Geometry of image formation

Tomáš Svoboda, svoboda@cmp.felk.cvut.cz
Czech Technical University in Prague, Center for Machine Perception
http://cmp.felk.cvut.cz

Last update: November 14, 2007

Talk Outline

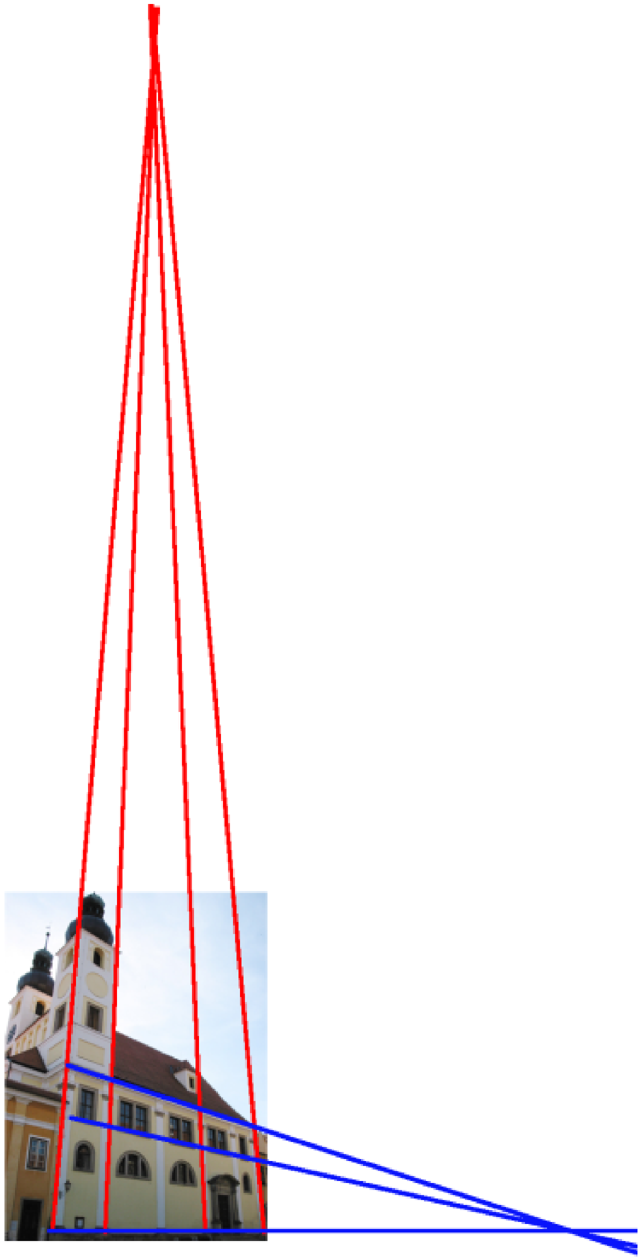
- Pinhole model
- Camera parameters
- Estimation of the parameters—Camera calibration

Motivation

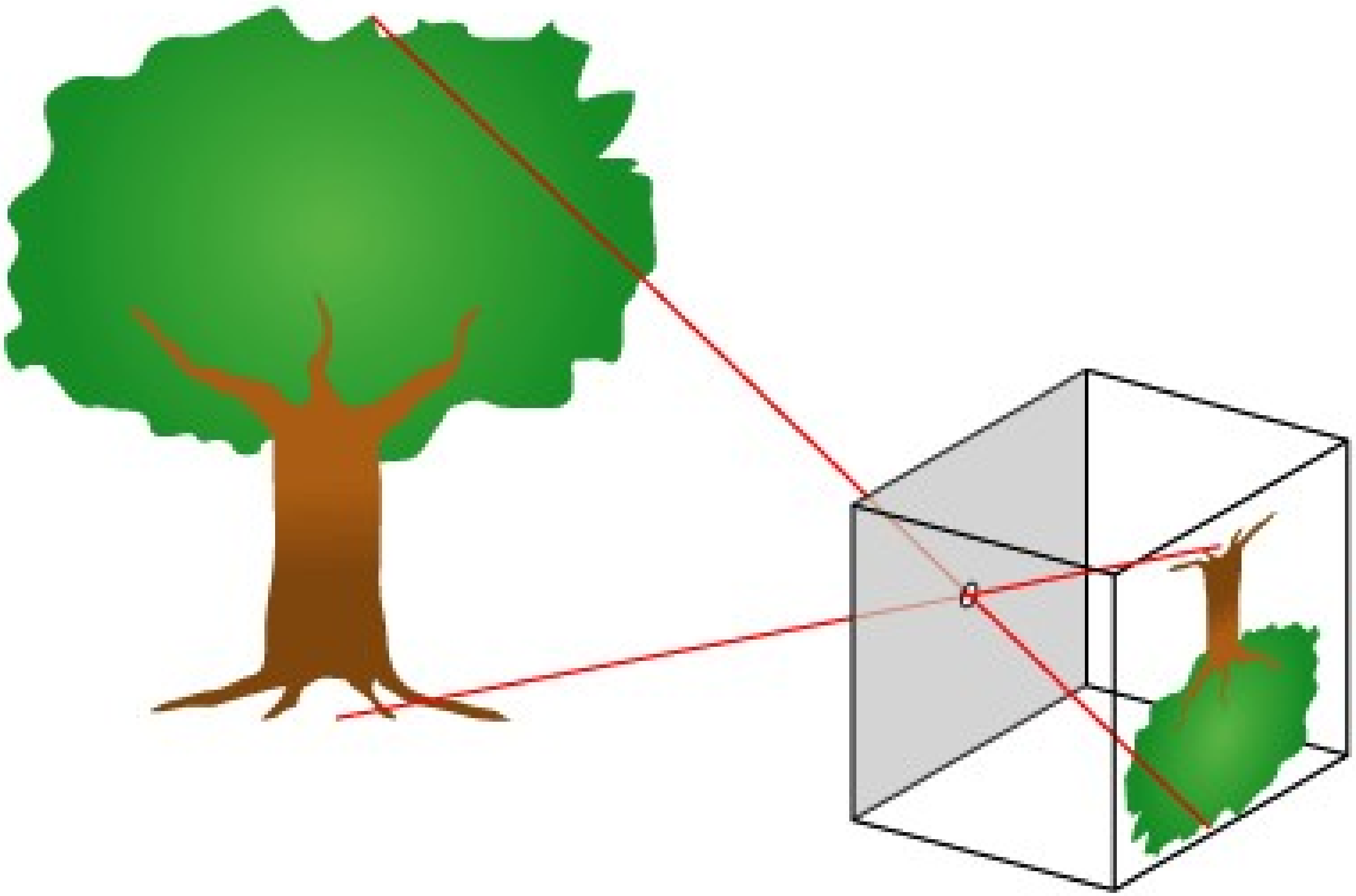
- parallel lines
- window sizes
- image units
- distance from the camera

What will we learn

- how does the 3D world project to 2D image plane?
- how is a camera modeled?
- how can we estimate the camera model?



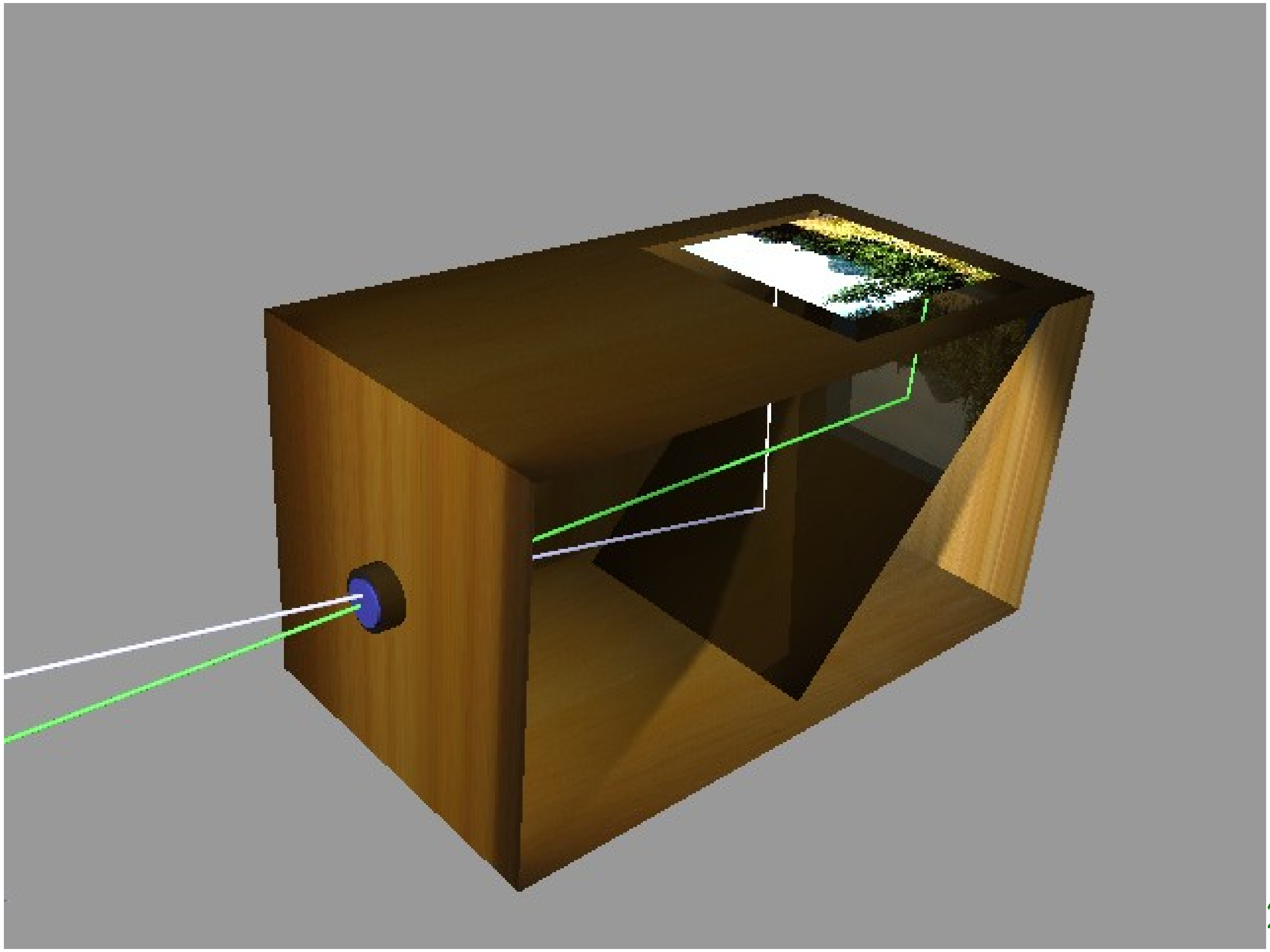
Pinhole camera



[1]

[1] http://en.wikipedia.org/wiki/Pinhole_camera

Camera Obscura

 [2]

[2] http://en.wikipedia.org/wiki/Camera_obscura

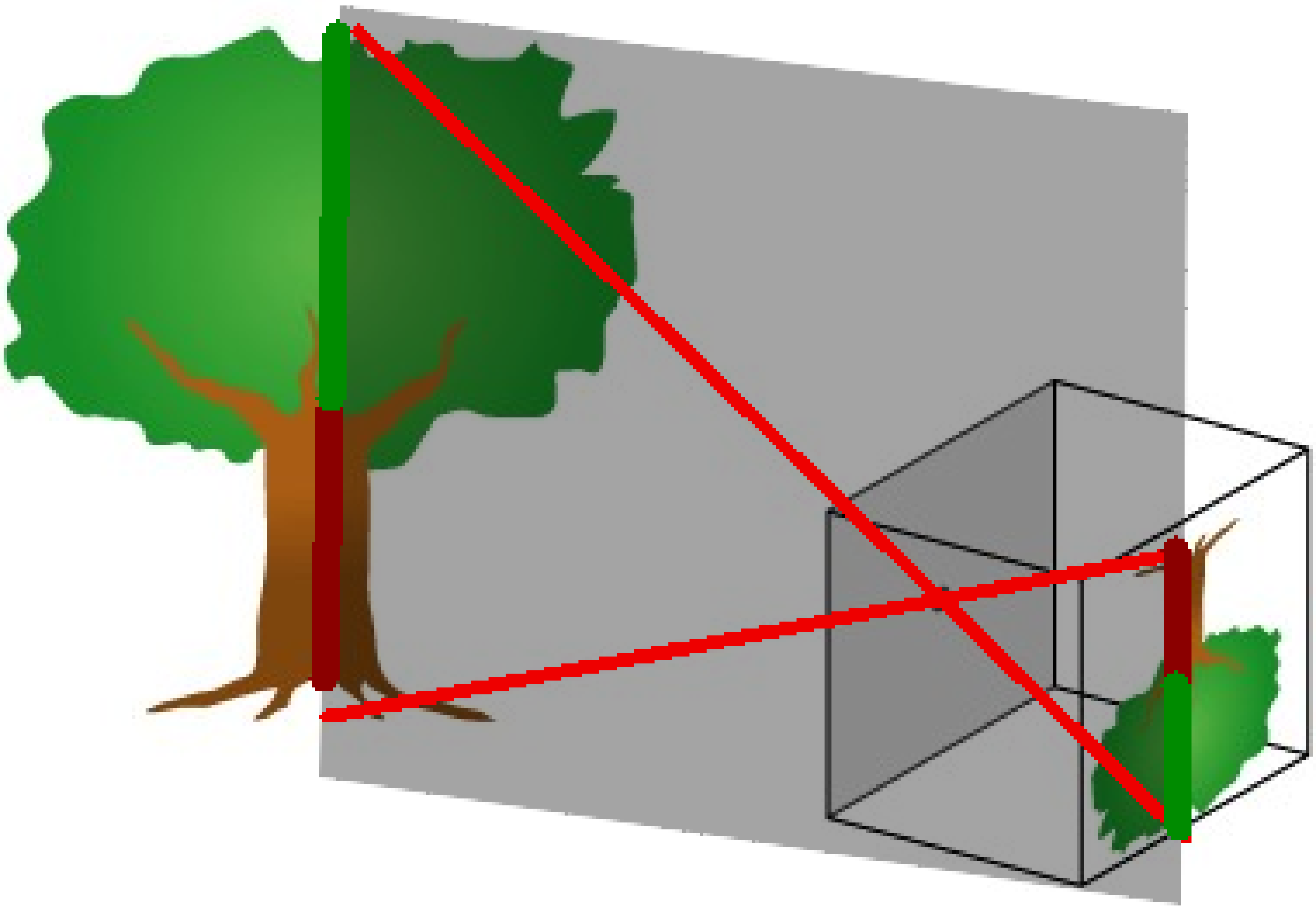
Camera Obscura — room-sized

[3]

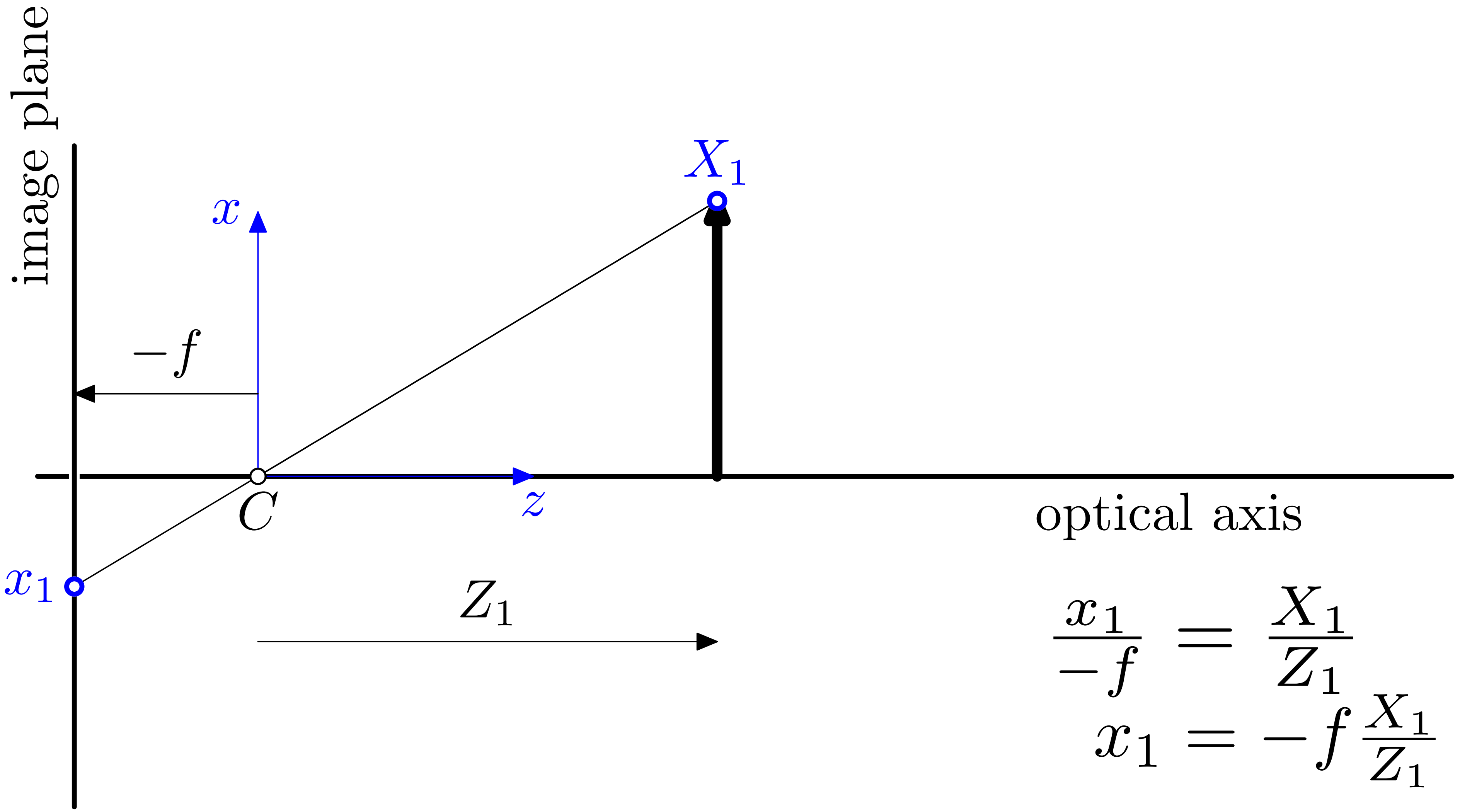
Used by the art department at the UNC at Chapel Hill

[3] http://en.wikipedia.org/wiki/Camera_obscura

1D Pinhole camera



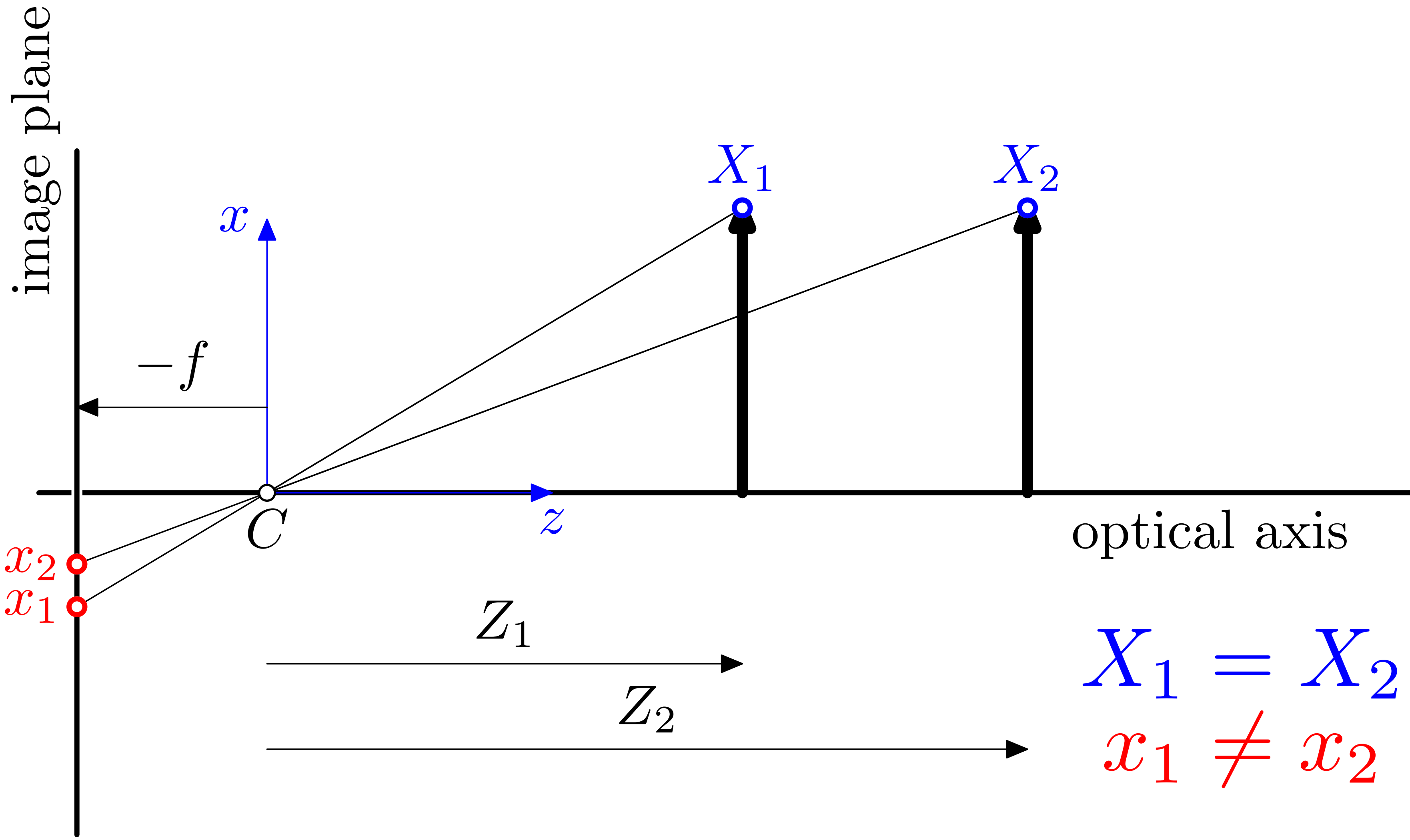
1D Pinhole camera projects 2D to 1D



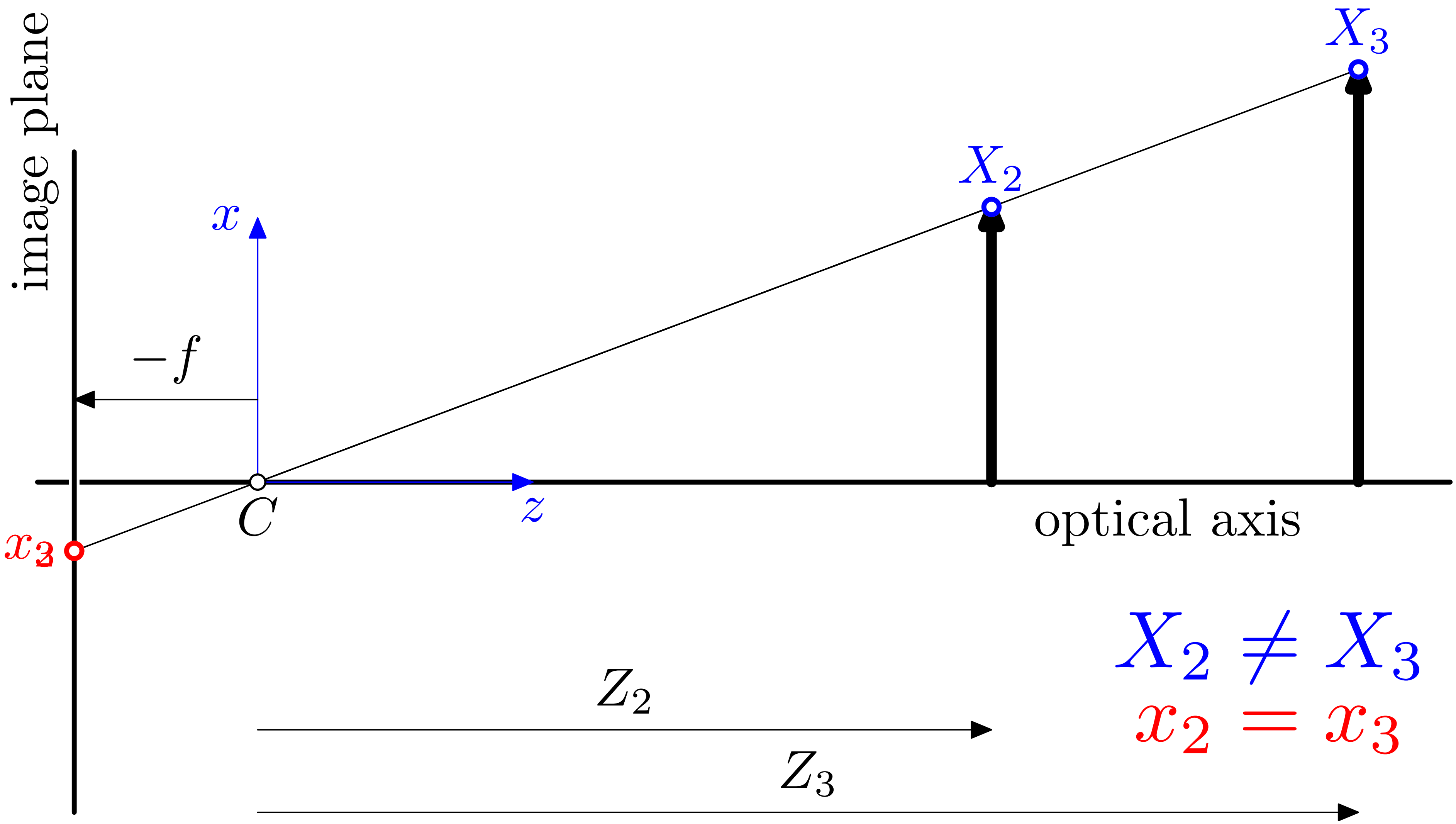
$$\frac{x_1}{-f} = \frac{X_1}{Z_1}$$

$$x_1 = -f\,\frac{X_1}{Z_1}$$

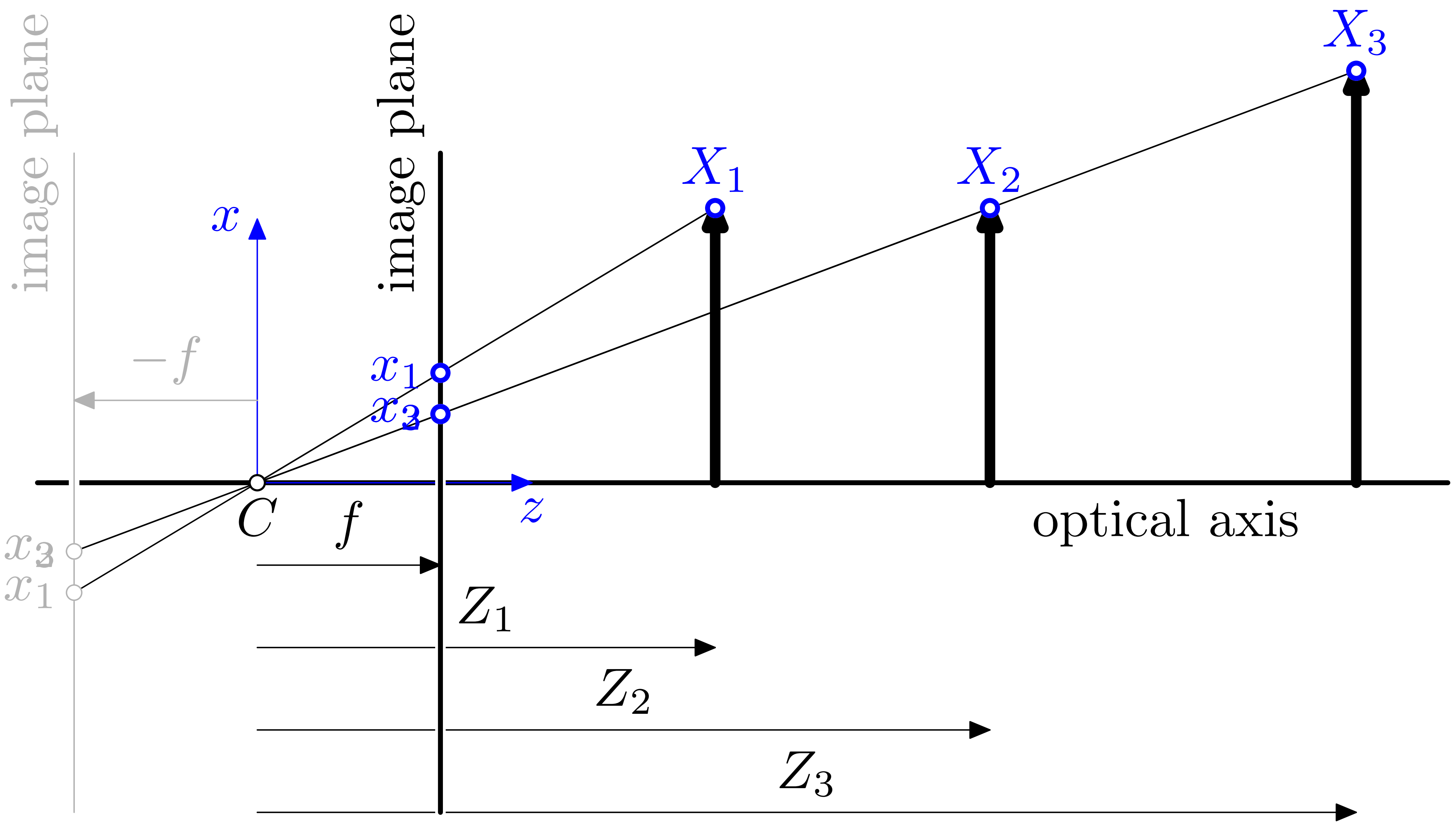
Problems with perspective I



Problems with perspective II

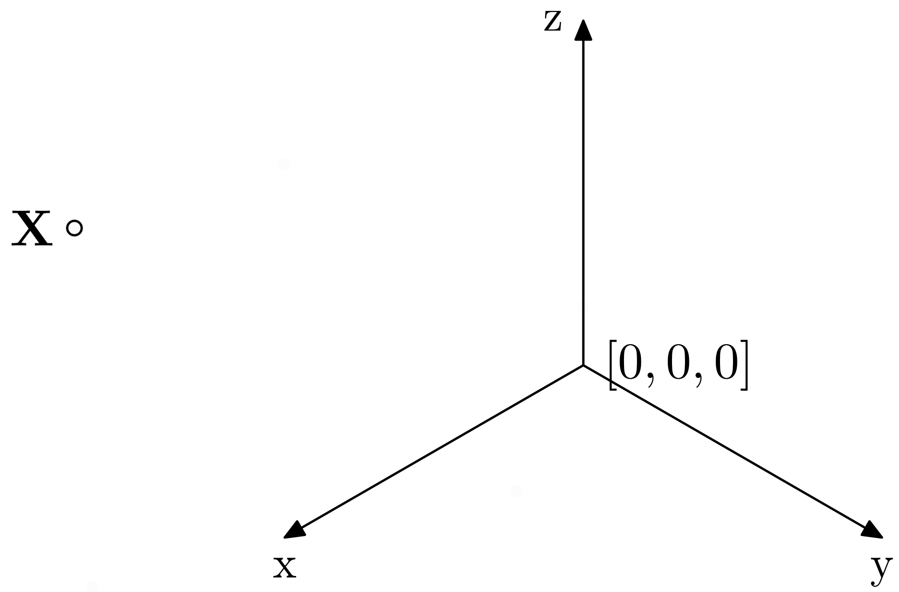


Get rid of the $(-)$ sign

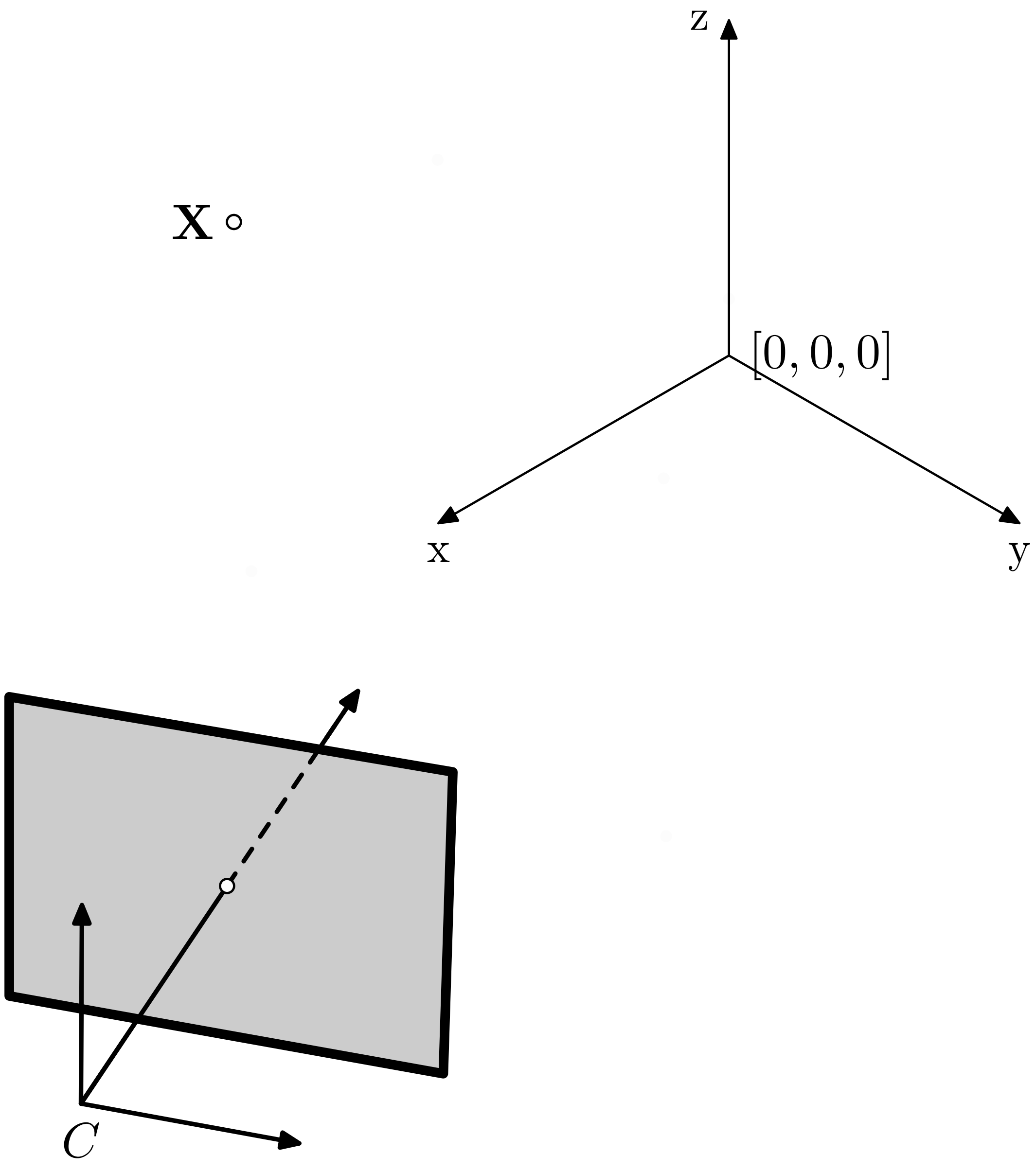


How does the 3D world
project to the 2D image plane?

