Robot, lidar, camera

Karel Zimmermann

http://cmp.felk.cvut.cz/~zimmerk/

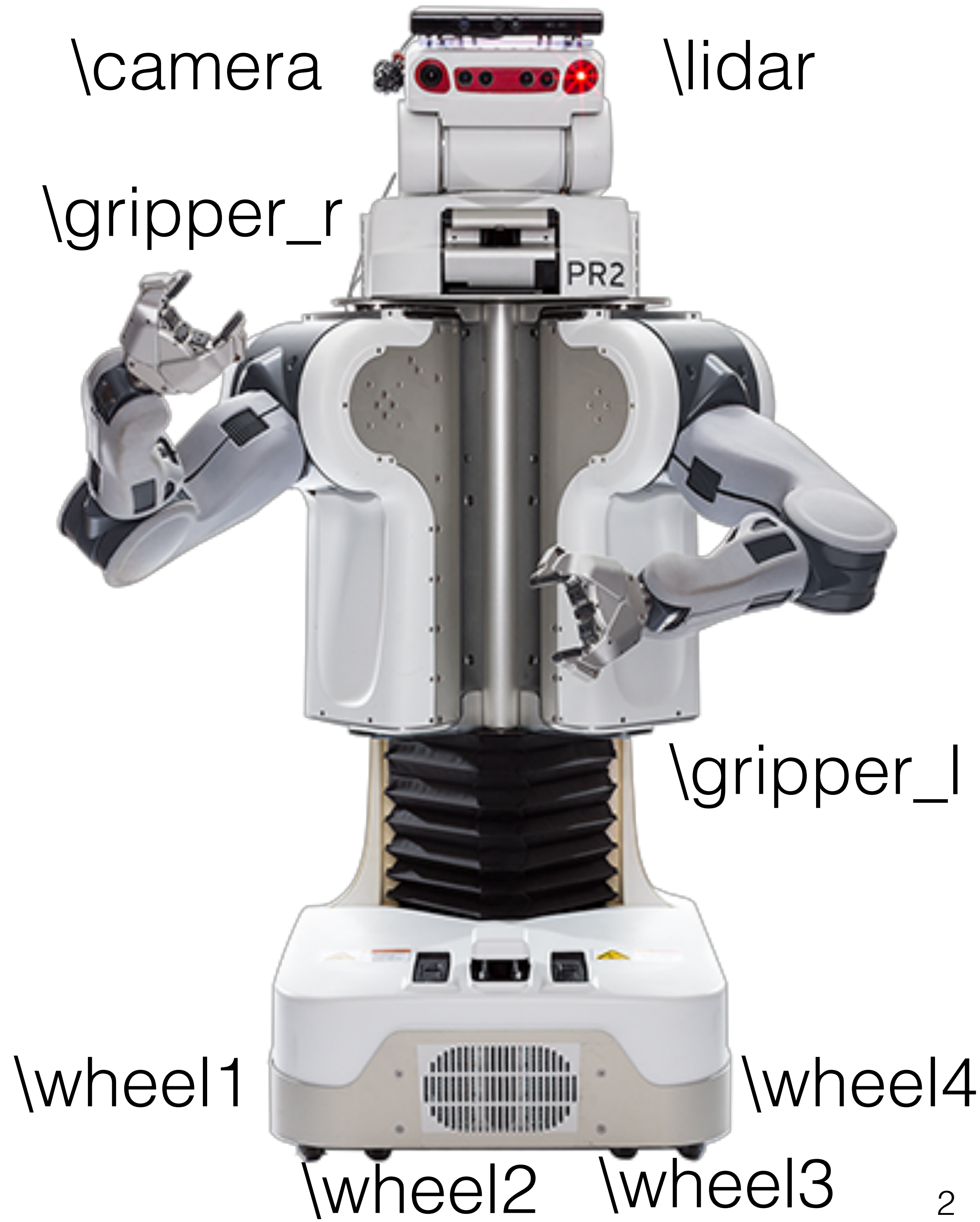
Vision for Robotics and Autonomous Systems
https://cyber.felk.cvut.cz/vras/

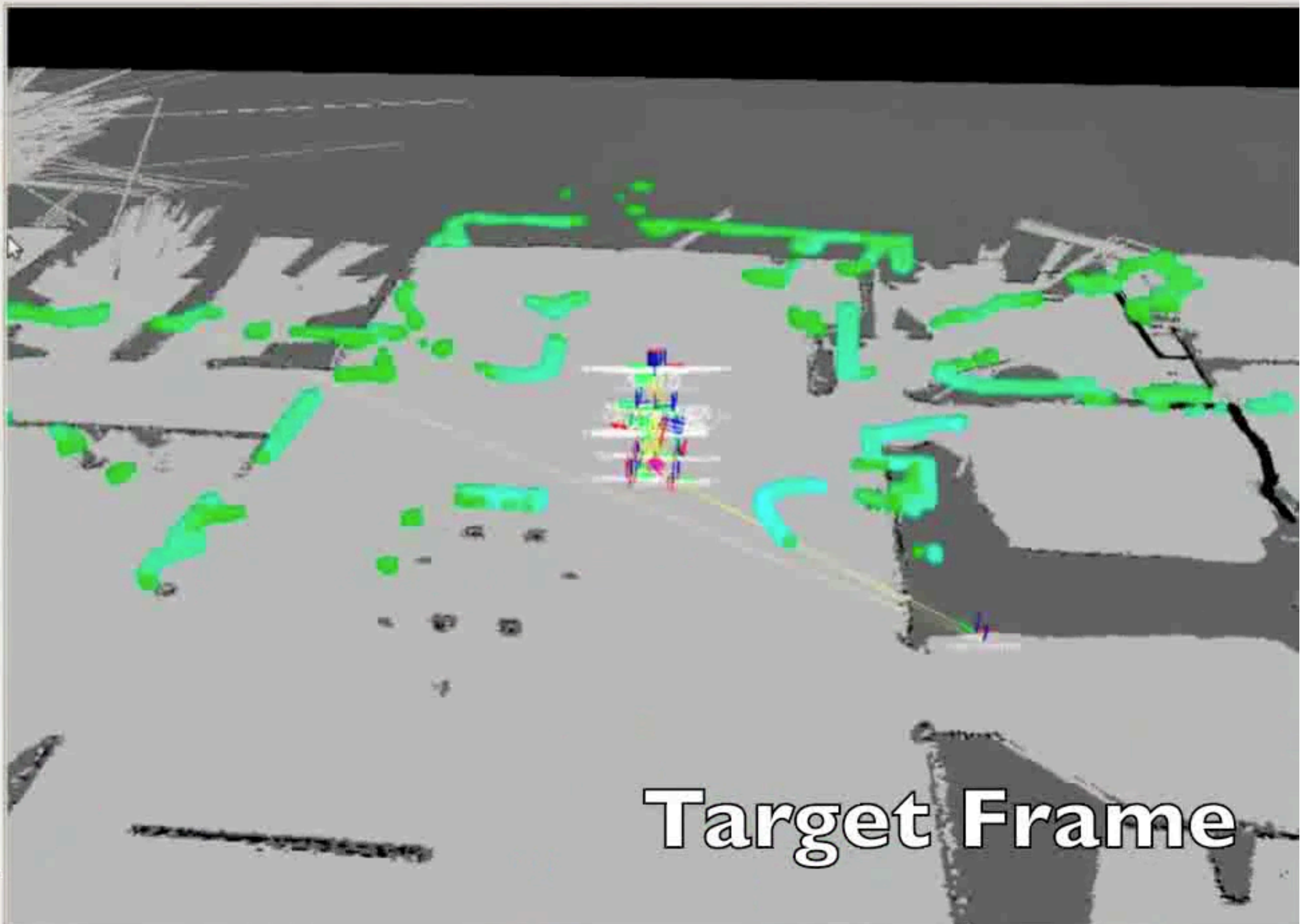
Center for Machine Perception
https://cmp.felk.cvut.cz

Department for Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague

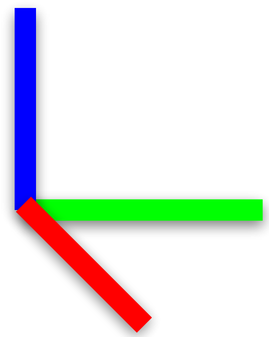
Why transformation among coordinate frames are important?

- Robot consist of many distributed components (sensors, actuators, joints)
- Each component operates in its own coordinate frame
- Robot moves => each coordinate frames changes in time.

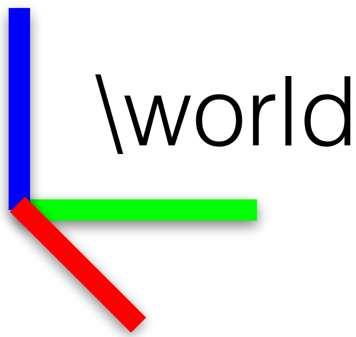



Target Frame

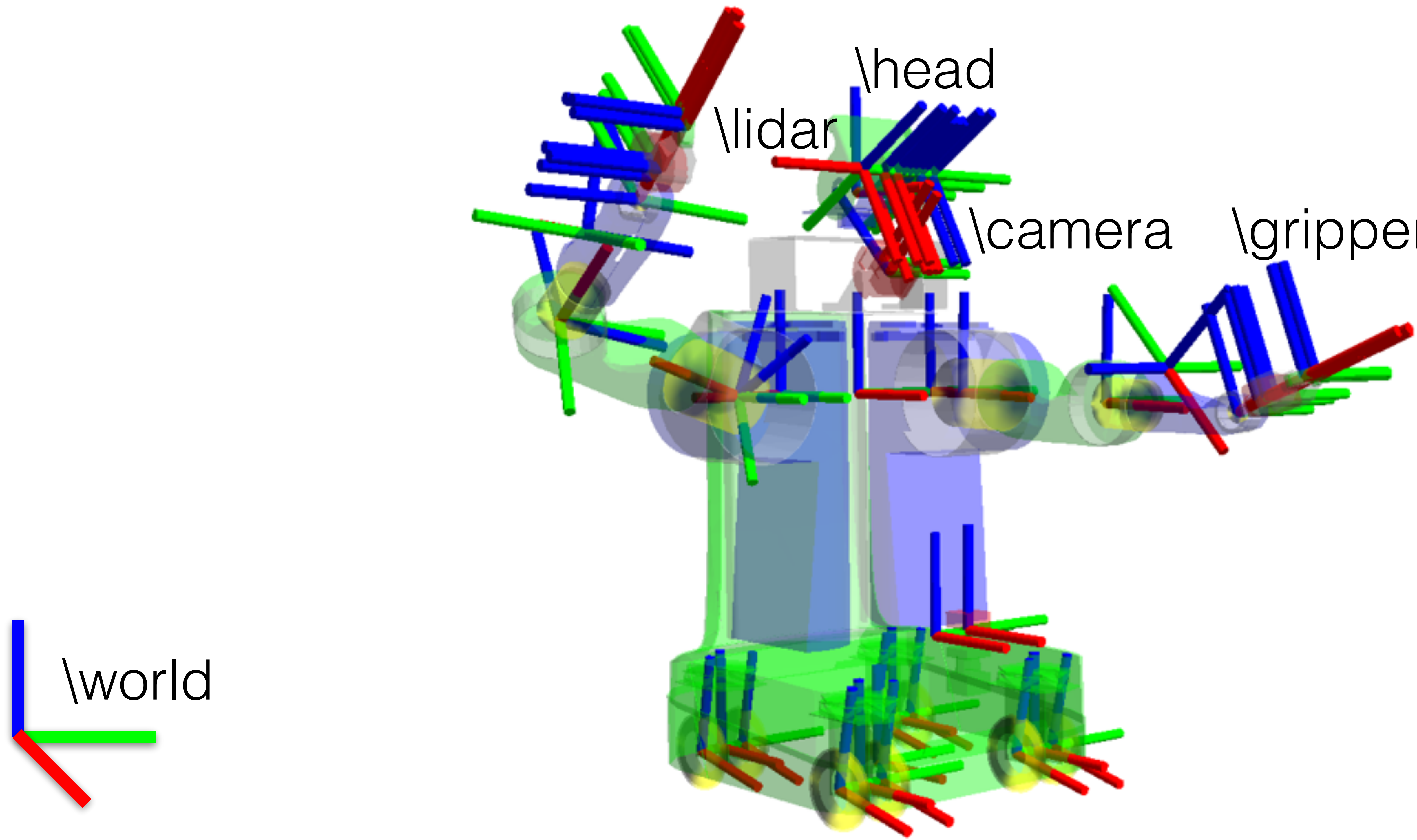
- Each coordinate frame define 3D Euclidean space.



- Each coordinate frame define 3D Euclidean space.
- It is uniquely determined by its name.

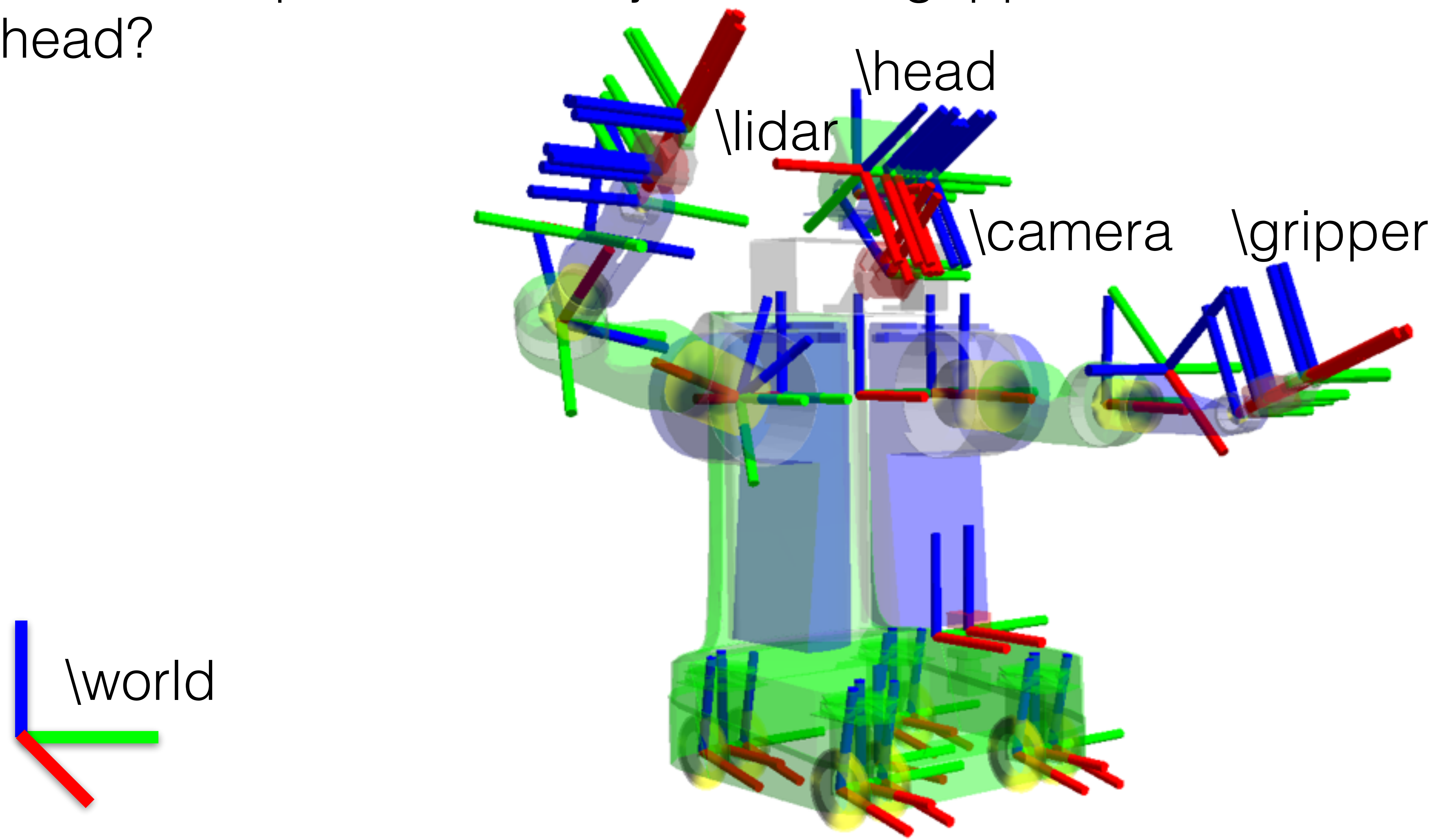


- Each coordinate frame define 3D Euclidean space.
- It is uniquely determined by its name.



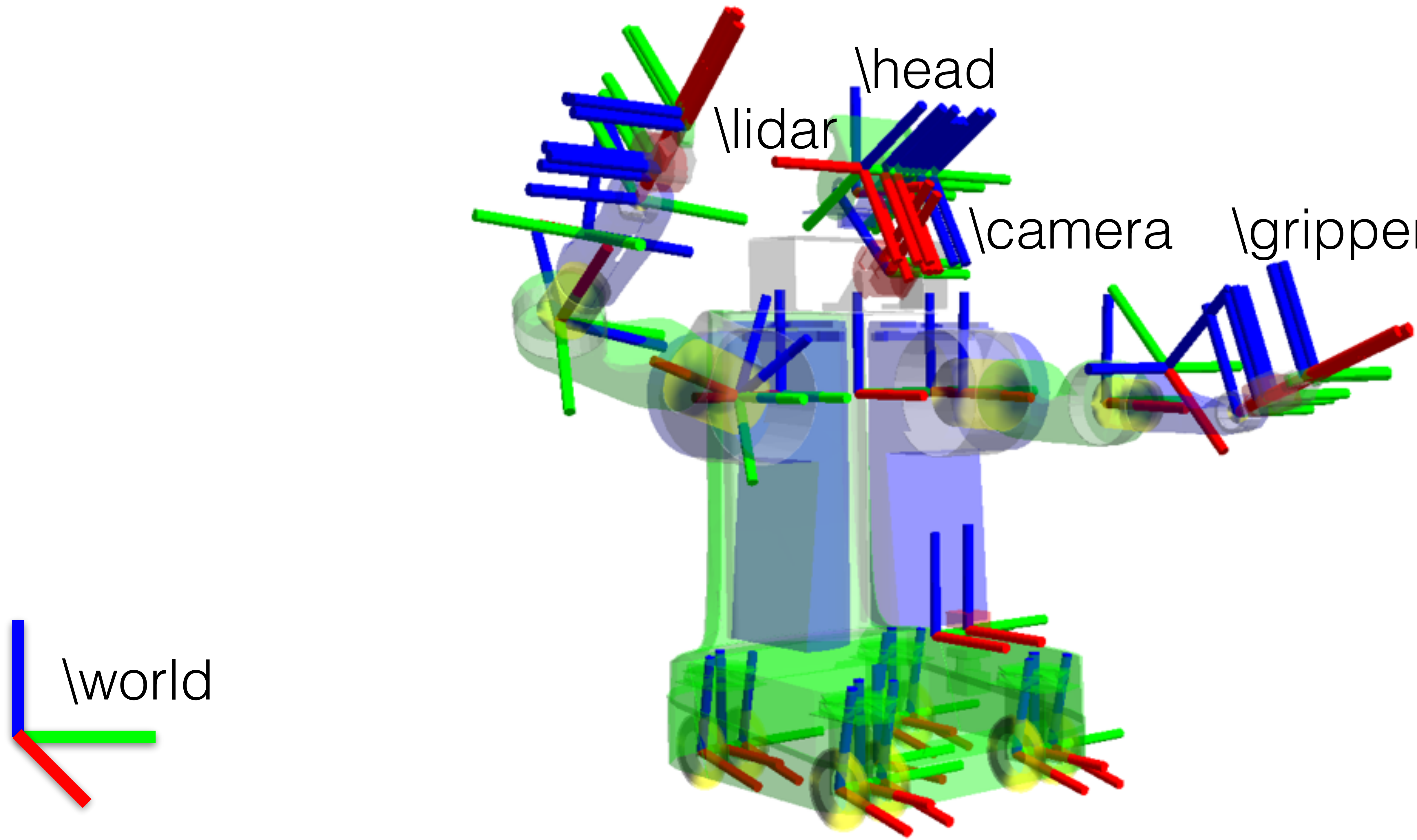
In arbitrary time we would like to answer questions like that:

- What is the pose of the \head in the \world?
- What color from \camera corresponds to 3D points measured by \lidar?
- What is the pose of the object in the \gripper relative to \head?



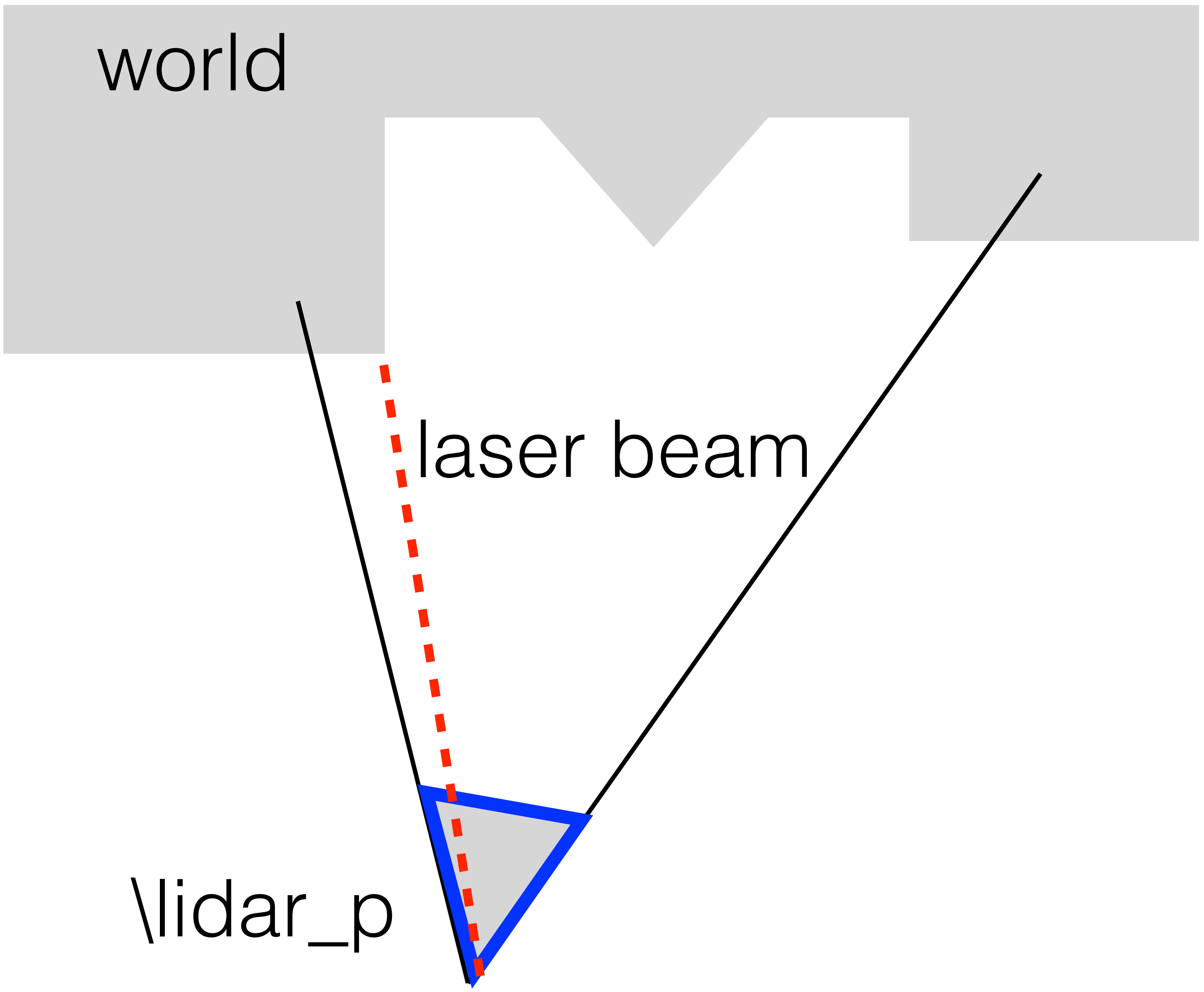
The topic of this lecture is to summarize:

- Lecture: mathematic definition of this transformations
- Labs: the way how ROS handles these transformations



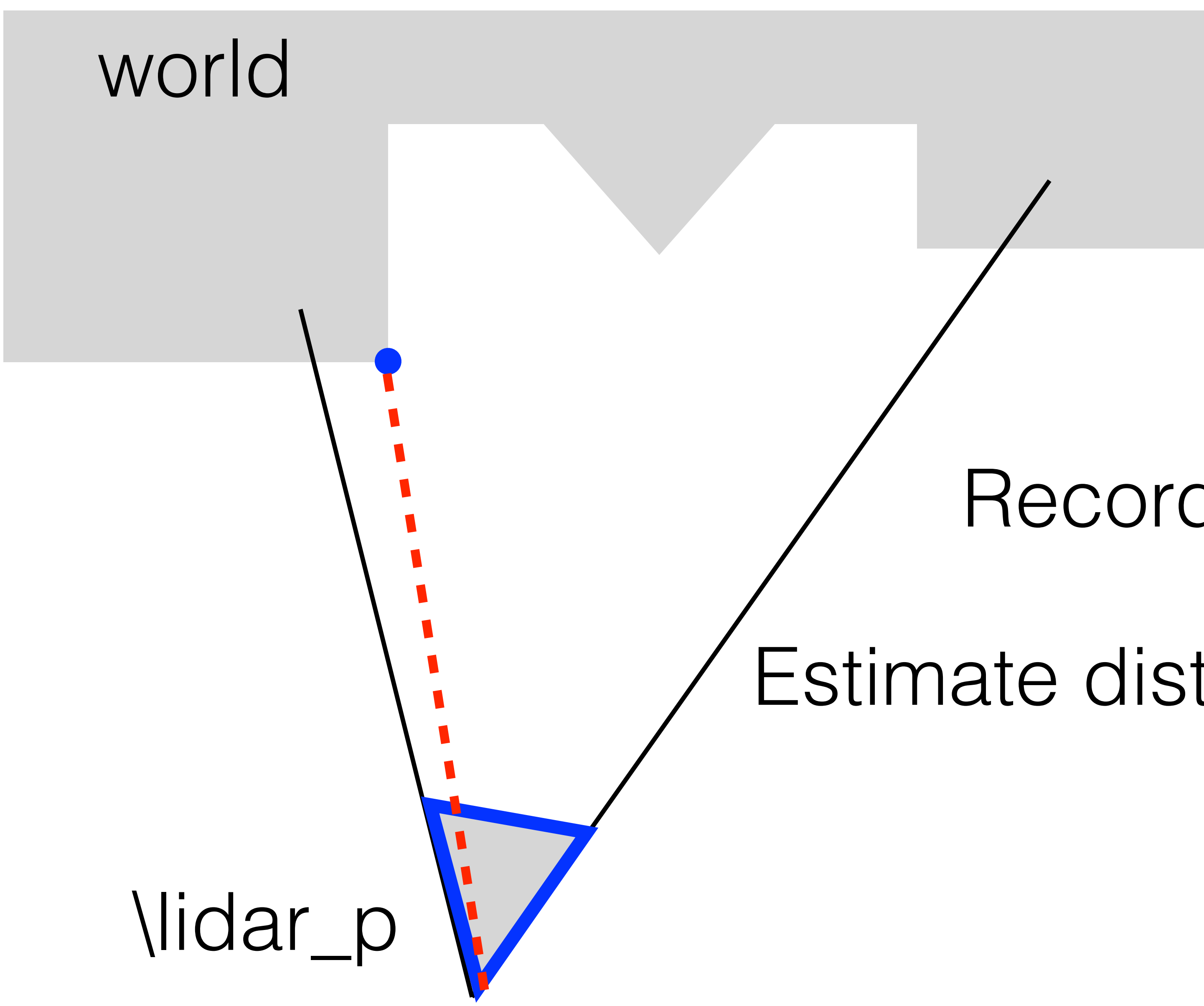
Lidar

Lidar is device which measure depth of some points in its field-of view by time-of-flight principle.



Lidar

Lidar is device which measure depth of some points in its field-of view by time-of-flight principle.

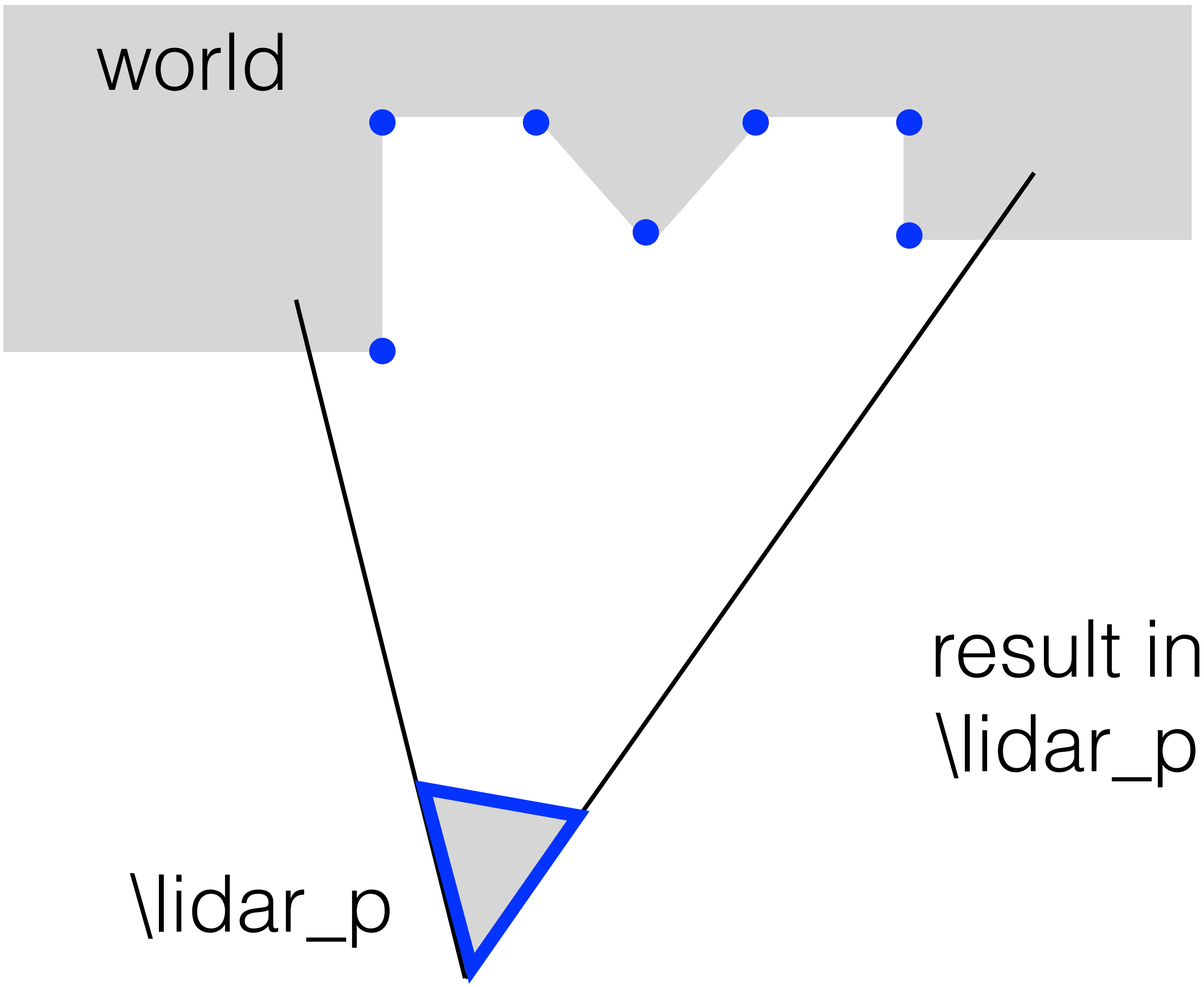


Record reflected beam

Estimate distance from time-of-flight.

Lidar

Lidar is device which measure depth of some points in its field-of view by time-of-flight principle.

