Pinhole модель камеры

Шишков Александр Валерьевич

1.

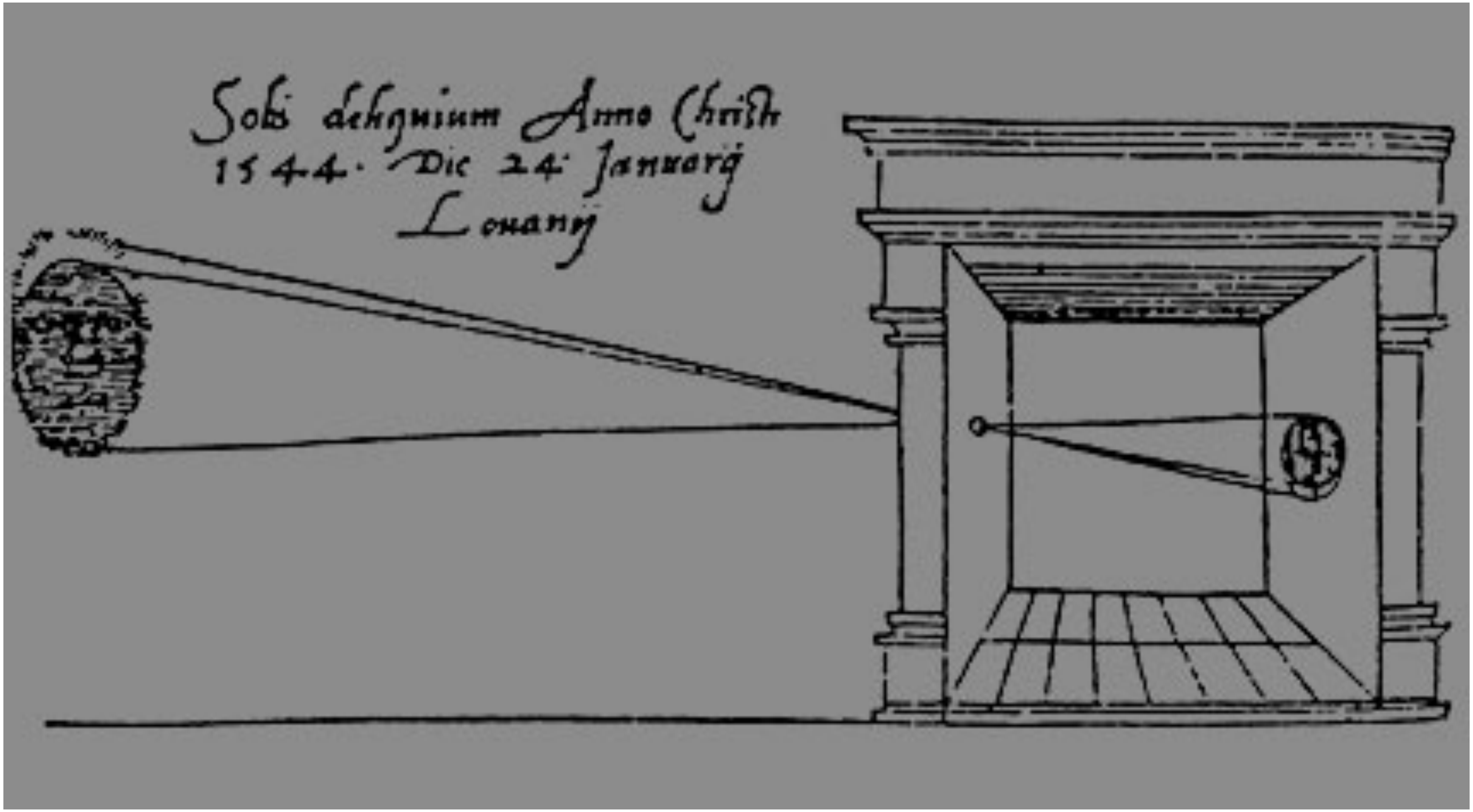
a. $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},\ AB = ?$

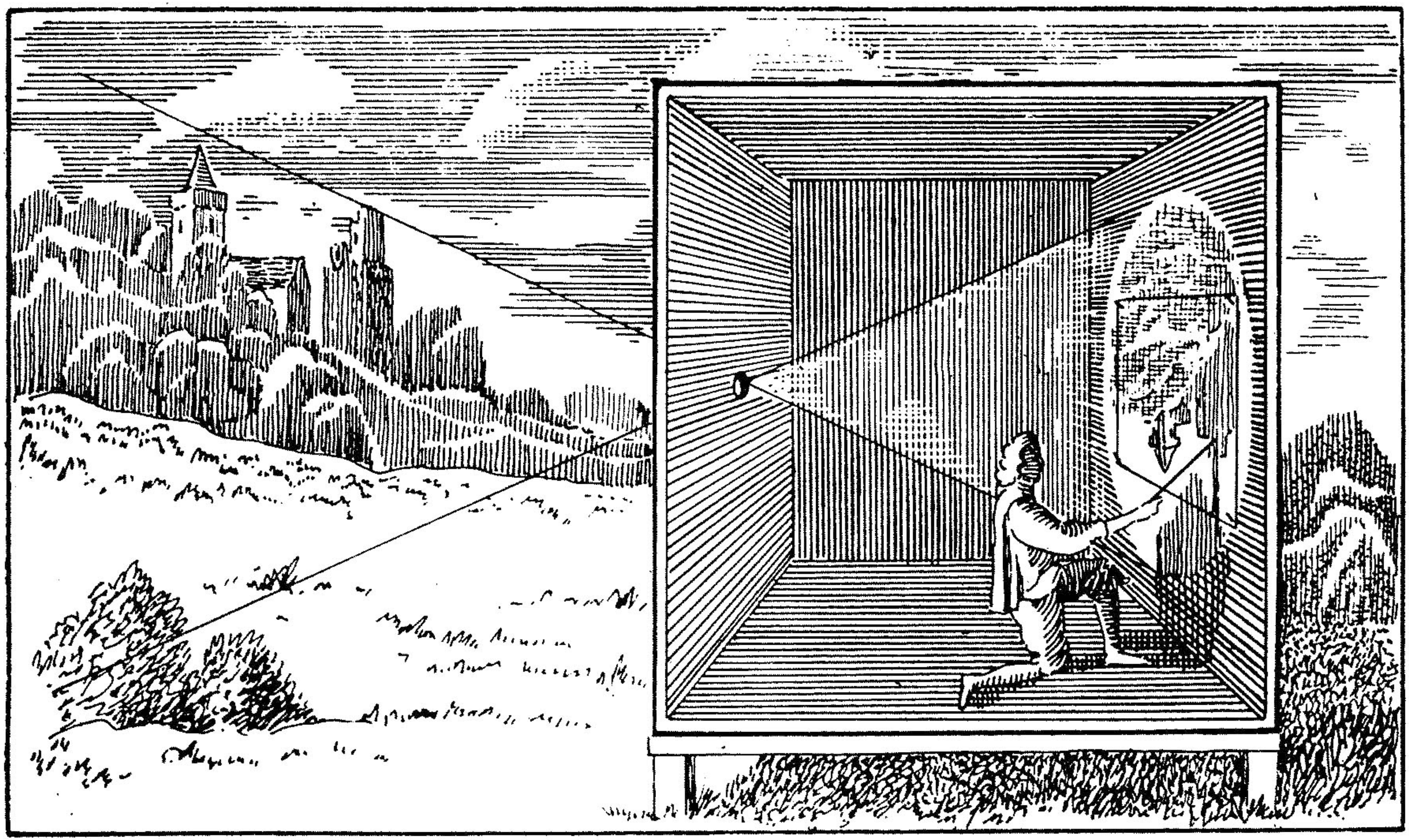
b. $A = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},\ AB = ?$

2. A, U, D и V – матрицы размера nXn, $A = UDV^T, UU^T = E,\ VV^T = E, D = diag(\sigma_1, \sigma_2, \dots, \sigma_n), \sigma_1 > \sigma_2 > \cdots > \sigma_n > 0$. Найти $\min_{|X|=1} |AX|$ и значение X, при котором минимум достигается

3. A, U, D и V – матрицы размера nXn, $A = UDV^T, UU^T = E,\ VV^T = E, D$ -- диагональная. Найти $\operatorname{argmin}_{BB^T=E} |A - B|$.

