Image Mosaic (Panorama)

Lecturer: Sang Hwa Lee

What is the Mosaic ? (1)

➔ Stitch multi-images into a seamless composite

What is the Mosaic ? (2)

- Definition:
  - Synthesize or construct a large image of the same viewpoint from many images
- Image set (assumption)
  - Partial scenes of the whole one with small overlapping
  - Spatial or temporal sequence of the same scene
- Common projection surface
  - All images are projected a common plane
  - Transformation by perspective projection
  - Types of the plane
    - 2D plane
    - 3D cylinder, hemisphere

Image Mosaic Examples (1)

- ➔ 1-D panoramic view: far distance

Image Mosaic Examples (2)

- ➔ 2-D panoramic view (individual images)






Image Mosaic Examples (2)

➔ 2-D panoramic view (synthesized image)

Image Mosaic Examples (3)

- Panoramic view: 360^{o} rotation

Basic procedure of Mosaic (1)

- 1. Take image sequence from the same scene
  - Global alignment of the images spatially
- 2. Warp the image pair
  - Compute 2D/3D projective transformation of images
  - Reprojection onto mosaic surface (plane, cylinder, sphere)
- 3. Local alignment
  - Move warped images to overlap exactly
- 4. Stitching (optional)
  - Find boundary between overlapped images
- 5. Composition of the warped images
  - Blend the two together to create a mosaic

Global Alignment of Images

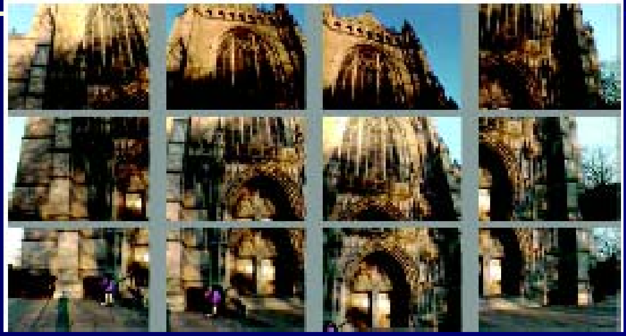


Image sequence

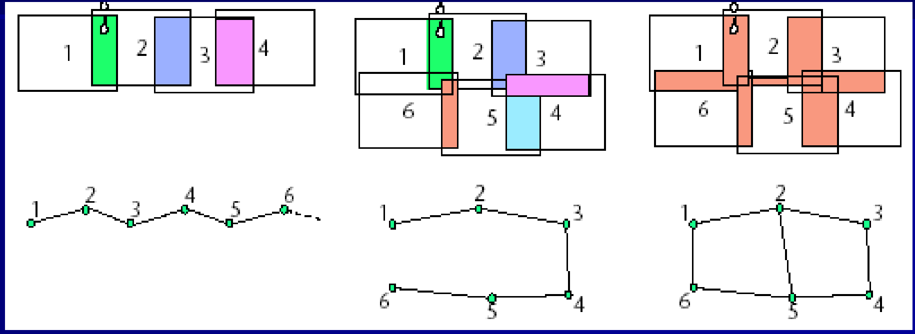
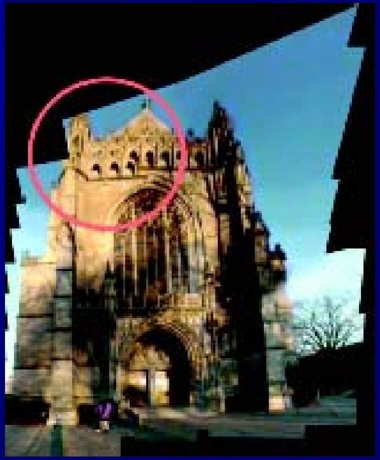
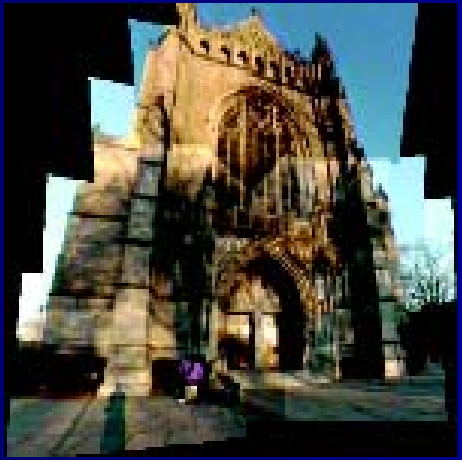


Image warping and local alignment



1D alignment

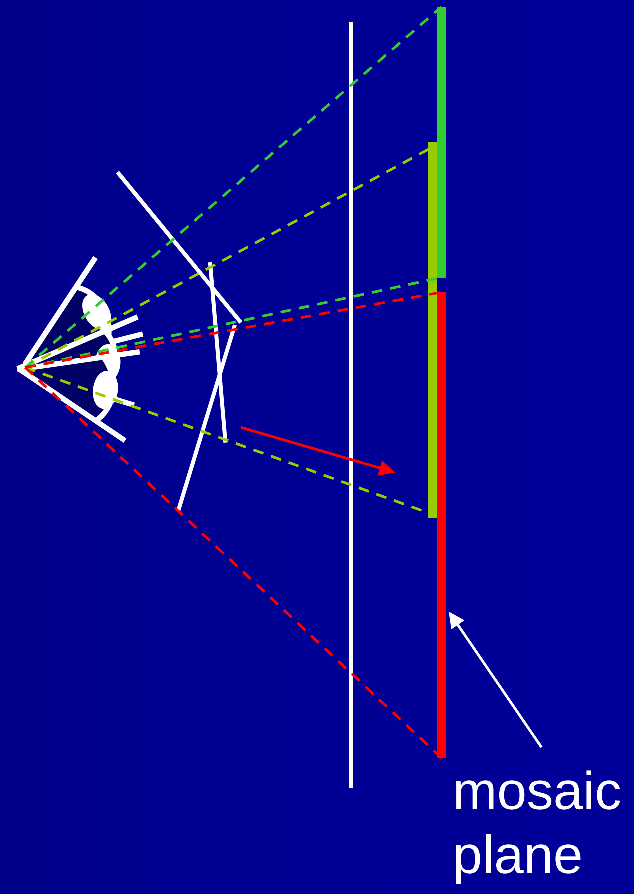


2D alignment

2D alignment and blending

Warping: Image Reprojection (1)

- The images are reprojected onto a common plane or surface
- Rather than thinking of this as a 3D reprojection, think of it as a 2D image warping from one image to another
  - No need of 3D depth information



Warping: Image Reprojection (2)

- Perspective projection of a plane
  - Lots of names for this:
    - homography, texture mapping, collineation, planar projective mapping
  - Modeled as a 2D warping using homogeneous coordinates
- p' = **Hp** (regular matrix multiplication)

$$\begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Local Alignment of Images

- Problems in alignment
  - Need in the image acquisition
    - Camera trajectory
    - Image order in the sequence
  - Overlap regions between image pairs
- Global alignment
  - Image-to-image alignment
- Local alignment
  - Corresponding points in overlap regions
  - Tiepoints (corresponding feature points)
    - Corner, SIFT
  - Most important for the mosaic performance

Stitching (1)

- Find the boundary between overlapped images
  - Assume the images are locally aligned
- Why do we use stitching as well as local alignment ?
  - Impossible to align the images exactly
  - Locally mismatches cause blurring in the overlapped regions
- Search for the most natural boundary on the overlapped regions
  - It requires blending at the boundary
  - How to do is an open problem!