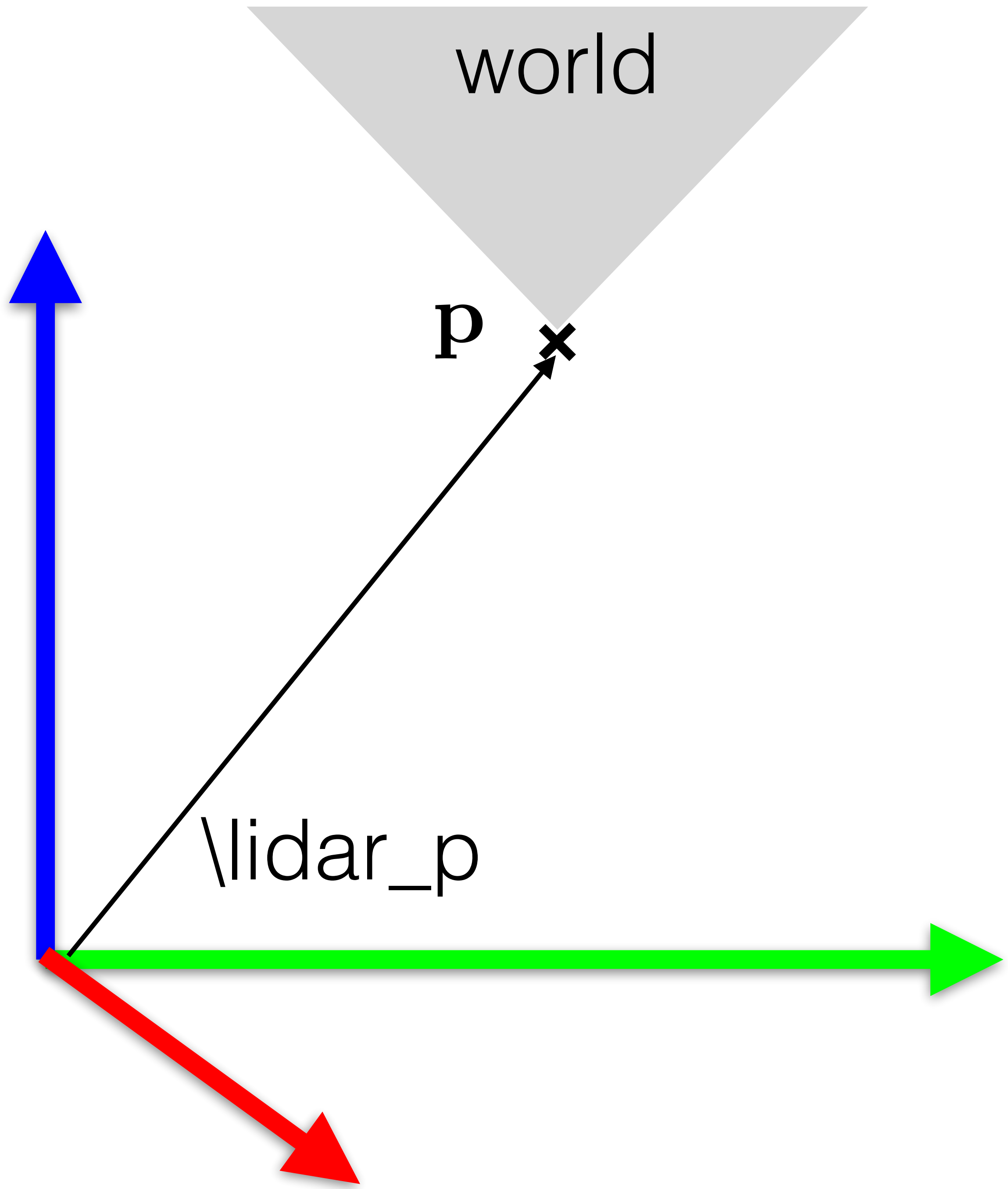


result in 3D point cloud in \lidar_p coordinate frame

Euclidean transformation of a rigid body

Let us now work only with two coordinate frames:

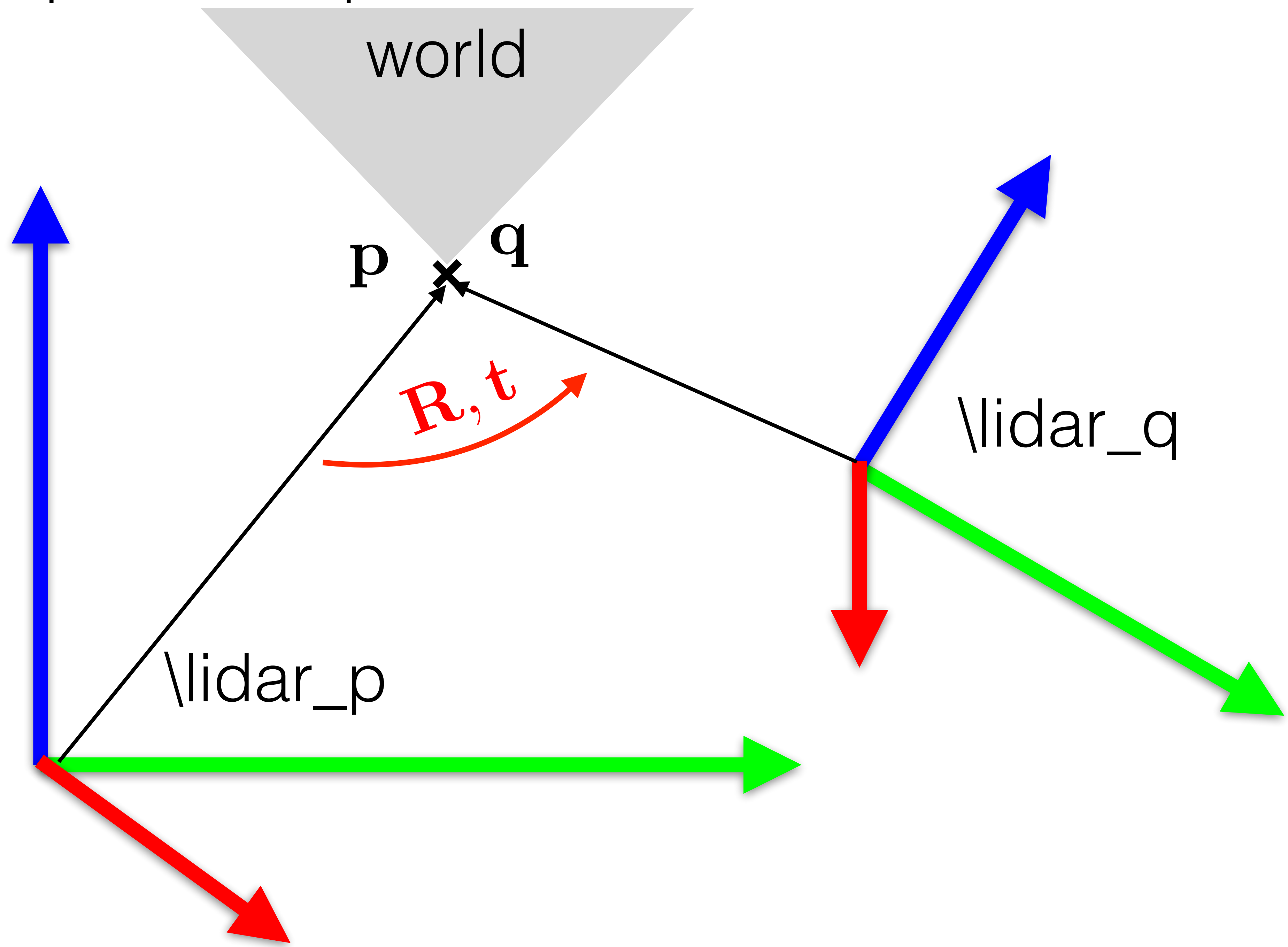
- \lidar_p in which points are denoted as $\mathbf{p} \in \mathcal{R}^3$



Euclidean transformation of a rigid body

Let us now work only with two coordinate frames:

- \lidar_p in which points are denoted as $\mathbf{p} \in \mathcal{R}^3$
- \lidar_q in which points are denoted as $\mathbf{q} \in \mathcal{R}^3$

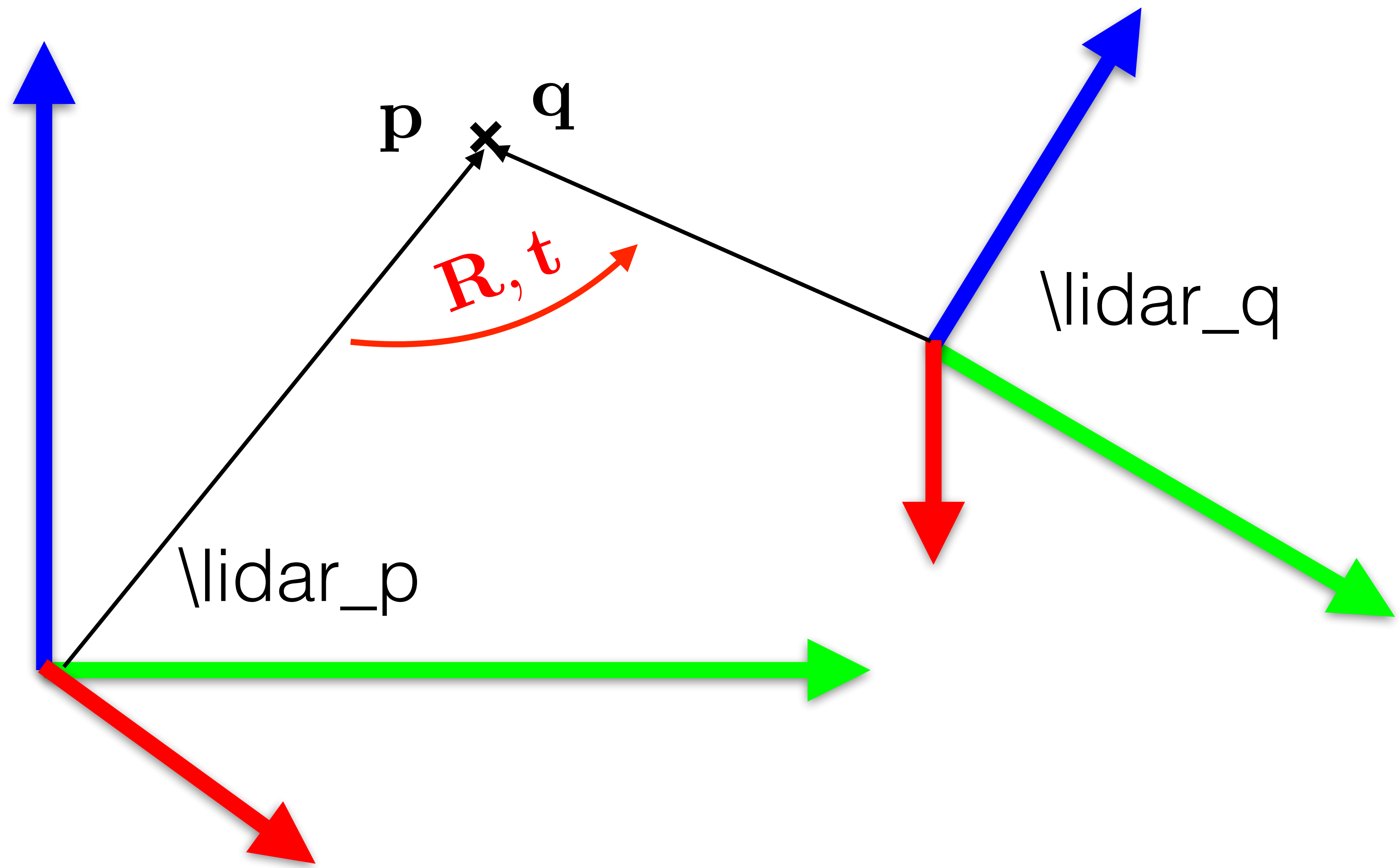


Euclidean transformation of a rigid body

Let us now work only with two coordinate frames:

- \lidar_p in which points are denoted as $\mathbf{p} \in \mathcal{R}^3$
- \lidar_q in which points are denoted as $\mathbf{q} \in \mathcal{R}^3$

$$\mathbf{q} = \mathbf{R}\mathbf{p} + \mathbf{t}$$



Euclidean transformation of a rigid body

- Assuming Euclidean motion (no squeezing)

$$\mathbf{q} = \mathbf{R}\mathbf{p} + \mathbf{t}$$

- where $\mathbf{R} \in \mathcal{SO}(3)$ is rotation and $\mathbf{t} \in \mathcal{R}^3$ is translation.

Euclidean transformation of a rigid body

- Assuming Euclidean motion (no squeezing)

$$\mathbf{q} = \mathbf{R}\mathbf{p} + \mathbf{t}$$

- where $\mathbf{R} \in \mathcal{SO}(3)$ is rotation and $\mathbf{t} \in \mathcal{R}^3$ is translation.
- Special orthogonal group

$$\mathcal{SO}(3)_{=}\{\mathbf{R} \in \mathcal{R}^{3\times 3} \mid \mathbf{R}^\top\mathbf{R} = \mathbf{I},\ \det(\mathbf{R}) = +1\}$$

Euclidean transformation of a rigid body

- Assuming Euclidean motion (no squeezing)

$$\mathbf{q} = \mathbf{R}\mathbf{p} + \mathbf{t}$$

- where $\mathbf{R} \in \mathcal{SO}(3)$ is rotation and $\mathbf{t} \in \mathcal{R}^3$ is translation.
- Special orthogonal group

$$\mathcal{SO}(3)_= \{\mathbf{R} \in \mathcal{R}^{3\times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I},\ \det(\mathbf{R}) = +1\}$$

- This is affine (not linear) transformation $\Rightarrow$ introduce homogeneous coordinates

$$\overline{\mathbf{p}} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}$$

- Euclidean transformation is given by matrix $\mathbf{g} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$

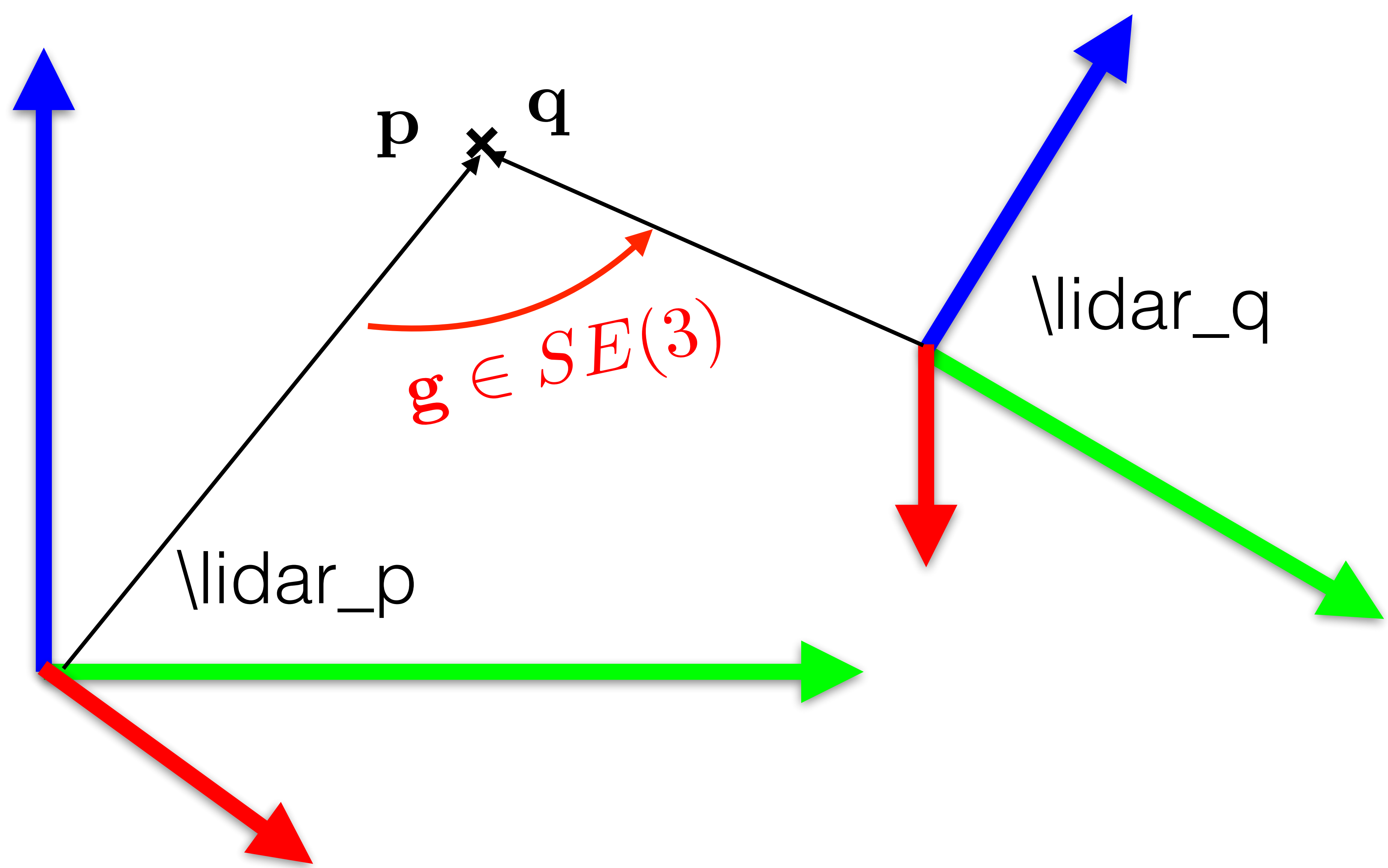
$$\overline{\mathbf{q}} = \mathbf{g}\,\overline{\mathbf{p}}$$

Euclidean transformation of a rigid body

- Set of all transformations forms Special Euclidean group

$$\mathcal{SE}(3) = \{\mathbf{g} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mid \mathbf{R} \in \mathcal{SO}(3), \mathbf{t} \in \mathcal{R}^3\}$$

where $\mathcal{SO}(3)_{=}\{\mathbf{R} \in \mathcal{R}^{3\times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I},\ \det(\mathbf{R}) = +1\}$



Euclidean transformation of a rigid body

Example:

Given transformation from \lidar1 to \lidar2 $\mathbf{g}_{12}$, what is inverse transformation $\mathbf{g}_{21}$

$$\mathbf{g}_{12} = \begin{bmatrix} \mathtt{R}_{12} & \mathbf{t}_{12} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{g}_{21} = ?$$

Summary

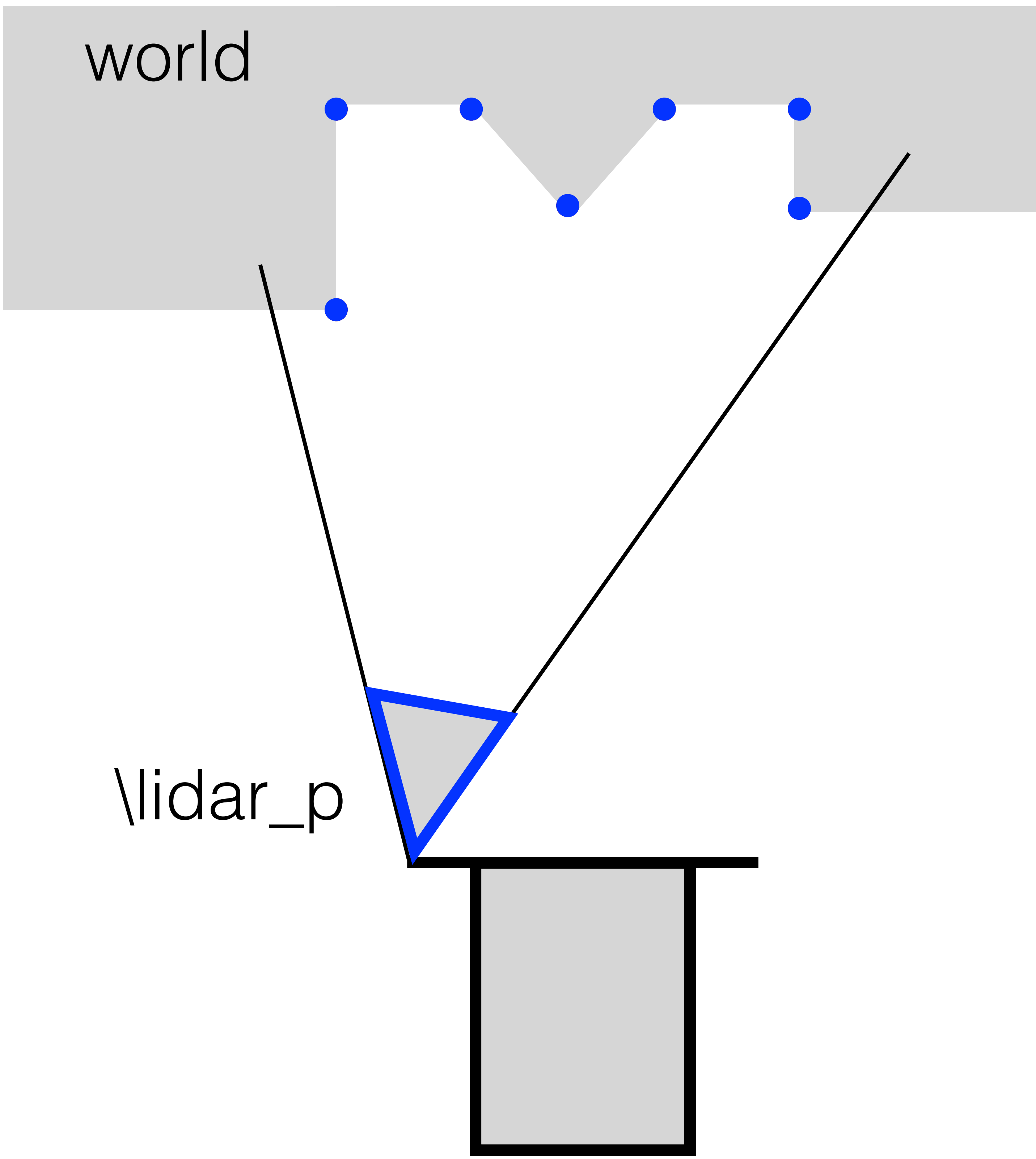
$$\mathcal{SE}(3) = \{\mathbf{g} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \ \mid \ \mathbf{R} \in \mathcal{SO}(3), \mathbf{t} \in \mathcal{R}^3\}$$

is set of transformations, which model spatio-temporal change of coordinate systems among sensors, actuators and world.

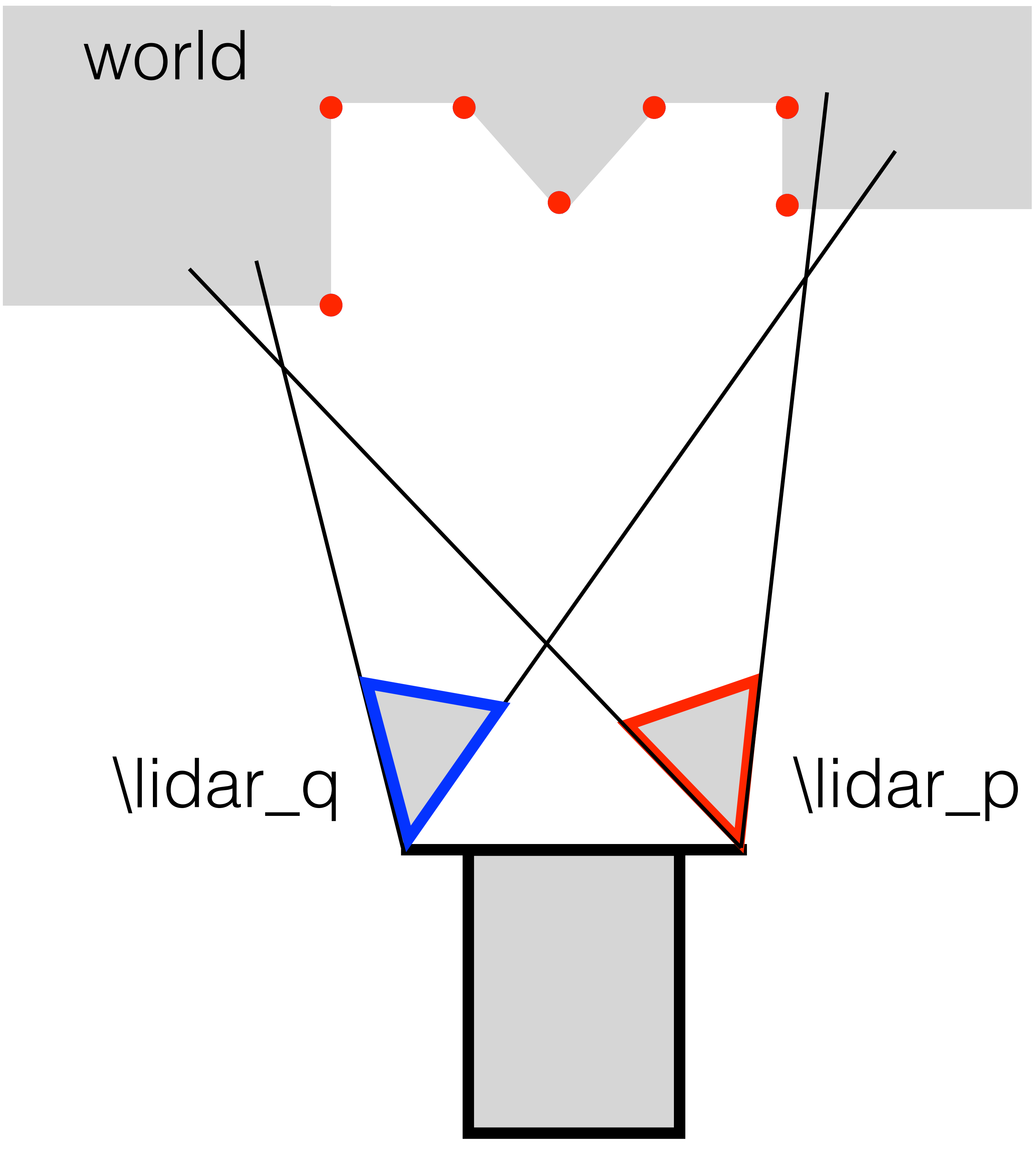
Mutual calibration of two coordinate frames

- Let us consider two lidars mounted on robot body.
- Each measures pointcloud in its own c.f.
- How can we estimate transformation between them?

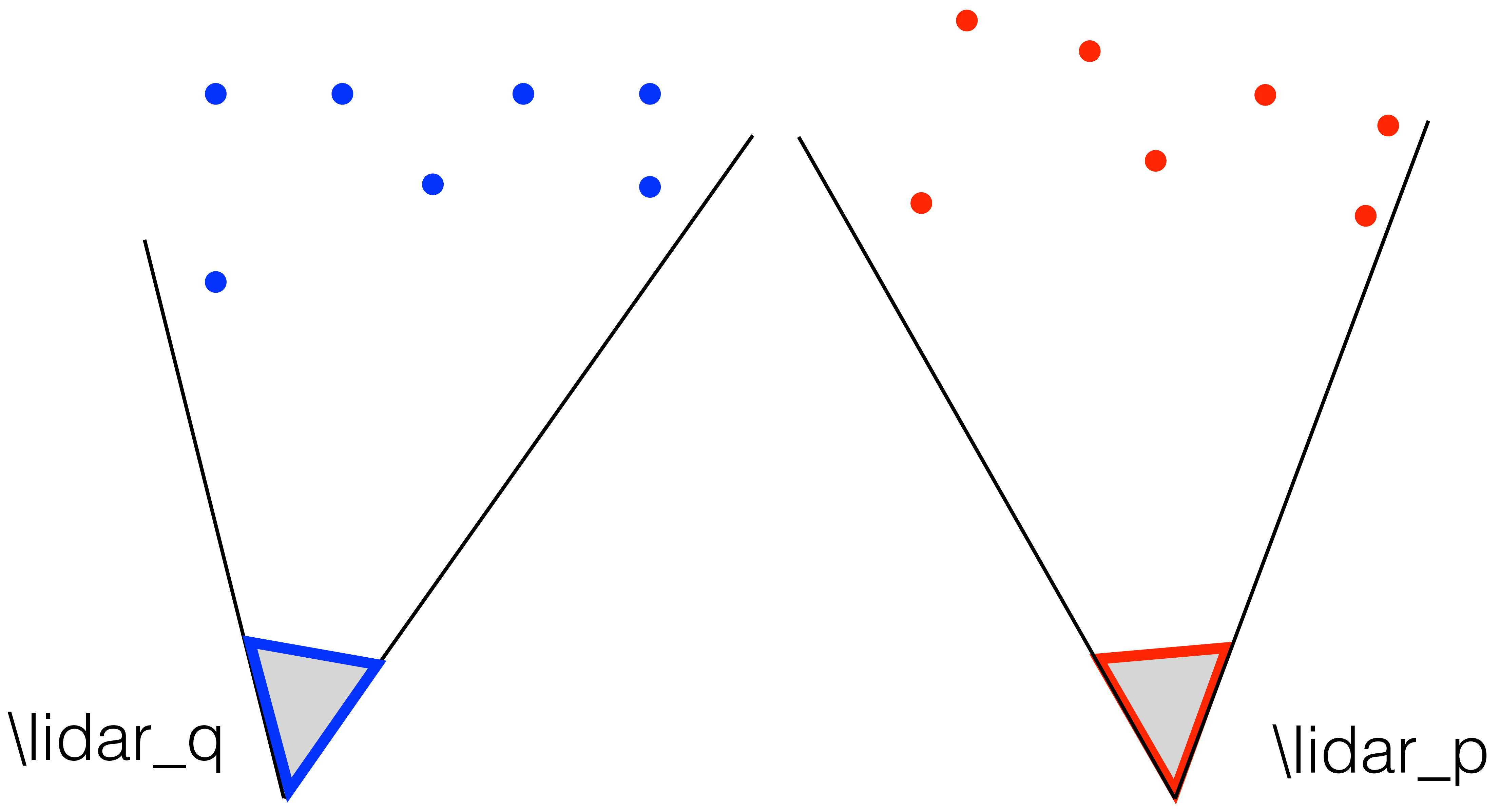
Mutual calibration of two coordinate frames



Mutual calibration of two coordinate frames

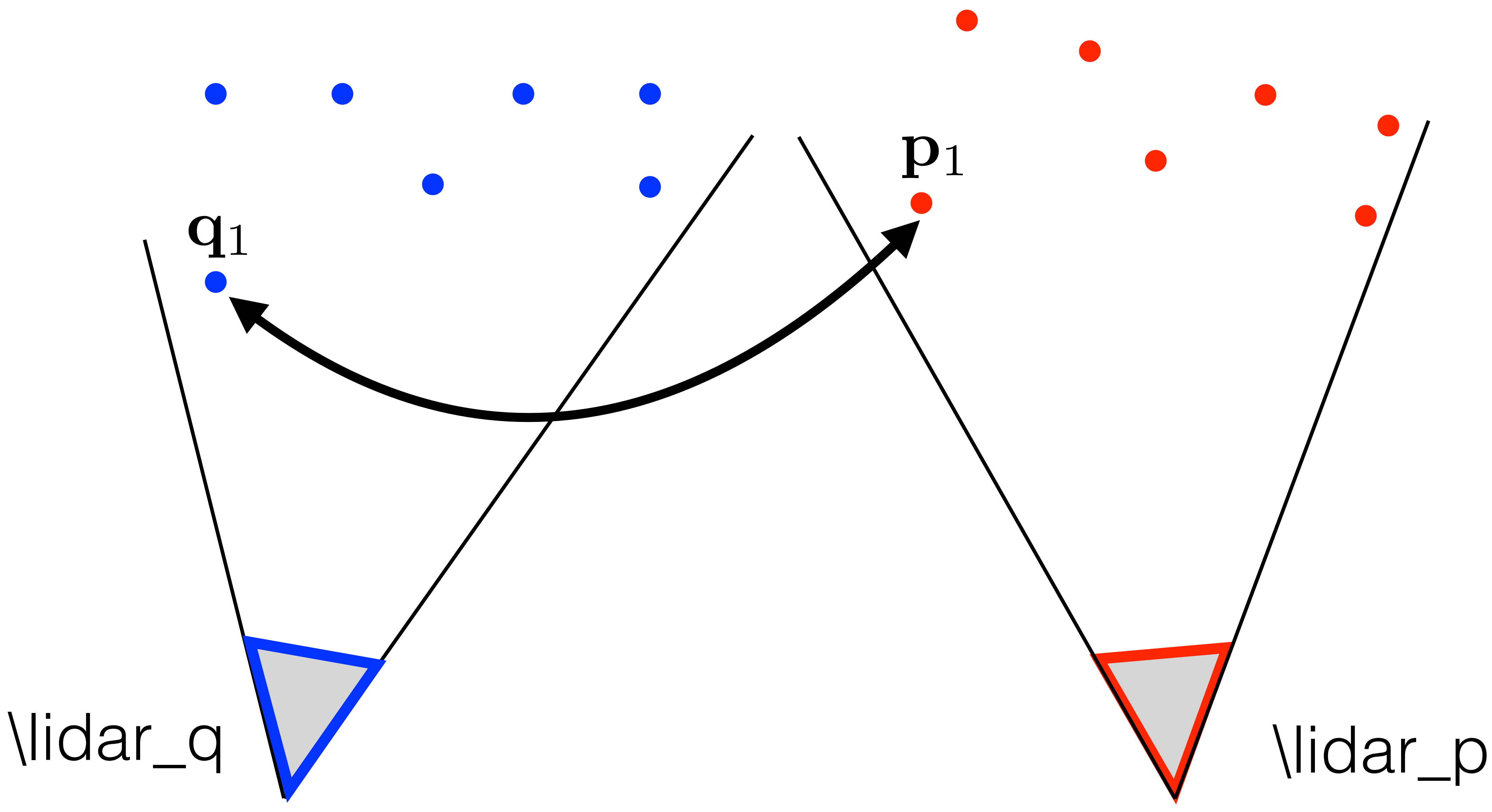


Mutual calibration of two coordinate frames



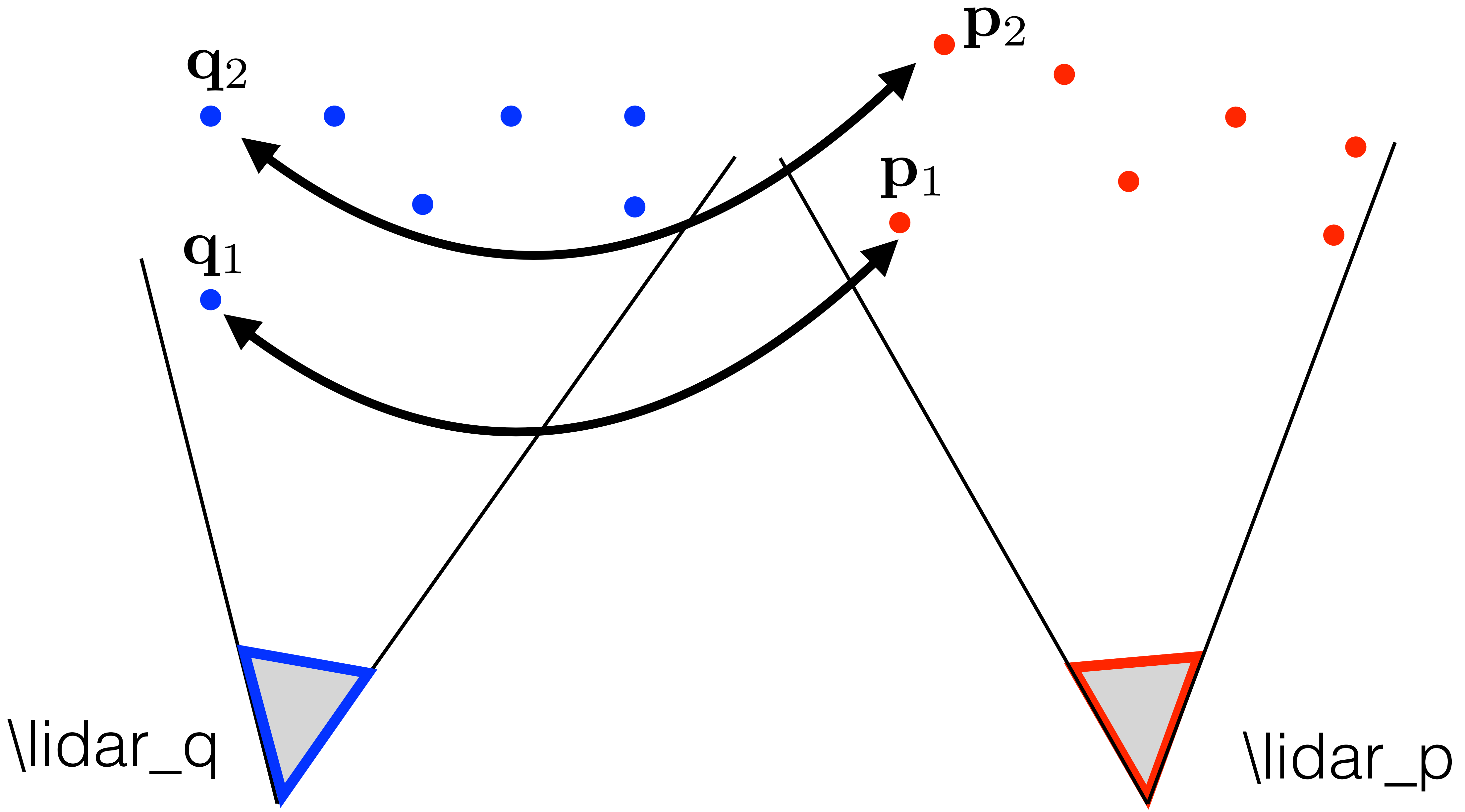
Mutual calibration of two coordinate frames

