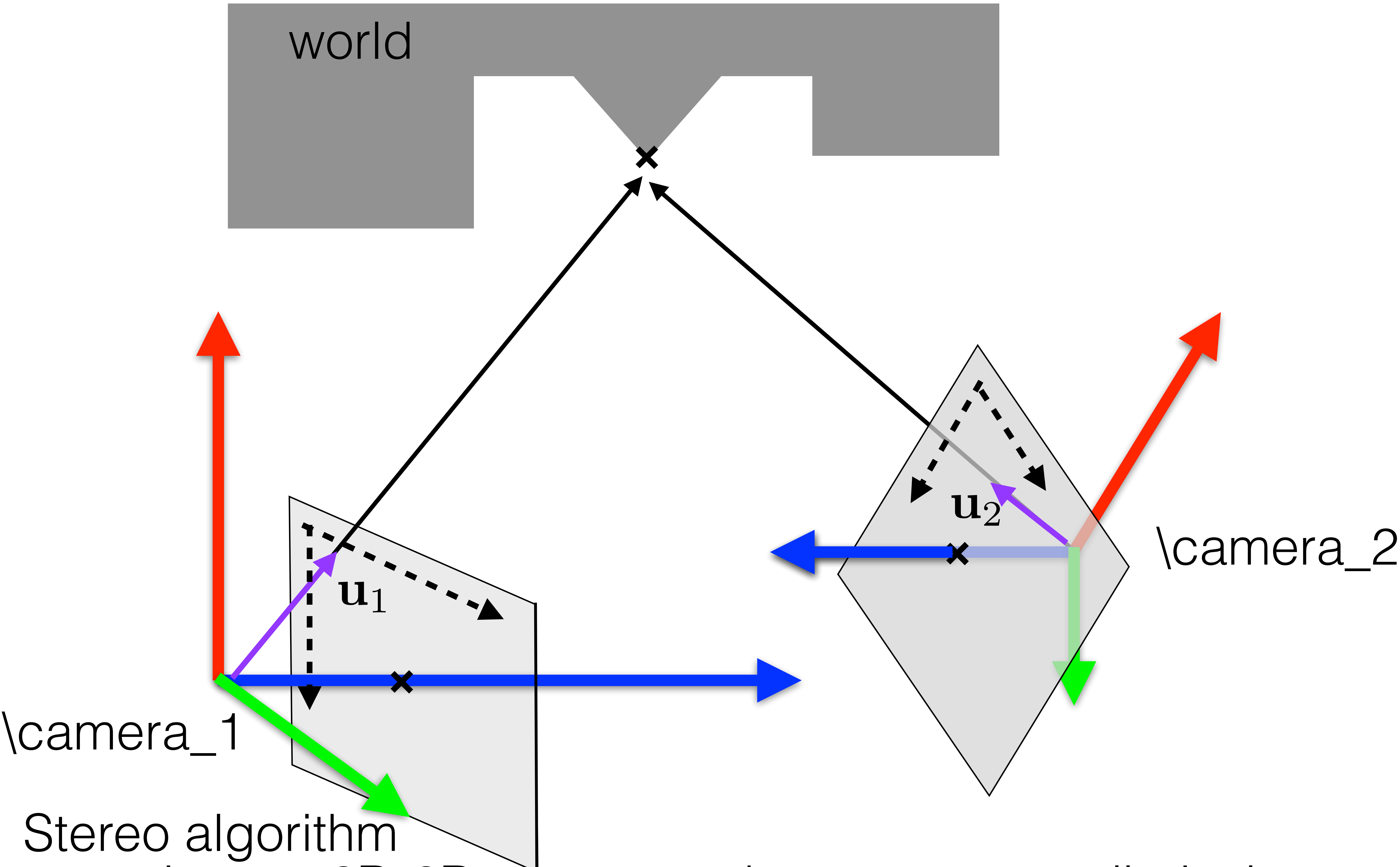
RGBD cameras and calibration

Karel Zimmermann

Stereo

- Pair of cameras mounted on a common rigid body, which provide depth (or 3D point cloud).
- Simulate human binocular vision.
- In contrast to lidar, it is a passive sensor.

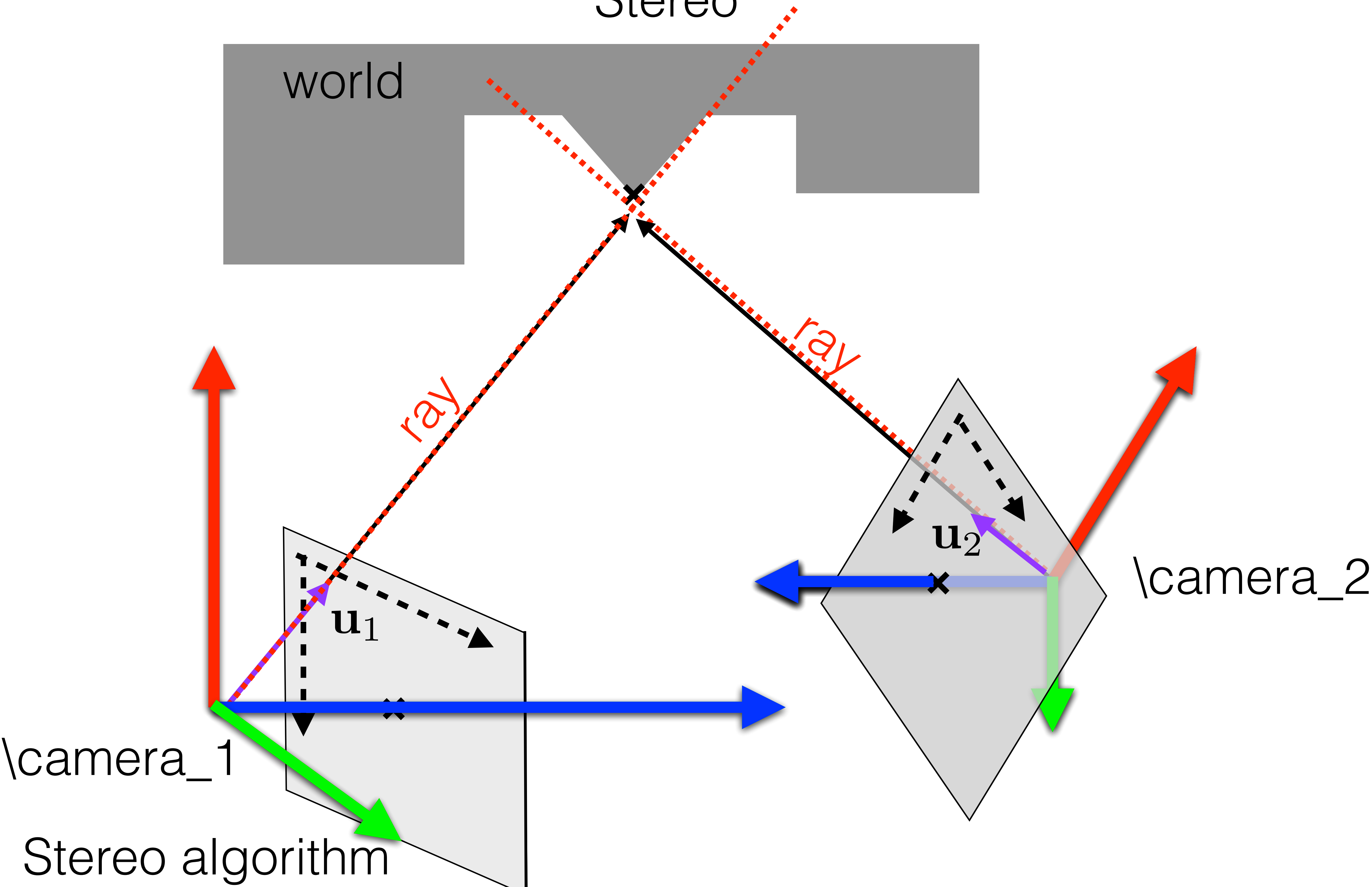
Stereo



Stereo algorithm

- estimates 2D-2D correspondences among all pixels

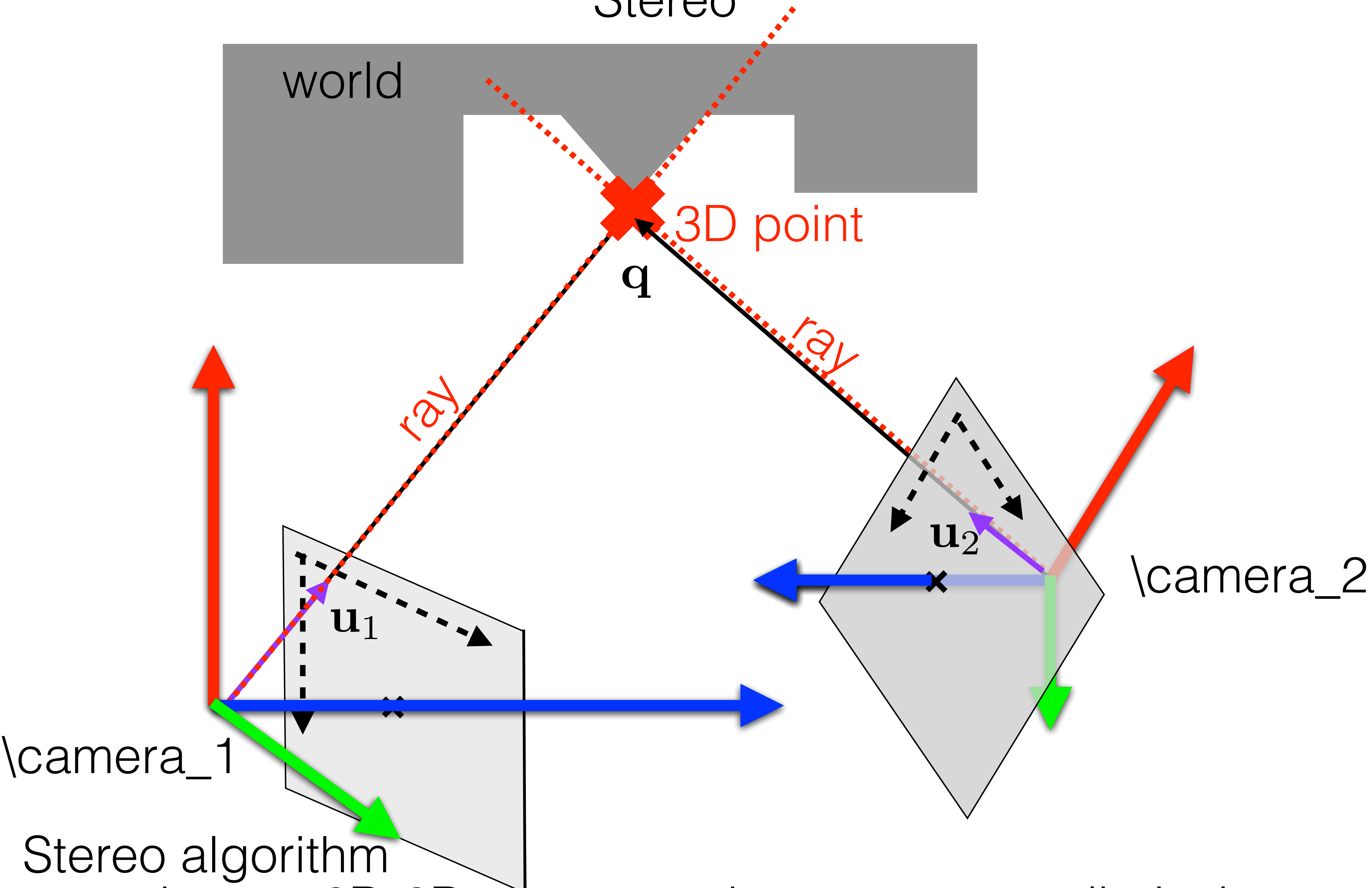
Stereo



Stereo algorithm

- estimates 2D-2D correspondences among all pixels
- for each 2D-2D correspondence cast two rays

