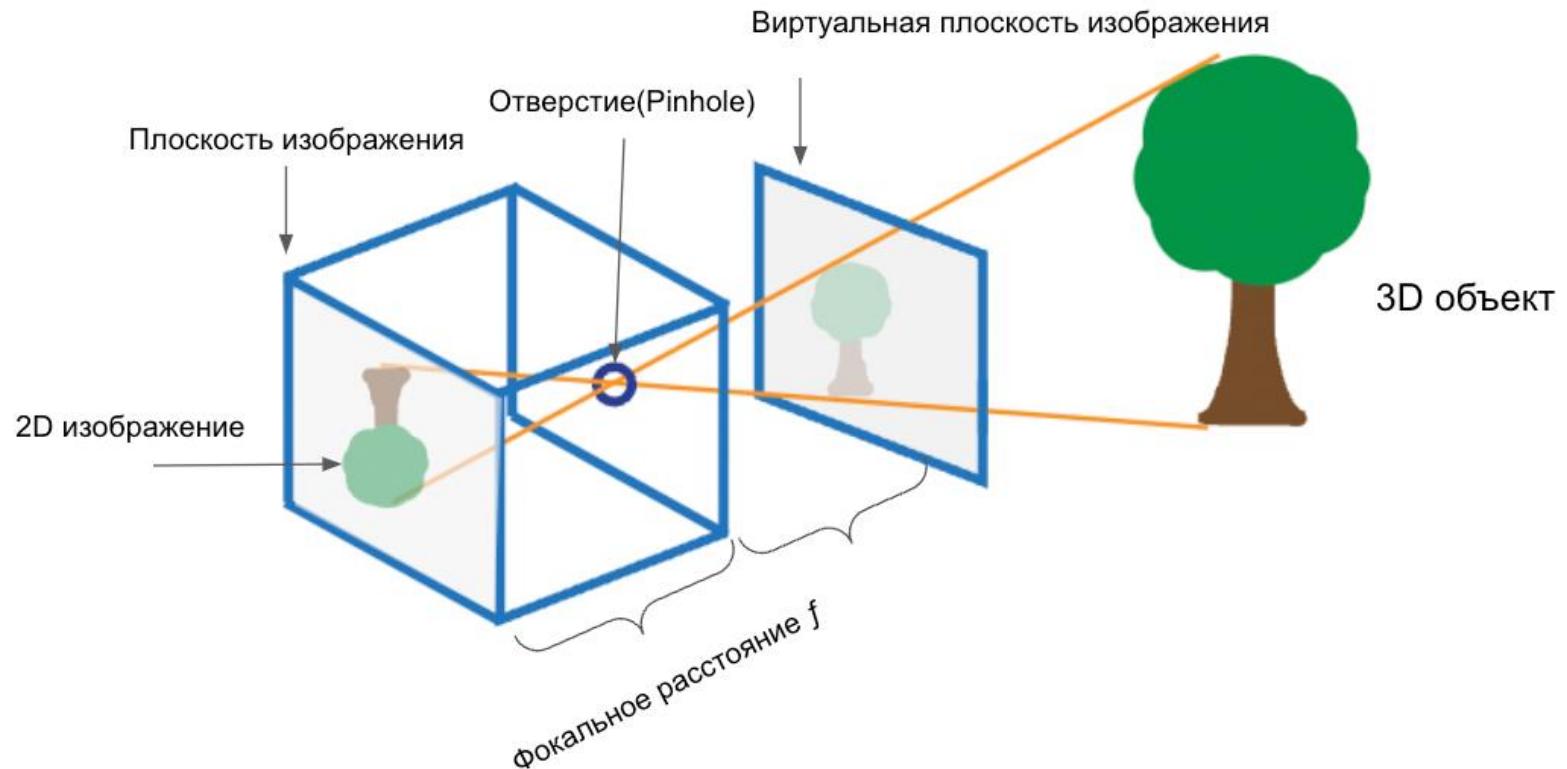
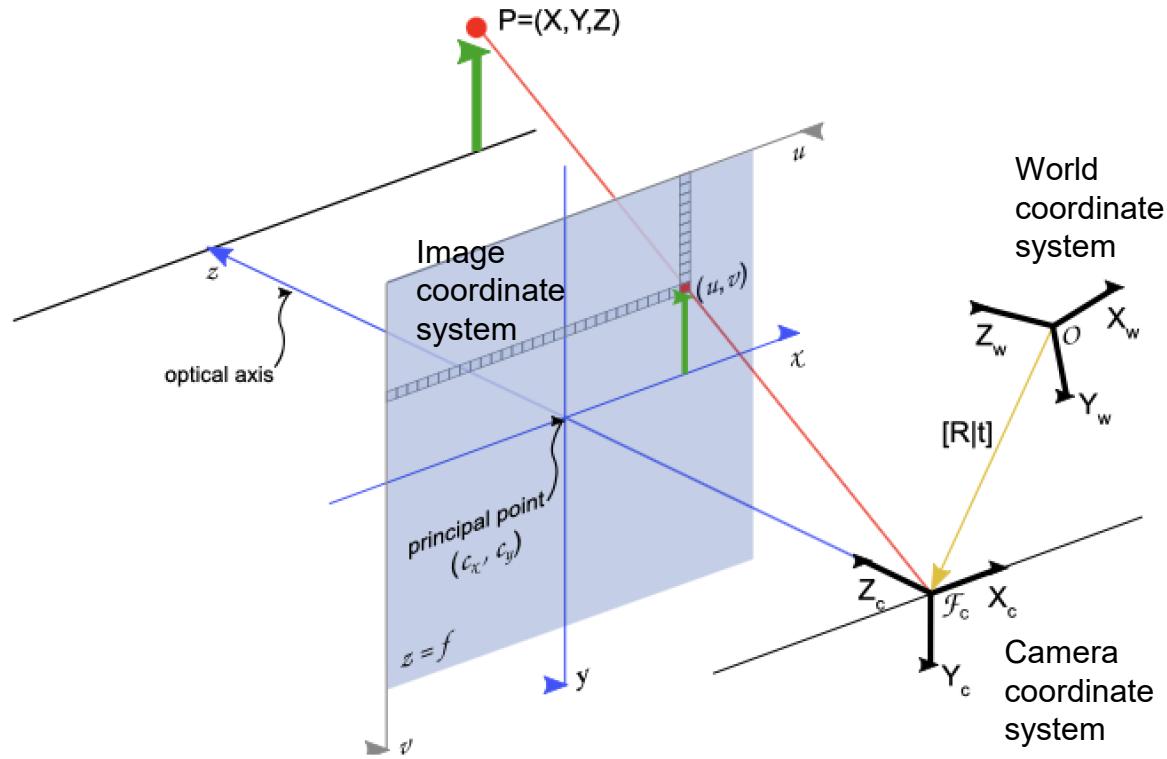