Stitching (2)

- Dynamic Programming for stitching
  - Pixel by pixel difference (cost) map
  - Cost aggregation from the top rows
  - Search for the minimum cost path

…	3	6	8	10	…
…	5	2	7	9	…
…	5	6	1	13	…
…	8	7	5	6	…
…	12	10	9	11	…
…	12	13	11	12	…
…	11	10	9	10	…
…	9	6	7	8	…
…	9	7	8	8	…

…	3	6	8	10	…
…	8	5	13	17	…
…	10	11	6	26	…
…	18	13	11	12	…
…	25	21	20	22	…
…	33	33	31	32	…
…	44	41	40	41	…
…	50	46	47	48	…
…	55	53	54	55	…

…	3	6	8	10	…
…	8	5	13	17	…
…	10	11	6	26	…
…	18	13	11	12	…
…	25	21	20	22	…
…	33	33	31	32	…
…	44	41	40	41	…
…	50	46	47	48	…
…	55	53	54	55	…

Stitching (2)

Warped 2 images

Stitching (2)

Linear blending: pixel averaging (blurring)

Stitching (2)

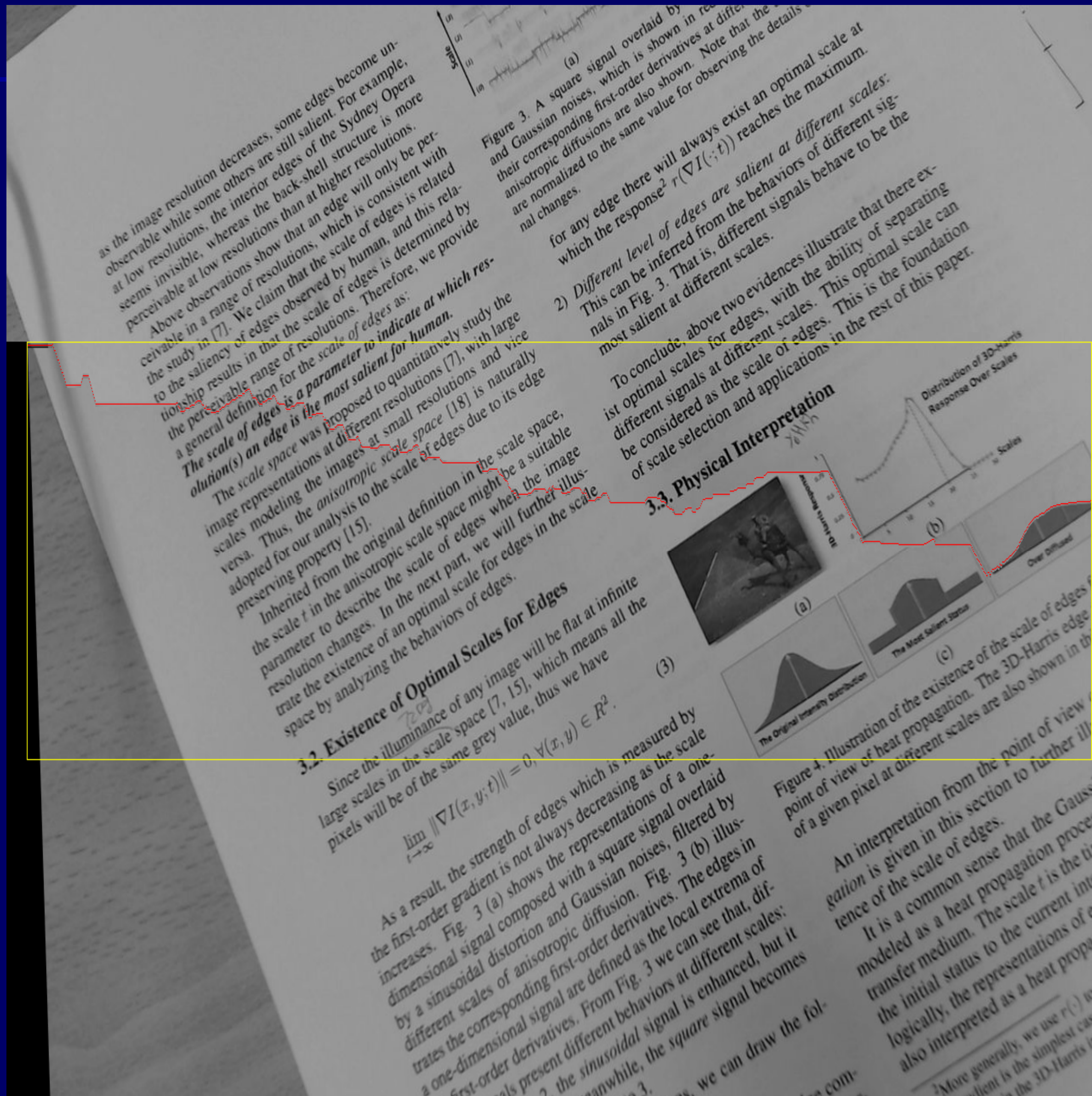
Boundary decision by stitching

Stitching (2)

Boundary decision and linear blending

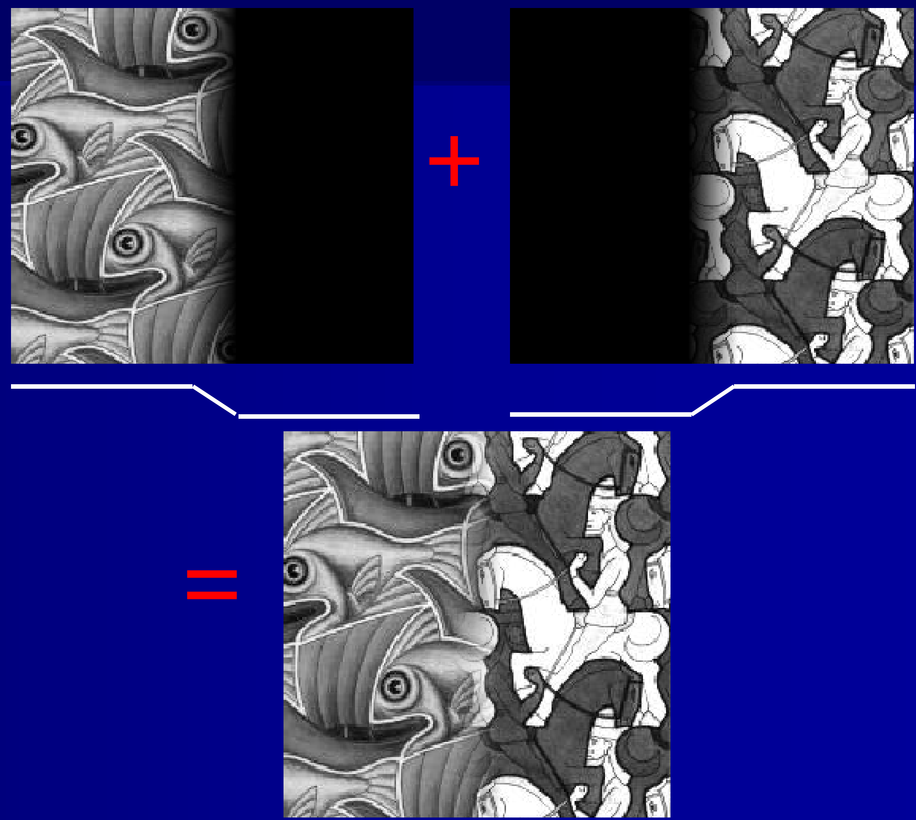
Stitching (3): Example

Stitching (4): Example



Blending (1)

