Projective Line

2D Point and Line Duality

The 2D line joining two points:

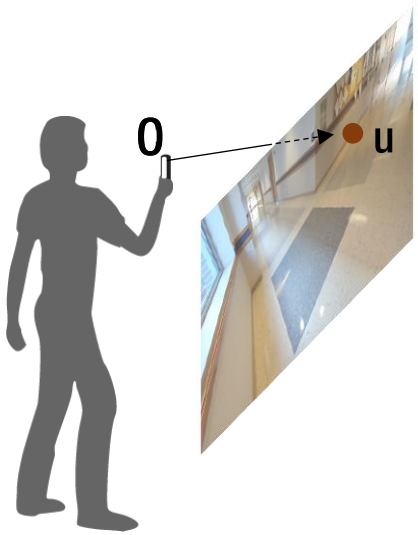
$$\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$$

The intersection between two lines:

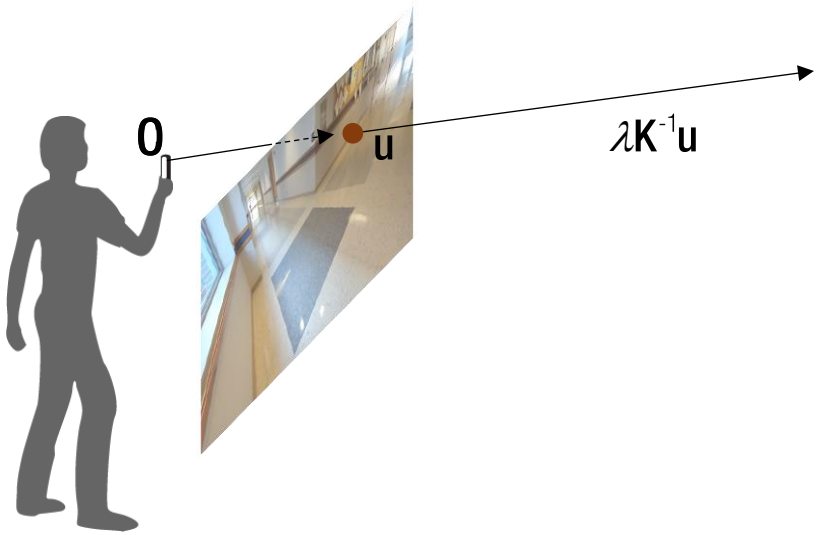
$$\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$$

Given any formula, we can switch the meaning of point and line to get another formula.

Geometric Interpretation (Point)



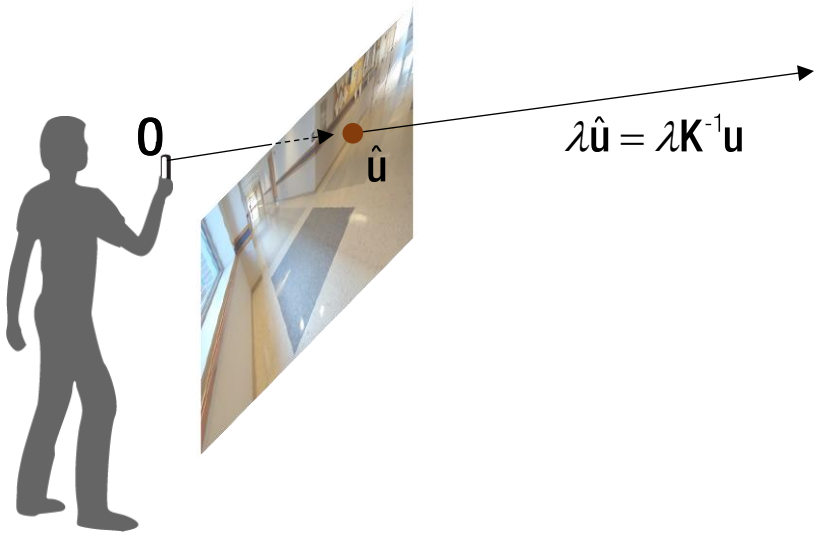
Geometric Interpretation (Point)



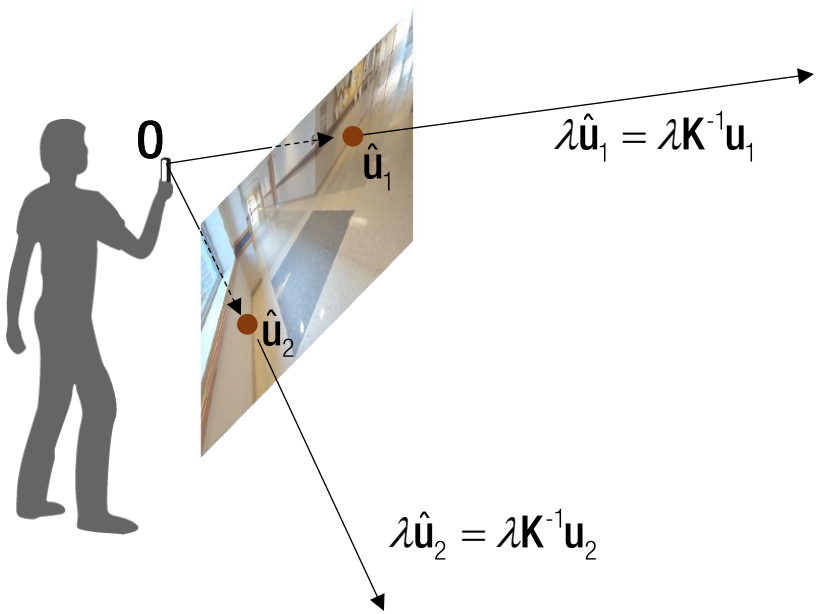
Geometric Interpretation (Point)

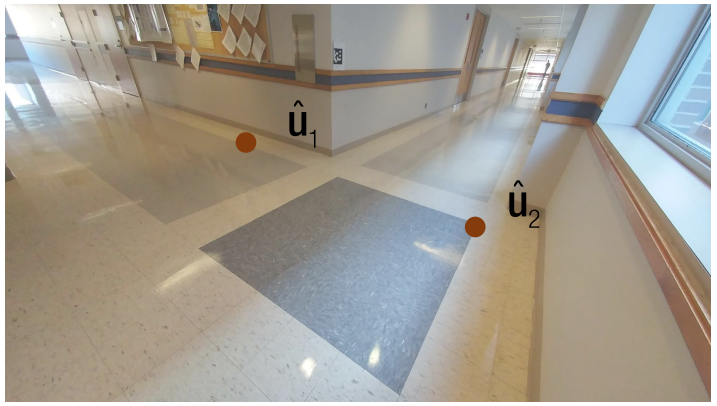
Normalized coordinate:

$$\hat{\mathbf{u}} = \mathbf{K}^{-1}\mathbf{u}$$



Geometric Interpretation (Point)

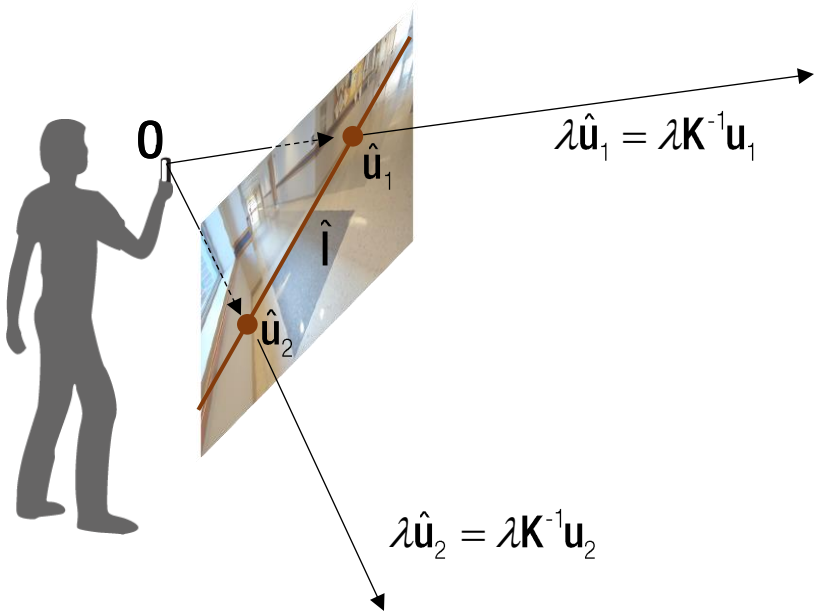


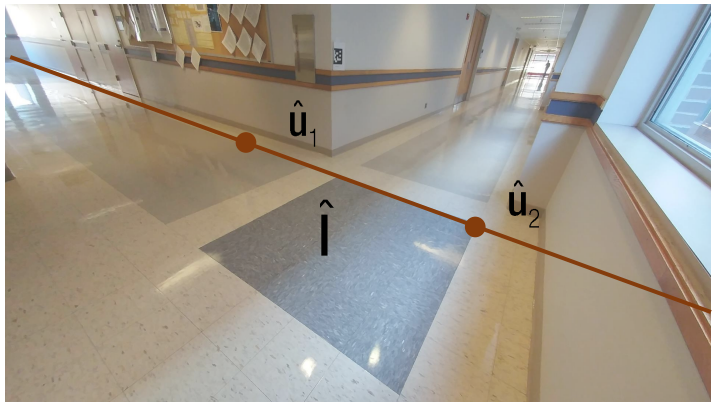


Normalized coordinate:

$$\hat{u}_1 = K^{-1}u_1 \qquad \hat{u}_2 = K^{-1}u_2$$

Geometric Interpretation (Line)





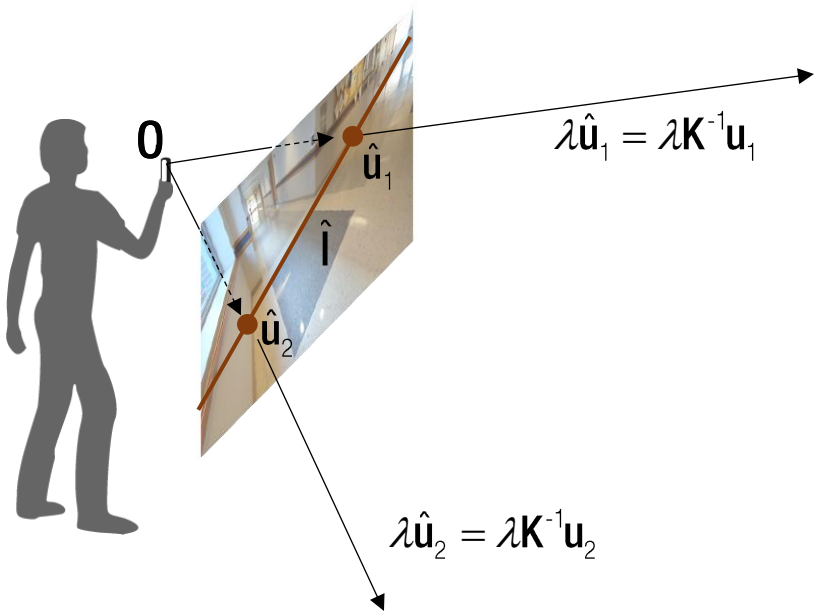
Normalized coordinate:

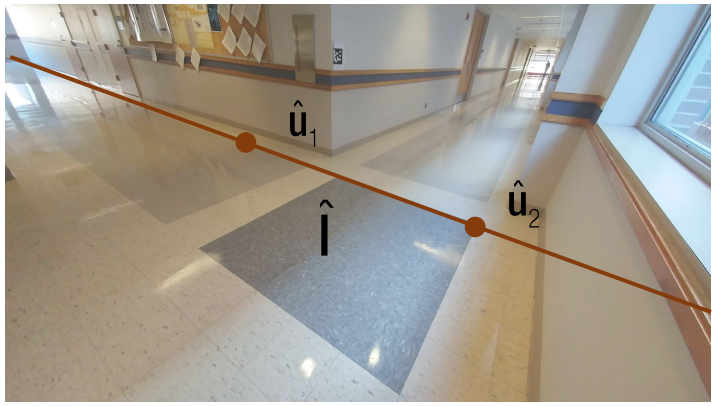
$$\hat{\mathbf{u}}_1 = \mathbf{K}^{-1}\mathbf{u}_1 \qquad \hat{\mathbf{u}}_2 = \mathbf{K}^{-1}\mathbf{u}_2$$

$$\longrightarrow \quad \hat{\mathbf{l}} = \hat{\mathbf{u}}_1 \times \hat{\mathbf{u}}_2$$

where $\hat{\mathbf{l}} = \ ?$

Geometric Interpretation (Line)





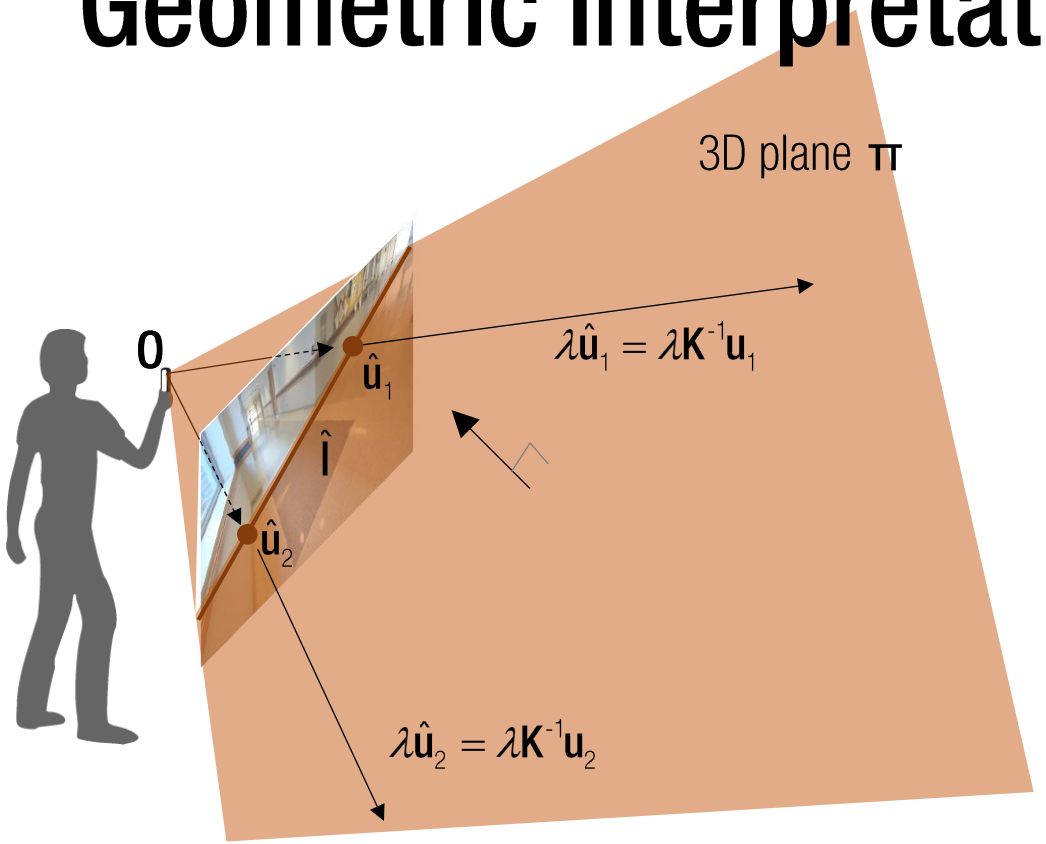
Normalized coordinate:

$$\hat{\mathbf{u}}_1 = \mathbf{K}^{-1}\mathbf{u}_1 \qquad \hat{\mathbf{u}}_2 = \mathbf{K}^{-1}\mathbf{u}_2$$

$$\longrightarrow \quad \hat{\mathbf{l}} = \hat{\mathbf{u}}_1 \times \hat{\mathbf{u}}_2$$

where $\hat{\mathbf{l}} = \left(\mathbf{K}^{-1}\right)^{-\mathsf{T}} \mathbf{l} = \mathbf{K}^{\mathsf{T}}\mathbf{l}$ due to duality

Geometric Interpretation (Line)



Normalized coordinate:

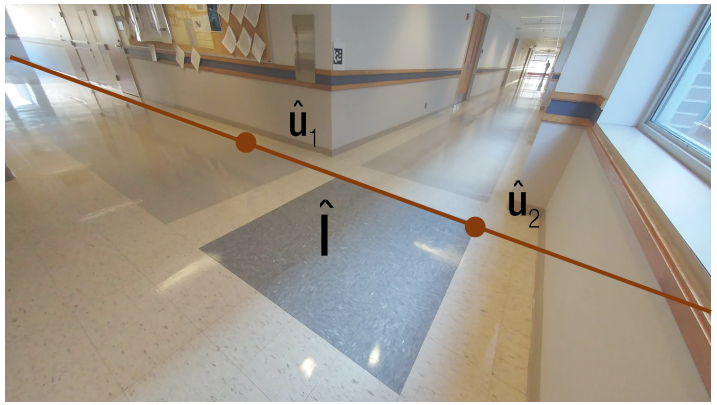
$$\hat{\mathbf{u}}_1 = \mathbf{K}^{-1}\mathbf{u}_1 \qquad \hat{\mathbf{u}}_2 = \mathbf{K}^{-1}\mathbf{u}_2$$

$$\longrightarrow \quad \hat{\mathbf{l}} = \hat{\mathbf{u}}_1 \times \hat{\mathbf{u}}_2$$

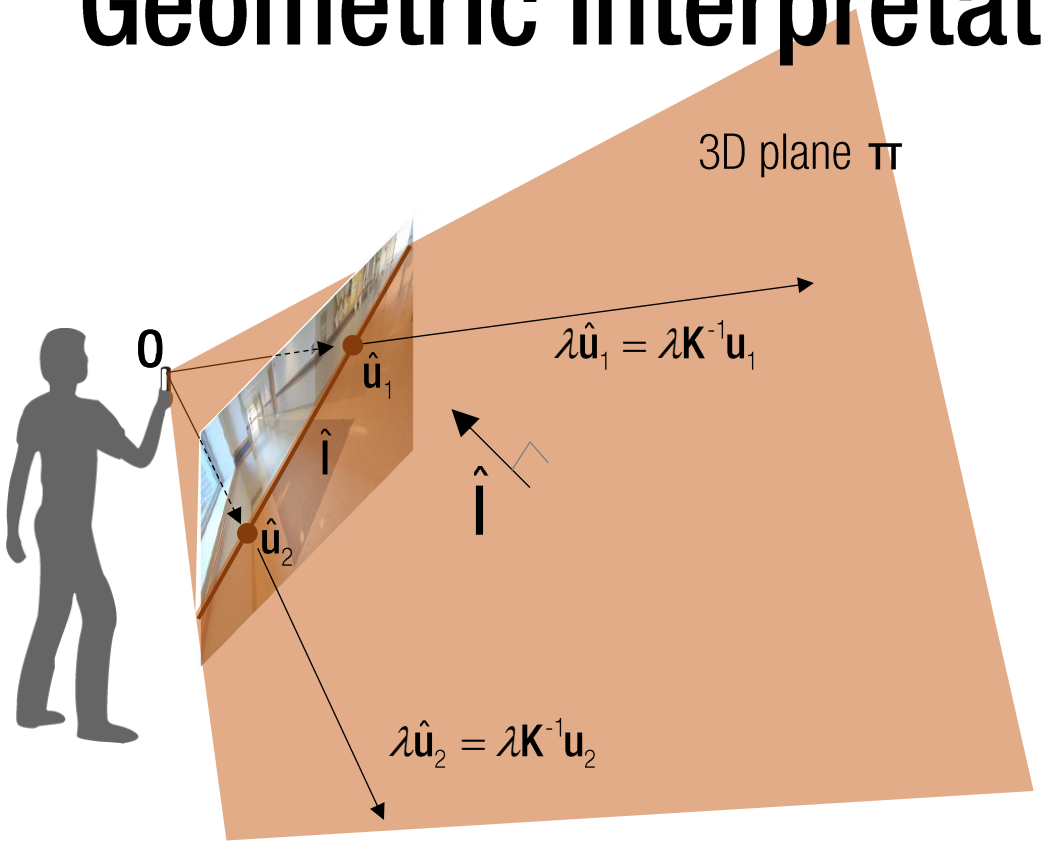
where $\hat{\mathbf{l}} = \left(\mathbf{K}^{-1}\right)^{-\mathsf{T}}\mathbf{l} = \mathbf{K}^{\mathsf{T}}\mathbf{l}$ due to duality

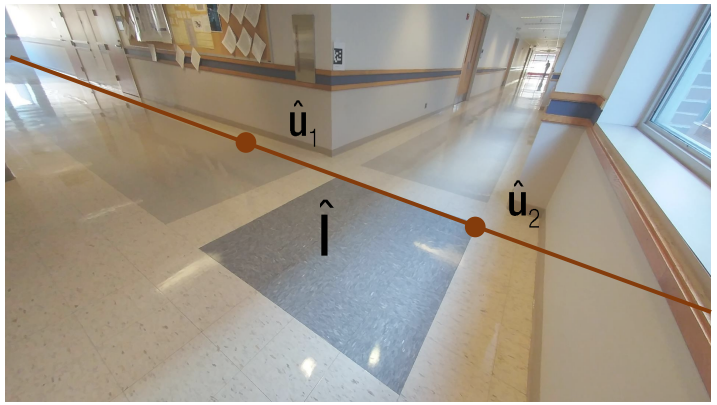
A 2D line in an image defines to a 3D plane passing the camera center:

$$\hat{\mathbf{l}} \rightarrow \pi$$



Geometric Interpretation (Line)





Normalized coordinate:

$$\hat{\mathbf{u}}_1 = \mathbf{K}^{-1}\mathbf{u}_1 \qquad \hat{\mathbf{u}}_2 = \mathbf{K}^{-1}\mathbf{u}_2$$

$$\longrightarrow \quad \hat{\mathbf{l}} = \hat{\mathbf{u}}_1 \times \hat{\mathbf{u}}_2$$

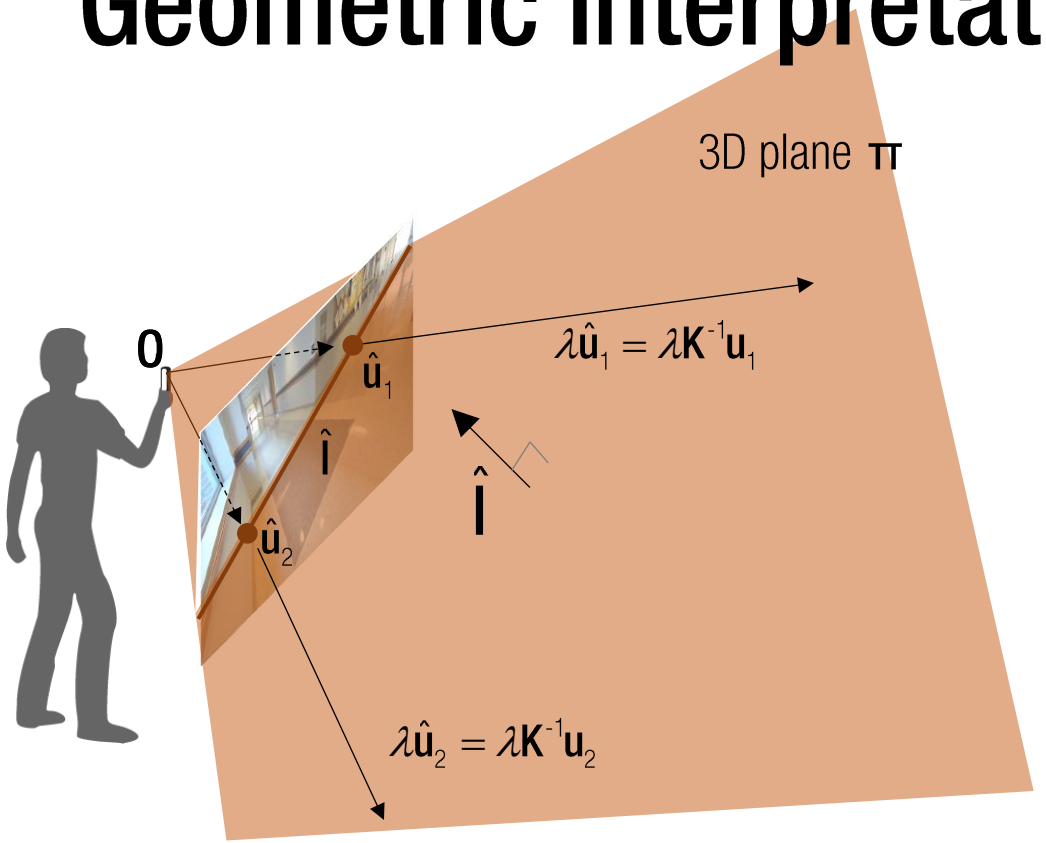
where $\hat{\mathbf{l}} = \left(\mathbf{K}^{-1}\right)^{-\mathsf{T}}\mathbf{l} = \mathbf{K}^{\mathsf{T}}\mathbf{l}$ due to duality

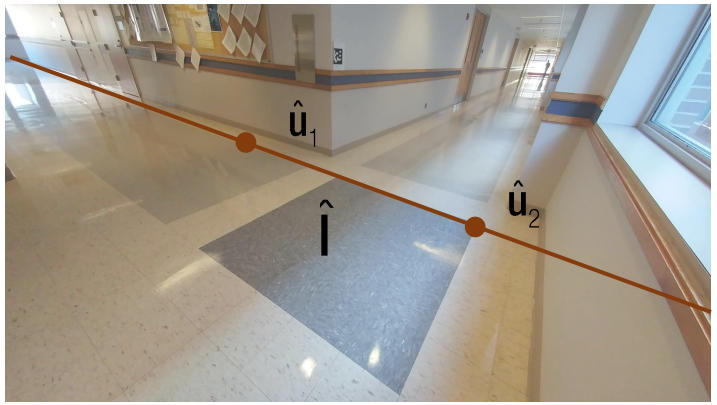
A 2D line in an image defines to a 3D plane passing the camera center:

$$\hat{\mathbf{l}} \rightarrow \pi$$

Plane normal: $\quad ? \quad = \lambda\hat{\mathbf{l}}$

Geometric Interpretation (Line)





Normalized coordinate:

$$\hat{u}_1 = K^{-1}u_1 \qquad \hat{u}_2 = K^{-1}u_2$$

$$\longrightarrow \quad \hat{l} = \hat{u}_1 \times \hat{u}_2$$

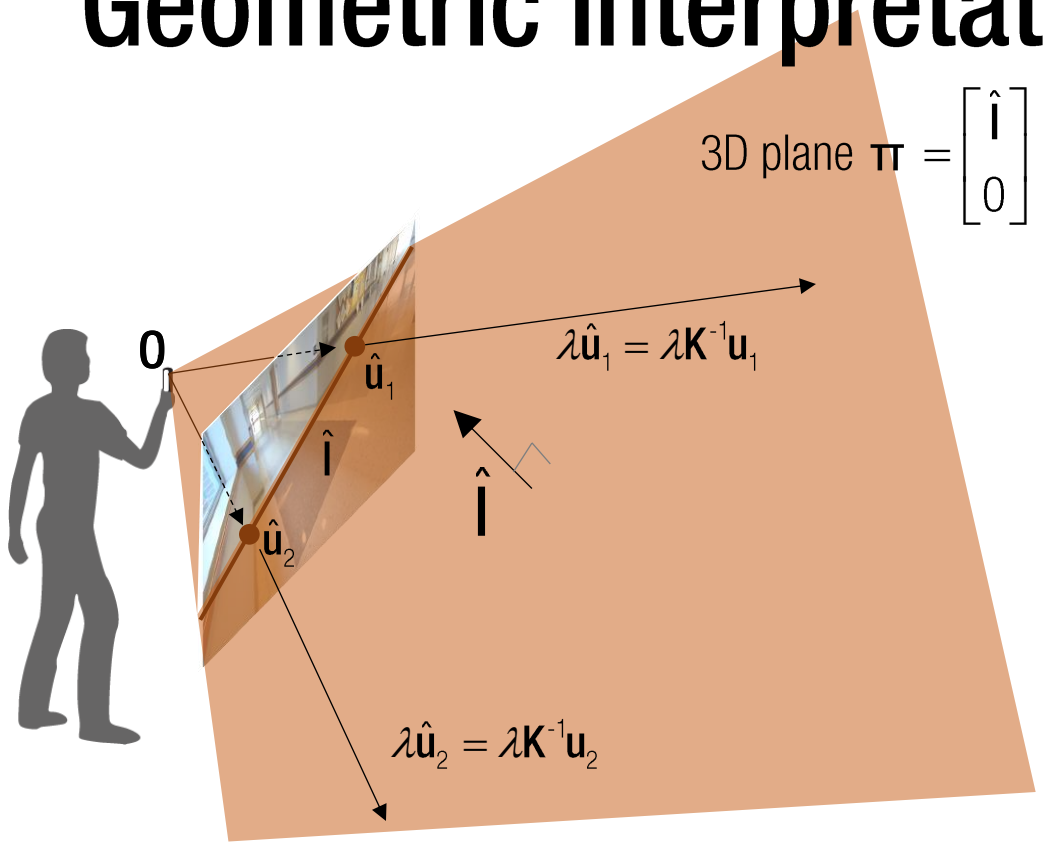
where $\hat{l} = \left(K^{-1}\right)^{-T} l = K^T l$ due to duality

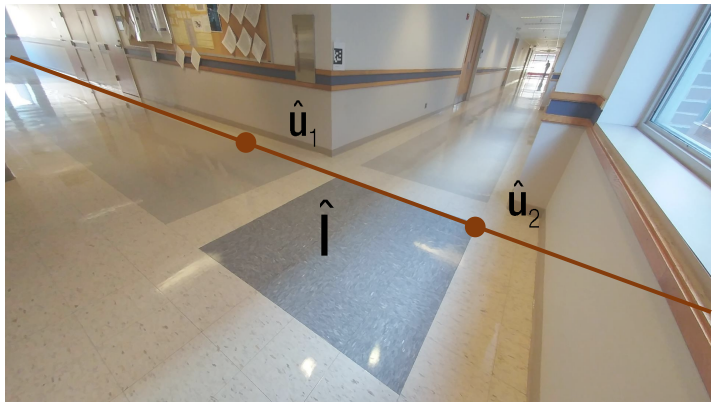
A 2D line in an image defines to a 3D plane passing the camera center:

$$\hat{l} \rightarrow \pi$$

Plane normal: $\left(\lambda_1\hat{u}_1\right) \times \left(\lambda_2\hat{u}_2\right) = \lambda\hat{l}$

Geometric Interpretation (Line)





Normalized coordinate:

$$\hat{u}_1 = K^{-1}u_1 \qquad \hat{u}_2 = K^{-1}u_2$$

$$\longrightarrow \quad \hat{l} = \hat{u}_1 \times \hat{u}_2$$

where $\hat{l} = \left(K^{-1}\right)^{-T} l = K^T l$ due to duality

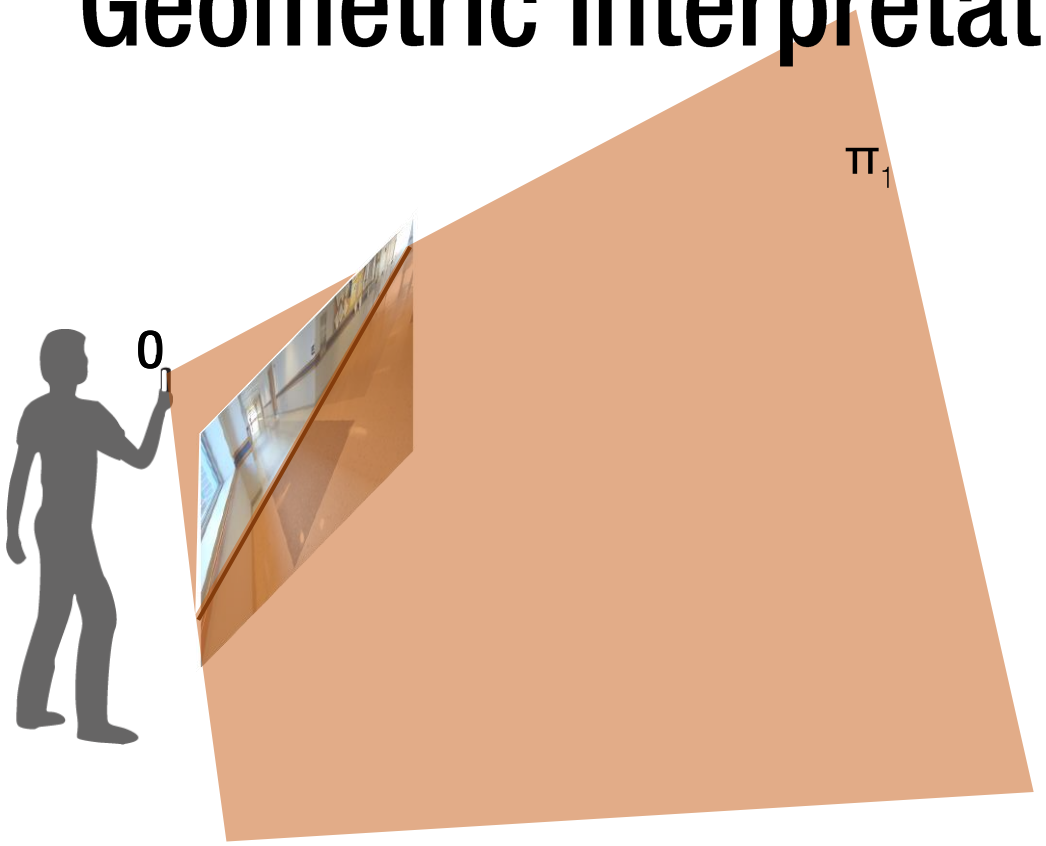
A 2D line in an image defines to a 3D plane passing the camera center:

$$\hat{l} \rightarrow \pi$$

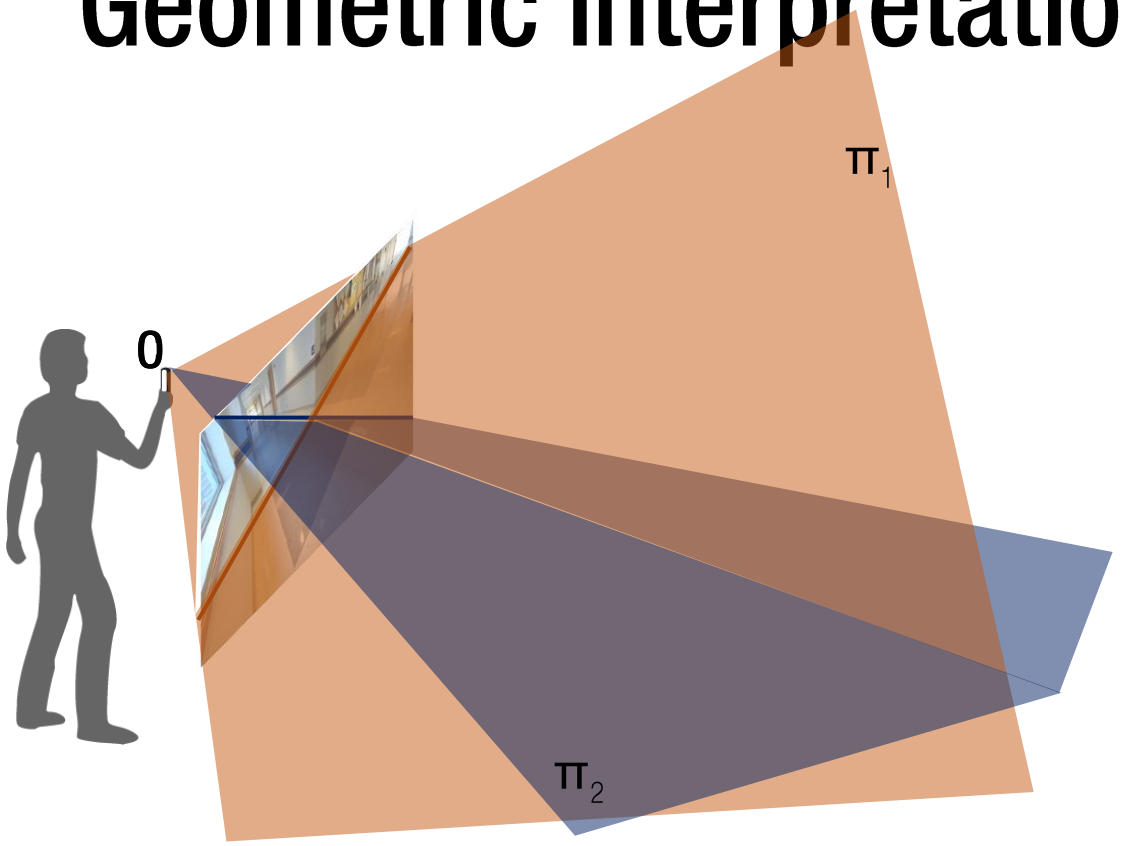
Plane normal: $\left(\lambda_1\hat{u}_1\right) \times \left(\lambda_2\hat{u}_2\right) = \lambda\hat{l}$

$$\therefore \pi = \begin{bmatrix} \hat{l} \\ 0 \end{bmatrix}$$

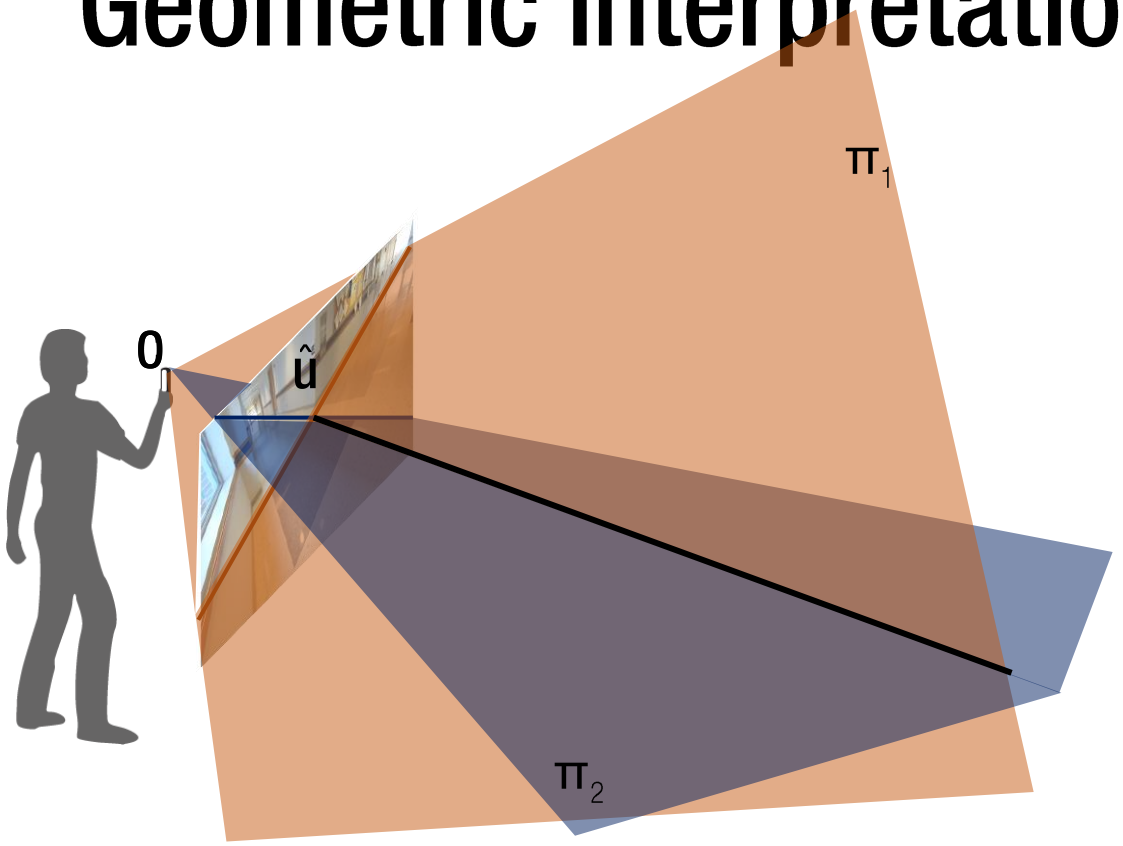
Geometric Interpretation (Line-Line)



Geometric Interpretation (Line-Line)

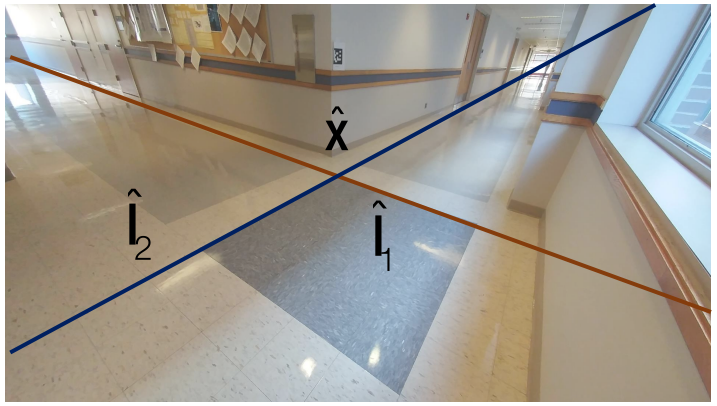


Geometric Interpretation (Line-Line)

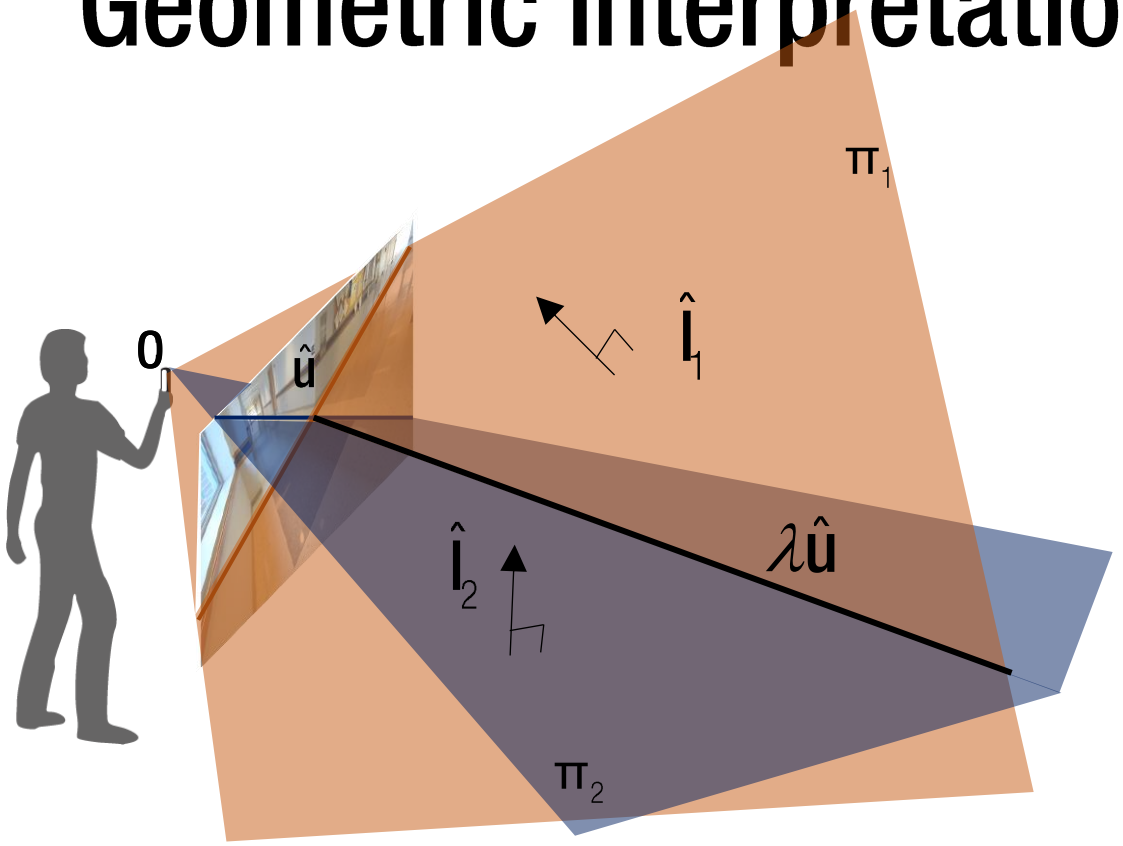


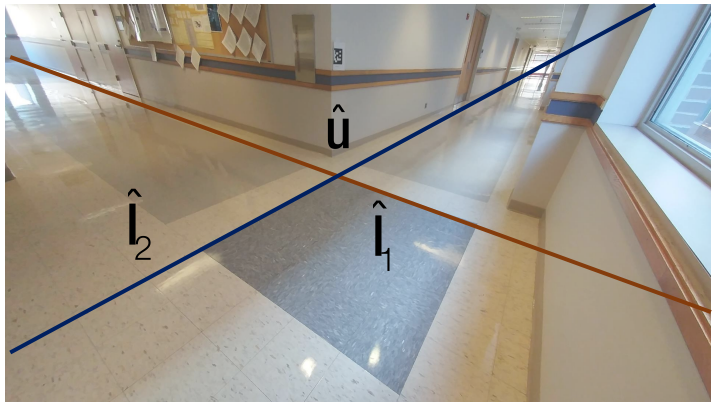
2D lines in an image intersect a 2D point corresponding to a 3D ray:

$$\hat{u} = \hat{l}_1 \times \hat{l}_2$$



Geometric Interpretation (Line-Line)