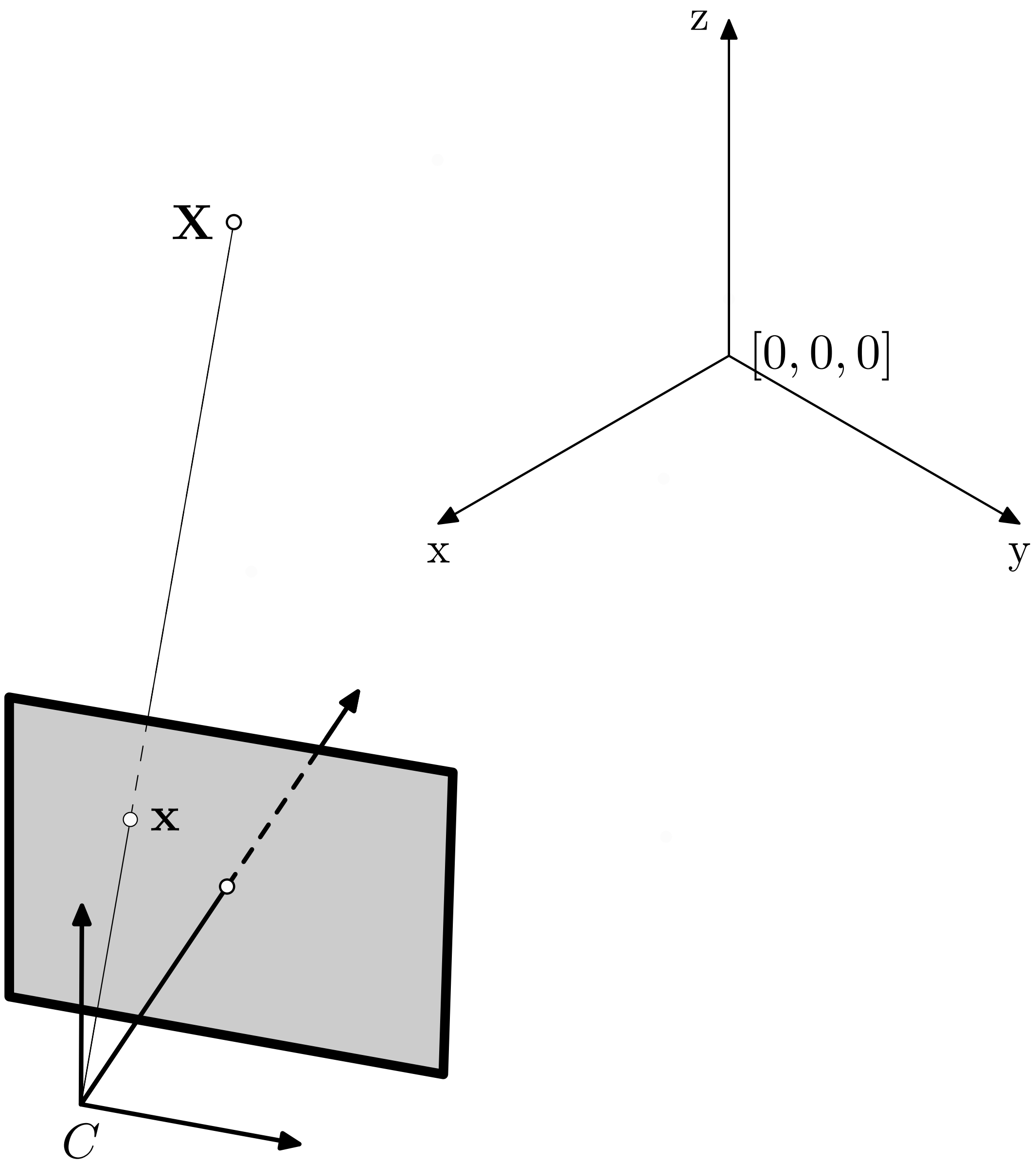
A 3D point $\mathbf{X}$ in a world coordinate system



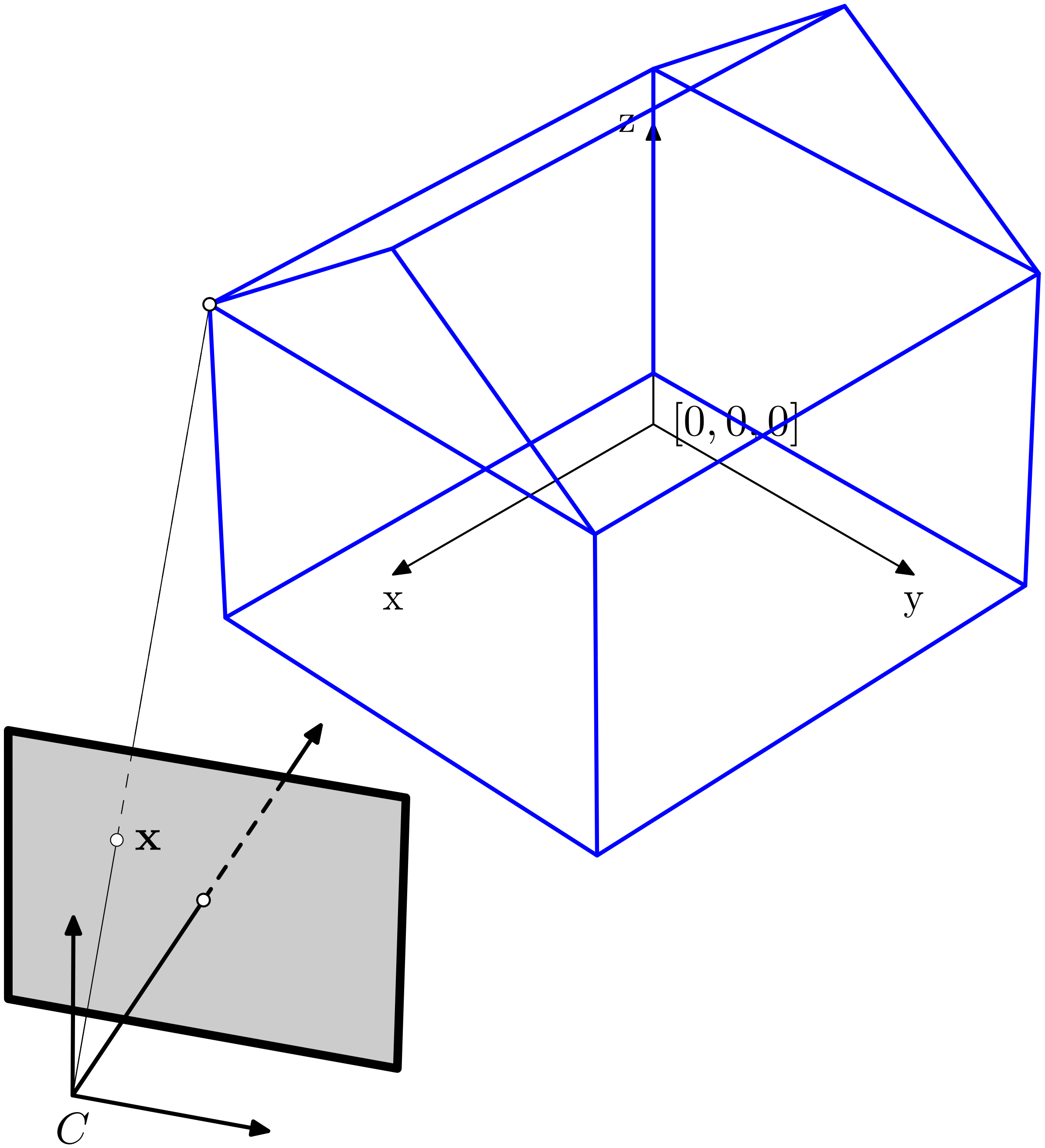
A pinhole camera observes a scene



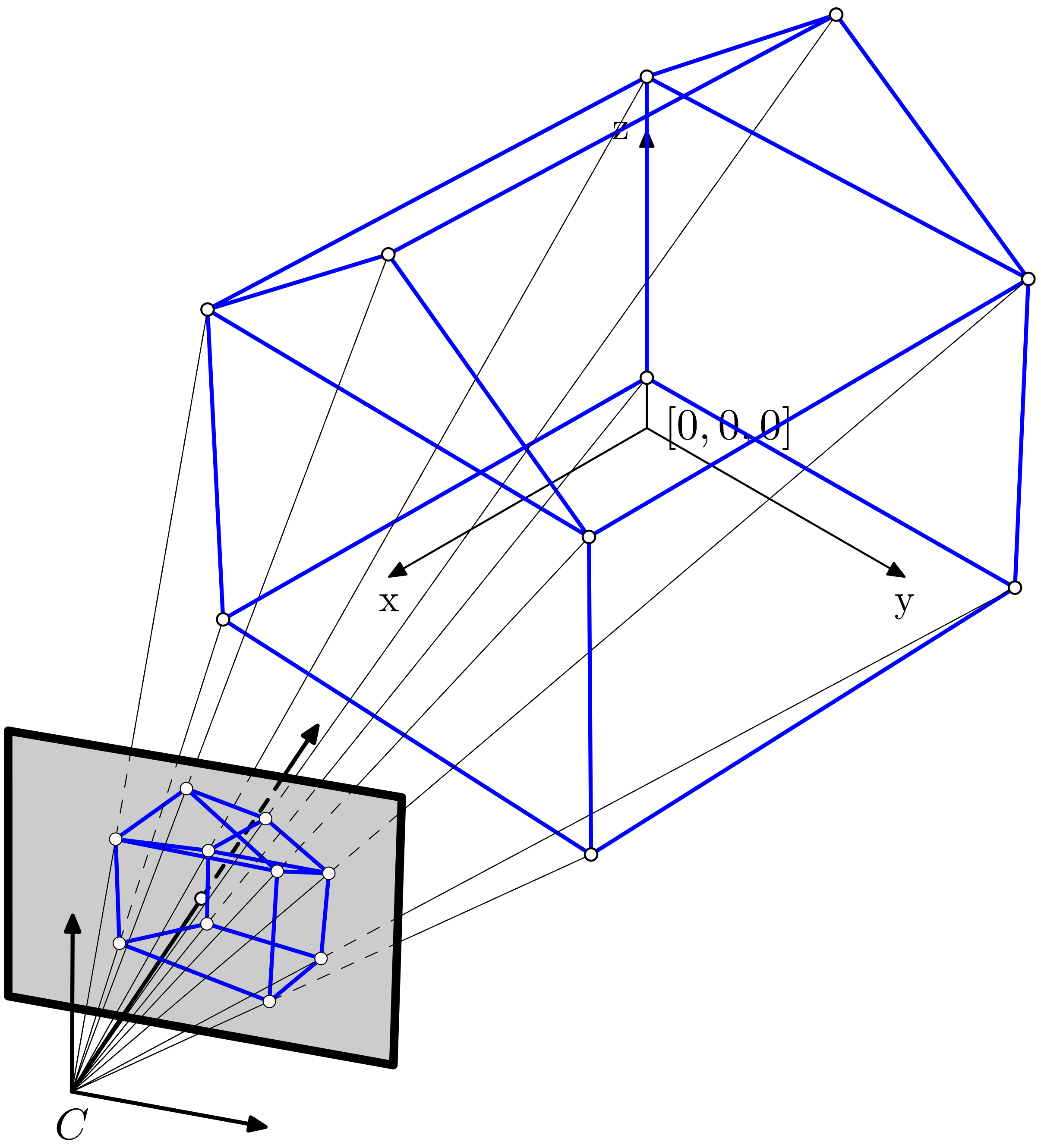
Point $\mathbf{X}$ projects to the image plane, point $\mathbf{x}$



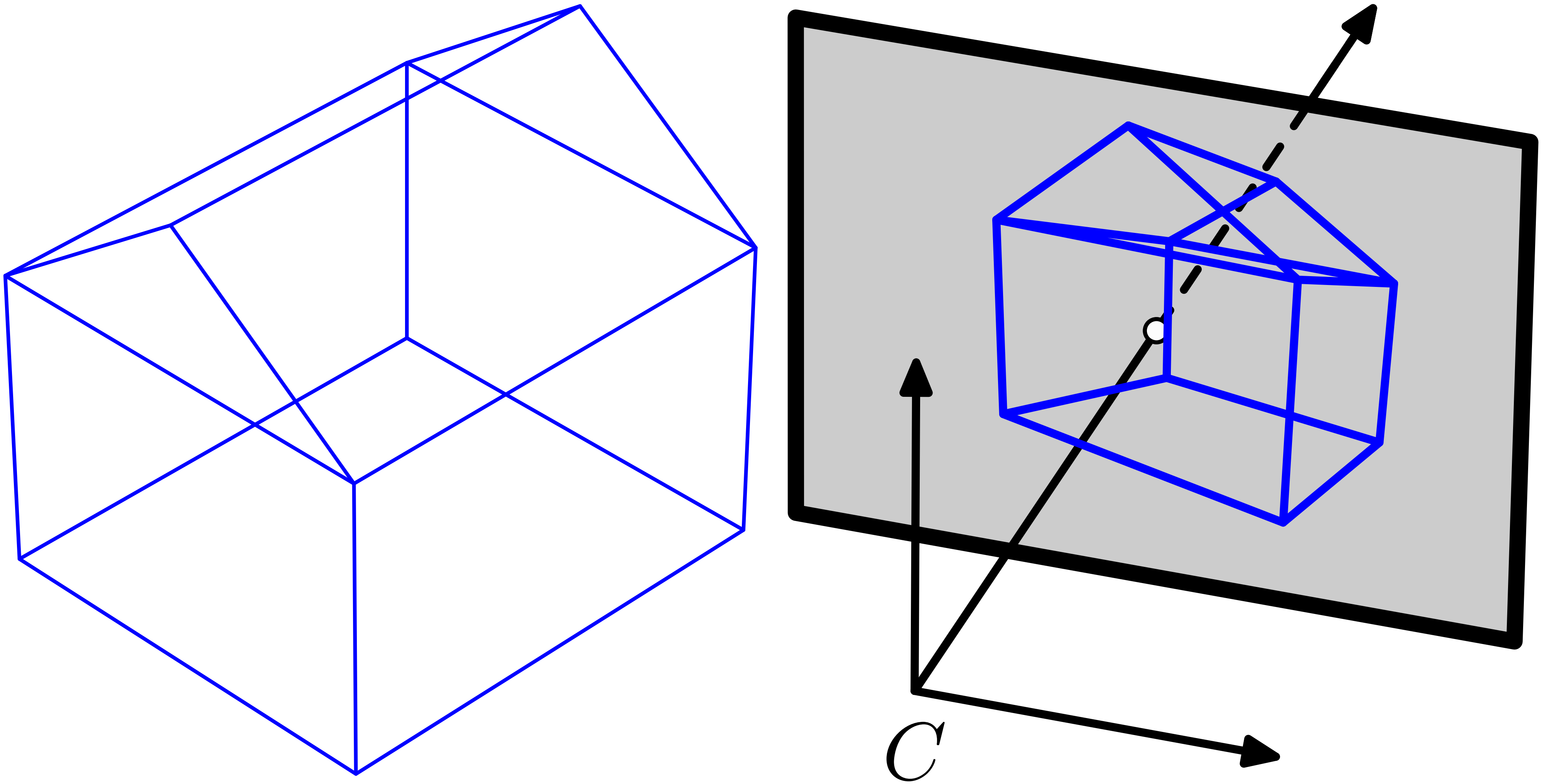
Scene projection



Scene projection



3D Scene projection – observations



- 3D lines project to 2D lines
- but the angles change, parallel lines are no more parallel.
- area ratios change, note the front and backside of the house

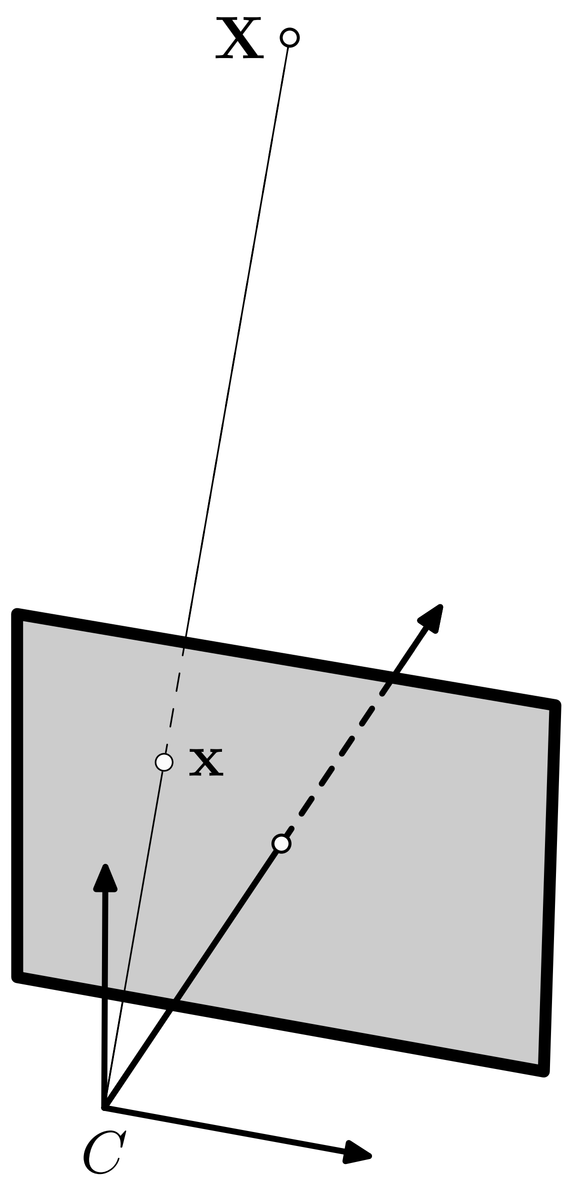
Put the sketches into equations

3D → 2D Projection

We remember that: $\mathbf{x} = [\frac{fX}{Z}, \frac{fY}{Z}]^\top$

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \simeq \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \simeq \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$



Use the homegeneous coordinates[4]

$$\lambda_{[1\times1]}\mathbf{x}_{[3\times1]} = \mathtt{K}_{[3\times3]} \left[\ \mathtt{I}|\mathbf{0} \ \right] \mathbf{X}_{[4\times1]}$$

but . . .

[4] for the notation conventions, see the talk notes

. . . we need the $\mathbf{X}$ in camera coordinate system

Rotate the vector:

$$\mathbf{X} = \mathtt{R}(\mathbf{X}_w - \mathbf{C}_w)$$

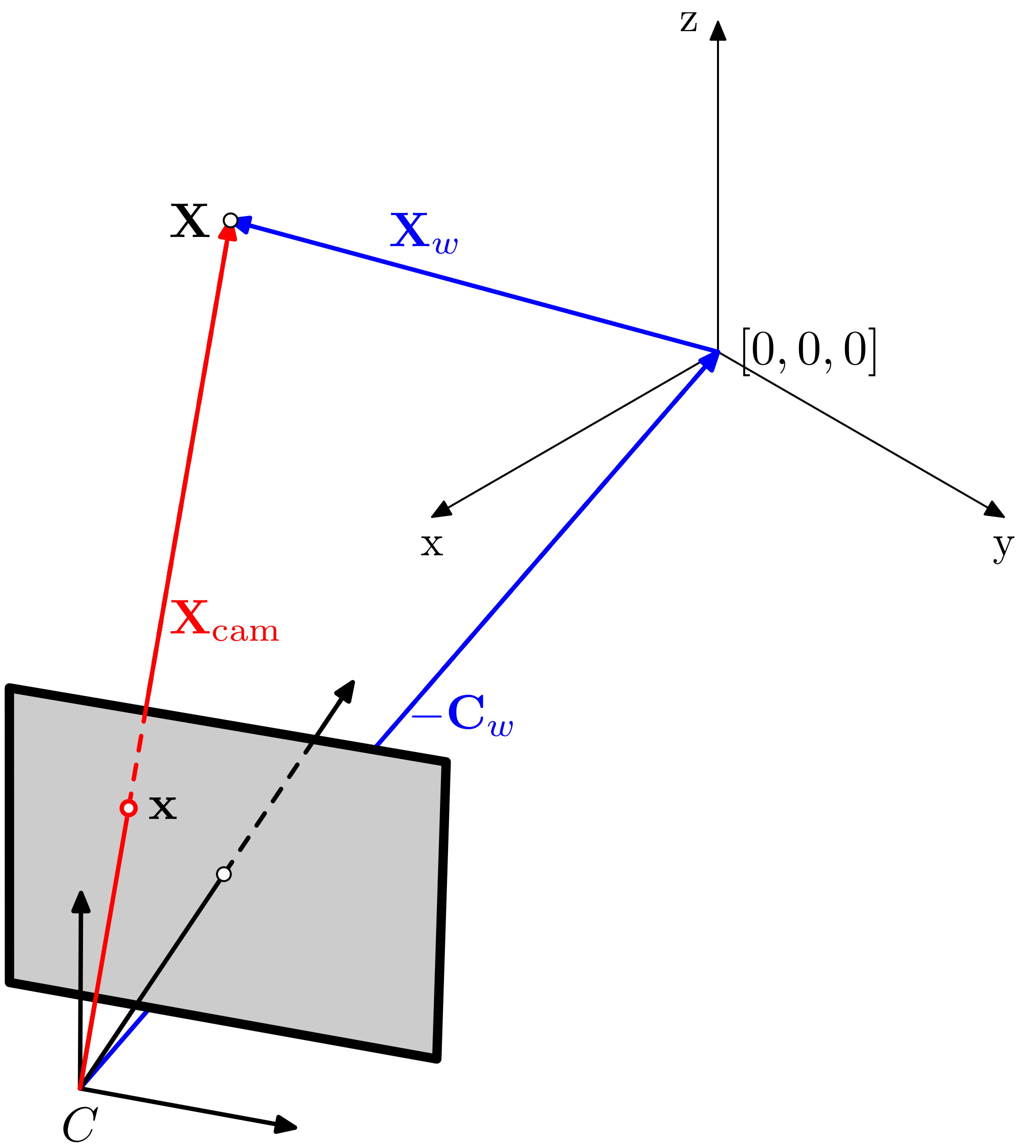
$\mathtt{R}$ is a 3×3 rotation matrix. The point coordinates $\mathbf{X}$ are now in the camera frame.

Use homogeneous coordinates to get a matrix equation

$$\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\mathbf{C}_w \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_w \\ 1 \end{bmatrix}$$

The camera center $\mathbf{C}_w$ is often replaced by the translation vector

$$\mathbf{t} = -\mathtt{R}\mathbf{C}_w$$

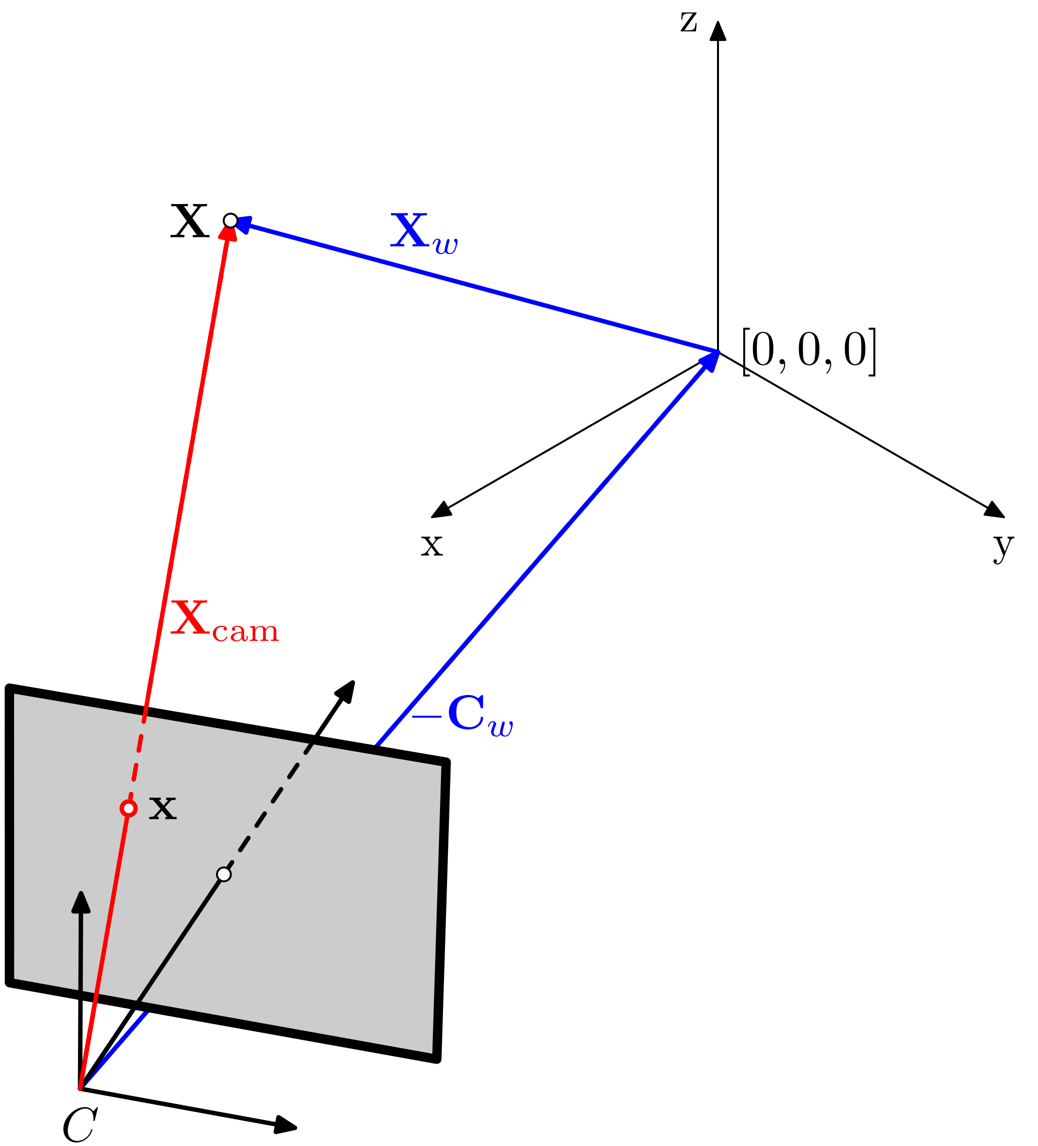


External (extrinsic parameters)

The translation vector $\mathbf{t}$ and the rotation matrix $\mathtt{R}$ are called External parameters of the camera.

$$\mathbf{x} \simeq \mathtt{K} \left[\; \mathtt{I} | \mathbf{0} \; \right] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

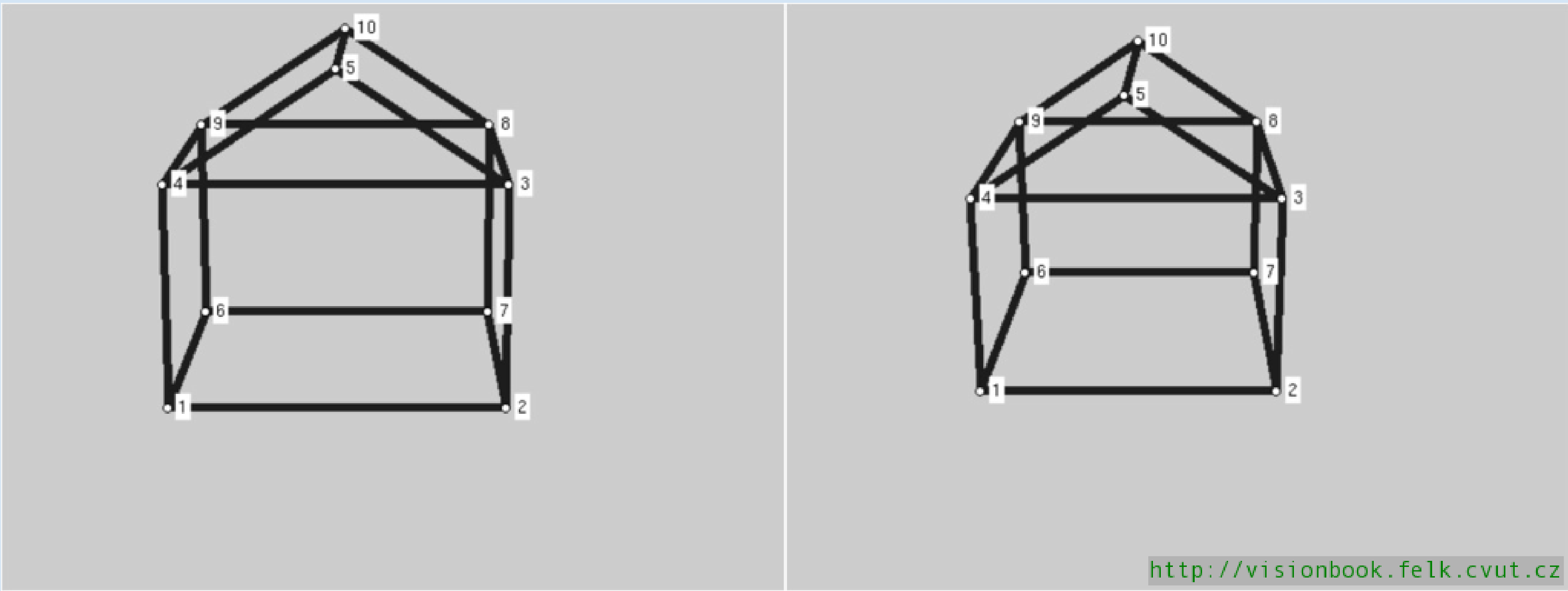
$$\lambda \mathbf{x} = \mathtt{K} \left[\; \mathtt{R} \quad \mathbf{t} \; \right] \begin{bmatrix} \mathbf{X}_w \\ 1 \end{bmatrix}$$



Camera parameters (so far): $f, \mathtt{R}, \mathbf{t}$

Is it all? What can we model?

