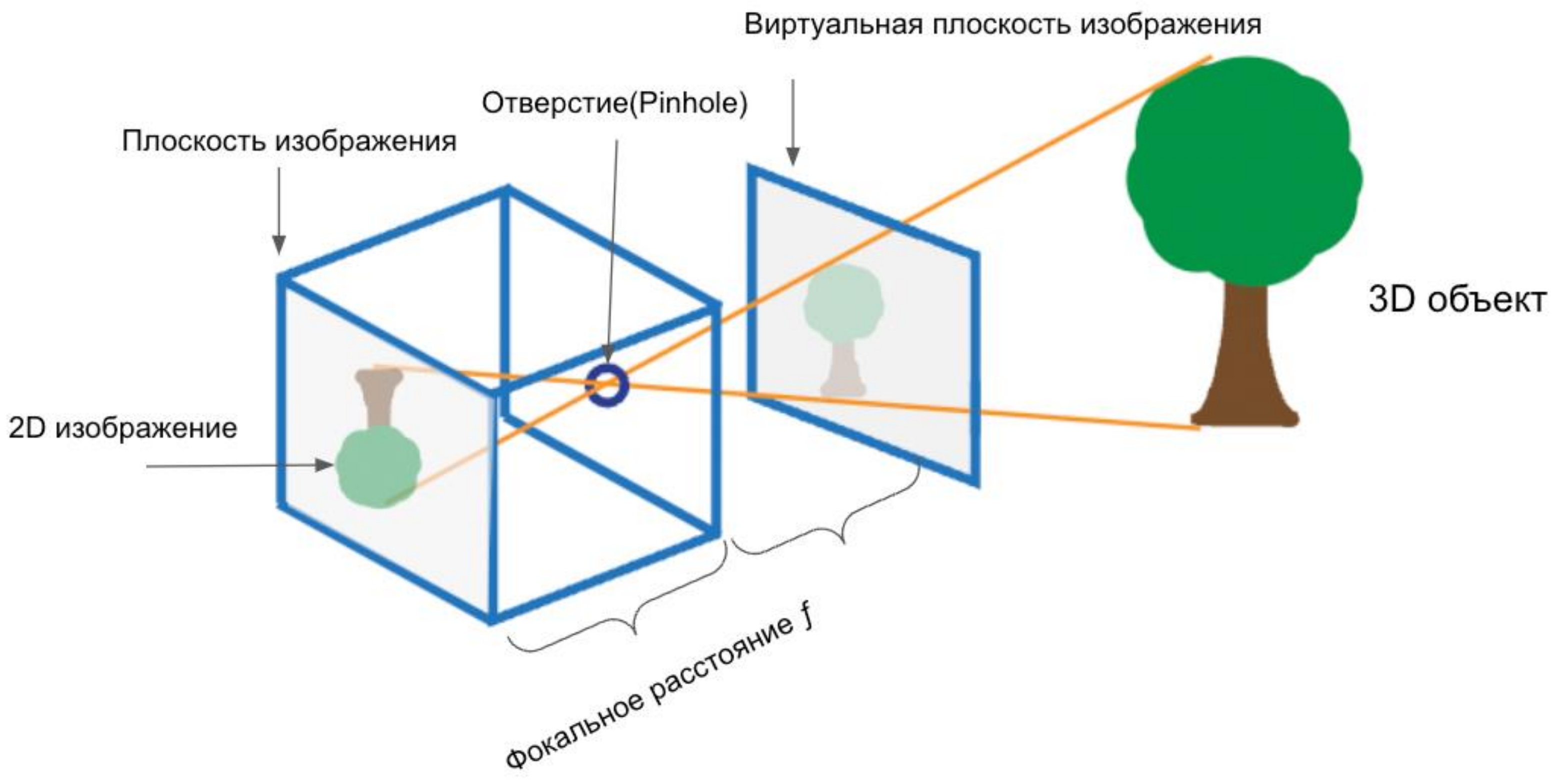
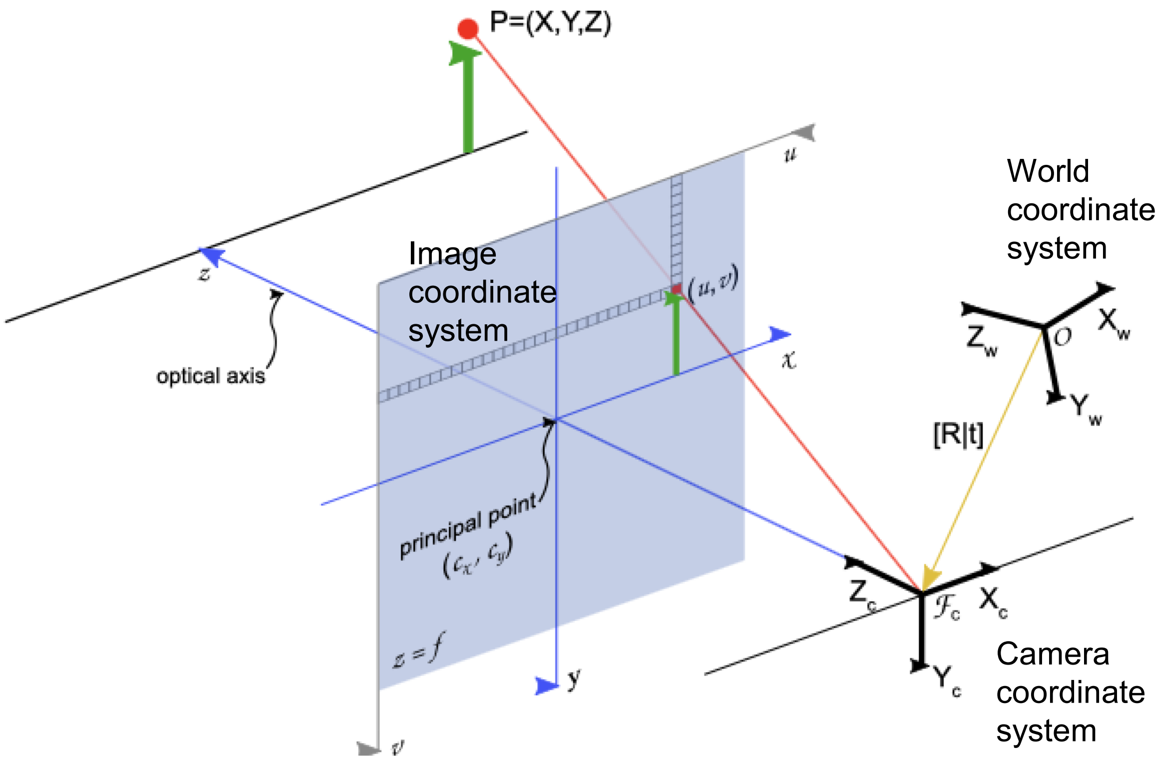
Модель камеры, калибровка, гомография

Камера-обскура (pinhole camera)



Камера-обскура (pinhole camera)



Однородные координаты

Перевод из обычных в однородные:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Однородные координаты
точки изображения

