

## TSBB21, Lecture 8

### Panorama Stitching. Mathematical tool: SVD

- Panorama Stitching
- Mathematical tool: SVD
- Literature
  - "Panorama Stitching. Supplementary Notes" by Per-Erik Forssén
  - Last in ... "Short about camera geometry and camera calibration" by Maria Magnusson
  - Parts of ... "Introduction to Representations and Estimation in Geometry" (IREG) by Klas Nordberg
  - Parts of ... "Mathematical Toolbox for Studies in Visual Computation at Linköping University" by Klas Nordberg

Maria Magnusson, CVL, Dept. of Electrical Engineering, Linköping University

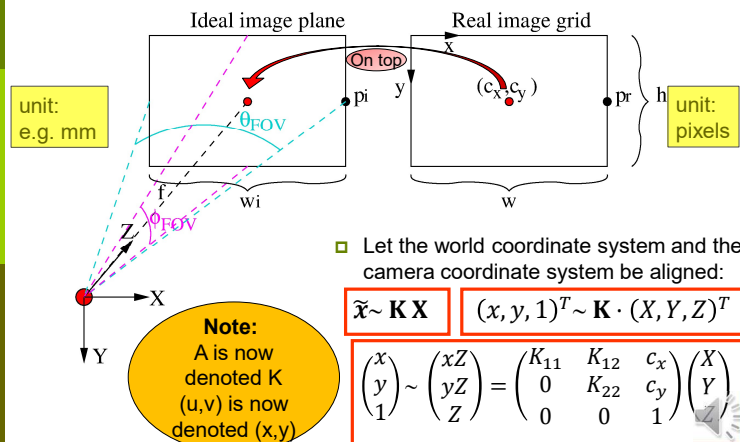
p. 1

## Panorama stitching



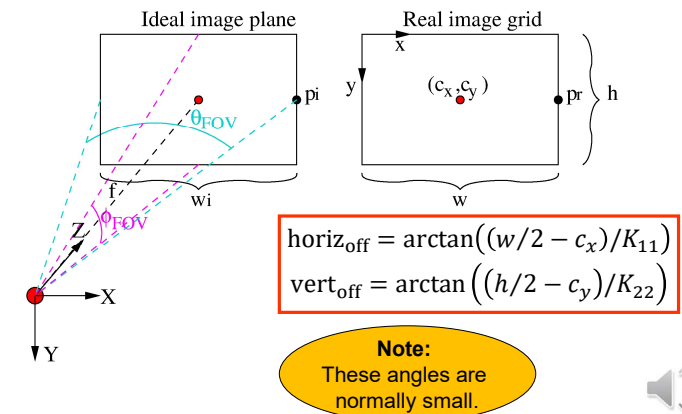
p. 2

## Determination of FOV and offset



p. 3

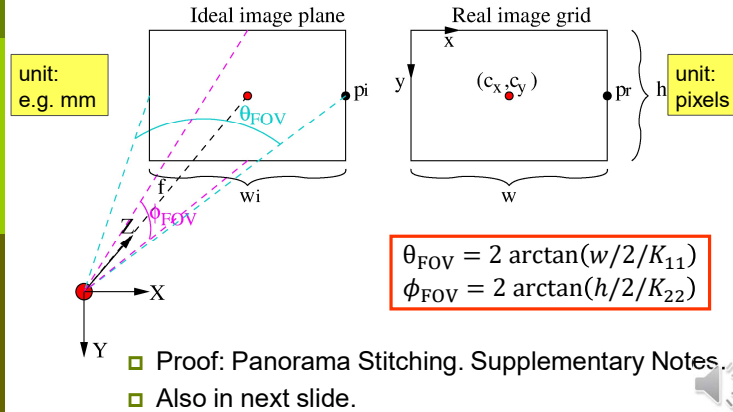
## Offset between optical centre and geometrical centre