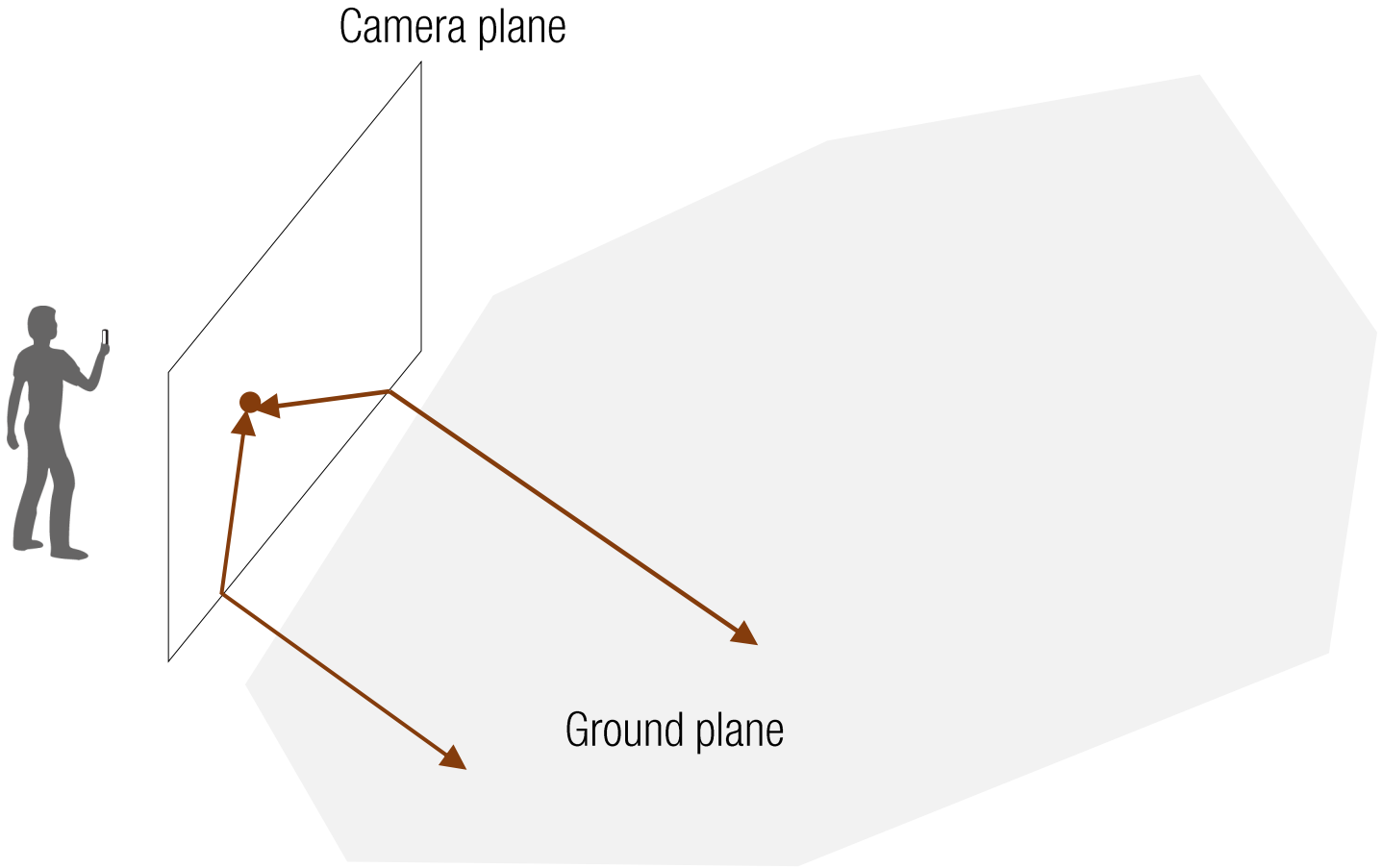


2D lines in an image intersect a 2D point corresponding to a 3D ray:

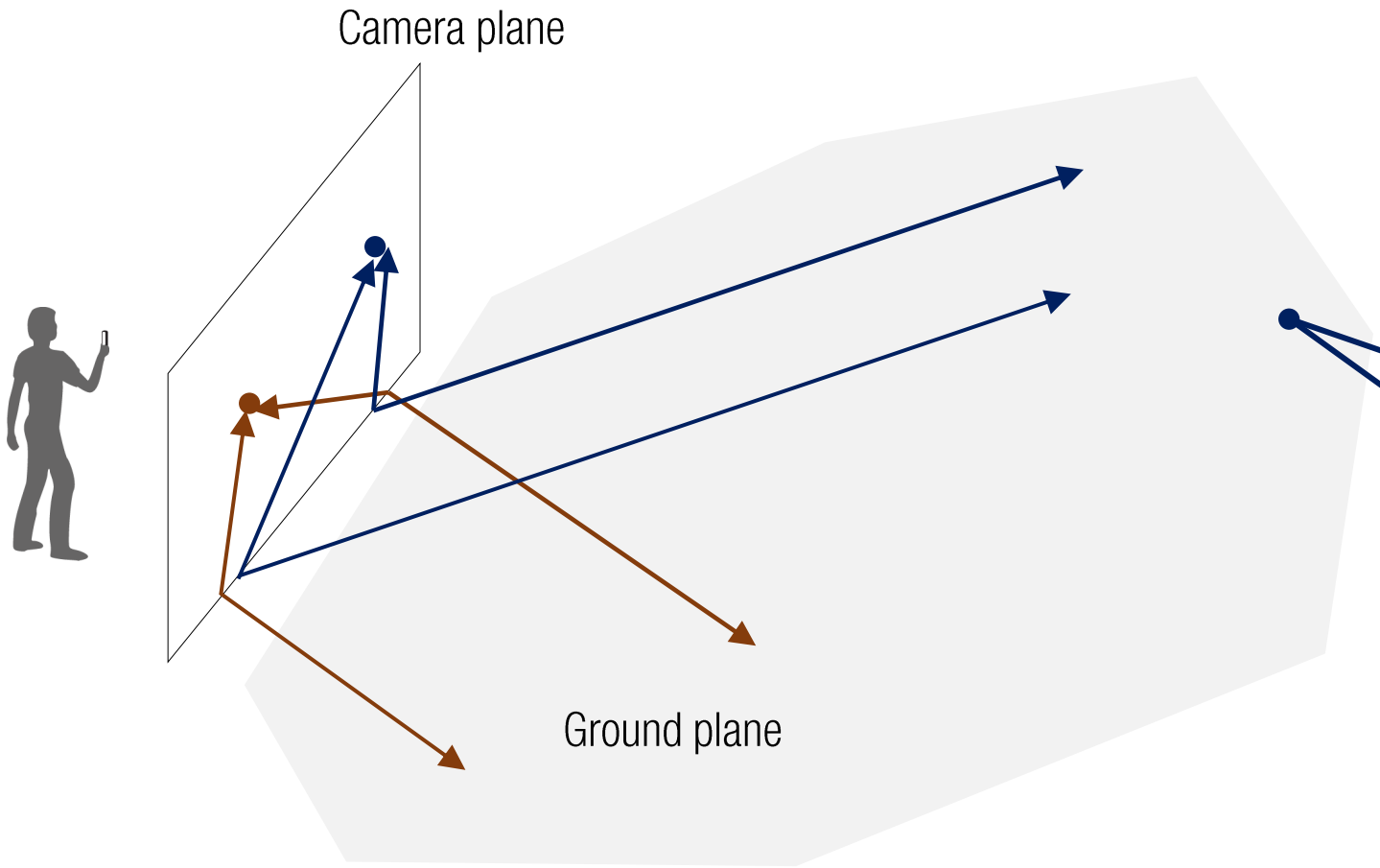
$$\hat{u} = \hat{l}_1 \times \hat{l}_2$$

: the 3D ray is perpendicular to two plane normals.

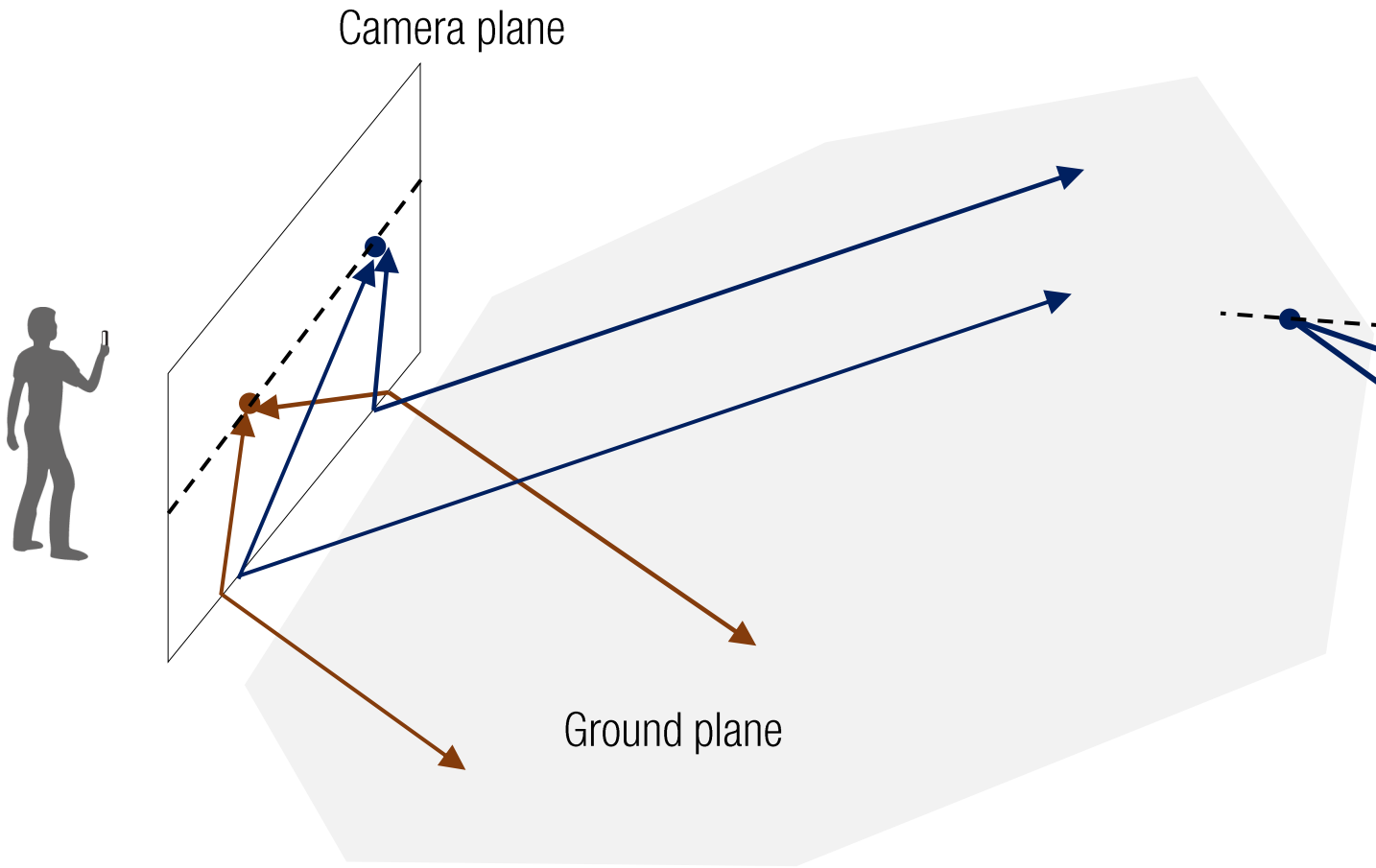
Geometric Interpretation of Vanishing Line



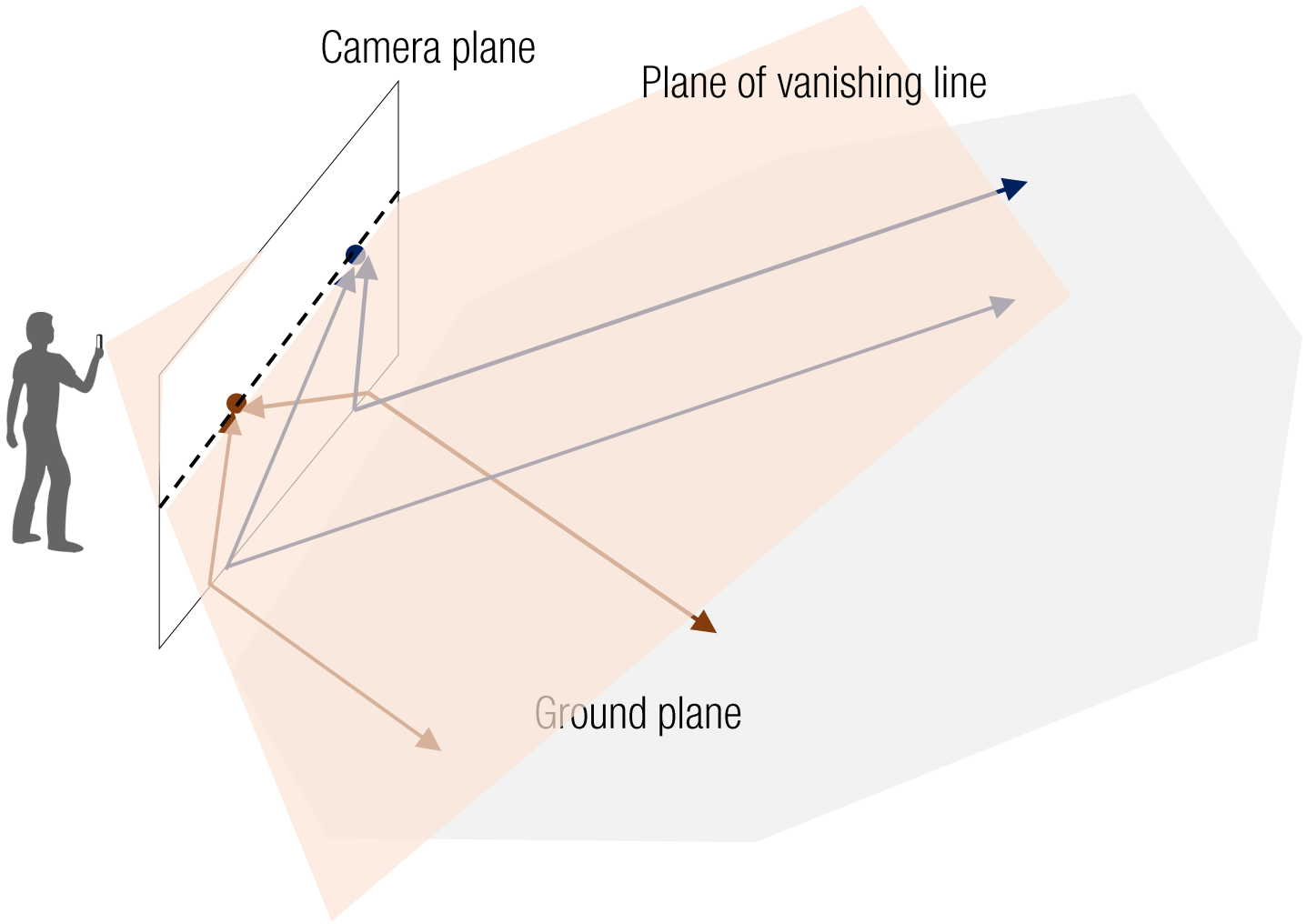
Geometric Interpretation of Vanishing Line

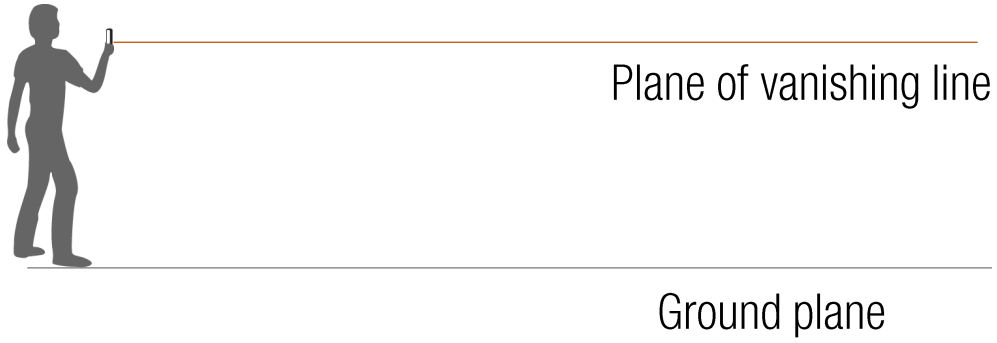


Geometric Interpretation of Vanishing Line



Geometric Interpretation of Vanishing Line





Side view

Geometric Interpretation of Vanishing Line

Side view

Where was I (how high)?

Where was I (how high)?

Taken from my hotel room (6th floor)

Taken from beach

Where was I (how high)?

Taken from my hotel room (6th floor)

Taken from beach

Where am I?
Using Vanishing Points

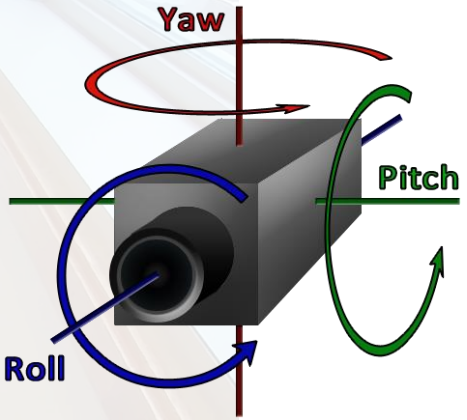
Vanishing point

Vanishing line for horizon

Vanishing point

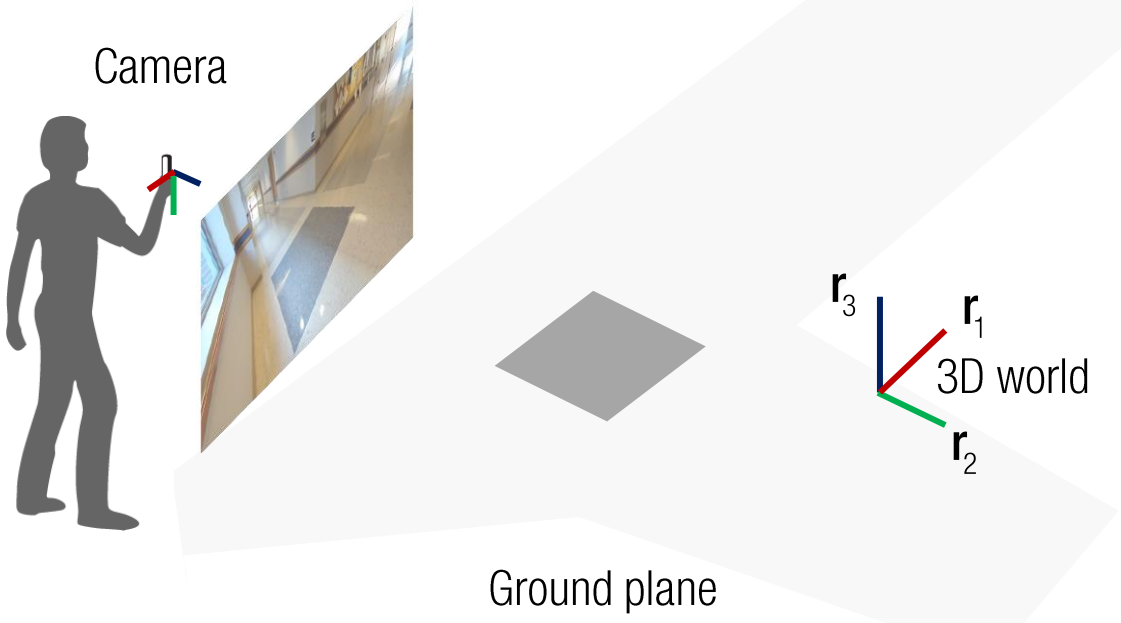
What can vanishing line tell us about me?

- Horizon
- Camera pitch angle (looking down)
- Camera roll angle (tilted toward right)



Keller Hall

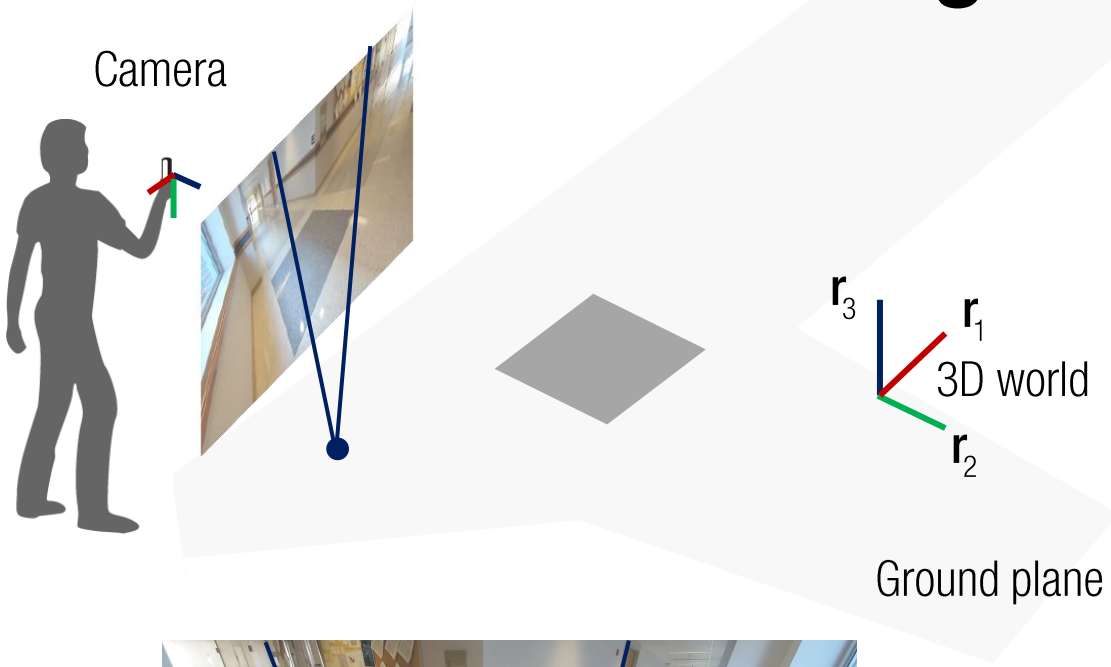
Where am I w.r.t. Ground Plane?

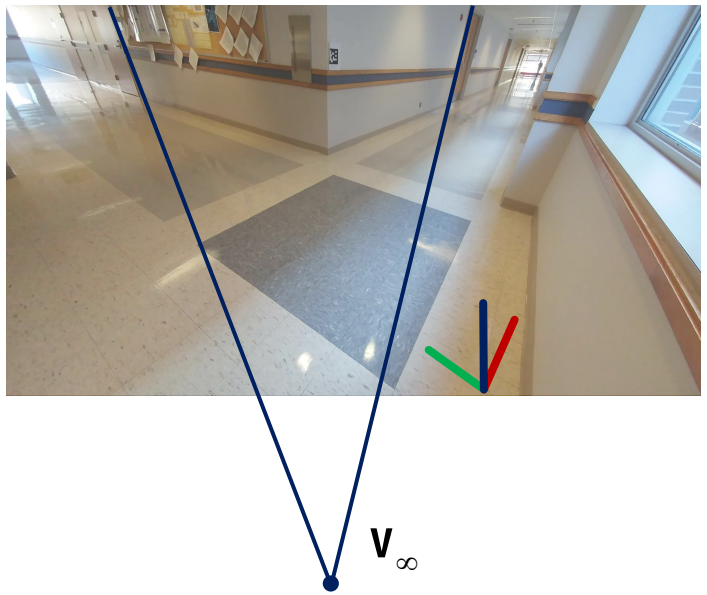


$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

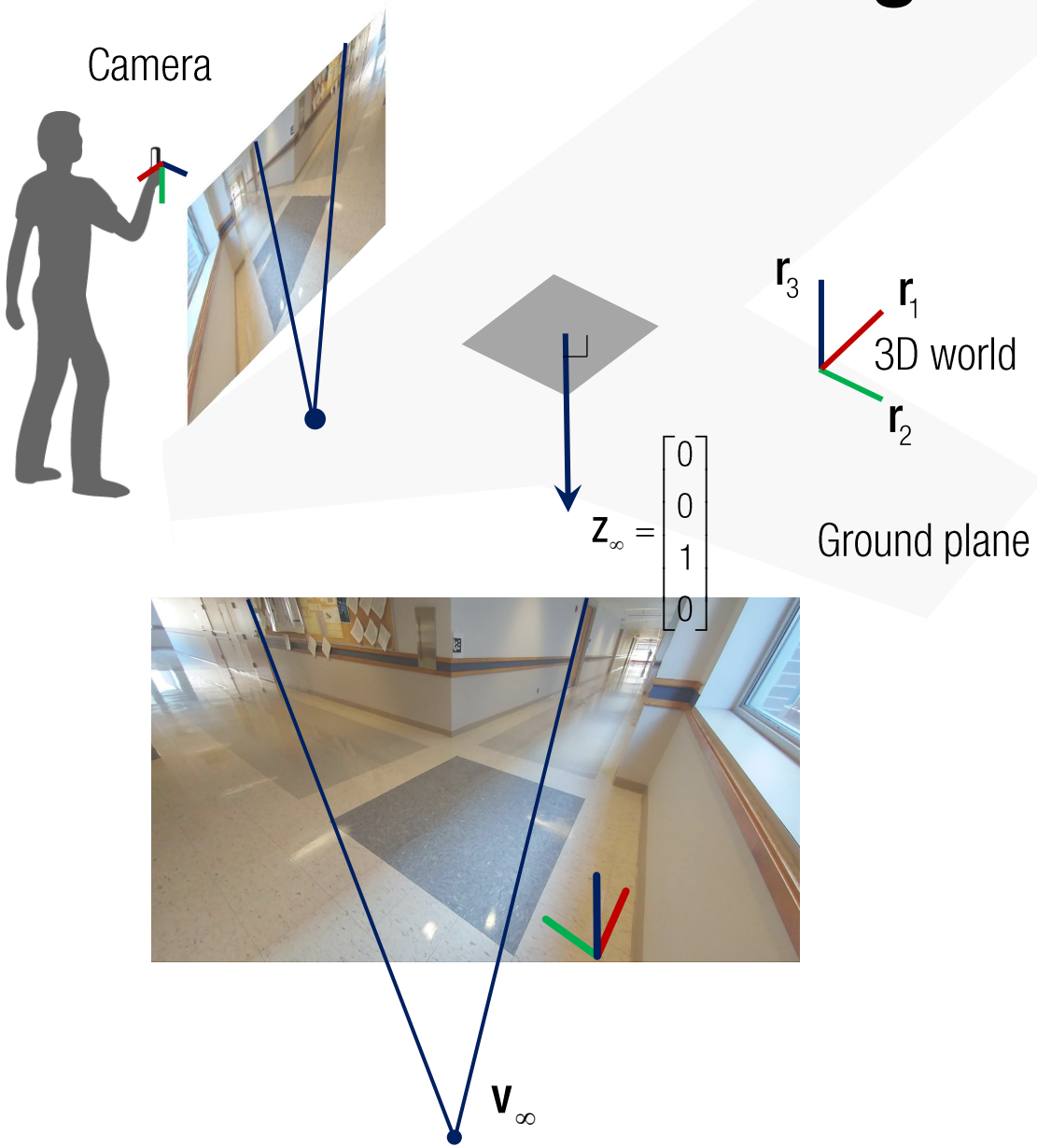
How to compute?

What can a Vanishing Point tell us about?

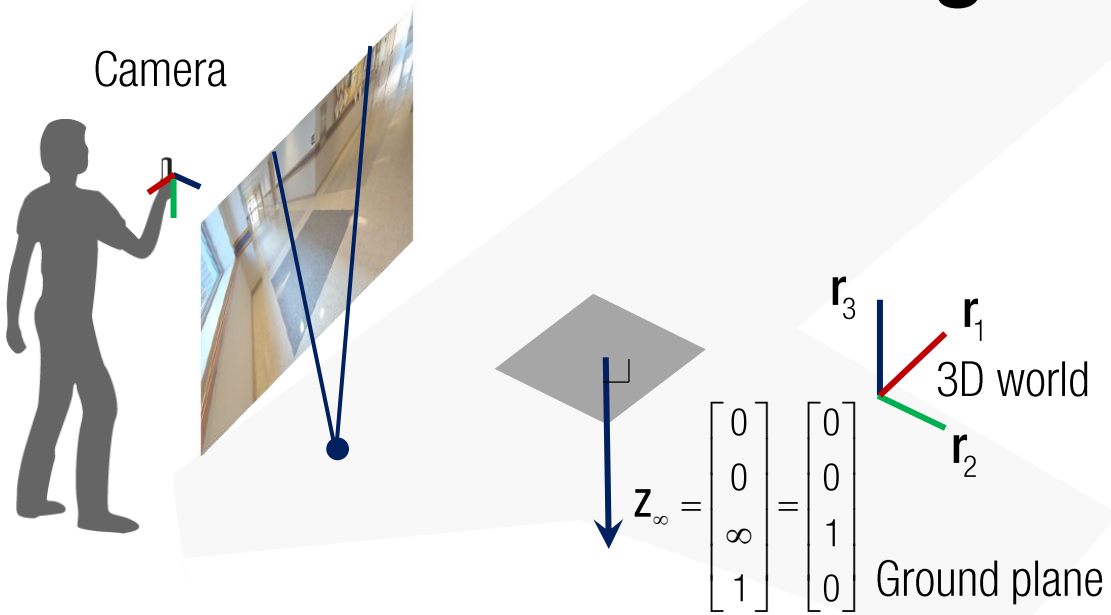




What can a Vanishing Point tell us about?



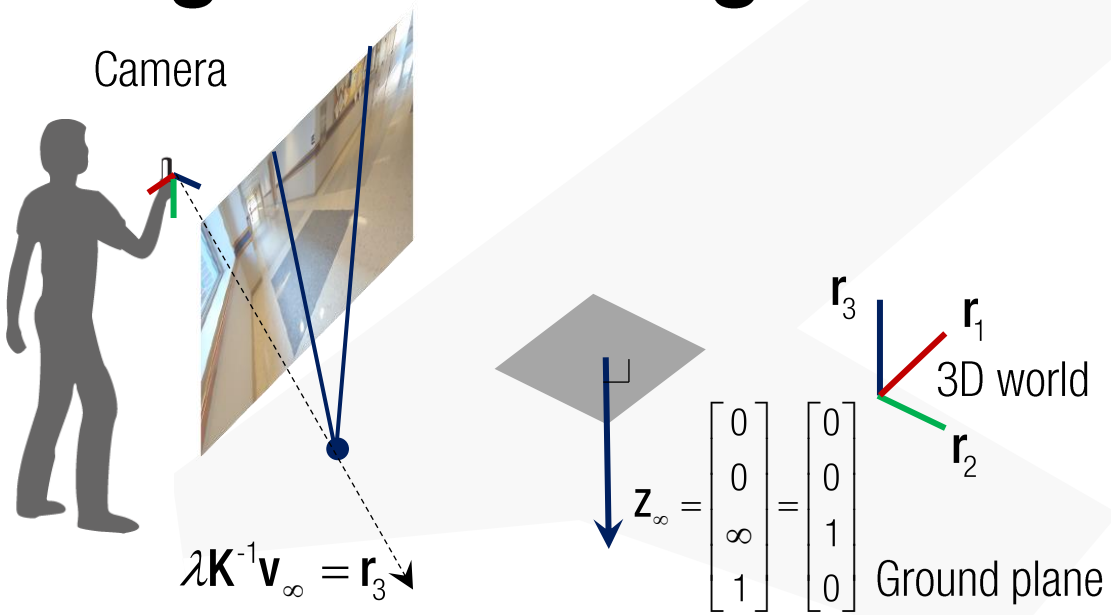
What can a Vanishing Point tell us about?



$$\lambda \mathbf{v}_\infty = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$\mathbf{v}_\infty$

Single Vanishing Point

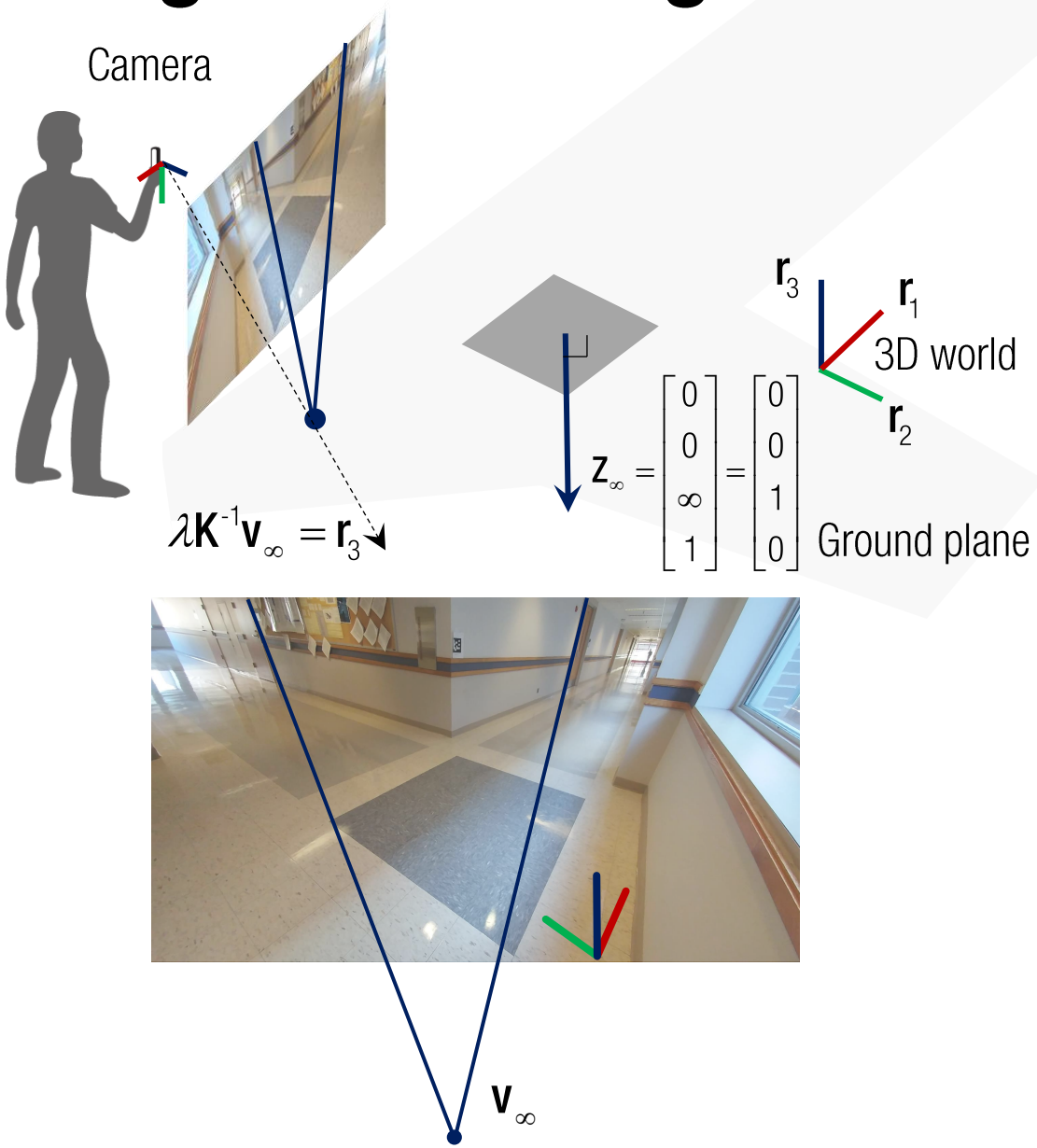


v_∞

$$\lambda \mathbf{v}_\infty = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\lambda \mathbf{v}_\infty = \mathbf{K}\mathbf{r}_3$$

Single Vanishing Point



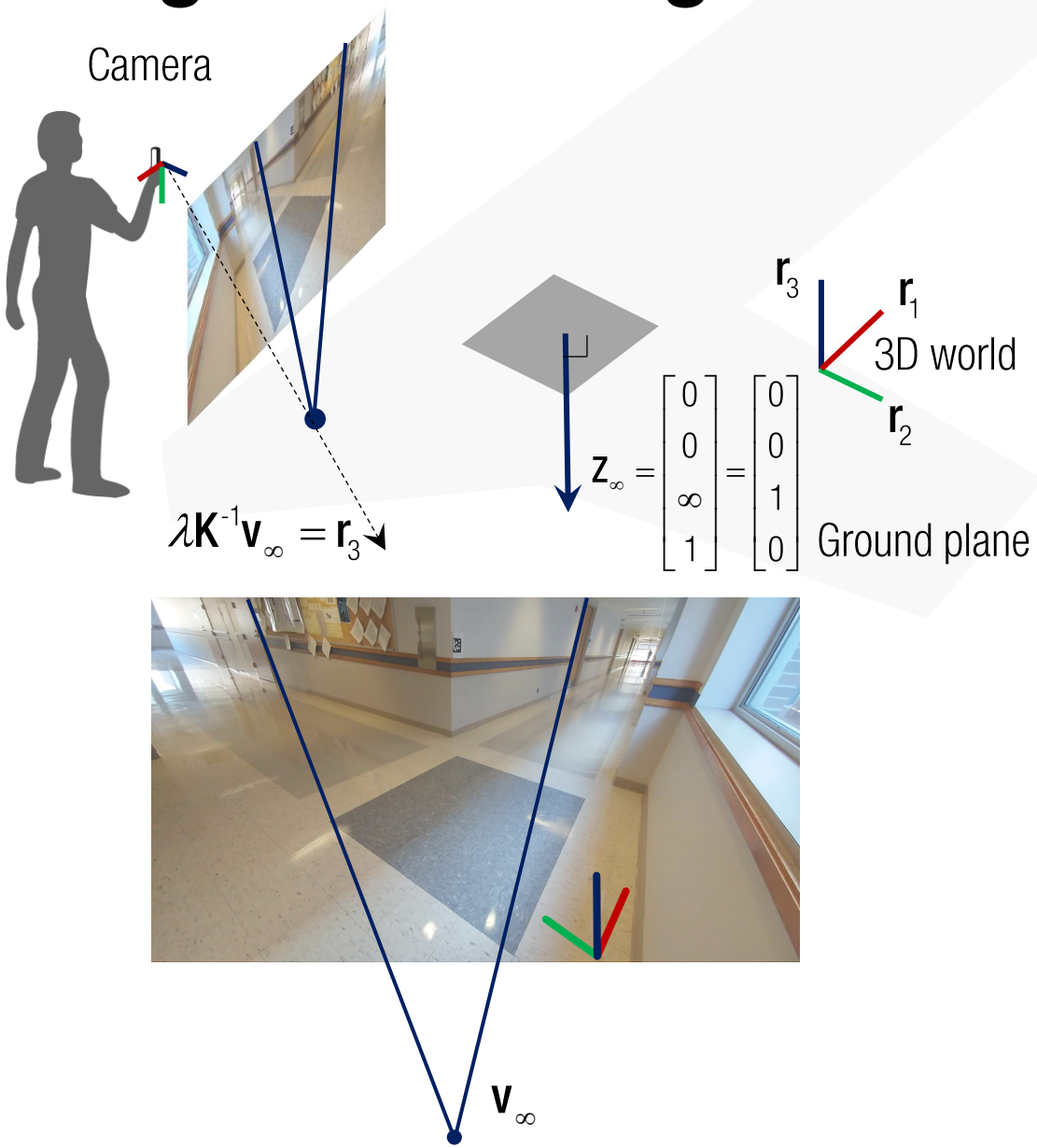
$$\lambda \mathbf{v}_\infty = \mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C\end{bmatrix}\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$$

$$\lambda \mathbf{v}_\infty = \mathbf{K}\mathbf{r}_3$$

$$\longrightarrow \mathbf{r}_3 = \frac{\mathbf{K}^{-1}\mathbf{v}_\infty}{\left\|\mathbf{K}^{-1}\mathbf{v}_\infty\right\|}$$ because $\mathbf{r}_3$ is a unit vector.

Z vanishing point tells us about the surface normal of the ground plane

Single Vanishing Point



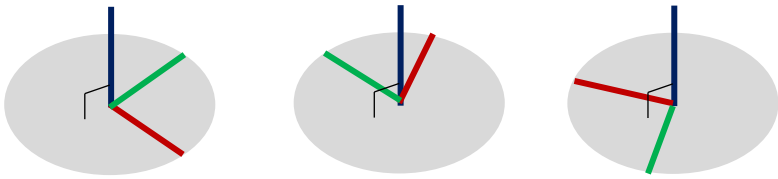
$$\lambda \mathbf{v}_\infty = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\lambda \mathbf{v}_\infty = \mathbf{K}\mathbf{r}_3$$

$$\longrightarrow \mathbf{r}_3 = \frac{\mathbf{K}^{-1}\mathbf{v}_\infty}{\left\|\mathbf{K}^{-1}\mathbf{v}_\infty\right\|}$$ because $\mathbf{r}_3$ is a unit vector.

