

Robot, lidar, Euclidean transformations

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>

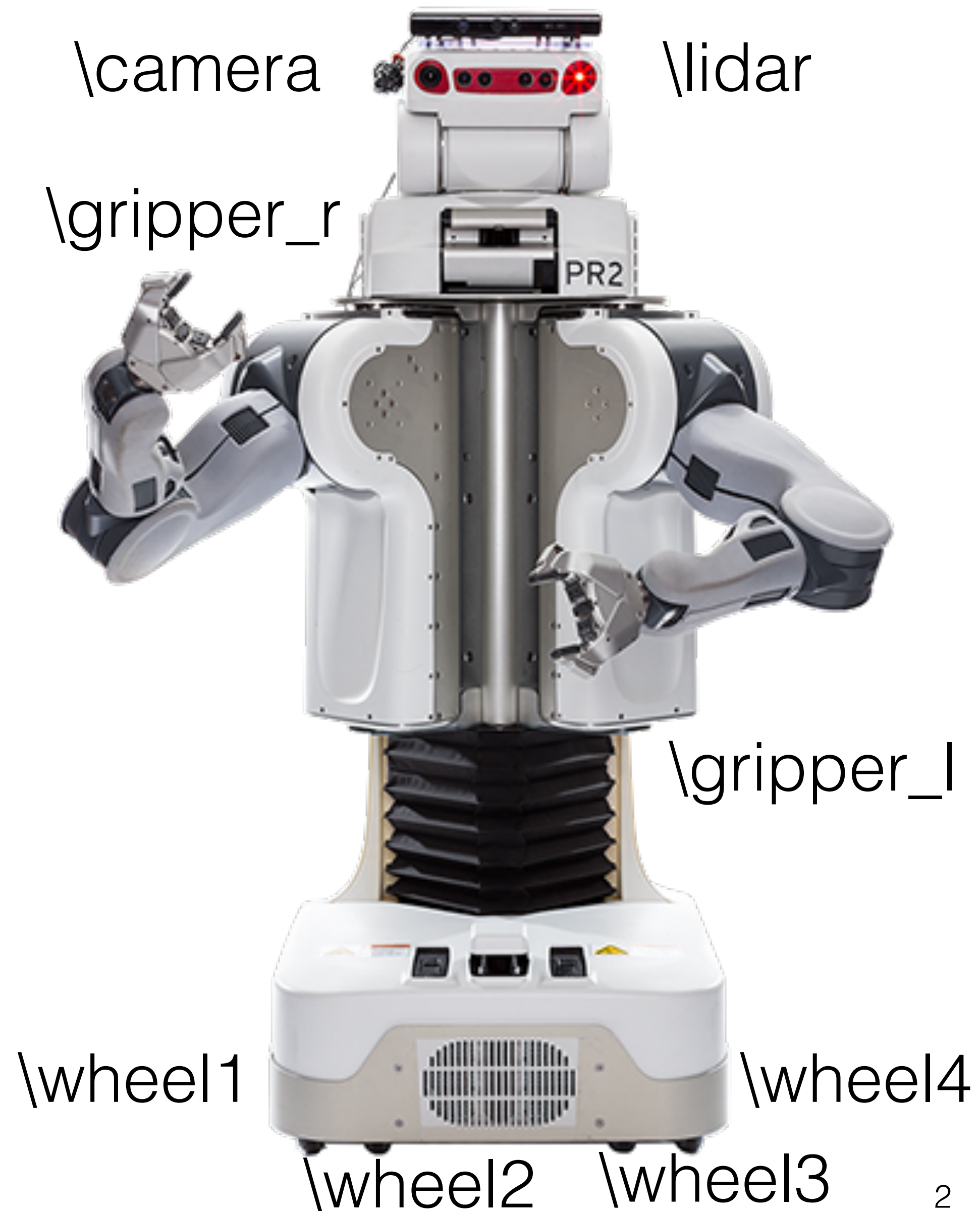


Department for Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague

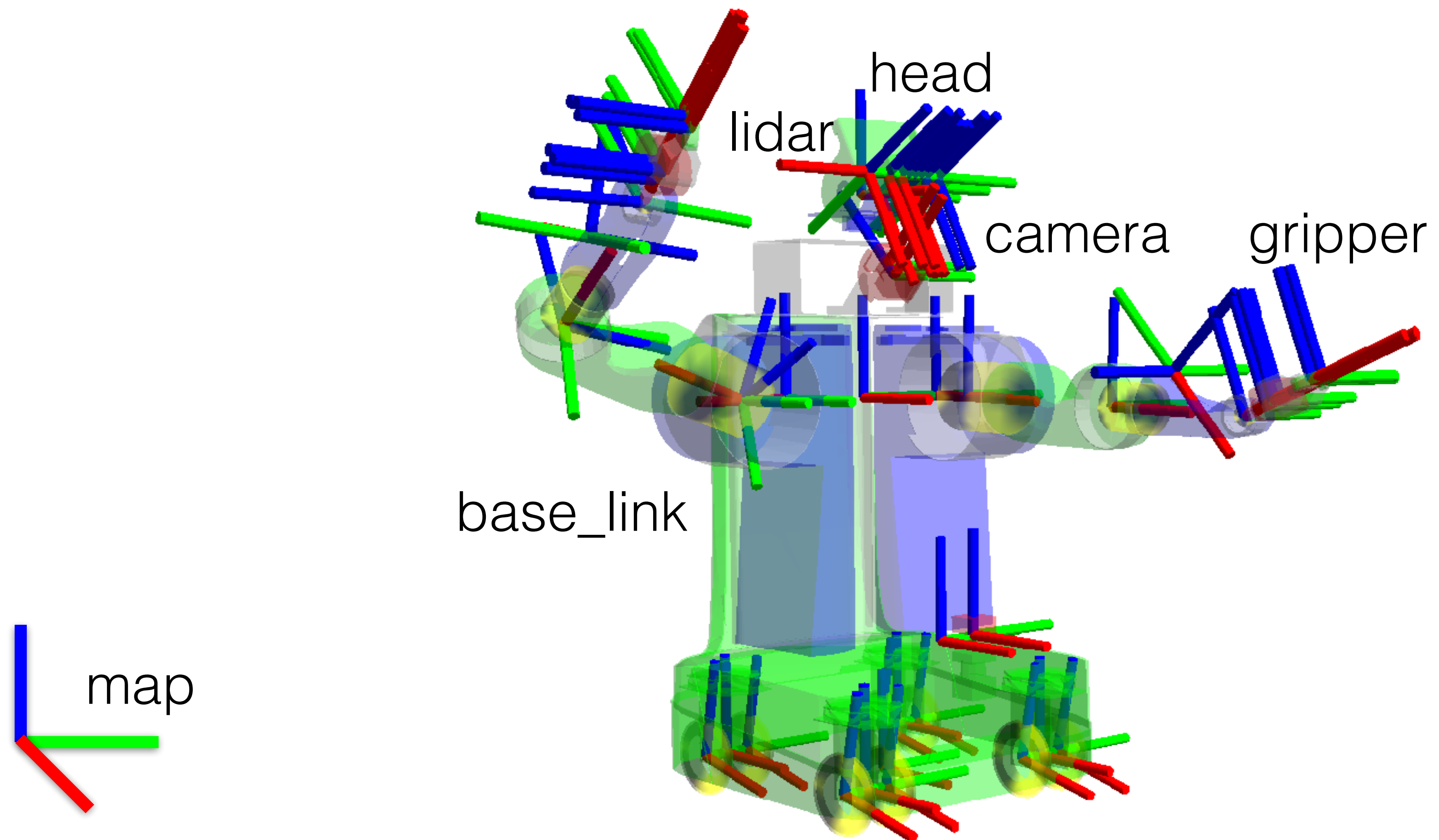


Why transformation among coordinate frames are important?

- 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.

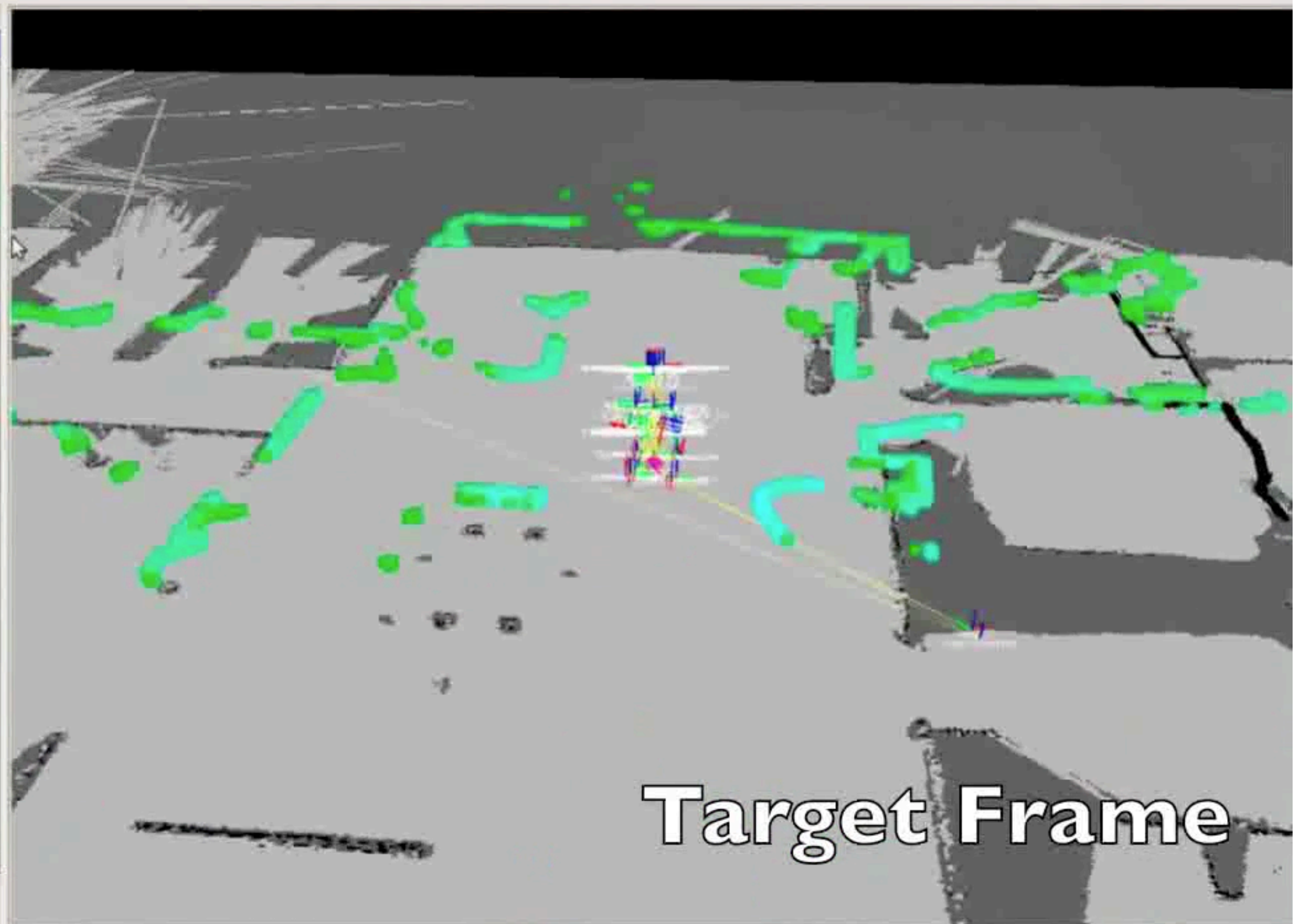


- Each coordinate frame define 3D Euclidean space.
- It is uniquely determined by its name.



(0,0,0)
/map
/base_link
Laser Scan)

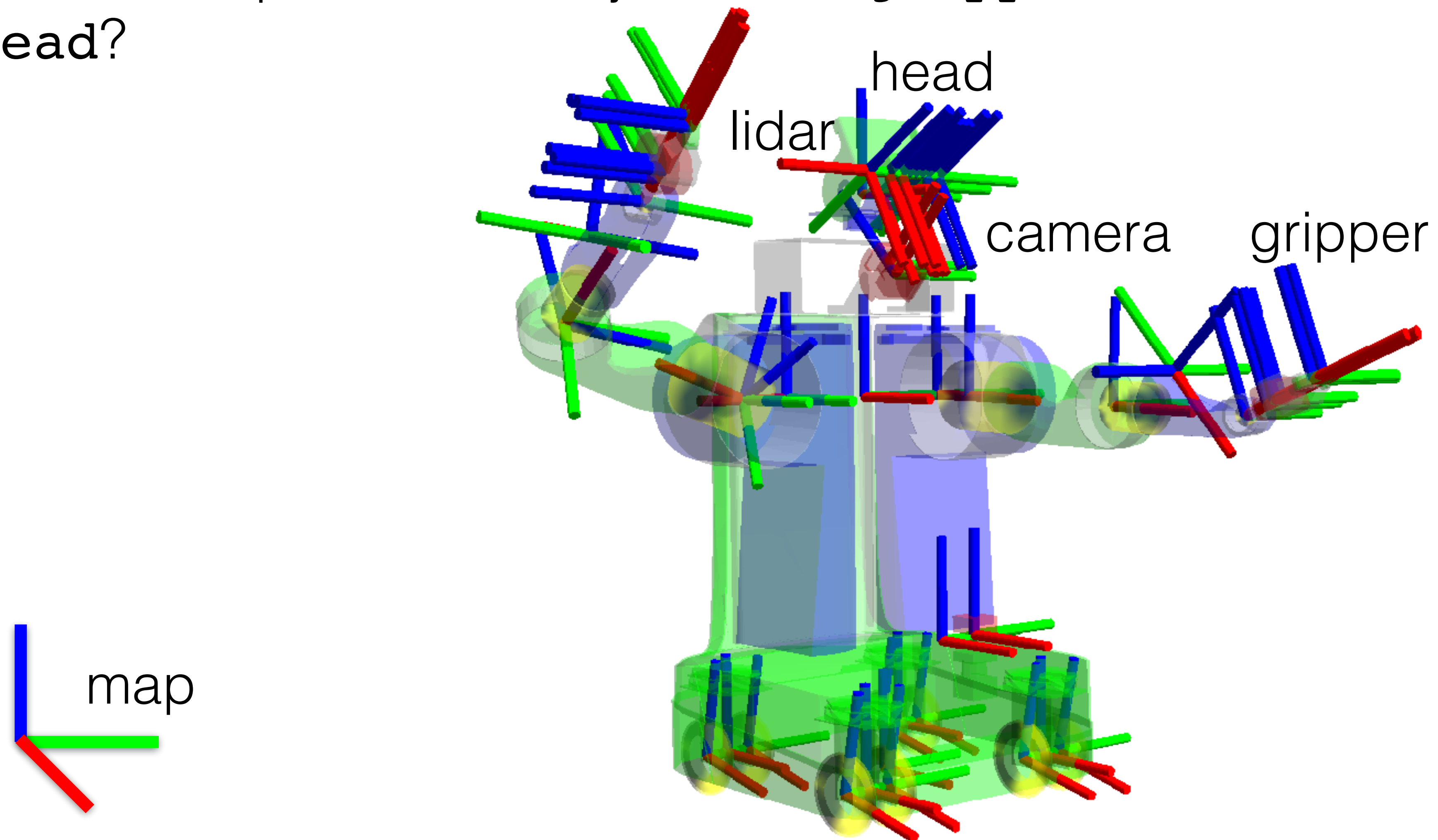
move



Target Frame

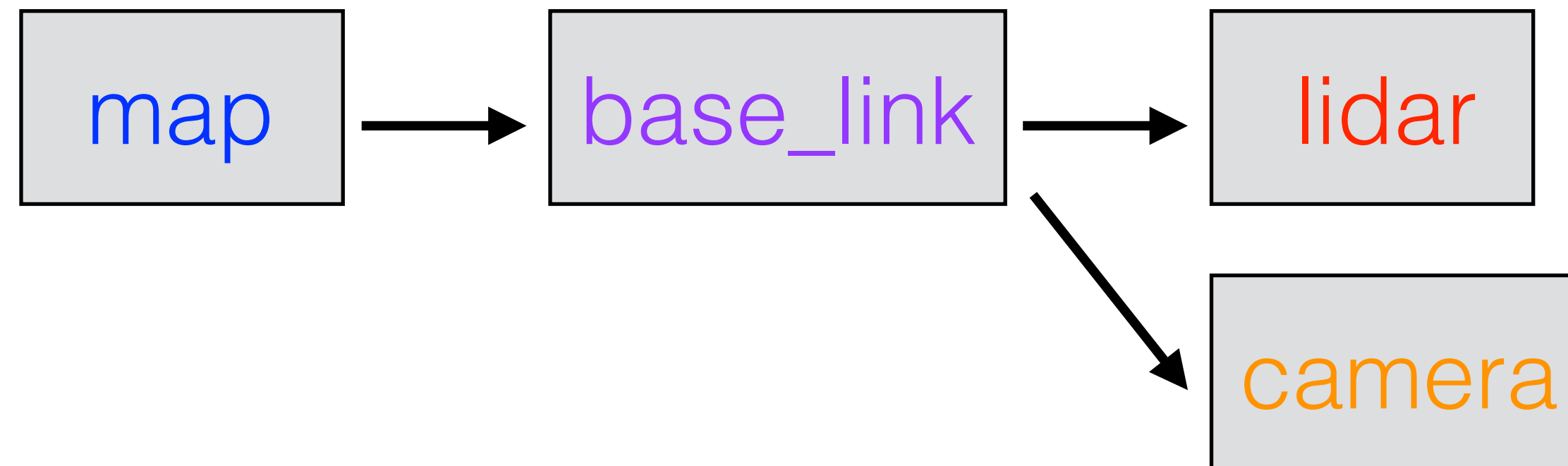
In arbitrary time we would like to answer questions like that:

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



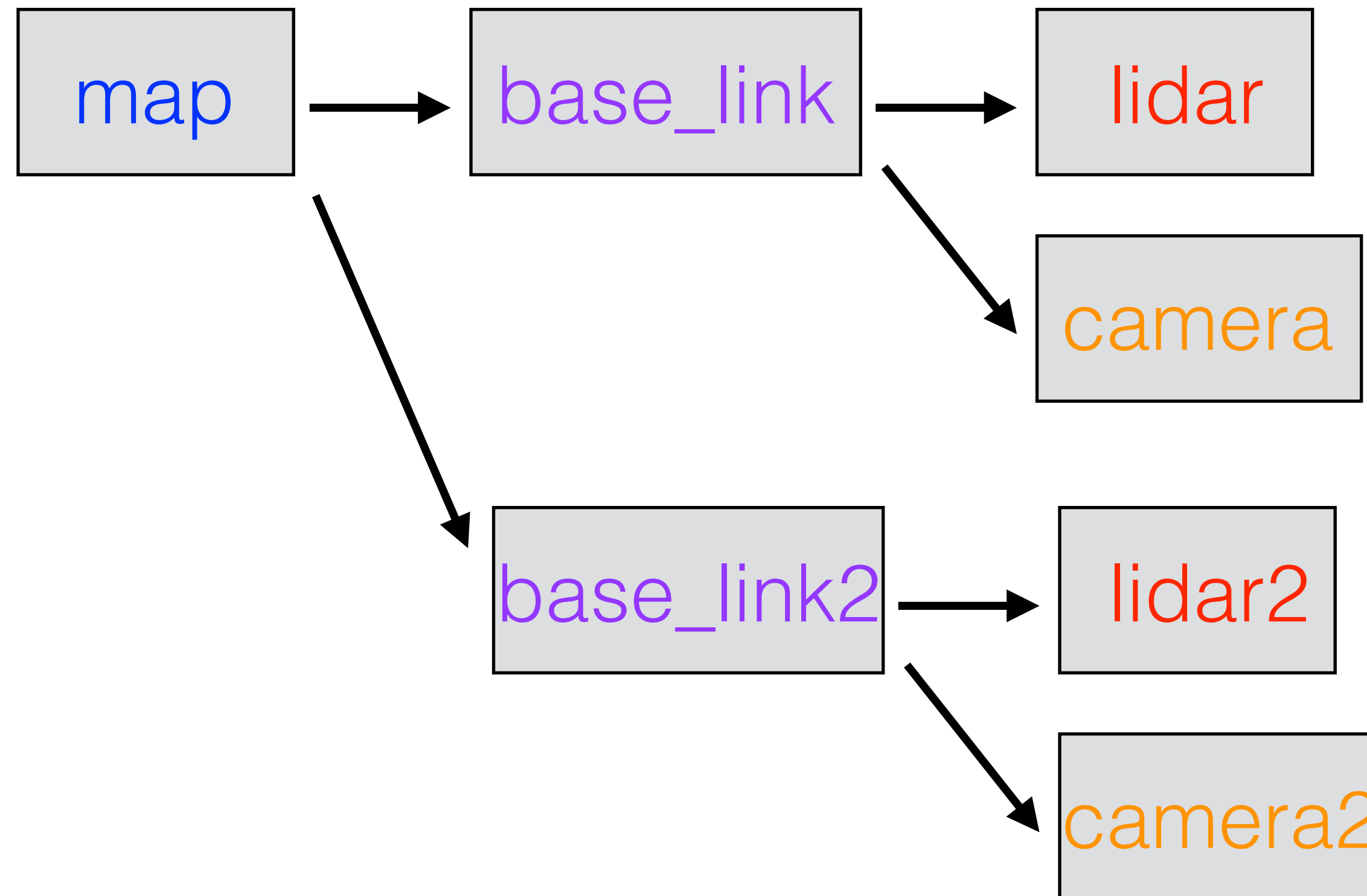
Coordinate frames & ROS

- Coordinate frames in ROS form a transformation tree: `tf`



Coordinate frames & ROS

- Transformation tree for two robots allow to estimate mutual position and use measurement from other robot.



