Master SIF - REP (Part 1)
Image acquisition and Projection Models

Thomas Maugey
thomas.maugey@inria.fr

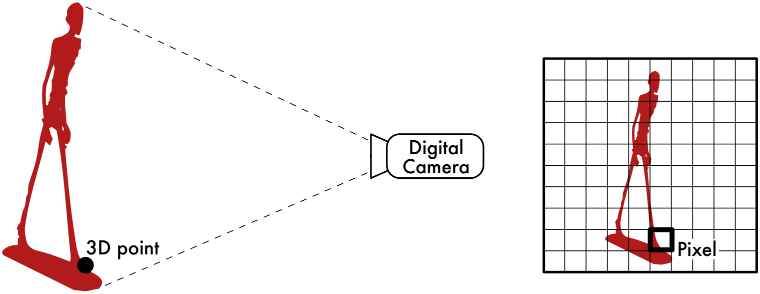
Fall 2023

Table of Contents

Table of Contents

What is a projection model?

Find the relationship between a point in the 3D world and the corresponding pixel in an image.



Photodetector

Sensor that converts a certain electromagnetic activity into a electrical current.

Usually a **semiconductor** that transforms a light photons into electrons only for a certain band of energy. The number of electrons collected is proportional to the quantity of light that is received.

One photodiode per Red/Green/Blue channel:

- CCD: charge-coupled device
- CMOS: complementary metal-oxide-semiconductor

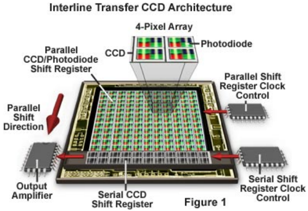
One photodiode for all Red/Green/Blue channels:

- Feoven

From photodiode to Pixel

A Pixel is a *picture element*

Active-Pixel Sensor (APS) associate to each pixel, one (or several) photodetector and an active amplifier.



APS based on CCD

How to capture the light ?

The issue is not only to capture the light intensity, but also the light direction

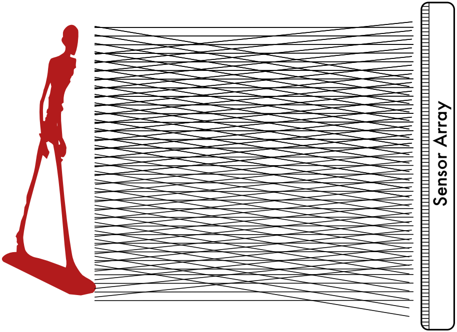
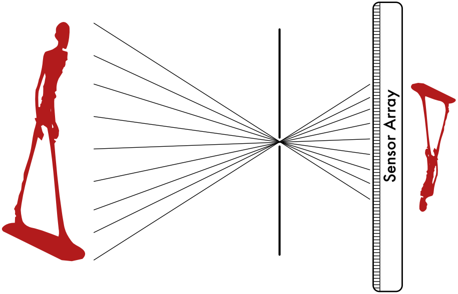


Table of Contents

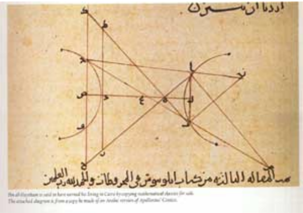
Pinhole capture = Perspective projection

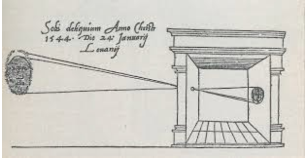
Filter the light with a hole, in order to have, at most, one ray per 3D point in the scene.



An old idea

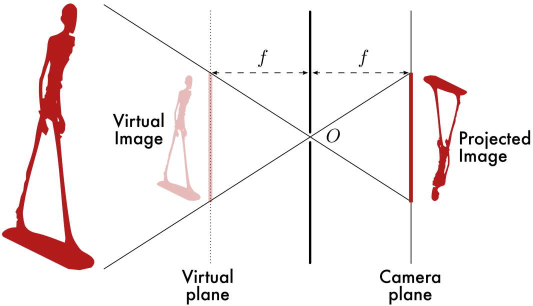
- Ibn Al-Haytham (965-1039)
- Leonardo Da Vinci (1514)
- Johann Zahn (1685)