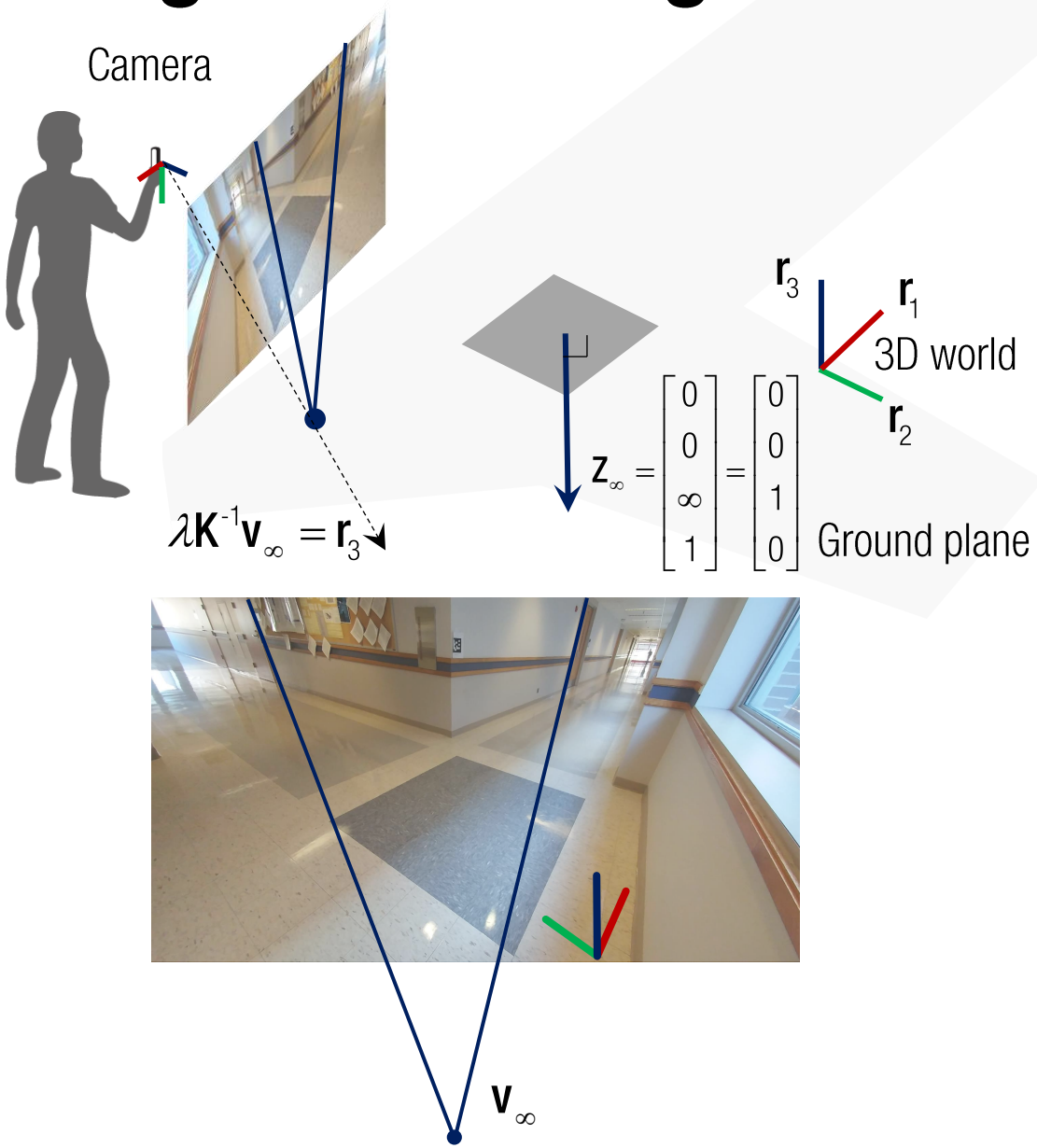
Z vanishing point tells us about the surface normal of the ground plane

Rotation ambiguity

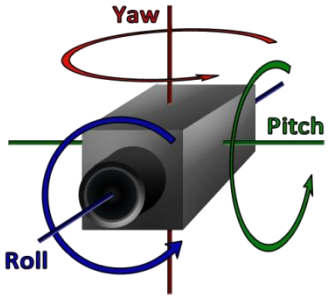


Ground plane

Single Vanishing Point

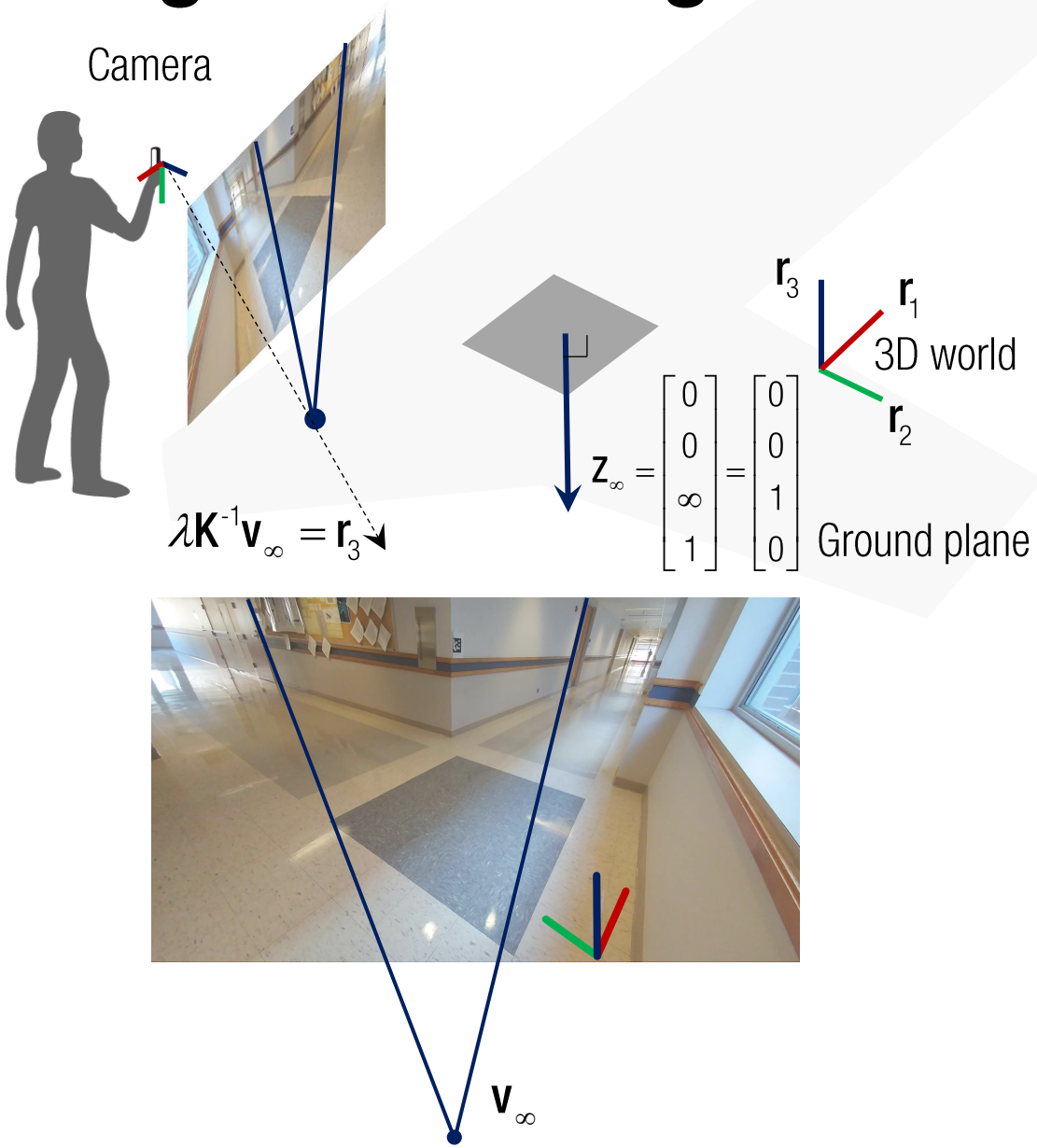


$$r_3 = \frac{K^{-1}v_\infty}{\|K^{-1}v_\infty\|}$$

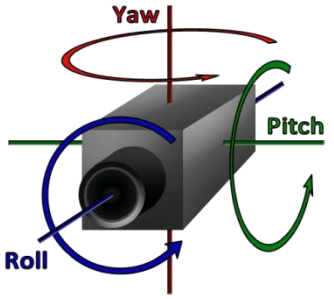


Roll and pitch angle can be computed by the ground plane.

Single Vanishing Point



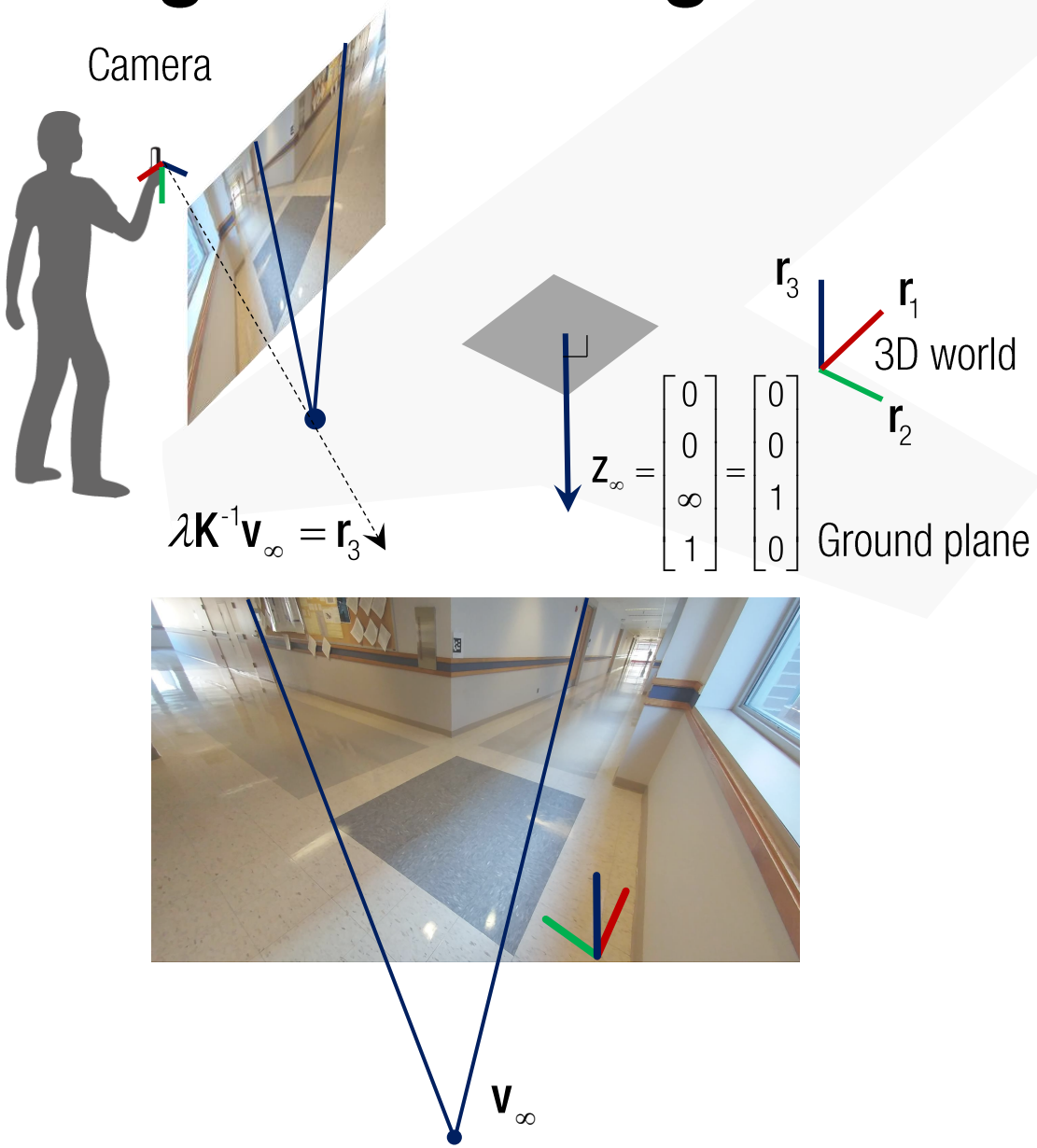
$$r_3 = \frac{K^{-1}v_\infty}{\|K^{-1}v_\infty\|}$$



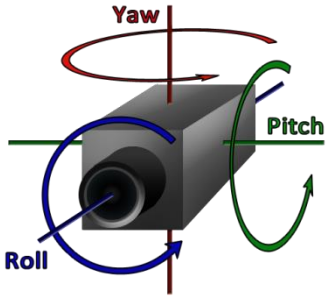
Roll and pitch angle can be computed by the ground plane.

Yaw: rotation about z axis: $R_{yaw} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Single Vanishing Point



$$r_3 = \frac{K^{-1}v_\infty}{\|K^{-1}v_\infty\|}$$

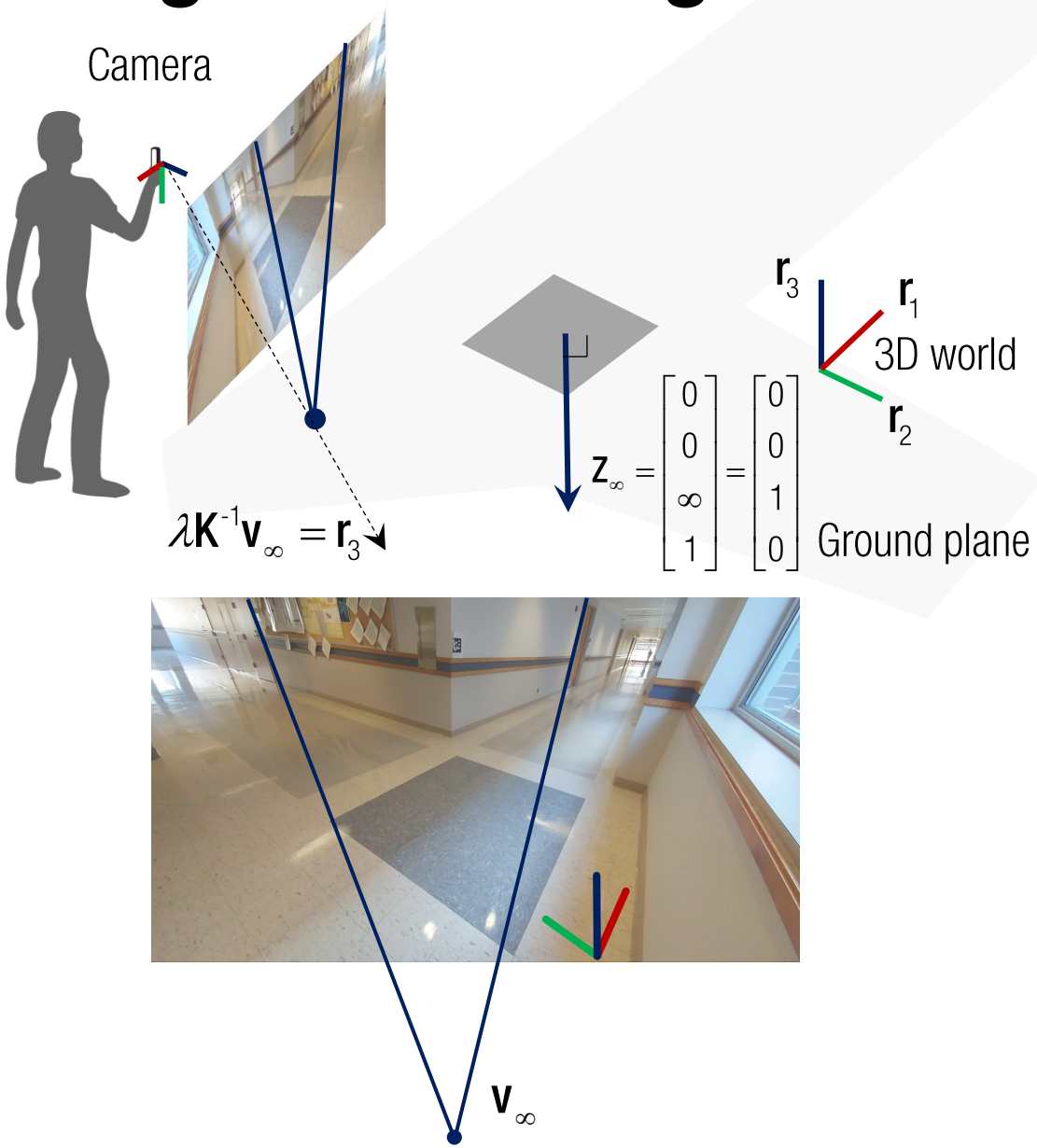


Roll and pitch angle can be computed by the ground plane.

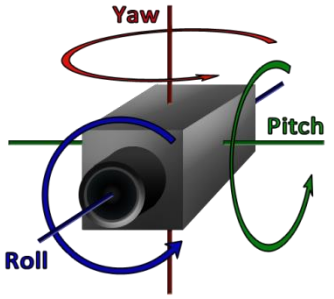
Yaw: rotation about z axis: $R_{yaw} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Pitch: rotation about y axis: $R_{pitch} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$

Single Vanishing Point



$$r_3 = \frac{K^{-1}v_\infty}{\|K^{-1}v_\infty\|}$$



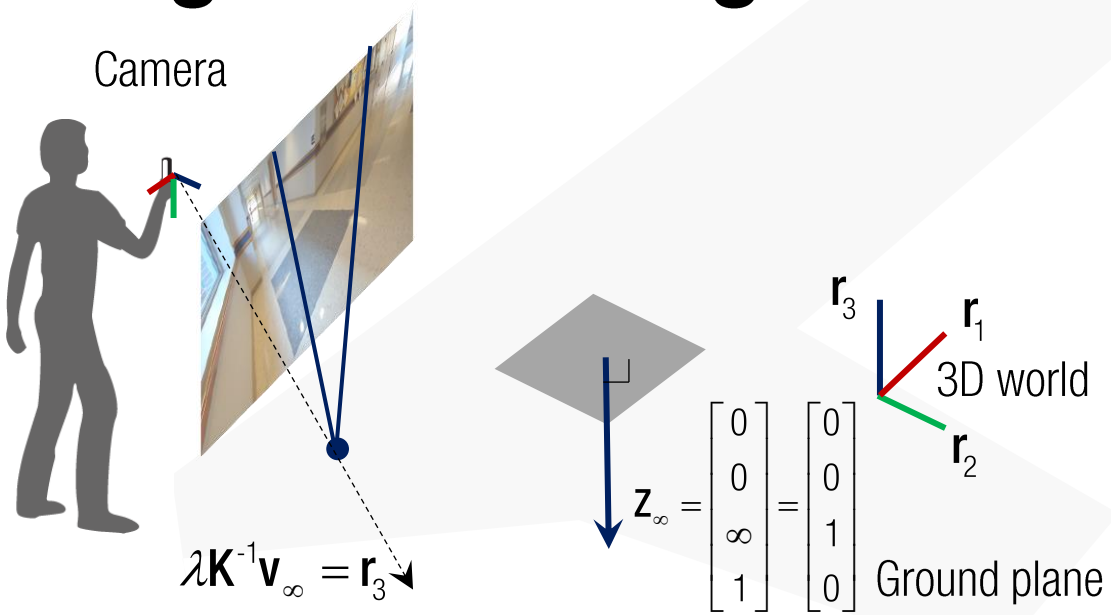
Roll and pitch angle can be computed by the ground plane.

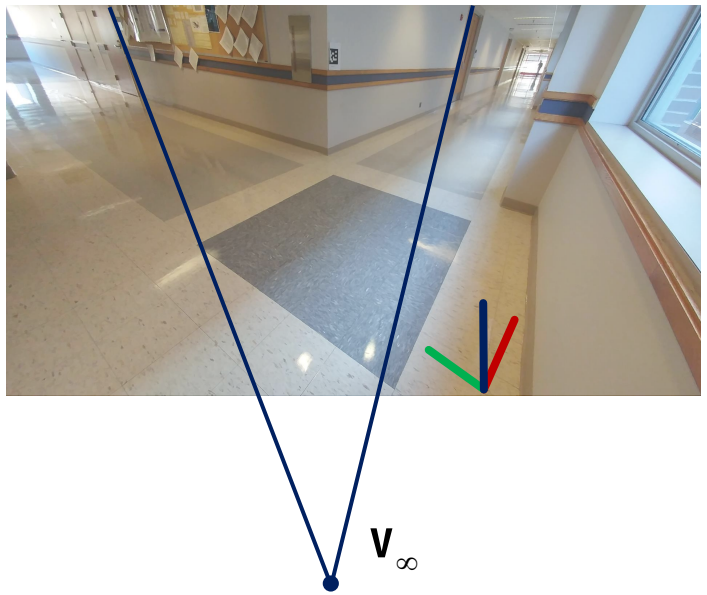
Yaw: rotation about z axis: $R_{yaw} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Pitch: rotation about y axis: $R_{pitch} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$

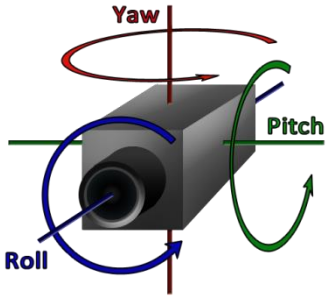
Roll: rotation about x axis: $R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$

Single Vanishing Point





$$r_3 = \frac{K^{-1}v_\infty}{\|K^{-1}v_\infty\|}$$



Roll and pitch angle can be computed by the ground plane.

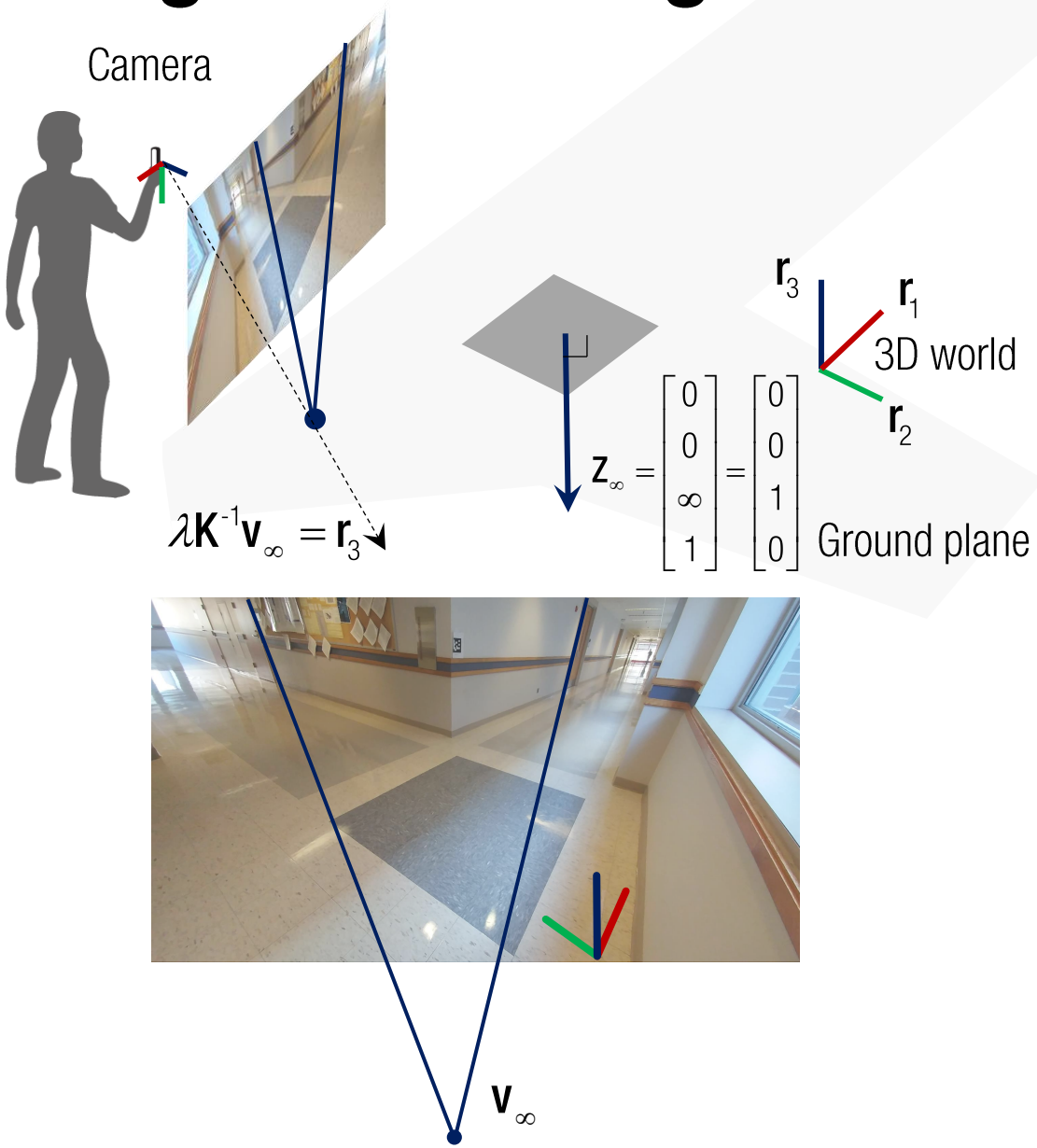
Yaw: rotation about z axis: $R_{yaw} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Pitch: rotation about y axis: $R_{pitch} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$

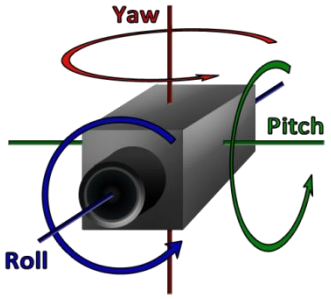
Roll: rotation about x axis: $R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$

$$\longrightarrow \quad [r_1 \quad r_2 \quad r_3] = (R_{yaw}R_{pitch}R_{roll})^T = \begin{bmatrix} \bullet & \bullet & \\ \bullet & \bullet & ? \\ \bullet & \bullet & \end{bmatrix}$$

Single Vanishing Point



$$r_3 = \frac{K^{-1}v_\infty}{\|K^{-1}v_\infty\|}$$



Roll and pitch angle can be computed by the ground plane.

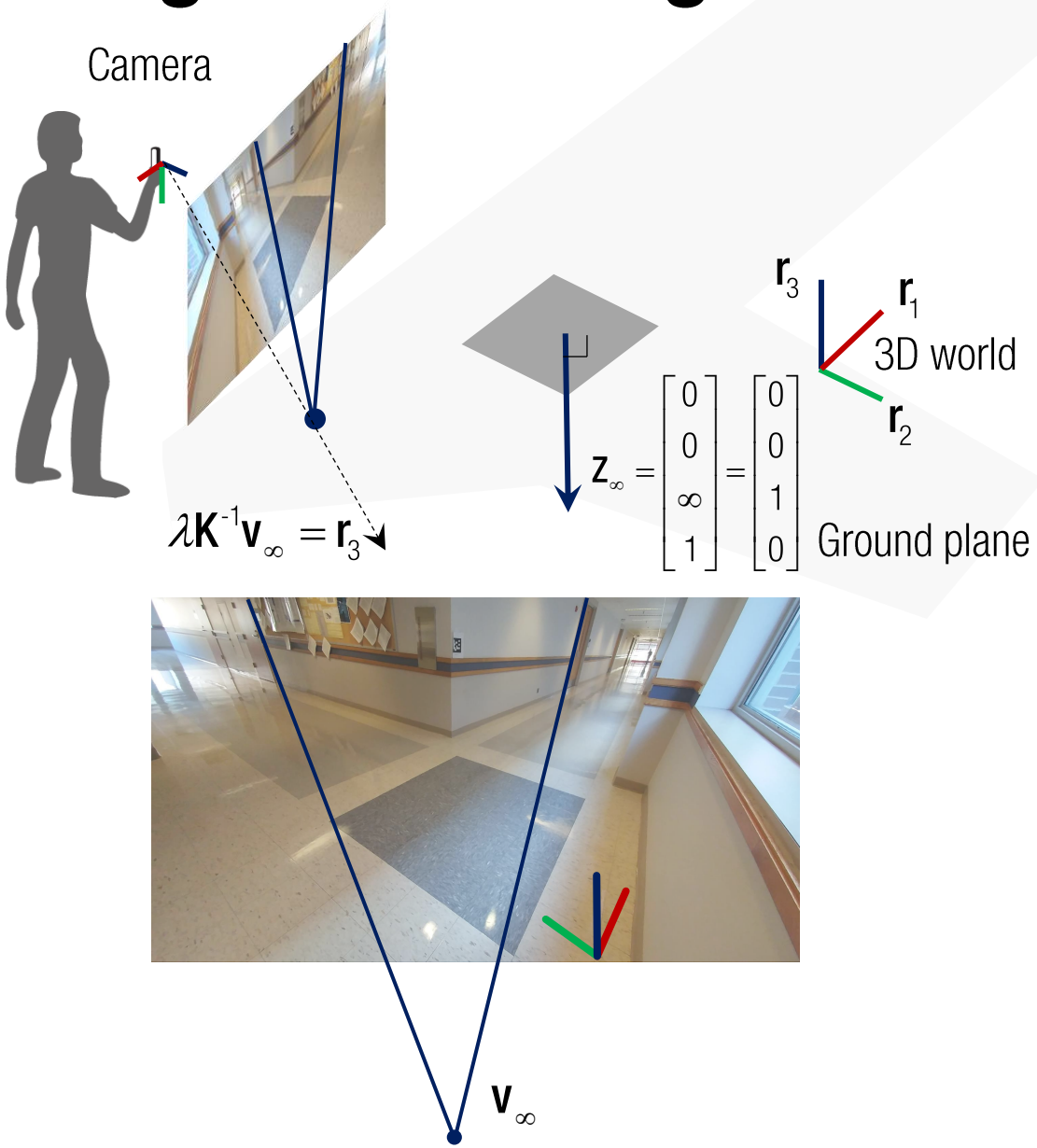
Yaw: rotation about z axis: $R_{yaw} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Pitch: rotation about y axis: $R_{pitch} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$

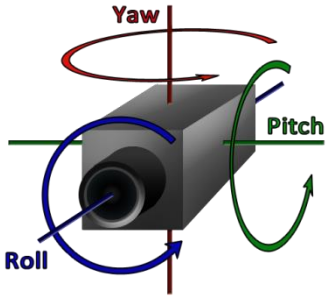
Roll: rotation about x axis: $R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$

$$\rightarrow \quad [r_1 \quad r_2 \quad r_3] = (R_{yaw}R_{pitch}R_{roll})^T = \begin{bmatrix} \bullet & \bullet & -\sin\beta \\ \bullet & \bullet & \cos\beta\sin\gamma \\ \bullet & \bullet & \cos\beta\cos\gamma \end{bmatrix}$$

Single Vanishing Point



$$r_3 = \frac{K^{-1}v_\infty}{\left\|K^{-1}v_\infty\right\|}$$



Roll and pitch angle can be computed by the ground plane.

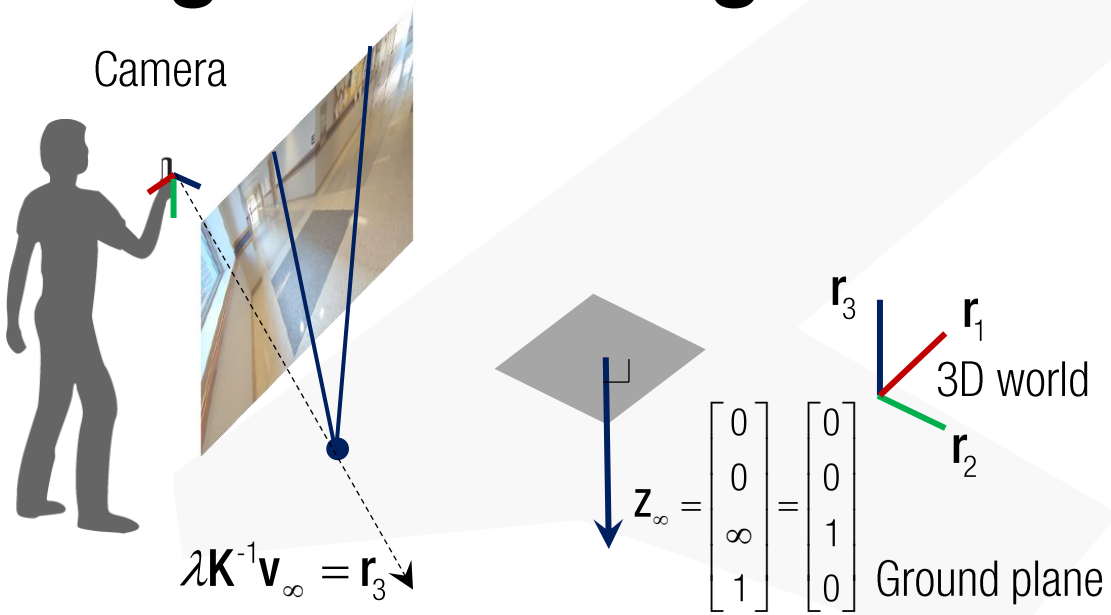
$$\begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} = \left(R_{yaw}R_{pitch}R_{roll}\right)^T = \begin{bmatrix} \bullet & \bullet & -\sin\beta \\ \bullet & \bullet & \cos\beta\sin\gamma \\ \bullet & \bullet & \cos\beta\cos\gamma \end{bmatrix}$$

Pitch: $\beta =$

→

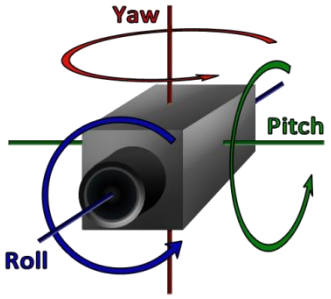
Roll: $\gamma =$

Single Vanishing Point



v_∞

$$r_3 = \frac{K^{-1}v_\infty}{\left\|K^{-1}v_\infty\right\|}$$



Roll and pitch angle can be computed by the ground plane.

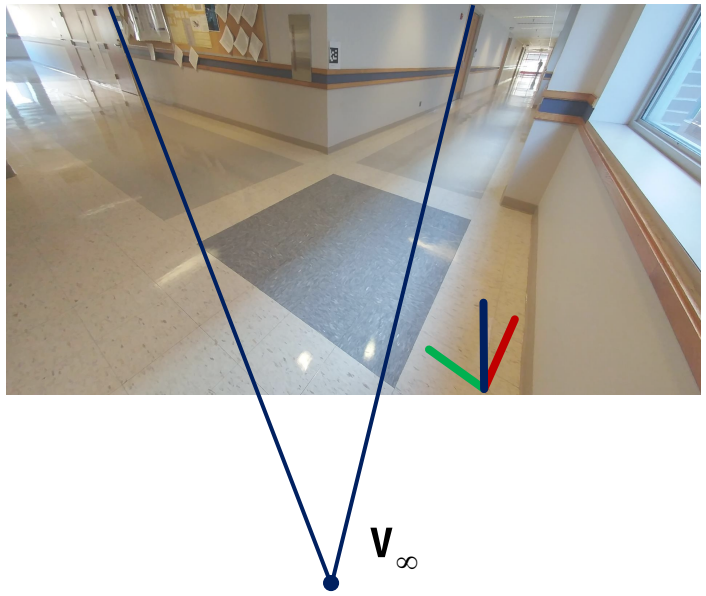
$$\begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} = \left(R_{yaw}R_{pitch}R_{roll}\right)^T = \begin{bmatrix} \bullet & \bullet & -\sin\beta \\ \bullet & \bullet & \cos\beta\sin\gamma \\ \bullet & \bullet & \cos\beta\cos\gamma \end{bmatrix}$$

$\longrightarrow$

Pitch: $\beta = \tan^{-1}\left(-\frac{r_{31}}{\sqrt{r_{32}^2 + r_{33}^2}}\right)$

Roll: $\gamma = \tan^{-1}\frac{r_{32}}{r_{33}}$

Single Vanishing Point (Exercise)

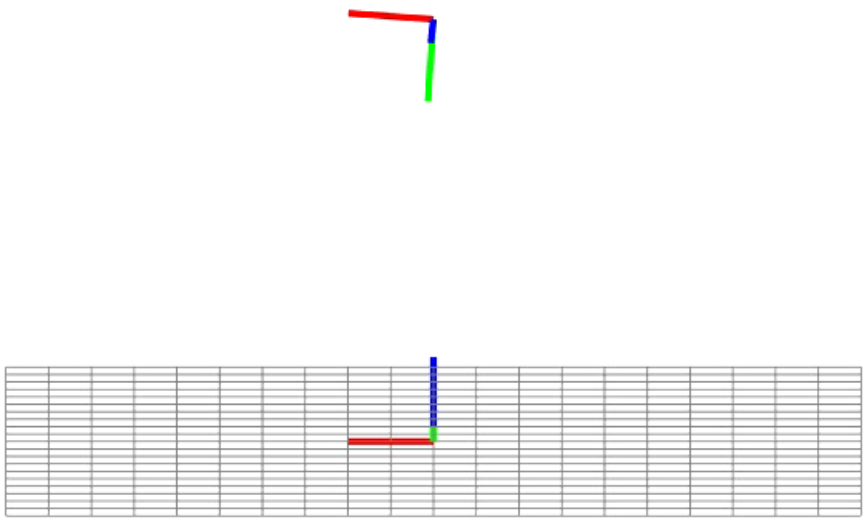


ComputeCameraUsingVanishingPoint.m

```
f = 1224;
K = [f 0 size(im,2)/2;
     0 f size(im,1)/2;
     0 0 1];

m1 = [2563;25;1];
m2 = [2439;545;1];
m3 = [571;25;1];
m4 = [723;498;1];

l1 = GetLineFromTwoPoints(m1,m2);
l2 = GetLineFromTwoPoints(m3,m4);

v1 = GetPointFromTwoLines(l1,l2);
v1 = v1/v1(3);

r3 = inv(K)*v1/norm(inv(K)*v1)
pitch = atan(-r3(1)/norm(r3(2:3)))
roll = atan(r3(2)/r3(3))
```

```
r3 =

  -0.0736
   0.8959
   0.4381


pitch =

   0.0736


roll =

   1.1160
```

Single Vanishing Point (Exercise)

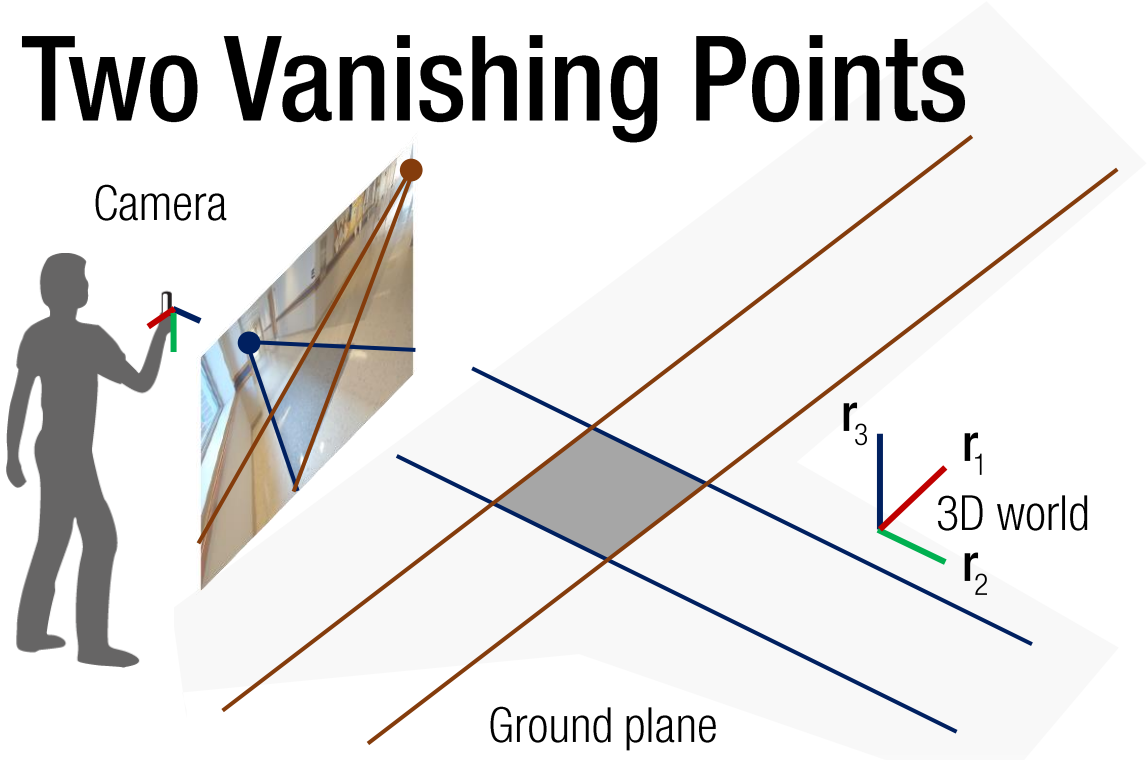


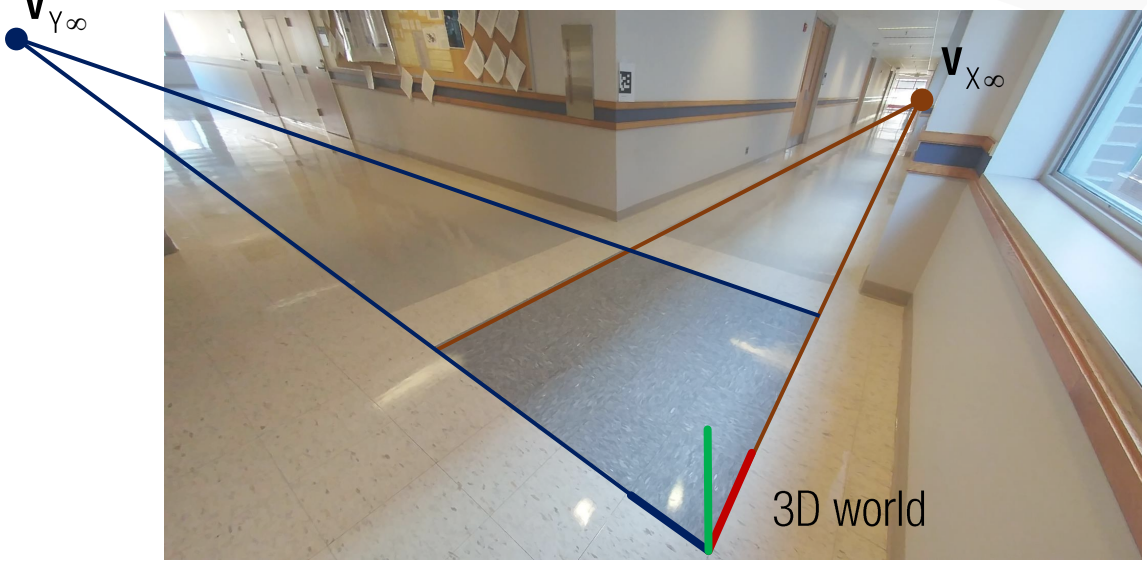
ComputeCameraUsingVanishingPoint.m

```
f = 1224;
K = [f 0 size(im,2)/2;
     0 f size(im,1)/2;
     0 0 1];

m1 = [2563;25;1];
m2 = [2439;545;1];
m3 = [571;25;1];
m4 = [723;498;1];

l1 = GetLineFromTwoPoints(m1,m2);
l2 = GetLineFromTwoPoints(m3,m4);

v1 = GetPointFromTwoLines(l1,l2);
v1 = v1/v1(3);

r3 = inv(K)*v1/norm(inv(K)*v1)
pitch = atan(-r3(1)/norm(r3(2:3)))
roll = atan(r3(2)/r3(3))
```