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

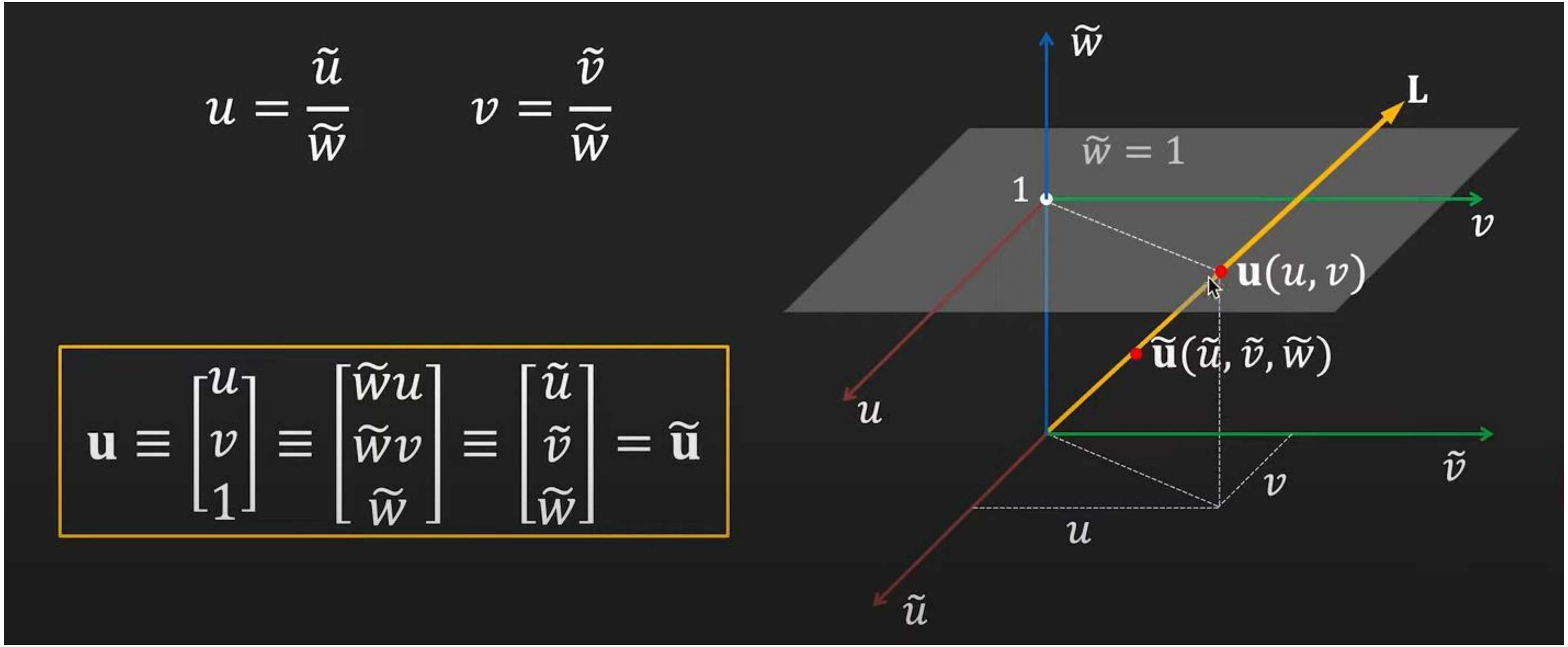
Однородные координаты
точки сцены

Перевод из однородных в обычные:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

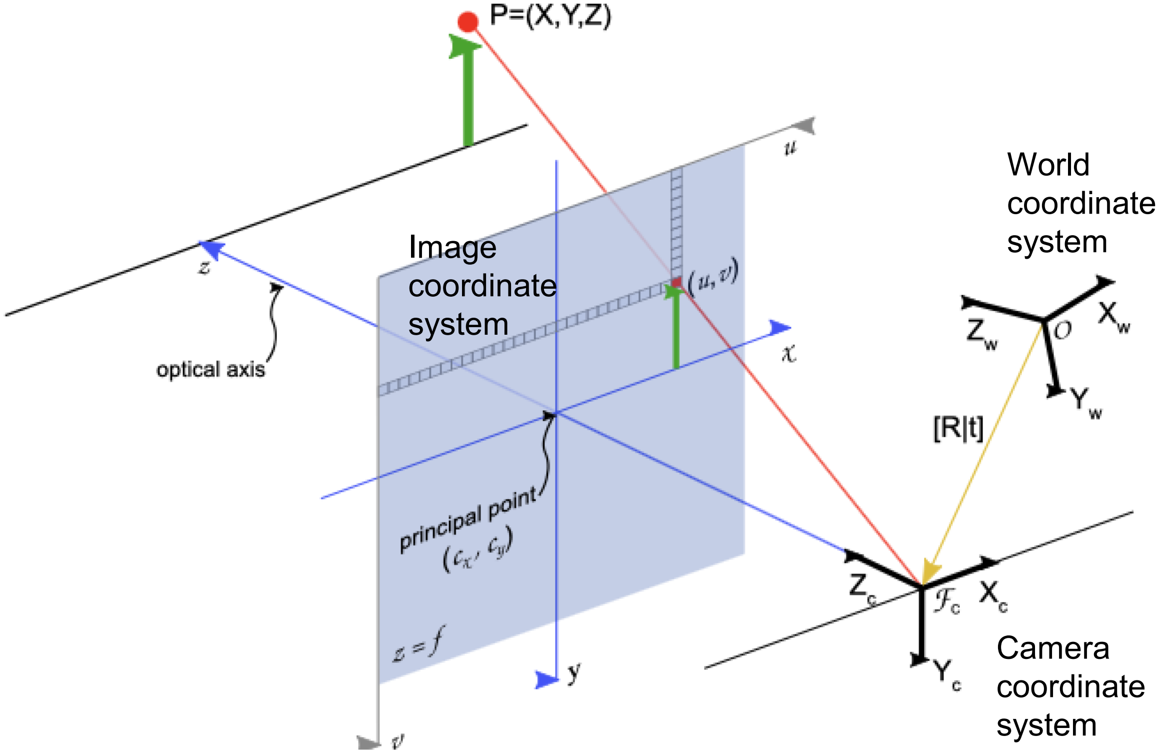
$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Однородные координаты



https://youtu.be/qByYk6JggQU?si=Hk2bGrwHTwNxzhS4

От пикселя к лучу

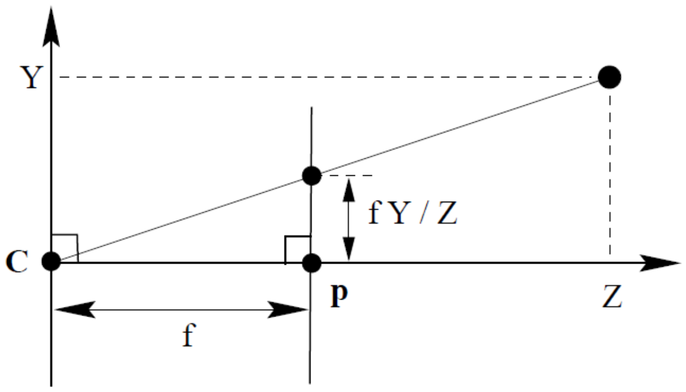


Был пиксель с координатами (u, v)

Стал луч с координатами (u, v, 1)

Можно двигаться как угодно вдоль луча, проекция на изображение не поменяется k(u, v, 1)

Focal length



$$\frac{Y}{v} = \frac{Z}{f} \quad \Rightarrow \quad v = f\frac{Y}{Z}$$

$$(X \quad Y \quad Z) \rightarrow (u \quad v) - ?$$

$$(u \quad v) = \left(f\frac{X}{Z} \quad f\frac{Y}{Z}\right)$$

$$\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{pmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \\ 1 \end{pmatrix}$$