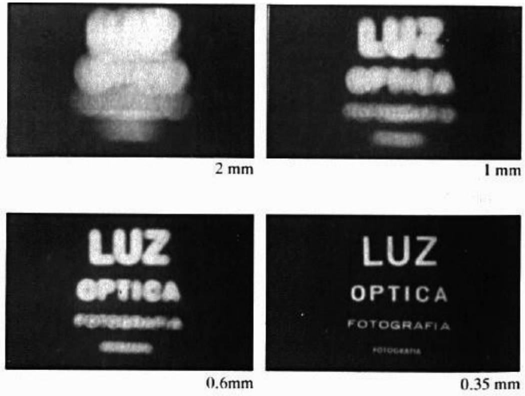
Aperture and focal length

- The **aperture** is the hole (pinhole) center O of the camera through which the rays are passing
- The **focal length** f is the distance between the aperture and the camera plane



Aperture's size

It controls the trade-off between the *quantity of light* and the *uniqueness of the ray direction* per sensor.

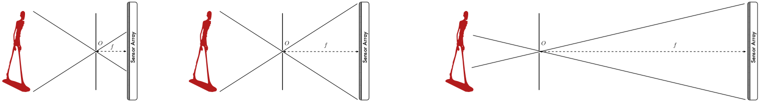


[Wikipedia]

In the following, we consider that it is a point.

Focal length

It controls the angle of view of the camera (and thus the zoom).



Camera objectives:

- Small f: wide angle
- High f: zoom

$f = 28$ mm

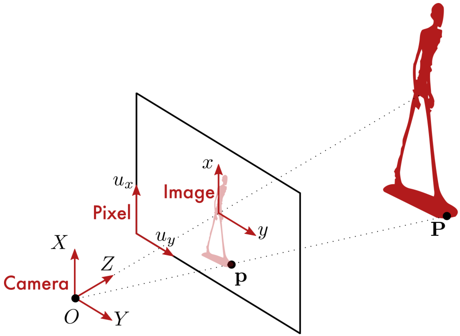
$f = 50$ mm

$f = 70$ mm

$f = 210$ mm

[Wikipedia]

Three coordinate systems



3D point:

$$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

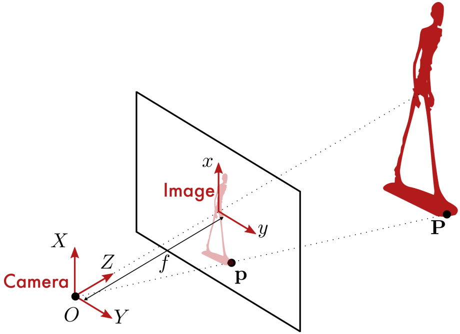
Projected point:

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Pixel:

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

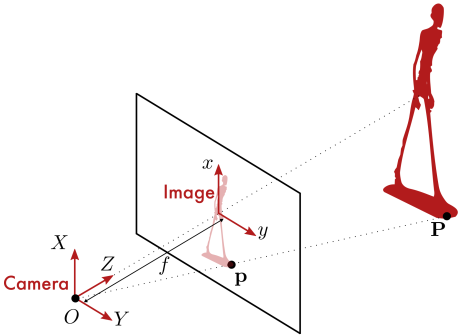
From Camera to Image coordinates



The relationship between $\mathbf{P}$ and $\mathbf{p}$ is given by:

$$\begin{cases} x = ? \\ y = ? \end{cases}$$

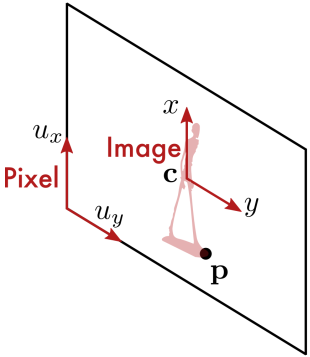
From Camera to Image coordinates



The relationship between $\mathbf{P}$ and $\mathbf{p}$ is given by:

$$\begin{cases} x = f\dfrac{X}{Z} \\ y = f\dfrac{Y}{Z} \end{cases}$$

From Image to Pixel coordinates



Camera center:

$$\mathbf{c} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

Resolution (pixel.mm^{-1}):

$$\mathbf{k} = \begin{bmatrix} k_x \\ k_y \end{bmatrix}$$

Pixel coordinates:

$$\begin{cases} u_x = k_x(x + x_0) \\ u_y = k_y(y + y_0) \end{cases}$$

Homogeneous Coordinates

Represent a n-dimensional coordinate with an $n+1$-dimension vector:

$$\begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \rightarrow \begin{bmatrix} v_1 \\ \vdots \\ v_n \\ 1 \end{bmatrix}$$