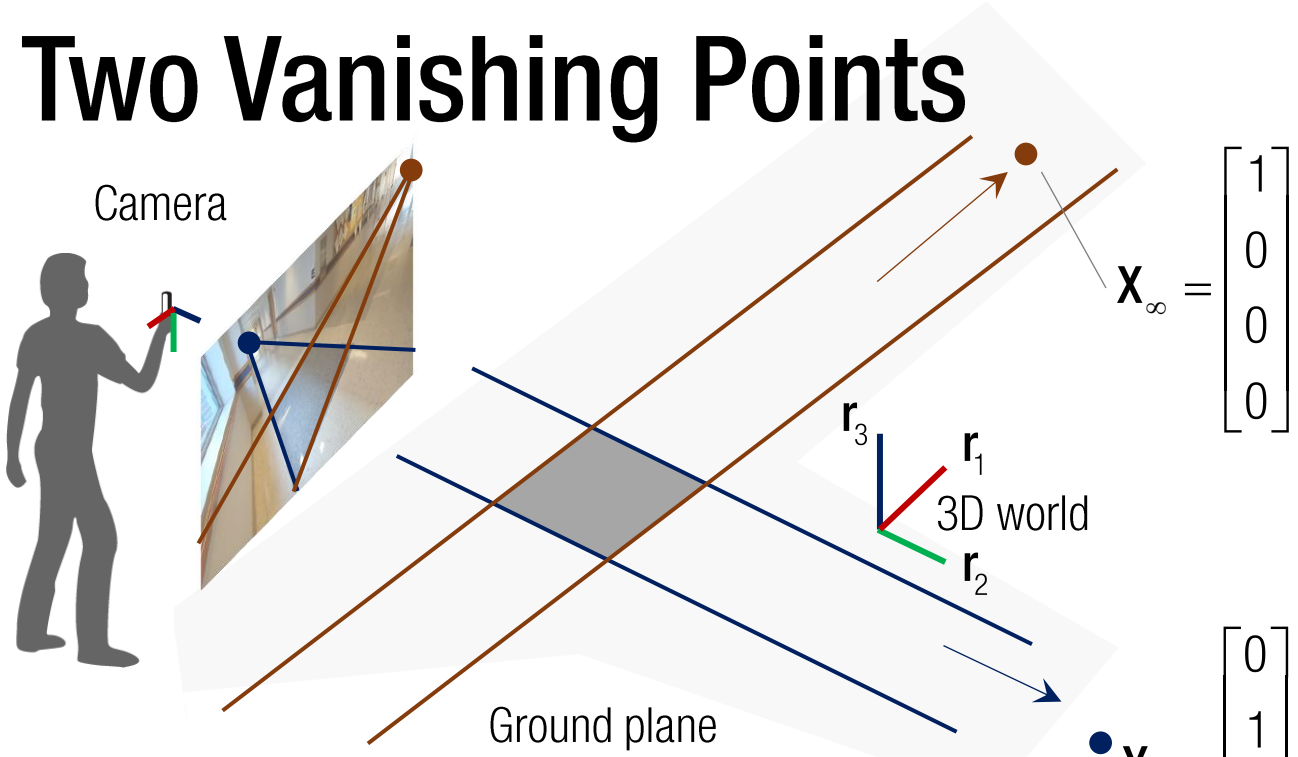
```
r3 =

  -0.0736
   0.8959
   0.4381


pitch =

   0.0736


roll =

   1.1160
```

Two Vanishing Points



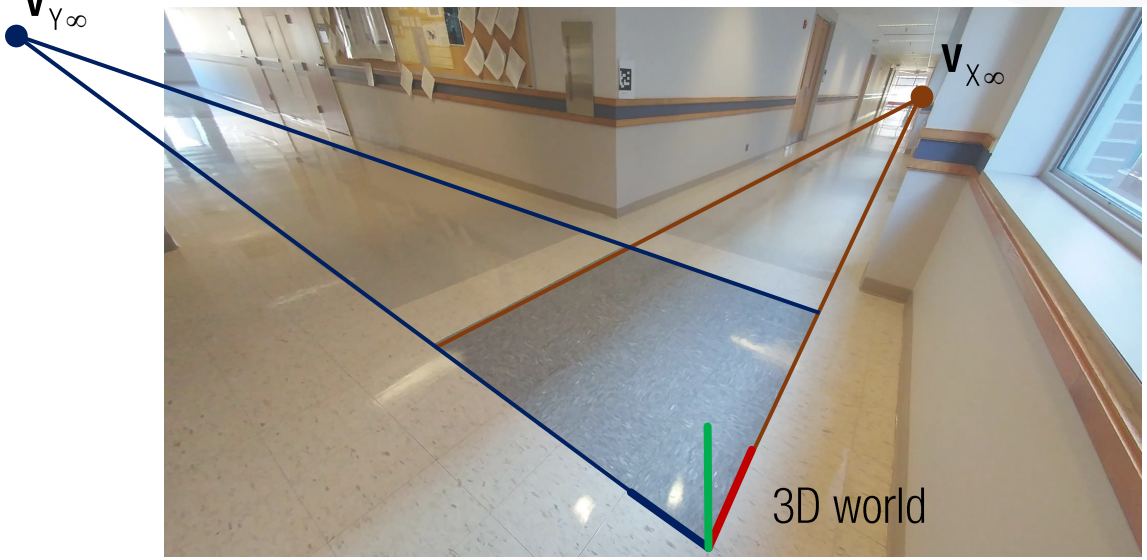


Two Vanishing Points

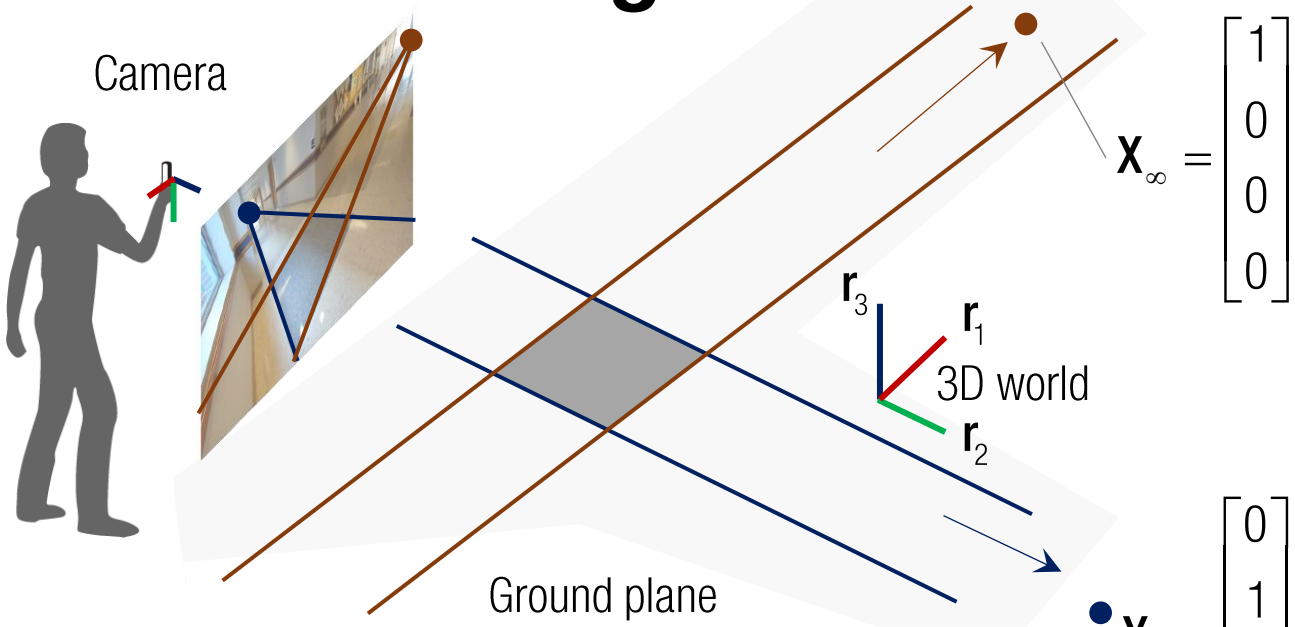


$$\mathbf{X}_\infty = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{Y}_\infty = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$



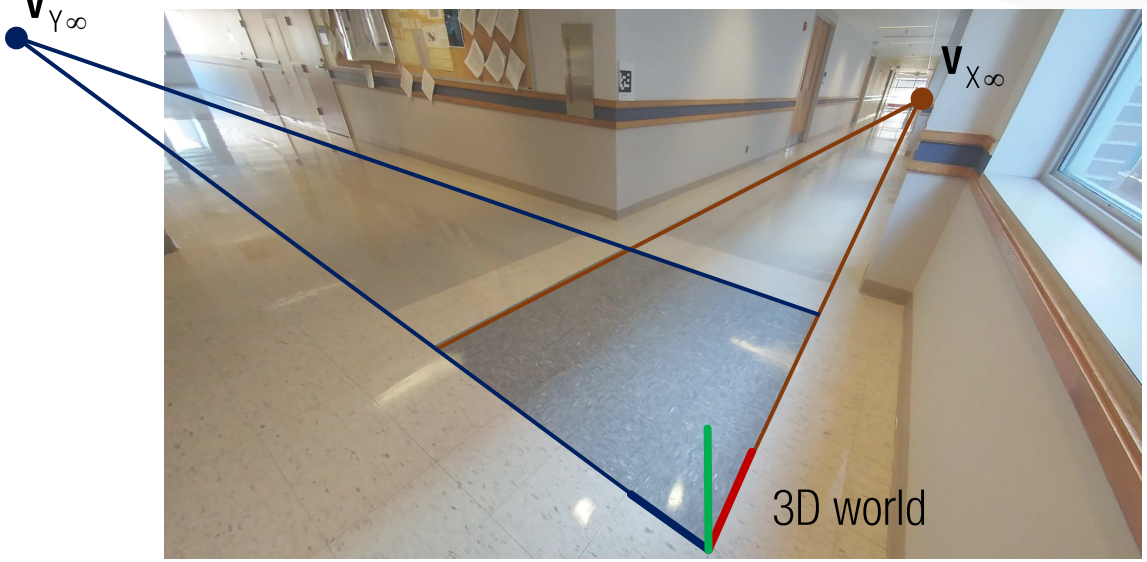
Two Vanishing Points



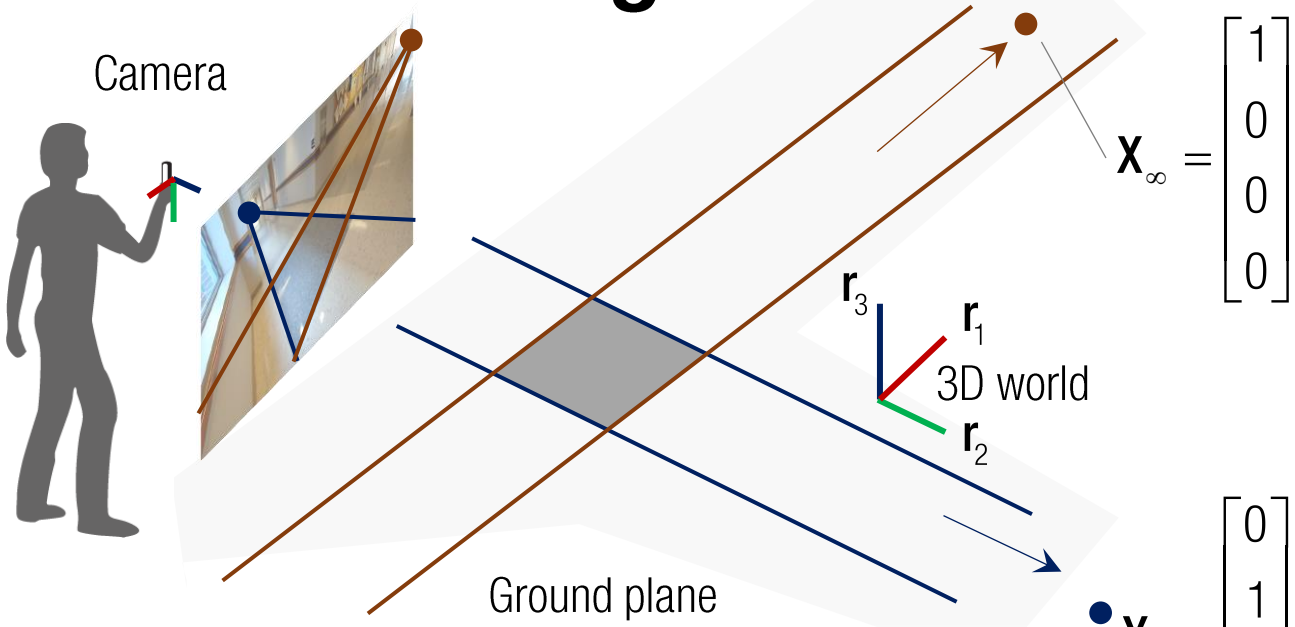
$$\mathbf{X}_\infty = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{Y}_\infty = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\lambda \mathbf{v}_{X\infty} = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C \end{bmatrix}\mathbf{X}_\infty$$
$$= \mathbf{K}\mathbf{r}_1$$

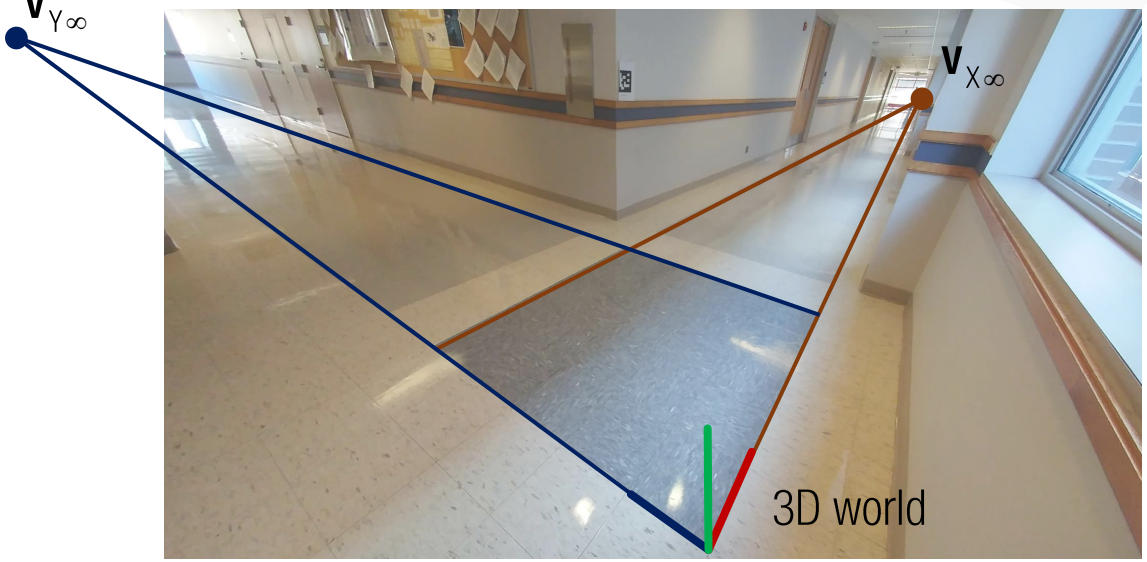


Two Vanishing Points



$$\mathbf{X}_\infty = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

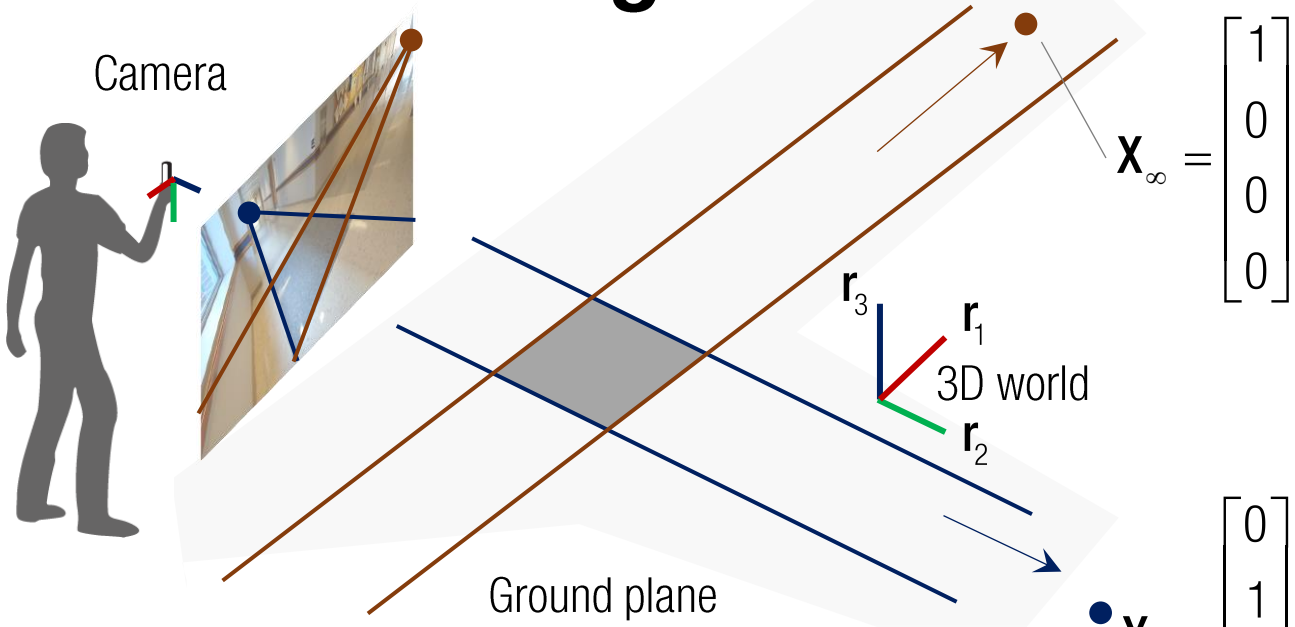
$$\mathbf{Y}_\infty = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$



$$\lambda \mathbf{v}_{X\infty} = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C \end{bmatrix}\mathbf{X}_\infty$$
$$= \mathbf{K}\mathbf{r}_1$$

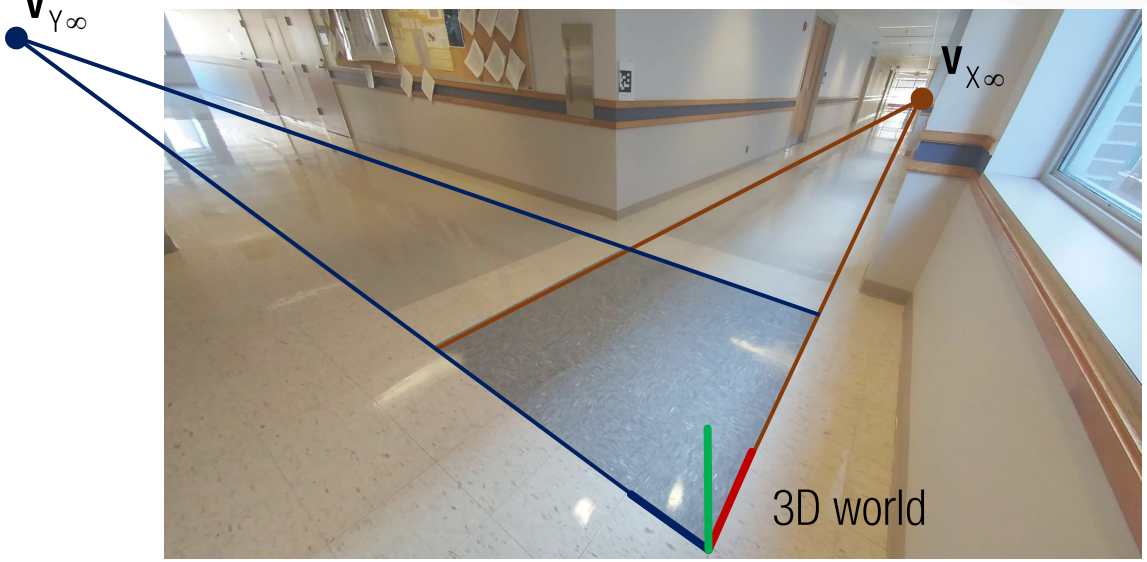
$$\lambda \mathbf{v}_{Y\infty} = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C \end{bmatrix}\mathbf{Y}_\infty$$
$$= \mathbf{K}\mathbf{r}_2$$

Two Vanishing Points



$$\mathbf{X}_\infty = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{Y}_\infty = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$



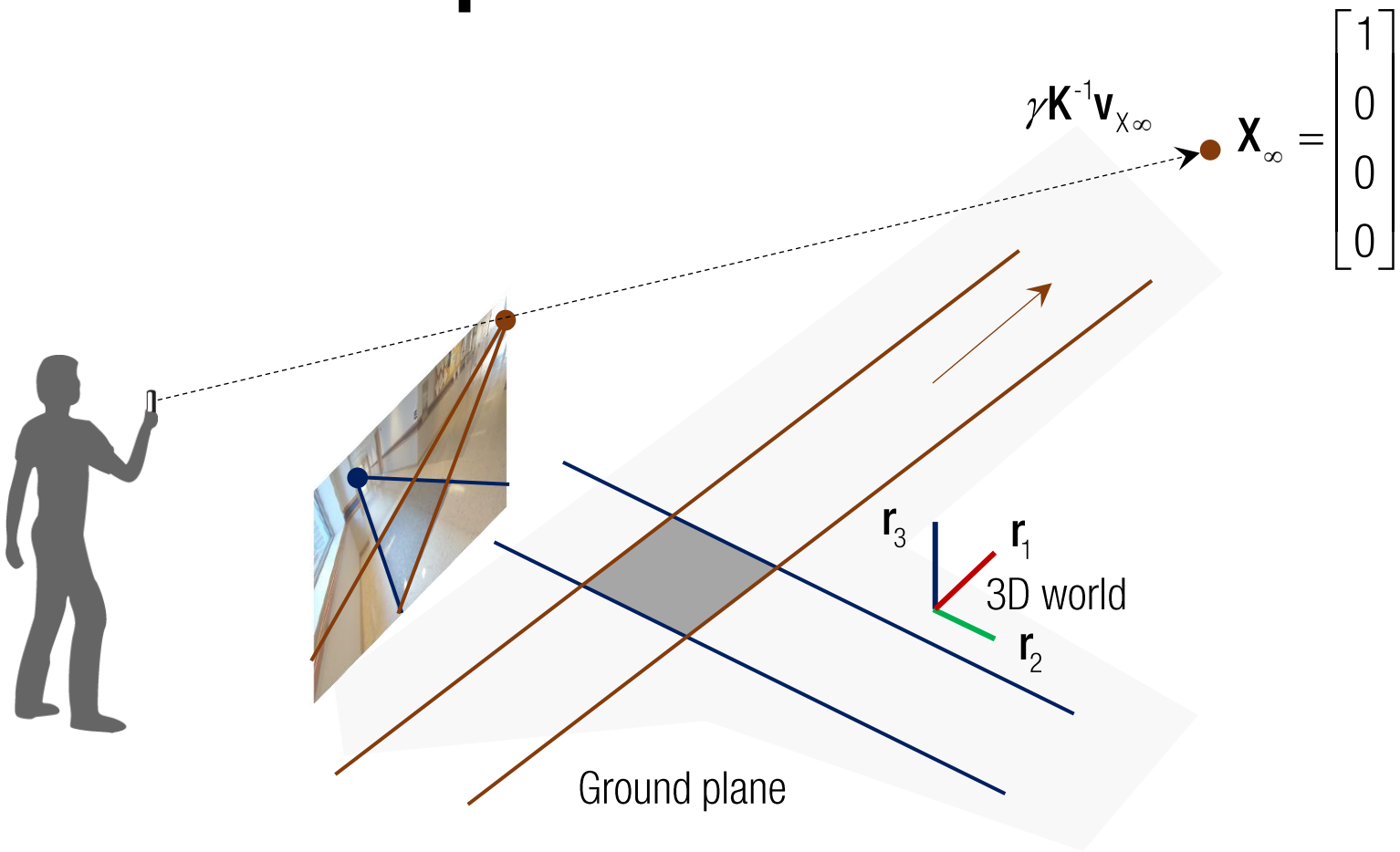
$$\lambda \mathbf{v}_{X\infty} = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C \end{bmatrix}\mathbf{X}_\infty = \mathbf{K}\mathbf{r}_1$$

$$\lambda \mathbf{v}_{Y\infty} = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}_W^C \end{bmatrix}\mathbf{Y}_\infty = \mathbf{K}\mathbf{r}_2$$

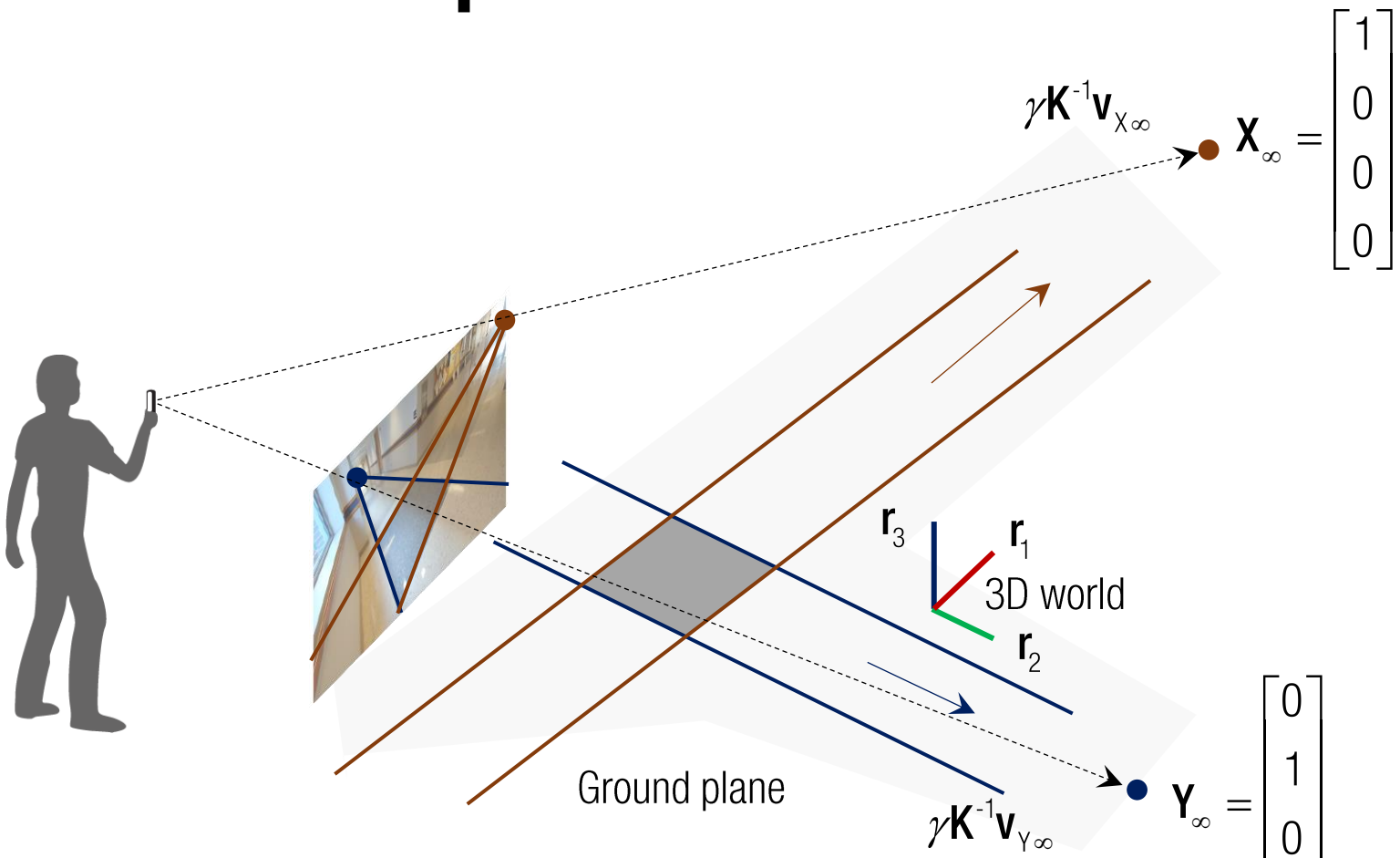
$$\rightarrow \quad \mathbf{r}_1 = \frac{\mathbf{K}^{-1}\mathbf{v}_{X\infty}}{\left\|\mathbf{K}^{-1}\mathbf{v}_{X\infty}\right\|}, \quad \mathbf{r}_2 = \frac{\mathbf{K}^{-1}\mathbf{v}_{Y\infty}}{\left\|\mathbf{K}^{-1}\mathbf{v}_{Y\infty}\right\|}$$

$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$: Orthogonality constraint

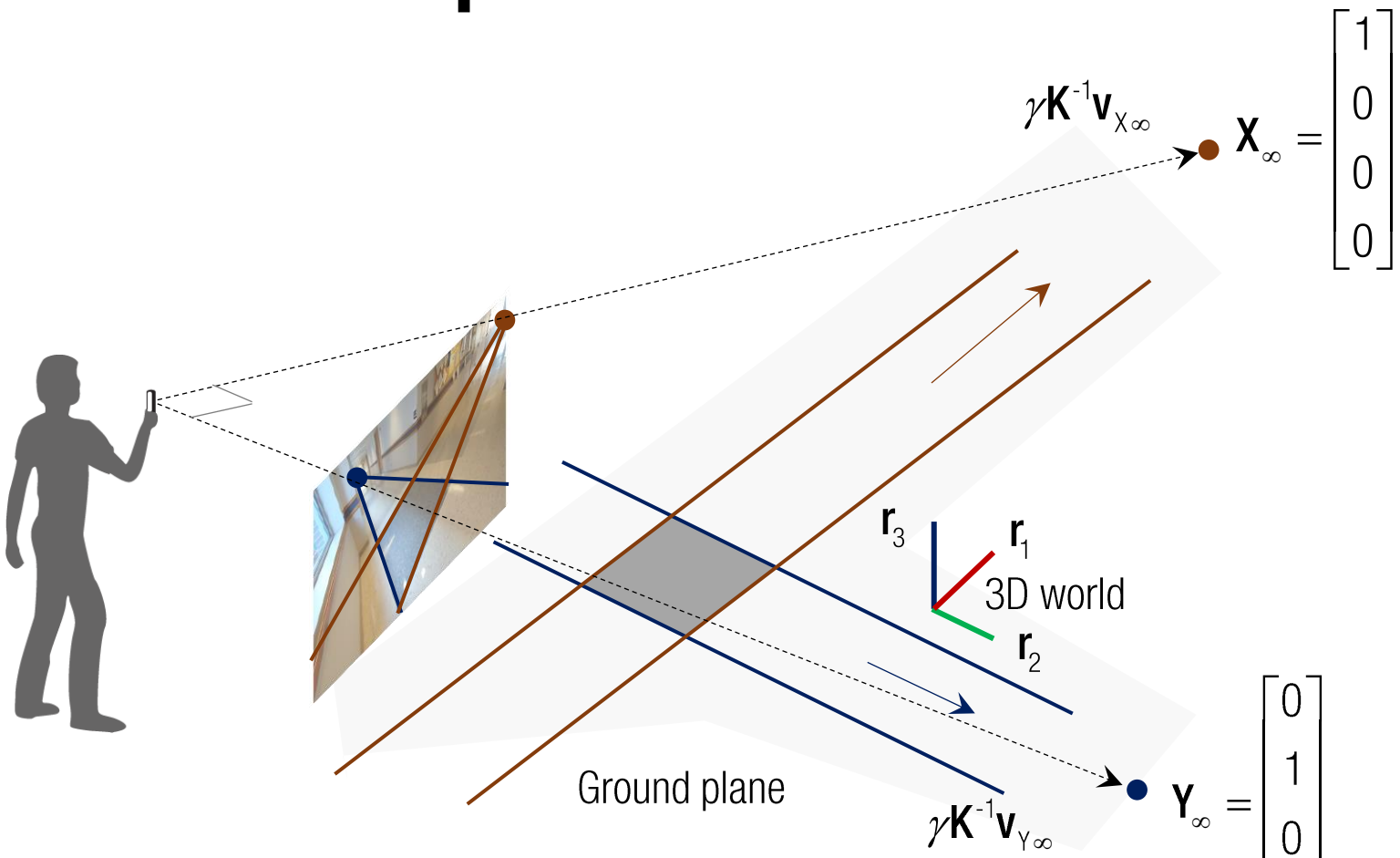
Geometric Interpretation



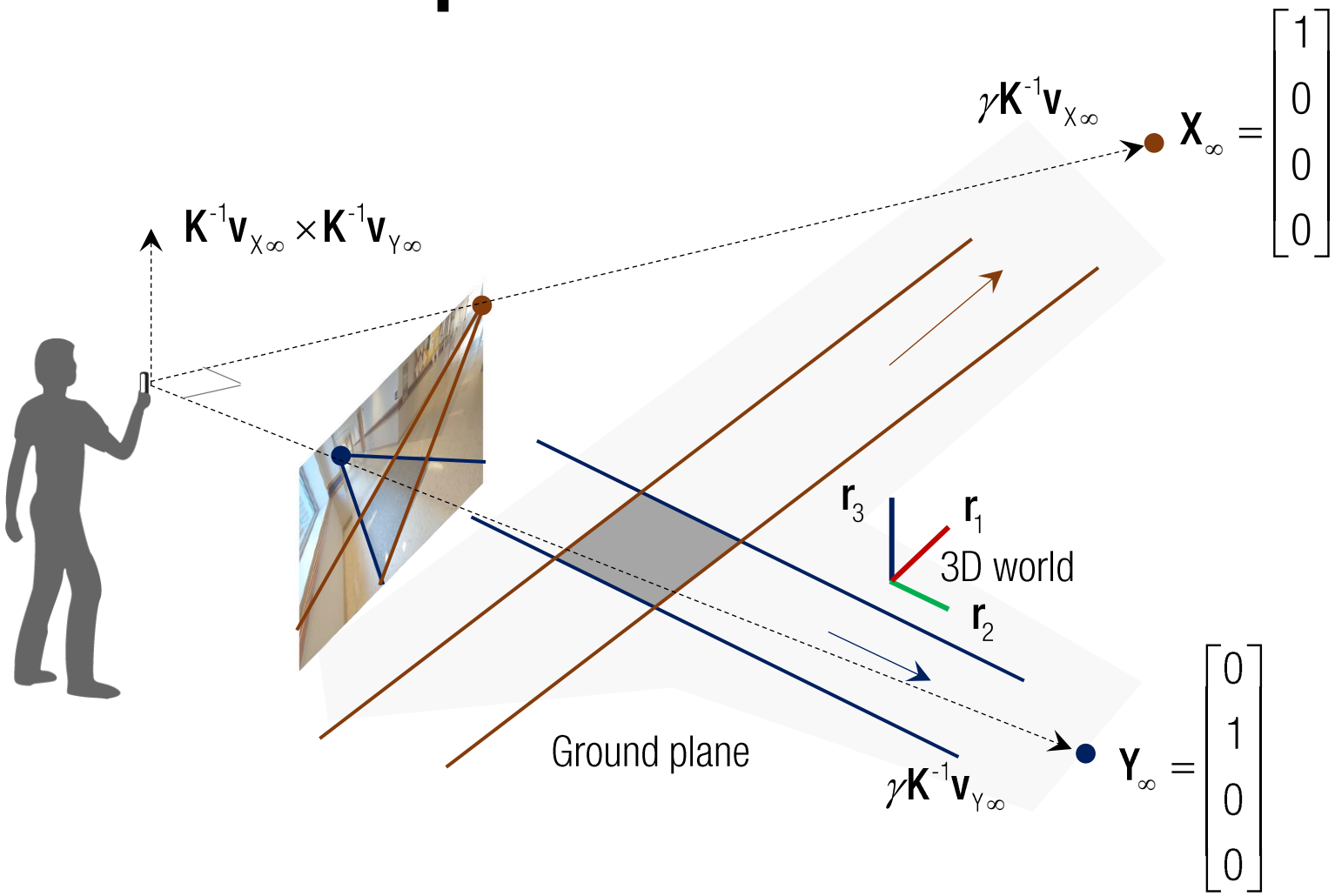
Geometric Interpretation



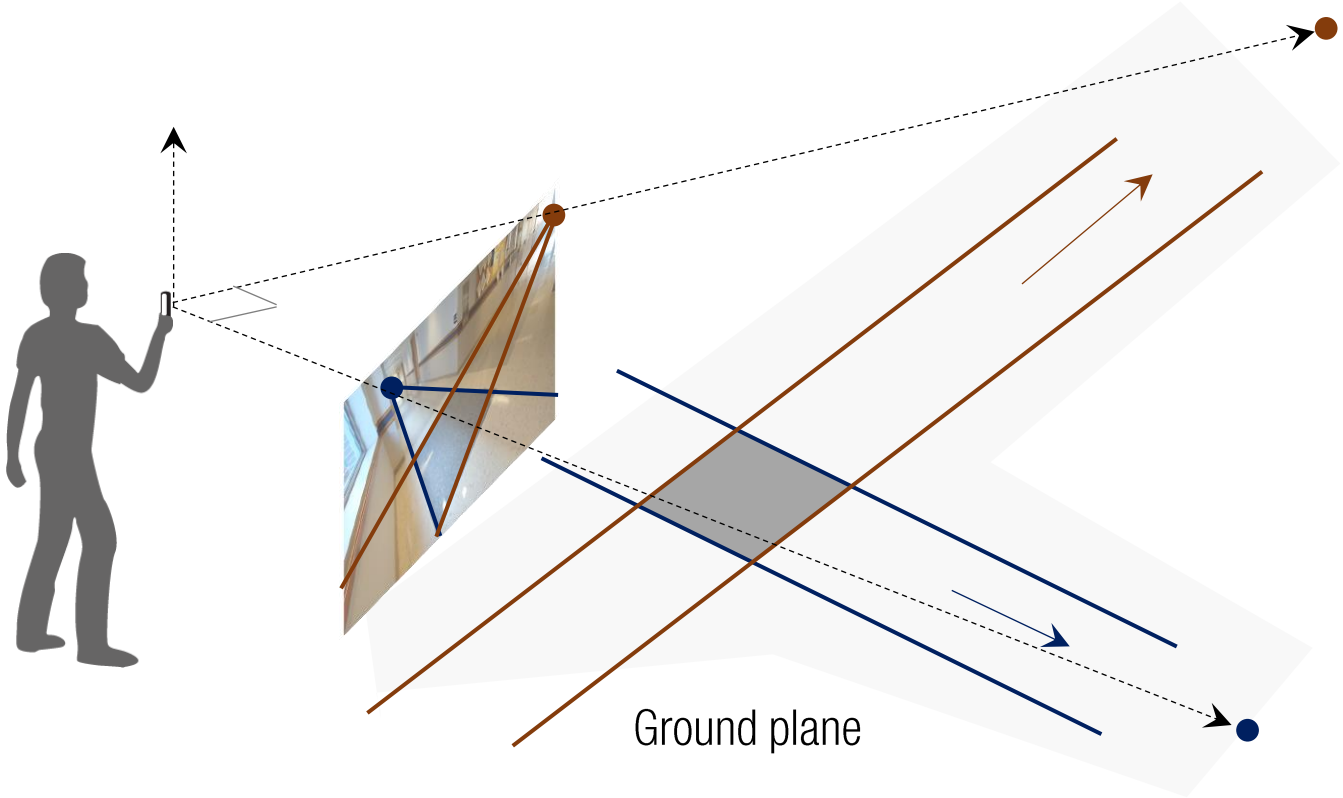
Geometric Interpretation



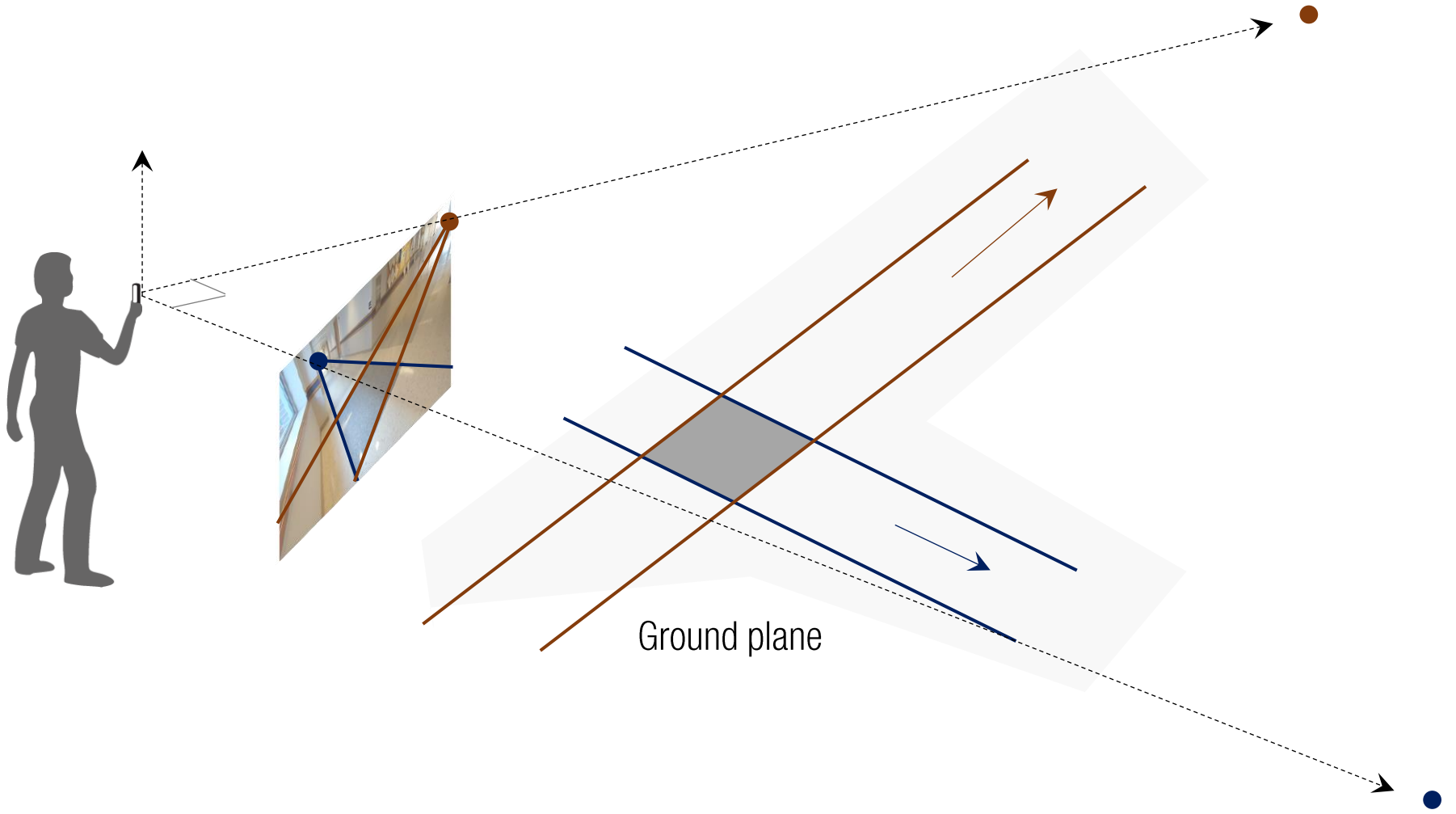
Geometric Interpretation



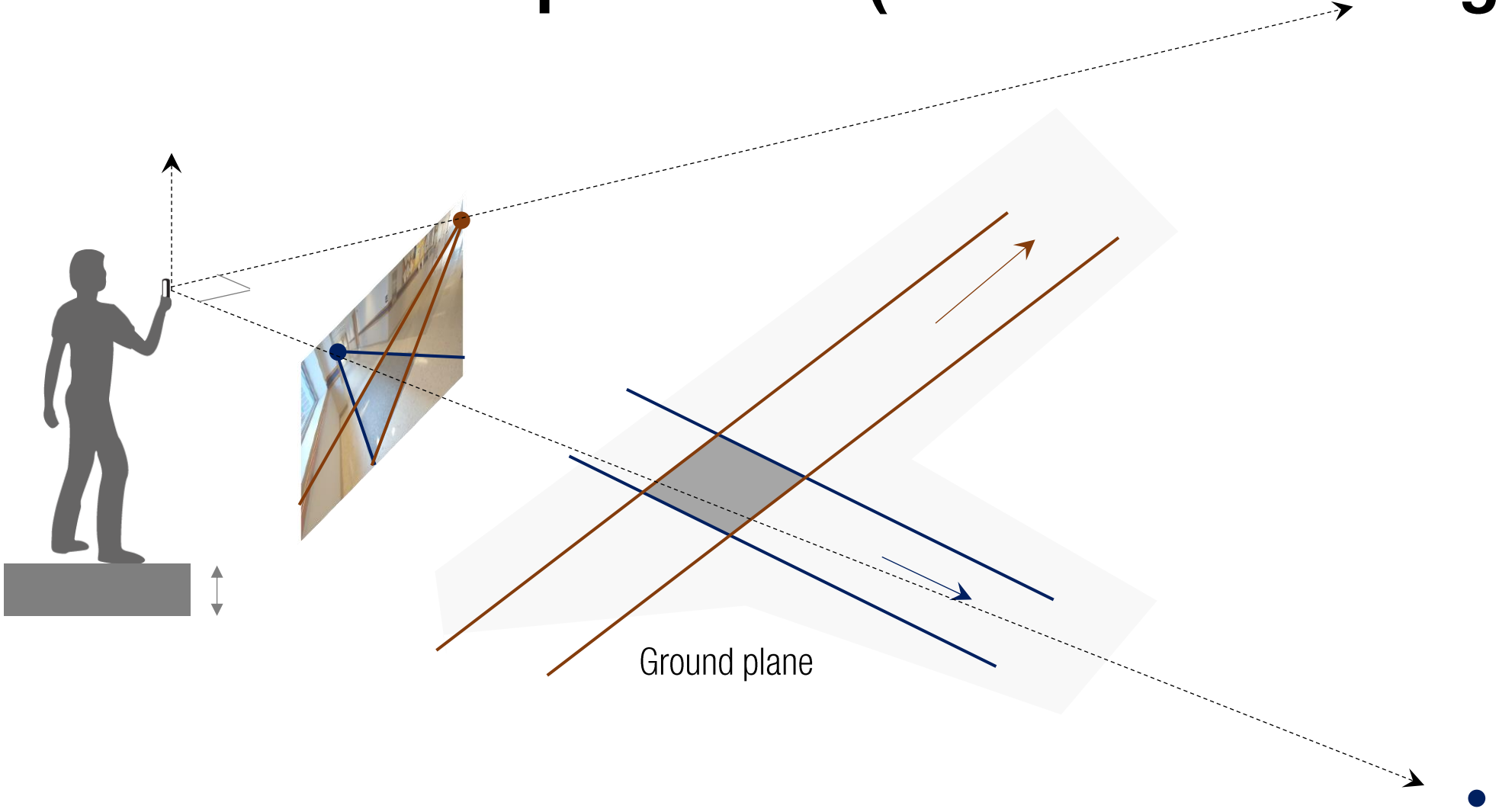
Geometric Interpretation (Translation Ambiguity)