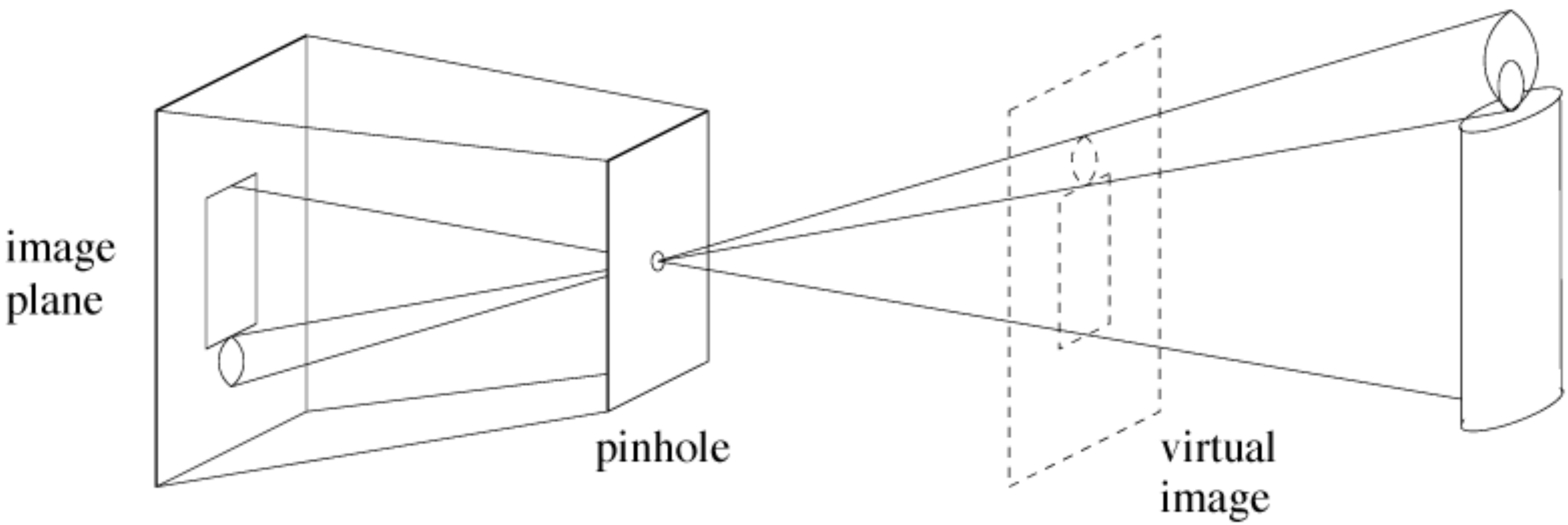
Камера-обскура





Pinhole camera

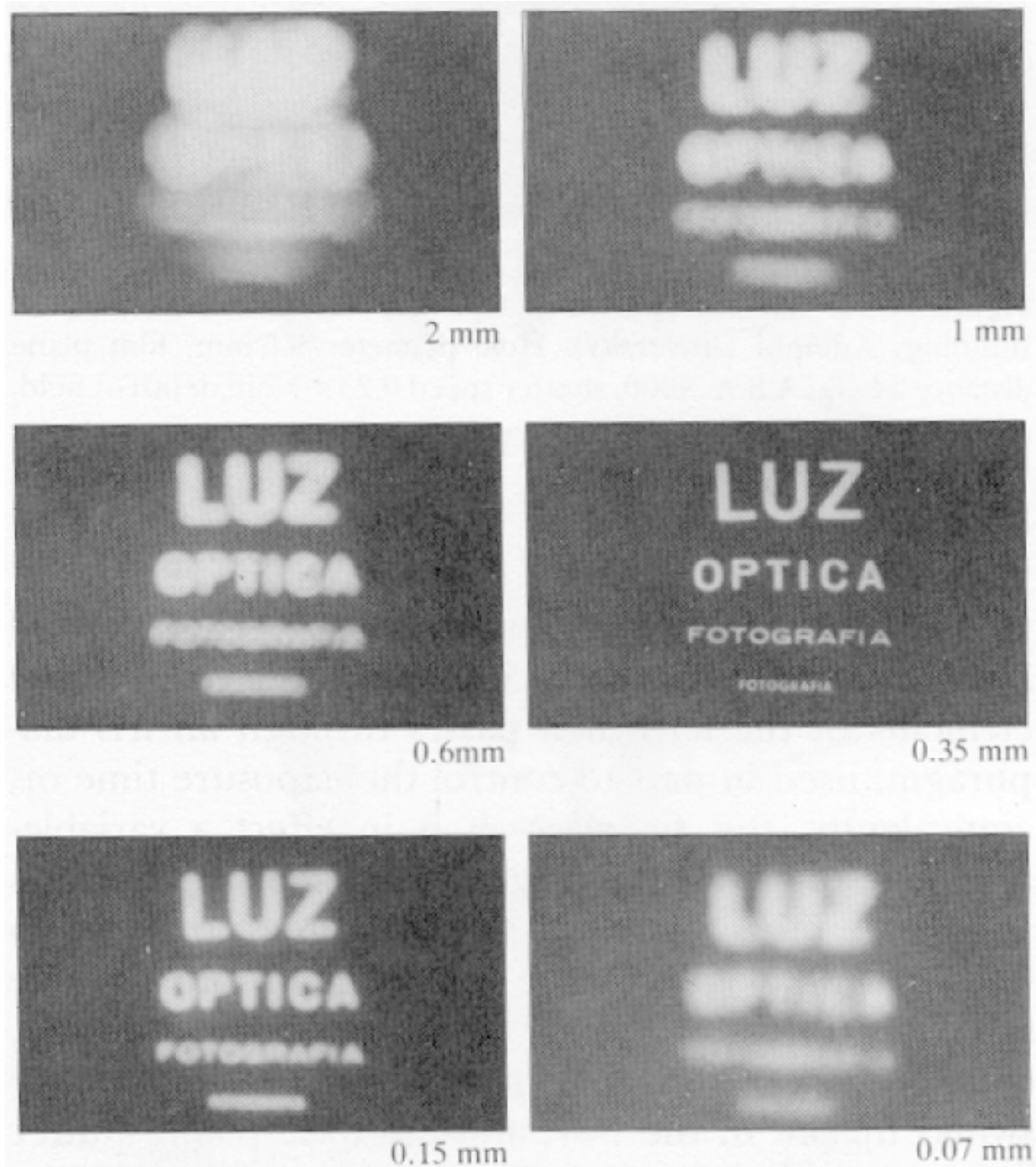
- Abstract camera model - box with a small hole in it
- Pinhole cameras work in practice