3D-3D correspondences

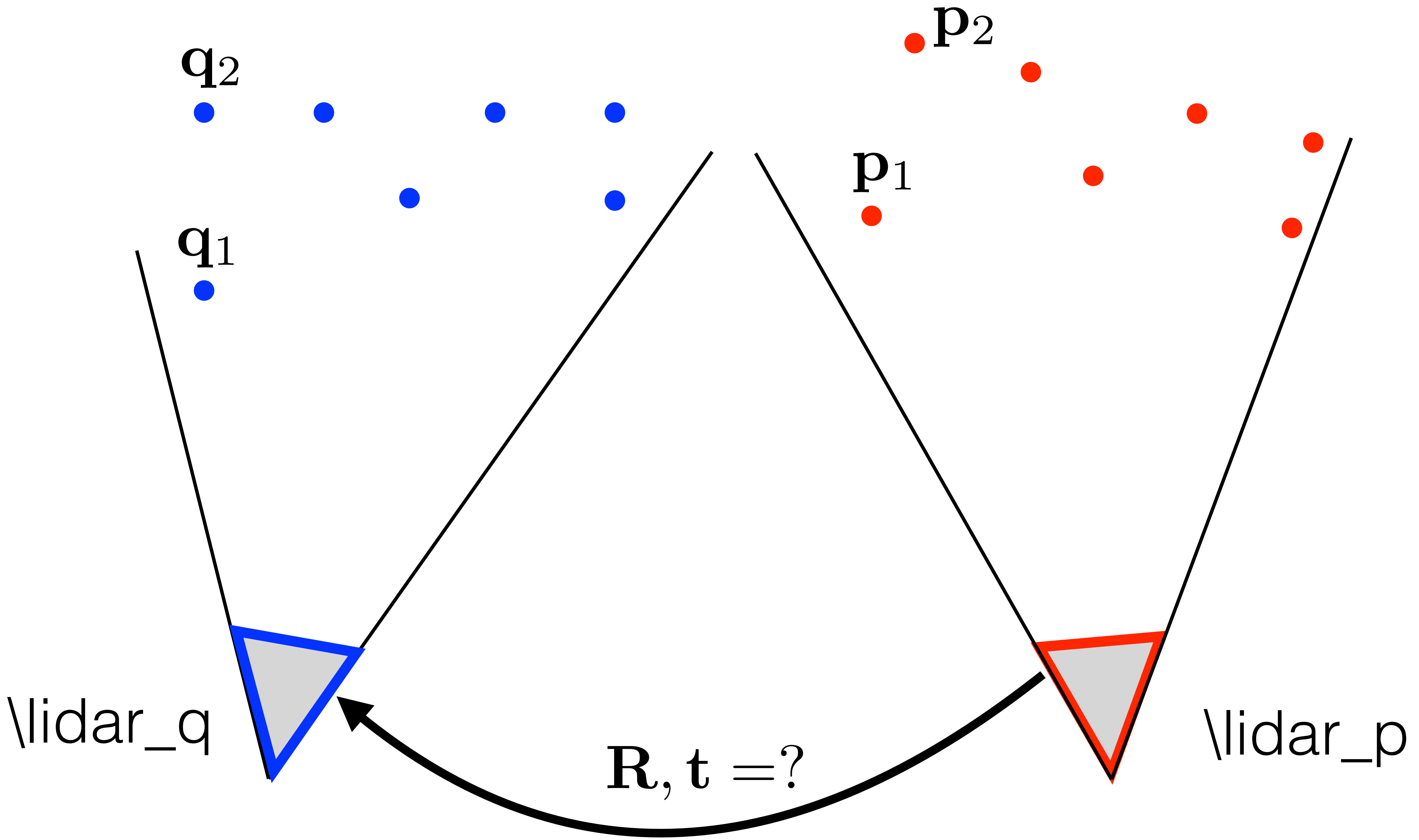


Mutual calibration of two coordinate frames

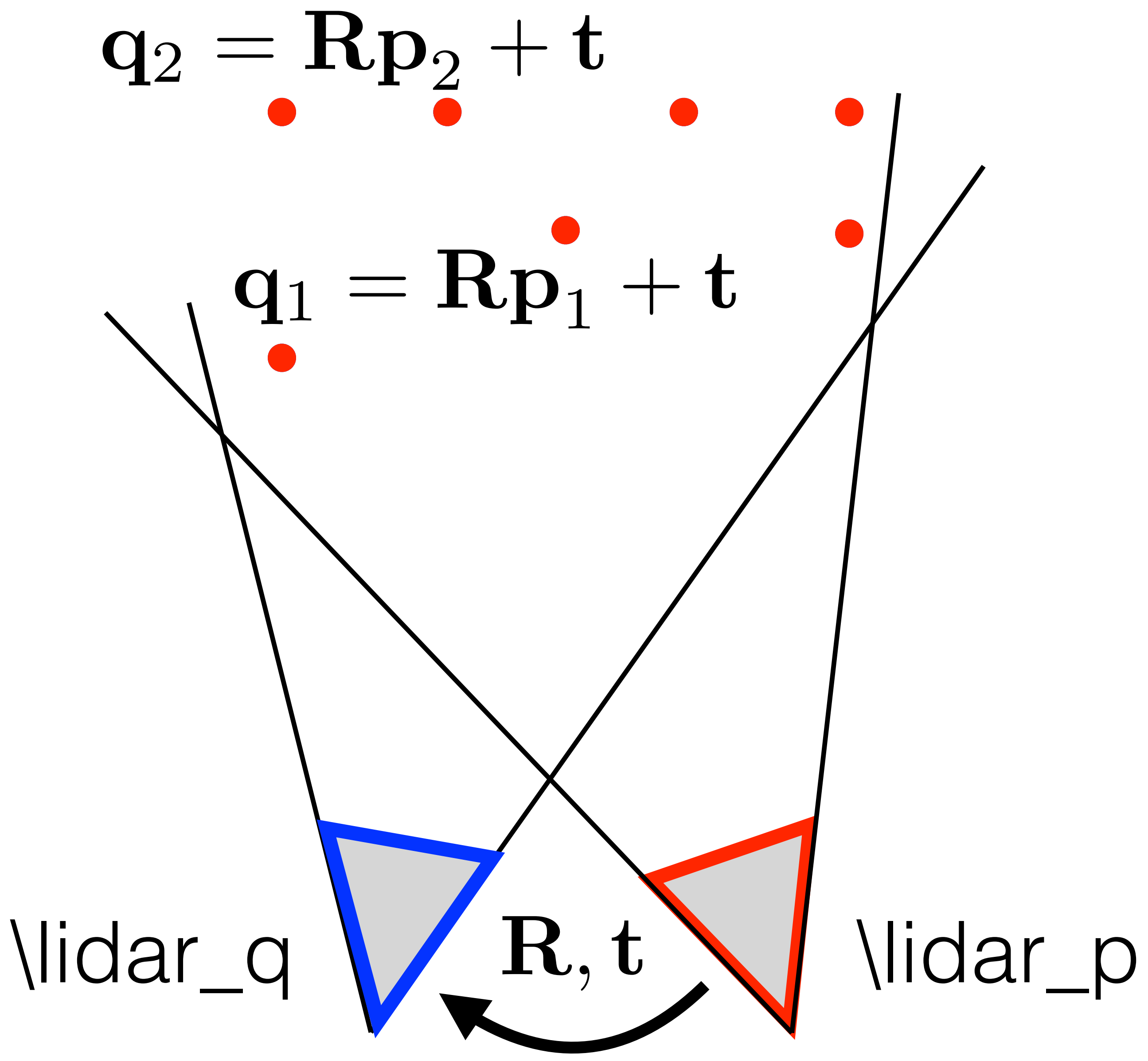
3D-3D correspondences



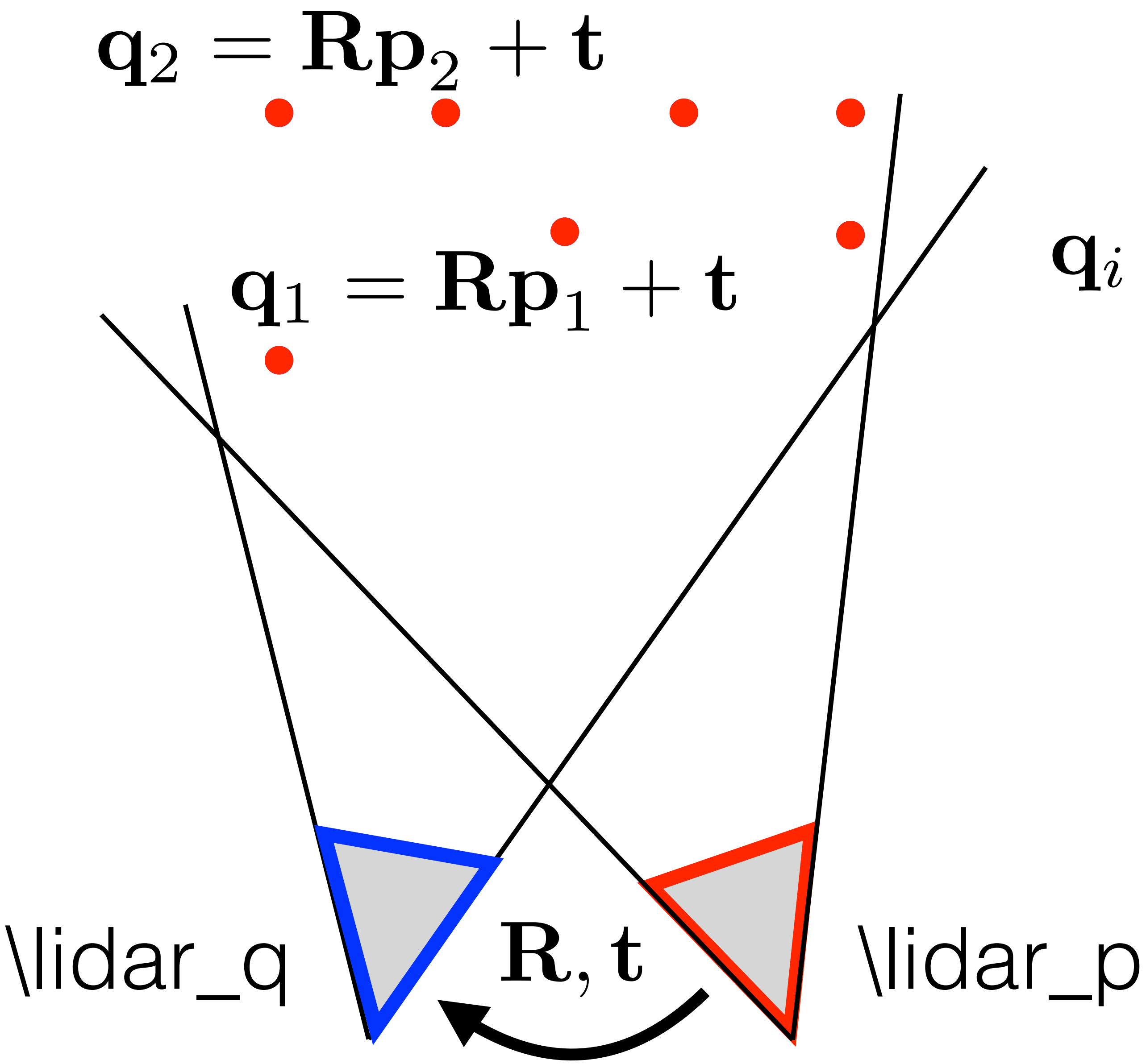
Mutual calibration of two coordinate frames



Mutual calibration of two coordinate frames



Mutual calibration of two coordinate frames



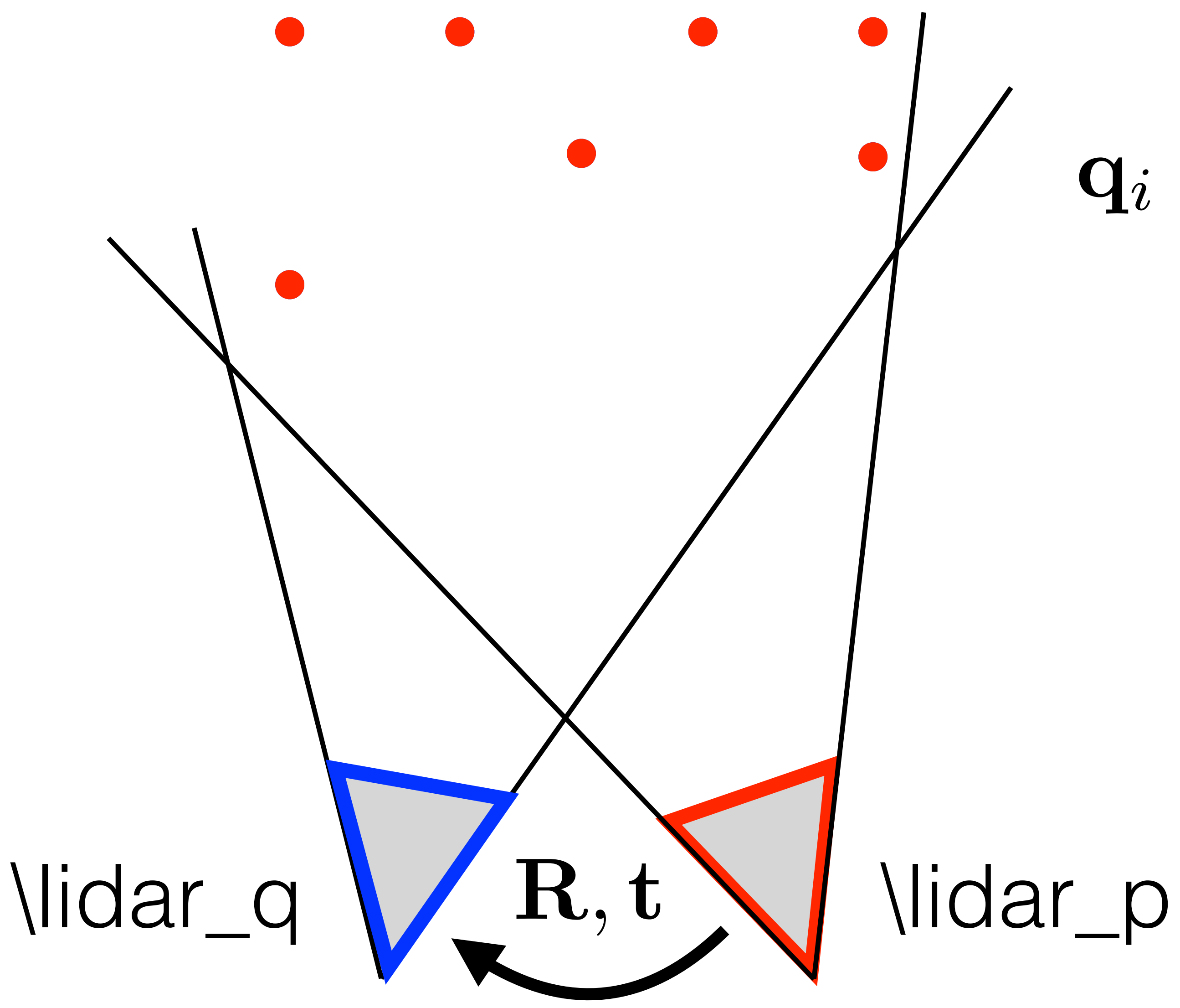
3N equations:

$$\mathbf{q}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t} \quad \forall_{i=1 \ldots N}$$

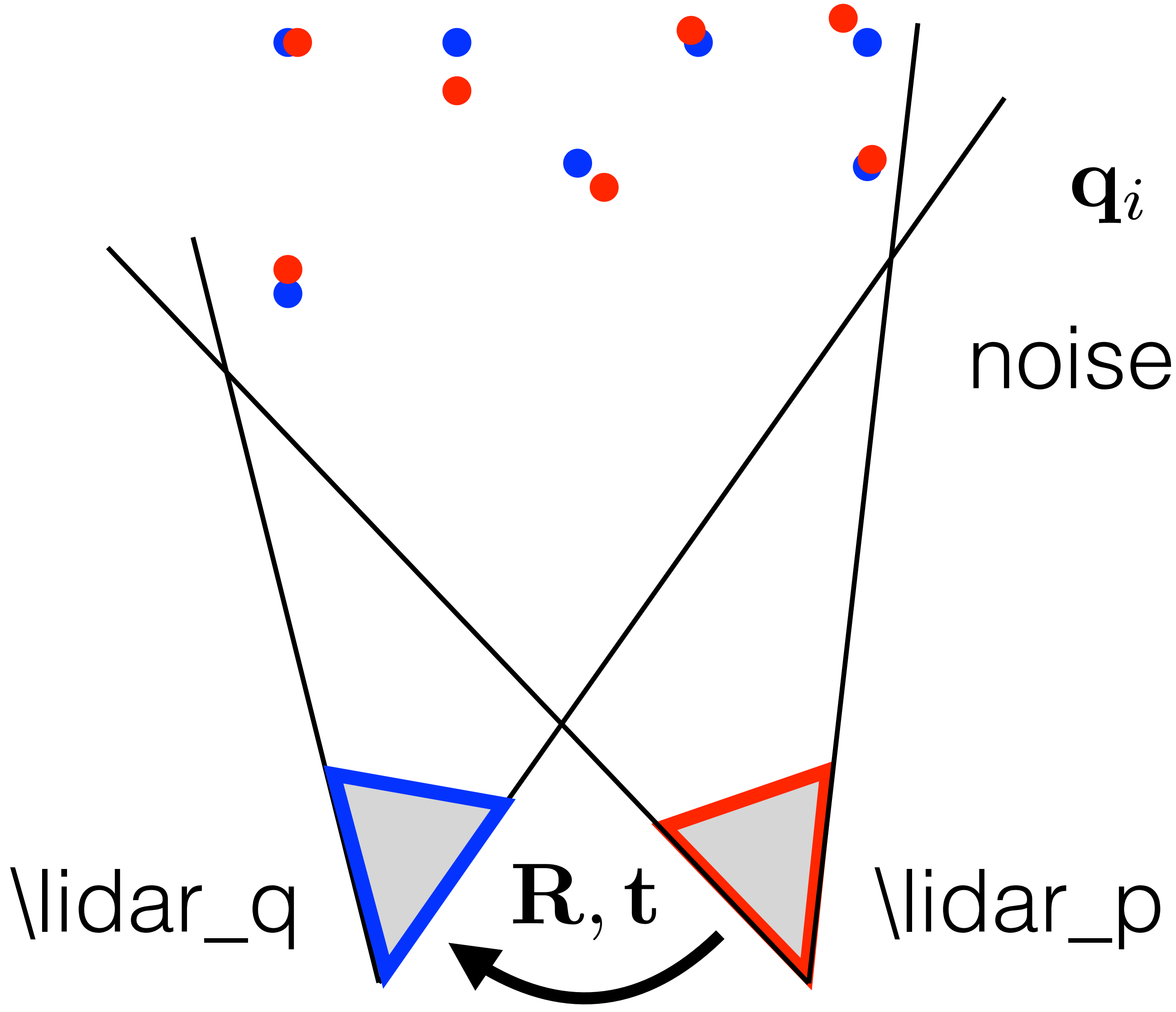
Mutual calibration of two coordinate frames

3N equations:

$$\mathbf{q}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t} \quad \forall_{i=1\ldots N}$$



Mutual calibration of two coordinate frames

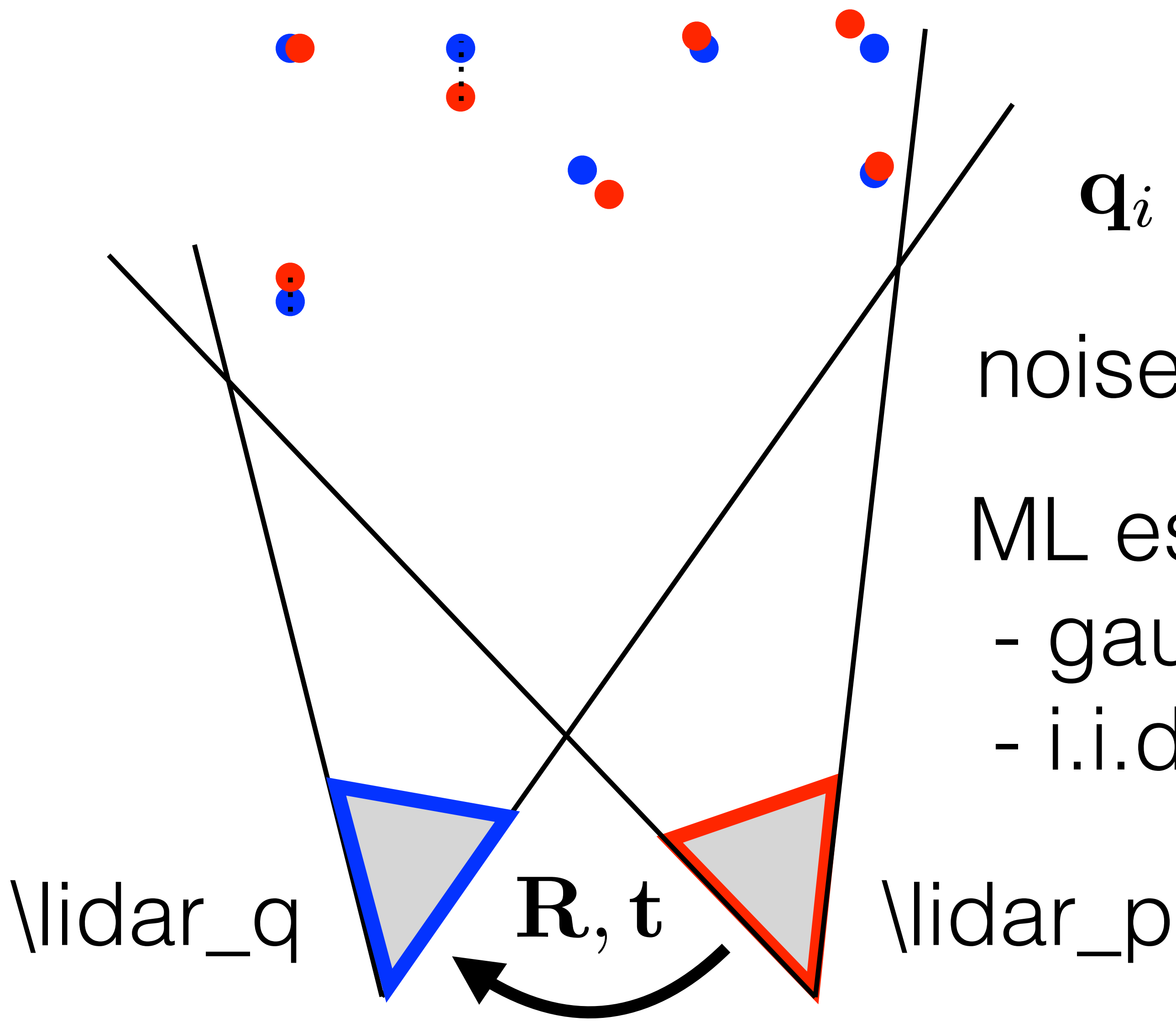


3N equations:

$$\mathbf{q}_i \neq \mathbf{R}\mathbf{p}_i + \mathbf{t} \quad \forall_{i=1\ldots N}$$

noise => no exact solution

Mutual calibration of two coordinate frames



3N equations:

$$\mathbf{q}_i \neq \mathbf{R}\mathbf{p}_i + \mathbf{t} \quad \forall_{i=1\ldots N}$$

noise => no exact solution

ML estimate wrt:

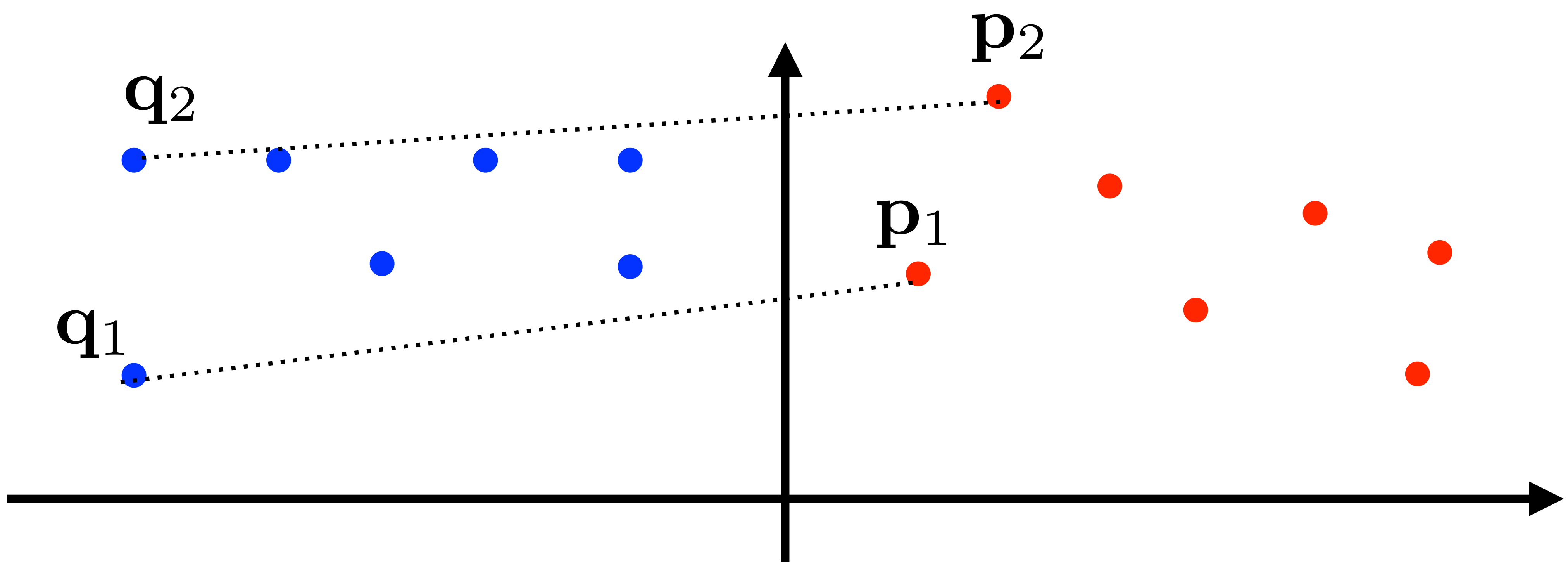
- gaussian noise
- i.i.d measurements

$$\mathbf{R}^*, \mathbf{t}^* = \arg\min_{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

Mutual calibration of two coordinate frames

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

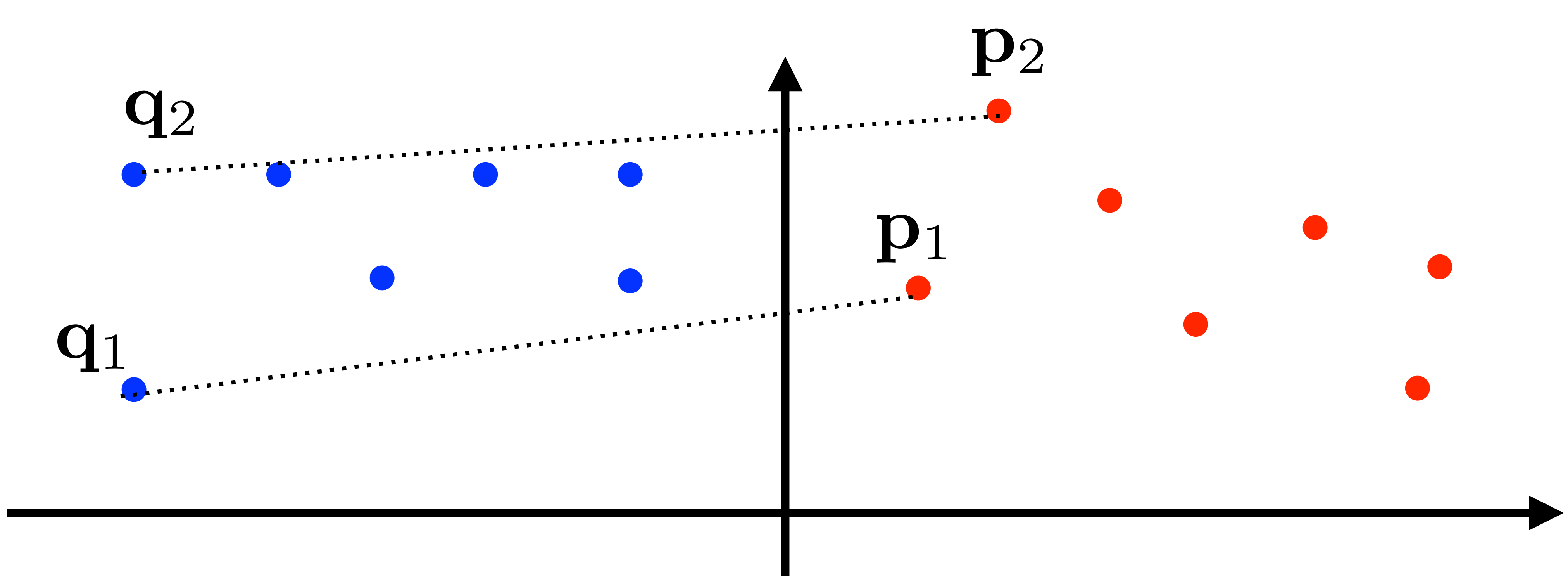
Solution:



Mutual calibration of two coordinate frames

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

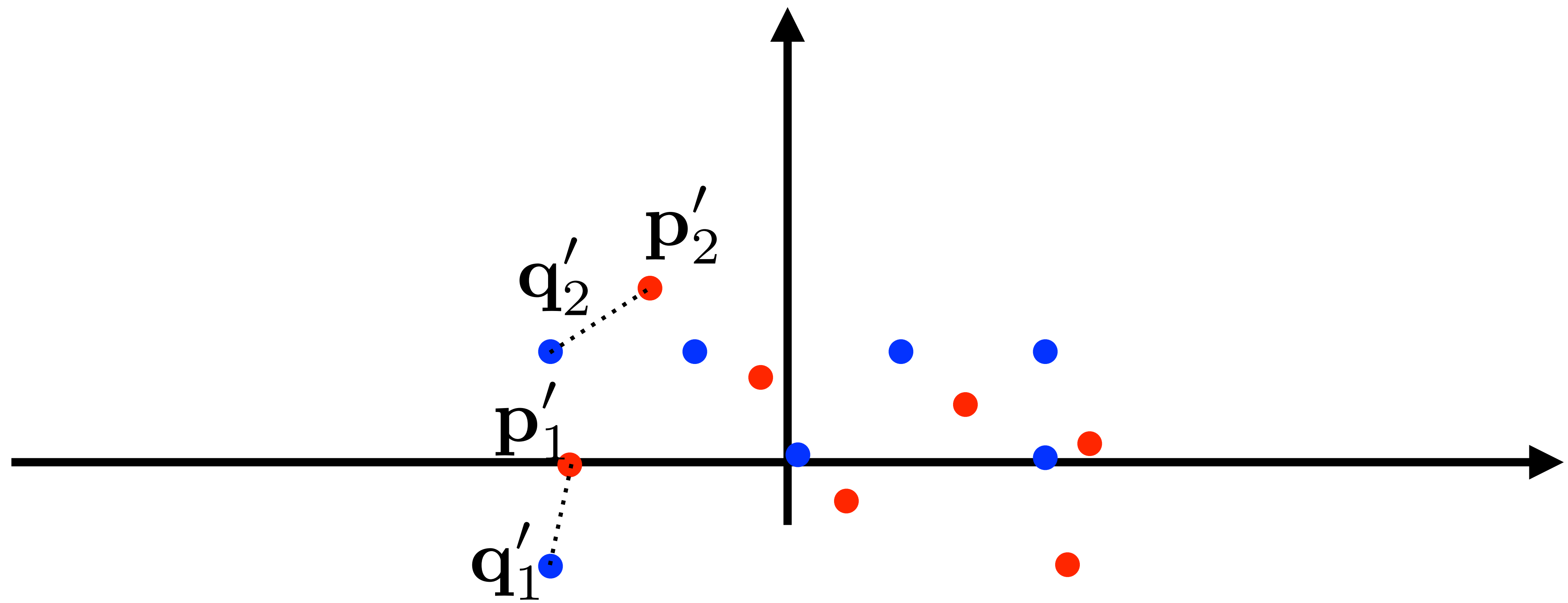
Solution: $$\mathbf{p}'_i = \mathbf{p}_i - \underbrace{\frac{1}{N}\sum_i \mathbf{p}_i}_{\widetilde{\mathbf{p}}}, \quad \mathbf{q}'_i = \mathbf{q}_i - \underbrace{\frac{1}{N}\sum_i \mathbf{q}_i}_{\widetilde{\mathbf{q}}}$$



Mutual calibration of two coordinate frames

$$\mathbf{R}^*, \mathbf{t}^* = \arg\min_{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

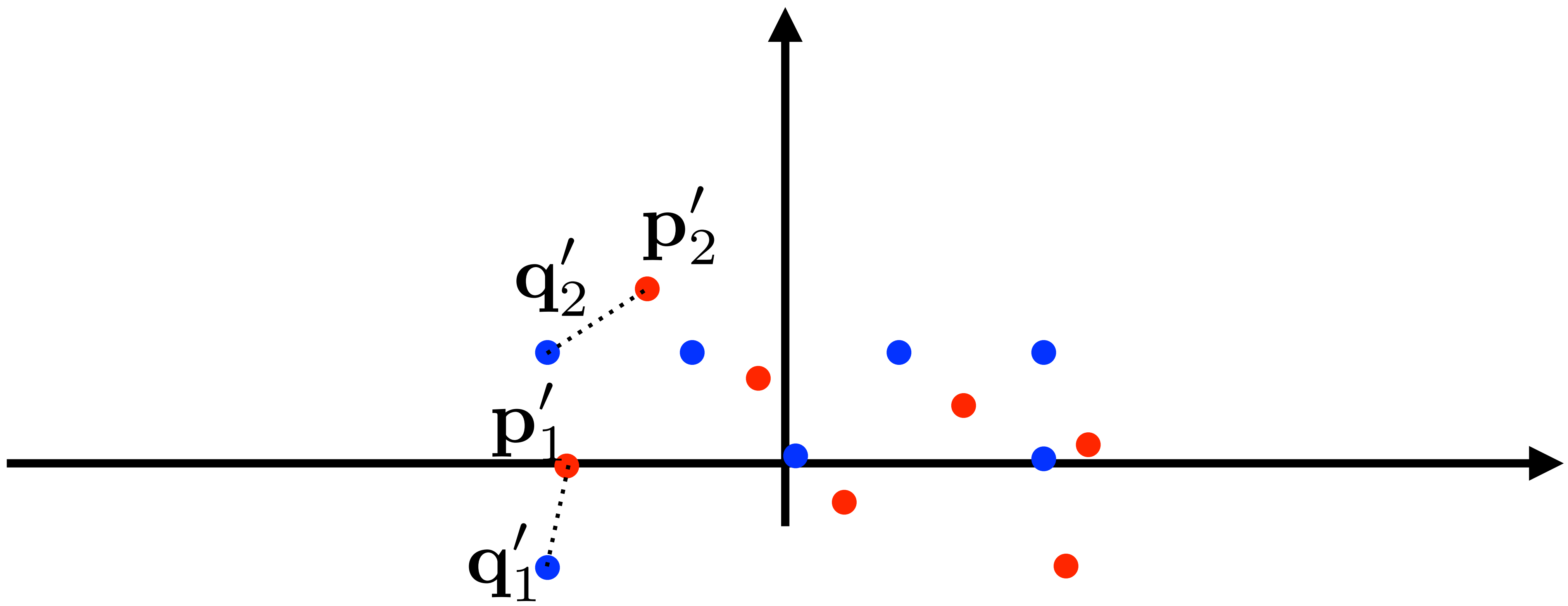
Solution: $$\mathbf{p}'_i = \mathbf{p}_i - \underbrace{\frac{1}{N}\sum_i \mathbf{p}_i}_{\widetilde{\mathbf{p}}}, \quad \mathbf{q}'_i = \mathbf{q}_i - \underbrace{\frac{1}{N}\sum_i \mathbf{q}_i}_{\widetilde{\mathbf{q}}}$$



Mutual calibration of two coordinate frames

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

Solution: estimate covariante matrix: $\mathbf{H} = \sum_i \mathbf{p}_i' \mathbf{q}_i'^\top$

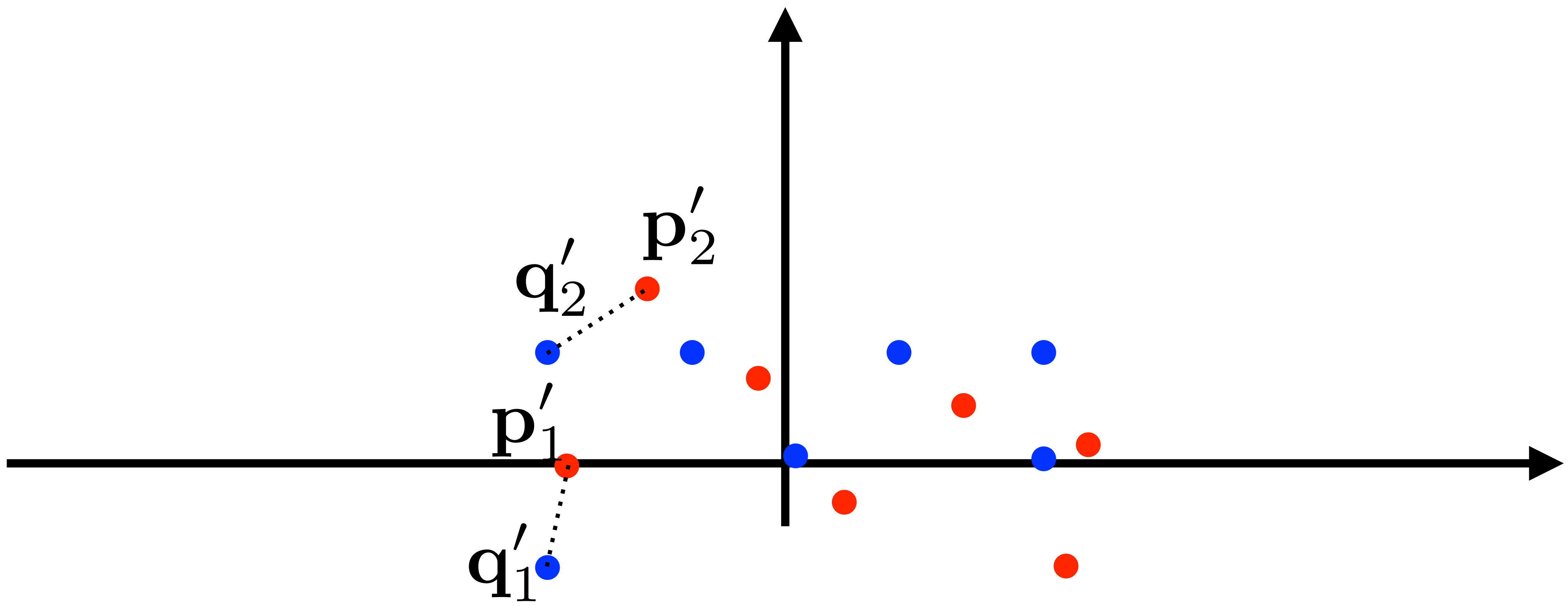


Mutual calibration of two coordinate frames

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

Solution: estimate covariante matrix: $\mathbf{H} = \sum_i \mathbf{p}'_i \mathbf{q}'^{\top}_i$

find SVD decomposition: $\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$



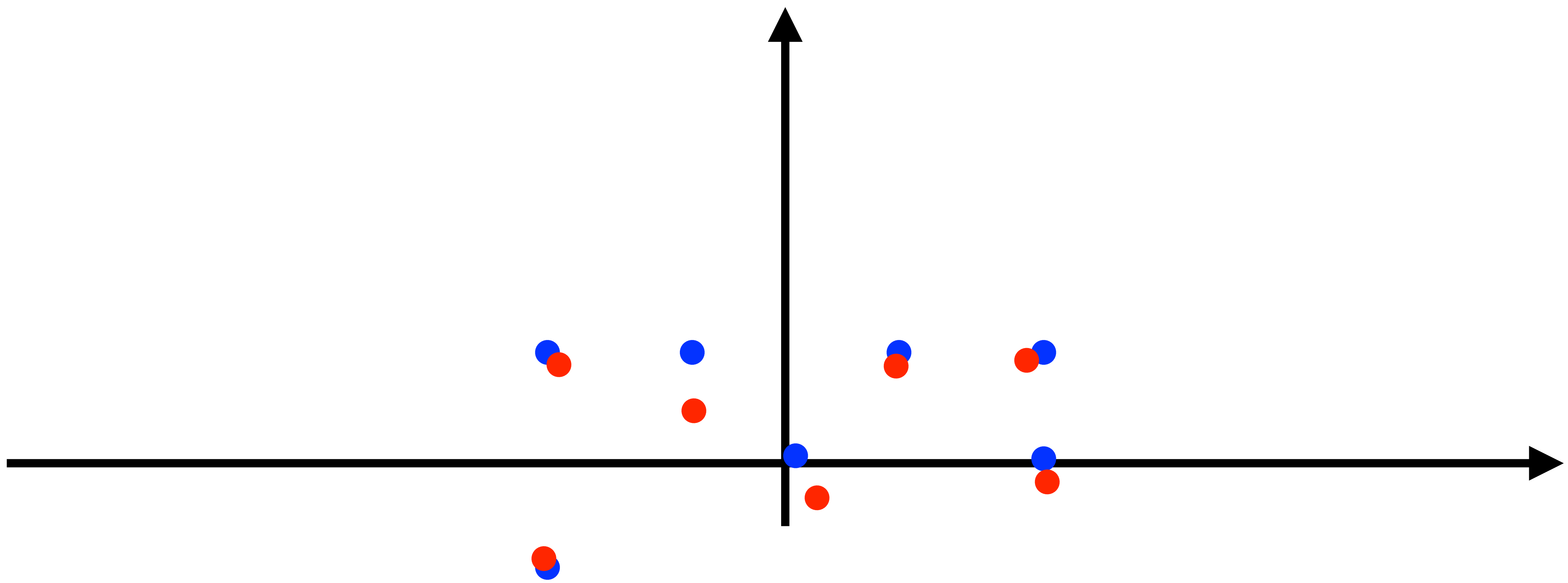
Mutual calibration of two coordinate frames