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

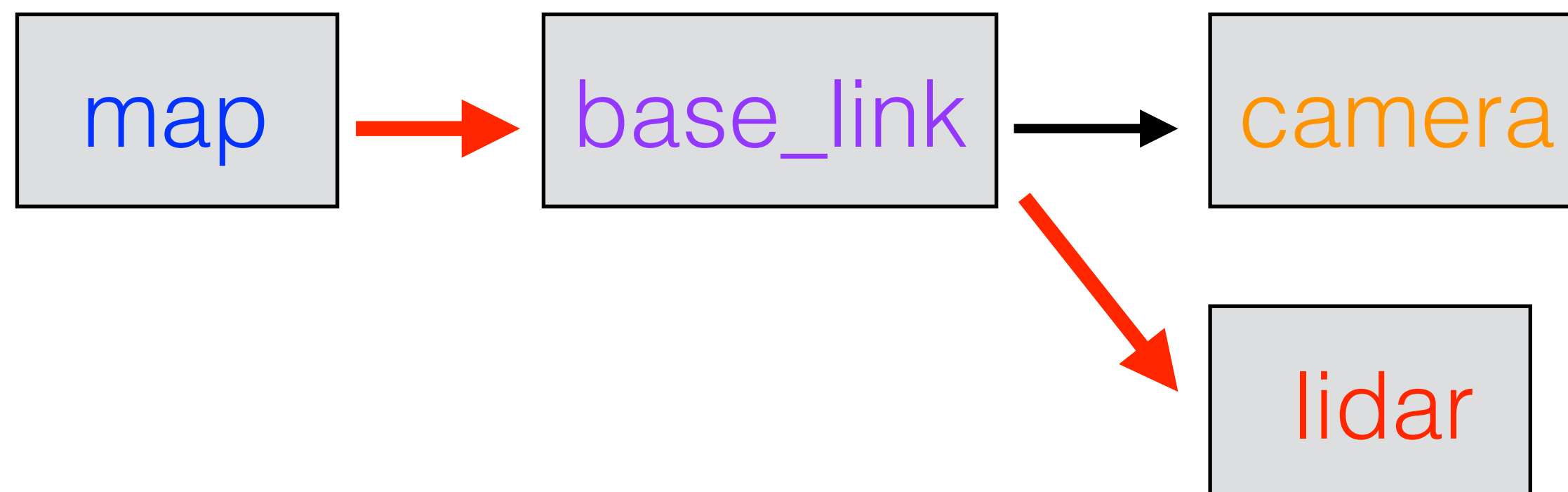
(1) in your own node:

```
# initialize listener (10 sec buffer)
buffer = tf2_ros.Buffer()
listener = tf2_ros.TransformListener(buffer)

# estimate transformation from lidar to map
transform = buffer.lookup_transform('lidar', 'map', rospy.Time())
```

(2) In the terminal:

```
$ rosrun tf tf_echo /map /lidar
At time 1263248513.809
- Translation: [2.398, 6.783, 0.000]
- Rotation: in Quaternion [0.000, 0.000, -0.707, 0.707]
```



Coordinate frames & ROS

- Coordinate frames in ROS form a transformation tree: `tf`



- In order to let ROS to keep the `tf` up-to-date
=> **parent-child transformations** between all coordinate frames need to be published by your nodes

Broadcasting static transformation between two c.f. in ROS

```
broadcaster = tf2_ros.StaticTransformBroadcaster()
transform = geometry_msgs.msg.TransformStamped()
# estimate R,t (e.g. measure or compute)
# ... "TOPIC OF FOLLOWING TWO LECTURES" => R,t
# convert rotation matrix into quaternion
q = mat2quat(R)
# fill-in transform between coordinate frames (q,t)
transform.translation.x = t[0]
transform.translation.y = t[1]
transform.translation.z = t[2]
transform.rotation.x = q[0]
transform.rotation.y = q[1]
transform.rotation.z = q[2]
transform.rotation.w = q[3]
transform.header.stamp = rospy.Time.now()
transform.header.frame_id = "base_link"
transform.child_frame_id = "lidar"

# publish transform between coordinate frames (q,t)
broadcaster.sendTransform(transform)
```

Outline

The topic of this lecture:

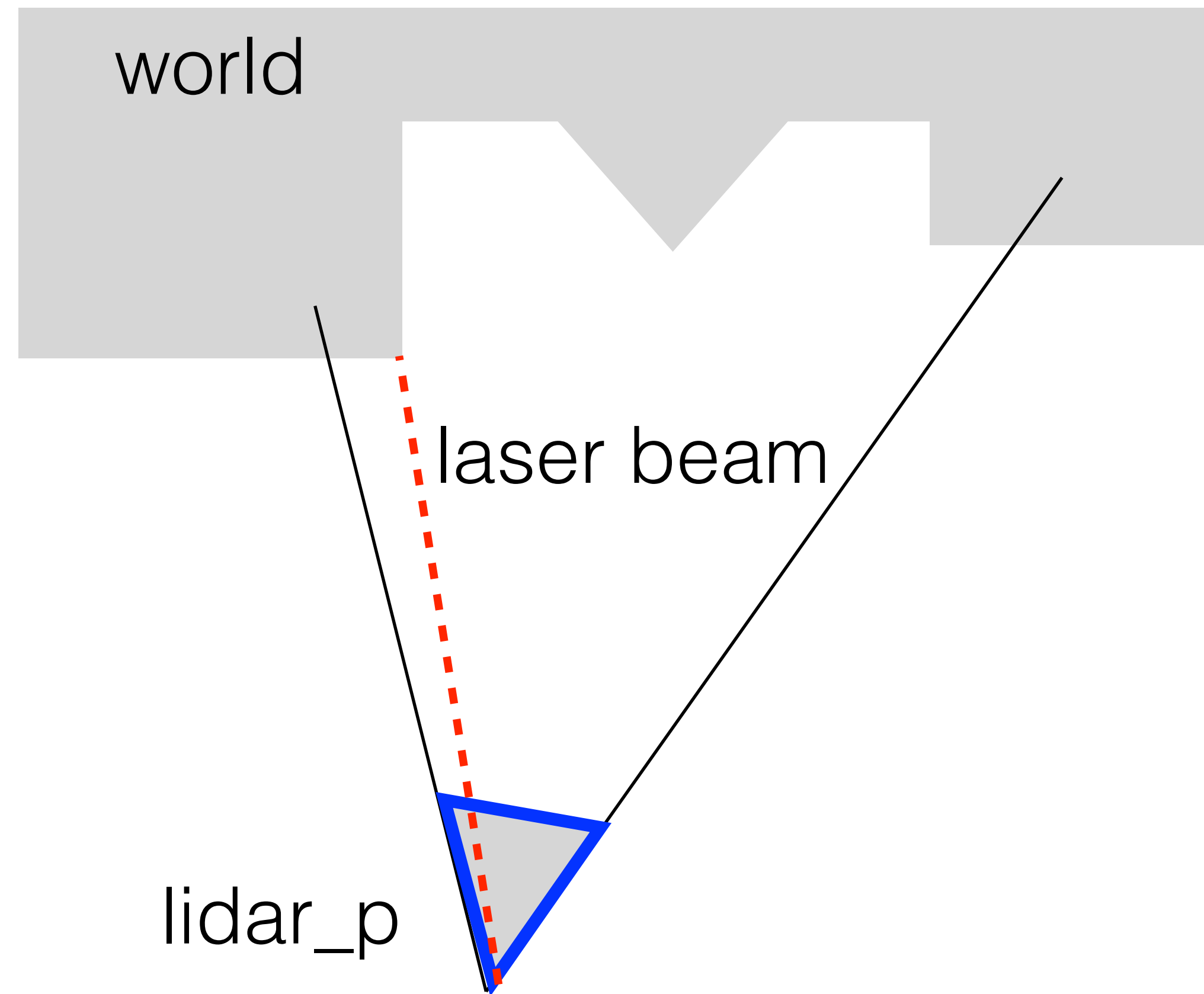
- estimating transformation between **static** coord. frames (sensor calibration)
- principle of lidar, camera, realsense, stereo ...

The topic of next lecture:

- estimating transformation between **dynamic** coord. frames (robot/sensor localization - SLAM)

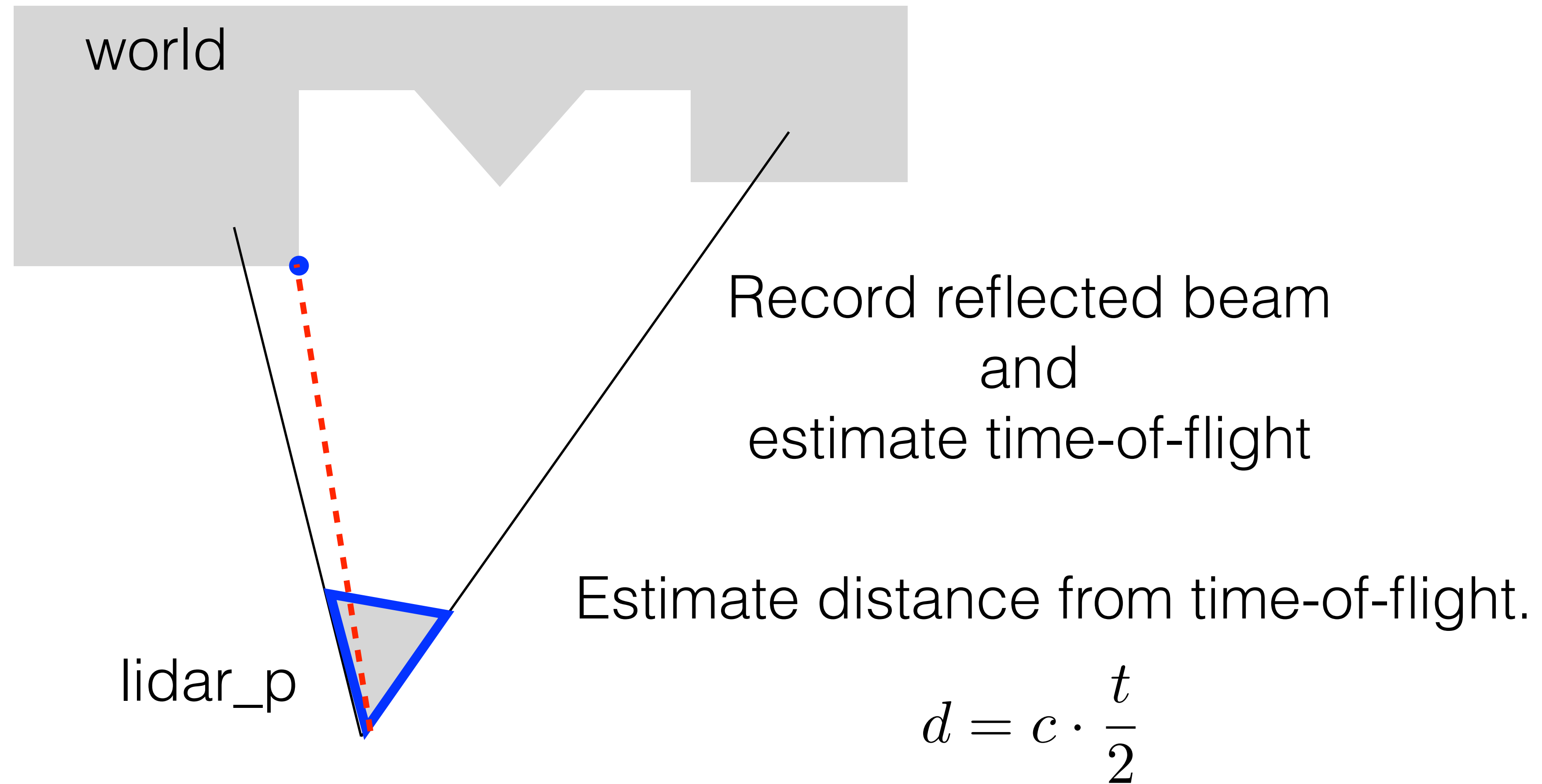
Lidar

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



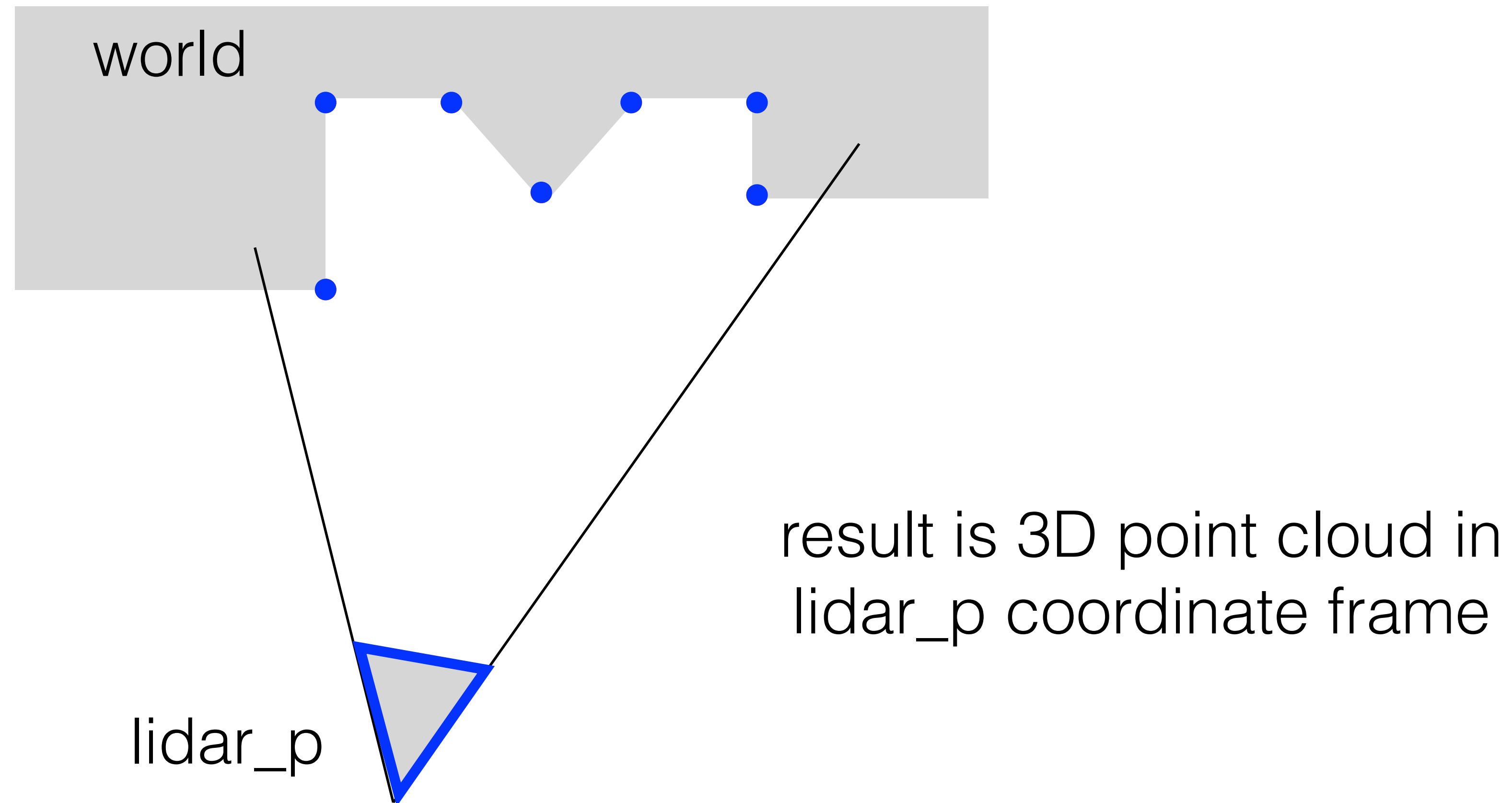
Lidar

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



Lidar

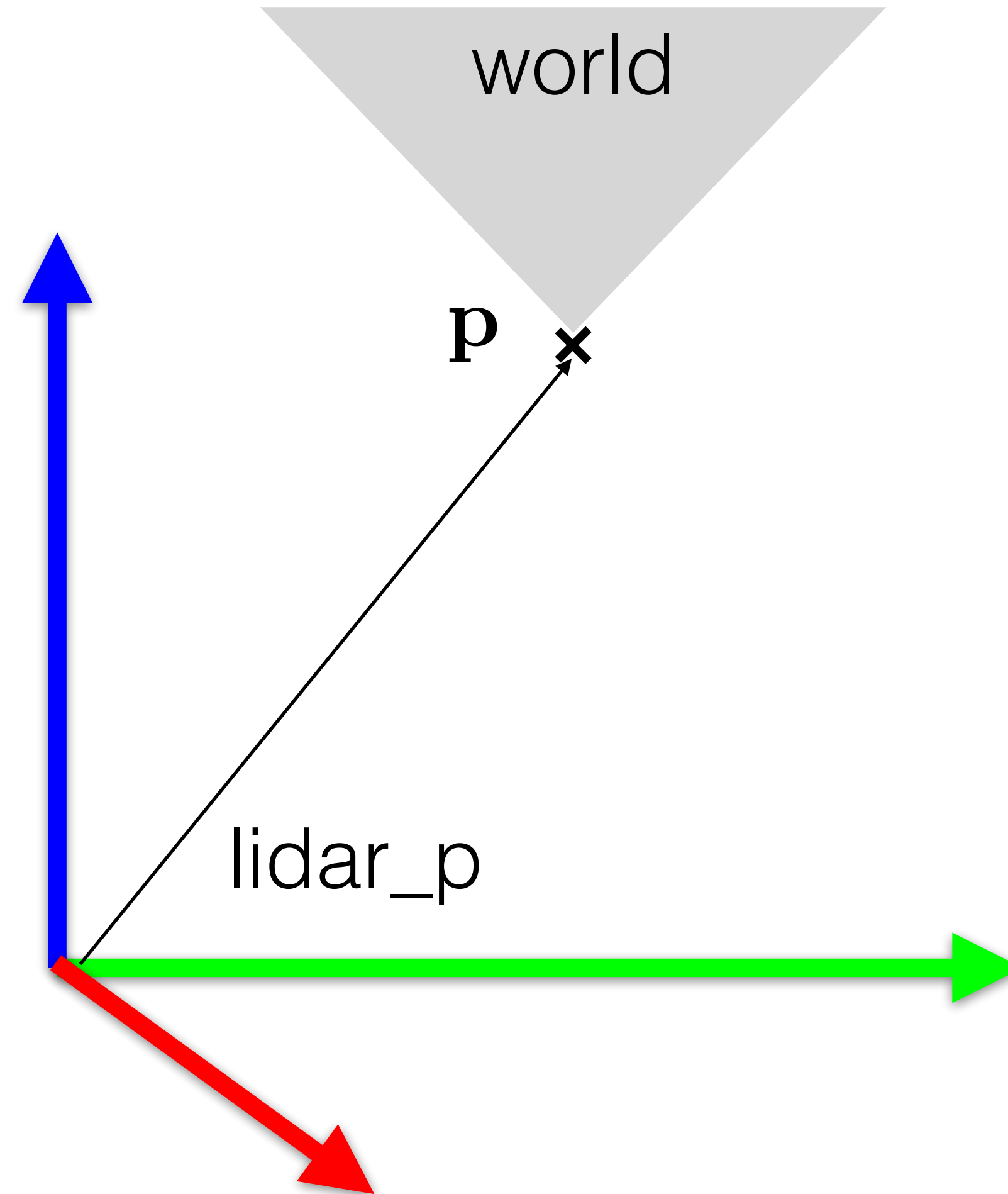
Lidar is device which measure depth of some points in its field-of view by time-of-flight principle.



