360 Panoramas

Luiz Velho
IMPA

Outlook

- Panoramic Surfaces / Parametrizations
- 360 Image Formats
- Omnidirectional Awareness

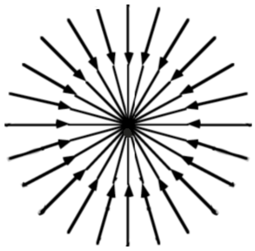
Panoramic Surfaces

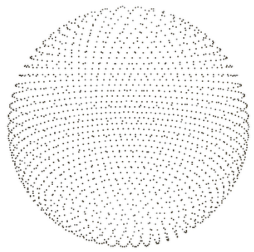
(AKA Plenoptic Surfaces)

Omnindirectional Image

The Set of <u>All Rays</u> incident at a point (x,y,z)

- Spherical Light Field   = 360 degrees

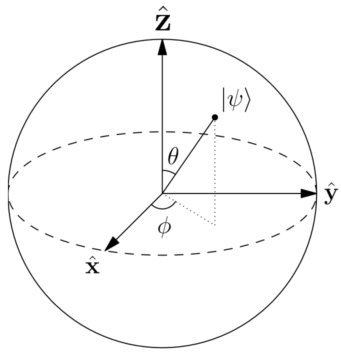




Canonical Representation

Spherical Representation

- Represent a point on the unit sphere with:
  - $\theta \in [0,2\pi)$: **azimuth** (longitude)
  - $\phi \in [0,\pi]$: **inclination** (colatitude, from north pole)



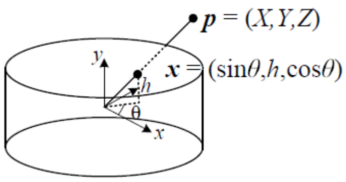
- Define Cartesian coordinates from spherical:
  - $(x, y, z) = (\sin\phi\cos\theta, \sin\phi\sin\theta, \cos\phi)$

Panoramic Surfaces

Generalized Support for Visual Information

- Data Representation

- example: ***Cilindrical Panorama***

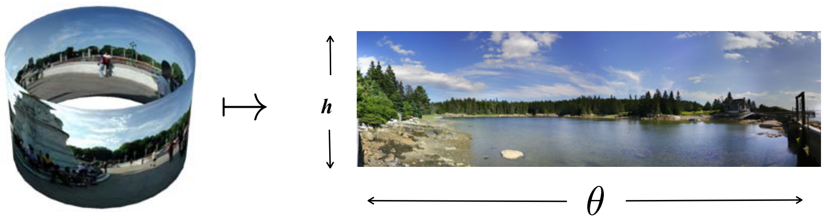


Parametrizations

Maps 2D Surface to Planar Domain

- Coordinate Systems

- example: ***Cilindrical Mapping***



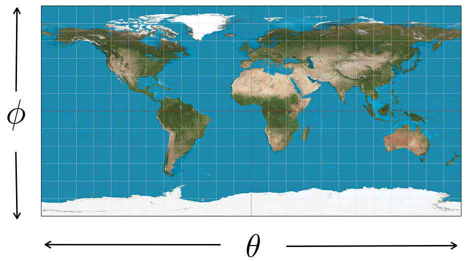
360 Image Formats

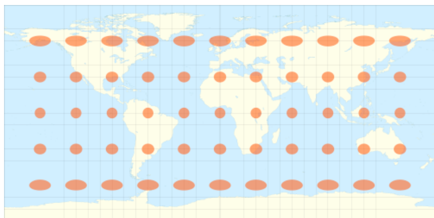
360° Image Formats

- Parametrizations of the Sphere
  - Lat-Long
  - Cube Map
  - Mirror Ball
  - Azimuthal
  - Stereographic

