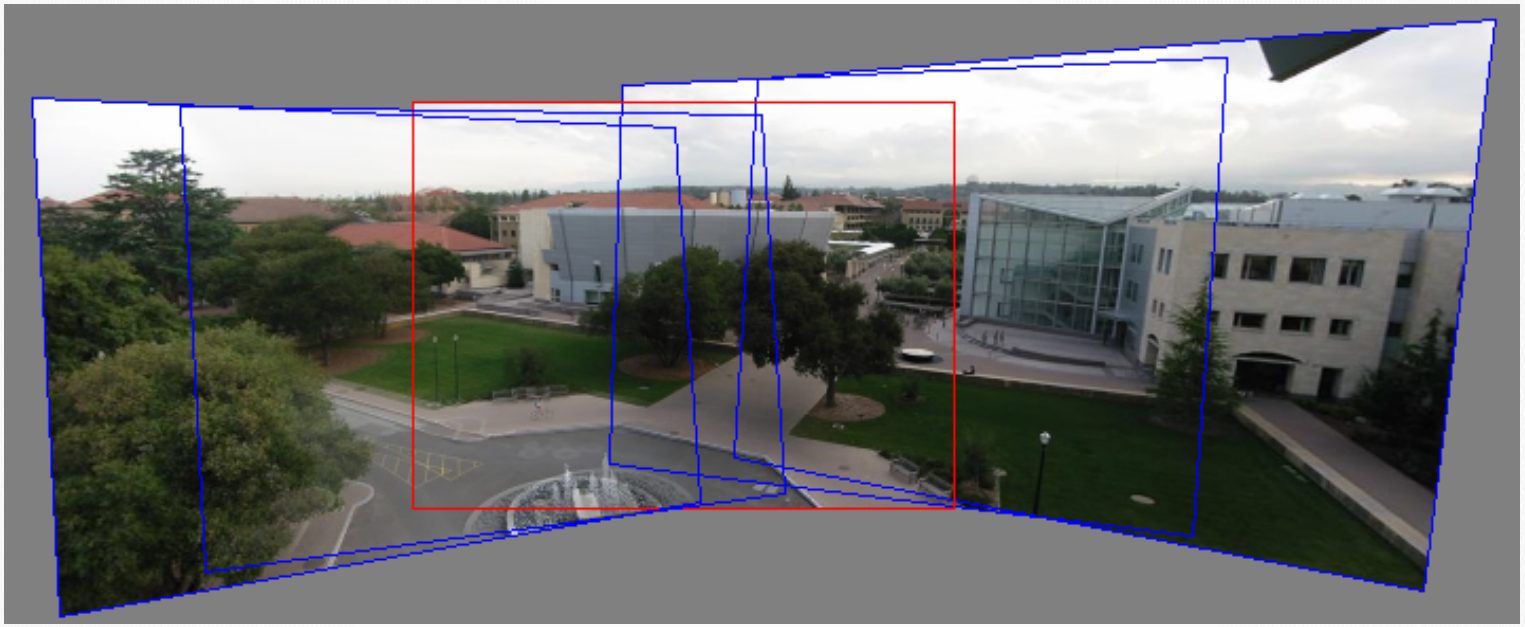
$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \sim \underbrace{\mathbf{K}_1 \mathbf{R}_2^T \mathbf{K}_2^{-1}} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

3x3 homography

How do we map points in image 2 into image 1?

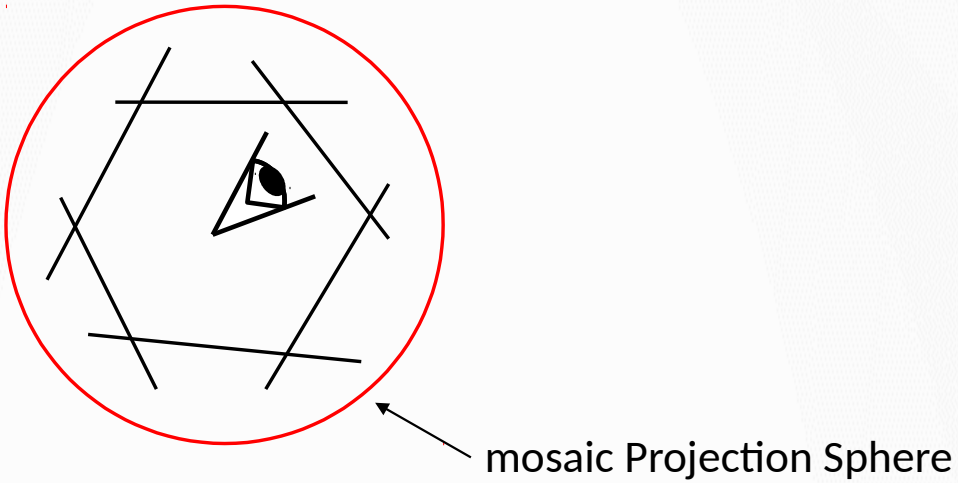
	image 1	image 2
intrinsics	$\mathbf{K}_1$	$\mathbf{K}_2$
extrinsics (rotation only)	$\mathbf{R}_1 = \mathbf{I}_{3\times 3}$	$\mathbf{R}_2$

Can we use homography to create a 360 panorama?



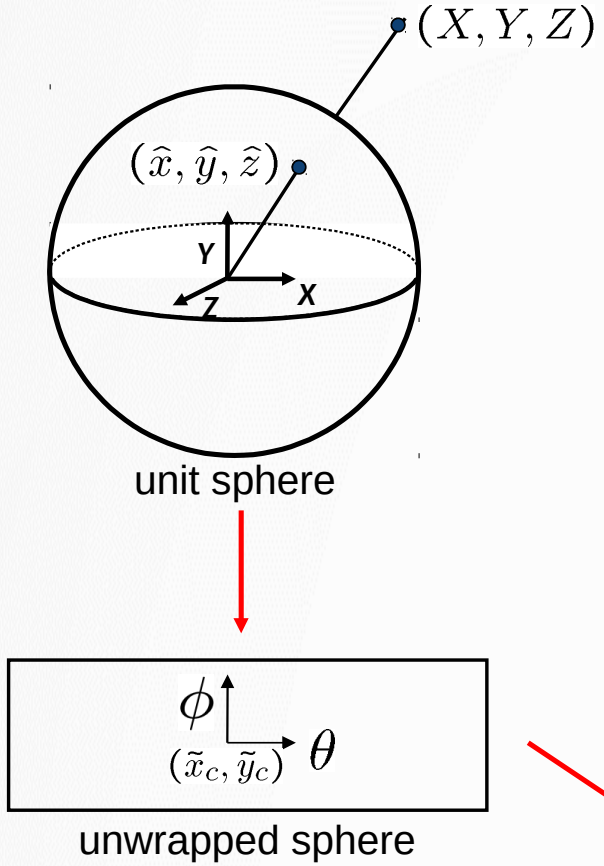
Panoramas

- What if you want a 360° field of view?



slide credit Noah Snavely

Spherical projection



- Map 3D point (X,Y,Z) onto sphere

$$(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2+Y^2+Z^2}}(X, Y, Z)$$

- Convert to spherical coordinates

$$(sin\theta cos\phi, sin\phi, cos\theta cos\phi) = (\hat{x}, \hat{y}, \hat{z})$$

- Convert to spherical image coordinates

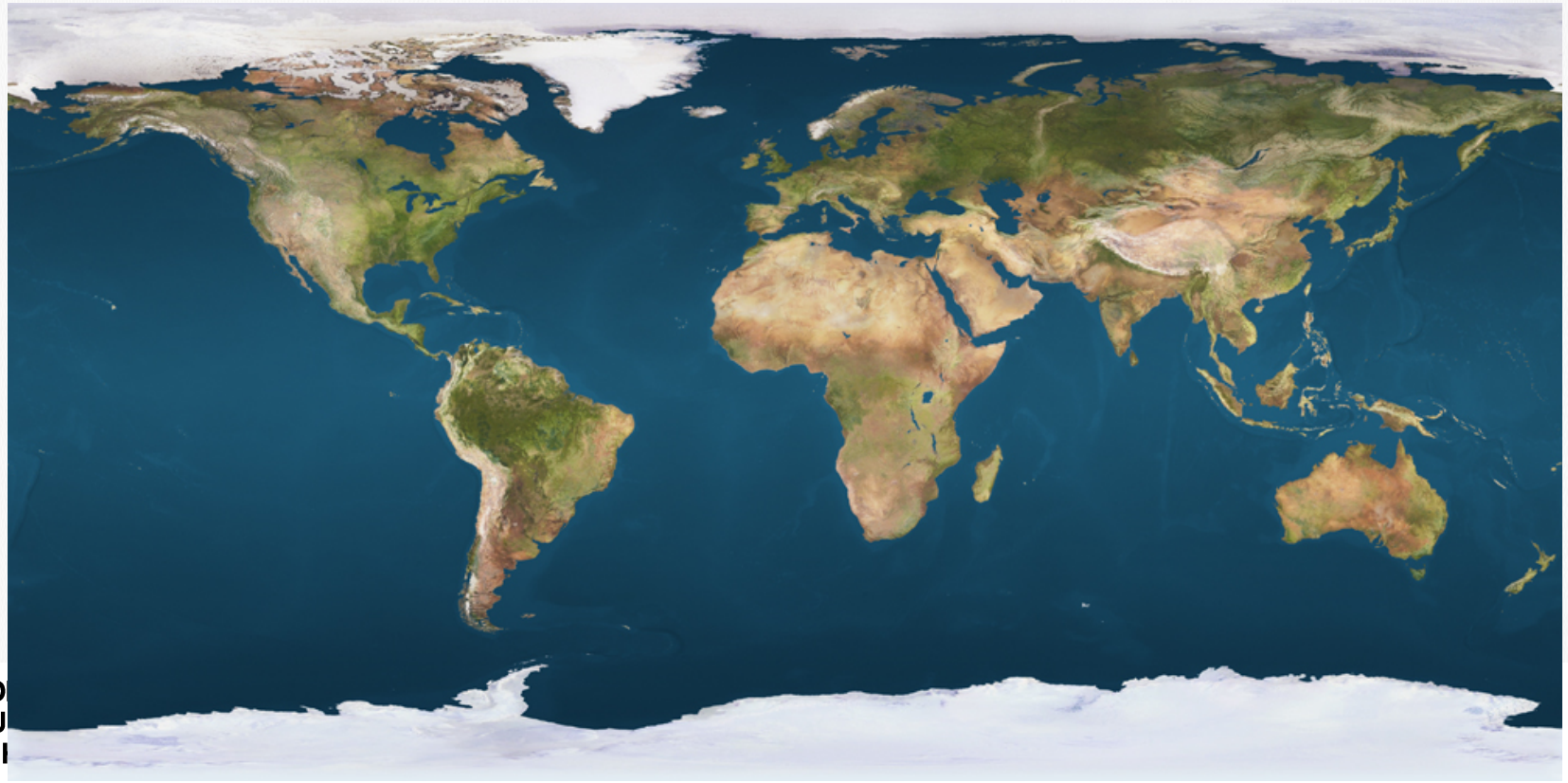
$$(\tilde{x}, \tilde{y}) = (s\theta, s\phi) + (\tilde{x}_c, \tilde{y}_c)$$