p. 4

## Determination of Field Of View (FOV)

p. 5



## Determination of Field Of View (FOV), proof

p. 6

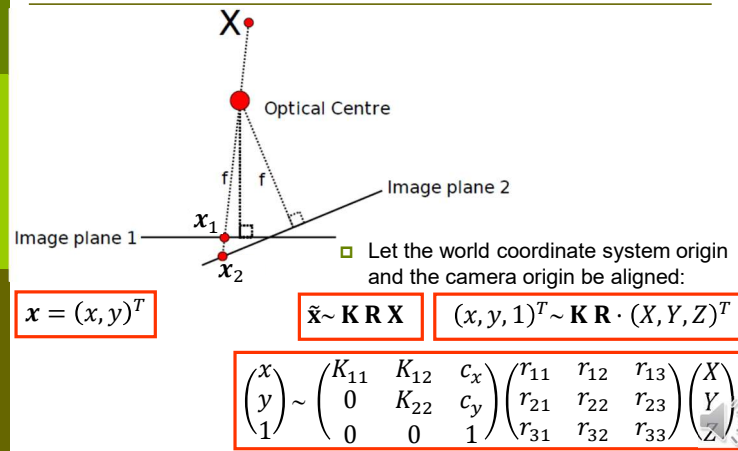
See figure 1. For simplicity, assume that  $\text{horiz}_{\text{off}} = \text{vert}_{\text{off}} = 0$ . Then the points  $p_i = (w_i/2, 0, f)^T$  and  $p_r = (c_x + w/2, c_y, 1)^T$ . Therefore

$$Z \begin{pmatrix} c_x + w/2 \\ c_y \\ 1 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & c_x \\ 0 & K_{22} & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_i/2 \\ 0 \\ f \end{pmatrix}$$

The first row gives  $Z(c_x + w/2) = K_{11}w_i/2 + c_x f$  and the third row gives  $Z = f$ . Therefore  $f(c_x + w/2) = K_{11}w_i/2 + c_x f$ , which gives  $w_i = wf/K_{11}$ . Finally,  $\theta_{FOV} = 2 \arctan(w_i/(2f)) = 2 \arctan(w/(2K_{11}))$ . Similarly,  $\phi_{FOV} = 2 \arctan(h/(2K_{22}))$ .

## Rotational Homographies

p. 7



## Rotational Homographies

p. 8

- The projections of a point  $X$  in the world to the points  $x_1$  and  $x_2$  in the two images:

$$\tilde{x}_1 \sim K R_1 X \quad \tilde{x}_2 \sim K R_2 X$$

- Assume the existence of a homography  $H_{21}$  that maps points from image2 to image1:

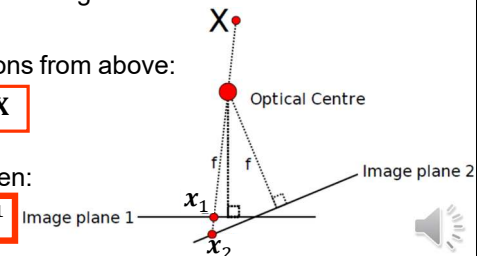
$$\tilde{x}_1 \sim H_{21} \tilde{x}_2$$

- Insert the expressions from above:

$$K R_1 X \sim H_{21} K R_2 X$$

- This is satisfied when:

$$H_{21} = K R_1 R_2^T K^{-1}$$



## Panorama stitching

- In panorama stitching, we have a set of images that share a common camera centre (origin), i.e. the images are all taken **from the same view-point** but in different directions.
- Given that the objects in the images are far away, the camera centres do not have to be exactly at the same point.
- Each image can be transformed into any other by a homography.



## Panorama stitching

- By applying the homography  $H_{21}$  to image2 (which is taken by camera2), it can be **stitched** onto image1 (which is taken by camera1).
- By applying the homography  $H_{31}$  to image3 (which is taken by camera3), it can be **stitched** onto image1 (which is taken by camera1).
- If both image2 and image3 are stitched onto image1, image1 works as a **reference image**.
- It is possible to stitch a whole set of images onto one reference image.