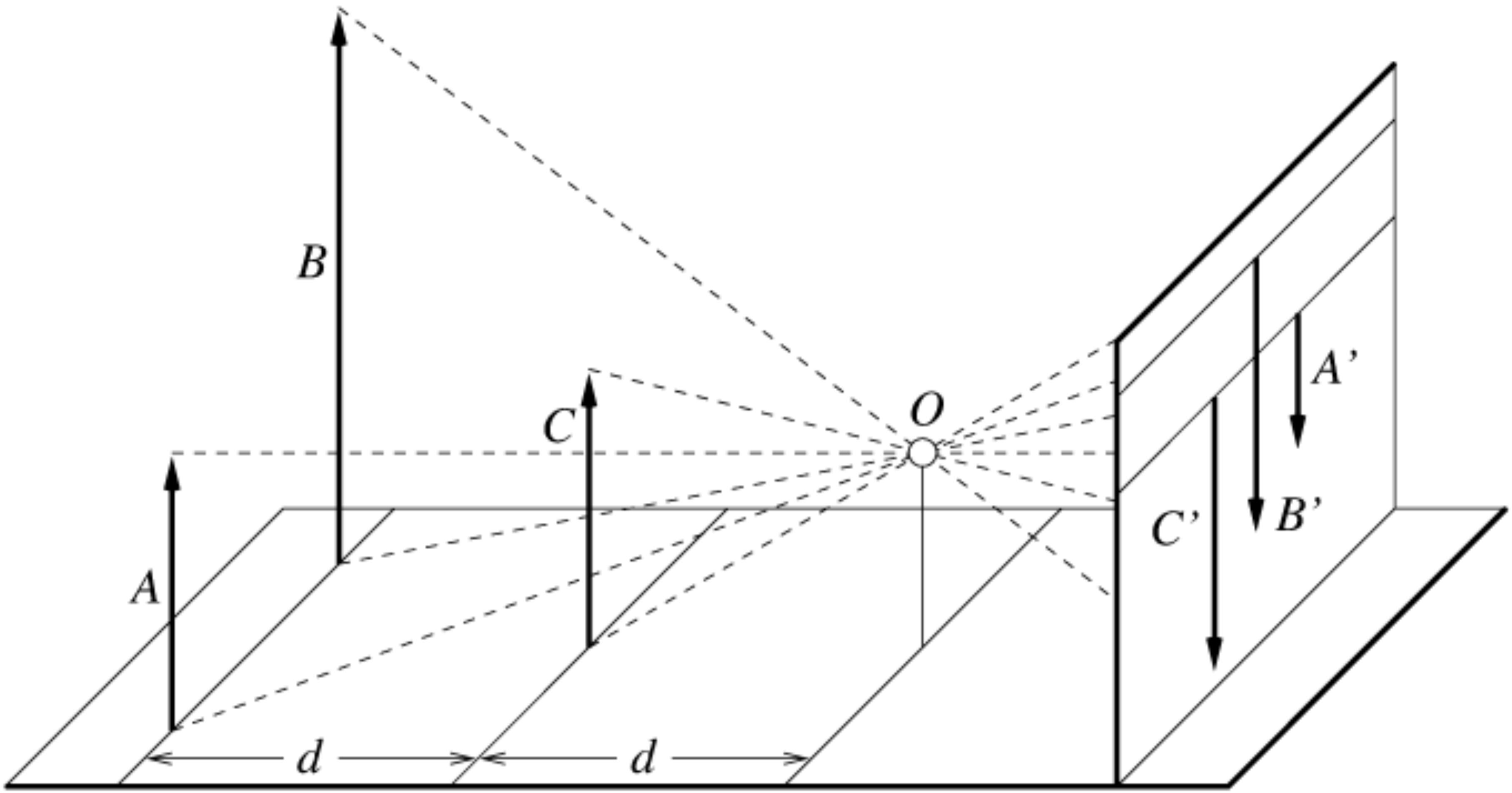
Pinhole too big - many directions are averaged, blurring the image

Pinhole too small- diffraction effects blur the image

Generally, pinhole cameras are *dark*, because a very small set of rays from a particular point hits the screen.



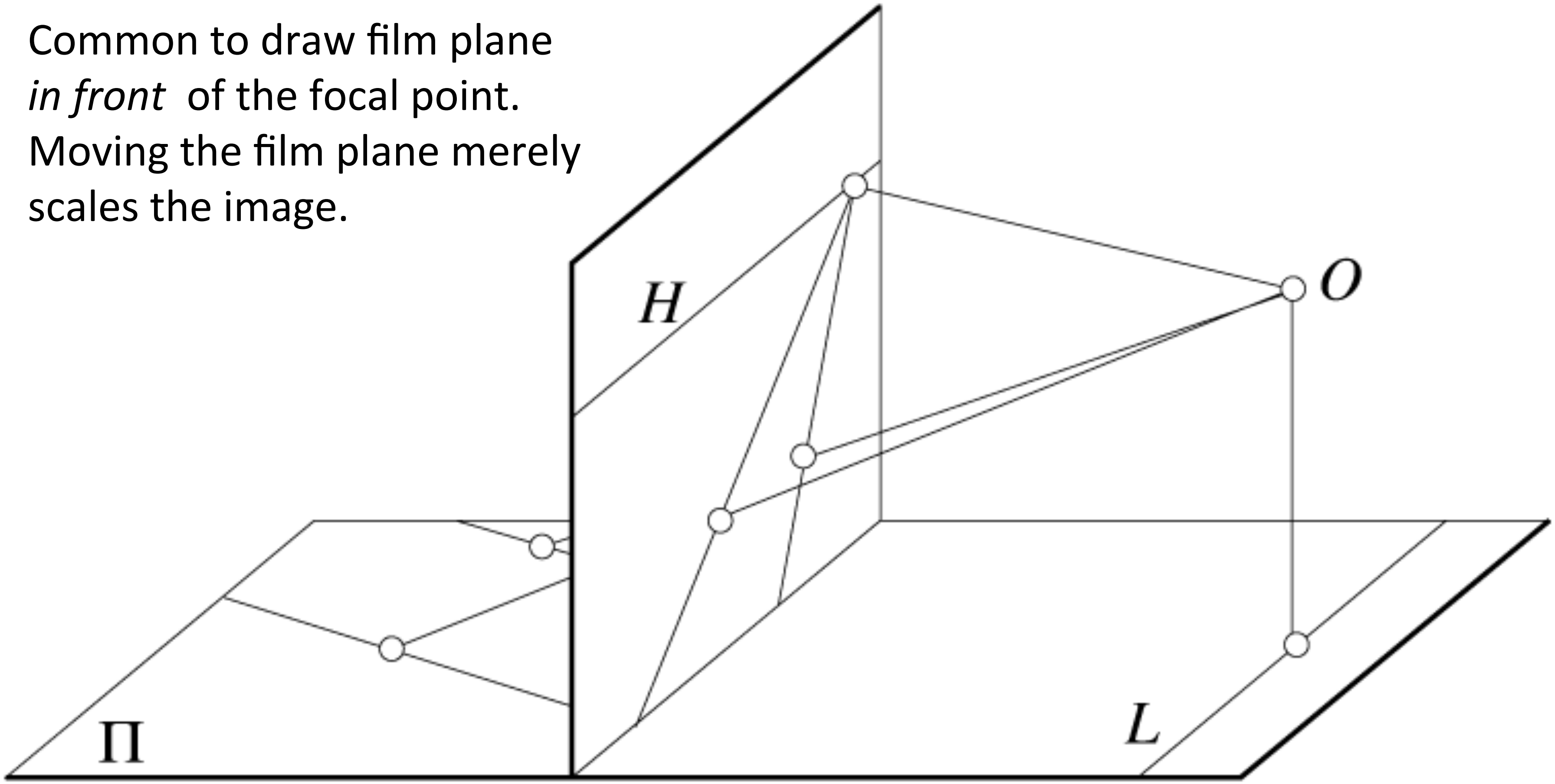
Distant objects are smaller

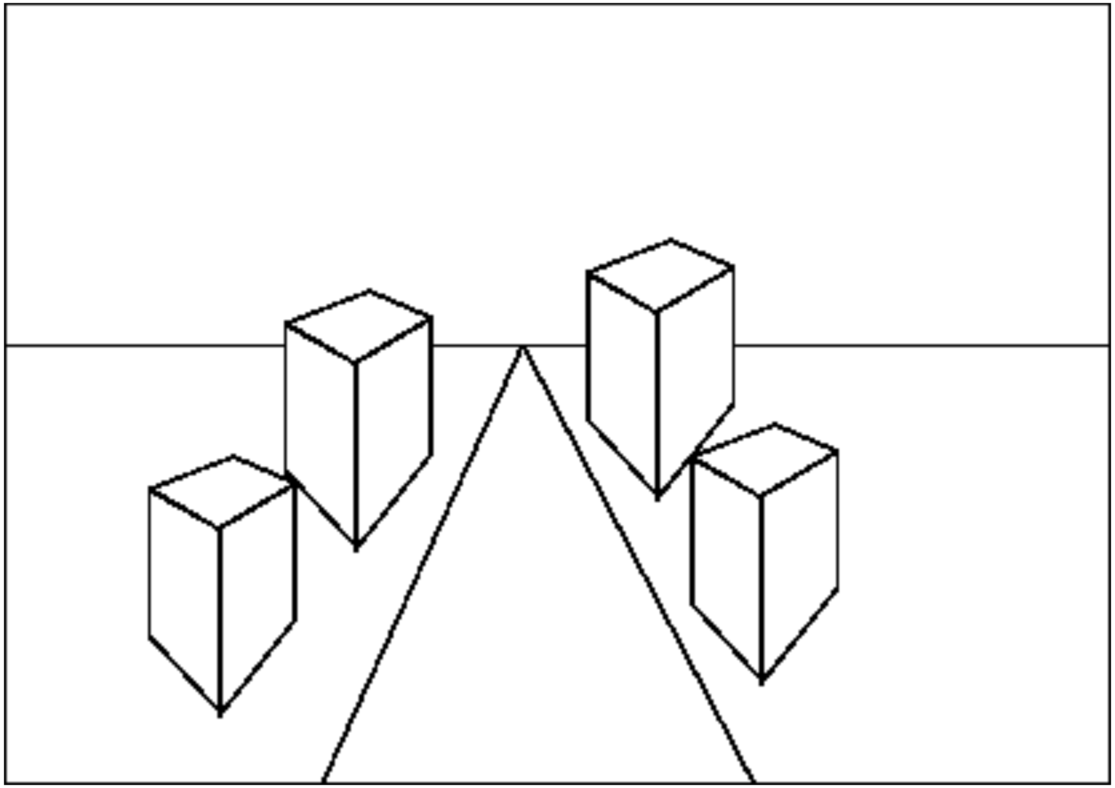


Computer Vision - A Modern Approach
Slides by D.A. Forsyth

Parallel lines meet

Common to draw film plane *in front* of the focal point. Moving the film plane merely scales the image.