  - s defines size of the final image
    - » often convenient to set s = camera focal length

slide credit Noah Snavely

Unwrapping a sphere

Credit: JHT's Planetary Pixel Emporium



Spherical reprojection

input

f = 200 (pixels)

f = 400

f = 800

- Map image to spherical coordinates
  - need to know the focal length

slide credit Noah Snavely

Aligning spherical images

- Suppose we rotate the camera by θ about the vertical axis
  - How does this change the spherical image?

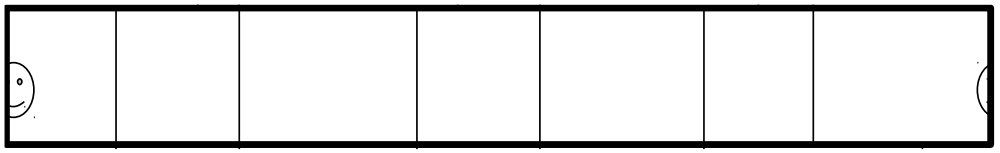
slide credit Noah Snavely

Aligning spherical images

- Suppose we rotate the camera by θ about the vertical axis
  - How does this change the spherical image?
    - Translation by θ
  - This means that we can align spherical images by translation

slide credit Noah Snavely

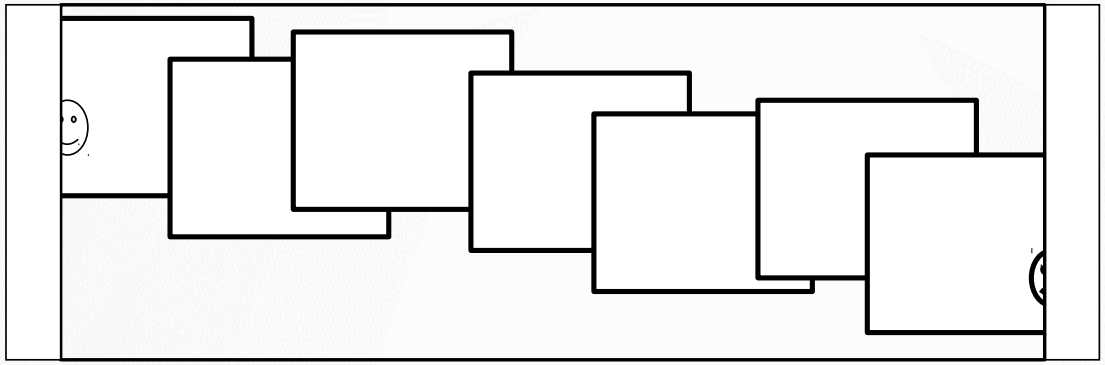
Assembling the panorama



- Stitch pairs together, blend, then crop

slide credit Noah Snavely

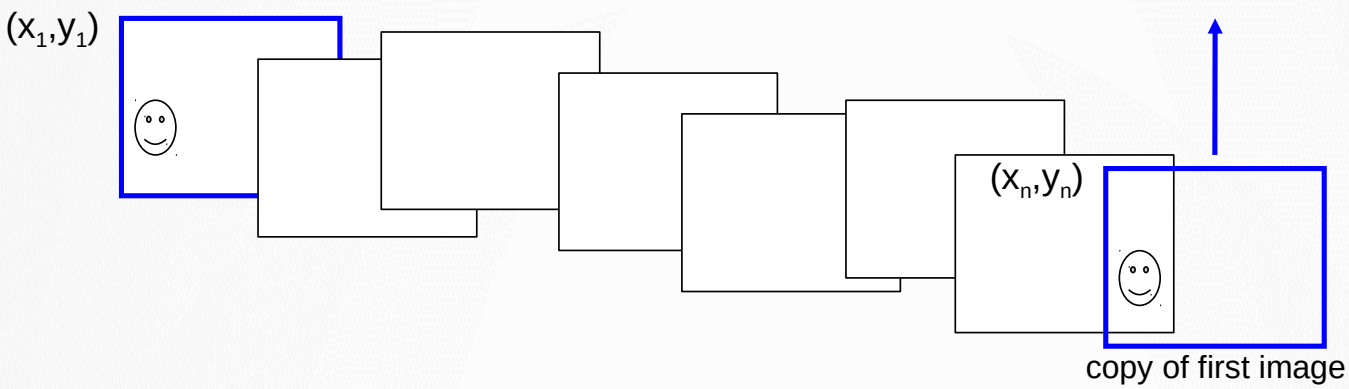
Problem: Drift



- Error accumulation
  - small errors accumulate over time

slide credit Noah Snavely

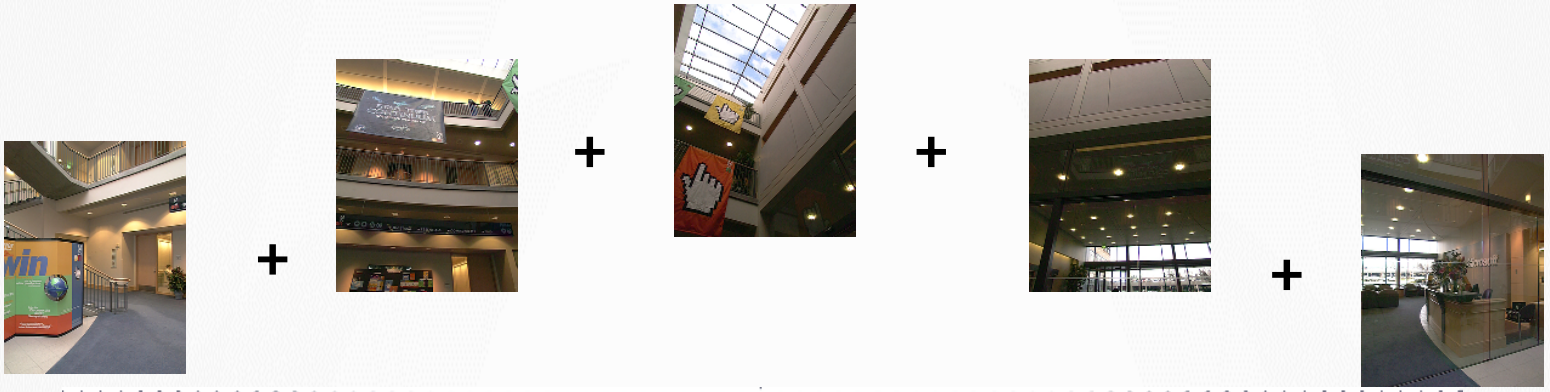
Problem: Drift



- Solution
  - add another copy of first image at the end
  - this gives a constraint: $y_n = y_1$
  - there are a bunch of ways to solve this problem
    - add displacement of $(y_1 - y_n)/(n-1)$ to each image after the first
    - **apply an affine warp: y' = y + ax**
    - run a big optimization problem, incorporating this constraint
      - best solution, but more complicated

slide credit Noah Snavely

Spherical panoramas



Microsoft Lobby: http://www.acm.org/pubs/citations/proceedings/graph/258734/p251-szeliski

Different projections are possible

Cube-map

Blending

- We’ve aligned the images – now what?