Equirectangular

- Latitude-Longitude Mapping (e.g., *Flickr)*





natural coordinate system ***distortion toward poles***

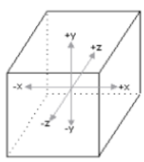
Most Convenient Format

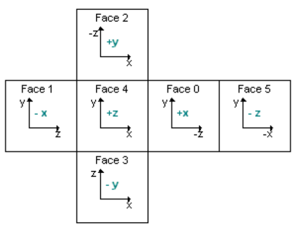
Equirectangular Projection (Latitude–Longitude)

- Parametrization
  - $(\theta, \phi) \in [0,2\pi] \times [0,\pi]$
- Pros:
  - Common in 360° photography / video
  - Simple mapping to/from sphere
- Cons:
  - Large distortion near poles (especially vertical stretching)

Cube Mapping

- 6 Perspective Projections





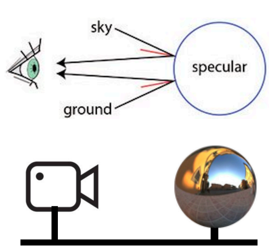
suitable for CG rendering

Cube Map Projection

- Sphere is projected onto the six faces of a cube
  - mapped from a corresponding direction: ±X, ±Y, ±Z.
- Pros:
  - Lower distortion compared to equirectangular
  - Efficient GPU rendering and sampling
- Cons:
  - Requires special handling for edges/corners
  - Discontinuous across face boundaries

Mirrored Ball

- Reflection Mapping



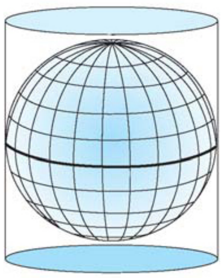
HDR Light Maps

Projection of Mirror Sphere

- Parametrization (*orthographic projection of reflective sphere onto a plane*)
  - For each pixel (x, y) inside the unit disk, the viewing direction is obtained from reflection on a virtual mirror ball
- Pros:
  - Matches certain real-world acquisition methods (e.g., mirrored spheres)
  - Direct relation to environment lighting
- Cons
  - Severe compression near center
  - Unused image corners (outside the circle)

Azimuthal Projection

- Hemispherical Mapping



Dome Master standard **- Used in fulldome theaters and planetariums**

Dome Master / Fisheye Projection

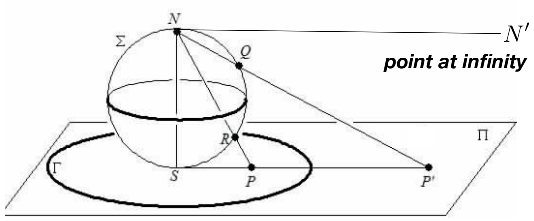
- Parametrization
  - $(r, \theta), \quad r = f(\phi)$
  - Dome Master format : $r = R \cdot \phi$

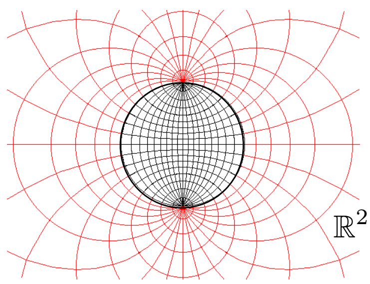
- Pros:
  - Natural for hemispherical displays
  - High angular accuracy in center

- Cons:
  - Non-uniform resolution
  - Needs remapping for VR/video

Stereographic

- Conformal Mapping (preserves angles)





singularity ***infinite plane***

Stereographic Projection

- Parametrization
  - Projects points from the sphere onto a plane from a fixed point (i.e., N / S pole)

$$(x, y) = \frac{(X, Y)}{1 + Z}, \qquad (X, Y, Z) \in \mathbb{S}^2$$

- Pros:
  - Conformal (angle-preserving)
  - Useful in visualization and computations
- Cons:
  - Not area-preserving
  - Can't capture the full sphere in a bounded region