Pinhole camera image

Amsterdam : what do you see in this picture?

- straight line
- size
- parallelism/angle
- shape
- shape of planes
- depth

Photo by Robert Kosara, robert@kosara.net
http://www.kosara.net/gallery/pinholeamsterdam/pic01.html
Slide: http://www.inf.u-szeged.hu/~kato/teaching/computervision/02-CameraGeometry.pdf

Pinhole camera image

✓straight line

- size
- parallelism/angle
- shape
- shape of planes
- depth

Amsterdam

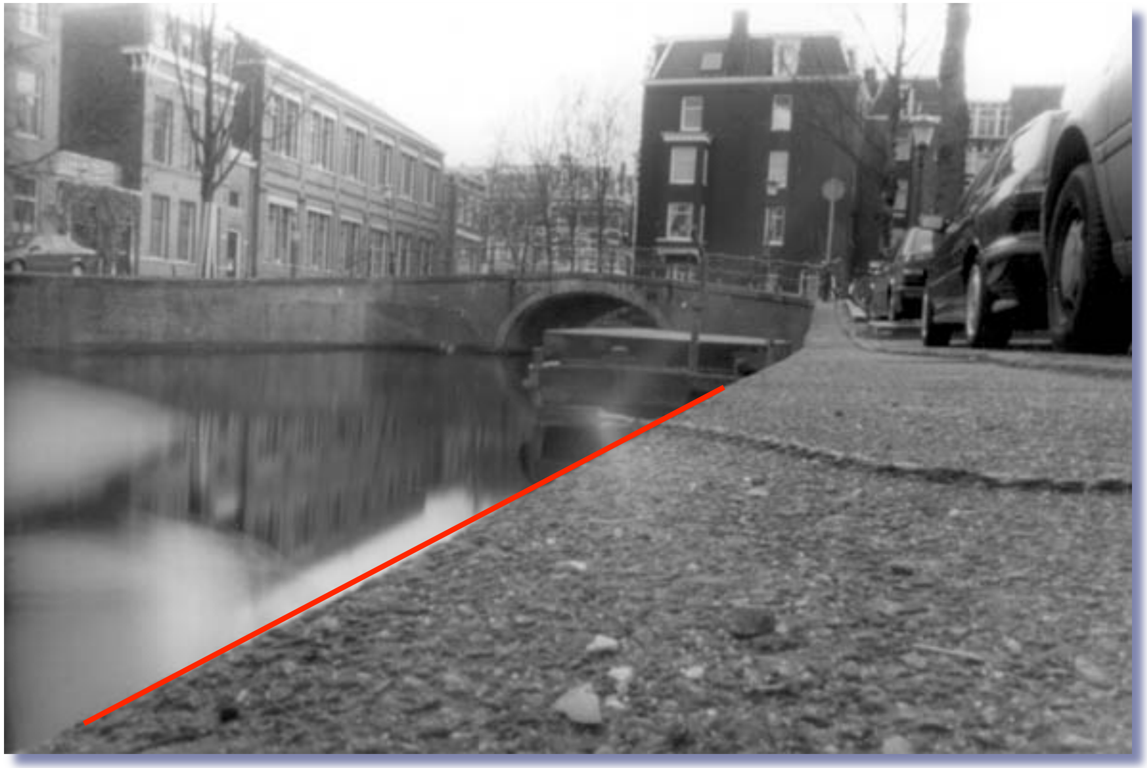


Photo by Robert Kosara, robert@kosara.net
http://www.kosara.net/gallery/pinholeamsterdam/pic01.html
Slide: http://www.inf.u-szeged.hu/~kato/teaching/computervision/02-CameraGeometry.pdf

Pinhole camera image

Amsterdam

- ✓straight line
- ×size
- ●parallelism/angle
- ●shape
- ●shape of planes
- ●depth

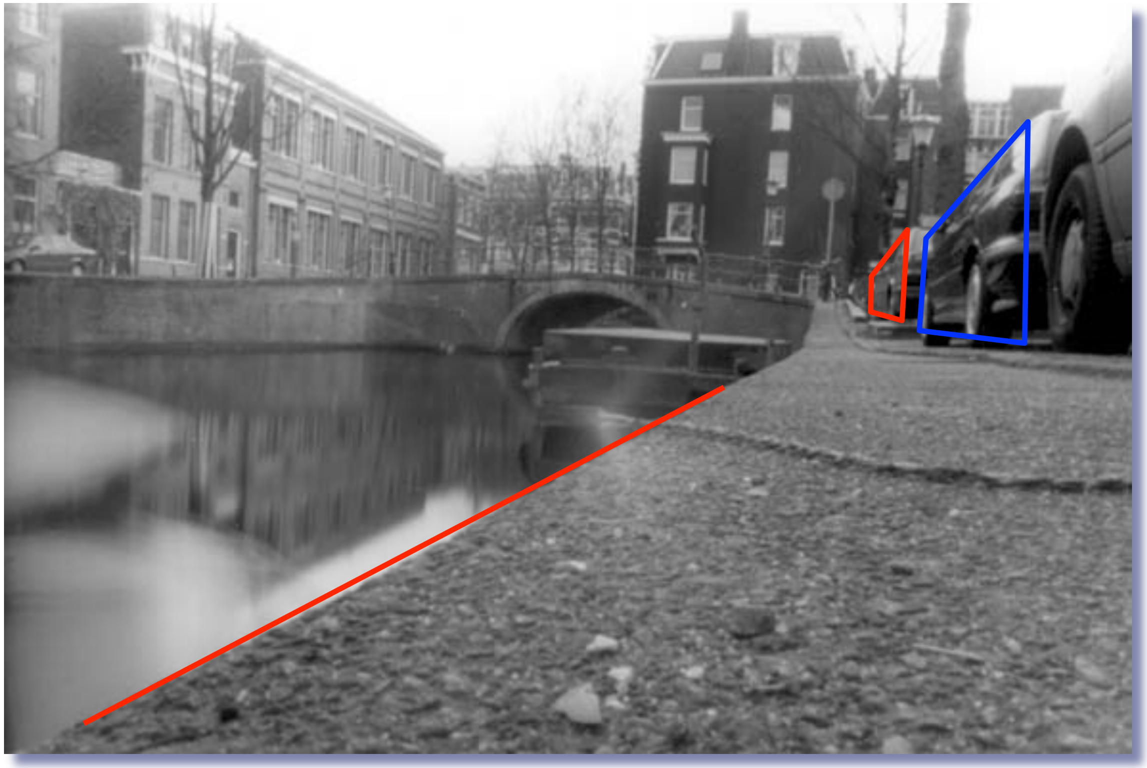


Photo by Robert Kosara, robert@kosara.net
http://www.kosara.net/gallery/pinholeamsterdam/pic01.html
Slide: http://www.inf.u-szeged.hu/~kato/teaching/computervision/02-CameraGeometry.pdf

Pinhole camera image

Amsterdam

- ✓straight line
- ×size
- ×parallelism/angle
- ●shape
- ●shape of planes
- ●depth