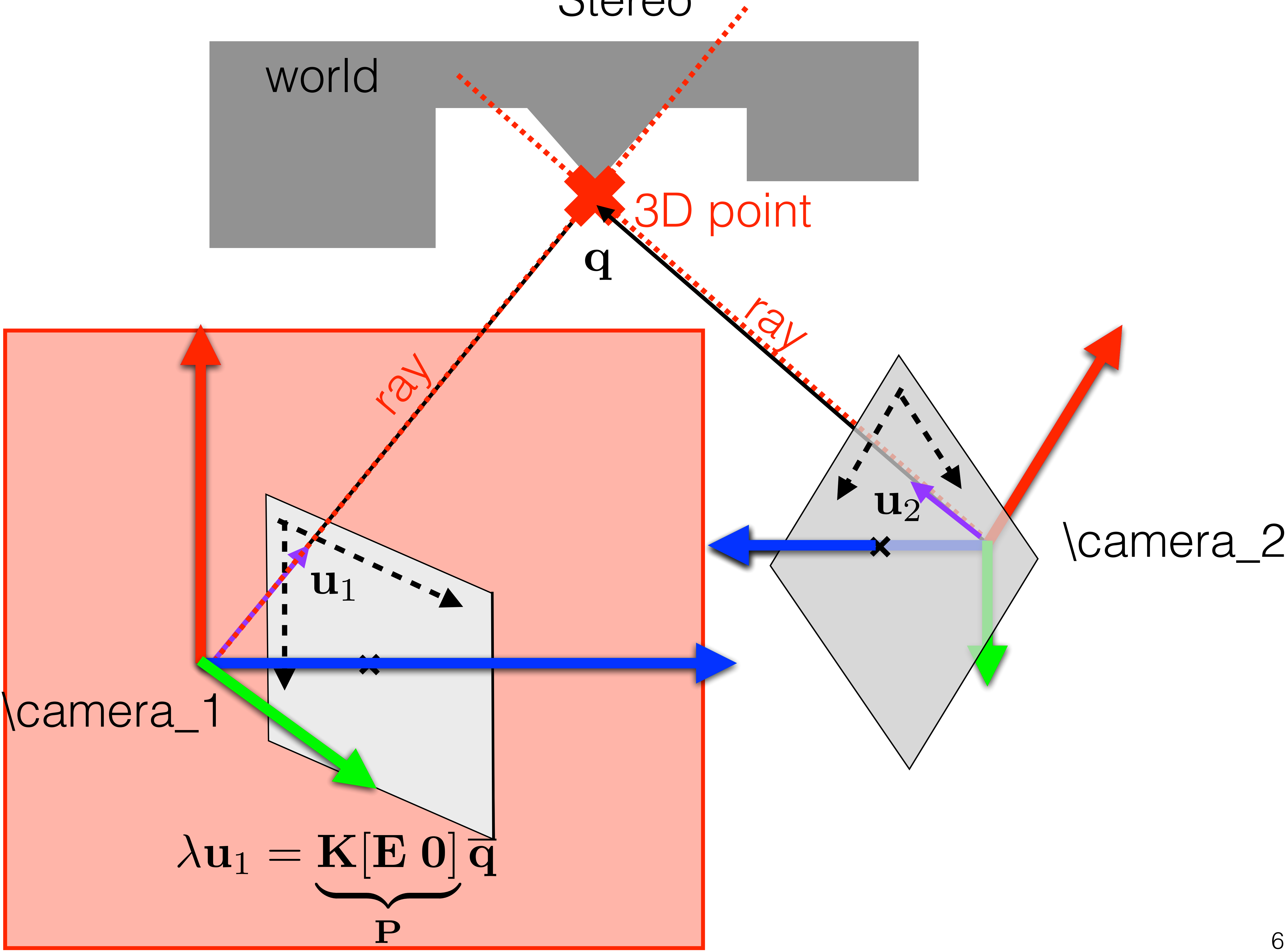
Stereo



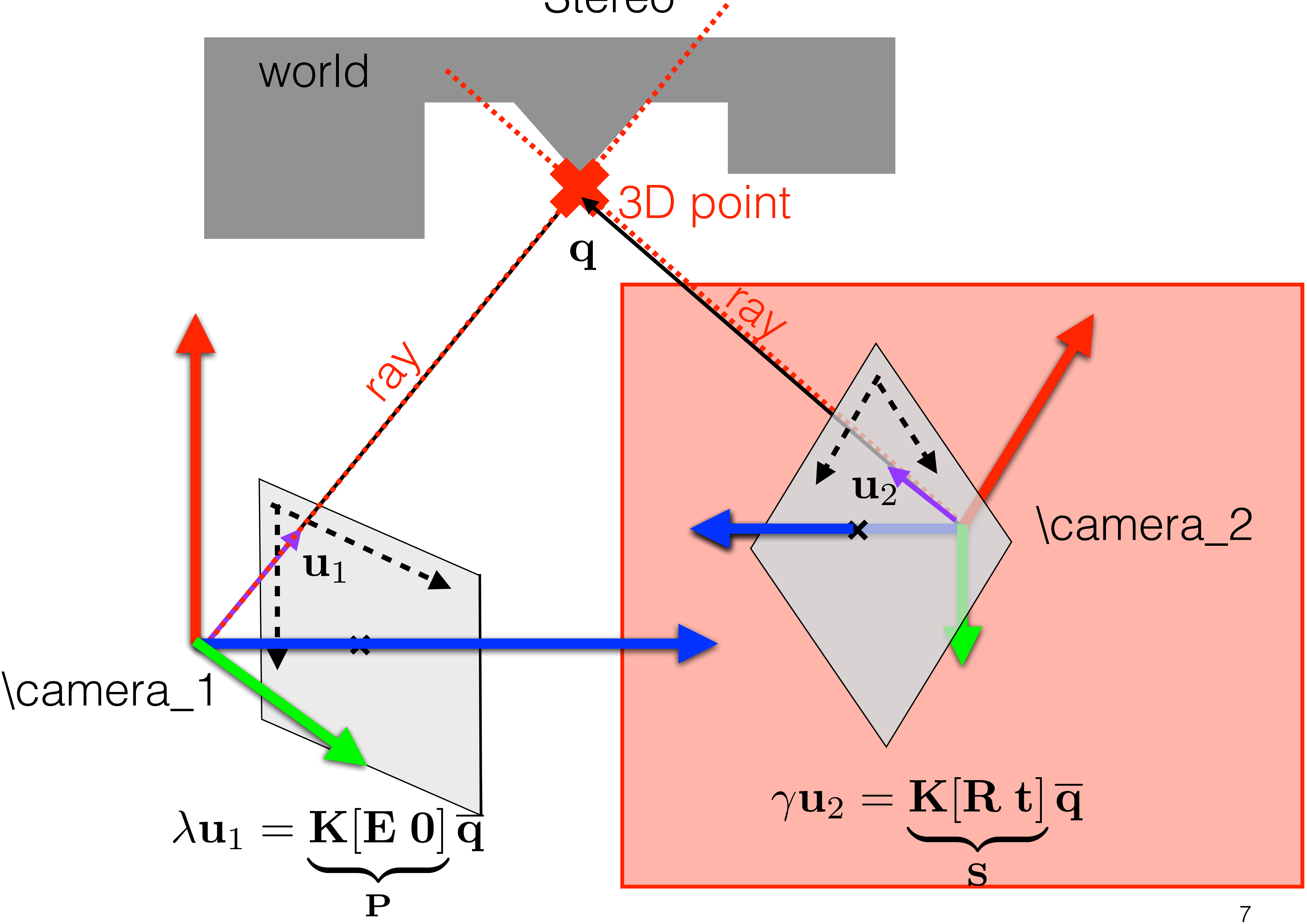
Stereo algorithm

- estimates 2D-2D correspondences among all pixels
- for each 2D-2D correspondence cast two rays
- compute their intersection => 3D point

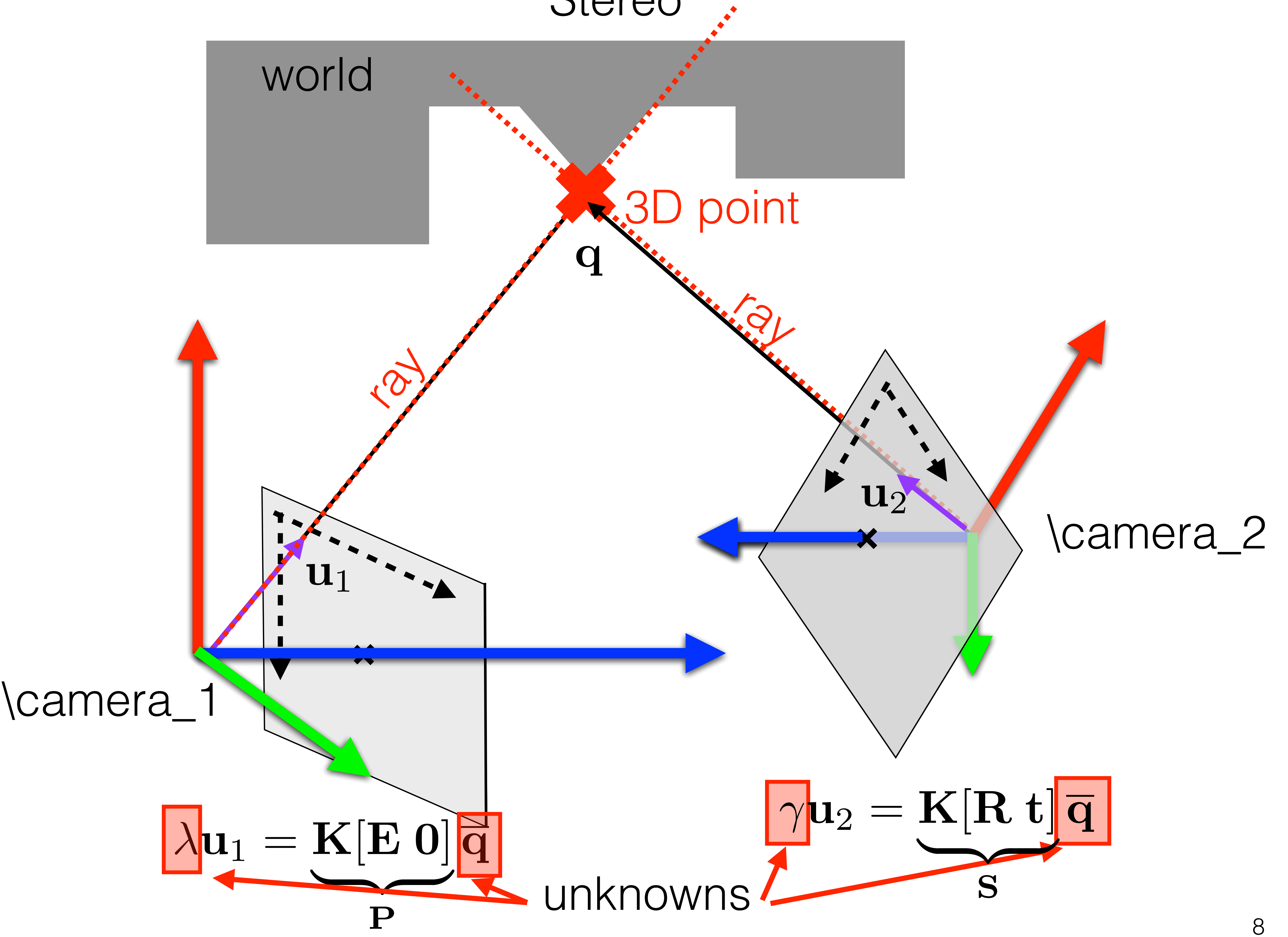
Stereo


world
3D point
$\mathbf{q}$
ray
ray
$\mathbf{u}_1$
$\mathbf{u}_2$
\camera_2
\camera_1
$\lambda \mathbf{u}_1 = \underbrace{\mathbf{K}[\mathbf{E}\ \mathbf{0}]}_{\mathbf{P}}\ \overline{\mathbf{q}}$