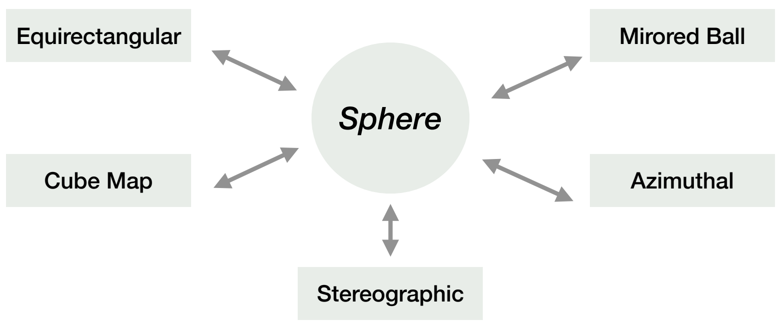
Summary Table

Format	Domain Shape	Sphere Coverage	Key Feature	Common Uses
Equirectangular	Rectangle	Full sphere	Simple mapping, heavy pole distortion	360° video, web rendering
Cube Map	6 square faces	Full sphere	Low distortion, face seams	Real-time rendering, environment map
Mirrored Ball	Circle in square	Hemisphere	Matches reflective sphere optics	Image-based lighting (IBL)
Dome Master (Azimuthal)	Circle	Hemisphere (often 180°)	Radial fisheye layout	Fulldome, planetarium, VR domes
Stereographic	Disk (unbounded)	Hemisphere	Angle-preserving projection	Mathematical visualization, conformal

Format Conversion

Basic Scheme

- Conversion formulas:
  - Between unit vector $(x, y, z) \in \mathbb{S}^2$ and corresponding image coordinates (u, v)
  - Assume the input vector is already normalized : $x^2 + y^2 + z^2 = 1$



Equirectangular Conversion

- Direct Mapping

$$u = \frac{\arctan 2(y, x)}{2\pi} \cdot W \bmod W$$

$$v = \frac{\arccos(z)}{\pi} \cdot H$$

- Inverse Mapping

Given pixel coordinates $(u, v) \in [0, W) \times [0, H)$:

$$\theta = 2\pi \cdot \frac{u}{W}, \quad \phi = \pi \cdot \frac{v}{H}$$

$$x = \sin\phi \cdot \cos\theta, \quad y = \sin\phi \cdot \sin\theta, \quad z = \cos\phi$$

Cube Map Conversion

- Direct Mapping
  - Determine major axis:
    major $= \arg\max(|x|, |y|, |z|)$
  - Project onto the corresponding face and compute $(s, t) \in [-1,1]$based on direction:

+X	$\mathrm{x} = \max(-z, x)$
-X	$\mathrm{x} = -\max(z, x)$
+Y	$\mathrm{y} = \max(x, y)$
-Y	$\mathrm{y} = -\max(x, y)$
+Z	$\mathrm{z} = \max(x, z)$
-Z	$\mathrm{z} = -\max(-x, z)$

  - Map. (s, t) to image pixels within the face

- Inverse Mapping
  - Given:
    - face index $F \in \{\pm X, \pm Y, \pm Z\}$, and
    - pixel coordinates in normalized face space $(s, t) \in [-1,1]^2$
  - Define the direction vector (x, y, z) per face F:

+X	(1, -t, -s)
-X	(-1, -t, s)
+Y	(s, 1, t)
-Y	(s, -1, -t)
+Z	(s, -t, 1)
-Z	(-s, -t, -1)

  - Normalize:

$$(x, y, z) \leftarrow \frac{(x, y, z)}{\sqrt{x^2 + y^2 + z^2}}$$

Mirrored Ball Conversion (*approx.*)

- Direct Mapping

$\theta = \arctan 2(y, x)$

$\phi = \arccos(z)$

$r = R \cdot \left(1 - \frac{\phi}{\pi}\right)$

$x' = r\cos\theta, \quad y' = r\sin\theta$

- Inverse Mapping

Given planar image coordinates (x', y') inside a circle of radius R:

$r = \sqrt{x'^2 + y'^2}, \quad \theta = \arctan 2(y', x')$

$\phi = \pi \cdot \left(1 - \frac{r}{R}\right)$

$x = \sin\phi \cdot \cos\theta, \quad y = \sin\phi \cdot \sin\theta, \quad z = \cos\phi$