Euclidean transformation of a rigid body

Let us now work only with two coordinate frames:

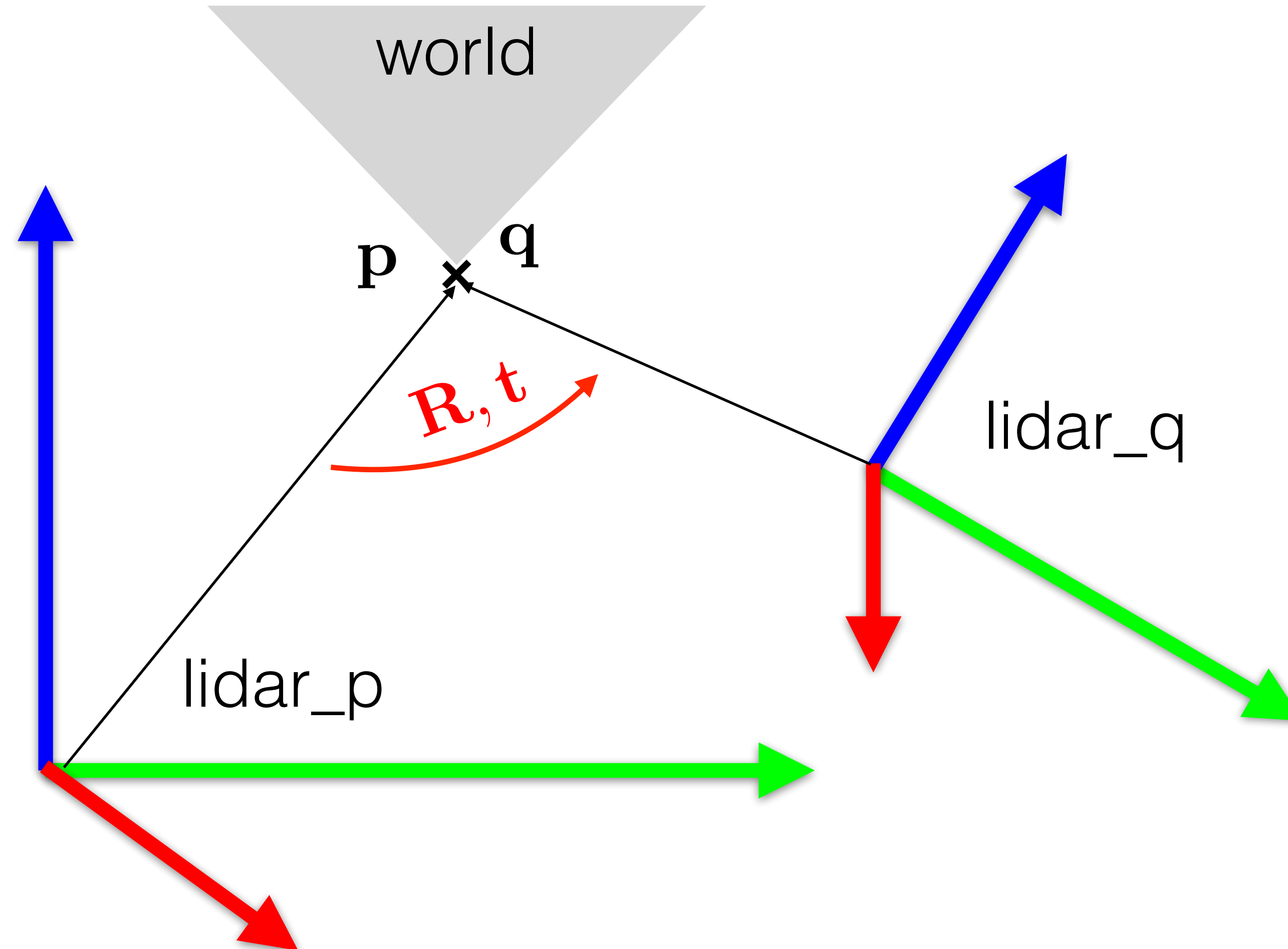
- lidar_p in which points are denoted as $\mathbf{p} \in \mathcal{R}^3$



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 $\mathbf{q} \in \mathcal{R}^3$

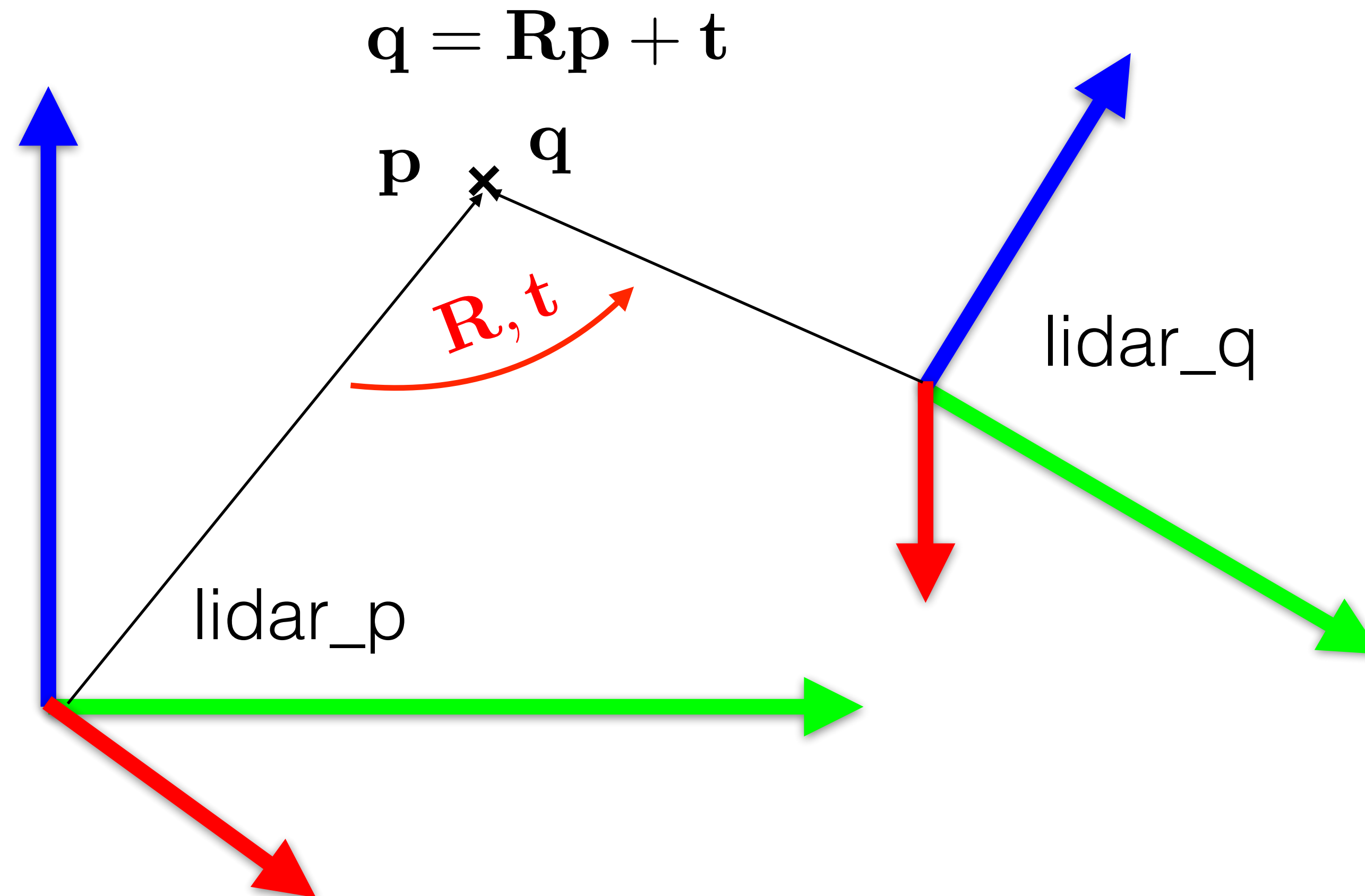


Euclidean transformation of a rigid body

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 $\mathbf{q} \in \mathcal{R}^3$

Transformation between measurements uniquely determined:



Summary: Euclidean transformation of a rigid body

- Euclidean transformation $\mathbf{q} = \mathbf{R}\mathbf{p} + \mathbf{t}$

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

- Euclidean transformation is given by matrix $\bar{\mathbf{q}} = \mathbf{g} \bar{\mathbf{p}}$

where $\mathbf{g} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$ $\bar{\mathbf{p}} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}$ $\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ 1 \end{bmatrix}$

- Set of all transformations forms Special Euclidean group

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

- Direct rotation representation is often undesirable (e.g. optimization of 3x3 matrix with determinant-induced constraints => 3-dim manifold in 9-dim space)
- Alternative representations are often used.

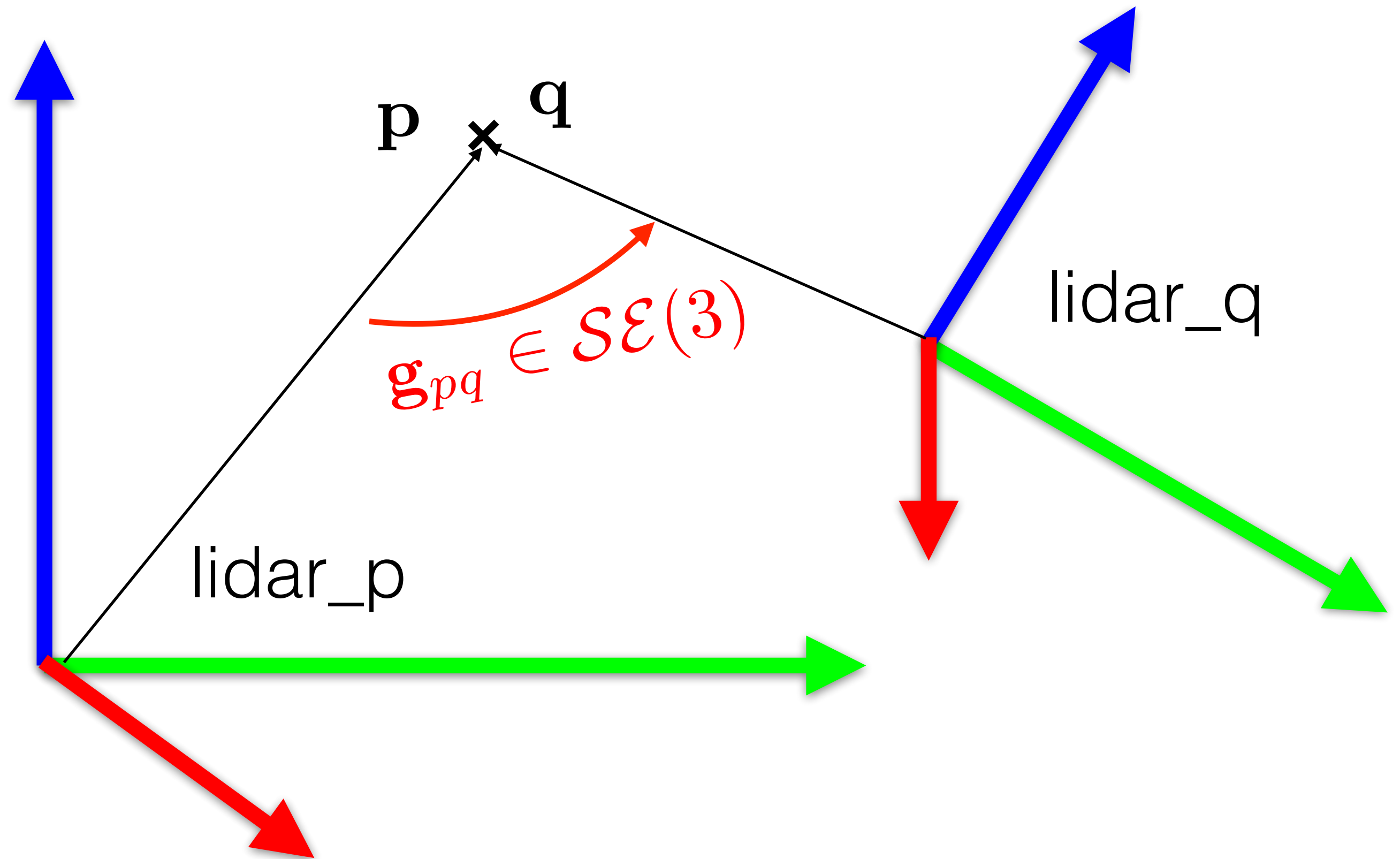
Euclidean transformation of a rigid body

Example 0:

Given transformation from lidar_p to lidar_q \mathbf{g}_{pq} ,
what is inverse transformation \mathbf{g}_{qp} ?

$$\mathbf{g}_{pq} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 000 & 1 \end{bmatrix}$$

$$\mathbf{g}_{qp} = ?$$



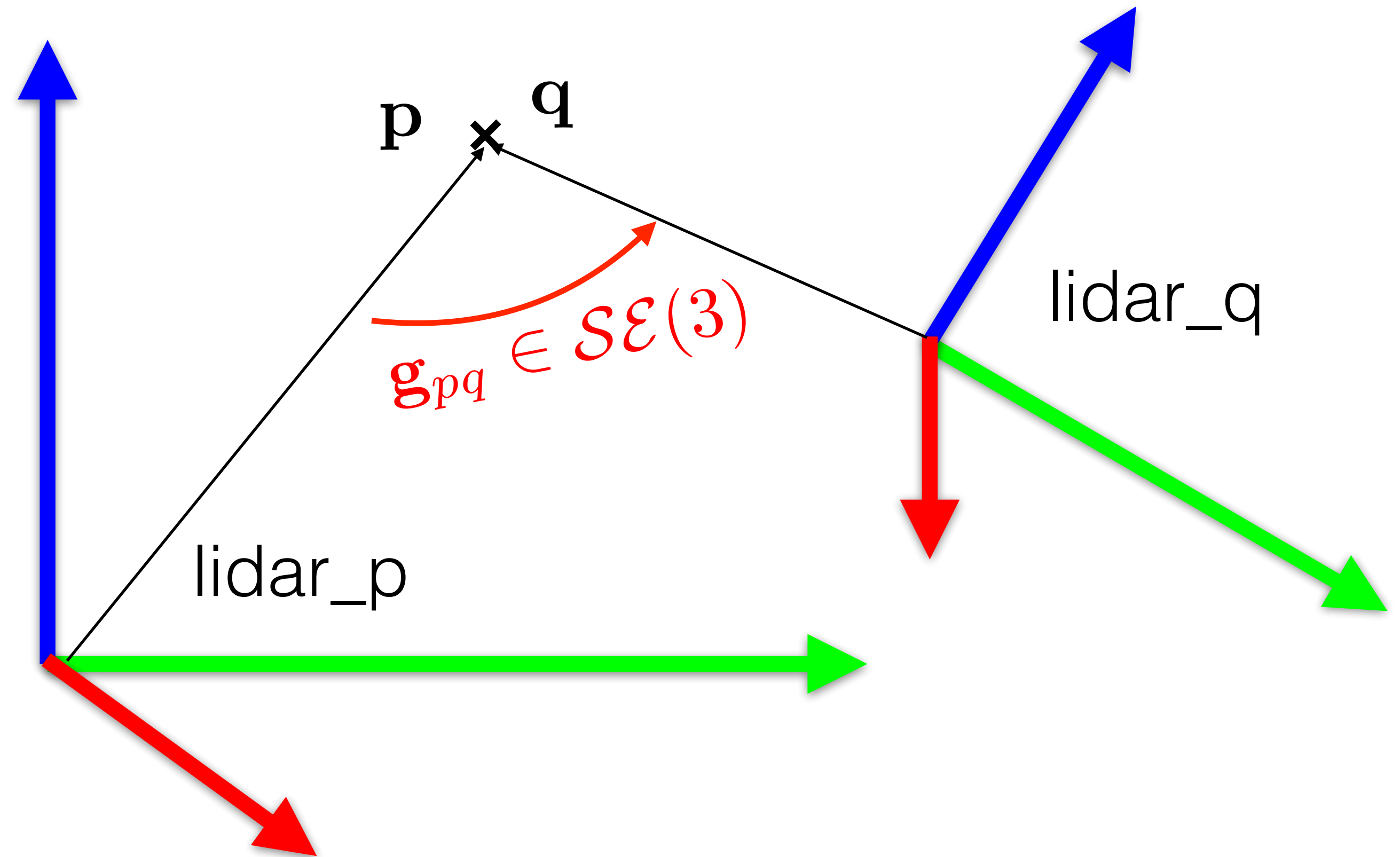
Euclidean transformation of a rigid body

Example 0:

Given transformation from lidar_p to lidar_q \mathbf{g}_{pq} ,
what is inverse transformation \mathbf{g}_{qp} ?

$$\mathbf{g}_{pq} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 000 & 1 \end{bmatrix}$$

$$\mathbf{g}_{qp} = \begin{bmatrix} \mathbf{R}^\top & -\mathbf{R}^\top \mathbf{t} \\ 000 & 1 \end{bmatrix}$$



Different representations of the rotation: Euler angles

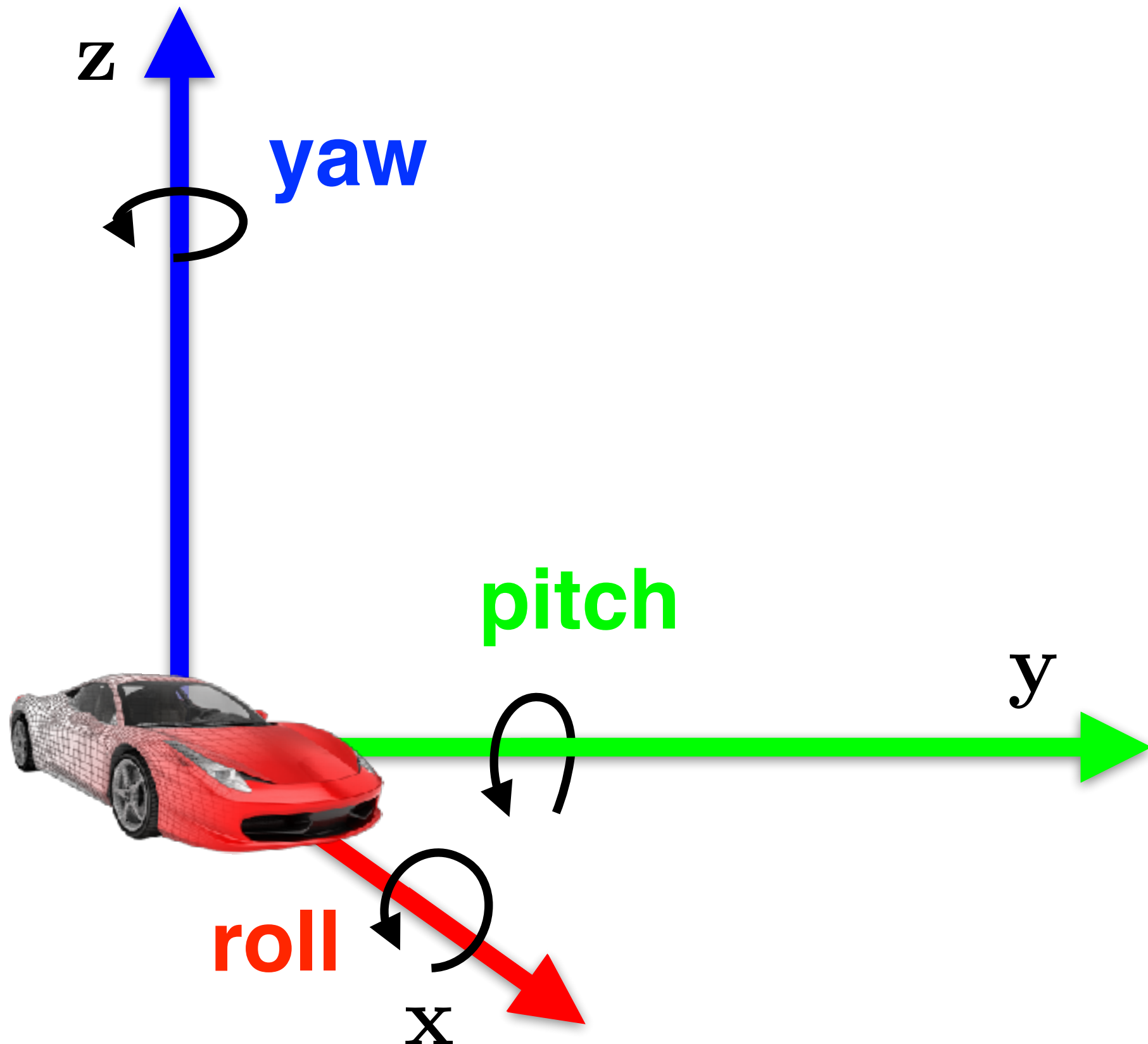
Any rotation can be achieved by 3 successive rotations around axes of coord. s.

1. roll: $\mathbf{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$

2. pitch: $\mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$

3. yaw: $\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma)$$



Different representations of the rotation: Euler angles

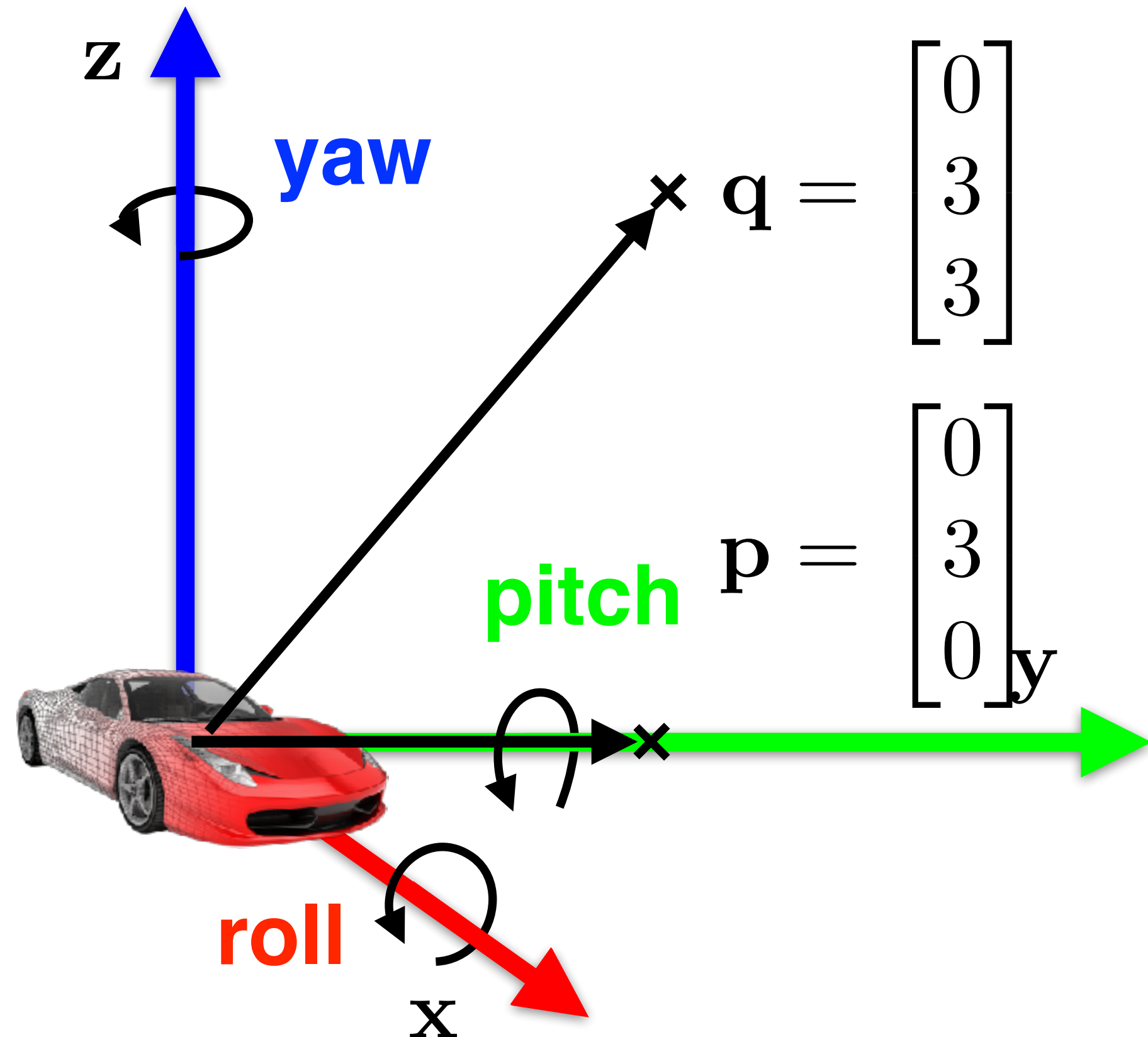
Any rotation can be achieved by 3 successive rotations around axes of coord. s.