- Composite the images to be natural
  - Compensation for Illumination/exposure differences
  - Global vs Local
- Blending with linear α map
  - At the boundary (local)
  - Transition width



$$I_{left}\,(x,y) = (\alpha R, \alpha G, \alpha B, \alpha)$$

$$I_{blend} = \alpha\, I_{left} + (1- \alpha) I_{right}$$

Blending (2): Example

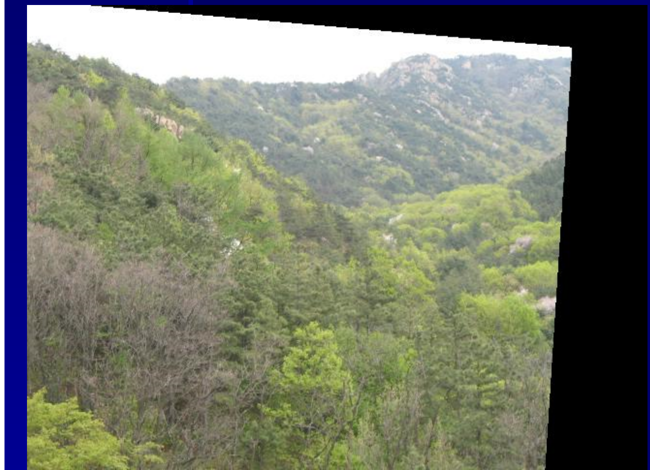
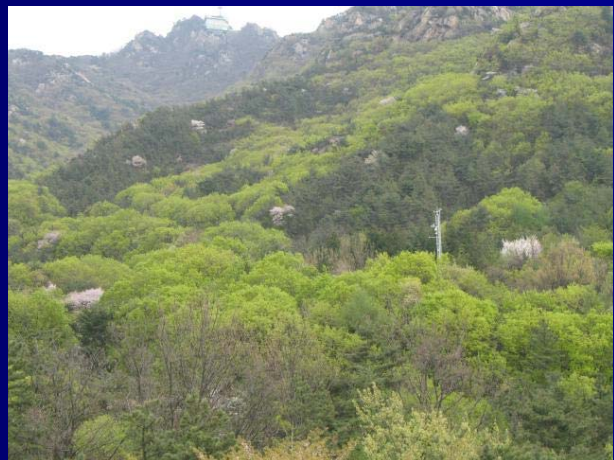
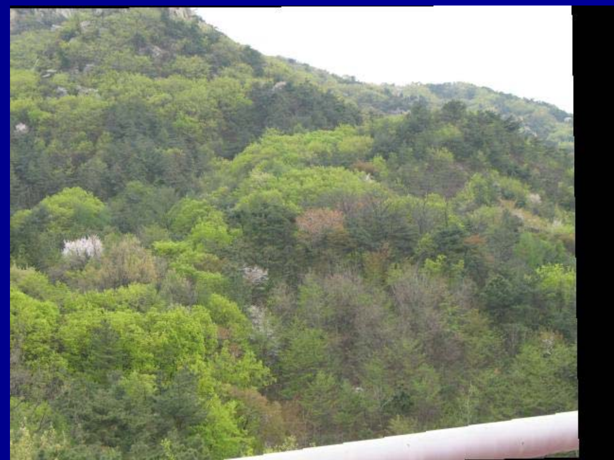
Stitching result by DP

Blending (3): Example

Linear blending with α map

An Experimental Overview (1)

1. 3 transformed data images

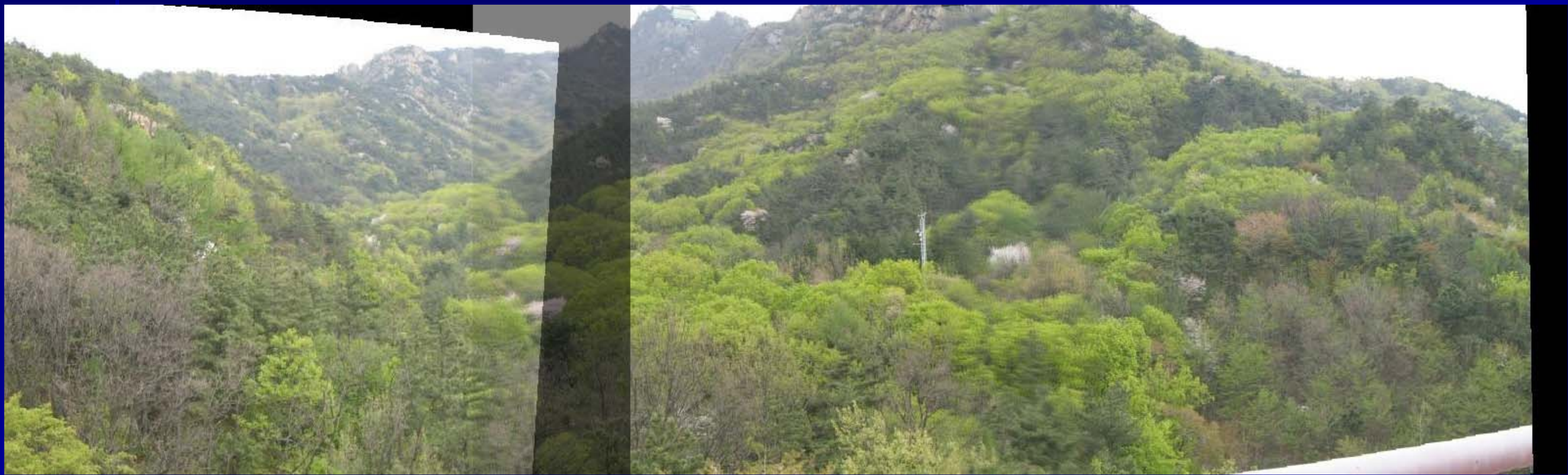




An Experimental Overview (2)

2. Transformed on the mosaic surface & local alignment

An Experimental Overview (3)

3. Blending by averaging



An Experimental Overview (4)

4. Stitched boundaries

An Experimental Overview (5)

5. Stitching result

An Experimental Overview (6)

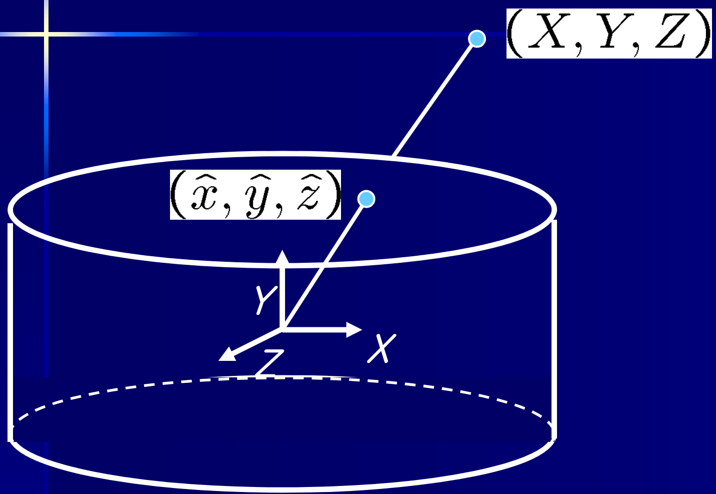
6. Linear blending with 16 pixels transition width