Однородные координаты
можно умножать/делить на любое число

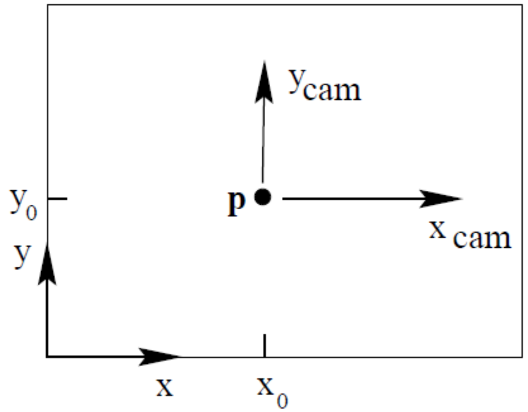
Principal point

$$(u \quad v) = \left(f\frac{X}{Z} \quad f\frac{Y}{Z}\right) \rightarrow (u \quad v) = \left(f\frac{X}{Z} + p_x \quad f\frac{Y}{Z} + p_y\right)$$

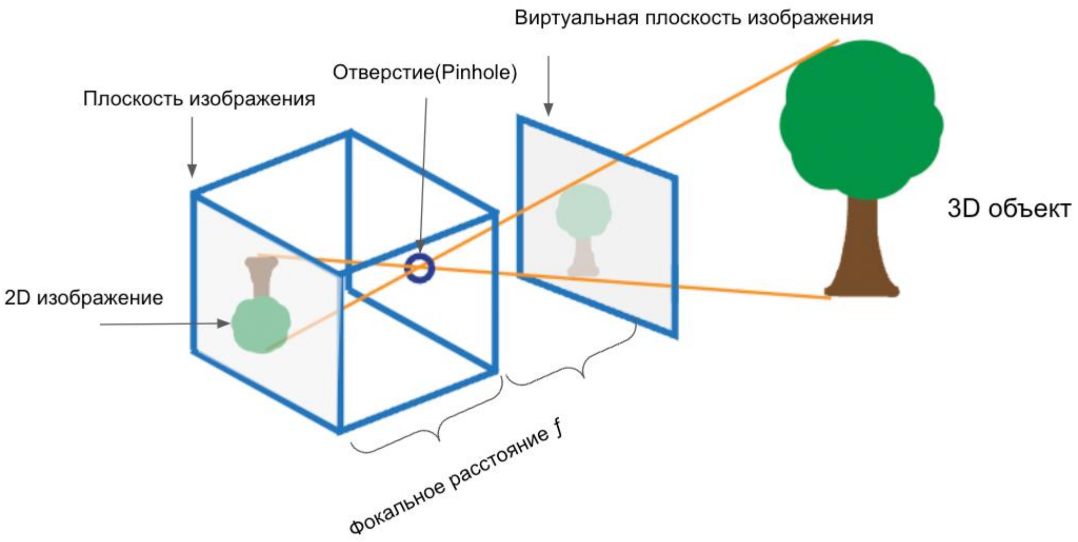
Было так

Хотим так

$$\begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{pmatrix} f\frac{X}{Z} + p_x \\ f\frac{Y}{Z} + p_y \\ 1 \end{pmatrix}$$



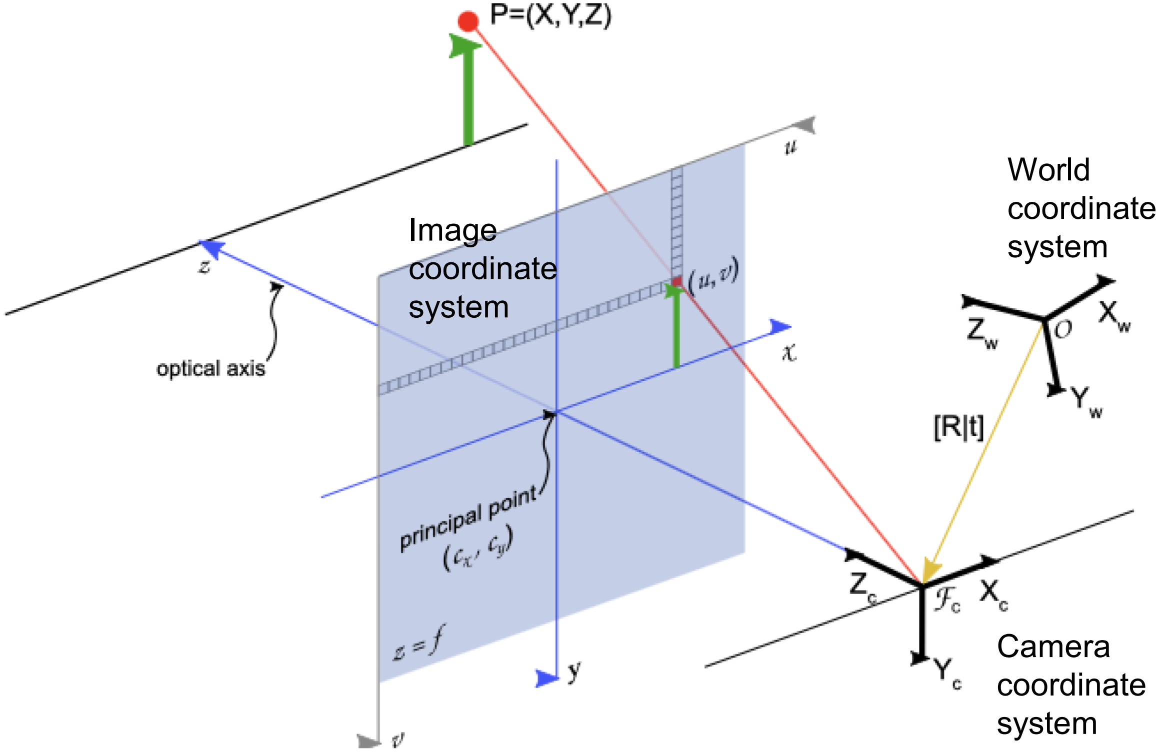
Модель камеры-обскуры



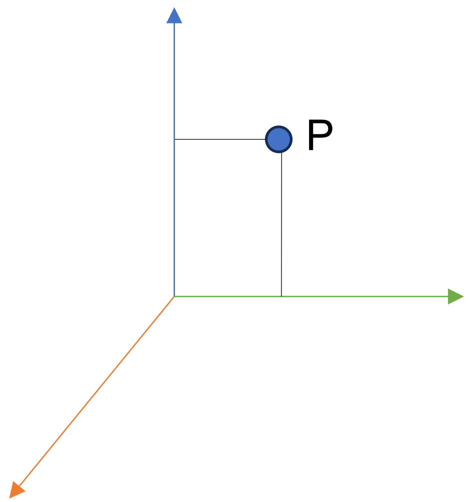
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \underbrace{\begin{pmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{Проецирование}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} f_x X + Z p_x \\ f_y Y + Z p_y \\ Z \end{pmatrix} = \begin{pmatrix} f_x \frac{X}{Z} + p_x \\ f_y \frac{Y}{Z} + p_y \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} u \\ v \end{pmatrix}$$