Azimuthal Conversion

- Direct Mapping

Only valid for upper hemisphere $z \geq 0$

$\theta = \arctan 2(y, x)$

$\phi = \arccos(z)$

$r = R \cdot \frac{\phi}{\pi/2} = \frac{2R\phi}{\pi}$

$x' = r\cos\theta, \quad y' = r\sin\theta$

- Inverse Mapping

Given $(x', y') \in \mathbb{R}^2$ in circular image of radius R:

$r = \sqrt{x'^2 + y'^2}, \quad \theta = \arctan 2(y', x')$

$\phi = \frac{\pi}{2} \cdot \frac{r}{R}$

$x = \sin\phi \cdot \cos\theta, \quad y = \sin\phi \cdot \sin\theta, \quad z = \cos\phi$

Note: Valid for $r \leq R$ → upper hemisphere only

Stereographic Conversion

- Direct Mapping

$$x' = \frac{x}{1+z}, \quad y' = \frac{y}{1+z}$$

(valid for $z \neq -1$, i.e., north hemisphere)

- Inverse Mapping

Given planar coordinates $(x', y') \in \mathbb{R}^2$:

$$r^2 = x'^2 + y'^2$$

$$x = \frac{2x'}{1+r^2}, \quad y = \frac{2y'}{1+r^2}, \quad z = \frac{r^2-1}{1+r^2}$$

OBS: Conversion to Complex Plane

The Complex Plane

Riemann Sphere / Complex Plane Conversion

The stereographic projection maps points on the unit sphere $\mathbb{S}^2 \subset \mathbb{R}^3$ to the extended complex plane $\widehat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$.

Stereographic Projection Formulas

- **From Sphere to Complex Plane:**

$$z_{\mathbb{C}} = \frac{x + iy}{1 - z} \quad \text{for } z \neq 1$$

- **From Complex Plane to Sphere:**
  Let $z_{\mathbb{C}} = u + iv$, with $r^2 = |z_{\mathbb{C}}|^2 = u^2 + v^2$, then:

$$x = \frac{2u}{1 + r^2}, \quad y = \frac{2v}{1 + r^2}, \quad z = \frac{r^2 - 1}{1 + r^2}$$

- **Point at Infinity:**

$$z_{\mathbb{C}} = \infty \quad \leftrightarrow \quad (x, y, z) = (0, 0, 1)$$

Summary Table

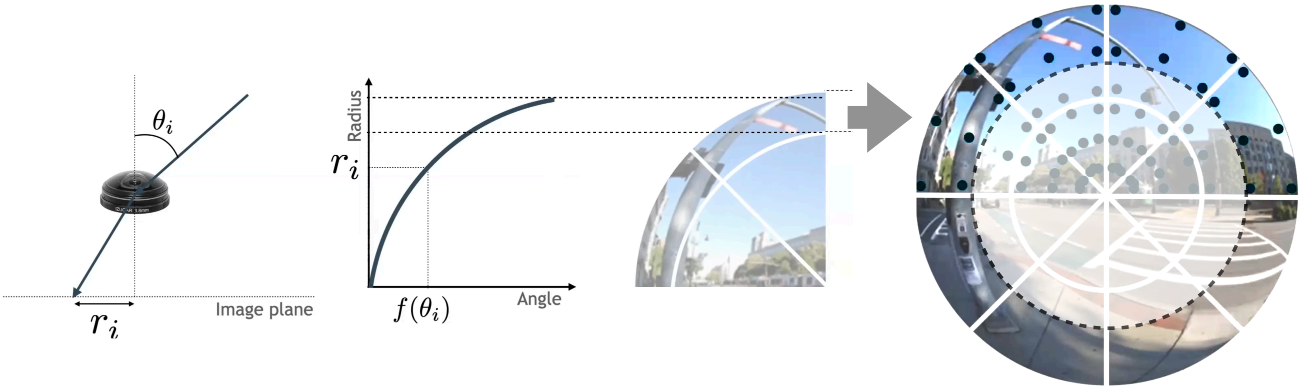
Conversion	**Formula**
Sphere → Complex Plane	$z_{\mathbb{C}} = \frac{x+iy}{1-z}$
Complex Plane → Sphere	$x = \frac{2\,\mathrm{Re}(z_{\mathbb{C}})}{1+\lvert z_{\mathbb{C}}\rvert^2} y = \frac{2\,\mathrm{Im}(z_{\mathbb{C}})}{1+\lvert z_{\mathbb{C}}\rvert^2} z = \frac{\lvert z_{\mathbb{C}}\rvert^2-1}{1+\lvert z_{\mathbb{C}}\rvert^2}$
$z_{\mathbb{C}} = \infty$	$(x, y, z) = (0, 0, 1)$