Curved Panoramic Mosaic

- Planar panoramic view
  - Wide field of view: up to 360°
  - Projection onto 2D plane
    - 2D to 2D homography
- Curved panoramic view
  - Curved mosaic surfaces
    - Cylinder
    - Hemisphere
  - More natural to human visual system

Mosaic Projection onto Cylinder Surface

Cylindrical Mosaic (1)



unit cylinder

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

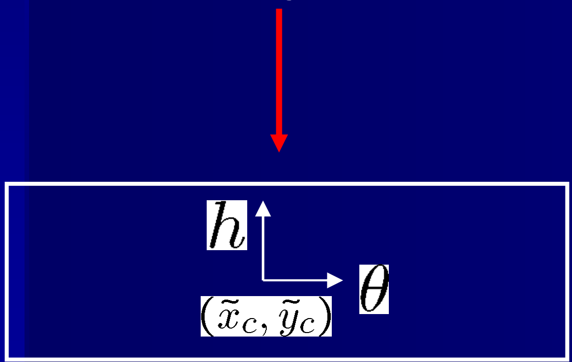
$$(\hat{x},\hat{y},\hat{z}) = \frac{1}{\sqrt{X^2+Z^2}}(X,Y,Z)$$

- Convert to cylindrical coordinates

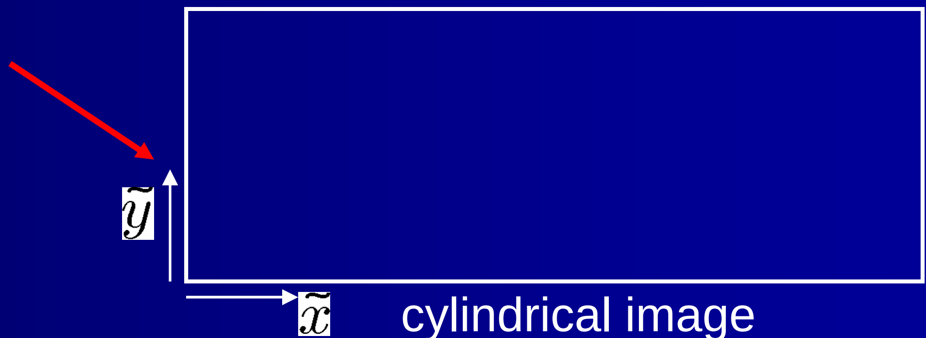
$$(sin\theta, h, cos\theta) = (\hat{x},\hat{y},\hat{z})$$

- Convert to cylindrical image coordinates

$$(\tilde{x},\tilde{y}) = (f\theta, fh) + (\tilde{x}_c,\tilde{y}_c)$$



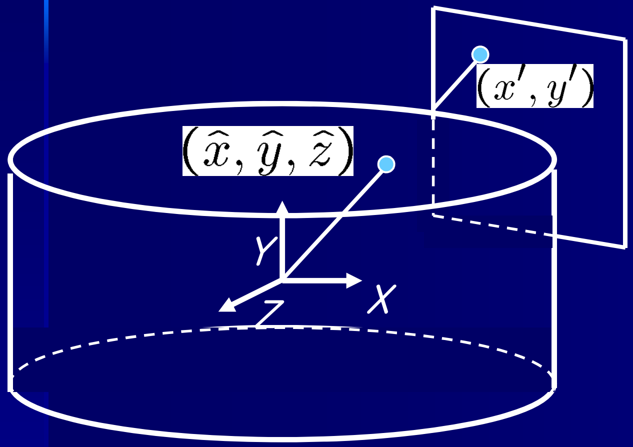
unwrapped cylinder



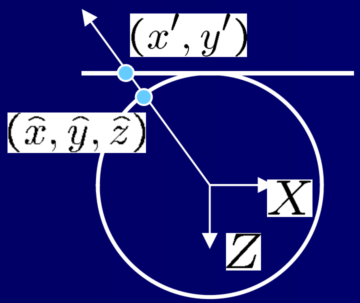
cylindrical image

Cylindrical Mosaic (2)

➔ How to map from a cylinder to a planar image?



side view



top-down view

- Apply camera projection matrix
  - focal length f
  - principle point is at center of image

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} -f & 0 & w/2 & 0 \\ 0 & -f & h/2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ 1 \end{bmatrix}$$

- Convert to image coordinates
  - divide by third coordinate (w)

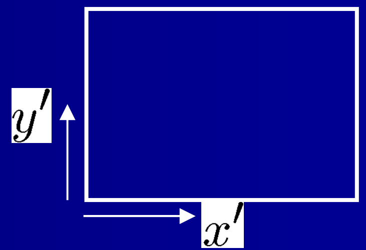


image coords

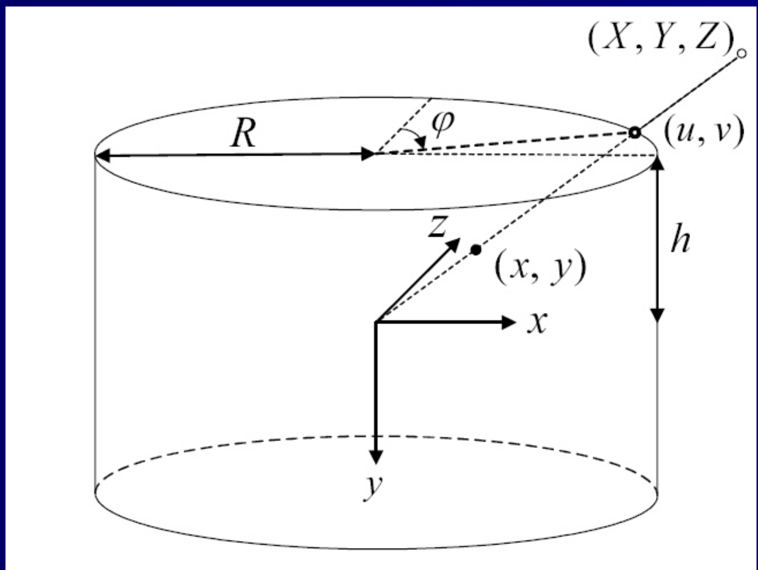
$(u, v) = (R\varphi, h)$

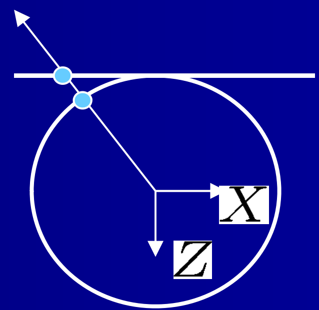
$$(x, y) = \left(f \frac{R \sin\varphi}{R\cos\varphi}, f \frac{h}{R\cos\varphi} \right) = \left(f \tan(u/R), f \frac{v}{R\cos(u/R)} \right).$$

Cylindrical Mosaic (3)

- Other formulation
  - S. J. Ha, et al, “Panorama mosaic optimization for mobile camera systems,” IEEE Trans. CE, pp. 1217-1225, Nov. 2007.