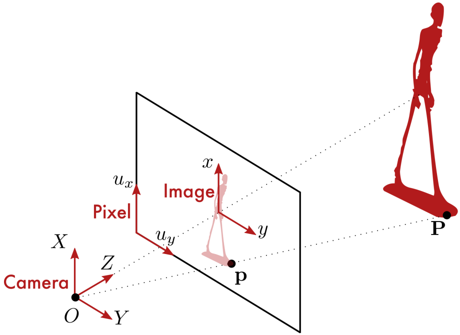
Homogeneous divide:

$$\begin{bmatrix} v_1 \\ \vdots \\ v_n \\ w \end{bmatrix} \rightarrow \begin{bmatrix} v_1/w \\ \vdots \\ v_n/w \\ 1 \end{bmatrix}$$

Two vectors are said **homogeneous** if their homogeneous divide is equal, *e.g.*,

$$\begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} 4 \\ 6 \\ 2 \end{bmatrix} \equiv \begin{bmatrix} 6 \\ 9 \\ 3 \end{bmatrix}$$

From Camera to Pixel coordinates



$$\begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} \equiv \begin{bmatrix} k_x f X + k_x x_0 Z \\ k_y f Y + k_y y_0 Z \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} k_x f & 0 & k_x x_0 \\ 0 & k_y f & k_y y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsic Matrix } \mathbf{K}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Intrinsic matrix

The intrinsic matrix is given by:

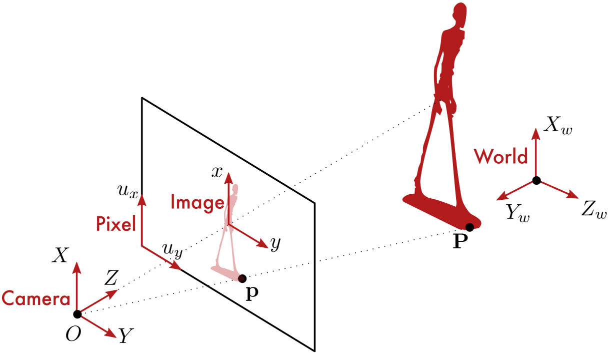
$$\mathbf{K} = \begin{bmatrix} k_x f & s & k_x x_0 \\ 0 & k_y f & k_y y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

with

- s: skew parameter (in pixels)
- (x_0, y_0) : principal point coordinates (in mm)
- f : focal length (in mm)
- $k_x,\ k_y$: vertical, horizontal resolution (in pixel.mm^{-1})

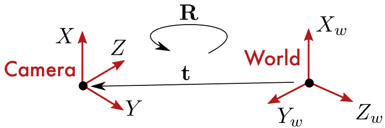
Play with it:
http://ksimek.github.io/2013/08/13/intrinsic/

World coordinates



The point $\mathbf{P}$ might be expressed in the world coordinate system: $\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$

Change of coordinate system



If (α, β, γ) are the euler angles of the rotation around respectively the (X_w, Y_w, Z_w) axis, the rotation matrix is given by:

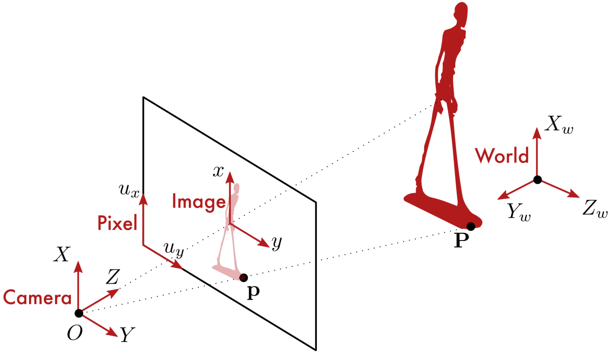
$$\mathbf{R} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

If the camera center O coordinates expressed in the world system are given by $\mathbf{t}$, the coordinate system change is expressed as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{R}\left(\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} - \mathbf{t}\right) = \underbrace{\begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{t} \end{bmatrix}}_{\text{Extrinsic Matrix } \mathbf{E}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Play with it: http://ksimek.github.io/2012/08/22/extrinsic/

From World to Pixel coordinates



$$\begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} \equiv \begin{bmatrix} k_x f & 0 & k_x x_0 \\ 0 & k_y f & k_y y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{Rt} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \mathbf{KE} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Perspective projection's properties