Stereographic projection between the Riemann Sphere and the extended complex plane.

Omnidirectional Awareness

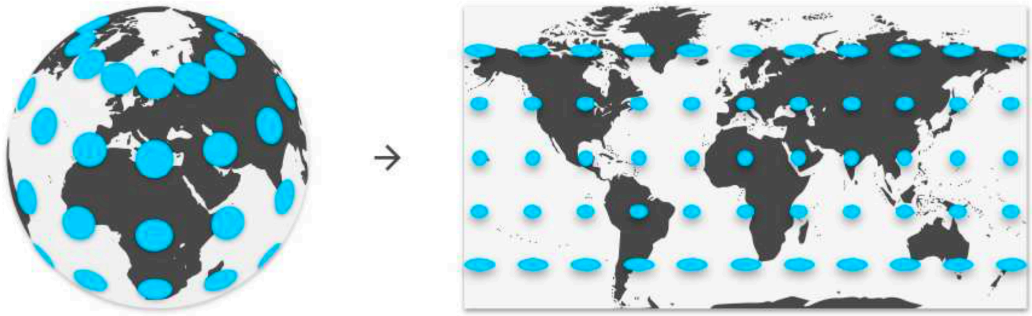
Lens Distortion

- Transfer Function



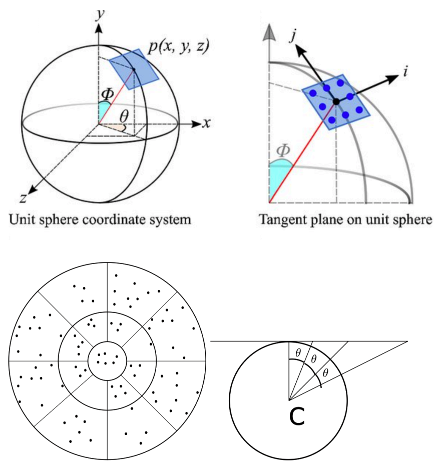
Parametrization Distortion

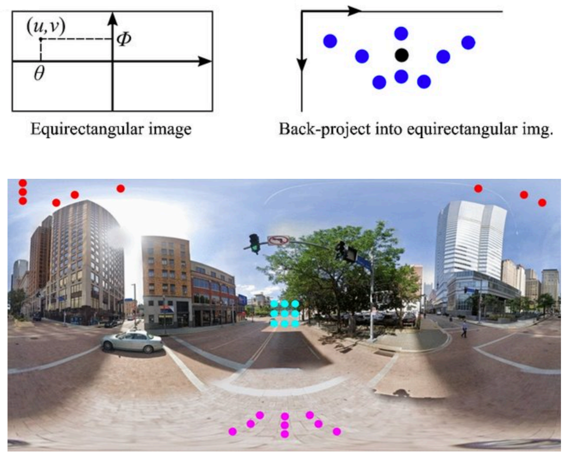
- Equirectangular Projection



Adaptive Sampling

- Non-Uniform Sampling



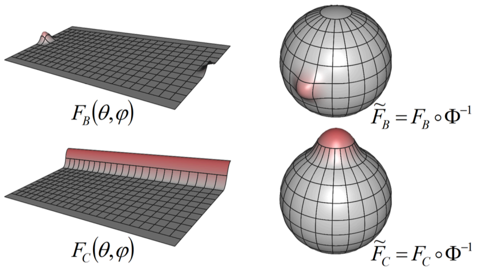


Awareness

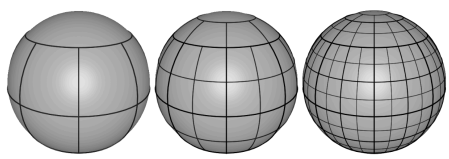
- Metric Aware
- Sampling Aware
- Computation Aware
- Content Aware
- View Aware