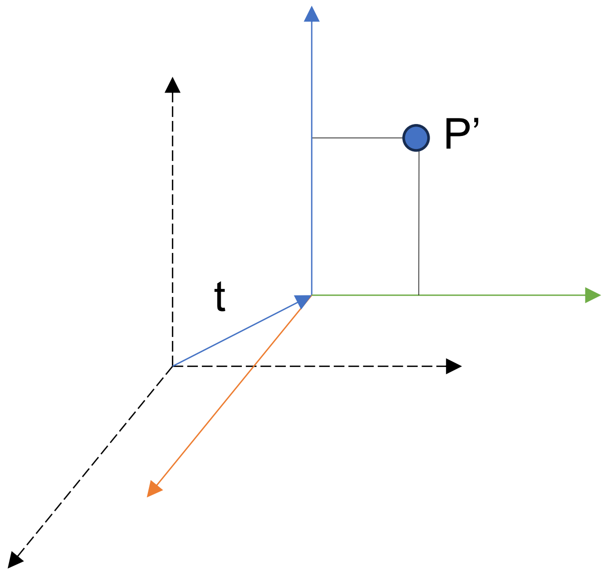
3D точка — 3D точка в однородных координатах — Проецирование — 3D точка в системе координат камеры — 2D точка в однородных координатах — Та же 2D точка в однородных координатах — Искомый пиксель в однородных координатах — Искомый пиксель

Если 3D точка не в системе координат камеры



Сдвиг системы координат

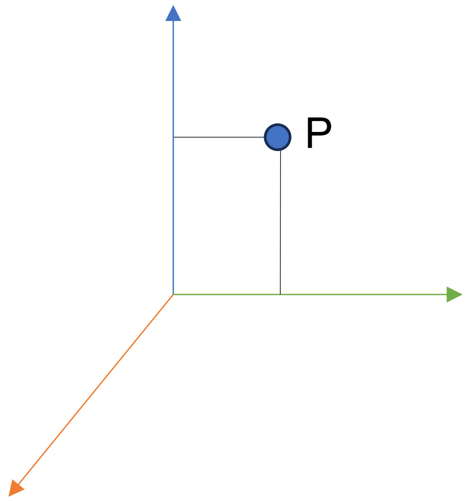


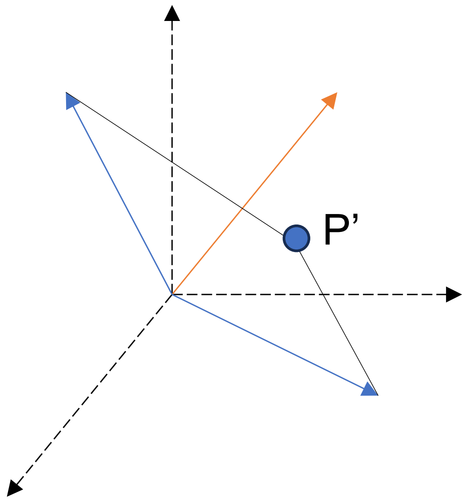


$$P = [x \quad y \quad z]$$

$$P' = [x \quad y \quad z] + [t_1 \quad t_2 \quad t_3] = P + t$$

Поворот системы координат

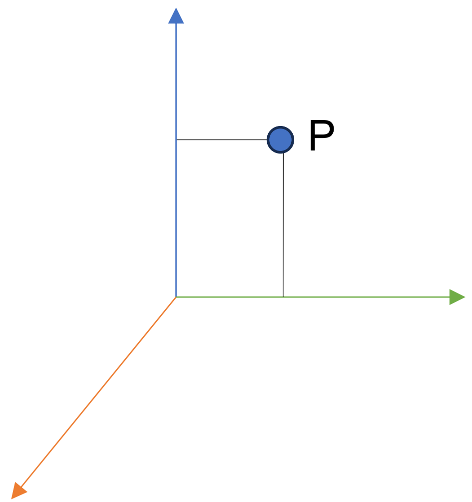


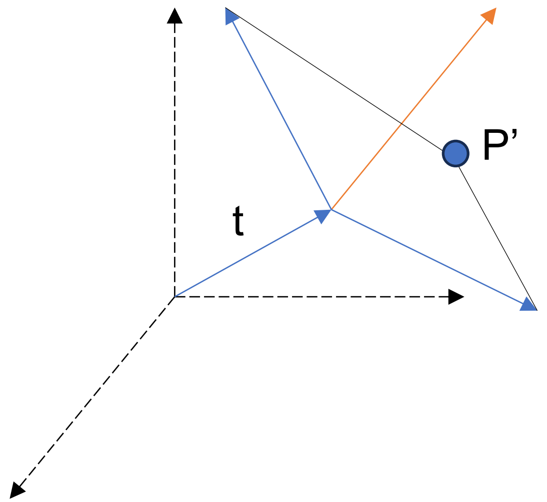


$$P = \begin{bmatrix} x & y & z \end{bmatrix}$$

$$P' = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = RP$$

Трансформация системы координат



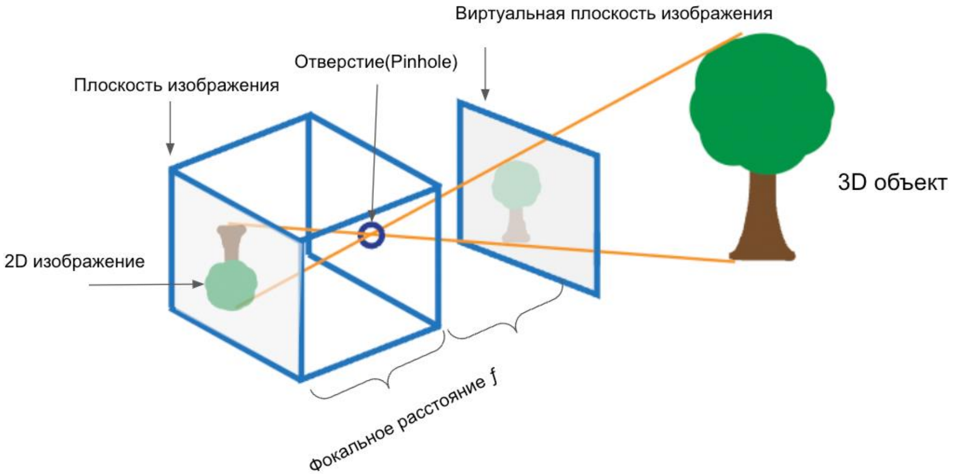


$$P = [x \quad y \quad z]$$

$$P' = RP + t$$

$$P' = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P \\ 1 \end{bmatrix}$$