# Модель камеры, калибровка, гомография

# 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:

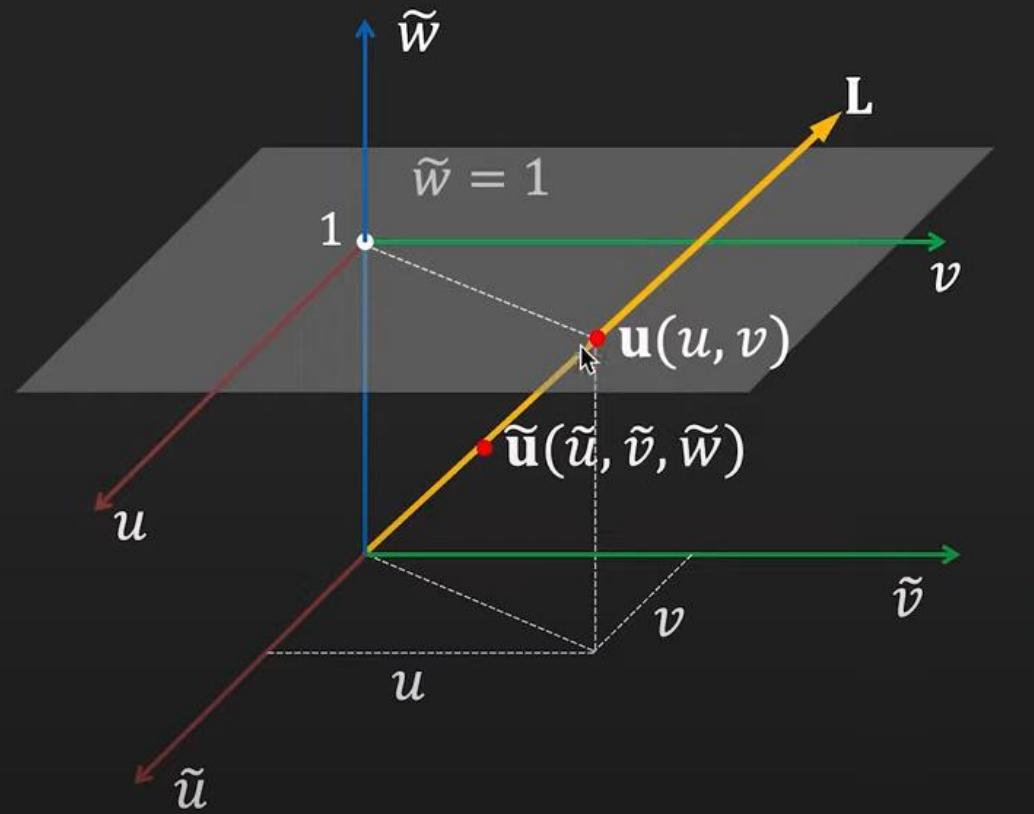
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

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

# Однородные координаты

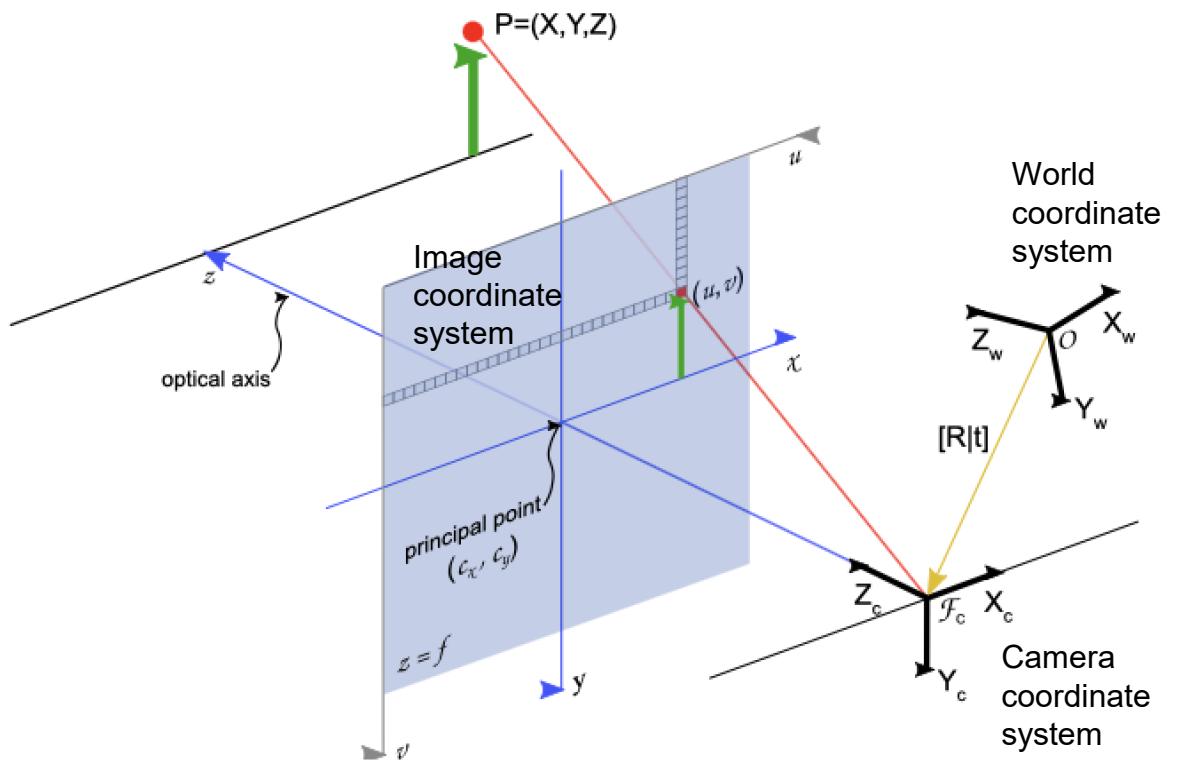
$$u = \frac{\tilde{u}}{\tilde{w}} \quad v = \frac{\tilde{v}}{\tilde{w}}$$

$$\mathbf{u} \equiv \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{w}u \\ \tilde{w}v \\ \tilde{w} \end{bmatrix} \equiv \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \tilde{\mathbf{u}}$$



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

# От пикселя к лучу

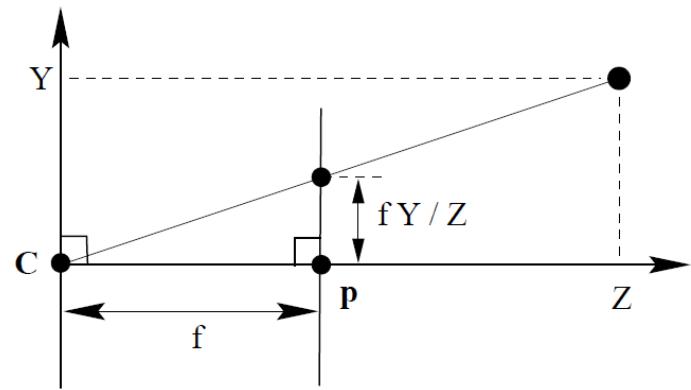


Был пиксель с координатами  $(u, v)$

Стал луч с координатами  $(u, v, 1)$

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

# Focal length



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

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

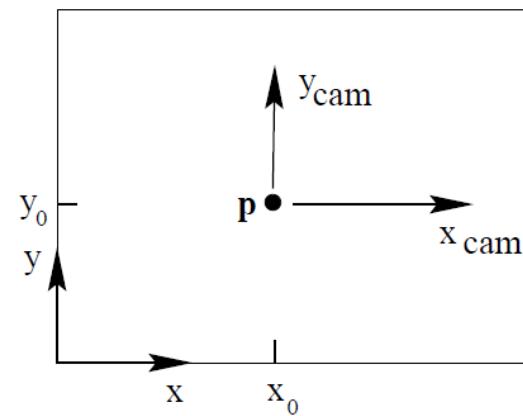
$$(u \ v) = \left( f \frac{X}{Z} \ f \frac{Y}{Z} \right)$$