Metric Aware

- Basis Functions

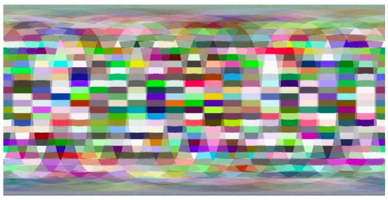


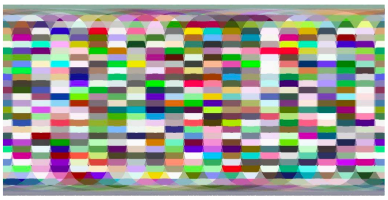
- Multiresolution

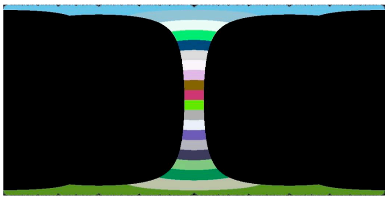


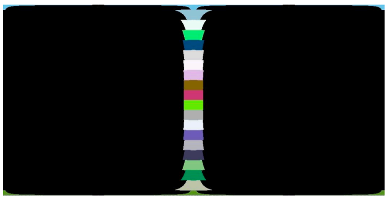
Sampling Aware

- Regular Sampling
- Adapted Sampling



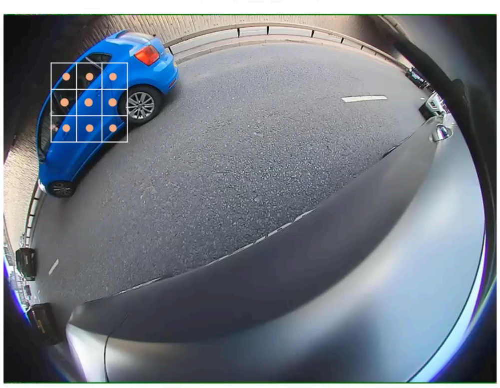


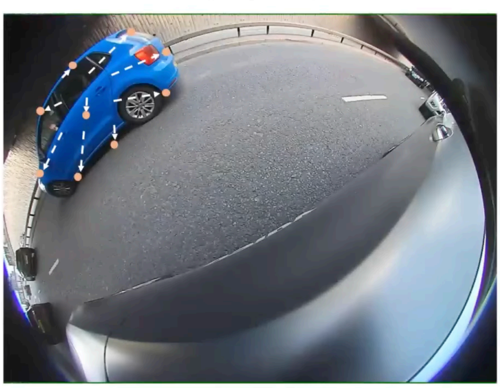




Computation Aware

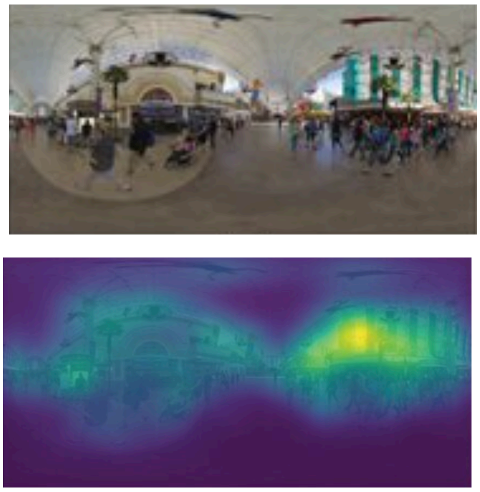
- Convolution Operator





Content Aware

- Saliency



- Segmentation

View Aware

- Gaze Direction