Модель камеры-обскуры



перекос γ между осями x и y датчика камеры

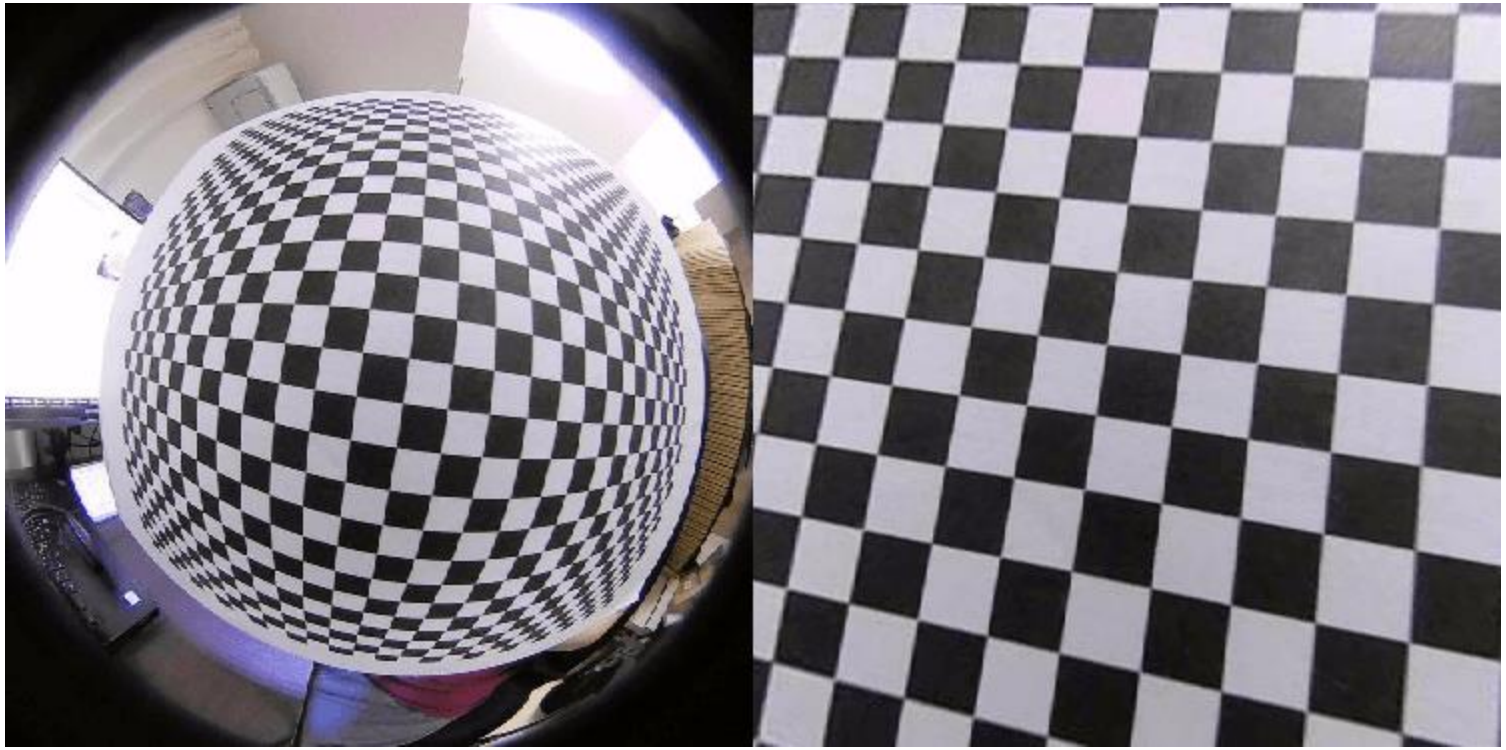
$$K = \begin{pmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} f_x & \gamma & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$\underbrace{\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}}_{\text{3D точка}} \rightarrow \underbrace{\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}}_{\text{3D точка в однородных координатах}} \rightarrow \underbrace{\begin{pmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{Проецирование}} \underbrace{\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}}_{\text{3D точка в системе координат камеры}} = \overbrace{\begin{pmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix}}^{\text{intrinsic matrix}}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \overbrace{\underbrace{\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}}_{\text{Трансформация между системами координат}}}^{\text{extrinsic matrix}} \underbrace{\begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix}}_{\text{3D точка в любой системе координат}} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} u \\ v \end{pmatrix}$$

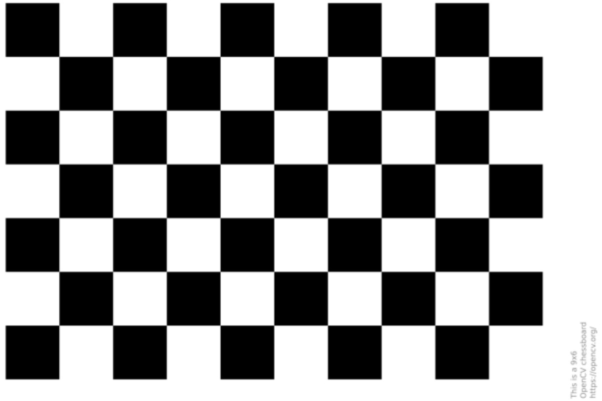
[Интерактивное демо]

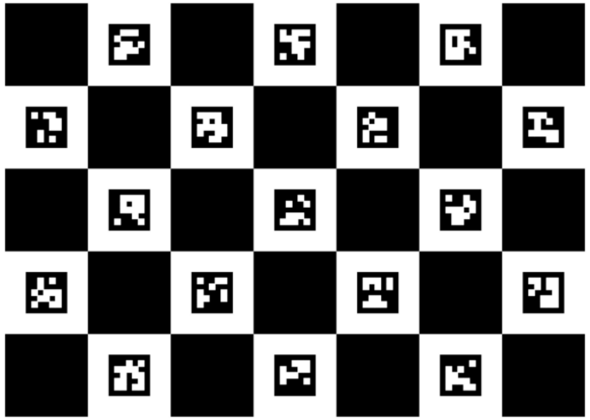
$$\mathrm{x} = \mathrm{PX}$$
$$\mathrm{x} = K[I|0][R|t]\mathrm{X}$$