## Panorama stitching, example

- Two images taken from approximately the same view-point:

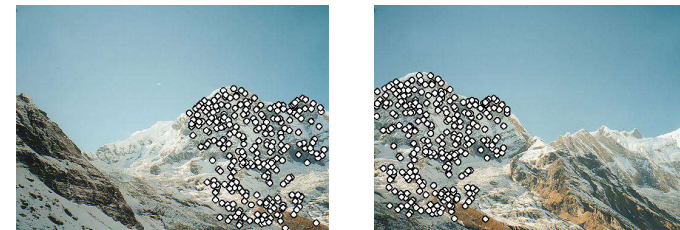


Images from: *Automatic Panoramic Image Stitching using Invariant Features*, IJCV 2007, Matthew Brown



## Panorama stitching, example

- Mark a set of corresponding points:



- From these points: Estimate a homography  $H$  that relates the 2 images.



## Panorama stitching, example

- The right image stitched onto the left image:



## Practical issues

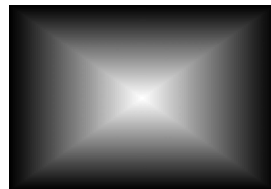
- The pixel values in overlapping regions may differ even if the geometric transformation is correct
  - Vignetting effects
  - Interpolation effects
  - Exposure time or illumination may be different in two images
  - Moving objects in the scene
- At each pixel:
  - Alternative 1: Take the value from only one of the two images
  - Alternative 2: Blend



## Blending weight

- For example, use a weight that is smaller at the edges of the image and larger at the center:

This is the weight image before the homography transformation



- Pano1 and Pano2 are the two images transformed to the reference grid.
- alpha1 and alpha2 are the weight image transformed to the reference grid.
- Normalized weighting: 
$$\text{Pano} = \frac{\alpha_1 \cdot \text{Pano}_1 + \alpha_2 \cdot \text{Pano}_2}{\alpha_1 + \alpha_2}$$



## Blending



With blending

Without blending



## Practical issues

- To assure a homography between any pair of images, we must have a pin-hole camera
  - No significant amount of lens distortion is allowed
  - Alternatively: lens distortion can be estimated and compensated for before the stitching
- In the panorama computer exercise, we use the following radial distortion model,

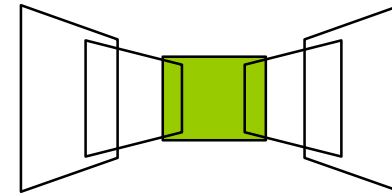
$$r_{\text{out}} = \frac{\arctan(r_{\text{in}} \cdot \gamma)}{\gamma}$$

- where the image is described in polar coordinates,  $(r, \theta)$ .
- You will manually estimate  $\gamma$  by undistorting the images using several different  $\gamma$ -values.  $\gamma$  is good when straight lines in the world give straight lines in the image.
- $\gamma$  is small, e.g.  $\gamma=0.001$



## Practical issues

- If the view direction between the reference image and the stitched image is very different, the 'resolution in' and the 'size of' the two images will vary a lot.
- Ex) 4 images stitched to a reference image (green):



- An attempt to stitch an image from 90° view direction onto an image from 0° view direction will result in infinity size of the stitched image.
- Solution: Map the images onto a cylinder or sphere and stitch them there instead.