Stereo


world
3D point
$\mathbf{q}$
ray
ray
$\mathbf{u}_1$
$\mathbf{u}_2$
\camera_1
\camera_2
$\lambda \mathbf{u}_1 = \underbrace{\mathbf{K}[\mathbf{E}\ \mathbf{0}]}_{\mathbf{P}} \overline{\mathbf{q}}$
$\gamma \mathbf{u}_2 = \underbrace{\mathbf{K}[\mathbf{R}\ \mathbf{t}]}_{\mathbf{S}} \overline{\mathbf{q}}$

Stereo


world
3D point
q
ray
ray
u1
u2
\camera_1
\camera_2
$\lambda \mathbf{u}_1 = \mathbf{K}[\mathbf{E}\ \mathbf{0}]\overline{\mathbf{q}}$
P
$\gamma \mathbf{u}_2 = \mathbf{K}[\mathbf{R}\ \mathbf{t}]\overline{\mathbf{q}}$
S
unknowns

Stereo

$$\lambda \mathbf{u}_1 = \underbrace{\mathbf{K}[\mathbf{E}\ \mathbf{0}]}_{\mathbf{P}} \overline{\mathbf{q}} \qquad \gamma \mathbf{u}_2 = \underbrace{\mathbf{K}[\mathbf{R}\ \mathbf{t}]}_{\mathbf{S}} \overline{\mathbf{q}}$$

unknowns (λ, $\overline{\mathbf{q}}$, γ, $\overline{\mathbf{q}}$)

$$\lambda u_{1x} = \mathbf{p}_1^\top \overline{\mathbf{q}} \qquad \gamma u_{2x} = \mathbf{s}_1^\top \overline{\mathbf{q}}$$

$$\lambda u_{1y} = \mathbf{p}_2^\top \overline{\mathbf{q}} \qquad \gamma u_{2y} = \mathbf{s}_2^\top \overline{\mathbf{q}}$$

$$\lambda = \mathbf{p}_3^\top \overline{\mathbf{q}} \qquad \gamma = \mathbf{s}_3^\top \overline{\mathbf{q}}$$

get rid of λ, γ

$$\mathbf{p}_3^\top \overline{\mathbf{q}} u_{1x} = \mathbf{p}_1^\top \overline{\mathbf{q}} \qquad \mathbf{s}_3^\top \overline{\mathbf{q}} u_{2x} = \mathbf{s}_1^\top \overline{\mathbf{q}}$$

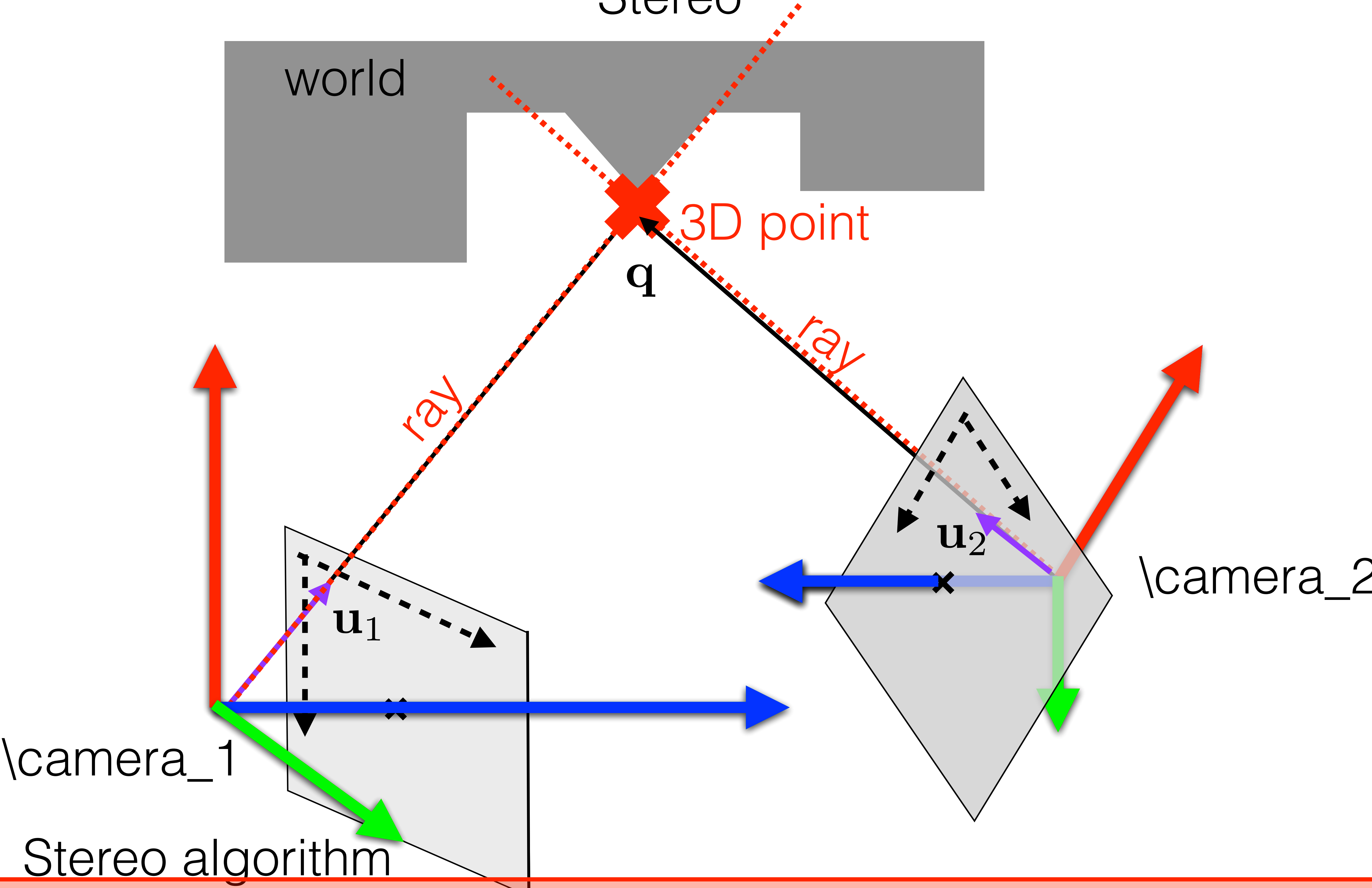
$$\mathbf{p}_3^\top \overline{\mathbf{q}} u_{1y} = \mathbf{p}_2^\top \overline{\mathbf{q}} \qquad \mathbf{s}_3^\top \overline{\mathbf{q}} u_{2y} = \mathbf{s}_2^\top \overline{\mathbf{q}}$$

rewrite to matrix form

$$\underbrace{\begin{bmatrix} \mathbf{p}_1^\top - \mathbf{p}_3^\top u_{1x} \\ \mathbf{p}_2^\top - \mathbf{p}_3^\top u_{1y} \\ \mathbf{s}_1^\top - \mathbf{s}_3^\top u_{2x} \\ \mathbf{s}_2^\top - \mathbf{s}_3^\top u_{2y} \end{bmatrix}}_{\mathbf{A}_{[4\times 4]}} \overline{\mathbf{q}} = \mathbf{0}$$

unknown ($\overline{\mathbf{q}}$)