- Distant objects look smaller (exercice)
- Lines project to lines (exercice)
- Parallel lines are in general no longer parallel (exercice)
- Parallel lines meet at a vanishing point
- Angles are not preserved
- 3D points can be retrieved from camera motion (cf. Epipolar Geometry)

Pose estimation

Unknown rotations and positions estimated thanks to known world coordinate positions and their associated pixel positions

$$\begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} \equiv \begin{bmatrix} k_x f & 0 & k_x x_0 \\ 0 & k_y f & k_y y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{Rt} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Algorithms

- Find many matches
- And minimize $\min_{(\mathbf{K},\mathbf{R},\mathbf{t})} \sum_i r_i(\mathbf{K},\mathbf{R},\mathbf{t})^2 = \min_{(\mathbf{K},\mathbf{R},\mathbf{t})} \sum_i ||p_i^{obs} - p_i^{est}(\mathbf{K},\mathbf{R},\mathbf{t})||_2^2$
- Gauss-Newton Solver
  - By first finding inital values $(\mathbf{K}_0, \mathbf{R}_0, \mathbf{t}_0)$
  - Then iteratively refine $(\mathbf{K}_{s+1}, \mathbf{R}_{s+1}, \mathbf{t}_{s+1}) = (\mathbf{K}_s, \mathbf{R}_s, \mathbf{t}_s) + \delta(\mathbf{K},\mathbf{R},\mathbf{t})$
  - where $\delta(\mathbf{K},\mathbf{R},\mathbf{t}) = -(\mathbf{J}_r^T\mathbf{J}_r)^{-1}\mathbf{J}_r^\top r$
- Levenberg-Marquardt

[Gennery, D. B. (1979, Nov.). Stereo-camera calibration. In Proceedings ARPA IUS Workshop (pp. 101-107)]

Pose estimation applications

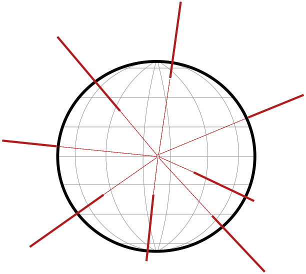
- Calibration
- Augmented reality
- Video summary

Table of Contents

What is an omnidirectional image?

Definition

An image that represents the light activity arriving at a point (the image center) from every direction (360^o field of view).



Applications:

- Virtual reality
  Head-Mounted Display (HDM)

- Free viewpoint Television
  More than 1 million videos uploaded on Youtube in 1 year
- Robotics

Omnidirectional capture?

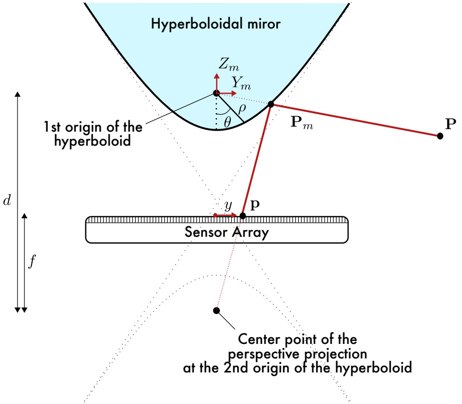
The main issue is to cover a wide angle of view (360^o)

- Multiple perspective projections by several small degree of view cameras (180^o or 360^o field of view)

- A curved mirror + one single perspective camera (180^o field of view)
- Fish-eye lenses (180^o field of view)

In the following, we present the two last ones.

Catadioptric cameras: hyper-catadioptric

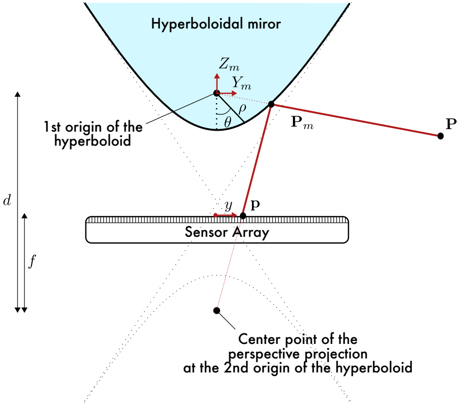


Projection on the mirror of equation $\rho = \frac{a}{1+e\cos\theta}$:

Perspective projection on the sensor array:

In the image coordinate:

Catadioptric cameras: hyper-catadioptric



Projection on the mirror of equation $\rho = \frac{a}{1+e\cos\theta}$:

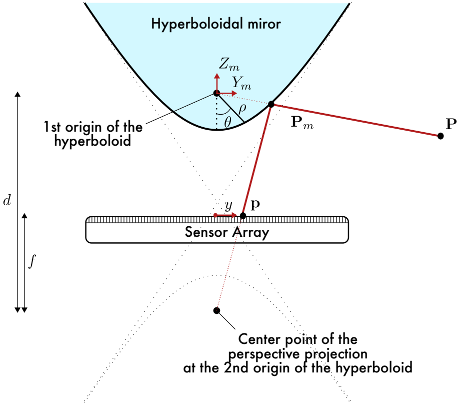
$$\mathbf{P}_m = \frac{\mathbf{P}}{||\mathbf{P}||}.\rho$$

$$\begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} = \frac{\rho}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Perspective projection on the sensor array:

In the image coordinate:

Catadioptric cameras: hyper-catadioptric



Projection on the mirror of equation $\rho = \frac{a}{1+e\cos\theta}$:

$$\mathbf{P}_m = \frac{\mathbf{P}}{||\mathbf{P}||}.\rho$$

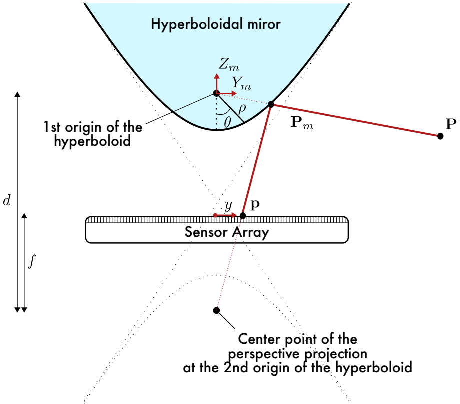
$$\begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} = \frac{\rho}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Perspective projection on the sensor array:

$$\begin{cases} x = f\frac{X_m}{Z_m+d} \\ y = f\frac{Y_m}{Z_m+d} \end{cases}$$

In the image coordinate:

Catadioptric cameras: hyper-catadioptric



Projection on the mirror of equation $\rho = \frac{a}{1+e\cos\theta}$:

$$\mathbf{P}_m = \frac{\mathbf{P}}{||\mathbf{P}||}.\rho$$

$$\begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} = \frac{\rho}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Perspective projection on the sensor array:

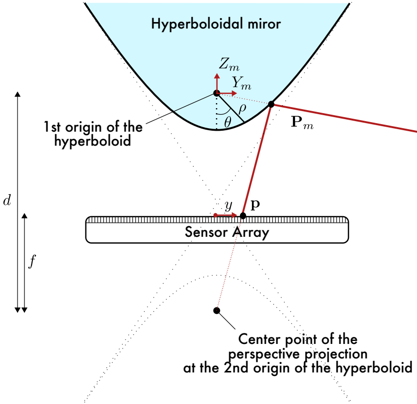
$$\begin{cases} x = f\frac{X_m}{Z_m+d} \\ y = f\frac{Y_m}{Z_m+d} \end{cases}$$

And:

$d = \frac{2ae}{1-e^2}$ and $\cos(\theta) = \frac{Z}{||\mathbf{P}||}$,

In the image coordinate:

Catadioptric cameras: hyper-catadioptric



Projection on the mirror of equation $\rho = \frac{a}{1+e\cos\theta}$:

$$\mathbf{P}_m = \frac{\mathbf{P}}{||\mathbf{P}||}.\rho$$

$$\begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} = \frac{\rho}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Perspective projection on the sensor array:

$$\begin{cases} x = f\frac{X_m}{Z_m+d} \\ y = f\frac{Y_m}{Z_m+d} \end{cases}$$

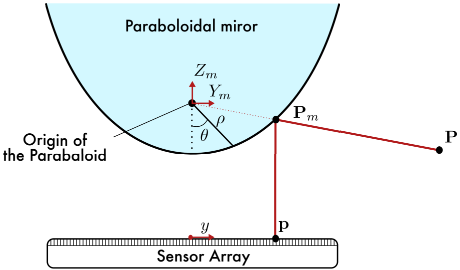
And:

$d = \frac{2ae}{1-e^2}$ and $\cos(\theta) = \frac{Z}{||\mathbf{P}||}$,

In the image coordinate:

$$\mathbf{p} = \left[\frac{\frac{1-e^2}{1+e^2} fX}{\frac{2e}{1+e^2}\sqrt{X^2+Y^2+Z^2}+Z}, \frac{\frac{1-e^2}{1+e^2} fY}{\frac{2e}{1+e^2}\sqrt{X^2+Y^2+Z^2}+Z} \right]^\top$$

Catadioptric cameras: Para-catadioptric

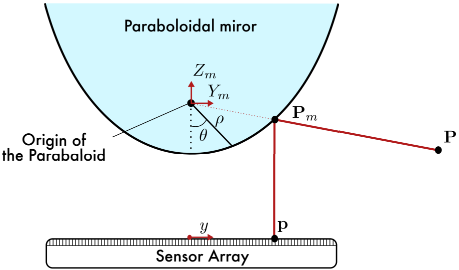


Projection on the mirror of equation $\rho = \frac{a}{1+\cos\theta}$:

Orthogonal projection on the sensor array:

In the image coordinate:

Catadioptric cameras: Para-catadioptric



Projection on the mirror of equation $\rho = \frac{a}{1+\cos\theta}$:

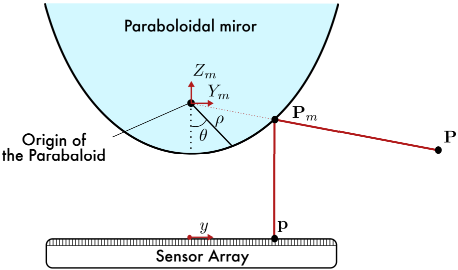
$$\mathbf{P}_m = \frac{\mathbf{P}}{||\mathbf{P}||}.\rho$$

$$\begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} = \frac{\rho}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Orthogonal projection on the sensor array:

In the image coordinate:

Catadioptric cameras: Para-catadioptric



Projection on the mirror of equation $\rho = \frac{a}{1+\cos\theta}$:

$$\mathbf{P}_m = \frac{\mathbf{P}}{||\mathbf{P}||}.\rho$$

$$\begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} = \frac{\rho}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

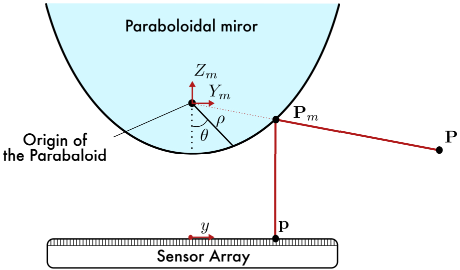
Orthogonal projection on the sensor array:

$$\begin{cases} x = X_m \\ y = Y_m \end{cases}$$

And $\cos(\theta) = \frac{Z}{||\mathbf{P}||}$,

In the image coordinate:

Catadioptric cameras: Para-catadioptric



Projection on the mirror of equation $\rho = \frac{a}{1+\cos\theta}$:

$$\mathbf{P}_m = \frac{\mathbf{P}}{||\mathbf{P}||}.\rho$$

$$\begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} = \frac{\rho}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Orthogonal projection on the sensor array:

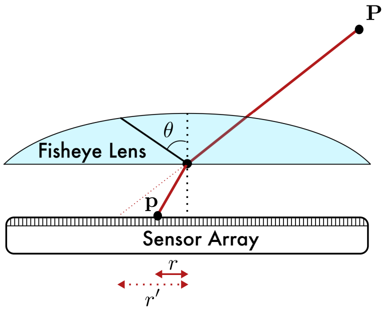
$$\begin{cases} x = X_m \\ y = Y_m \end{cases}$$

And $\cos(\theta) = \frac{Z}{||\mathbf{P}||}$,

In the image coordinate:

$$\mathbf{p} = \left[\frac{aX}{\sqrt{X^2+Y^2+Z^2}+Z}, \frac{aY}{\sqrt{X^2+Y^2+Z^2}+Z}\right]^\top$$

Fisheye lens



Radial distortion of the lens:

$$r \neq r'$$

Example of radial distortion [F01]:

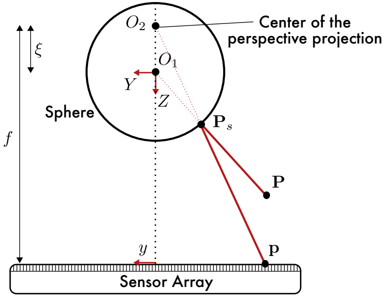
$$r' = \frac{k_1 r}{1 - k_2 r^2}$$

Usually, this distortion reads [C07]:

$$r = f(\theta)$$

[F01] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In CVPR (1), pages 125—132, 2001.
[C07] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A generic fisheye camera model for robotic applications. In IROS, pages 1683{1688, 2007

Unified Spherical Model



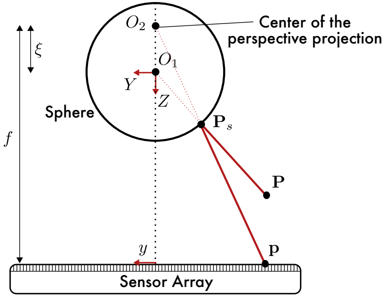
Projection on the sphere of center O_1:

Perspective projection of center O_2 on the sensor array:

In the image coordinates:

[J. Courbon et al. 2012. Evaluation of the Unified Model of the Sphere for Fisheye Cameras in Robotic Applications]

Unified Spherical Model



Projection on the sphere of center O_1:

$$\mathbf{P}_s = \frac{\mathbf{P}}{||\mathbf{P}||}$$

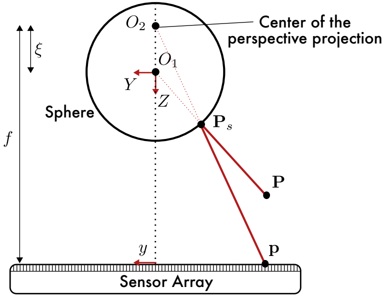
$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = \frac{1}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Perspective projection of center O_2 on the sensor array:

In the image coordinates:

[J. Courbon et al. 2012. Evaluation of the Unified Model of the Sphere for Fisheye Cameras in Robotic Applications]

Unified Spherical Model



Projection on the sphere of center O_1:

$$\mathbf{P}_s = \frac{\mathbf{P}}{||\mathbf{P}||}$$

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = \frac{1}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

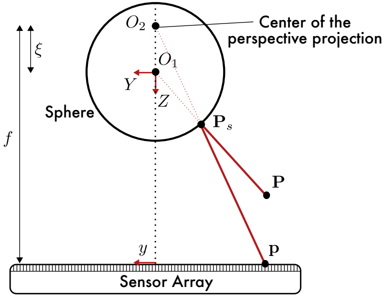
Perspective projection of center O_2 on the sensor array:

$$\begin{cases} x = f\frac{X_s}{Z_s+\xi} \\ y = f\frac{Y_s}{Z_s+\xi} \end{cases}$$

In the image coordinates:

[J. Courbon et al. 2012. Evaluation of the Unified Model of the Sphere for Fisheye Cameras in Robotic Applications]

Unified Spherical Model



Projection on the sphere of center O_1:

$$\mathbf{P}_s = \frac{\mathbf{P}}{||\mathbf{P}||}$$

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = \frac{1}{\sqrt{X^2+Y^2+Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Perspective projection of center O_2 on the sensor array:

$$\begin{cases} x = f\frac{X_s}{Z_s+\xi} \\ y = f\frac{Y_s}{Z_s+\xi} \end{cases}$$

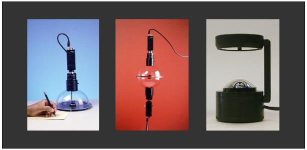
In the image coordinates:

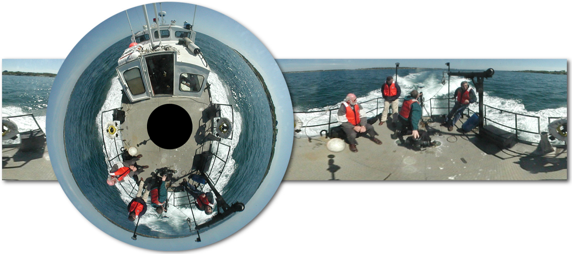
$$\mathbf{p} = \left[\frac{fX}{\xi\sqrt{X^2+Y^2+Z^2}+Z}, \frac{fY}{\xi\sqrt{X^2+Y^2+Z^2}+Z}\right]^\top$$

[J. Courbon et al. 2012. Evaluation of the Unified Model of the Sphere for Fisheye Cameras in Robotic Applications]

Example of Captured 360^o image

Catadioptric Cameras





[S.K. Nayar and V.N. Peri, "Folded Catadioptric Cameras," Panoramic Vision, pp. 103-119, R., Springer-Verlag, Apr. 2001.]
[S. Baker and S.K. Nayar, ,"Single Viewpoint Catadioptric Cameras," Panoramic Vision, pp. 39-71, R., Springer-Verlag, Apr. 2001.]
[S. Baker and S.K. Nayar, "A Theory of Single-Viewpoint Catadioptric Image Formation," International Journal on Computer Vision, Vol. 35, No. 2, pp. 175-196, Nov. 1999.]