$$(x, y) = \left(f \frac{R \sin\varphi}{R\cos\varphi}, f \frac{h}{R\cos\varphi} \right) = \left(f \tan(u/R), f \frac{v}{R\cos(u/R)} \right)$$





top-down view

Cylindrical Mosaic (4)

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

Image 384x300

f = 180 (pixels)

f = 280

f = 380

Cylindrical Mosaic (5)

- Non-linear distortions
  - Straight lines are curved in the case of close scene
  - How to control the focal length and cylinder radius

Spherical Mosaic (1)

- Mosaic surface: Hemisphere

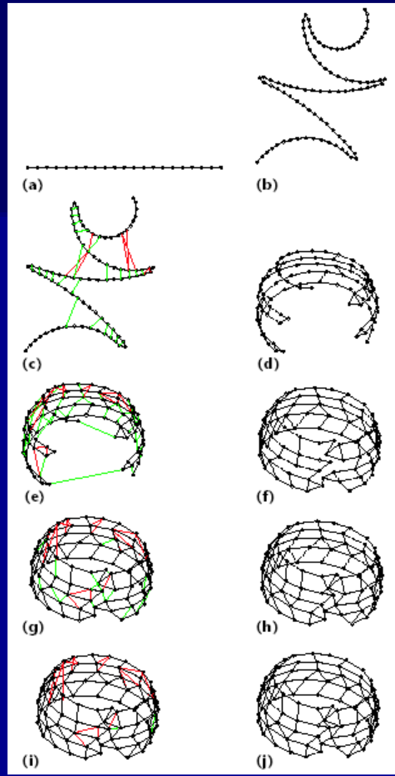
Image sequence

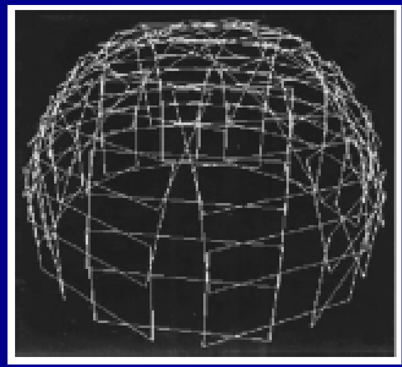
Spherical panorama

Mercator view

Spherical Mosaic (2)

- Image sequence by multiple rotational camera
- Immersive image rendering
  - Virtual reality
  - Closed space modeling
- Similar approach to cylindrical panorama
  - Spherical coordinate
  - Projection onto spherical surface





Spherical Mosaic (3)

➔ Results of spherical mosaic

Spherical Mosaic Systems

- Photosynth (by MS)
  - PC based
  - http://photosynth.net/
- Photo Sphere (By Google)
  - Mobile device based
  - https://www.youtube.com/watch?feature=player_embedded&v=0poff-mHQ4Q

Spherical Panorama System (1) - Overview

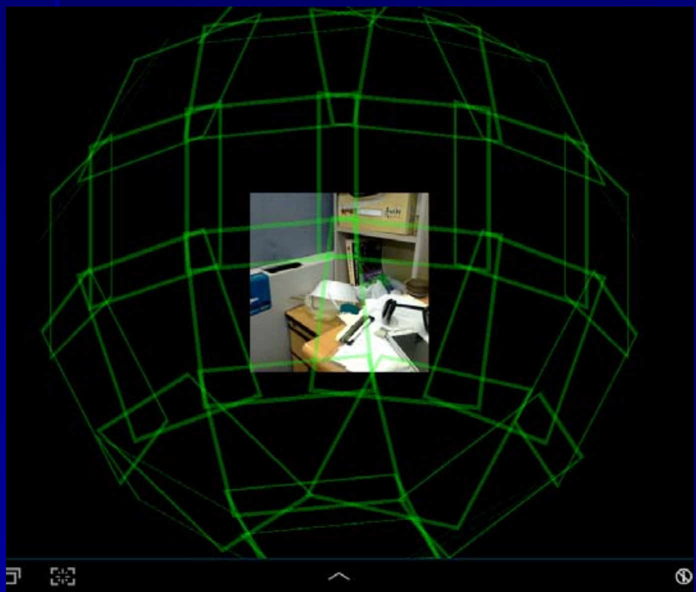
- User Interface for image acquisition
- Exposure compensation
- Image alignment
- Spherical mapping
- Blending