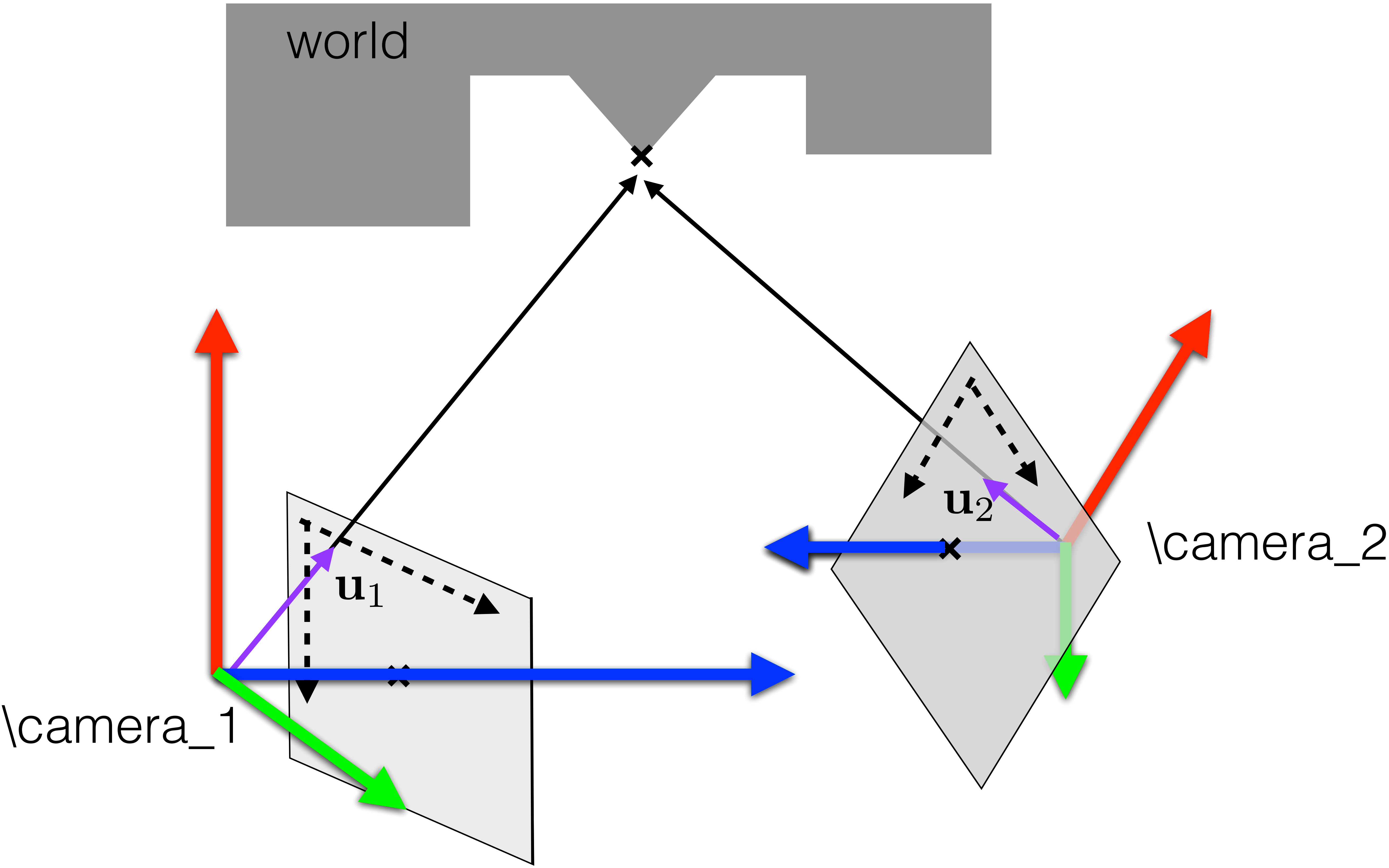
Stereo



Stereo algorithm

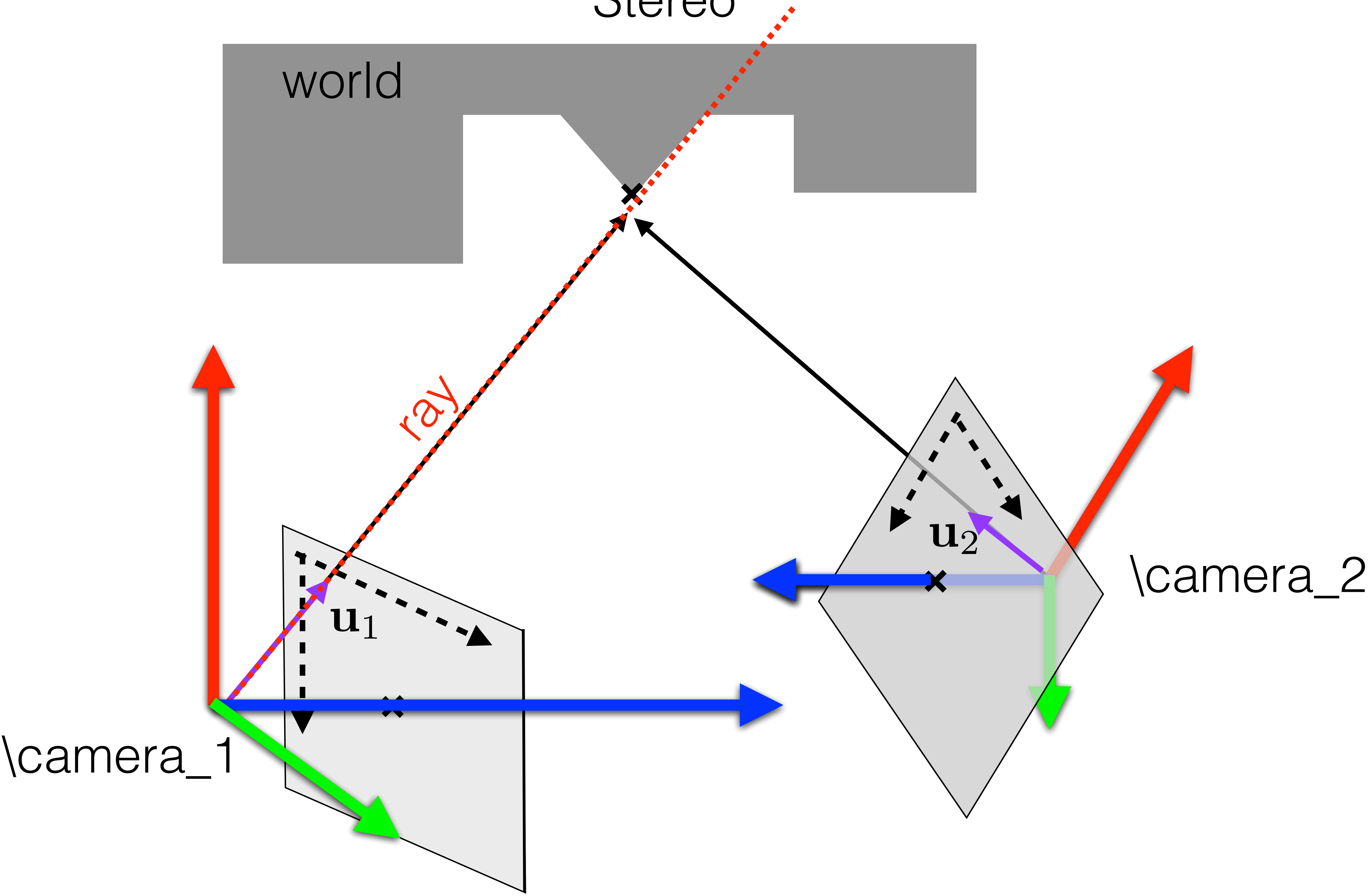
- estimates 2D-2D correspondences among all pixels
- for each 2D-2D correspondence cast two rays
- compute their intersection => 3D point

Stereo



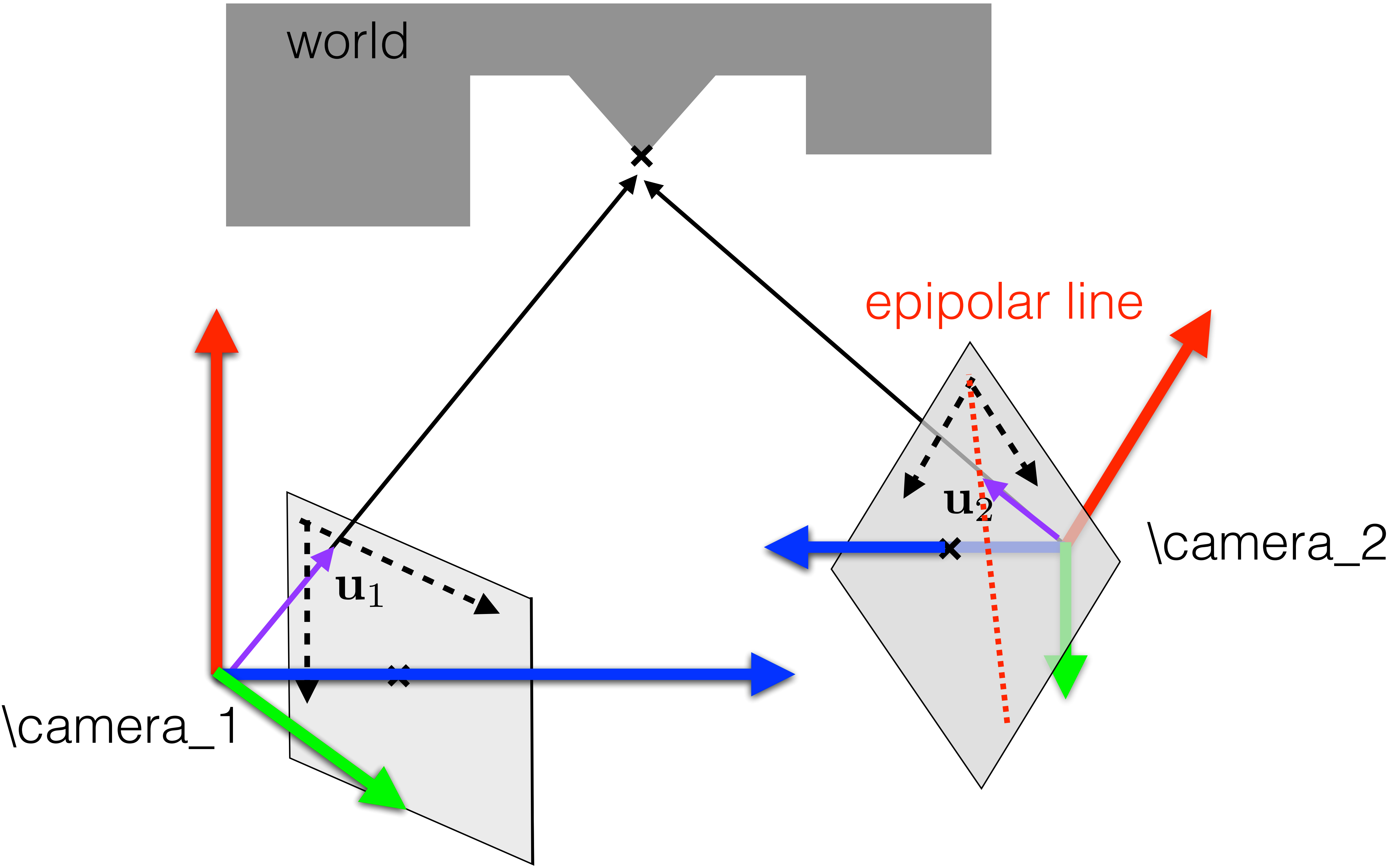
Given pixel $\mathbf{u}_1$ in \camera_1, where does the corresponding pixel $\mathbf{u}_2$ lie in \camera_2?

Stereo



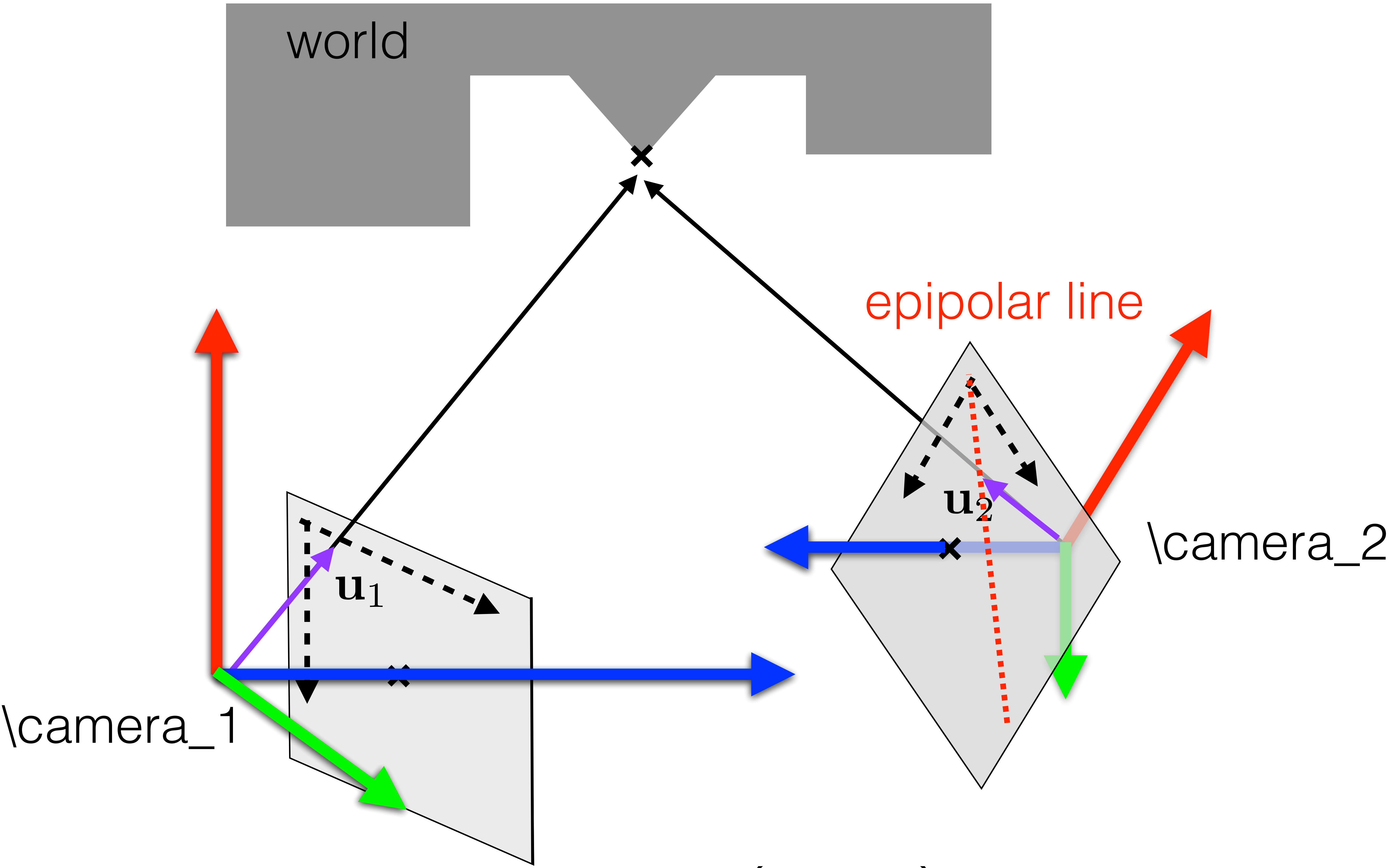
Given pixel $\mathbf{u}_1$ in \camera_1, where does the corresponding pixel $\mathbf{u}_2$ lie in \camera_2?