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

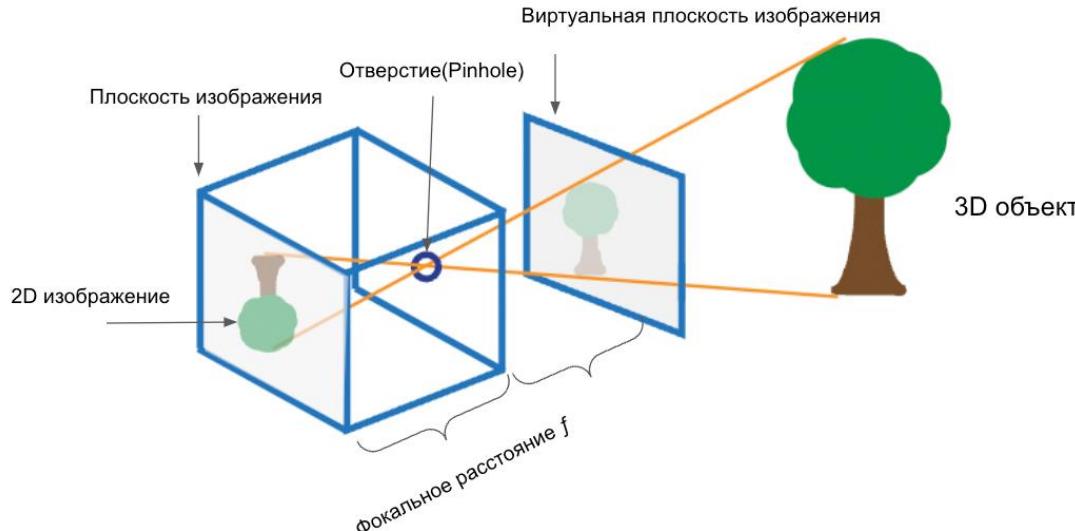
Однородные координаты  
можно умножать/делить на любое число

# Principal point

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



# Модель камеры-обскуры



$$\begin{aligned}
 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} &\rightarrow \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \xrightarrow{\text{Проектирование}} \underbrace{\begin{pmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{3D точка в однородных координатах}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} f_x X + Z p_x \\ f_y Y + Z p_y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} f_x \frac{X}{Z} + p_x \\ f_y \frac{Y}{Z} + p_y \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ v \\ 1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} u \\ v \end{pmatrix}
 \end{aligned}$$