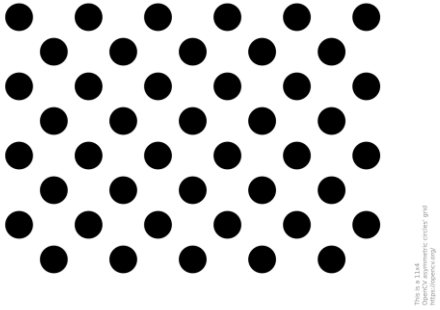
Калибровка

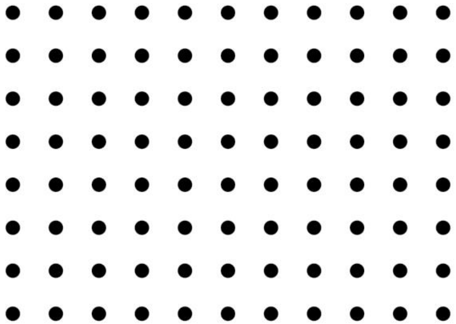


Калибровка







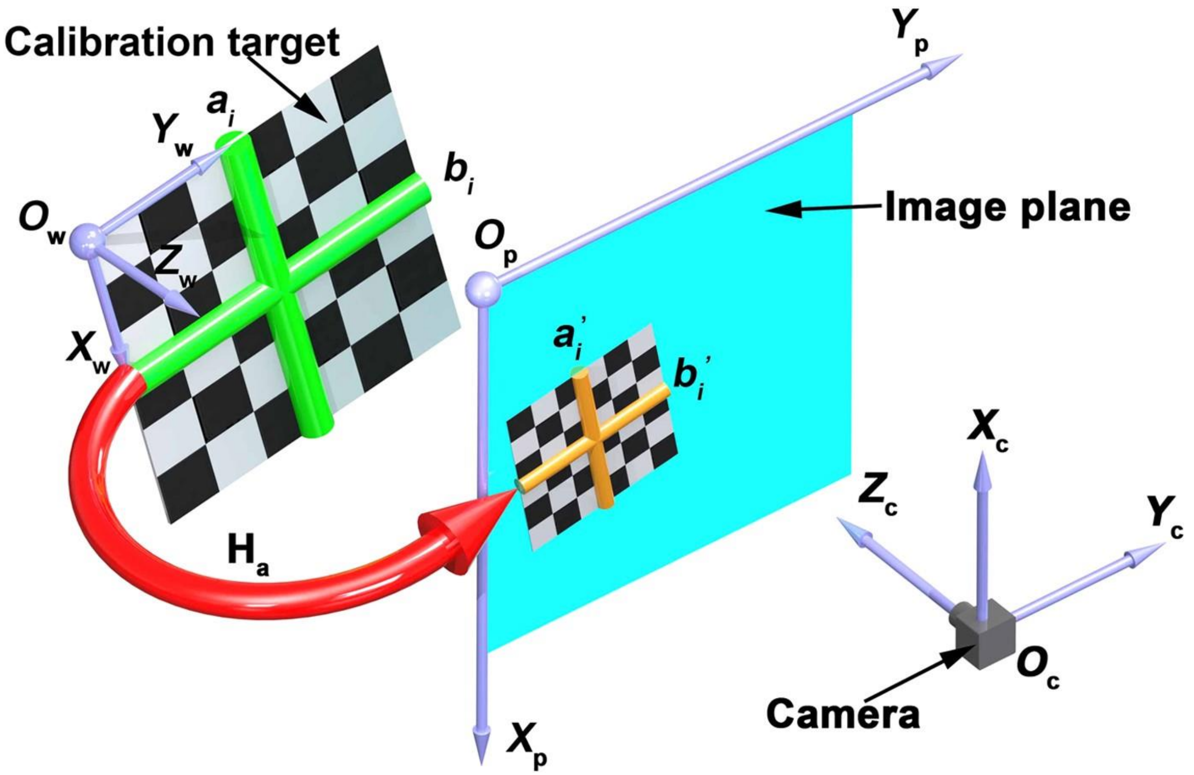


Определение параметров камеры по снимкам

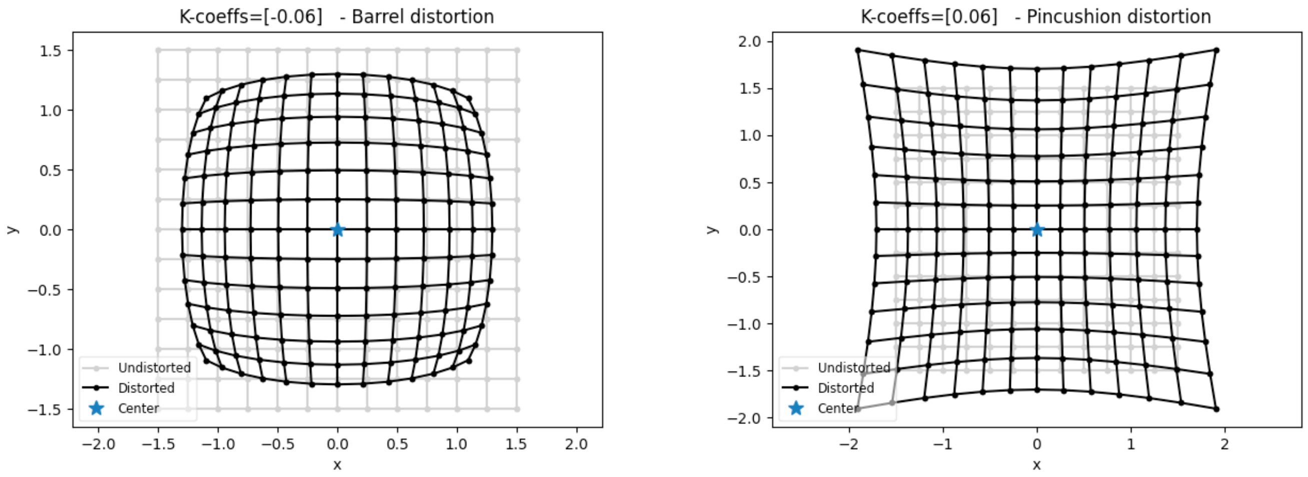
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2D точки с паттерна на изображении, находятся простыми алгоритмами

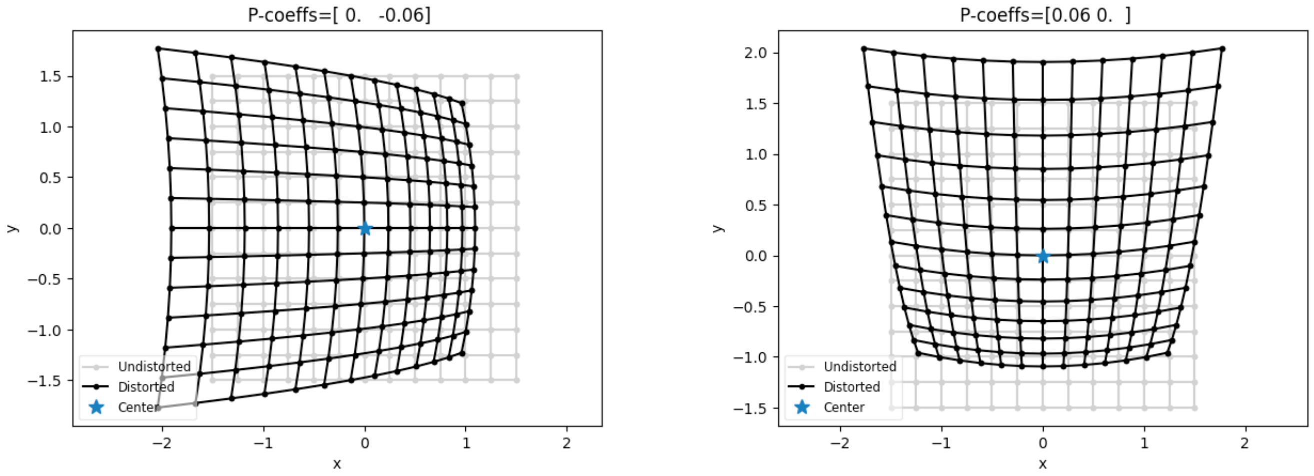
Координаты маркеров, известны заранее



Дисторсия



Радиальная дисторсия



Тангенциальная дисторсия

Добавление дисторсии в модель

$$\begin{cases} u_{dist} = f_1(u, v) \\ v_{dist} = f_2(u, v) \end{cases}$$

Общий вид дисторсии

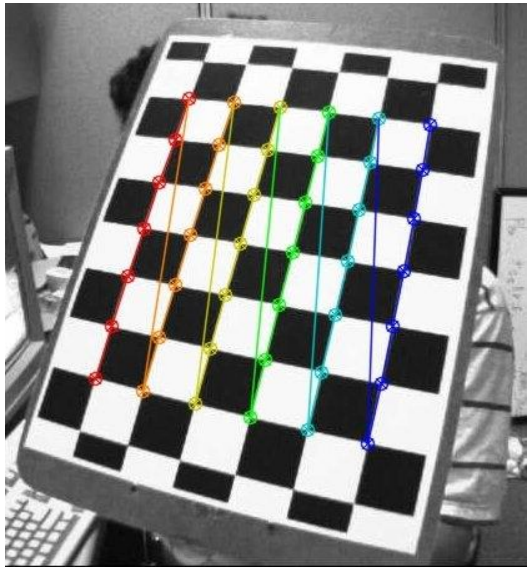
$$\begin{cases} u_{dist} = u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ v_{dist} = v(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases}, \quad r = \sqrt{u^2 + v^2}$$