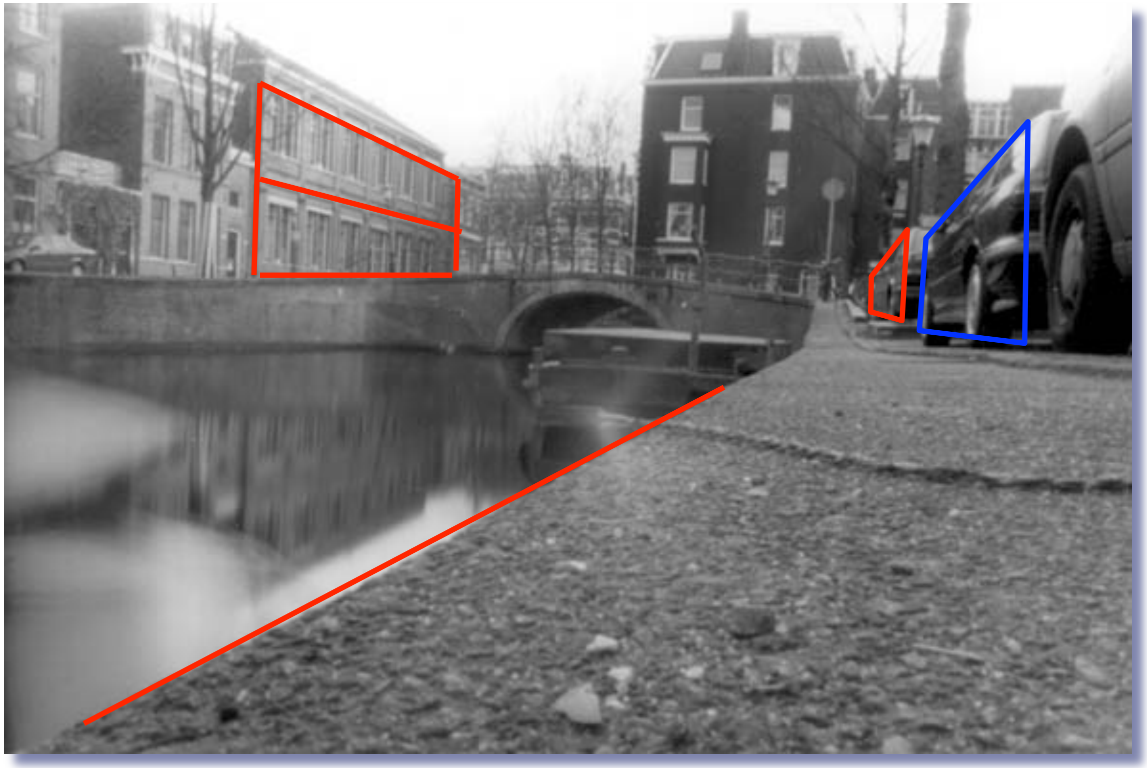


Photo by Robert Kosara, robert@kosara.net
http://www.kosara.net/gallery/pinholeamsterdam/pic01.html
Slide: http://www.inf.u-szeged.hu/~kato/teaching/computervision/02-CameraGeometry.pdf

Pinhole camera image

Amsterdam

- ✓straight line
- ×size
- ×parallelism/angle
- ×shape
- ●shape of planes
- ●depth

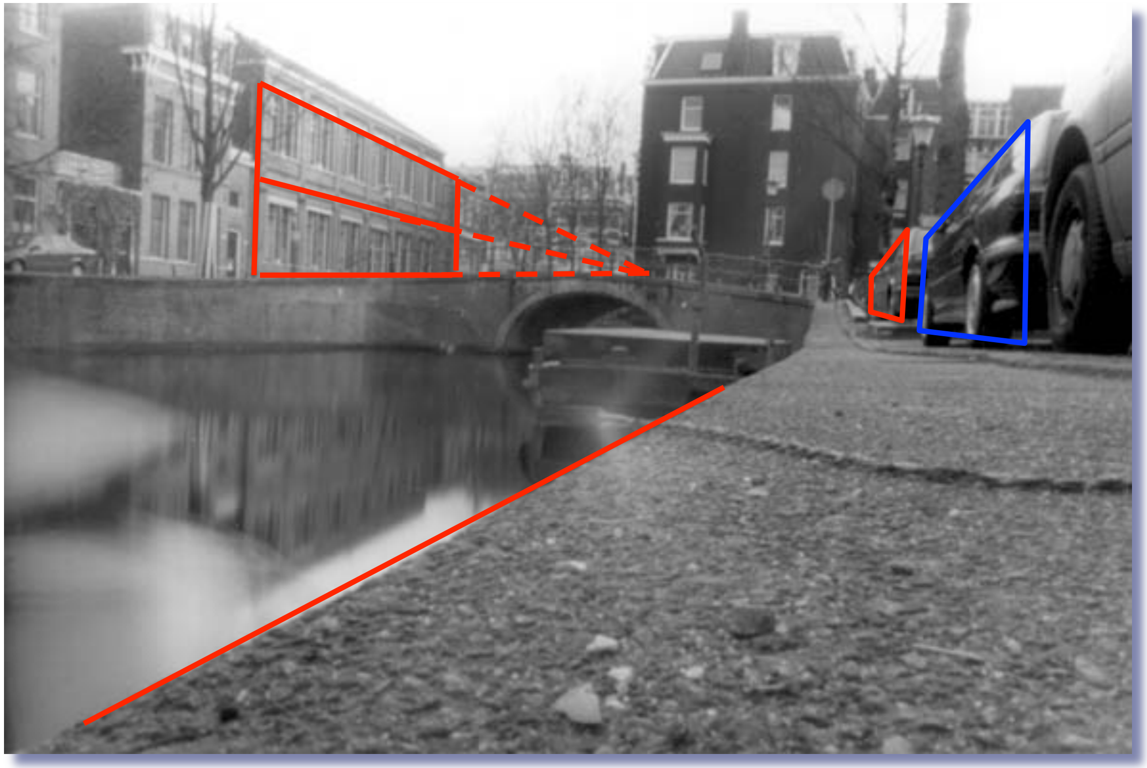


Photo by Robert Kosara, robert@kosara.net
http://www.kosara.net/gallery/pinholeamsterdam/pic01.html
Slide: http://www.inf.u-szeged.hu/~kato/teaching/computervision/02-CameraGeometry.pdf

Pinhole camera image

Amsterdam

- ✓ straight line
- × size
- × parallelism/angle
- × shape
- ● shape of planes
- ✓ parallel to image
- ● depth

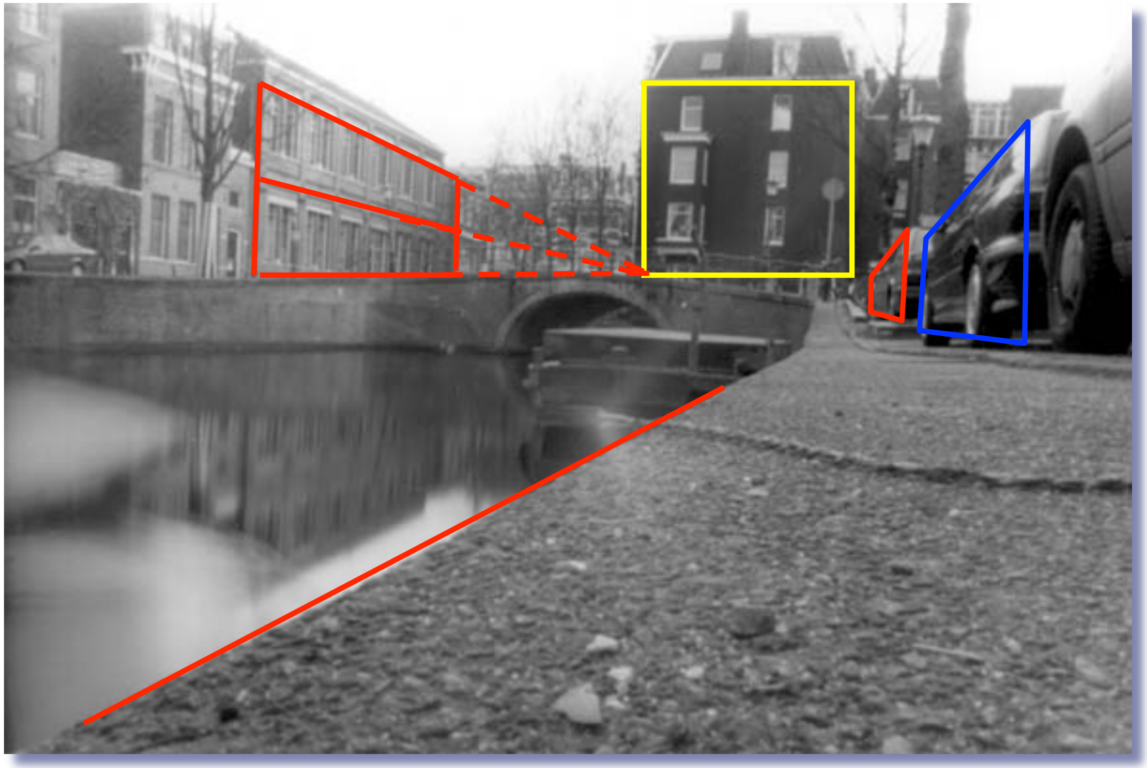


Photo by Robert Kosara, robert@kosara.net
http://www.kosara.net/gallery/pinholeamsterdam/pic01.html
Slide: http://www.inf.u-szeged.hu/~kato/teaching/computervision/02-CameraGeometry.pdf

Pinhole camera image

Amsterdam: what do you see?