3D точка в однородных координатах

Проектирование

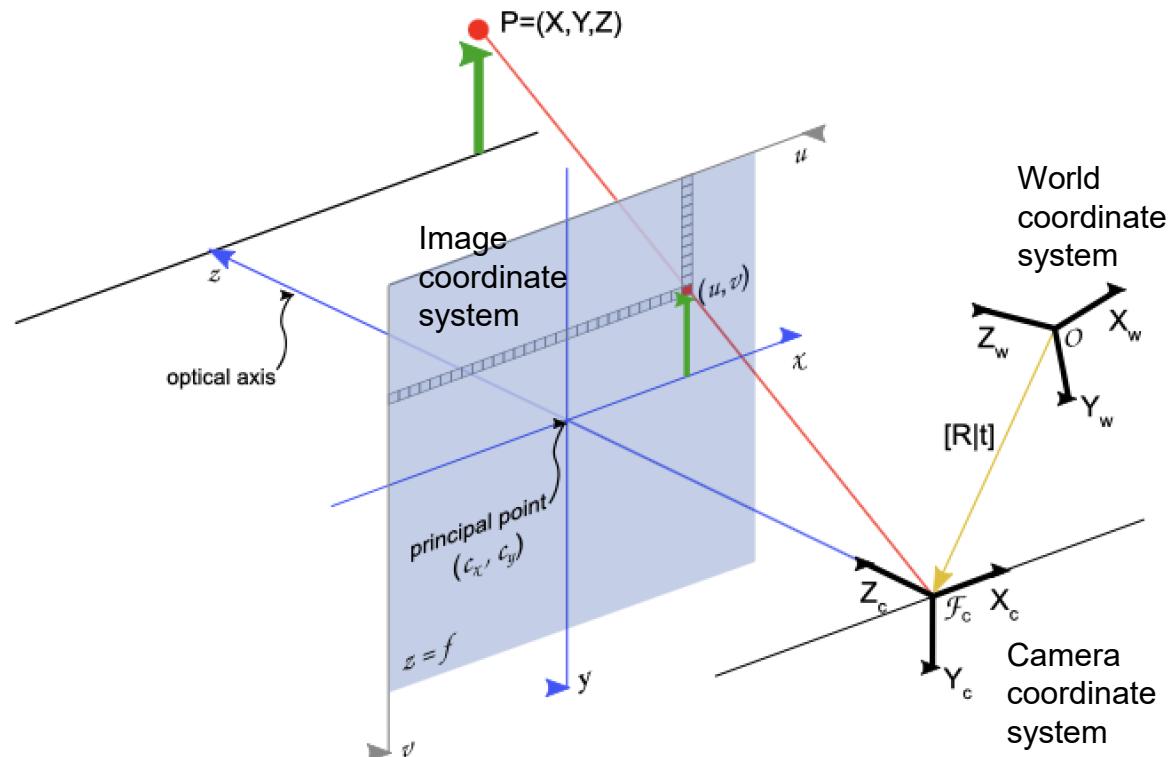
3D точка в системе координат камеры

2D точка в однородных координатах

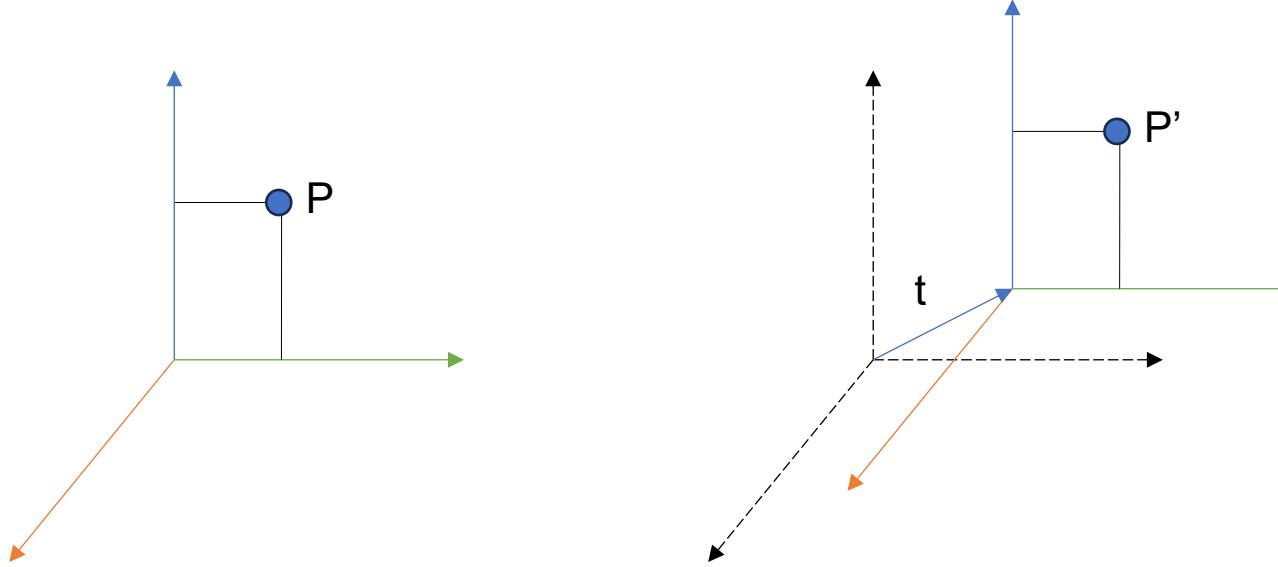
Та же 2D точка в однородных координатах

Искомый пиксель в однородных координатах

# Если 3D точка не в системе координат камеры



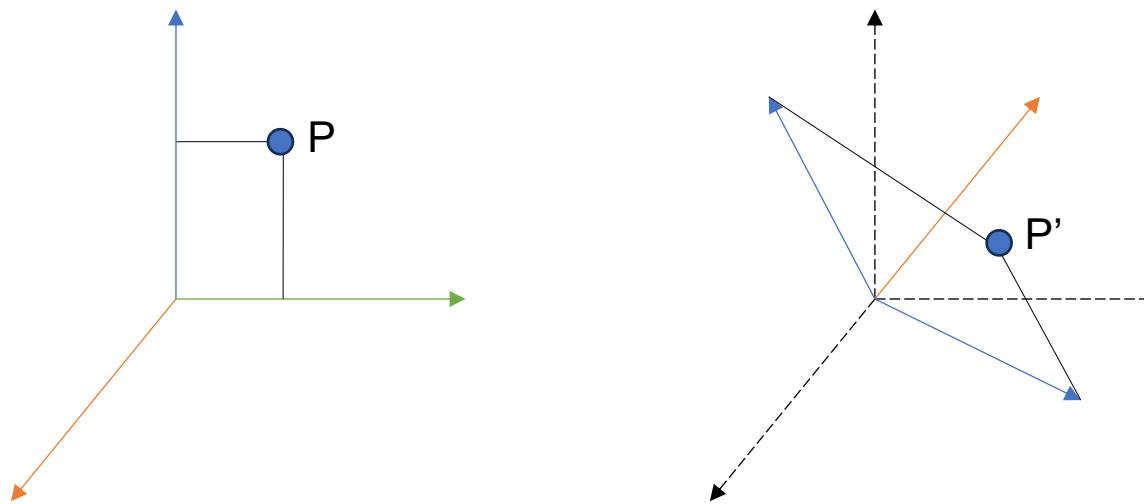
# Сдвиг системы координат



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

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

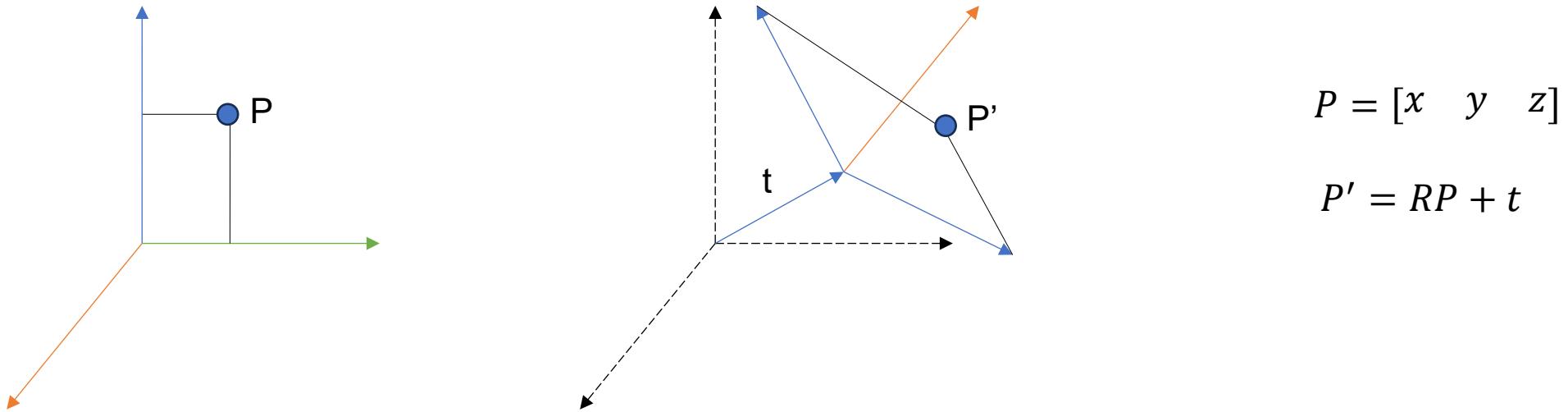
# Поворот системы координат



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

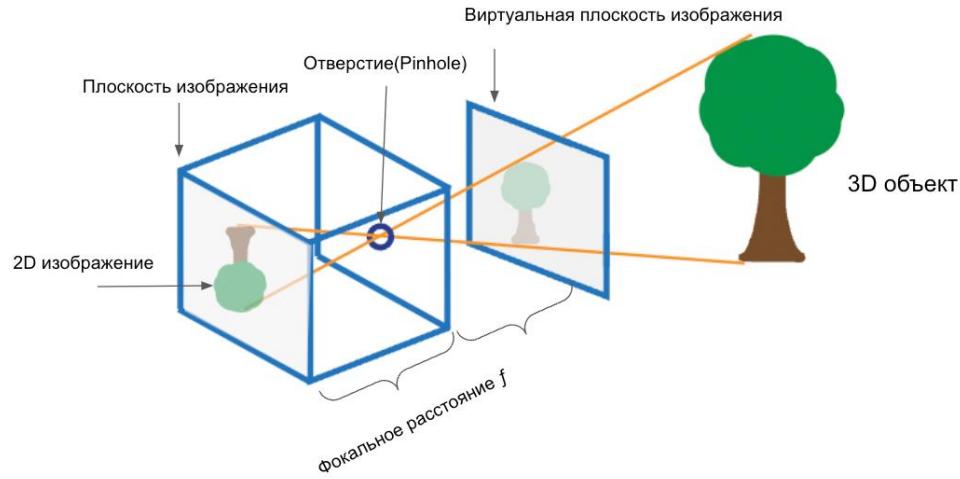
$$P' = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = RP$$