Example 1:

What is 90-rotation around axis y?

What does this rotation preserve?

What is the meaning of R-columns?



1. roll: $\mathbf{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$

2. pitch: $\mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$

3. yaw: $\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma)$$

Different representations of the rotation: Euler angles

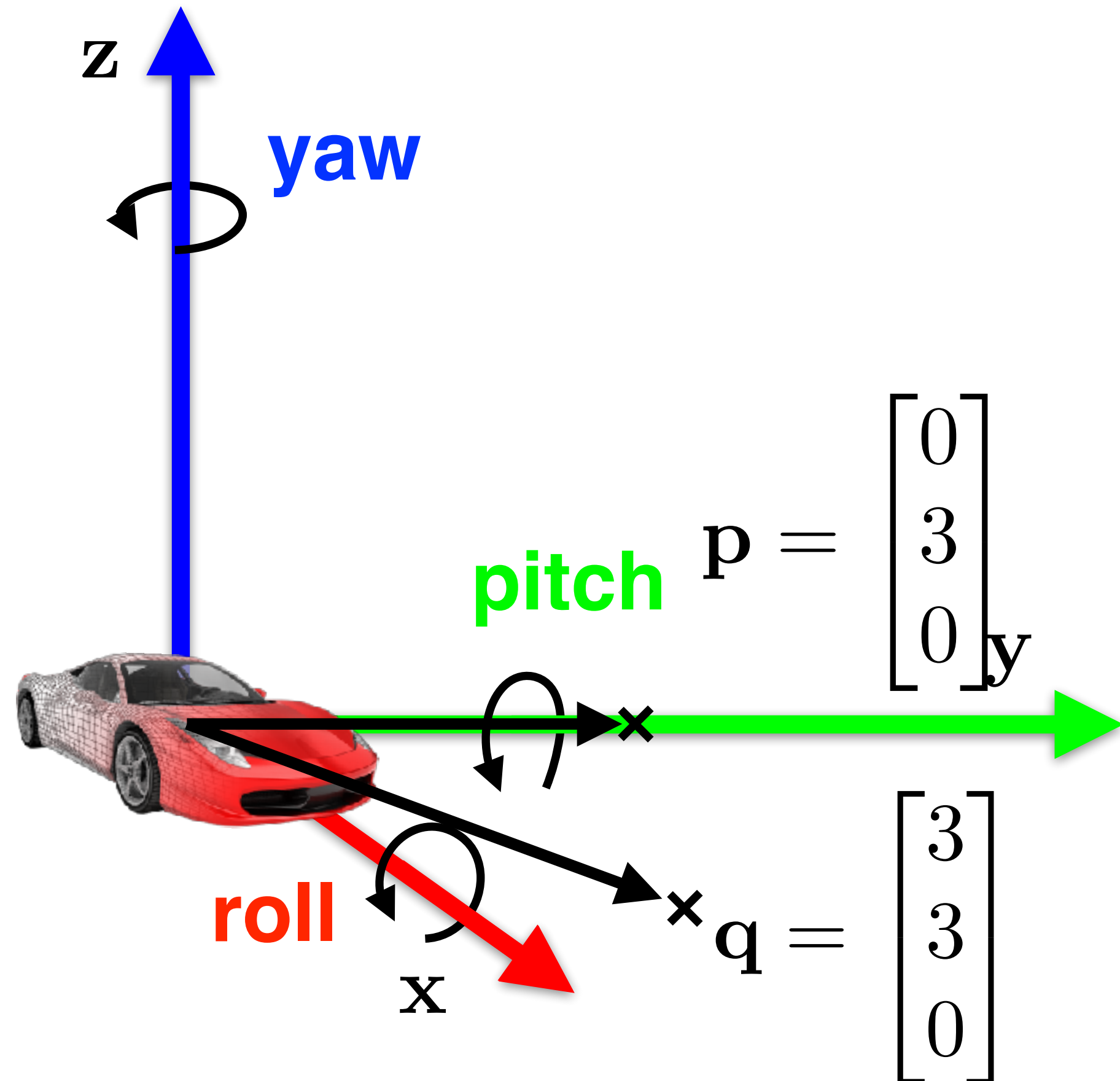
Any rotation can be achieved by 3 successive rotations around axes of coord. s.

Example 1:

What is 90-rotation around axis y?

What does this rotation preserve?

What is the meaning of R-columns?



1. roll: $\mathbf{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$

2. pitch: $\mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$

3. yaw: $\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

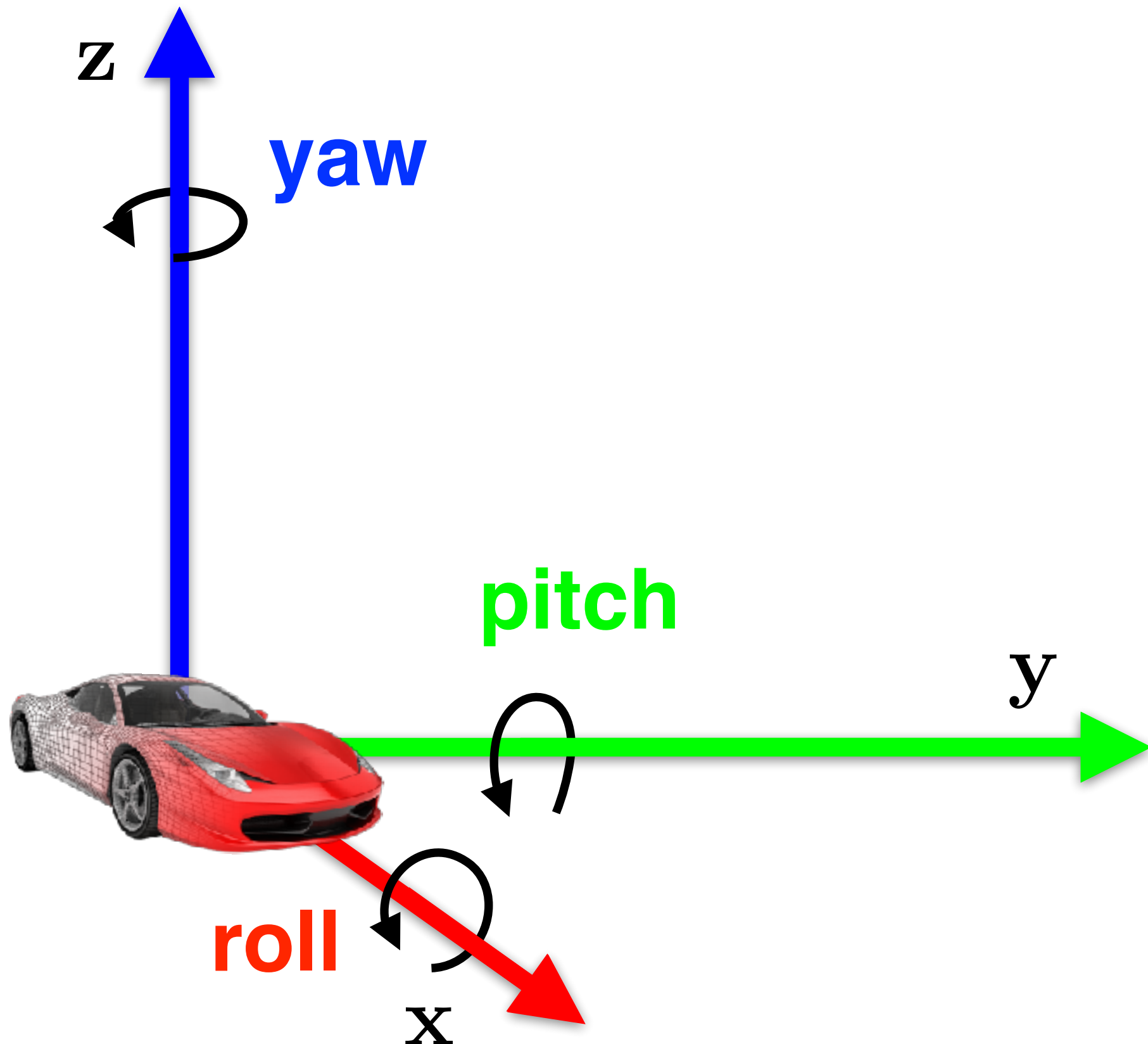
$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma)$$

Different representations of the rotation: Euler angles

Any rotation can be achieved by 3 successive rotations around axes of coord. s.

roll **pitch** **yaw**

Gimbal lock: $\begin{bmatrix} -90, & 90, & 0 \\ 0, & 90, & 90 \end{bmatrix}$



1. roll: $\mathbf{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$

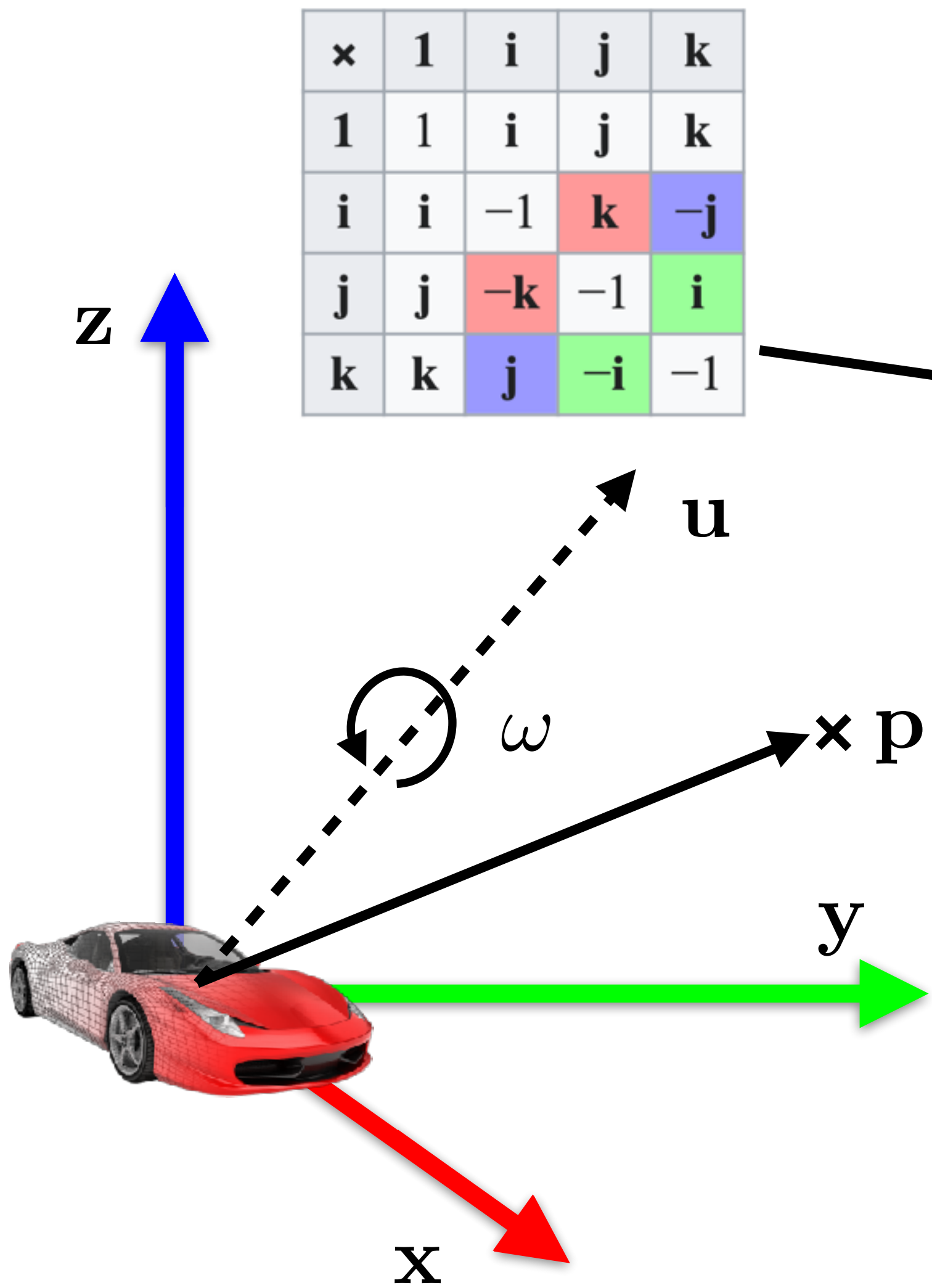
2. pitch: $\mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$

3. yaw: $\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma)$$

Different representations of the rotation: Quaternions

Euler rotation theorem: Any sequence of rotations is equivalent to single rotation around fixed axis.



$$\mathbf{p} \in \mathbb{R}^3$$

$$\mathbf{p} = (p_x, p_y, p_z) = p_x \mathbf{i} + p_y \mathbf{j} + p_z \mathbf{k}$$

$\mathbf{i}, \mathbf{j}, \mathbf{k}$... cartesian axes

$$\mathbb{p} \in \mathbb{H}$$

$$\mathbb{p} = (0, p_x, p_y, p_z) = 0 + p_x \hat{\mathbf{i}} + p_y \hat{\mathbf{j}} + p_z \hat{\mathbf{k}}$$

$\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$... fundamental quaternions units
such that $\hat{\mathbf{i}}^2 = \hat{\mathbf{j}}^2 = \hat{\mathbf{k}}^2 = \hat{\mathbf{i}}\hat{\mathbf{j}}\hat{\mathbf{k}} = -1$

$$\mathbb{q} \in \mathbb{H}$$

$$\mathbb{q} = \cos \frac{\omega}{2} + (u_x \hat{\mathbf{i}} + u_y \hat{\mathbf{j}} + u_z \hat{\mathbf{k}}) \sin \frac{\omega}{2}$$

$$\mathbb{q}^{-1} \in \mathbb{H}$$

$$\mathbb{q}^{-1} = \cos \frac{\omega}{2} - (u_x \hat{\mathbf{i}} + u_y \hat{\mathbf{j}} + u_z \hat{\mathbf{k}}) \sin \frac{\omega}{2}$$

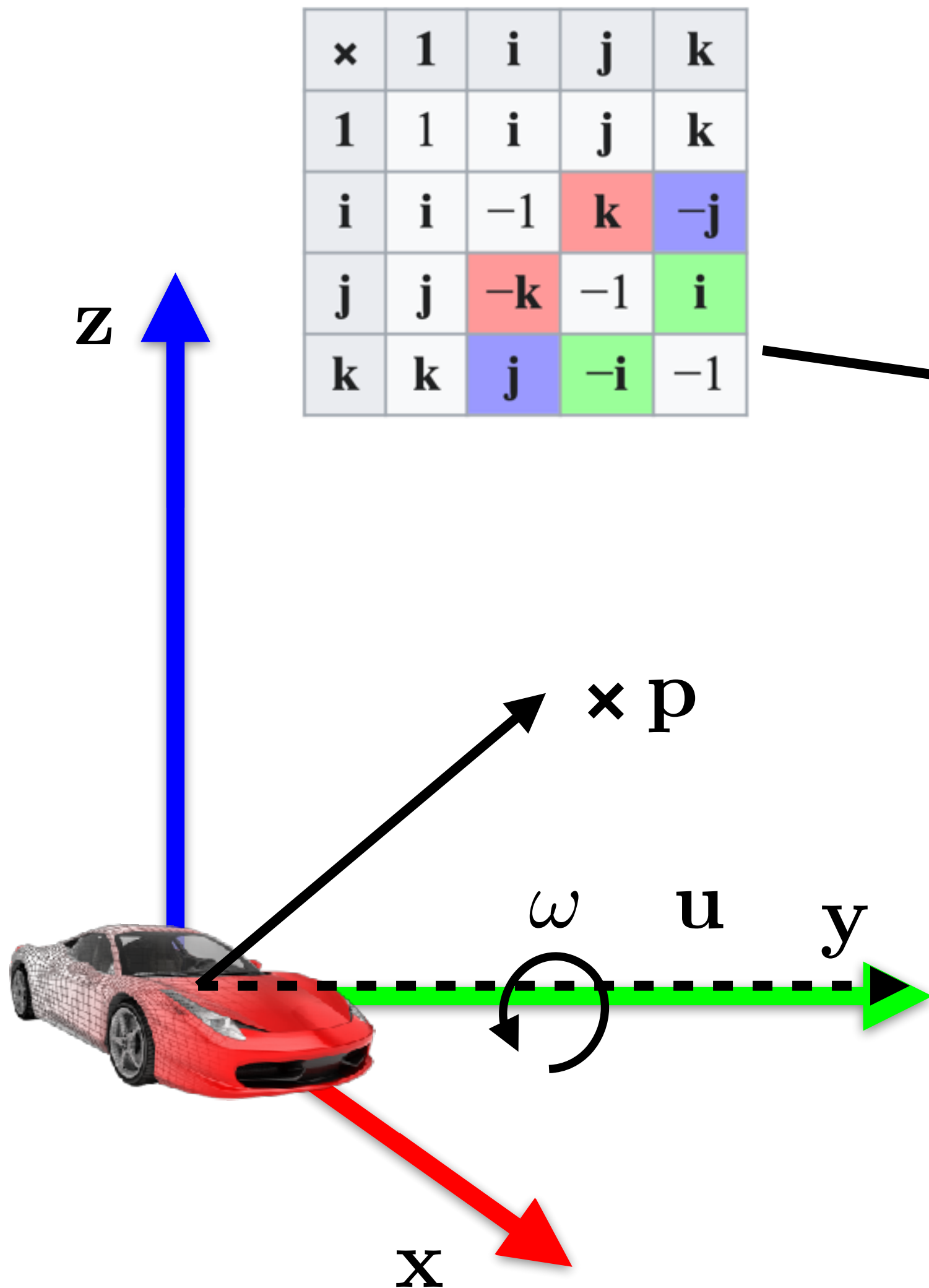
$$\mathbb{p}' \in \mathbb{H}$$

$$\mathbb{p}' = \mathbb{q}\mathbb{p}\mathbb{q}^{-1} \dots \text{rotated point in quaternion}$$

$$\mathbb{p}' = (0, p'_x, p'_y, p'_z) \in \mathbb{H} \Rightarrow \mathbf{p}' = (p'_x, p'_y, p'_z) \in \mathbb{R}^3$$

Different representations of the rotation: Quaternions

Example 2: Rotate given point $p = [0,3,3]$ around axis y by 90 deg via quaternions.



x	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

$$\mathbf{p} \in \mathbb{R}^3$$

$$\mathbf{p} = (p_x, p_y, p_z) = p_x \mathbf{i} + p_y \mathbf{j} + p_z \mathbf{k}$$

$\mathbf{i}, \mathbf{j}, \mathbf{k}$... cartesian axes

$$\mathbb{p} \in \mathbb{H}$$

$$\mathbb{p} = (0, p_x, p_y, p_z) = 0 + p_x \hat{\mathbf{i}} + p_y \hat{\mathbf{j}} + p_z \hat{\mathbf{k}}$$

$\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$... fundamental quaternions units
such that $\hat{\mathbf{i}}^2 = \hat{\mathbf{j}}^2 = \hat{\mathbf{k}}^2 = \hat{\mathbf{i}}\hat{\mathbf{j}}\hat{\mathbf{k}} = -1$

$$q \in \mathbb{H}$$

$$q = \cos \frac{\omega}{2} + (u_x \hat{\mathbf{i}} + u_y \hat{\mathbf{j}} + u_z \hat{\mathbf{k}}) \sin \frac{\omega}{2}$$

$$q^{-1} \in \mathbb{H}$$

$$q^{-1} = \cos \frac{\omega}{2} - (u_x \hat{\mathbf{i}} + u_y \hat{\mathbf{j}} + u_z \hat{\mathbf{k}}) \sin \frac{\omega}{2}$$