## Mapping a point $(X,Y,Z)^T$ to the unit sphere

- A point  $(X,Y,Z)^T$  is projected to the normalized image plane  $(x_n, y_n, 1)^T$  and then transformed to a point  $(x, y, 1)^T$  on the real image grid as:

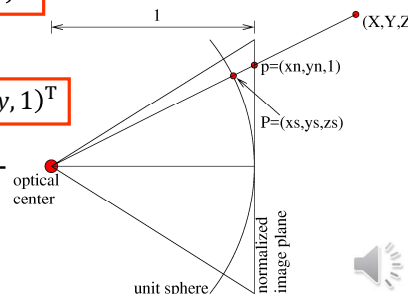
$$(x, y, 1)^T = \mathbf{K} \cdot (x_n, y_n, 1)^T$$

- Consequently:

$$(x_n, y_n, 1)^T = \mathbf{K}^{-1} \cdot (x, y, 1)^T$$

- Finally, a simple geometrical consideration gives:

$$(x_s, y_s, z_s)^T = \dots$$



## The Orthogonal Procrustes problem (OPP)

- If two corresponding sets of 3D points from two images are mapped to the unit sphere, it is possible to determine the rotation between them using OPP.
- Consider two sets of 3D points  $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$  and  $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_N)$  that are related as:  $\mathbf{X}_k = \mathbf{R}\mathbf{Y}_k + \epsilon_k$ ,
- where  $\mathbf{R}$  is an orthogonal matrix and  $\epsilon_k$  is an additive Gaussian noise term.
- It can be shown (see e.g. Panorama Stitching. Supplementary Notes) that if the SVD of  $\mathbf{XY}^T$  gives:  $\mathbf{XY}^T = \mathbf{UDV}^T$  Then:  $\mathbf{R} = \mathbf{UV}^T$

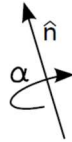


## Axis-Angle Representation

- When  $\mathbf{R}$  is determined, it is possible to find the rotation axis  $\hat{\mathbf{n}}$  and the rotation angle  $\alpha \in [0, \pi[$
- $\hat{\mathbf{n}}$  is an eigenvector of  $\mathbf{R}$  with eigenvalue 1:  

$$\hat{\mathbf{n}} = \mathbf{R} \hat{\mathbf{n}}$$
- The other two eigenvalues are:  

$$e^{i\alpha} \quad e^{-i\alpha}$$
- Use the Matlab command  $[\mathbf{V} \mathbf{D}] = \text{eig}(\mathbf{R})$ ;
- The eigenvectors are in  $\mathbf{V}$  and the eigenvalues are in  $\mathbf{D}$ .
- More information on this can be found in e.g. *Panorama Stitching. Supplementary...*



## Resampling to Spherical Coordinates

- Now we have computed different  $\mathbf{R}$  matrices for each image to be stitched. They can then be resampled to spherical coordinates:

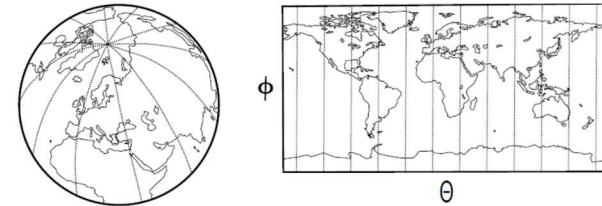


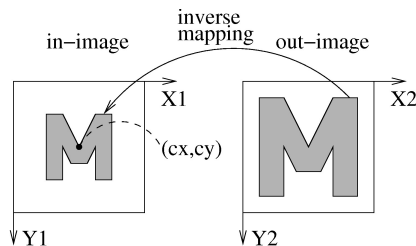
Fig. Illustration of spherical coordinates. Left: A world map painted on a sphere. Right: The same map in longitude-latitude space.