Stereo



Corresponding pixel lies on the epipolar line
(i.e. projection of the ray from camera_1 to \camera_2)

Stereo



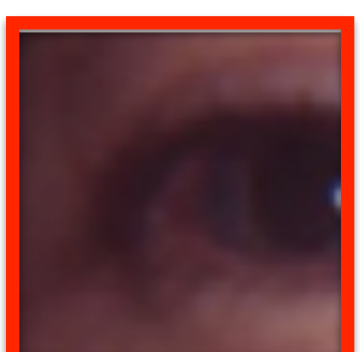
$$\mathcal{E} = \{\mathbf{u}_2 \in \mathcal{R}^2 \mid \; \overline{\mathbf{u}}_2^\top \underbrace{\mathbf{K}^{-\top}(\mathbf{R} \times \mathbf{t})\mathbf{K}^{-1}}_{\mathbf{F}} \overline{\mathbf{u}}_1 = 0\}$$

Stereo

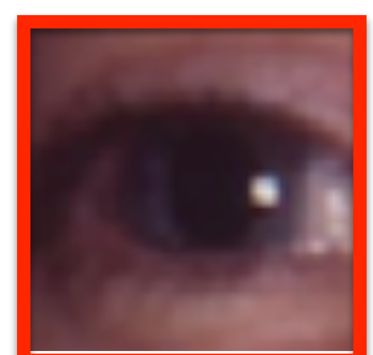
Stereo

Stereo

Stereo

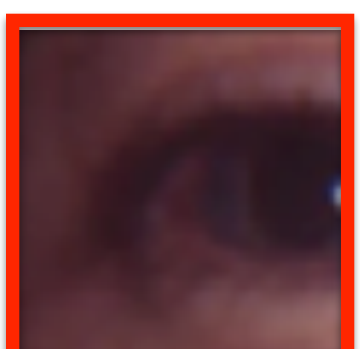
$I(\mathbf{x}) =$ 

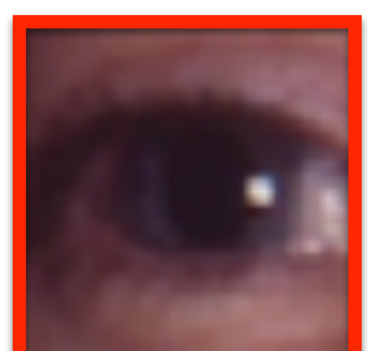
$\mathbf{x} \in \mathcal{W}$

$J(\mathbf{e} + \mathbf{x}) =$ 

$\mathbf{x} \in \mathcal{W}$
$\mathbf{e} \in \mathcal{E}$

Stereo

$I(\mathbf{x}) =$  $\mathbf{x} \in \mathcal{W}$

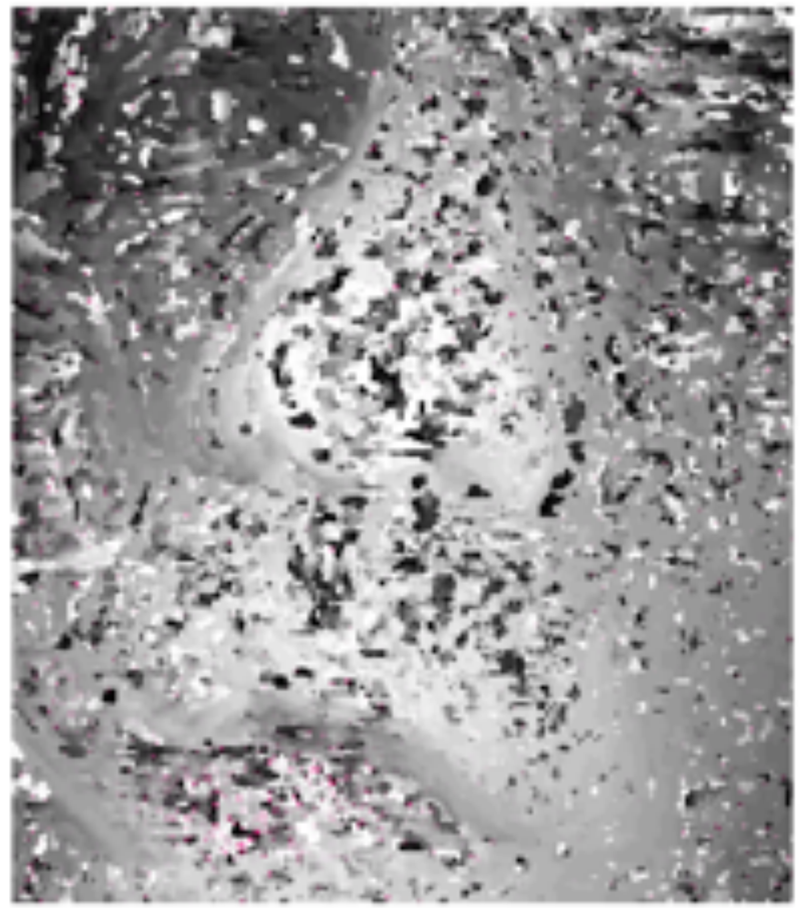
$J(\mathbf{e}+\mathbf{x}) =$  $\mathbf{x} \in \mathcal{W}$, $\mathbf{e} \in \mathcal{E}$

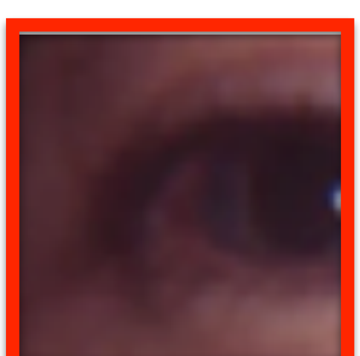
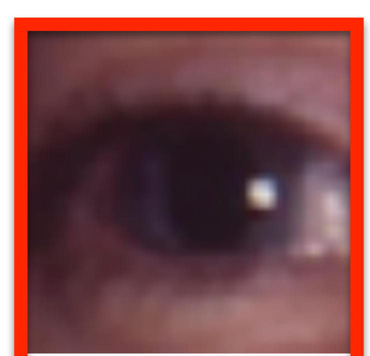
Similarity function:

$$\sum_{\mathbf{x}\in\mathcal{W}} \Big(J(\mathbf{e}+\mathbf{x}) - I(\mathbf{x}) \Big)^2$$

Stereo

greedy solution



$I(\mathbf{x}) =$  $J(\mathbf{e}+\mathbf{x}) =$ 

$\mathbf{x} \in \mathcal{W}$

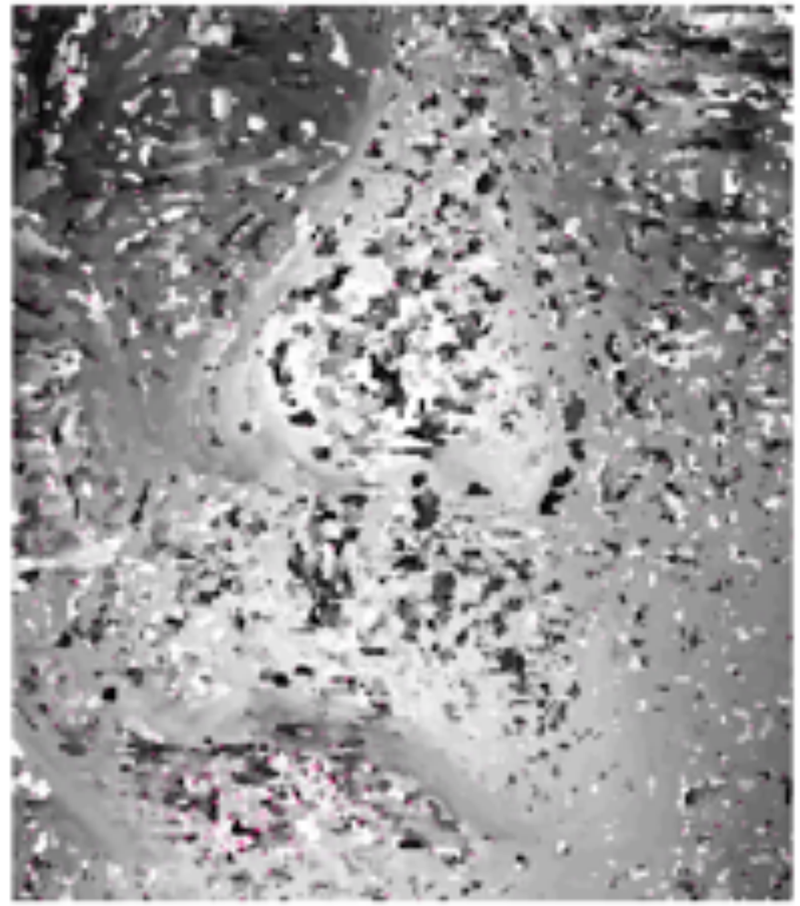
$\mathbf{x} \in \mathcal{W}$
$\mathbf{e} \in \mathcal{E}$

Similarity function:

$$\sum_{\mathbf{x}\in\mathcal{W}} \Big(J(\mathbf{e}+\mathbf{x}) - I(\mathbf{x}) \Big)^2$$