Example of Captured 360^o image

Fisheye Cameras

[http://polymathprogrammer.com/2009/10/15/convert-360-degree-fisheye-image-to-landscape-mode/]

Line projections

Let us take a line of equation

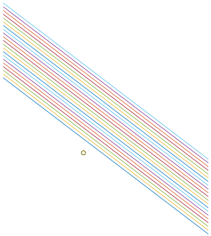
$$\begin{cases} X = a_x t + X_0 \\ Y = a_y t + Y_0 \\ Z = a_z t + Z_0 \end{cases}$$

If $k_x = k_y = f = 1$ and $x_0 = y_0 = 0$. We can write

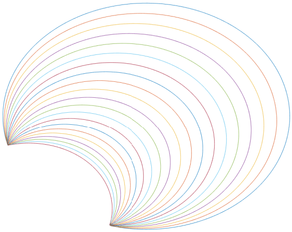
$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} \frac{a_x t + X_0}{\xi\sqrt{(a_x t+X_0)^2+(a_y t+Y_0)^2+(a_z t+Z_0)^2}+a_z t+Z_0} \\ \frac{a_y t + Y_0}{\xi\sqrt{(a_x t+X_0)^2+(a_y t+Y_0)^2+(a_z t+Z_0)^2}+a_z t+Z_0} \end{bmatrix}$$

The projection of lines are curves in the spherical image.

Parallel lines projections



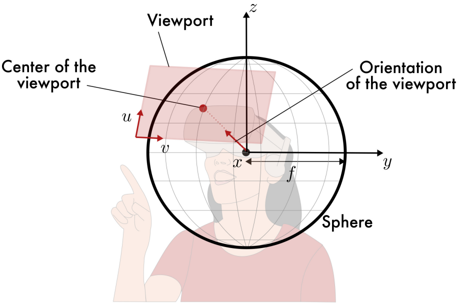
Parallel lines in the 3D space



Projection in the spherical camera

The vanishing points are visible in the scene.

Viewport rendering



The pixels of the spherical image are placed on the sphere

$$\mathbf{P}_s = [x, y, z]^\top$$

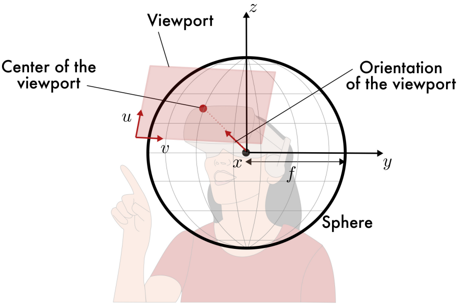
The viewport is oriented towards a direction whose rotation matrix is given by $\mathbf{R}$.

The center of the viewport is at (c_u, c_v), with corresponding resolutions (k_u, k_v).

The projection of $\mathbf{P}_s$ on the viewport is:

[De Simone, Francesca et al. "Geometry-driven quantization for omnidirectional image coding." PCS (2016).]

Viewport rendering



The pixels of the spherical image are placed on the sphere $\mathbf{P}_s = [x, y, z]^\top$

The viewport is oriented towards a direction whose rotation matrix is given by $\mathbf{R}$.

The center of the viewport is at (c_u, c_v), with corresponding resolutions (k_u, k_v).

The projection of $\mathbf{P}_s$ on the viewport is:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \begin{bmatrix} k_u c_u & 0 & k_u f \\ k_v c_v & k_v f & 0 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

[De Simone, Francesca et al. "Geometry-driven quantization for omnidirectional image coding." PCS (2016).]

Table of Contents

References

- Radke, R. J. (2013). Computer vision for visual effects. Cambridge University Press.
- Forsyth, D. A., and Ponce, J. (2003). A modern approach. Computer vision: a modern approach, 88-101.
- http://ksimek.github.io/