## Resampling and interpolation from in-image Im1 to out-image Im2

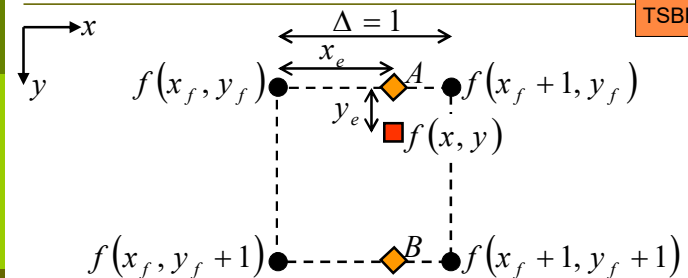
Repetition from TSBB08, TSBB31

- Suppose  $(X_2, Y_2)^T = \mathbf{T} \cdot (X_1, Y_1)^T$
- For all points  $(X_2, Y_2)$  in the out-image:
  - Perform an inverse mapping  $\text{inv}(\mathbf{T})$  to  $(X_1, Y_1)$  in the in-image.
  - Perform interpolation, e.g. bilinear interpolation, in the in-image, obtain a value.
  - Put the value at position  $(X_2, Y_2)$  in the out-image



## Bilinear interpolation

Repetition from TSBB08, TSBB31



$$\begin{cases} A = f(x_f, y_f) \cdot (1 - x_e) + f(x_f + 1, y_f) \cdot x_e \\ B = f(x_f, y_f + 1) \cdot (1 - x_e) + f(x_f + 1, y_f + 1) \cdot x_e \\ f(x, y) = A \cdot (1 - y_e) + B \cdot y_e \end{cases}$$

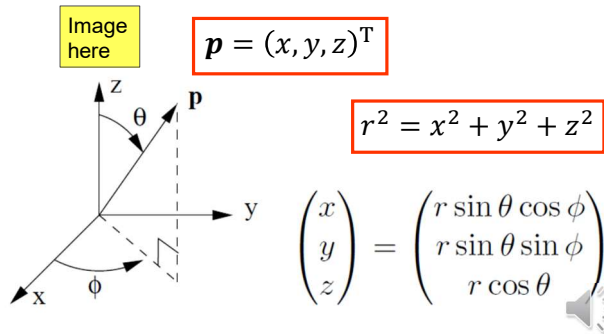




## Spherical coordinate system: standard form

p. 25

- Disadvantage: Singularities at  $\theta = \{0, \pi\}$

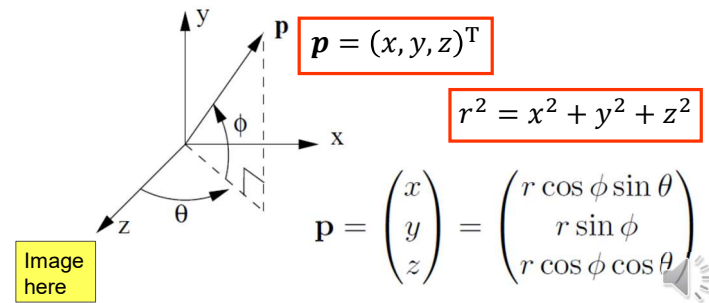


## Spherical coordinate system: longitude-latitude form

p. 26

- Singularities at  $\phi = \{-\frac{\pi}{2}, \frac{\pi}{2}\}$

i.e. above and below the camera, north and south pole.



## Recipe

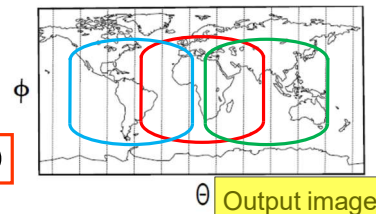
p. 27

- For all points  $p(\phi, \theta)$  in the output image:

- Transform it with

$$(sx, sy, s)^T = \tilde{x} \sim \mathbf{KR}p(\phi, \theta)$$

- Receive the position  $(x, y)$  in the input image.
- Perform bilinear interpolation in the input image, obtain a value.
- Put the value at position  $(\phi, \theta)$  in the output image



Note that, normally, there are several input images, e.g. three. The reference image (red) have the rotation  $\mathbf{R}=\mathbf{I}$ , the identity matrix. The other images (blue and green) have rotations  $\mathbf{R}=\mathbf{R}_1$  and  $\mathbf{R}=\mathbf{R}_2$ , relative to the reference image.