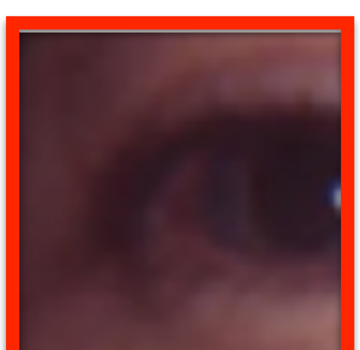
Stereo

greedy solution



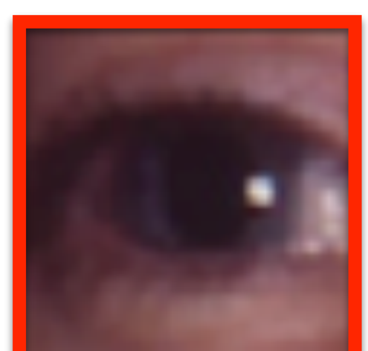
line smoothness

$I(\mathbf{x}) =$



$\mathbf{x} \in \mathcal{W}$

$J(\mathbf{e} + \mathbf{x}) =$



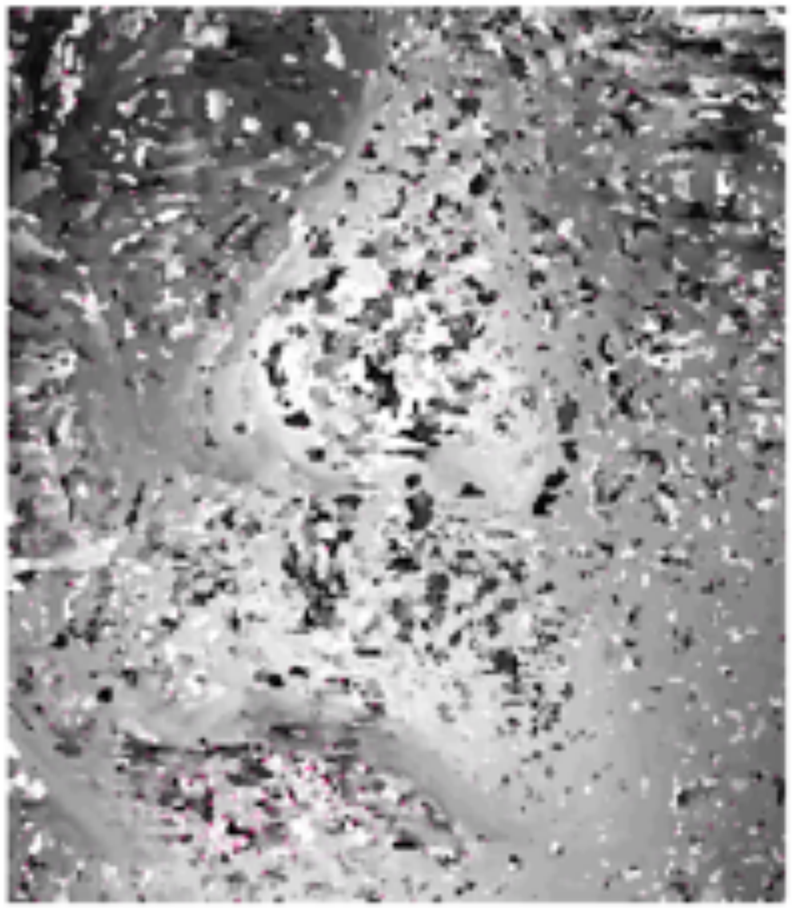
$\mathbf{x} \in \mathcal{W}$
$\mathbf{e} \in \mathcal{E}$

Similarity function:

$$\sum_{\mathbf{x} \in \mathcal{W}} \Big(J(\mathbf{e} + \mathbf{x}) - I(\mathbf{x}) \Big)^2$$

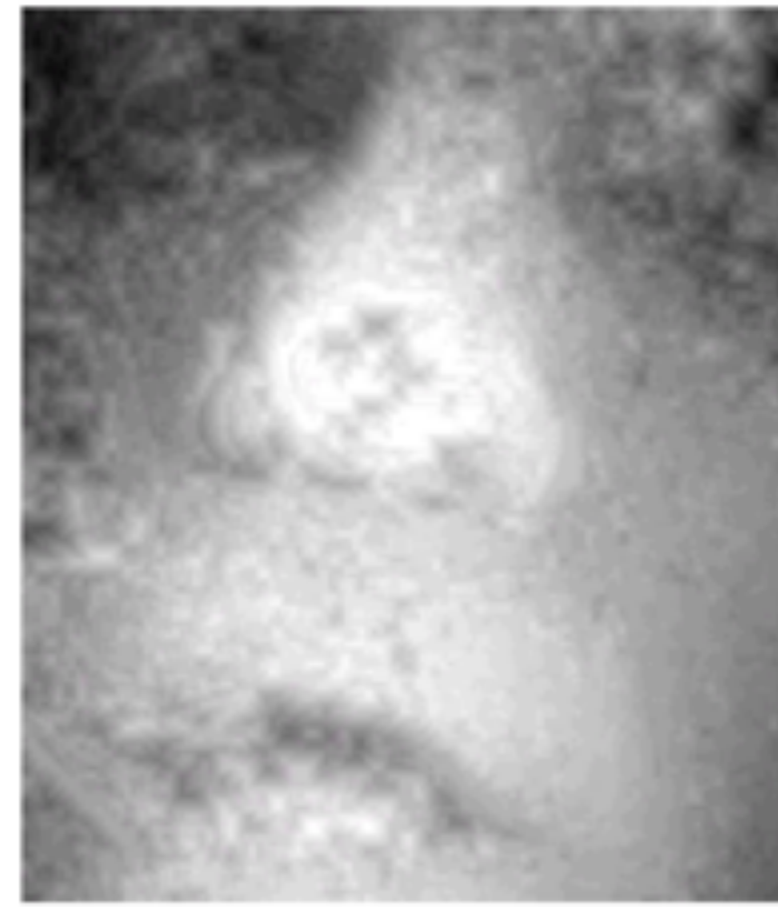
Stereo

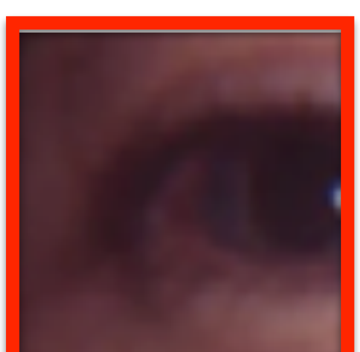
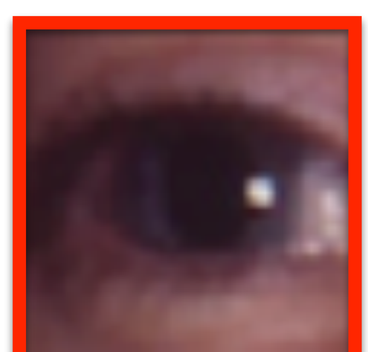
greedy solution



line smoothness

neighbourhood smoothness



$I(\mathbf{x}) =$  $J(\mathbf{e}+\mathbf{x}) =$ 

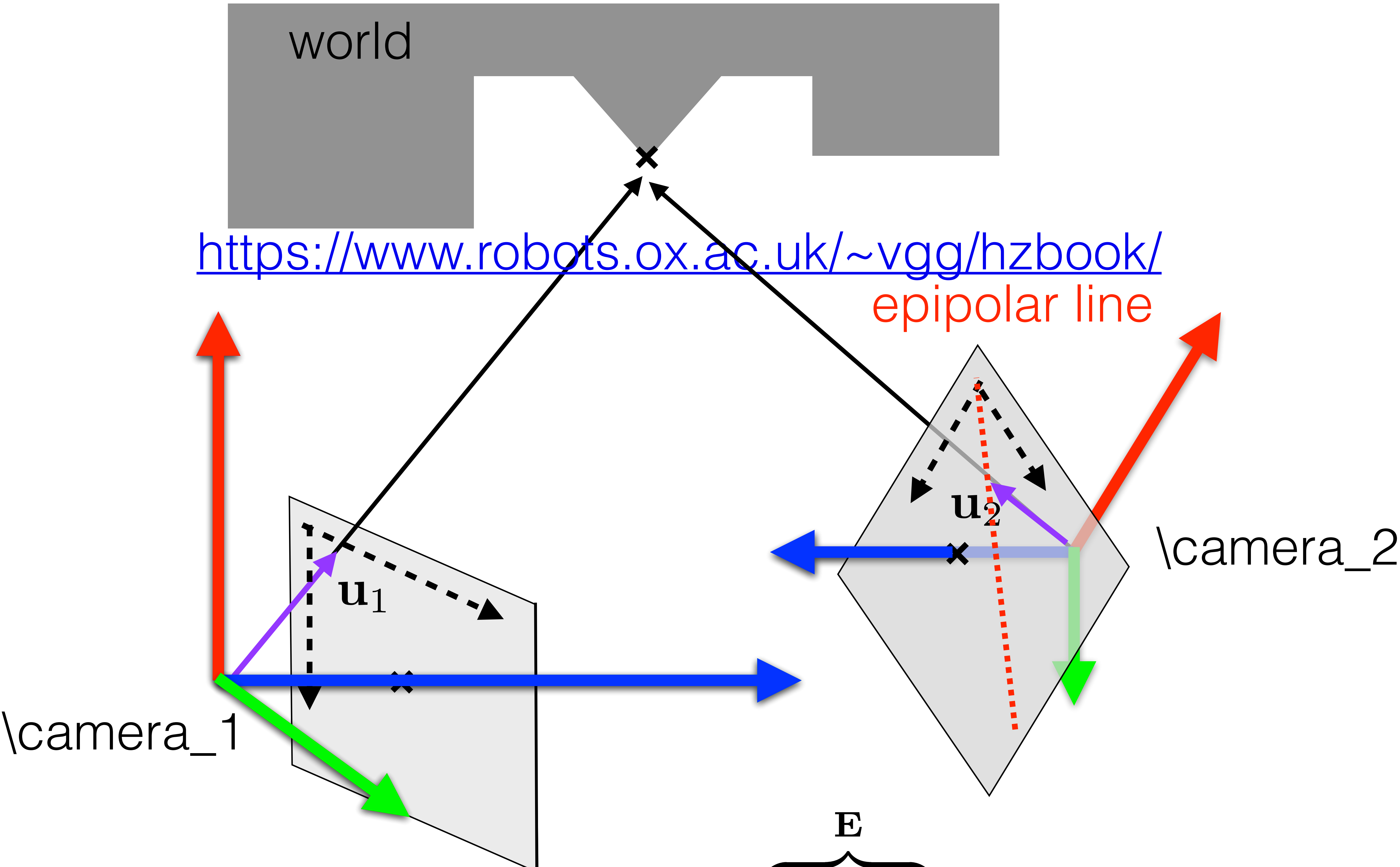
$\mathbf{x} \in \mathcal{W}$

$\mathbf{x} \in \mathcal{W}$
$\mathbf{e} \in \mathcal{E}$

Similarity function:

$$\sum_{\mathbf{x}\in\mathcal{W}} \Big(J(\mathbf{e}+\mathbf{x}) - I(\mathbf{x})\Big)^2$$

Calibration of the stereo



https://www.robots.ox.ac.uk/~vgg/hzbook/

$$\mathcal{E} = \{\mathbf{u}_2 \in \mathcal{R}^2 \mid \; \overline{\mathbf{u}}_2^\top \underbrace{\mathbf{K}^{-\top} \overbrace{(\mathbf{R} \times \mathbf{t})}^{\mathbf{E}} \mathbf{K}^{-1}}_{\mathbf{F}} \overline{\mathbf{u}}_1 = 0\}$$