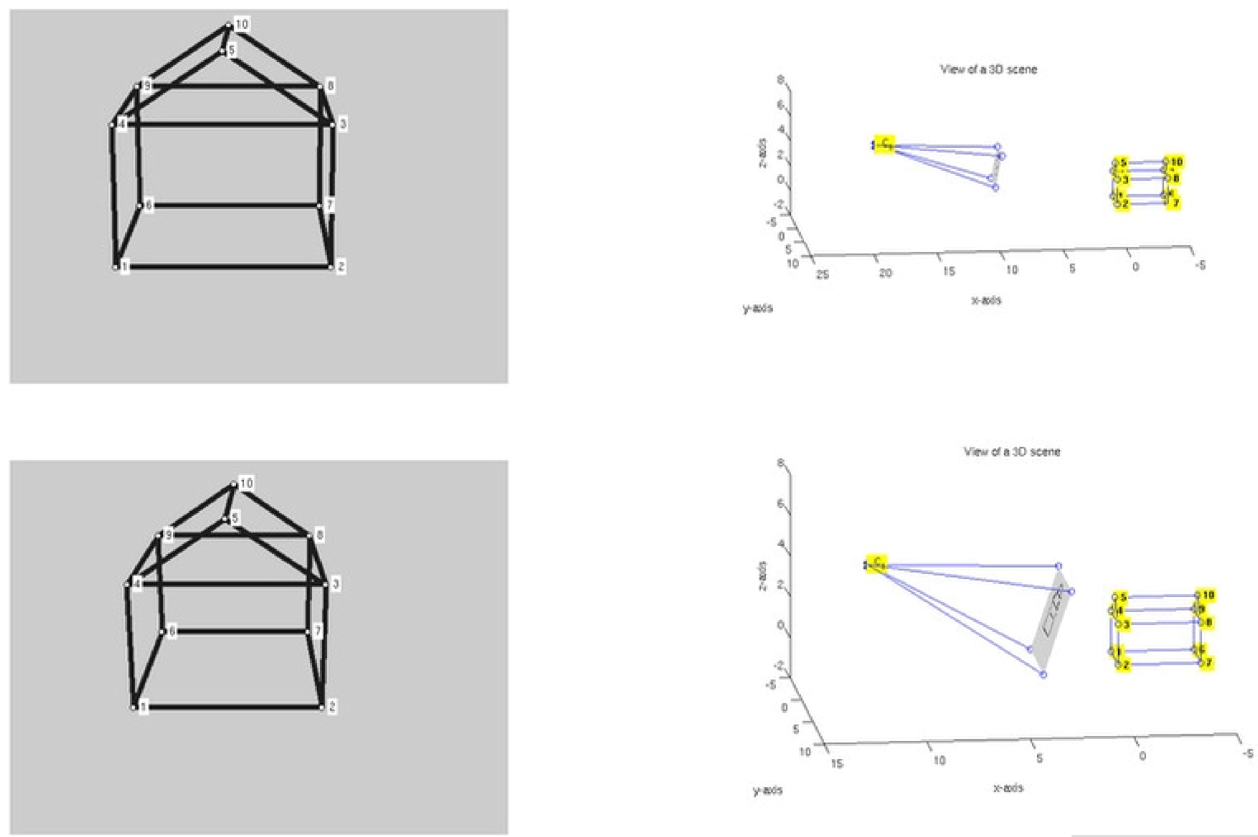
What is the geometry good for?



video: Zoom out vs. motion away from scene

- How would you characterize the difference?
- Would you guess the motion type?

What is the geometry good for?



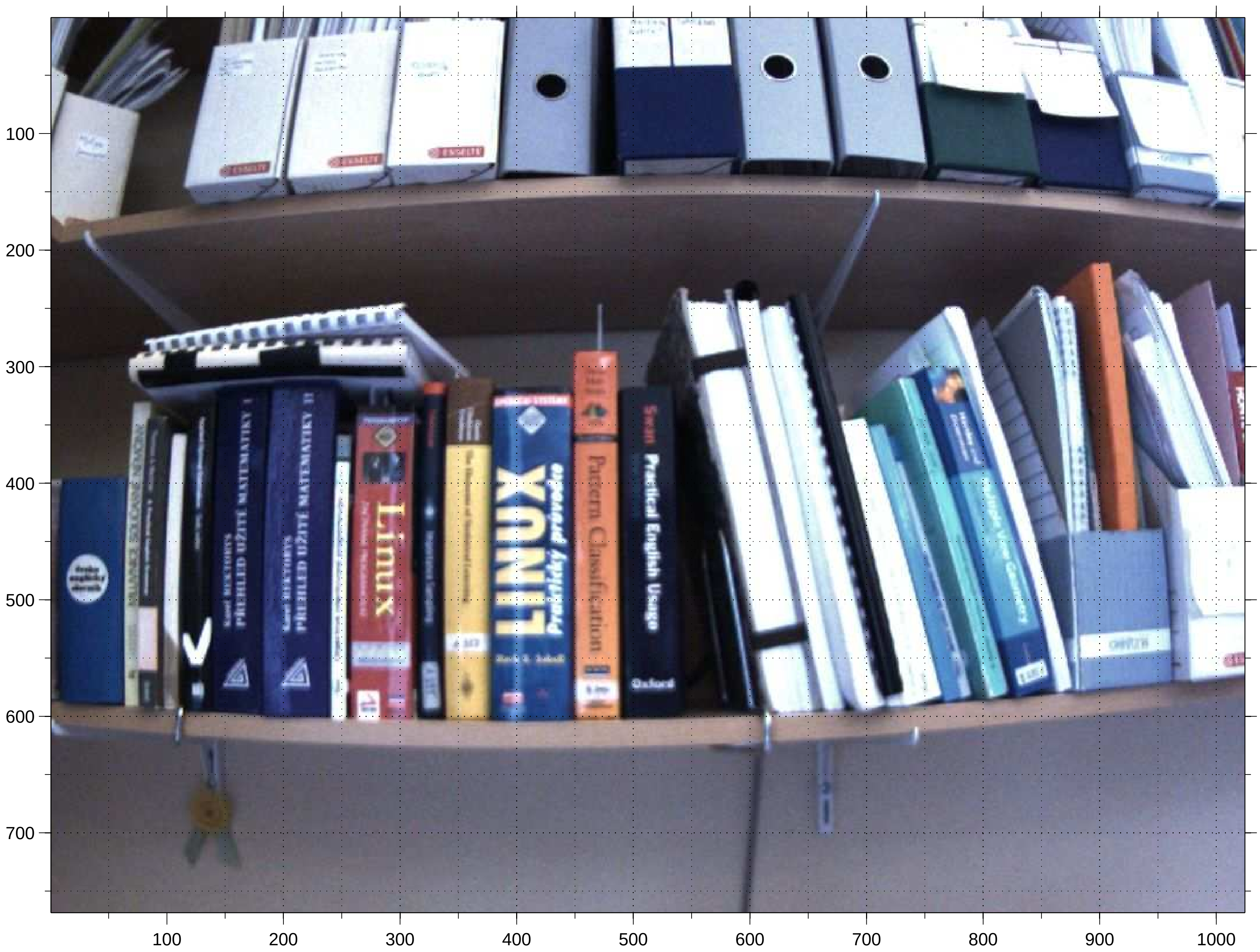
http://visionbook.felk.cvut.cz

video: Zoom out vs. motion away from scene

Enough geometry[5], look at real images

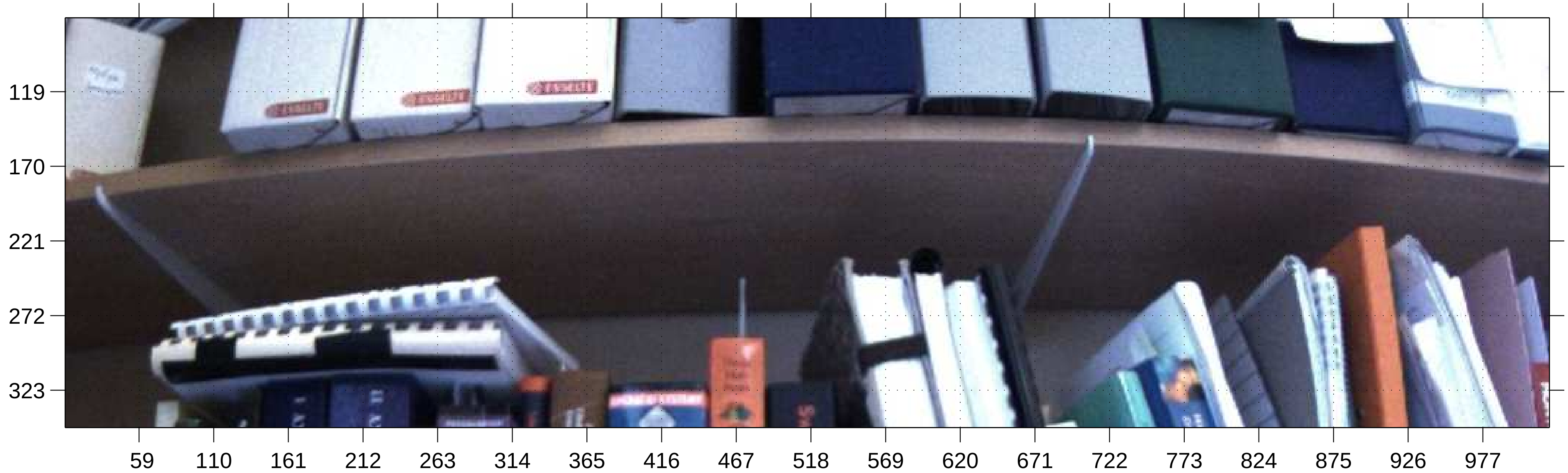
[5] just for a moment

From geometry to pixels and back again

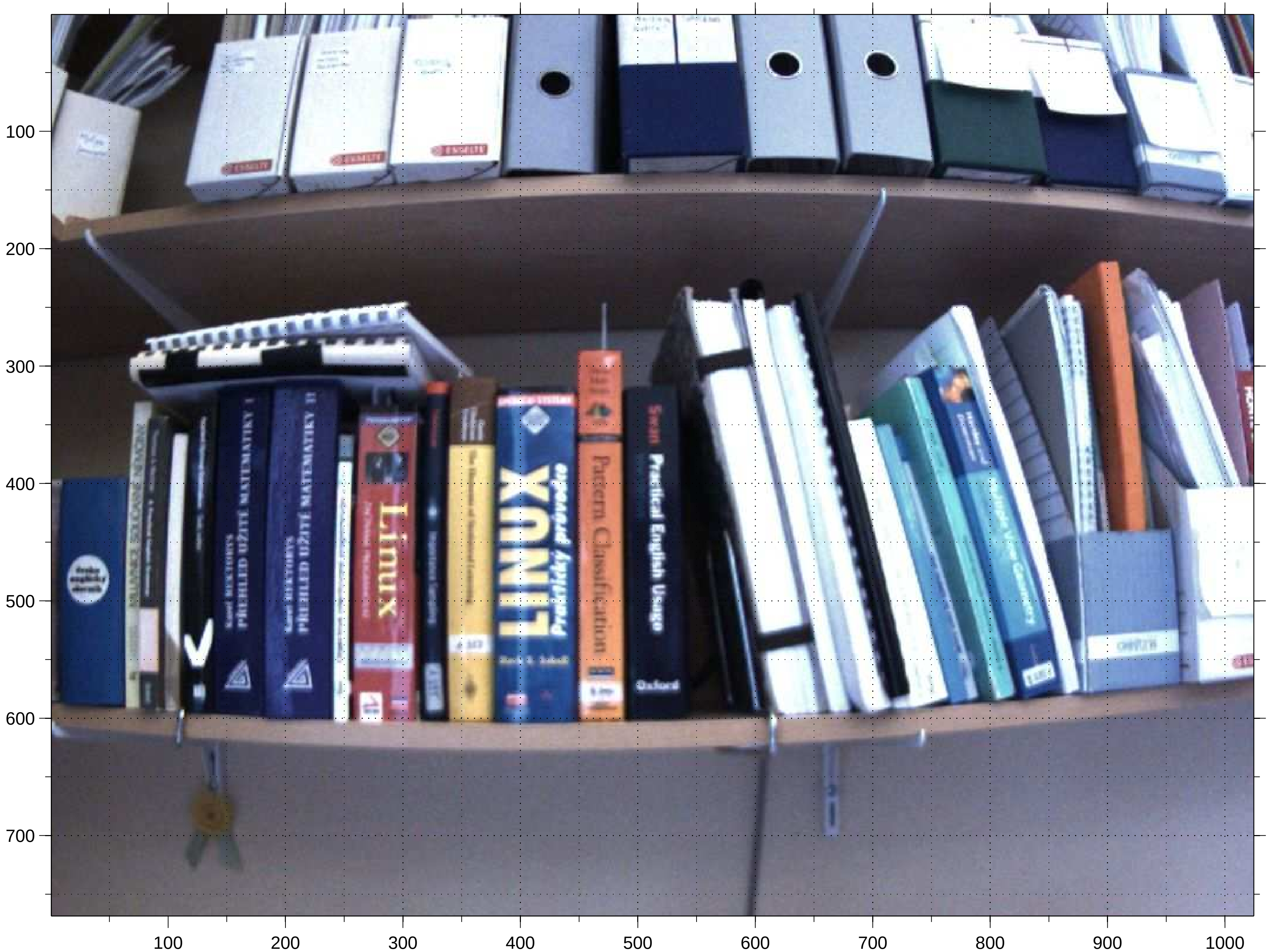


Problems with pixels

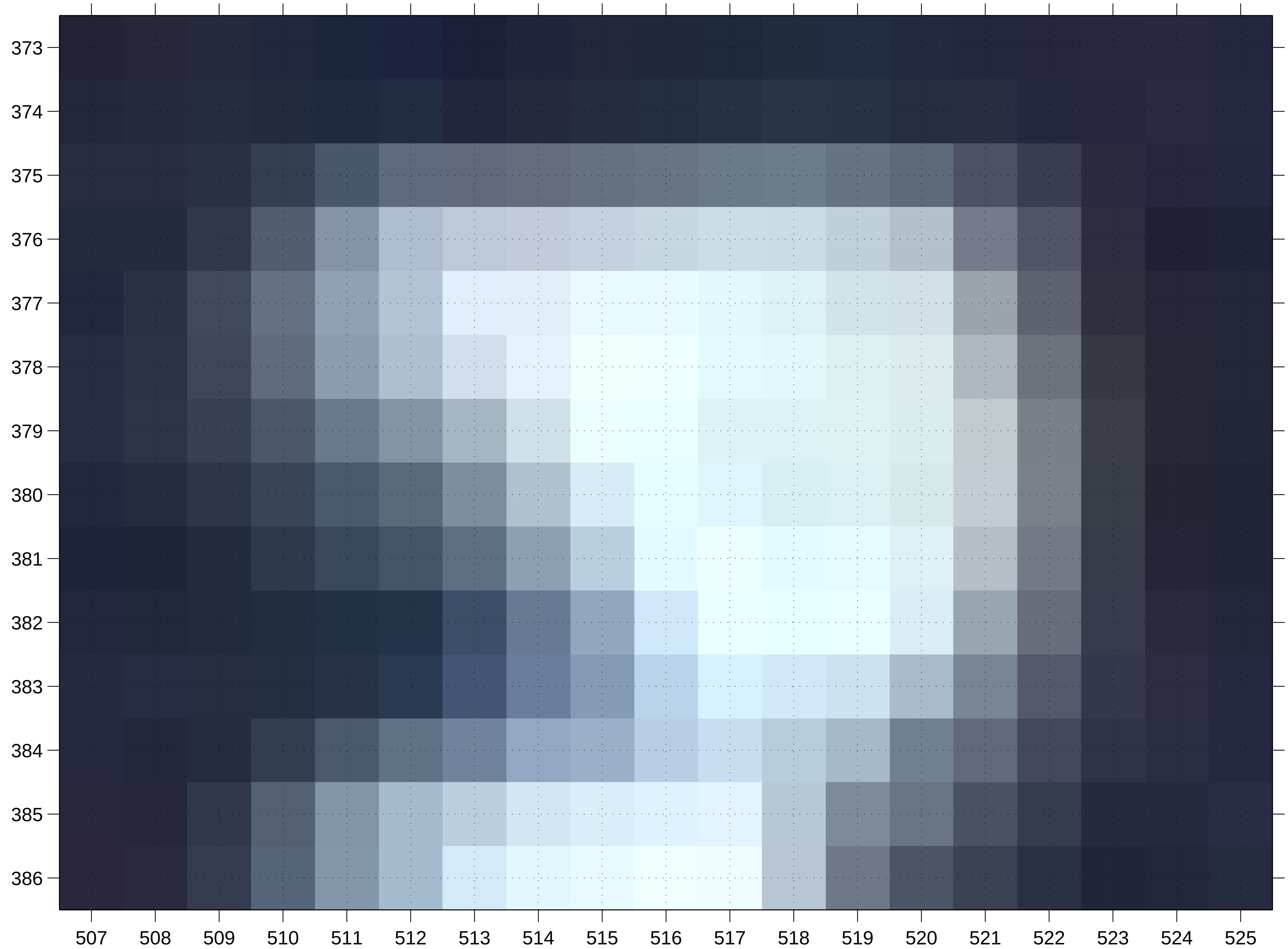
Is this a stright line?



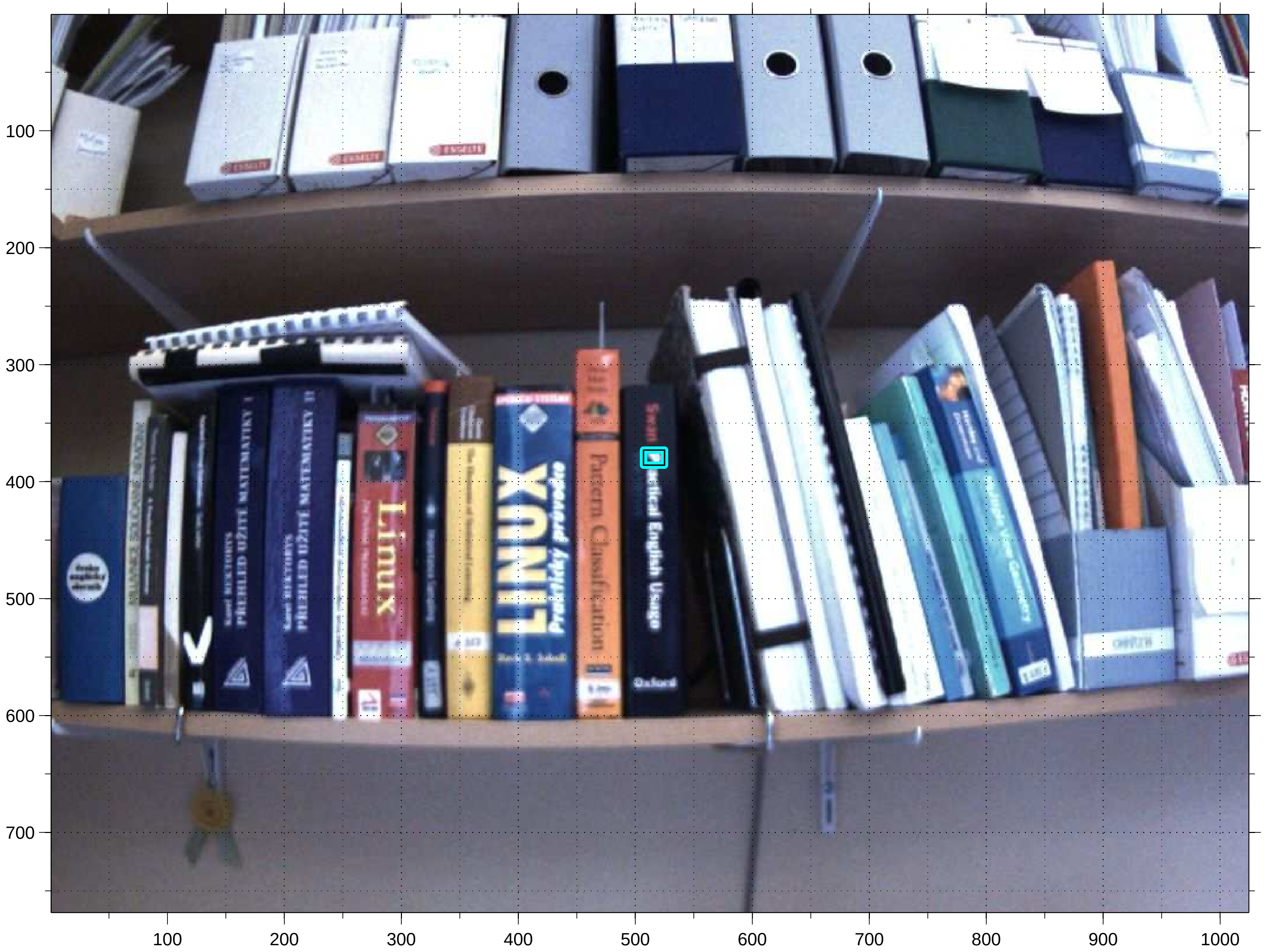
Problems with pixels



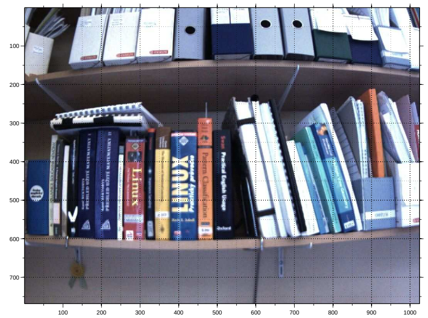
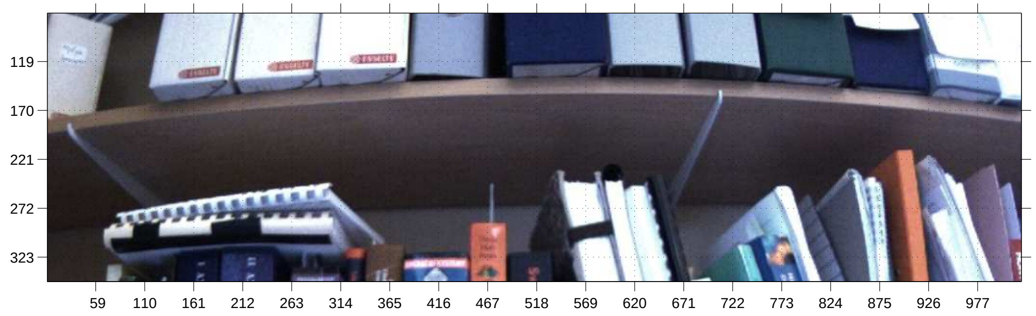
What are we looking at?

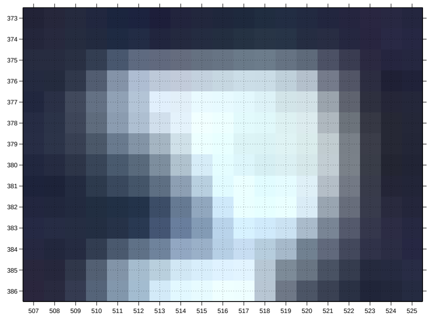


Did you recognize it?



Pixel images revisited



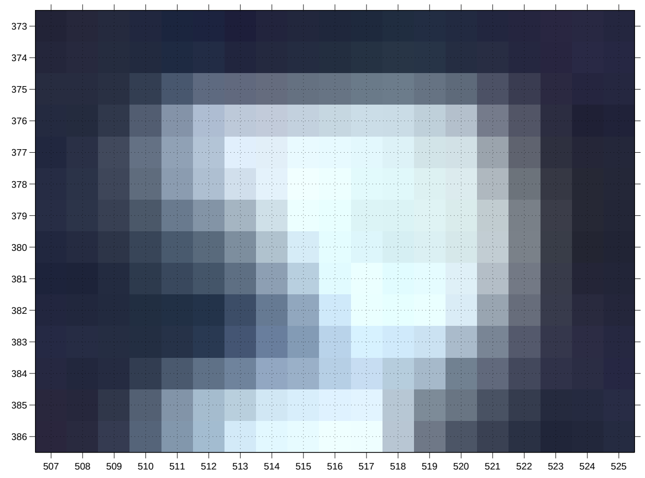
- There are no negative coordinates. Where is the principal point?
- Lines are not lines any more.
- Pixels, considered independently, do not carry much information.

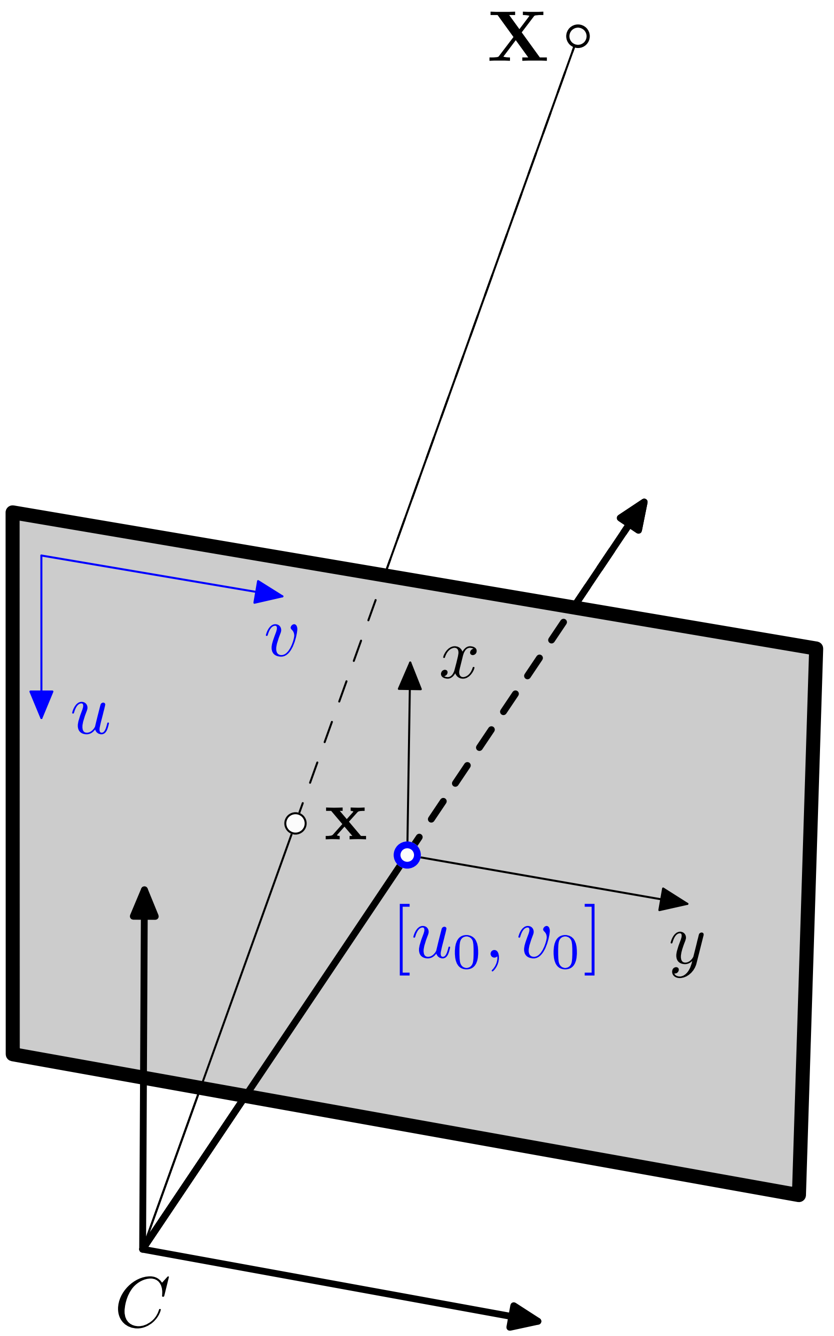
Pixel coordinate system

Assume normalized geometrical coordinates $\mathbf{x} = [x, y, 1]^\top$

$$\begin{aligned} u &= m_u(-x) + u_0 \\ v &= m_v y + v_0 \end{aligned}$$

where m_u, m_v are sizes of the pixels and $[u_0, v_0]^\top$ are coordinates of the principal point.





Put pixels and geometry together

From 3D to image coordinates: $$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathtt{R} & \mathbf{t} \end{bmatrix} \mathbf{X}_{[4\times 1]}$$

From normalized coordinates to pixels: $$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -m_u & 0 & u_0 \\ 0 & m_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Put them together: $$\frac{1}{\lambda} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -fm_u & 0 & u_0 \\ 0 & fm_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathtt{R} & \mathbf{t} \end{bmatrix} \mathbf{X}$$

Finally: $\mathbf{u} \simeq \mathtt{K} \begin{bmatrix} \mathtt{R} & \mathbf{t} \end{bmatrix} \mathbf{X}$

Introducing a 3×4 camera projection matrix $\mathtt{P}$: $\boxed{\mathbf{u} \simeq \mathtt{P}\mathbf{X}}$

Non-linear distortion

Several models exist. Less standardized than the linear model. We will consider a simple on-parameter radial distortion. $\mathbf{x}_n$ denote the linear image coordinates, $\mathbf{x}_d$ the distorted ones.

$$\mathbf{x}_d = (1 + \kappa r^2)\mathbf{x}_n$$

where κ is the distortion parameter, and $r^2 = x_n^2 + y_n^2$ is the distance from the principal point.

Observable are the distorted pixel coordinates

$$\mathbf{u}_d = \mathtt{K}\mathbf{x}_d$$

Assume that we know κ. How to get the lines back?