## Singular value decomposition (SVD)

p. 28

Theorem:

- For any  $N \times M$  real valued matrix  $\mathbf{X}$ ,
  - we can find an  $N \times N$  orthonormal matrix  $\mathbf{U}$
  - we can find an  $M \times M$  orthonormal matrix  $\mathbf{V}$
  - we can find an  $N \times M$  real diagonal matrix  $\mathbf{S}$
  - such that:

$$\mathbf{X} = \mathbf{USV}^T$$

- This is the **singular value decomposition (SVD)** of  $\mathbf{X}$

## Singular value decomposition (SVD)

p. 29

- $\mathbf{S}$  is  $N \times M$  diagonal (non-zero values only in the diagonal)
- The diagonal elements of  $\mathbf{S}$ ,  $\sigma_1, \dots, \sigma_P$ , are **real** and **non-negative** (with  $P = \min(N, M)$ )
- The diagonal elements of  $\mathbf{S}$  are the **singular values** of  $\mathbf{X}$
- The singular values are usually ordered such that  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_P$



## Singular value decomposition (SVD)

p. 30

- For singular value  $\sigma_k$ , the corresponding columns  $\mathbf{u}_k$  and  $\mathbf{v}_k$  of  $\mathbf{U}$  and  $\mathbf{V}$  are the **left and right singular vectors** of  $\mathbf{X}$ , respectively.

- Notice that

$$\begin{aligned}\mathbf{X}\mathbf{v}_k &= \sigma_k \mathbf{u}_k \\ \mathbf{X}^T \mathbf{u}_k &= \sigma_k \mathbf{v}_k\end{aligned}$$

- Remember that

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$



## Singular value decomposition (SVD)

p. 31

- In the case of a non-square matrix  $\mathbf{X}$  there will be some left (or right) singular vectors that neither have a corresponding singular value  $\sigma_k$  nor a right (or a left) singular vector.
- In this case they are simply said to have singular value 0 since, for example

$$\mathbf{X} \mathbf{v}_k = \mathbf{0} \quad (k > P, \text{ and } N < M)$$



## Solution of a homogeneous system of equations using SVD

p. 32

- Regard the following homogeneous system of equations:

$$\mathbf{X}\mathbf{b} = \mathbf{0}$$

- Perform SVD and rewrite:

$$\begin{aligned}\mathbf{U}\mathbf{S}\mathbf{V}^T \mathbf{b} &= \mathbf{0} \\ \mathbf{U}^T \mathbf{U} \mathbf{S} \mathbf{V}^T \mathbf{b} &= \mathbf{0} \\ \mathbf{S} \mathbf{V}^T \mathbf{b} &= \mathbf{0}\end{aligned}$$

- Solution:

$$\mathbf{b} = \mu \mathbf{v}_n$$

- where  $\mu$  is a scale factor and  $\mathbf{v}_n$  is the last column of  $\mathbf{V}$

- Matlab code:

```
[U, S, V] = svd(X);  
b = V(:, n);
```





## Example: Solve $\mathbf{X}\mathbf{b} = 0$

- ▣ Suppose that  $\mathbf{X}$  is a  $4 \times 3$ -matrix. Since  $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ , we have:

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ u_{41} & u_{42} & u_{43} & u_{44} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{pmatrix}$$

- ▣ Previous slide gave:  $\mathbf{S}\mathbf{V}^T\mathbf{b} = 0$

- ▣ Solution:  $\mathbf{b} = \mu\mathbf{v}_3 = \mu(v_{13}, v_{23}, v_{33})^T$  Insert:

$$\begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{pmatrix} \mu \begin{pmatrix} v_{13} \\ v_{23} \\ v_{33} \end{pmatrix} = \mu \begin{pmatrix} 0 \\ 0 \\ \sigma_3 \end{pmatrix}$$

- ▣ The smaller  $\sigma_3$ , the better solution and  $\sigma_3 = 0$  solves  $\mathbf{X}\mathbf{b} = 0$  perfectly.