✓straight line

×size

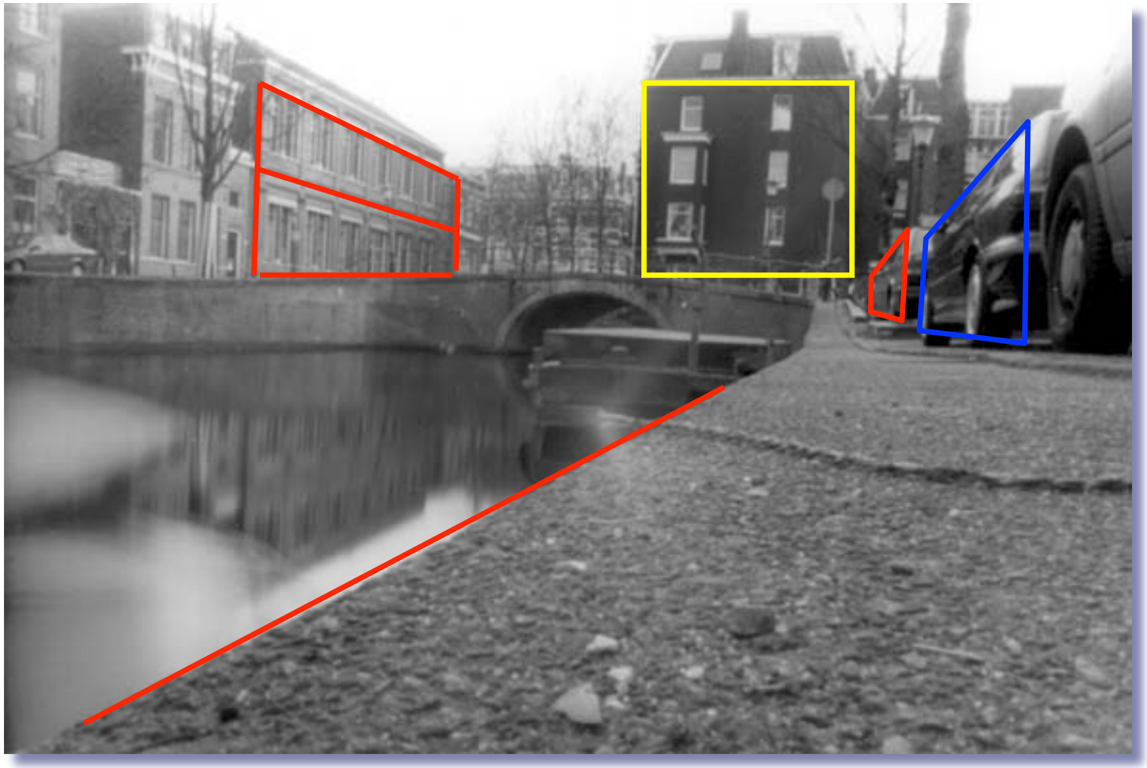
×parallelism/angle

×shape

- shape of planes

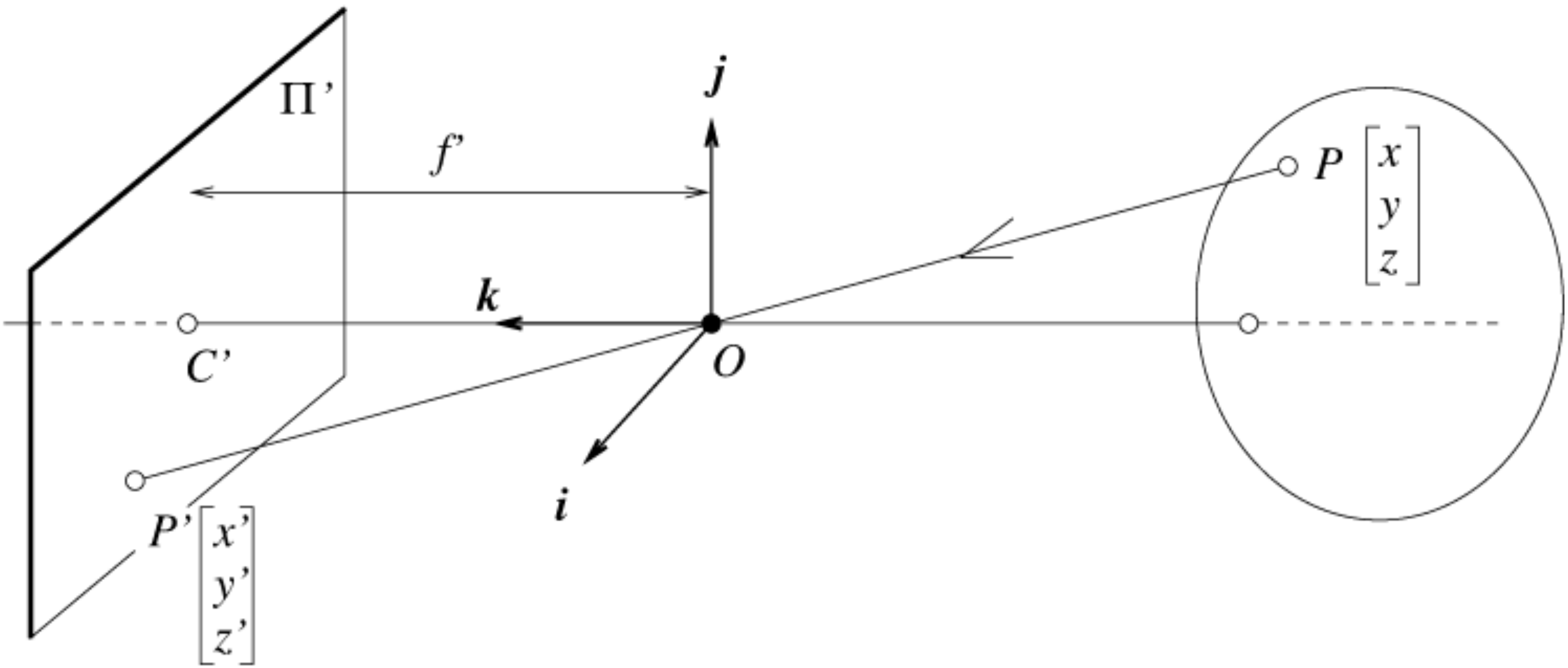
✓ parallel to image

- Depth ?
  - stereo
  - motion
  - size
  - structure …



- We see spatial shapes rather than individual pixels
- Knowledge: top-down vision belongs to human
- Stereo &Motion most successful in 3D CV & application
- You can see it but you don't know how…