# Трансформация системы координат



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

# Модель камеры-обскуры



$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \xrightarrow{\text{Проектирование}} \begin{pmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

3D точка в однородных координатах      3D точка в однородных координатах      Проектирование      3D точка в системе координат камеры

перекос  $\gamma$  между осями  $x$  и  $y$  датчика камеры

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

intrinsic matrix      extrinsic matrix

$$\begin{pmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} u \\ v \end{pmatrix}$$

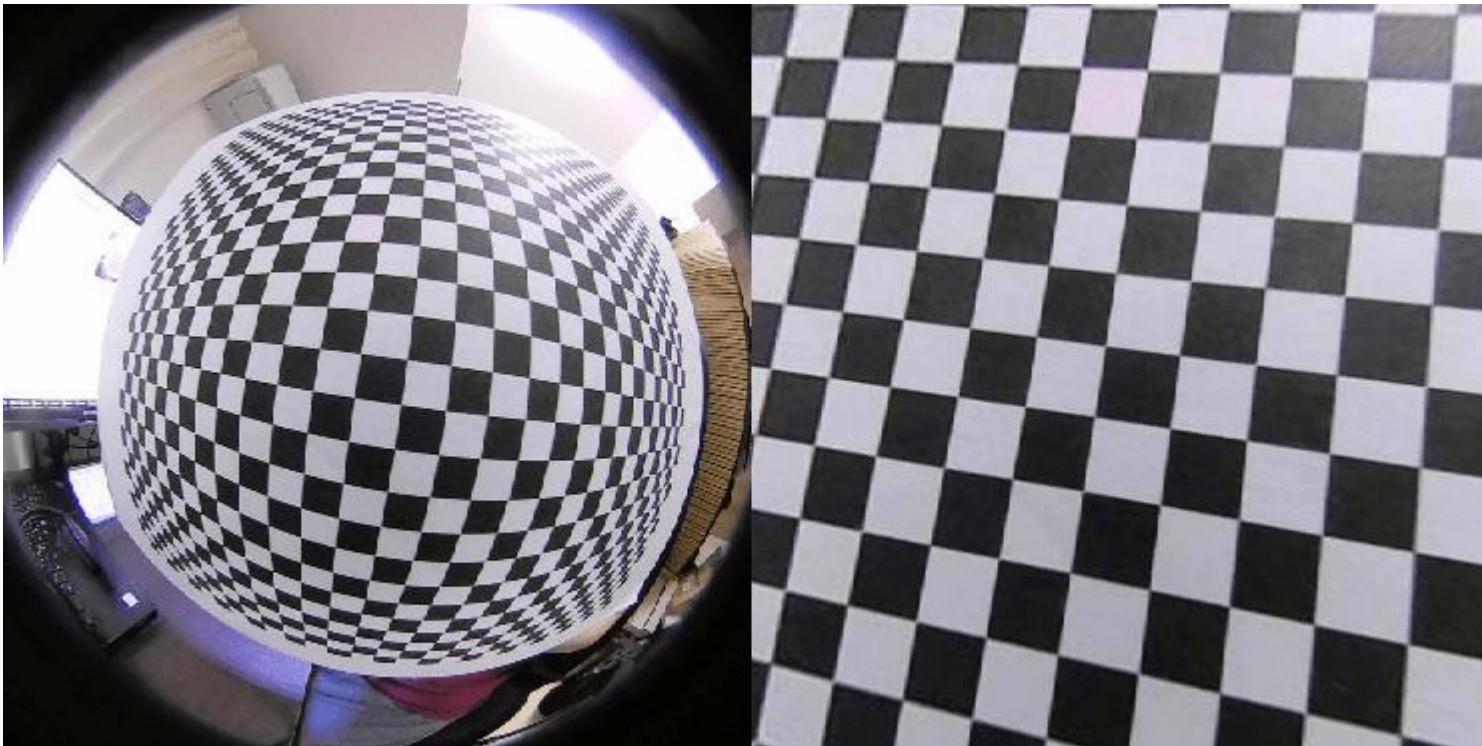
Трансформация между системами координат      3D точка в любой системе координат

$$\mathbf{x} = \mathbf{P}\mathbf{X}$$

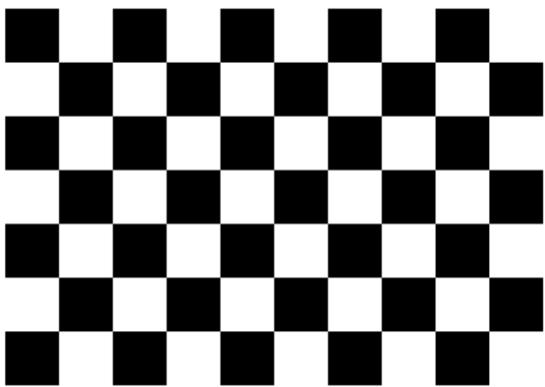
$$\mathbf{x} = K[I|0][R|t]\mathbf{X}$$

[Интерактивное демо](#)

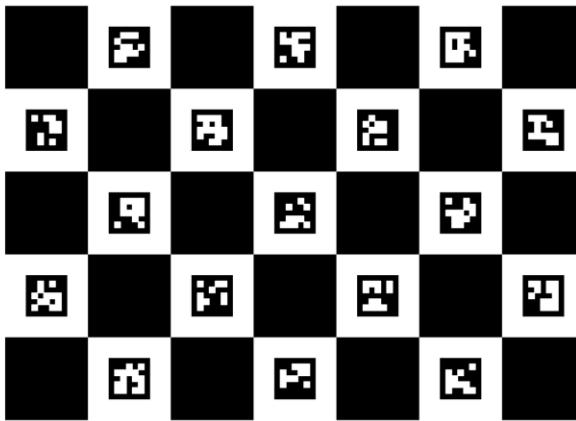
# Калибровка



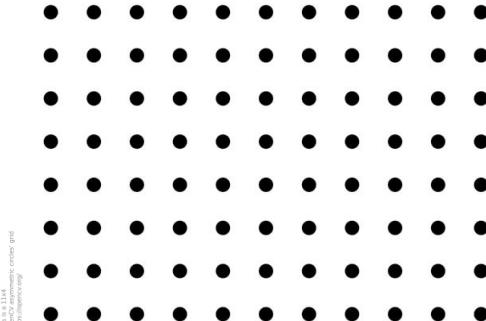
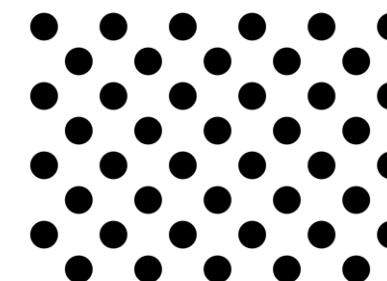
# Калибровка