Радиальная дисторсия

$$\begin{cases} u_{dist} = u + \left(2p_1 uv + p_2(r^2 + 2u^2)\right) \\ v_{dist} = v + (p_1(r^2 + 2v^2) + 2p_2 uv) \end{cases}, \quad r = \sqrt{u^2 + v^2}$$

Тангенциальная дисторсия

Калибровка в OpenCV



cv2.findChessboardCorners(…) →

cv2.calibrateCamera(…) → K, distortion, R, t

cv2.initUndistortRectifyMap(…) →

Обратная проекция

Было $\begin{pmatrix} X & Y & Z \end{pmatrix} \rightarrow \begin{pmatrix} u & v \end{pmatrix}$

Хотим $\begin{pmatrix} u & v \end{pmatrix} \rightarrow \begin{pmatrix} X & Y & Z \end{pmatrix}$

$$\begin{pmatrix} u \\ v \end{pmatrix} \rightarrow \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} \frac{1}{f_x} & 0 & -\frac{p_x}{f_x} \\ 0 & \frac{1}{f_y} & -\frac{p_y}{f_y} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{u}{f_x} - \frac{p_x}{f_x} \\ \frac{v}{f_y} - \frac{p_y}{f_y} \\ 1 \end{pmatrix} = \begin{pmatrix} Z\left(\frac{u}{f_x} - \frac{p_x}{f_x}\right) \\ Z\left(\frac{v}{f_y} - \frac{p_y}{f_y}\right) \\ Z \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Координаты пикселя	Однородные координаты пикселя	3D Обратная матрица к K	Луч в однородных координатах	Глубину Z нужно откуда-то взять	Искомая 3D точка

Гомография

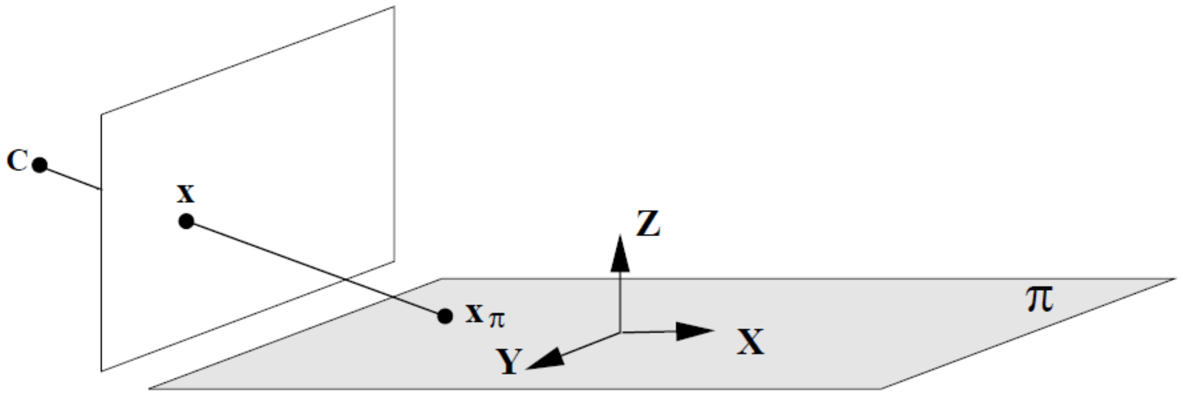
$$\begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

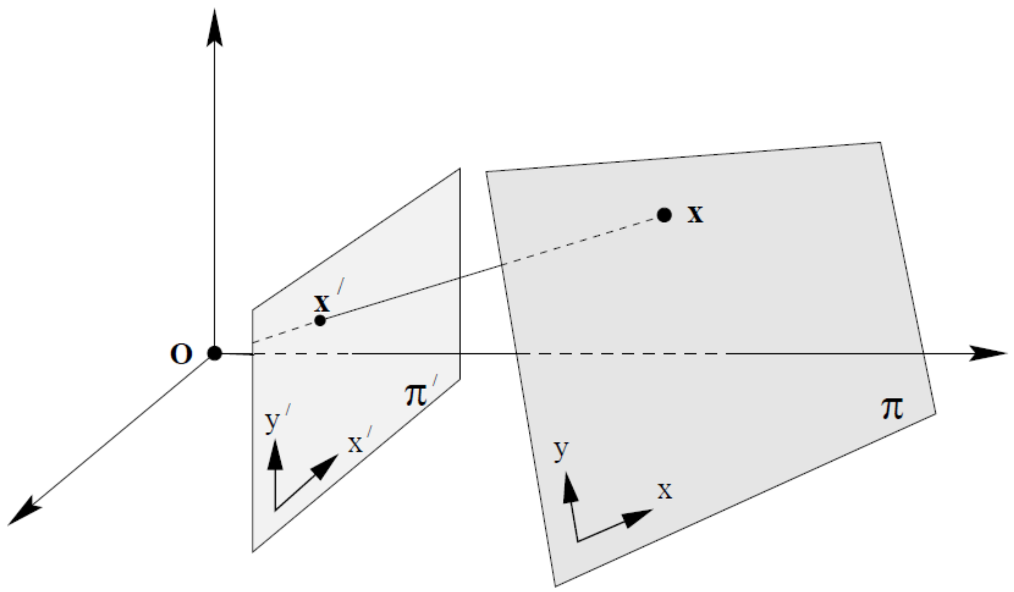
Гомография – это несингулярное отображение одной проективной плоскости в другую

Важно, что эта матрица определена в точности до множителя, так как однородные матрицы инварианты к умножению на скаляр. То есть, эта матрица определена 8 числами, а не 9. У такой матрицы 8 степеней свободы (degrees of freedom – dof).

Довольно часто проективную трансформацию называют гомография.