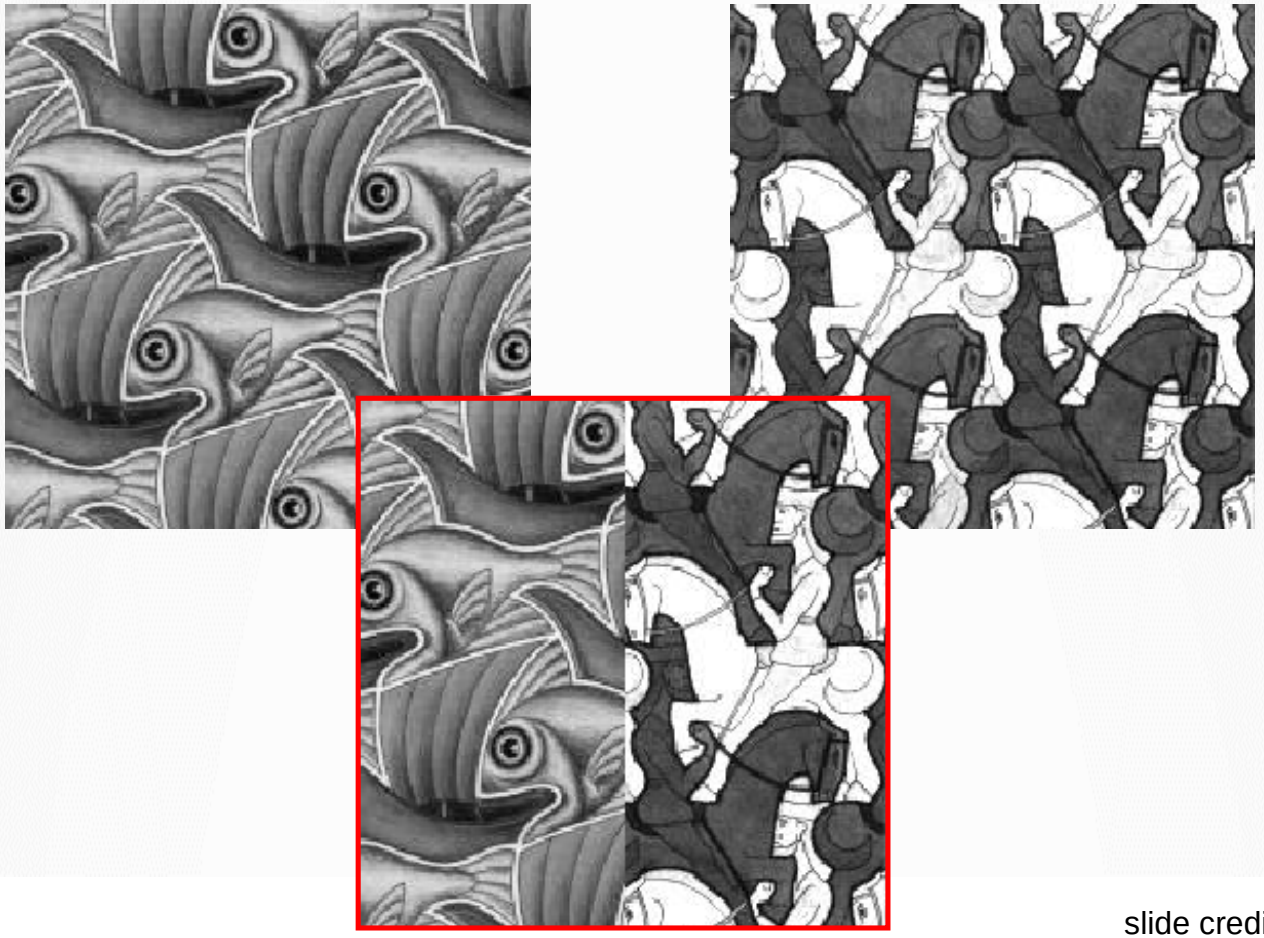
slide credit Noah Snavely

Blending

- Want to seamlessly blend them together

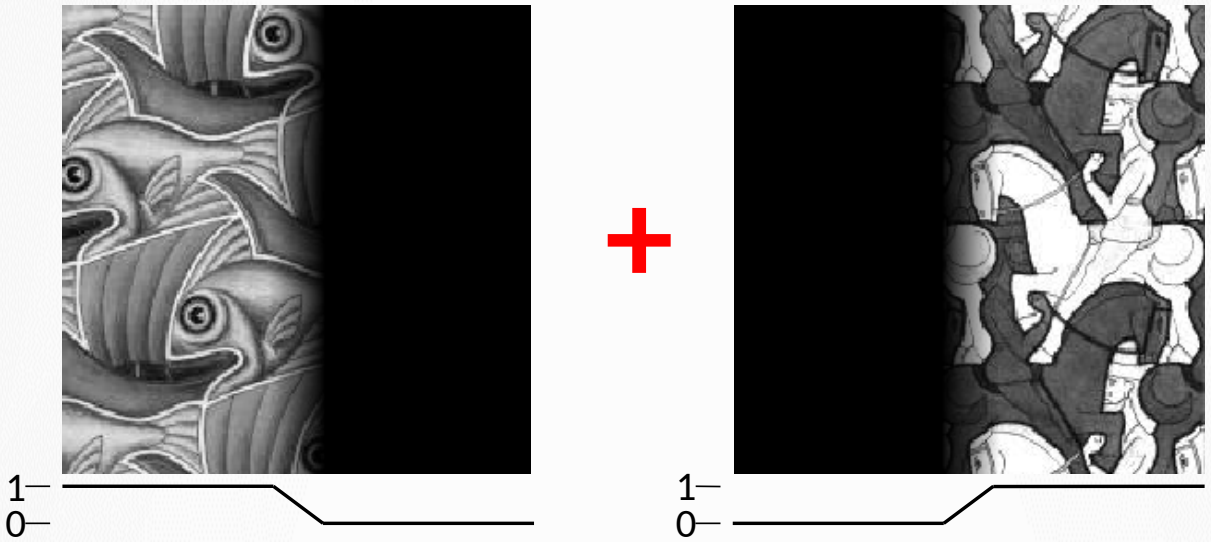
slide credit Noah Snavely

Image Blending



slide credit Noah Snavely

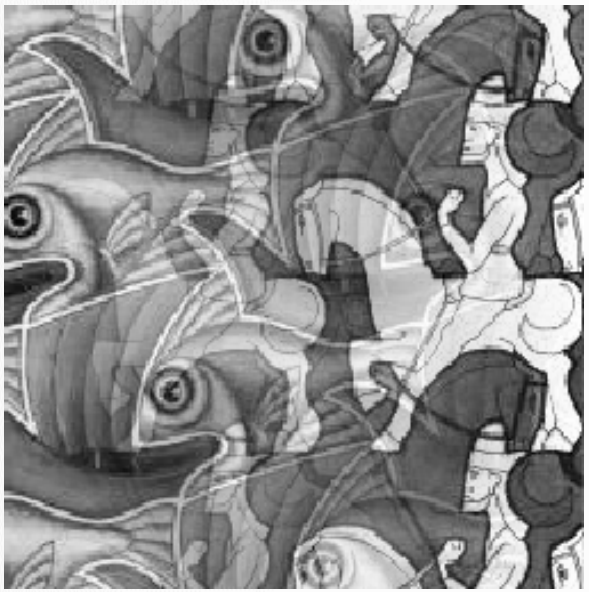
Feathering

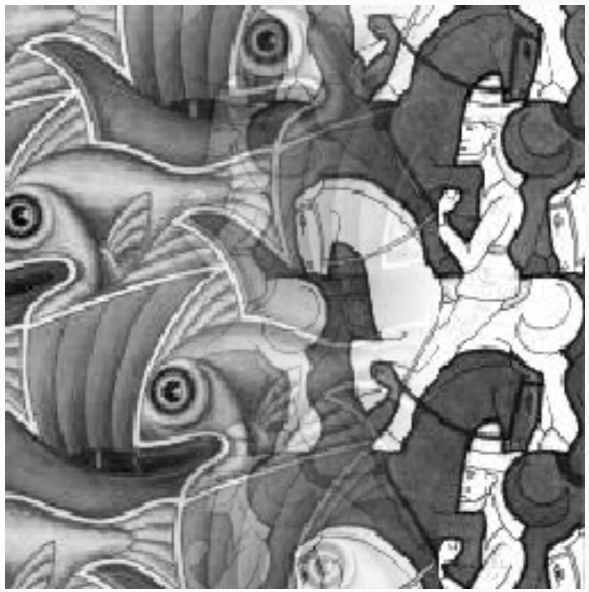


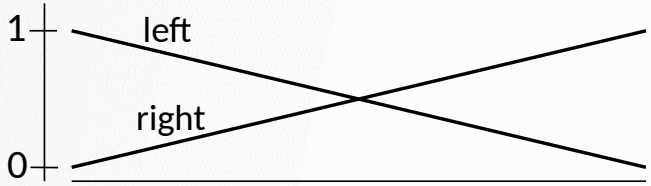
HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