This is a 9x9 OpenCV chessboard  
<https://opencv.org/>



This is a 9x9 OpenCV chessboard  
<https://opencv.org/>



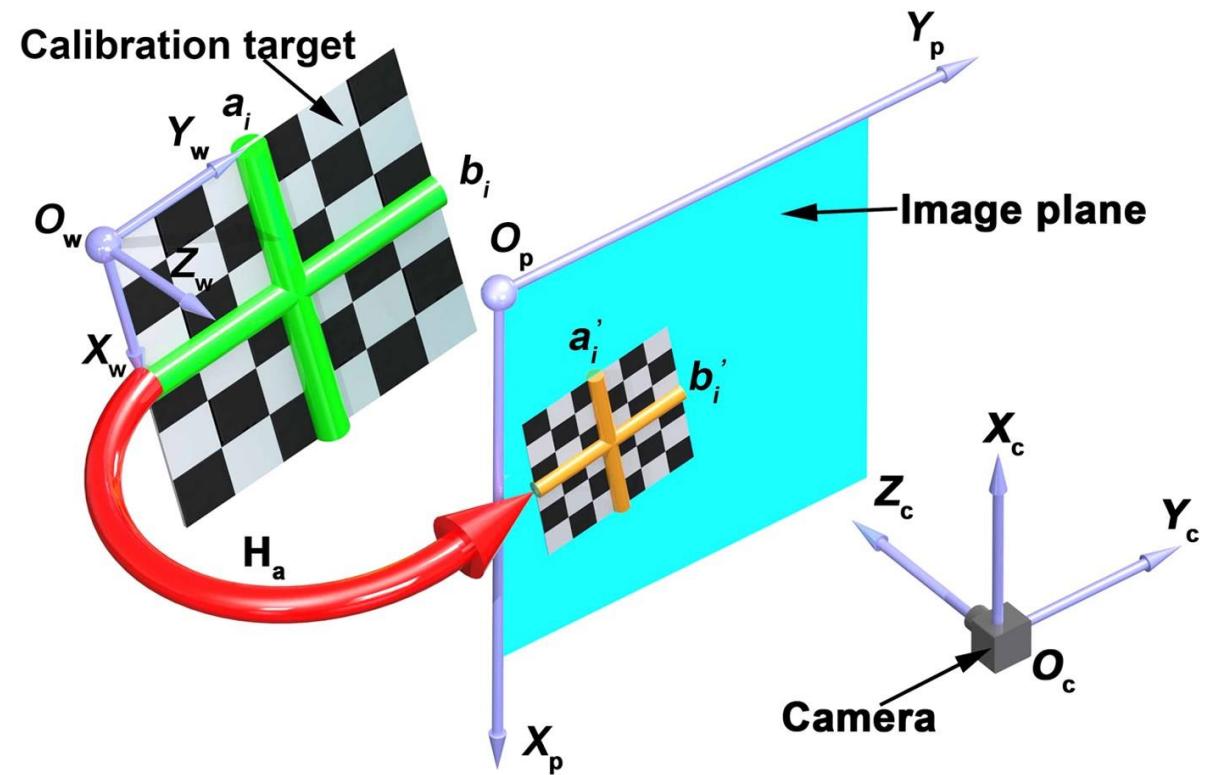
This is a 9x14 OpenCV chessboard  
<https://opencv.org/>

# Определение параметров камеры по снимкам

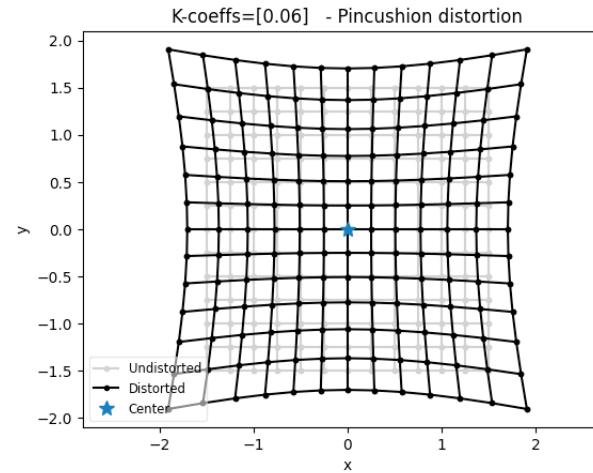
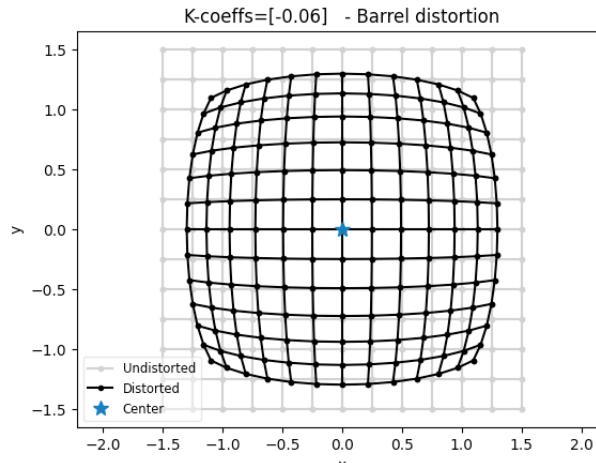
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2D точки с паттерна на изображении, находятся простыми алгоритмами

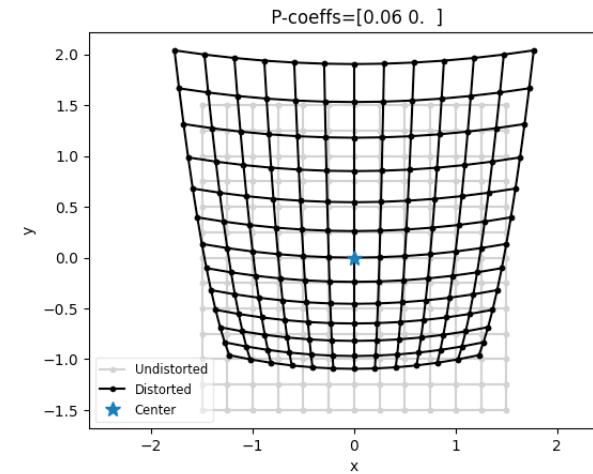
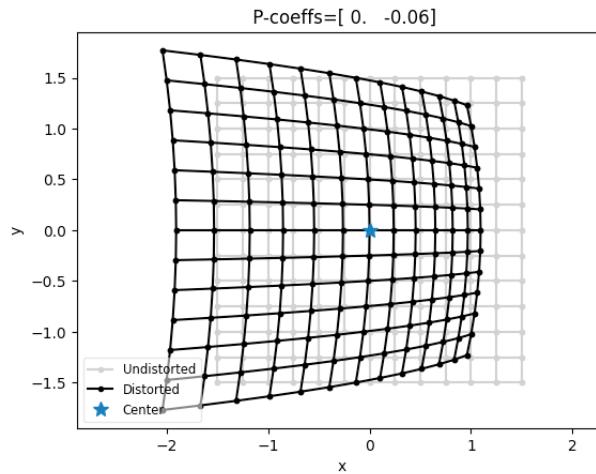
Координаты маркеров, известны заранее