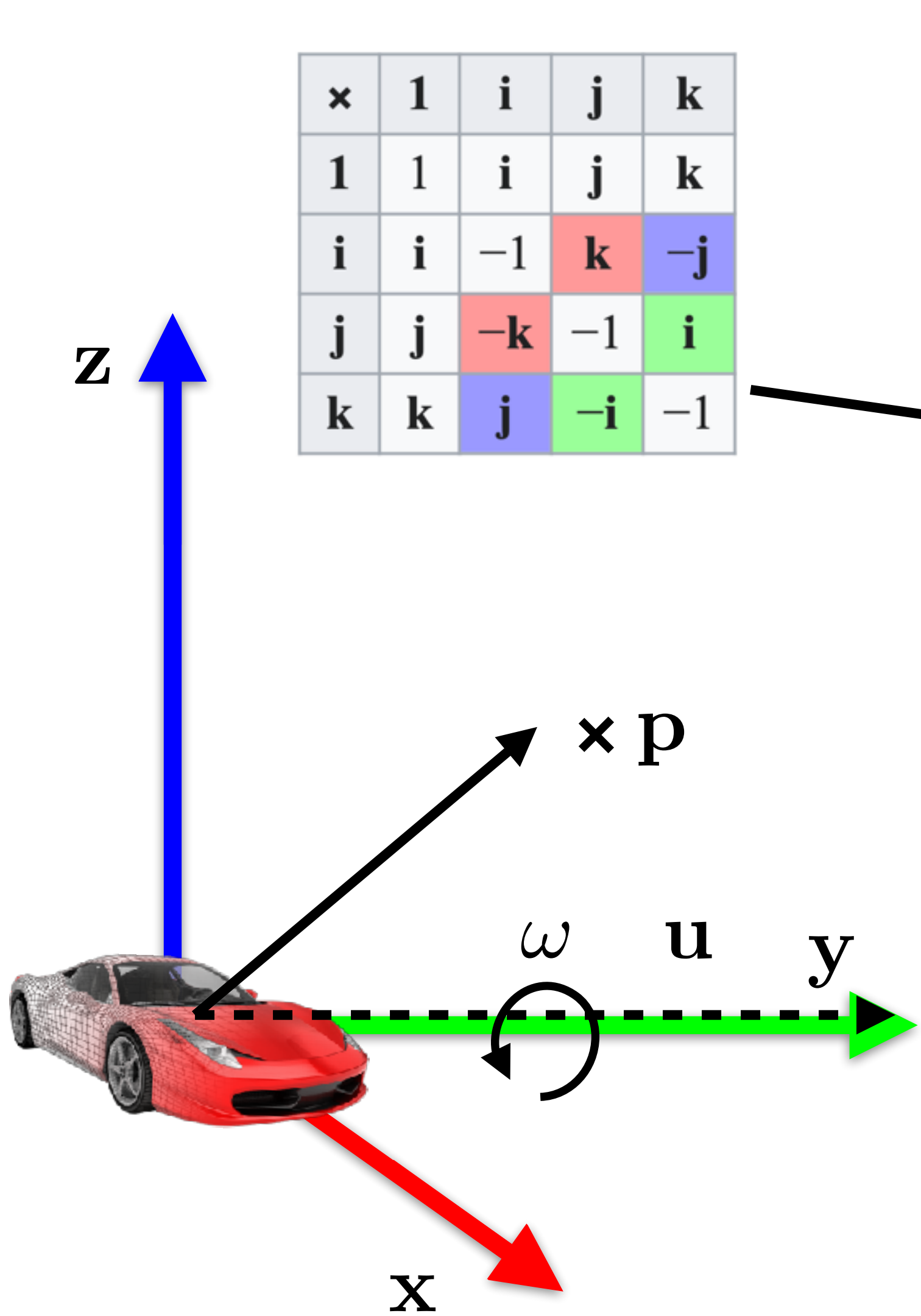
$$\mathbb{p}' \in \mathbb{H}$$

$$\mathbb{p}' = q \mathbb{p} q^{-1} \dots \text{rotated point in quaternion}$$

$$\mathbb{p}' = (0, p'_x, p'_y, p'_z) \in \mathbb{H} \Rightarrow \mathbf{p}' = (p'_x, p'_y, p'_z) \in \mathbb{R}^3$$

Different representations of the rotation: Quaternions

Example 3: Show that successive rotations of p by q_1, q_2 is eq. to rot. by $(q_1.q_2)$



x	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

$$p \in \mathbb{R}^3$$

$$p = (p_x, p_y, p_z) = p_x \mathbf{i} + p_y \mathbf{j} + p_z \mathbf{k}$$

$\mathbf{i}, \mathbf{j}, \mathbf{k}$... cartesian axes

$$p \in \mathbb{H}$$

$$p = (0, p_x, p_y, p_z) = 0 + p_x \hat{\mathbf{i}} + p_y \hat{\mathbf{j}} + p_z \hat{\mathbf{k}}$$

$\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$... fundamental quaternions units
such that $\hat{\mathbf{i}}^2 = \hat{\mathbf{j}}^2 = \hat{\mathbf{k}}^2 = \hat{\mathbf{i}}\hat{\mathbf{j}}\hat{\mathbf{k}} = -1$

$$q \in \mathbb{H}$$

$$q = \cos \frac{\omega}{2} + (u_x \hat{\mathbf{i}} + u_y \hat{\mathbf{j}} + u_z \hat{\mathbf{k}}) \sin \frac{\omega}{2}$$

$$q^{-1} \in \mathbb{H}$$

$$q^{-1} = \cos \frac{\omega}{2} - (u_x \hat{\mathbf{i}} + u_y \hat{\mathbf{j}} + u_z \hat{\mathbf{k}}) \sin \frac{\omega}{2}$$

$$p' \in \mathbb{H}$$

$$p' = qpq^{-1} \dots \text{rotated point in quaternion}$$

$$p' = (0, p'_x, p'_y, p'_z) \in \mathbb{H} \Rightarrow p' = (p'_x, p'_y, p'_z) \in \mathbb{R}^3$$

Summary: Rotation parameterizations

(1) Rotation matrix $\mathbf{R} \in \mathcal{SO}(3)$

$$\mathbf{p}'(\mathbf{R}) = \mathbf{R}\mathbf{p} \quad \text{with constrain of } \mathbf{R} \in \mathcal{SO}(3)$$

- Special orthogonal group

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

(2) Euler angles $(\alpha, \beta, \gamma) \in [0; 2\pi]^3$

$$\mathbf{p}'(\alpha, \beta, \gamma) = \mathbf{R}(\alpha, \beta, \gamma)\mathbf{p}$$

(3) Quaternions $\mathbf{q} \in \mathbb{H}$

$$\mathbf{p}'(\mathbf{q}) = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$$

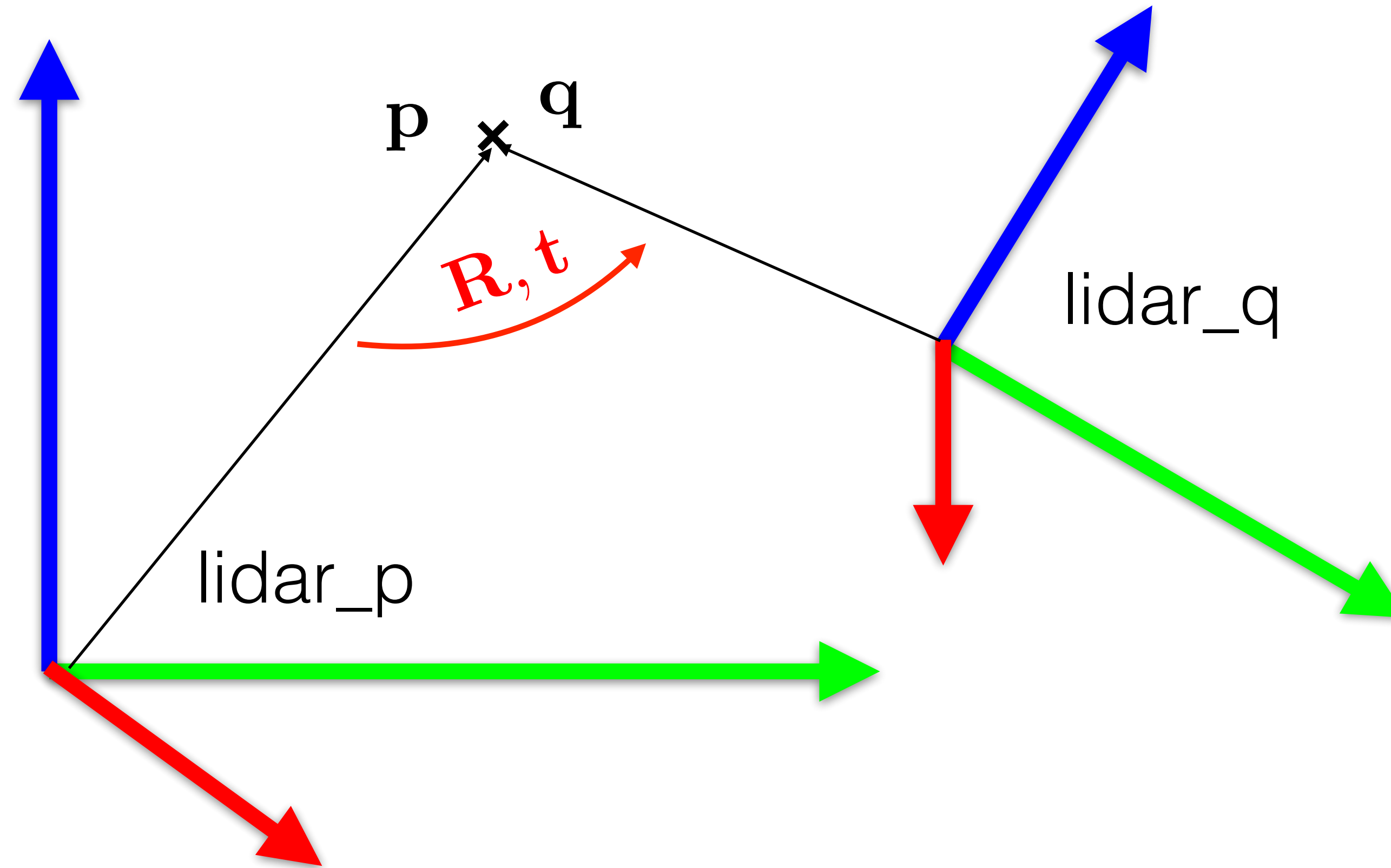
```
from scipy.spatial.transform import Rotation as Rot
Rot.from_matrix(R).as_euler('xyz')
Rot.from_quat(q).to_matrix('xyz')
```

Summary

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

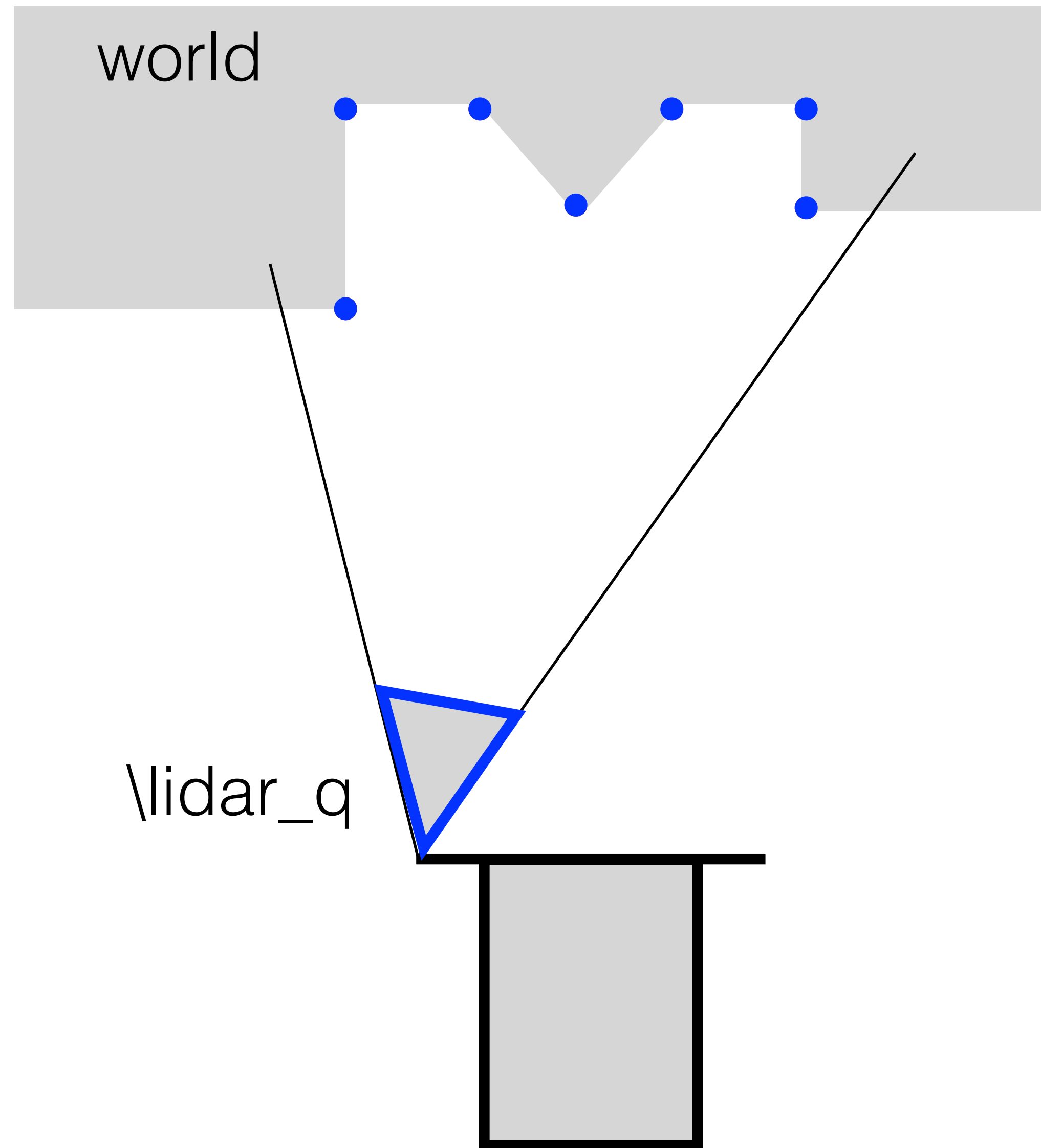
is set of transformations, which model spatio-temporal change of coordinate frames among sensors, actuators and world map.

Mutual calibration of two coordinate frames

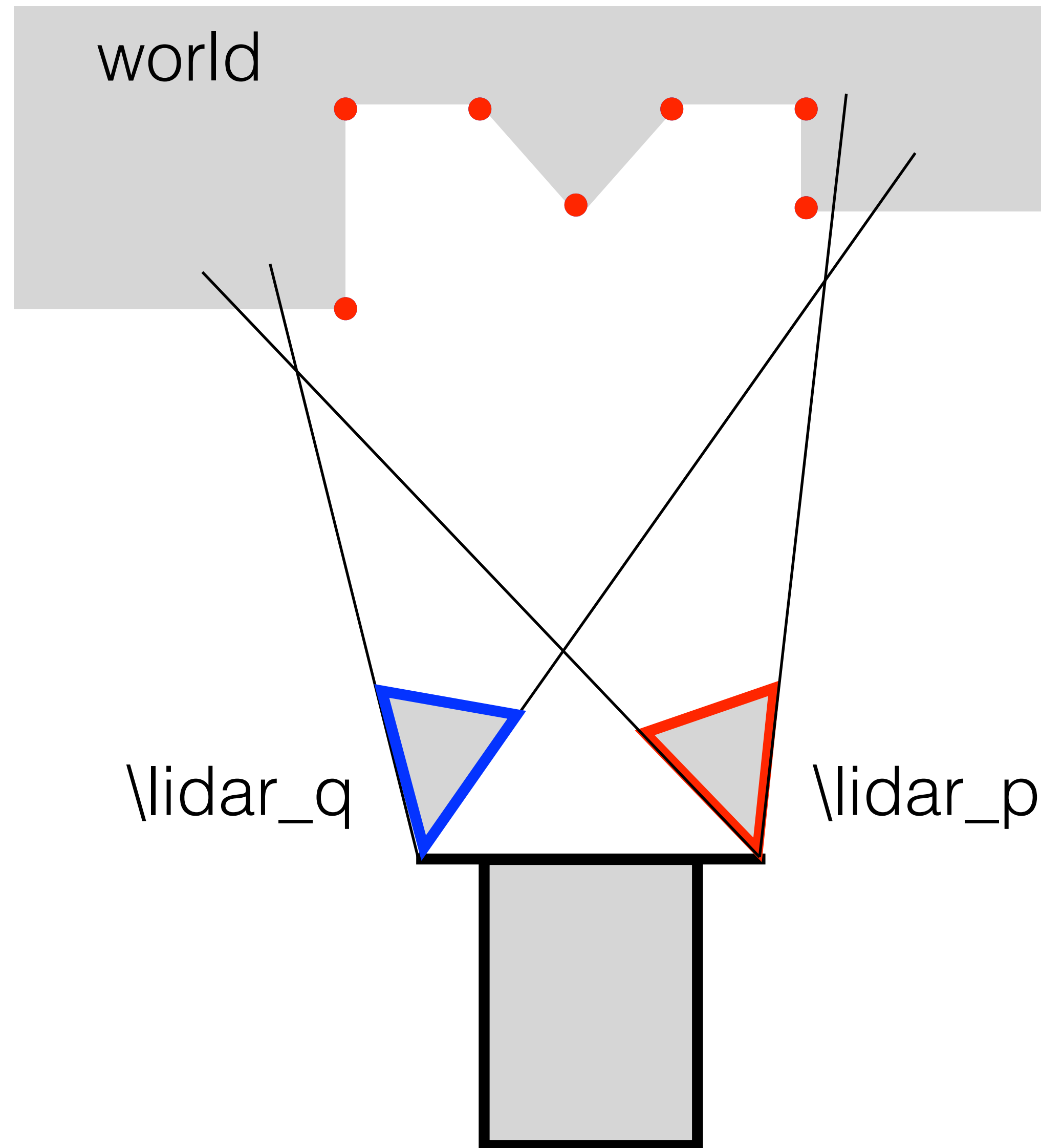


- How do we estimate \mathbf{R}, \mathbf{t} ?

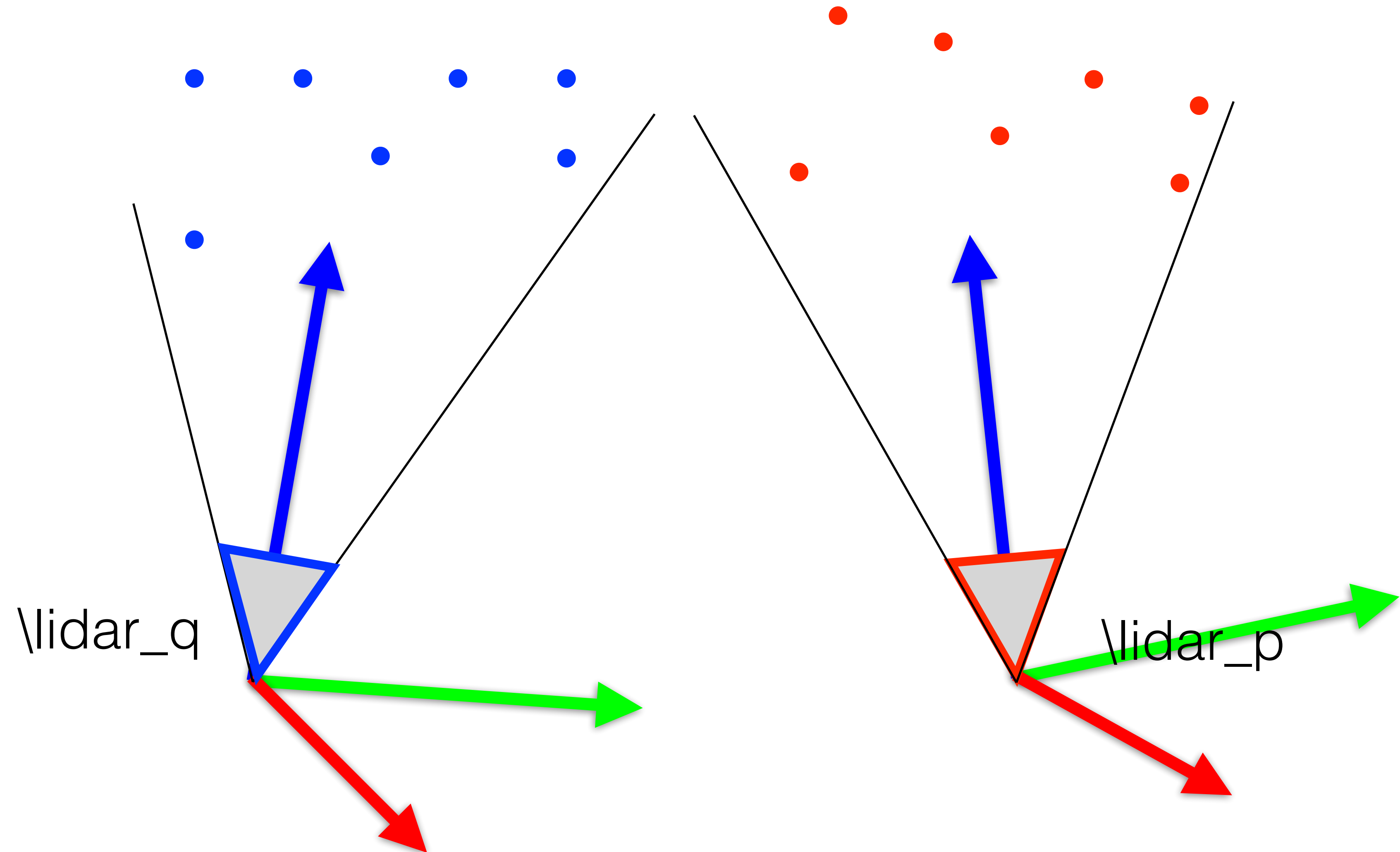
Mutual calibration of two coordinate frames