Spherical Panorama System (2)
- A Result

Spherical Panorama System (3) - User Interface

➔ Guide for image acquisition of 360 world

- Wire frame + IMU sensor
- Divide 3-D circumstance into subregions



Spherical Panorama System (4) - Exposure compensation

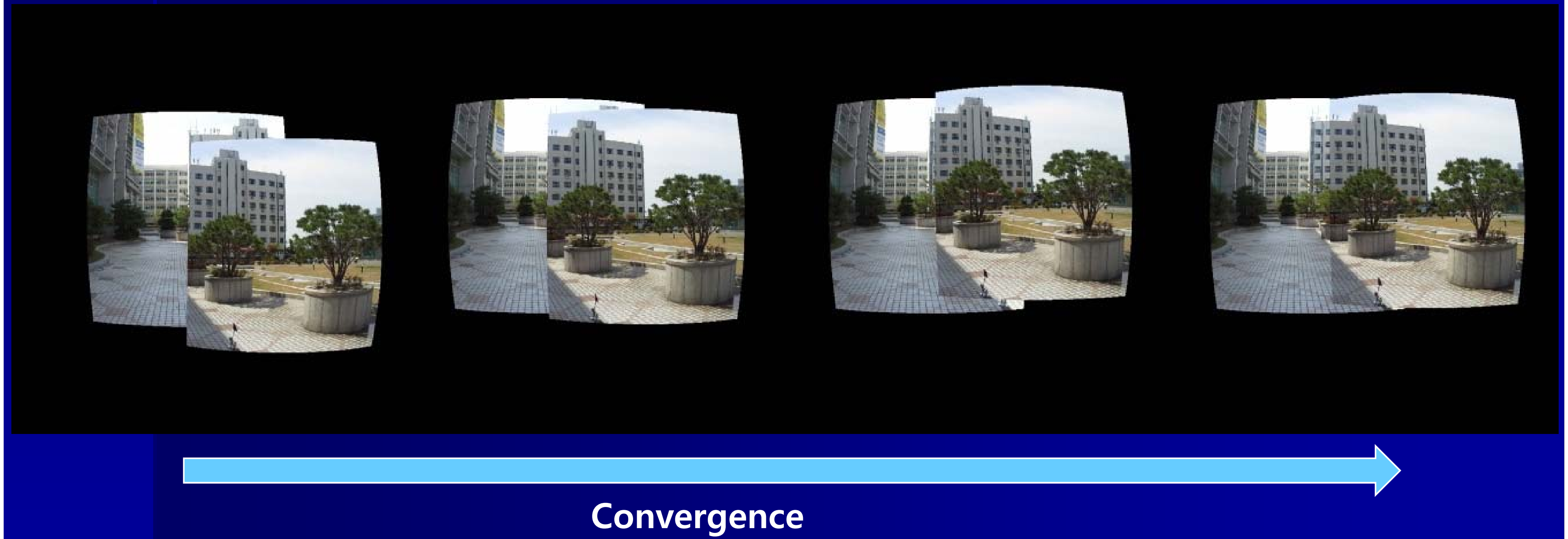
➔ Different brightness compensation

- Using the mean and variance in the overlapped region



Spherical Panorama System (5) - Local alignment

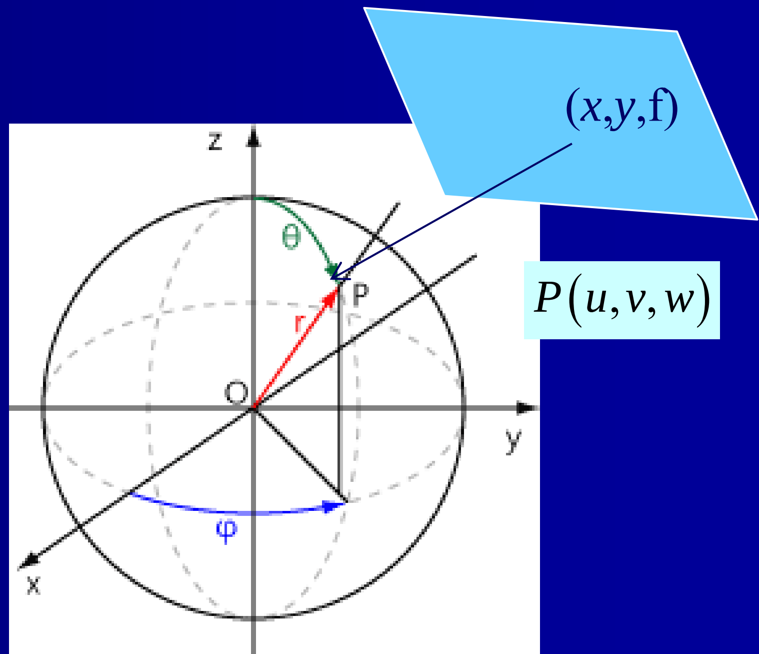
➔ Realignment of images on the spherical surface
  - Refining angles of spherical coordinate



Spherical Panorama System (6) - Spherical Mapping

- ➔ Reprojection onto spherical surface
  - ➢ Center of sphere = optical projection center
  - ➢ Each image should be rotated w.r.t the captured direction.

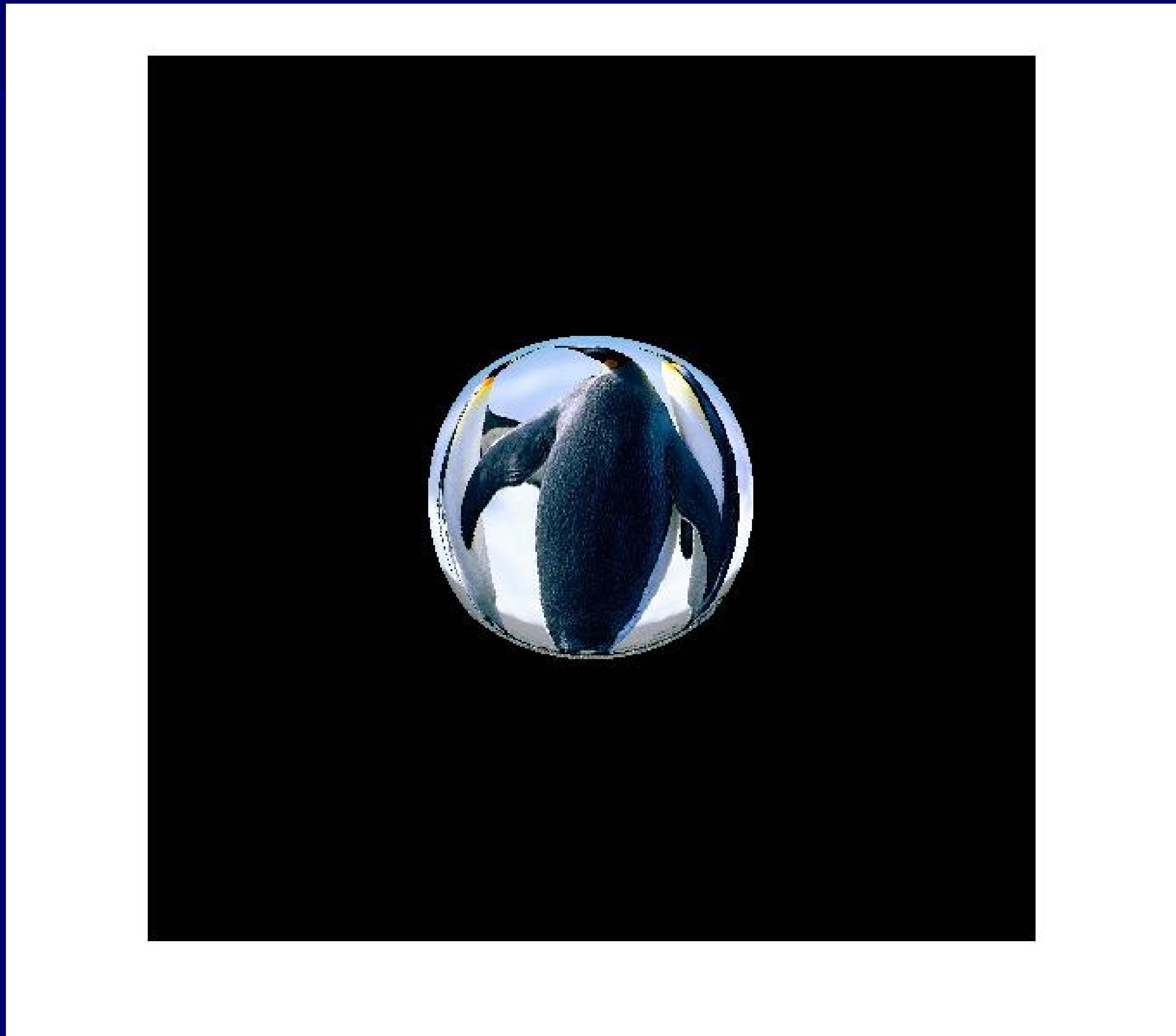
$$
\begin{aligned}
(u, v, w) &= \frac{r}{\sqrt{x^2 + y^2 + f^2}}(x, y, f) \\
&= (r\cos\theta\cos\varphi, r\cos\theta\sin\varphi, r\sin\theta)
\end{aligned}
$$



Result of Spherical Projection (1)

- R=100
- F=100

원영상



Result of Spherical Projection (2)