Stereo: summary

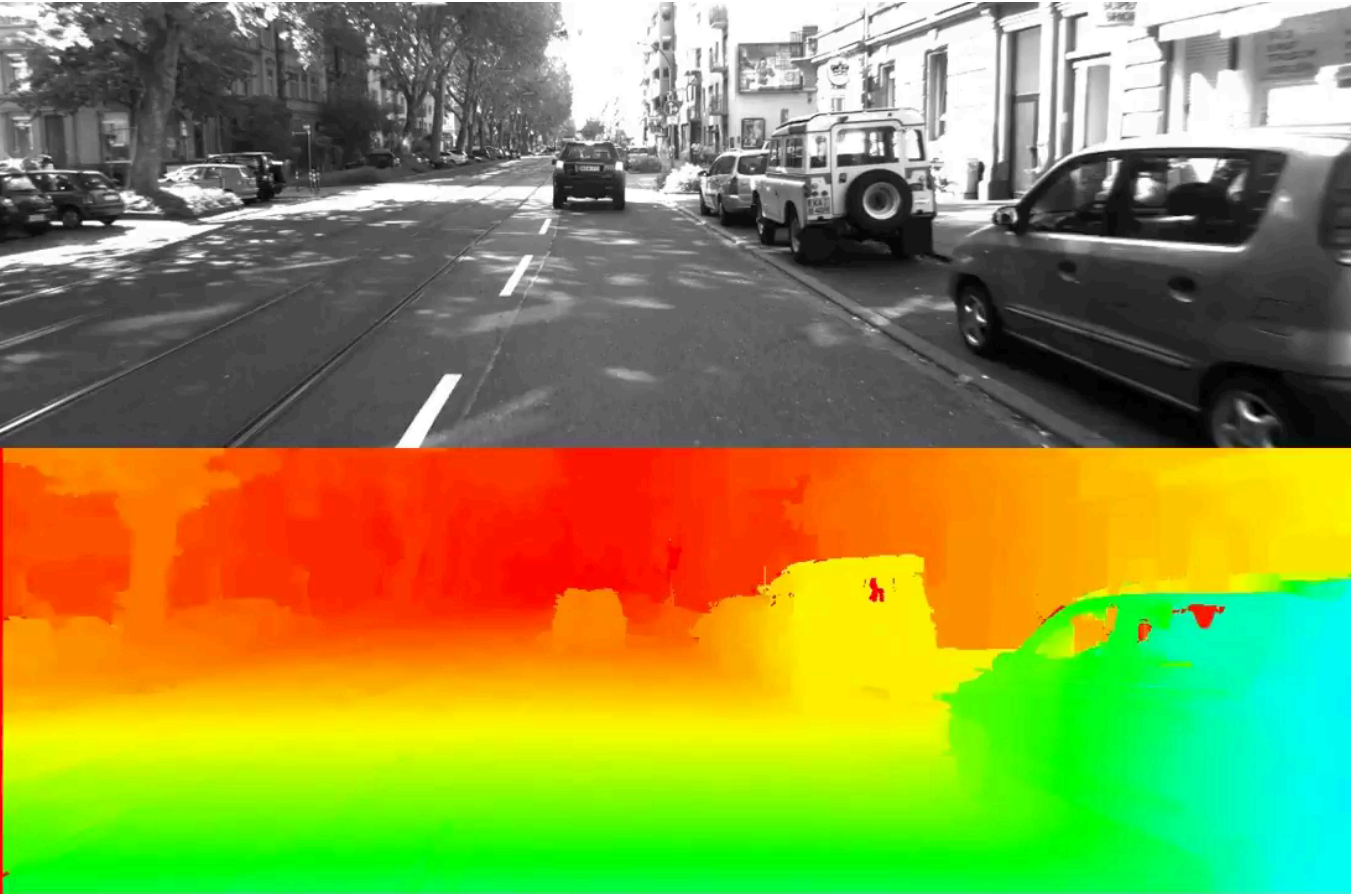
- Passive depth sensor created from pair of cameras.
- Inaccurate on long distance (due to limited resolution).
- Works well on textured, not reflective, smooth surfaces.
- Computationally demanding optimisation.
- Some OpenCV implementation:

```
stereo = cv2.createStereoBM(numDisparities=16,
blockSize=15)
depth = stereo.compute(imgL,imgR)
```

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_depthmap/py_depthmap.html

https://docs.opencv.org/3.1.0/d3/d14/tutorial_ximgproc_disparity_filtering.html#gsc.tab=0

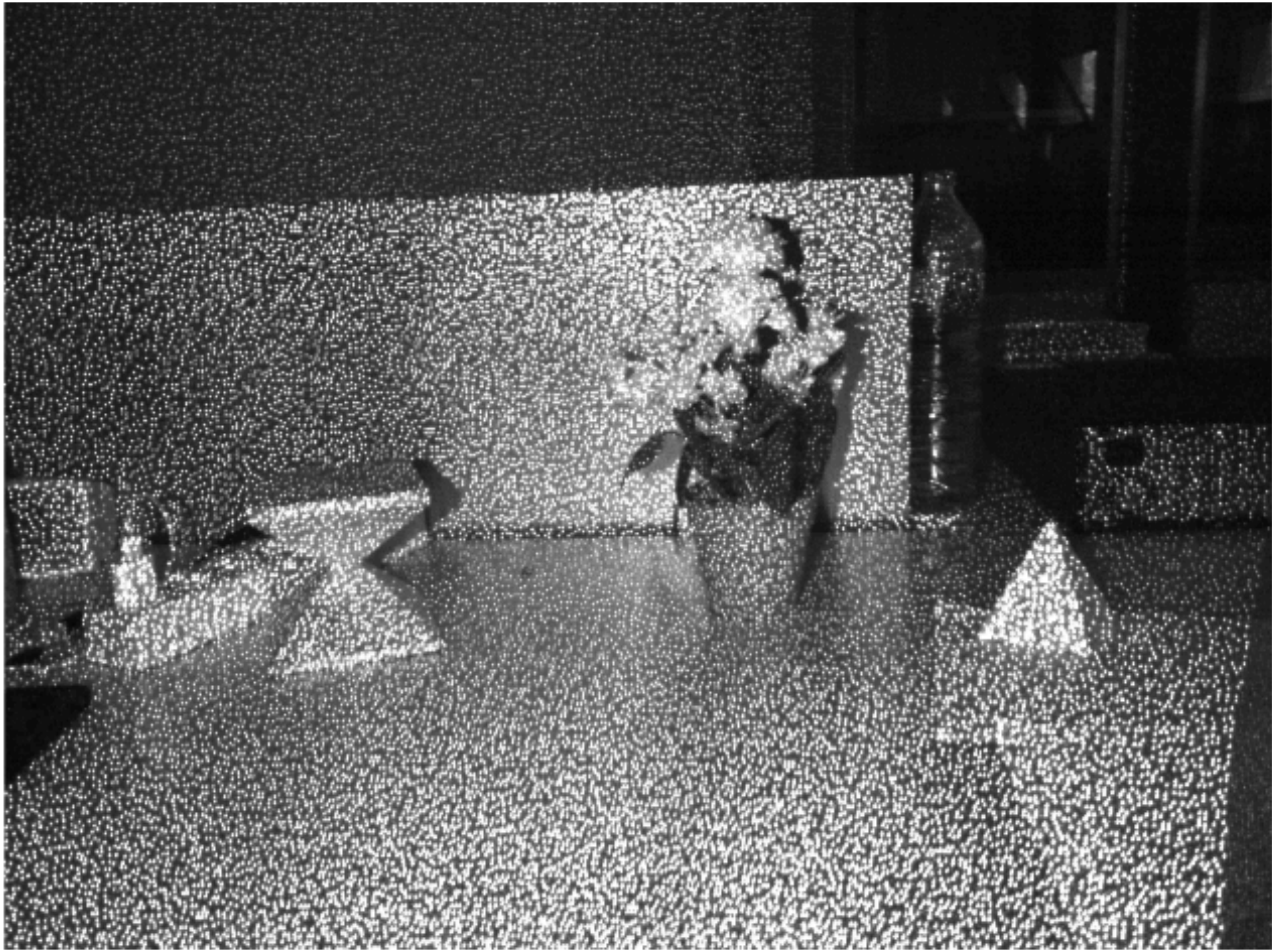
Stereo: summary



Kinect

- **Stereo** looks at the same object two-times and estimate its depth from two RGB images.
- **Kinect** avoid ambiguity by actively projecting a unique IR pattern on the object and search for its known appearance in the IR camera.

Kinect

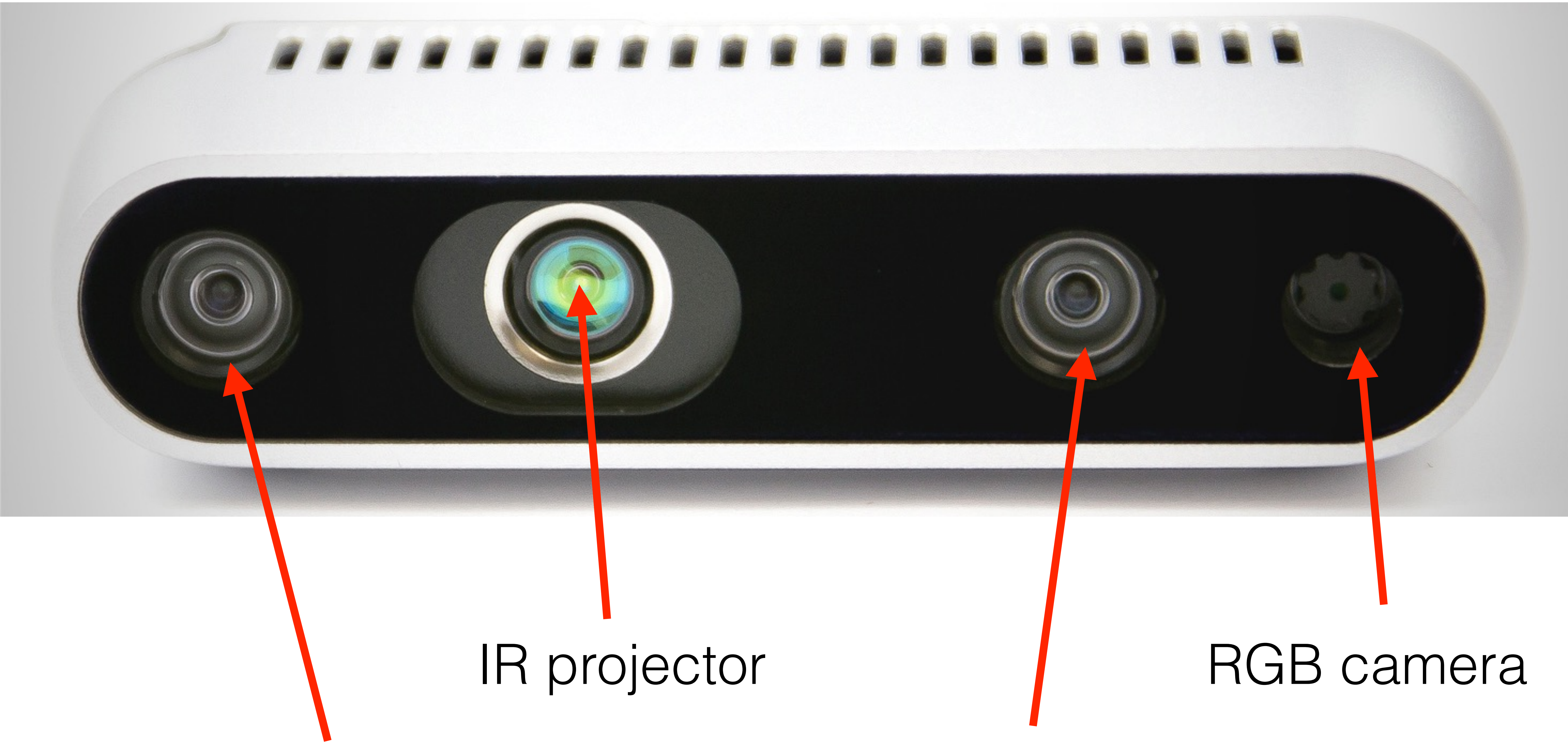


- Correspondence between projected patch and observed patch lies on the epipolar line.

Summary: Kinect

- Active depth sensor consisting of IR camera and projector.
- Does not work outdoor due to strong illumination.
- Inaccurate on long distances.
- It does not require well textured surface.
- Cheap and fast solution for indoor robotics.