Mutual calibration of two coordinate frames

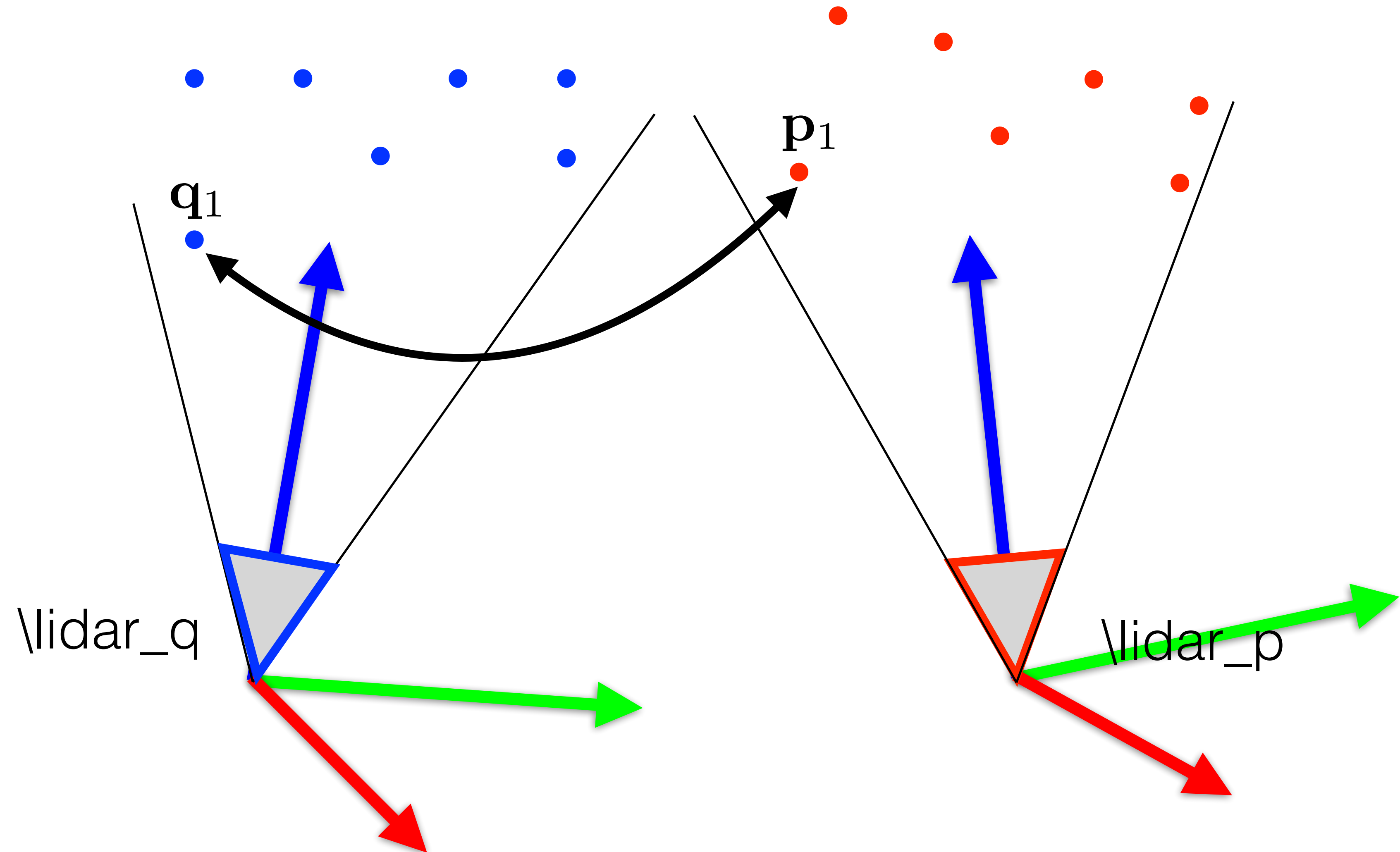


Mutual calibration of two coordinate frames



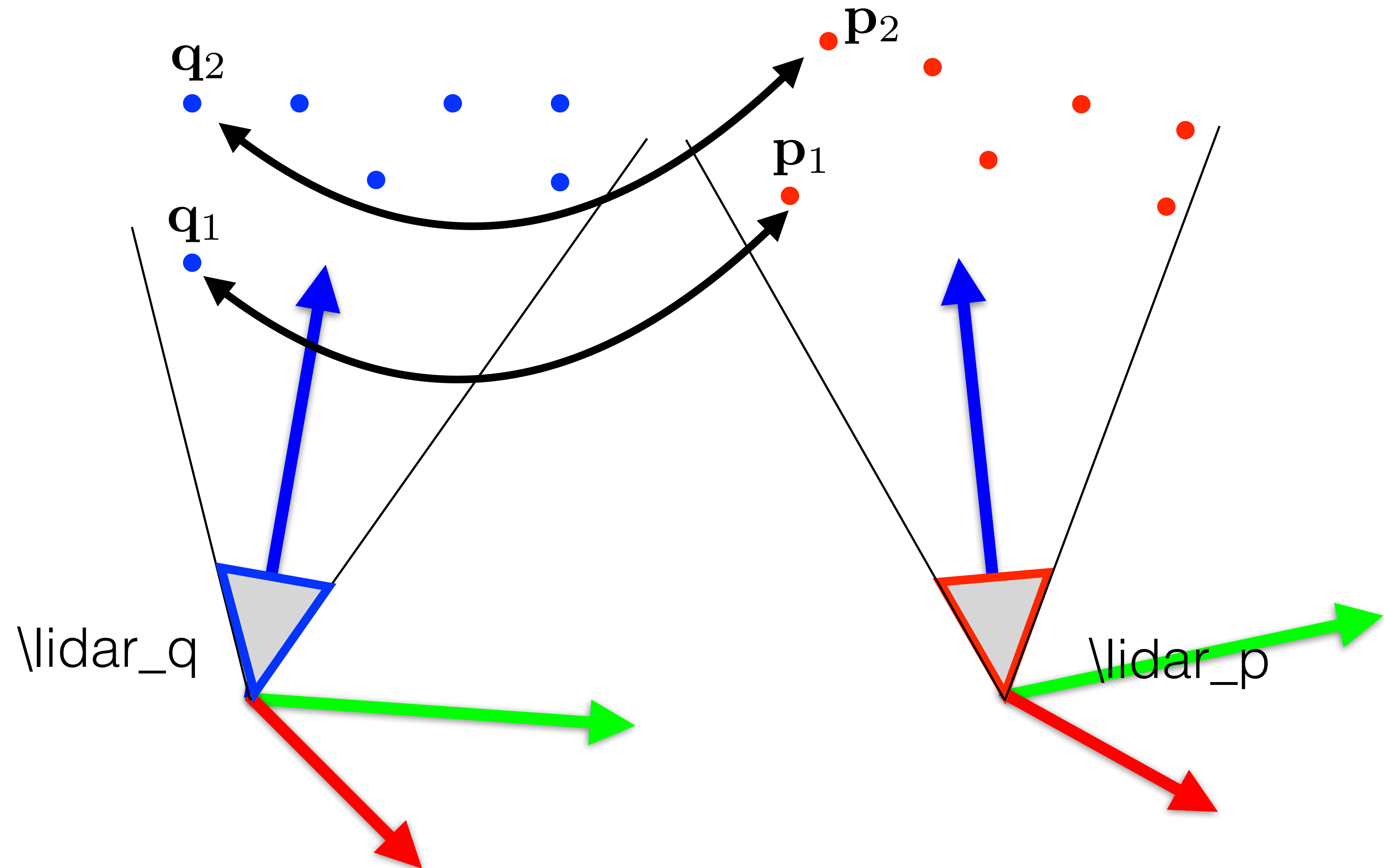
Mutual calibration of two coordinate frames

3D-3D correspondences

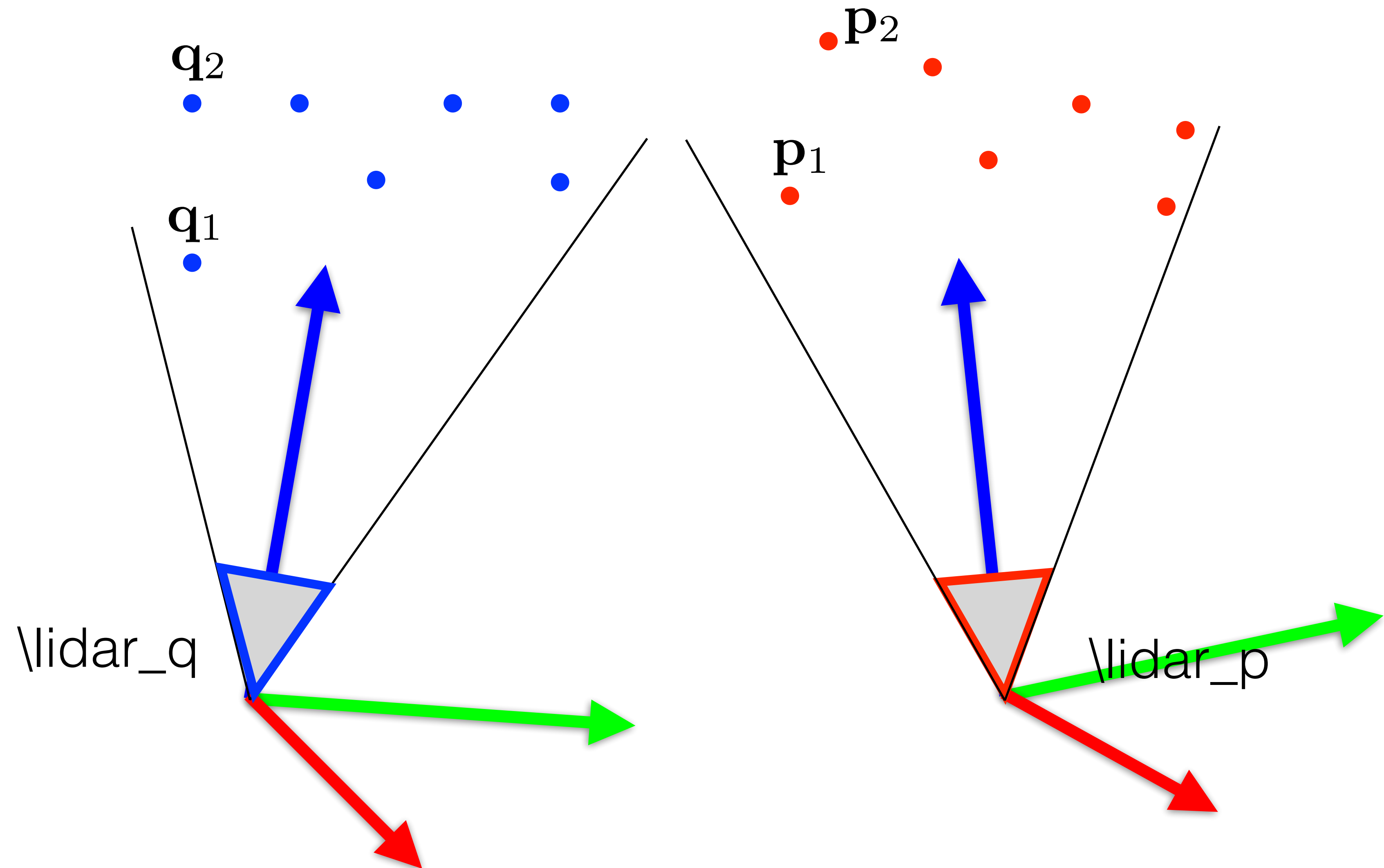


Mutual calibration of two coordinate frames

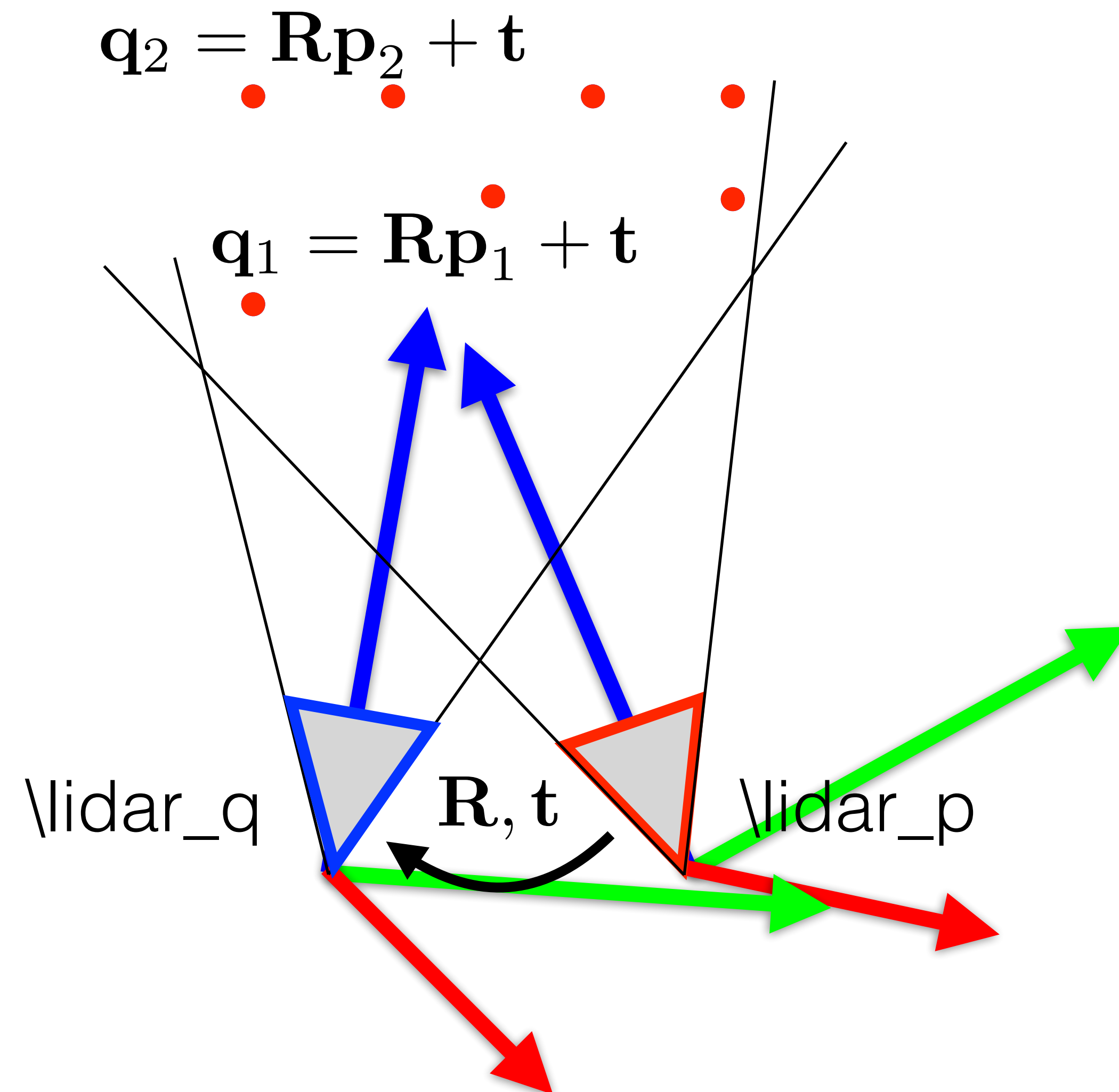
3D-3D correspondences



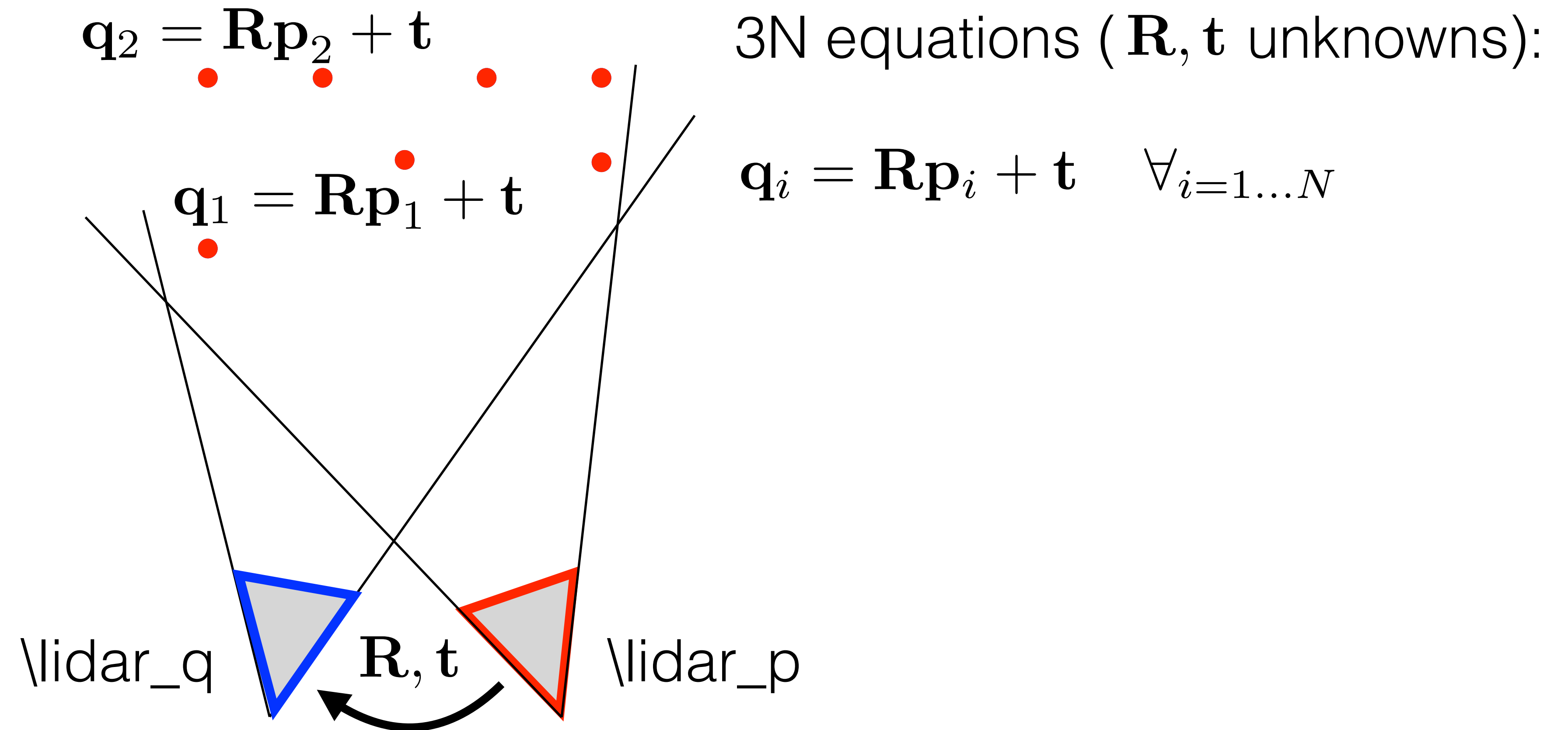
Mutual calibration of two coordinate frames



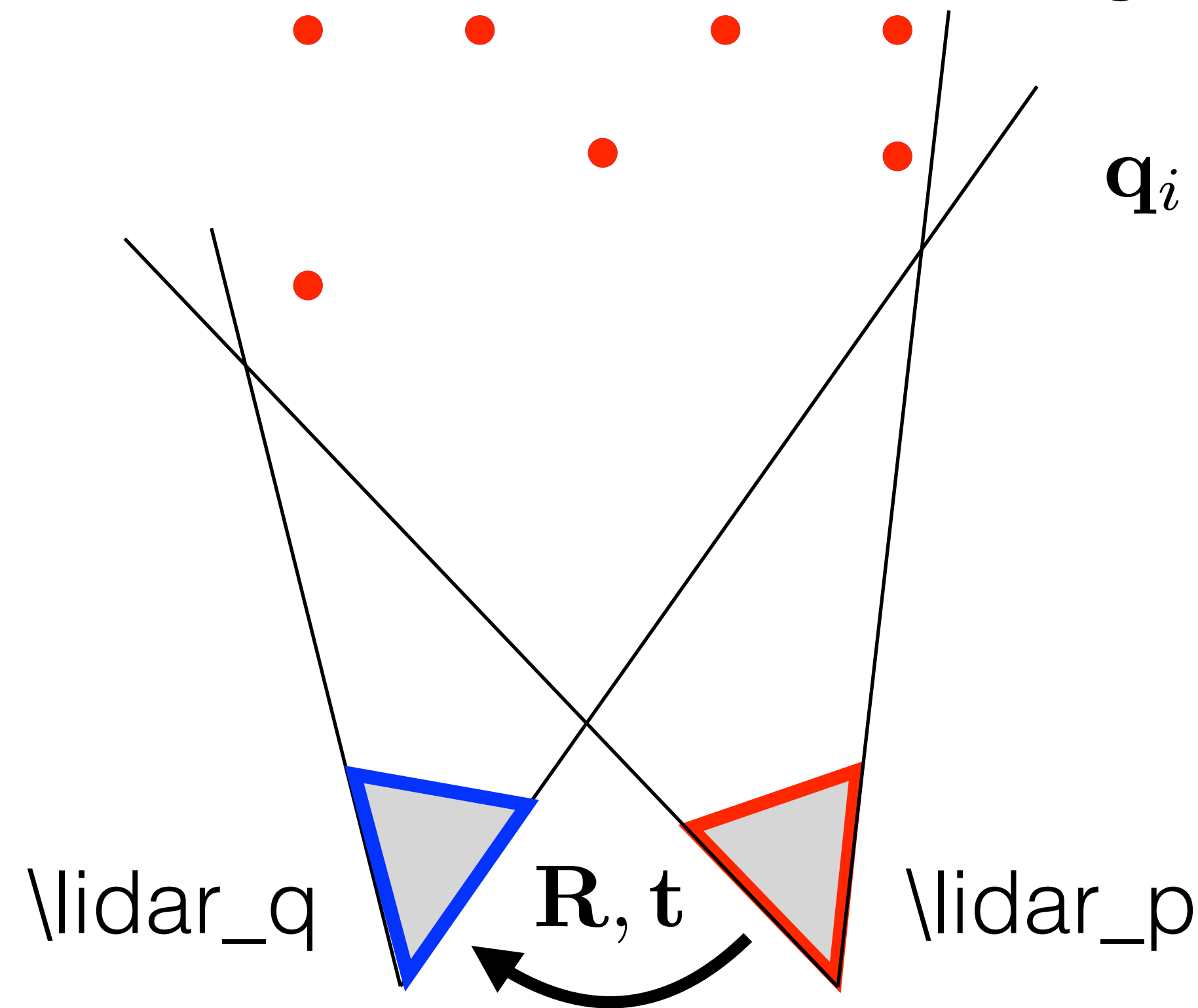
Mutual calibration of two coordinate frames