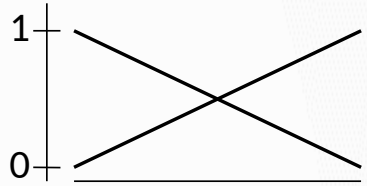
slide credit Noah Snavely

Effect of window size



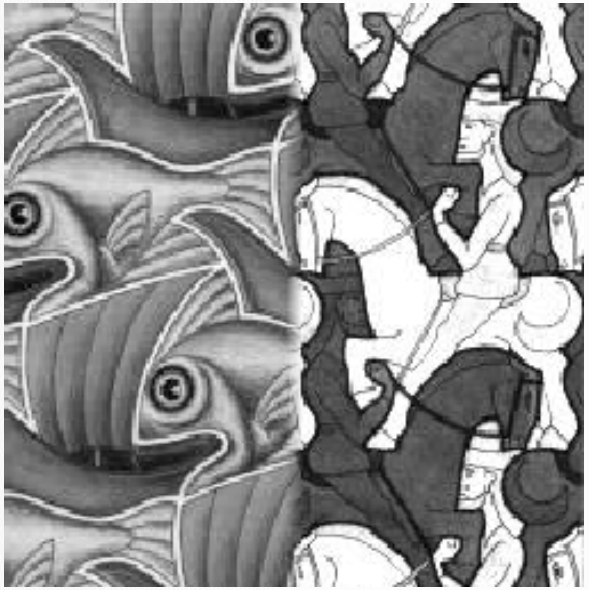


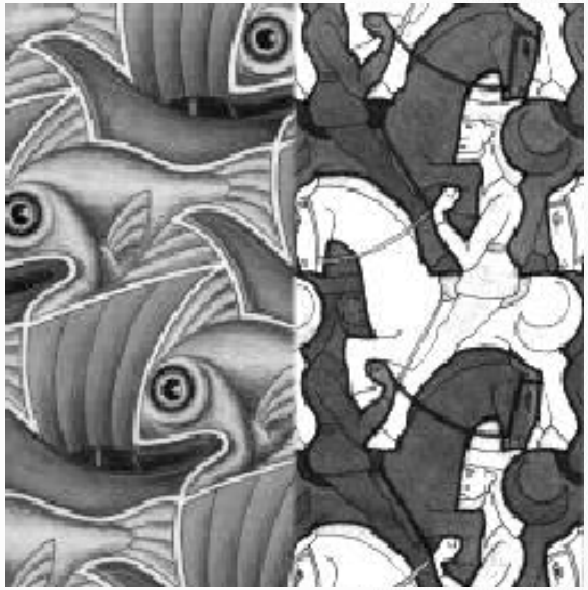


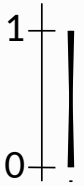


slide credit Noah Snavely

Effect of window size

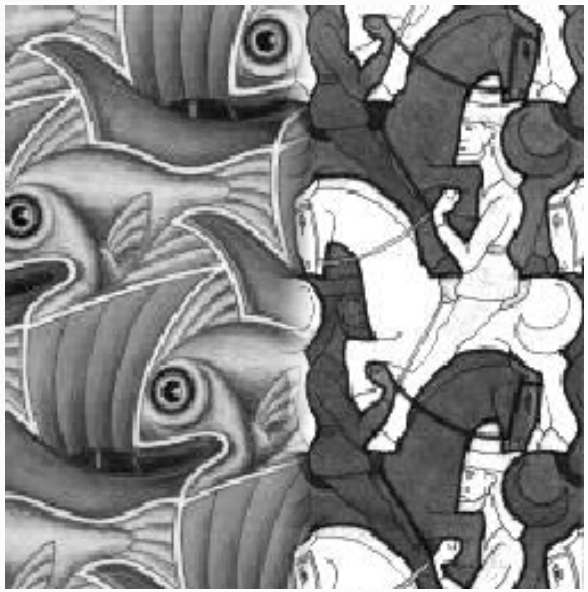


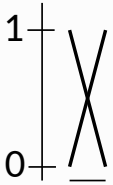




slide credit Noah Snavely

Good window size



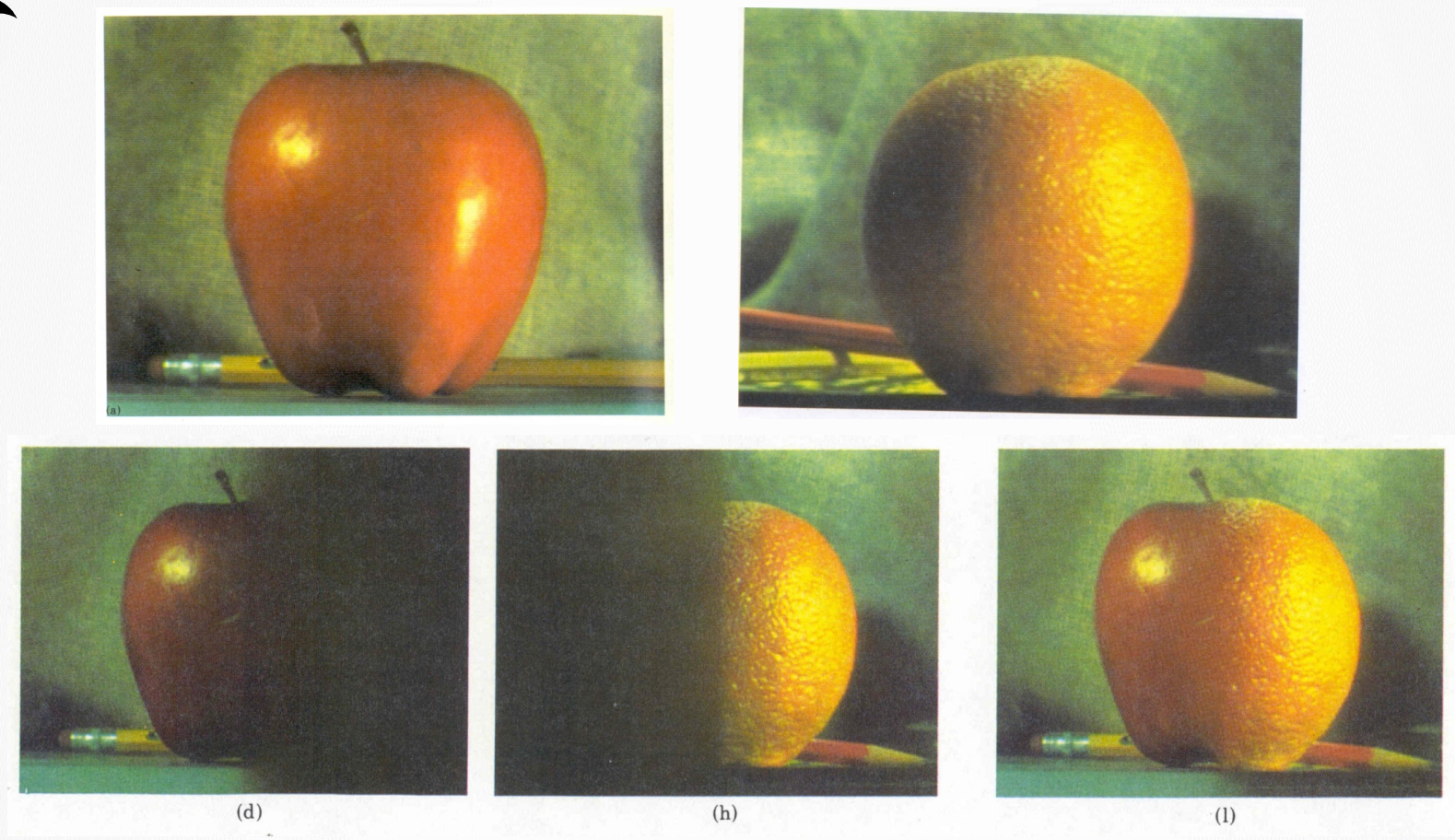


"Optimal" window: smooth but not ghosted

- Doesn't always work...

slide credit Noah Snavely

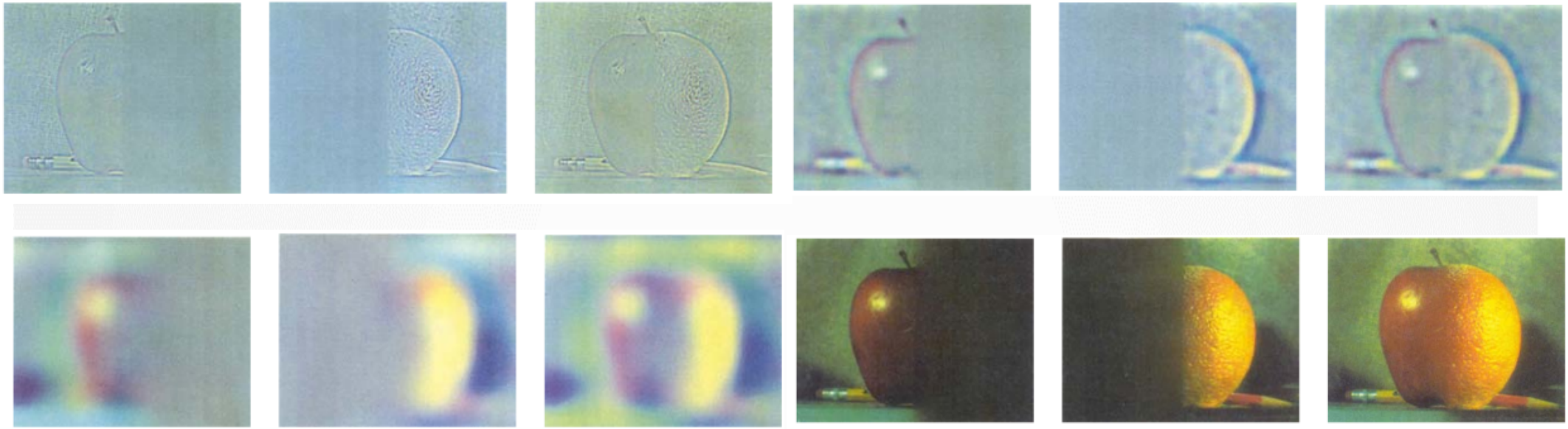
Pyramid blending



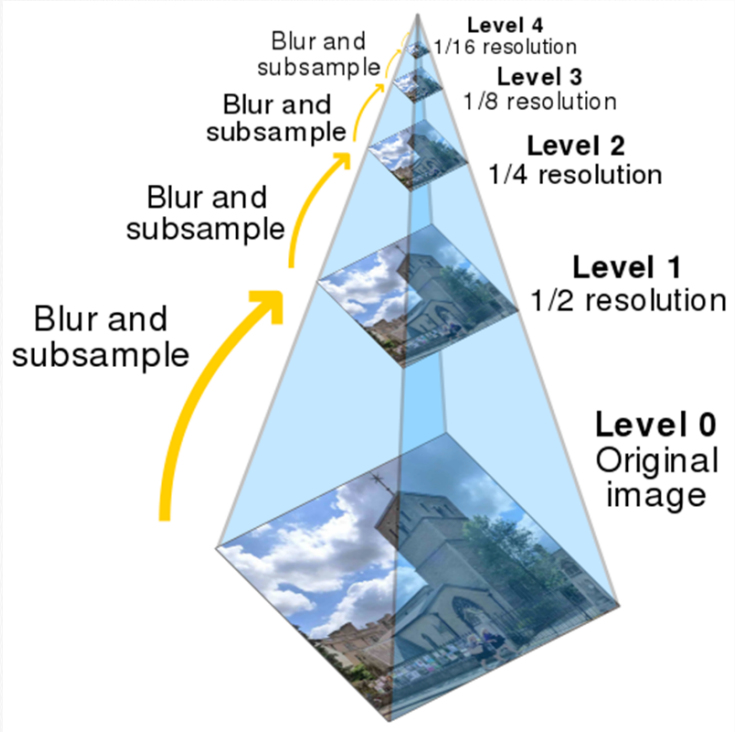
Create a Laplacian pyramid, blend each level

- Burt, P. J. and Adelson, E. H., A multiresolution spline with applications to image mosaics, ACM Transactions on Graphics, 42(4), October 1983, 217-236.