Undoing Radial Distortion

From pixels to distorted image coordinates: $\mathbf{x}_d = \mathtt{K}^{-1}\mathbf{u}_d$

From distorted to linear image coordinates: $\mathbf{x}_n = \frac{\mathbf{x_d}}{1+\kappa r^2}$

Where is the problem? $r^2 = x_n^2 + y_n^2$. We have unknowns on both sides of the equation.

Iterative solution:

1. initialize $\mathbf{x}_n = \mathbf{x}_d$
2. $r^2 = x_n^2 + y_n^2$
3. compute $\mathbf{x}_n = \frac{\mathbf{x_d}}{1+\kappa r^2}$
4. go to 2. (and repeat few times)

And back to pixels $\mathbf{u}_n = \mathtt{K}\mathbf{x}_n$

Undoing Radial Distortion

video

Estimation of camera parameters—camera calibration

The goal: estimate the 3×4 camera projection matrix $\mathtt{P}$ and possibly the parameters of the non-linear distortion κ from images.

Assume a known projection $[u, v]^\top$ of a 3D point $\mathbf{X}$ with known coordinates

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1^\top \\ \mathbf{P}_2^\top \\ \mathbf{P}_3^\top \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\frac{\lambda u}{\lambda} = \frac{\mathbf{P}_1^\top \mathbf{X}}{\mathbf{P}_3^\top \mathbf{X}} \quad \text{and} \quad \frac{\lambda v}{\lambda} = \frac{\mathbf{P}_2^\top \mathbf{X}}{\mathbf{P}_3^\top \mathbf{X}}$$

Re-arrange and assume[6] $\lambda \neq 0$ to get set of homegeneous equations

$$\begin{aligned} u\mathbf{X}^\top \mathbf{P}_3 - \mathbf{X}^\top \mathbf{P}_1 &= 0 \\ v\mathbf{X}^\top \mathbf{P}_3 - \mathbf{X}^\top \mathbf{P}_2 &= 0 \end{aligned}$$

[6] see some notes about $\lambda = 0$ in the talk notes

Estimation of the $\mathtt{P}$ matrix

$$
\begin{aligned}
u\mathbf{X}^\top\mathbf{P}_3 - \mathbf{X}^\top\mathbf{P}_1 &= 0 \\
v\mathbf{X}^\top\mathbf{P}_3 - \mathbf{X}^\top\mathbf{P}_2 &= 0
\end{aligned}
$$

Re-shuffle into a matrix form:

$$
\underbrace{\begin{bmatrix} -\mathbf{X}^\top & \mathbf{0}^\top & u\mathbf{X}^\top \\ \mathbf{0}^\top & -\mathbf{X}^\top & v\mathbf{X}^\top \end{bmatrix}}_{\mathtt{A}_{[2\times 12]}} \underbrace{\begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix}}_{\mathbf{p}_{[12\times 1]}} = \mathbf{0}_{[2\times 1]}
$$

A correspondece $\mathbf{u}_i \leftrightarrow \mathbf{X}_i$ forms two homogeneous equations. $\mathtt{P}$ has 12 parameters but scale does not matter. We need at least 6 2D $\leftrightarrow$ 3D pairs to get a solution. We constitute $\mathtt{A}_{[\geq 12\times 12]}$ data matrix and solve

$$
\mathbf{p}^* = \mathrm{argmin}\|\mathtt{A}\mathbf{p}\| \;\; \text{subject to} \;\; \|\mathbf{p}\| = 1
$$

which is a constrained LSQ problem. $\mathbf{p}^*$ minimizes algebraic error

Decomposition of $\mathtt{P}$ into the calibration parameters

$$\mathtt{P} = \left[\begin{array}{cc} \mathtt{KR} & \mathtt{K}\mathbf{t} \end{array}\right] \quad \text{and} \quad \mathbf{C} = -\mathtt{R}^{-1}\mathbf{t}$$

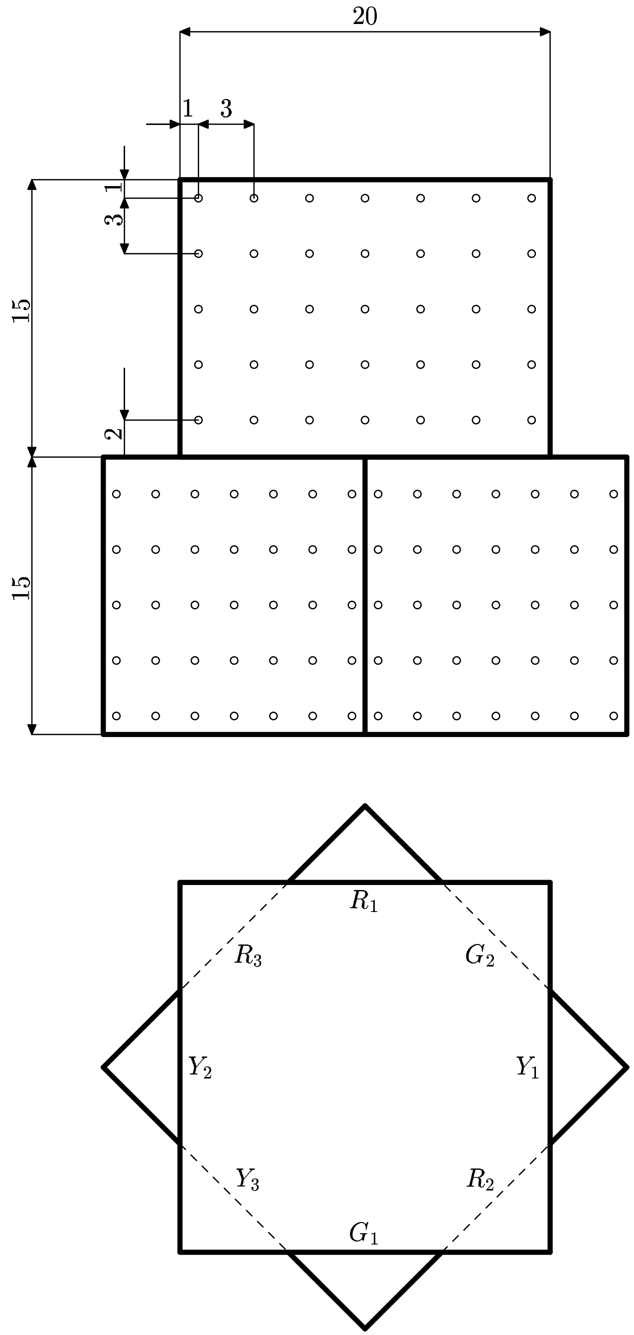
We know that $\mathtt{R}$ should be 3×3 orthonormal, and $\mathtt{K}$ upper triangular.

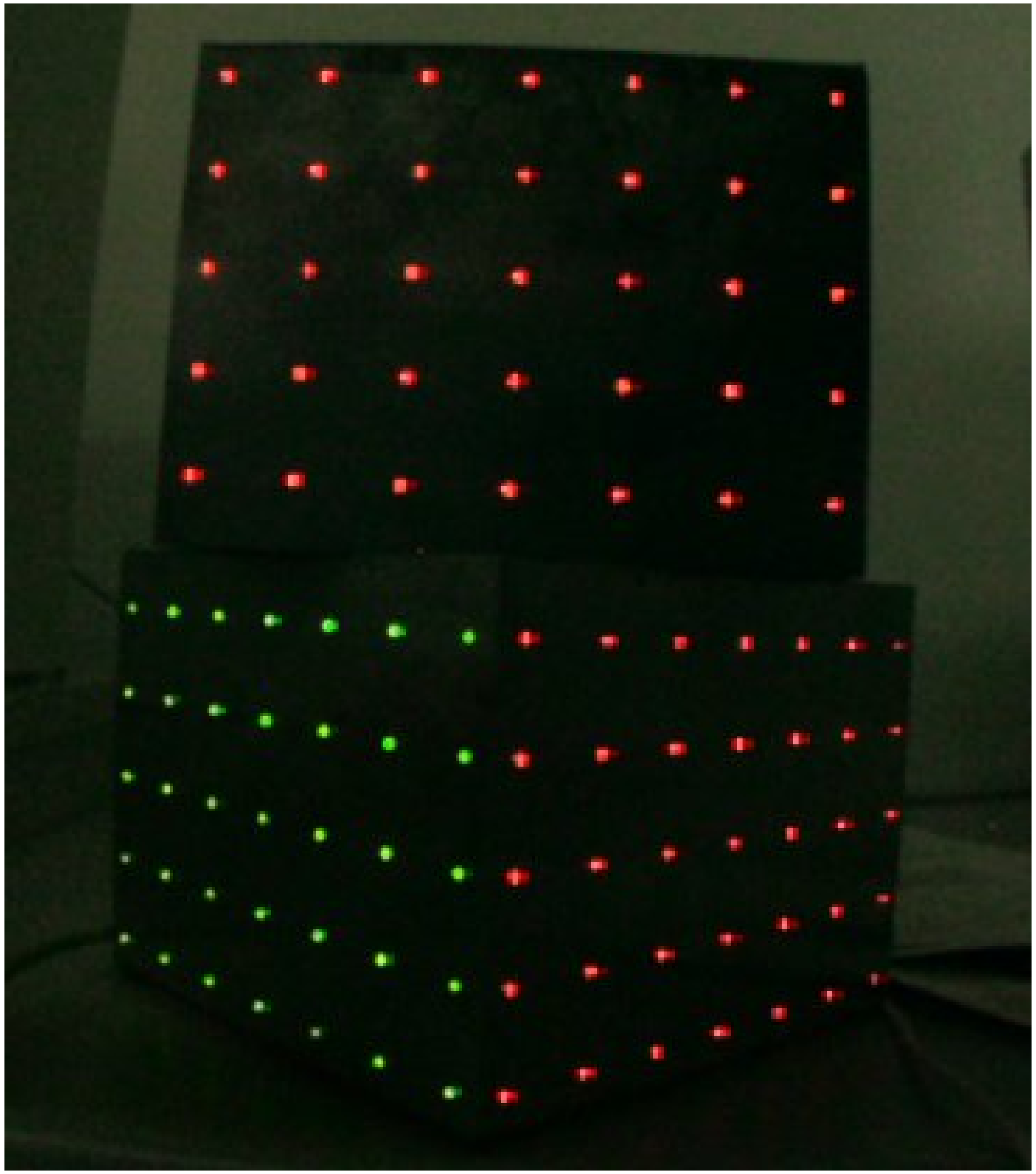
```
P = P./norm(P(3,1:3));

[K,R] = rq(P(:,1:3));
t = inv(K)*P(:,4);
C = -R'*t;
```

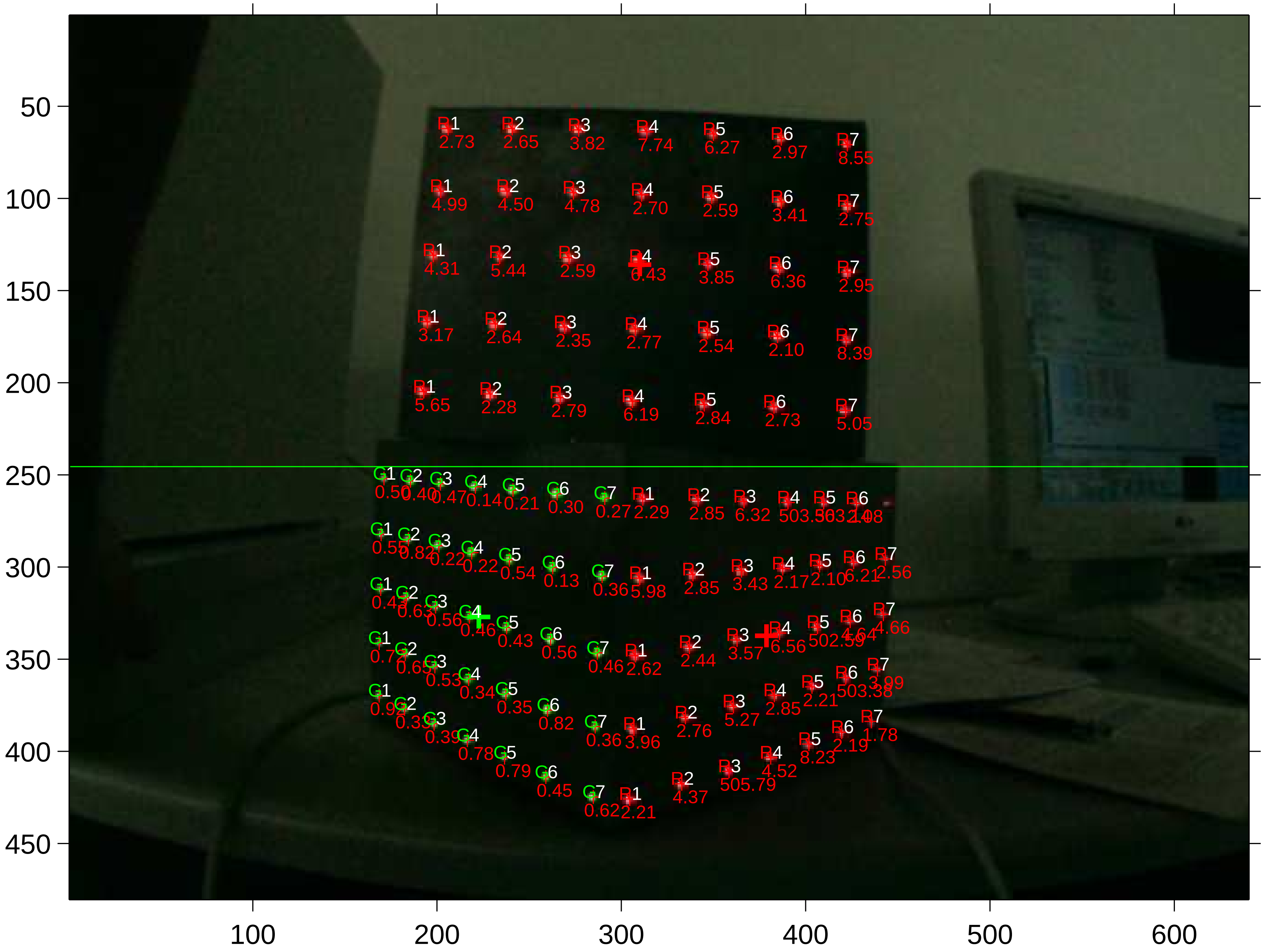
See the slide notes for more details.

An example of an calibration object

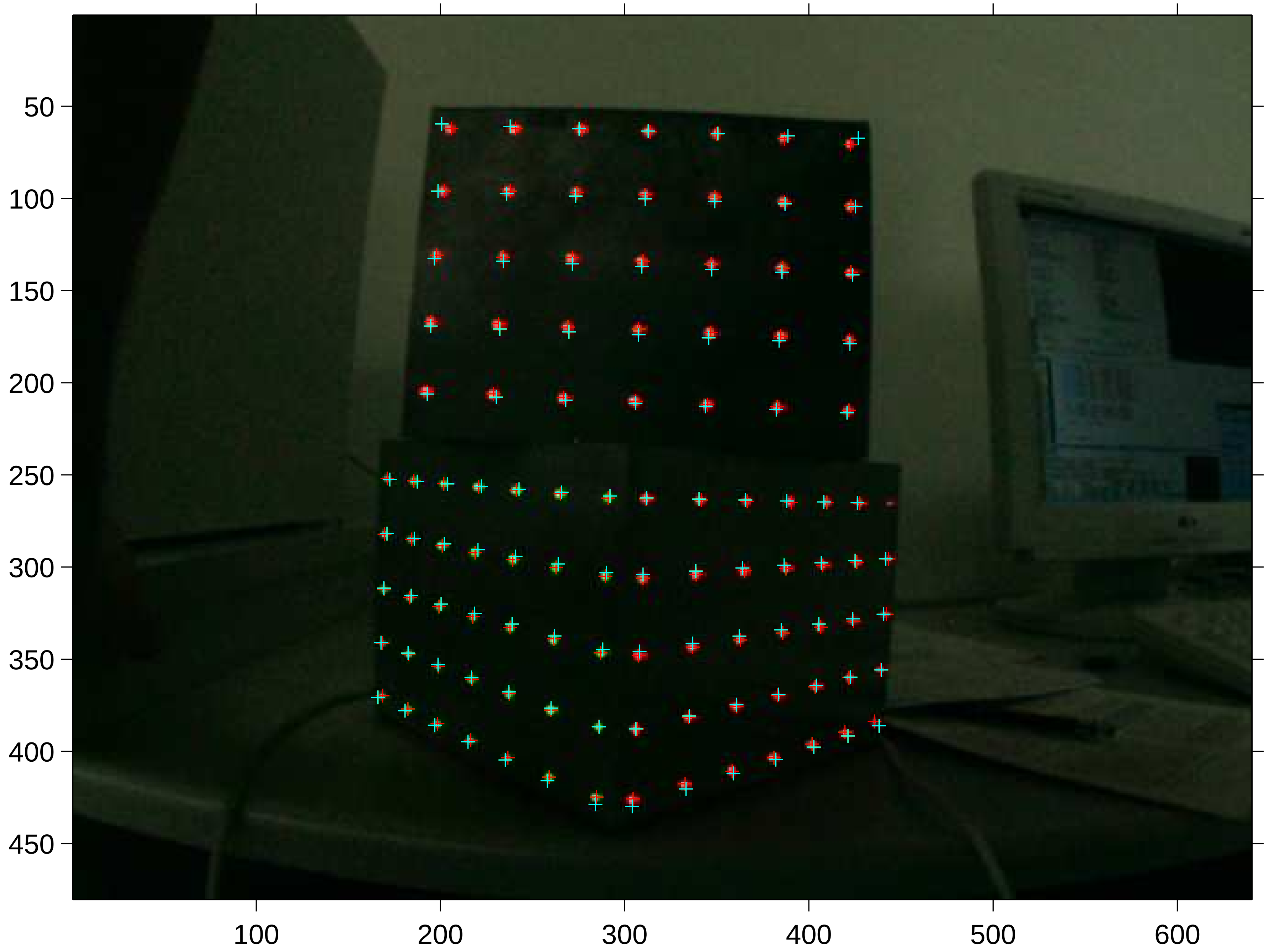




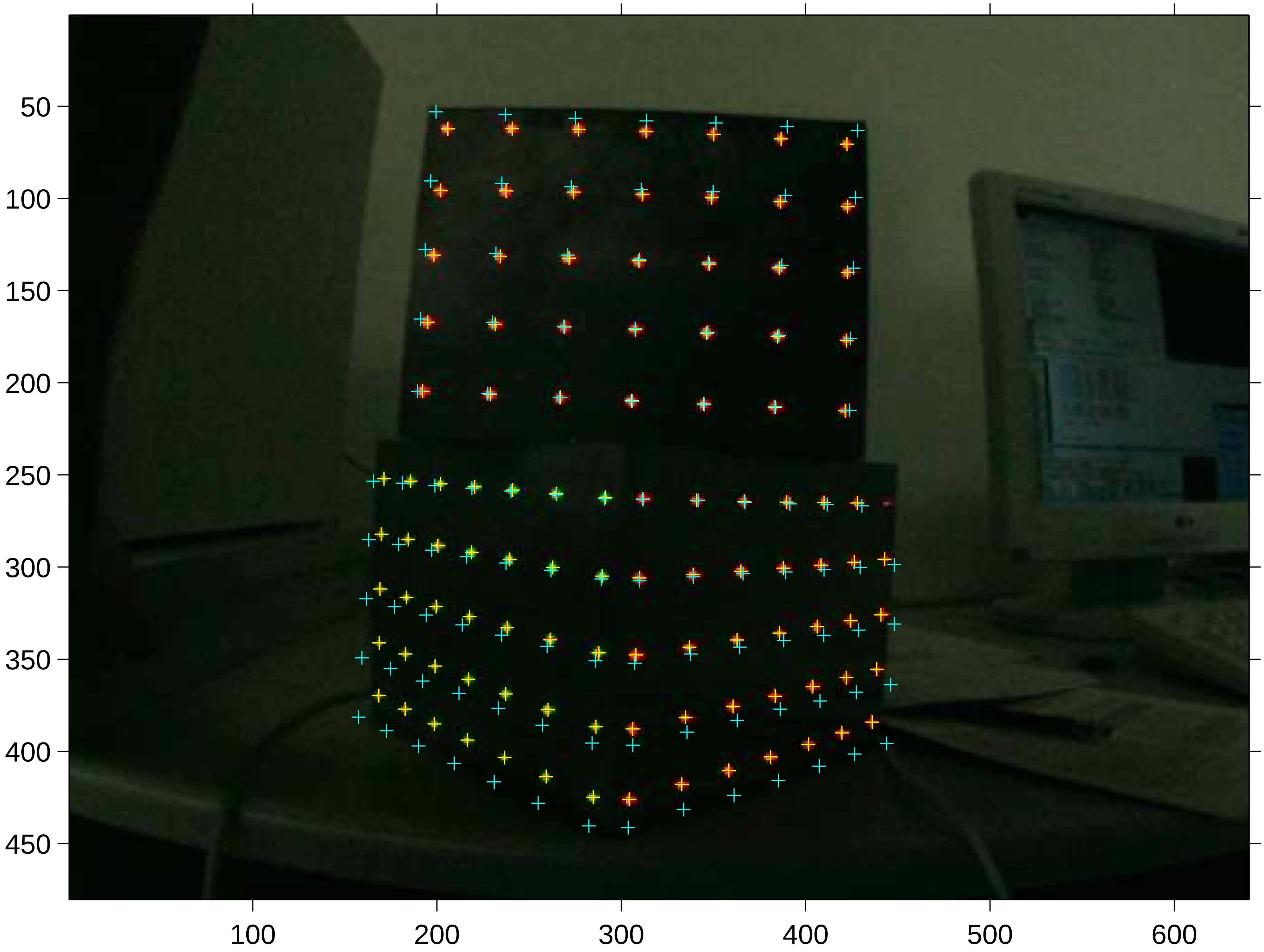
2D projections localized



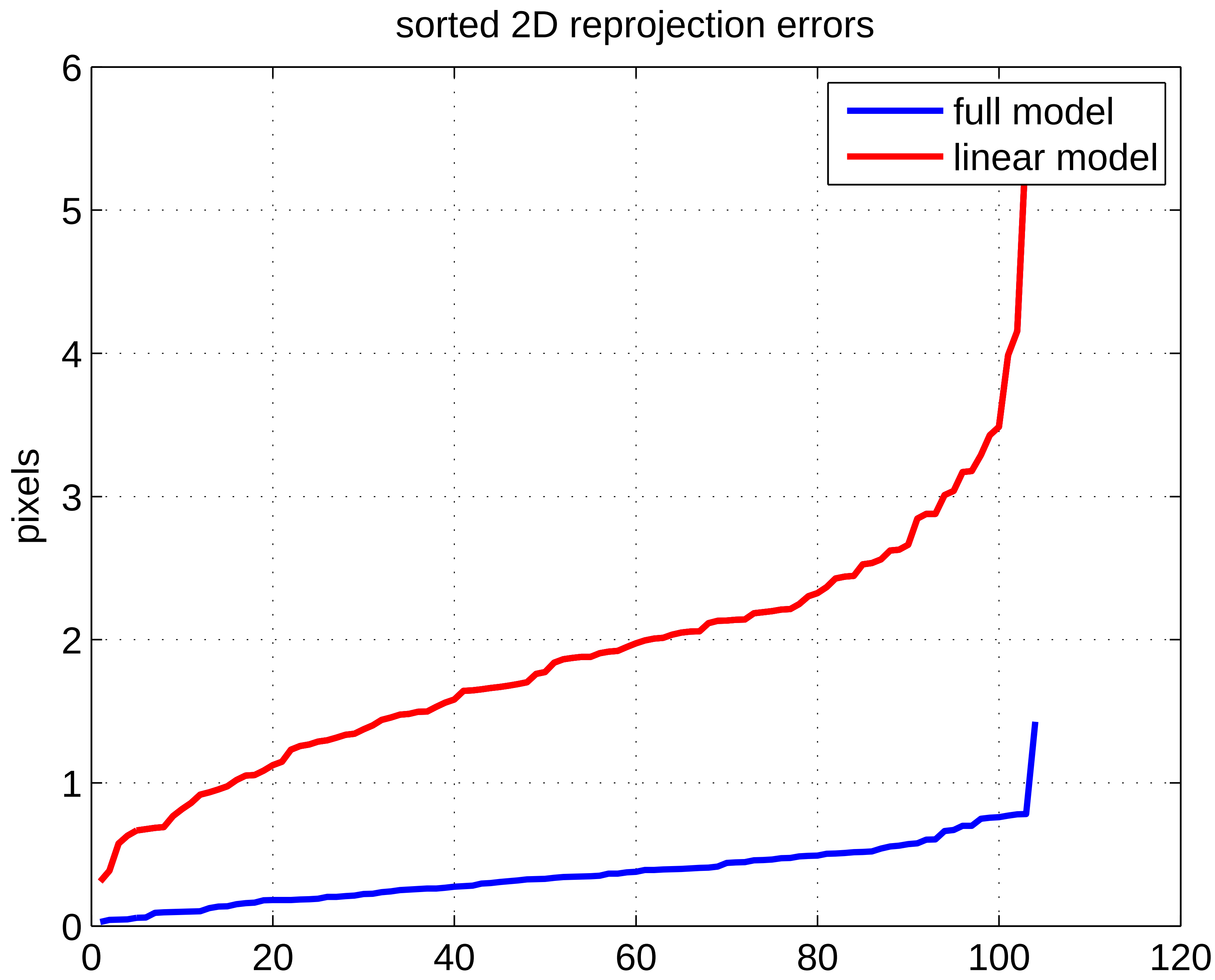
Reprojection for linear model



Reprojection for full model



Reprojection errors—comparison between full and linear model



References

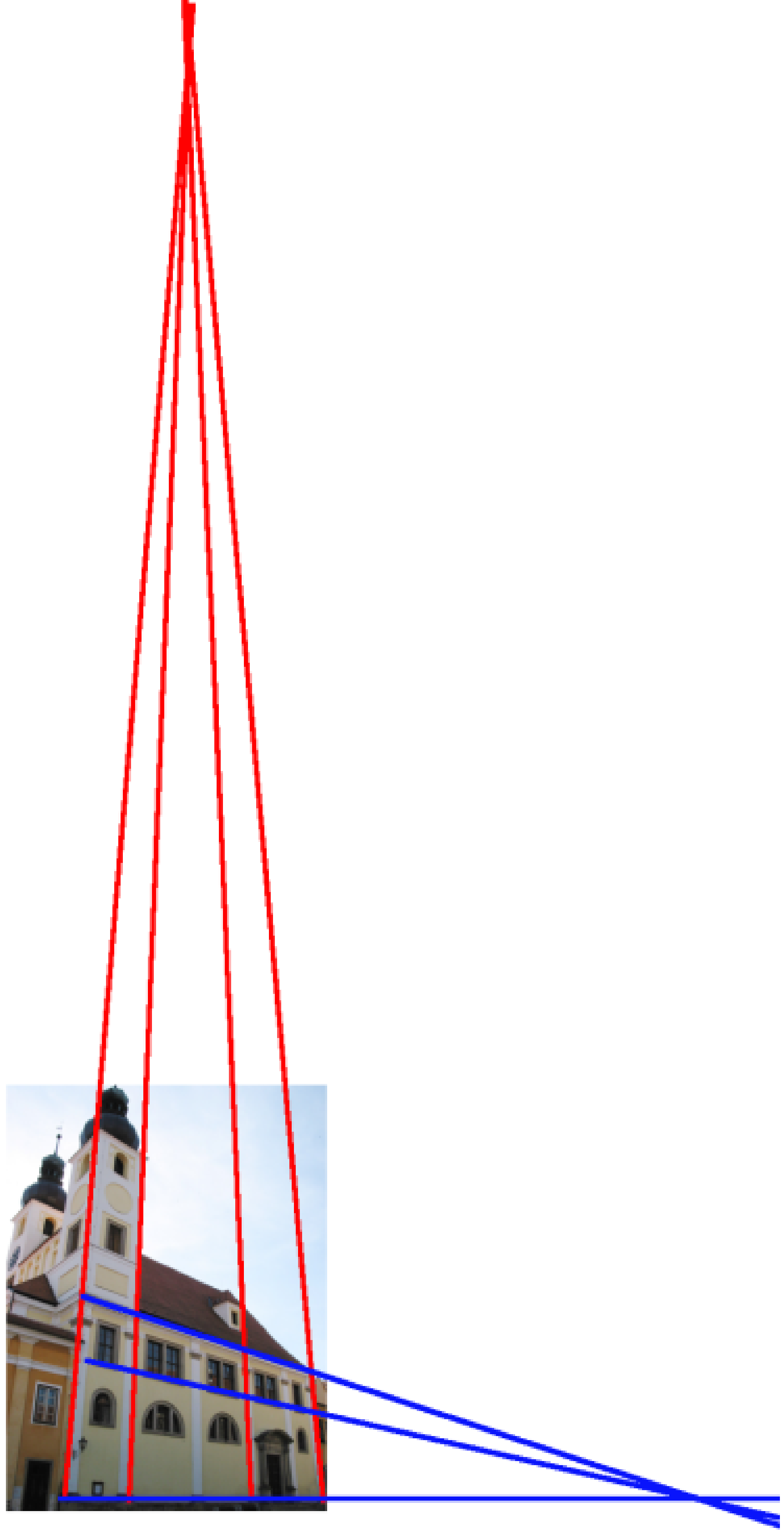
The book [2] is the ultimate reference. It is a must read for anyone wanting use cameras for 3D computing.

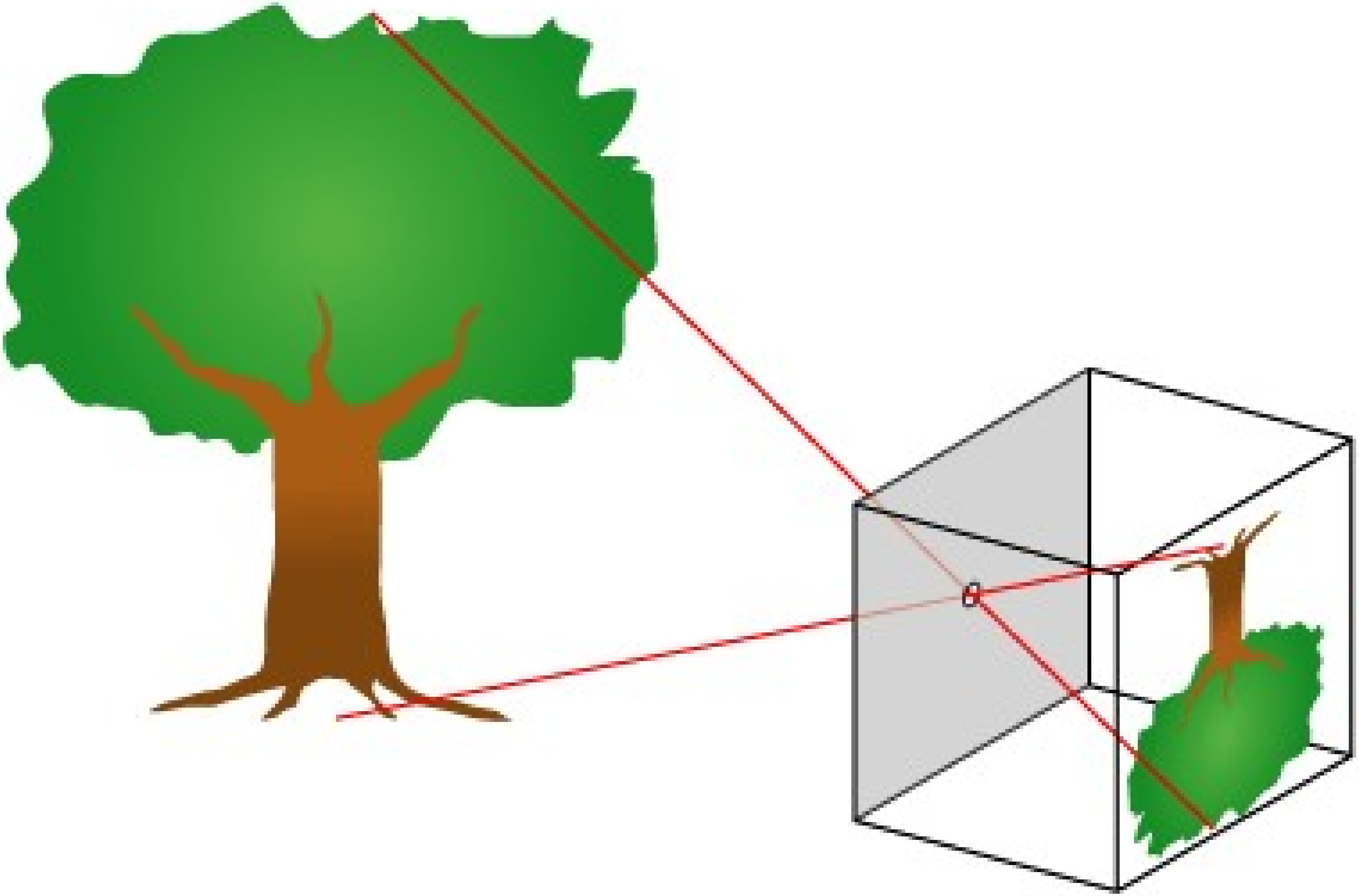
Details about matrix decompositions used throughout the lecture can be found at [1]

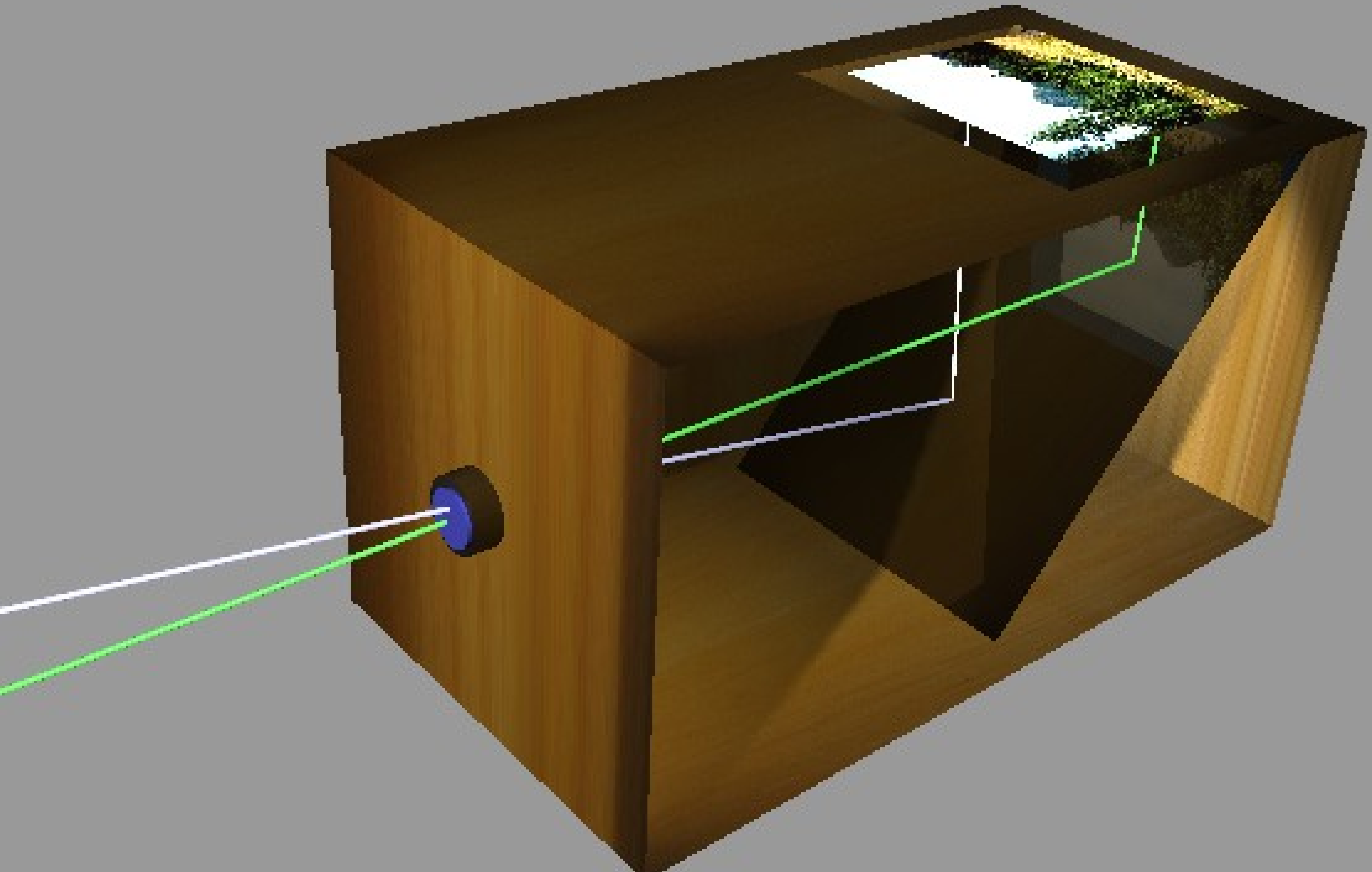
[1] Gene H. Golub and Charles F. Van Loan. Matrix Computation. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, USA, 3rd edition, 1996.

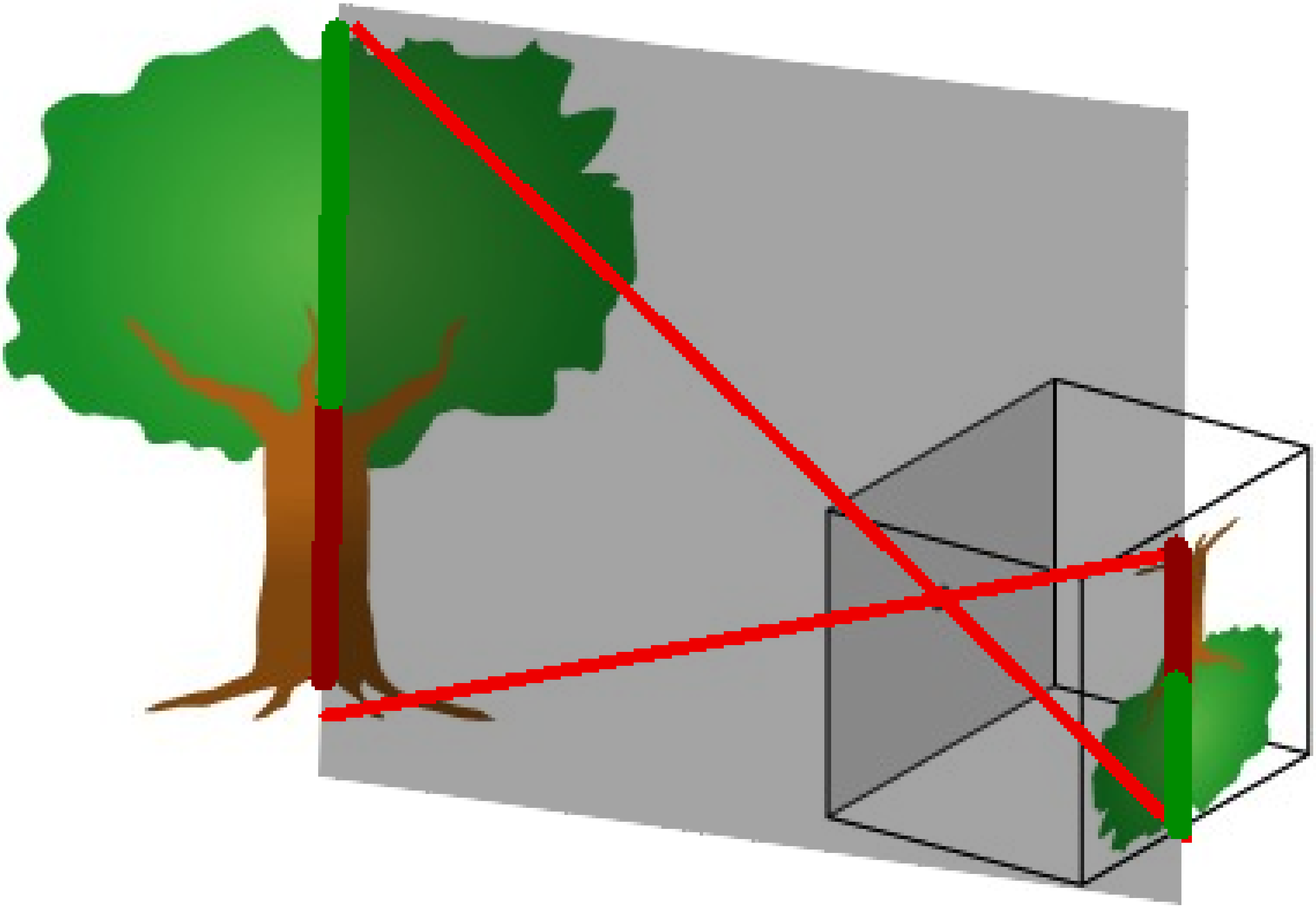
[2] Richard Hartley and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge University, Cambridge, 2nd edition, 2003.

End









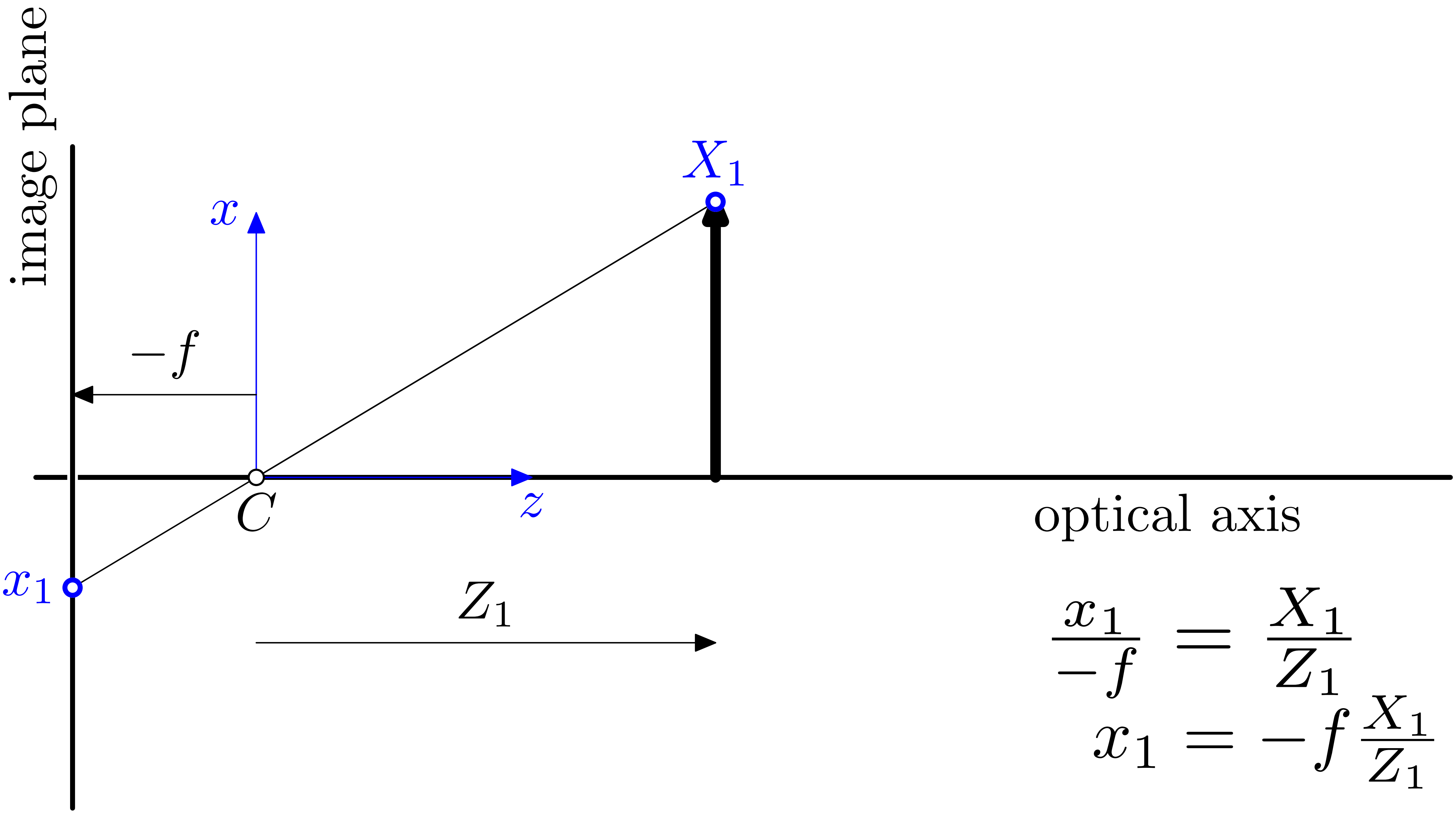


image plane
X_1
x
$-f$
C
z
optical axis
x_1
Z_1
$\frac{x_1}{-f} = \frac{X_1}{Z_1}$
$x_1 = -f\frac{X_1}{Z_1}$

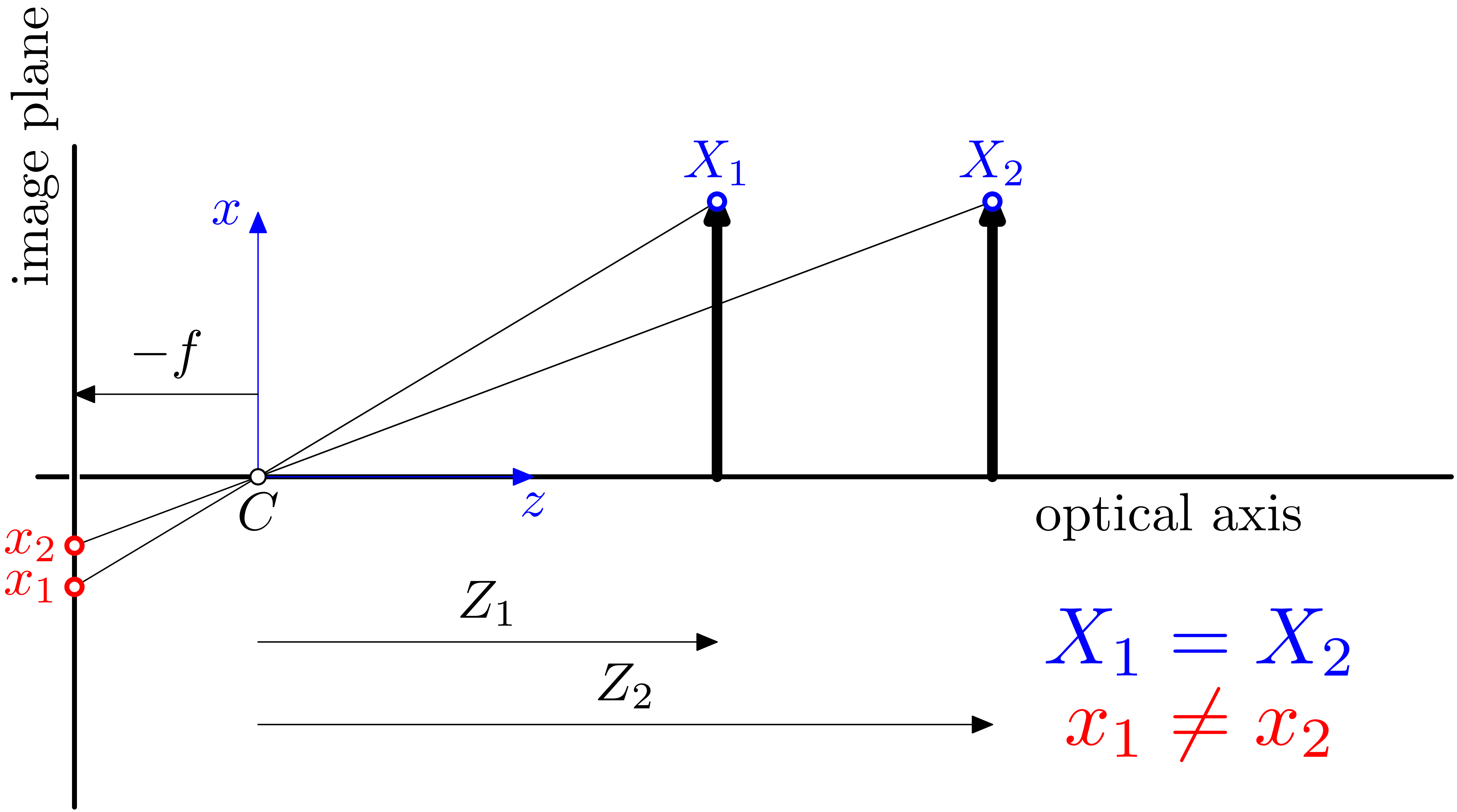


image plane
X_1
X_2
x
$-f$
C
z
optical axis
x_2
x_1
Z_1
Z_2
$X_1 = X_2$
$x_1 \neq x_2$

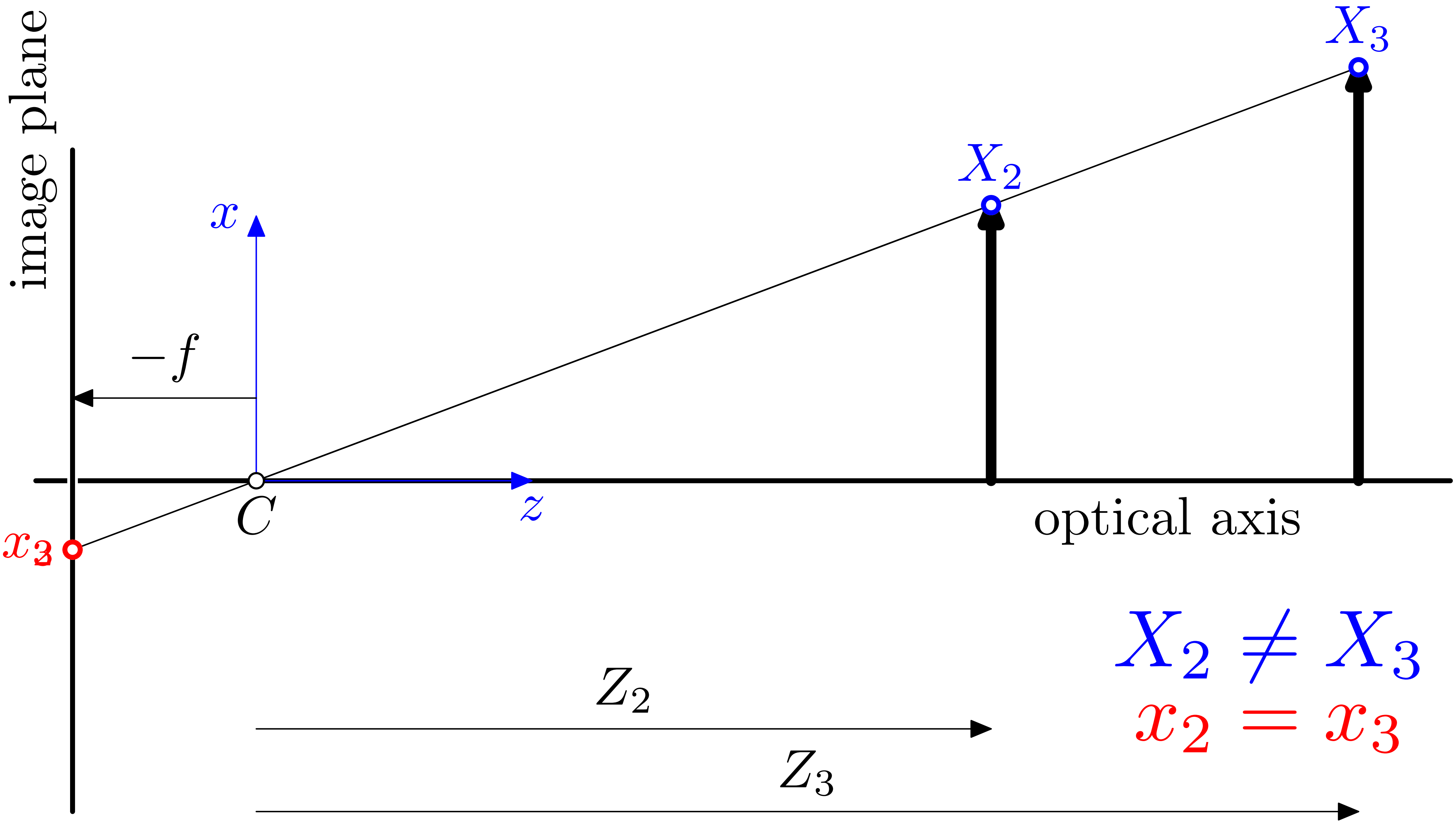


image plane
x
$-f$
C
z
optical axis
X_2
X_3
x_3
x_2
Z_2
Z_3
$X_2 \neq X_3$
$x_2 = x_3$

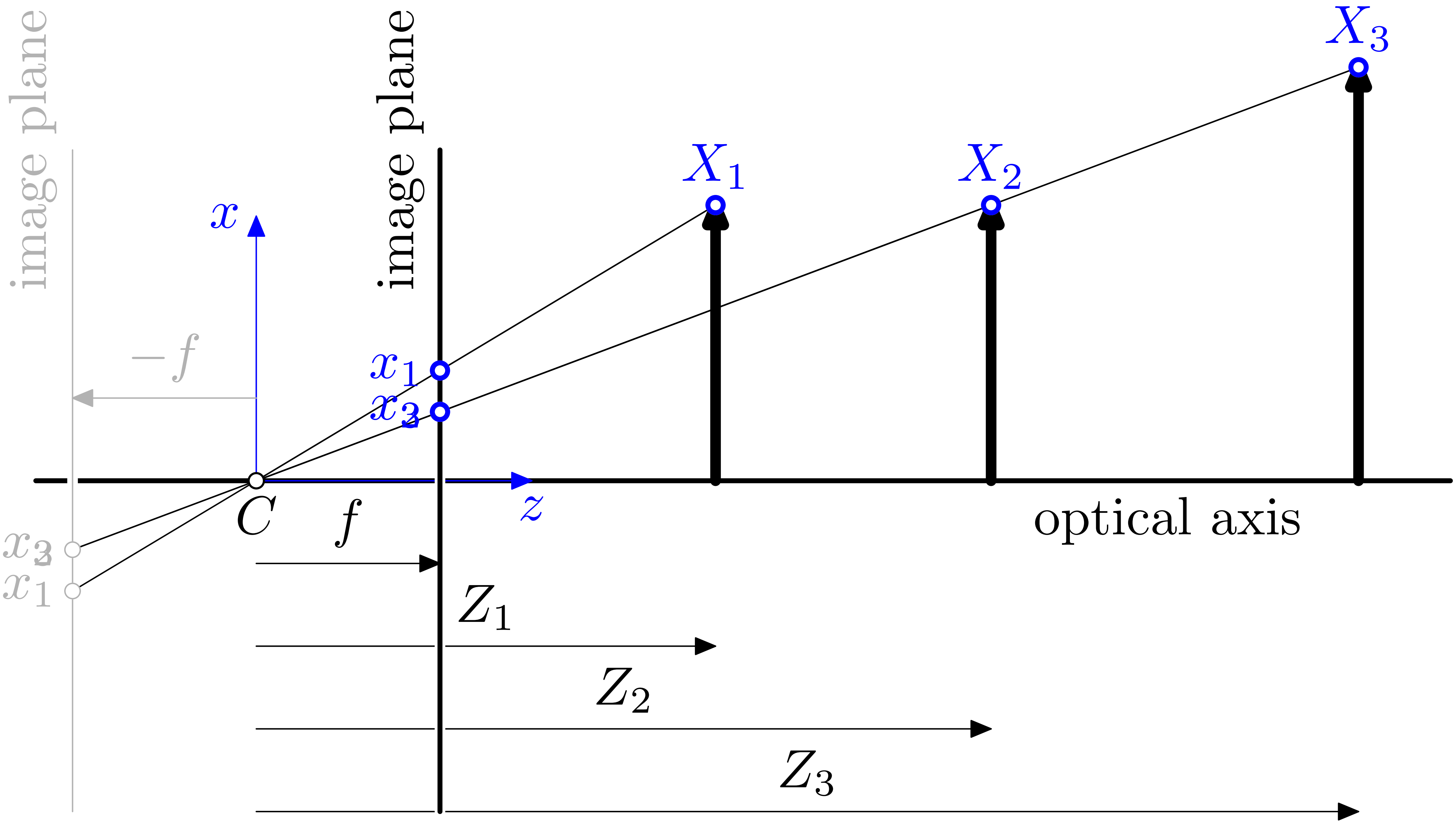
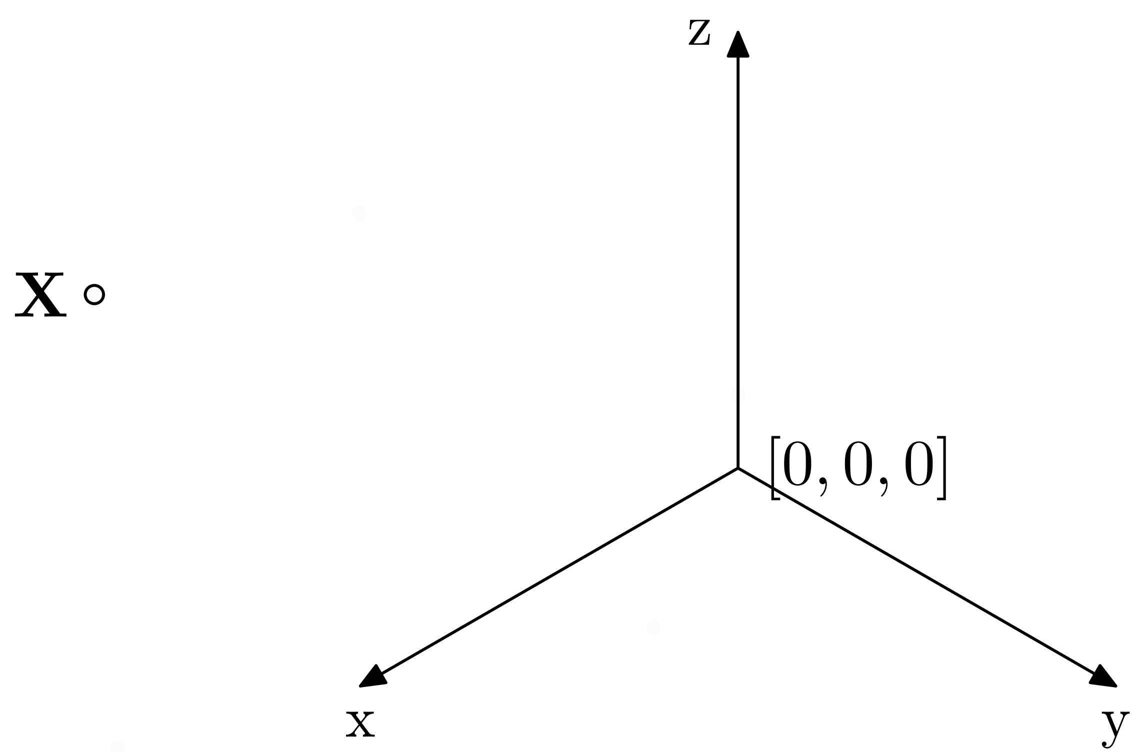
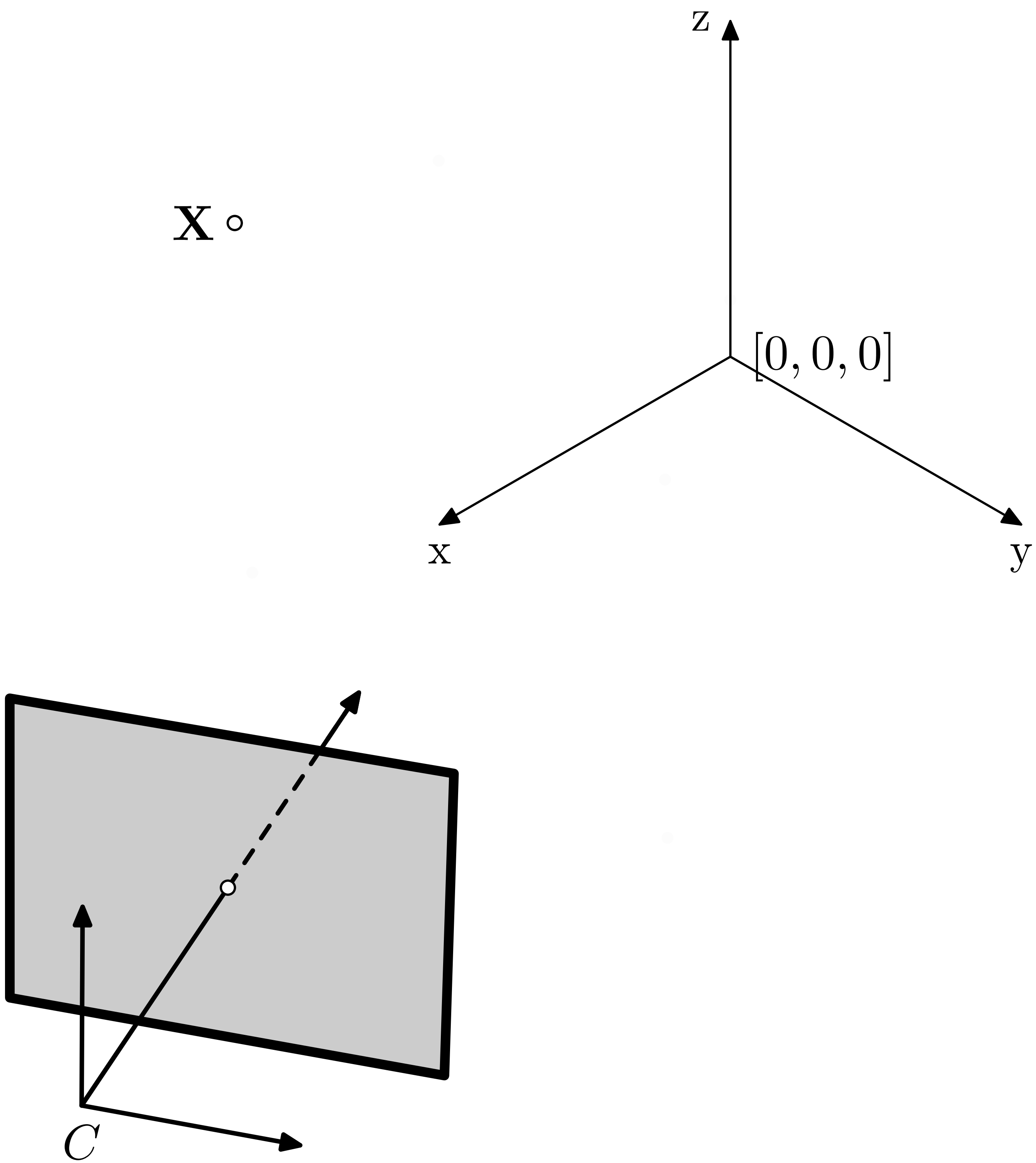
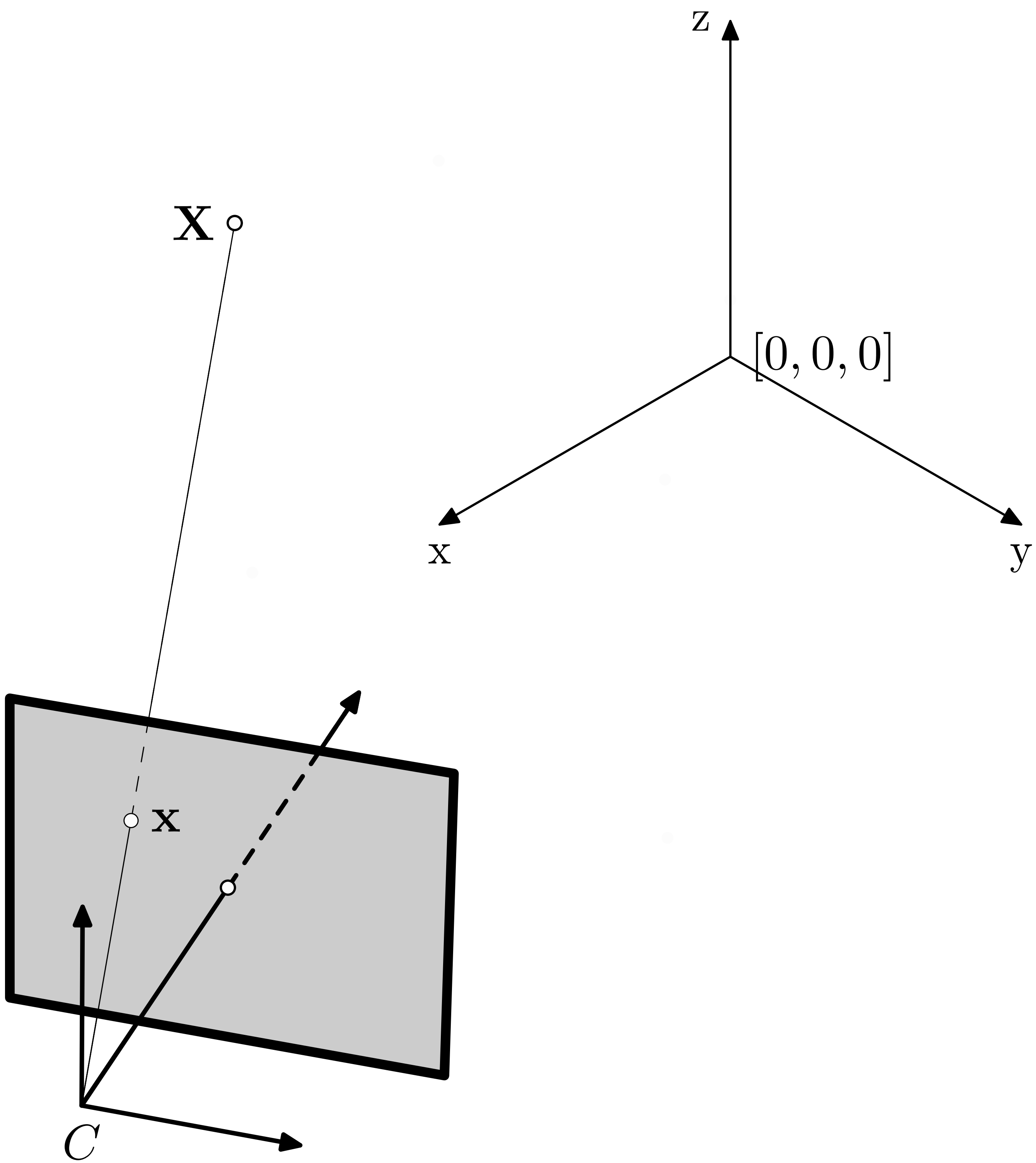
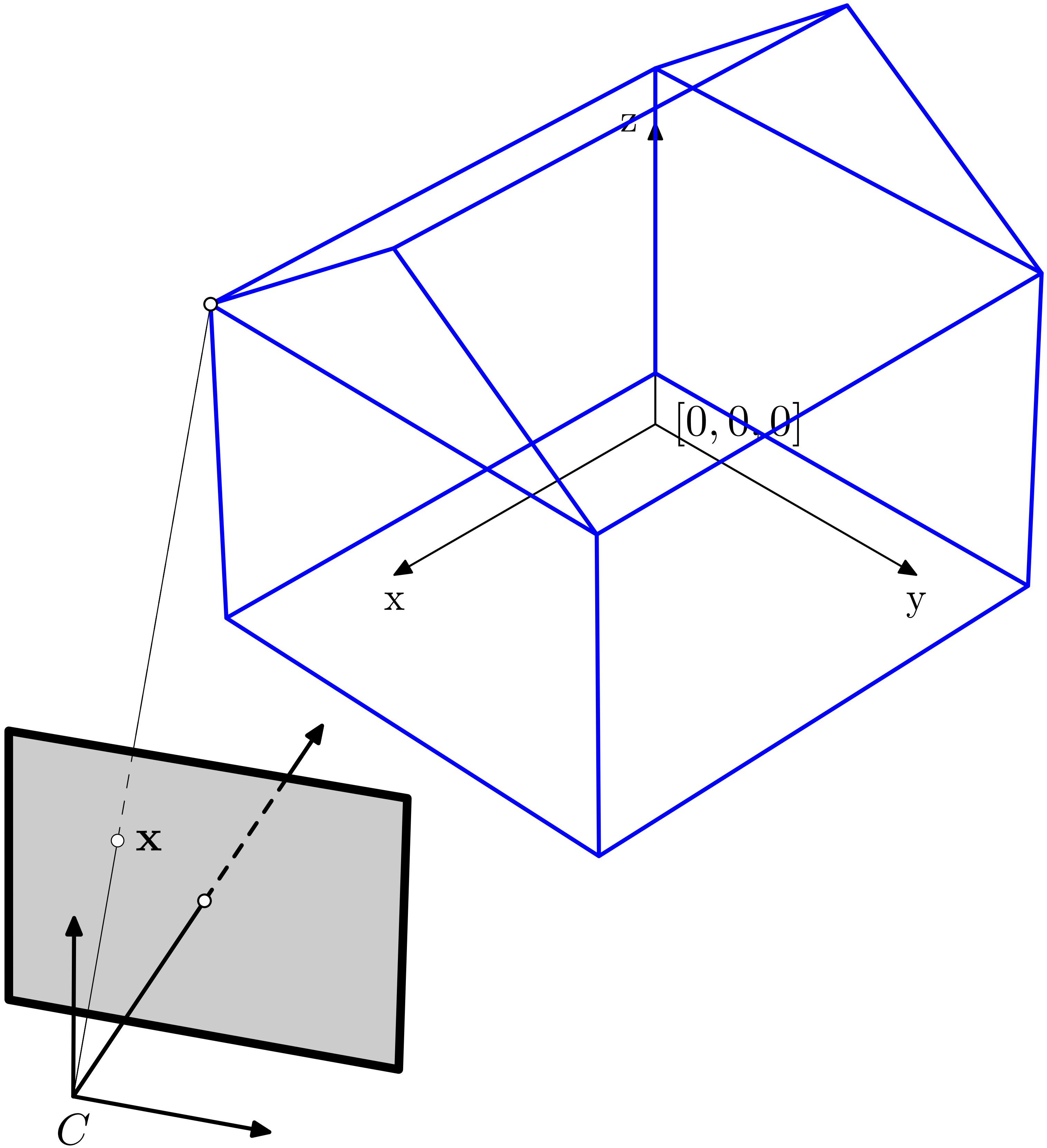


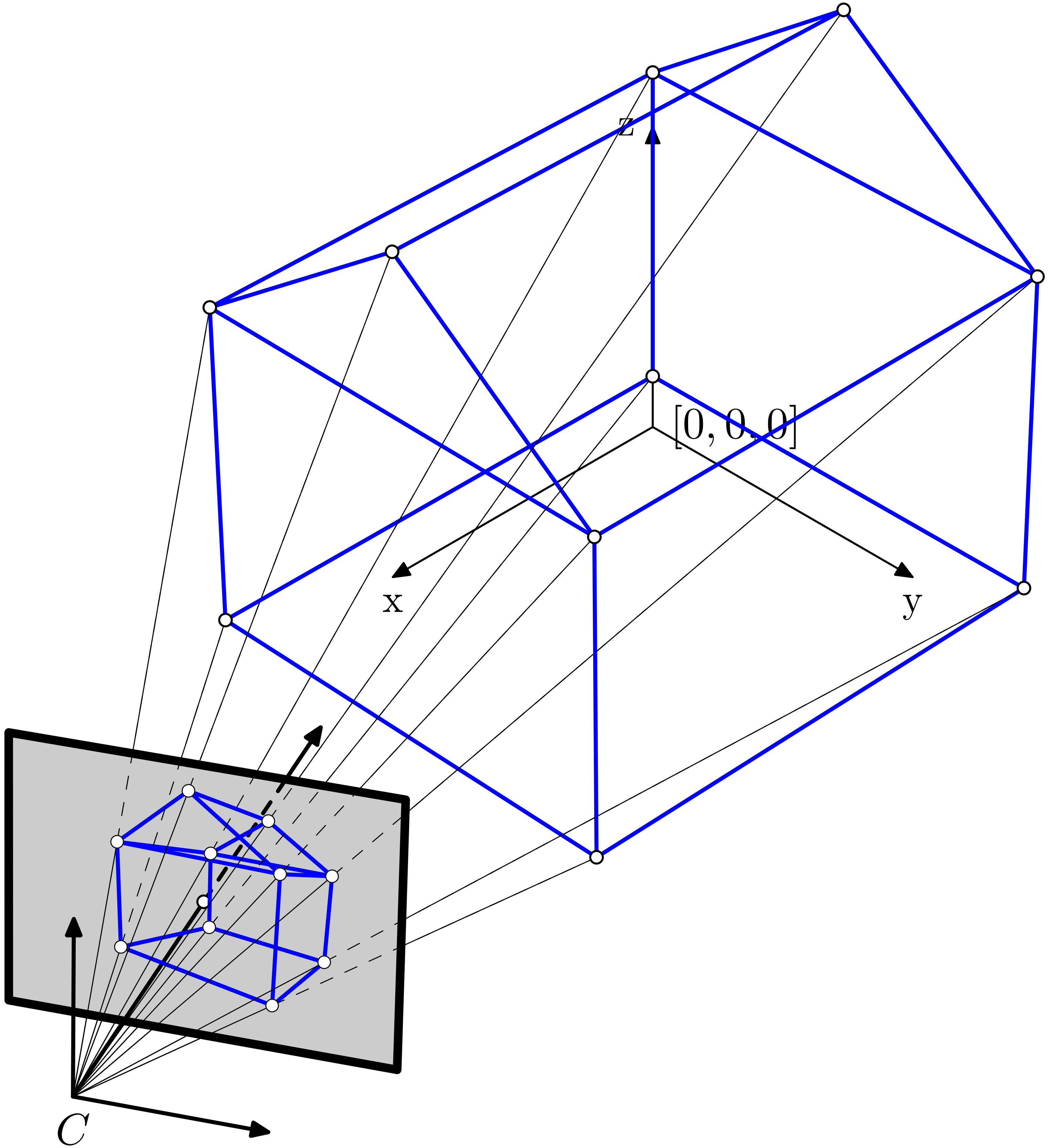
image plane
image plane
x
$-f$
x_1
x_2
x_3
X_1
X_2
X_3
C
f
z
optical axis
x_2
x_3
x_1
Z_1
Z_2
Z_3

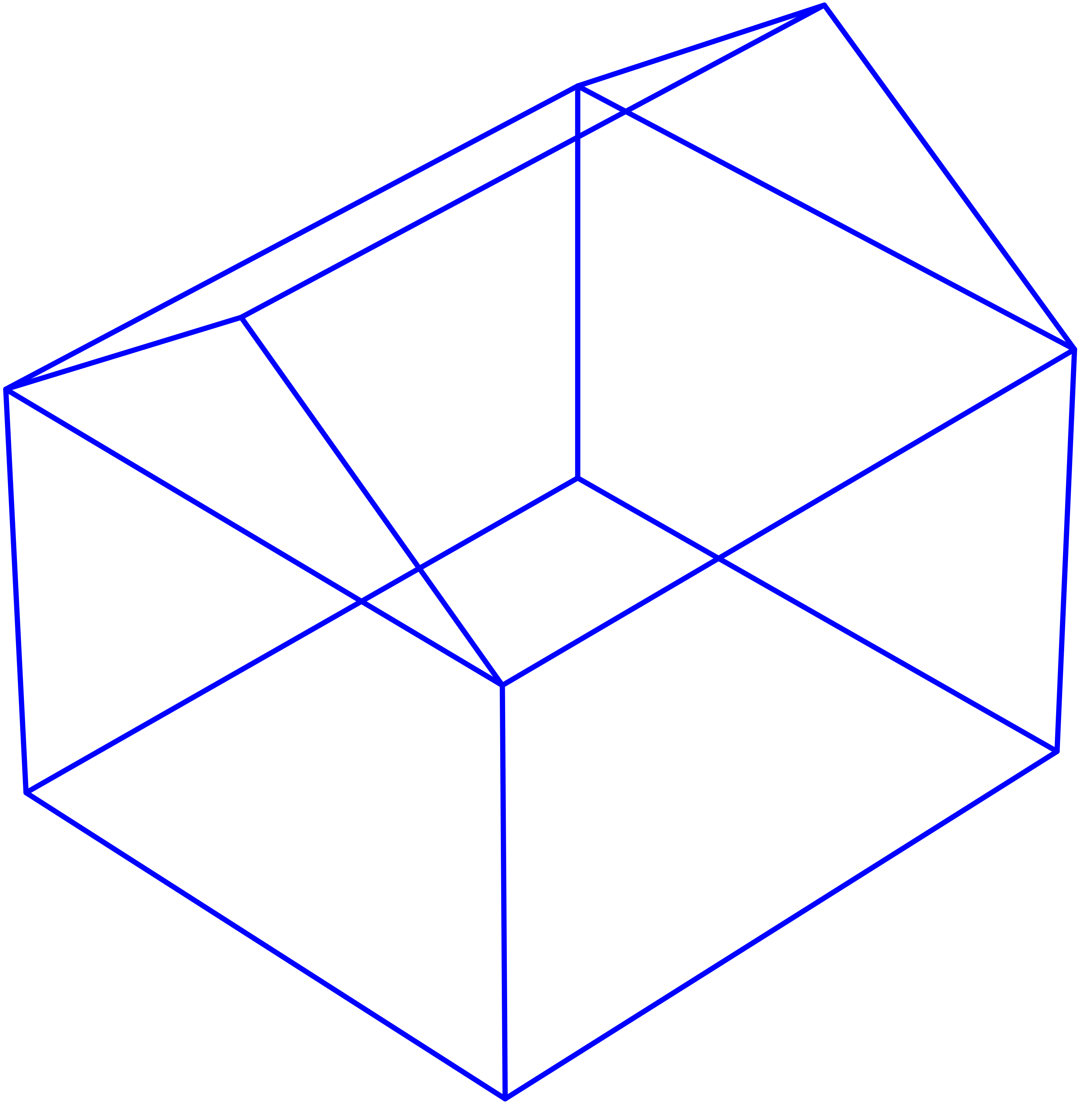

z
X
[0, 0, 0]
x
y

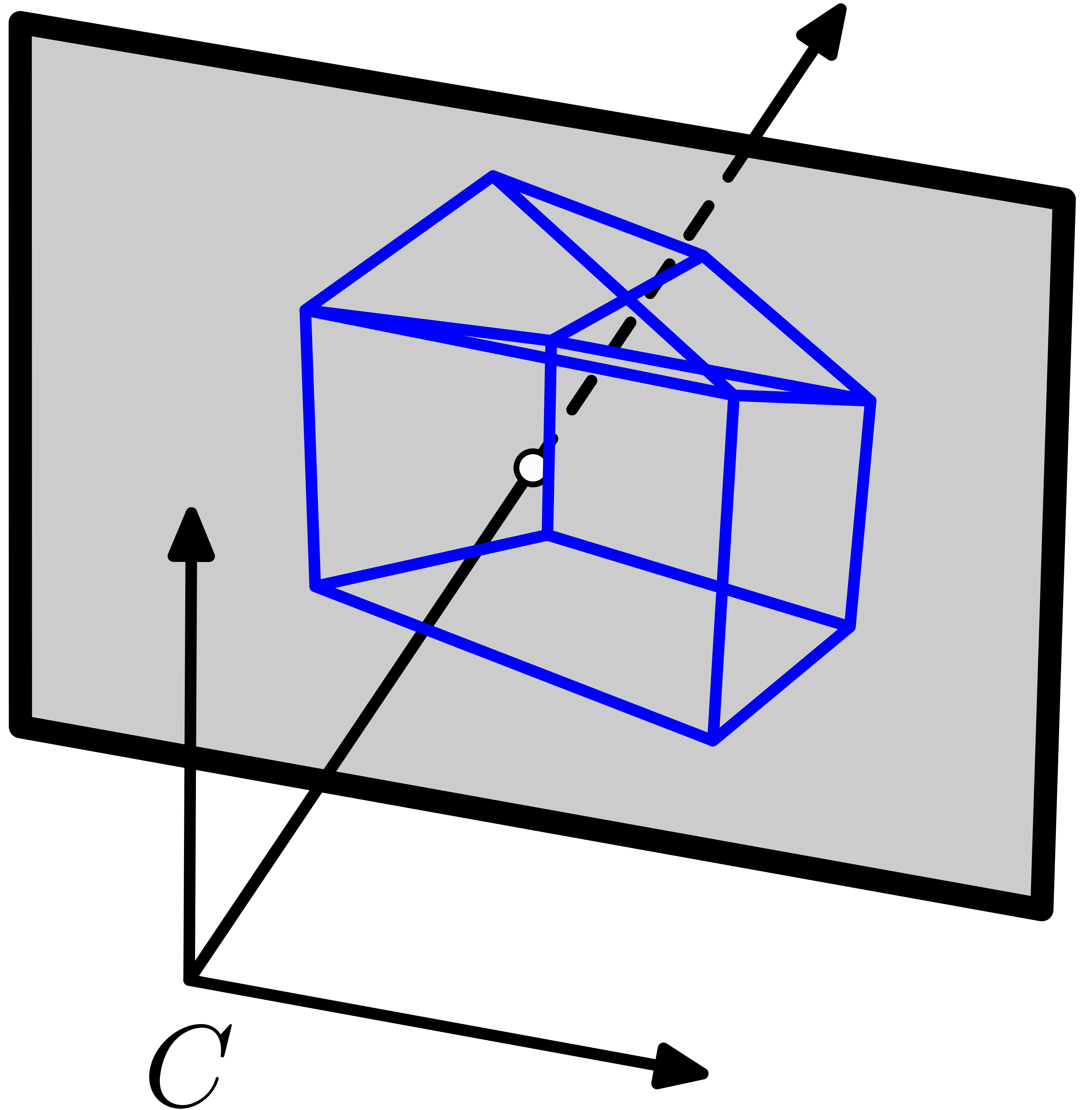

z
X
[0, 0, 0]
x
y
C

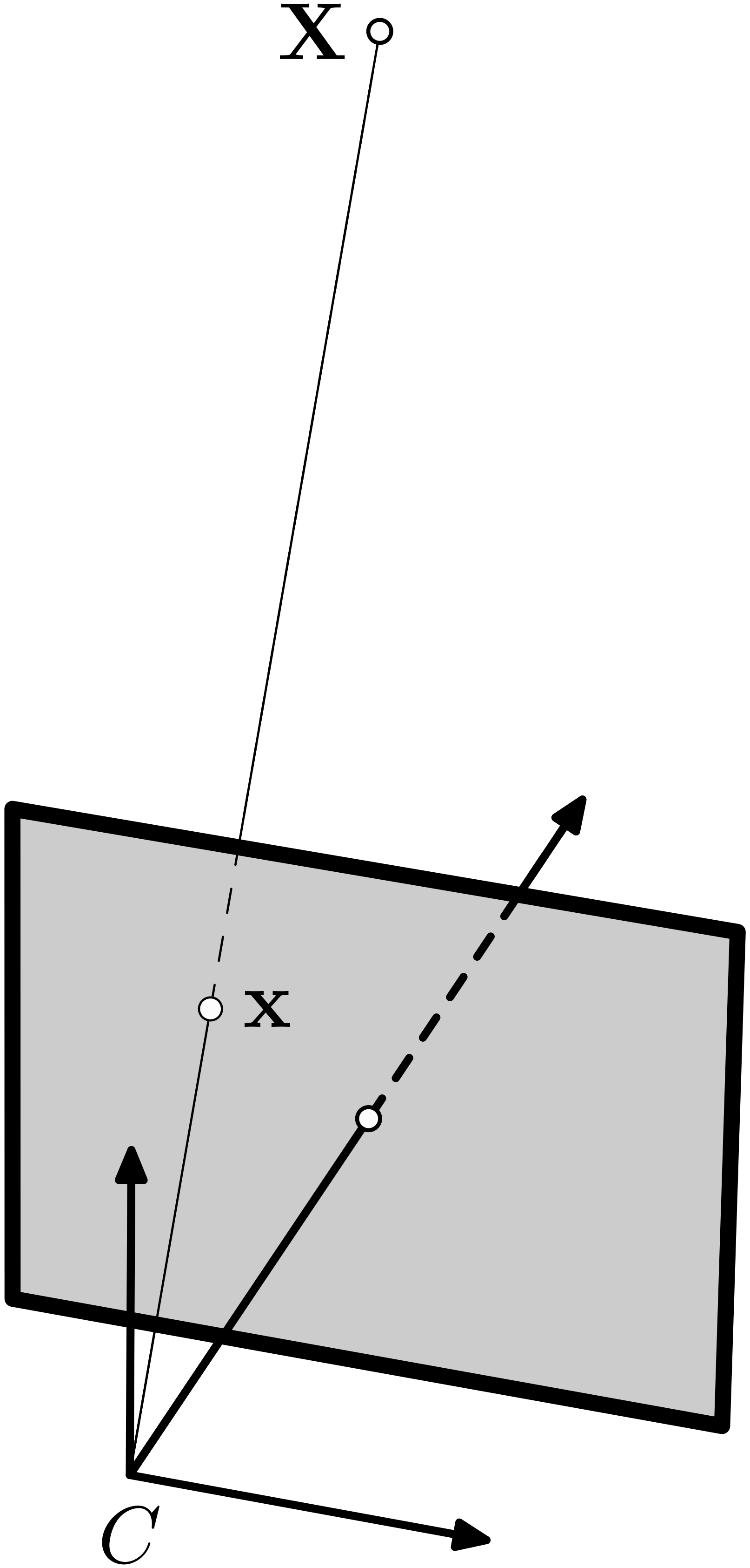

z
X
[0, 0, 0]
x
y
x
C


z
[0, 0, 0]
x
y
x
C

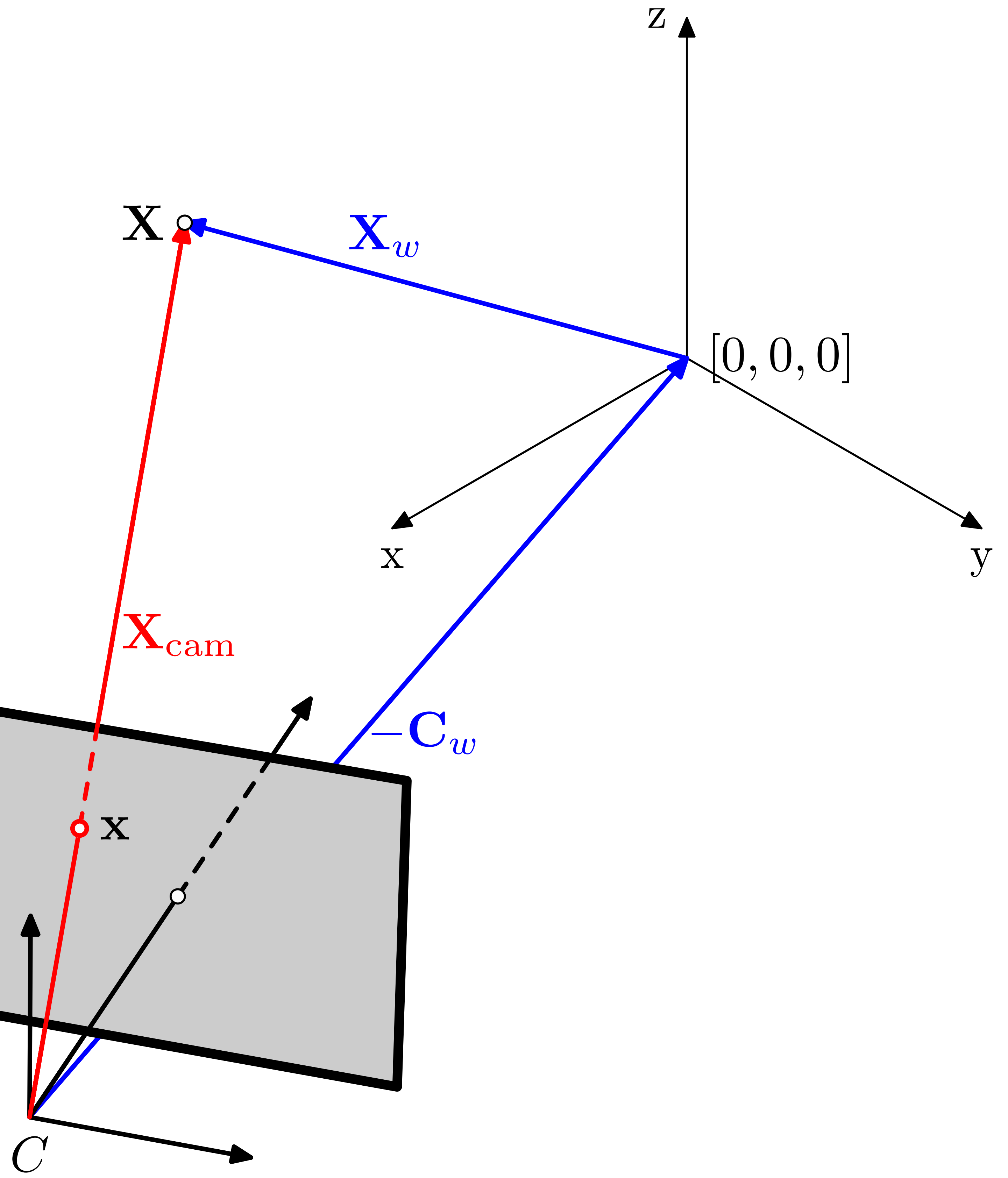

z
[0, 0, 0]
x
y
C



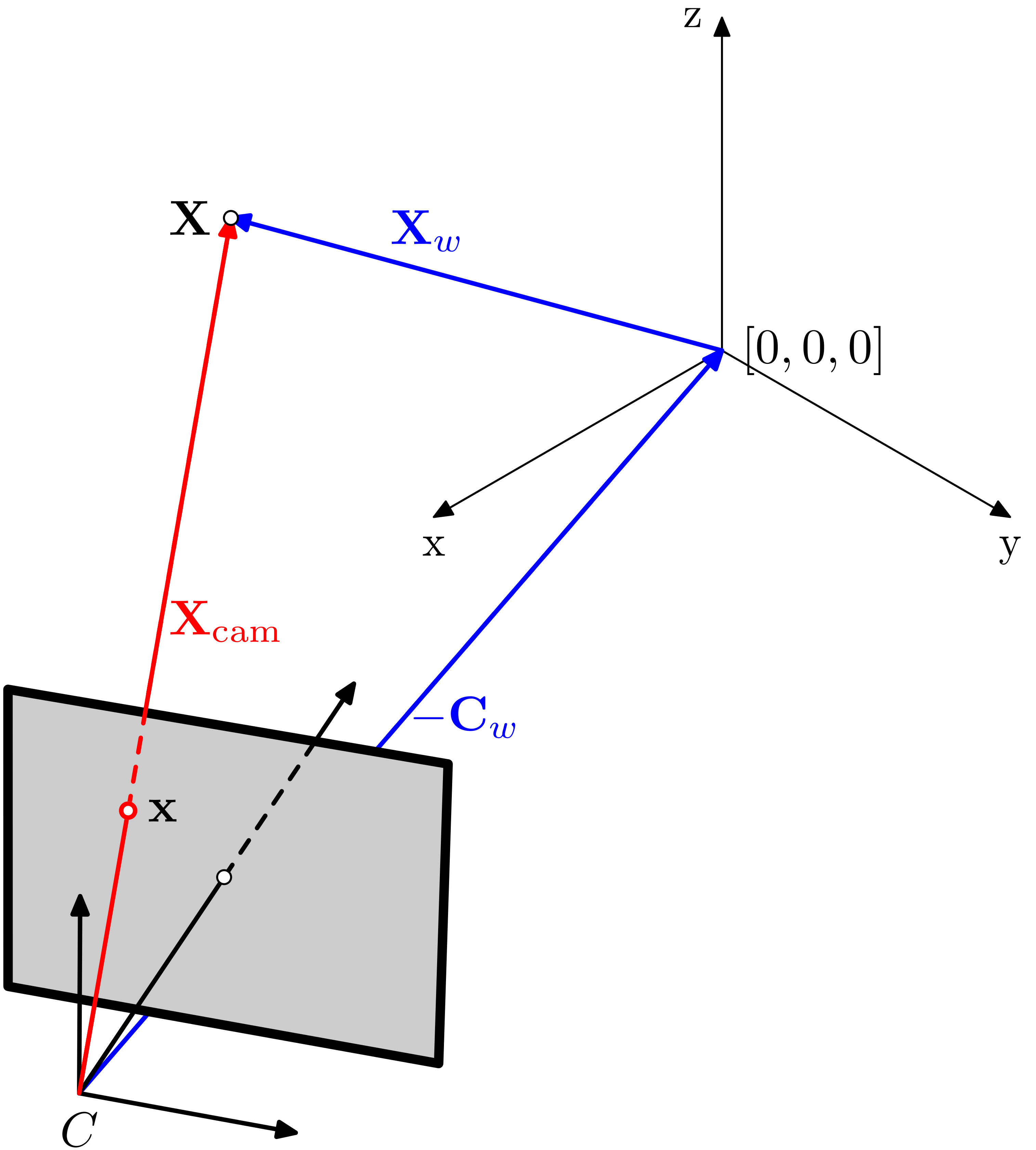

C



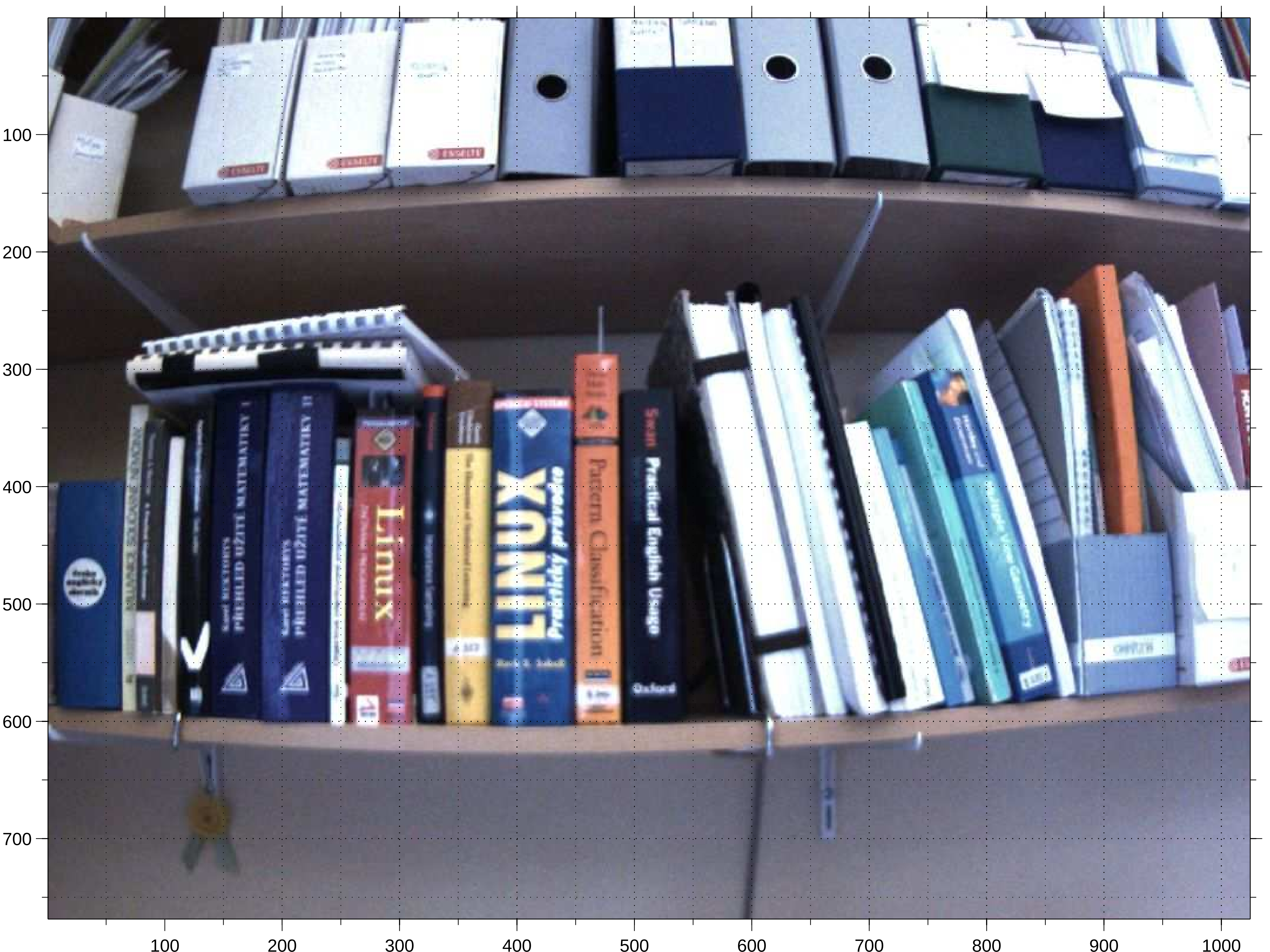
X
x
C

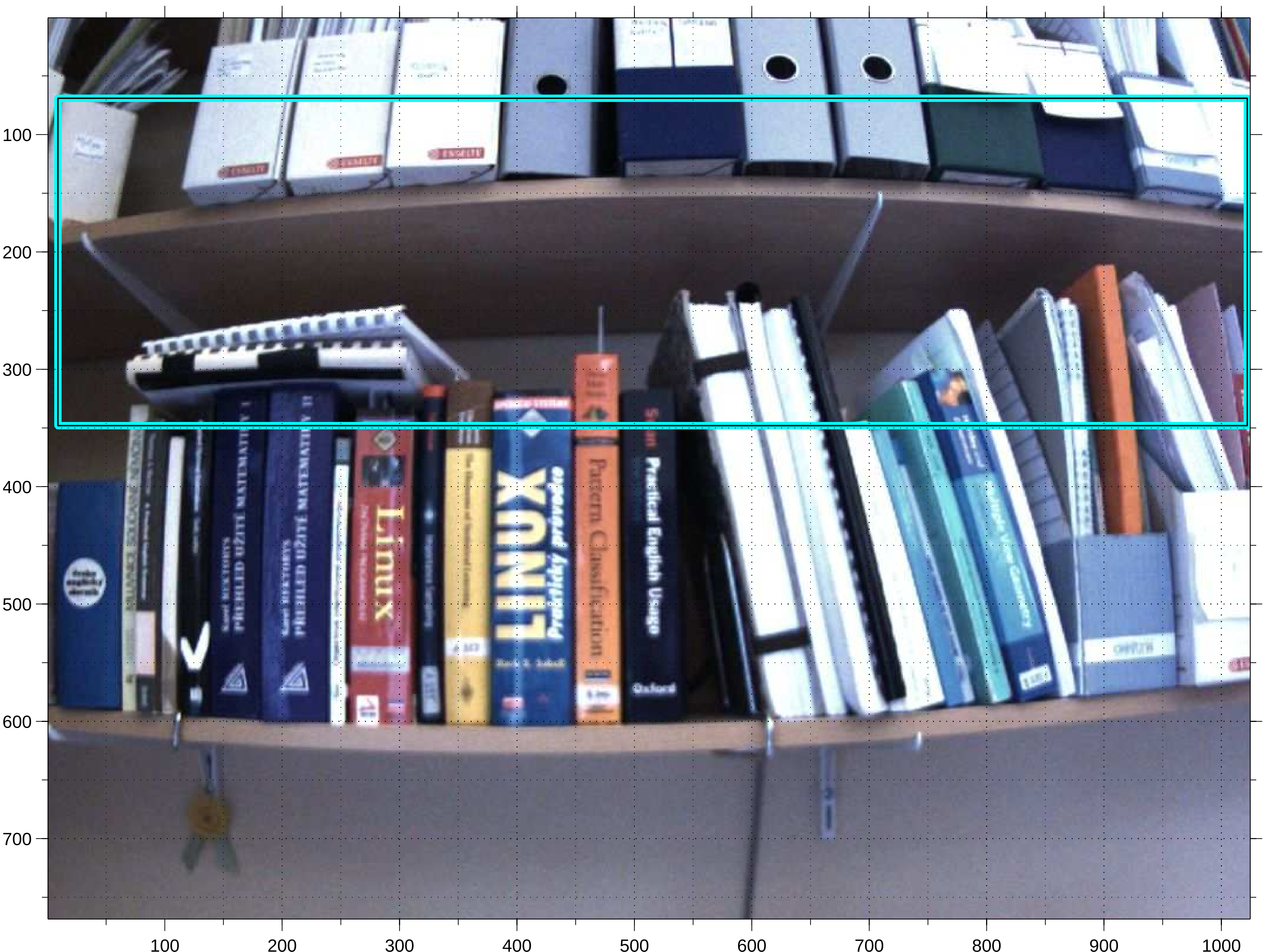


z
X
X_w
[0, 0, 0]
x
y
X_cam
−C_w
x
C

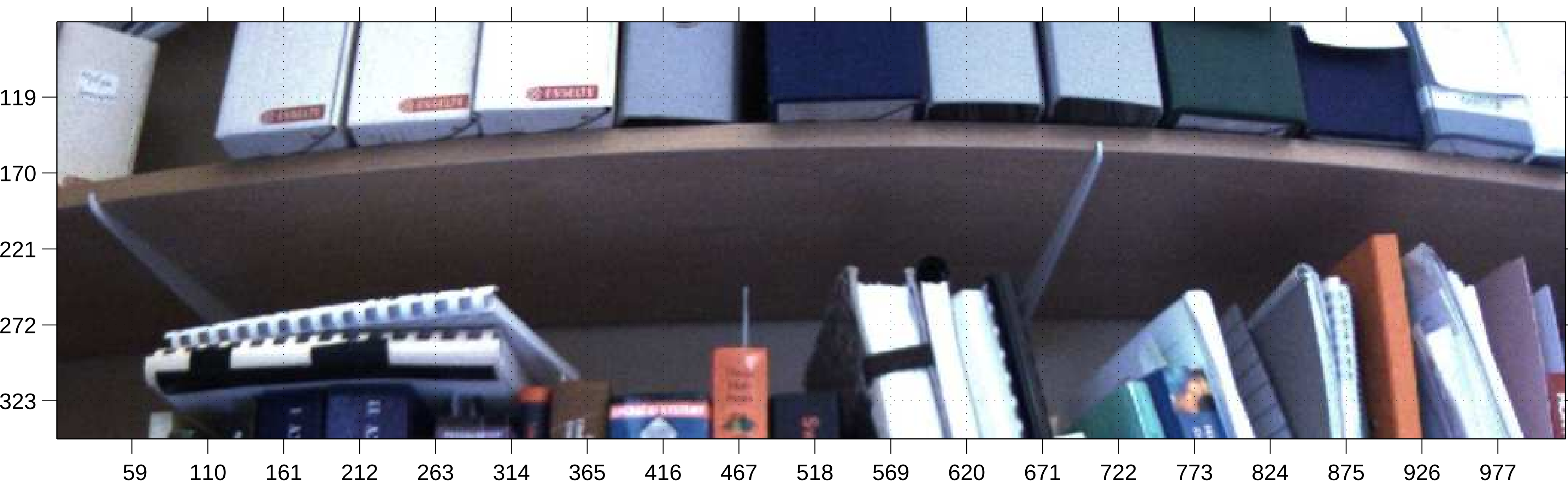


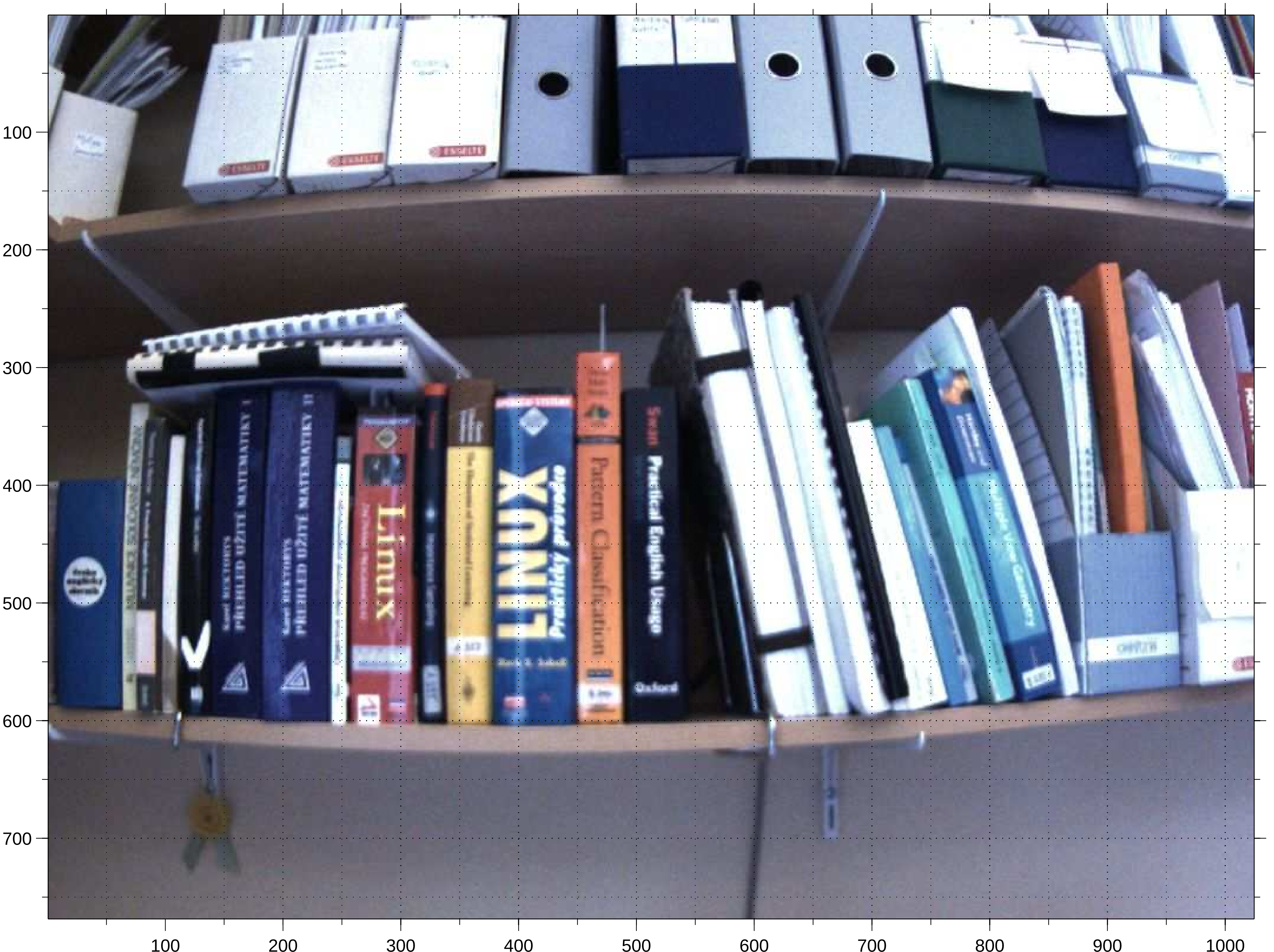
z
X
$\mathbf{X}_w$
[0, 0, 0]
x
y
$\mathbf{X}_{\mathrm{cam}}$
$-\mathbf{C}_w$
x
C

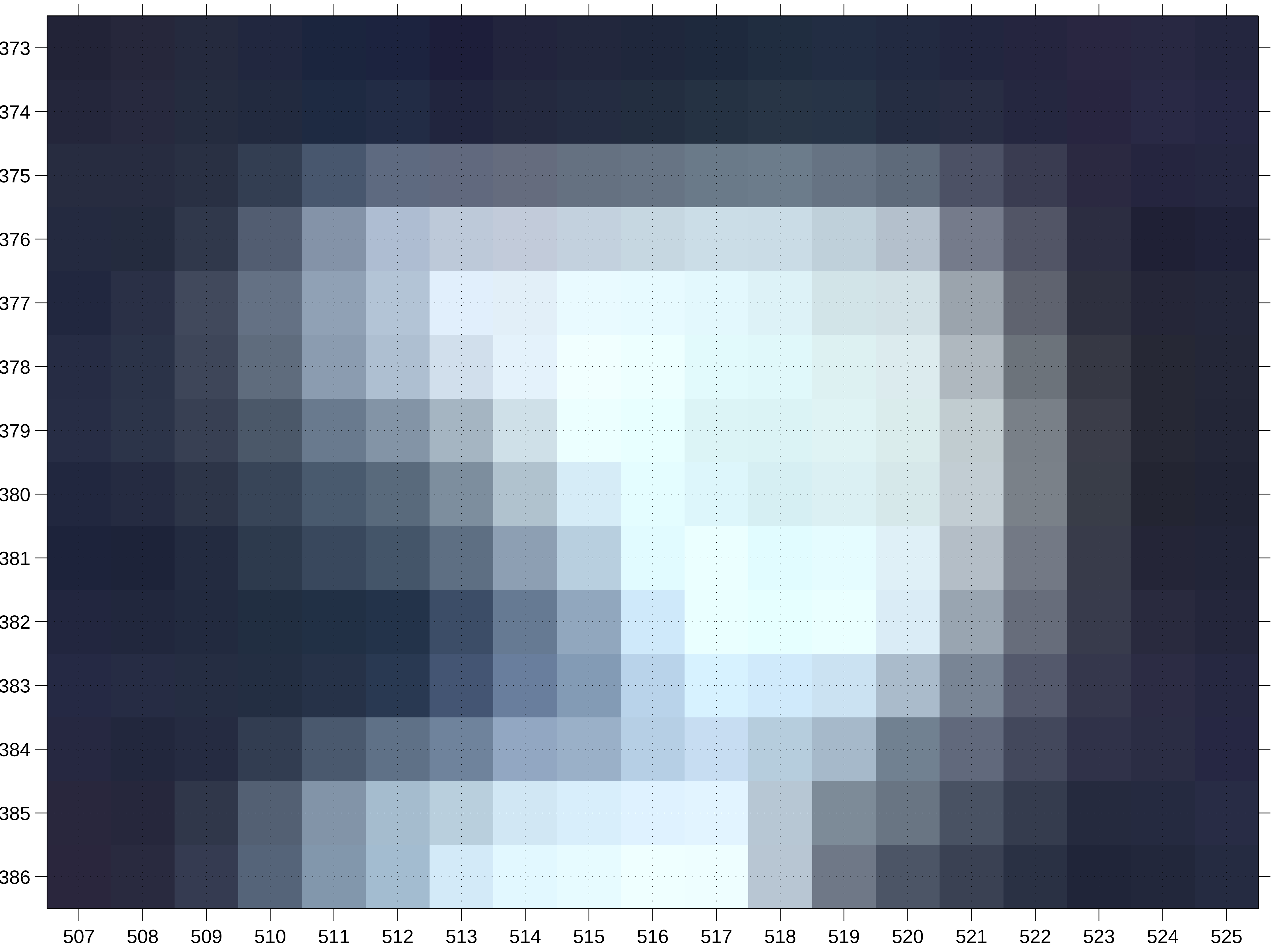

100
200
300
400
500
600
700
100
200
300
400
500
600
700
800
900
1000
PŘEHLED UŽITÉ MATEMATIKY I
PŘEHLED UŽITÉ MATEMATIKY II
Linux
LINUX
Pattern Classification
Practical English Usage
Oxford



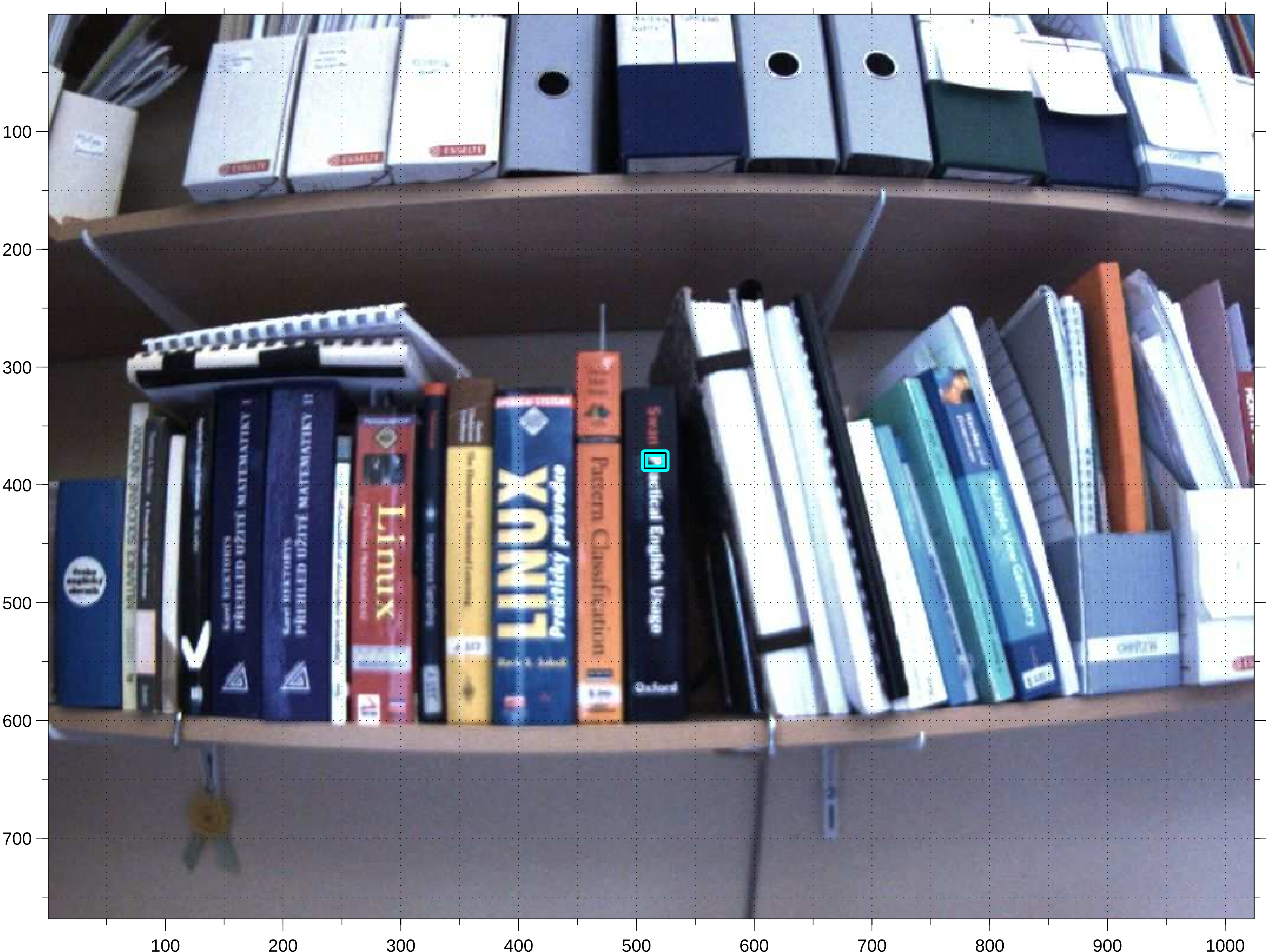
100
200
300
400
500
600
700
800
900
1000
PŘEHLED UŽITÉ MATEMATIKY I
PŘEHLED UŽITÉ MATEMATIKY II
Linux
LINUX
Pattern Classification
Practical English Usage
Oxford

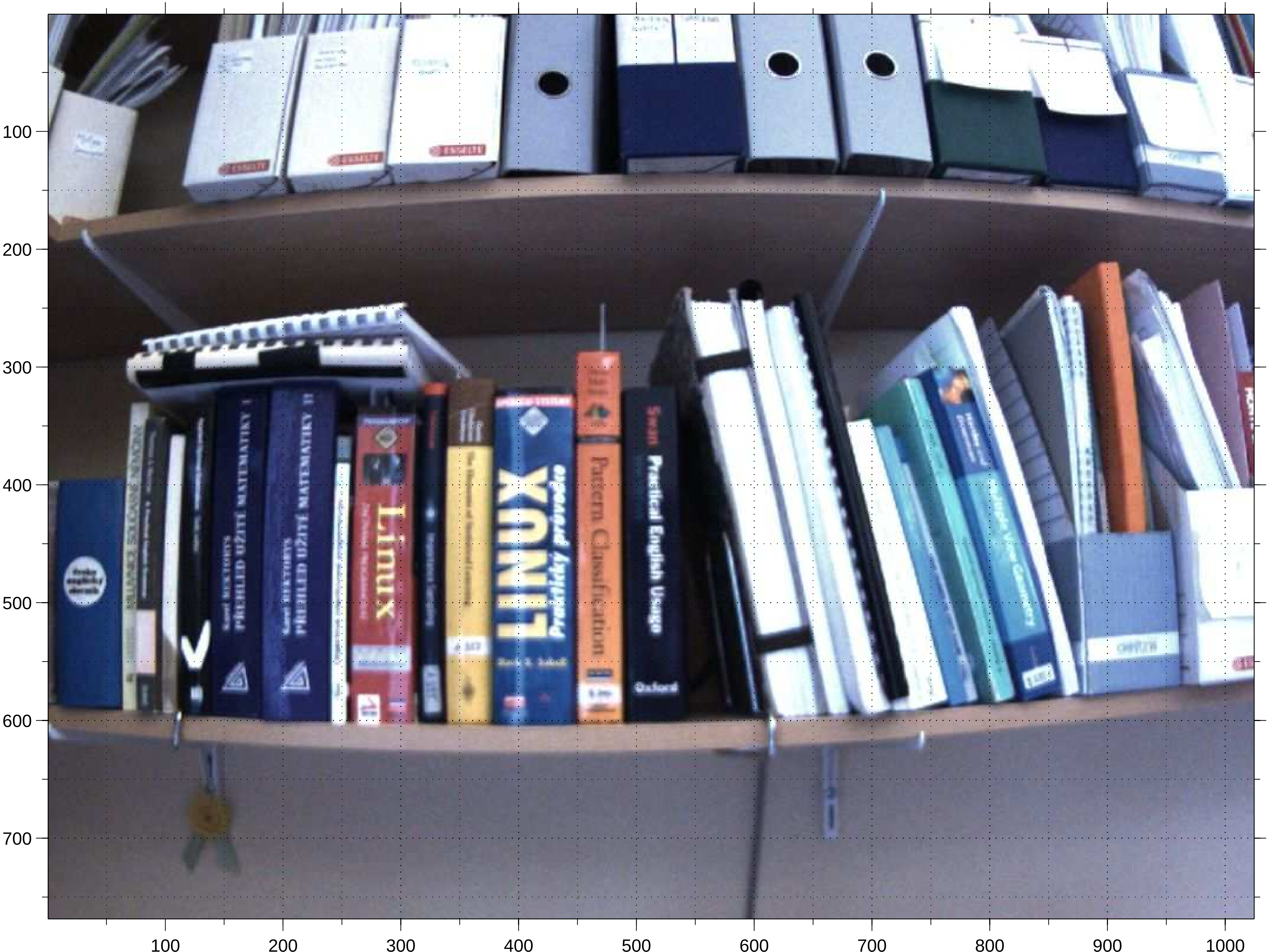

119
170
221
272
323
59
110
161
212
263
314
365
416
467
518
569
620
671
722
773
824
875
926
977

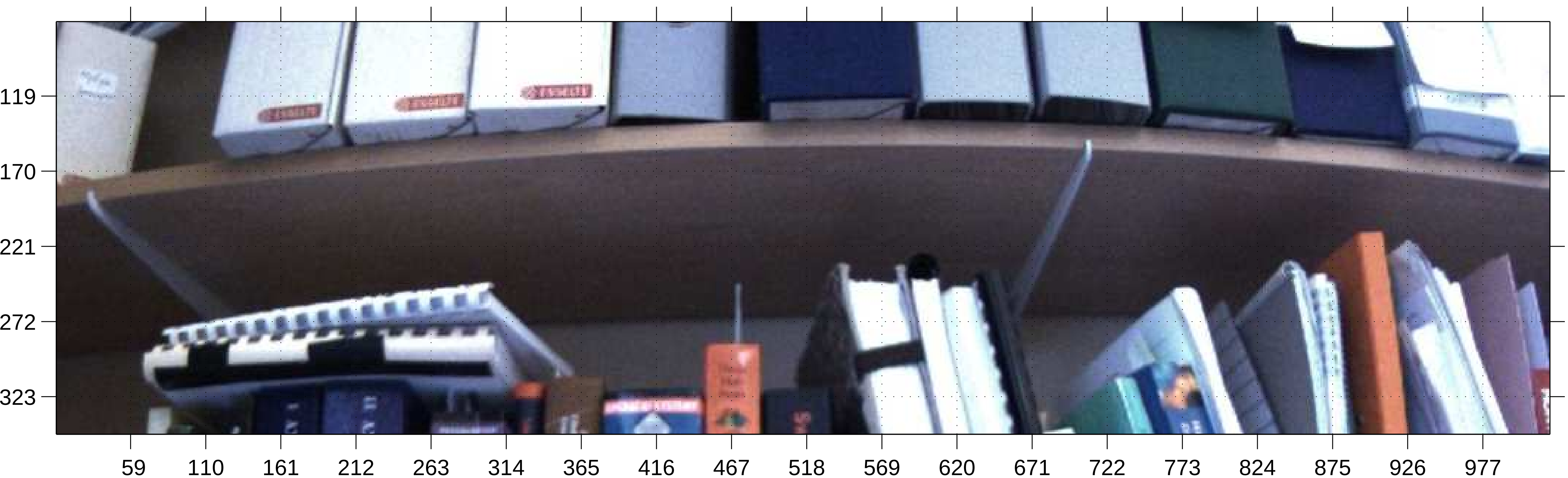

100
200
300
400
500
600
700
100
200
300
400
500
600
700
800
900
1000
Karel Rektorys
PŘEHLED UŽITÉ MATEMATIKY I
Karel Rektorys
PŘEHLED UŽITÉ MATEMATIKY II
Linux
LINUX
Prakticky průvodce
Pattern Classification
Swan
Practical English Usage
Oxford

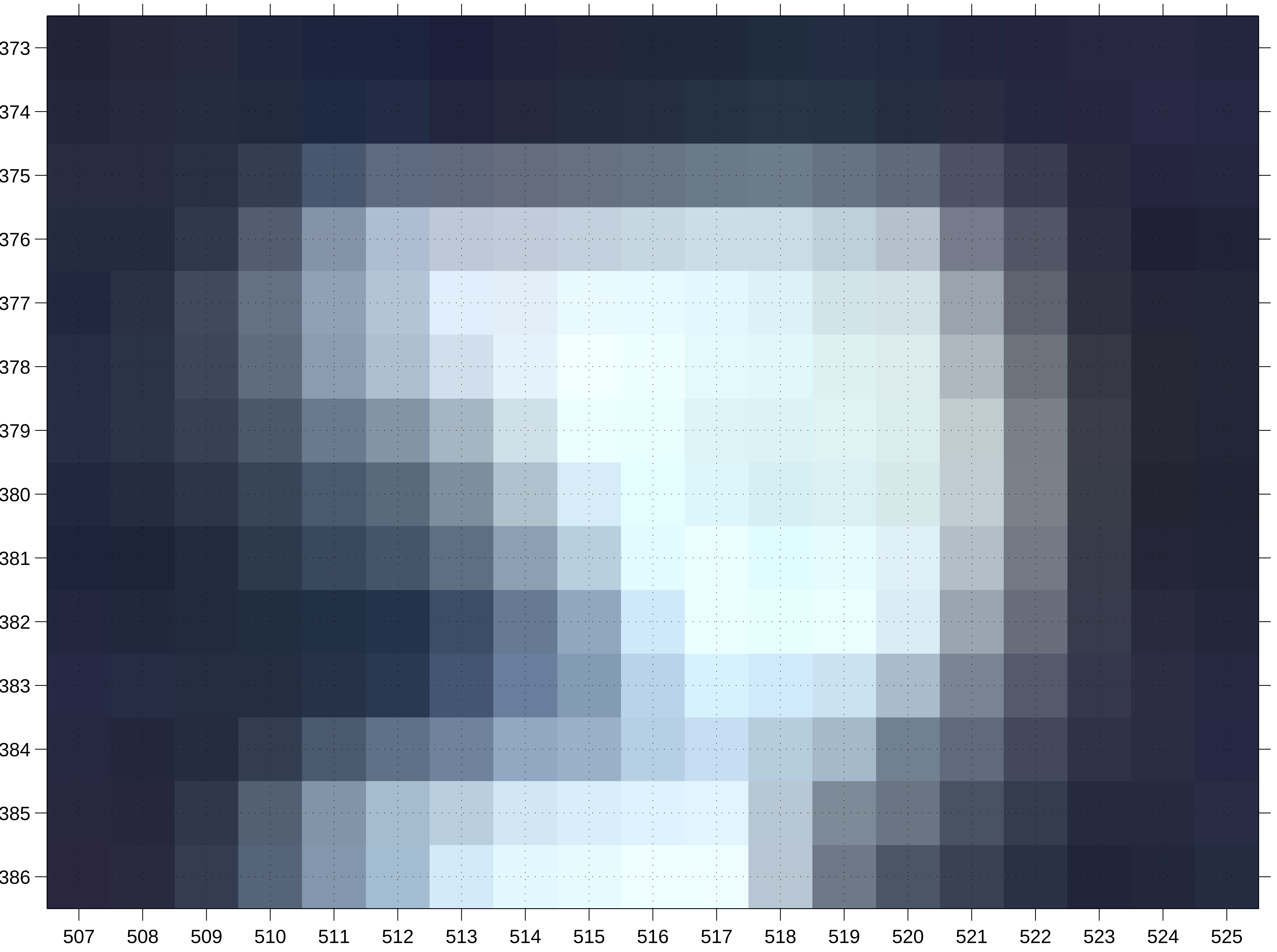


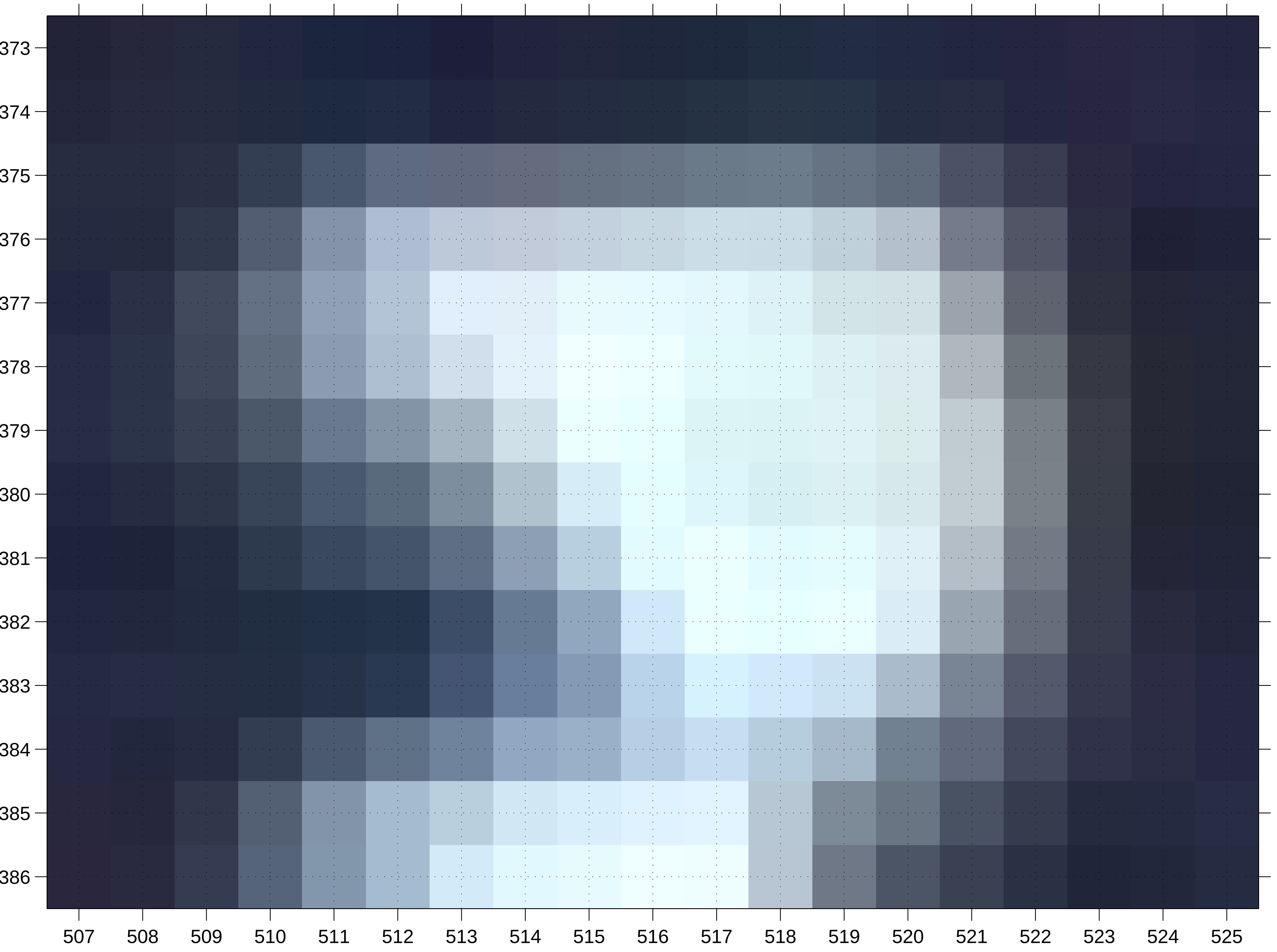
373
374
375
376
377
378
379
380
381
382
383
384
385
386
507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525


100
200
300
400
500
600
700
100
200
300
400
500
600
700
800
900
1000
Linux
LINUX
Pattern Classification
PŘEHLED UŽITÉ MATEMATIKY
Oxford

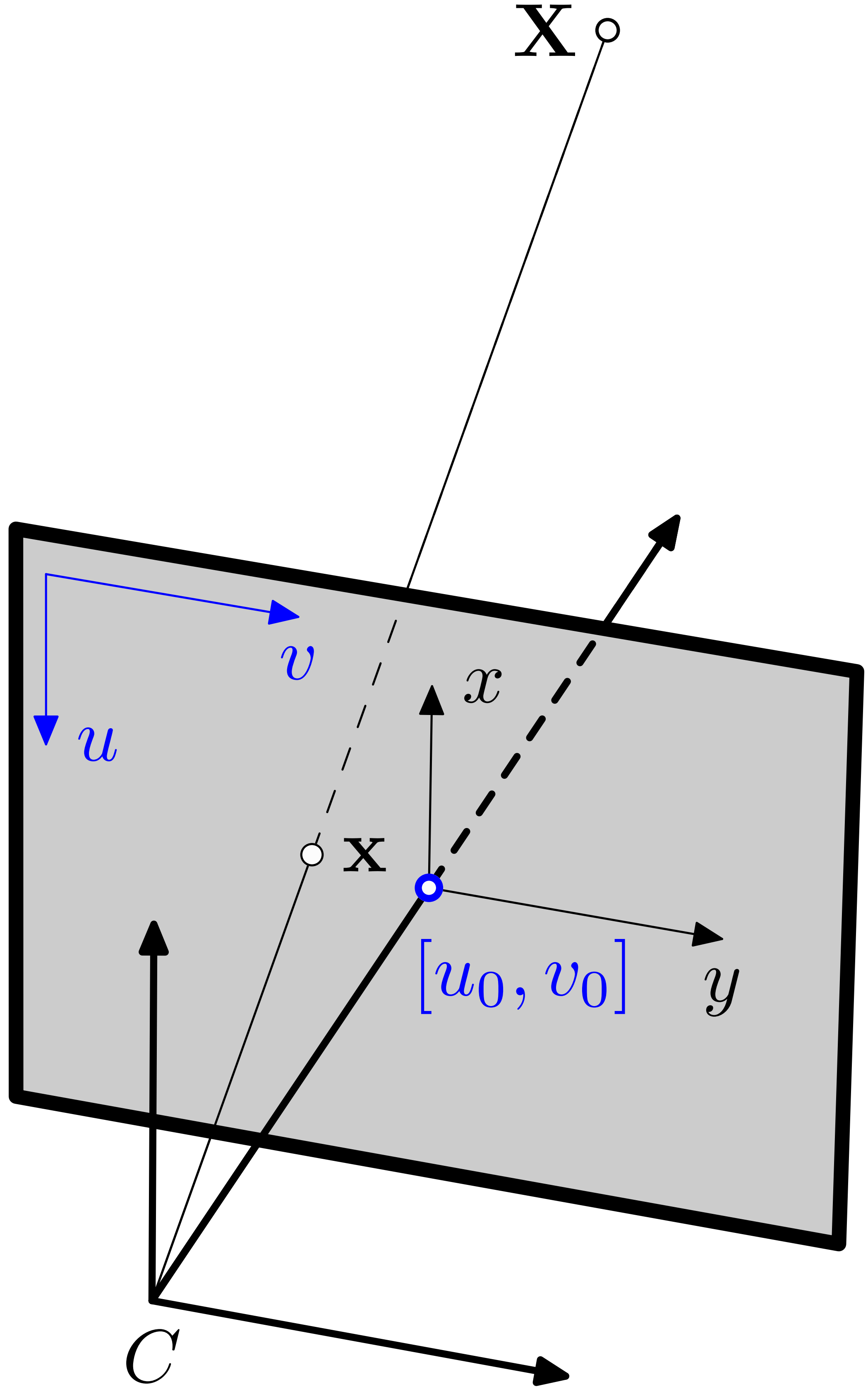

100
200
300
400
500
600
700
100
200
300
400
500
600
700
800
900
1000
PŘEHLED UŽITÉ MATEMATIKY I
PŘEHLED UŽITÉ MATEMATIKY II
Linux
LINUX
Pattern Classification
Swan
Practical English Usage
Oxford


119
170
221
272
323
59 110 161 212 263 314 365 416 467 518 569 620 671 722 773 824 875 926 977

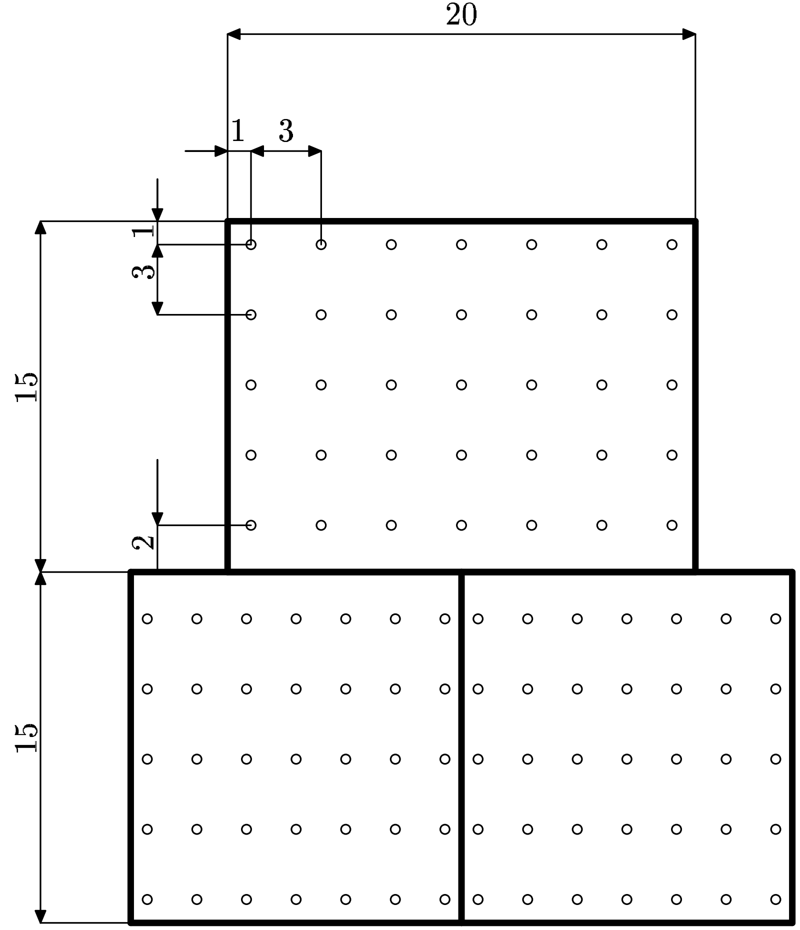

373
374
375
376
377
378
379
380
381
382
383
384
385
386
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525

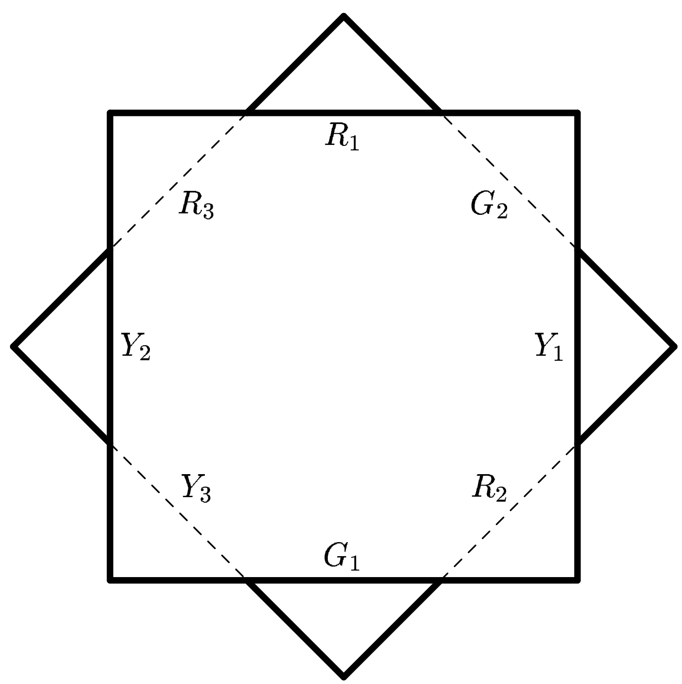


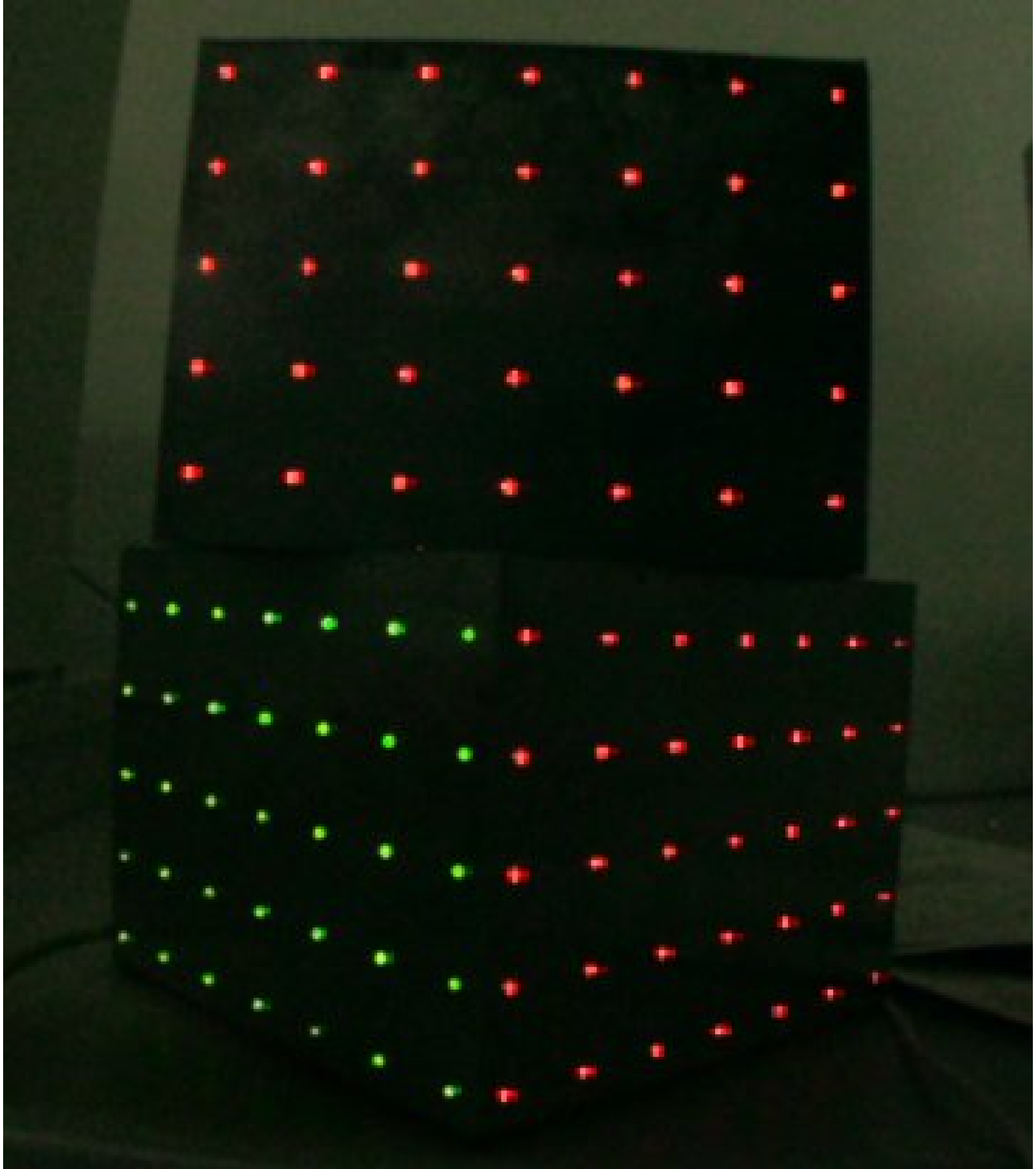
373
374
375
376
377
378
379
380
381
382
383
384
385
386
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525

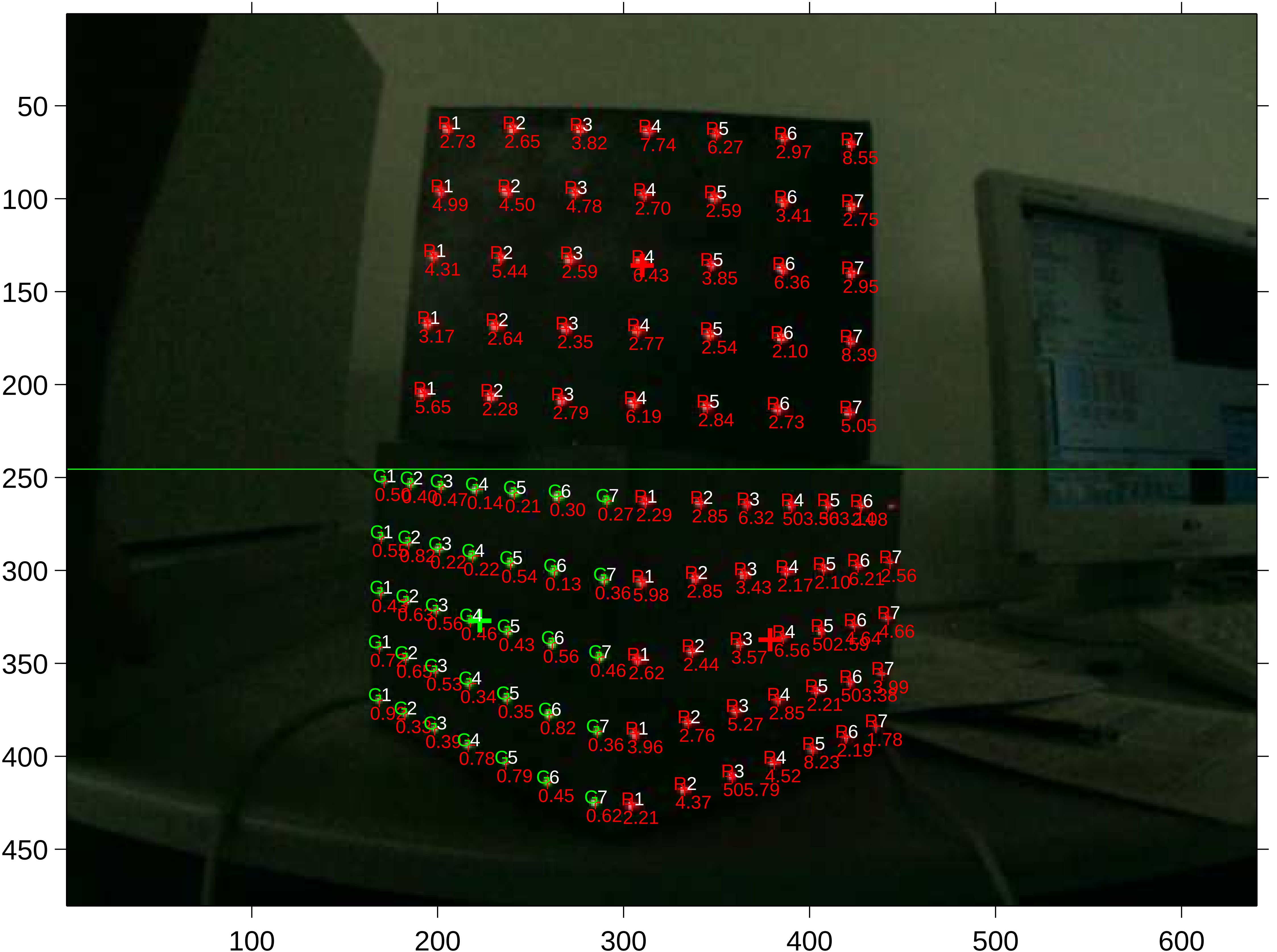


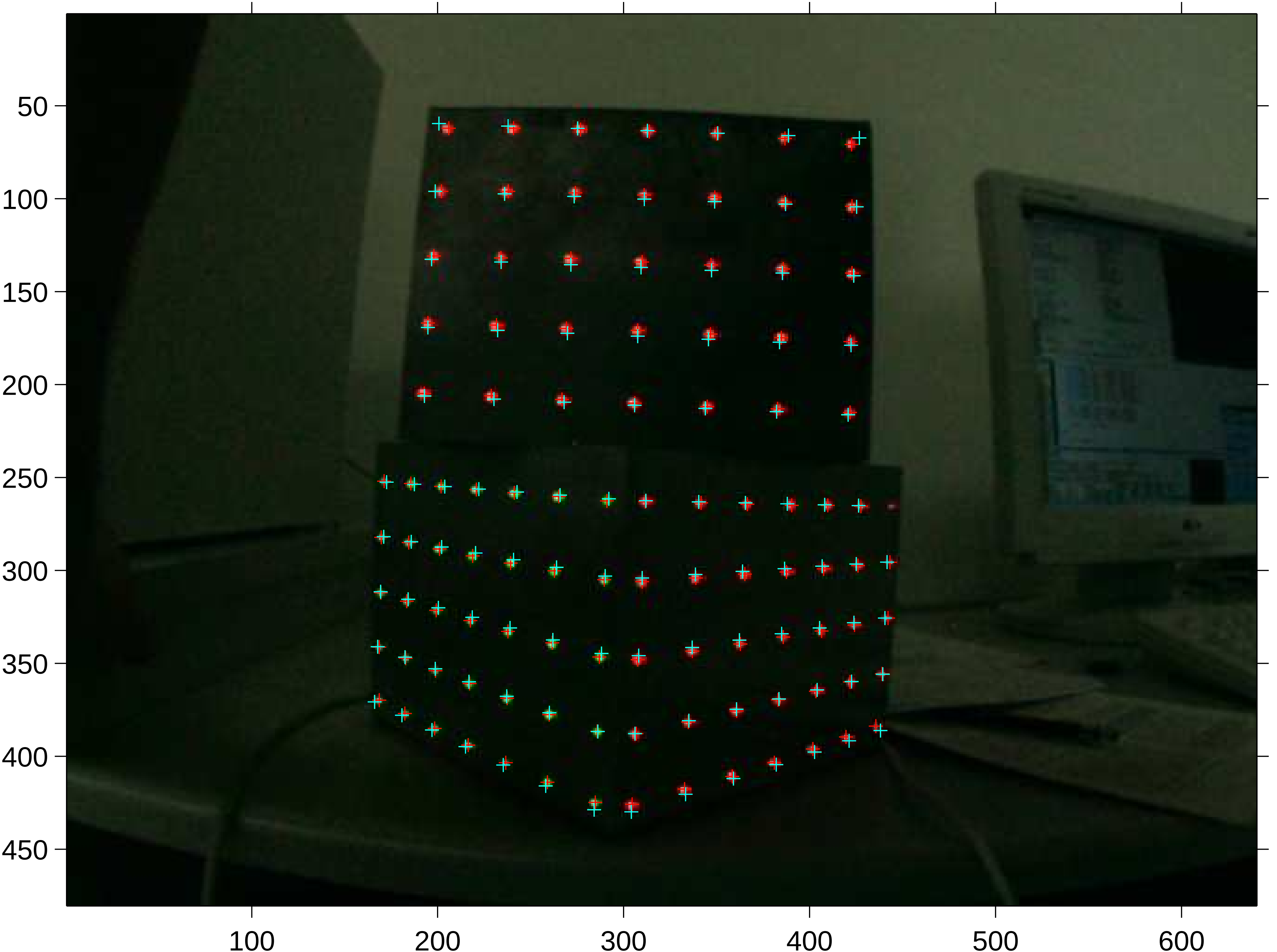
X
v
u
x
x
[u0, v0]
y
C

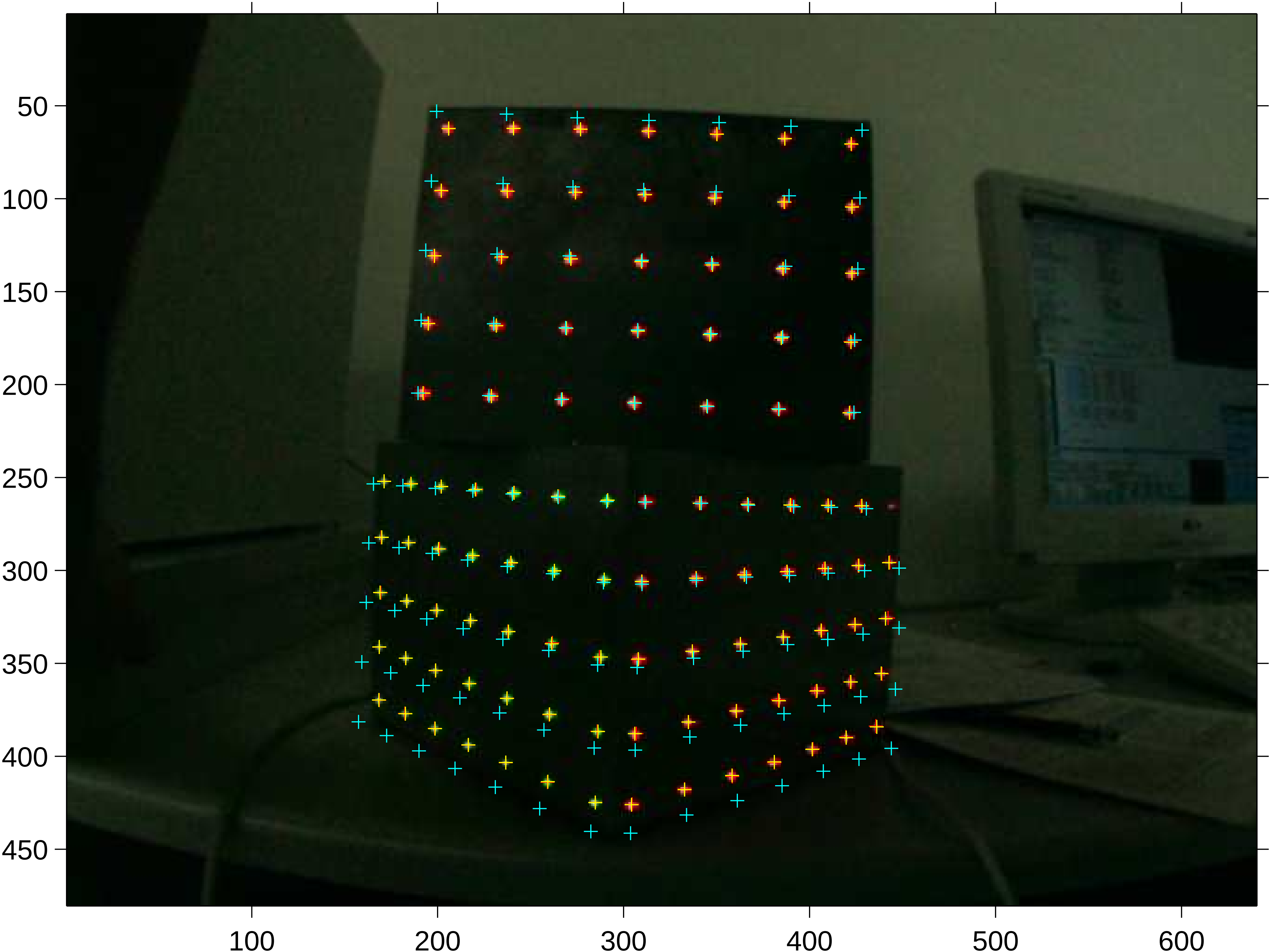

20
1
3
1
3
15
2
15

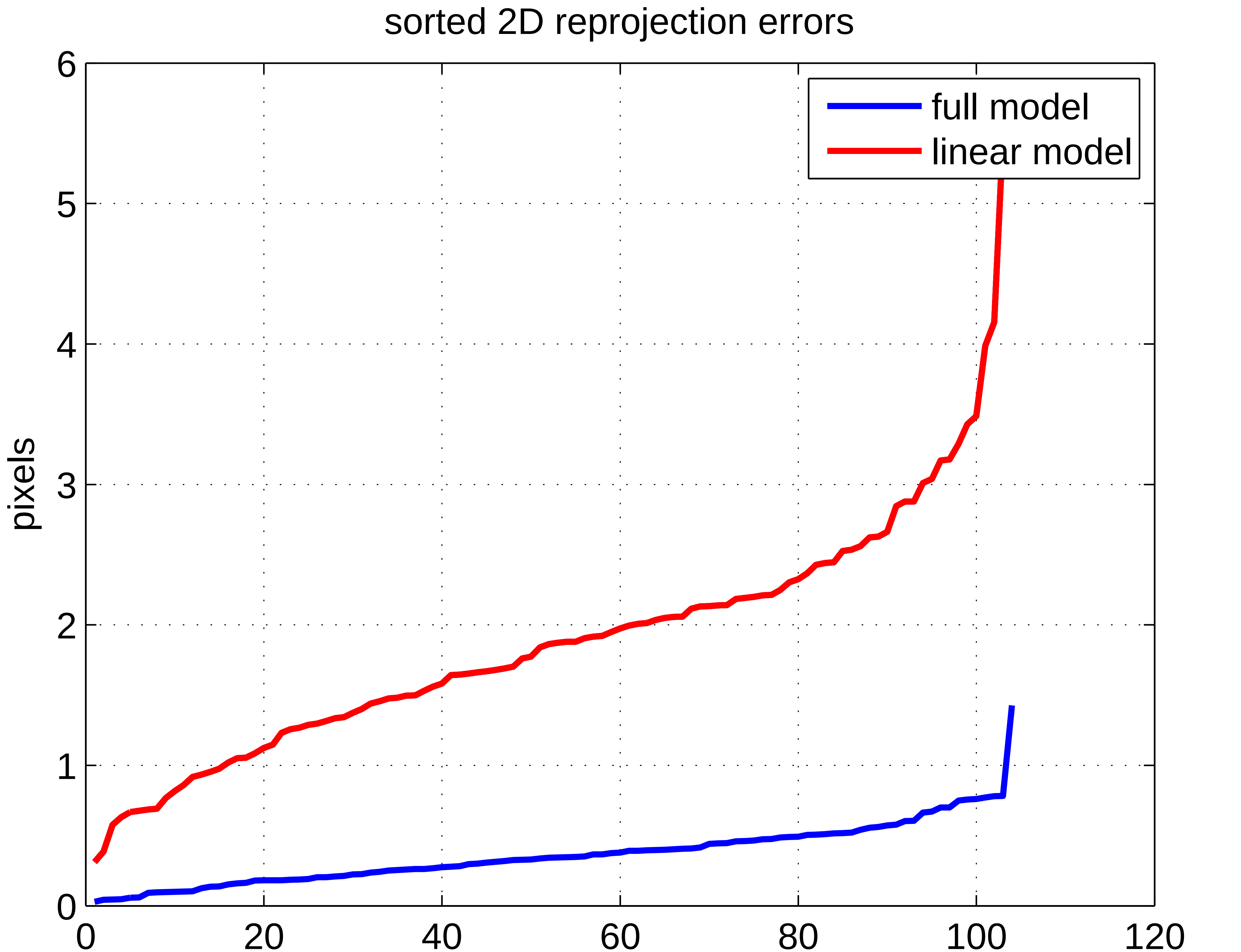

R_1
R_3
G_2
Y_2
Y_1
Y_3
R_2
G_1




50
100
150
200
250
300
350
400
450
100
200
300
400
500
600
R1 2.73
R2 2.65
R3 3.82
R4 7.74
R5 6.27
R6 2.97
R7 8.55
R1 4.99
R2 4.50
R3 4.78
R4 2.70
R5 2.59
R6 3.41
R7 2.75
R1 4.31
R2 5.44
R3 2.59
R4 6.43
R5 3.85
R6 6.36
R7 2.95
R1 3.17
R2 2.64
R3 2.35
R4 2.77
R5 2.54
R6 2.10
R7 8.39
R1 5.65
R2 2.28
R3 2.79
R4 6.19
R5 2.84
R6 2.73
R7 5.05


50
100
150
200
250
300
350
400
450
100
200
300
400
500
600


50
100
150
200
250
300
350
400
450
100
200
300
400
500
600



sorted 2D reprojection errors
pixels
full model
linear model
0
1
2
3
4
5
6
0
20
40
60
80
100
120