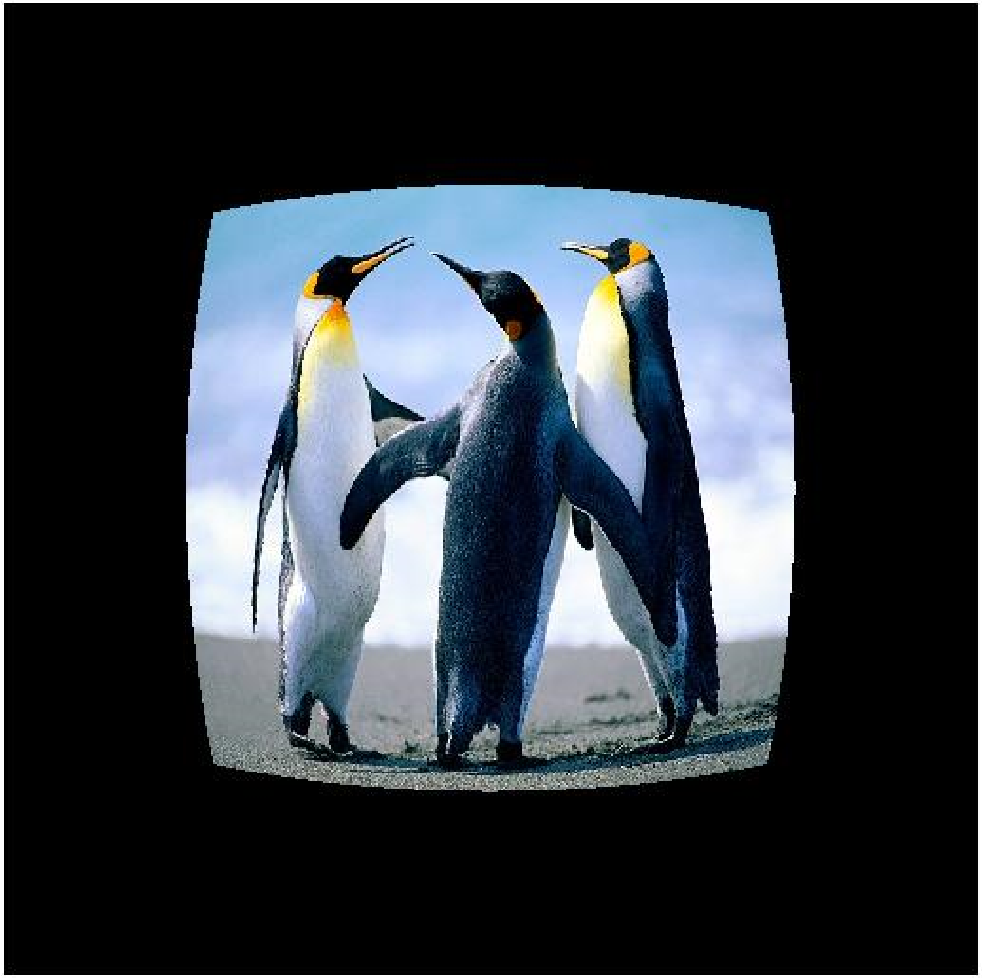
- R=100
- F=200

원영상

Result of Spherical Projection (3)

- R=350
- F=500

원영상

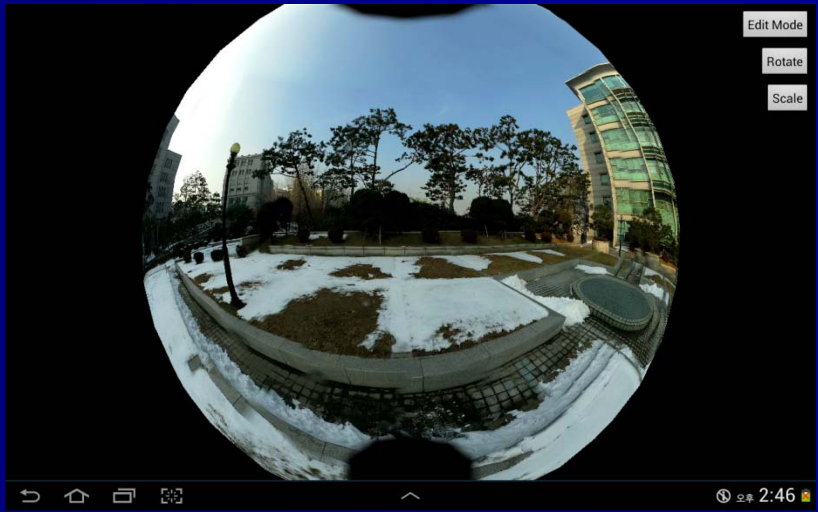


Result of Spherical Projection (4)

- R=500
- F=700

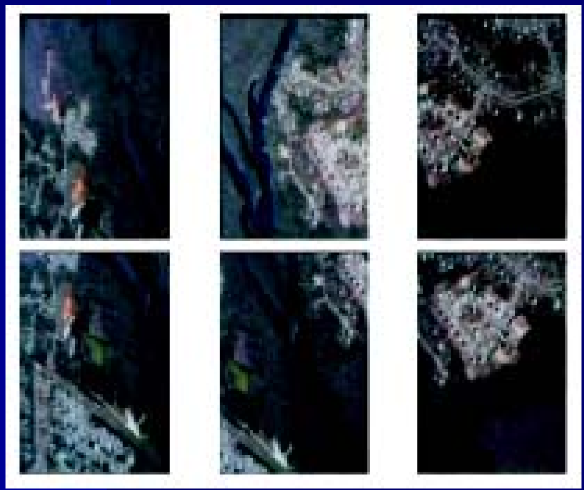
원영상

Spherical Panorama Results



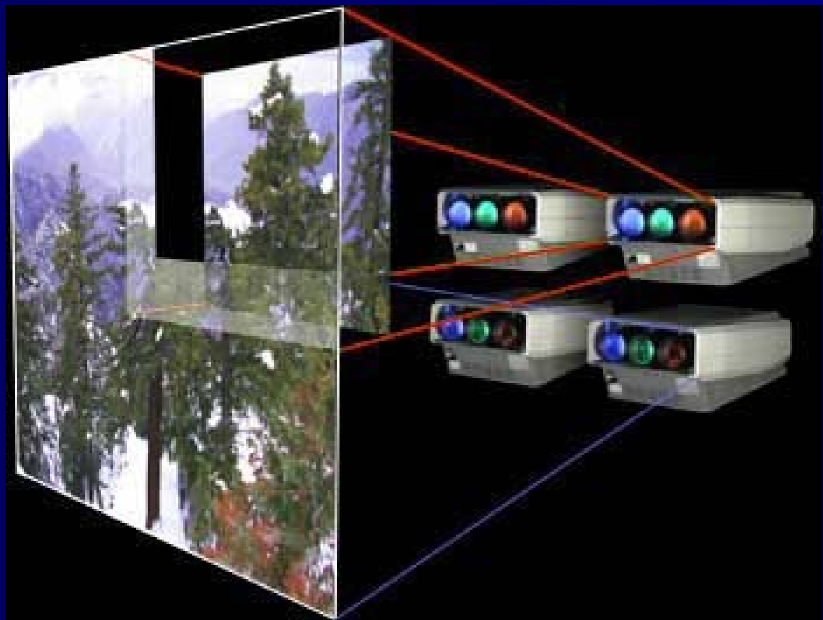
Applications of the Mosaic (1)

- Geographic map
  - Aerial Image sequence
  - Geographical information system



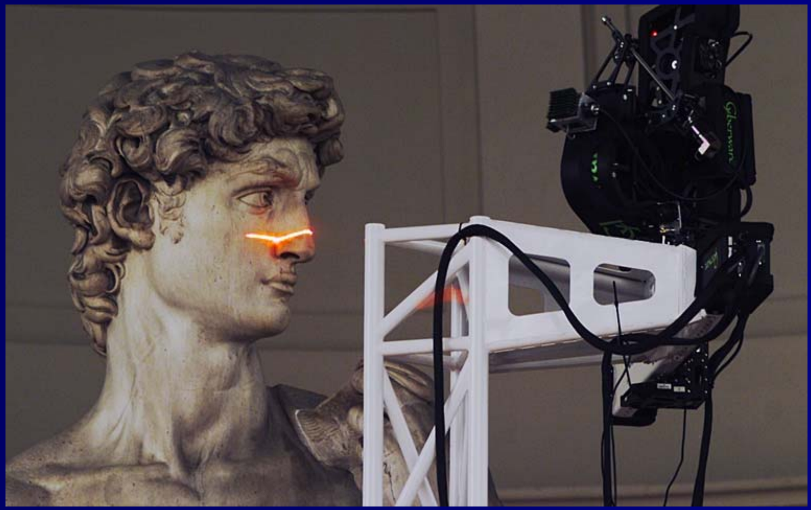
Applications of the Mosaic (2)