Mutual calibration of two coordinate frames



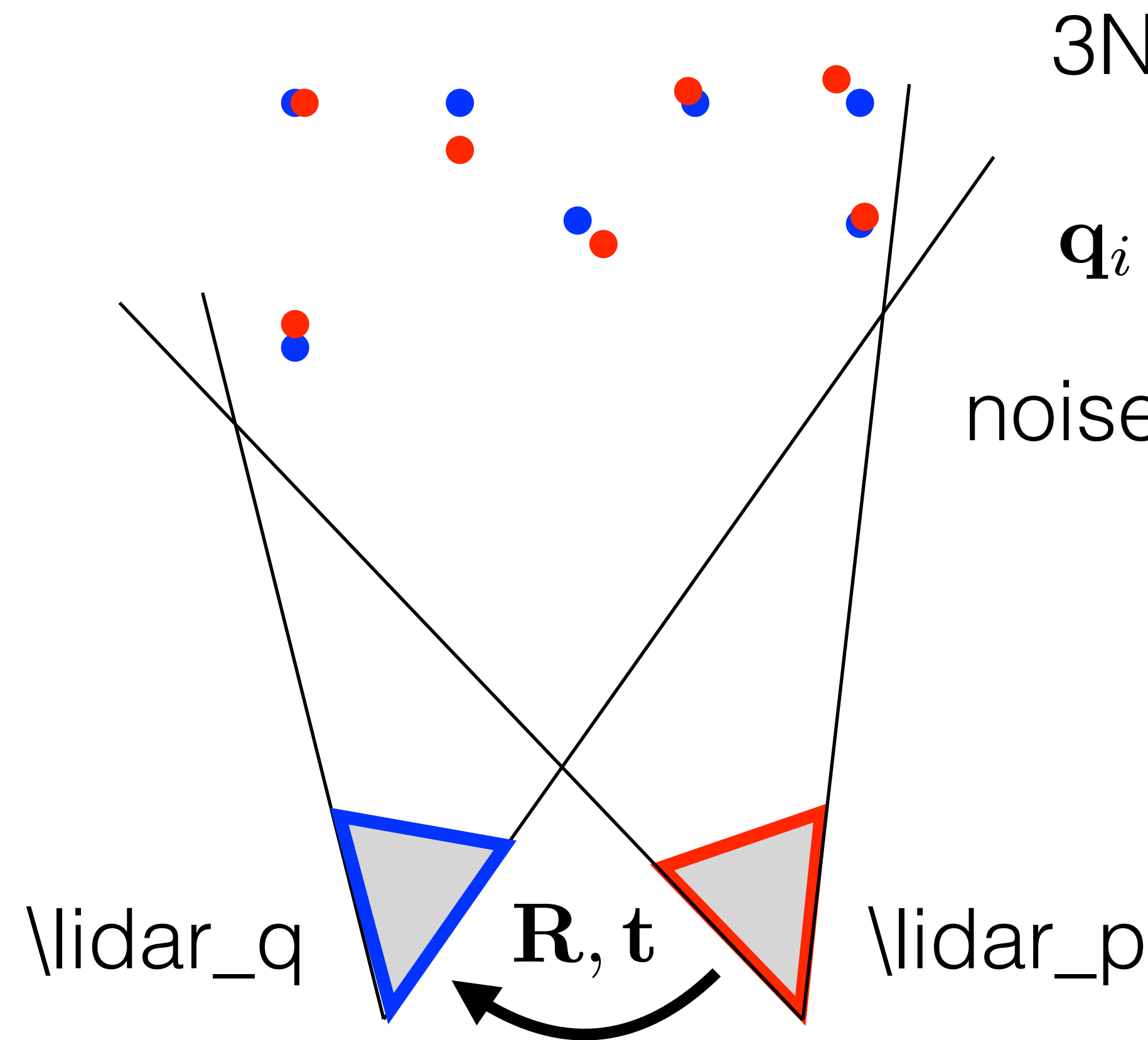
Mutual calibration of two coordinate frames



3N equations (\mathbf{R}, \mathbf{t} unknowns):

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

Mutual calibration of two coordinate frames

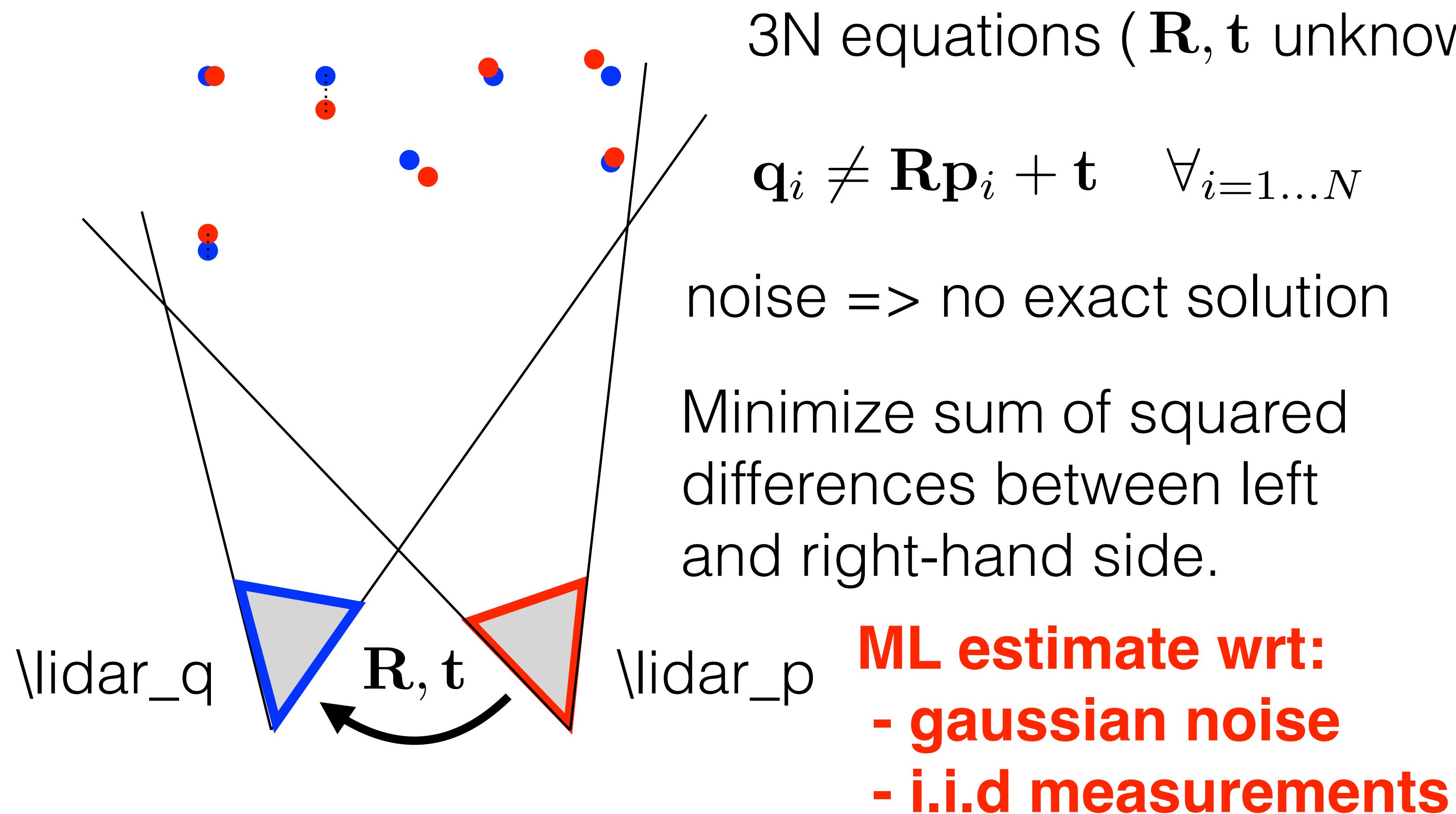


3N equations (\mathbf{R}, \mathbf{t} unknowns):

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

noise \Rightarrow no exact solution

Mutual calibration of two coordinate frames

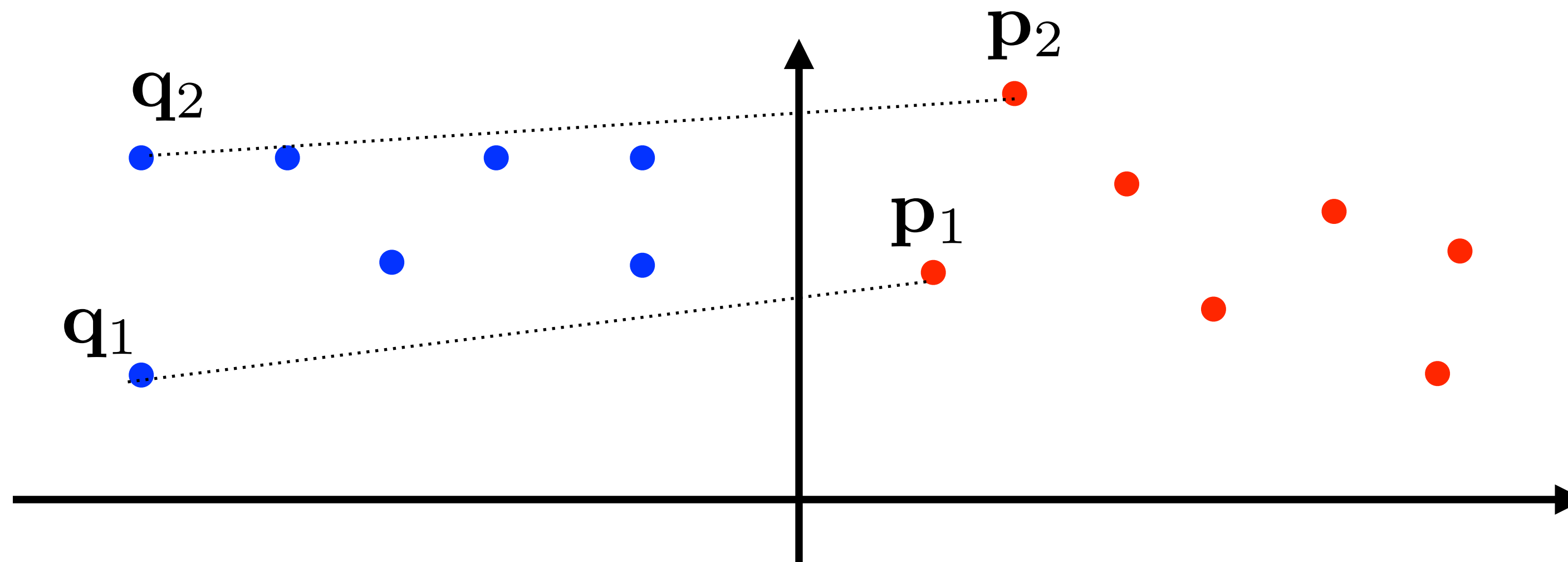


$$\mathbf{R}^*, \mathbf{t}^* = \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2$$

Mutual calibration of two coordinate frames

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

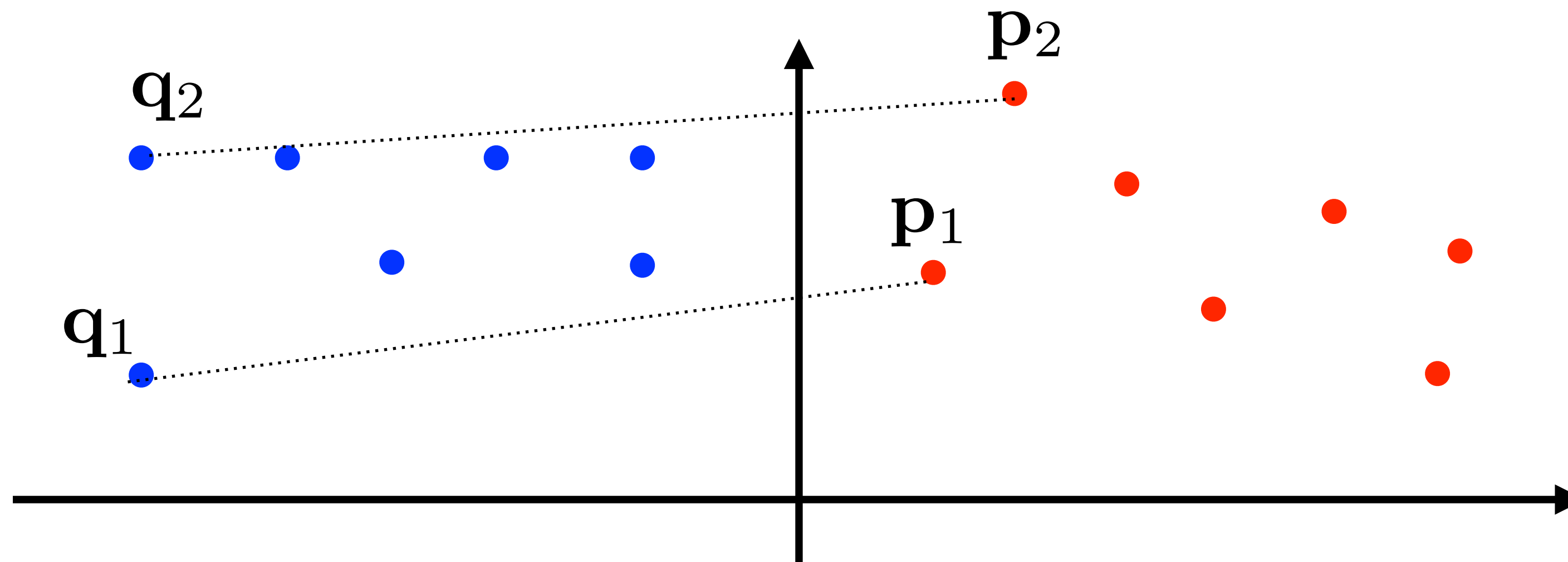
Solution:



Mutual calibration of two coordinate frames

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

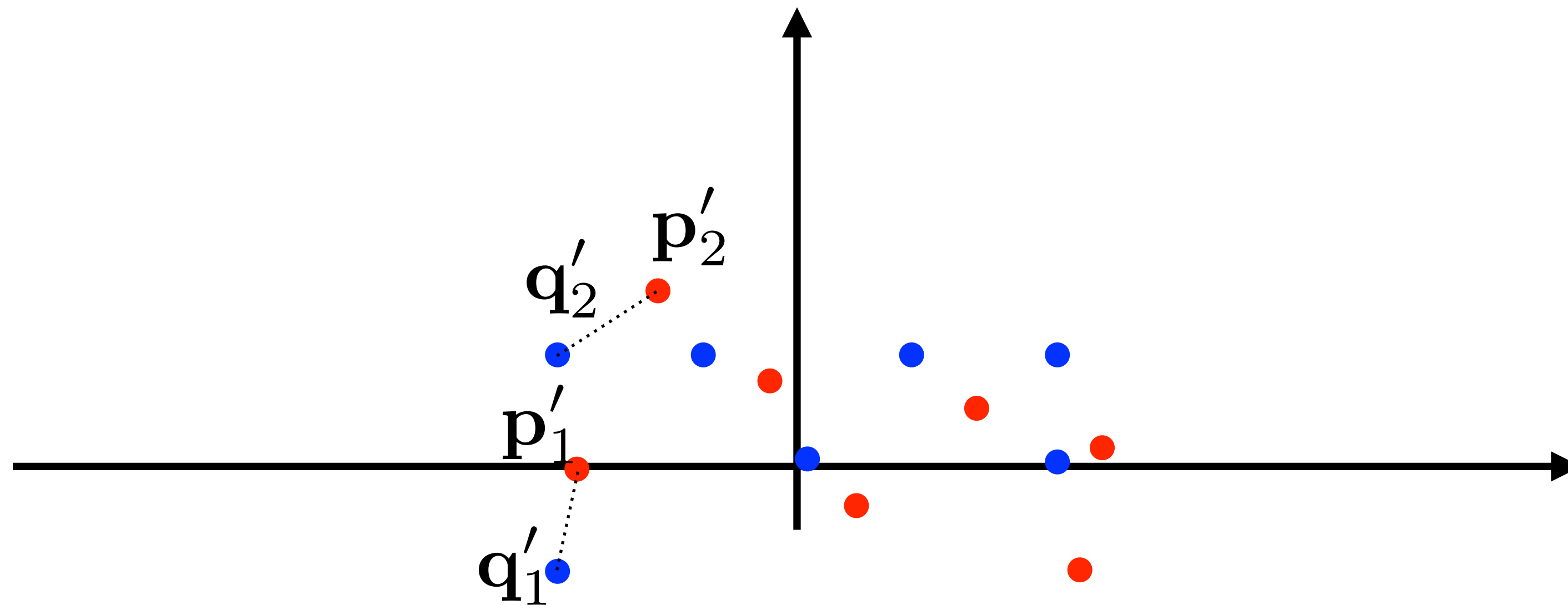
Solution: $\mathbf{p}'_i = \mathbf{p}_i - \underbrace{\frac{1}{N} \sum_i \mathbf{p}_i}_{\tilde{\mathbf{p}}}, \quad \mathbf{q}'_i = \mathbf{q}_i - \underbrace{\frac{1}{N} \sum_i \mathbf{q}_i}_{\tilde{\mathbf{q}}}$



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

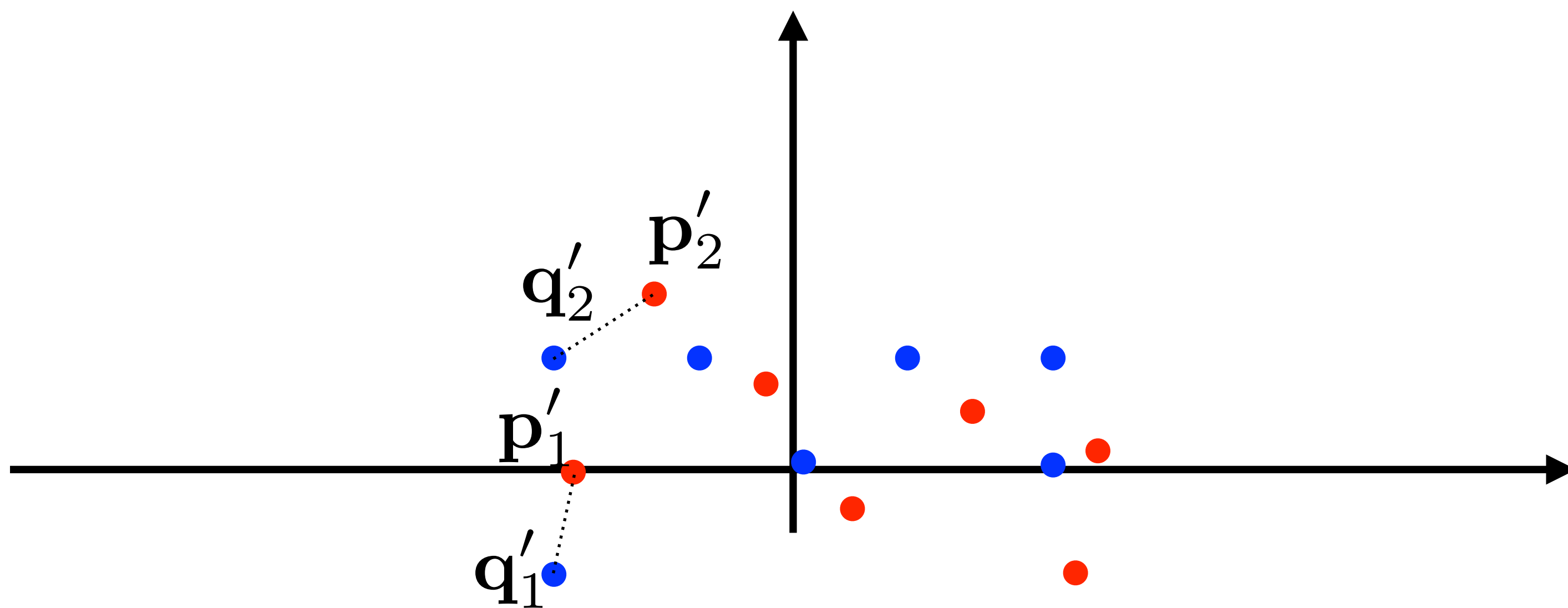


Mutual calibration of two coordinate frames

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

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

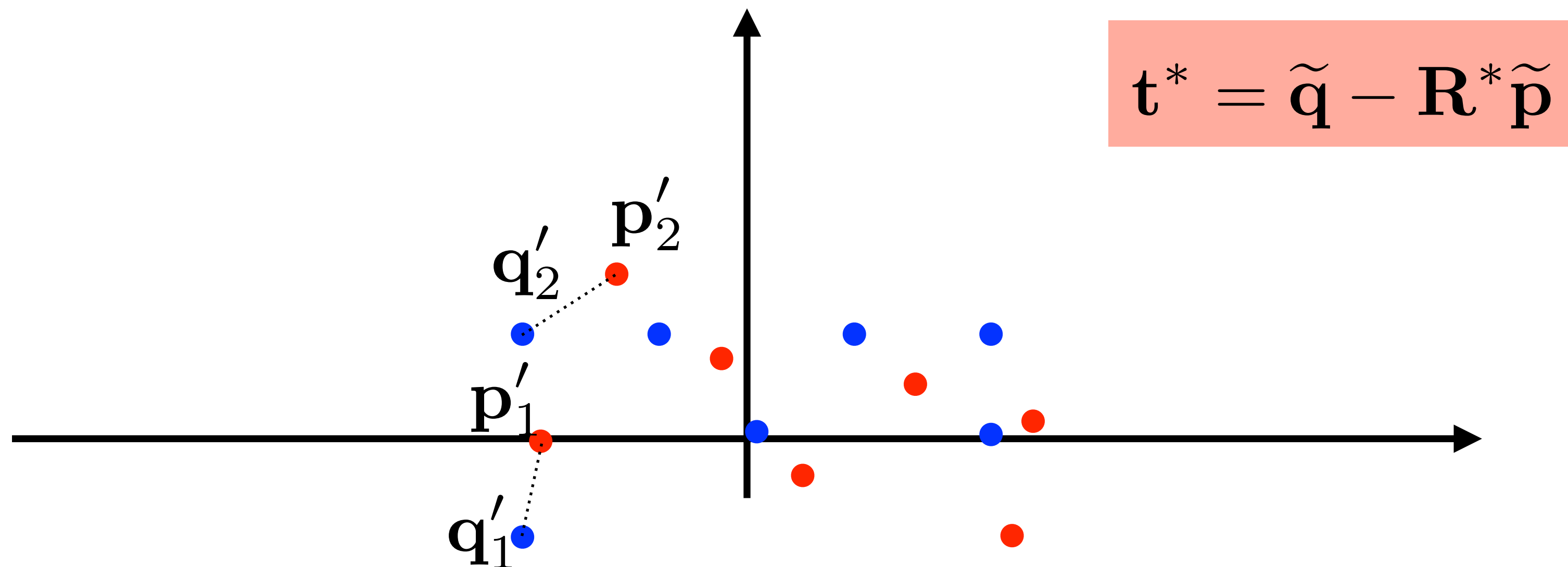
Solution: $\mathbf{p}'_i = \mathbf{p}_i - \underbrace{\frac{1}{N} \sum_i \mathbf{p}_i}_{\tilde{\mathbf{p}}}, \quad \mathbf{q}'_i = \mathbf{q}_i - \underbrace{\frac{1}{N} \sum_i \mathbf{q}_i}_{\tilde{\mathbf{q}}}$