Гомография

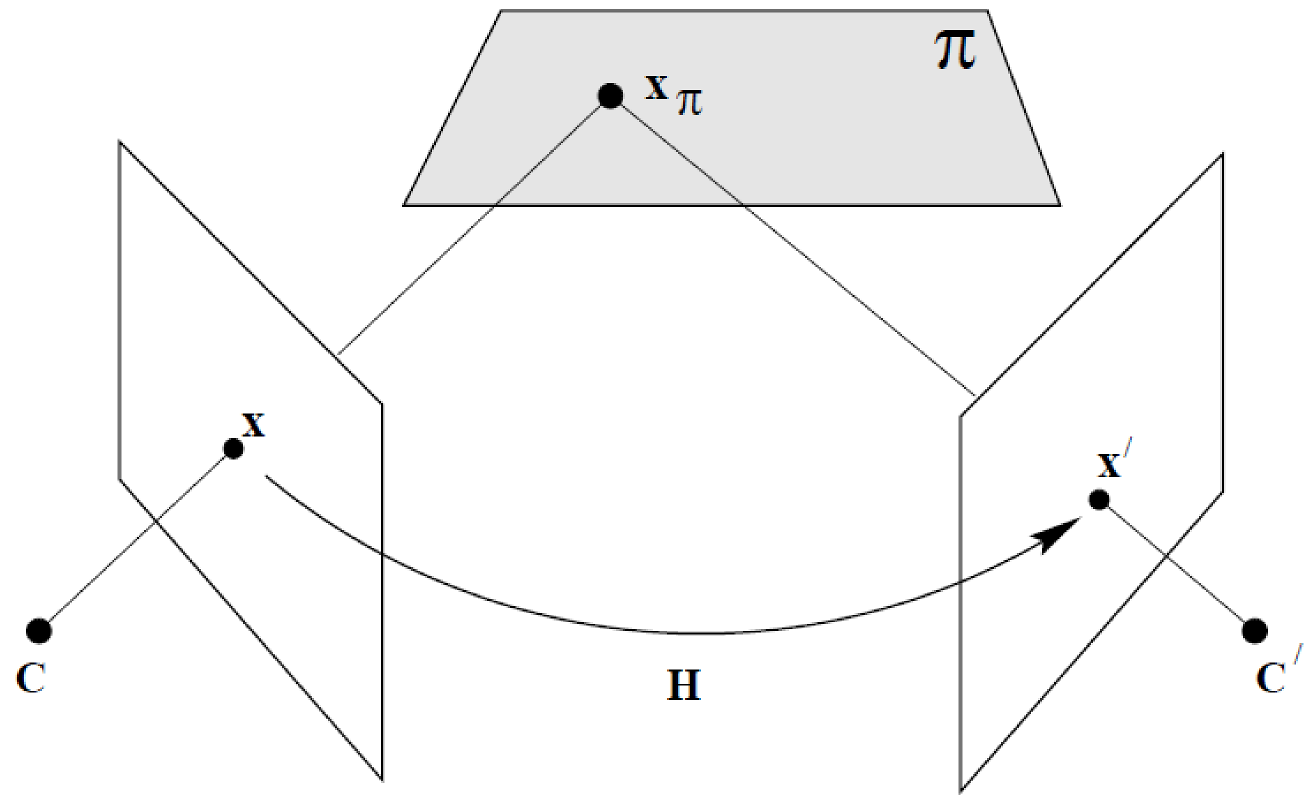




$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

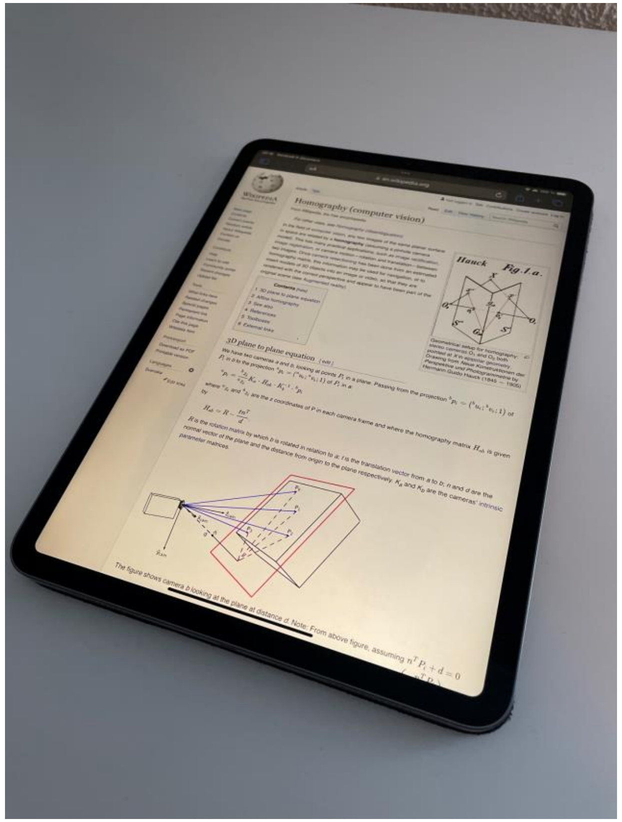
$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

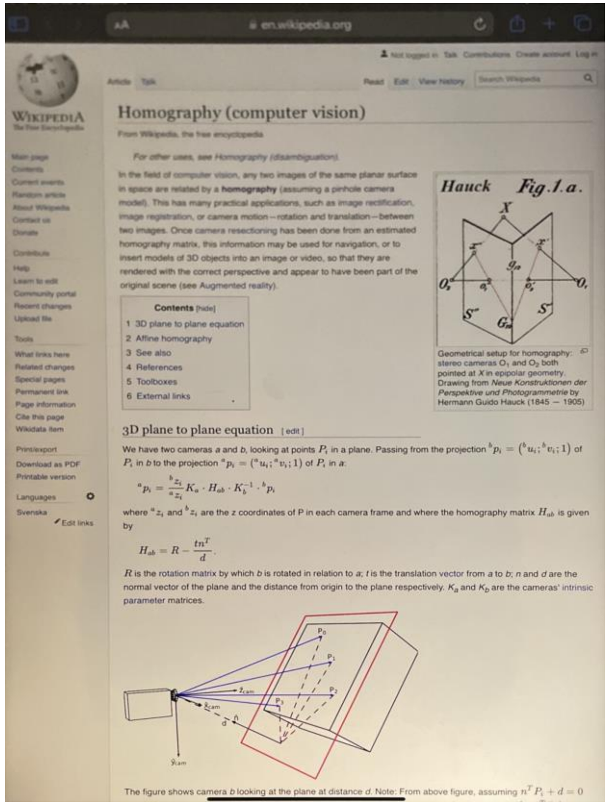
Плоская сцена



Исправление искажения изображения

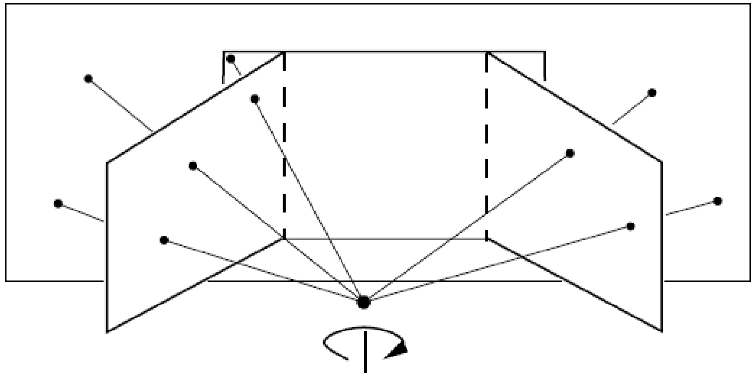
Исправление искажения изображения



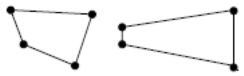
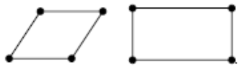
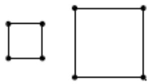
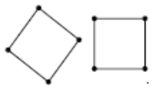


https://github.com/tprigent/image-homography

Построение панорам



Простейшие случаи гомографии

Группа	Матрица	Пример	Свойства
Проективное (projective) 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Прямая отображается в прямую
Аффинное (affine) 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Сохраняет параллельность
Преобразование подобия (similarity) 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Сдвигает, поворачивает и масштабирует объекты
Евклидово (euclidean) 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Только сдвигает и поворачивает объекты

The Direct Linear Transformation Algorithm DLT

Подробный алгоритм нахождения матрицы гомографии H
с помощью DLT расписан в файле
10.2_2D_Alignment__DLT.pdf

Одна из реализаций на github:
https://github.com/tprigent/image-homography

Использованные материалы

- Курс 3DCV от DeepSchool
- Видеокурс Computer Vision, FAU Erlangen-Nürnberg
- Геометрия формирования изображения
- Подробное описание калибровки камеры
- Видеокурс о калибровке камер и создании панорам
- Лекции по фотограмметрии
- Лекции А. Конушина, дополнительные главы компьютерного зрения
- Материалы курса Computer Vision, Carnegie Mellon
- Hartley R., Zisserman A. Multiple view geometry in computer vision