# 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

$$\begin{cases} u_{dist} = u(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ v_{dist} = v(1 + k_1r^2 + k_2r^4 + k_3r^6) \end{cases}, \quad r = \sqrt{u^2 + v^2}$$

Радиальная  
дисторсия

$$\begin{cases} u_{dist} = u + (2p_1uv + p_2(r^2 + 2u^2)) \\ v_{dist} = v + (p_1(r^2 + 2v^2) + 2p_2uv)' \end{cases} \quad r = \sqrt{u^2 + v^2}$$

Тангенциальная  
дисторсия

# Калибровка в OpenCV

```
cv2.findChessboardCorners(...)
```



```
cv2.calibrateCamera(...)
```



```
K, distortion, R, t
```

```
cv2.initUndistortRectifyMap(...)
```



# Обратная проекция

Было  $(X \quad Y \quad Z) \rightarrow (u \quad v)$

Хотим  $(u \quad v) \rightarrow (X \quad Y \quad Z)$

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

Координаты  
пикселя  
Однородные  
координаты  
пикселя

3D Обратная  
матрица к K

Луч в  
однородных  
координатах

Глубину Z нужно  
откуда-то взять

Искомая  
3D точка

# Гомография

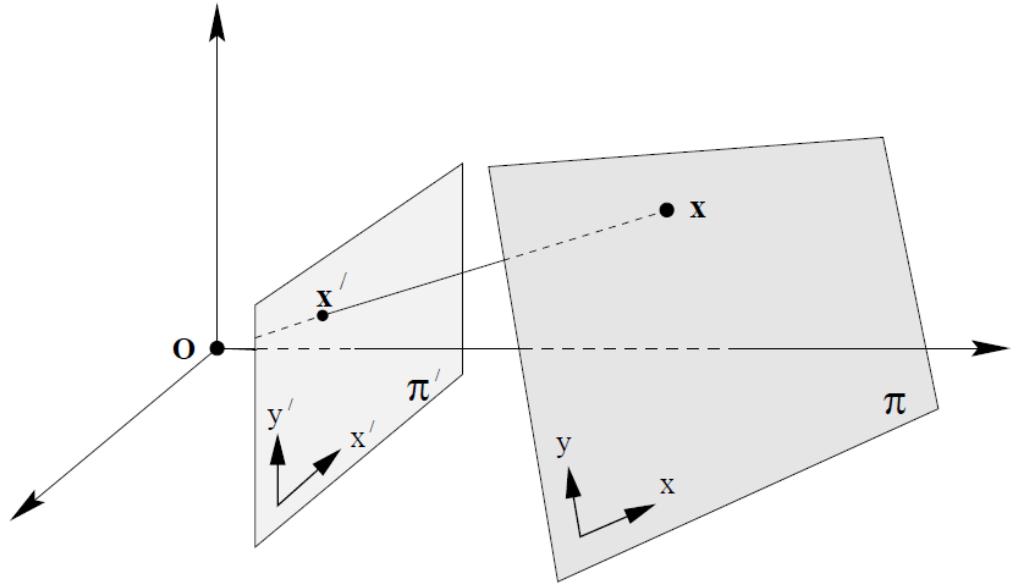
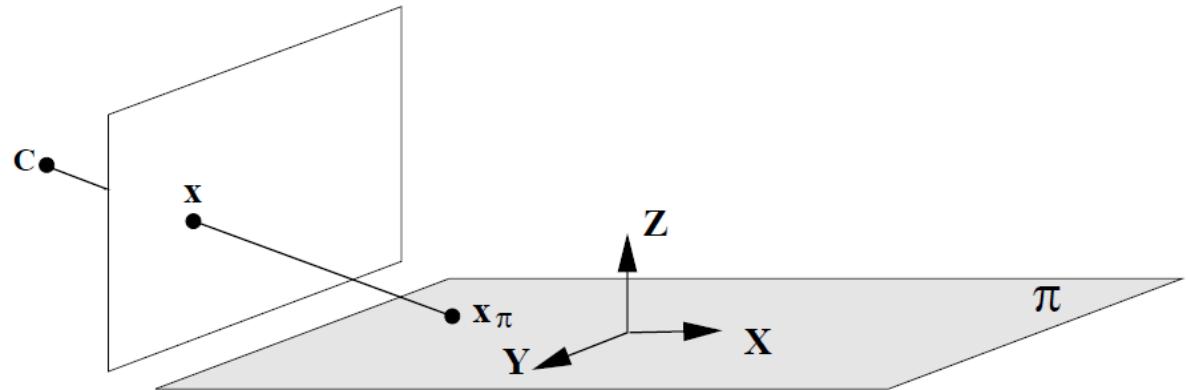
$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Гомография – это несингулярное отображение одной проективной плоскости в другую

Важно, что эта матрица определена в точности до множителя, так как однородные матрицы инвариантны 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.