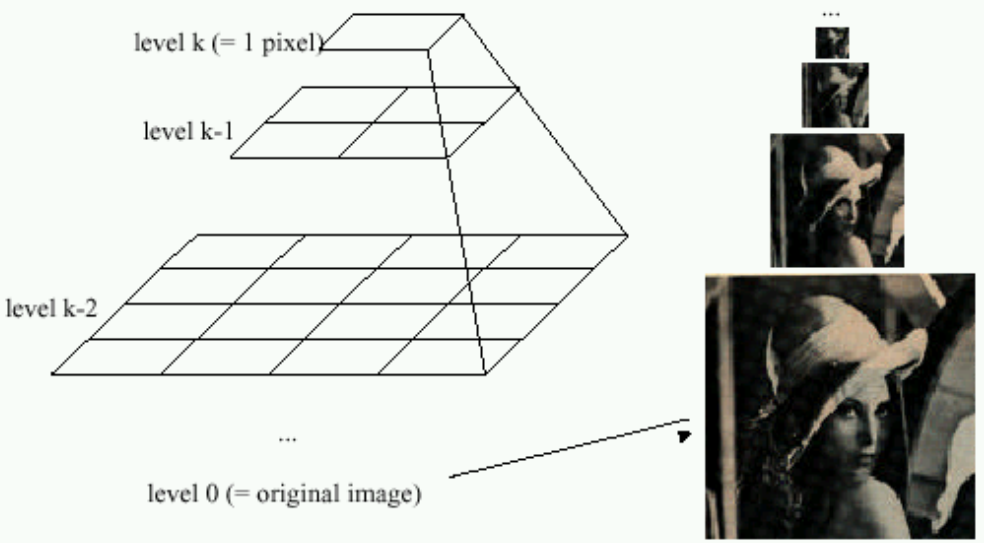
IMAGE BLENDING



RECAP (LECT 2) - GAUSSIAN PYRAMIDS



Idea: Represent NxN image as a "pyramid" of 1x1, 2x2, 4x4,…, 2^kx2^k images (assuming N=2^k)

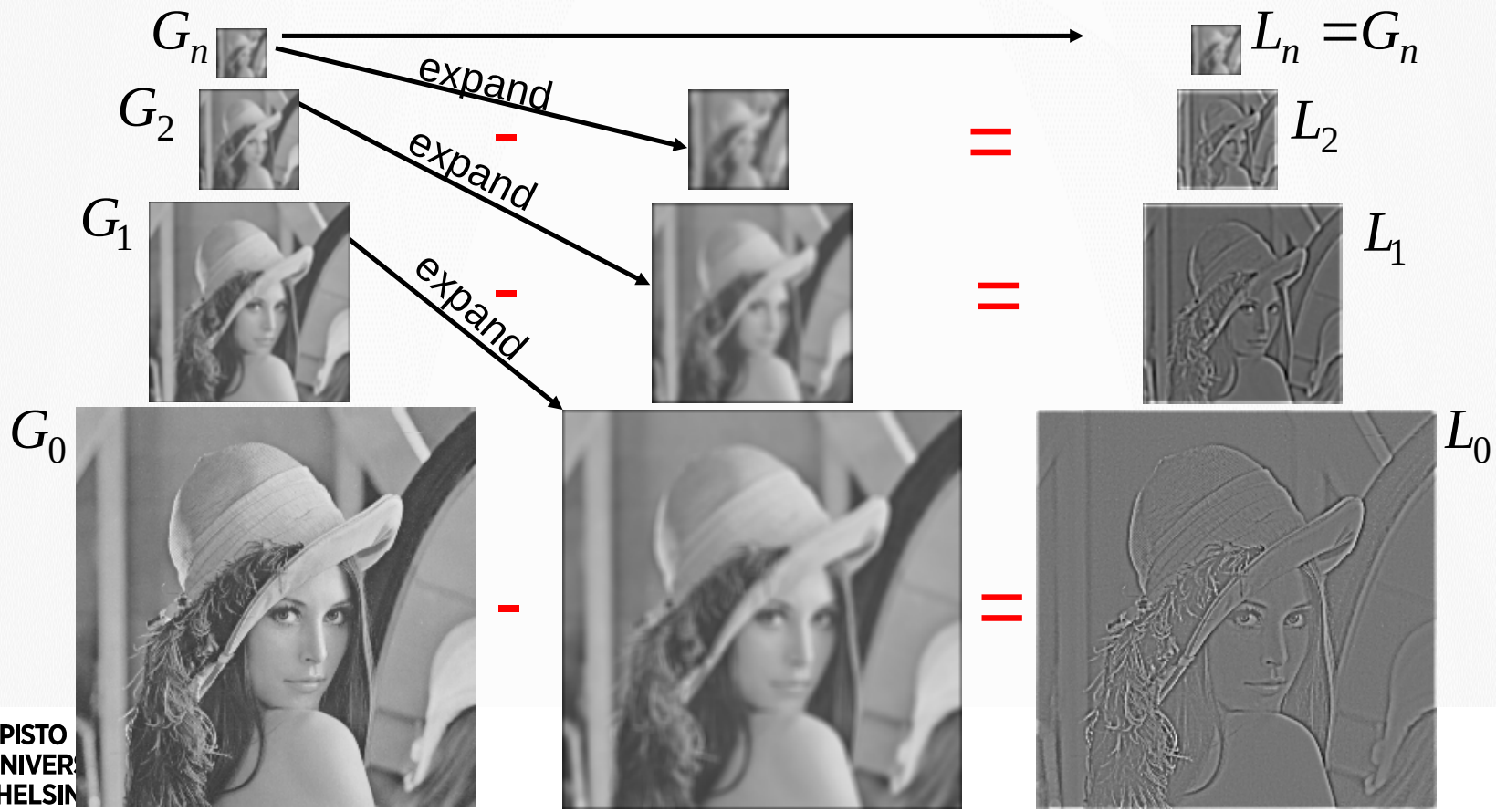


Source: S. Seitz

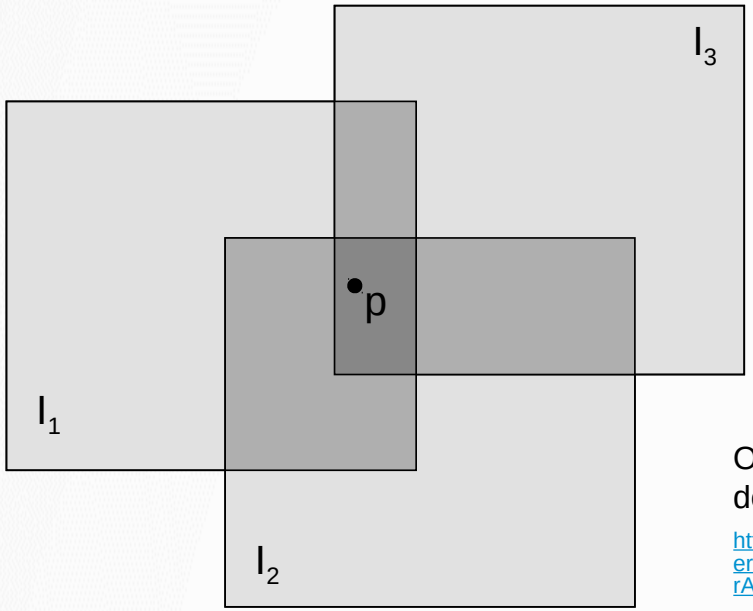
The Laplacian Pyramid

$$L_i = G_i - \text{expand}(G_{i+1})$$

Gaussian Pyramid $G_i = L_i + \text{expand}(G_{i+1})$ Laplacian Pyramid



Alpha Blending



Optional: see Blinn (CGA, 1994) for details:

http://ieeexplore.ieee.org/iel1/38/7531/00310740.pdf?isNumber=7531&prod=JNL&arnumber=310740&arSt=83&ared=87&arAuthor=Blinn%2C+J.F

.

Encoding blend weights: I(x,y) = (αR, αG, αB, α)

color at p = $$\frac{(\alpha_1 R_1,\ \alpha_1 G_1,\ \alpha_1 B_1) + (\alpha_2 R_2,\ \alpha_2 G_2,\ \alpha_2 B_2) + (\alpha_3 R_3,\ \alpha_3 G_3,\ \alpha_3 B_3)}{\alpha_1 + \alpha_2 + \alpha_3}$$

Implement this in two steps:

1. accumulate: add up the (α premultiplied) RGBα values at each pixel
2. normalize: divide each pixel's accumulated RGB by its α value

   Q: what if α = 0?

slide credit Noah Snavely

Some panorama examples

- Every image on Google Streetview

Magic: ghost removal

M. Uyttendaele, A. Eden, and R. Szeliski.
Eliminating ghosting and exposure artifacts in image mosaics.
In Proceedings of the Interational Conference on Computer Vision and Pattern Recognition, volume 2, pages 509--516, Kauai, Hawaii, December 2001.

Magic: ghost removal

M. Uyttendaele, A. Eden, and R. Szeliski.
Eliminating ghosting and exposure artifacts in image mosaics.
In Proceedings of the Interational Conference on Computer Vision and Pattern Recognition, volume 2, pages 509--516, Kauai, Hawaii, December 2001.