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

Solution: estimate covariante matrix: $\mathbf{H} = \sum_i \mathbf{p}_i' \mathbf{q}_i'^\top$

find SVD decomposition: $\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$

estimate optimal rotation: $\mathbf{R}^* = \mathbf{V}\mathbf{U}^\top$



Mutual calibration of two coordinate frames

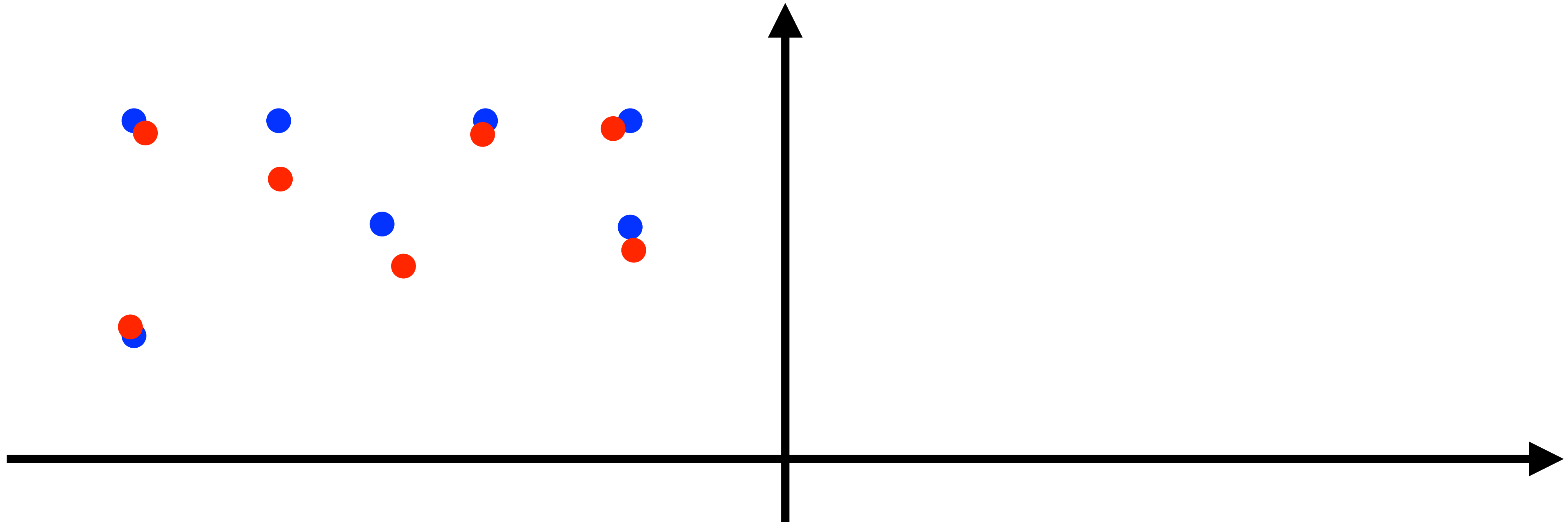
$$\mathbf{R}^*, \mathbf{t}^* = \arg\min_{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

Solution: estimate covariante matrix: $\mathbf{H} = \sum_i \mathbf{p}'_i \mathbf{q}'^{\top}_i$

find SVD decomposition: $\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$

estimate optimal rotation: $\mathbf{R}^* = \mathbf{V}\mathbf{U}^\top$

$$\mathbf{t}^* = \widetilde{\mathbf{q}} - \mathbf{R}^*\widetilde{\mathbf{p}}$$



Mutual calibration of two coordinate frames

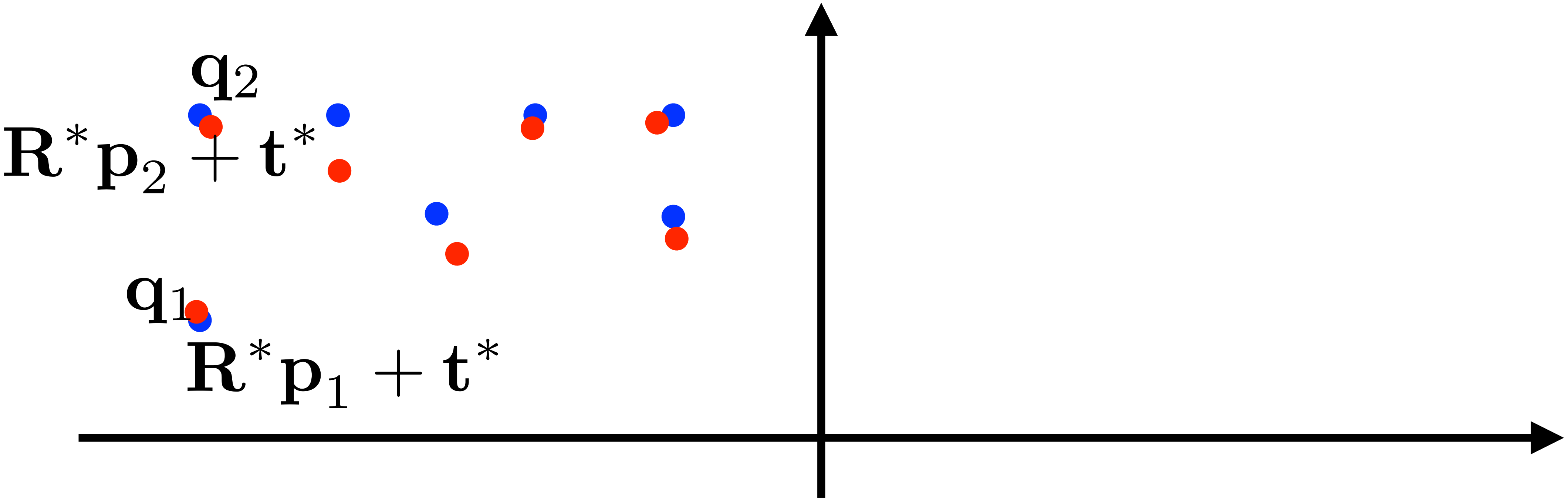
$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

Solution: estimate covariante matrix: $\mathbf{H} = \sum_i \mathbf{p}'_i \mathbf{q}'^{\top}_i$

find SVD decomposition: $\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$

estimate optimal rotation: $\mathbf{R}^* = \mathbf{V}\mathbf{U}^\top$

$$\mathbf{t}^* = \widetilde{\mathbf{q}} - \mathbf{R}^*\widetilde{\mathbf{p}}$$



Mutual calibration of two coordinate frames

(1) Record pointclouds and manually estimate 3D-3D correspondences

(2) Solve: $$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

Solution: $$\mathbf{R}^* = \mathbf{V}\mathbf{U}^\top$$
$$\mathbf{t}^* = \widetilde{\mathbf{q}} - \mathbf{R}^*\widetilde{\mathbf{p}}$$

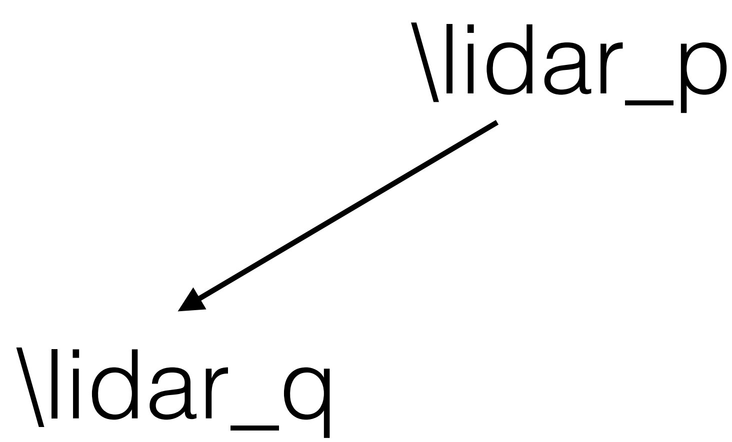
SVD in python:

```
U, S, V = np.linalg.svd(H, full_matrices=True)
```

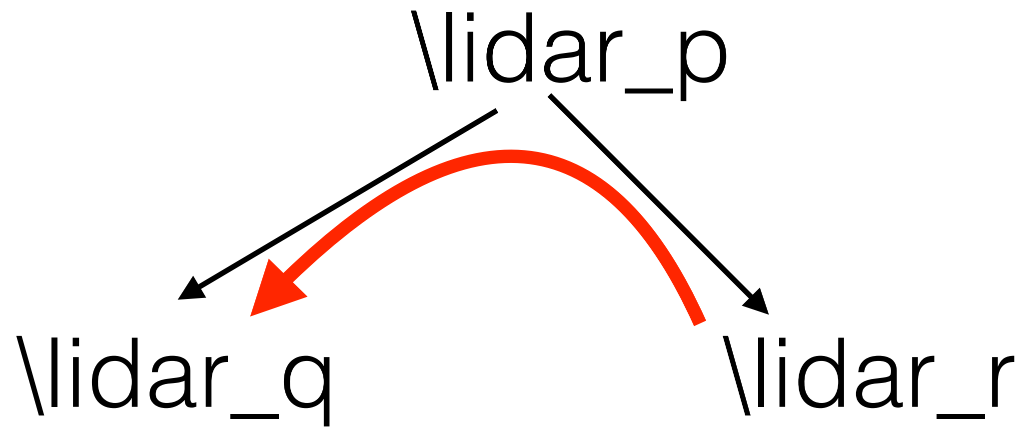
Broadcasting static transformation between two c.f. in ROS:

```
broadcaster = tf2_ros.StaticTransformBroadcaster()
transform = geometry_msgs.msg.TransformStamped()
# compute transform from 3D-3D correspondences
broadcaster.sendTransform(transform)
```

Mutual calibration of two coordinate frames



Mutual calibration of two coordinate frames



Mutual calibration of two coordinate frames

- Application in robotics for SLAM.
- Application in Computer graphics for alignment of 3D models

Proof [Arun-TPAMI-87]

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2 =$$

Proof [Arun-TPAMI-87]

$$\mathbf{R}^*, \mathbf{t}^* = \arg\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2 =$$

$$= \arg\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}(\mathbf{p}_i' + \widetilde{\mathbf{p}}) + \mathbf{t} - \mathbf{q}_i' - \widetilde{\mathbf{q}}\|_2^2 =$$

Proof [Arun-TPAMI-87]

$$\mathbf{R}^*, \mathbf{t}^* = \arg\min_{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2 =$$

$$= \arg\min_{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3} \sum_i \|\mathbf{R}(\mathbf{p}_i' + \widetilde{\mathbf{p}}) + \mathbf{t} - \mathbf{q}_i' - \widetilde{\mathbf{q}}\|_2^2 =$$

$$= \arg\min_{\mathbf{R}\in SO(3), \mathbf{t}\in\mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i' - \mathbf{q}_i' + \underbrace{\mathbf{R}\widetilde{\mathbf{p}} + \mathbf{t} - \widetilde{\mathbf{q}}}_{\mathbf{t}'}\|_2^2 =$$

Proof [Arun-TPAMI-87]

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2 =$$

$$= \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}(\mathbf{p}'_i + \widetilde{\mathbf{p}}) + \mathbf{t} - \mathbf{q}'_i - \widetilde{\mathbf{q}}\|_2^2 =$$

$$= \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i + \underbrace{\mathbf{R}\widetilde{\mathbf{p}} + \mathbf{t} - \widetilde{\mathbf{q}}}_{\mathbf{t}'}\|_2^2 =$$

$$= \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i (\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i + \mathbf{t}')^\top (\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i + \mathbf{t}') =$$

Proof [Arun-TPAMI-87]

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R}\in SO(3),\mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2 =$$

$$= \underset{\mathbf{R}\in SO(3),\mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}(\mathbf{p}'_i + \widetilde{\mathbf{p}}) + \mathbf{t} - \mathbf{q}'_i - \widetilde{\mathbf{q}}\|_2^2 =$$

$$= \underset{\mathbf{R}\in SO(3),\mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i + \underbrace{\mathbf{R}\widetilde{\mathbf{p}} + \mathbf{t} - \widetilde{\mathbf{q}}}_{\mathbf{t}'}\|_2^2 =$$

$$= \underset{\mathbf{R}\in SO(3),\mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i (\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i + \mathbf{t}')^\top (\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i + \mathbf{t}') =$$

$$= \underset{\mathbf{R}\in SO(3),\mathbf{t}\in\mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i\|_2^2 + \underbrace{\sum_i 2(\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i)\mathbf{t}'}_{=0} + \|\mathbf{t}'\|_2^2 =$$

Proof [Arun-TPAMI-87]

$$\mathbf{R}^*, \mathbf{t}^* = \arg\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2 =$$

$$= \arg\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}(\mathbf{p}_i' + \widetilde{\mathbf{p}}) + \mathbf{t} - \mathbf{q}_i' - \widetilde{\mathbf{q}}\|_2^2 =$$

$$= \arg\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i' - \mathbf{q}_i' + \underbrace{\mathbf{R}\widetilde{\mathbf{p}} + \mathbf{t} - \widetilde{\mathbf{q}}}_{\mathbf{t}'}\|_2^2 =$$

$$= \arg\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i (\mathbf{R}\mathbf{p}_i' - \mathbf{q}_i' + \mathbf{t}')^\top (\mathbf{R}\mathbf{p}_i' - \mathbf{q}_i' + \mathbf{t}') =$$

$$= \arg\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i' - \mathbf{q}_i'\|_2^2 + \underbrace{\sum_i 2(\mathbf{R}\mathbf{p}_i' - \mathbf{q}_i')\mathbf{t}'}_{=0} + \|\mathbf{t}'\|_2^2 =$$

$$= \arg\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i' - \mathbf{q}_i'\|_2^2 + \|\mathbf{t}'\|_2^2$$

we can reach second term zero by $\mathbf{t} = \widetilde{\mathbf{q}} - \mathbf{R}\widetilde{\mathbf{p}} = \mathbf{t}^*$

Proof [Arun-TPAMI-87]

$$= \arg\min_{\mathbf{R}\in SO(3),\mathbf{t}\in\mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i' - \mathbf{q}_i'\|_2^2 + \|\mathbf{t}'\|_2^2$$

we can reach second term zero by $\mathbf{t} = \widetilde{\mathbf{q}} - \mathbf{R}\widetilde{\mathbf{p}} = \mathbf{t}^*$

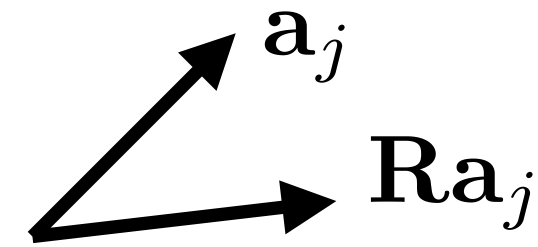
$$\arg\min_{\mathbf{R}\in SO(3)} \sum_i \|\mathbf{R}\mathbf{p}_i' - \mathbf{q}_i'\|_2^2 = \arg\max_{\mathbf{R}\in SO(3)} \sum_i \mathbf{q}_i'^\top \mathbf{R}\mathbf{p}_i' =$$

$$= \arg\max_{\mathbf{R}\in SO(3)} \ \text{trace}\{\mathbf{R}\mathbf{H}\} = \mathbf{V}\mathbf{U}^\top$$

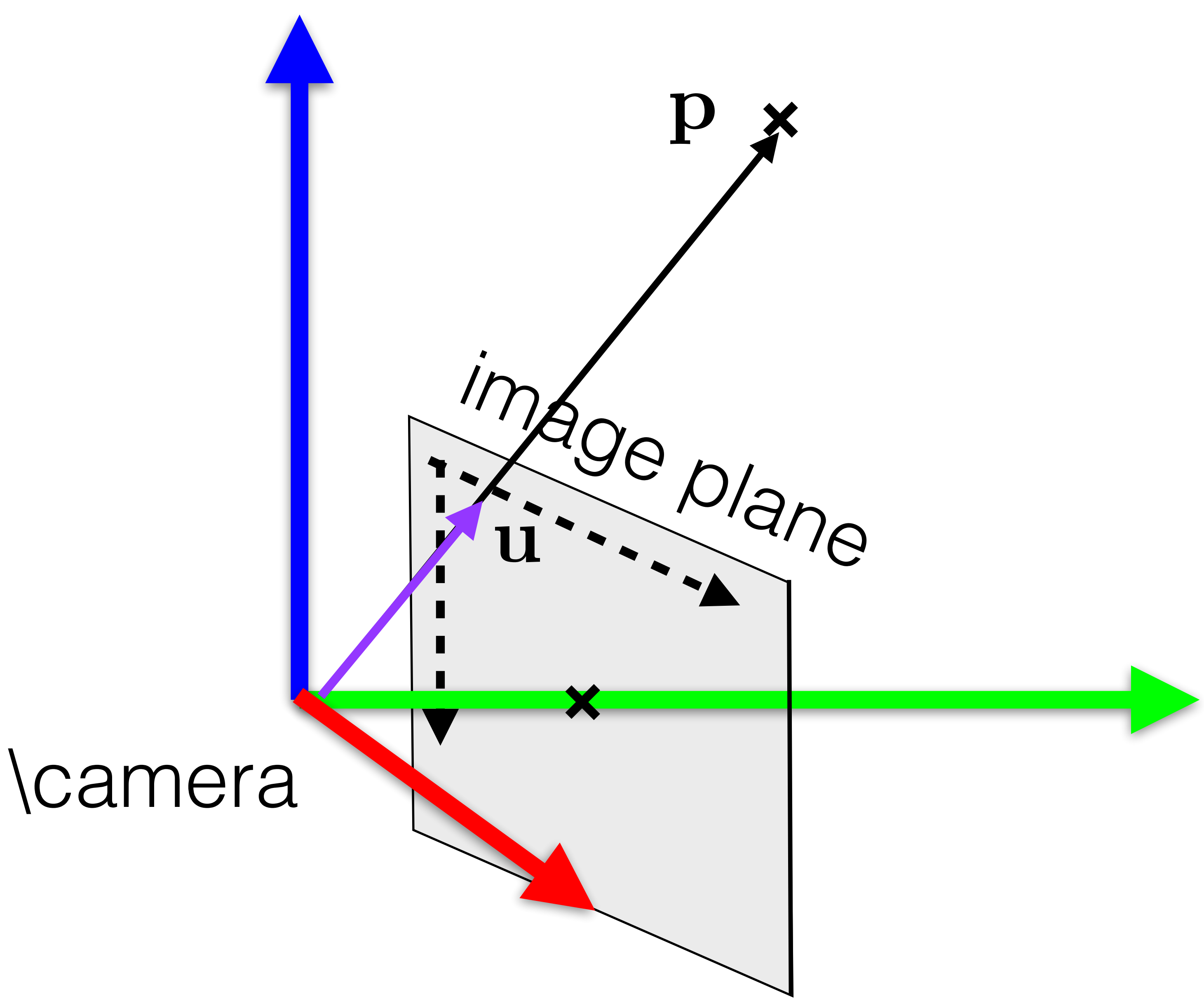
because when you substitute this rotation into criterion you will obtain value higher then for any other rotation

$$\text{trace}\{\mathbf{R}^*\mathbf{H}\} = \text{trace}\{\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{S}\mathbf{V}^\top\} = \text{trace}\{\mathbf{V}\mathbf{S}\mathbf{V}^\top\} =$$

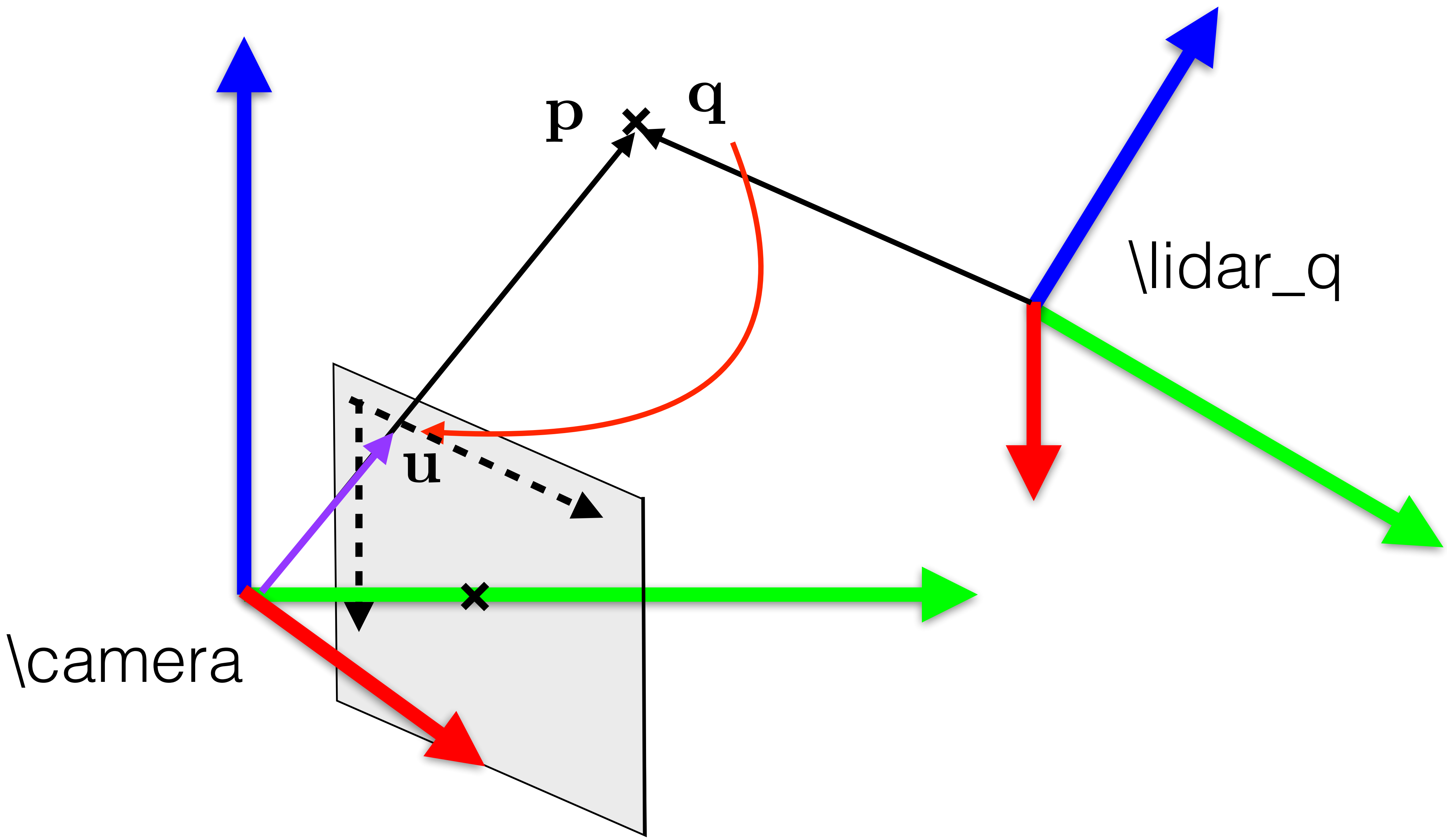
$$= \text{trace}\{\underbrace{(\mathbf{V}\sqrt{\mathbf{S}})}_{\mathbf{A}}\underbrace{(\sqrt{\mathbf{S}}\mathbf{V})^\top}_{\mathbf{A}^\top}\} = \sum_j \mathbf{a}_j^\top \mathbf{a}_j \geq \text{trace}(\mathbf{R}\mathbf{A})\mathbf{A}^\top$$



Camera captures RGB colors of objects projected on image plane (light sensitive sensor which provide images).



Let us have one lidar and one camera

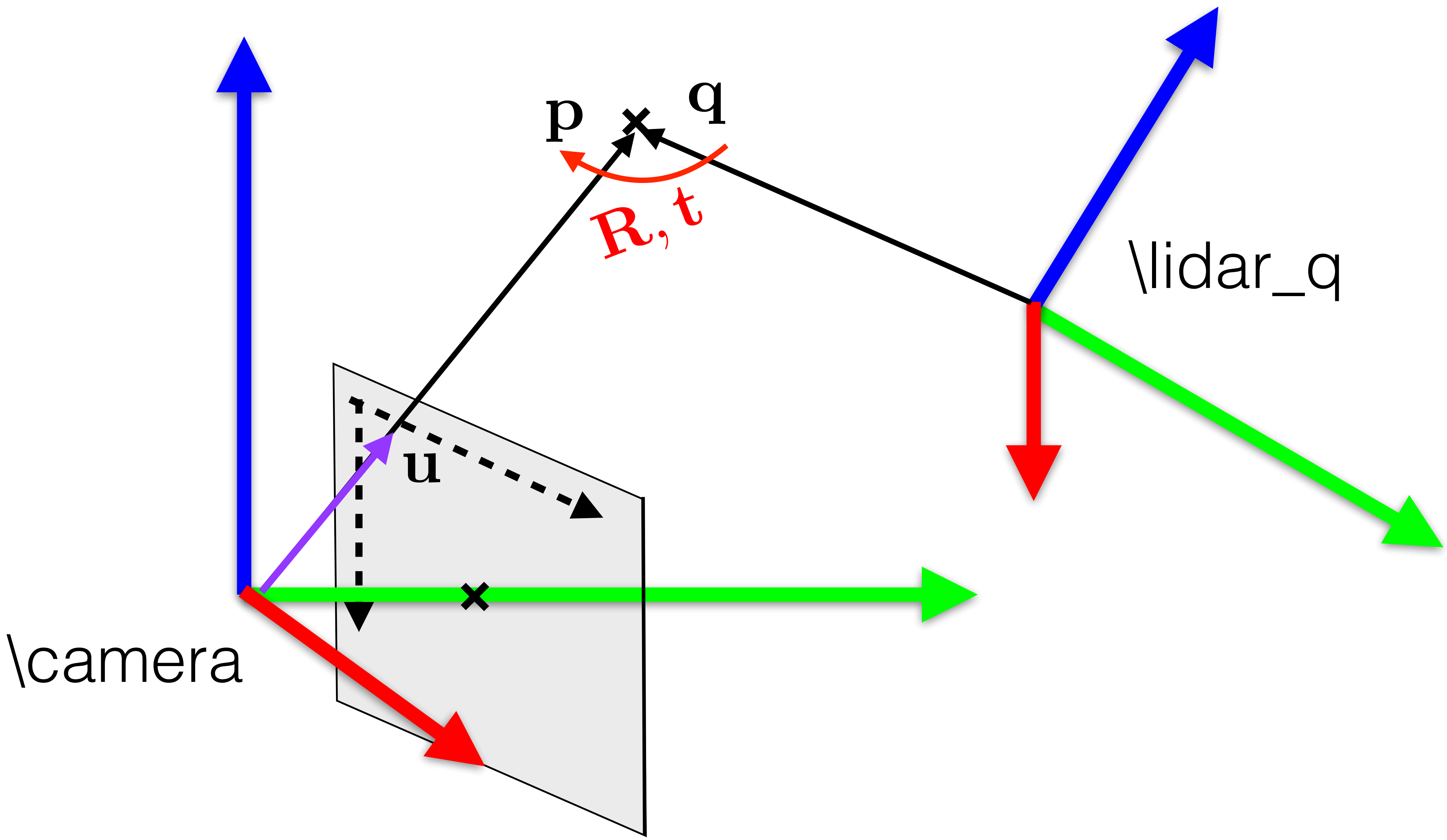


Camera

Projection from lidar to image plane consists of two steps:

- transform of 3D point in \lidar_q $\mathbf{q} \in \mathcal{R}^3$ to \camera $\mathbf{p} \in \mathcal{R}^3$

$$\mathbf{p} = \mathbf{R}\mathbf{q} + \mathbf{t}$$

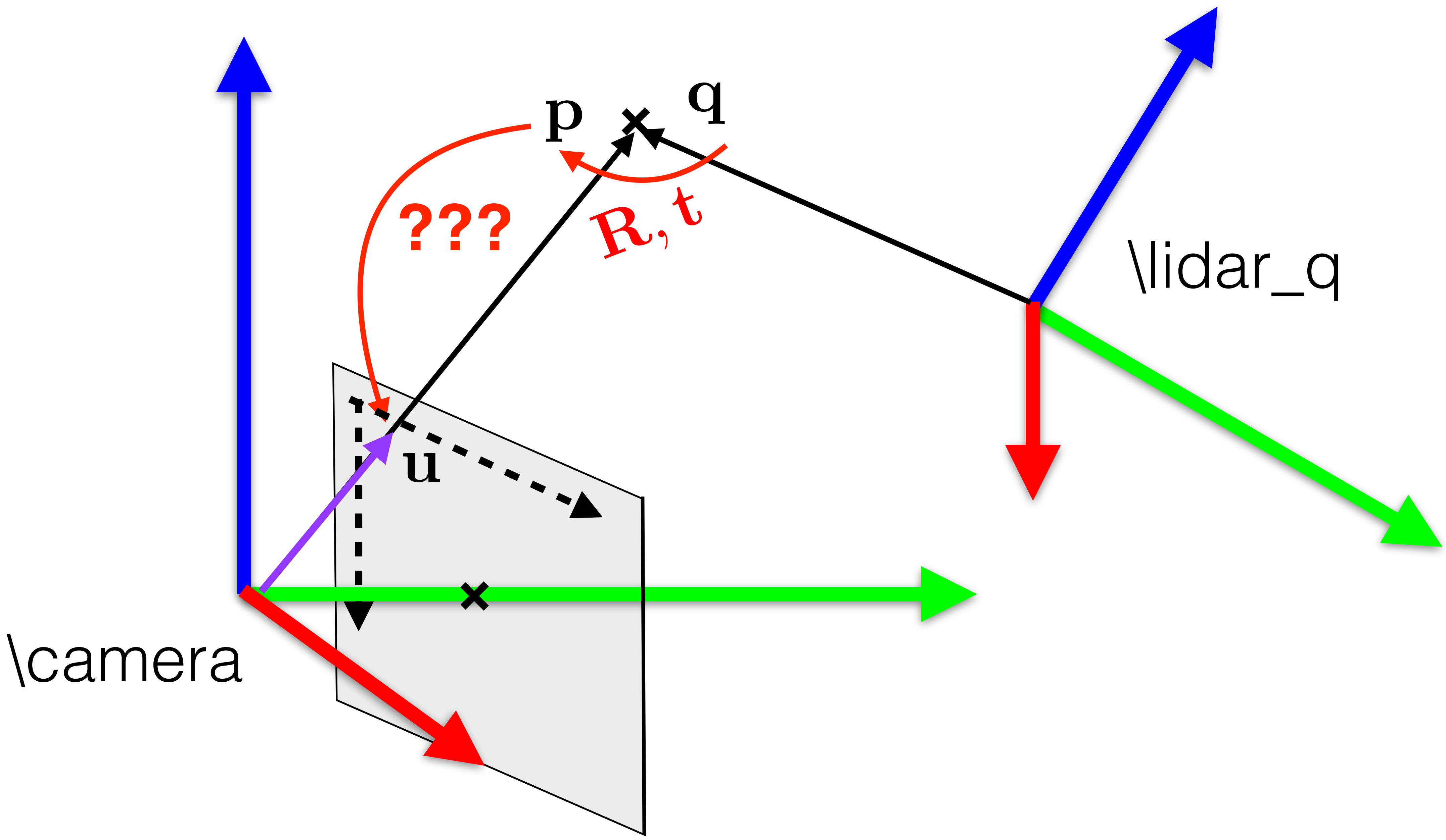


Camera

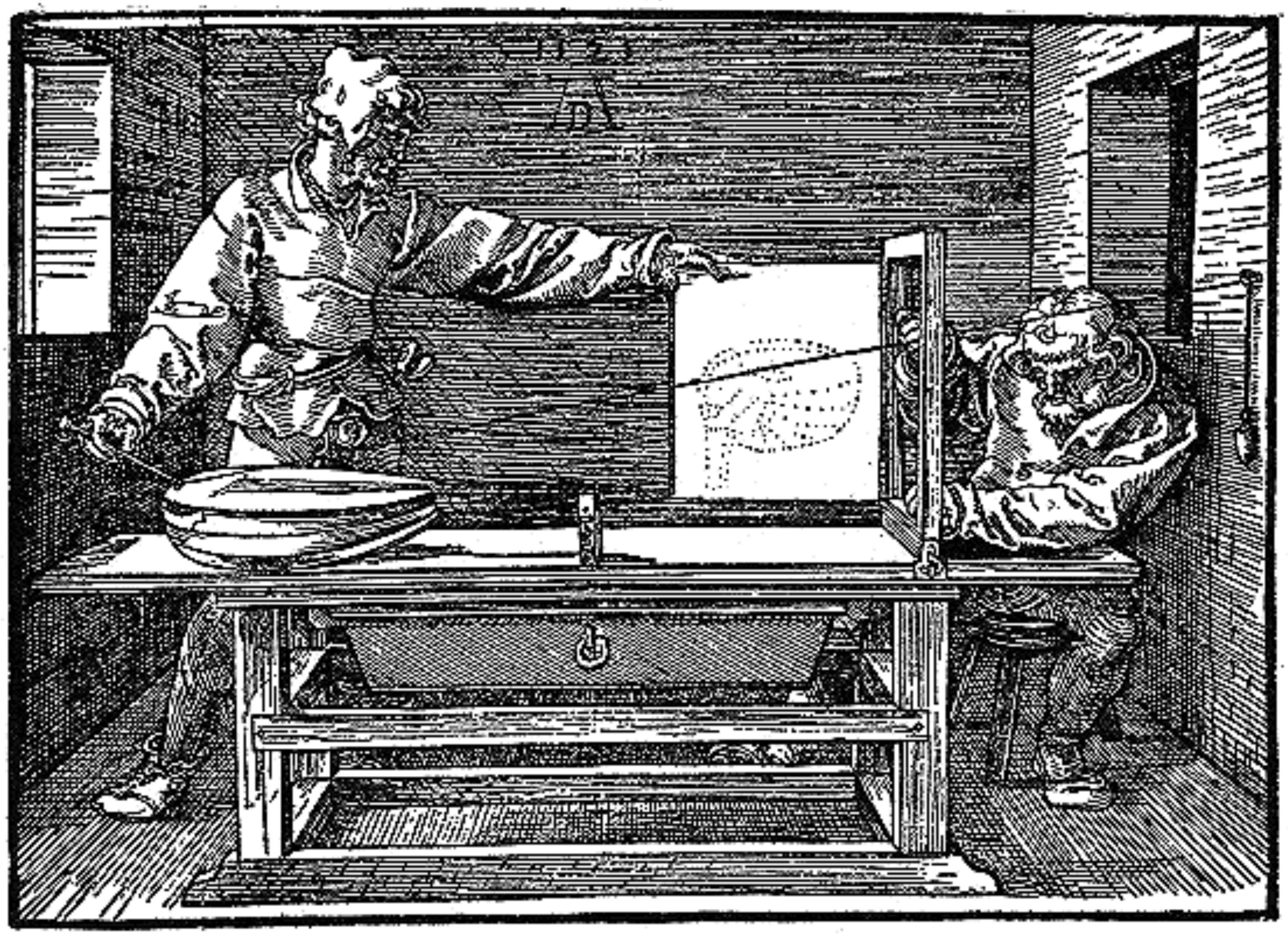
Projection from lidar to image plane consists of two steps:

- transform of 3D point in \lidar_q $\mathbf{q} \in \mathcal{R}^3$ to \camera $\mathbf{p} \in \mathcal{R}^3$
- projection of 3D point in \camera on image plane $\mathbf{u} \in \mathcal{R}^2$

$$\mathbf{p} = \mathbf{R}\mathbf{q} + \mathbf{t}$$



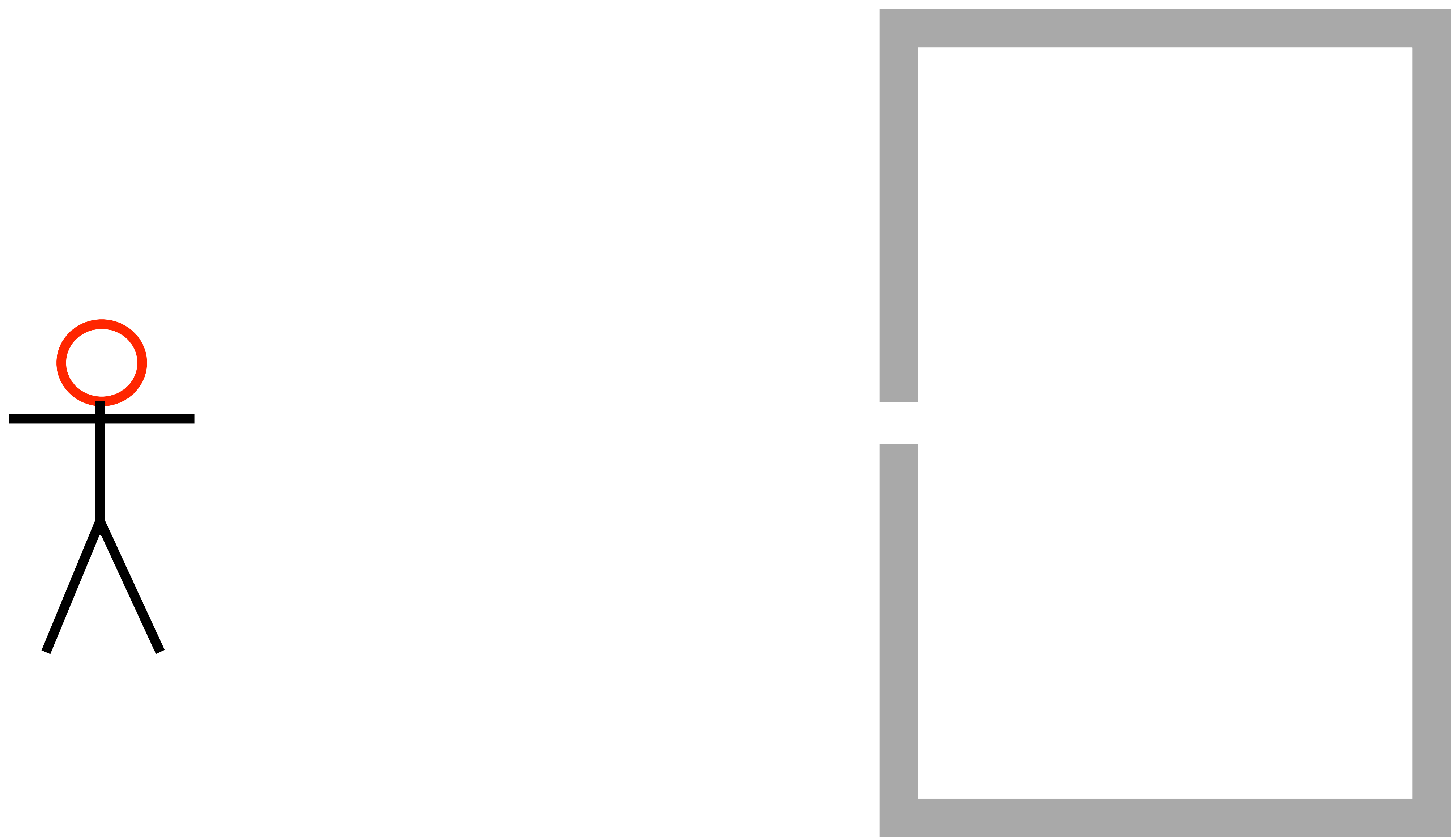
Projection of 3D point in \camera on image plane



Albrecht Durer (1545), Hitachi Viewmuseum

Projection of 3D point in \camera on image plane

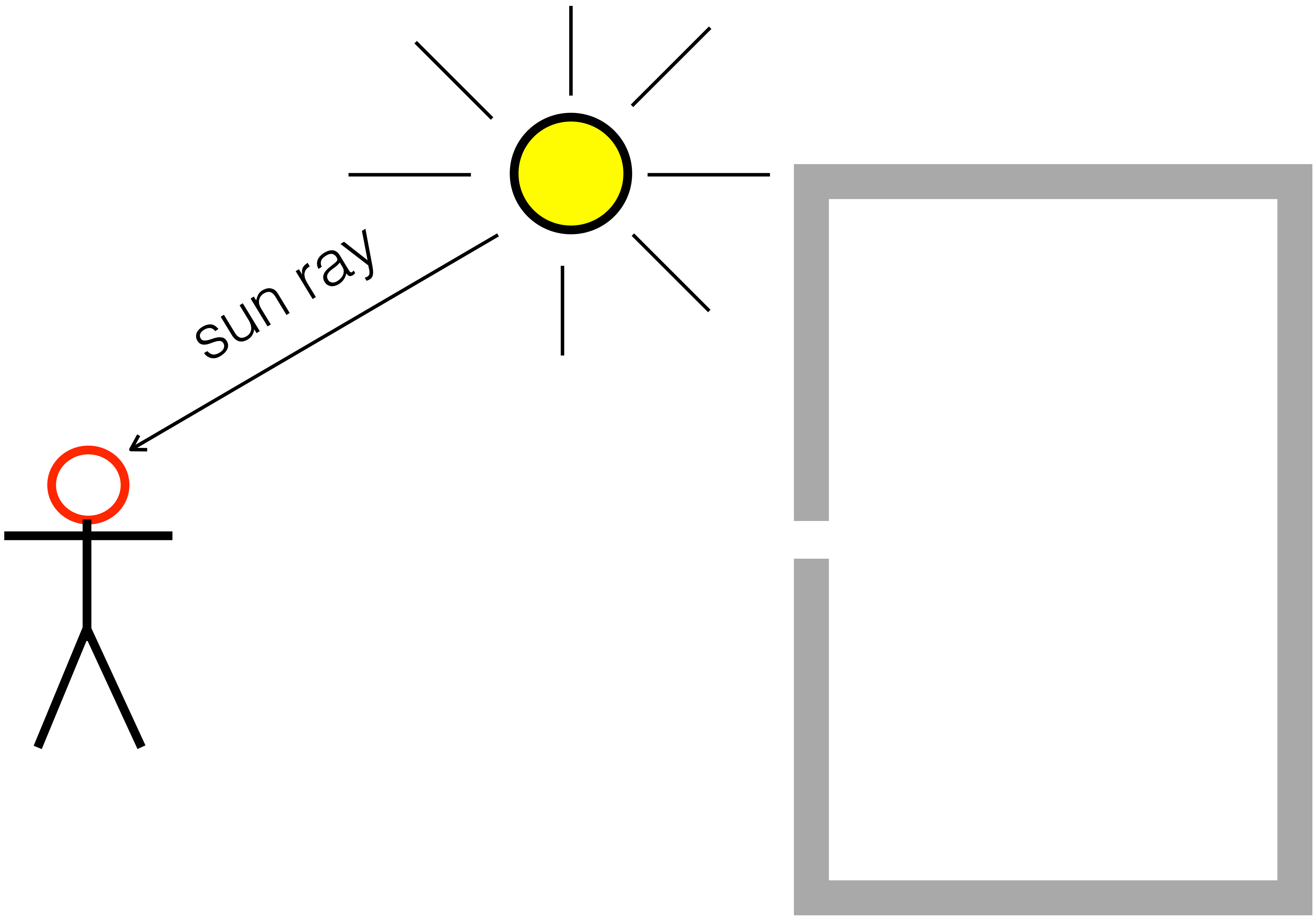
Projection of 3D point in \camera on image plane



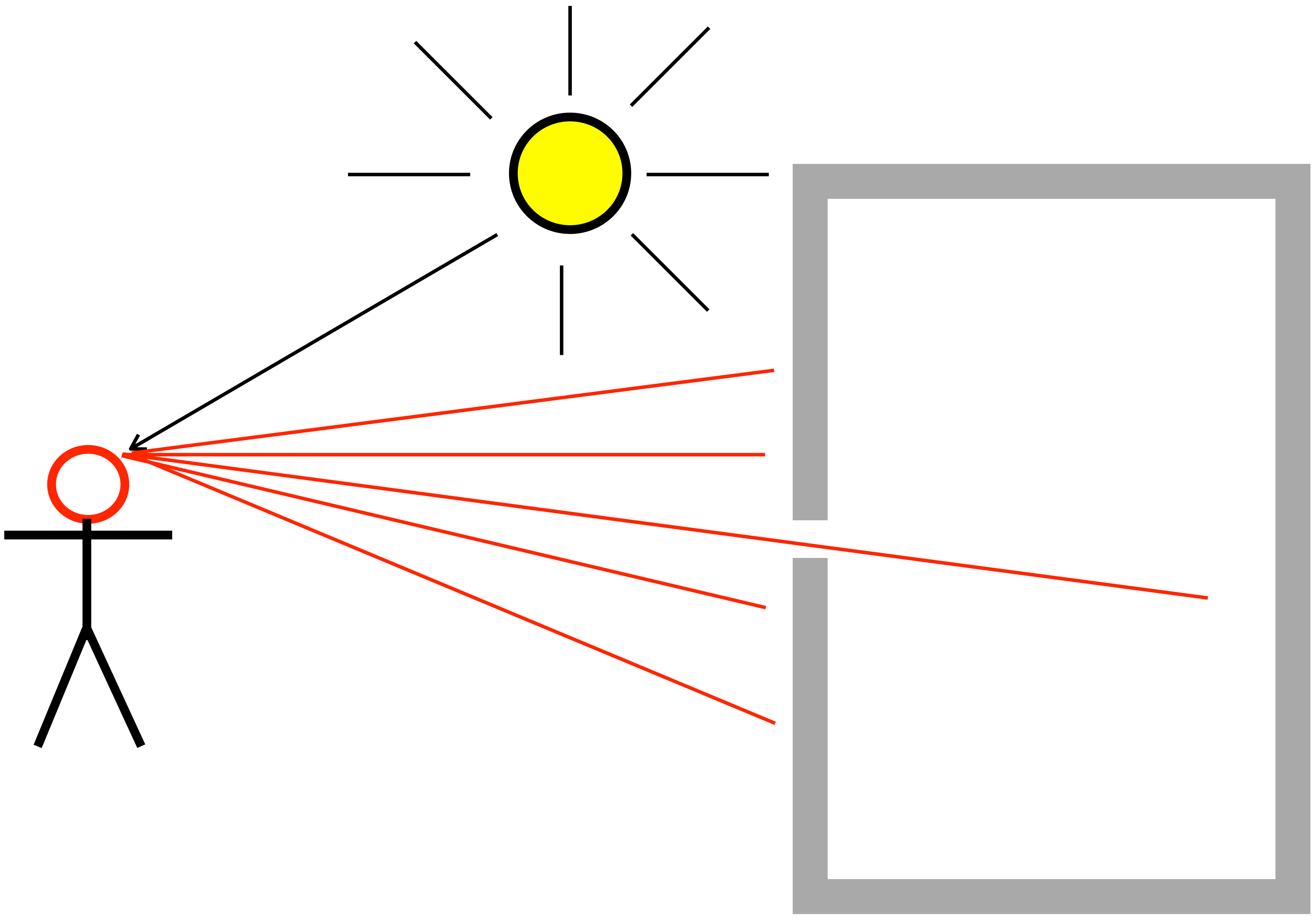
Object in front of camera

Pinhole camera model

Projection of 3D point in \camera on image plane

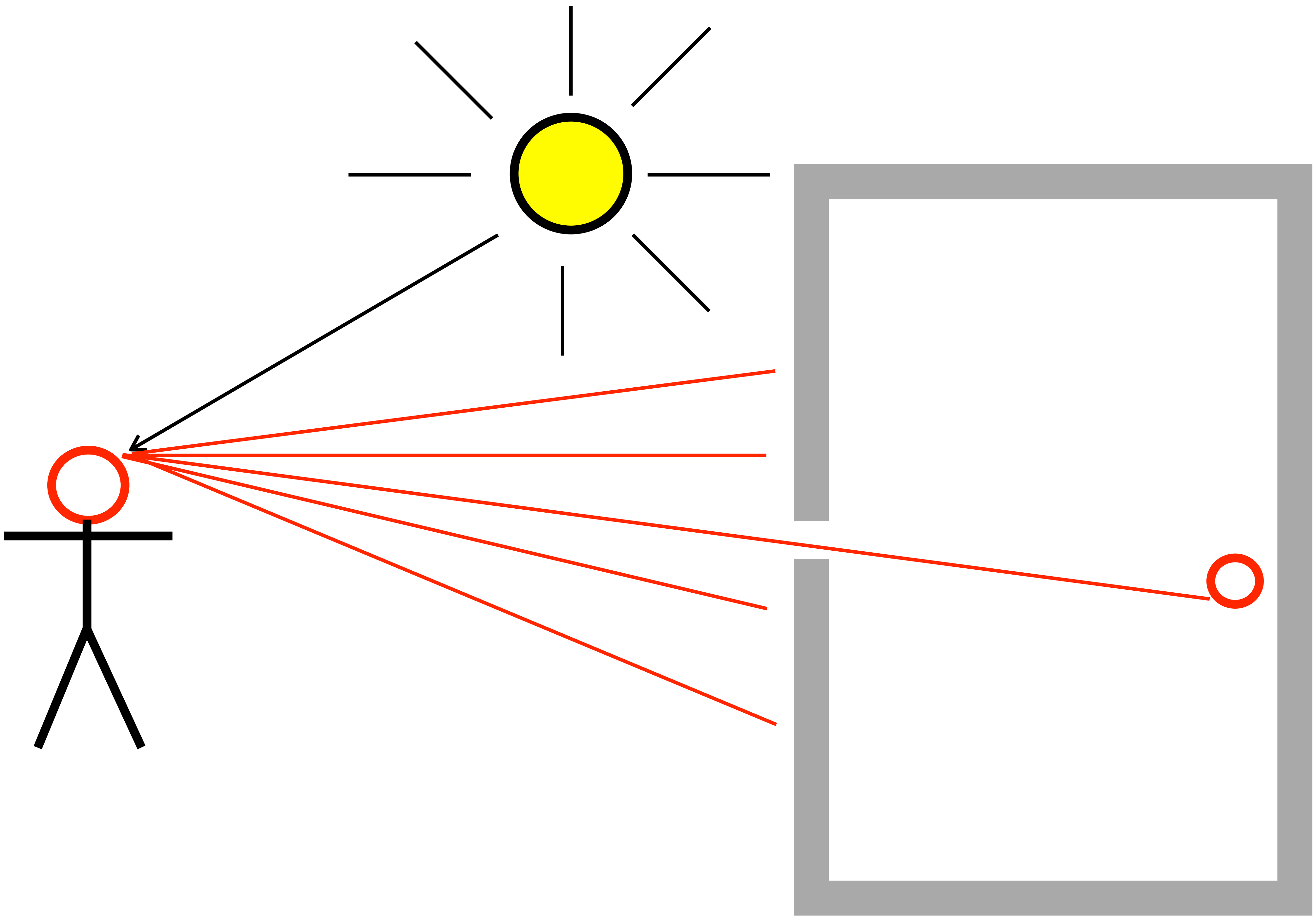


Projection of 3D point in \camera on image plane



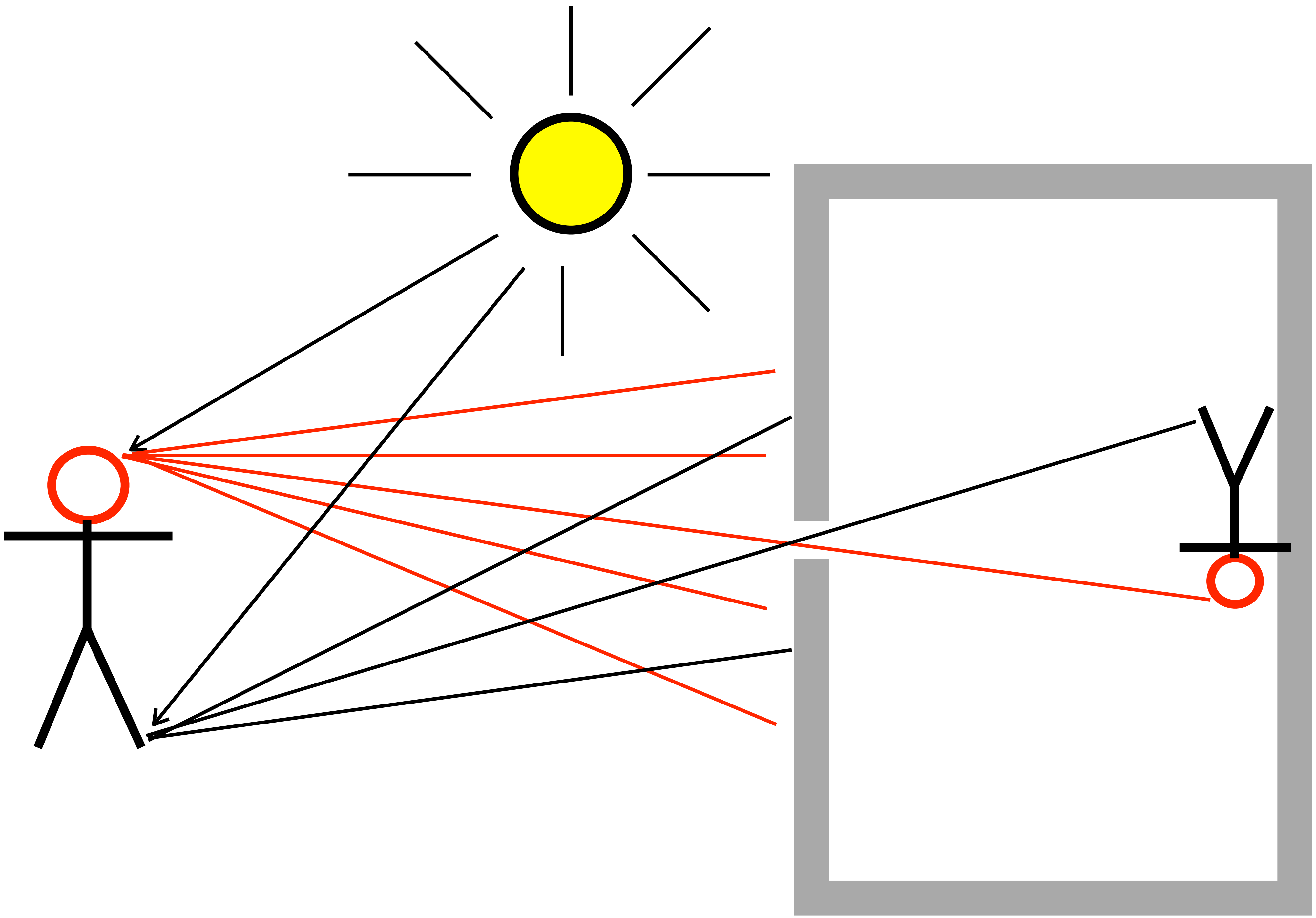
Sun ray is reflected from Lambertian surface in hemisphere

Projection of 3D point in \camera on image plane



Reflected ray (red) forms inverted image of the object

Projection of 3D point in \camera on image plane

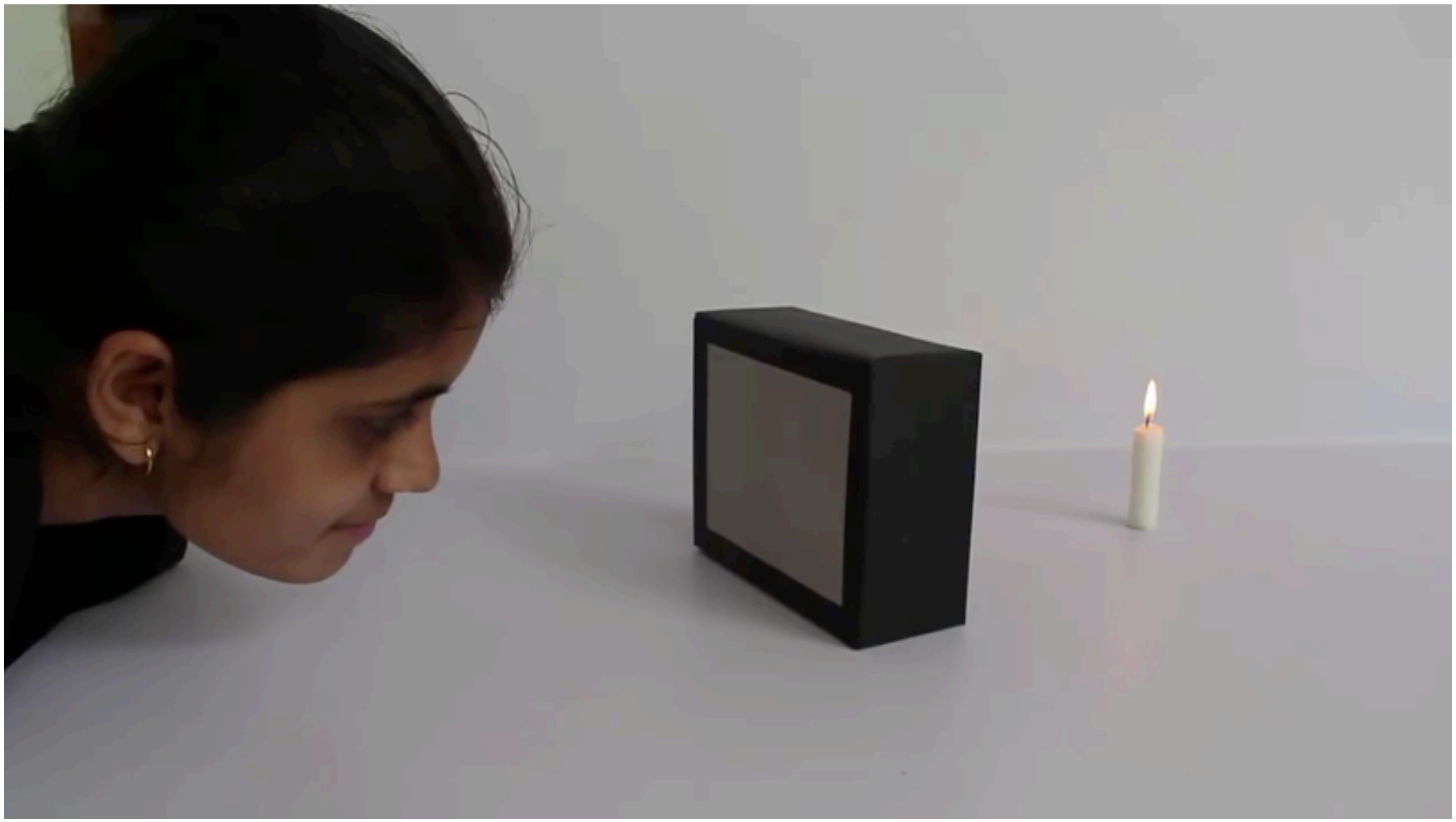


Projection of 3D point in \camera on image plane

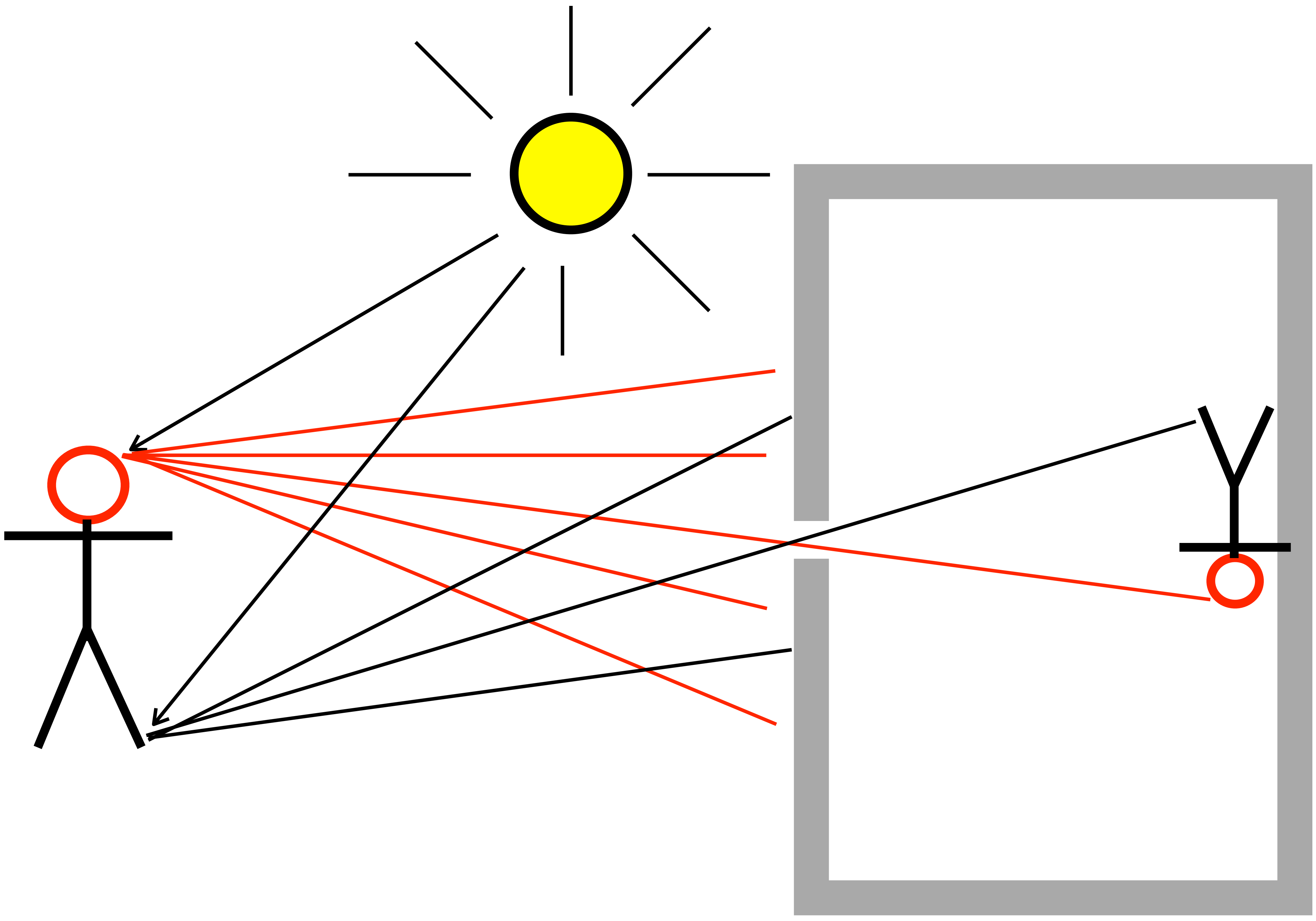
Projection of 3D point in \camera on image plane
Pinhole camera model

Projection of 3D point in \camera on image plane

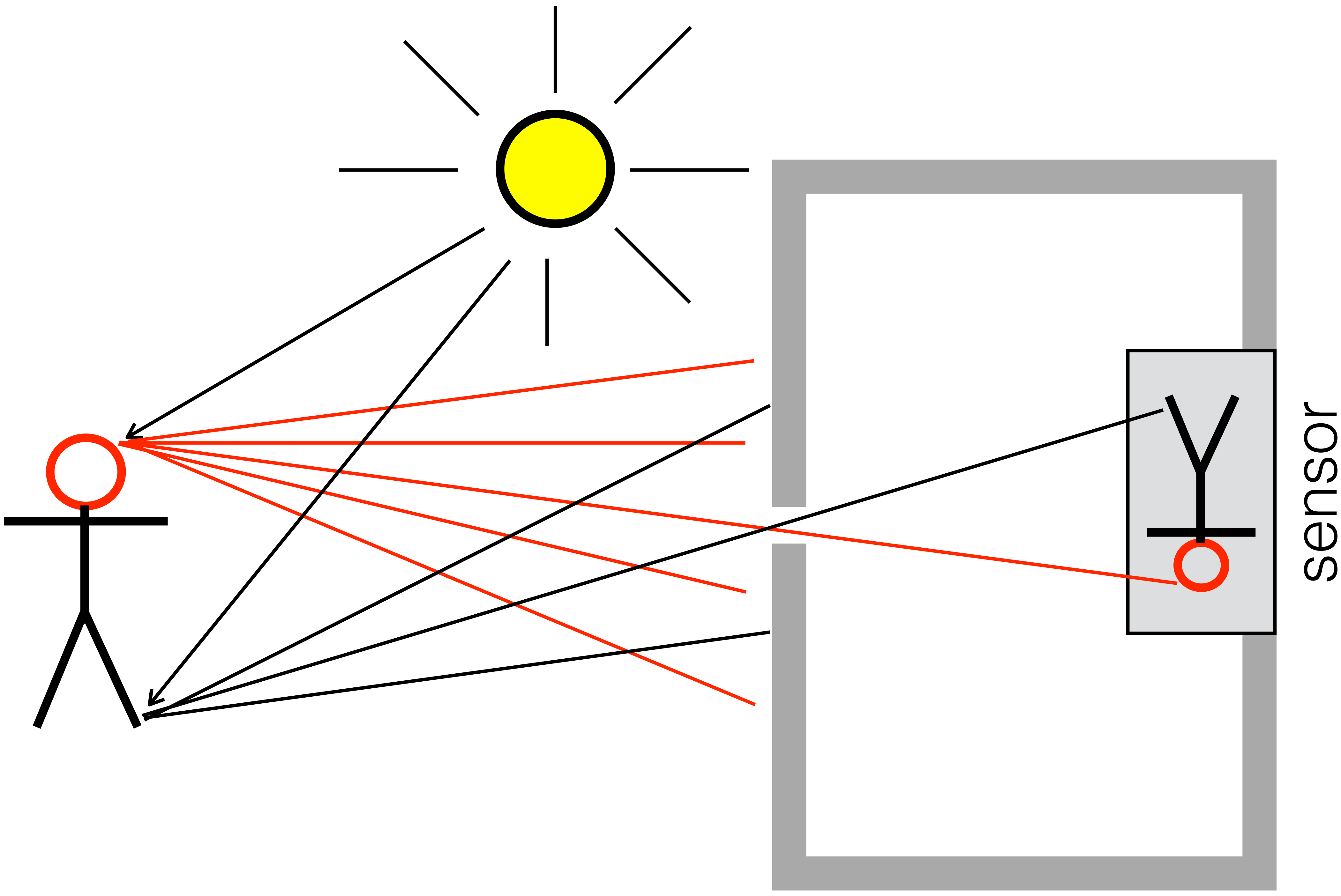
Pinhole camera model



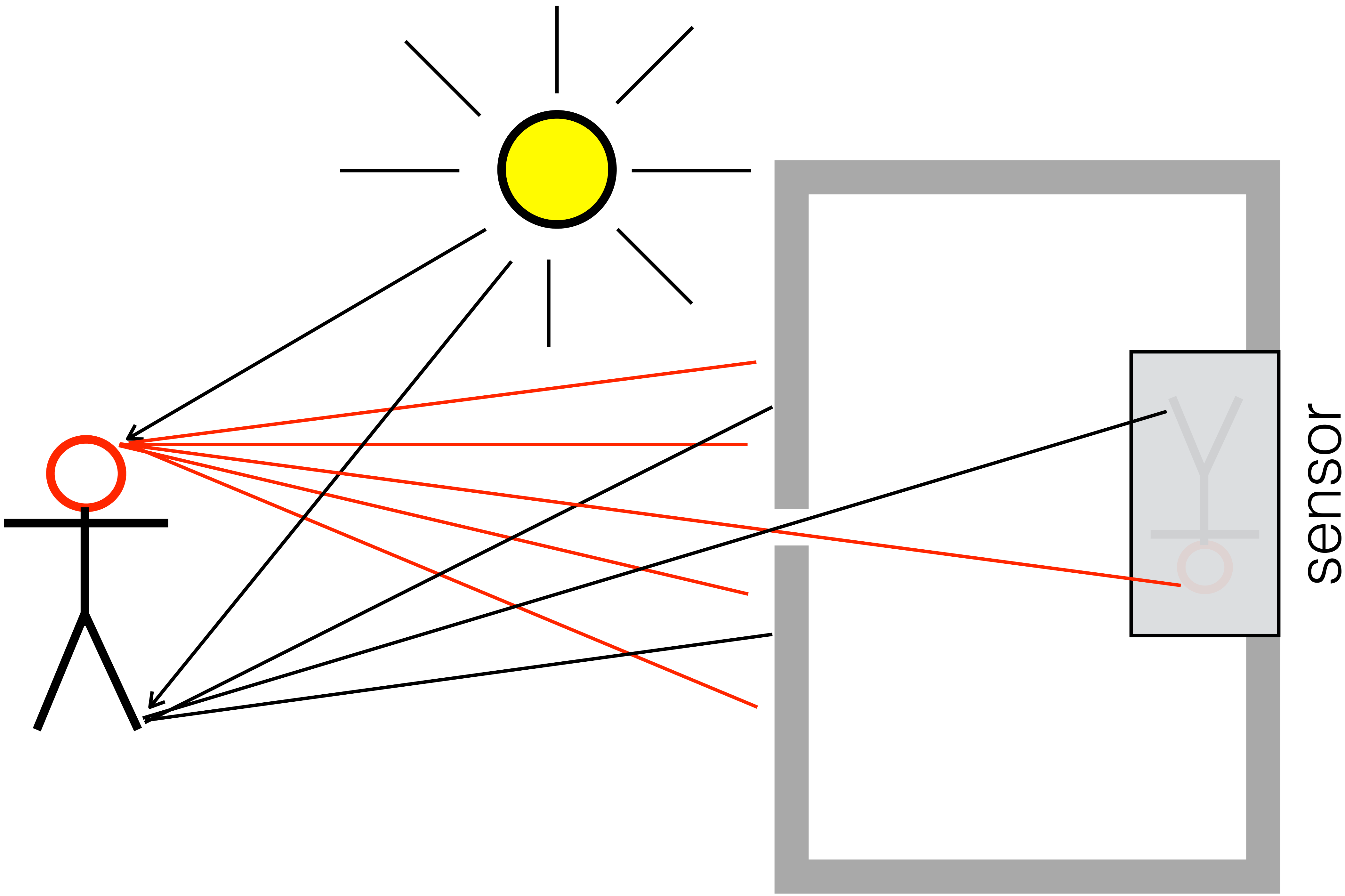
Projection of 3D point in \camera on image plane