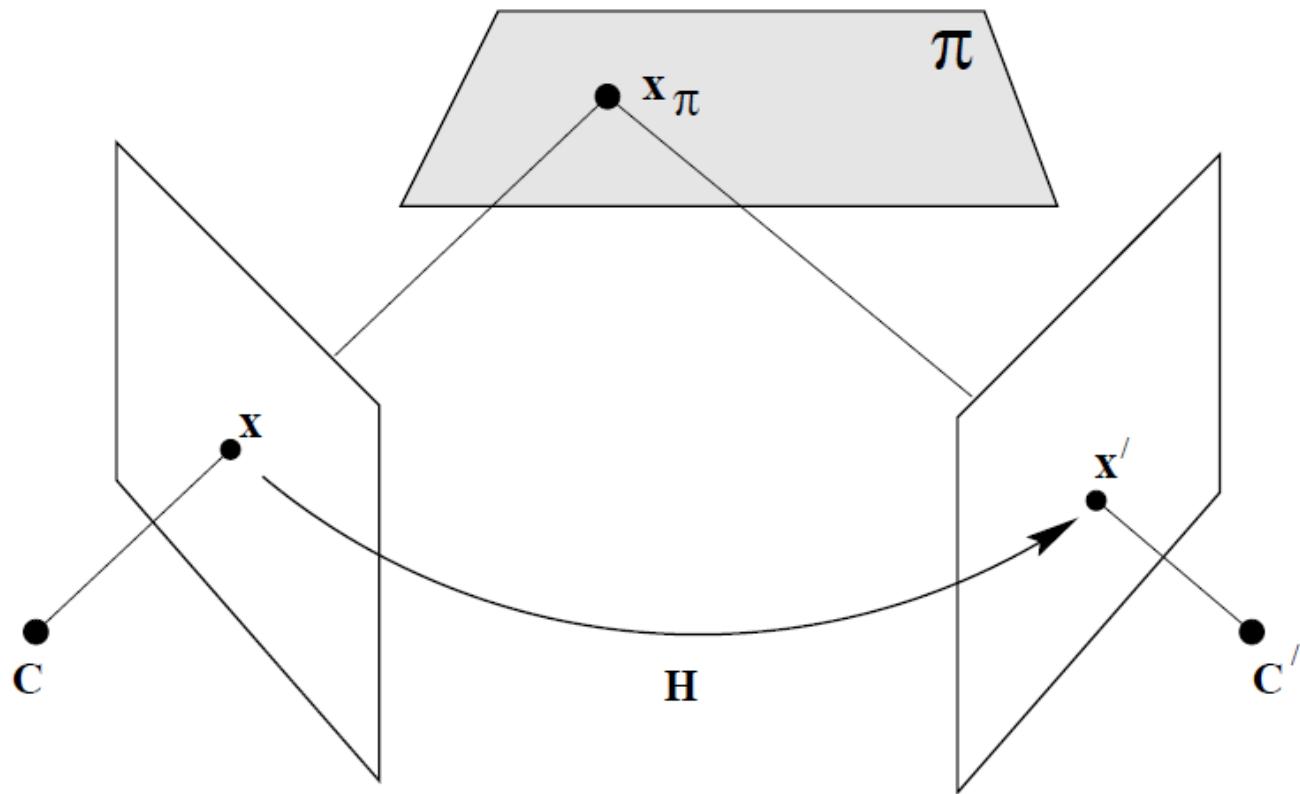
# Гомография



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

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

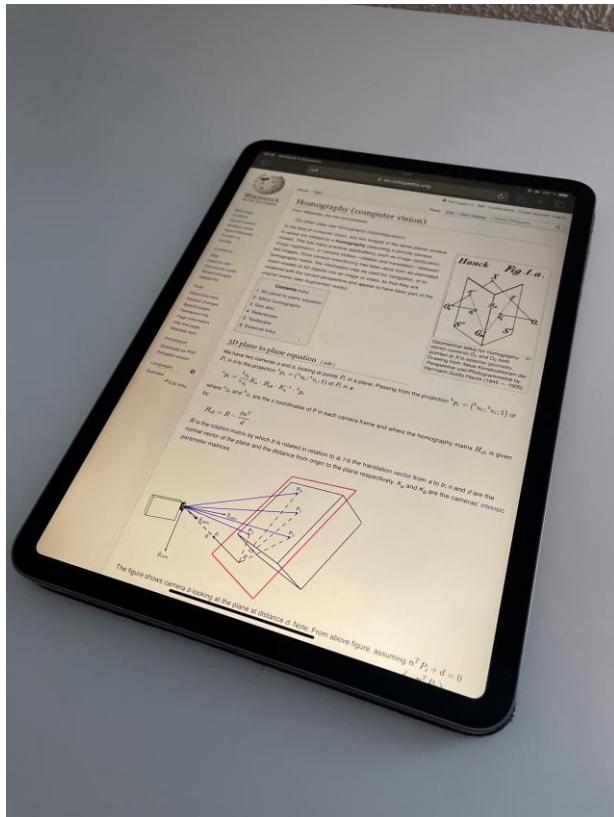
# Плоская сцена



# Исправление искажения изображения



# Исправление искажения изображения

A screenshot of the same Wikipedia page on Homography (computer vision). The left sidebar contains navigation links like "Main page", "Contents", "Current events", "Random article", "About Wikipedia", "Contact us", "Donate", "Community portal", "Help", "Learn to edit", "Community portal", "Recent changes", "Upload file", "Tools", "What links here", "Related changes", "Special pages", "Permanent link", "Page information", "Cite the page", "Wikidata item", "Print/export", "Download as PDF", "Printable version", and "Languages". The right side of the page features the "Hauck Fig. 1.a." diagrams and the "3D plane to plane equation" section. The "3D plane to plane equation" section contains the following text and formula:

In the field of computer vision, any two images of the same planar surface in space are related by a **homography** (assuming a pinhole camera model). This has many practical applications, such as image rectification, image registration, camera motion estimation, rotation and translation – between two images. Once camera rectification has been done from an estimated homography matrix, this information may be used for navigation, or to insert models of 3D objects into an image or video, so that they are rendered with the correct perspective and appear to have been part of the original scene (see Augmented reality).

**Contents** [hide]

- 1 3D plane to plane equation
- 2 Affine homography
- 3 See also
- 4 References
- 5 Toolbars
- 6 External links

**3D plane to plane equation** [edit]

We have two cameras  $a$  and  $b$ , looking at points  $P_i$  in a plane. Passing from the projection  ${}^b p_i = ({}^b u_i; {}^b v_i; 1)$  of  $P_i$  in  $b$  to the projection  ${}^a p_i = ({}^a u_i; {}^a v_i; 1)$  of  $P_i$  in  $a$ :

$${}^a p_i = \frac{{}^b z_i}{{}^a z_i} K_a \cdot H_{ab} \cdot K_b^{-1} \cdot {}^b p_i$$

where  ${}^a z_i$  and  ${}^b z_i$  are the  $z$  coordinates of  $P_i$  in each camera frame and where the homography matrix  $H_{ab}$  is given by

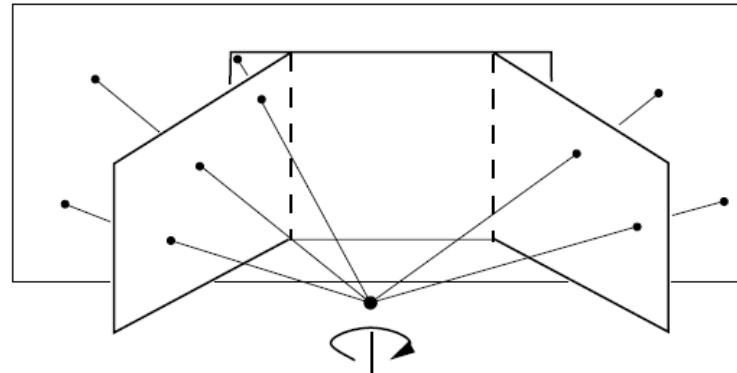
$$H_{ab} = R - \frac{tn^T}{d}$$

$R$  is the rotation matrix by which  $b$  is rotated in relation to  $a$ ;  $t$  is the translation vector from  $a$  to  $b$ ;  $n$  and  $d$  are the normal vector of the plane and the distance from origin to the plane respectively.  $K_a$  and  $K_b$  are the cameras' intrinsic parameter matrices.

The figure shows camera  $b$  looking at the plane at distance  $d$ . Note: From above figure, assuming  $n^T P_1 + d = 0$

<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