The equation of projection

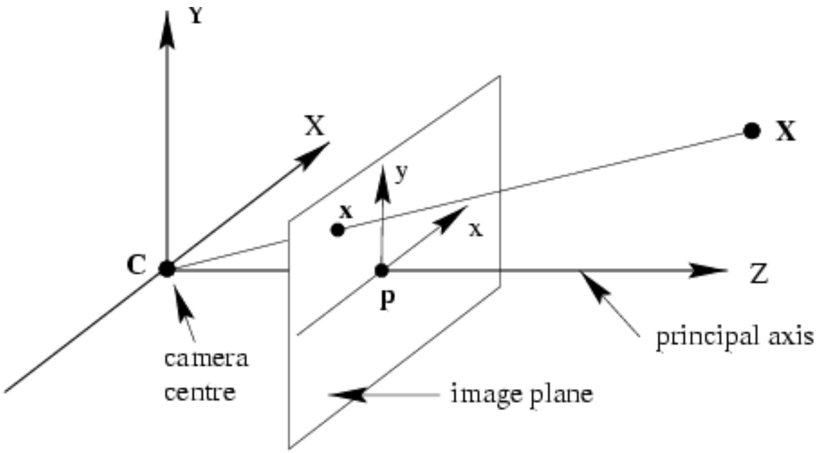


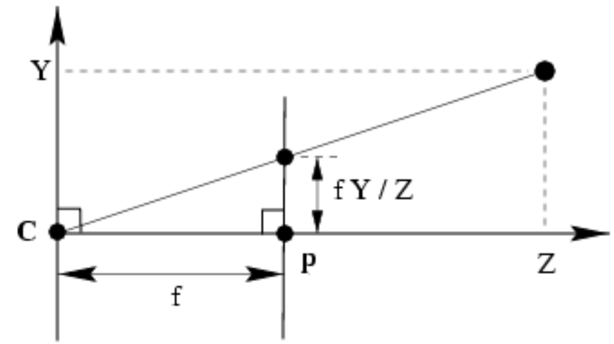
Homogenous coordinates

- Add an extra coordinate and use an equivalence relation
- for 2D
  - equivalence relation k*(X,Y,Z) is the same as (X,Y,Z)
- for 3D
  - equivalence relation k*(X,Y,Z,T) is the same as (X,Y,Z,T)

- Basic notion
  - Possible to represent points "at infinity"
    - Where parallel lines intersect
    - Where parallel planes intersect
  - Possible to write the action of a perspective camera as a matrix

Pinhole camera model

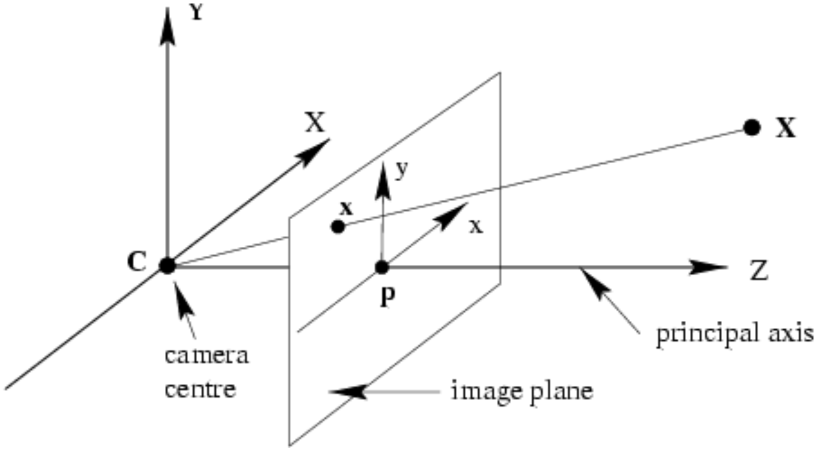


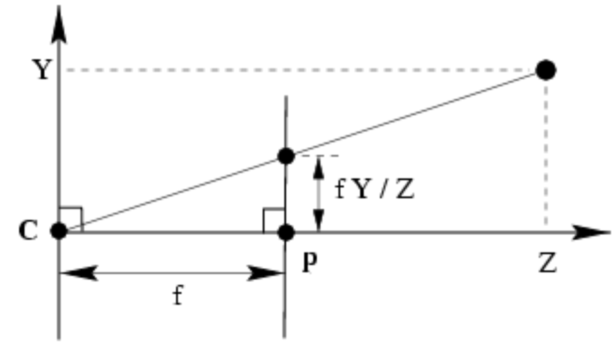


$$(X, Y, Z)^T \mapsto (fX / Z, fY / Z)^T$$

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

Pinhole camera model



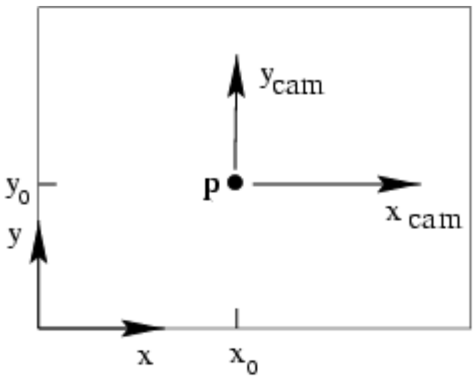


$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

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

$$\mathrm{P} = \mathrm{diag}(f, f, 1)\left[\mathrm{I} \mid 0\right]$$

Principal point offset

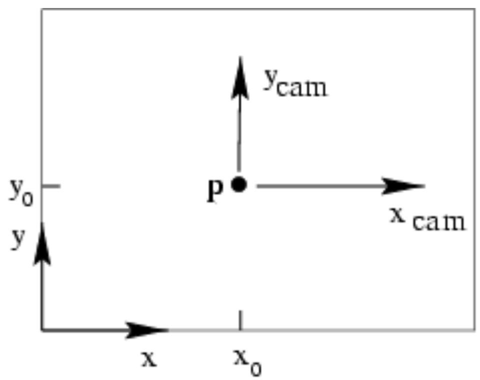


$$(X, Y, Z)^T \mapsto (fX/Z + p_x, fY/Z + p_y)^T$$

$(p_x, p_y)^T$ principal point

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

Principal point offset

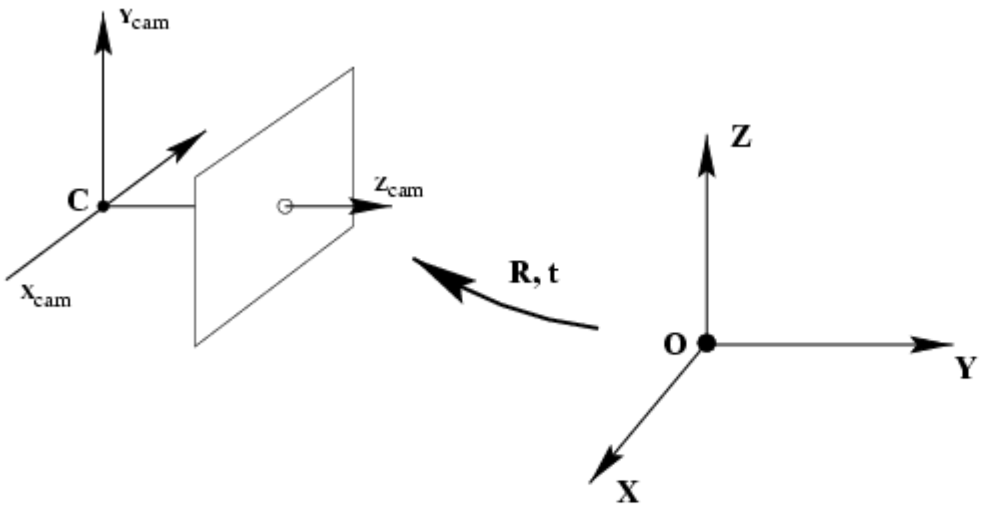


$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_x \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\mathrm{x} = \mathrm{K}\left[\mathrm{I} \mid 0\right] \mathrm{X}_{\mathrm{cam}}$$

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$ calibration matrix

Camera rotation and translation



$$\tilde{X}_{cam} = R\left(\tilde{X} - \tilde{C}\right)$$

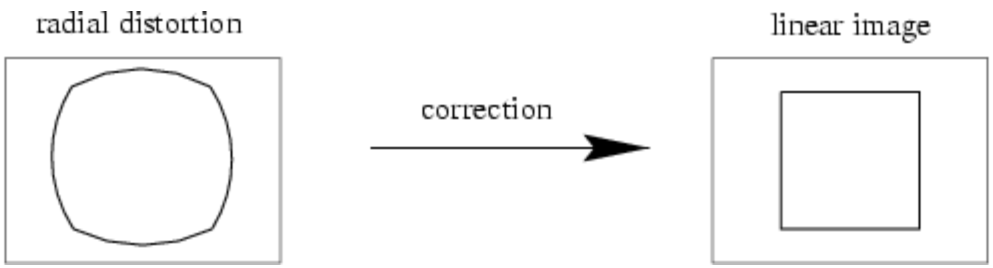
$$X_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} X$$

$$\mathbf{x} = \mathbf{K}\left[I \,|\, 0\right] X_{cam}$$

$$\mathbf{x} = \mathbf{K}R\left[I \,|\, -\tilde{C}\right] X$$

$$x = PX \qquad P = K\left[R \,|\, t\right] \qquad t = -R\tilde{C}$$



$$(\tilde{x}, \tilde{y}, 1)^\top = [\mathtt{I} \mid \mathbf{0}]\mathbf{X}_{\mathrm{cam}}$$