Projection of 3D point in \camera on image plane

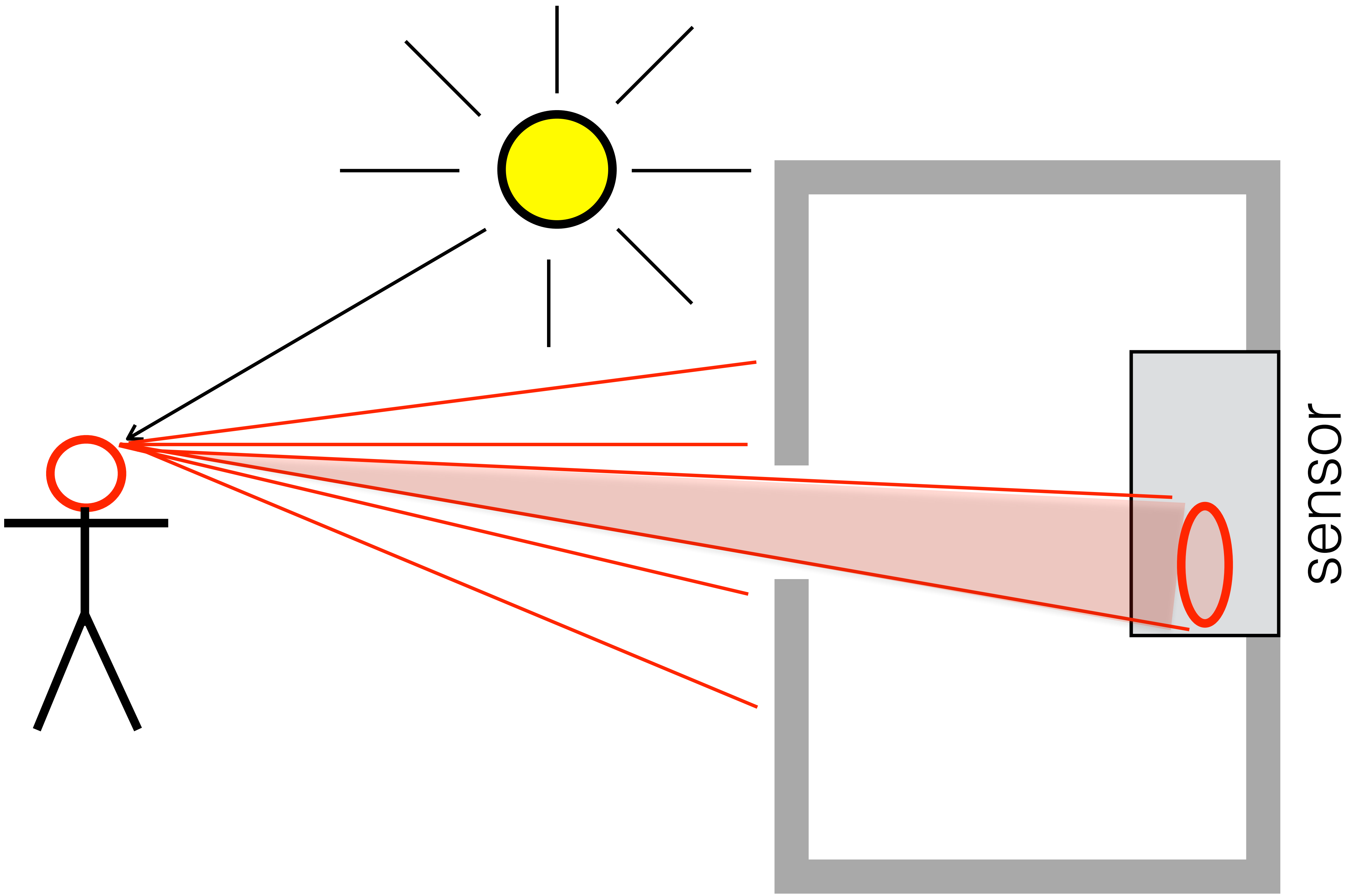


Projection of 3D point in \camera on image plane



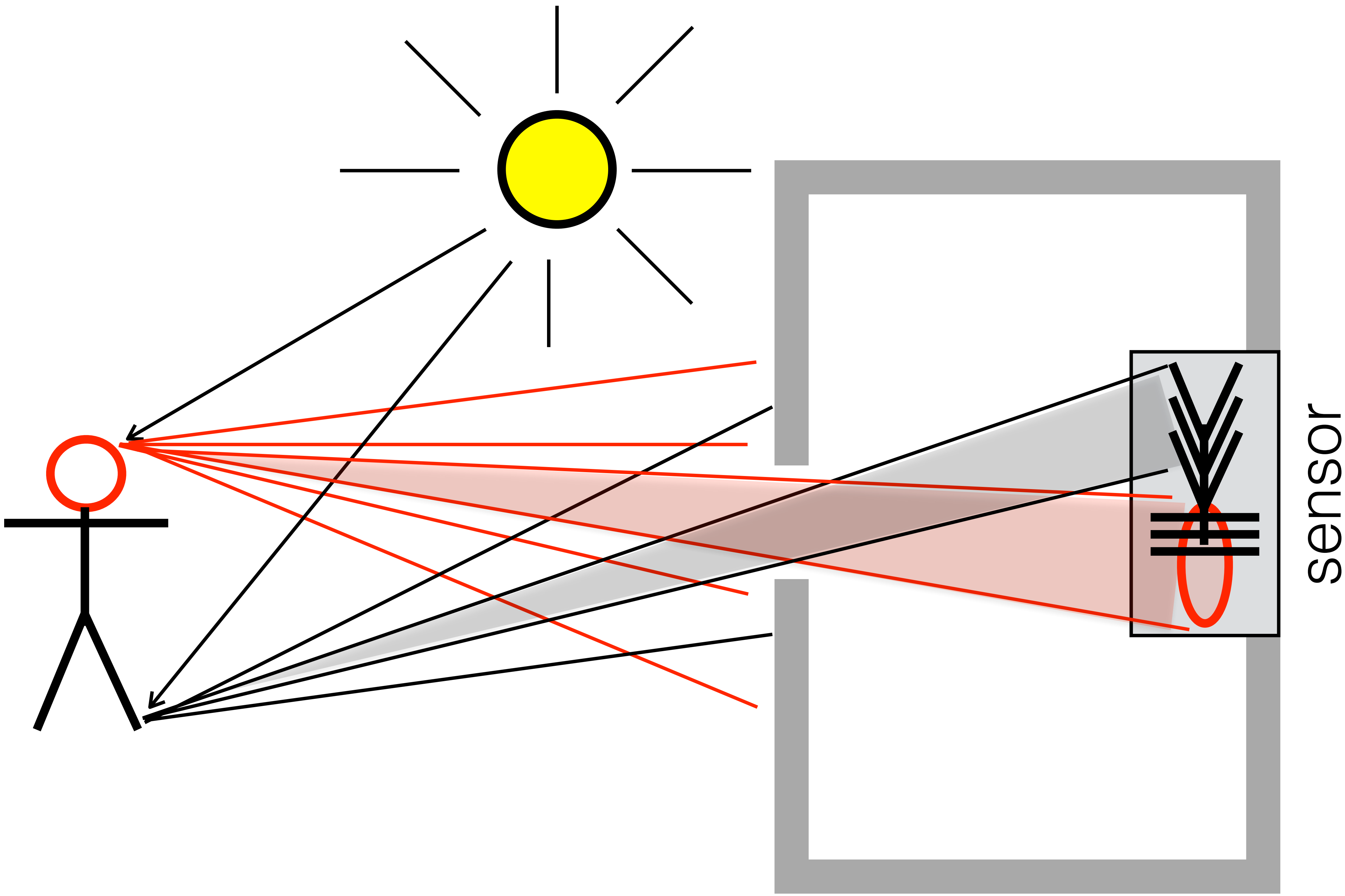
Light energy from rays traversed through pinhole is small

Projection of 3D point in \camera on image plane



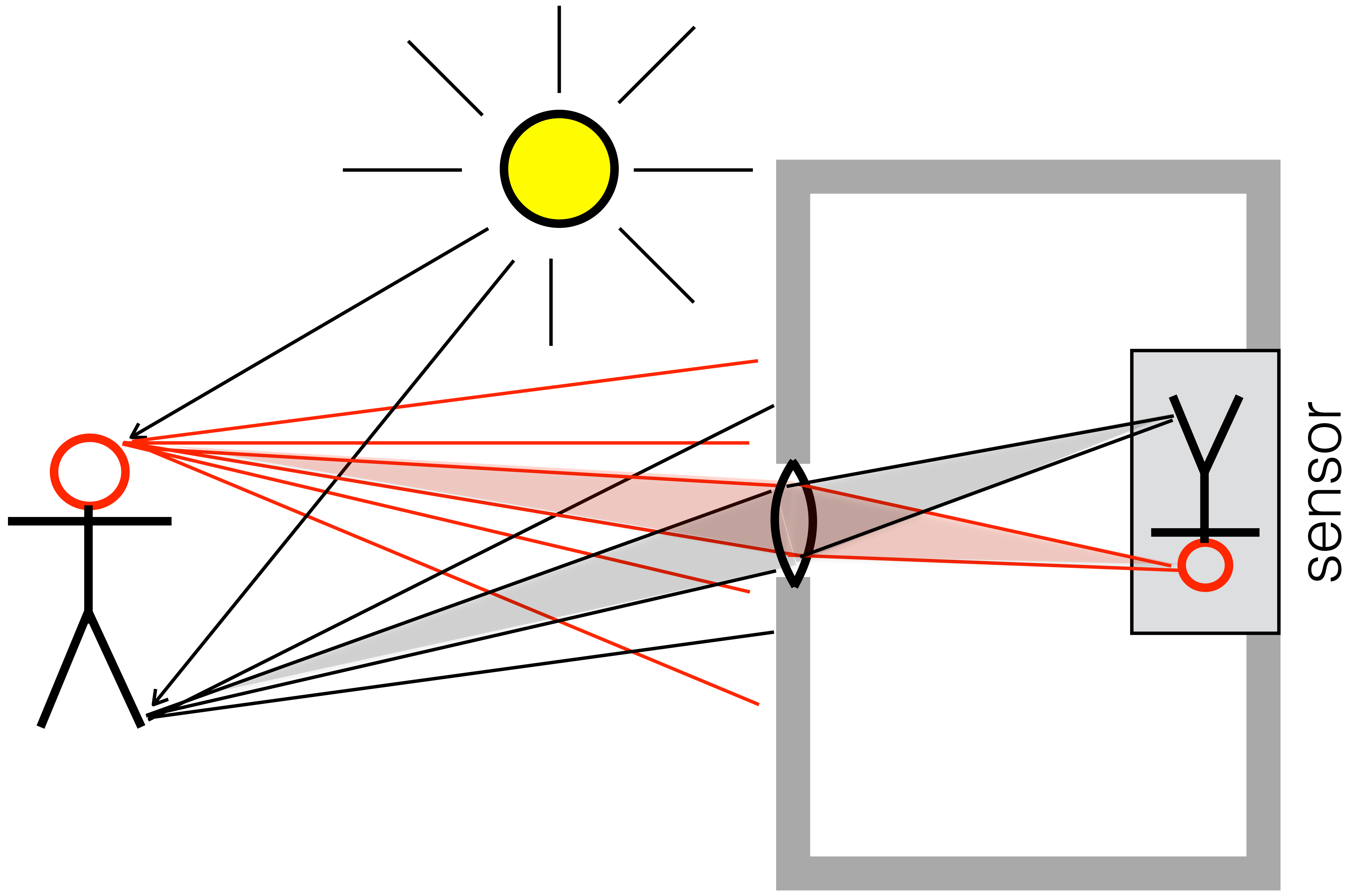
Increasing hole size yields more energy but blurs image

Projection of 3D point in \camera on image plane



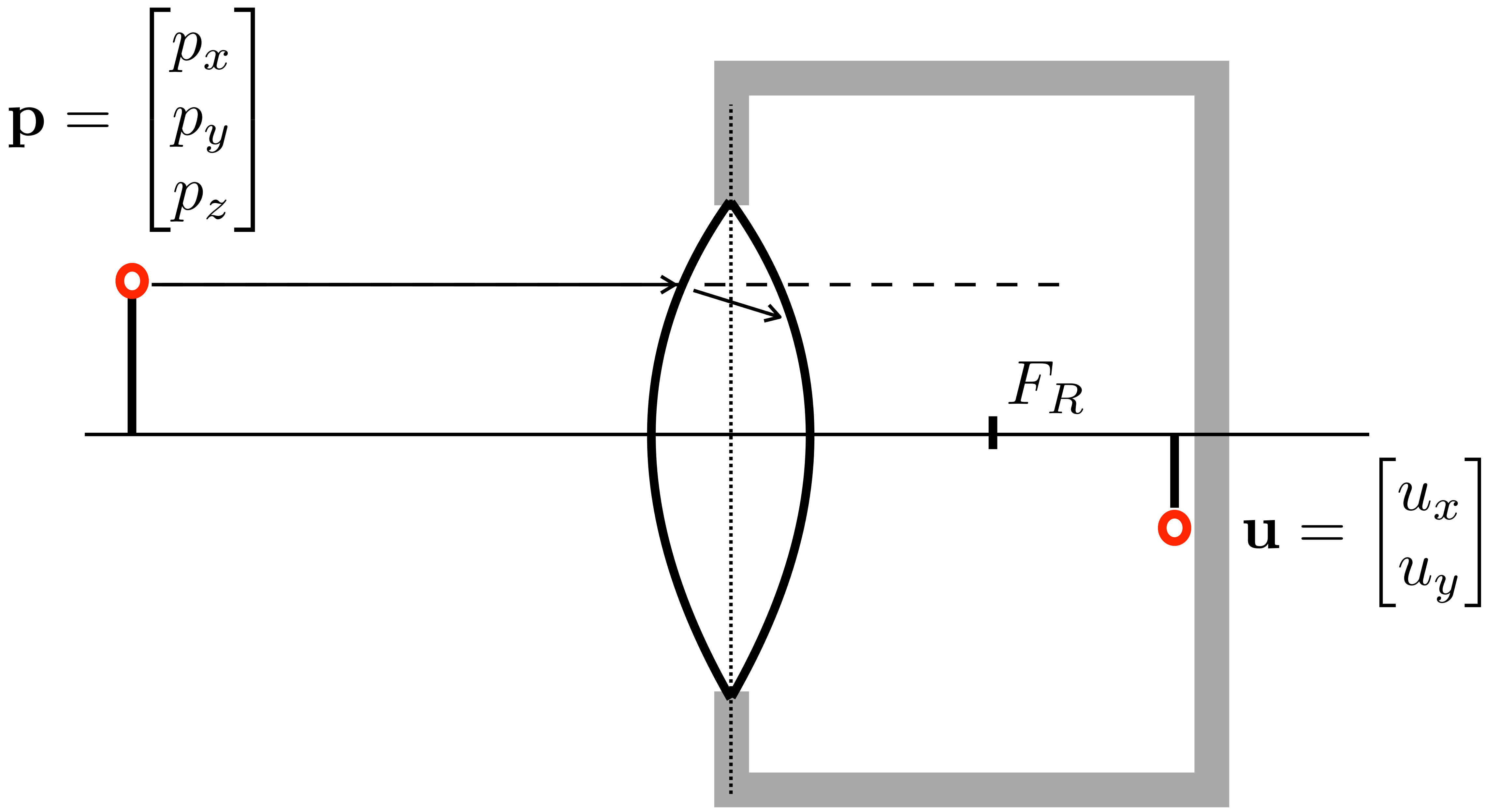
Increasing hole size yields more energy but blurs image

Projection of 3D point in \camera on image plane



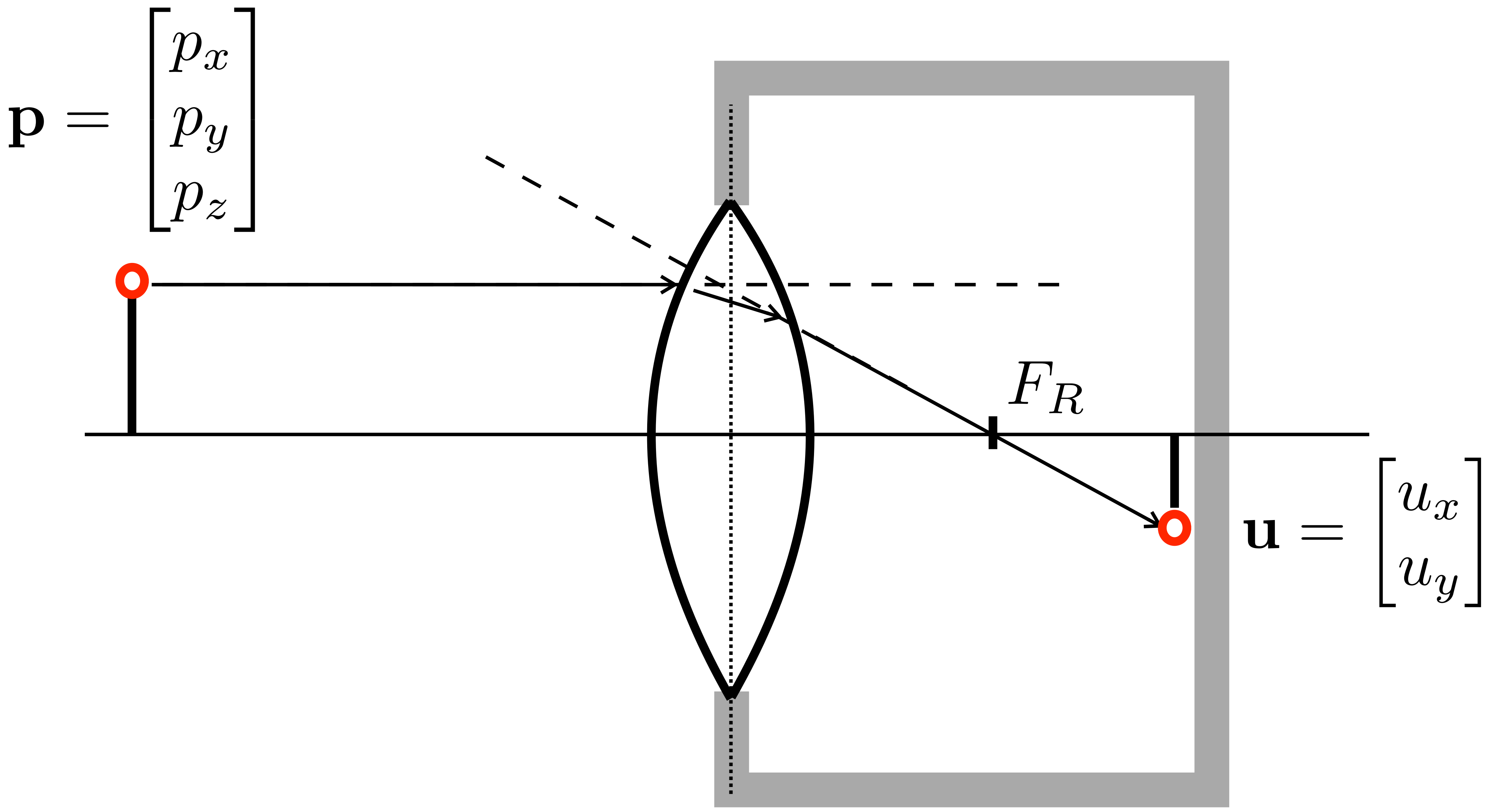
Lens focus cone of light rays in a single point

Projection of 3D point in \camera on image plane

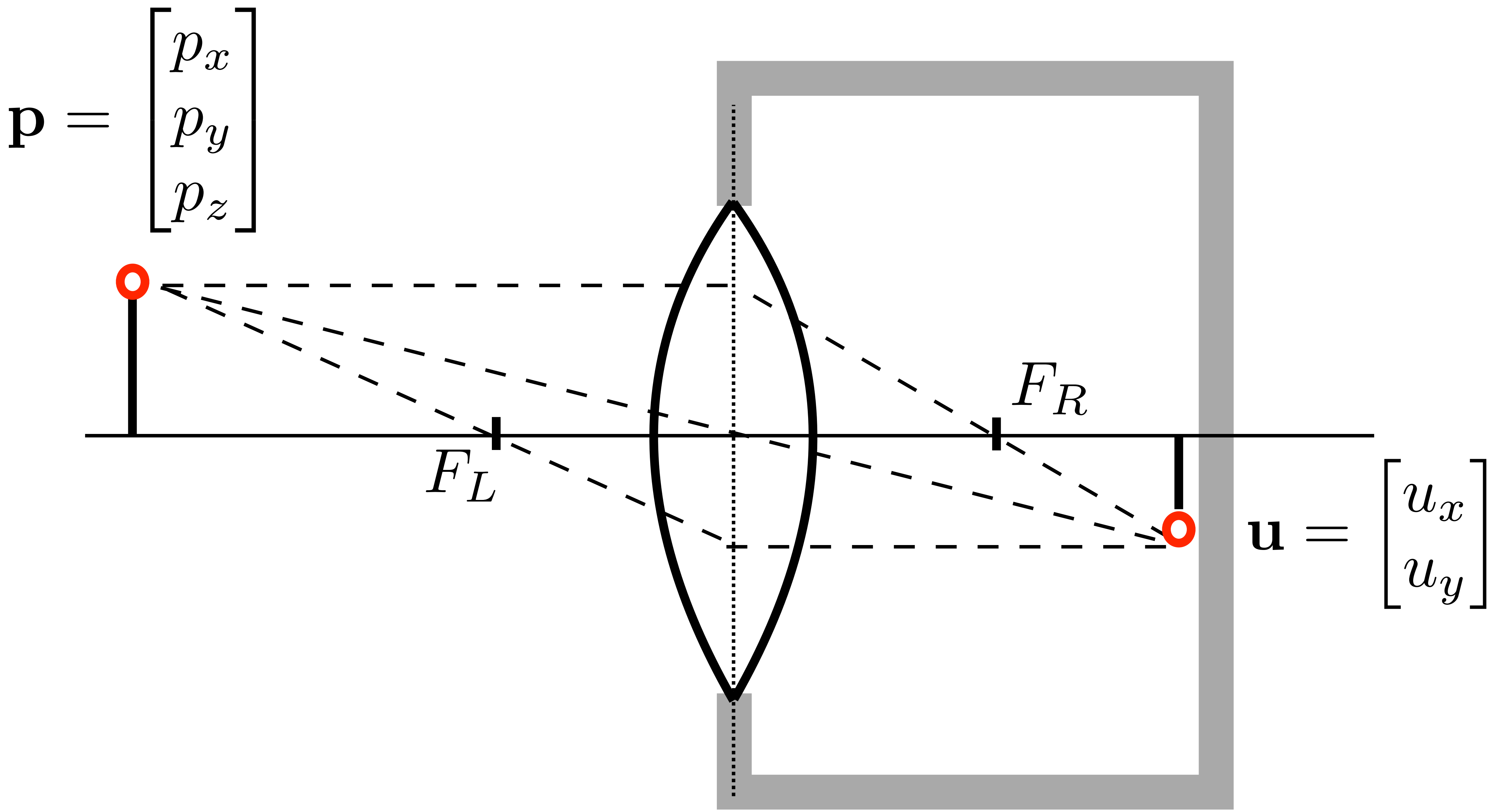


Light travels differently in different materials (refraction index)

Projection of 3D point in \camera on image plane

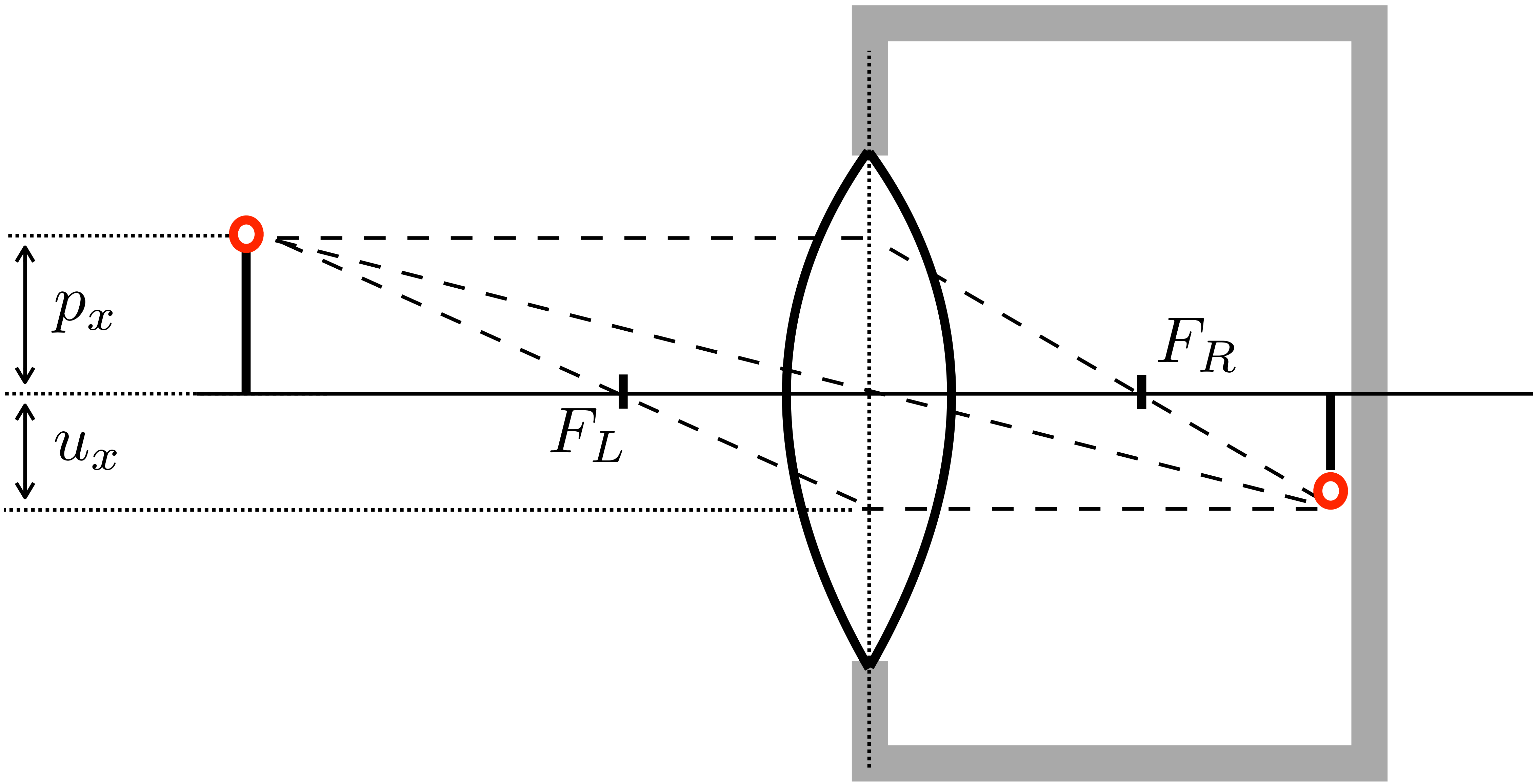


Projection of 3D point in \camera on image plane



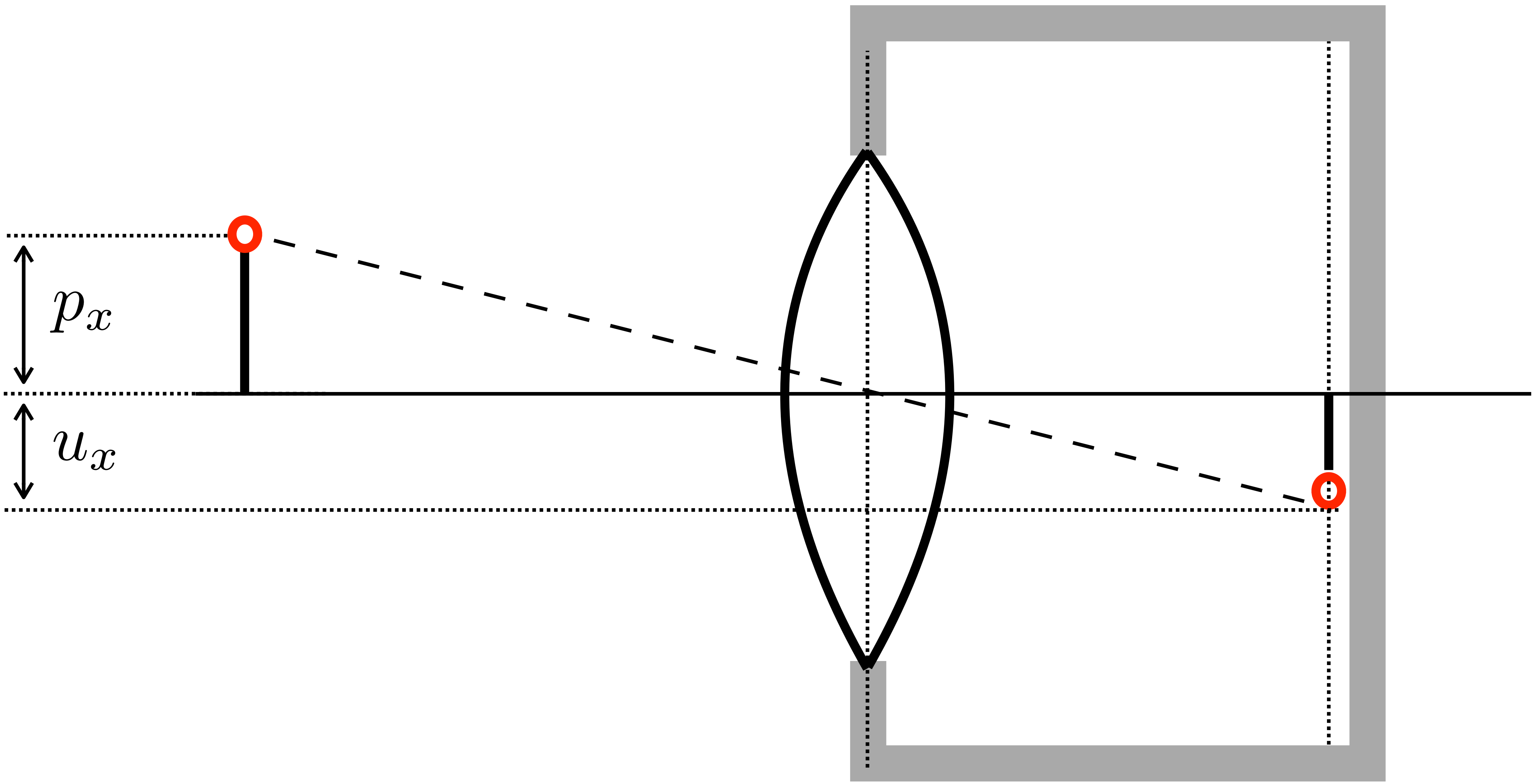
Lens geometry

Projection of 3D point in \camera on image plane



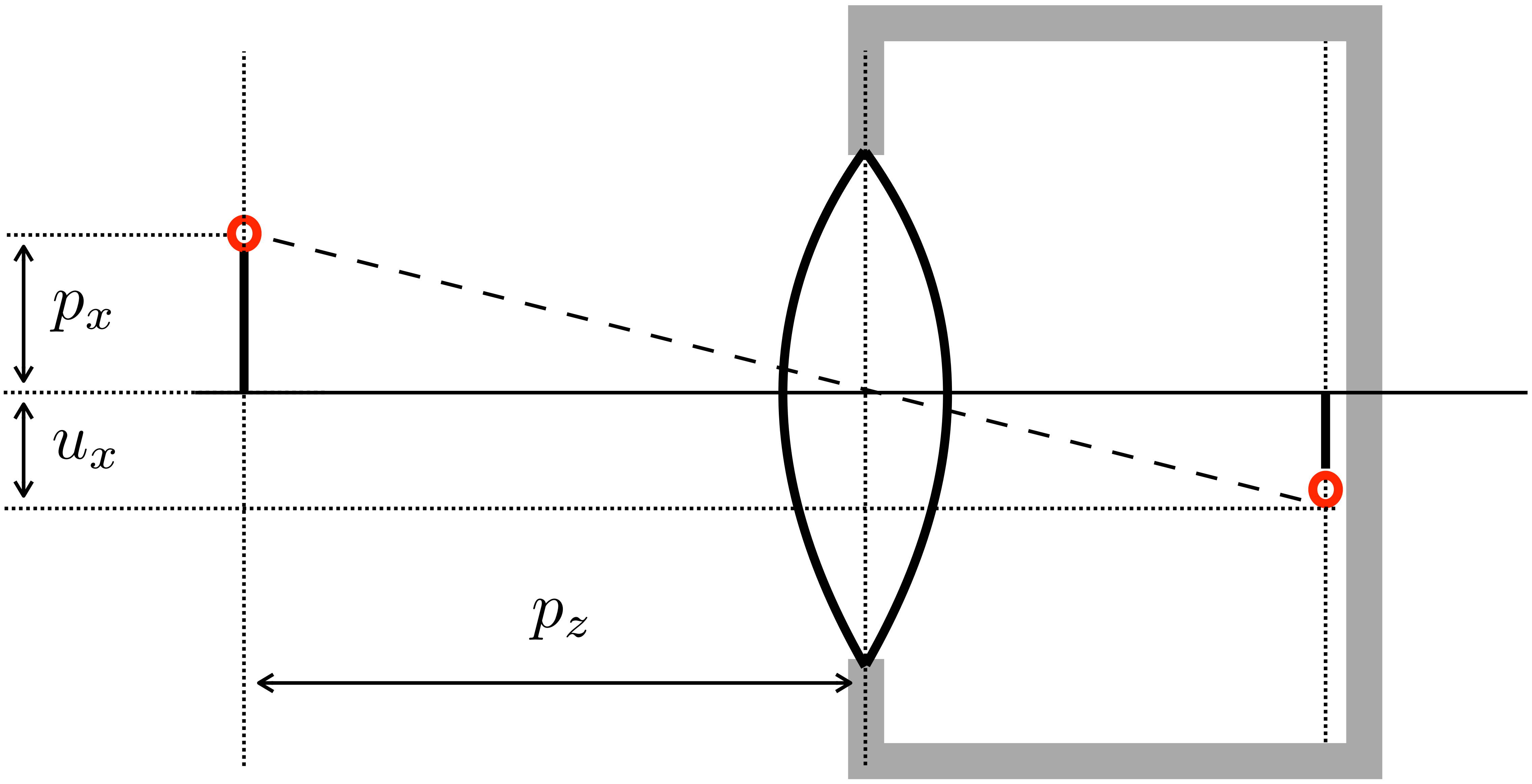
Lens geometry

Projection of 3D point in \camera on image plane

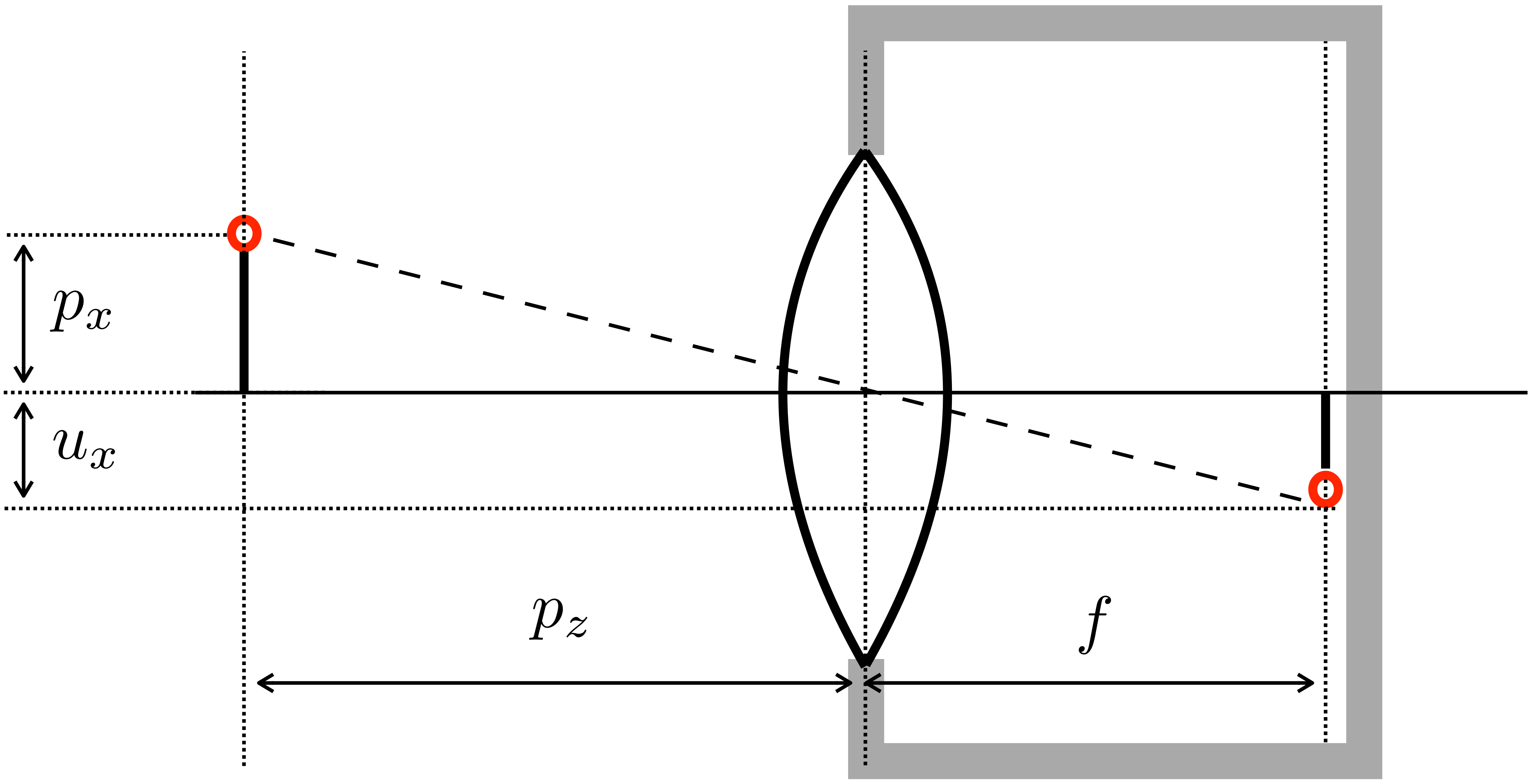


Simplified geometry

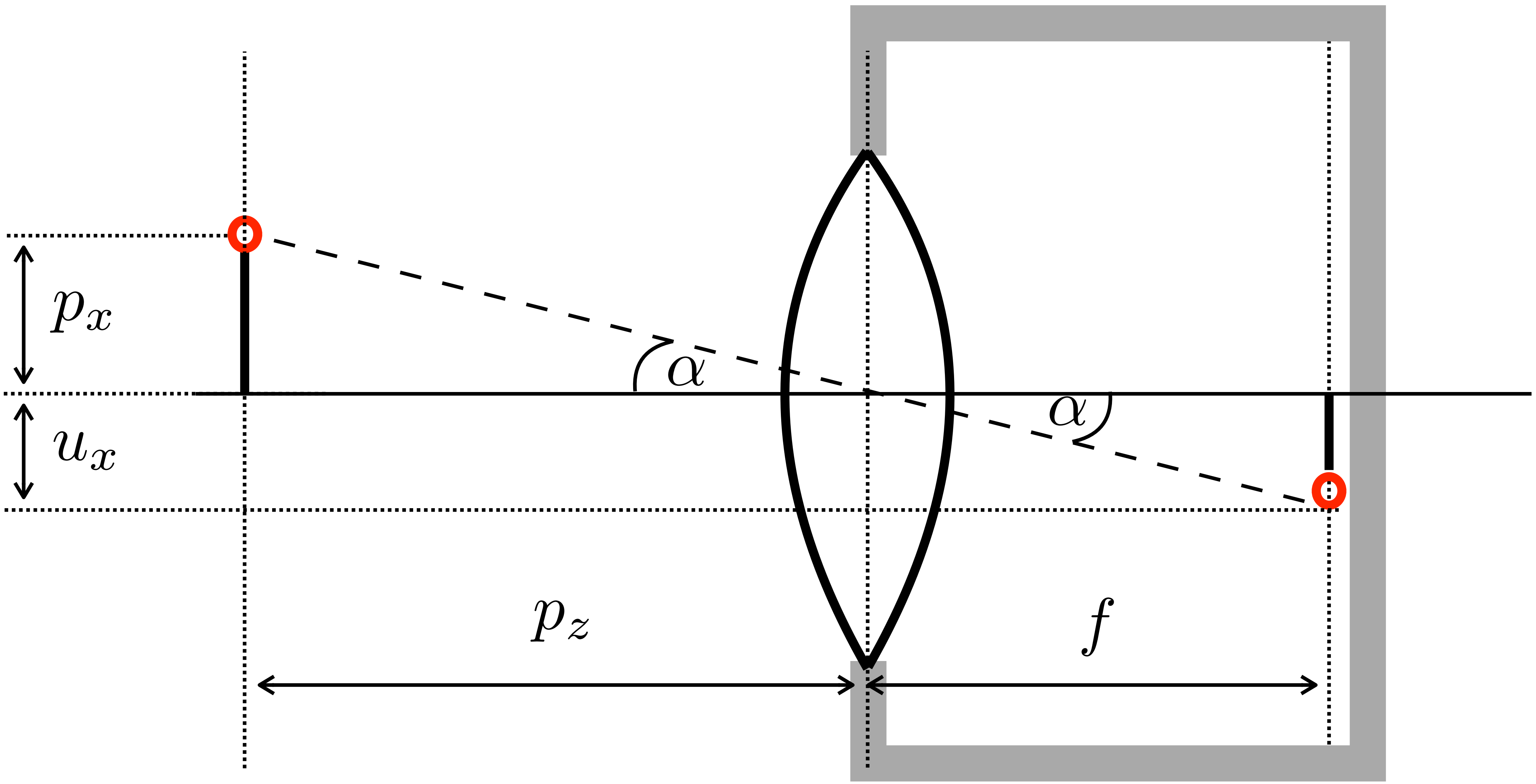
Projection of 3D point in \camera on image plane