RealSense

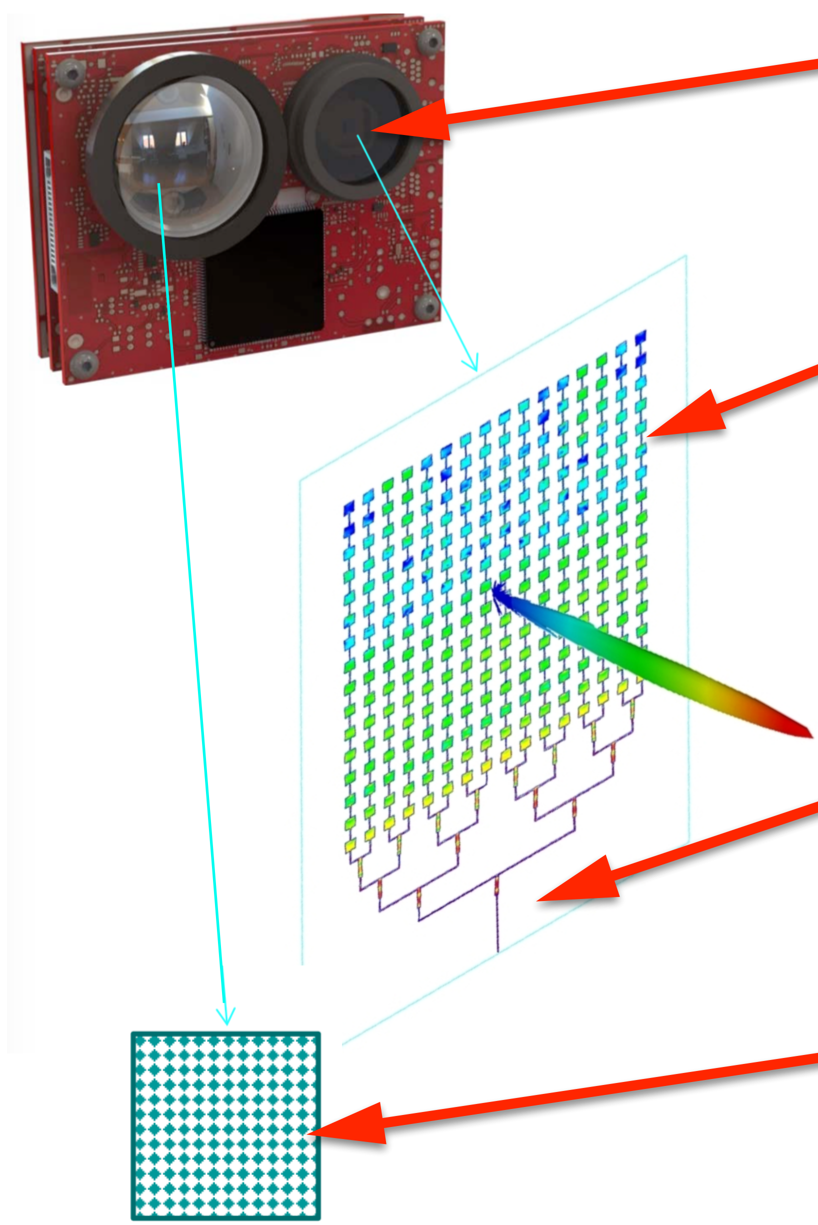


- **Indoor:** IR projector avoid ambiguities by projecting unique IR pattern. It works like stereo enhanced by IR pattern pattern.
- **Outdoor:** It is normal stereo

Solid-state lidar

Lidar with independent steering of depth-measuring rays

S3 principle



Emitted laser beams

Transmitted through Optical Phased Array

Controlling optical properties of OPA elements, allows to steer laser beams in desired directions

Reflected laser beams are captured by SPAD array

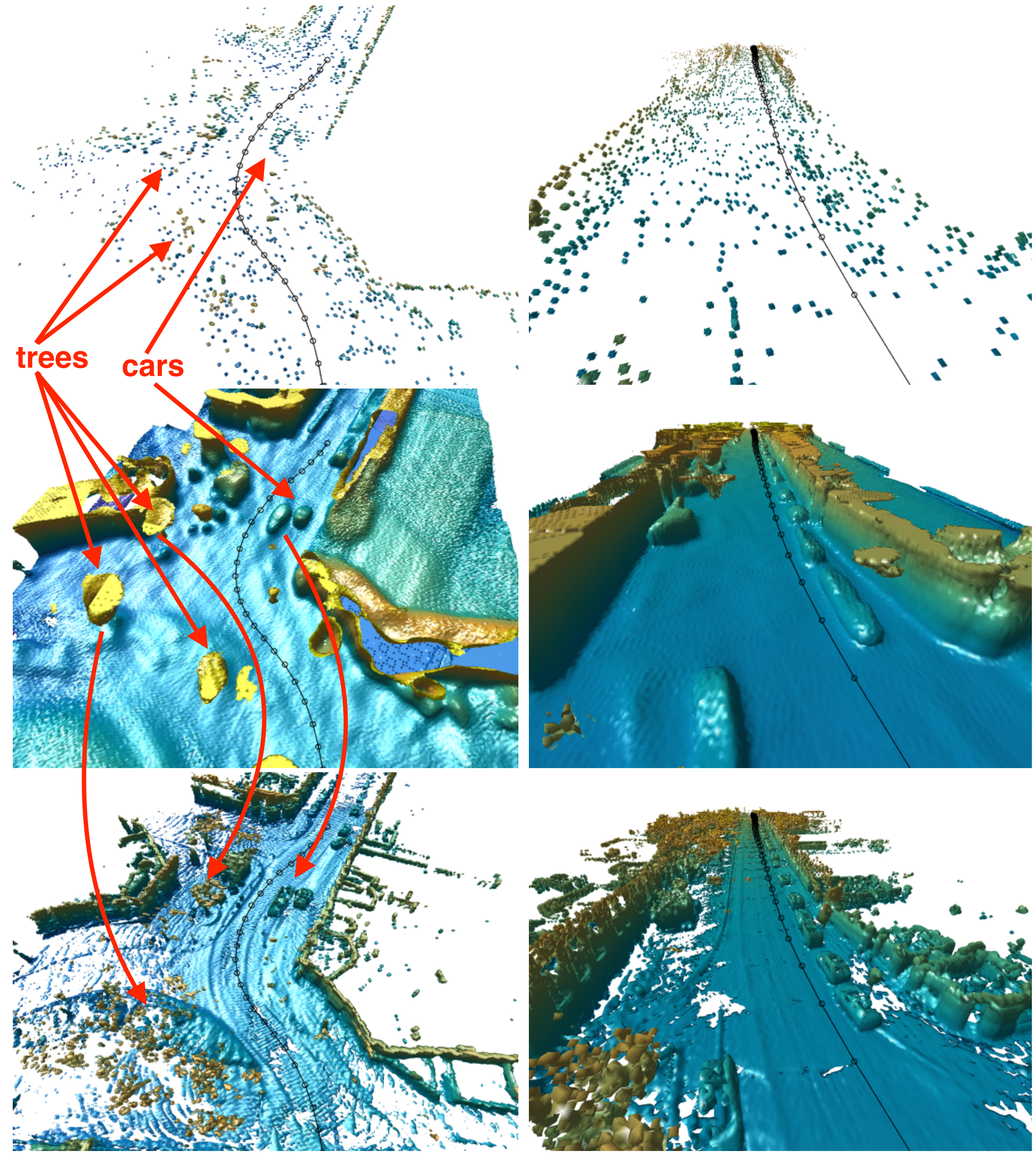
Images of S3 Lidar redistributed with permission of Quanergy Systems (http://quanergy.com)

Experiment: Qualitative evaluation

Sparse measurements

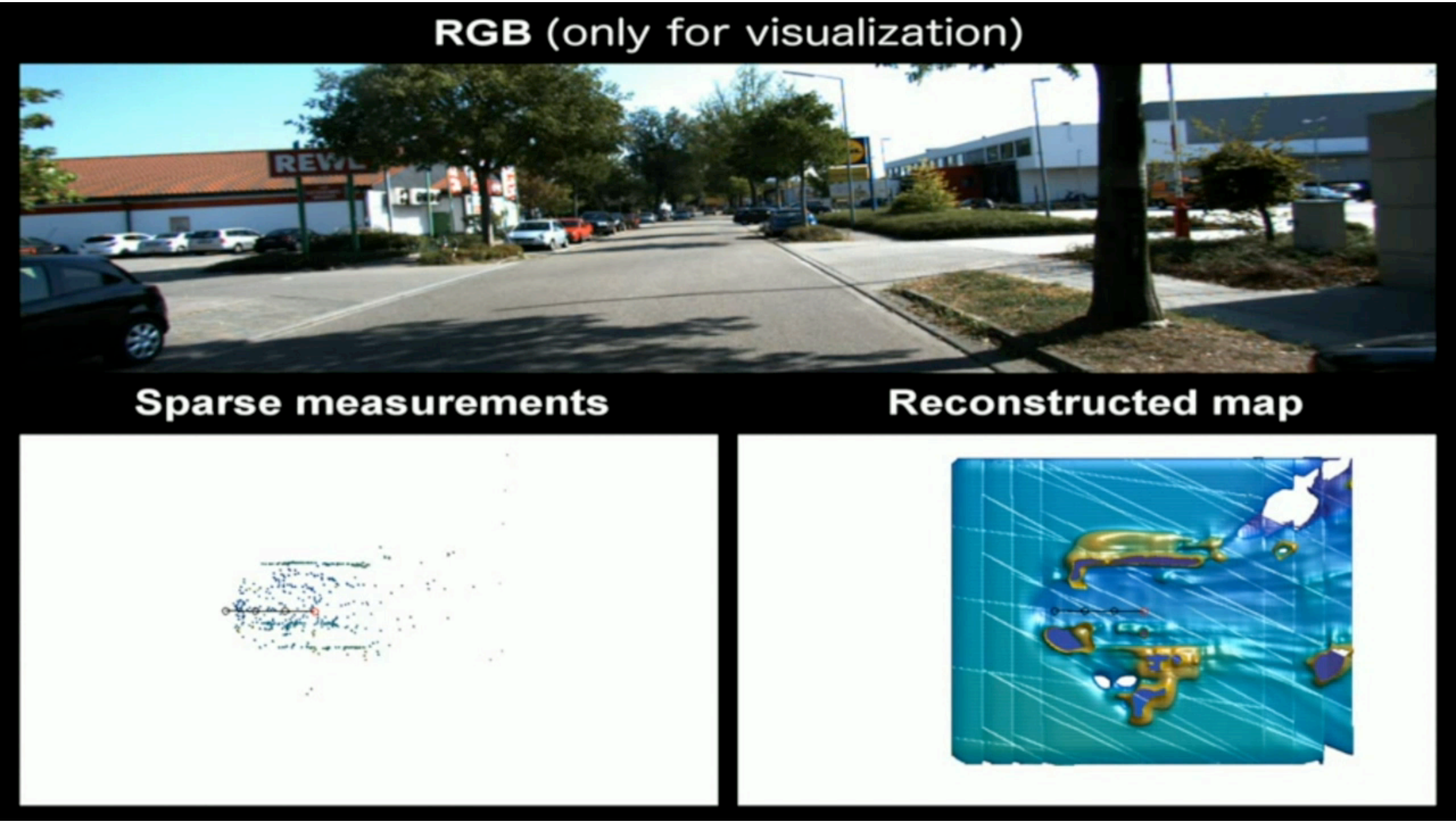
Reconstructed map

Ground truth



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

Active mapping [Zimmermann, Petricek et al. ICCV 2017]



[1] Zimmermann, Petricek, Salansky, Svoboda, Learning for Active 3D Mapping, **ICCV oral**, 2017 https://arxiv.org/abs/1708.02074