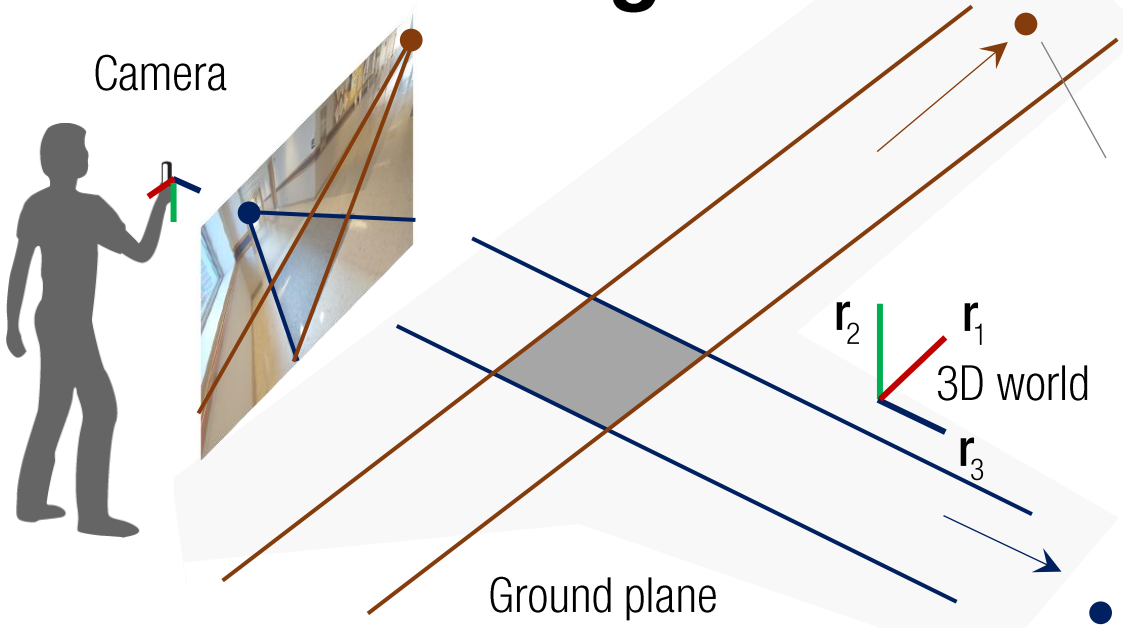
Geometric Interpretation (Translation Ambiguity)

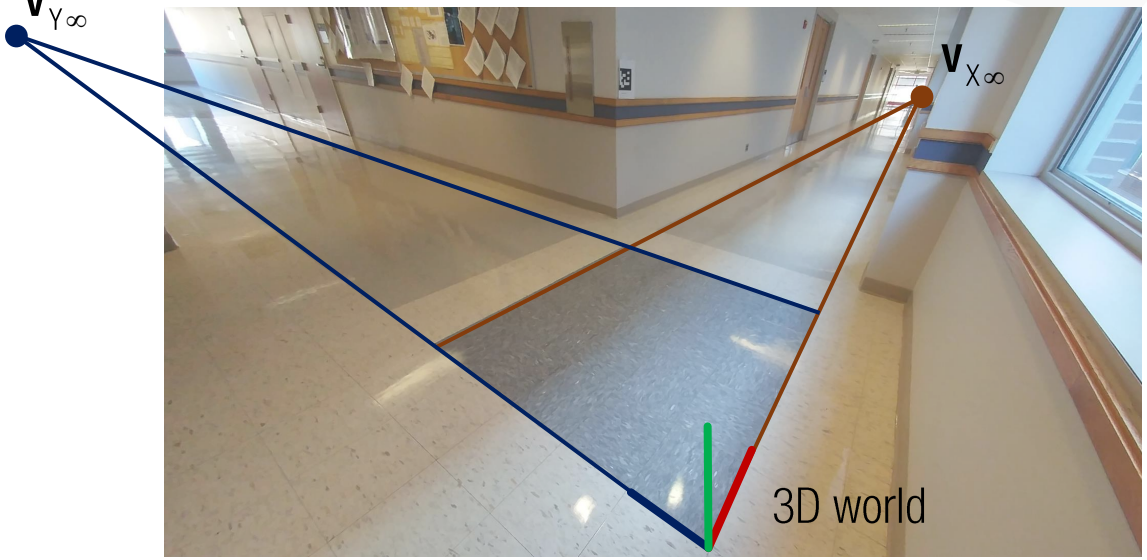


Geometric Interpretation (Translation Ambiguity)



Two Vanishing Points





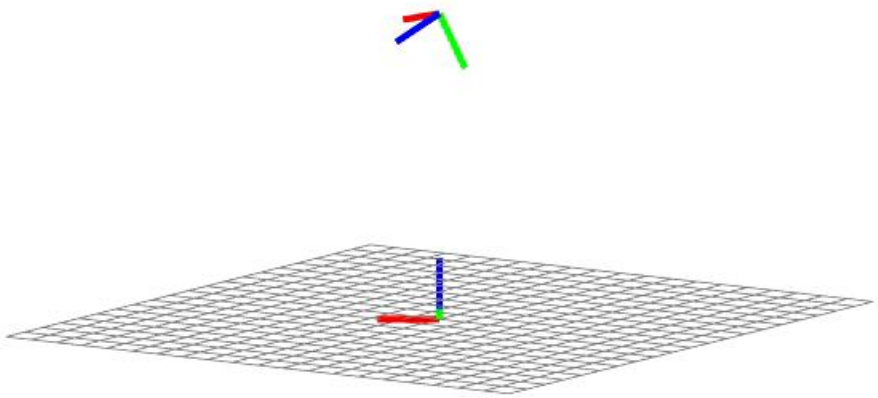
ComputeCameraUsingTwoVanishingPoints.m

```
f = 4000;
K = [f 0 size(im,2)/2;
     0 f size(im,1)/2;
     0 0 1];

l11 = GetLineFromTwoPoints(m11,m12);
l12 = GetLineFromTwoPoints(m13,m14);

l21 = GetLineFromTwoPoints(m21,m22);
l22 = GetLineFromTwoPoints(m23,m24);

v1 = GetPointFromTwoLines(l11,l12);
v2 = GetPointFromTwoLines(l21,l22);

r1 = inv(K)*v1/norm(inv(K)*v1);
r2 = inv(K)*v2/norm(inv(K)*v2);

r3 = Vec2Skew(r1)*r2;
```

Not orthogonal matrix!

```
R =
    0.2448   -0.5178    0.0424
   -0.1737   -0.1960   -0.6978
    0.9539    0.8327   -0.1379

det(R) =
    0.5077

R'*R =
    0.3299    0.0294   -0.2036
    0.0294    0.5555   -0.2327
   -0.2036   -0.2327    1.6224
```

Two Vanishing Points



ComputeCameraUsingTwoVanishingPoints.m

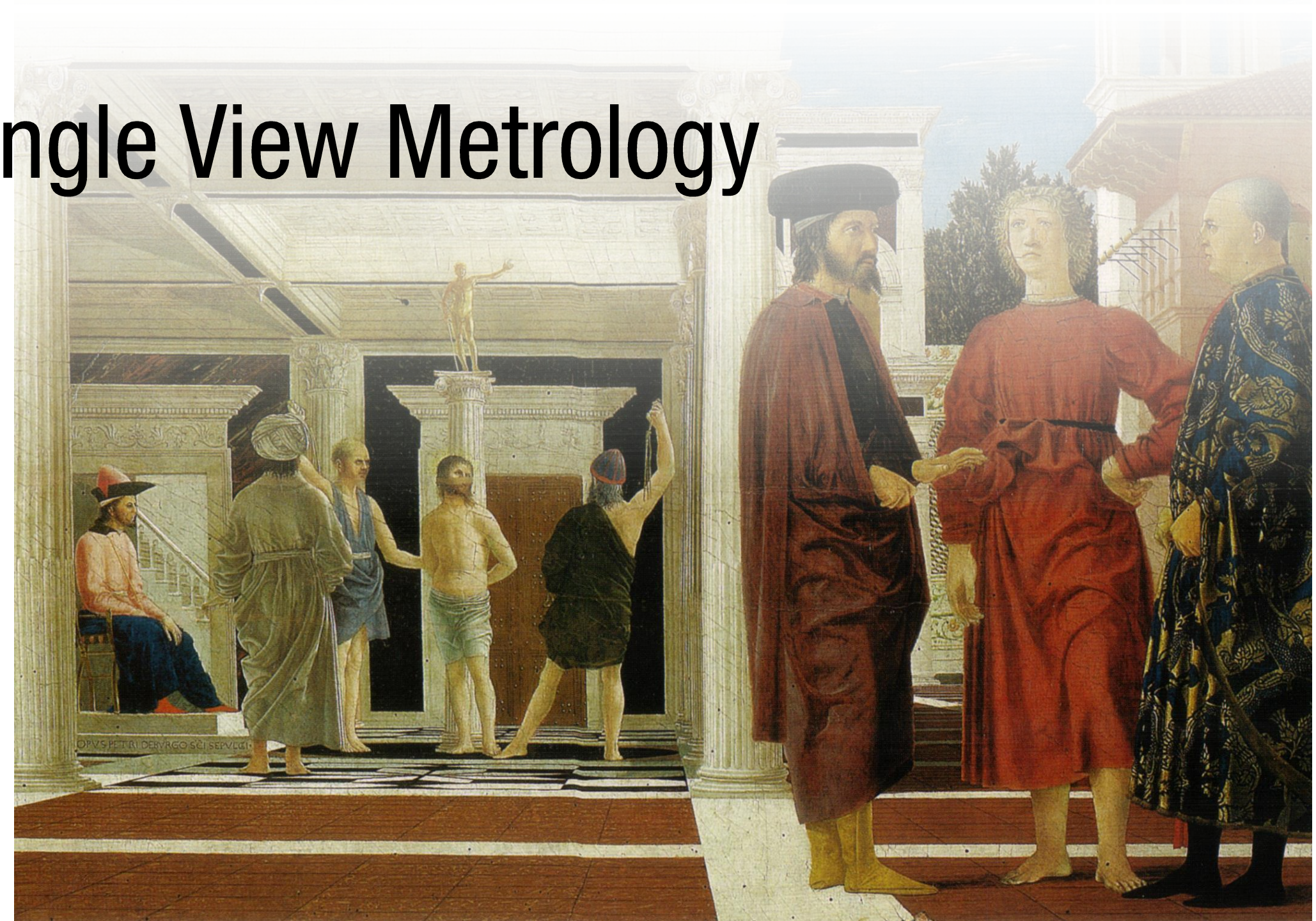
```
f = 1224;
K = [f 0 size(im,2)/2;
    0 f size(im,1)/2;
    0 0 1];

l11 = GetLineFromTwoPoints(m11,m12);
l12 = GetLineFromTwoPoints(m13,m14);

l21 = GetLineFromTwoPoints(m21,m22);
l22 = GetLineFromTwoPoints(m23,m24);

v1 = GetPointFromTwoLines(l11,l12);
v2 = GetPointFromTwoLines(l21,l22);

r1 = inv(K)*v1/norm(inv(K)*v1);
r2 = inv(K)*v2/norm(inv(K)*v2);

r3 = Vec2Skew(r1)*r2;
```

Change focal length

Orthogonal matrix!

```
R =
   0.5846   -0.8496    0.0508
  -0.4149   -0.3216   -0.8367
   0.6972    0.4180   -0.5405

det(R) =
   0.9948

R'*R =
   1.0662   -0.0118    0.0250
  -0.0118    0.9757    0.0285
   0.0250    0.0285    0.9530
```

Single View Metrology

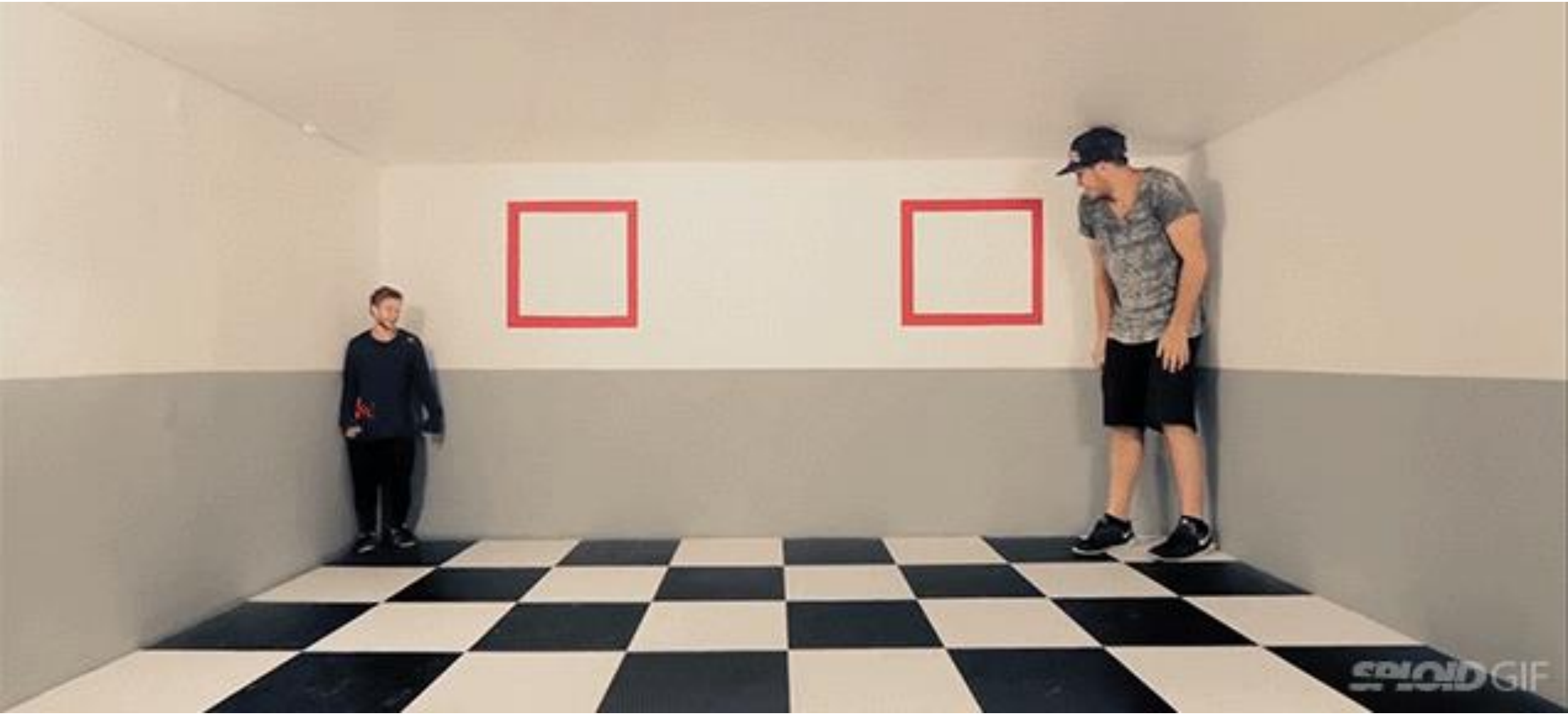


Criminisi et al. “Single View Geometry”, ICCV 1999

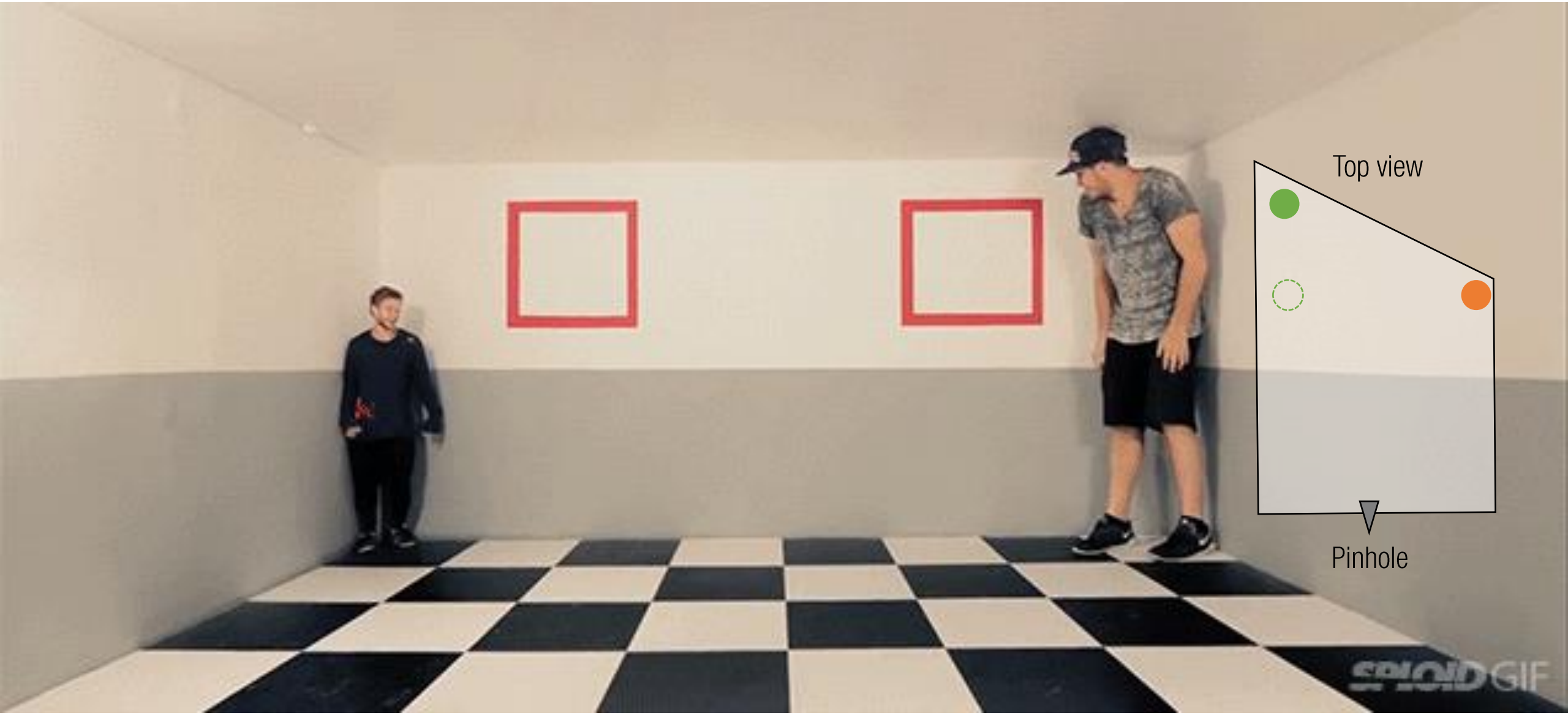
Criminisi et al. “Single View Geometry”, ICCV 1999

Criminisi et al. “Single View Geometry”, ICCV 1999

Hoiem et al. “Automatic Photo Pop-up”, SIGGRAPH 2005



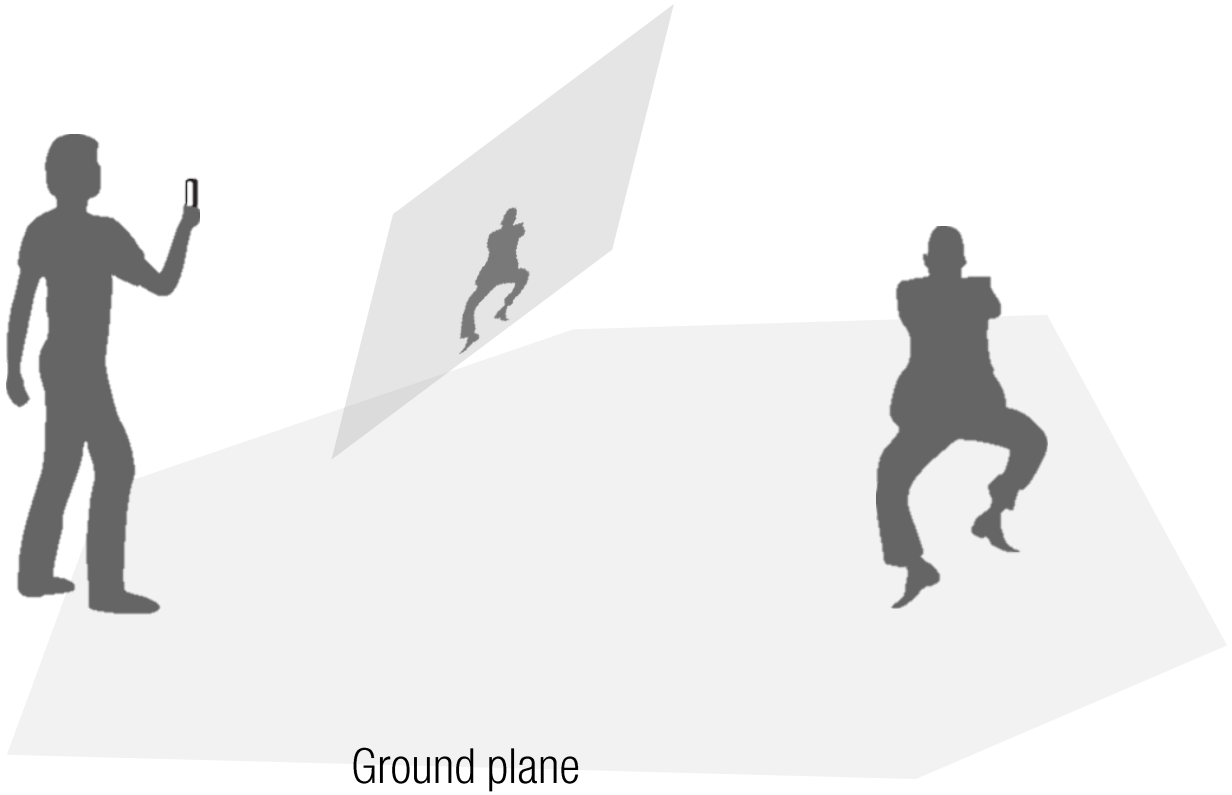
Ames room



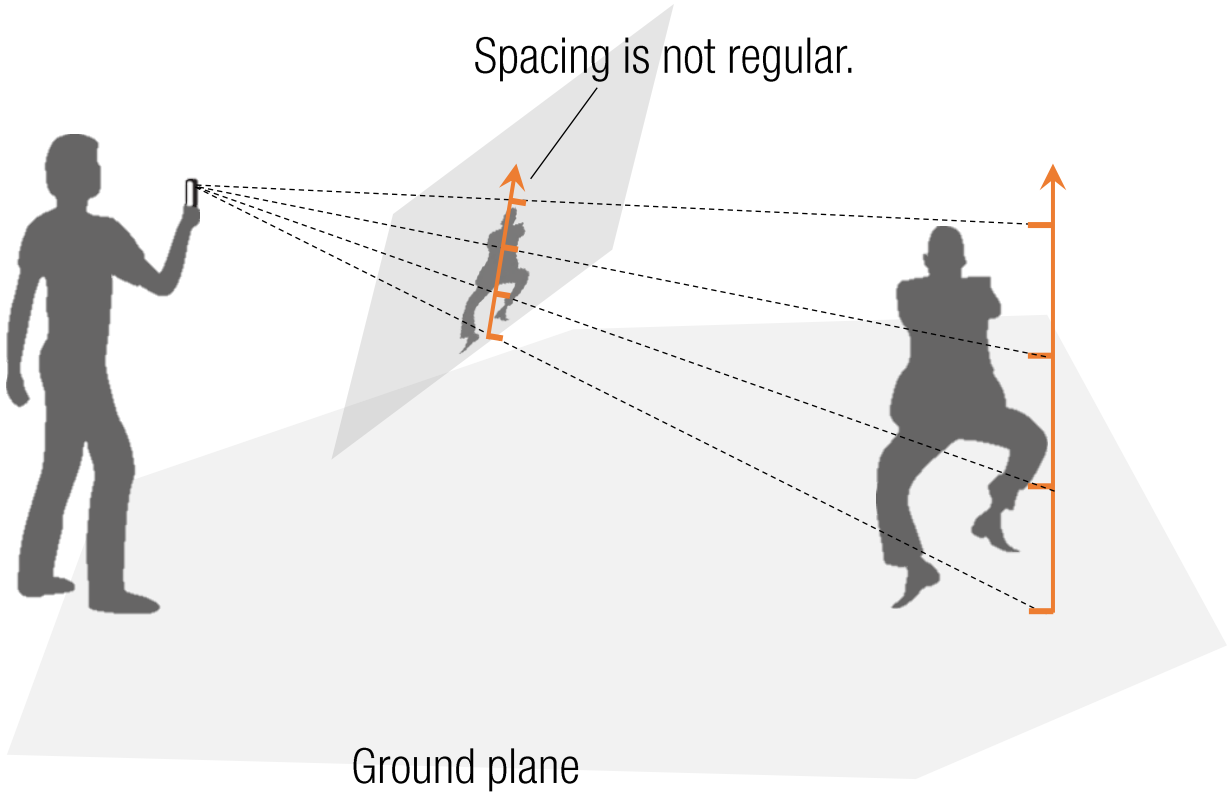
Ames room


MLPS-St. Paul International Airport
Vanishing line
WOMEN
E5
BURGER KING
Distance cannot be trusted.

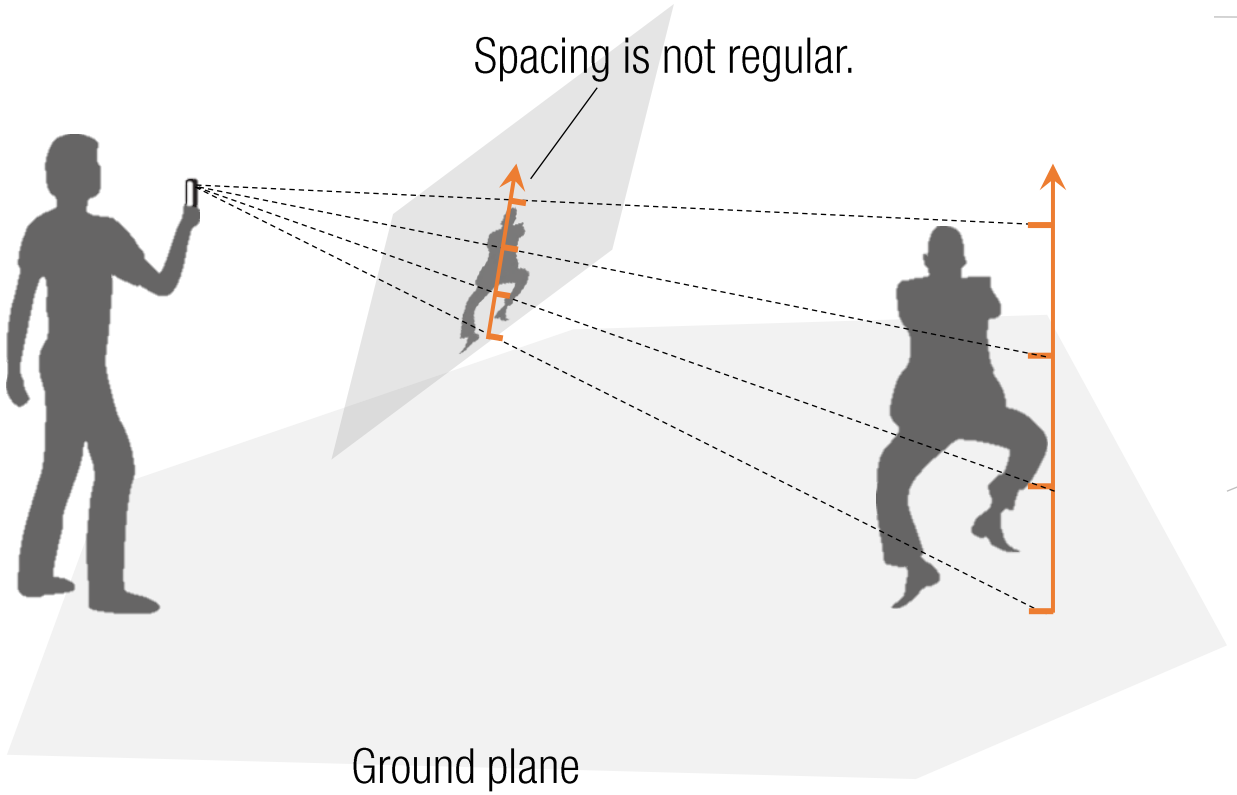
Height from Image

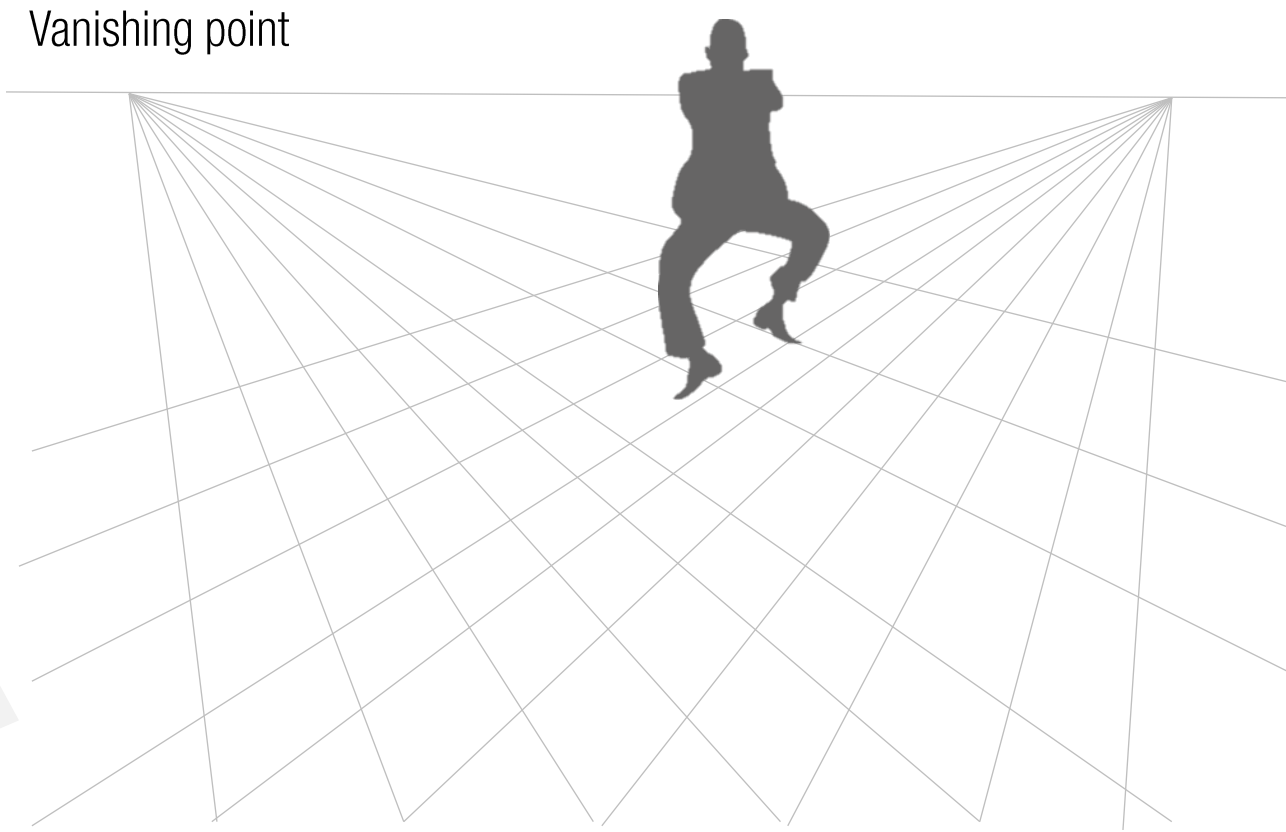


Height from Image

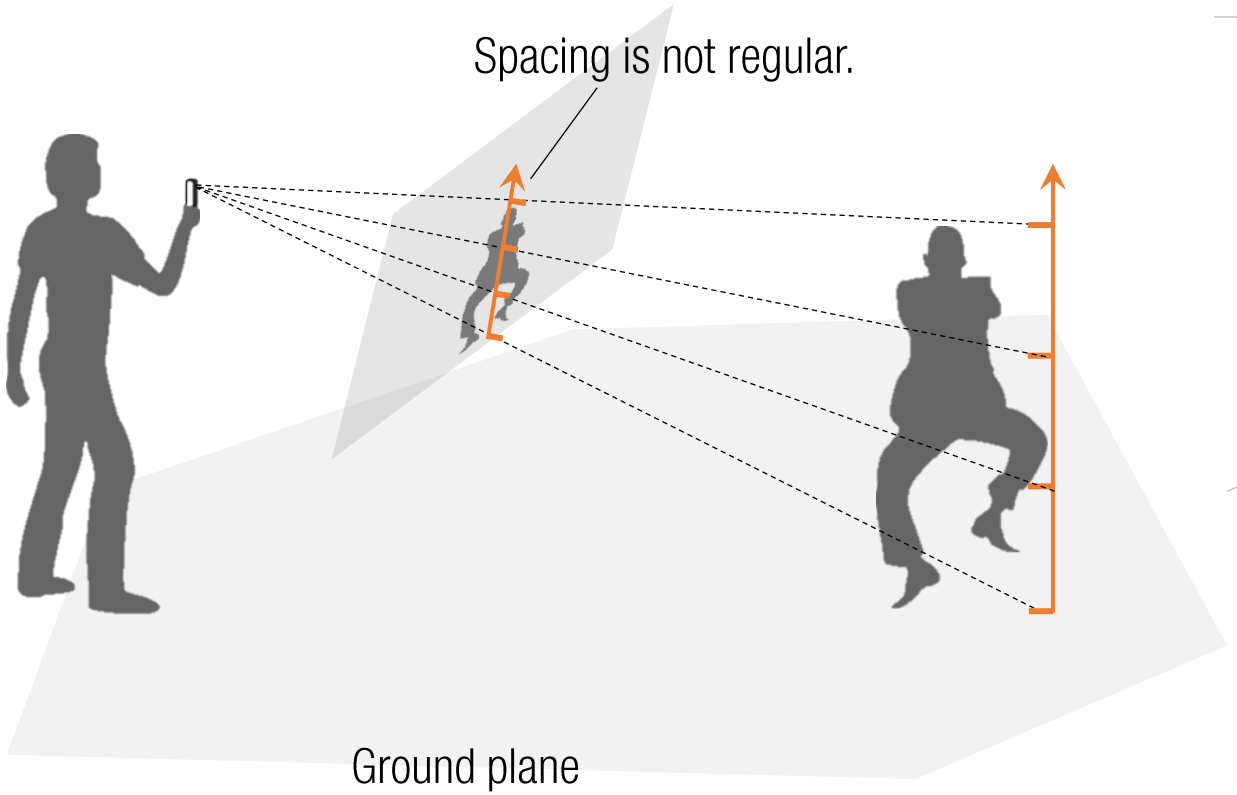


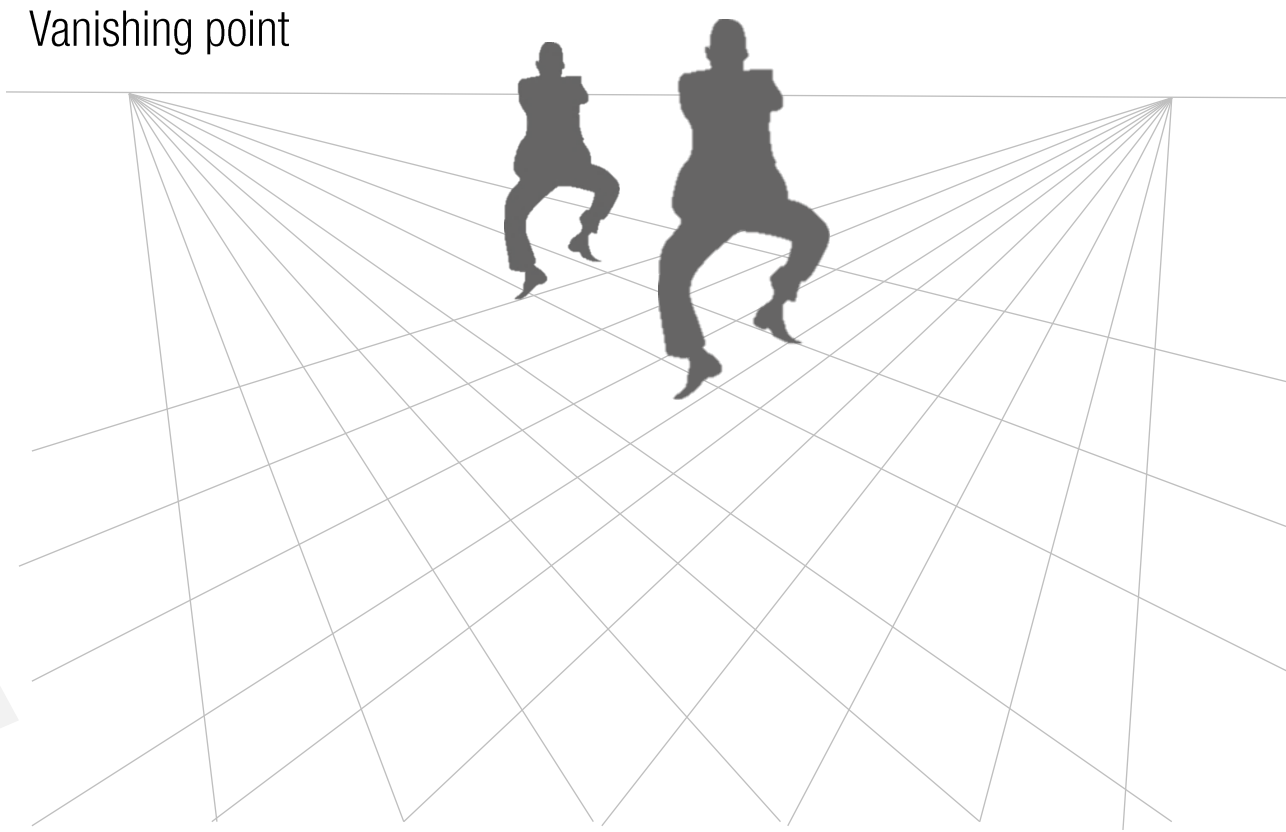
Height from Image



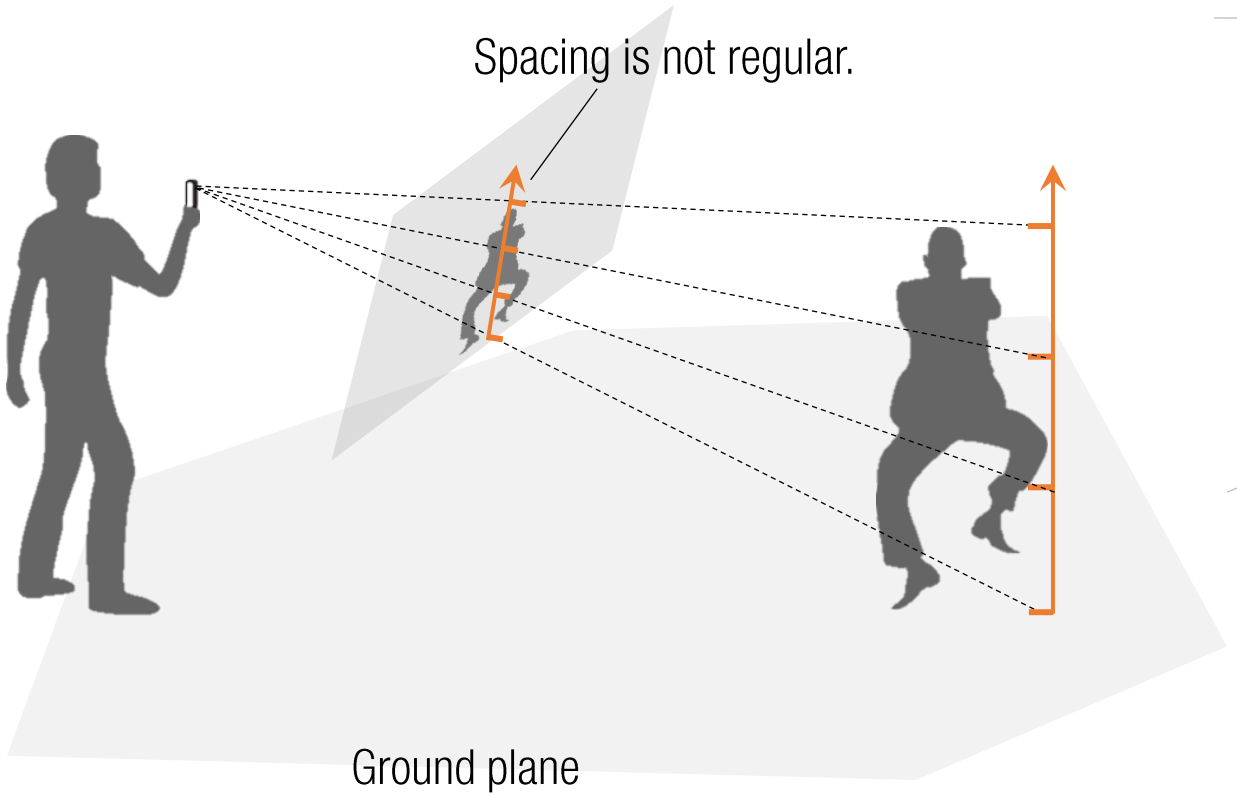


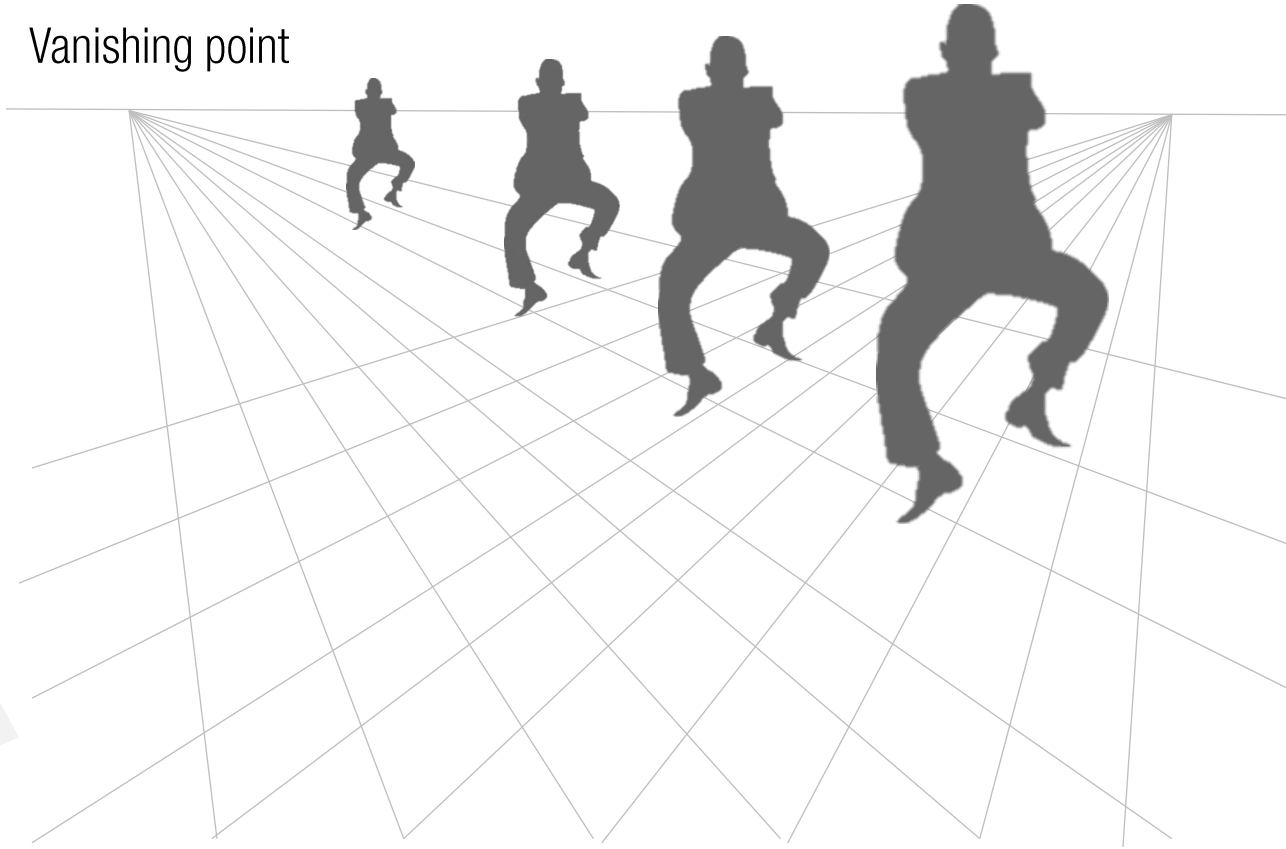
Height from Image



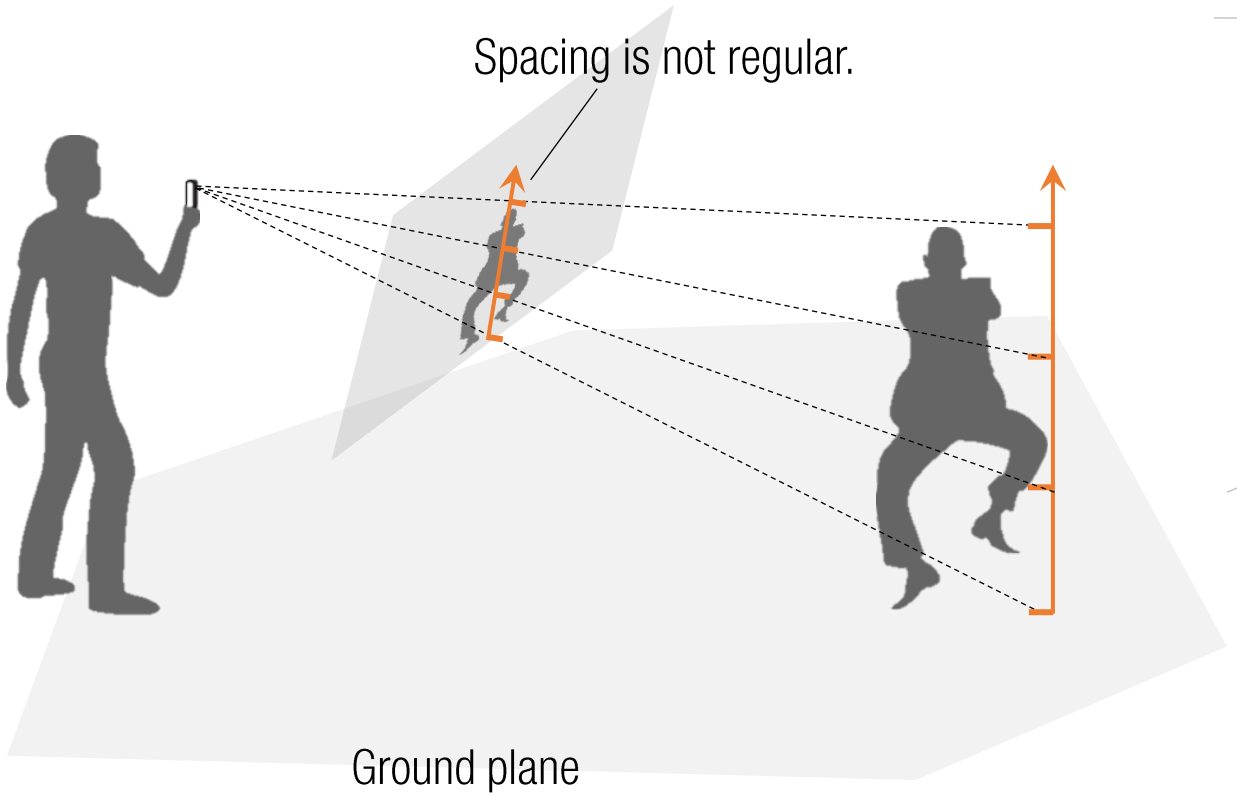


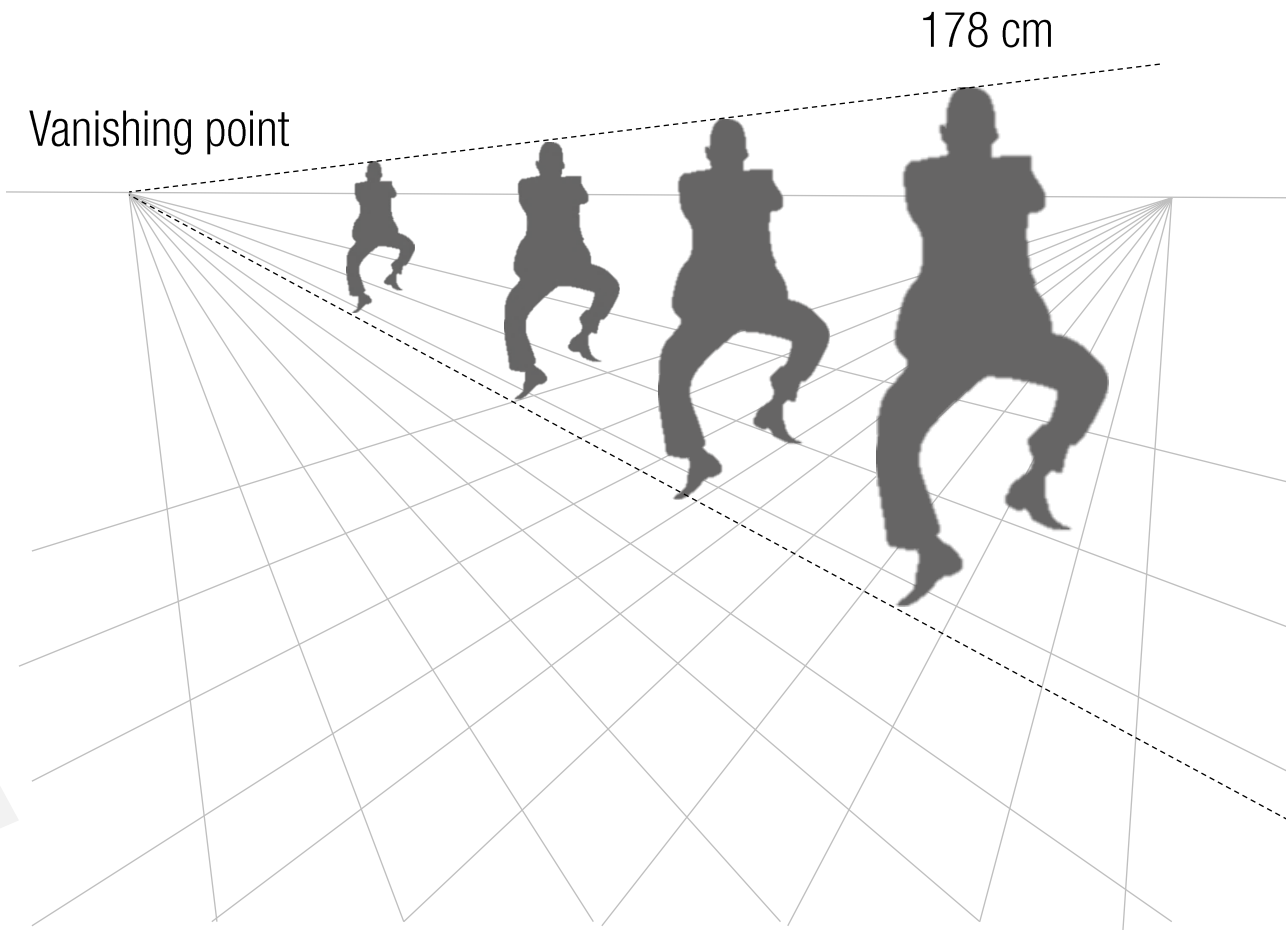
Height from Image



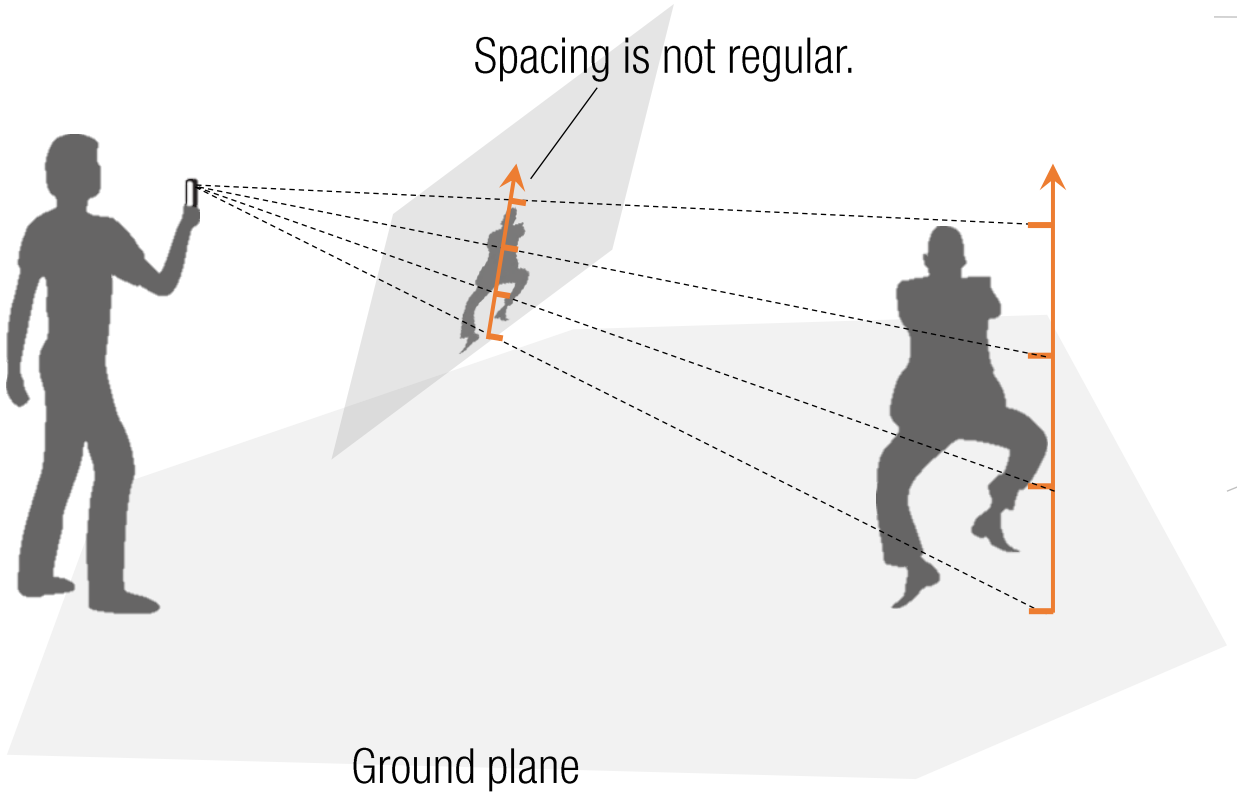


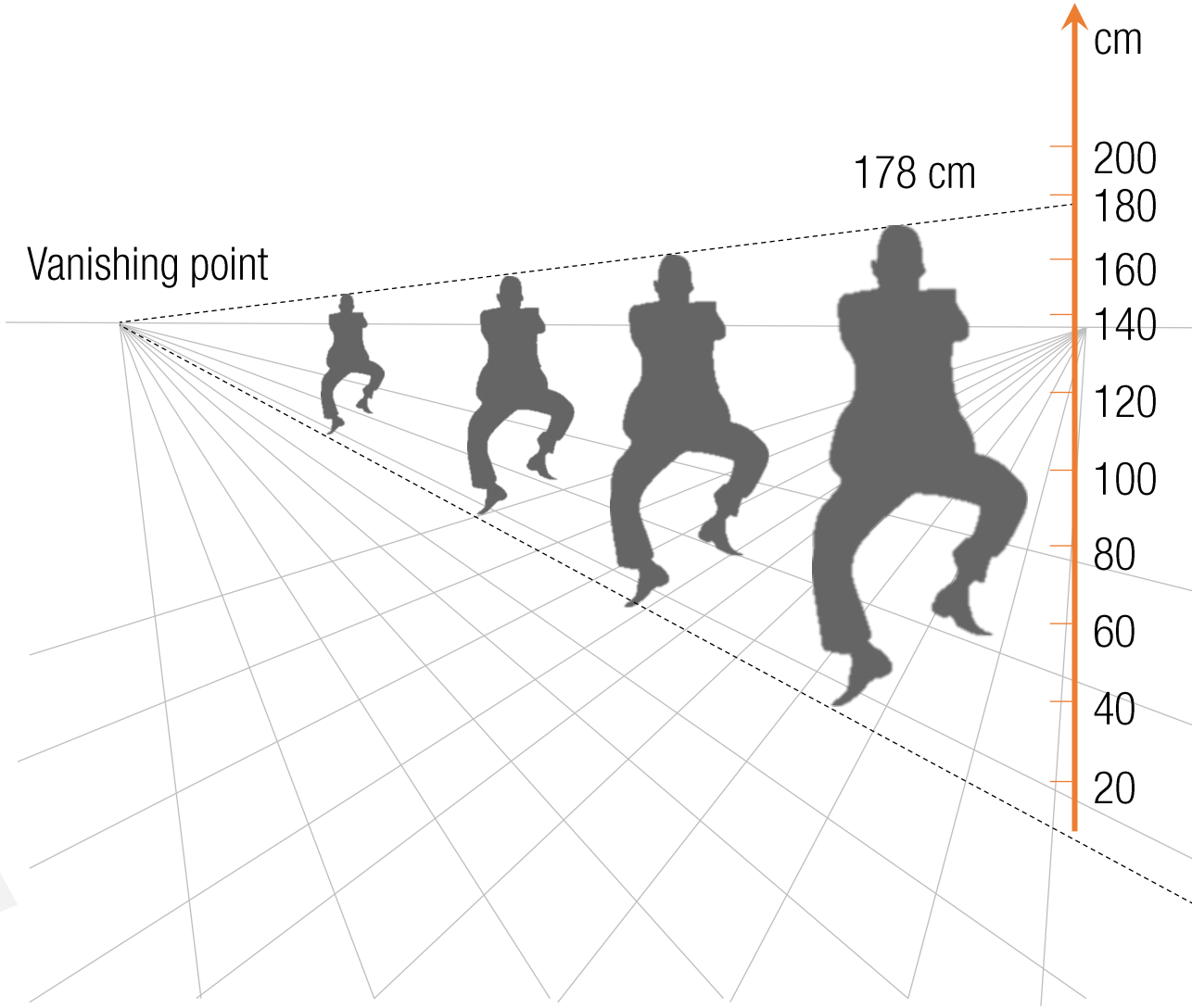
Height from Image



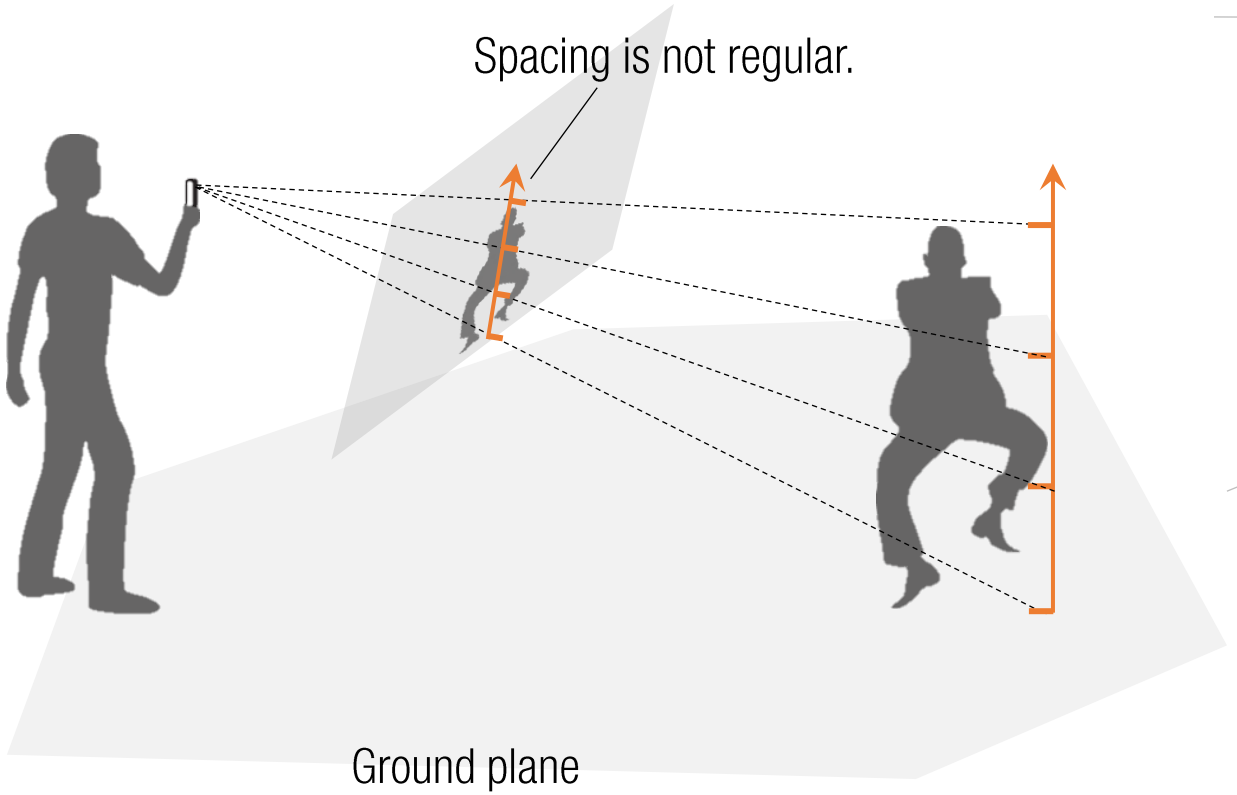


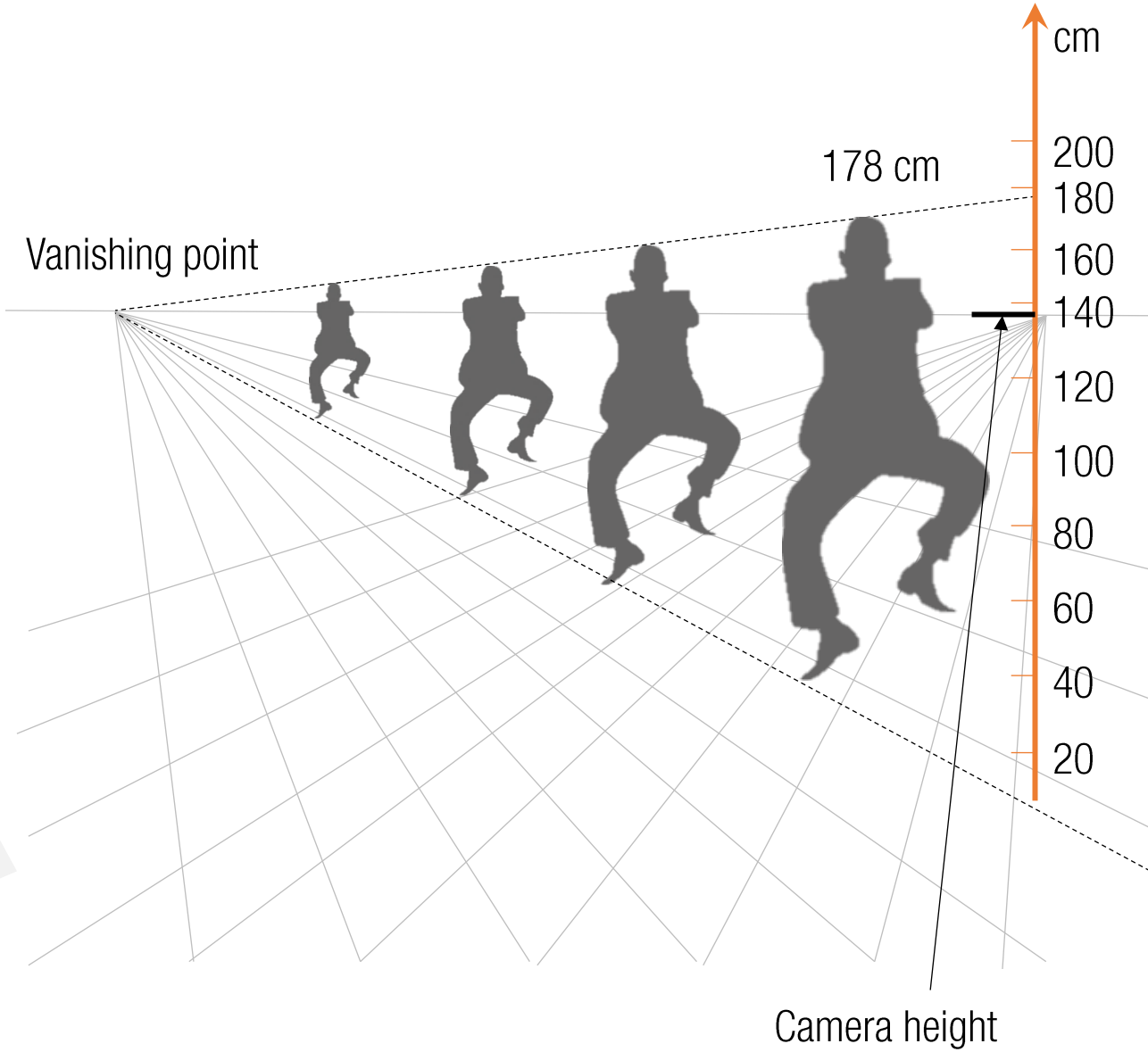
Height from Image





Height from Image



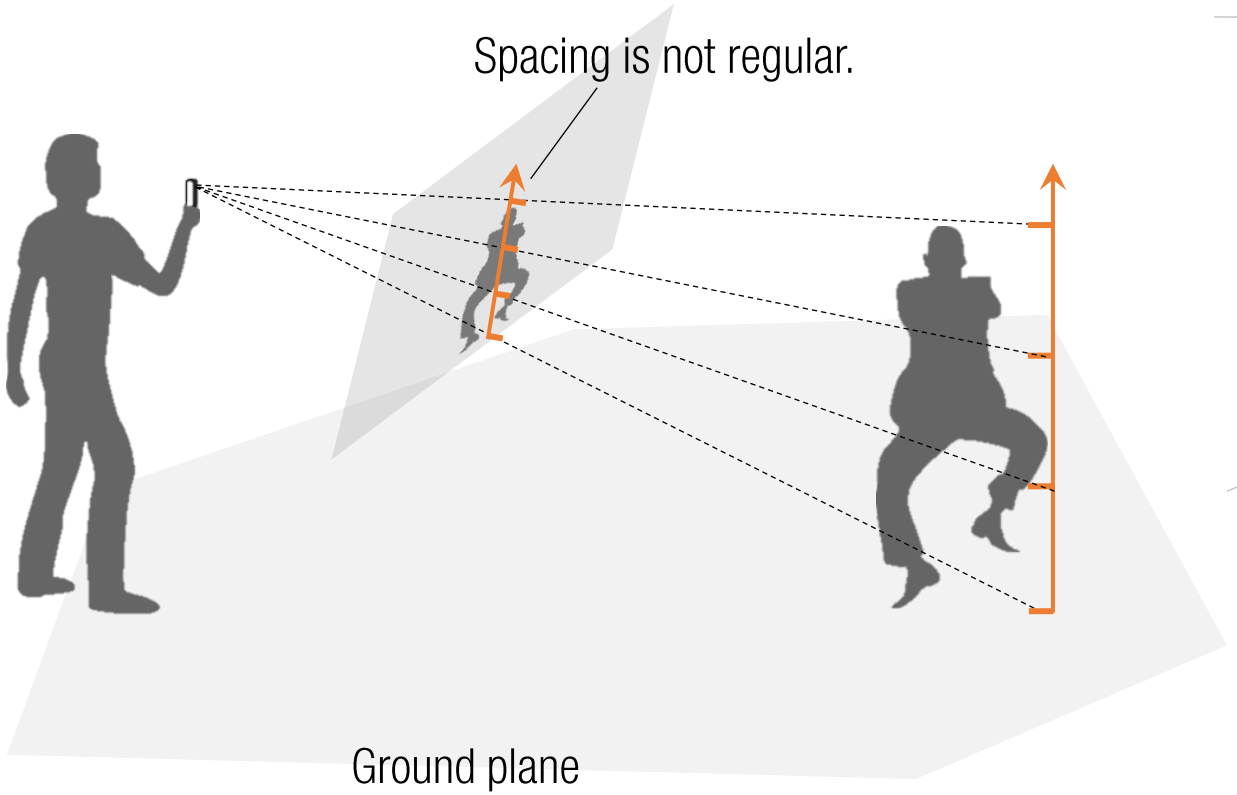


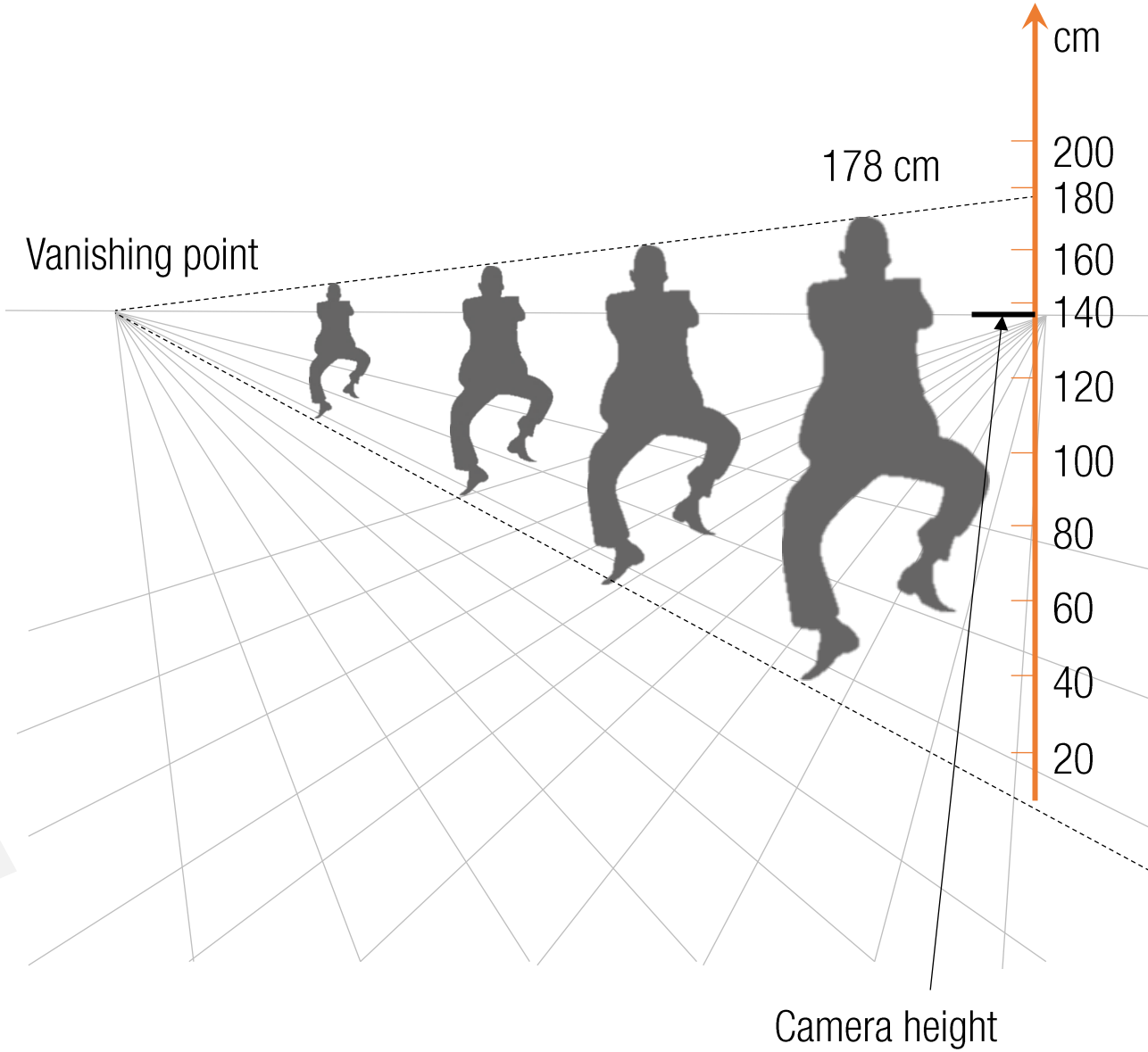
Where was I (how high)?

Taken from my hotel room (6th floor)

Taken from beach

Height from Image

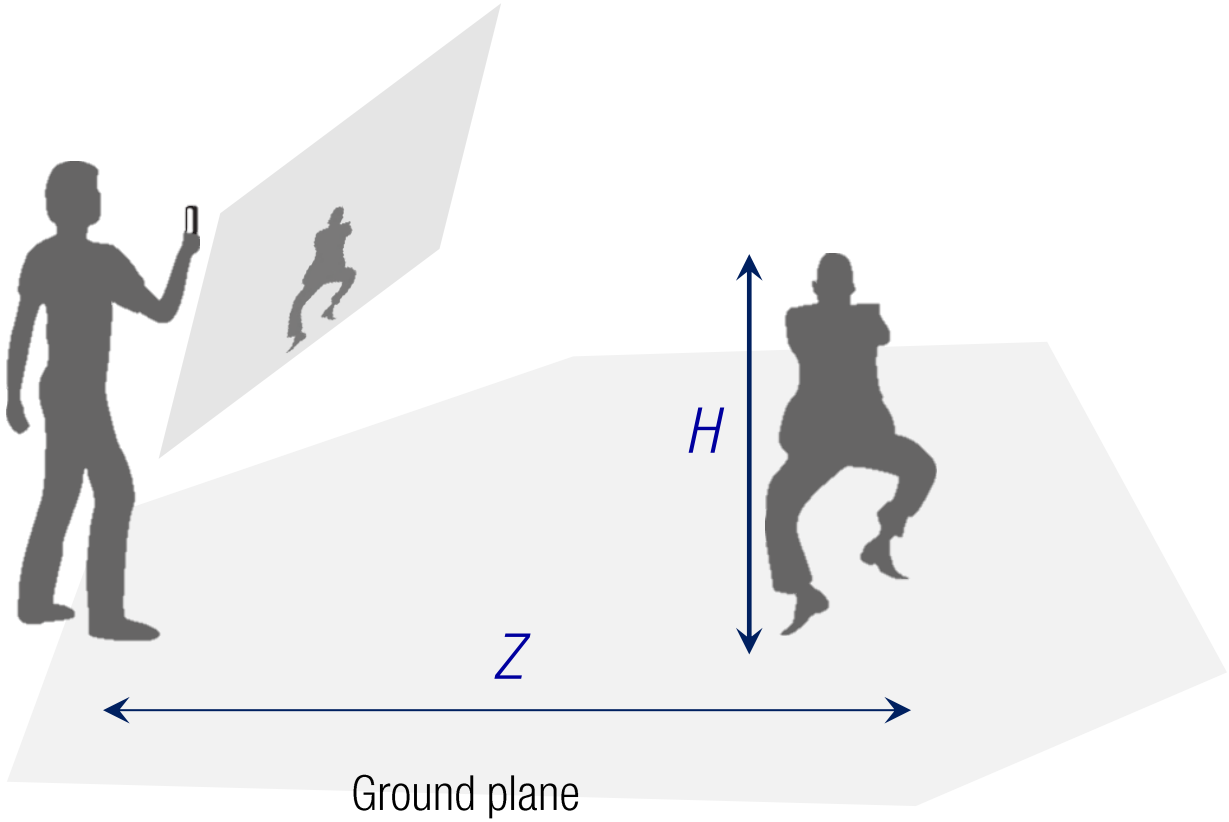


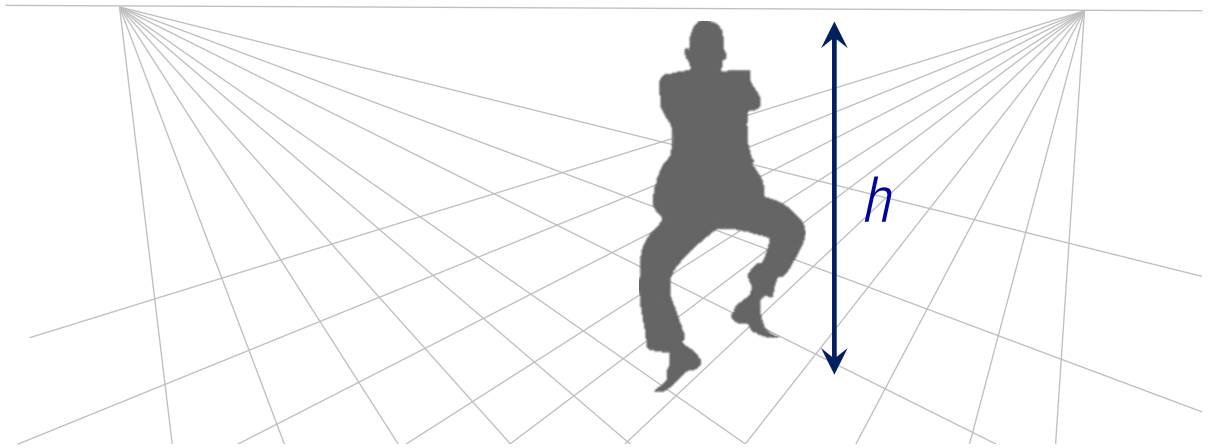


Height from Image

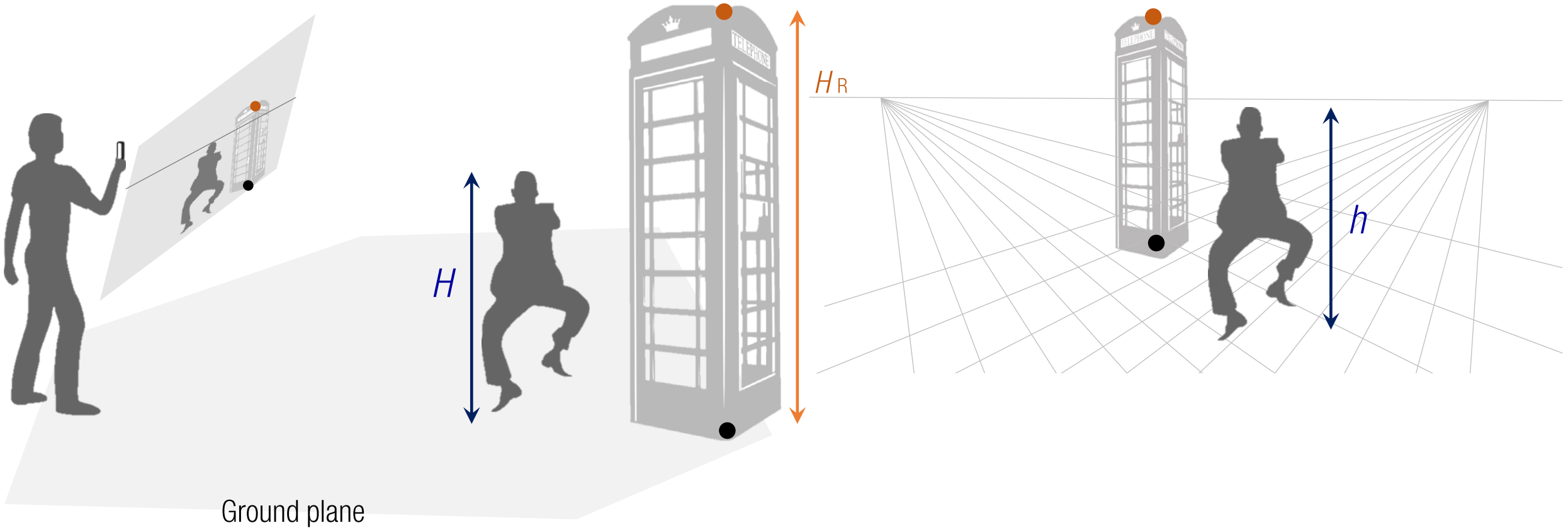
It is impossible to compute the height.

$$h = f\frac{H}{Z}$$

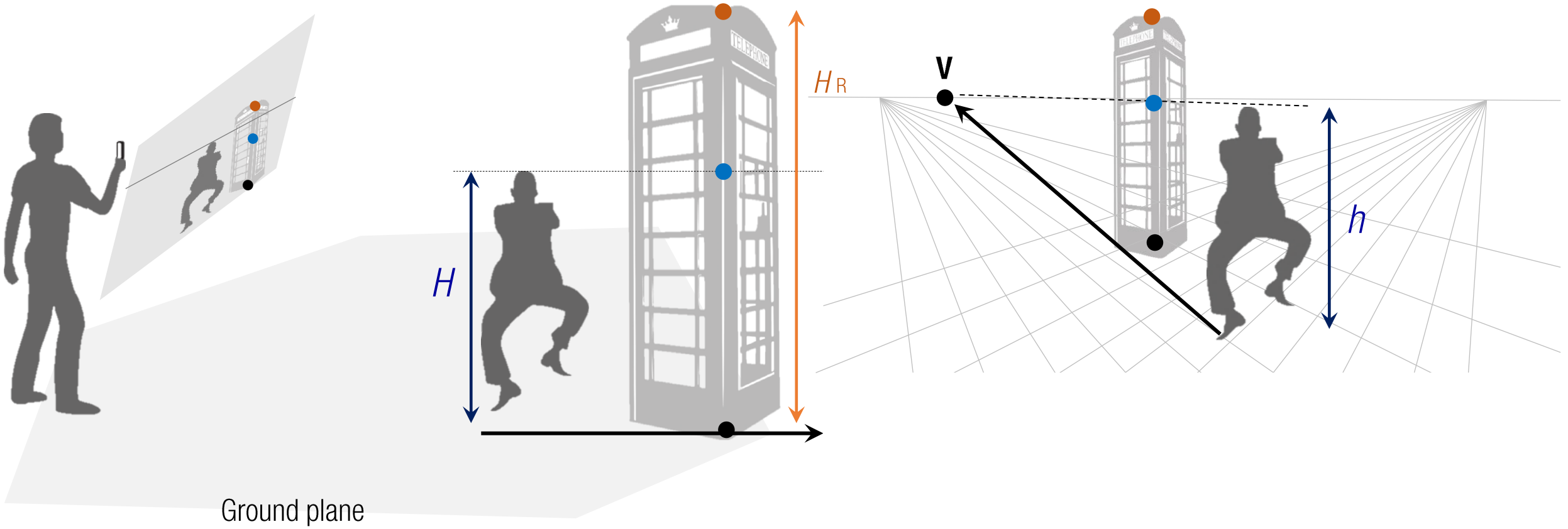




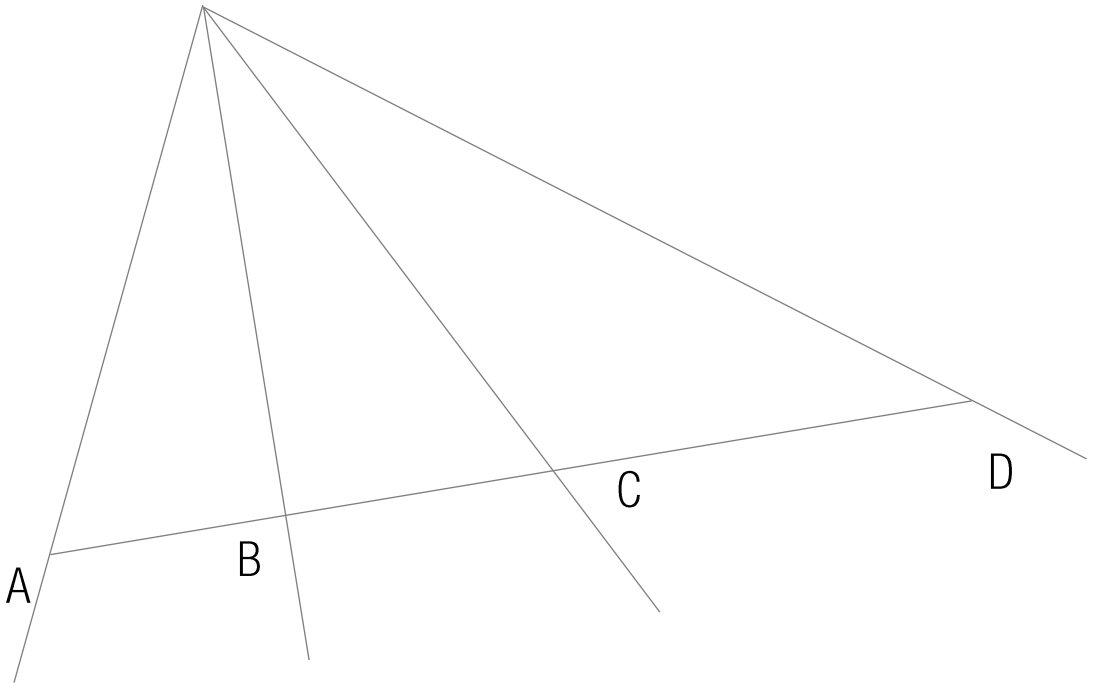
Height from Image



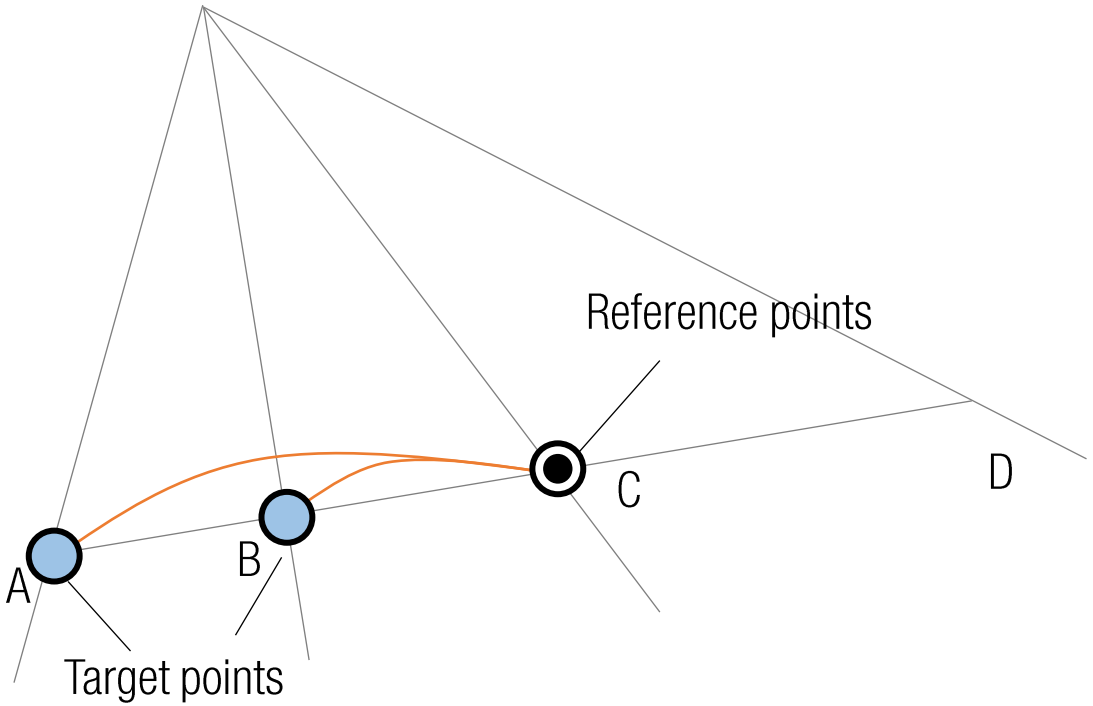
Height from Image



Cross Ratio

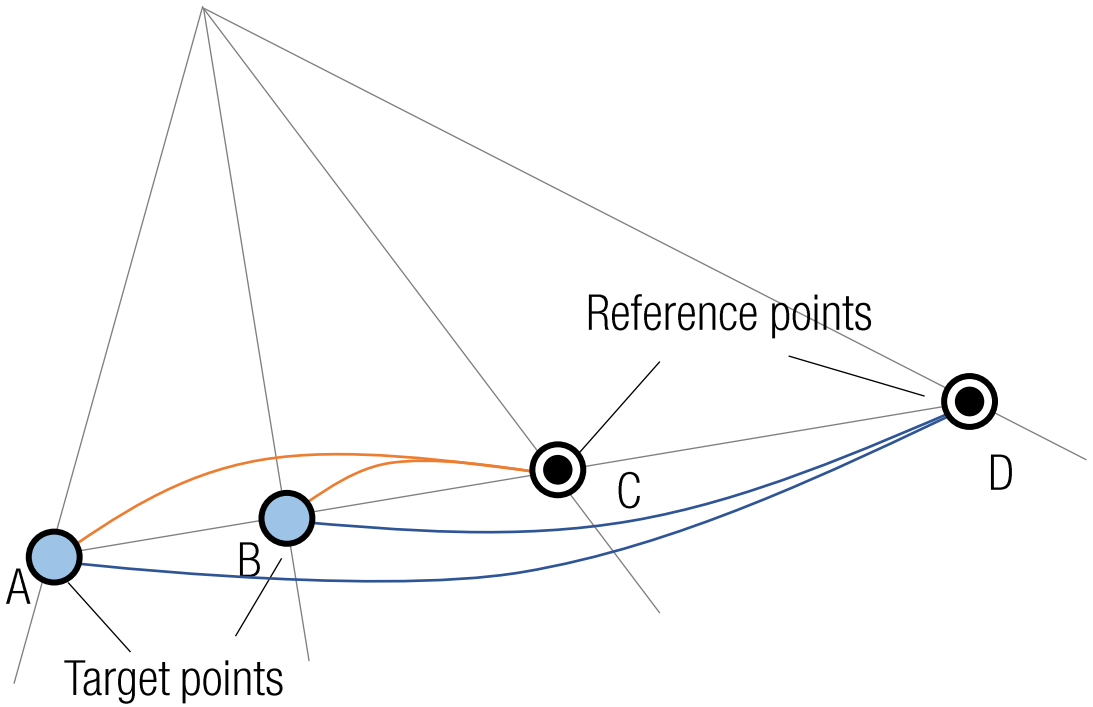


Cross Ratio



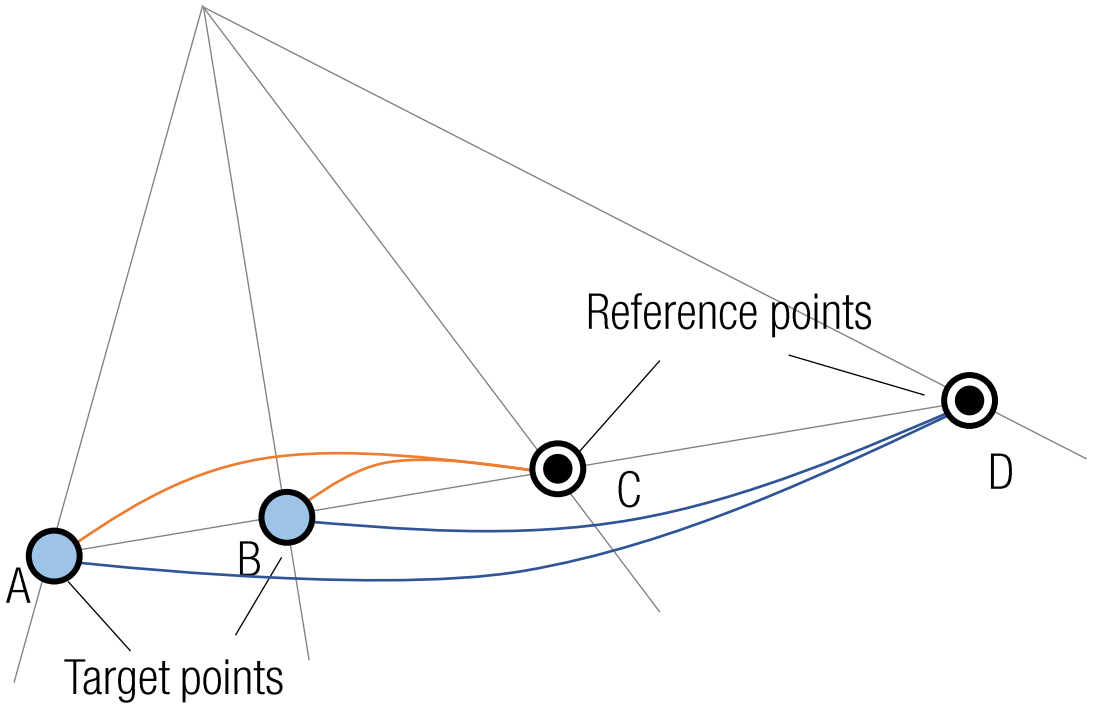
$$\frac{\overline{AC}}{\overline{BC}}$$

Cross Ratio



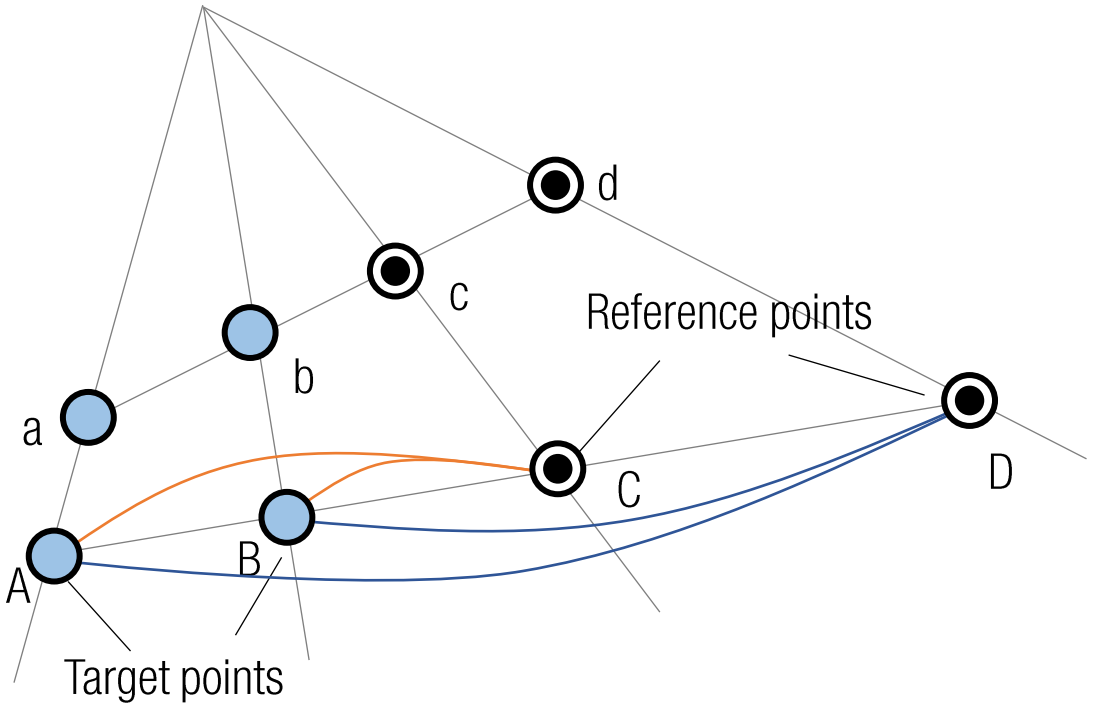
$$\frac{\overline{AC}\,\overline{BD}}{\overline{BC}\,\overline{AD}}$$

Cross Ratio



$$\frac{\overline{AC}}{\overline{BC}}\,\frac{\overline{BD}}{\overline{AD}}$$

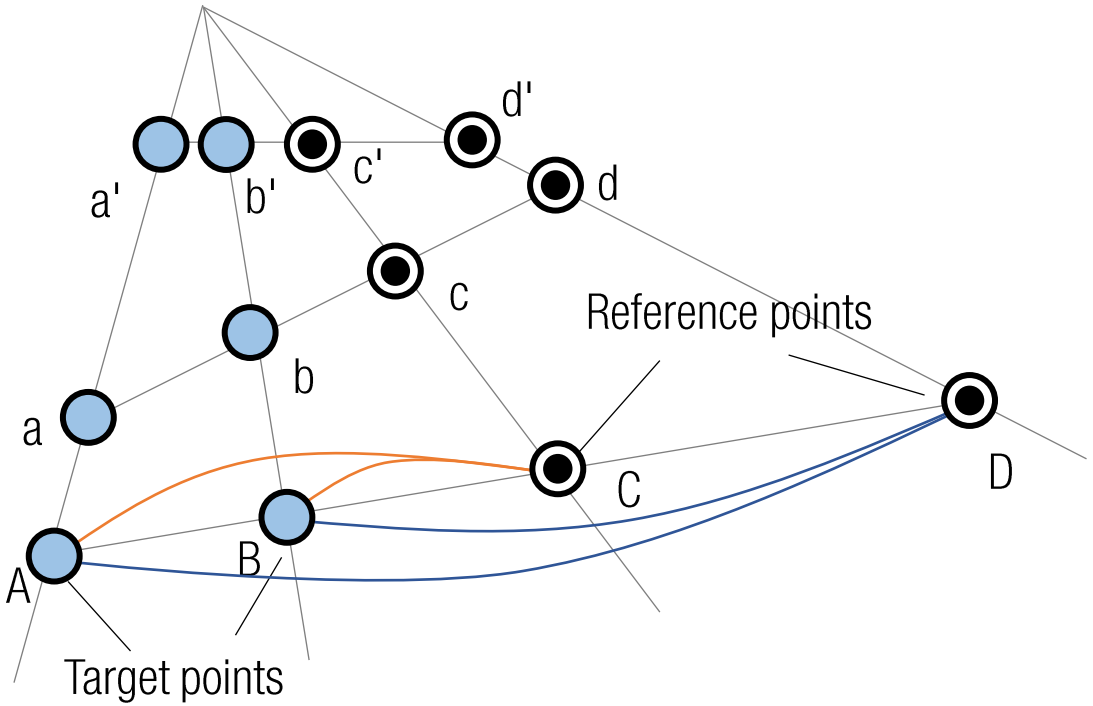
Cross Ratio



$$\frac{\overline{AC}\,\overline{BD}}{\overline{BC}\,\overline{AD}} = \frac{\overline{ac}\,\overline{bd}}{\overline{bc}\,\overline{ad}}$$

Cross ratio (perspective transformation invariant)

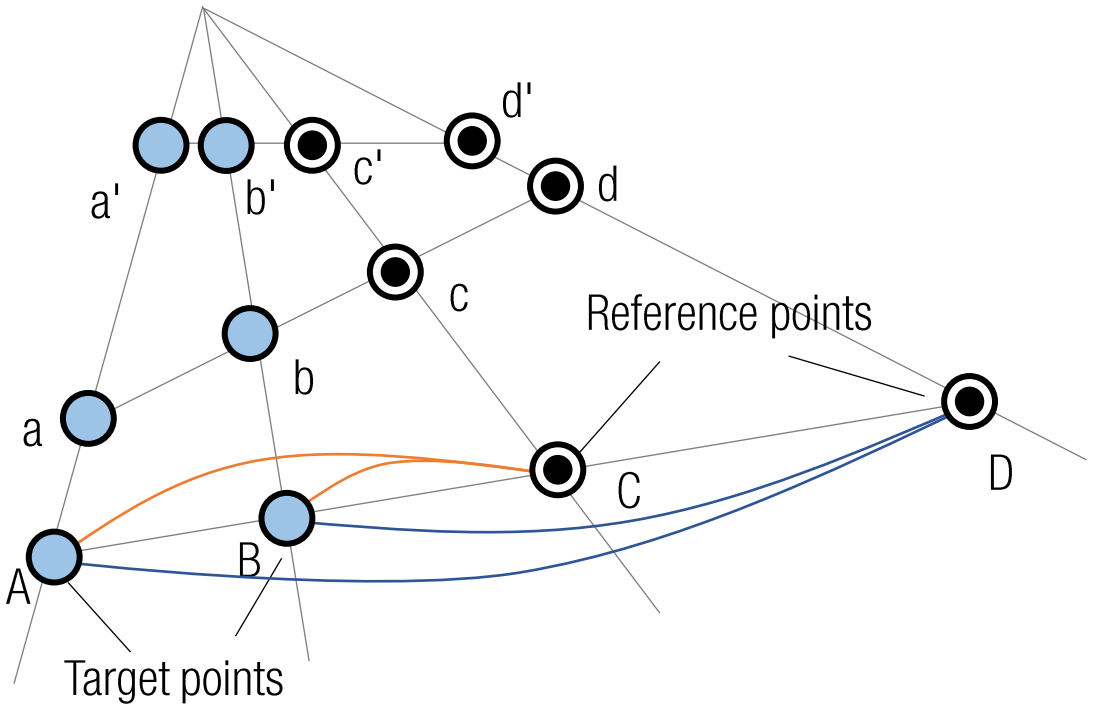
Cross Ratio



$$\frac{\overline{AC}\,\overline{BD}}{\overline{BC}\,\overline{AD}} = \frac{\overline{ac}\,\overline{bd}}{\overline{bc}\,\overline{ad}} = \frac{\overline{a'c'}\,\overline{b'd'}}{\overline{b'c'}\,\overline{a'd'}}$$

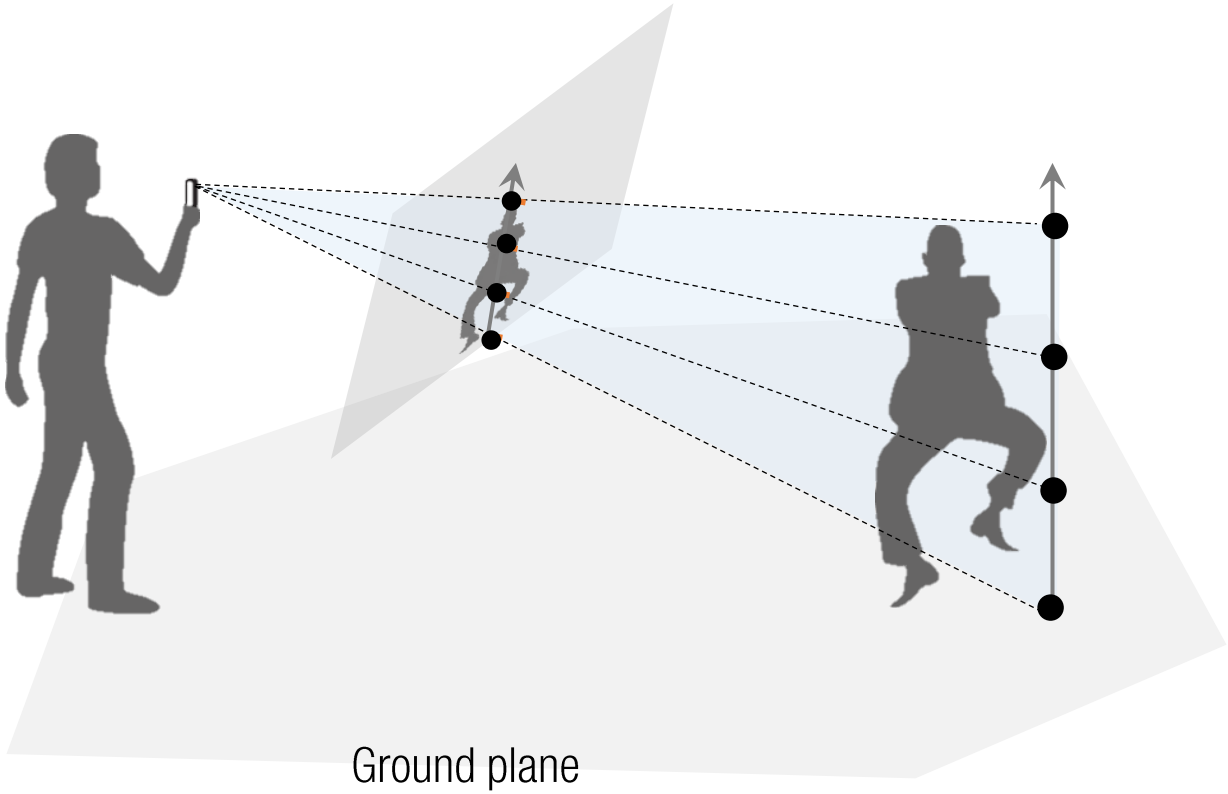
Cross ratio (perspective transformation invariant)

Cross Ratio

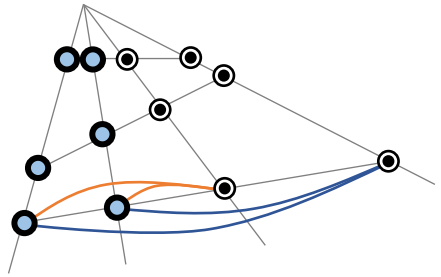


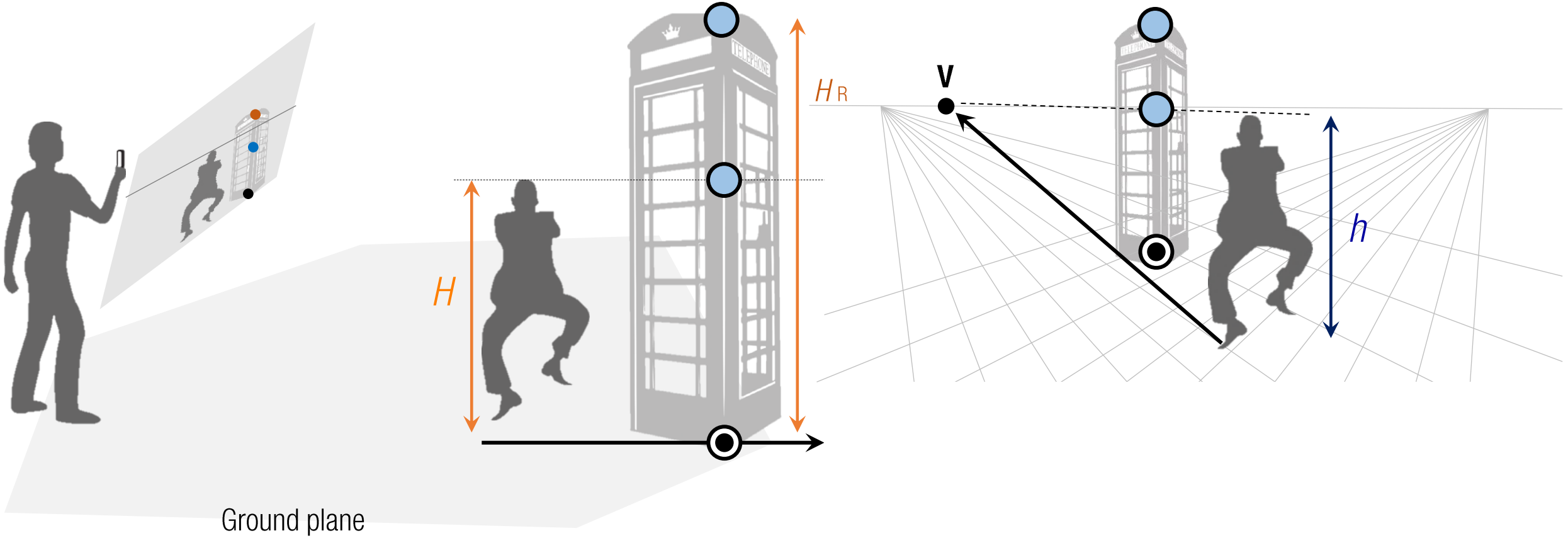
$$\frac{\overline{AC}\,\overline{BD}}{\overline{BC}\,\overline{AD}} = \frac{\overline{ac}\,\overline{bd}}{\overline{bc}\,\overline{ad}} = \frac{\overline{a'c'}\,\overline{b'd'}}{\overline{b'c'}\,\overline{a'd'}}$$

Cross ratio (perspective transformation invariant)

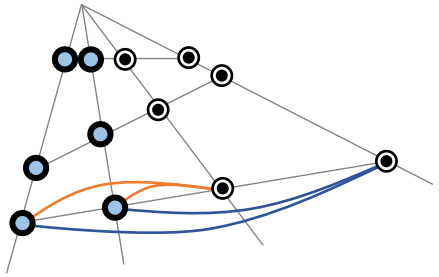


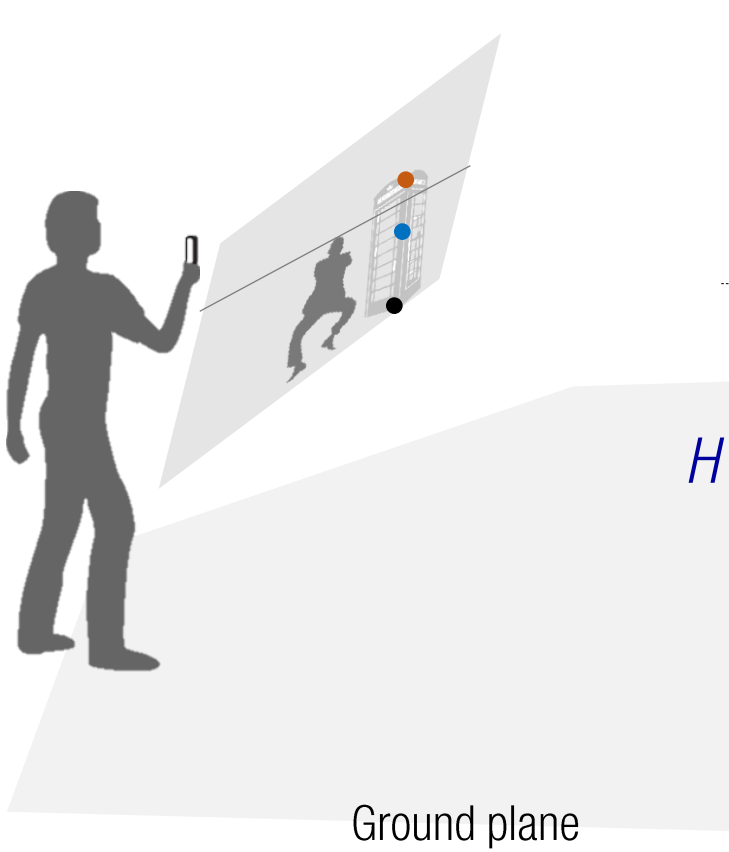
Height from Image

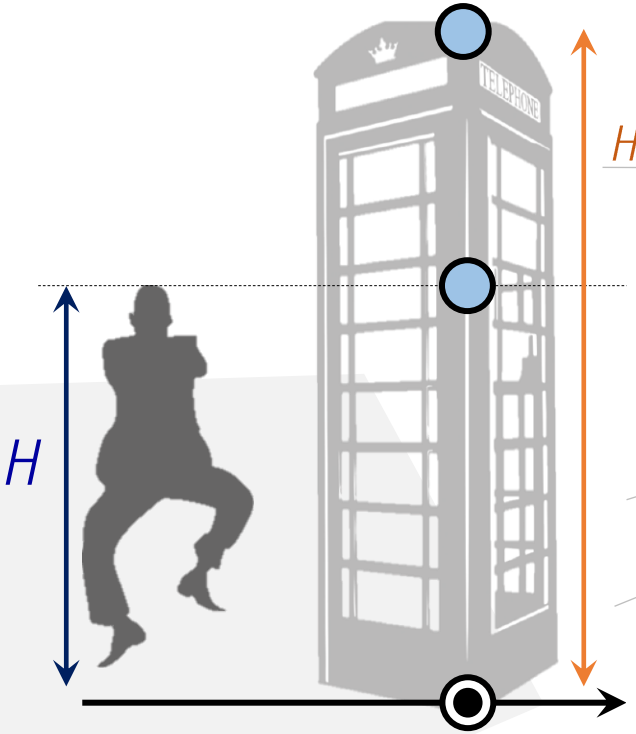


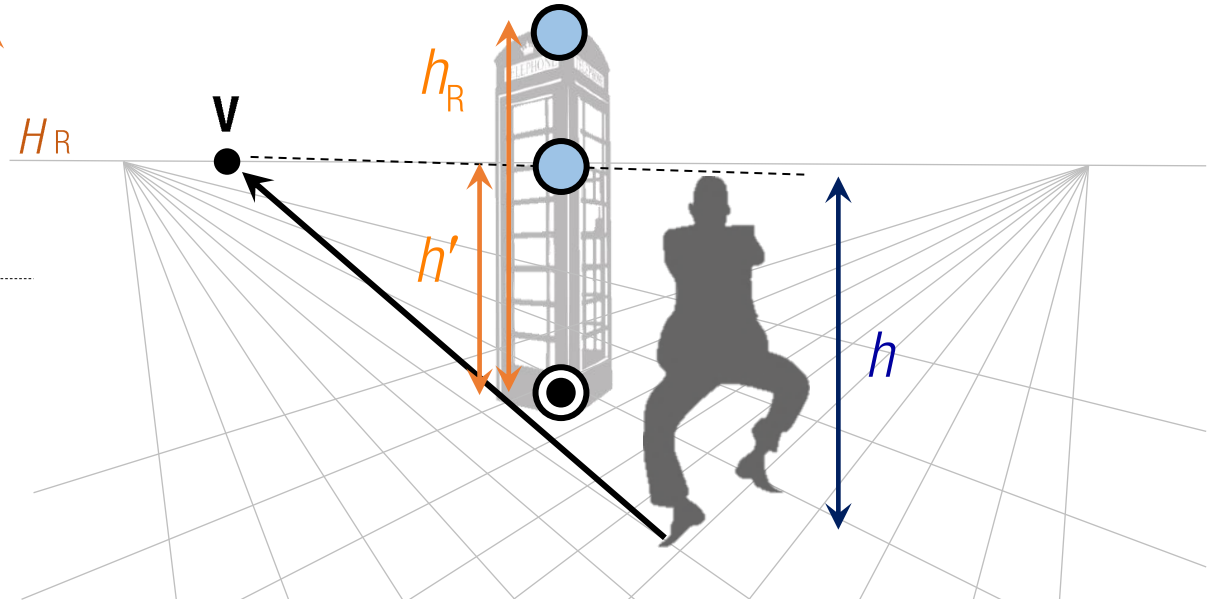


Height from Image





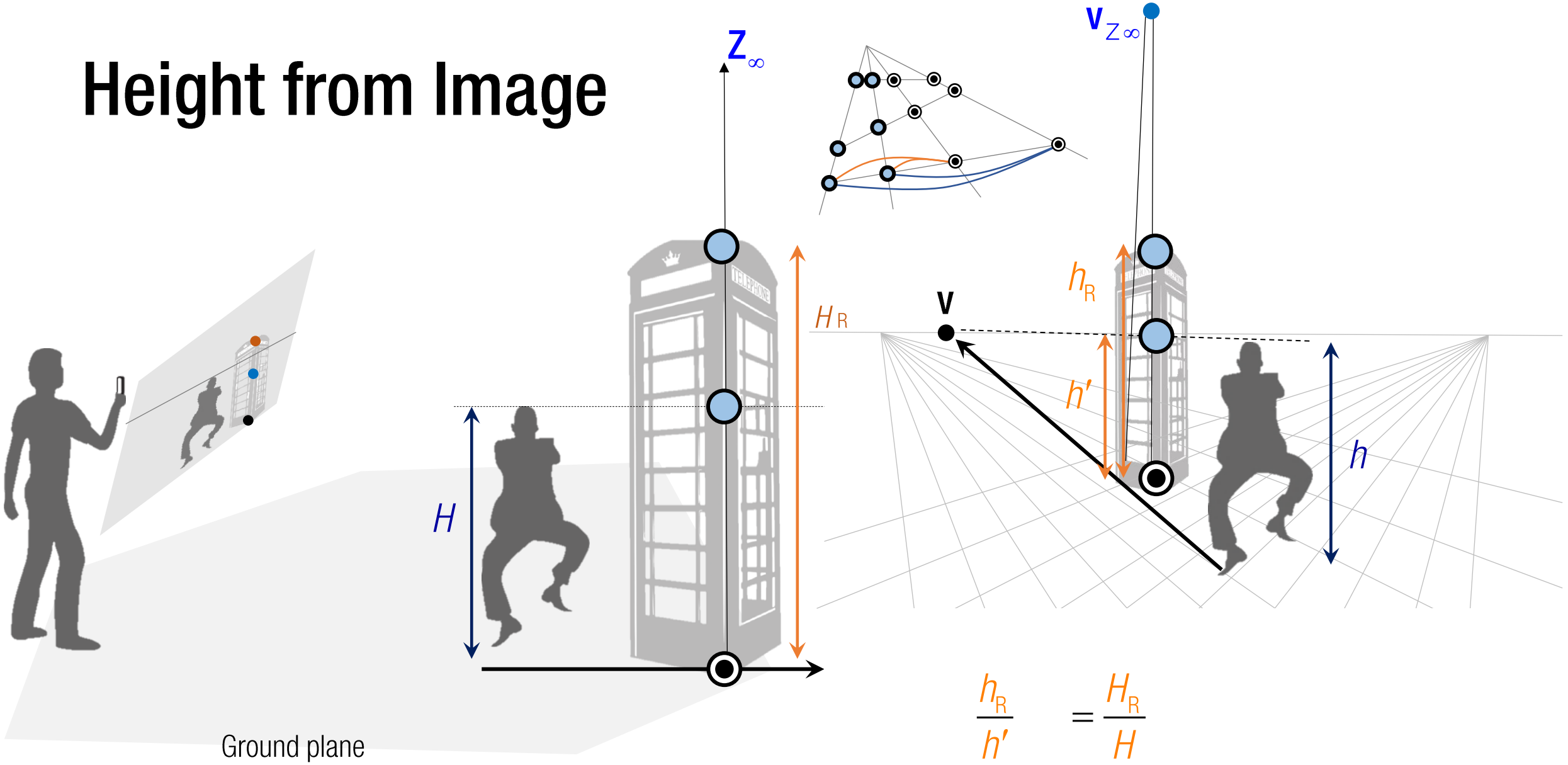




Ground plane

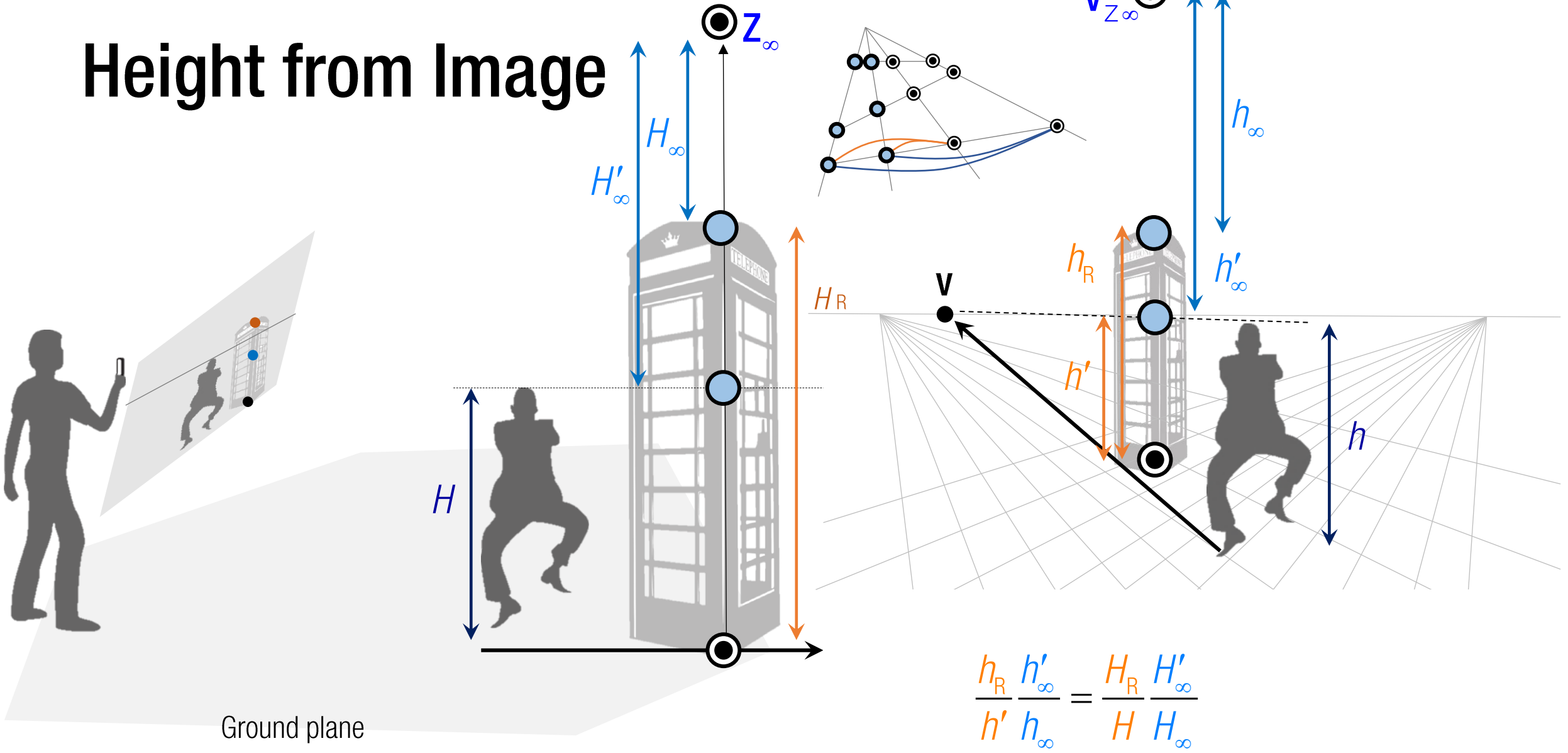
$$\frac{h_R}{h'} = \frac{H_R}{H}$$

Height from Image



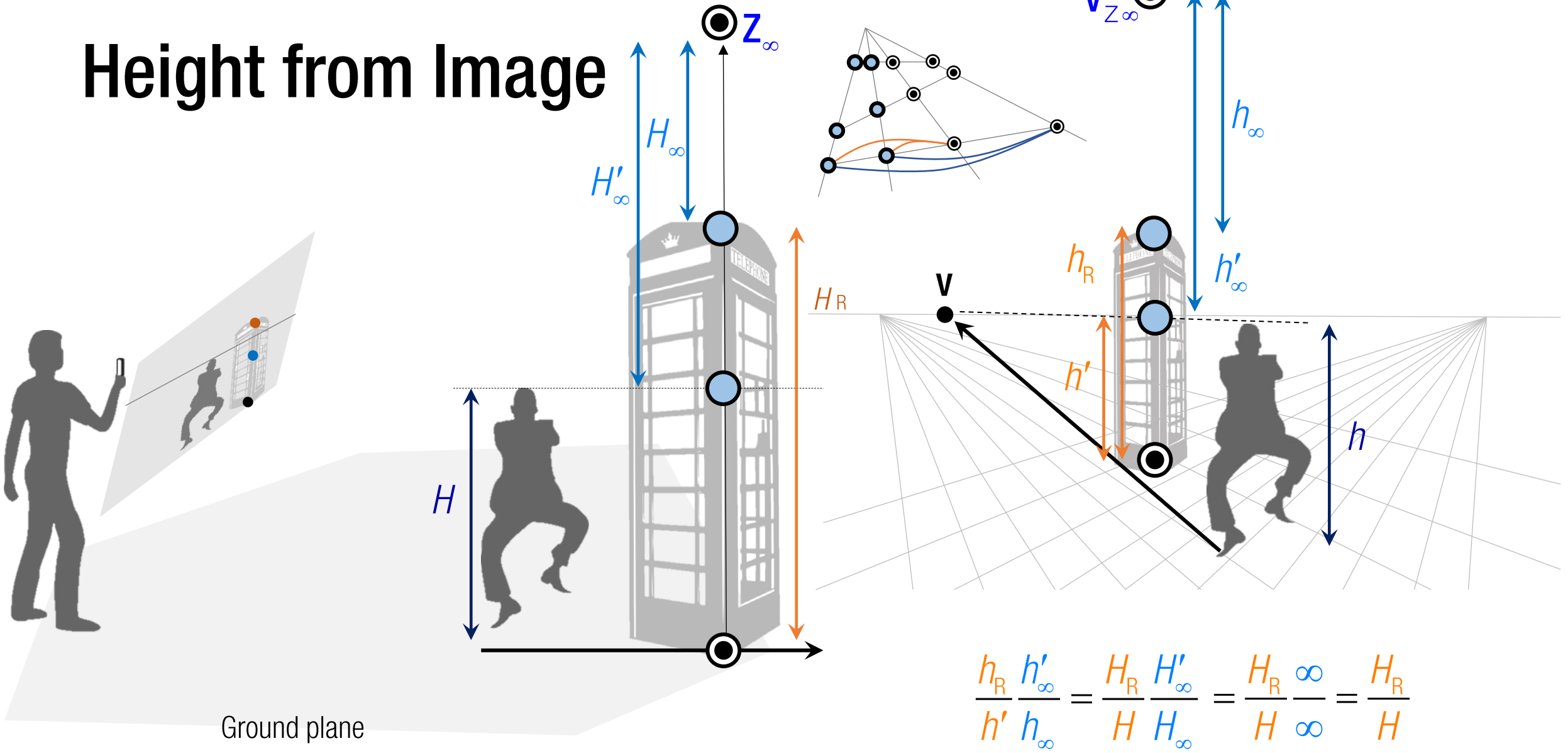
$$\frac{h_R}{h'} = \frac{H_R}{H}$$

Height from Image



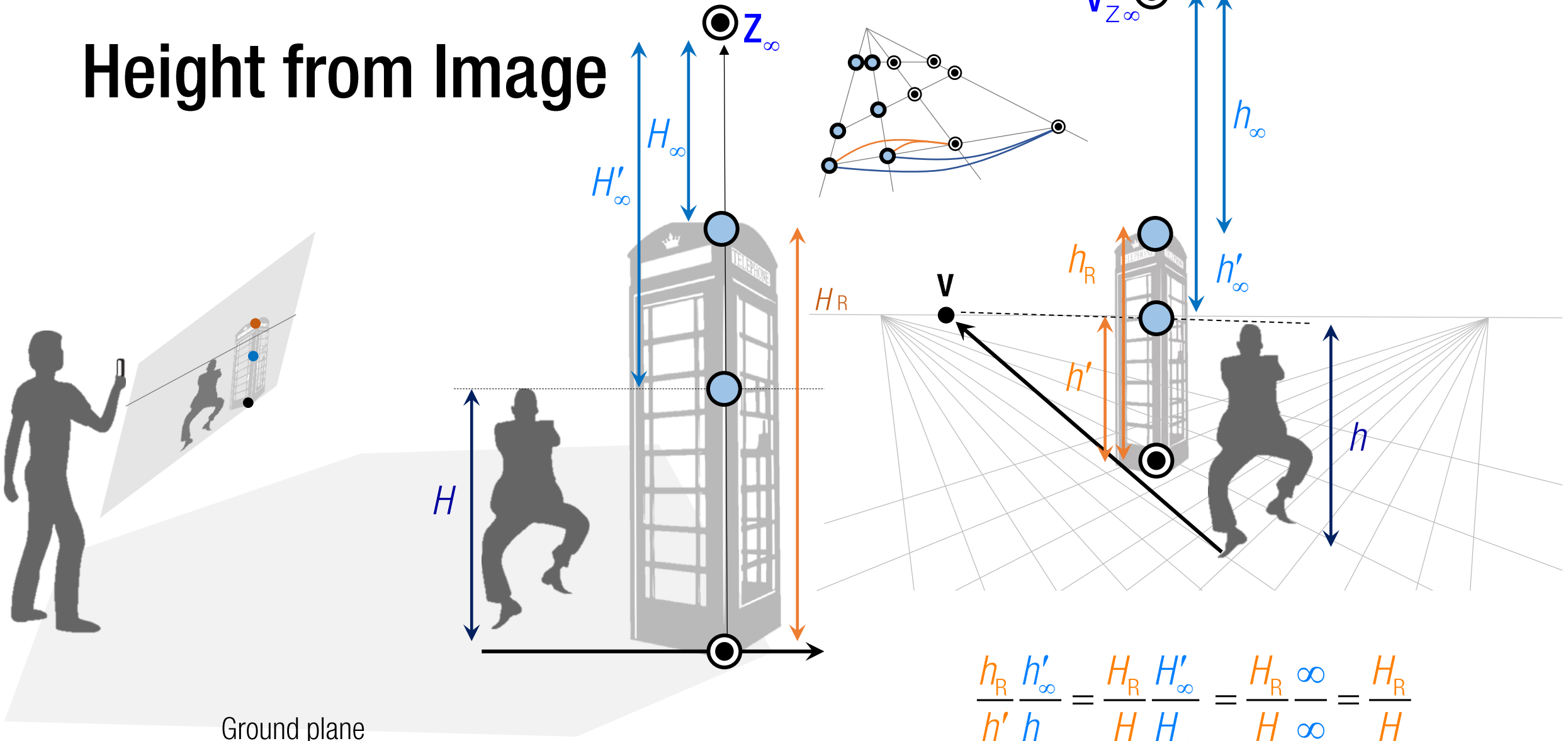
$$\frac{h_R}{h'}\frac{h'_\infty}{h_\infty} = \frac{H_R}{H}\frac{H'_\infty}{H_\infty}$$

Height from Image



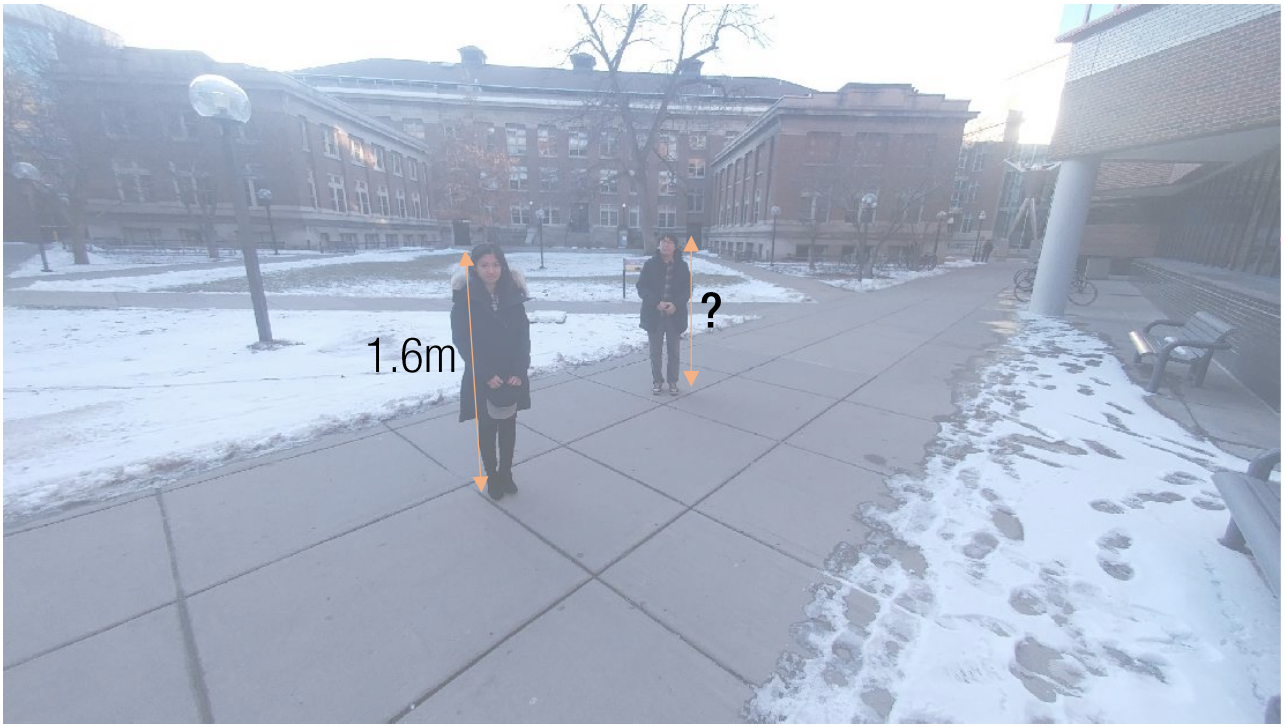
$$\frac{h_R}{h'}\frac{h'_\infty}{h_\infty} = \frac{H_R}{H}\frac{H'_\infty}{H_\infty} = \frac{H_R}{H}\frac{\infty}{\infty} = \frac{H_R}{H}$$

Height from Image

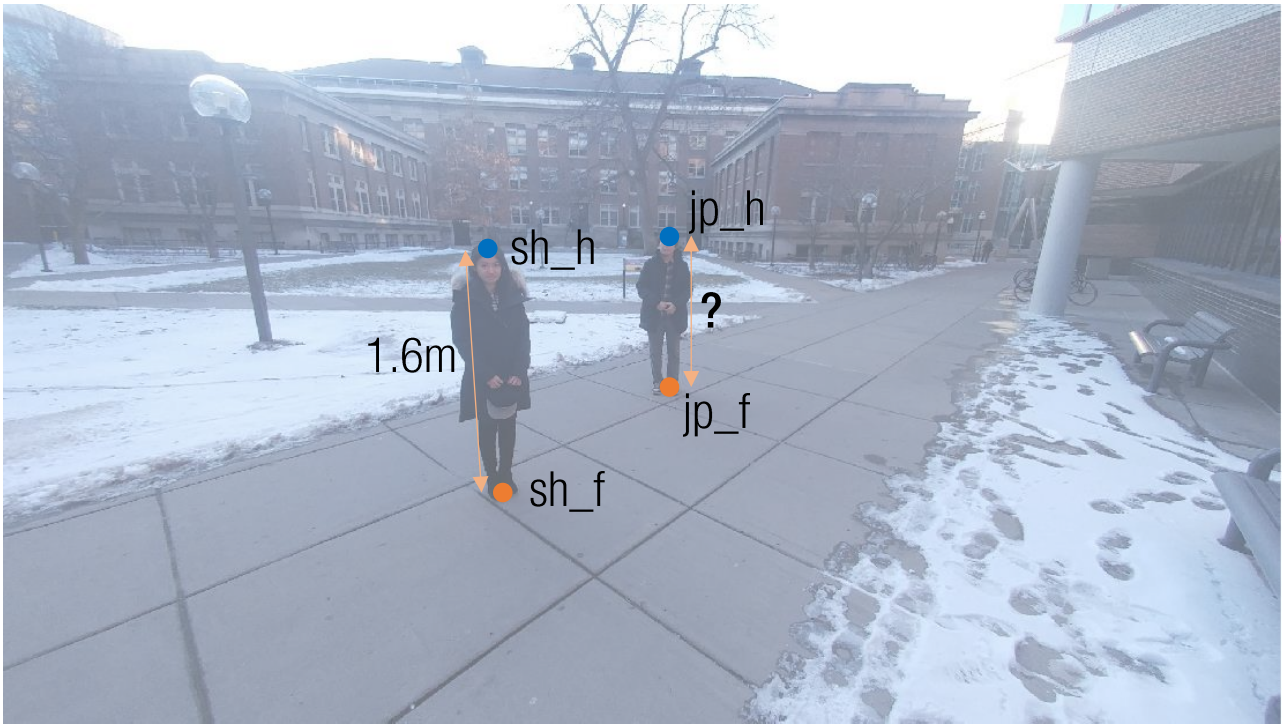


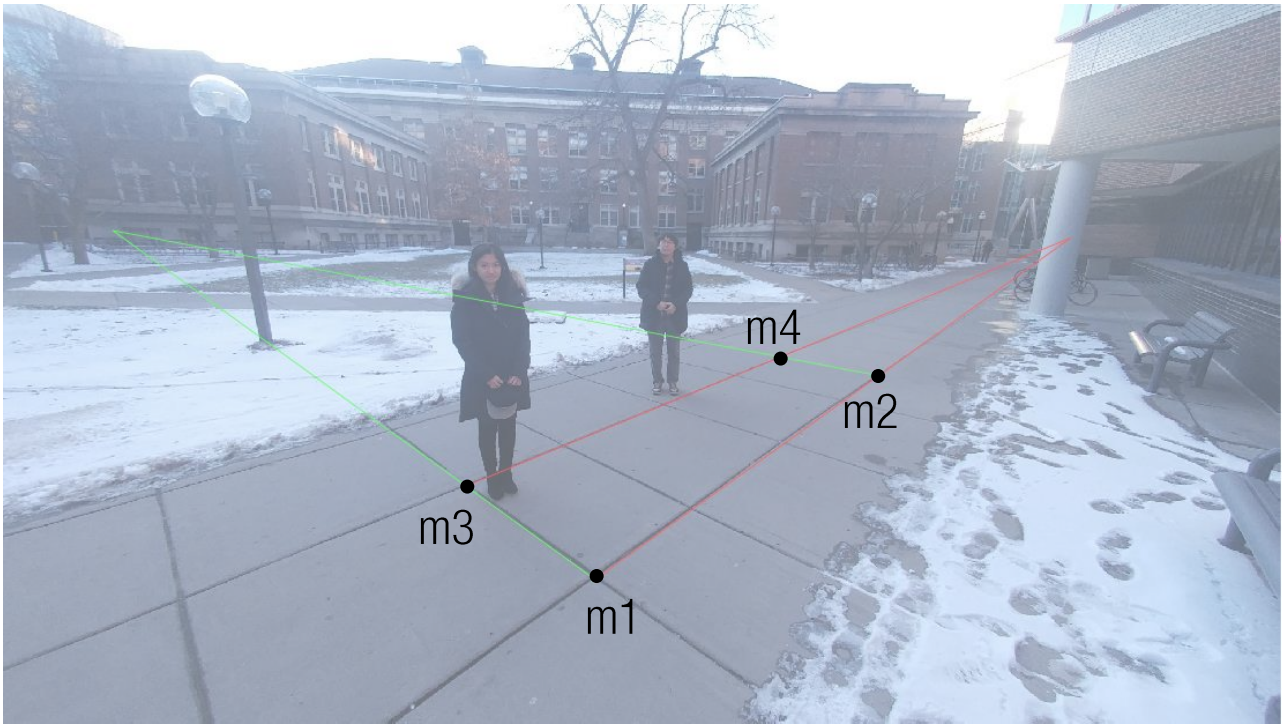
$$\frac{h_R}{h'}\frac{h'_\infty}{h_\infty} = \frac{H_R}{H}\frac{H'_\infty}{H_\infty} = \frac{H_R}{H}\frac{\infty}{\infty} = \frac{H_R}{H}$$

$$\longrightarrow H = H_R \frac{h' h_\infty}{h_R h'_\infty}$$

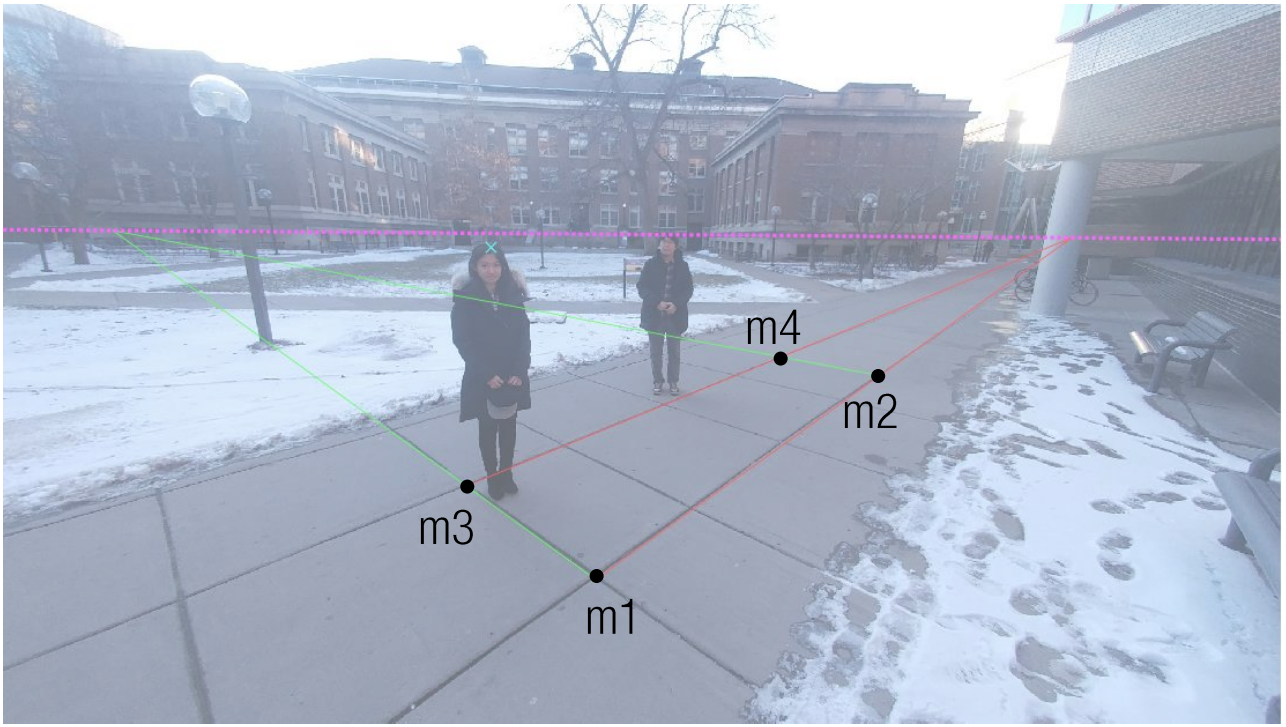

1.6m
?

sh_f = [1504;1447;1];
sh_h = [1468;730;1];
jp_f = [1997;1175;1];
jp_h = [1997;695;1];



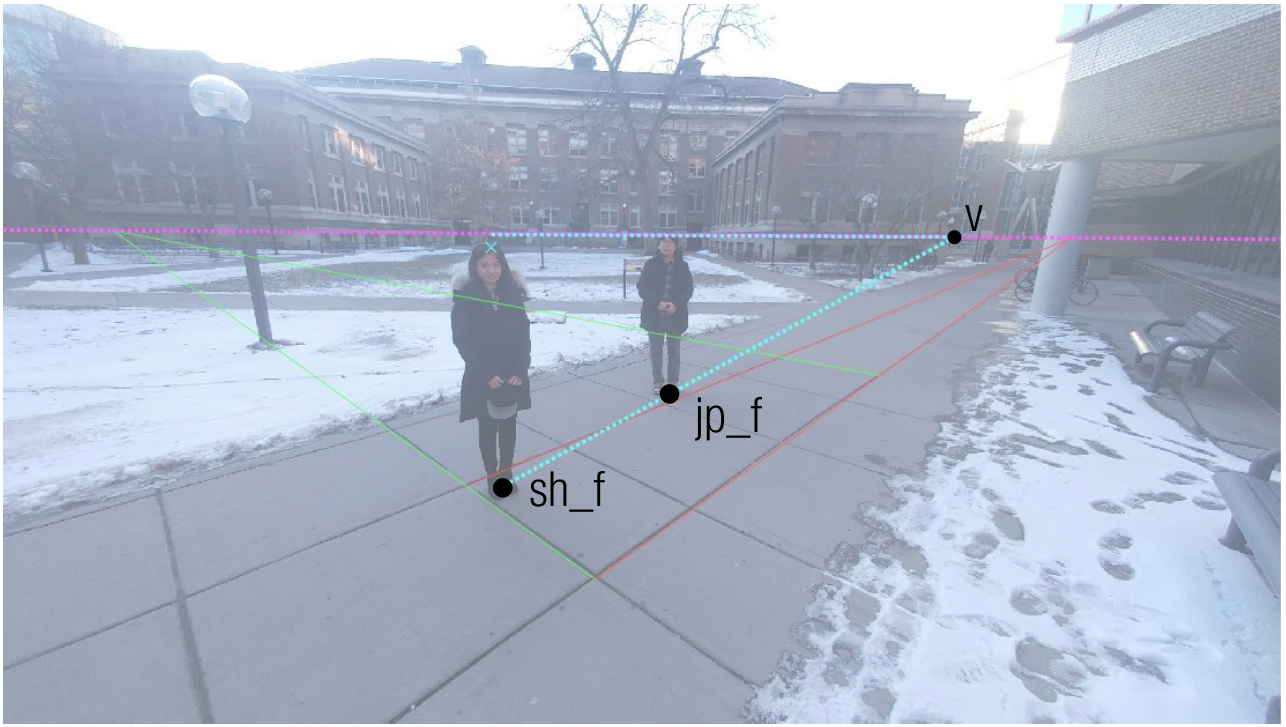



```
sh_f = [1504;1447;1];
sh_h = [1468;730;1];
jp_f = [1997;1175;1];
jp_h = [1997;695;1];

l11 = GetLineFromTwoPoints(m1,m2);
l12 = GetLineFromTwoPoints(m3,m4);
l21 = GetLineFromTwoPoints(m1,m3);
l22 = GetLineFromTwoPoints(m2,m4);
```




```
sh_f = [1504;1447;1];
sh_h = [1468;730;1];
jp_f = [1997;1175;1];
jp_h = [1997;695;1];

l11 = GetLineFromTwoPoints(m1,m2);
l12 = GetLineFromTwoPoints(m3,m4);
l21 = GetLineFromTwoPoints(m1,m3);
l22 = GetLineFromTwoPoints(m2,m4);

v1 = GetPointFromTwoLines(l11,l12);
v2 = GetPointFromTwoLines(l21,l22);
l = GetLineFromTwoPoints(v1,v2);
```

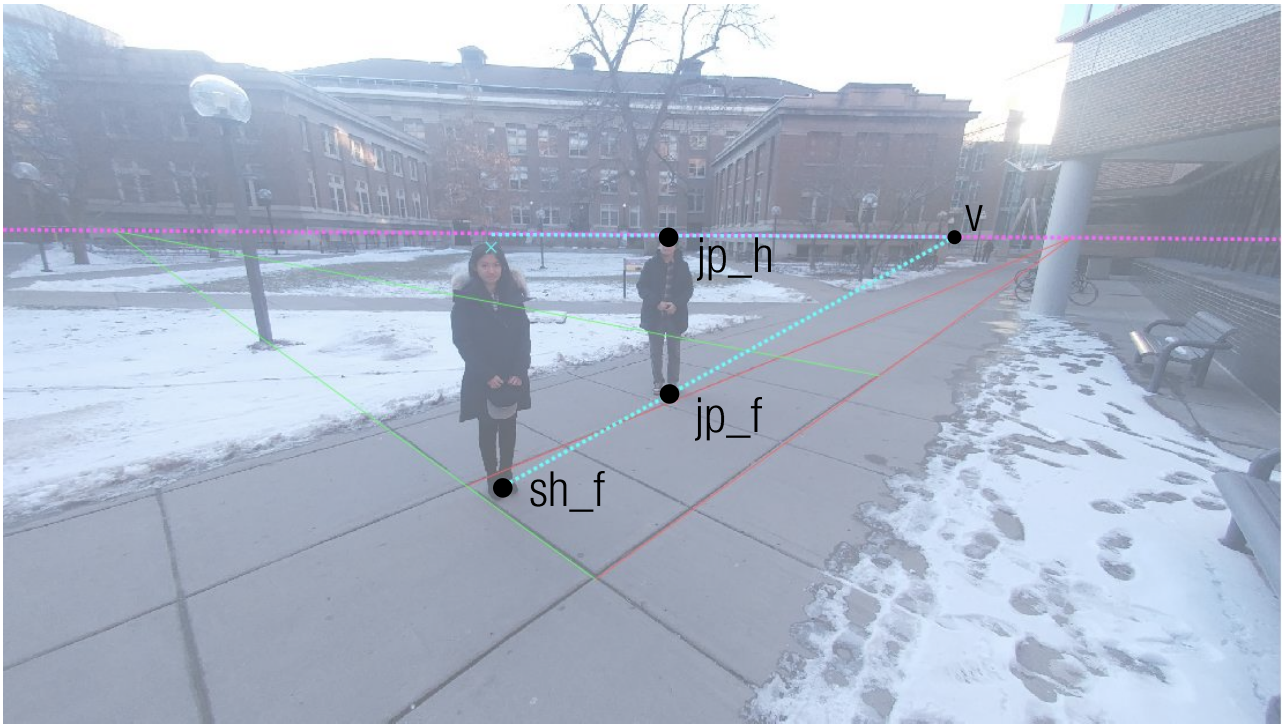
← Vanishing line




```
sh_f = [1504;1447;1];
sh_h = [1468;730;1];
jp_f = [1997;1175;1];
jp_h = [1997;695;1];

l11 = GetLineFromTwoPoints(m1,m2);
l12 = GetLineFromTwoPoints(m3,m4);
l21 = GetLineFromTwoPoints(m1,m3);
l22 = GetLineFromTwoPoints(m2,m4);

v1 = GetPointFromTwoLines(l11,l12);
v2 = GetPointFromTwoLines(l21,l22);
l = GetLineFromTwoPoints(v1,v2);
```

← Vanishing line

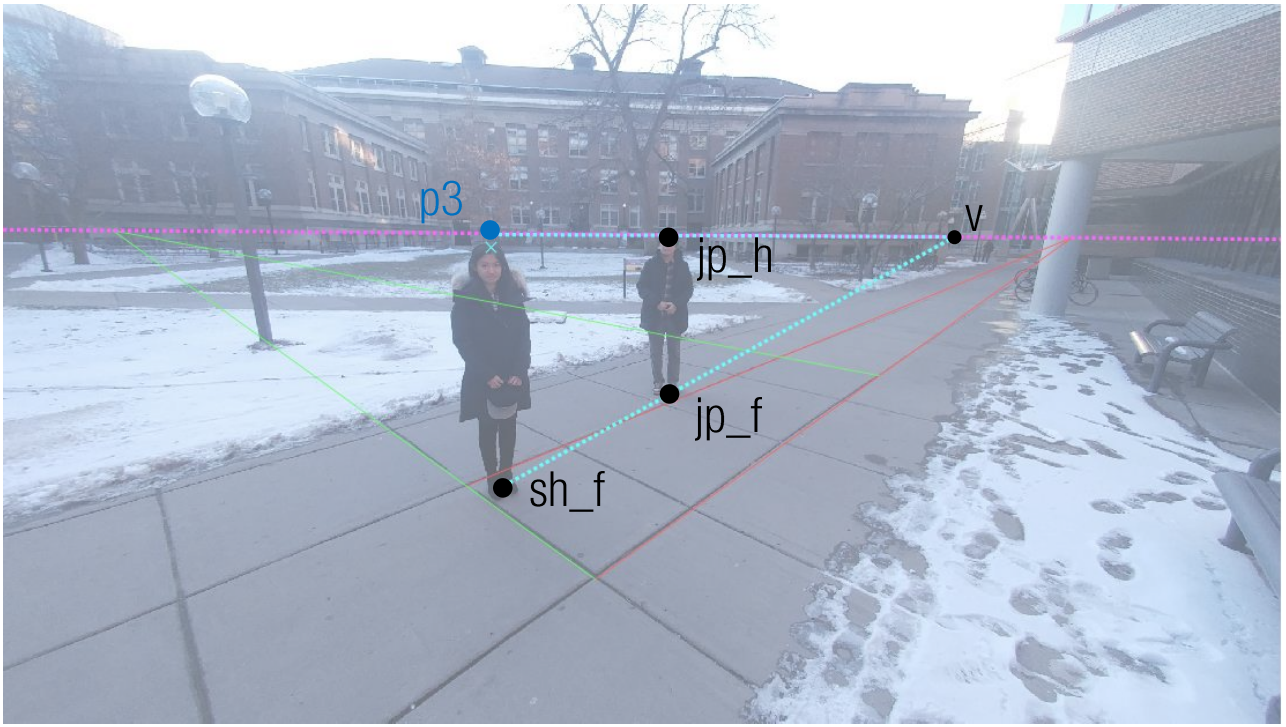
```
line_sh_jp_f = GetLineFromTwoPoints(sh_f, jp_f);
v = GetPointFromTwoLines(line_sh_jp_f, l);
```




```
sh_f = [1504;1447;1];
sh_h = [1468;730;1];
jp_f = [1997;1175;1];
jp_h = [1997;695;1];

l11 = GetLineFromTwoPoints(m1,m2);
l12 = GetLineFromTwoPoints(m3,m4);
l21 = GetLineFromTwoPoints(m1,m3);
l22 = GetLineFromTwoPoints(m2,m4);

v1 = GetPointFromTwoLines(l11,l12);
v2 = GetPointFromTwoLines(l21,l22);
l = GetLineFromTwoPoints(v1,v2);
```

← Vanishing line

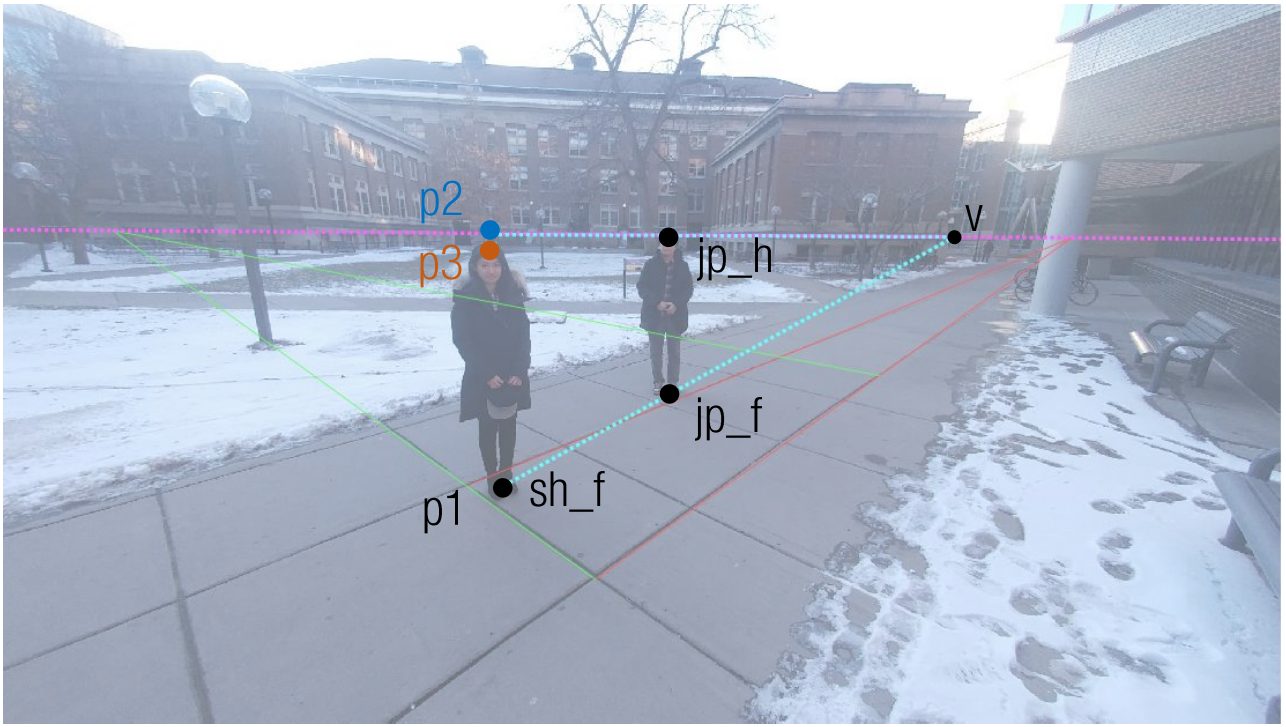
```
line_sh_jp_f = GetLineFromTwoPoints(sh_f, jp_f);
v = GetPointFromTwoLines(line_sh_jp_f, l);

line_jp_h_v = GetLineFromTwoPoints(jp_head, v);
```




```
sh_f = [1504;1447;1];
sh_h = [1468;730;1];
jp_f = [1997;1175;1];
jp_h = [1997;695;1];

l11 = GetLineFromTwoPoints(m1,m2);
l12 = GetLineFromTwoPoints(m3,m4);
l21 = GetLineFromTwoPoints(m1,m3);
l22 = GetLineFromTwoPoints(m2,m4);

v1 = GetPointFromTwoLines(l11,l12);
v2 = GetPointFromTwoLines(l21,l22);
l = GetLineFromTwoPoints(v1,v2);
```

← Vanishing line

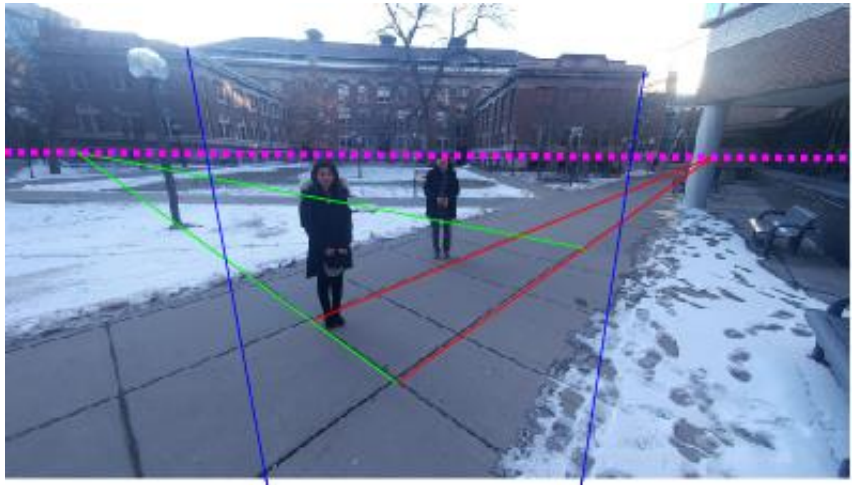
```
line_sh_jp_f = GetLineFromTwoPoints(sh_f, jp_f);
v = GetPointFromTwoLines(line_sh_jp_f, l);

line_jp_h_v = GetLineFromTwoPoints(jp_head, v);
line_sh = GetLineFromTwoPoints(sh_h, sh_f);
p2  = GetPointFromTwoLines(line_jp_head_v, line_sh);
```




```
sh_f = [1504;1447;1];
sh_h = [1468;730;1];
jp_f = [1997;1175;1];
jp_h = [1997;695;1];

l11 = GetLineFromTwoPoints(m1,m2);
l12 = GetLineFromTwoPoints(m3,m4);
l21 = GetLineFromTwoPoints(m1,m3);
l22 = GetLineFromTwoPoints(m2,m4);

v1 = GetPointFromTwoLines(l11,l12);
v2 = GetPointFromTwoLines(l21,l22);
l = GetLineFromTwoPoints(v1,v2);
```

← Vanishing line

```
line_sh_jp_f = GetLineFromTwoPoints(sh_f, jp_f);
v = GetPointFromTwoLines(line_sh_jp_f, l);

line_jp_h_v = GetLineFromTwoPoints(jp_head, v);
line_sh = GetLineFromTwoPoints(sh_h, sh_f);
p2  = GetPointFromTwoLines(line_jp_head_v, line_sh);
p3 = sh_h;
p1 = sh_f;
```

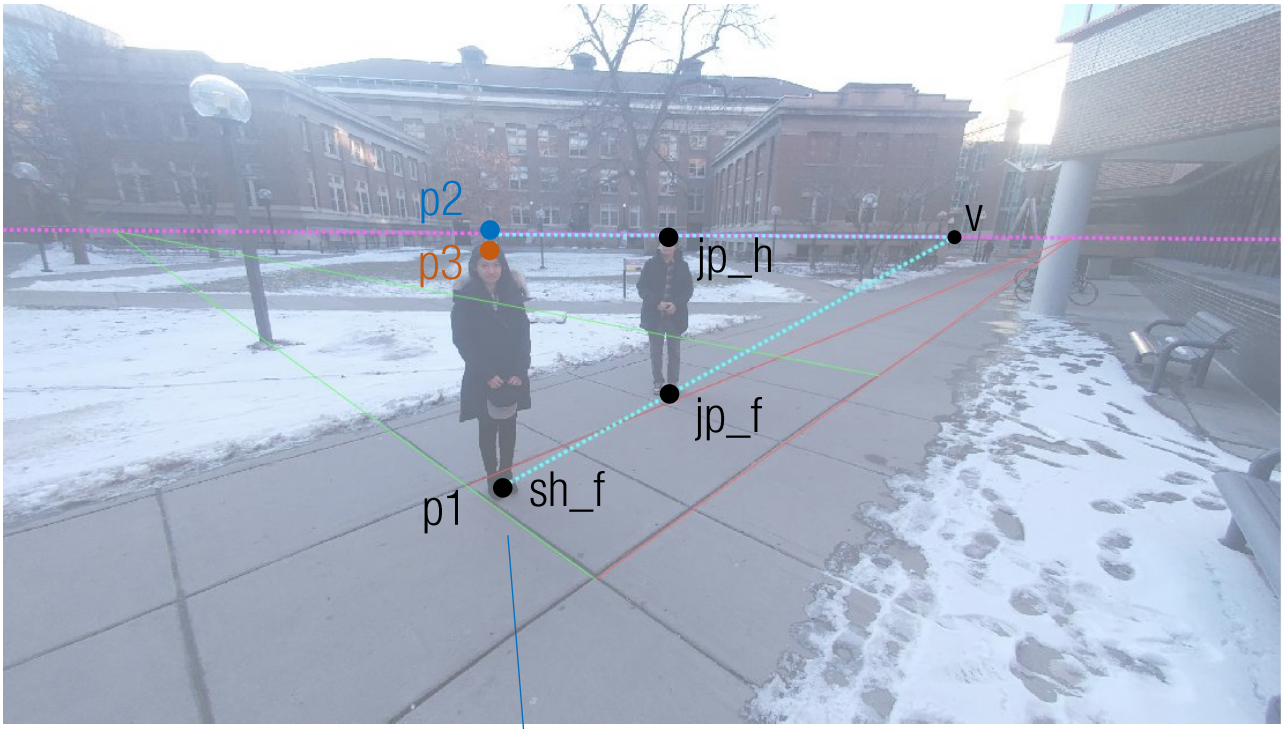


p4

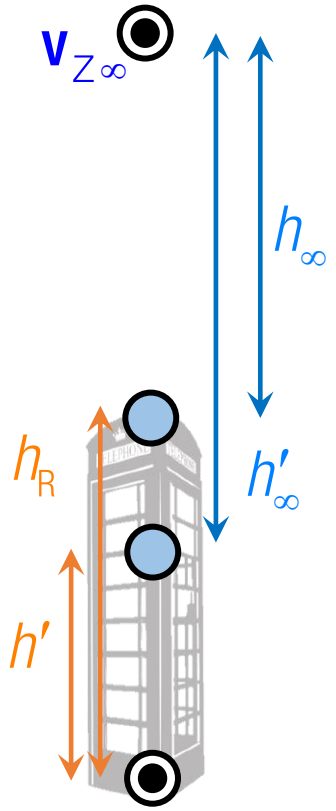
```
sh_f = [1504;1447;1];
sh_h = [1468;730;1];
jp_f = [1997;1175;1];
jp_h = [1997;695;1];

l11 = GetLineFromTwoPoints(m1,m2);
l12 = GetLineFromTwoPoints(m3,m4);
l21 = GetLineFromTwoPoints(m1,m3);
l22 = GetLineFromTwoPoints(m2,m4);

v1 = GetPointFromTwoLines(l11,l12);
v2 = GetPointFromTwoLines(l21,l22);
l = GetLineFromTwoPoints(v1,v2);

line_sh_jp_f = GetLineFromTwoPoints(sh_f, jp_f);
v = GetPointFromTwoLines(line_sh_jp_f, l);

line_jp_h_v = GetLineFromTwoPoints(jp_head, v);
line_sh = GetLineFromTwoPoints(sh_h, sh_f);
p3  = GetPointFromTwoLines(line_jp_head_v, line_sh);
p2 = sh_h;
p1 = sh_f;

l31 = GetLineFromTwoPoints(m5,m6);
l32 = GetLineFromTwoPoints(m7,m8)
v3 = GetPointFromTwoLines(l31,l32);
p4 = v3;
```

← Vanishing line

ComputeHeightFromCrossRatio.m

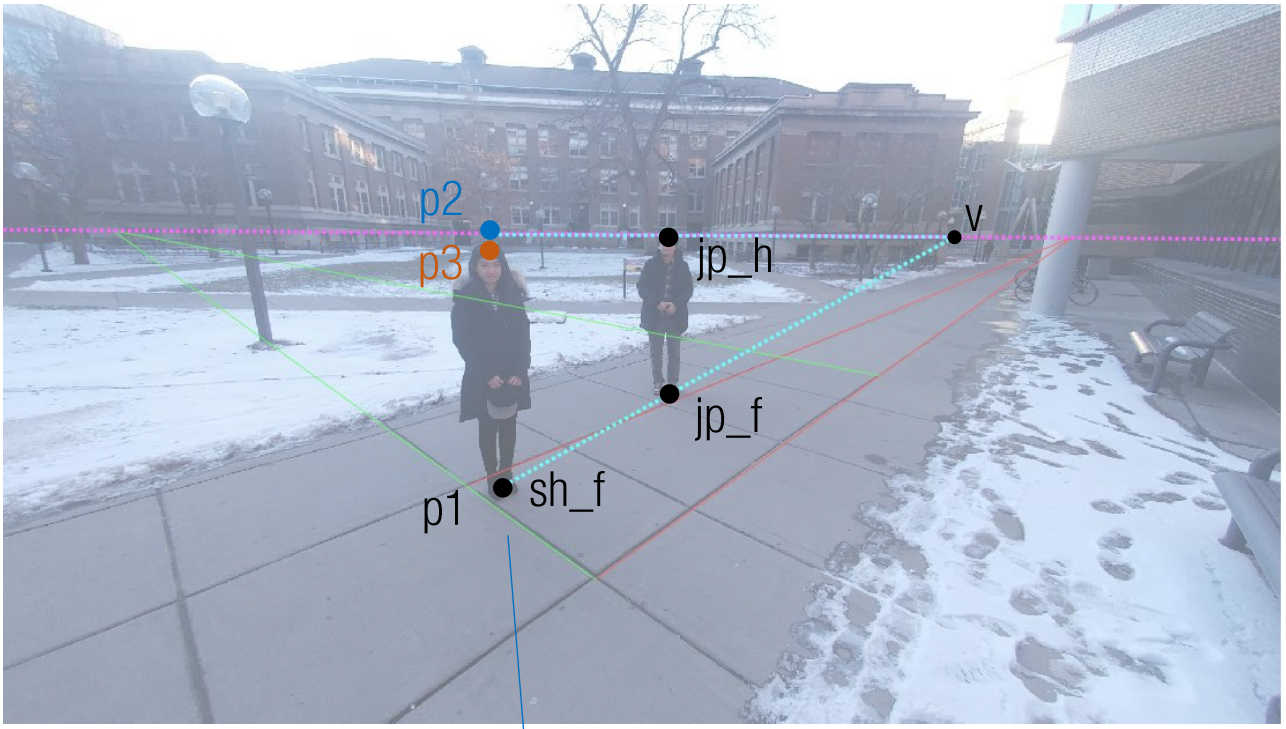



```
h_prime = norm(p1-p2);
h_R = norm(p1-p3);
h_prime_inf = norm(p4-p2);
h_inf = norm(p4-p3);
```



$$H = H_R \frac{h' h_\infty}{h_R h'_\infty}$$

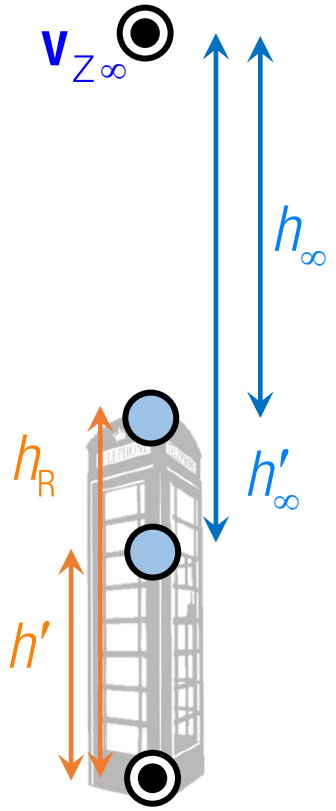
ComputeHeightFromCrossRatio.m




```
h_prime = norm(p1-p2);
h_R = norm(p1-p3);
h_prime_inf = norm(p4-p2);
h_inf = norm(p4-p3);

H = H_R * h_prime * h_prime_inf / h_R / h_inf
```

H =
1.6779

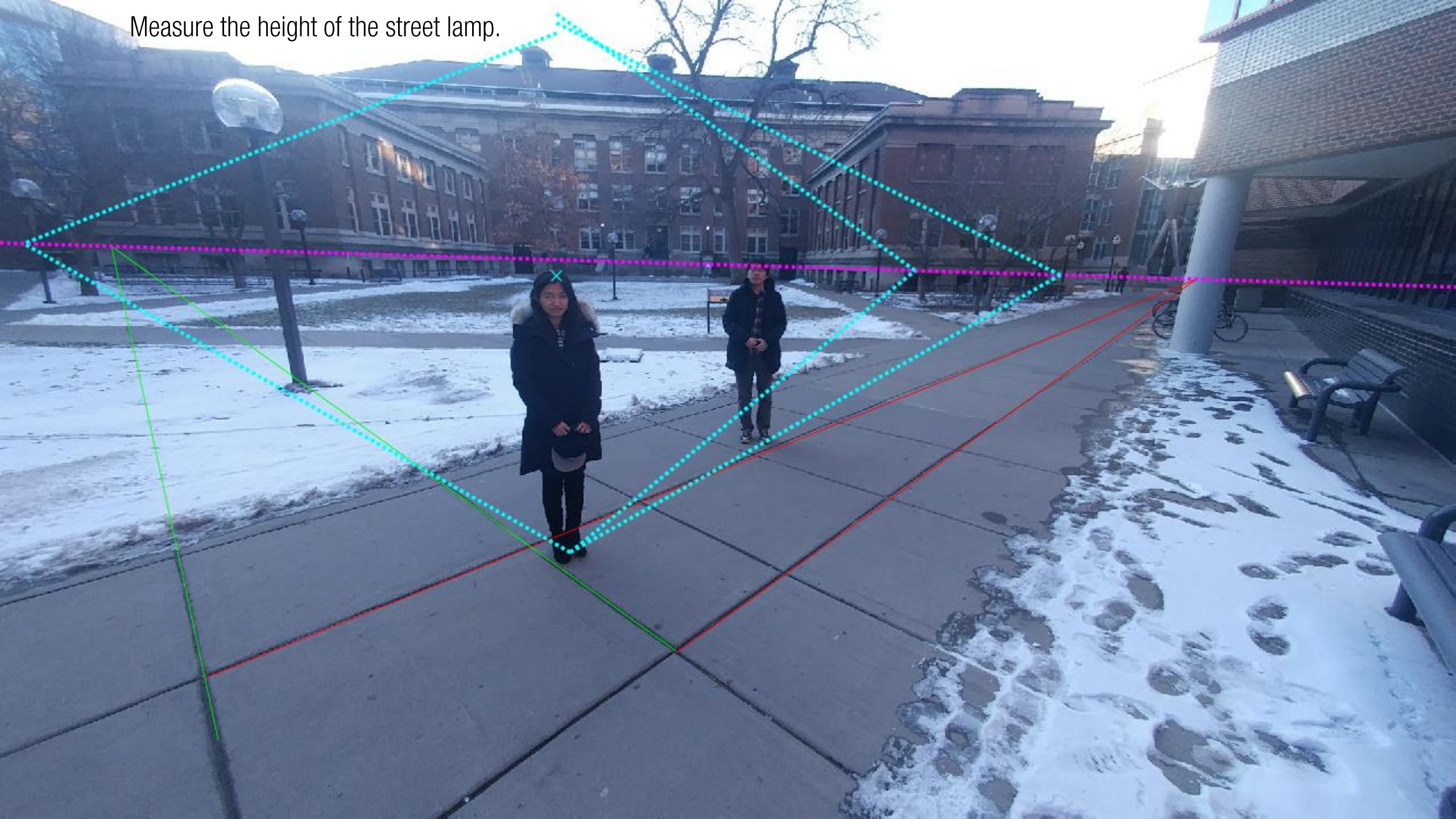
Ground truth: 1.7m



$$H = H_R \frac{h' h_\infty}{h_R h'_\infty}$$

ComputeHeightFromCrossRatio.m

Measure the height of the street lamp.



Measure the height of the street lamp.

H = 2.8108m