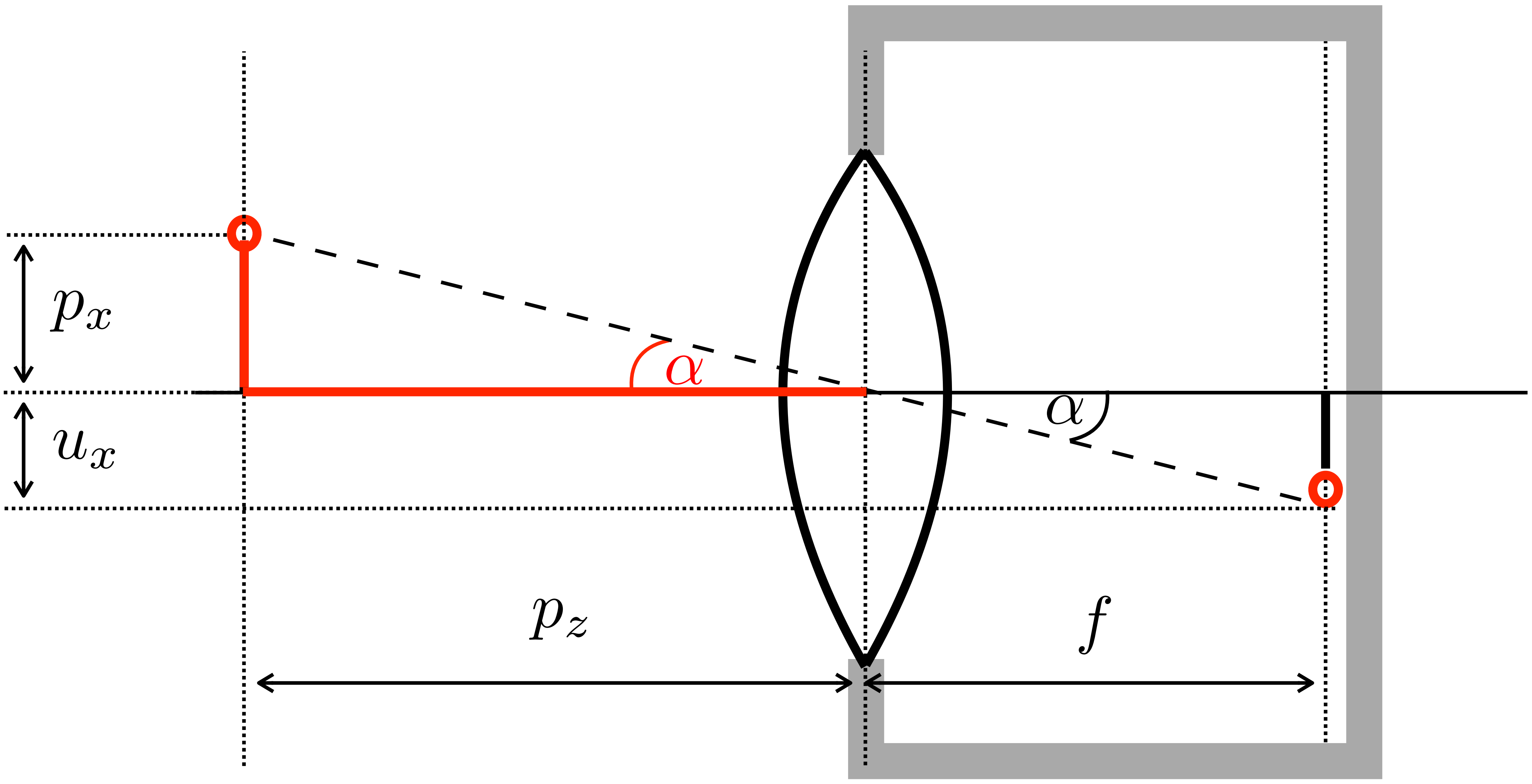
Projection of 3D point in \camera on image plane



Projection of 3D point in \camera on image plane

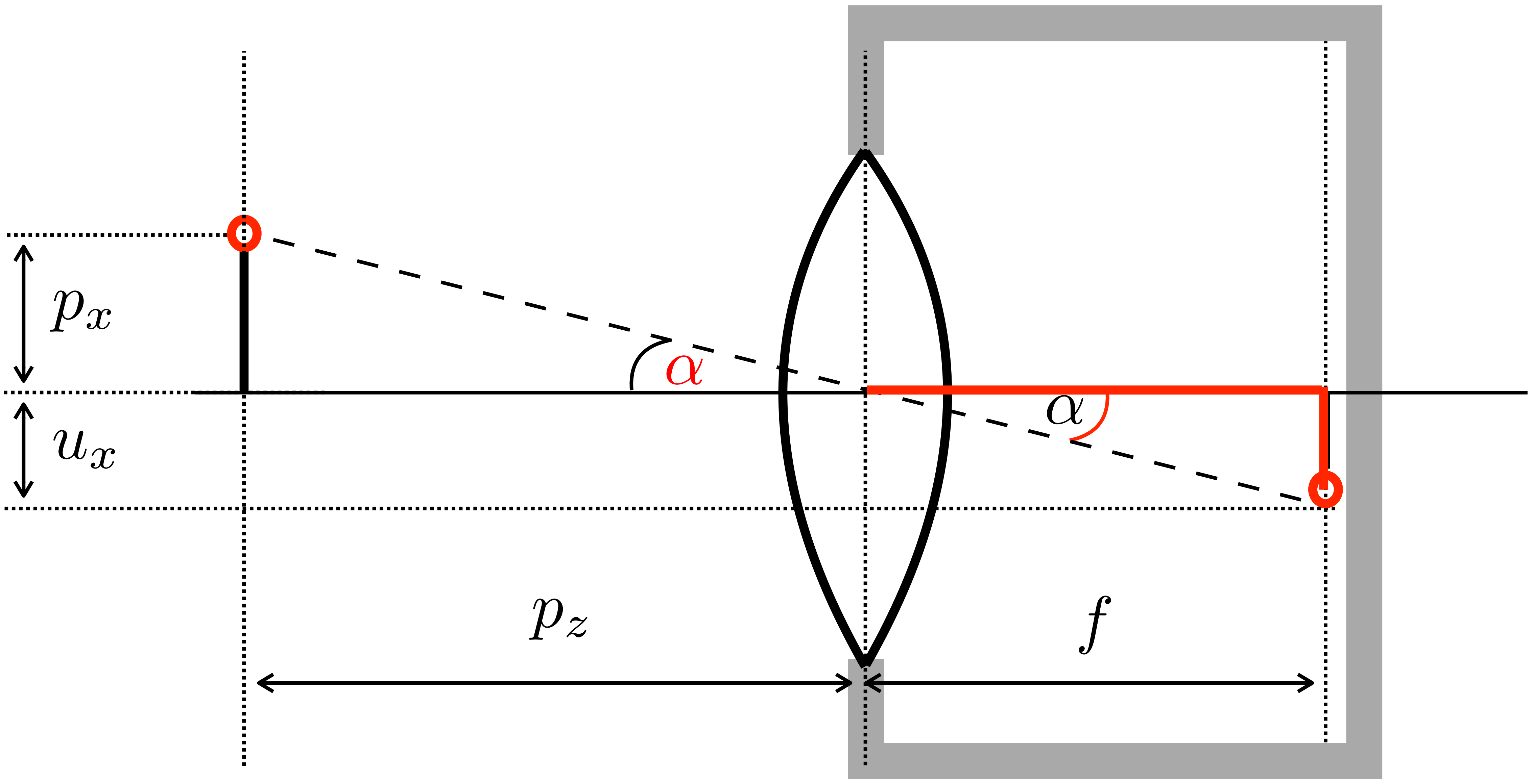


Projection of 3D point in \camera on image plane



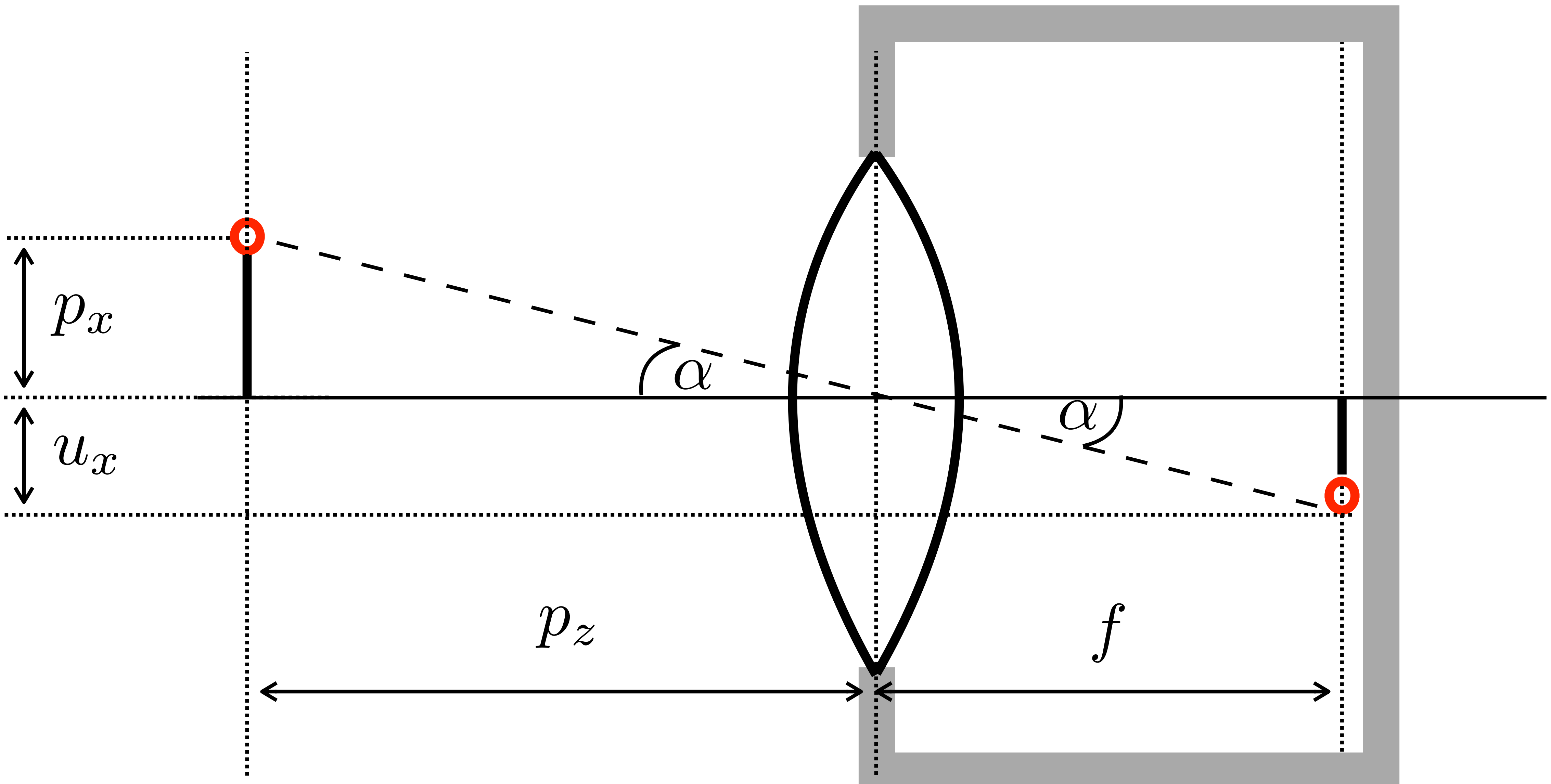
$$\mathrm{tg}(\alpha) = \frac{p_x}{p_z}$$

Projection of 3D point in \camera on image plane



$$\mathrm{tg}(\alpha) = \frac{p_x}{p_z} = \frac{u_x}{f}$$

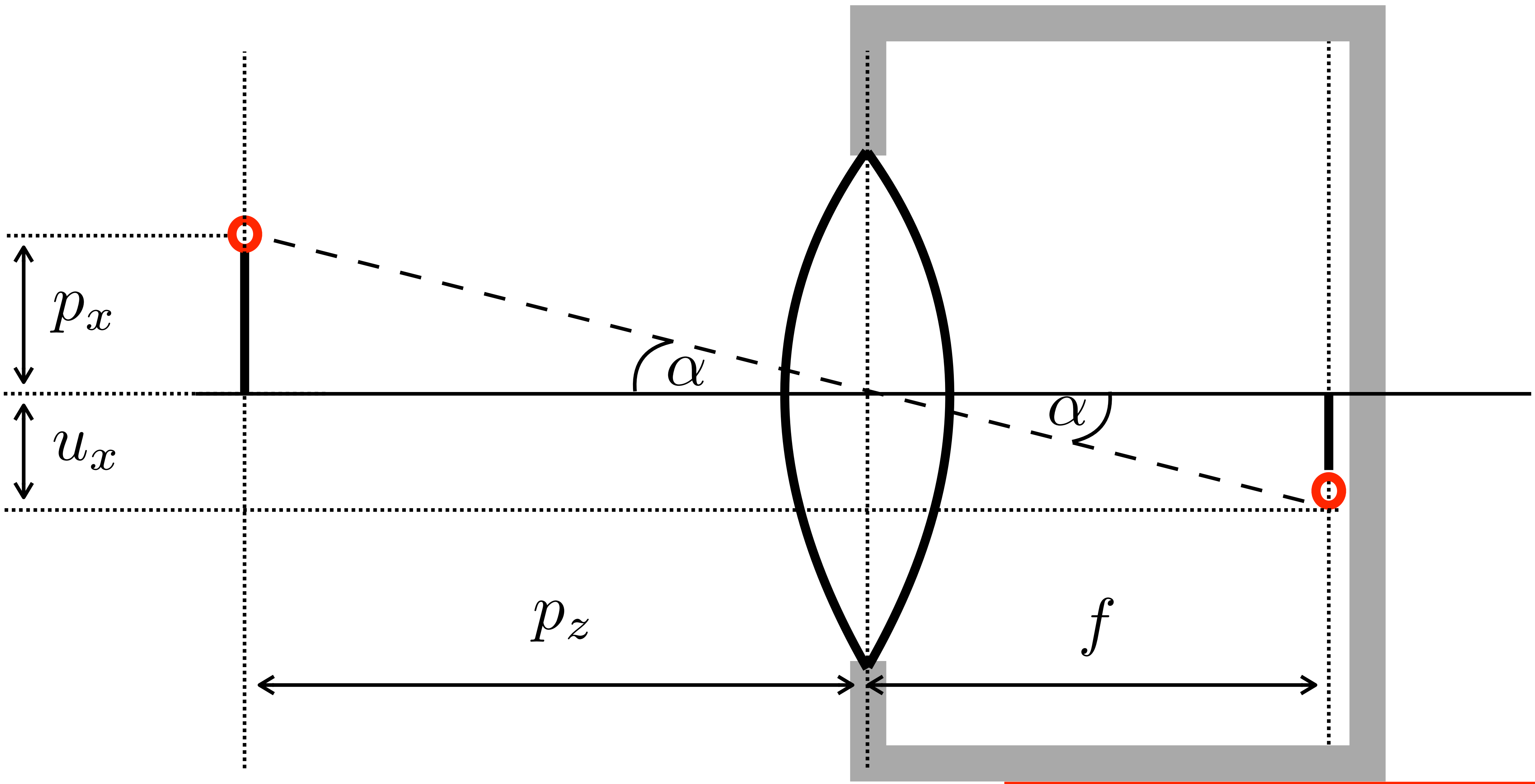
Projection of 3D point in \camera on image plane



$$\lambda u_x = f p_x$$
$$\lambda u_y = f p_y$$
$$\lambda = p_z$$

$$\Rightarrow \lambda \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

Projection of 3D point in \camera on image plane



$$\lambda \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

$$\lambda \bar{\mathbf{u}} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$

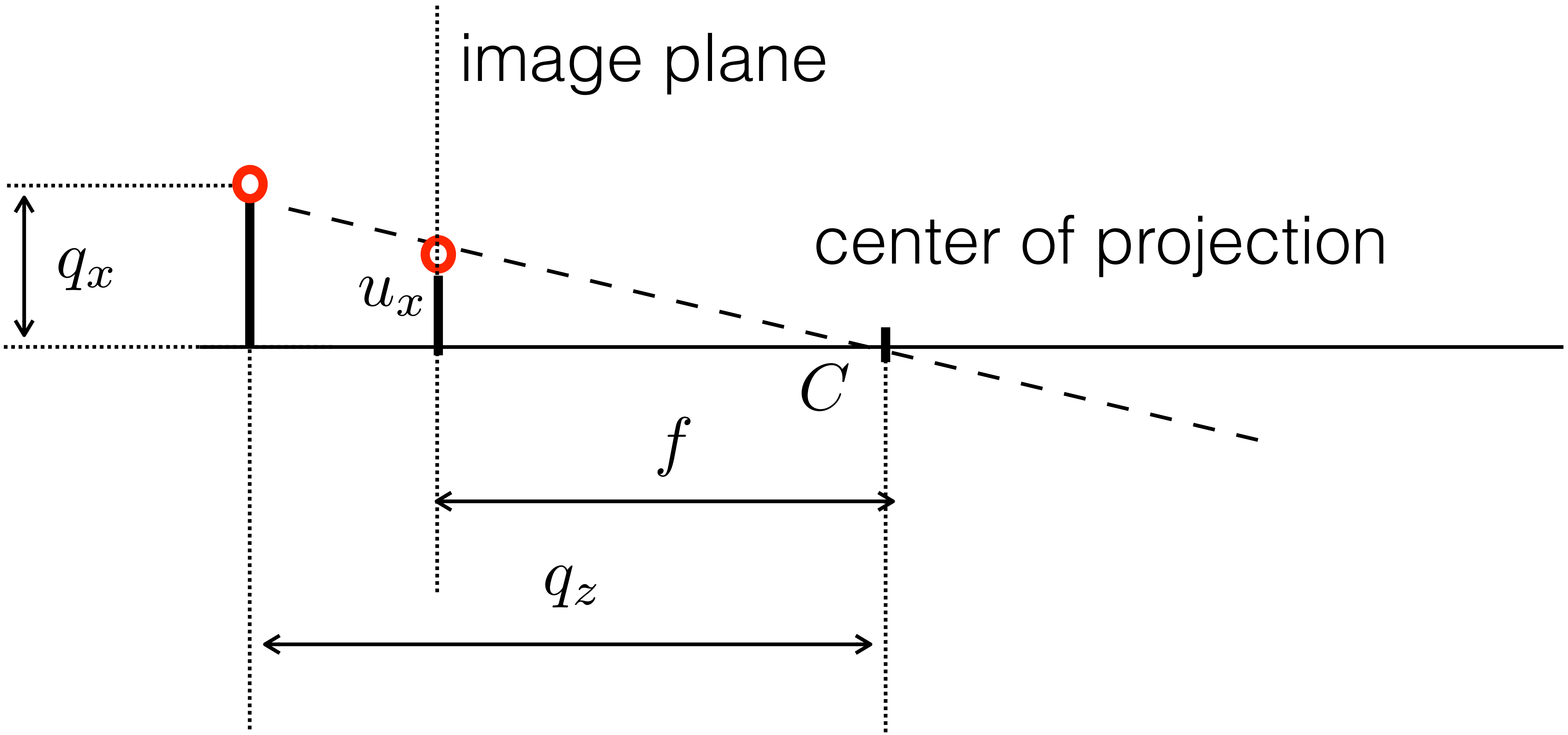
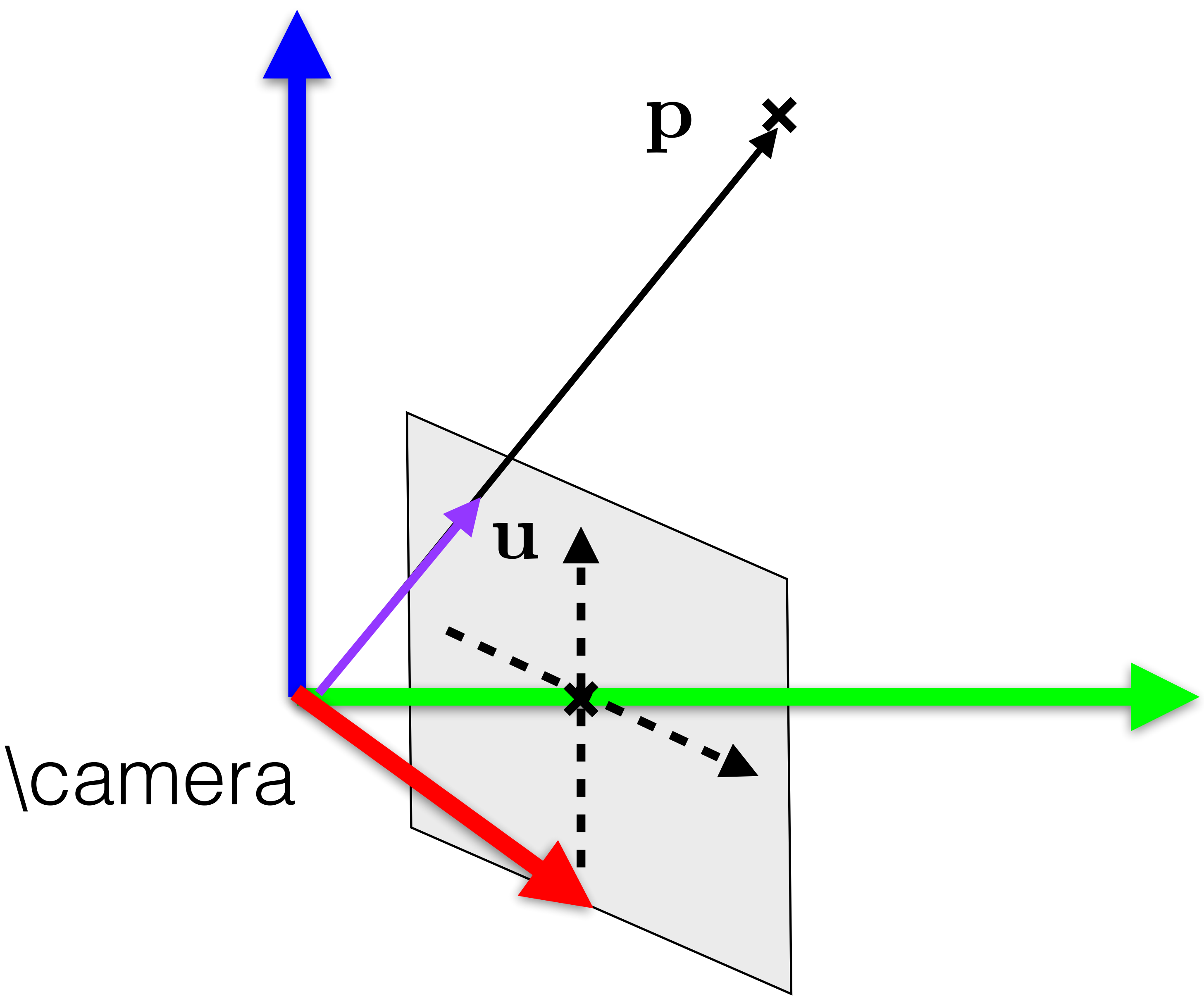


image plane
center of projection
q_x
u_x
C
f
q_z

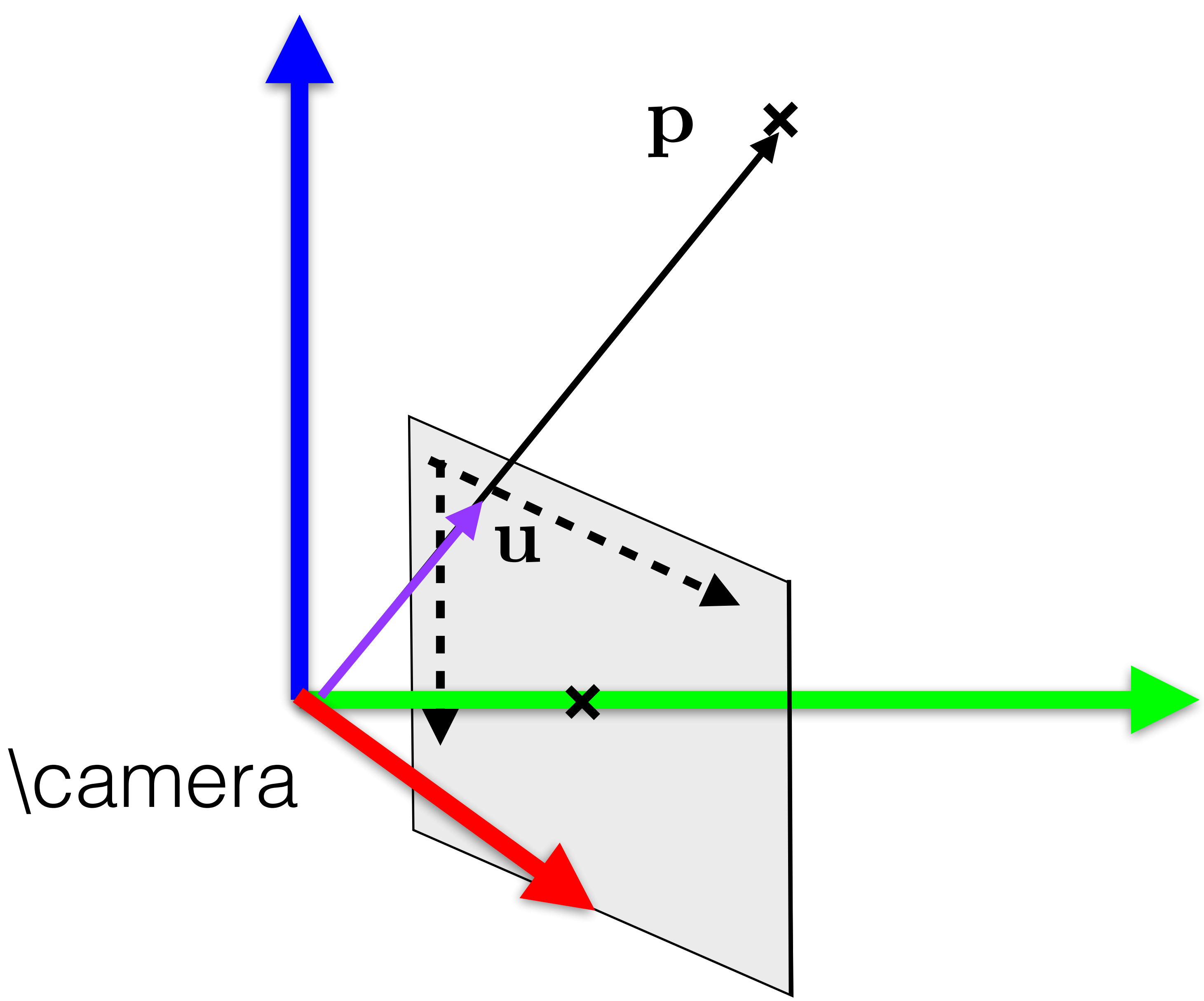
Projection 3D points on the image plane

$$\lambda \bar{\mathbf{u}} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$



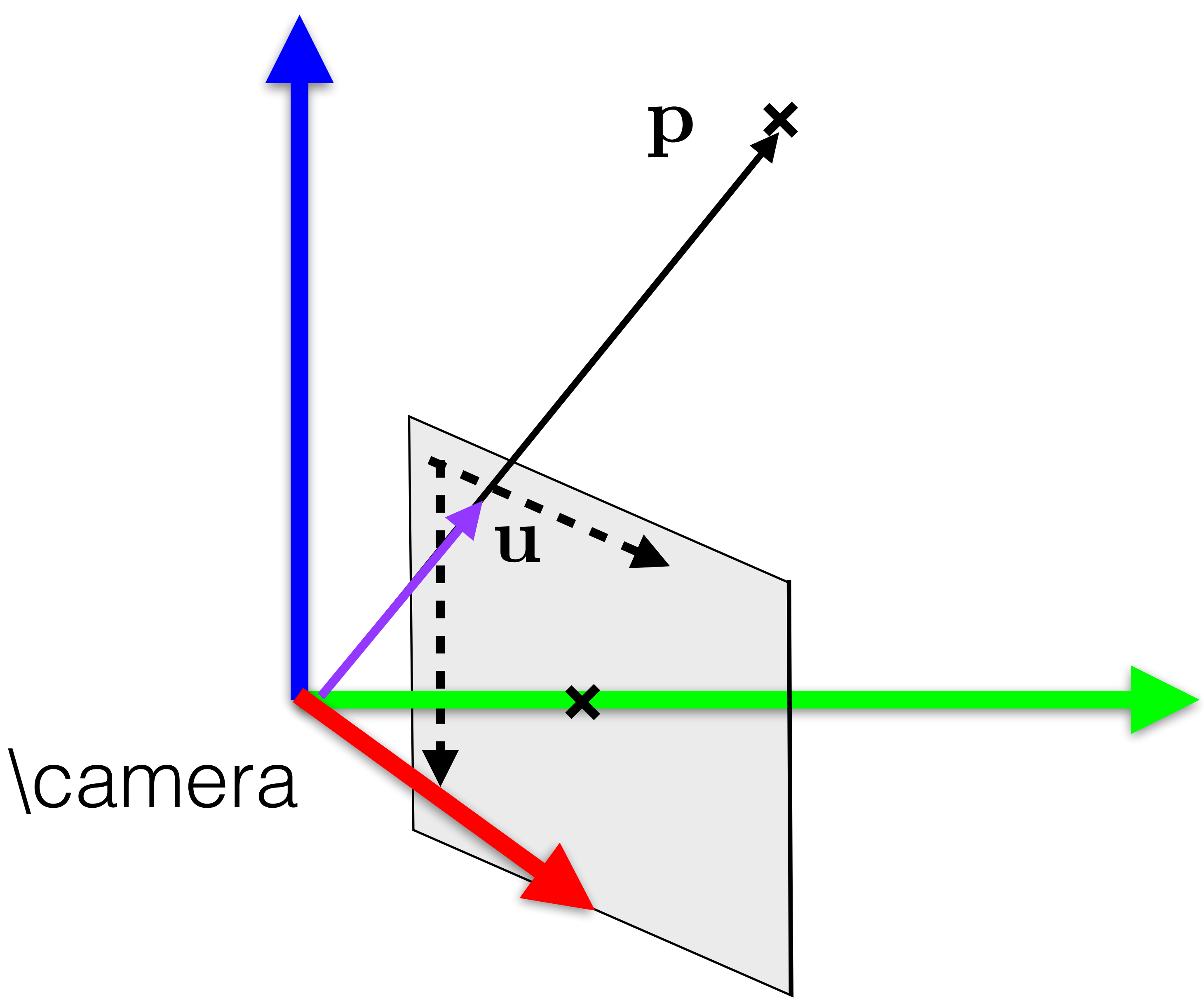
Projection 3D points on the image plane

$$\lambda \bar{\mathbf{u}} = \begin{bmatrix} 1 & 0 & o_x \\ 0 & 1 & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$