Mutual calibration of two coordinate frames

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

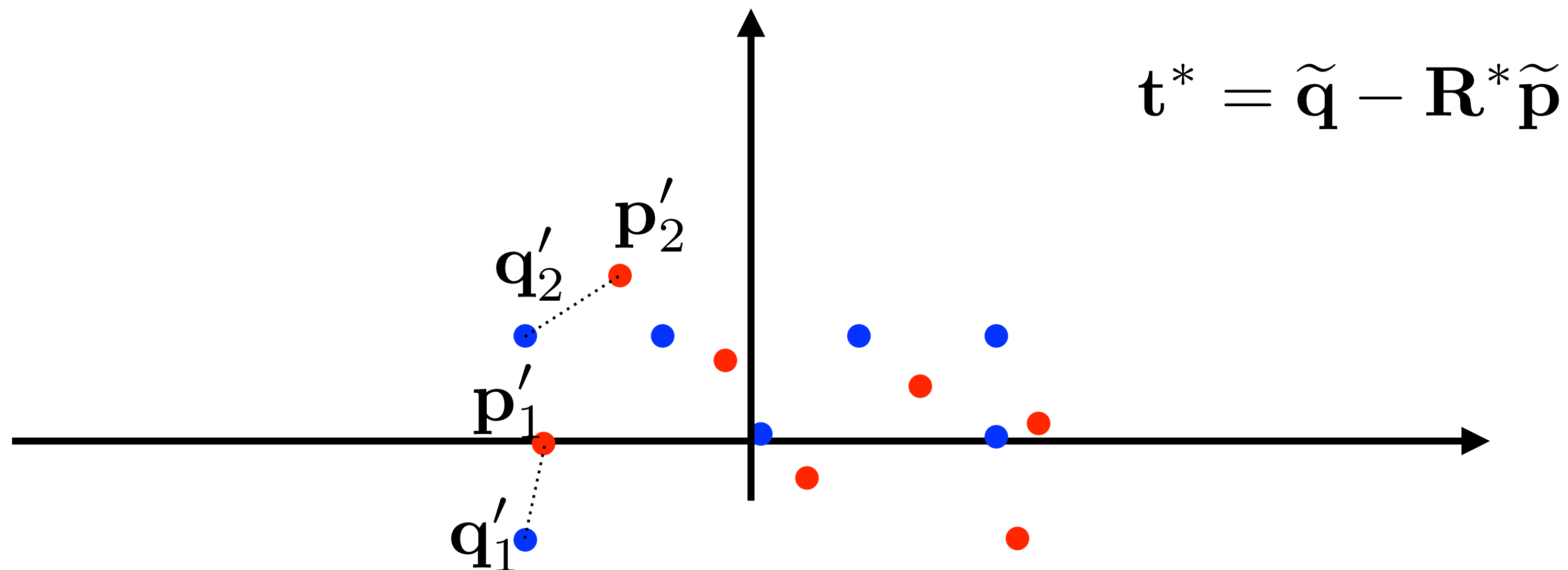
Solution: $\mathbf{p}'_i = \mathbf{p}_i - \underbrace{\frac{1}{N} \sum_i \mathbf{p}_i}_{\tilde{\mathbf{p}}}, \quad \mathbf{q}'_i = \mathbf{q}_i - \underbrace{\frac{1}{N} \sum_i \mathbf{q}_i}_{\tilde{\mathbf{q}}}$



Mutual calibration of two coordinate frames

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

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

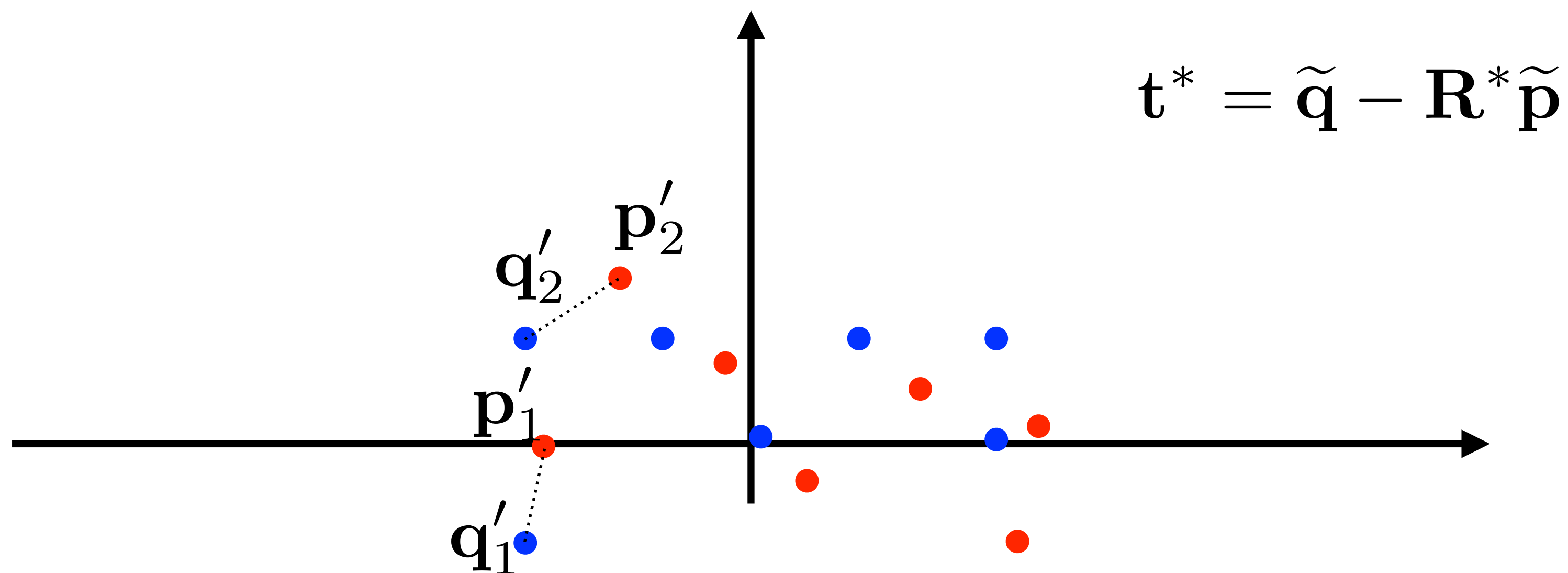


Mutual calibration of two coordinate frames

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

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

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



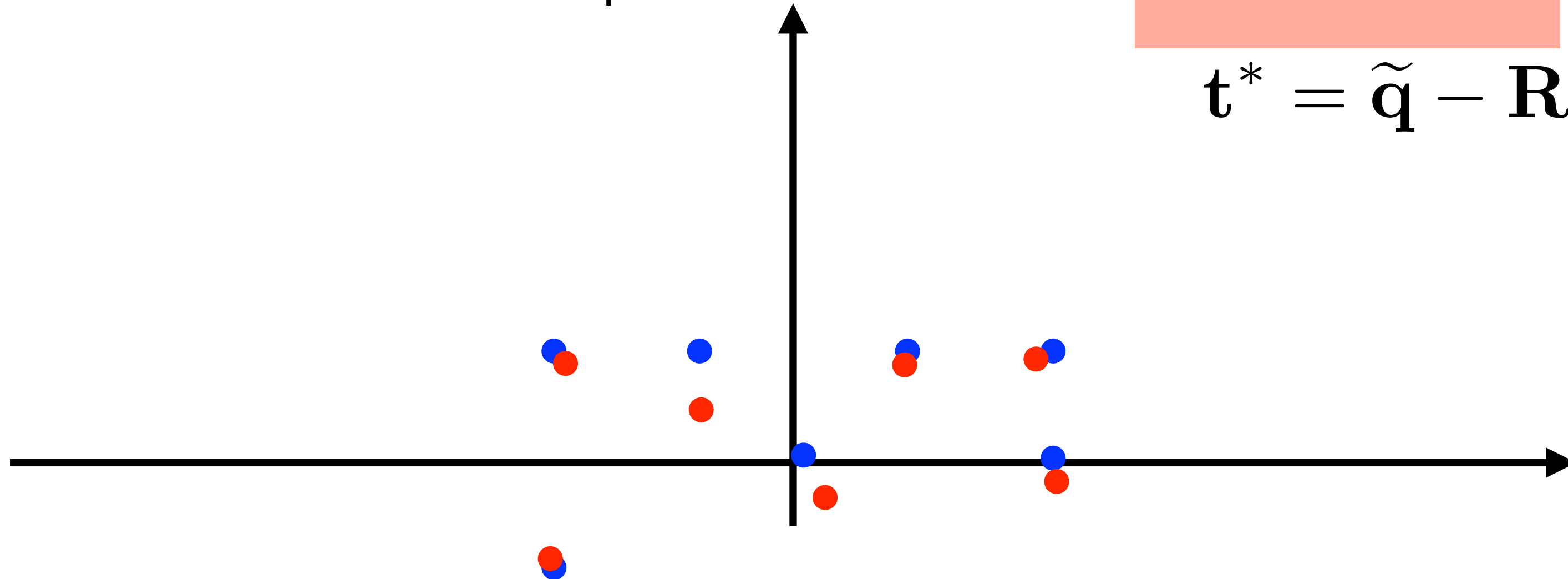
Mutual calibration of two coordinate frames

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

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



Mutual calibration of two coordinate frames

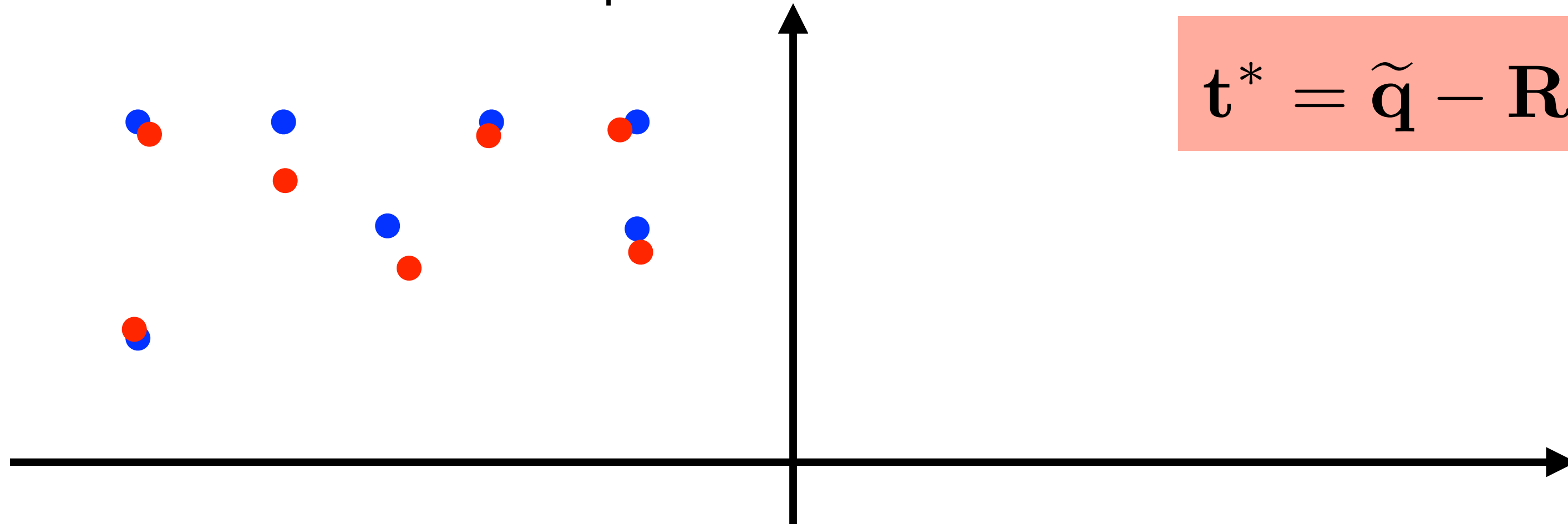
$$\begin{aligned}\mathbf{R}^*, \mathbf{t}^* &= \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2 \\ &= \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i\|_2^2 +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

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

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



Mutual calibration of two coordinate frames

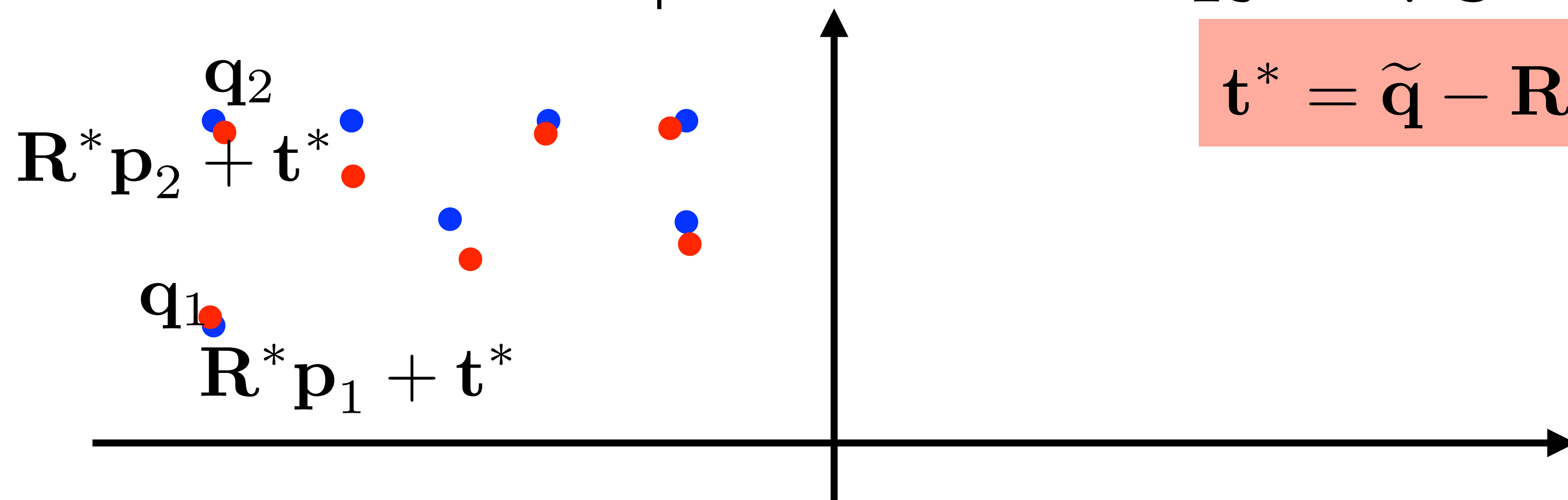
$$\begin{aligned} \mathbf{R}^*, \mathbf{t}^* &= \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2 \\ &= \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i\|_2^2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

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

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



Mutual calibration of two coordinate frames

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

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

$$\begin{aligned} \text{Solution: } \mathbf{R}^* &= \mathbf{V}\mathbf{U}^\top \\ \mathbf{t}^* &= \tilde{\mathbf{q}} - \mathbf{R}^* \tilde{\mathbf{p}} \end{aligned}$$

In python:

```
H = P @ Q.T
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

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

Proof [Arun-TPAMI-87]

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

Proof [Arun-TPAMI-87]

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

Proof [Arun-TPAMI-87]

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

Proof [Arun-TPAMI-87]

$$\begin{aligned}
 \mathbf{R}^*, \mathbf{t}^* &= \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2 = \\
 &= \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}(\mathbf{p}'_i + \tilde{\mathbf{p}}) + \mathbf{t} - \mathbf{q}'_i - \tilde{\mathbf{q}}\|_2^2 = \\
 &= \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i \|\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i + \underbrace{\mathbf{R}\tilde{\mathbf{p}} + \mathbf{t} - \tilde{\mathbf{q}}}_{\mathbf{t}'}\|_2^2 = \\
 &= \arg \min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3} \sum_i (\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i + \mathbf{t}')^\top (\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i + \mathbf{t}') =
 \end{aligned}$$

Proof [Arun-TPAMI-87]

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

Proof [Arun-TPAMI-87]

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

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

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

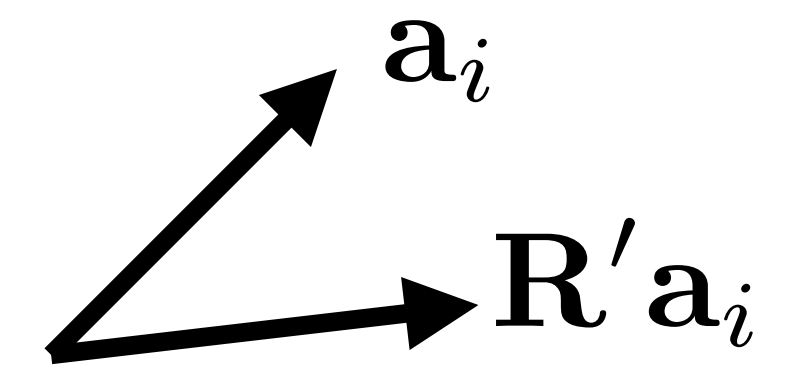
$$\arg \min_{\mathbf{R} \in SO(3)} \sum_i \|\mathbf{R}\mathbf{p}'_i - \mathbf{q}'_i\|_2^2 = \arg \max_{\mathbf{R} \in SO(3)} \sum_i \mathbf{q}'_i{}^\top \mathbf{R}\mathbf{p}'_i =$$

$$= \arg \max_{\mathbf{R} \in SO(3)} \sum_i \underbrace{\mathbf{q}'_i{}^\top}_{\mathbf{a}_i} \underbrace{\mathbf{R}\mathbf{p}'_i}_{\mathbf{b}_i} = \arg \max_{\mathbf{R} \in SO(3)} \text{trace } \mathbf{R} \underbrace{\mathbf{P}\mathbf{Q}^\top}_{\mathbf{H}} = \mathbf{V}\mathbf{U}^\top$$

$$\arg \max_{\mathbf{R}', \mathbf{R}^* \in SO(3)} \text{trace } \mathbf{R}' \mathbf{R}^* \mathbf{U}\mathbf{S}\mathbf{V}^\top \quad \dots \text{ expand into two rotations}$$

$$\arg \max_{\mathbf{R}' \in SO(3)} \text{trace } \mathbf{R}' \underbrace{\mathbf{V}\mathbf{U}^\top}_{\mathbf{R}^*} \underbrace{\mathbf{U}\mathbf{S}\mathbf{V}^\top}_{\mathbf{H}} = \arg \max_{\mathbf{R}' \in SO(3)} \text{trace } \mathbf{R}' \underbrace{(\mathbf{V}\sqrt{\mathbf{S}})}_{\mathbf{A}} \underbrace{(\sqrt{\mathbf{S}}\mathbf{V})^\top}_{\mathbf{A}^\top} =$$

$$= \arg \max_{\mathbf{R}' \in SO(3)} \sum_i \mathbf{a}_i{}^\top \mathbf{R}' \mathbf{a}_i = \mathbf{E}$$



$$\text{trace } \mathbf{B}\mathbf{A}^\top = \sum_i \mathbf{a}_i{}^\top \mathbf{b}_i$$