CHAPTER 3: TRACKING

Augmented Reality – Principles and Practice

http://www.augmentedrealitybook.org

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

TRACKING, CALIBRATION, AND REGISTRATION

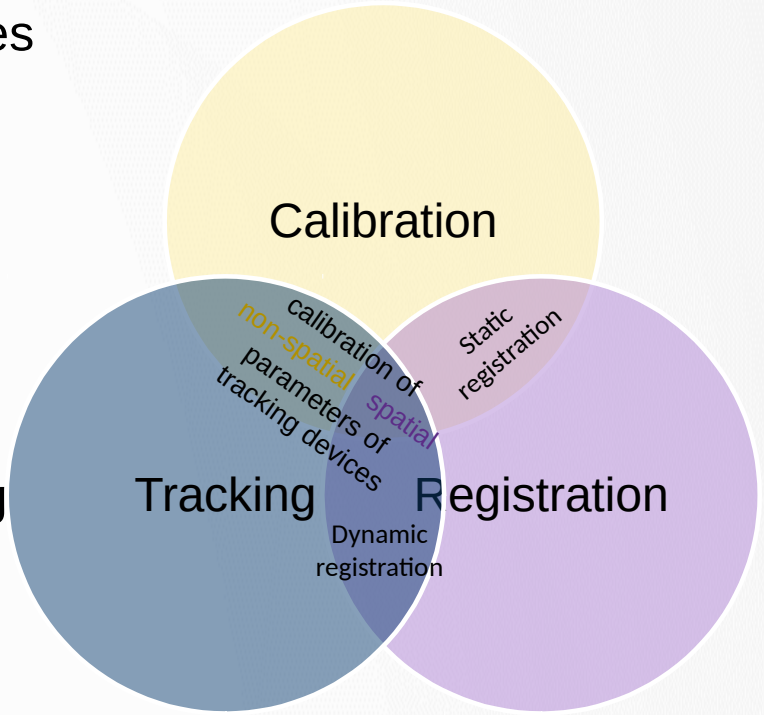
Registration = alignment of spatial properties

Calibration = offline adjustment of measurements

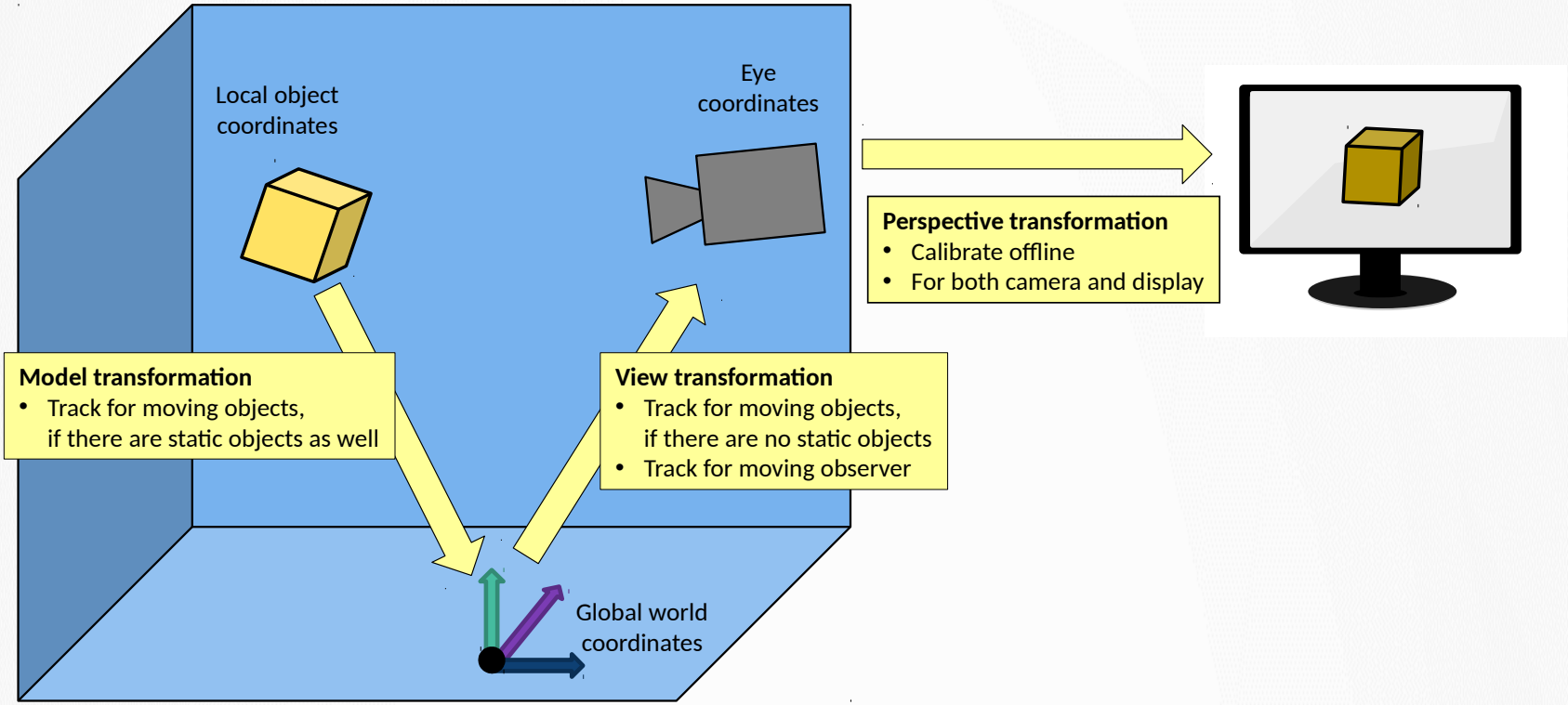
- Spatial calibration yields static registration
- Offline: once in lifetime or once at startup
- Alternative: autocalibration

Tracking = dynamic sensing and measuring of spatial properties

- Tracking yields dynamic registration
- Tracking in AR/VR always means “in 3D”!