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(\tilde{r}) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}$$

Модель искажений линзы

$$x'' = x'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x' y' + p_2(r^2 + 2x'^2)$$
$$y'' = y'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y'^2) + 2p_2 x' y'$$
$$\text{where} \quad r^2 = x'^2 + y'^2$$
$$u = f_x * x'' + c_x$$
$$v = f_y * y'' + c_y$$

Pose estimation

Detected image points: $\left\{ \begin{pmatrix} u_i \\ v_i \end{pmatrix} \right\}_{i=1..n}$

Train object points: $\left\{ \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \right\}_{i=1..n}$

A class of object transformations:

$$f\left[\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, R, T \right] = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + T$$

Reprojection error

$$w_i \begin{pmatrix} u_i^p \\ v_i^p \\ 1 \end{pmatrix} = P \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix}$$

$$error(P) = \sum_i \left[\begin{pmatrix} u_i \\ v_i \end{pmatrix} - \begin{pmatrix} u_i^p \\ v_i^p \end{pmatrix} \right]^2$$

Perspective-n-Points problem

$$\min_{R,\,T} \operatorname{error}(K, R, T)$$

Direct Linear Transformation

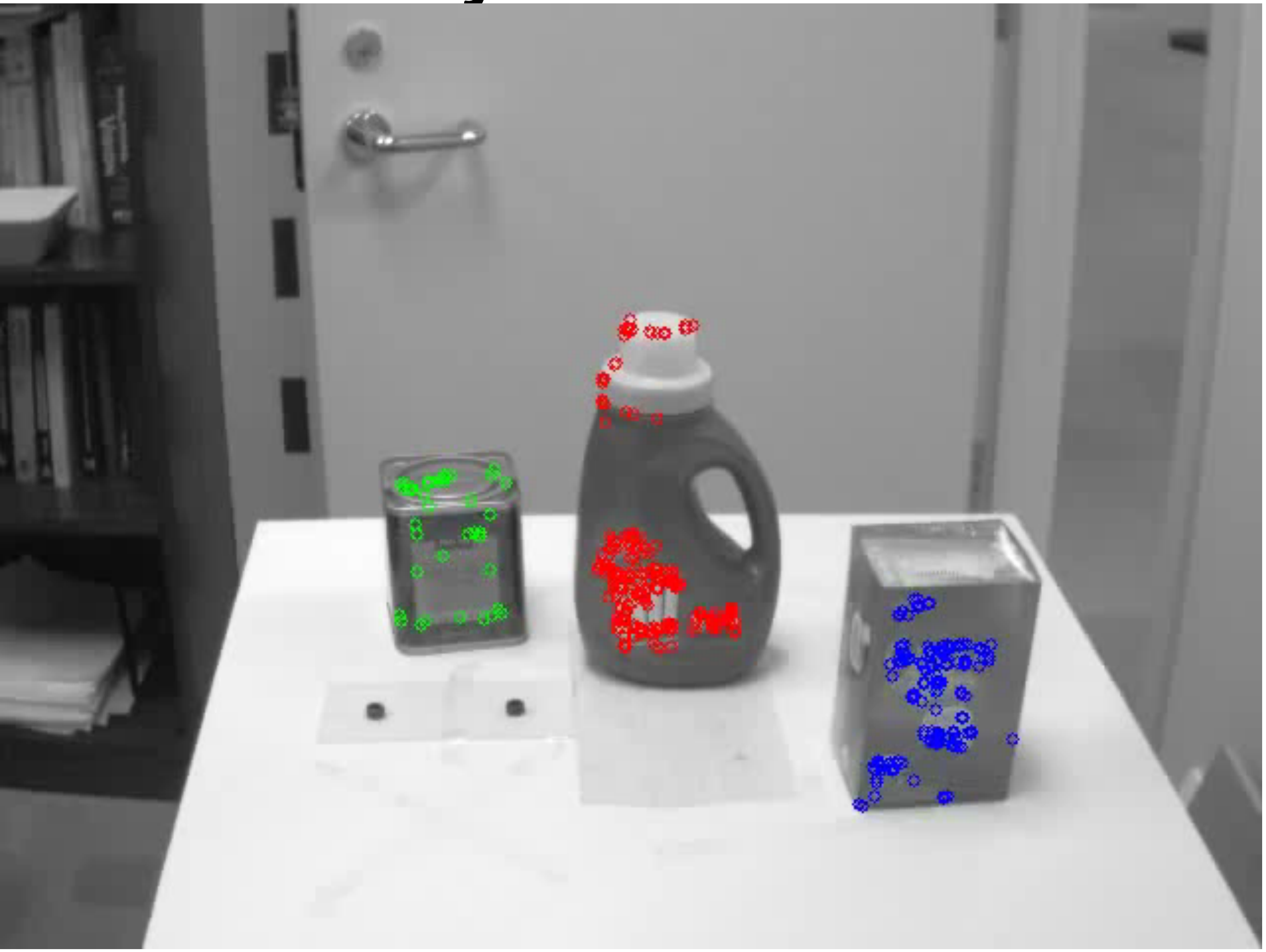
$$\begin{pmatrix} u_i^p w_i \\ v_i^p w_i \\ w_i \end{pmatrix} = P \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} u_i^p \\ v_i^p \\ 1 \end{pmatrix} \times P \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = 0$$

$$\left[R \mid T \right] = K^{-1} P$$

Other methods

- O(N) closed-form methods
- Levenberg-Marquardt
- P3P, P4P
- RANSAC

Object detection example



Iryna Gordon and David G. Lowe, "What and where: 3D object recognition with accurate pose," in *Toward Category-Level Object Recognition, eds. J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, (Springer-Verlag, 2006), pp. 67-82.*

Manuel Martinez Torres, Alvaro Collet Romea, and Siddhartha Srinivasa, MOPED: A Scalable and Low Latency Object Recognition and Pose Estimation System, Proceedings of ICRA 2010, May, 2010.

Optimization: Gauss-Newton method

$$\min C\left(\vec{\beta}\right) = \sum_i \left(y_i - f\left(\vec{x}_i, \vec{\beta}\right)\right)^2 = \vec{\varepsilon}^T\left(\vec{\beta}\right)\vec{\varepsilon}\left(\vec{\beta}\right)$$

$$C\left(\vec{\beta}+\vec{\delta}\right) \approx \left(\vec{\varepsilon}\left(\vec{\beta}\right)+J\vec{\delta}\right)^T\left(\vec{\varepsilon}\left(\vec{\beta}\right)+J\vec{\delta}\right)+O\left(\|\delta\|^2\right)$$

$$\nabla C\left(\vec{\beta}+\vec{\delta}\right) \approx -2J^T\left(\vec{\varepsilon}\left(\vec{\beta}\right)+J\vec{\delta}\right)$$

$$J^T J\vec{\delta} = -J^T\vec{\varepsilon}$$

Optimization: Levenberg method

$$\left(J^T J + \lambda E\right)\vec{\delta} = J^T \vec{\varepsilon}$$

$$\left\|\lambda E\right\| >> \left\|J^T J\right\|:\ \lambda\vec{\delta} = J^T \vec{\varepsilon}, \textit{this is gradient descent}$$

$$\left\|\lambda E\right\| << \left\|J^T J\right\|:\ J^T J\vec{\delta} = J^T \vec{\varepsilon}, \textit{this is Gauss} - \textit{Newton}$$

Optimization: Levenberg-Marquardt method

$$\left(J^T J + \lambda diag\left(J^T J\right)\right)\vec{\delta} = J^T \vec{\varepsilon}$$

Practical considerations

- Cholesky decomposition
  - The left-hand side matrix can be forced to be positive-definite
- The matrix is usually sparse, so sparse Cholesky decomposition is possible