- Virtual reality
  - Construct an environment for the navigation
  - Large display
    - Camera-projector system



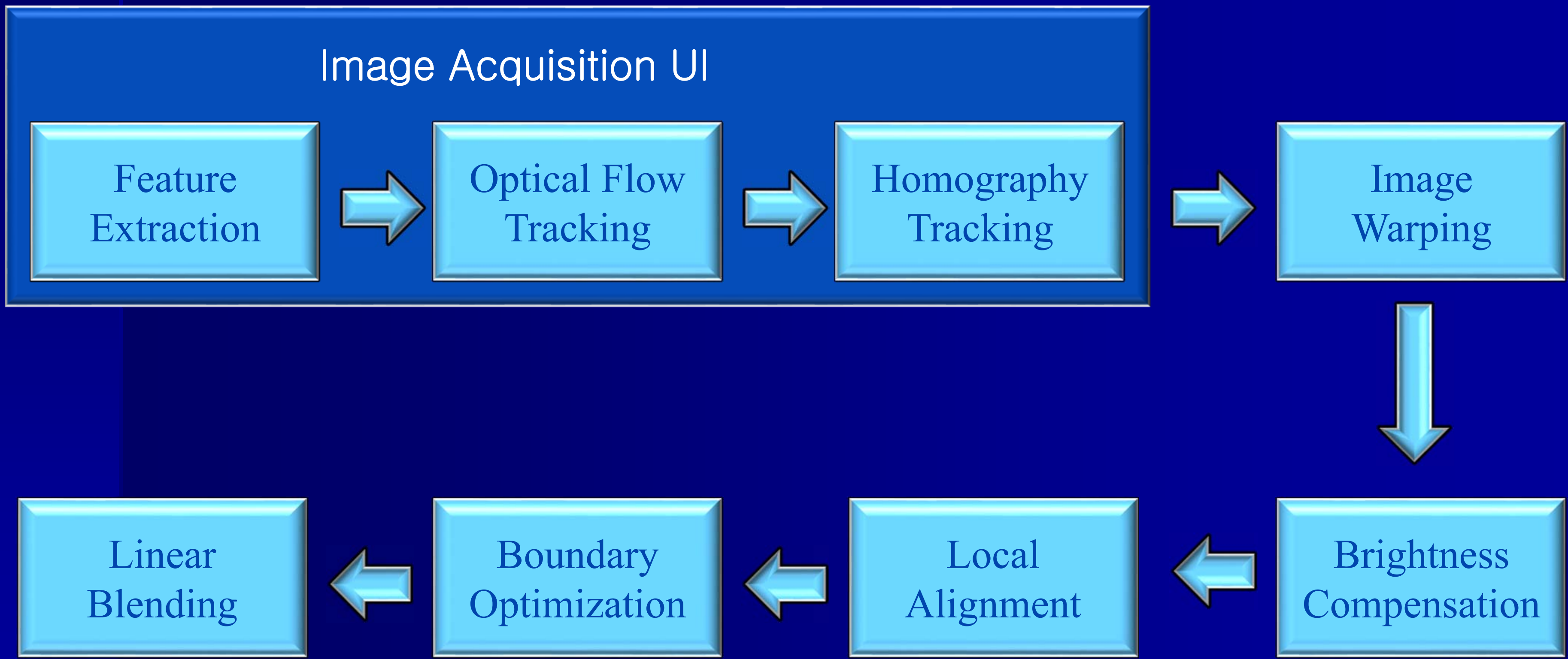
Applications of the Mosaic (3)

- Registration of 3D shapes
  - Construct a full 3D model of the object and scene
  - 3D modeling
  - Use of depth information and 3D projection



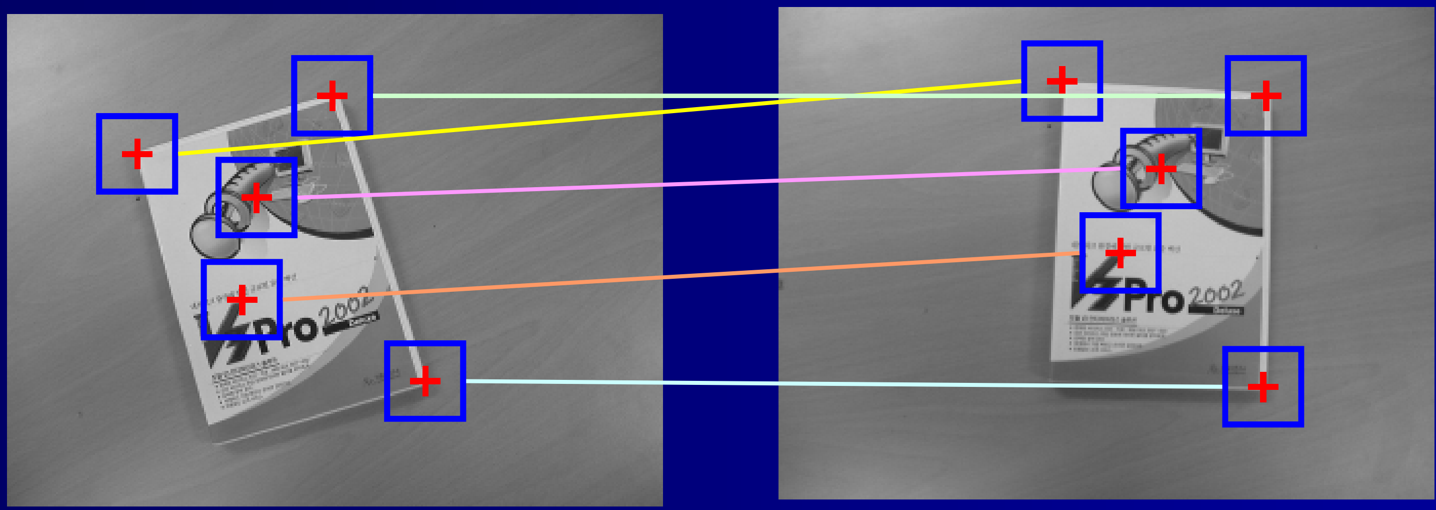
Another Issues in Mosaic (1)

- Mobile camera application



Another Issues in Mosaic (2)

- Features extraction and matching
  - Keypoints to determine transformations of images
  - They must be robust and stable
  - BMA, corners, edges
  - **SIFT, SURF, FAST corner**



Current Issues in Mosaic (2)

- Lens distortion
  - Geometric distortion
    - Straight lines are curved due to projection onto cylindrical surface
  - Radiometric distortion
    - Vignette
    - Related to exposure compensation

Current Issues in Mosaic (2)

Bottom: Anti-vignette

Current Issues in Mosaic (3)

- Bundle Adjustment
  - Find the optimal mosaic surface globally
  - Align and stitch images with rotation, focal length of best matching image

Current Issues in Mosaic (4)

- Real-time operation
  - Using Sensors, and specific UI
  - AR viewer
- Fast feature matching and tracking