COORDINATE SYSTEMS



FRAMES OF REFERENCE

Word-stabilized

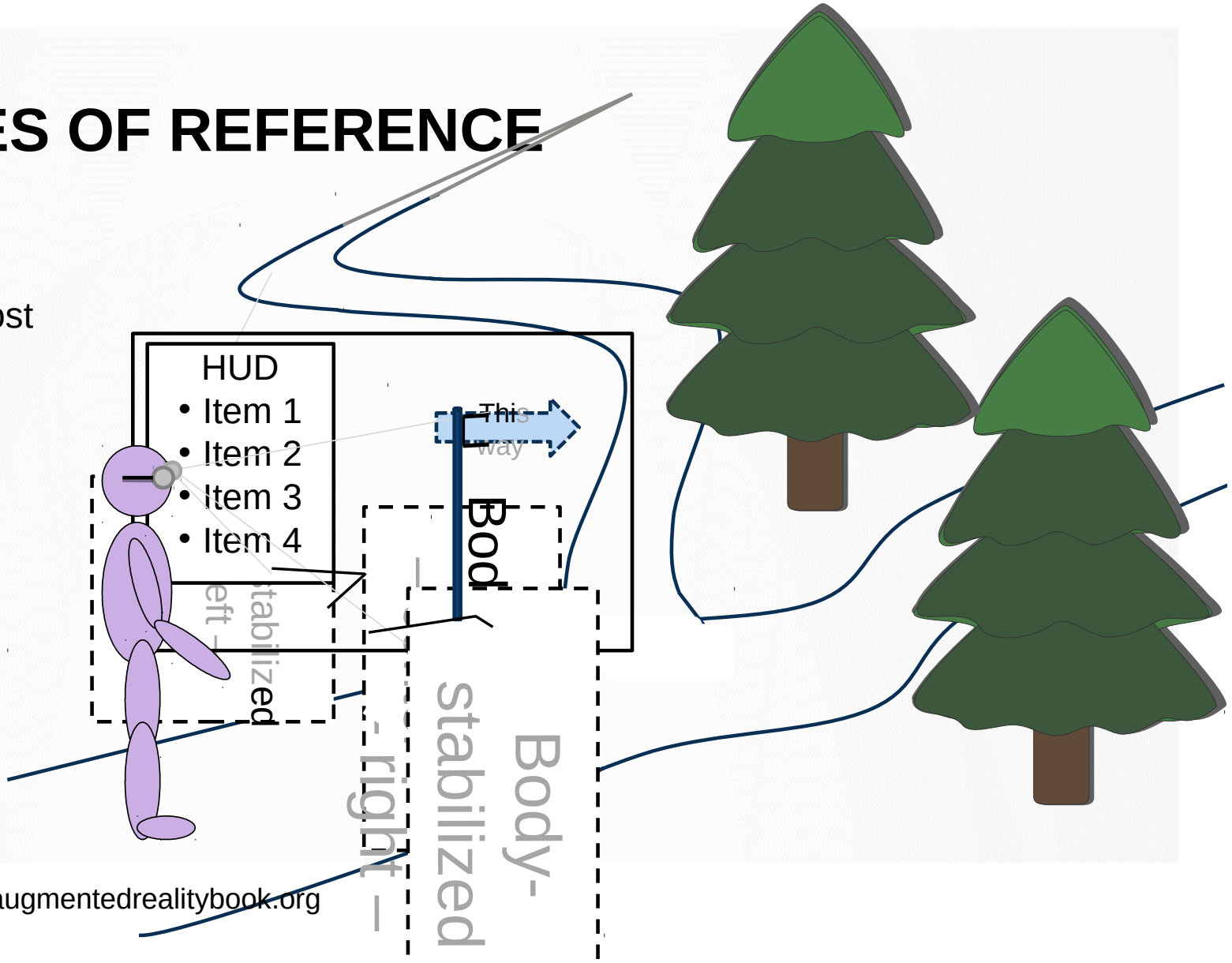
E.g., billboard or signpost

Body-stabilized

E.g., virtual tool-belt

Screen-stabilized

Heads-up display

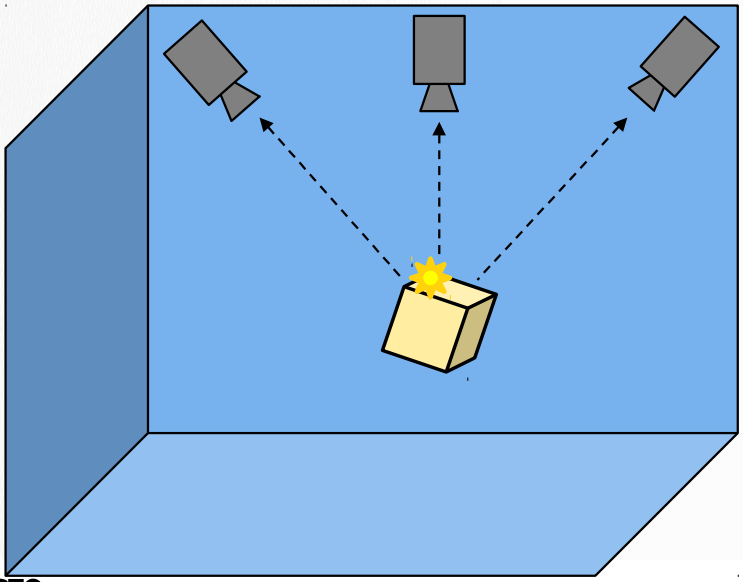


SENSOR GROUP ARRANGEMENT

Outside-in

Stationary mounted sensors

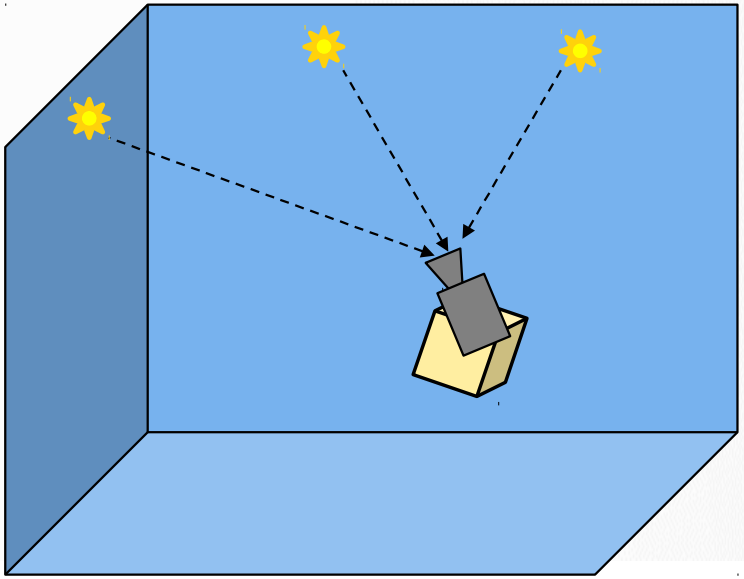
Good position, poor orientation



Inside-out

Mobile sensor(s)

Good orientation, poor position



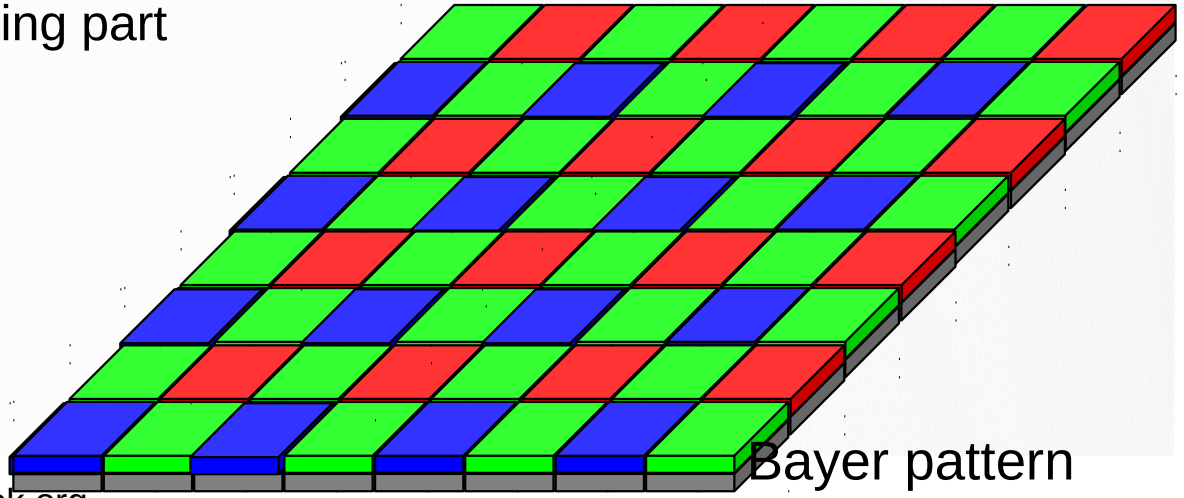
OPTICAL SENSORS

Digital cameras are cheap and powerful

- CCD (charge coupled devices) – professional photography
- CMOS (complementary metal oxide semiconductor) – fast and cheap

Computer vision techniques improve with Moore's law

Lenses are becoming the most limiting part



Bayer pattern

MODEL-BASED VERSUS MODEL-FREE TRACKING

Model-based

A tracking model representing the 3D world is available

Compare the model to observations in the images

Model-free

At start-up, no tracking model is available

Most build a temporary tracking model while tracking

Measurements only relative to starting point

LEAP MOTION

2 cameras, 3 infrared LEDs

Short-distance reflection
of the hands

VALVE/HTC VIVE

“Lighthouses” = two scanning infrared lasers

Photodiodes on head pick up lasers

MARKERS VS NATURAL FEATURES

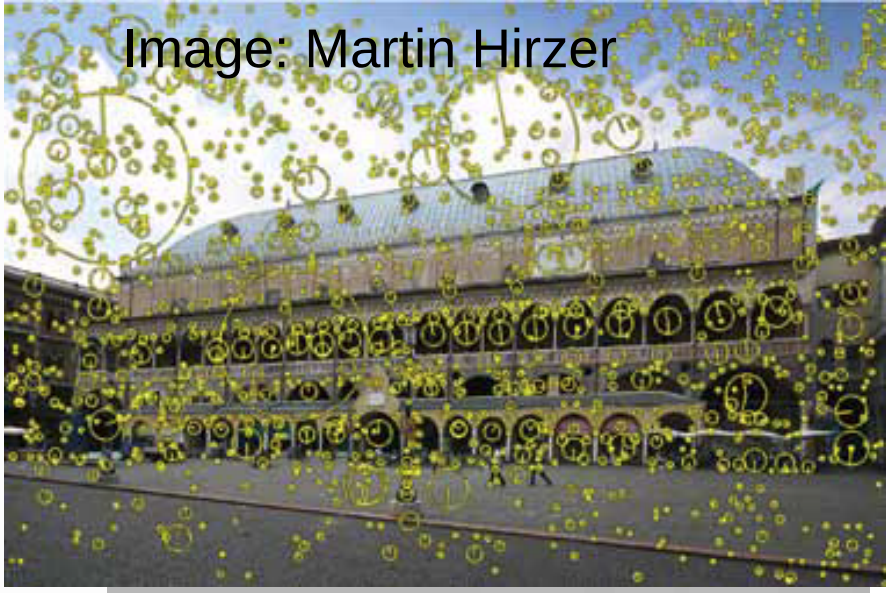
- Fiducials markers
  - Artificial tracking targets
  - Square shapes yield 4 points (enough for pose)
  - Circular shapes yield only 1 point
  - Digital marker model exists first,
    marker manufactured second (e.g., printing)
- Natural feature tracking
  - Existing visual features in the environment
  - Physical features exist first,
    tracking model reconstructed second

Image: Daniel Wagner

Image: Andrei State, UNC Chapel Hill

NATURAL FEATURES

- Detect salient interest points in image
  - Must be easily found
  - Location in image should remain stable when viewpoint changes
  - Requires textured surfaces
  - Alternative: can use edge features (less discriminative)
- Match interest points to tracking model database
  - Database filled with results of 3D reconstruction
  - Matching entire (sub-)images is too costly
  - Typically interest points are compiled into “descriptors”



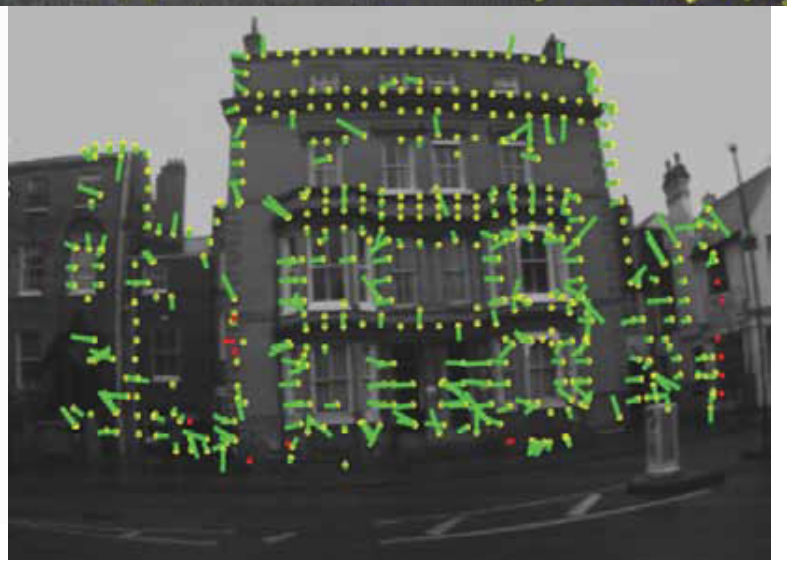


Image: Gerhard Reitmayr

MARKER TARGET IDENTIFICATION

- More targets or features ⇒ more easily confused
- Must be as unique as possible
- Square markers
  - 2D barcodes with error correction
  - E.g., 6x6=36 bits (2 orientation, 6-12 payload, rest for error correction)
  - Marker tapestries
- Spherical targets
  - 5 spheres in different geometric configurations
  - Can distinguish 10-20 targets
- Pulsed LEDs