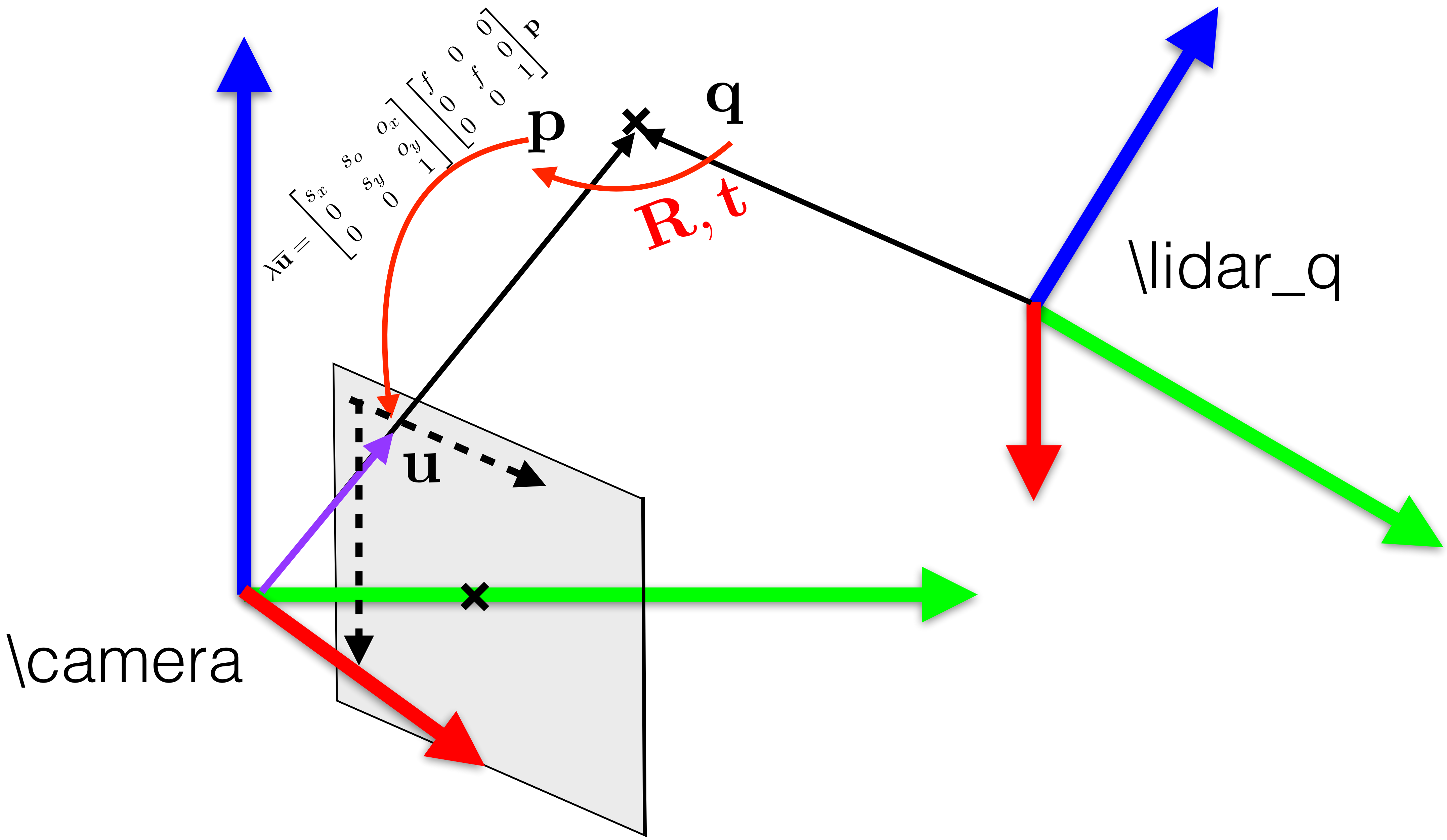
Projection 3D points on the image plane

$$\lambda \bar{\mathbf{u}} = \begin{bmatrix} s_x & s_o & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$



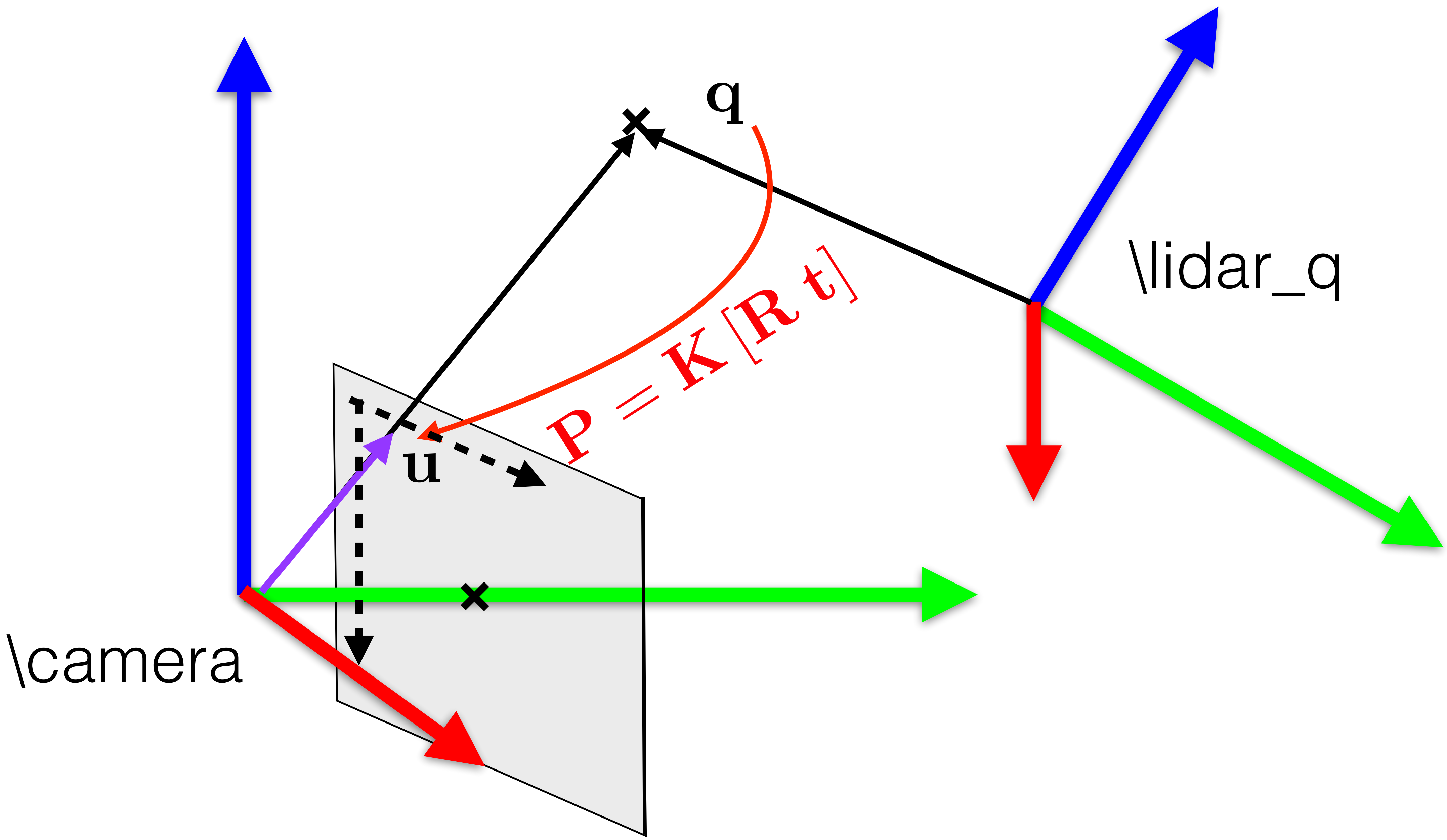
Projection 3D points on the image plane

$$\lambda \bar{\mathbf{u}} = \begin{bmatrix} s_x & s_o & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \bar{\mathbf{q}}$$



\camera to \lidar_q calibration

$$\lambda \overline{\mathbf{u}} = \underbrace{\mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}}_{\mathbf{P}} \overline{\mathbf{q}}$$



Projection 3D points on the image plane

$$\lambda\overline{\mathbf{u}} = \begin{bmatrix} s_x & s_o & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \overline{\mathbf{q}}$$

$\mathbf{K} \in \mathcal{R}^{3\times 3}$

$\mathbf{P} \in \mathcal{R}^{3\times 4}$

$\mathbf{K} \in \mathcal{R}^{3\times 3}$ .... intrinsic parameters

$\mathbf{R} \in \mathcal{SO}(3), \mathbf{t} \in \mathcal{R}^3$...... extrinsic parameters

$\mathbf{P} \in \mathcal{R}^{3\times 4}$ .... camera projection matrix

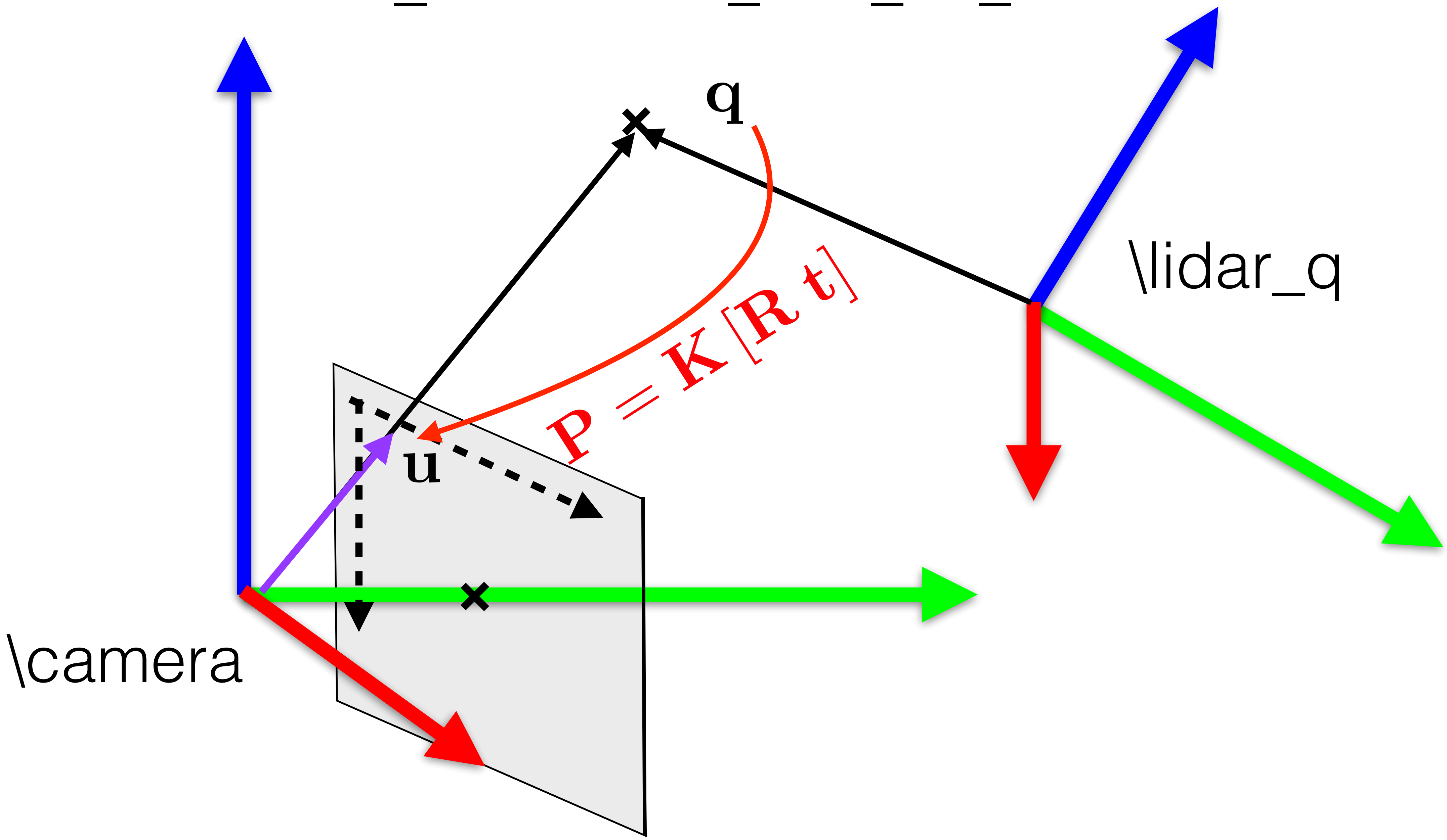
Example 1: Project point to a given camera.

Example 2: What is a ray of a pixel?

\camera to \lidar_q calibration

$$\lambda \overline{\mathbf{u}} = \underbrace{\mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}}_{\mathbf{P}} \overline{\mathbf{q}}$$

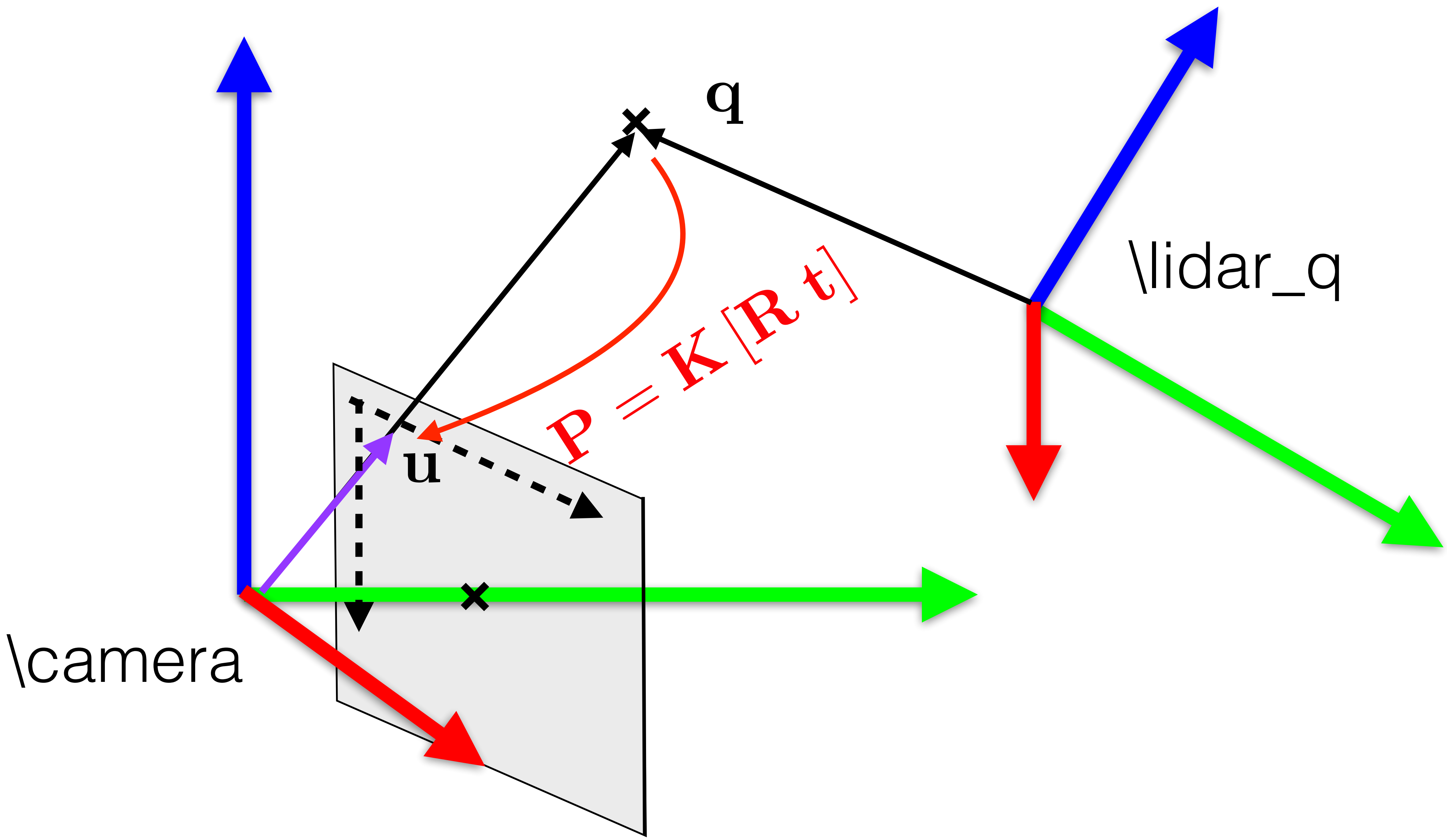
openCV: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html



\camera to \lidar_q calibration

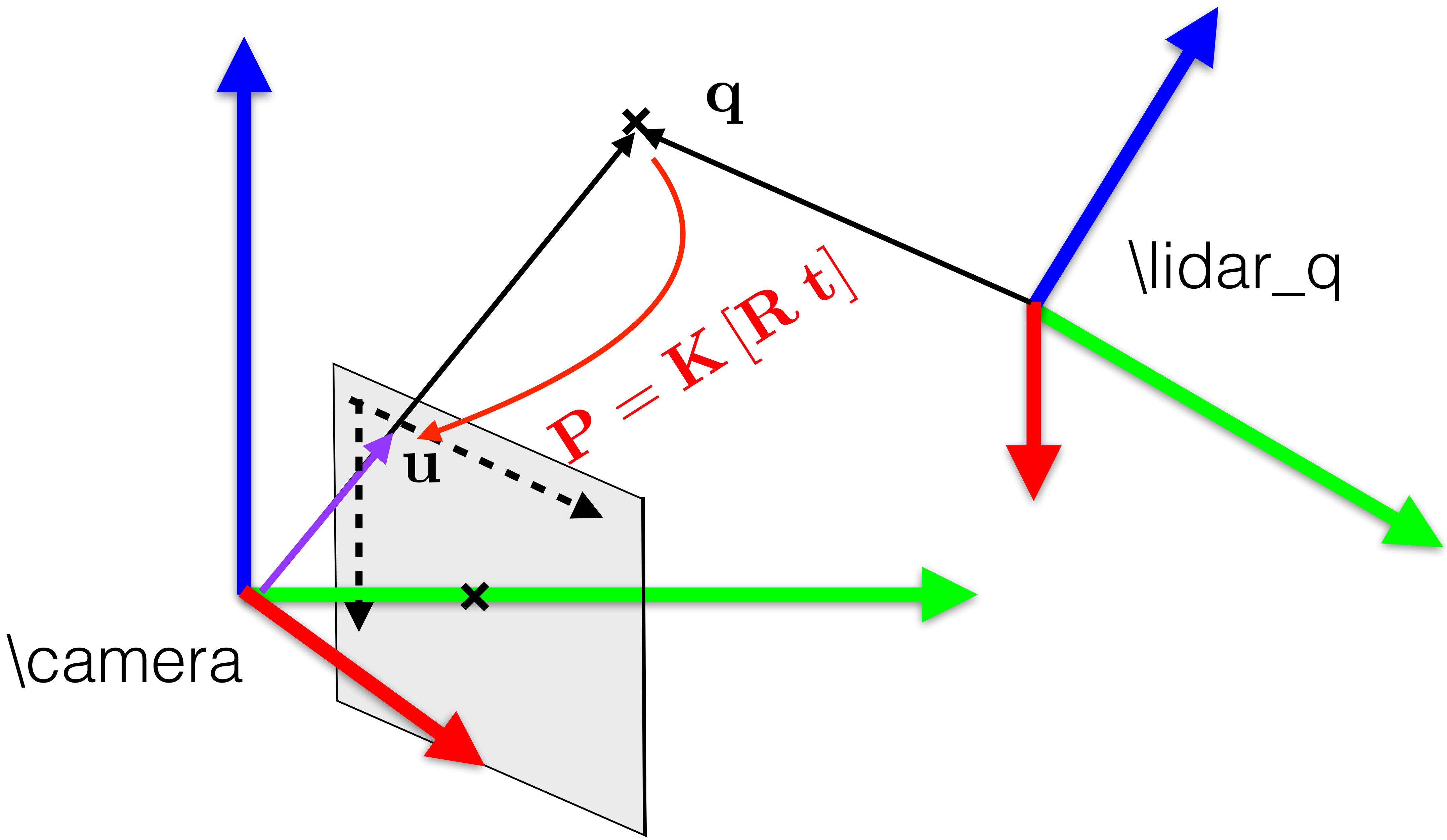
$$\lambda\bar{\mathbf{u}} = \mathbf{P}\bar{\mathbf{q}}$$

unknown



\camera to \lidar_q calibration

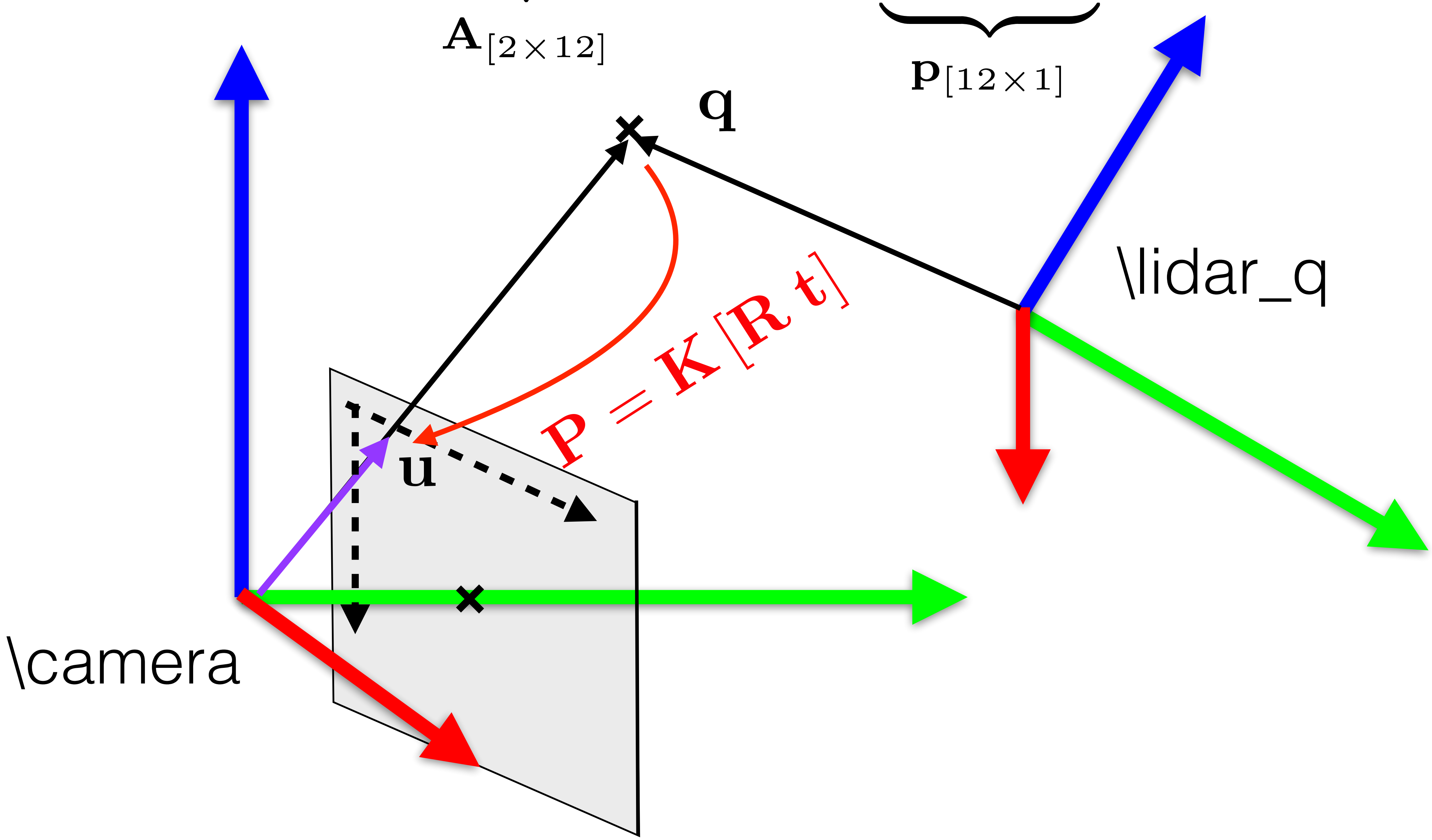
$$\lambda \overline{\mathbf{u}} = \mathbf{P}\overline{\mathbf{q}} \quad \Rightarrow \quad \underbrace{\begin{bmatrix} -\overline{\mathbf{q}}^\top & \mathbf{0}^\top & u_x \overline{\mathbf{q}}^\top \\ \mathbf{0}^\top & -\overline{\mathbf{q}}^\top & u_y \overline{\mathbf{q}}^\top \end{bmatrix}}_{\mathbf{A}_{[2\times 12]}} \underbrace{\begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}}_{\mathbf{p}_{[12\times 1]}} = \mathbf{0}_{[2\times 1]}$$



\camera to \lidar_q calibration

Each 2D-3D correspondence yields two equations:

$$\underbrace{\begin{bmatrix} -\bar{\mathbf{q}}_i^\top & \mathbf{0}^\top & u_{xi}\bar{\mathbf{q}}_i^\top \\ \mathbf{0}^\top & -\bar{\mathbf{q}}_i^\top & u_{yi}\bar{\mathbf{q}}_i^\top \end{bmatrix}}_{\mathbf{A}_{[2\times 12]}} \underbrace{\begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}}_{\mathbf{p}_{[12\times 1]}} = \mathbf{0}_{[2\times 1]}$$



\camera to \lidar_q calibration

Each 2D-3D correspondence yields two equations:

$$\underbrace{\begin{bmatrix} -\overline{\mathbf{q}}_i^\top & \mathbf{0}^\top & u_{xi}\overline{\mathbf{q}}_i^\top \\ \mathbf{0}^\top & -\overline{\mathbf{q}}_i^\top & u_{yi}\overline{\mathbf{q}}_i^\top \end{bmatrix}}_{\mathbf{A}_{[2\times 12]}} \underbrace{\begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}}_{\mathbf{p}_{[12\times 1]}} = \mathbf{0}_{[2\times 1]}$$

For N 2D-3D correspondences, we obtain (2Nx12) homogeneous linear system $\mathbf{A}\mathbf{p} = \mathbf{0}$

Assuming

- i.i.d. measurements and
- gaussian noise between left-hand-side and right-hand-side

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

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

Lagrange function:

$$L(\mathbf{p}, \lambda) = \|\mathtt{A}\mathbf{p}\| + \lambda(1 - \|\mathbf{p}\|) = \mathbf{p}^\top \mathtt{A}^\top \mathtt{A}\mathbf{p} + \lambda(1 - \mathbf{p}^\top \mathbf{p})$$

Critical points:

$$\frac{\partial L(\mathbf{p}, \lambda)}{\partial \mathbf{p}} = 2\mathtt{A}^\top \mathtt{A}\mathbf{p} - 2\lambda\mathbf{p} = \mathbf{0}$$

$$\frac{\partial L(\mathbf{p}, \lambda)}{\partial \lambda} = 1 - \mathbf{p}^\top \mathbf{p} = \mathbf{0}$$

First equation is characteristic equation $(\mathtt{A}^\top \mathtt{A} - \lambda \mathtt{I})\mathbf{p} = \mathbf{0}$

Every eigen-vector of $\mathtt{A}^\top \mathtt{A}$ is the critical point   choose one

Cost function in these eigen vectors is equal to eigen-values

$$\|\mathtt{A}\mathbf{p}\| = \mathbf{p}^\top \mathtt{A}^\top \mathtt{A}\mathbf{p} = \mathbf{p}^\top \lambda \mathbf{p} = \lambda \mathbf{p}^\top \mathbf{p} = \lambda\|\mathbf{p}\| = \lambda$$

Solution is the eigen-vector of $\mathtt{A}^\top \mathtt{A}$ with the smallest eigen-value

Summary camera calibration

- Manually estimate 2D-3D correspondences
- Build matrix

$$\mathbf{A} = \begin{bmatrix} -\mathbf{q}^\top & \mathbf{0}^\top & u_x\mathbf{q}^\top \\ \mathbf{0}^\top & -\mathbf{q}^\top & u_y\mathbf{q}^\top \\ . & . & . \\ . & . & . \\ . & . & . \end{bmatrix}$$

- Find eigen-values and eigen-vectors of $\mathbf{A}^\top\mathbf{A}$ (python: `numpy.linalg.eig`)
- Reshape the eigen-vector $\mathbf{p} \in \mathcal{R}^{12\times 1}$ with the smallest eigen-value to camera matrix $\mathbf{P} \in \mathcal{R}^{3\times 4}$
- Scale does not matter: $\mathbf{P} = \mathbf{P}/\|[\mathbf{p}_{31}, \mathbf{p}_{32}, \mathbf{p}_{33}]^\top\|$
- Optionally decompose:

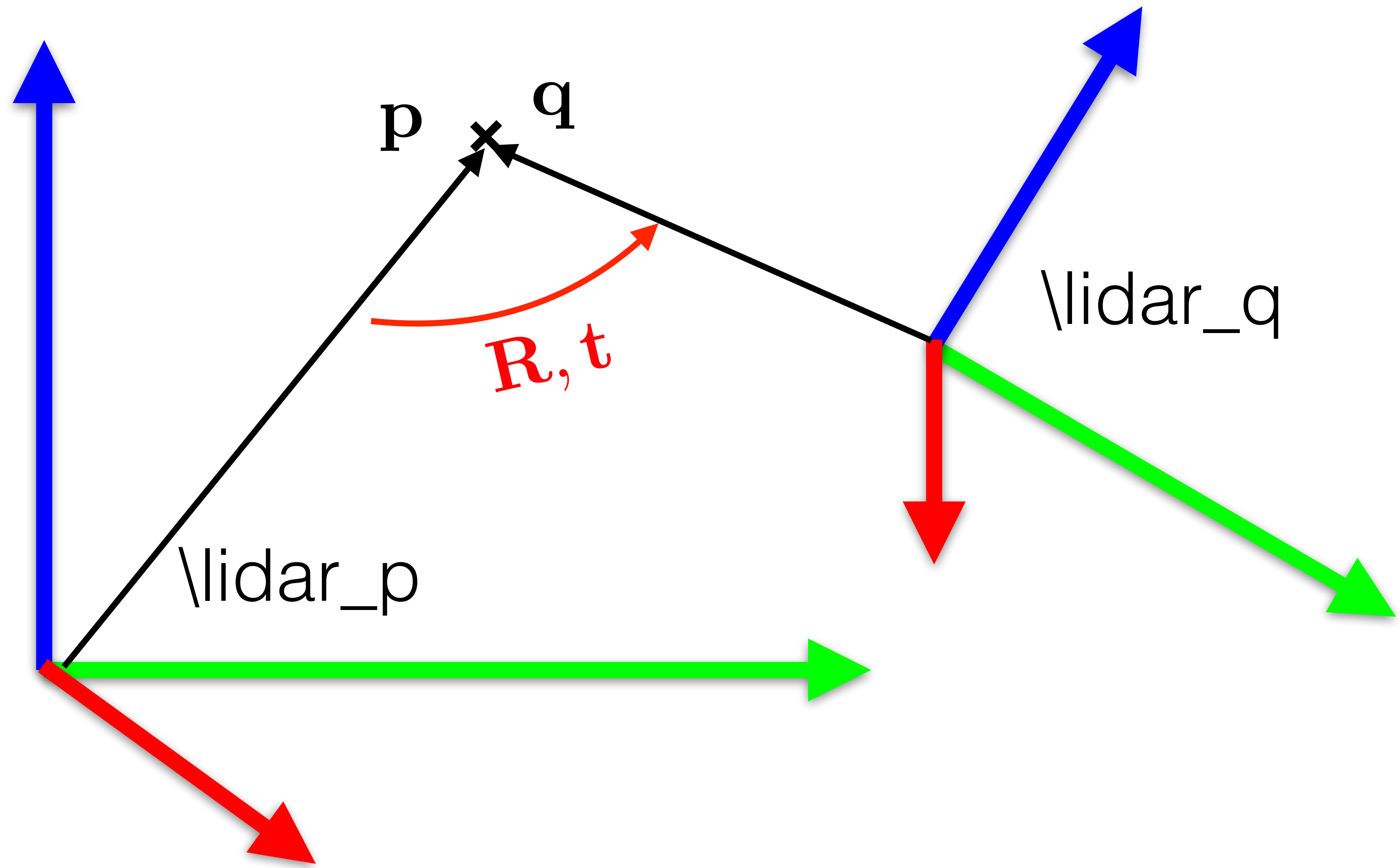
$$\mathbf{P} = [\underbrace{\mathbf{KR}}_{\mathbf{B}} \ \underbrace{\mathbf{Kt}}_{\mathbf{c}}] = [\mathbf{B} \ \ \mathbf{c}]$$

$$\mathbf{K}, \mathbf{R} = qr(\mathbf{B})$$
$$\mathbf{t} = \mathbf{K}^{-1}\mathbf{c}$$

(python: `numpy.linalg.qr`)

Summary

lidar-lidar calibration from 3D-3D correspondences



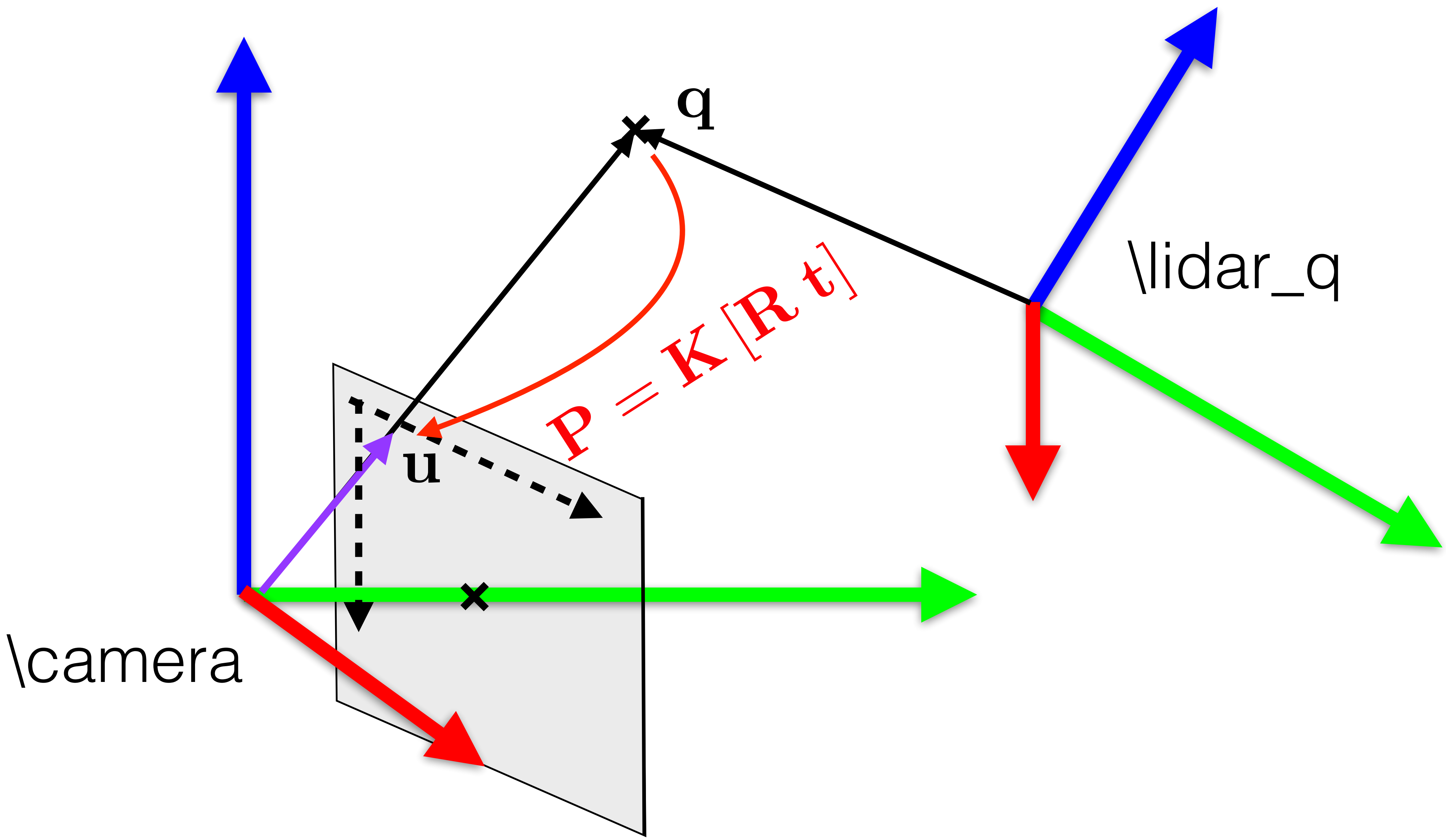
Solve: $\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\arg\min} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$

Solution: $\mathbf{R}^* = \mathbf{V}\mathbf{U}^\top$

$\mathbf{t}^* = \widetilde{\mathbf{q}} - \mathbf{R}^* \widetilde{\mathbf{p}}$

Summary

camera-lidar calibration from 2D-3D correspondences



Solve: $\mathbf{p}^* = \text{argmin} ||\mathbf{A}\mathbf{p}|| \text{ subject to } ||\mathbf{p}|| = 1$

Solution: smallest eigen-vector of $\mathbf{A}^\top \mathbf{A}$