Image: Mark Fiala

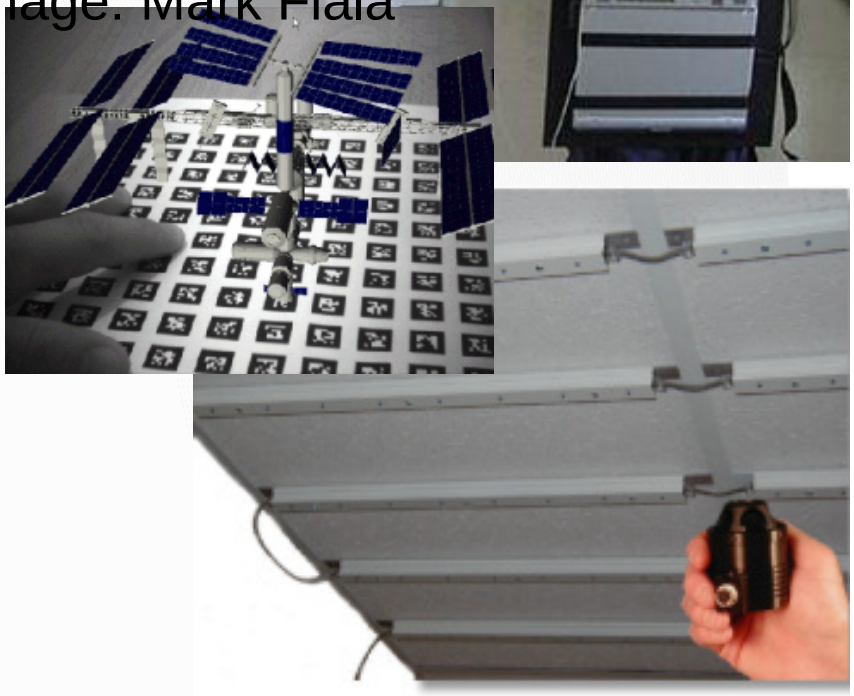


Image: Greg Welch, UNC Chapel Hill

NATURAL FEATURE TARGET IDENTIFICATION

- Individual natural interest points too easily confused
- Rely on co-occurency of interest points for detection
- Probabilistic search methods used to deal with errors
  - Vocabulary trees
  - Random sampling consensus

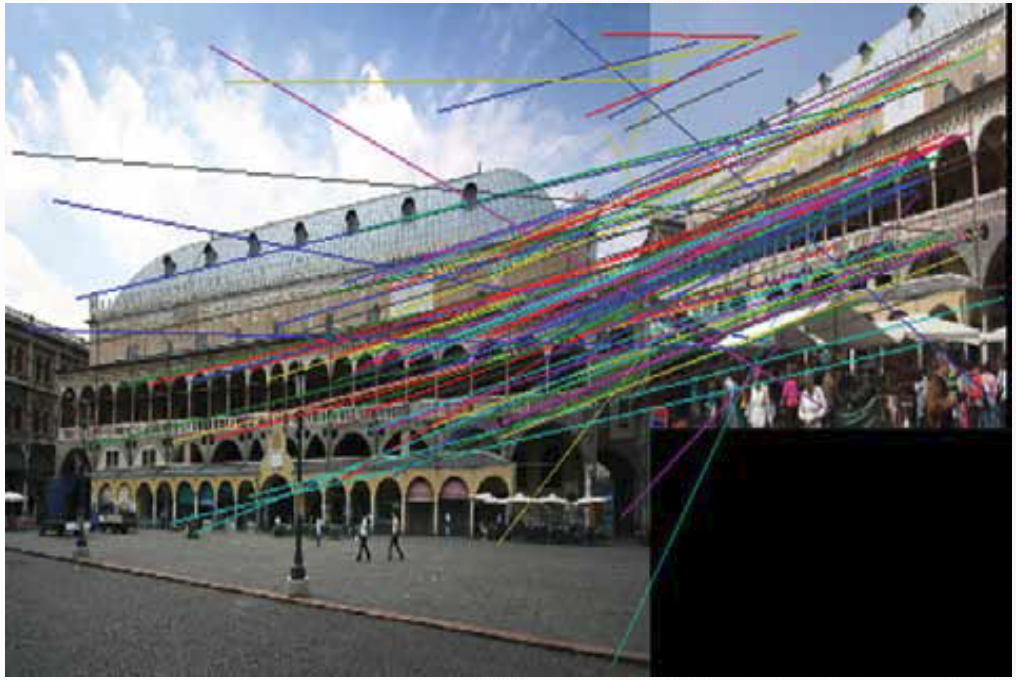


Image: Martin Hirzer

COOPERATIVE SENSOR FUSION

Primary sensor relies on information from secondary sensor to obtain its measurements

E.g., A-GPS combines celltower + GPS

Combination of inside-out + outside-in

Stereo cameras with known epipolar geometry

Non-overlapping cameras (e.g., 360°)

Indirect sensing (cont'd)

PointGrey LadyBu

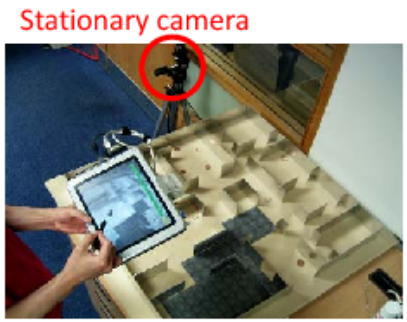
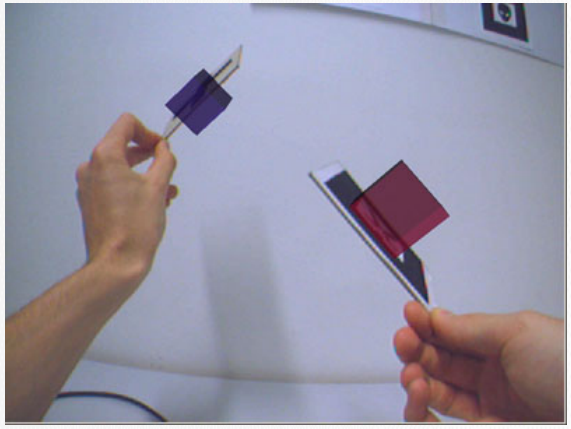


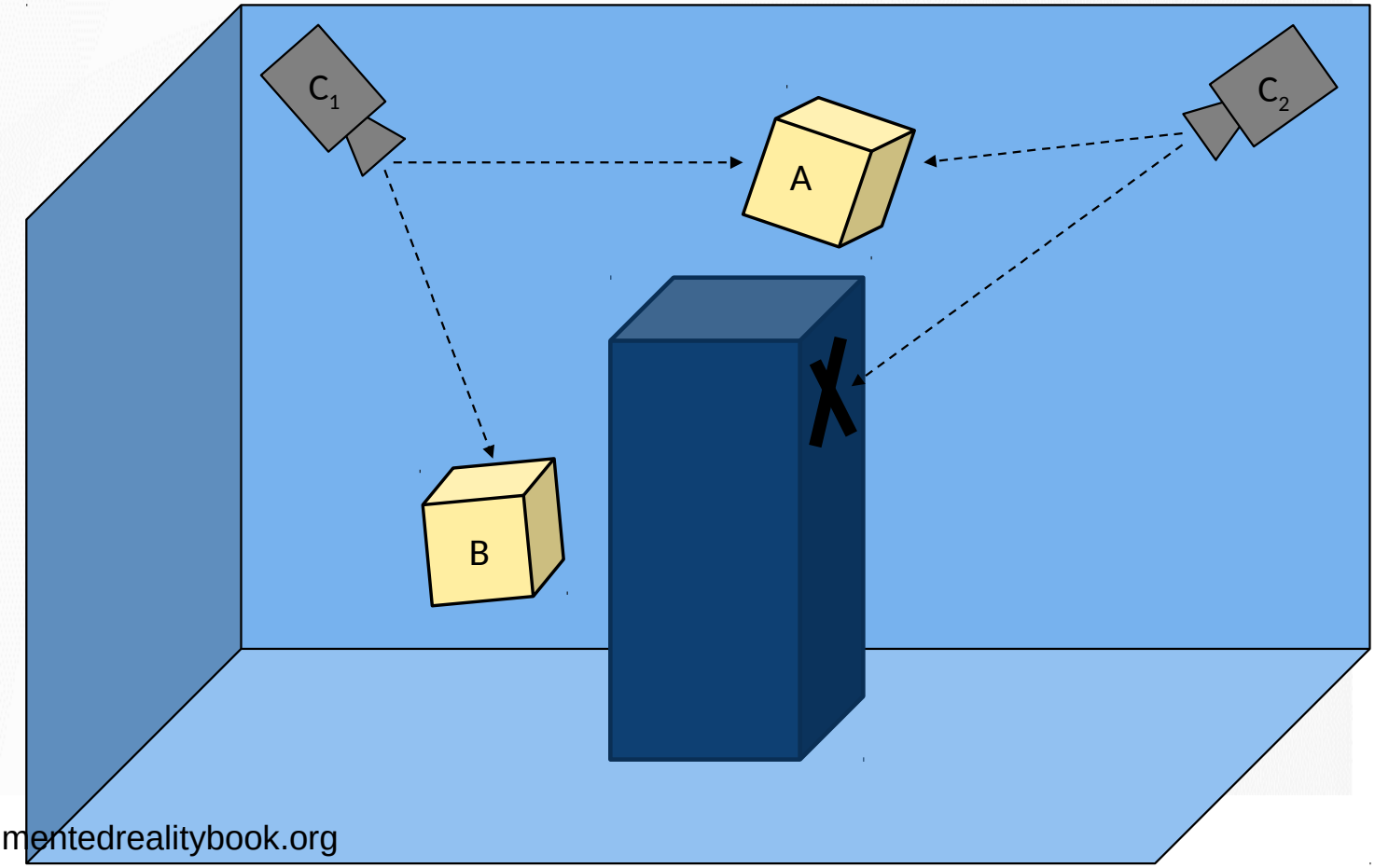
Image: Georg Klein

www.augmentedrealitybook.org

COOPERATIVE SENSOR FUSION FOR INDIRECT SENSING

"Track around the corner"





EXERCISE 6 AND LECTURE 14

- Lecture 14 will be a summary lecture, but as communicated at the first lecture it will be given by the students
- 4 groups, 4 papers, 15 minutes presentation + 5 minutes discussion each
  - Describe the topic and innovation of the paper
  - Map the paper to the computer vision principles you have learned during the course, what were the principles and why were they relevant in the method discussed in the paper
- You may use exercise 6 for preparing the slides and asking about the things you did not understand (in the paper or in lectures)
- Papers will be found from Wiki
- I'll name all students who have participate in the course into a group. In case you want to change the group, agree that straight with someone in another group who is willing to change places with you
- Agree straight with your group members how you'll arrange the work. In case you can't reach them via email please contact me

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI