3-d Reconstruction from a Single View

David Capel
CSE 597F

(with many slides from talks by Aloysha Efros and Steve Seitz)

Breaking into the third dimension

Virtual tourism with 2-d mosaics

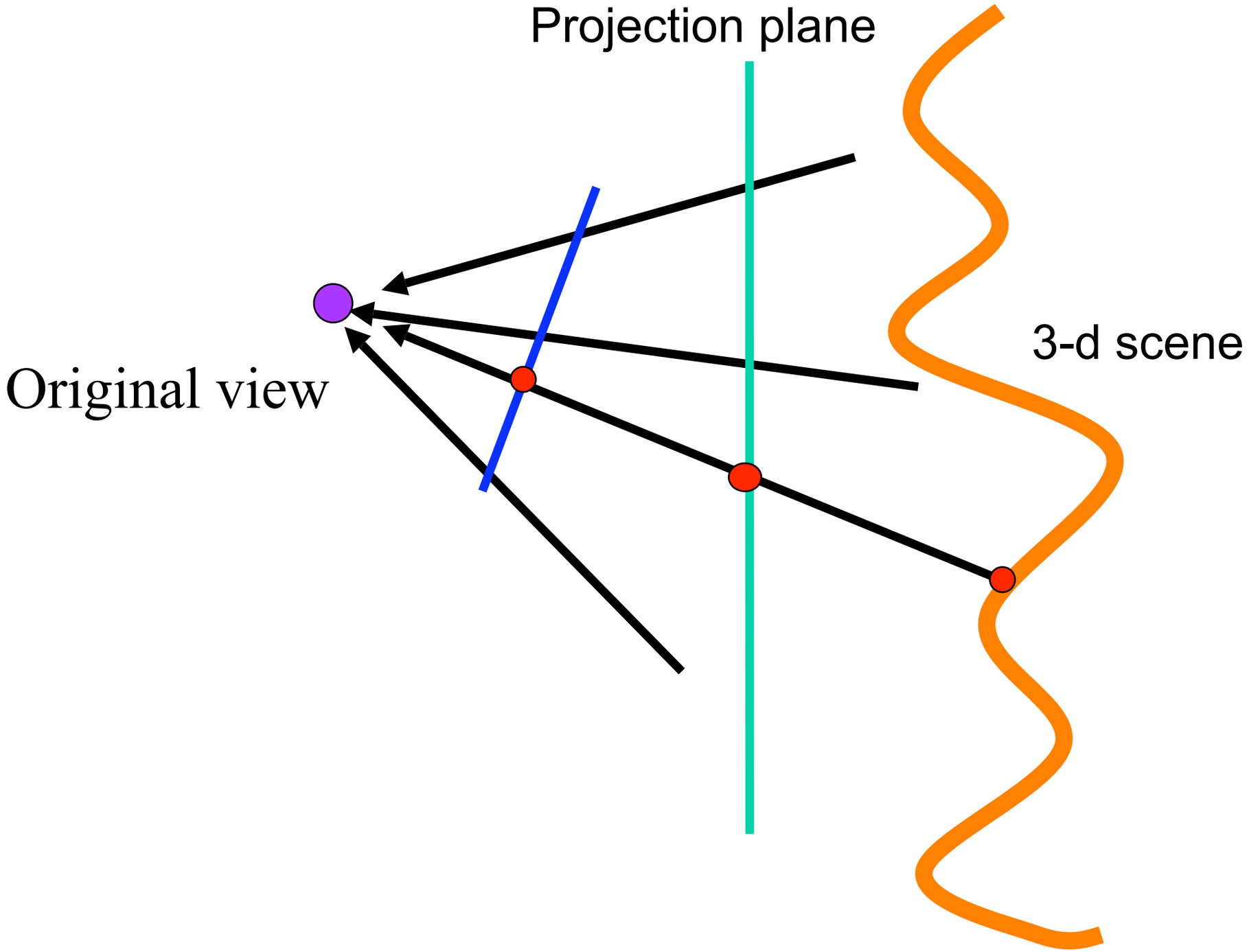
e.g. Quicktime VR, Google StreetView

- Powerful and easy to create
- But 3-d parallax is lost
- Virtual viewpoint can't *move*, only rotate/zoom

How can we visit the missing third dimension?

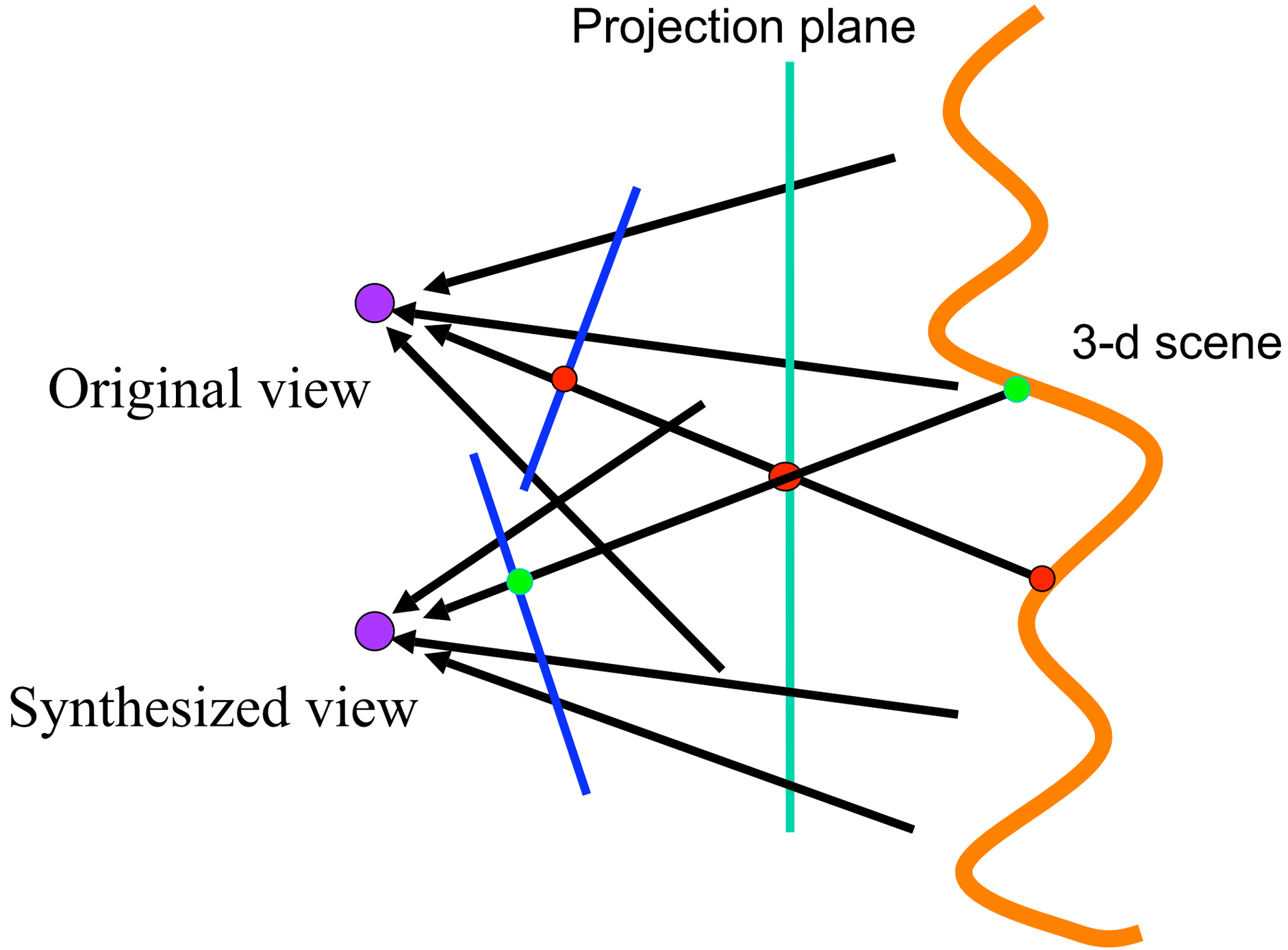
Can we use a homography to change viewpoint?

- For general 3-d scenes, no.



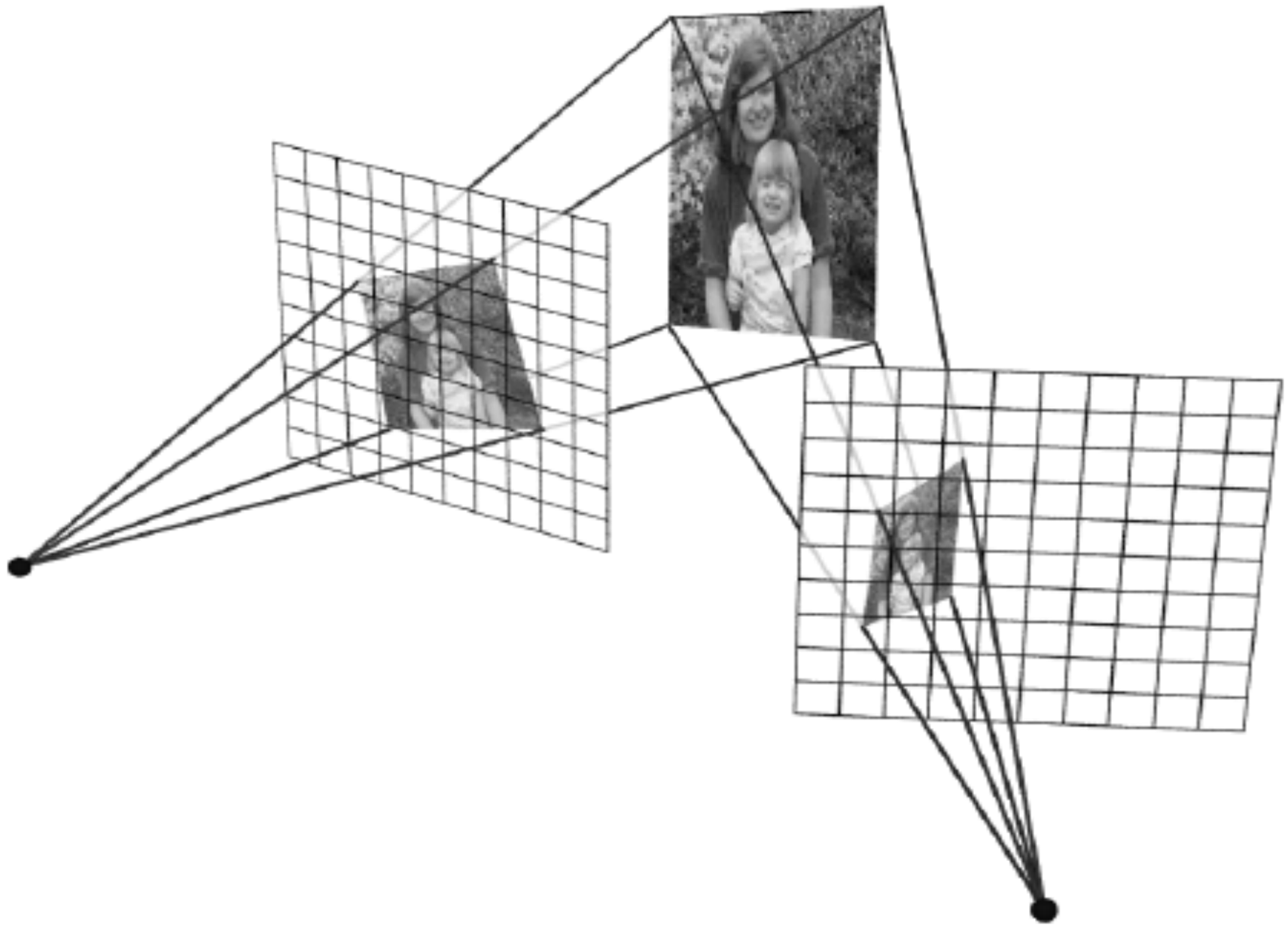
Can we use a homography to change viewpoint?

- For general 3-d scenes, no.



Can we use a homography to change viewpoint?

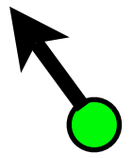
- Yes - if the scene is flat



Can we use a homography to change viewpoint?

Original view

Can we use a homography to change viewpoint?

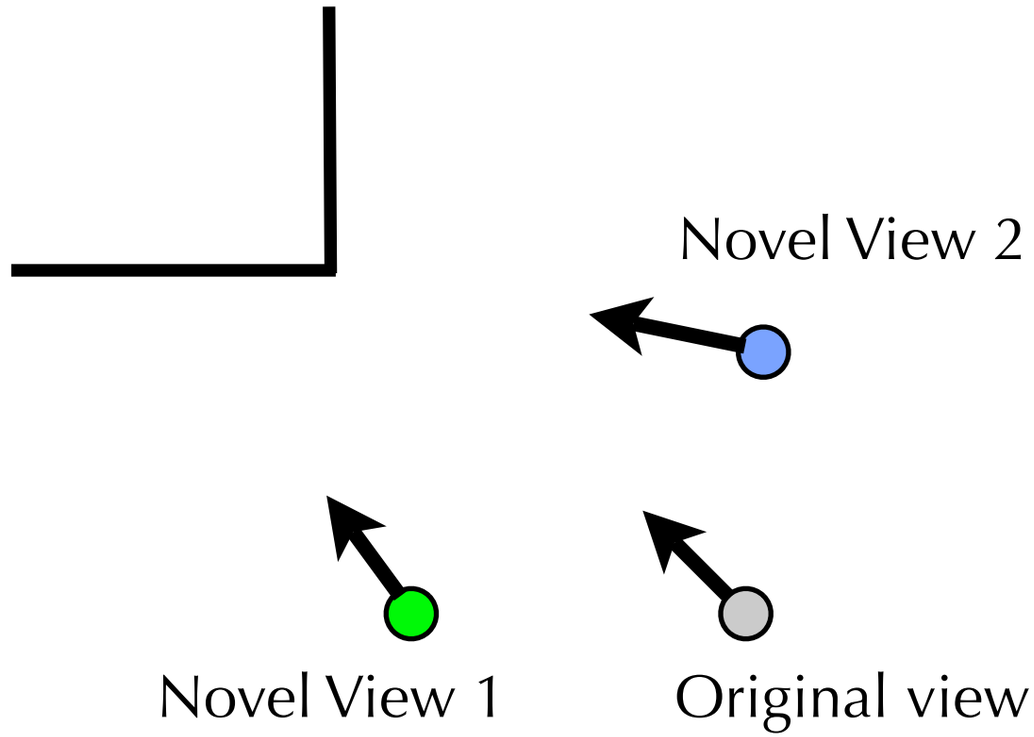


Novel View 1

Original view

Novel View 1

Can we use a homography to change viewpoint?



Novel View 1

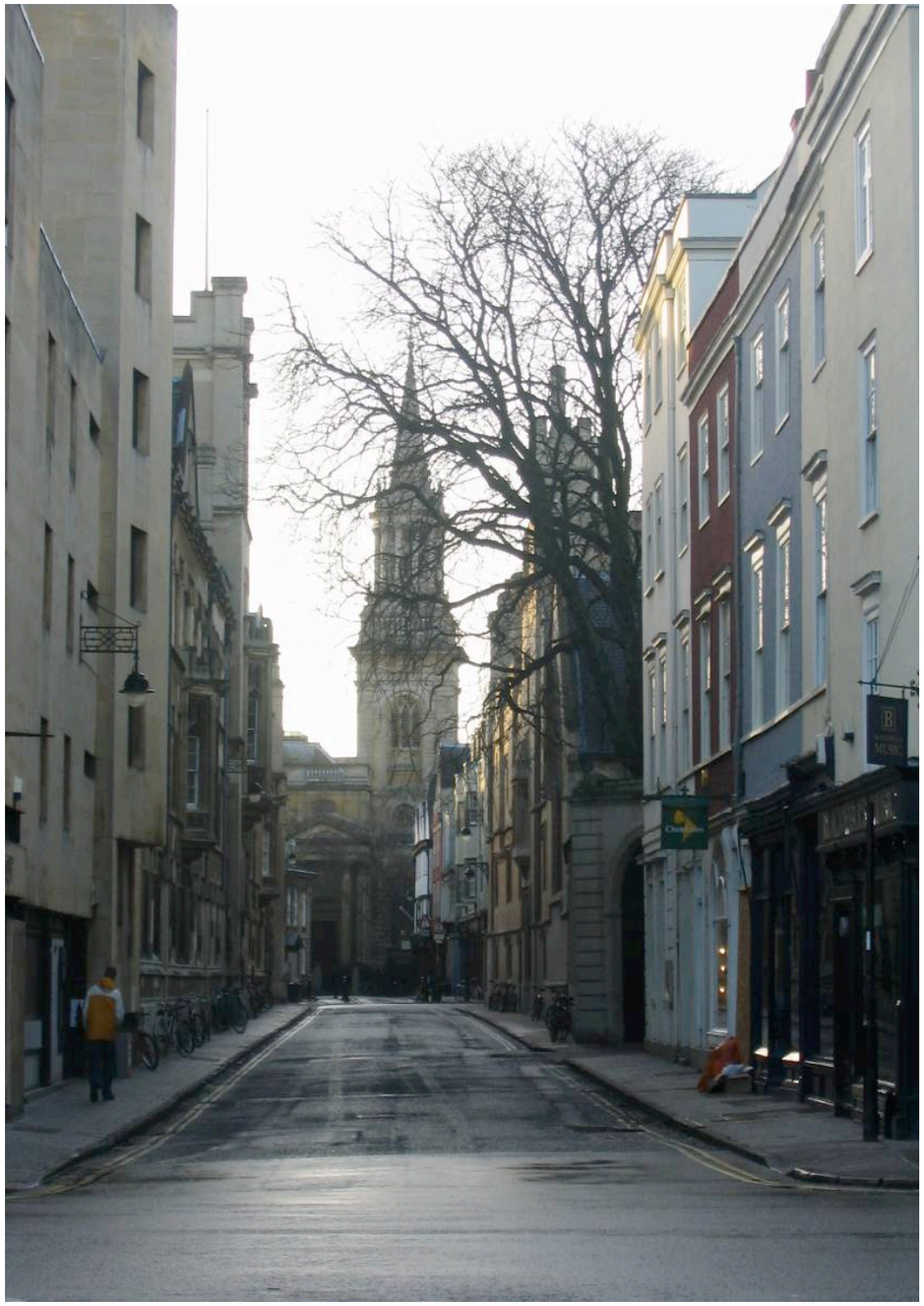
Novel View 2

Breaking into the third dimension

Idea

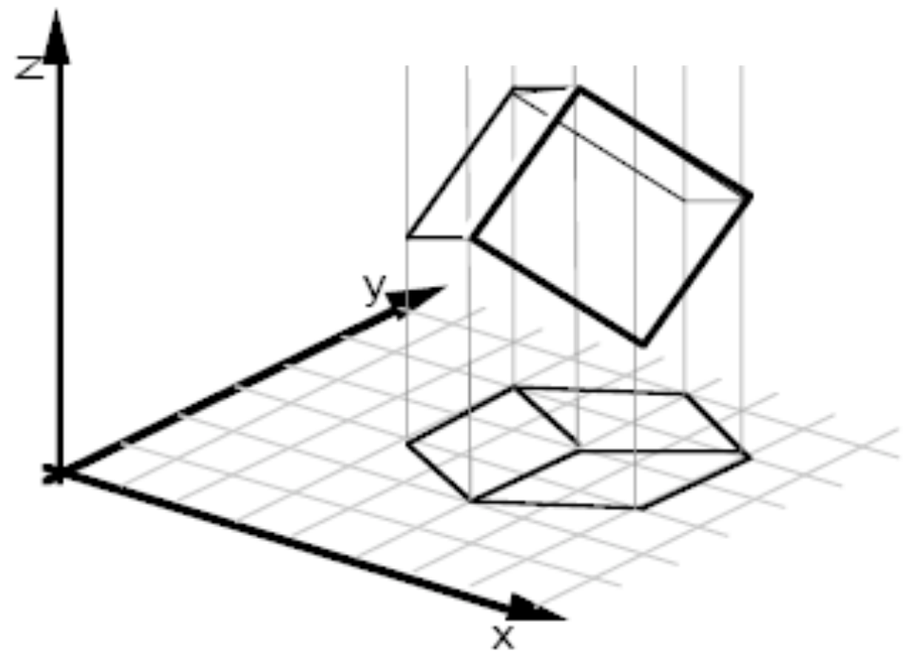
-- Model the scene as a set of planes

-- Now we just need to find the position and orientation of those planes



The Problem

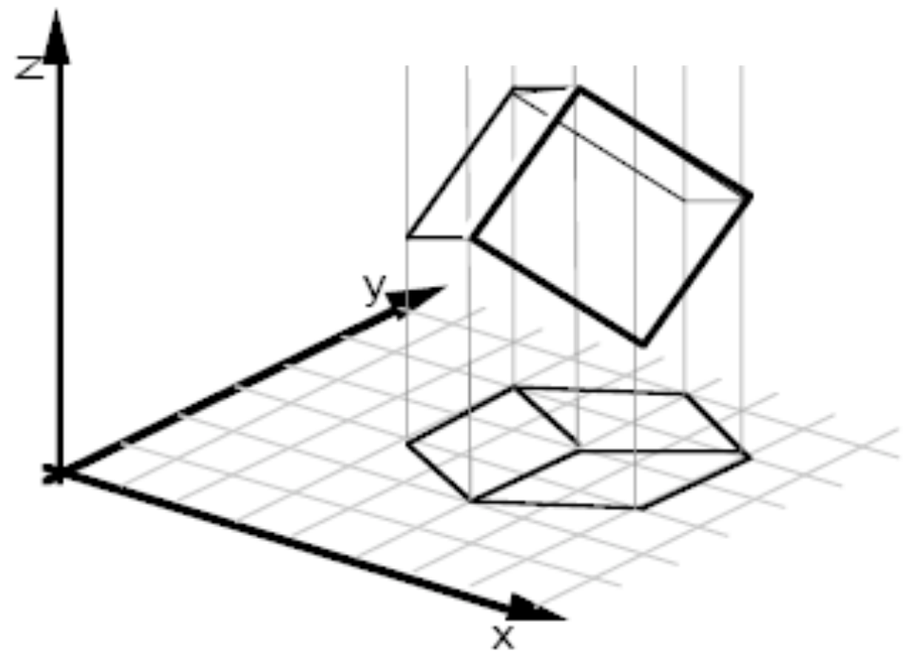
- Recovering 3D geometry from **single** 2D projection



from [Sinha and Adelson 1993]

The Problem

- Recovering 3D geometry from **single** 2D projection



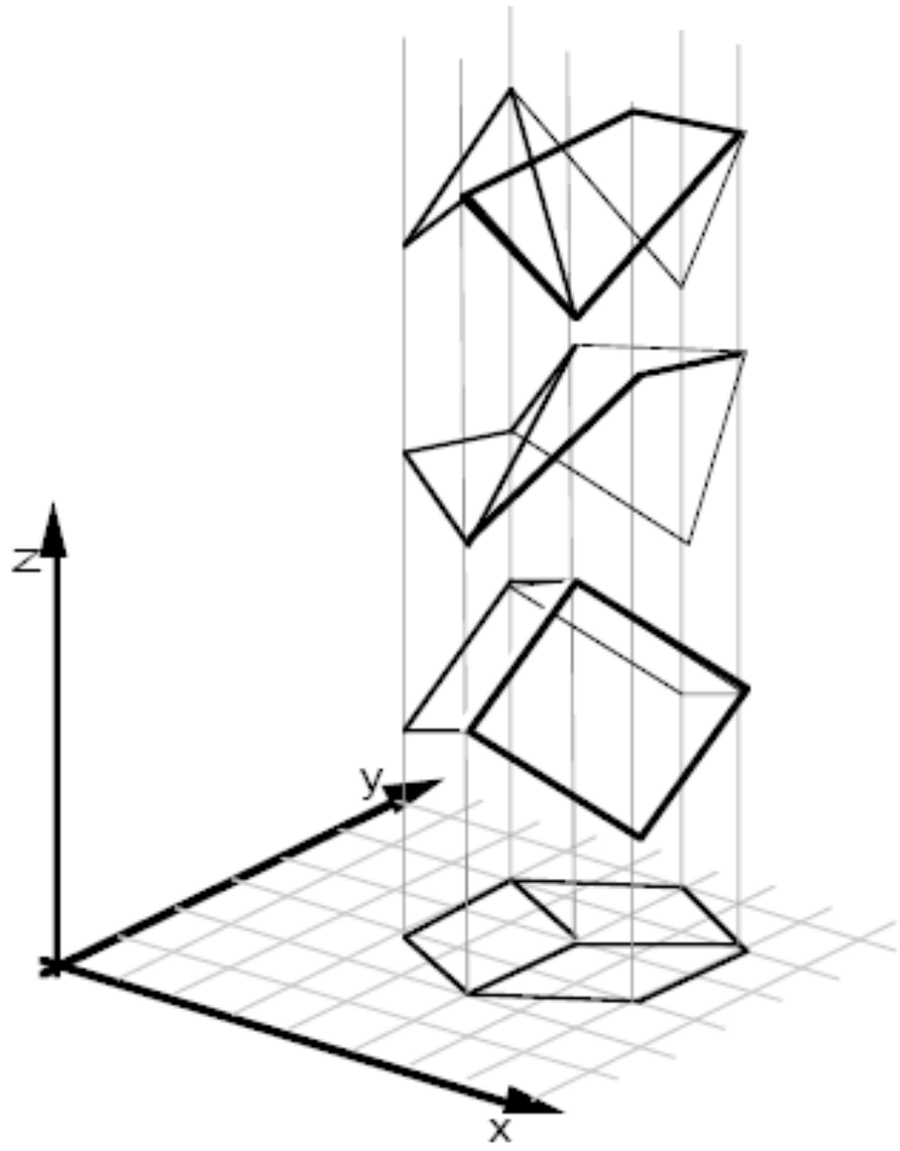
from [Sinha and Adelson 1993]

The Problem

- Recovering 3D geometry from **single** 2D projection

- Infinite number of possible solutions!

- How do we choose the right one?

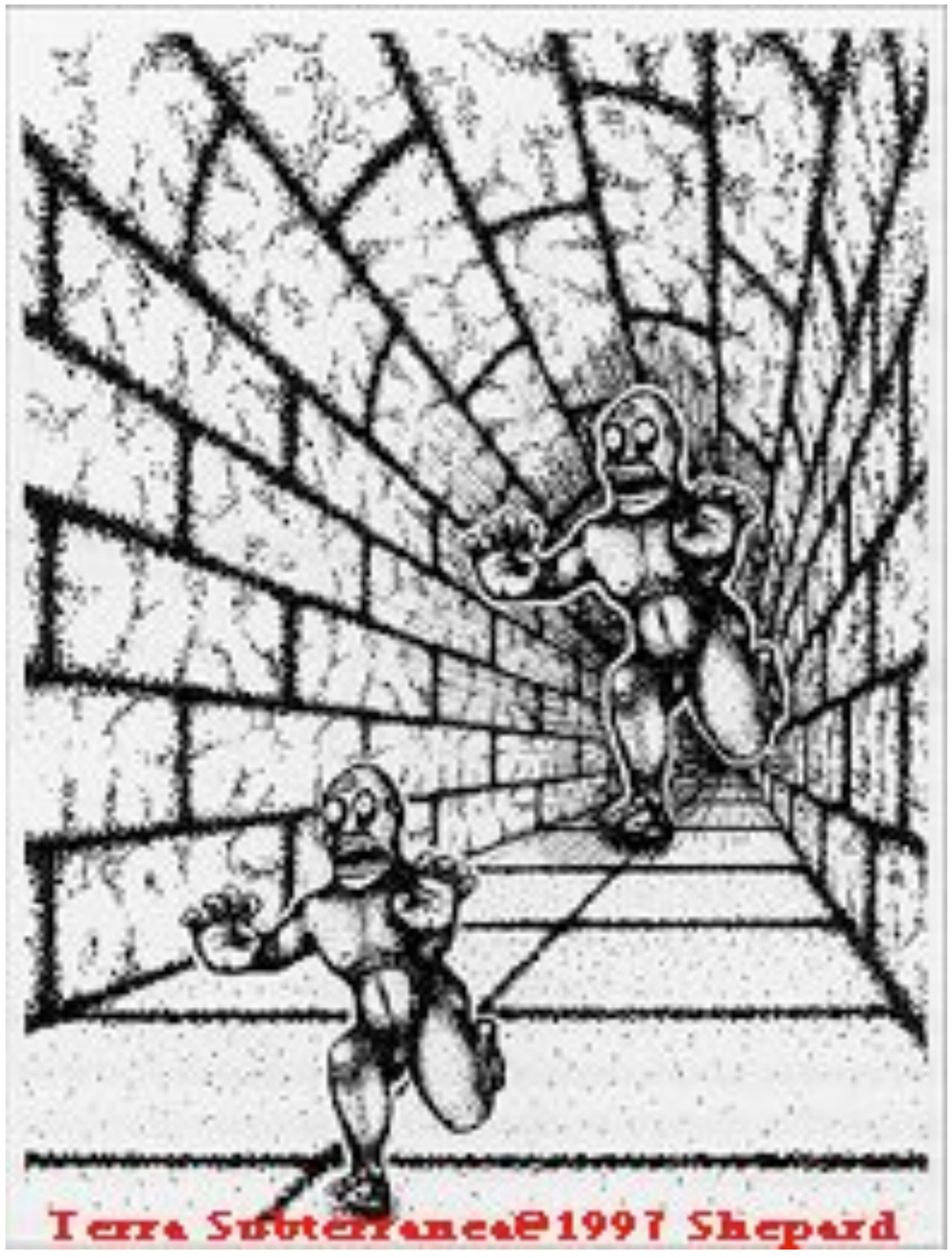


from [Sinha and Adelson 1993]

Parallel Lines?

Perspective cues

Perspective cues : identical monsters?

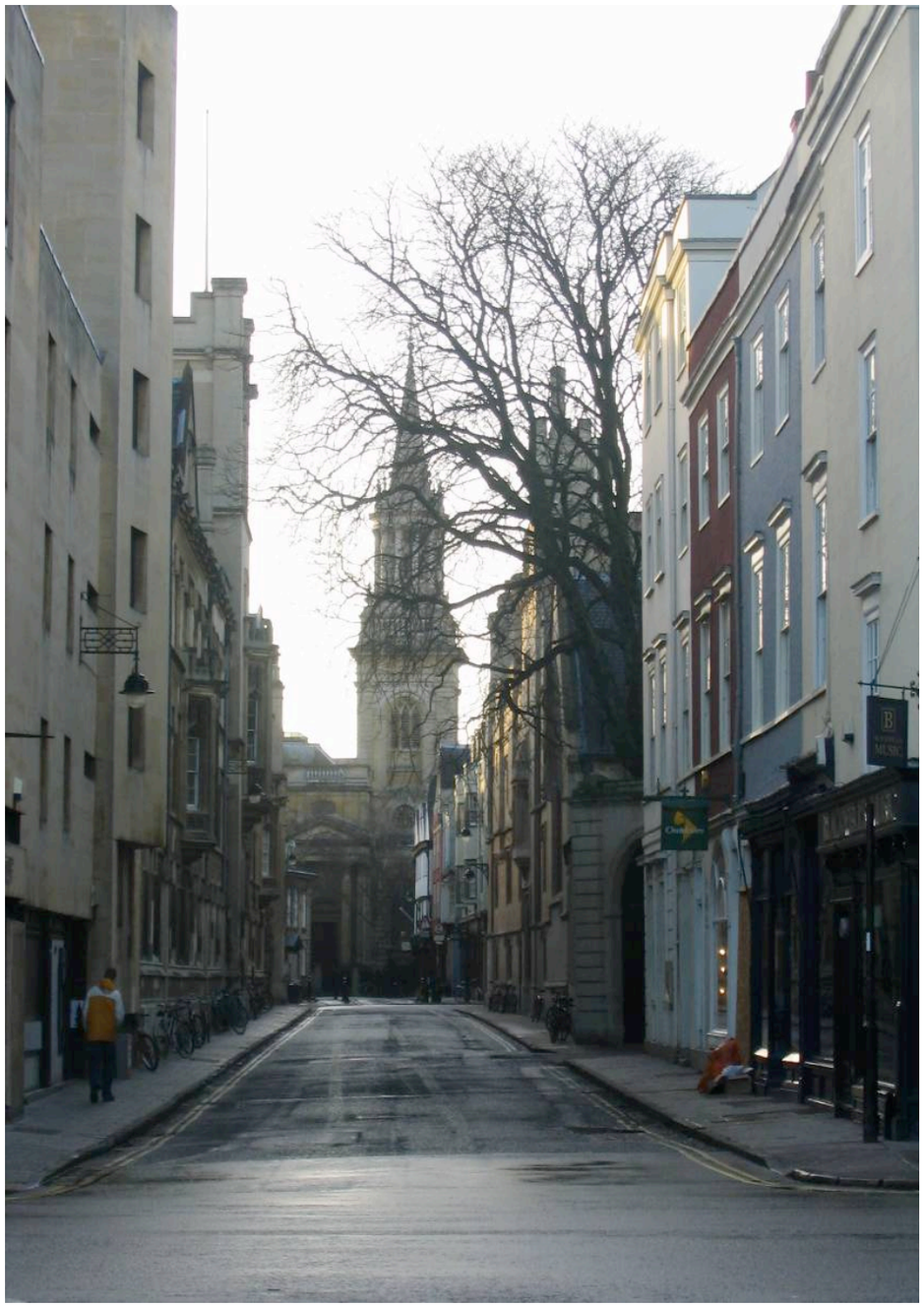

Terra Subterranea©1997 Shepard

Breaking into the third dimension

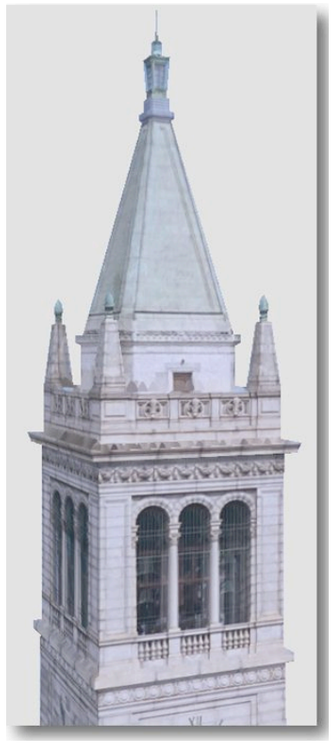
Idea

-- Model the scene as a set of planes

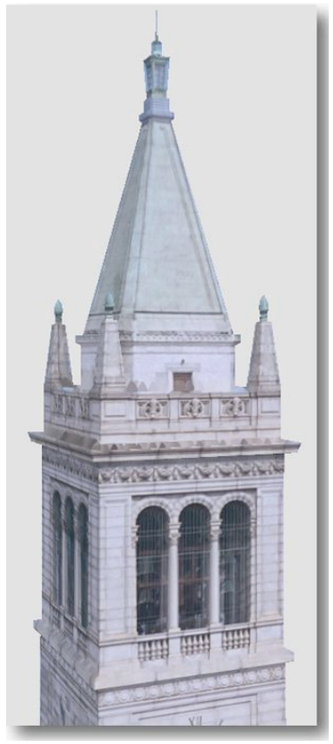
-- Use perspective to help find position and orientation of planes



Related Work

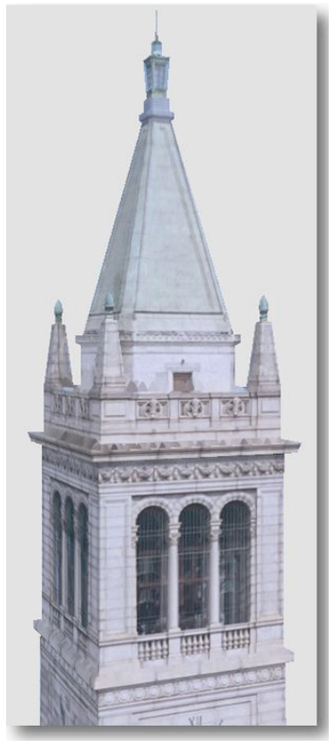


Related Work



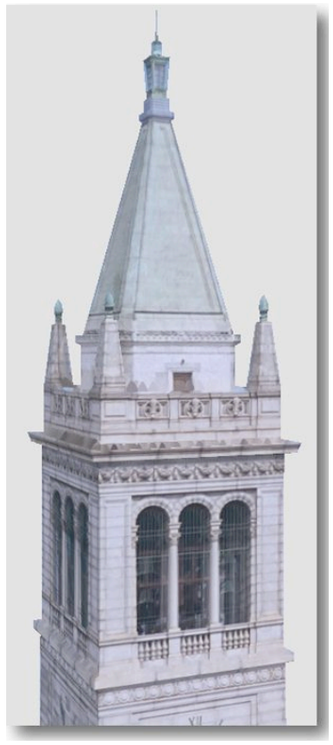
- Multiple Images
  - Manual Reconstruction
    - Façade [Debevec et al. 1996], REALVIZ Image Modeler, Photobuilder [Cipolla et al. 1999], [Zieglier et al. 2003], etc.
  - Automatic Reconstruction

Related Work



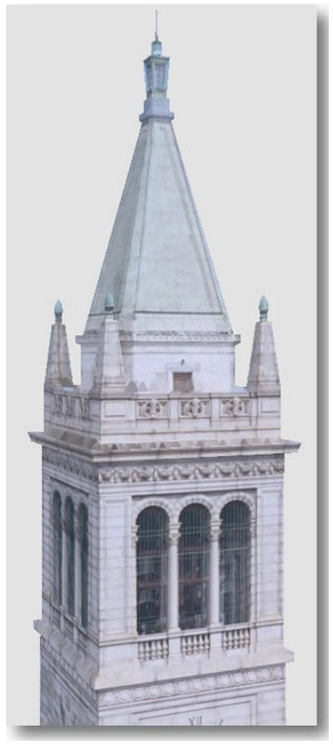
- Multiple Images
  - Manual Reconstruction
    - Façade [Debevec et al. 1996], REALVIZ Image Modeler, Photobuilder [Cipolla et al. 1999], [Zieglier et al. 2003], etc.
  - Automatic Reconstruction
    - Structure from Motion (e.g. [Nister 2001], [Pollefeys et al. 2004])

Related Work



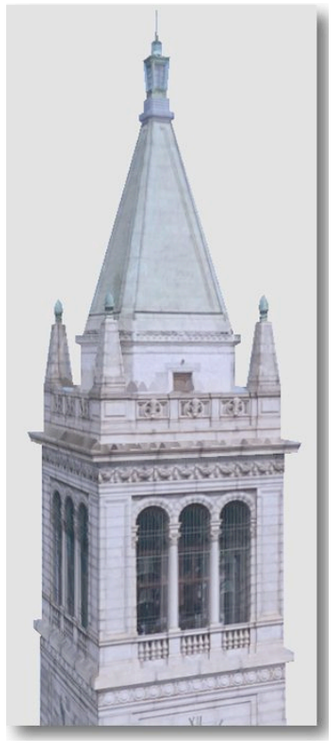
- Multiple Images
  - Manual Reconstruction
    - Façade [Debevec et al. 1996], REALVIZ Image Modeler, Photobuilder [Cipolla et al. 1999], [Zieglier et al. 2003], etc.
  - Automatic Reconstruction
    - Structure from Motion (e.g. [Nister 2001], [Pollefeys et al. 2004])
- Single Image

Related Work



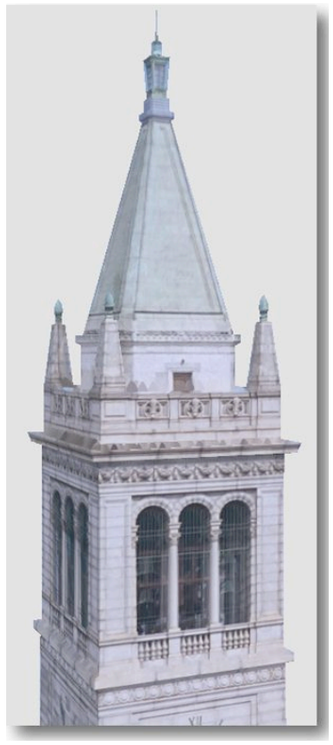
- Multiple Images
  - Manual Reconstruction
    - Façade [Debevec et al. 1996], REALVIZ Image Modeler, Photobuilder [Cipolla et al. 1999], [Zieglier et al. 2003], etc.
  - Automatic Reconstruction
    - Structure from Motion (e.g. [Nister 2001], [Pollefeys et al. 2004])
- Single Image
  - Manual Reconstruction

Related Work



- Multiple Images
  - Manual Reconstruction
    - Façade [Debevec et al. 1996], REALVIZ Image Modeler, Photobuilder [Cipolla et al. 1999], [Zieglier et al. 2003], etc.
  - Automatic Reconstruction
    - Structure from Motion (e.g. [Nister 2001], [Pollefeys et al. 2004])
- Single Image
  - Manual Reconstruction
    - Metric ([Liebowitz et al. 1999], [Criminisi et al. 1999])

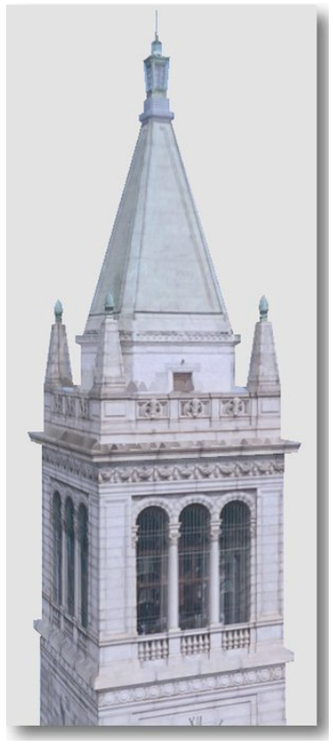
Related Work



- Multiple Images
  - Manual Reconstruction
    - Façade [Debevec et al. 1996], REALVIZ Image Modeler, Photobuilder [Cipolla et al. 1999], [Zieglier et al. 2003], etc.
  - Automatic Reconstruction
    - Structure from Motion (e.g. [Nister 2001], [Pollefeys et al. 2004])

- Single Image
  - Manual Reconstruction
    - Metric ([Liebowitz et al. 1999], [Criminisi et al. 1999])
    - Approximate: *Tour into the Picture* [Horry et al. 1997];

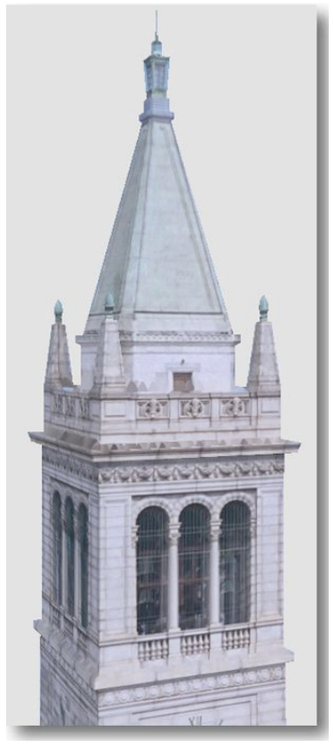
Related Work



- Multiple Images
  - Manual Reconstruction
    - Façade [Debevec et al. 1996], REALVIZ Image Modeler, Photobuilder [Cipolla et al. 1999], [Zieglier et al. 2003], etc.
  - Automatic Reconstruction
    - Structure from Motion (e.g. [Nister 2001], [Pollefeys et al. 2004])

- Single Image
  - Manual Reconstruction
    - Metric ([Liebowitz et al. 1999], [Criminisi et al. 1999])
    - Approximate: *Tour into the Picture* [Horry et al. 1997];
  - Automatic Reconstruction

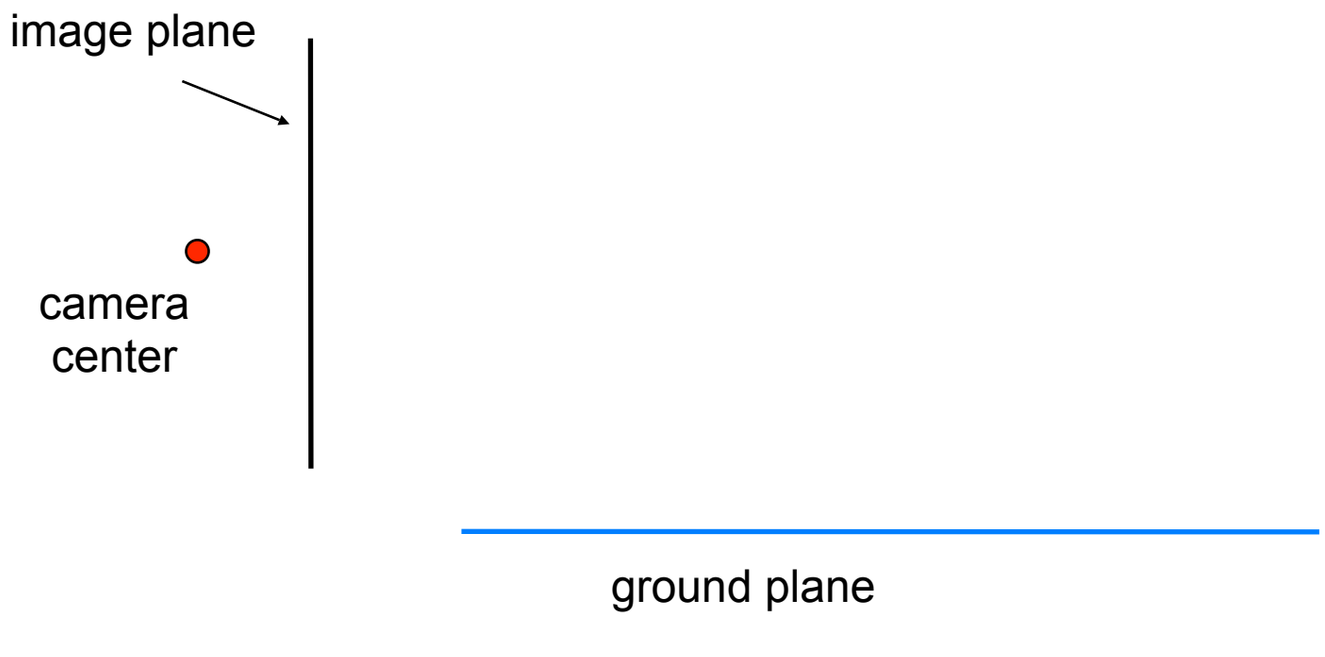
Related Work



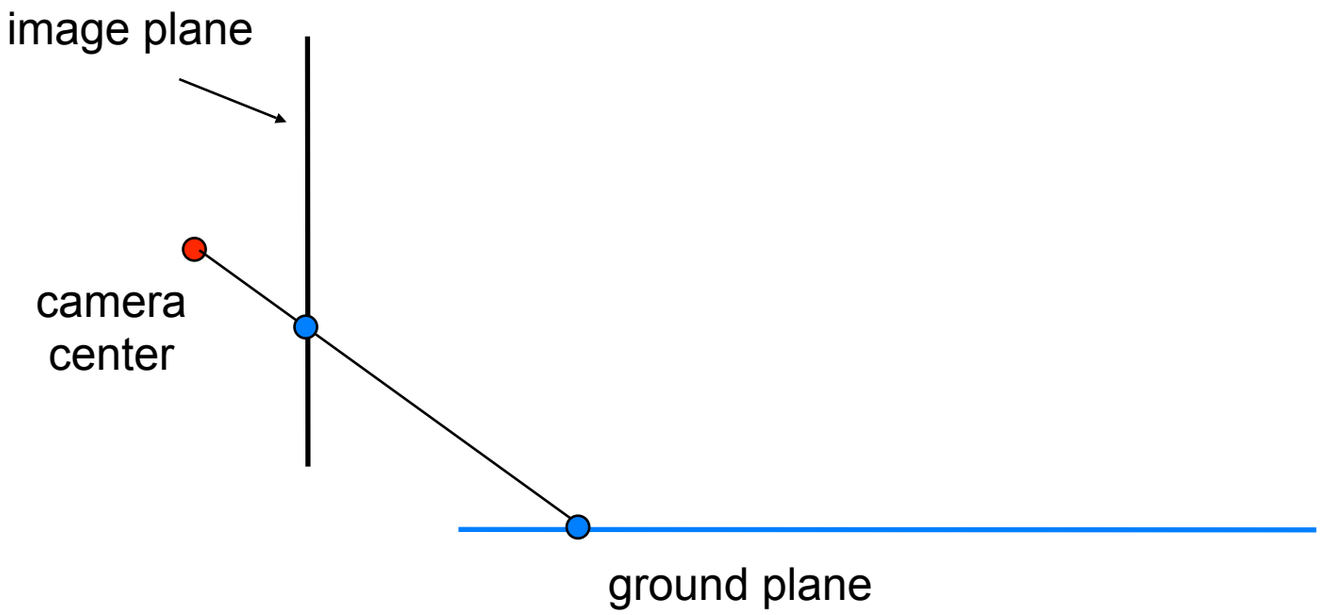
- Multiple Images
  - Manual Reconstruction
    - Façade [Debevec et al. 1996], REALVIZ Image Modeler, Photobuilder [Cipolla et al. 1999], [Zieglier et al. 2003], etc.
  - Automatic Reconstruction
    - Structure from Motion (e.g. [Nister 2001], [Pollefeys et al. 2004])

- Single Image
  - Manual Reconstruction
    - Metric ([Liebowitz et al. 1999], [Criminisi et al. 1999])
    - Approximate: *Tour into the Picture* [Horry et al. 1997];
  - Automatic Reconstruction
    - *Automatic Photo Pop-Up* [Hoiem et al. 2005]

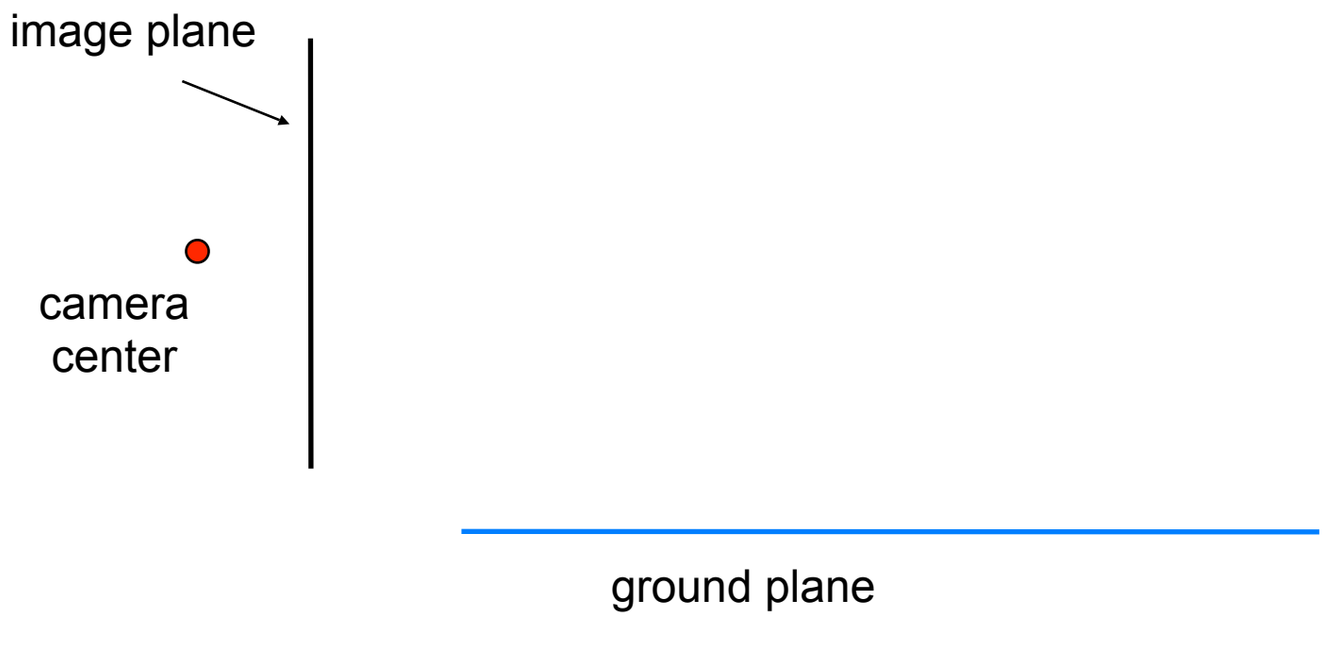
Vanishing points



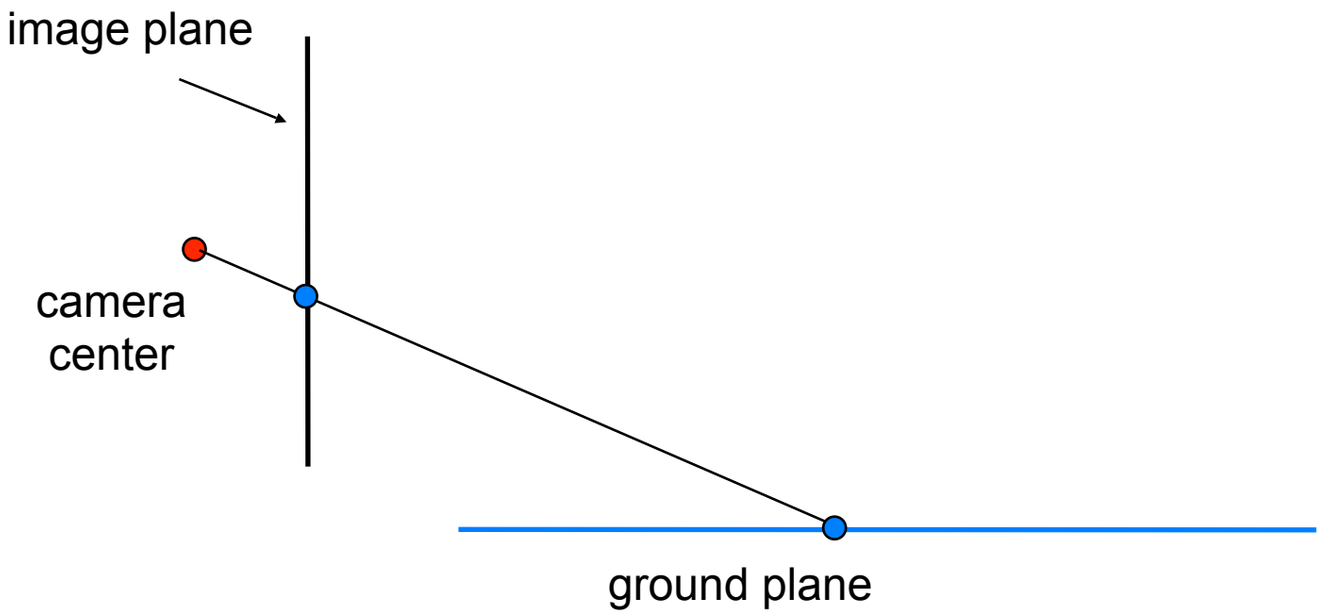
Vanishing points



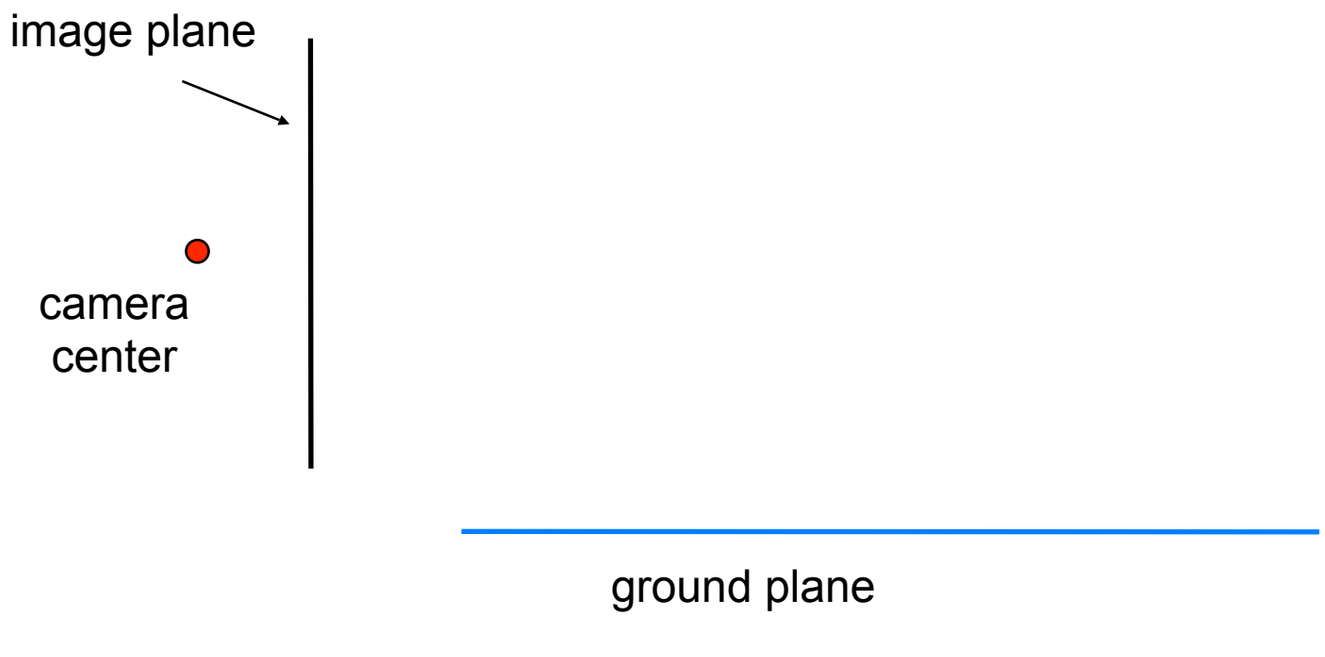
Vanishing points



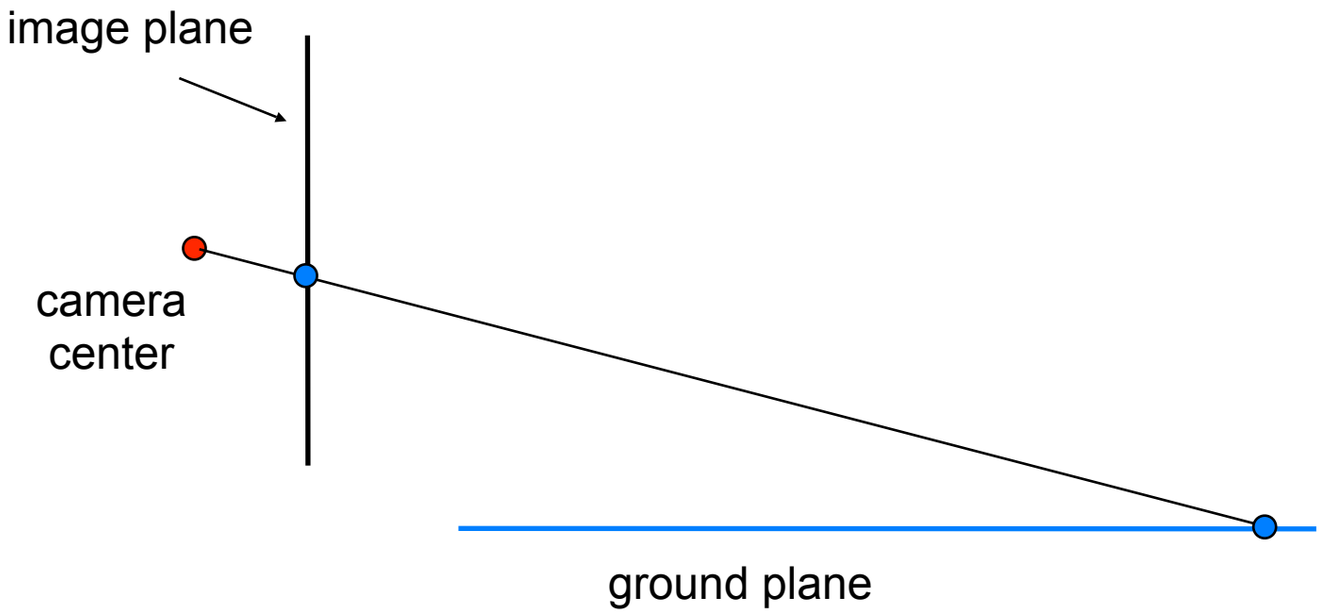
Vanishing points



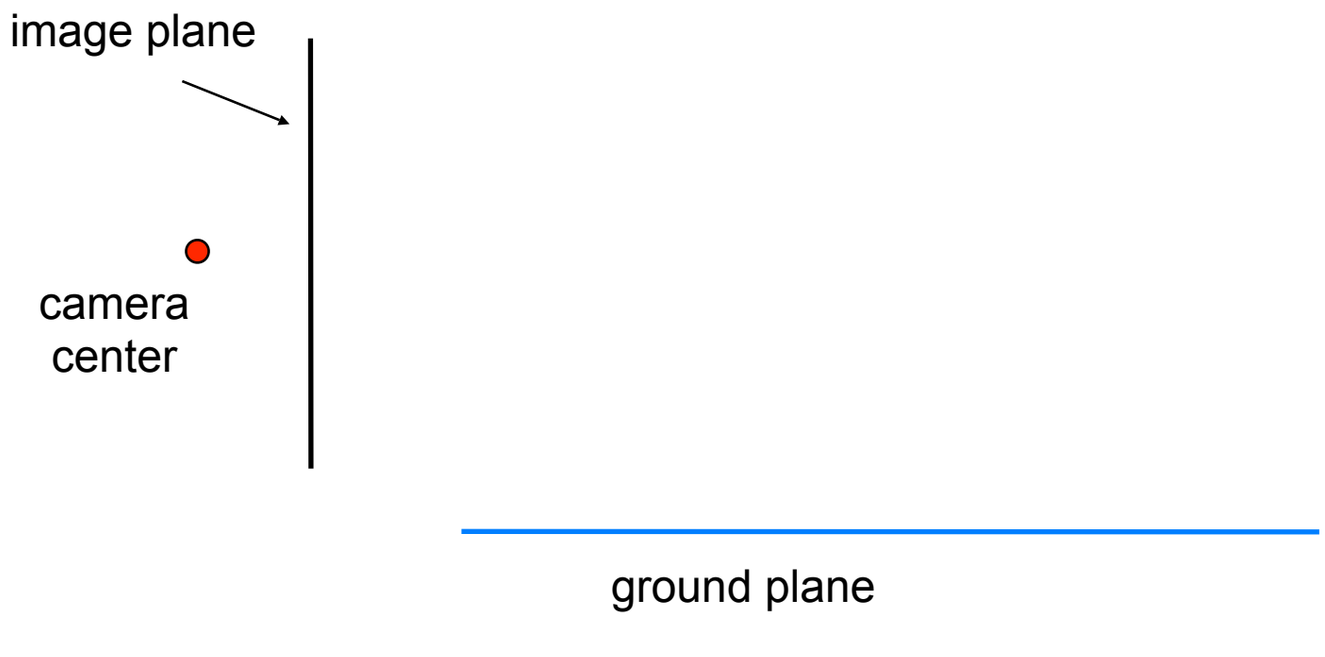
Vanishing points



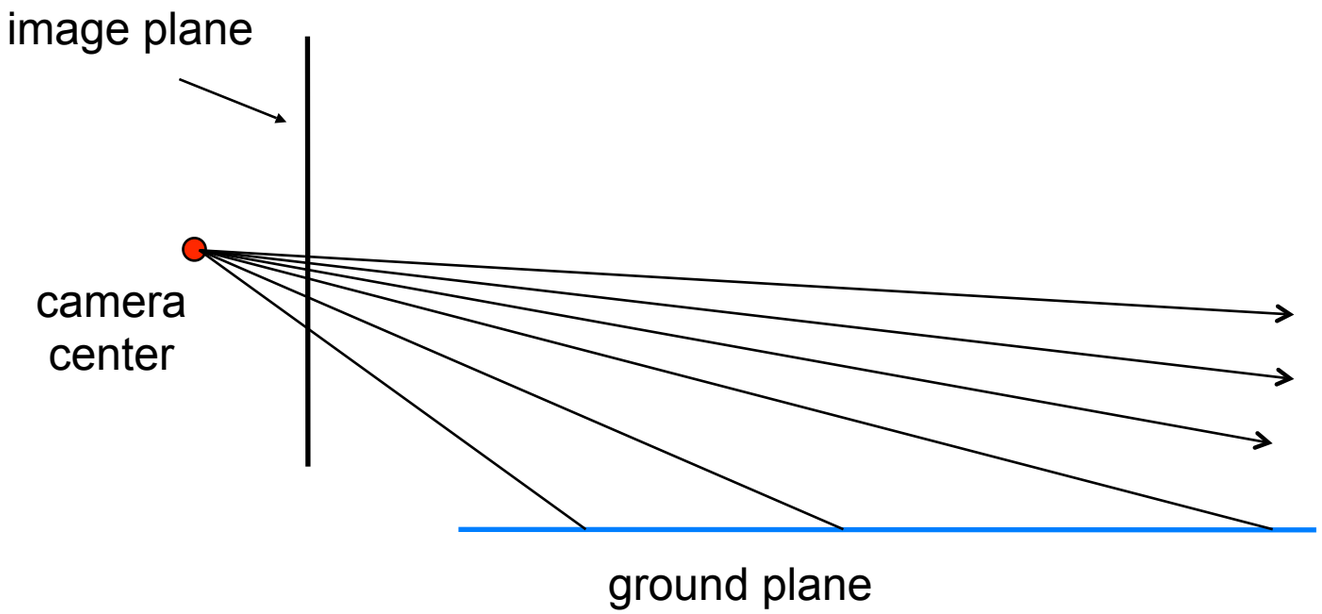
Vanishing points



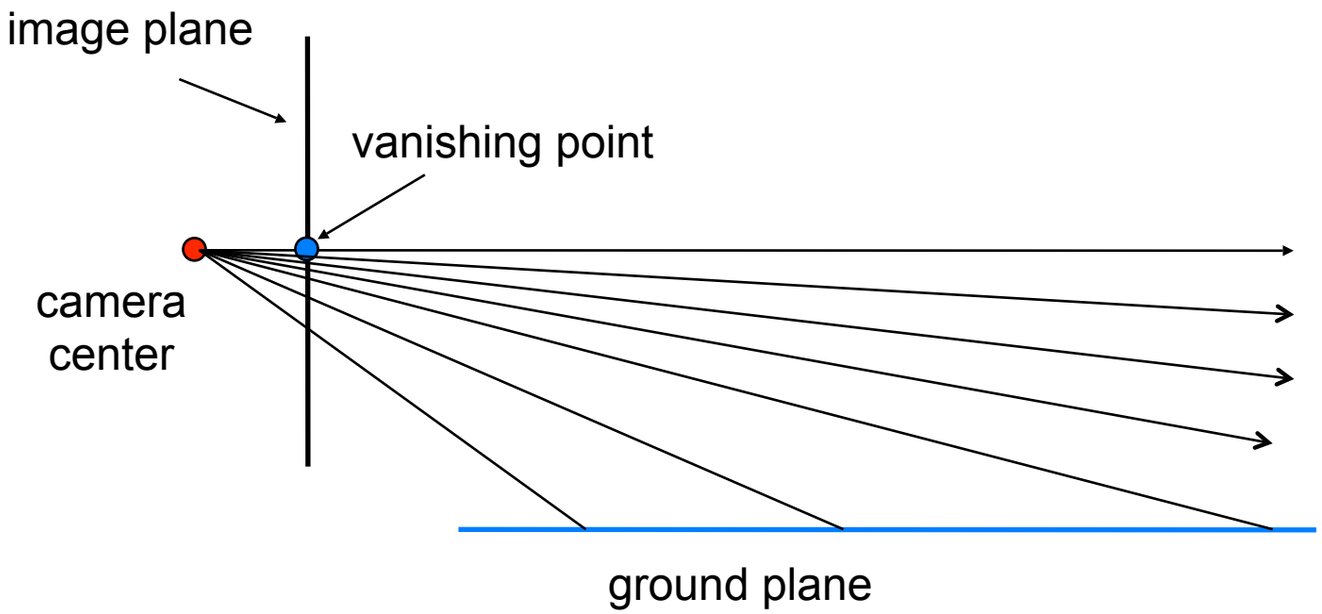
Vanishing points



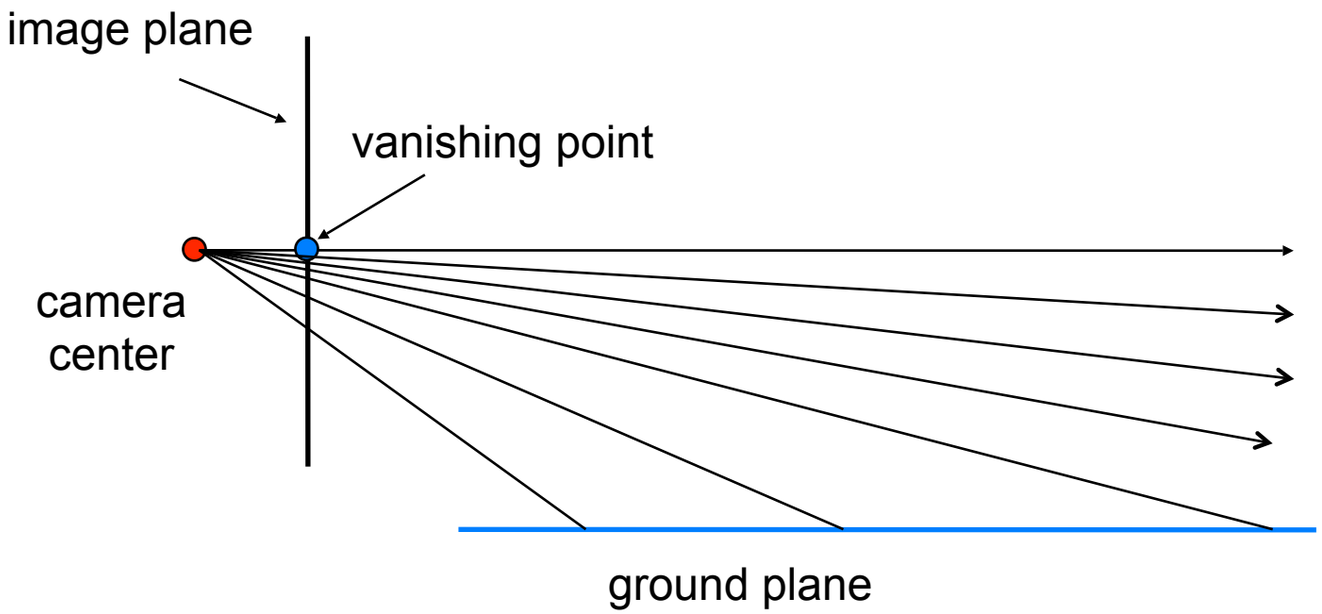
Vanishing points



Vanishing points



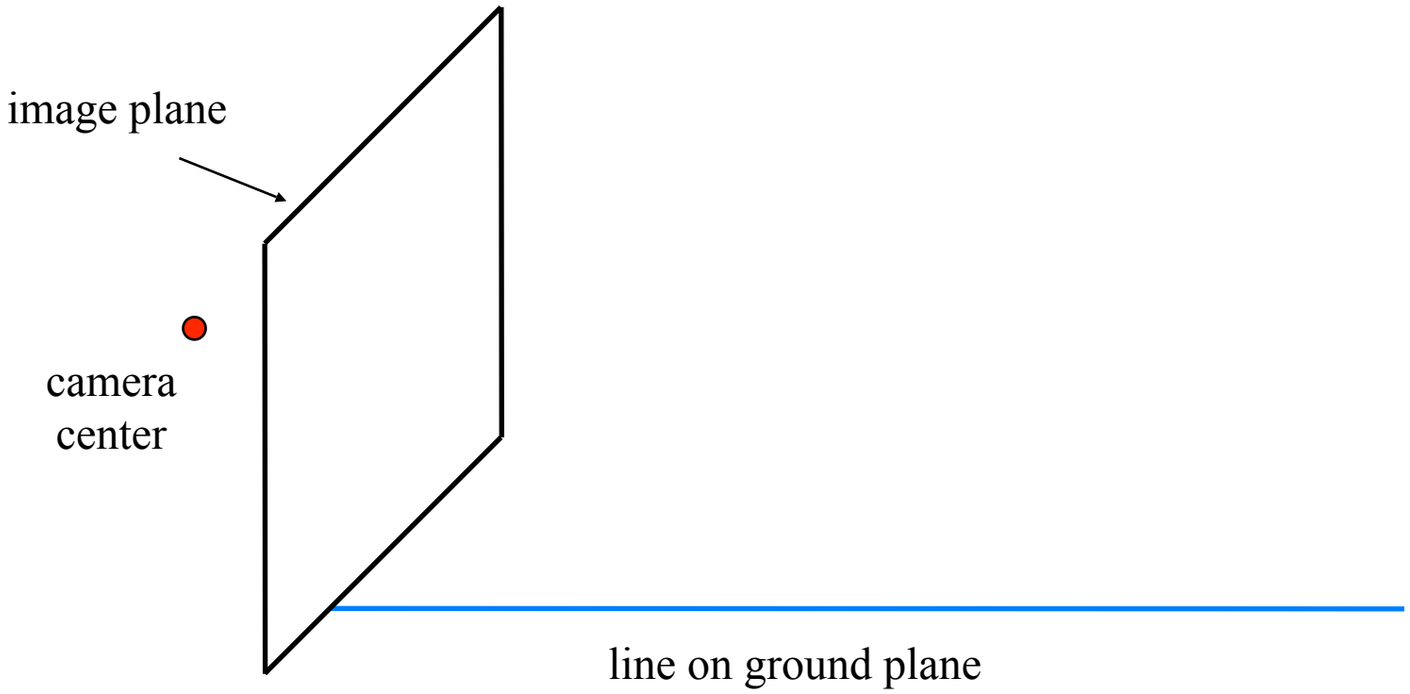
Vanishing points



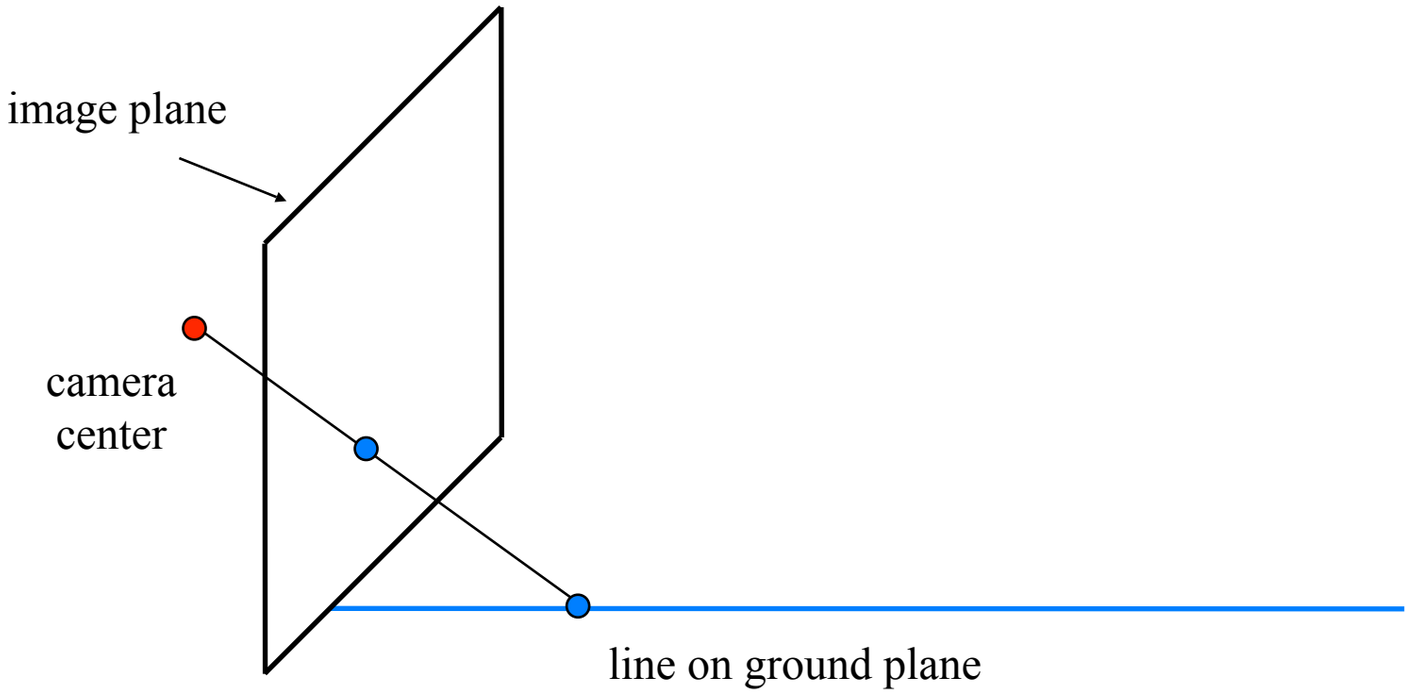
Vanishing point

- projection of a point at infinity

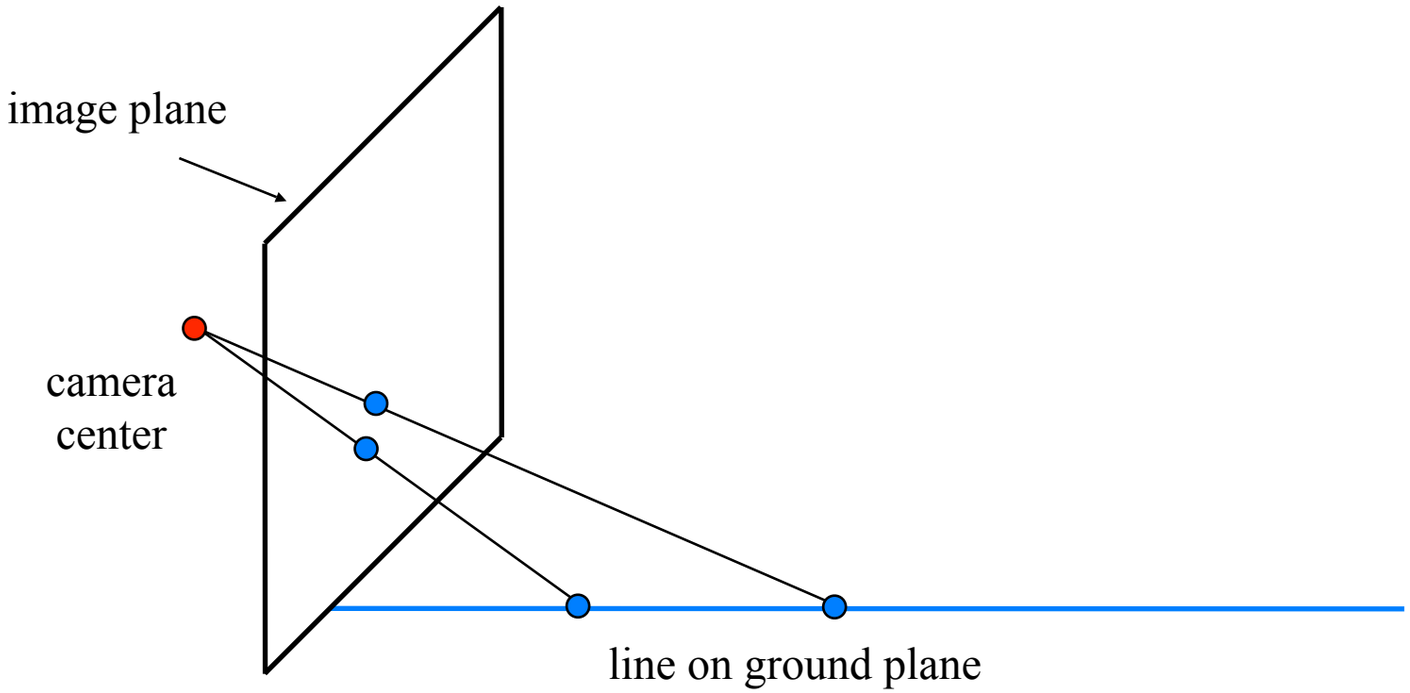
Vanishing points (2D)



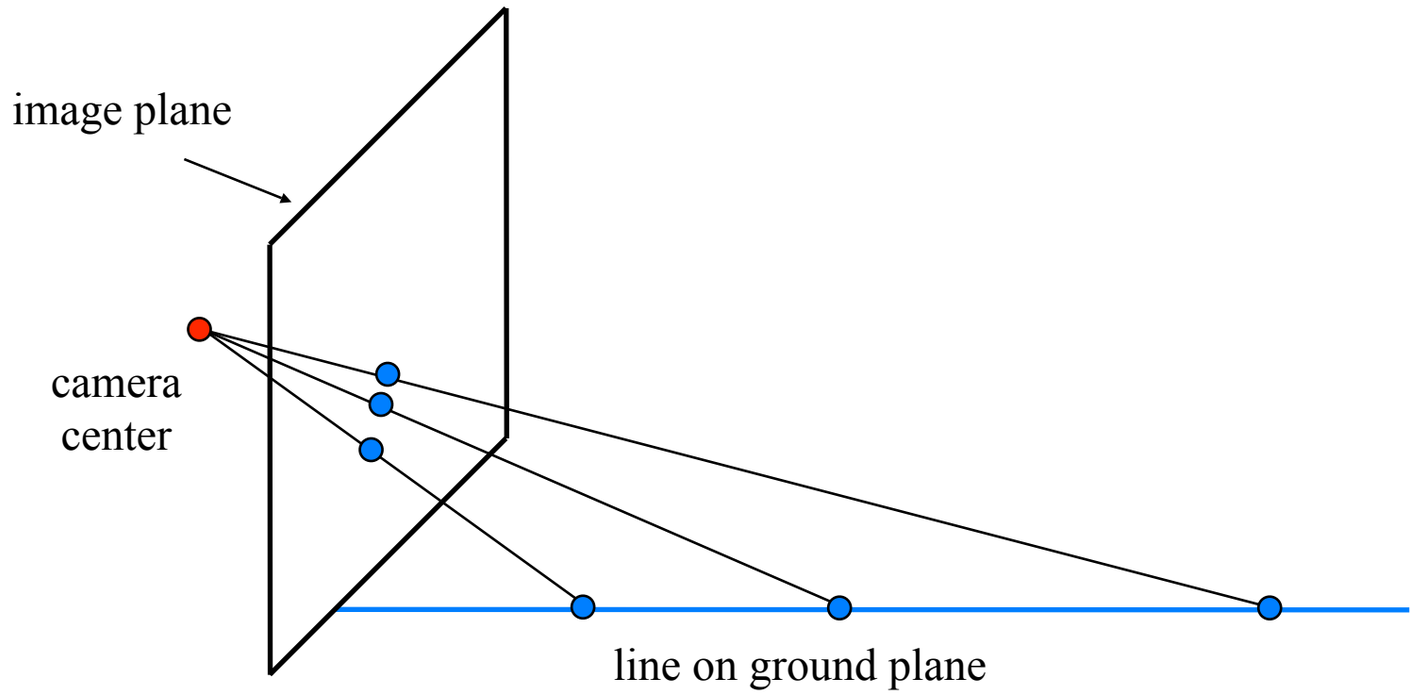
Vanishing points (2D)



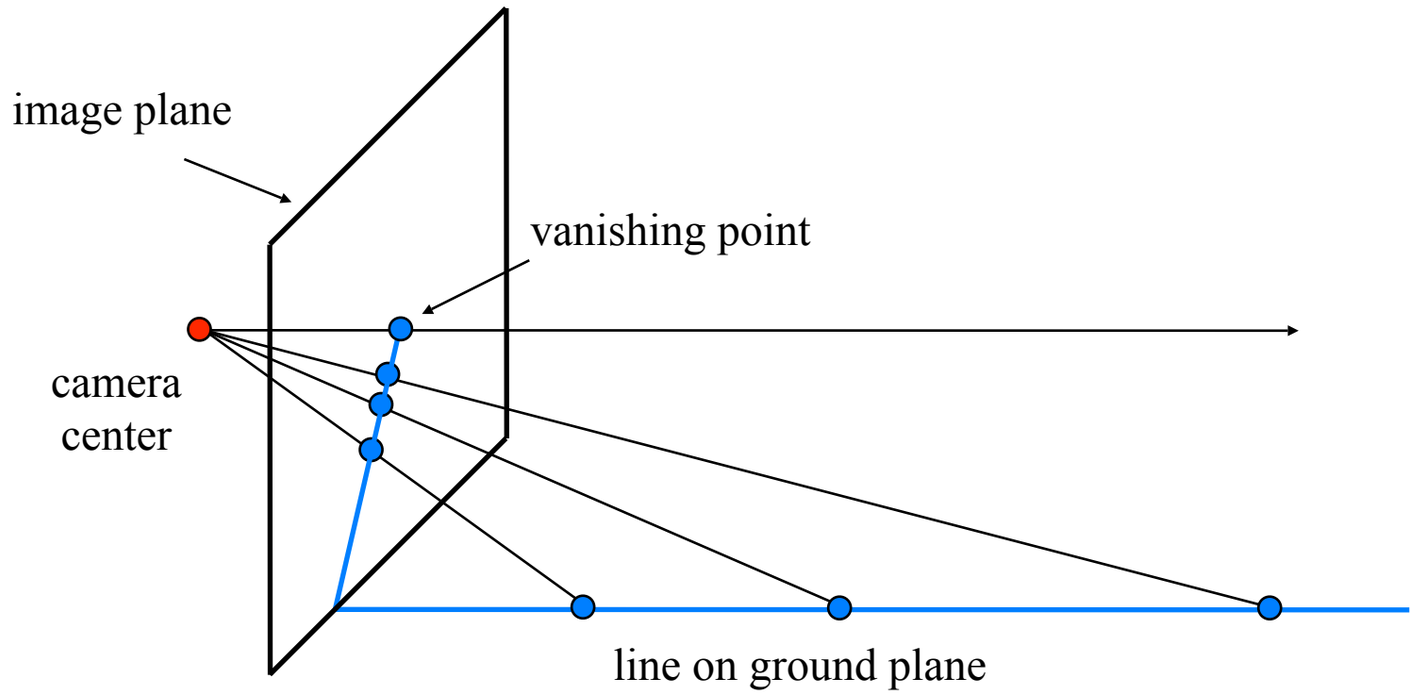
Vanishing points (2D)



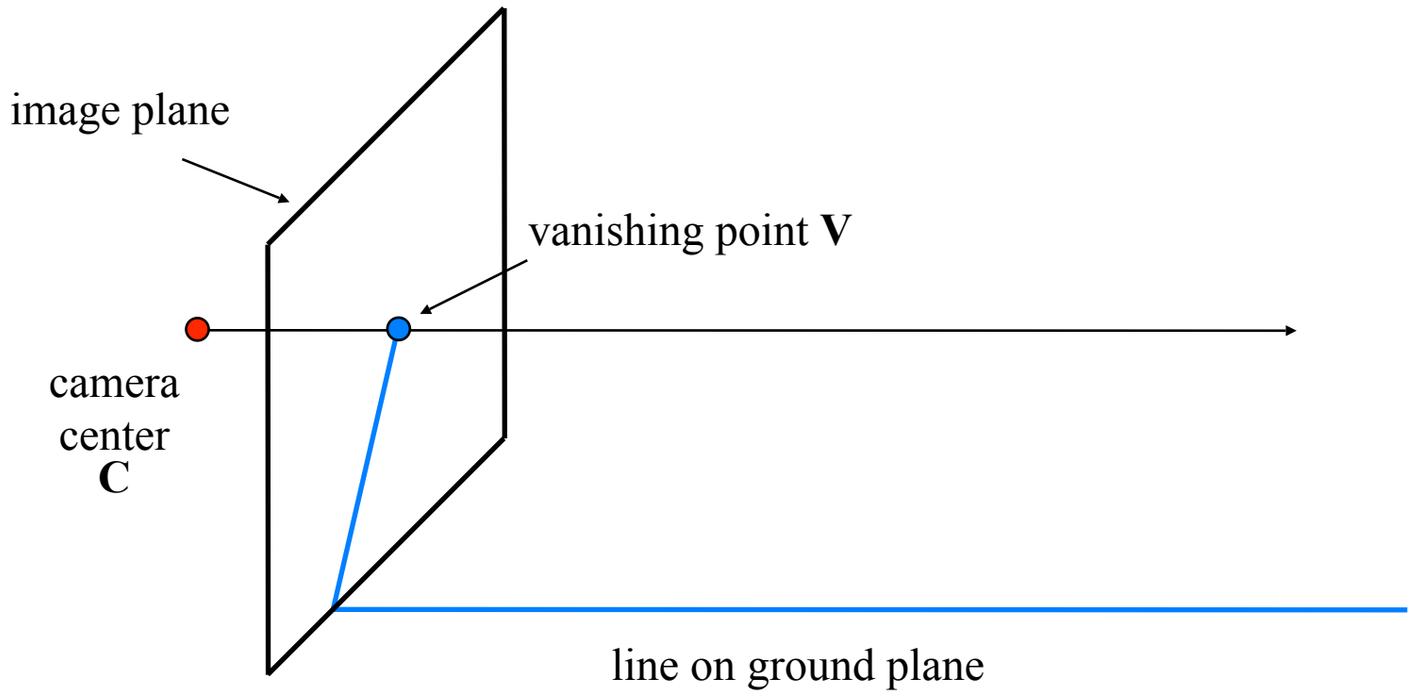
Vanishing points (2D)



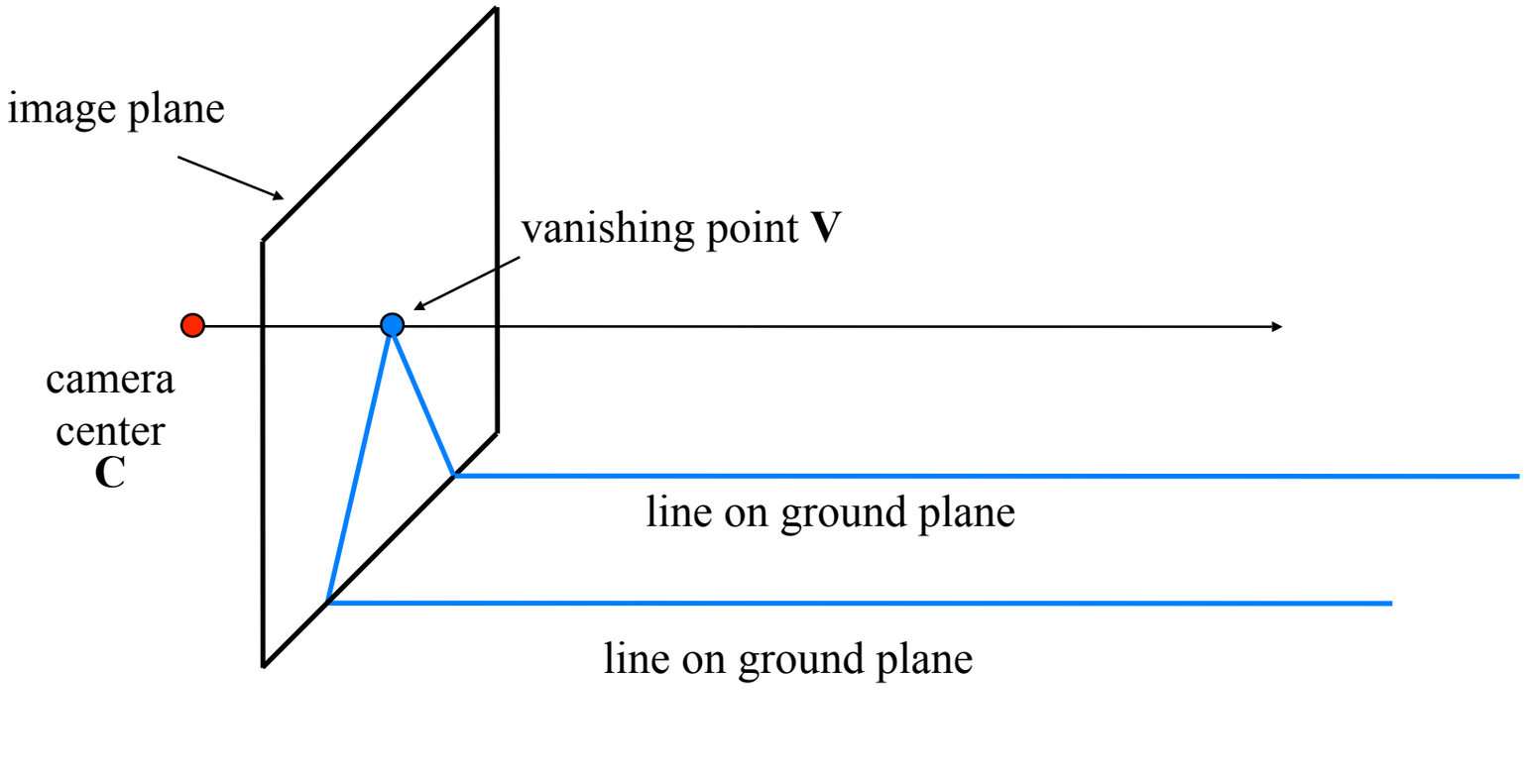
Vanishing points (2D)



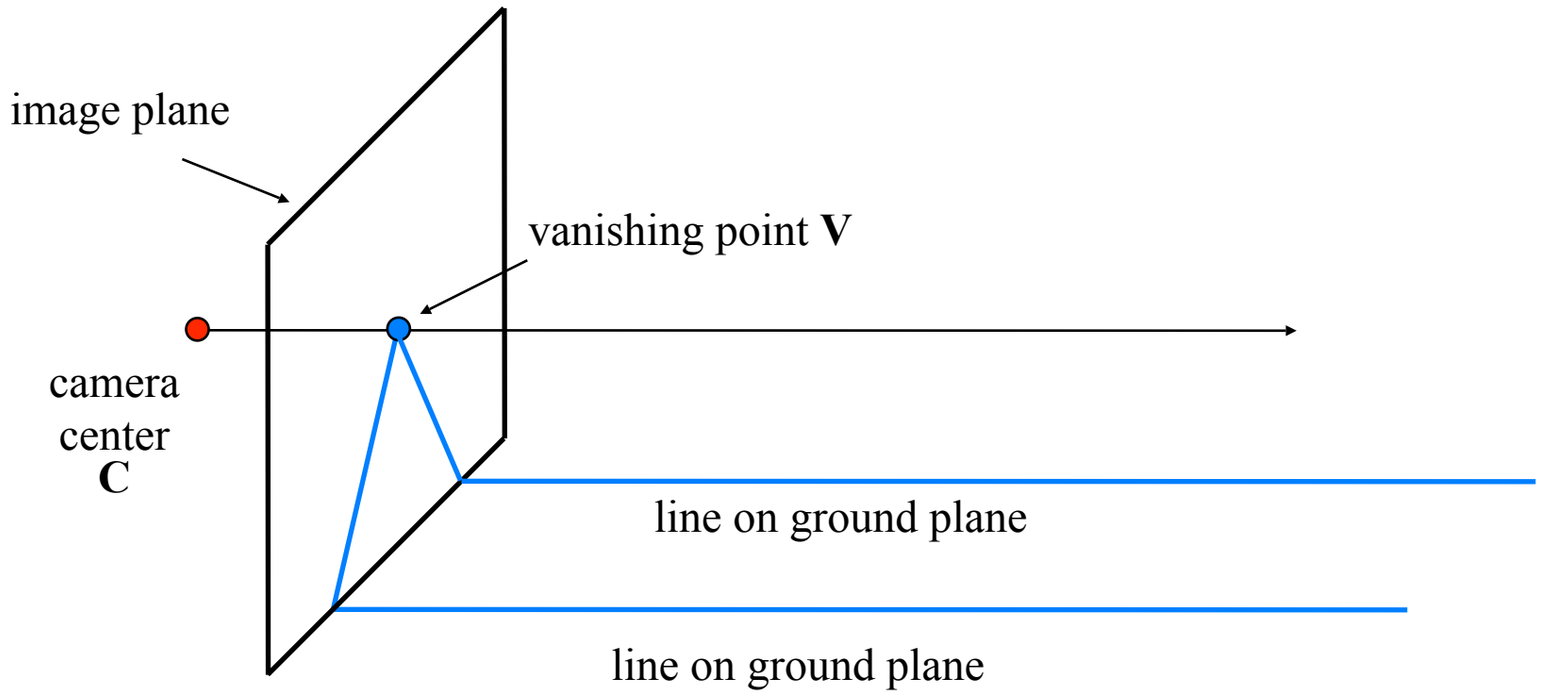
Vanishing points



Vanishing points



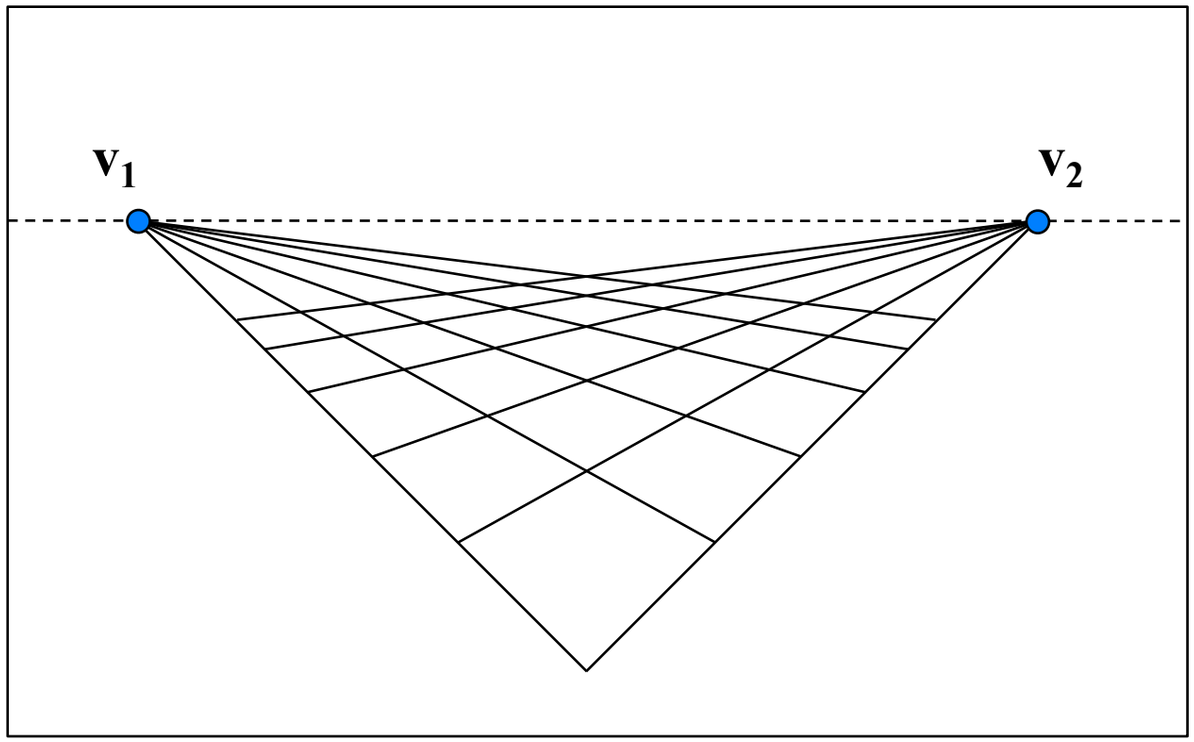
Vanishing points



Properties

- Any two parallel lines have the same vanishing point **v**
- The ray from **C** through **v** is parallel to the lines
- An image may have more than one vanishing point
  - in fact every pixel is a potential vanishing point

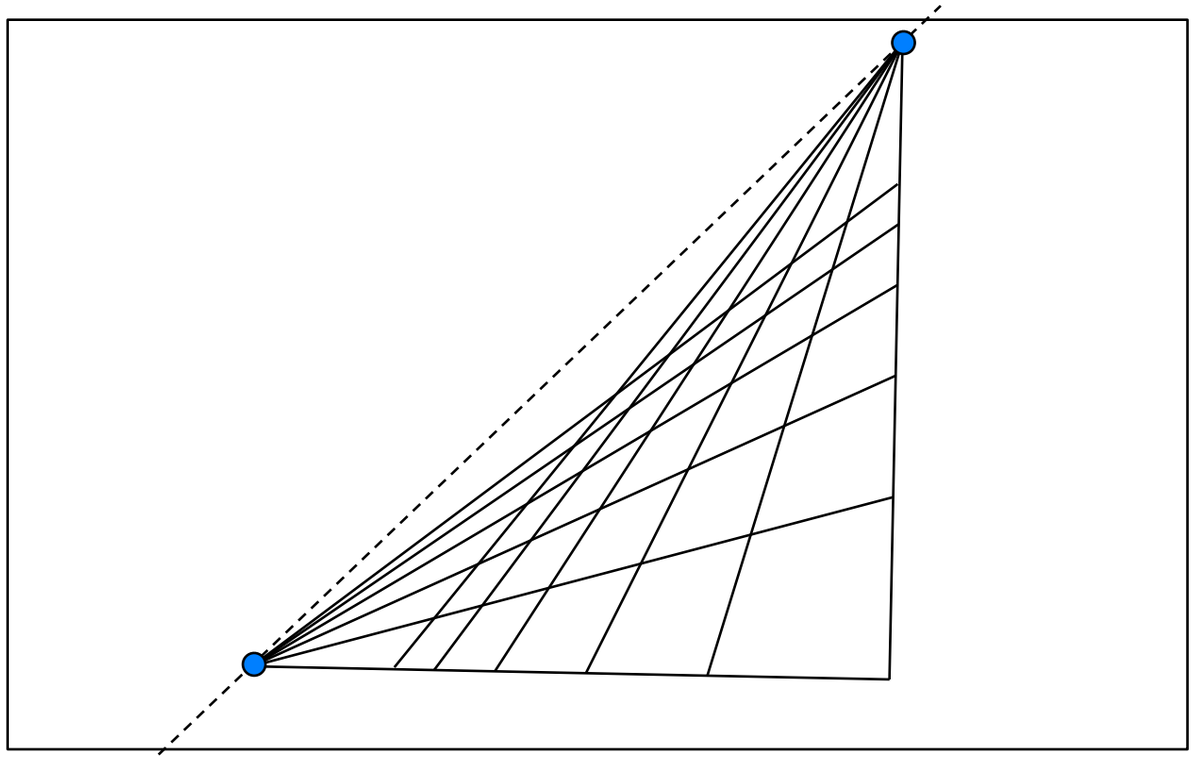
Vanishing lines



Multiple Vanishing Points

- Any set of parallel lines on the plane define a vanishing point
- The union of all of these vanishing points is the *horizon line*
  - also called *vanishing line*
- Note that different planes define different vanishing lines

Vanishing lines



Multiple Vanishing Points

- Any set of parallel lines on the plane define a vanishing point
- The union of all of these vanishing points is the *horizon line*
  - also called *vanishing line*
- Note that different planes define different vanishing lines

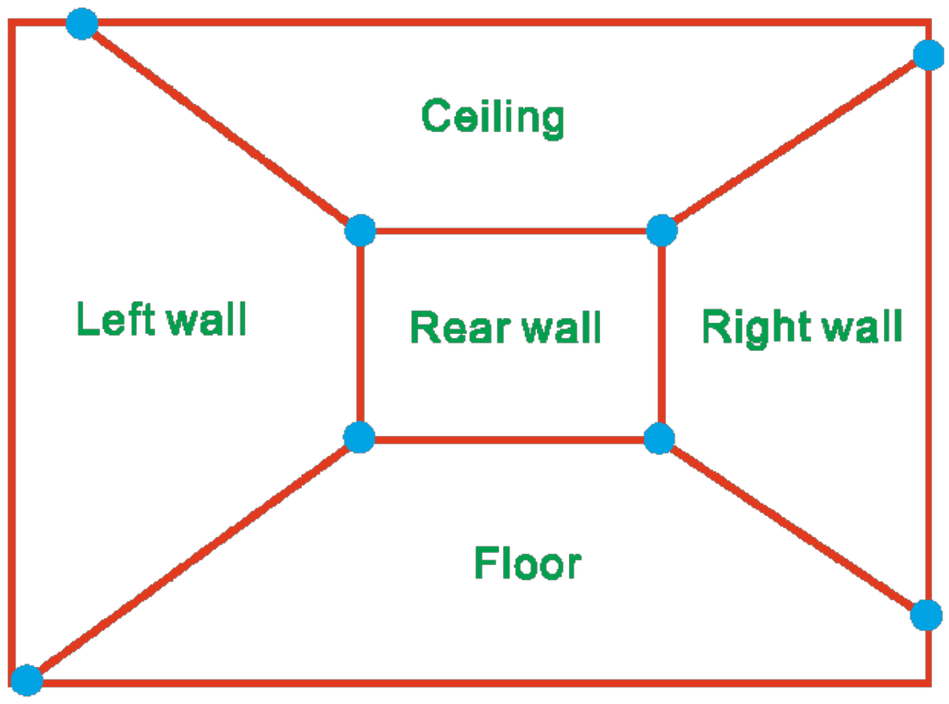
“Tour into the Picture” (SIGGRAPH ’97)

Create a 3D “theatre stage” of five billboards

Specify foreground objects through bounding polygons

Use camera transformations to navigate through the scene

The idea



The idea

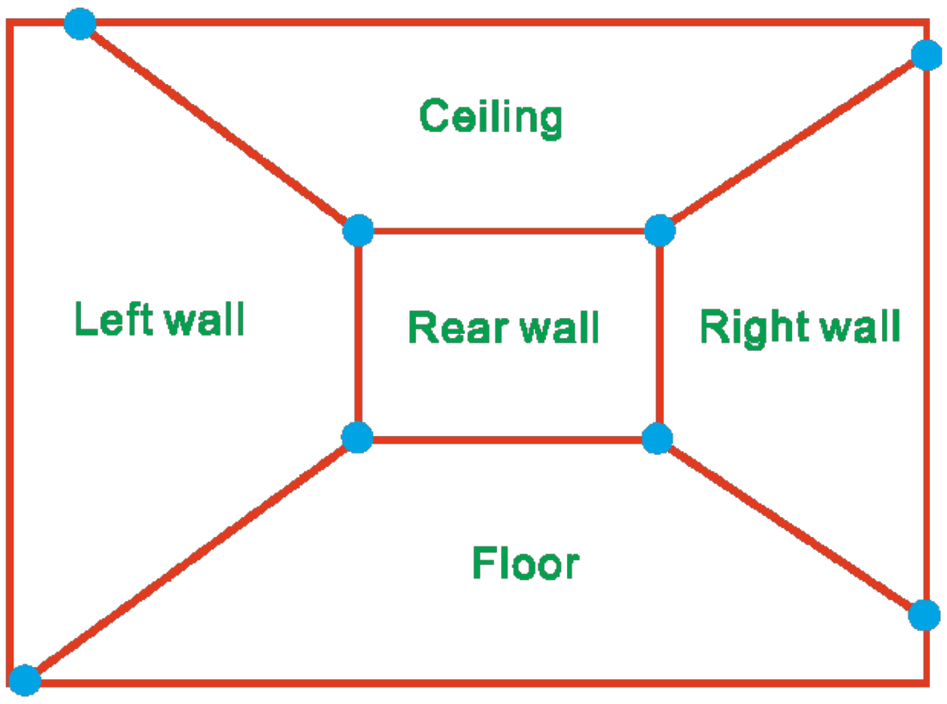
Many scenes (especially paintings), can be represented as an axis-aligned box volume (i.e. a stage)

Key assumptions:

- All walls of volume are orthogonal
- Camera view plane is parallel to back of volume
- Camera up is normal to volume bottom

How many vanishing points does the box have?

- Three, but two at infinity
- Single-point perspective



The idea

Many scenes (especially paintings), can be represented as an axis-aligned box volume (i.e. a stage)

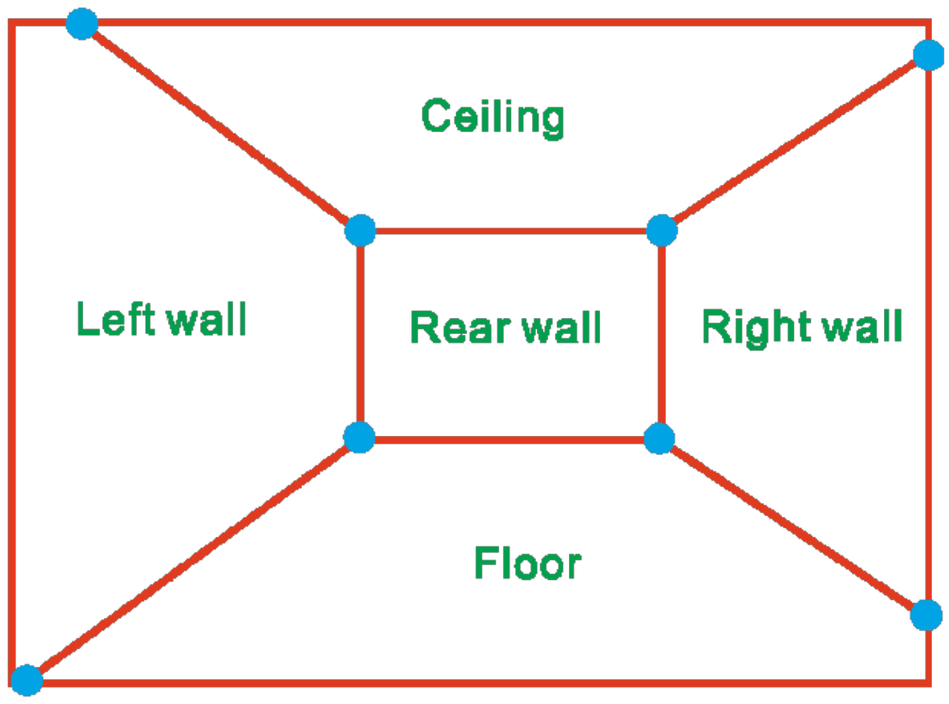
Key assumptions:

- All walls of volume are orthogonal
- Camera view plane is parallel to back of volume
- Camera up is normal to volume bottom

How many vanishing points does the box have?

- Three, but two at infinity
- Single-point perspective

Can use the vanishing point to fit the box to the particular



The idea

Many scenes (especially paintings), can be represented as an axis-aligned box volume (i.e. a stage)

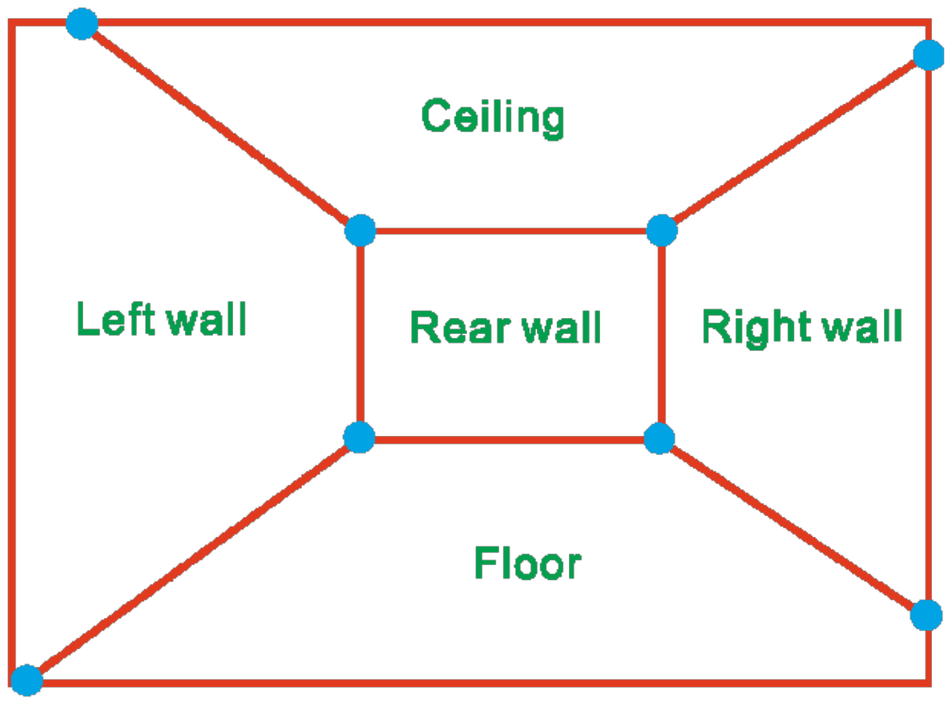
Key assumptions:

- All walls of volume are orthogonal
- Camera view plane is parallel to back of volume
- Camera up is normal to volume bottom

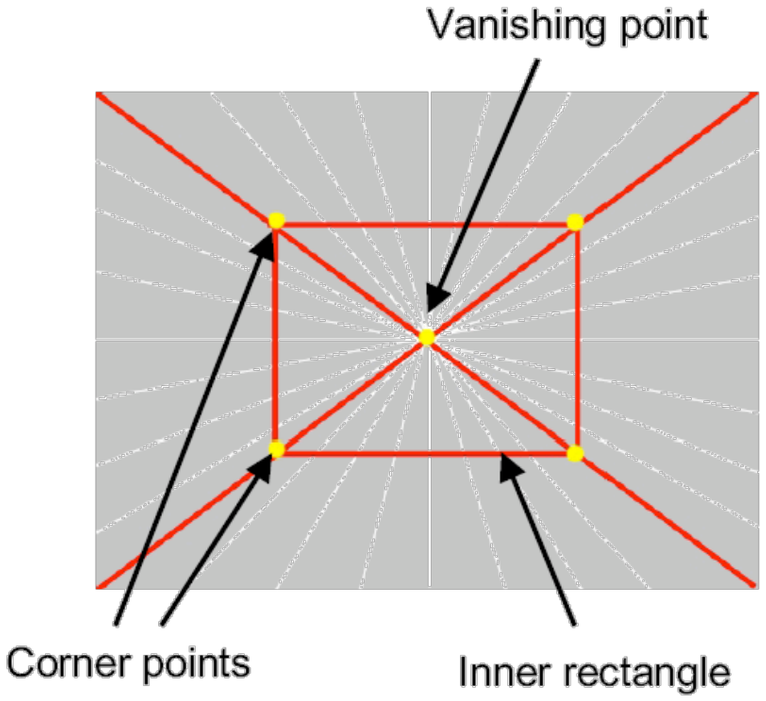
How many vanishing points does the box have?

- Three, but two at infinity
- Single-point perspective

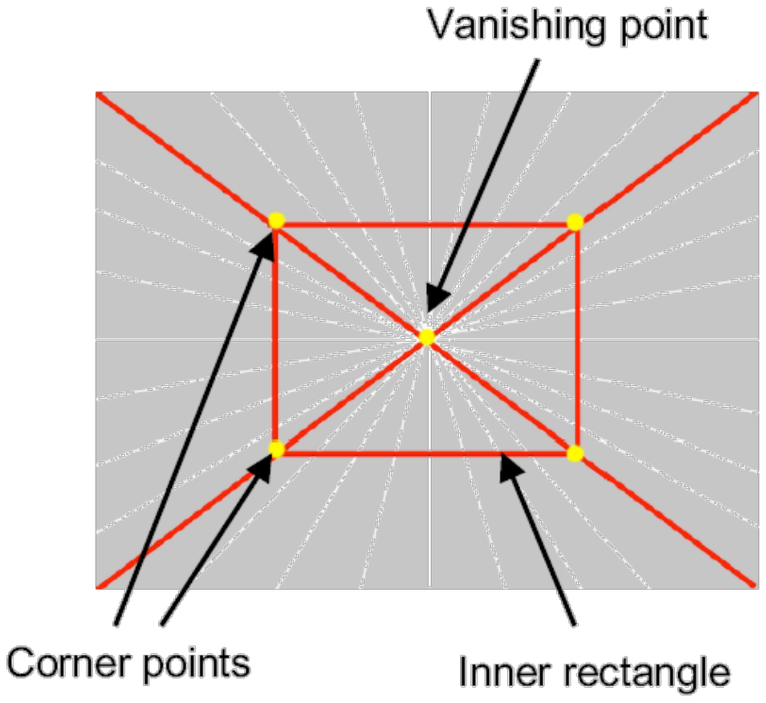
Can use the vanishing point to fit the box to the particular Scene!



Fitting the box volume

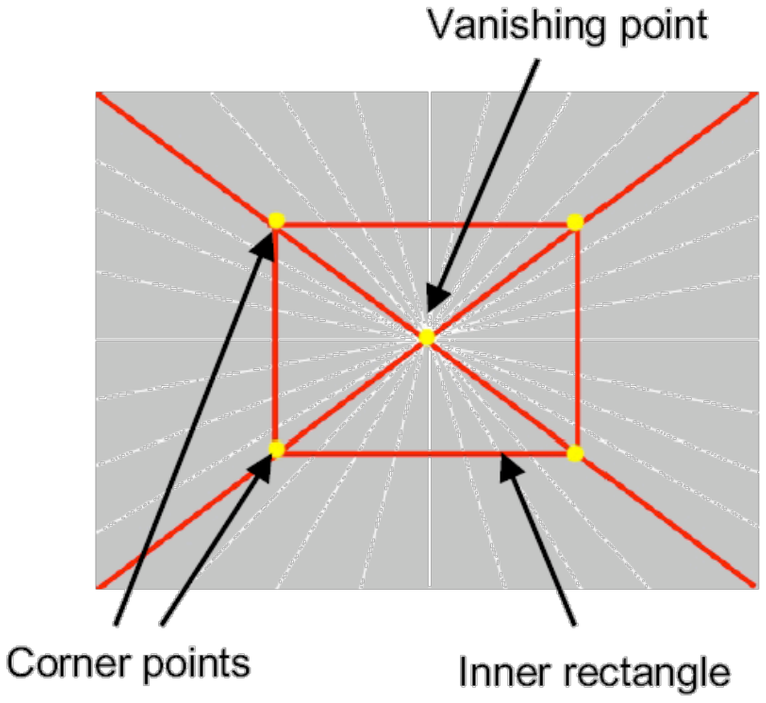


Fitting the box volume



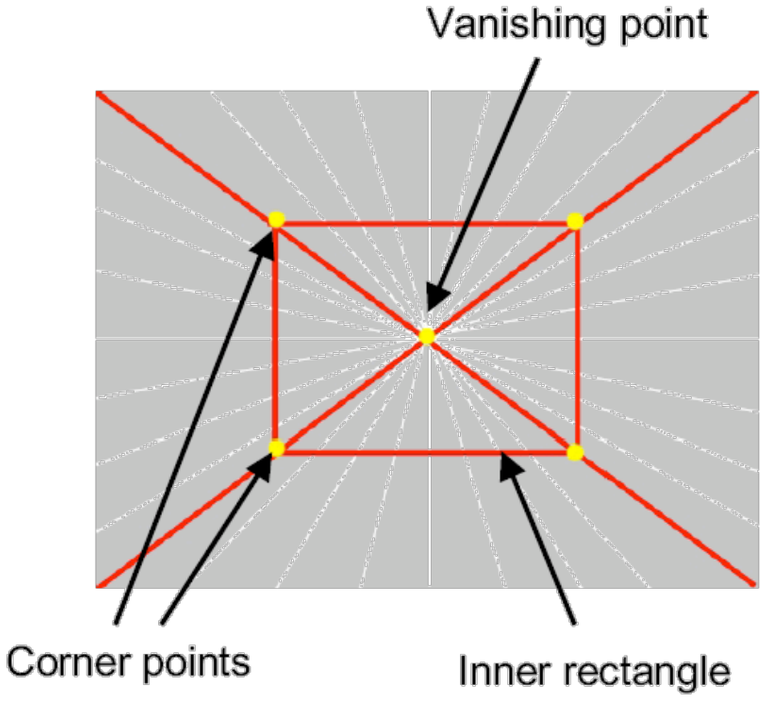
User controls the inner box and the vanishing point placement (# of DOF???)

Fitting the box volume



User controls the inner box and the vanishing point placement (# of DOF???)

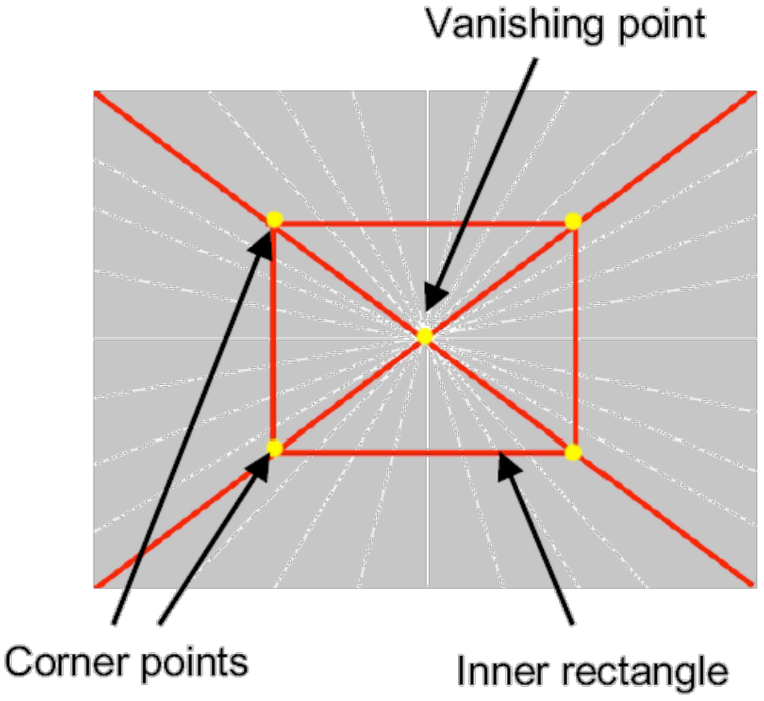
Fitting the box volume



User controls the inner box and the vanishing point placement (# of DOF???)

Q: What's the significance of the vanishing point location?

Fitting the box volume

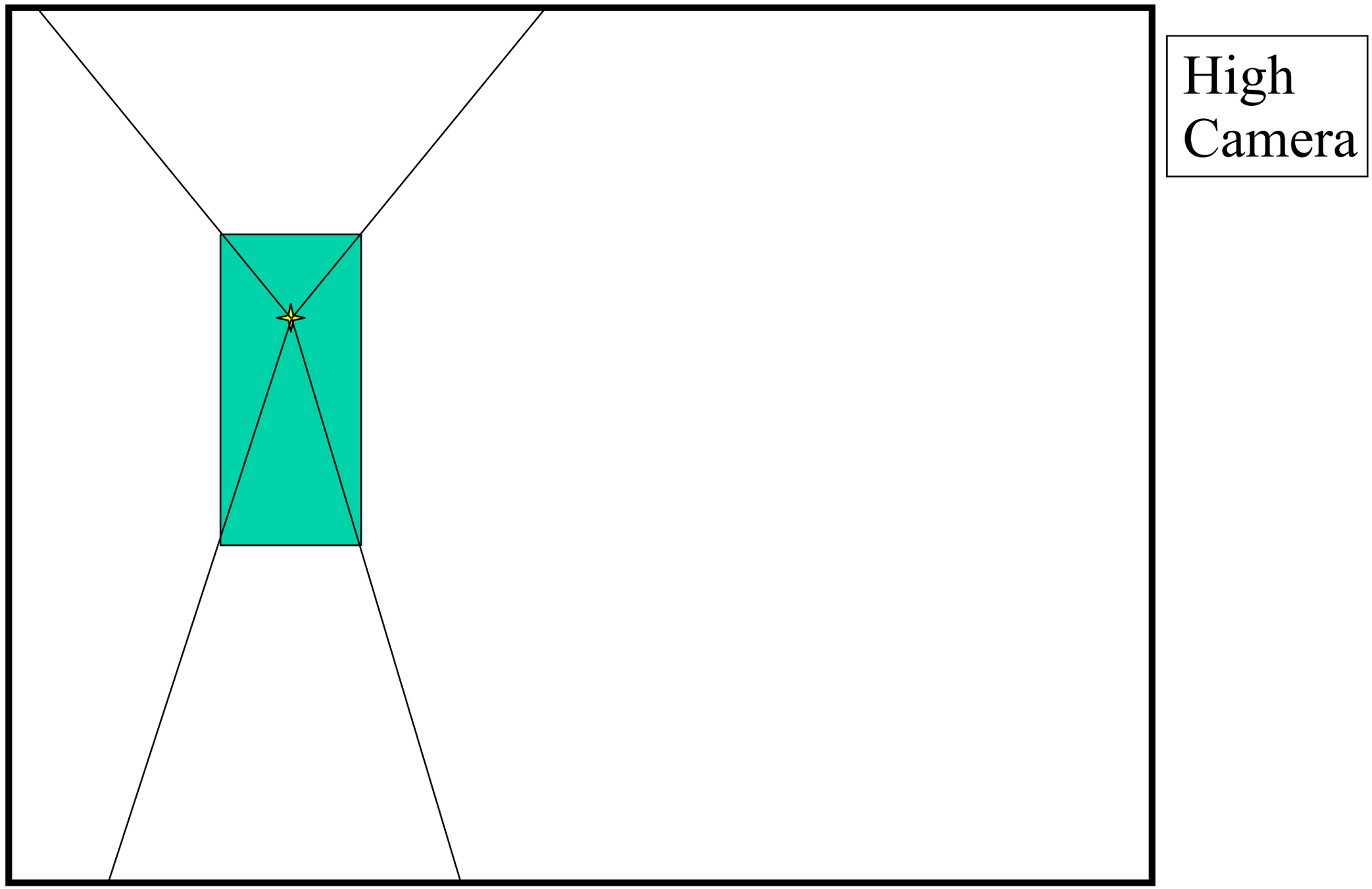


User controls the inner box and the vanishing point placement (# of DOF???)

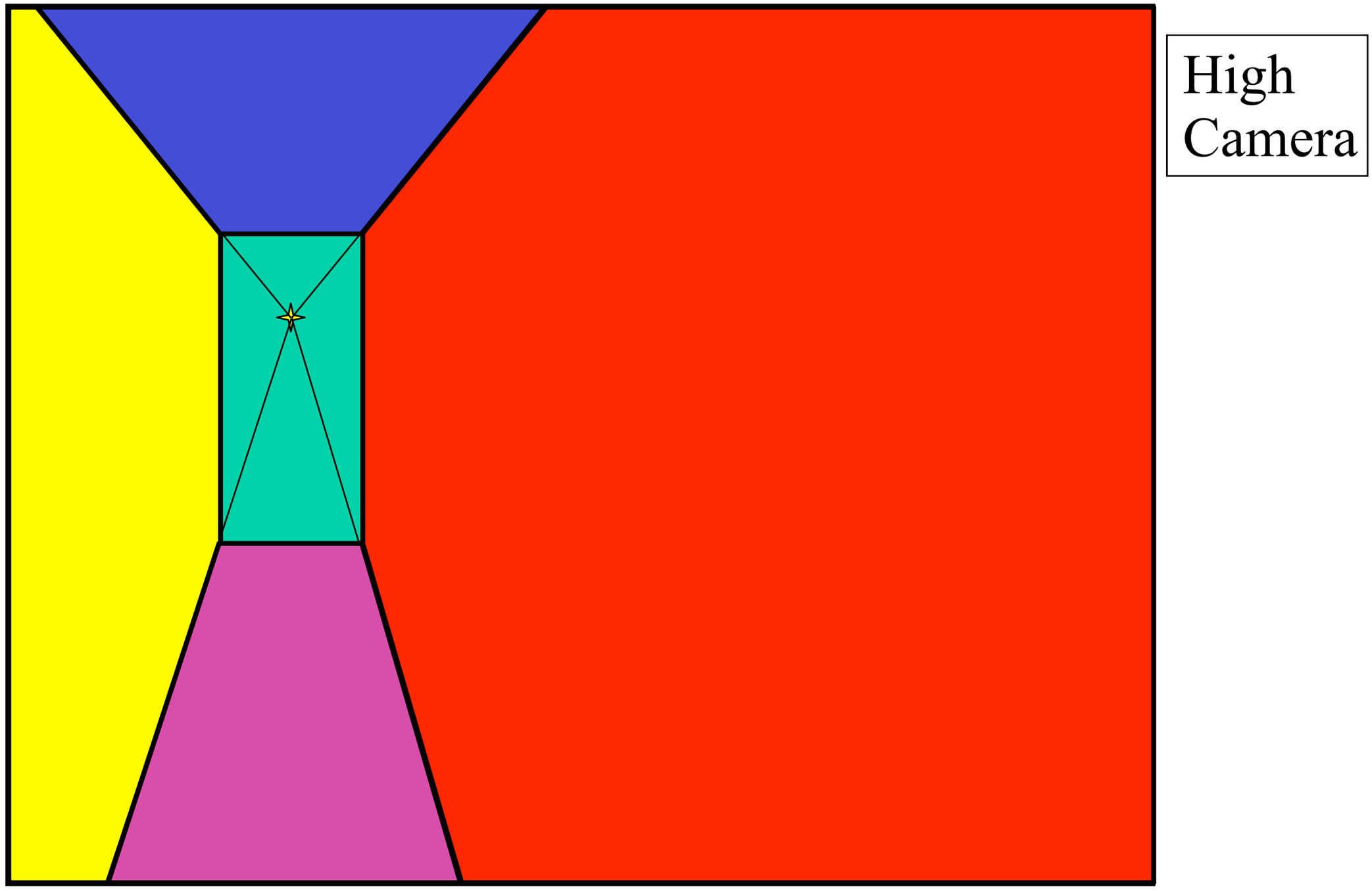
Q: What's the significance of the vanishing point location?

A: It's at eye level: ray from COP to VP is perpendicular to image plane.

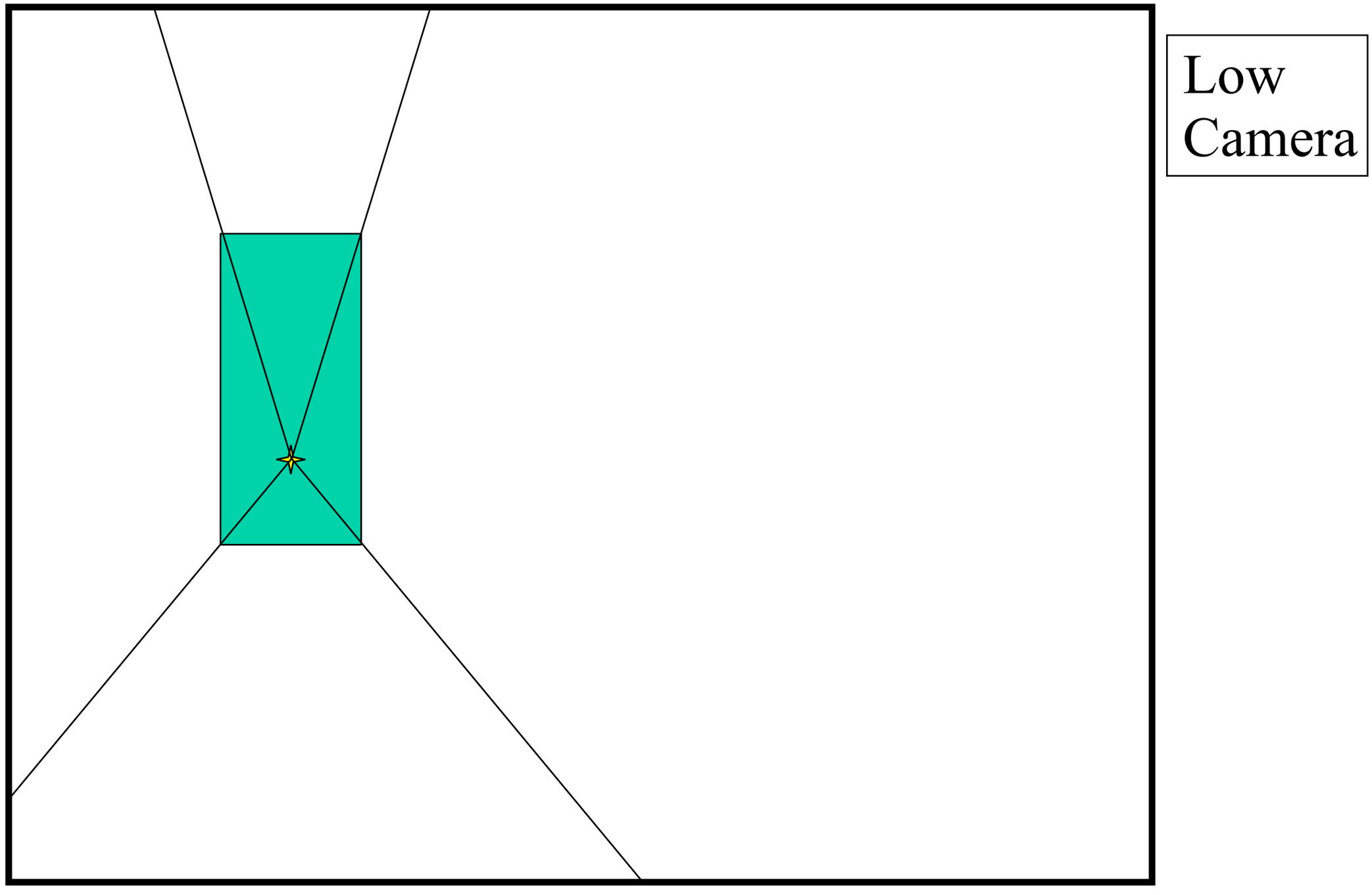
Example of user input: vanishing point and back face of view volume are defined



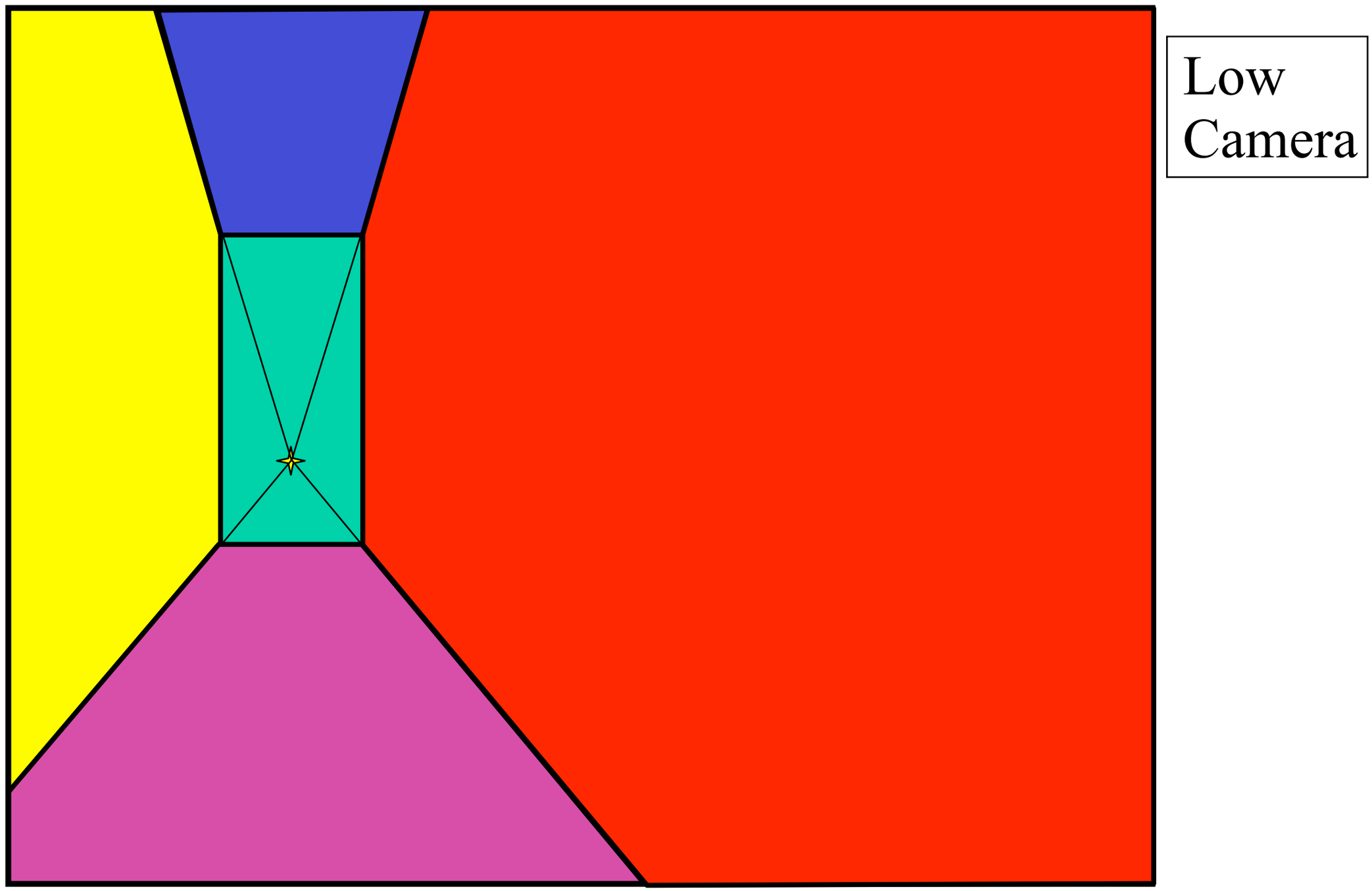
Example of user input: vanishing point and back face of view volume are defined



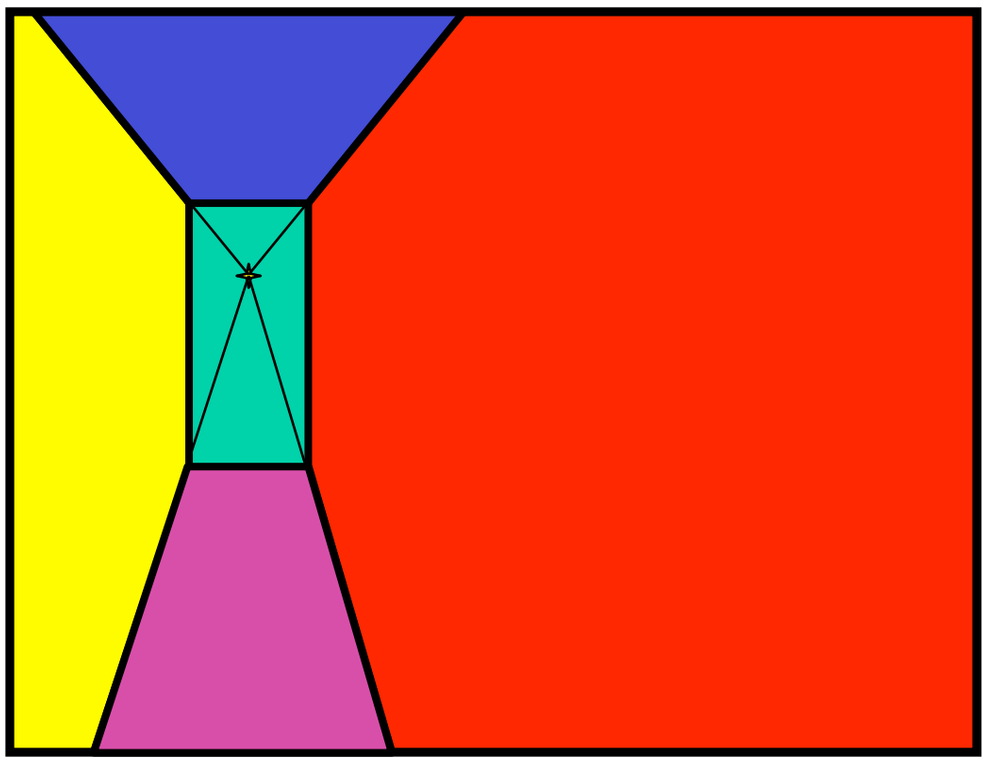
Example of user input: vanishing point and back face of view volume are defined



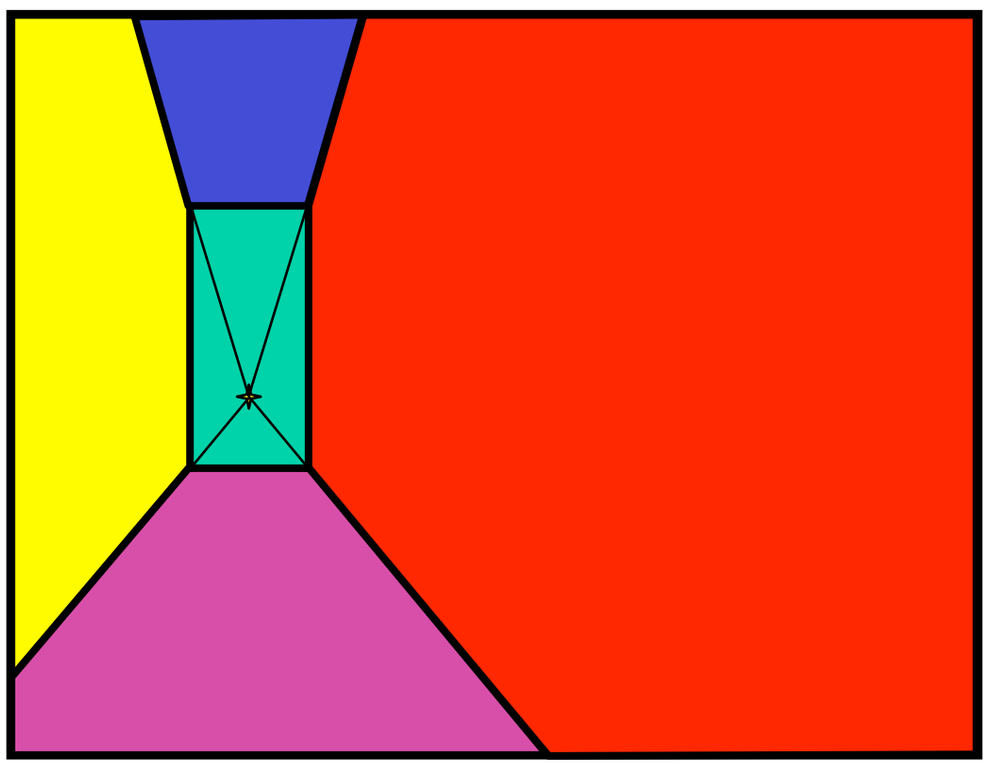
Example of user input: vanishing point and back face of view volume are defined



Comparison of how image is subdivided based on two different camera positions. You should see how moving the vanishing point corresponds to moving the eyepoint in the 3D world.

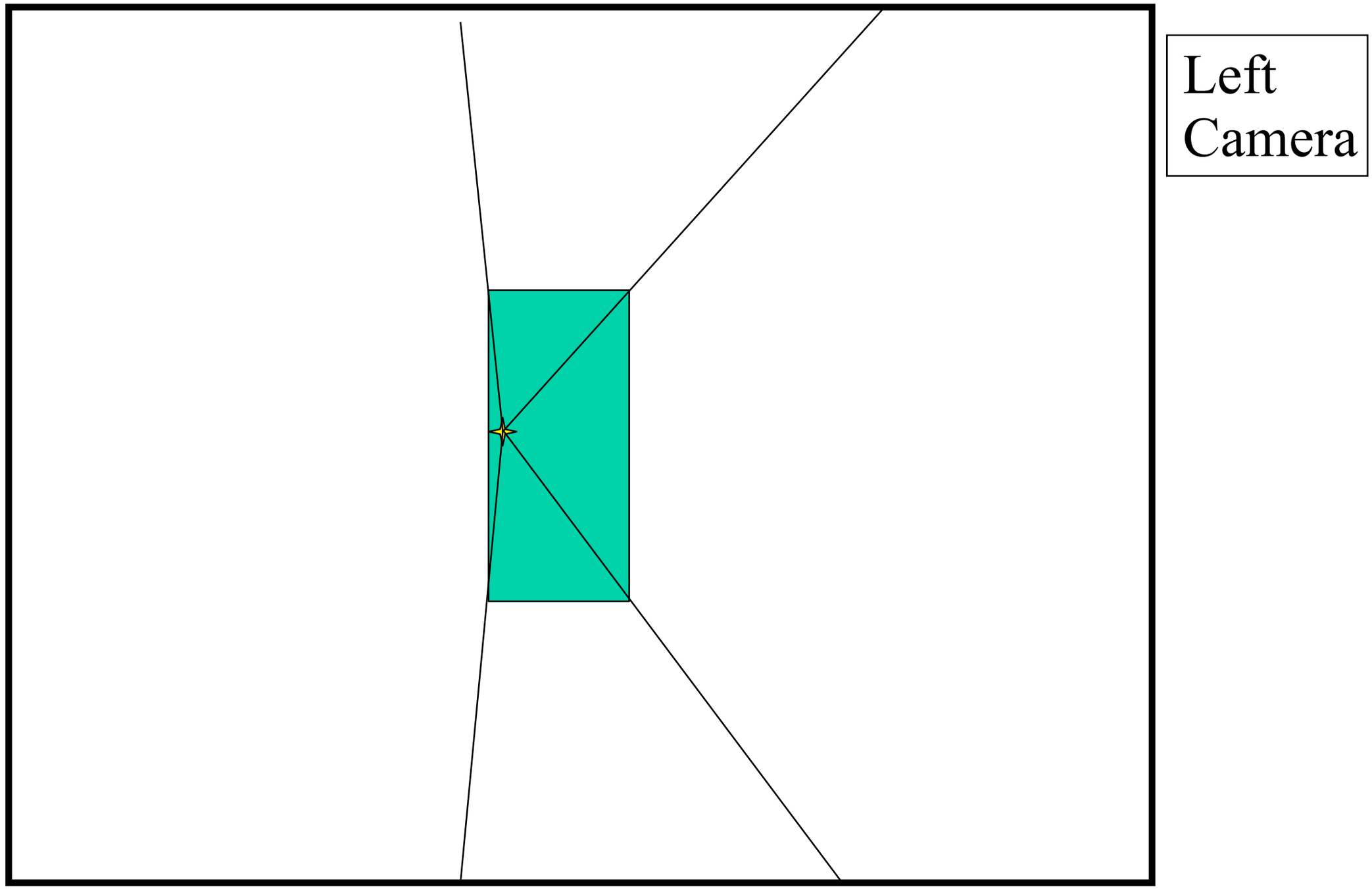


High Camera

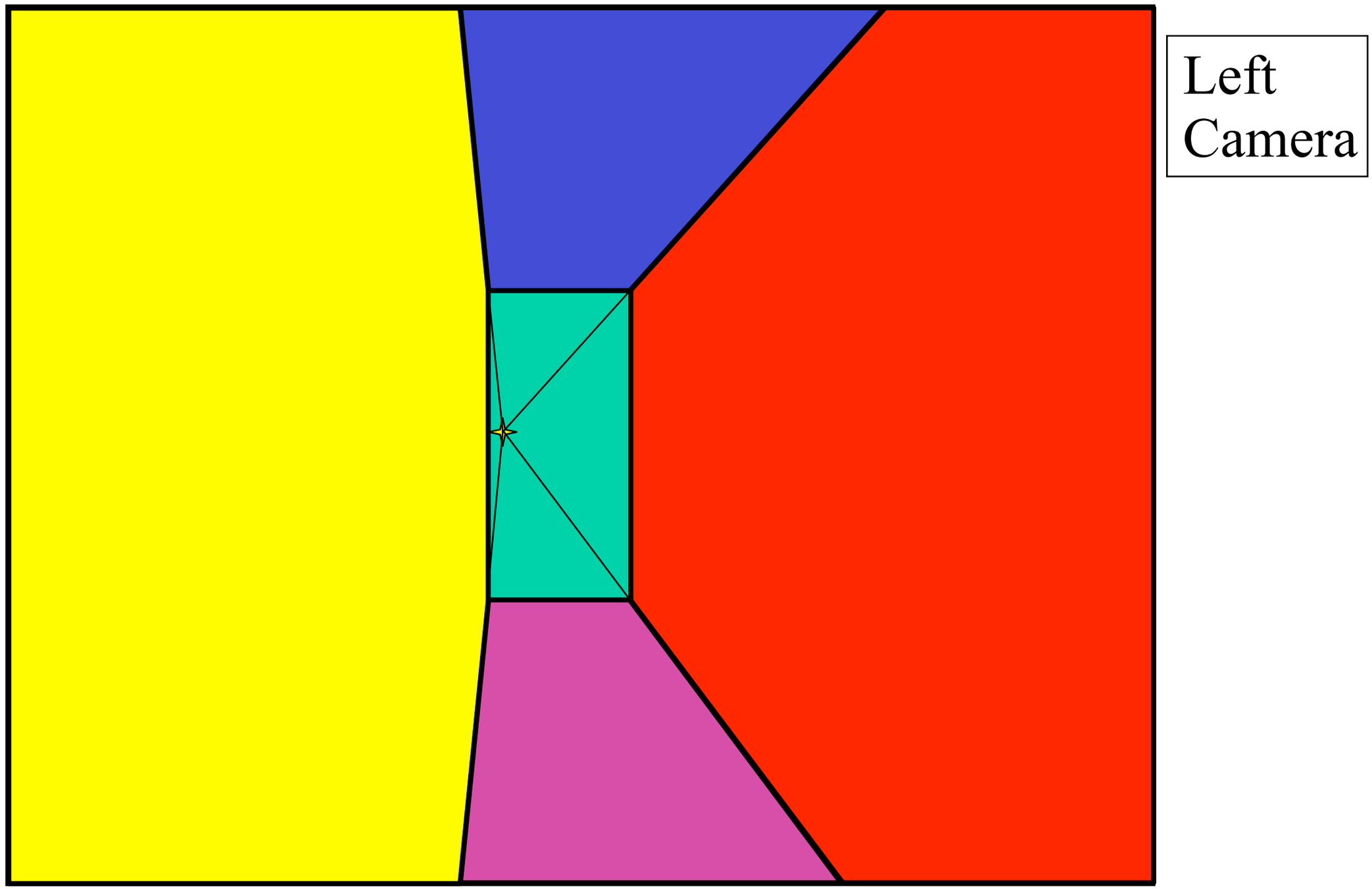


Low Camera

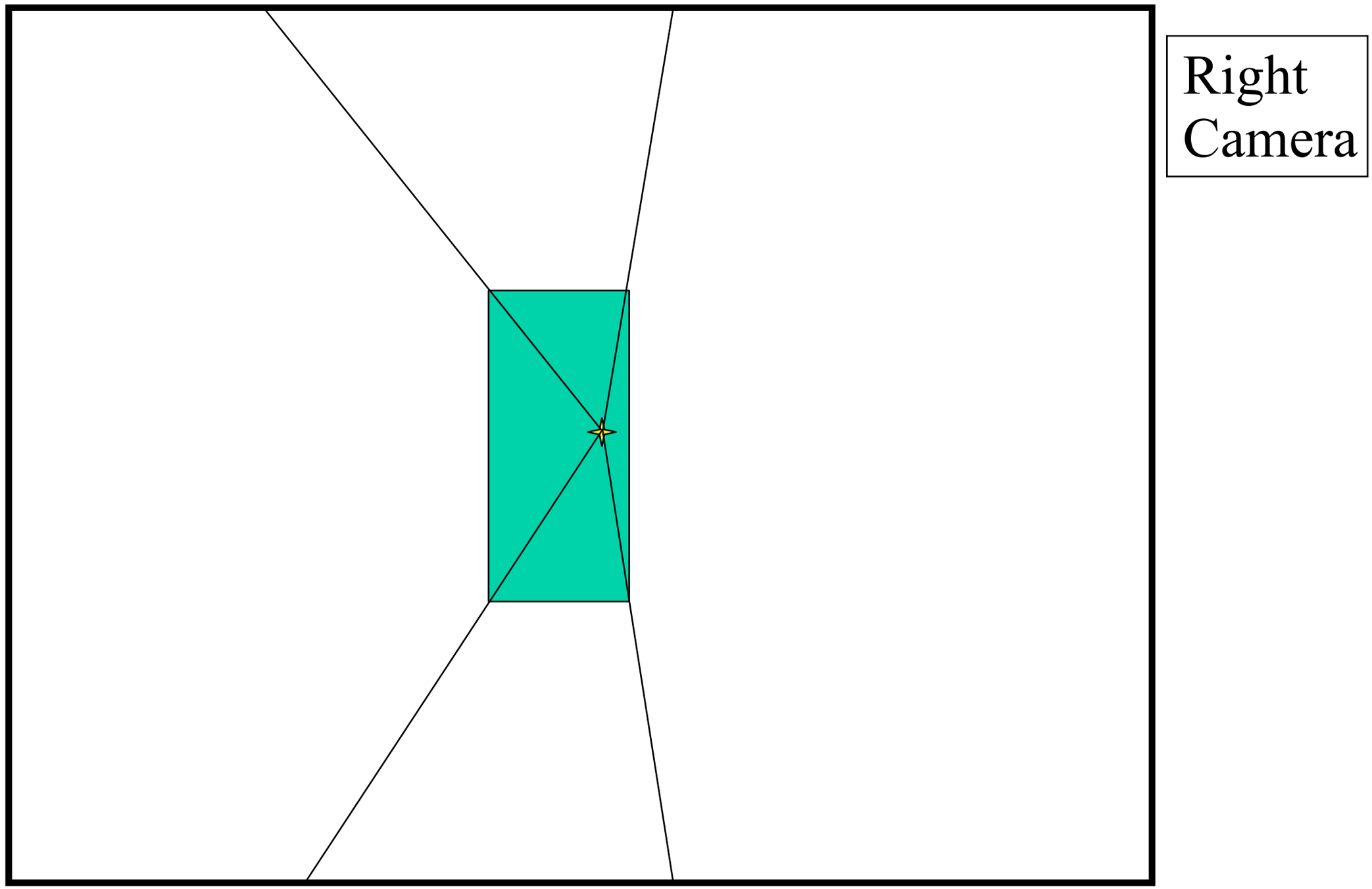
Another example of user input: vanishing point and back face of view volume are defined



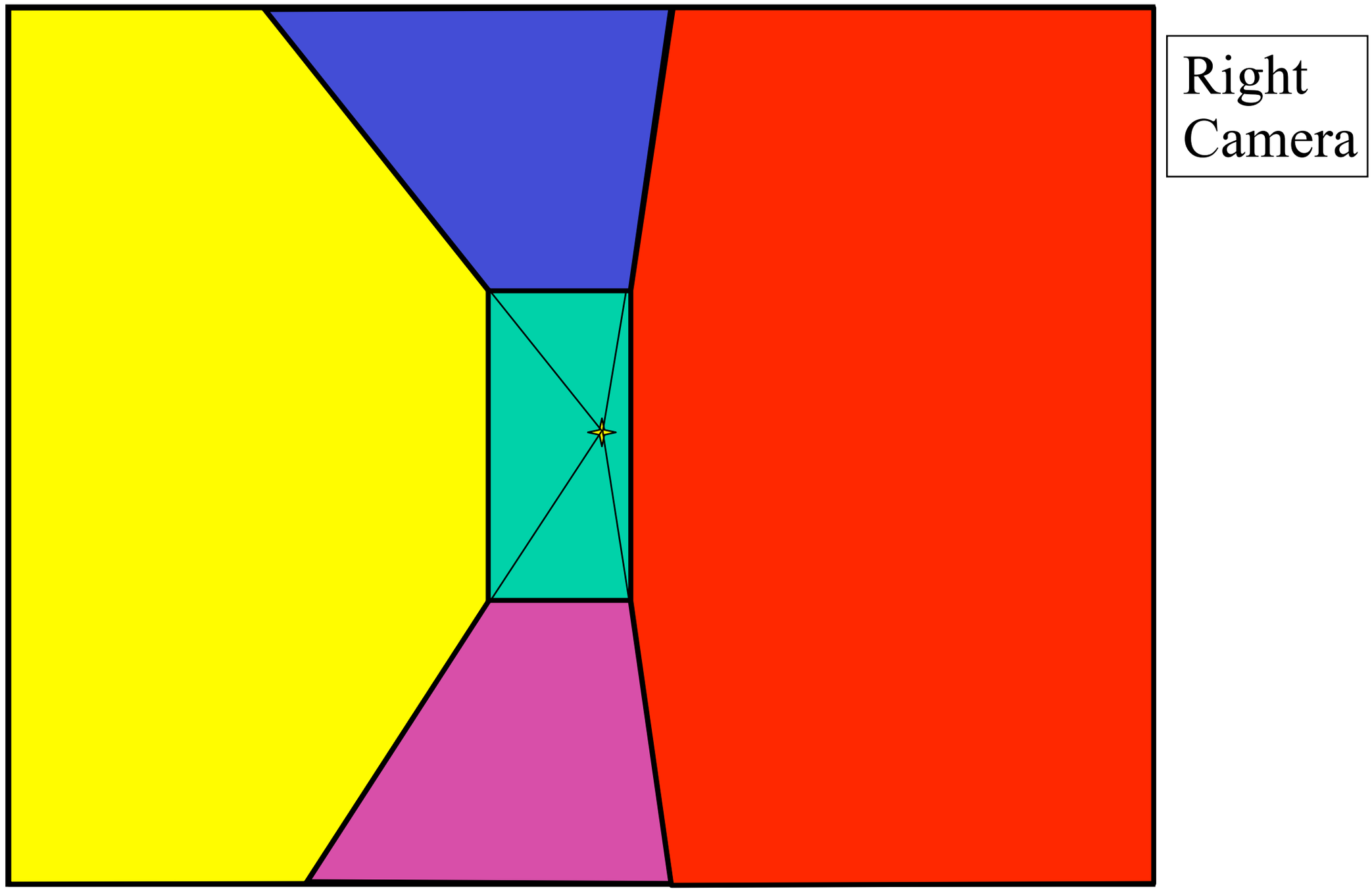
Another example of user input: vanishing point and back face of view volume are defined



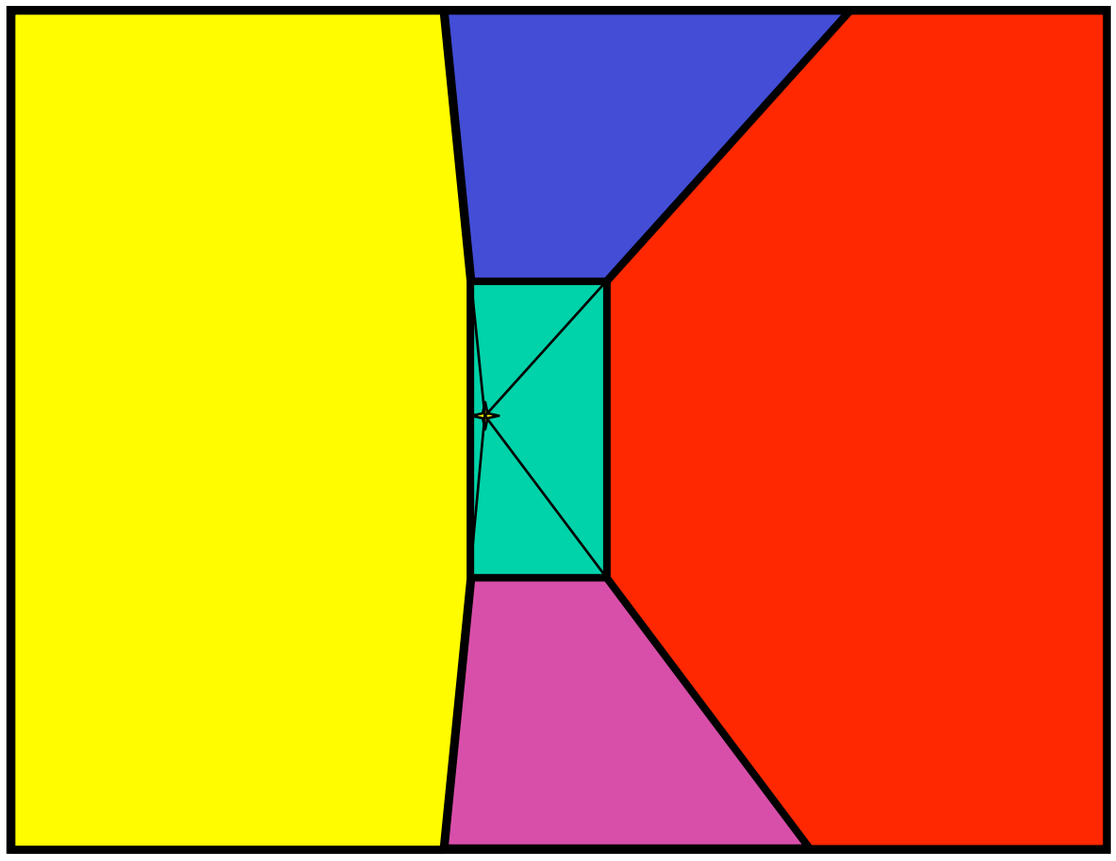
Another example of user input: vanishing point and back face of view volume are defined



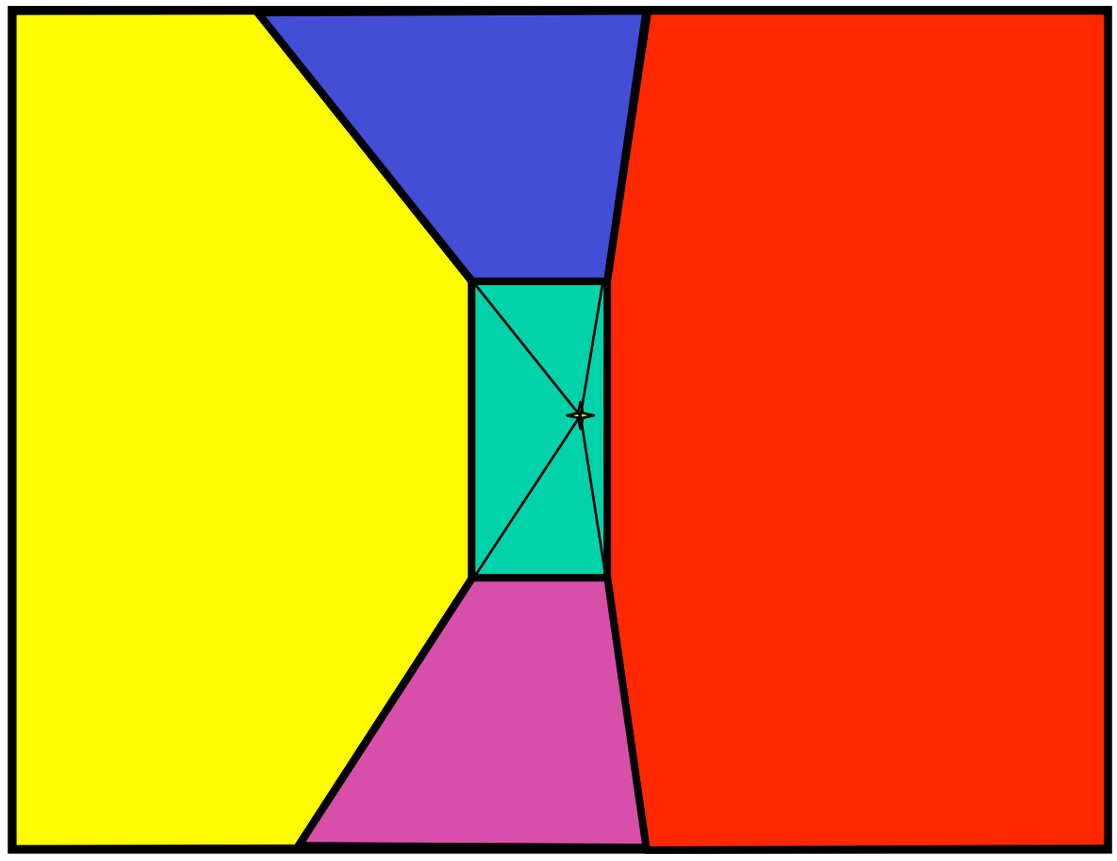
Another example of user input: vanishing point and back face of view volume are defined



Comparison of two camera placements – left and right. Corresponding subdivisions match view you would see if you looked down a hallway.



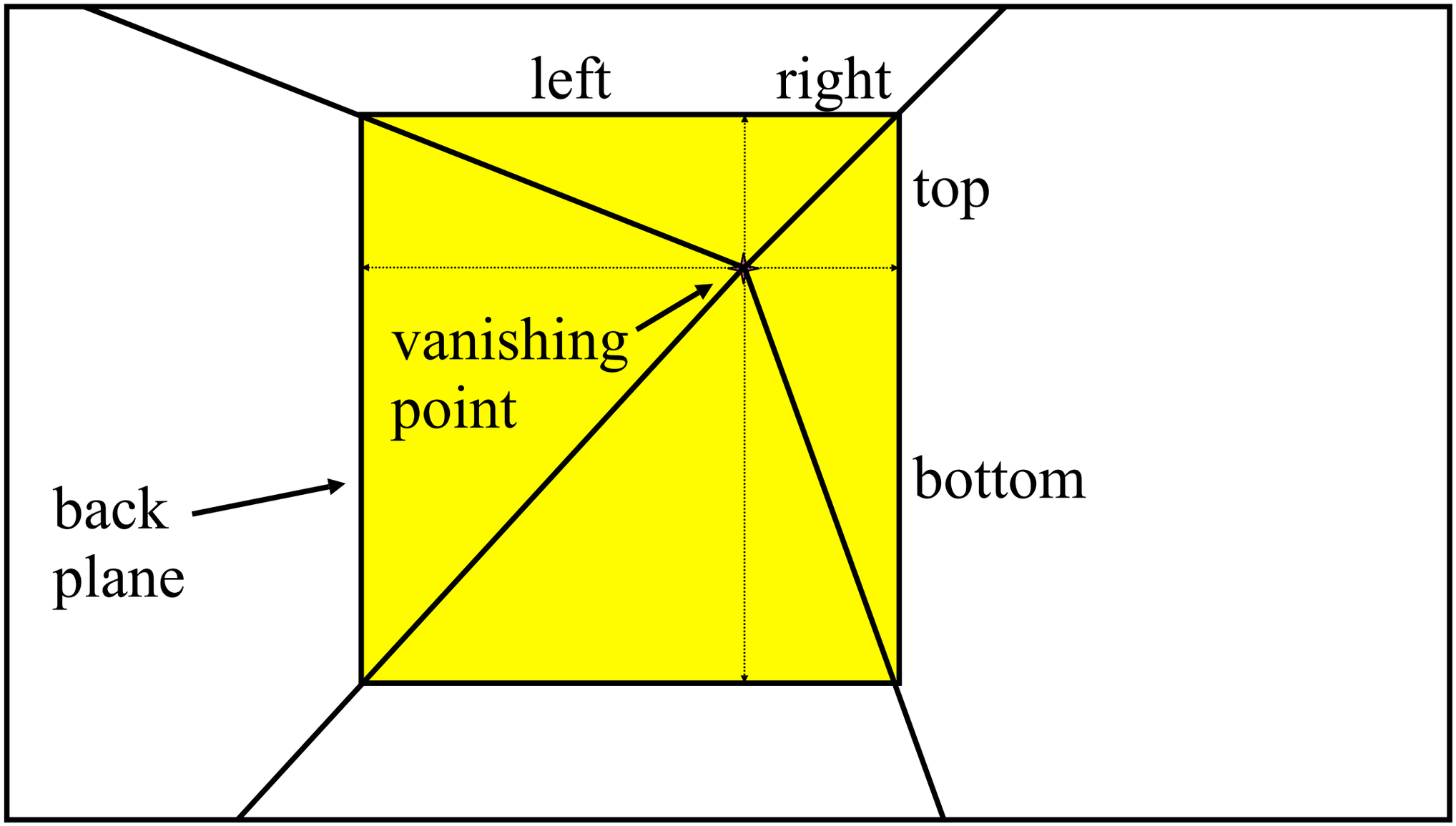
Left Camera



Right Camera

2D to 3D conversion

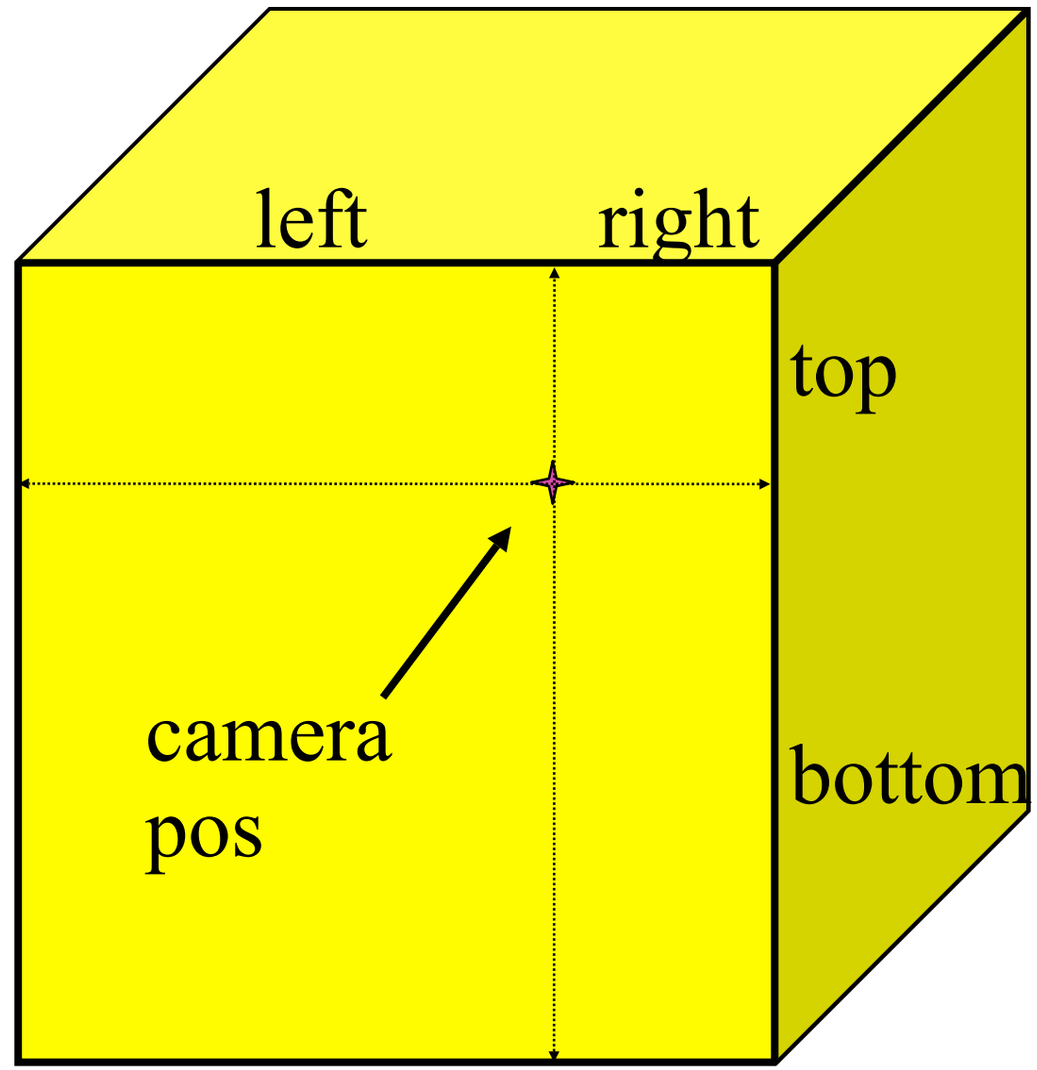
First, we can get ratios



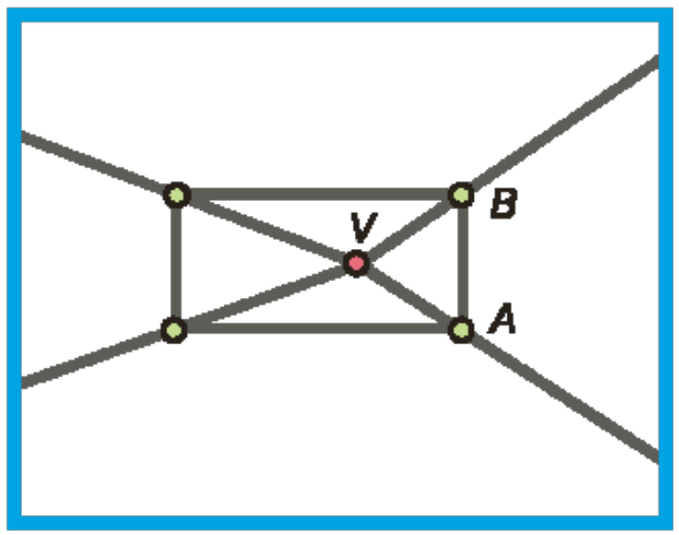
2D to 3D conversion

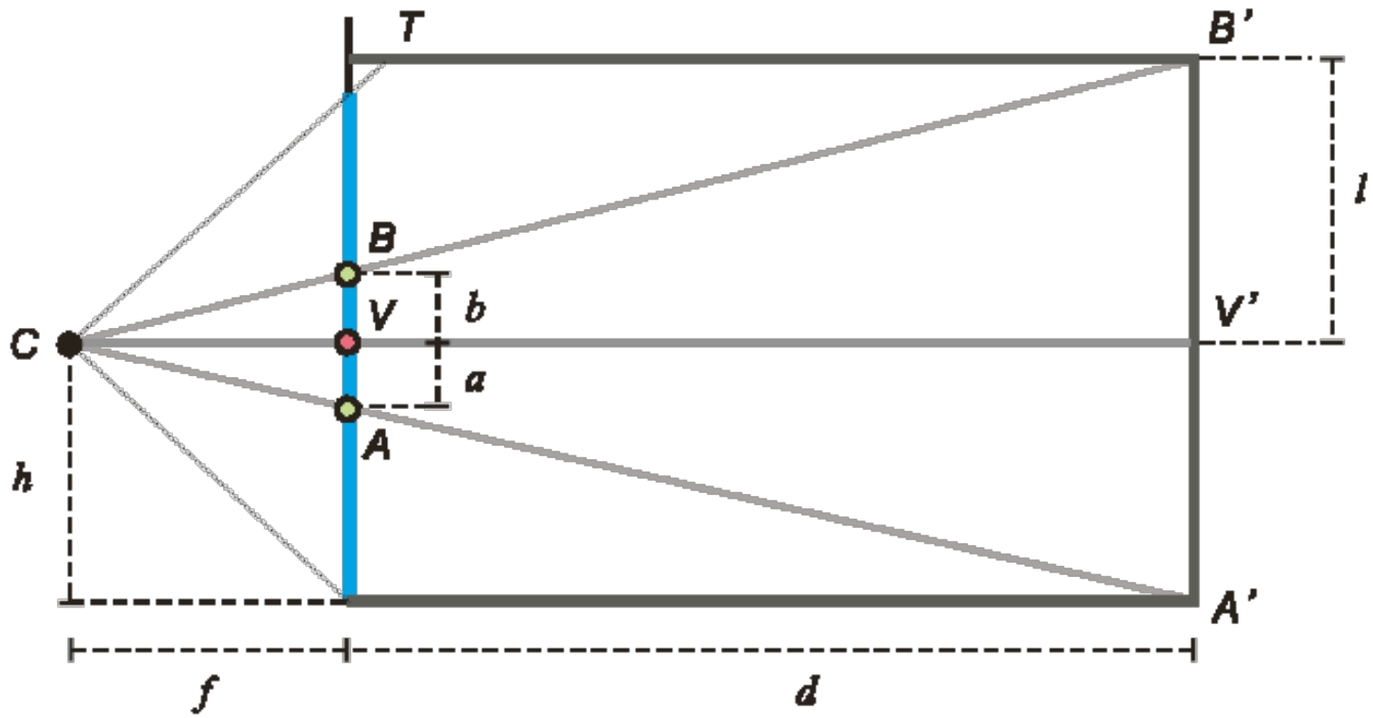
- Use top versus side ratio to determine relative height and width dimensions of box

- Left/right and top/bot ratios determine part of 3D camera placement



Depth of the box





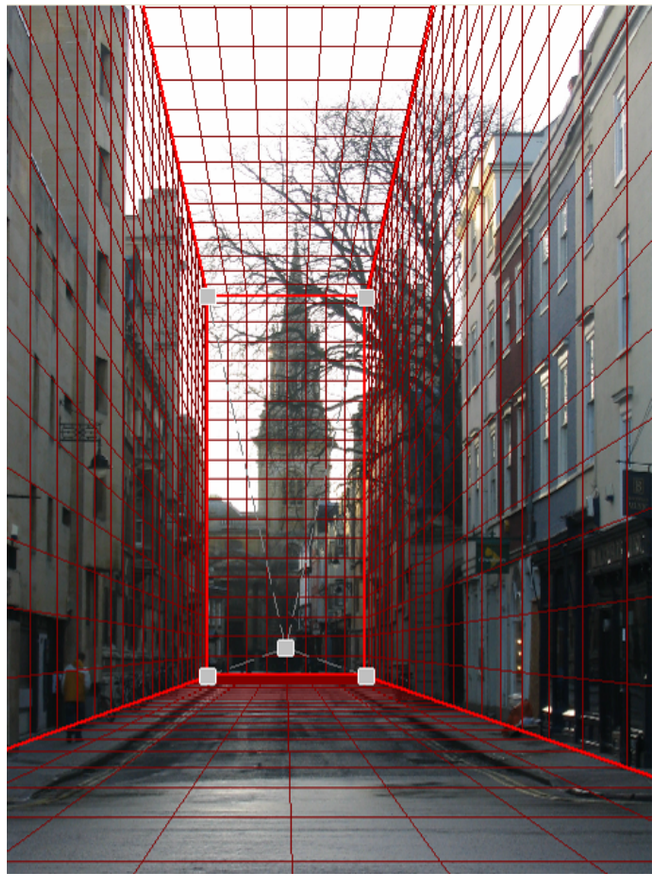
Can compute by similar triangles (CVA vs. CV'A')
Need to know focal length f (or FOV)

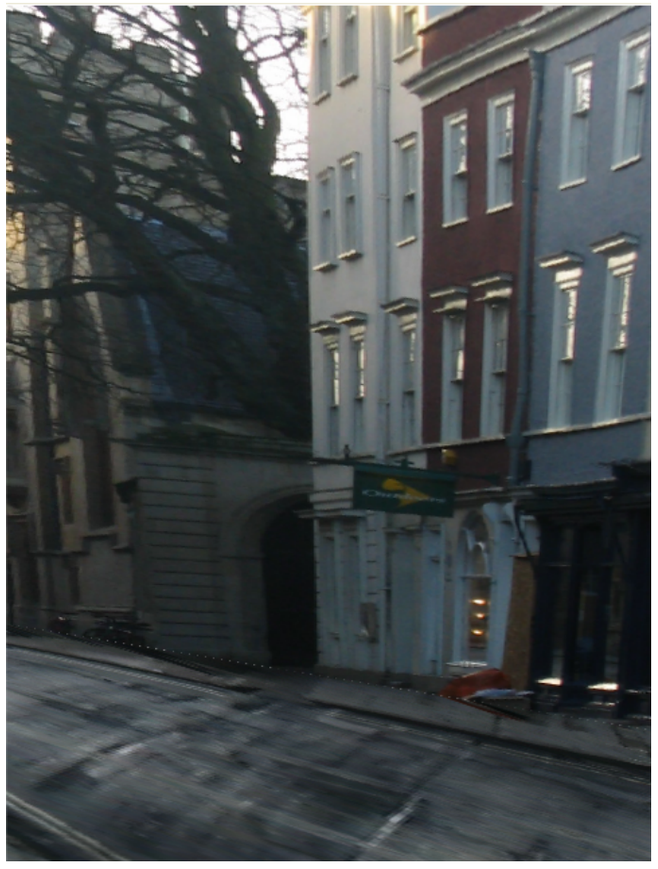
Note: can compute position on any object on the ground

DEMO

Now, we know the 3D geometry of the box

We can texture-map the box walls with texture from the image



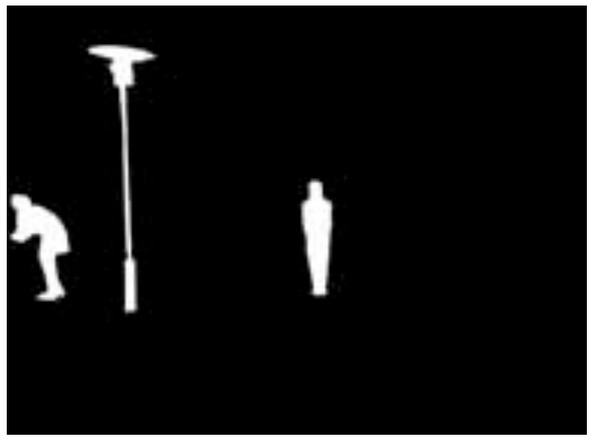


Foreground Objects

Use separate billboard for each

For this to work, three separate images used:

- Original image.
- Mask to isolate desired foreground images.
- Background with objects removed



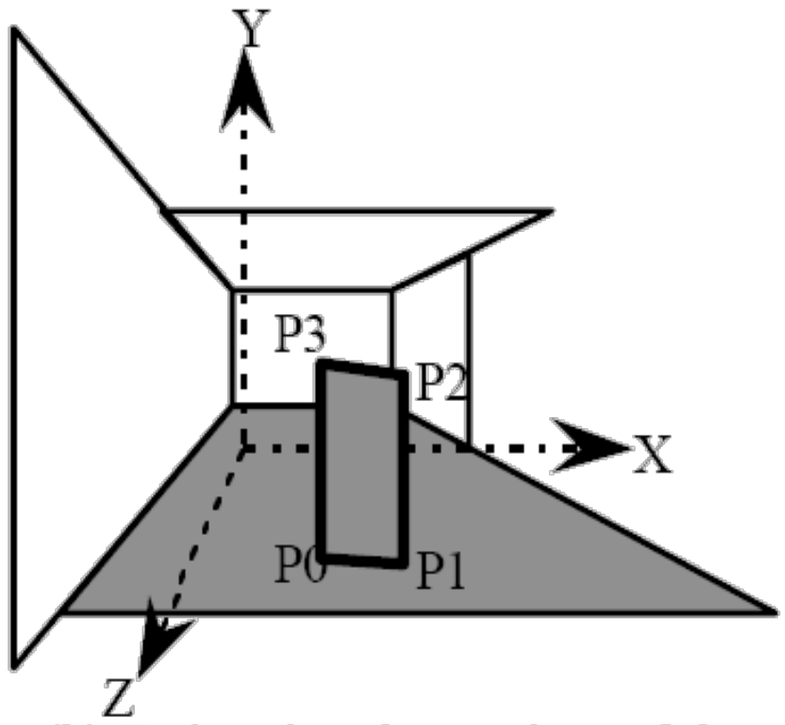
Foreground Objects

Add vertical rectangles for each foreground object

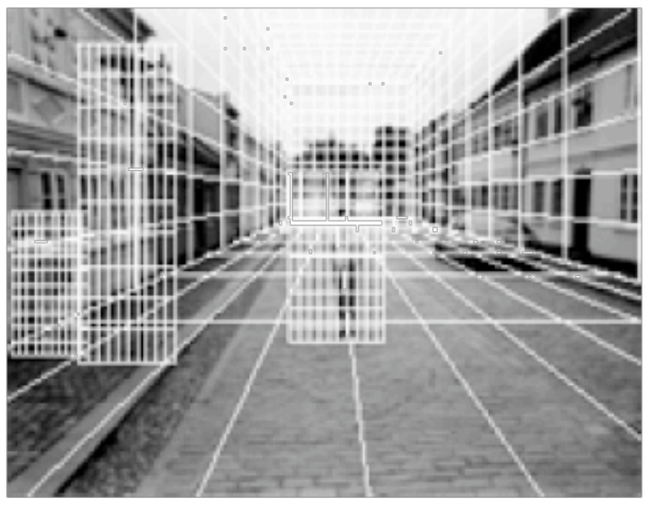
Can compute 3D coordinates P0, P1 since they are on known plane.

P2, P3 can be computed using similar triangles

(a) Specifying of a foreground object

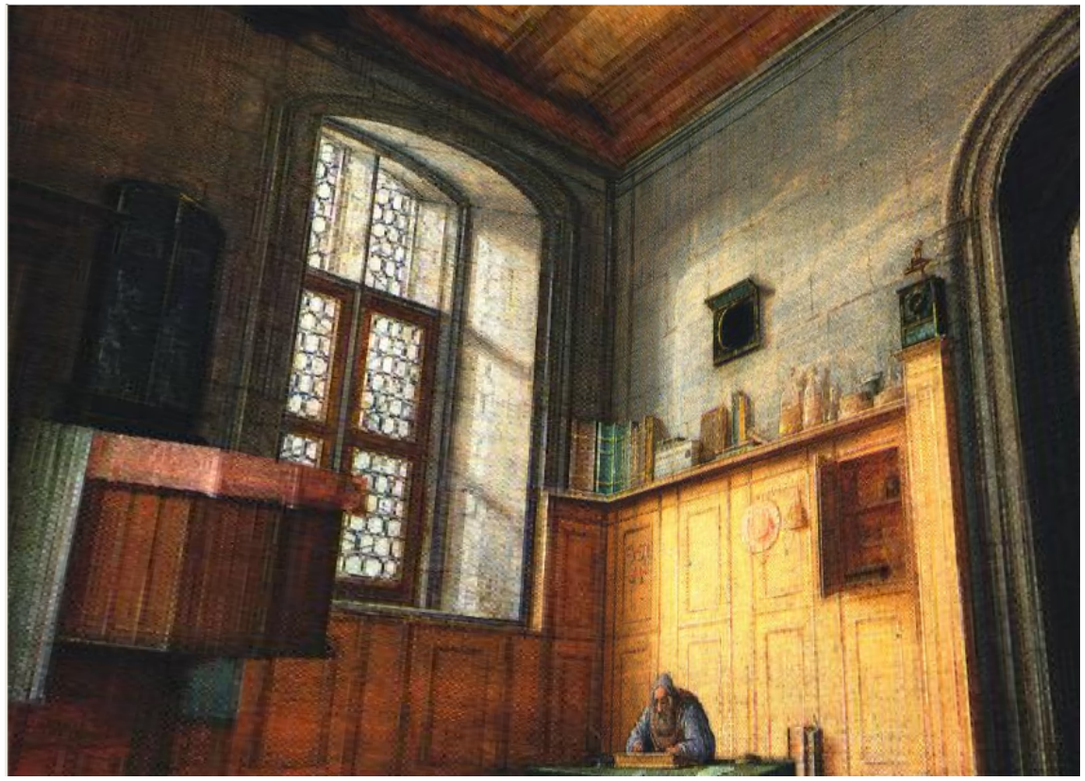


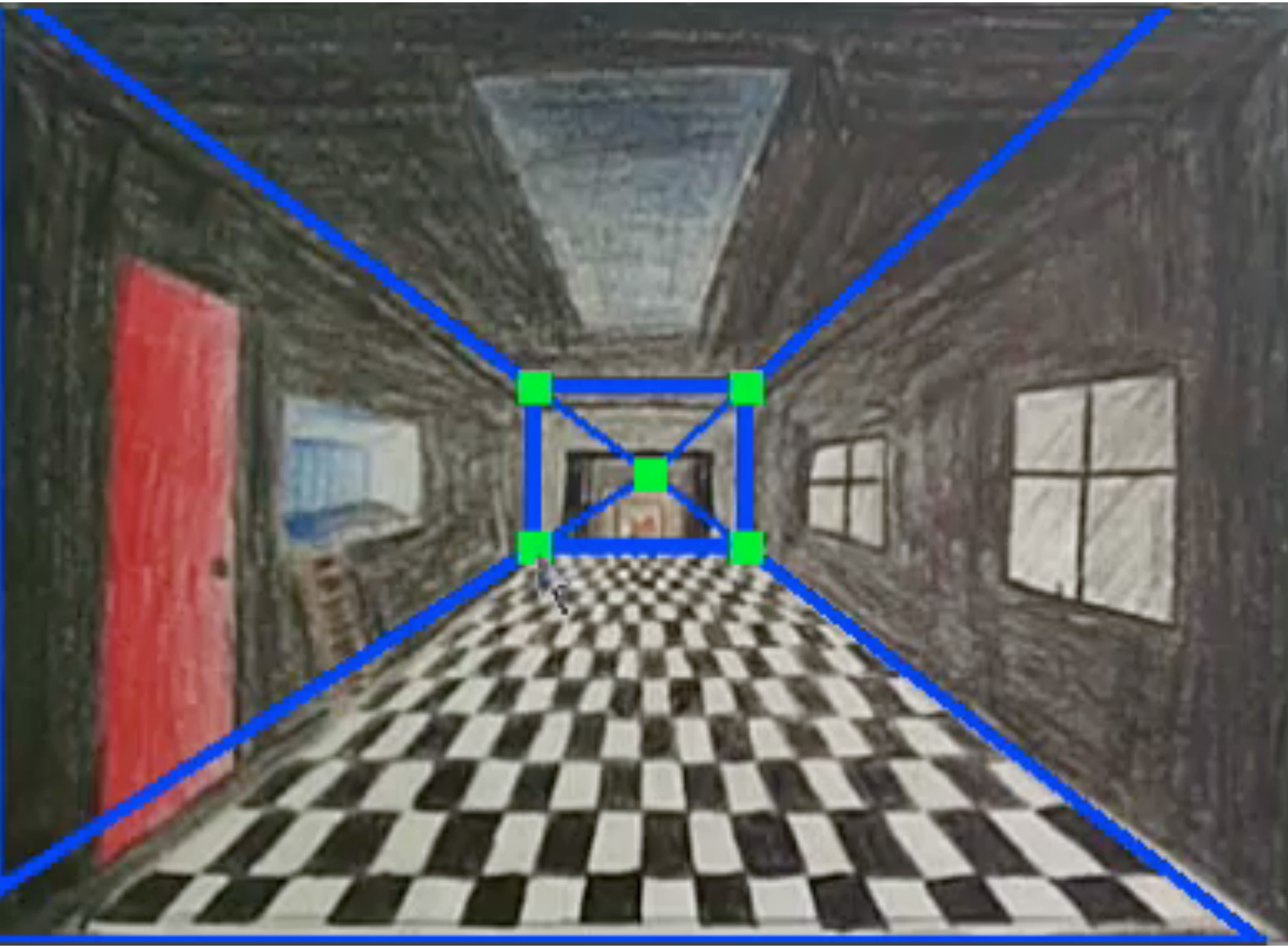
(b) Estimating the vertices of the foreground object model



(c) Three foreground object models

Foreground DEMO





Automatic Photo Pop-up

Derek Hoiem Alexei A. Efros Martial Hebert

Carnegie Mellon University




Goals

Pop-up Book

Automatic Photo Pop-up

Derek Hoiem Alexei A. Efros Martial Hebert

Carnegie Mellon University



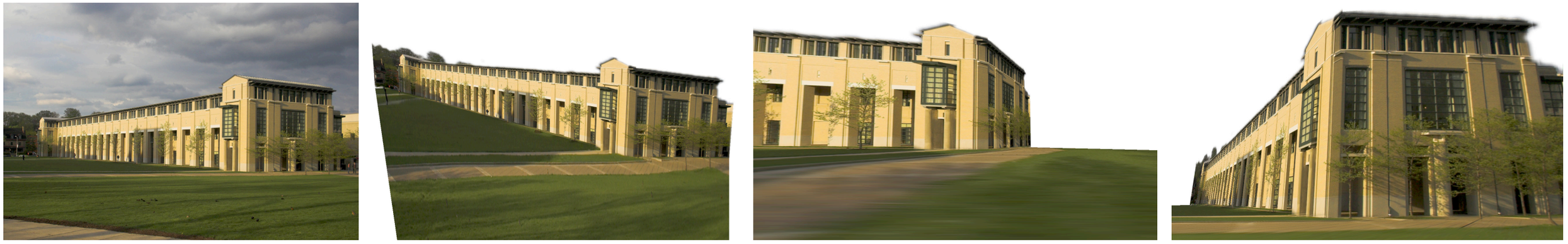

Goals

Pop-up Book

Automatic Photo Pop-up

Derek Hoiem Alexei A. Efros Martial Hebert

Carnegie Mellon University



Goals

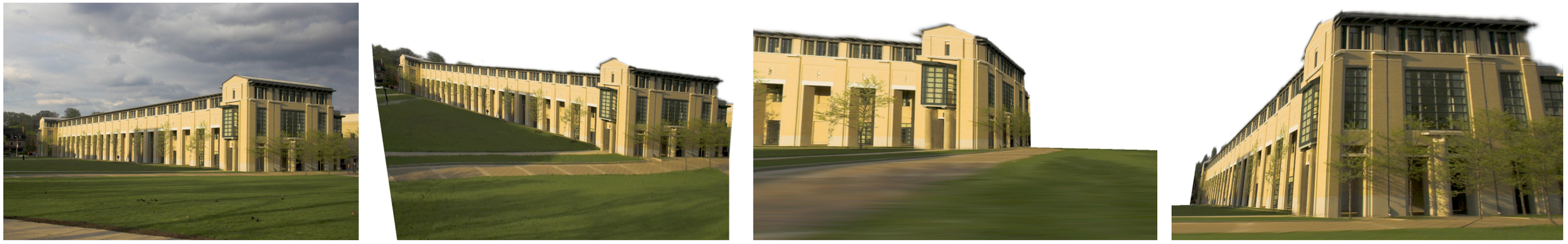
- Simple, piecewise planar models

Pop-up Book

Automatic Photo Pop-up

Derek Hoiem Alexei A. Efros Martial Hebert

Carnegie Mellon University



Goals

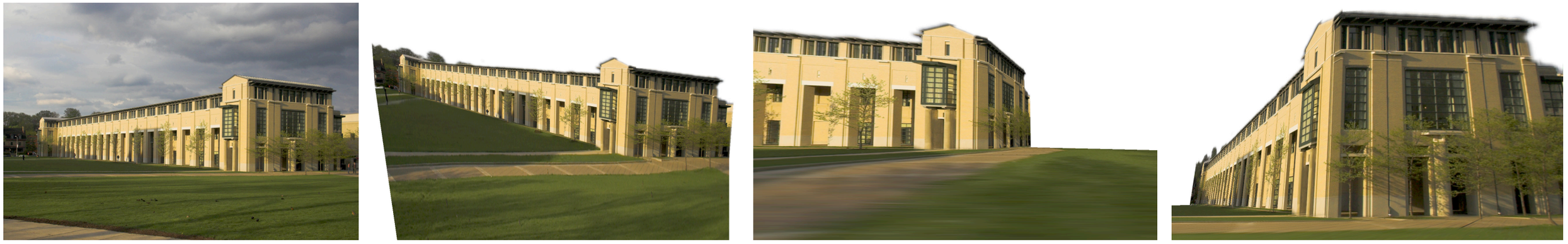
- Simple, piecewise planar models

Pop-up Book

Automatic Photo Pop-up

Derek Hoiem Alexei A. Efros Martial Hebert

Carnegie Mellon University



Goals

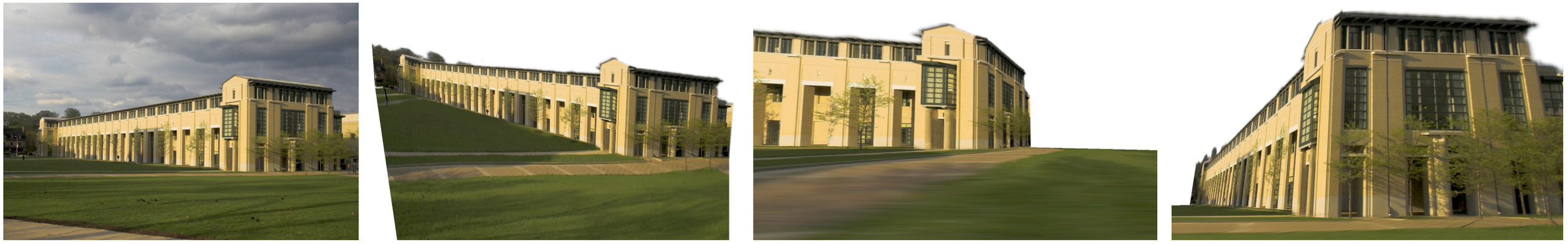
- Simple, piecewise planar models
- Outdoor scenes

Pop-up Book

Automatic Photo Pop-up

Derek Hoiem Alexei A. Efros Martial Hebert

Carnegie Mellon University



Goals

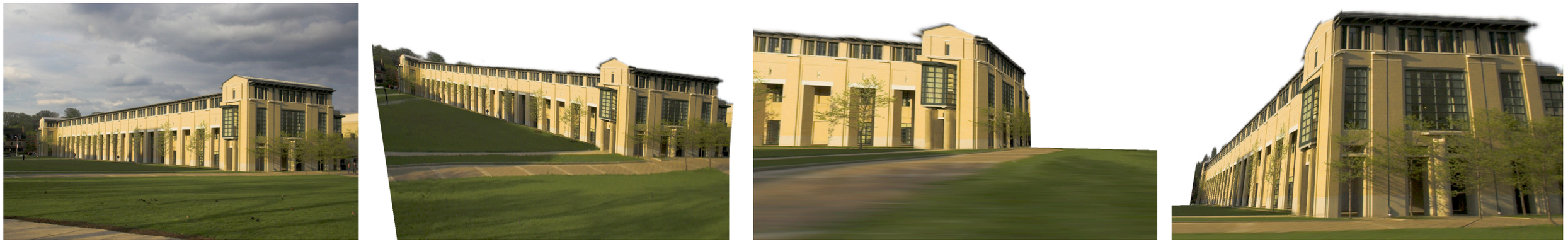
- Simple, piecewise planar models
- Outdoor scenes

Pop-up Book

Automatic Photo Pop-up

Derek Hoiem Alexei A. Efros Martial Hebert

Carnegie Mellon University



Goals

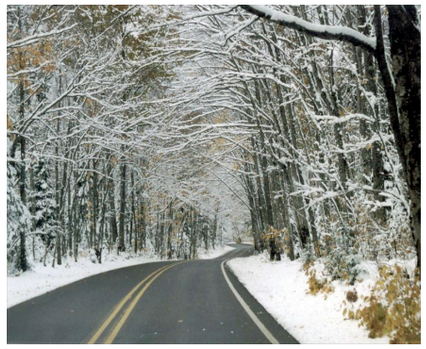
- Simple, piecewise planar models
- Outdoor scenes
- Doesn’t need to work all the time (~35%)

Pop-up Book

Our Approach: Learning

- Learn structure of the world and appearance-based models of geometry












Overview

Input

Image

Learned Models

Overview

Input

Image

Learned Models

Geometric Labels

Ground

Vertical

Sky

Overview

Input Geometric Labels Cut’n’Fold

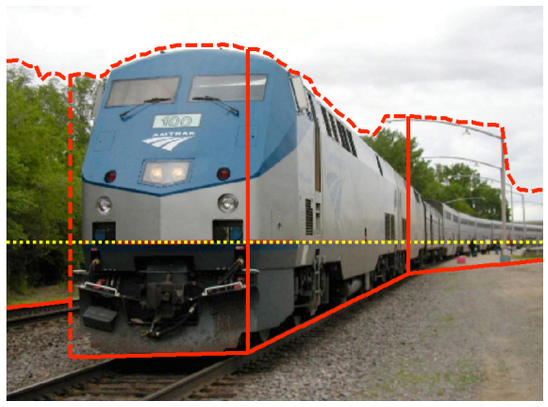
Image

Learned Models

Ground

Vertical

Sky



Overview

Input | Geometric Labels | Cut'n'Fold | 3D Model

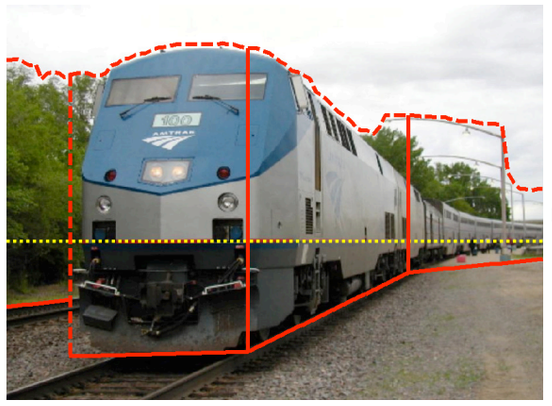
Image

Learned Models

Ground

Vertical

Sky



Geometric Cues

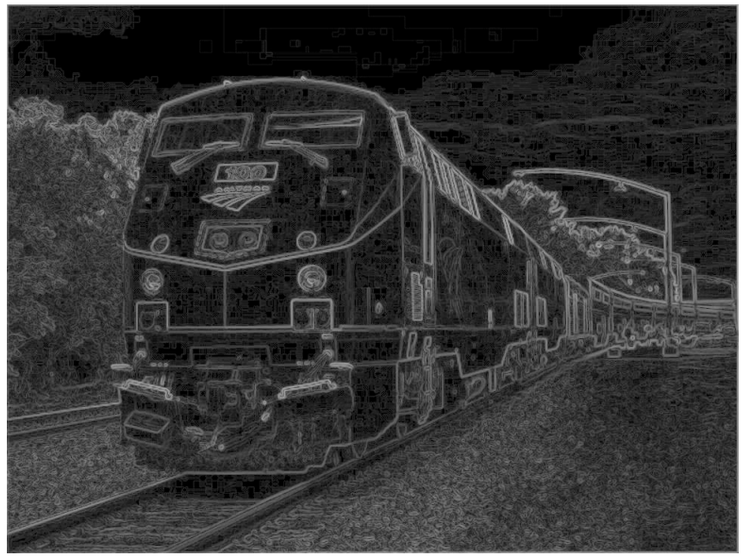
Geometric Cues

Color

Geometric Cues

Color

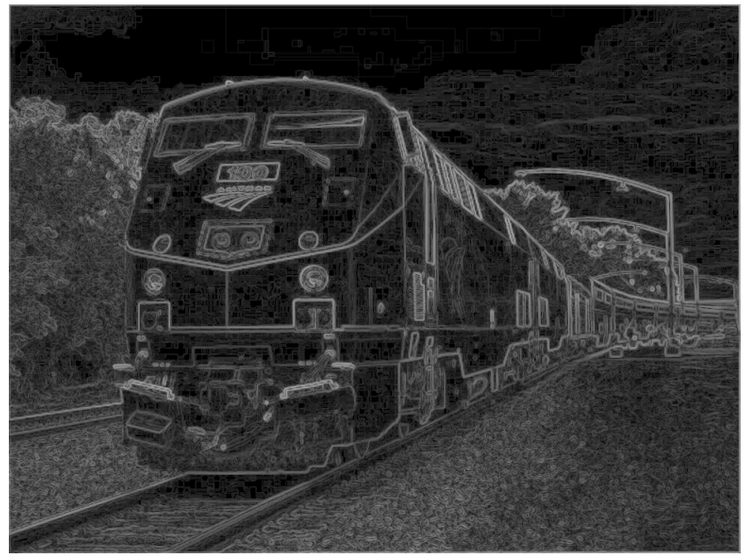
Texture



Geometric Cues

Color

Texture

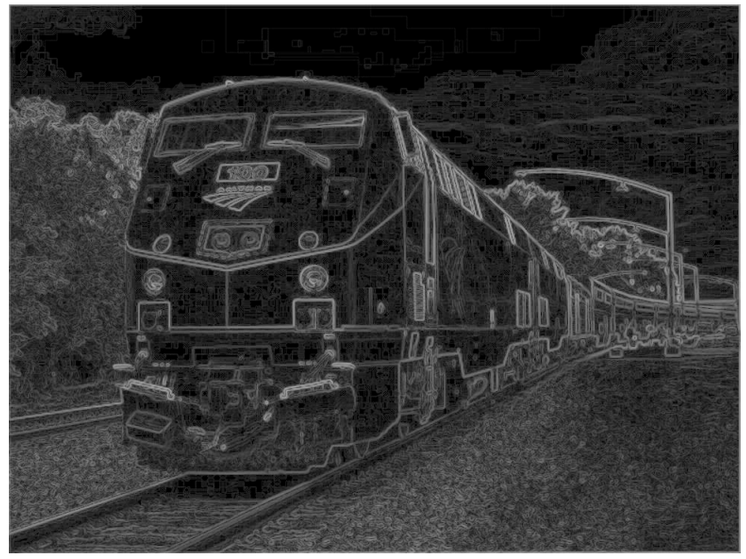


Location

Geometric Cues

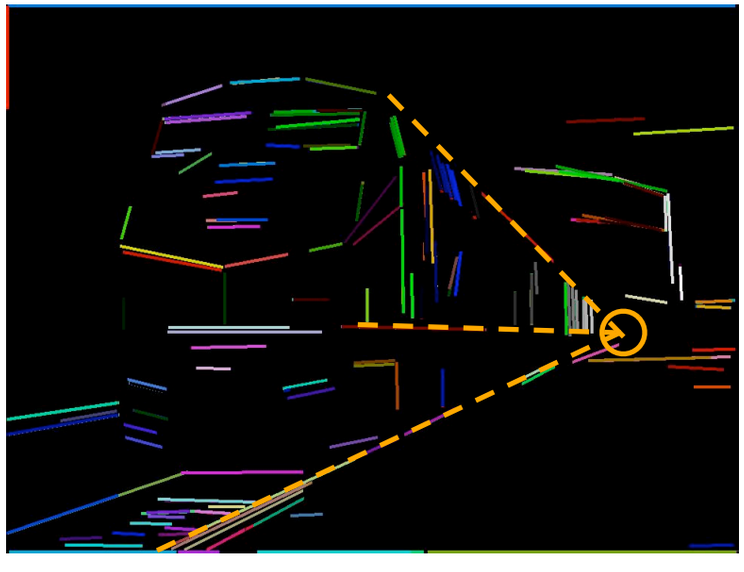
Color

Texture



Location

Perspective



Robust Spatial Support

RGB Pixels

Superpixels

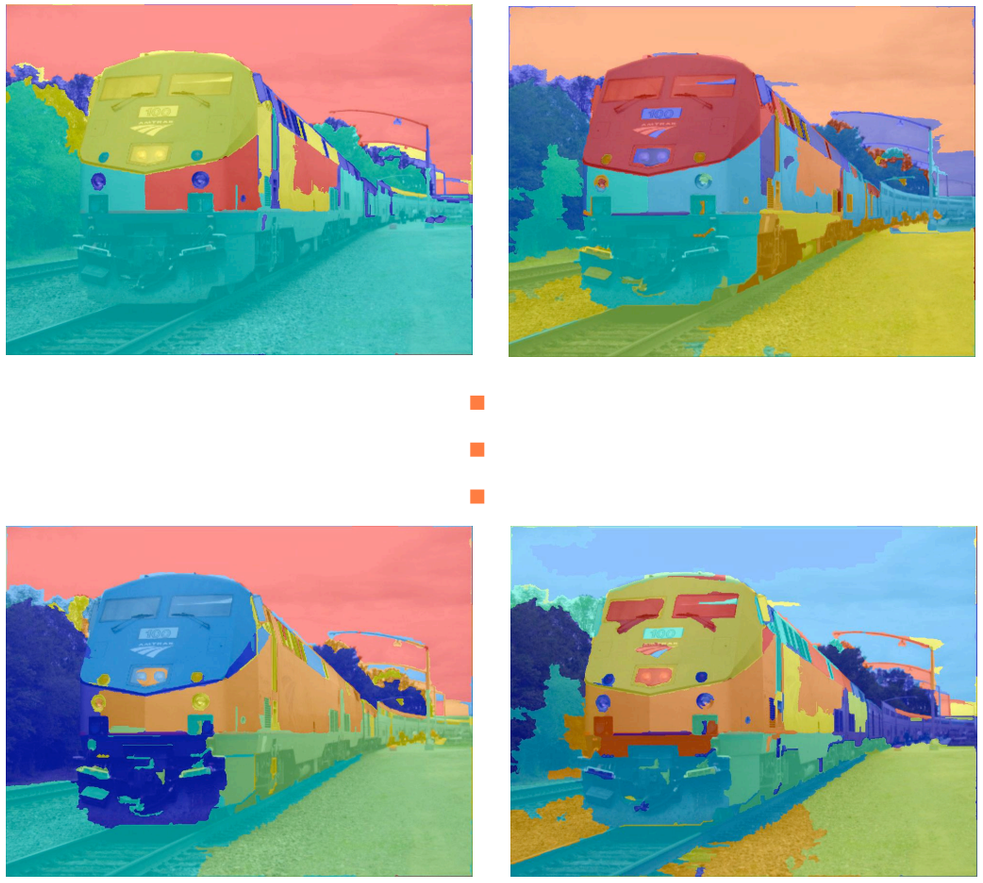
[Felzenszwalb and Huttenlocher 2004]

- Safe oversegmentation of image
- Better but not still not enough spatial support

Multiple Segmentations

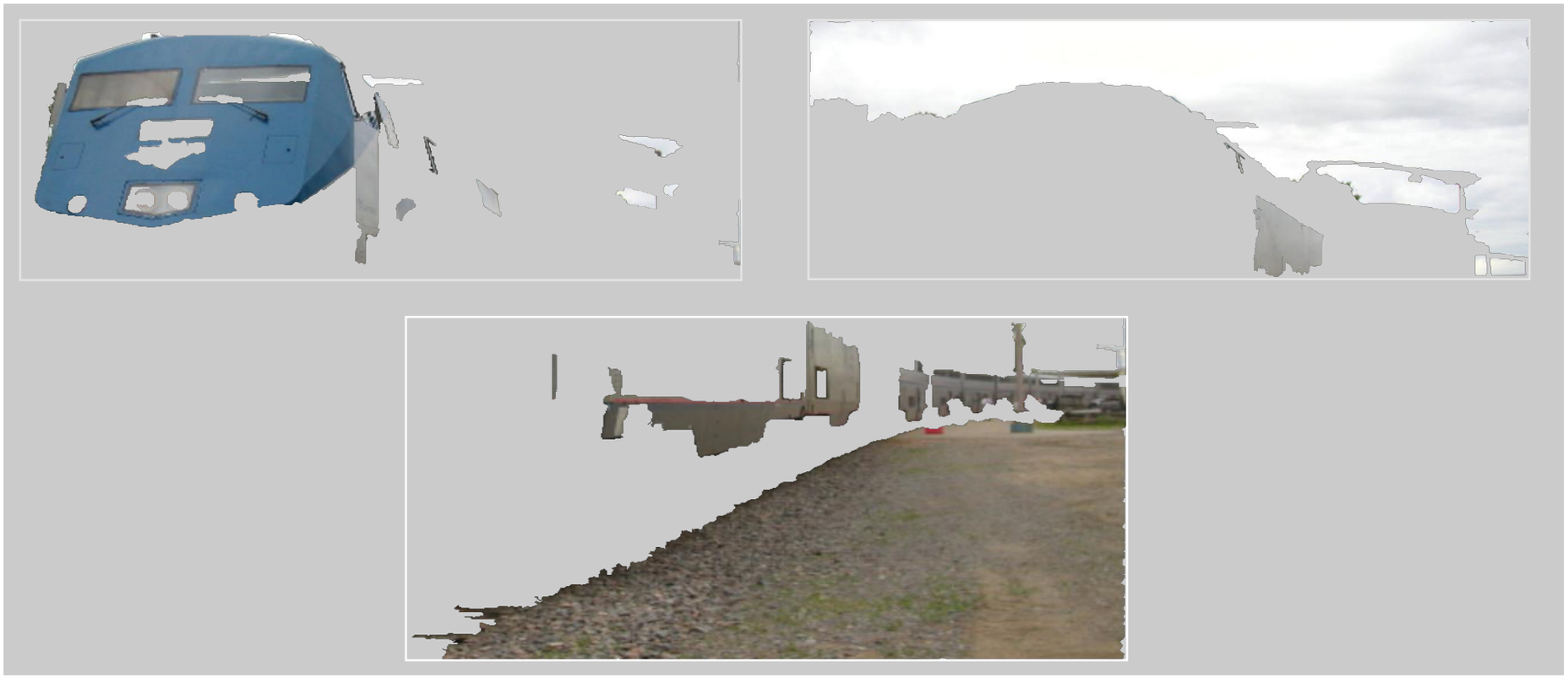
Superpixels

Multiple Segmentations



- Group superpixels likely to be from the same surface into segments
- Single segmentation unreliable
- Create multiple segmentations

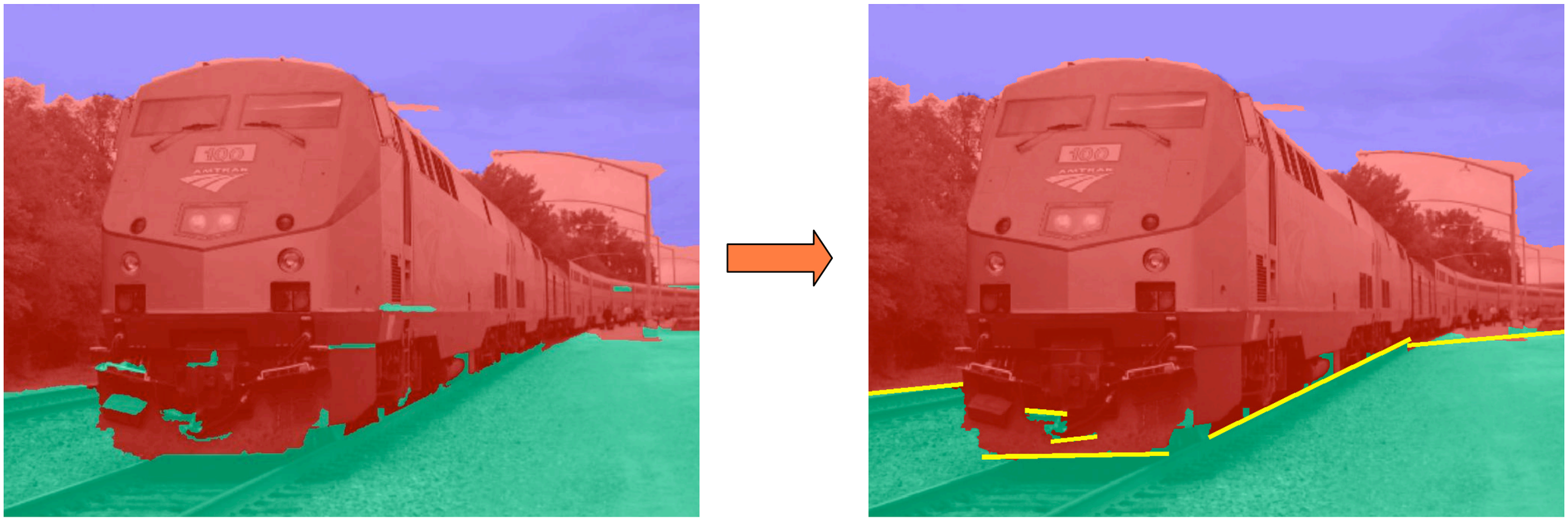
Learning Appearance-based Geometry



- All geometric cues available
- Does this segment correspond to a single surface? (*homogeneity likelihood*)
- If so, what is the geometry of that surface? (*label likelihood*)

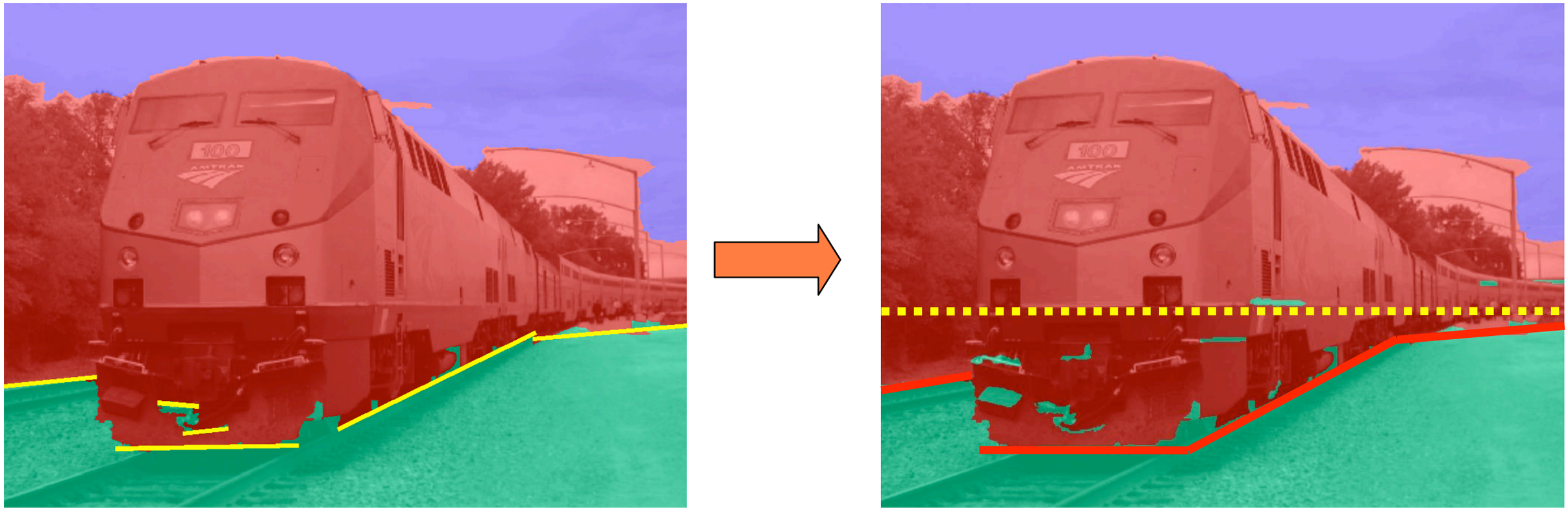
Feature Descriptions	Num	Used
Color	**15**	**15**
C1. RGB values: mean	3	3
C2. HSV values: conversion from mean RGB values	3	3
C3. Hue: histogram (5 bins) and entropy	6	6
C4. Saturation: histogram (3 bins) and entropy	3	3
Texture	**29**	**13**
T1. DOOG Filters: mean abs response	12	3
T2. DOOG Filters: mean of variables in T1	1	0
T3. DOOG Filters: id of max of variables in T1	1	1
T4. DOOG Filters: (max - median) of variables in T1	1	1
T5. Textons: mean abs response	12	7
T6. Textons: max of variables in T5	1	0
T7. Textons: (max - median) of variables in T5	1	1
Location and Shape	**12**	**10**
L1. Location: normalized x and y, mean	2	2
L2. Location: norm. x and y, 10^{th} and 90^{th} percentile	4	4
L3. Location: norm. y wrt horizon, 10^{th} and 90^{th} pctl	2	2
L4. Shape: number of superpixels in constellation	1	1
L5. Shape: number of sides of convex hull	1	0
L6. Shape: $num\ pixels/area(convex\ hull)$	1	1
L7. Shape: whether the constellation region is contiguous	1	0
3D Geometry	**35**	**28**
G1. Long Lines: total number in constellation region	1	1
G2. Long Lines: % of nearly parallel pairs of lines	1	1
G3. Line Intersection: hist. over 12 orientations, entropy	13	11
G4. Line Intersection: % right of center	1	1
G5. Line Intersection: % above center	1	1
G6. Line Intersection: % far from center at 8 orientations	8	4
G7. Line Intersection: % very far from center at 8 orientations	8	5
G8. Texture gradient: x and y "edginess" (T2) center	2	2

Cutting and Folding



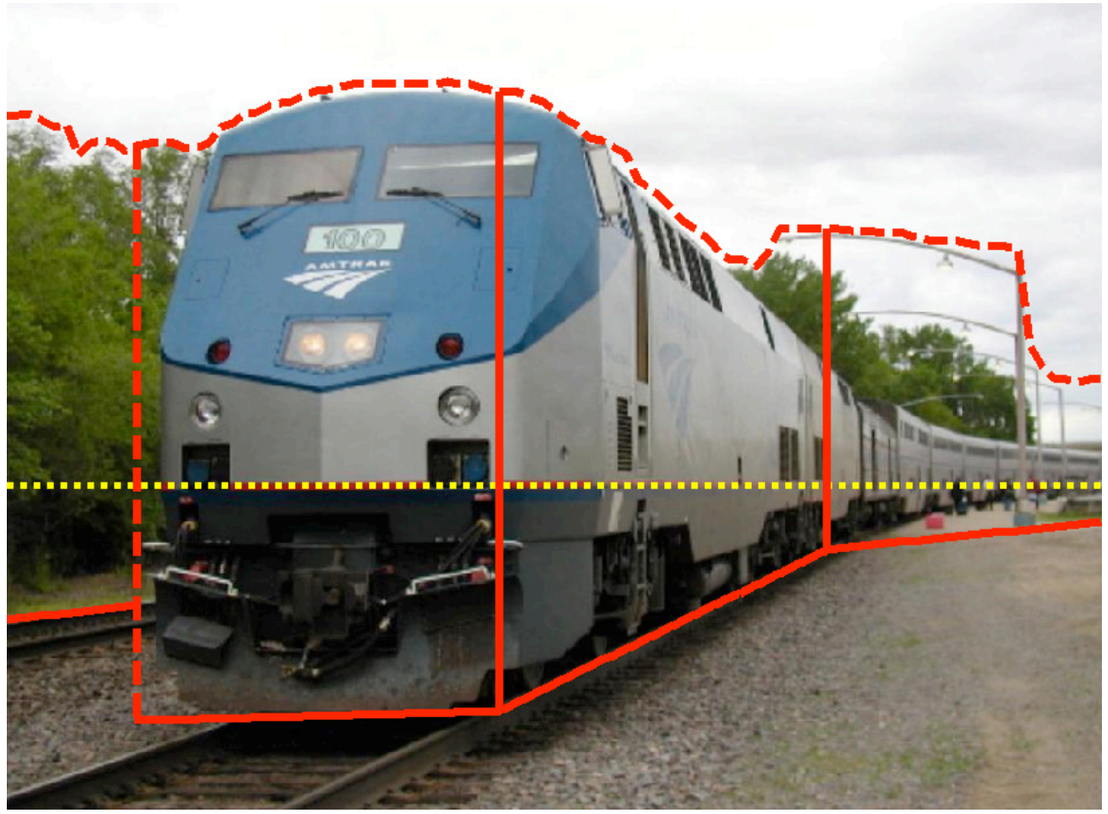
- Fit ground-vertical boundary
  - Iterative Hough transform

Cutting and Folding



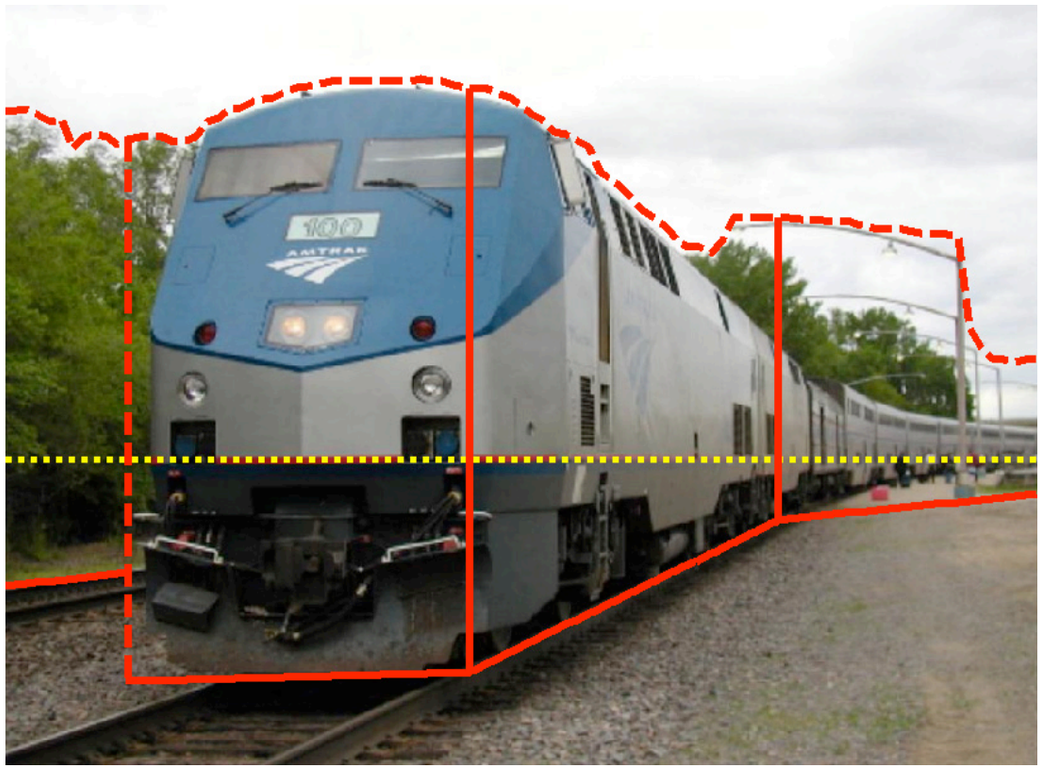
- Form polylines from boundary segments
  - Join segments that intersect at slight angles
  - Remove small overlapping polylines
- Estimate horizon position from perspective cues

Cutting and Folding



- ``Fold'' along polylines and at corners
- ``Cut'' at ends of polylines and along vertical-sky boundary

Cutting and Folding



- Construct 3D model
- Texture map

Failures

Labeling Errors

Conclusion

Conclusion

- First system to automatically recover 3D scene from single image!

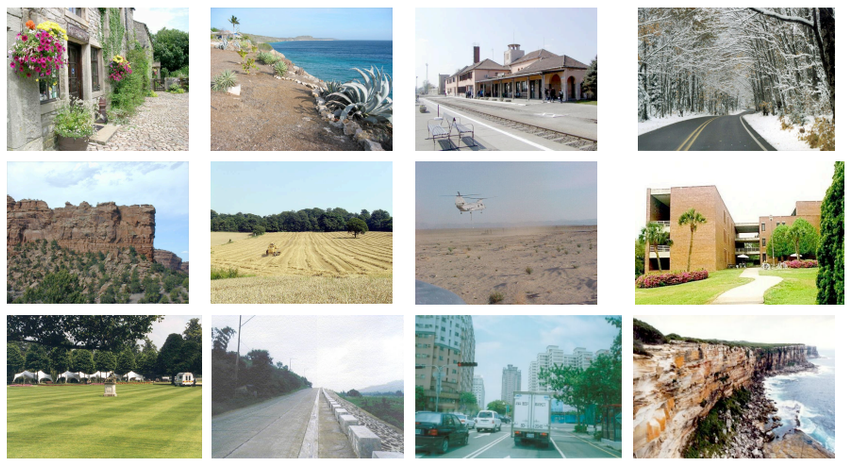
Conclusion

- First system to automatically recover 3D scene from single image!

Conclusion

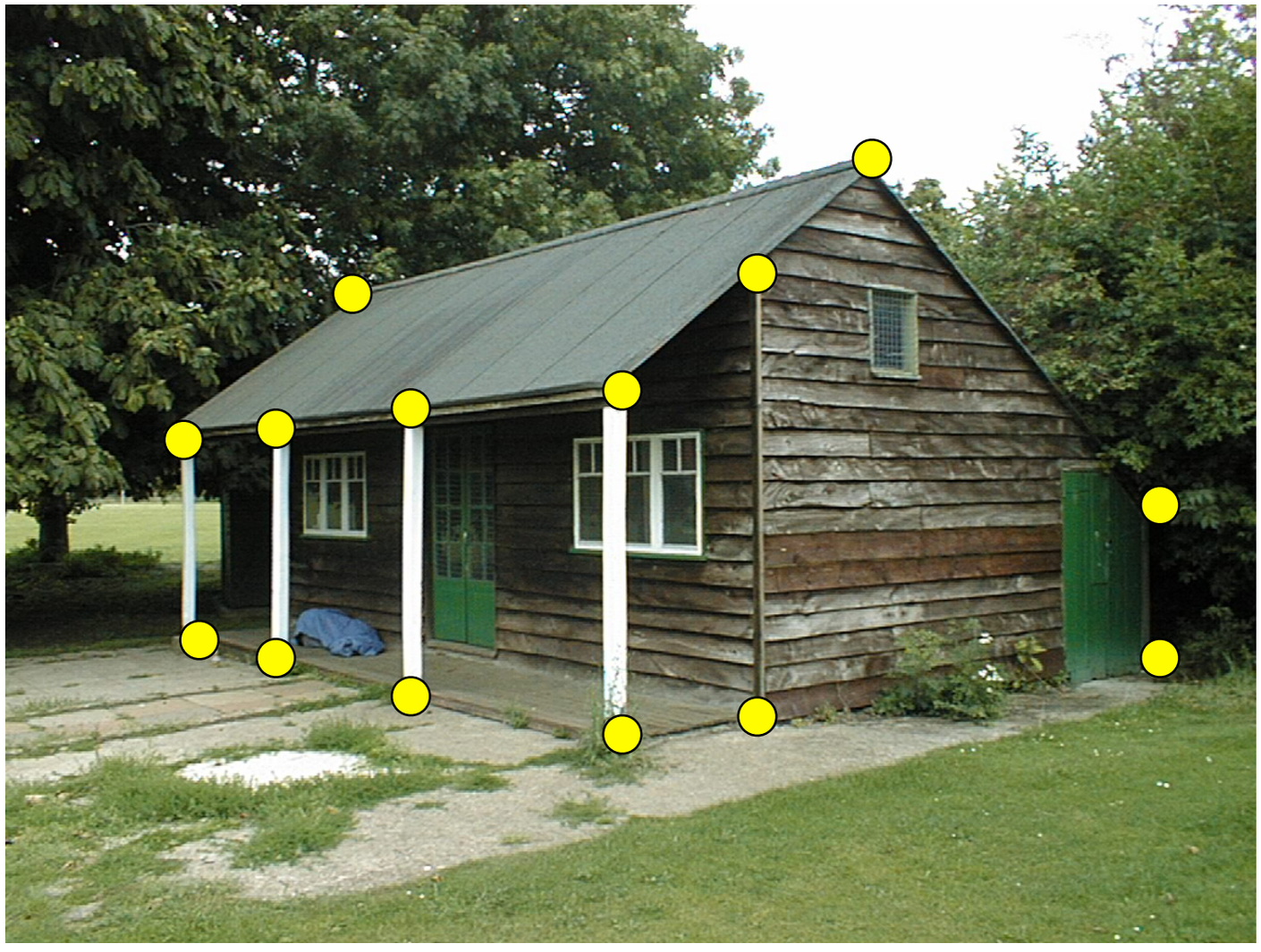
- First system to automatically recover 3D scene from single image!

- *Learn* statistics of our world from training images



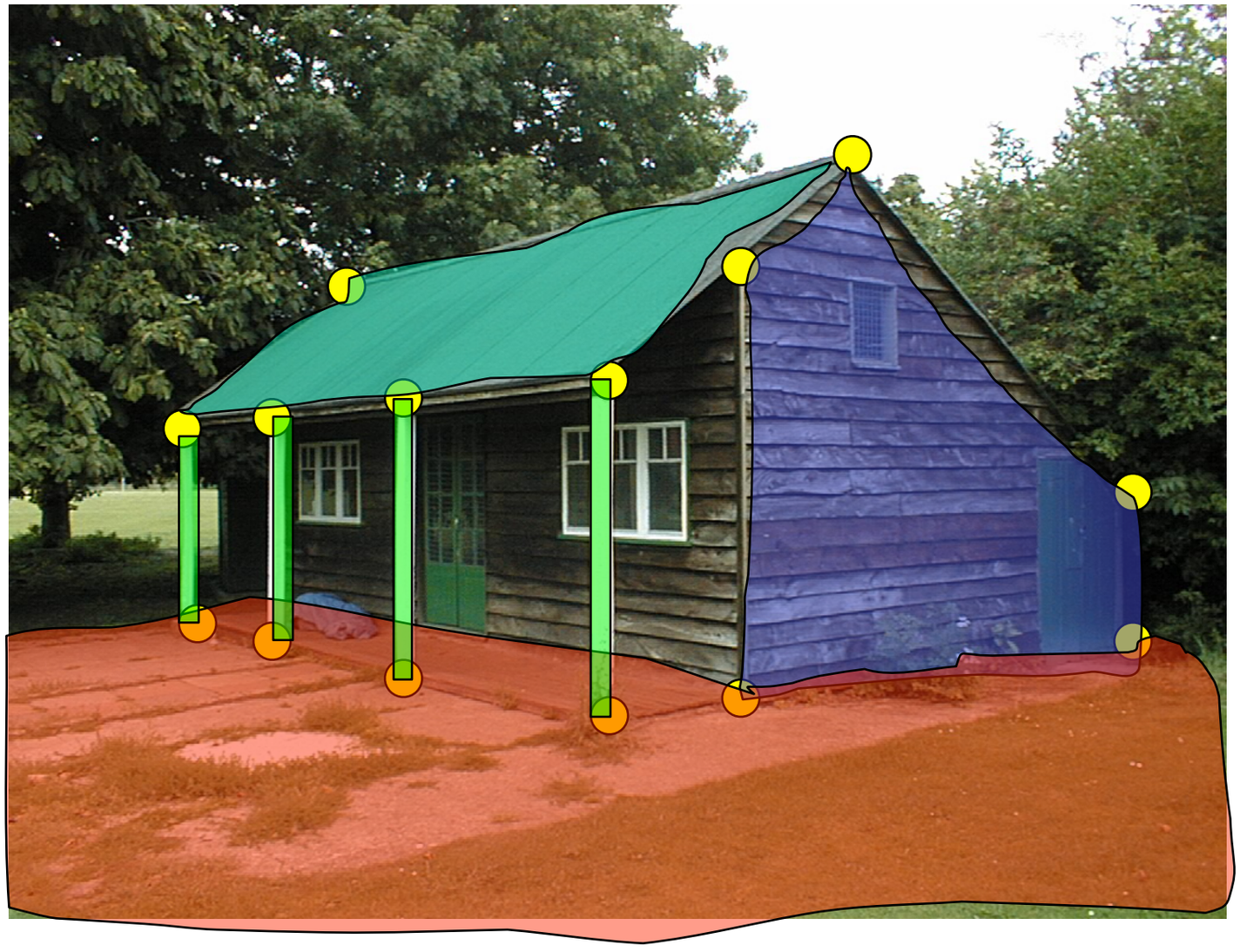
How can we model more complex scene?

How can we model more complex scene?



1. Find world coordinates (X,Y,Z) for a few points

How can we model more complex scene?



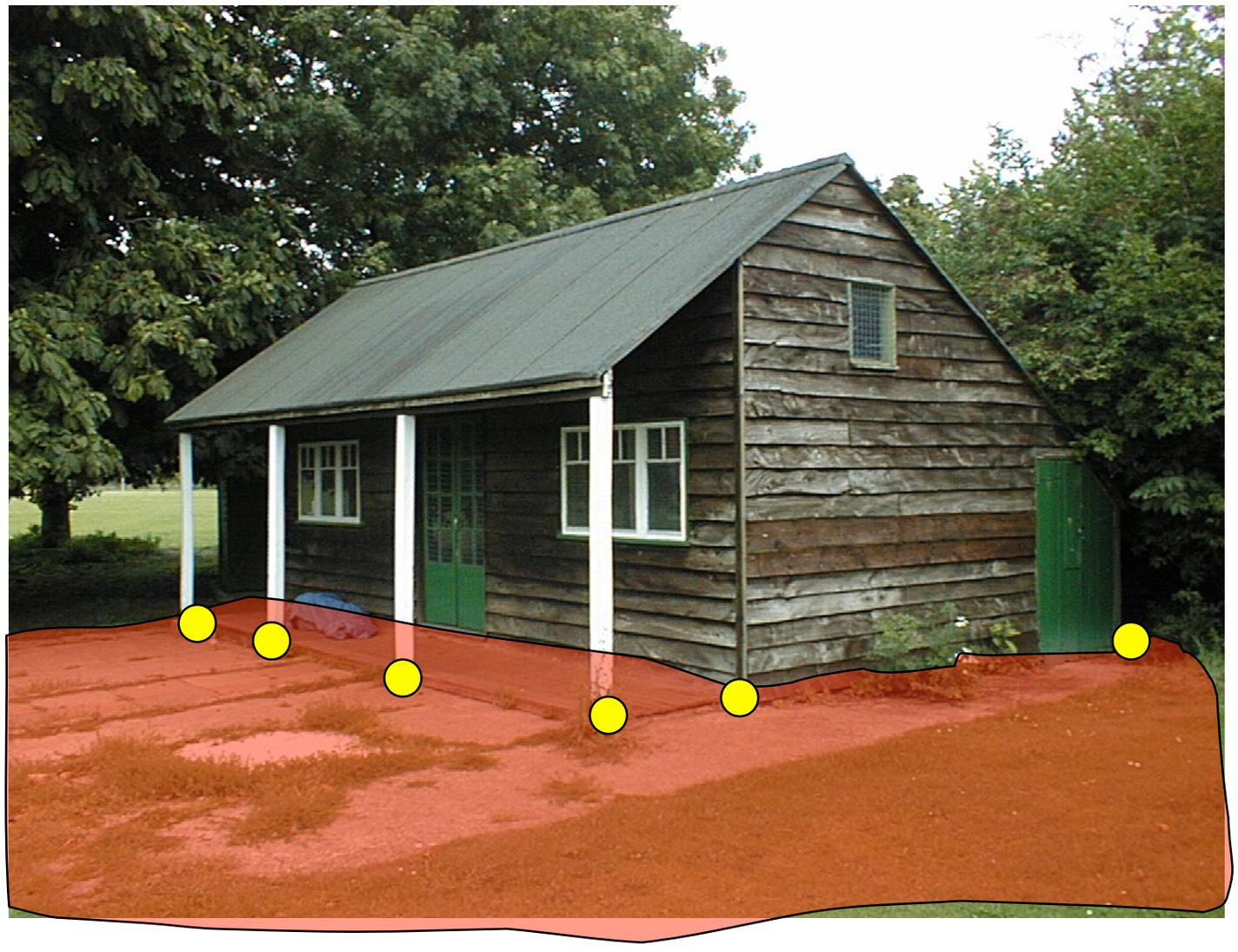
1. Find world coordinates (X,Y,Z) for a few points
2. Connect the points with planes to model geometry
   - Texture map the planes

Finding world coordinates (X,Y,Z)

Finding world coordinates (X,Y,Z)

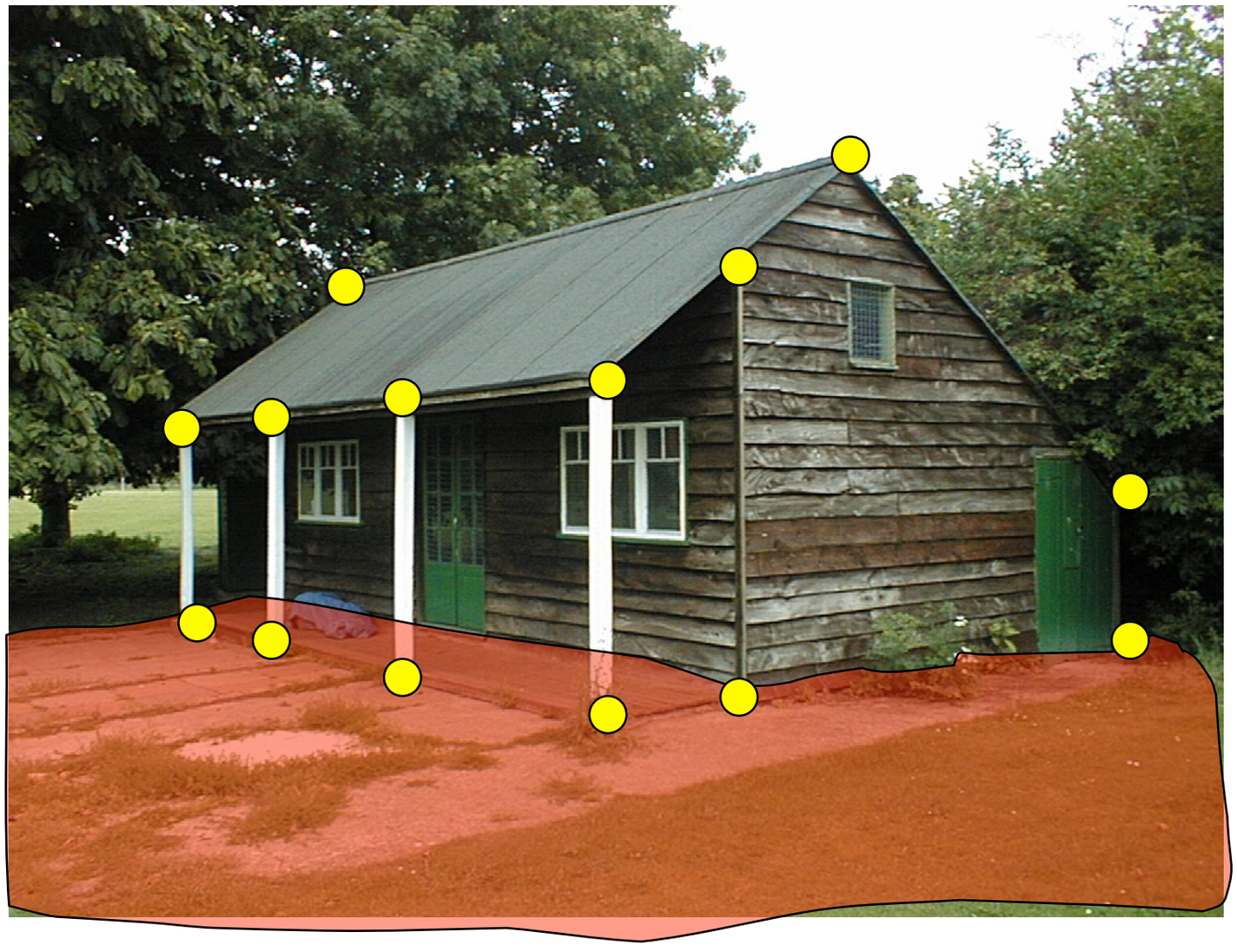
1. Define the ground plane (Z=0)

Finding world coordinates (X,Y,Z)



1. Define the ground plane (Z=0)
2. Compute points (X,Y,0) on that plane

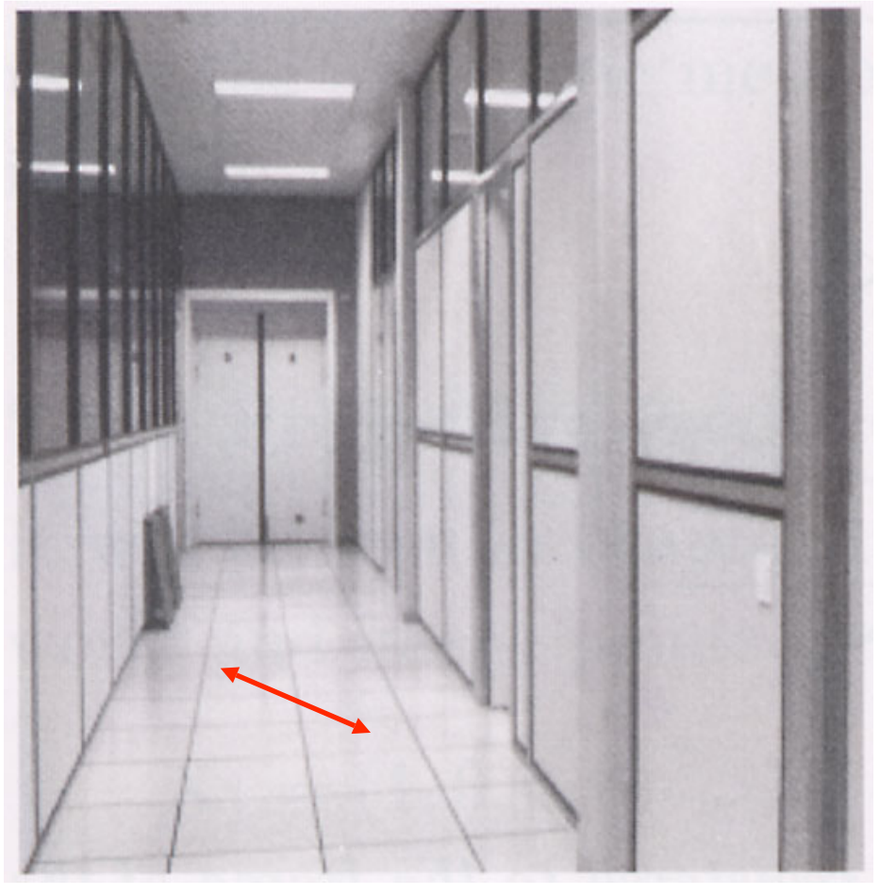
Finding world coordinates (X,Y,Z)



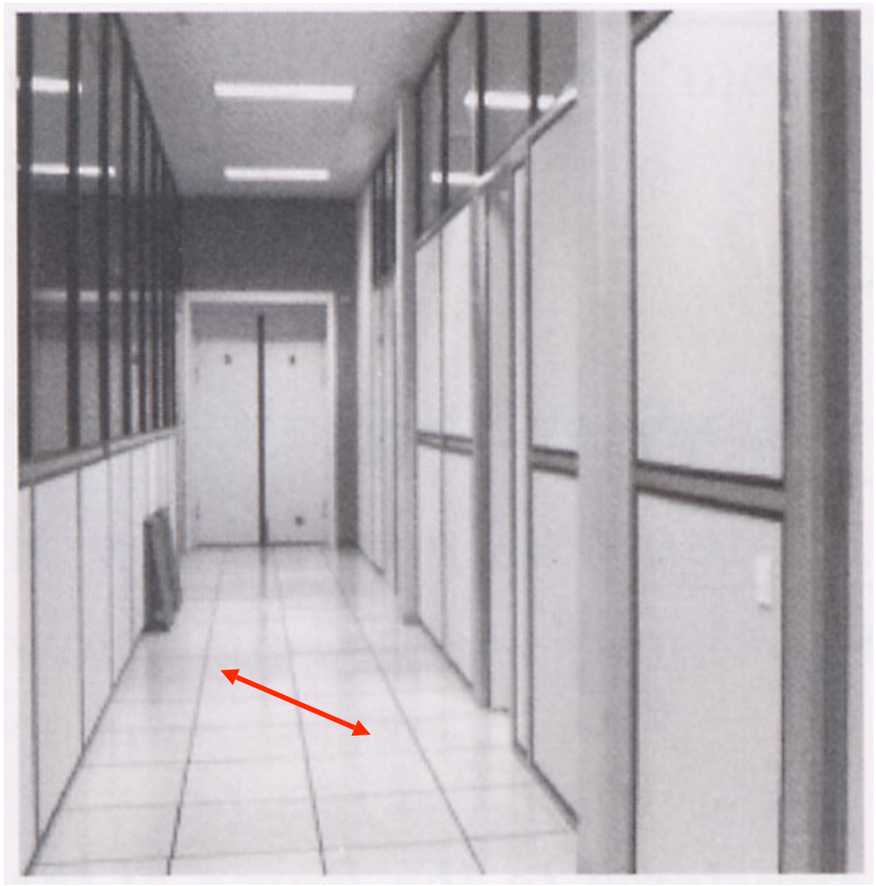
1. Define the ground plane (Z=0)
2. Compute points (X,Y,0) on that plane
3. Compute the *heights* Z of all other points

Measurements on planes

Measurements on planes

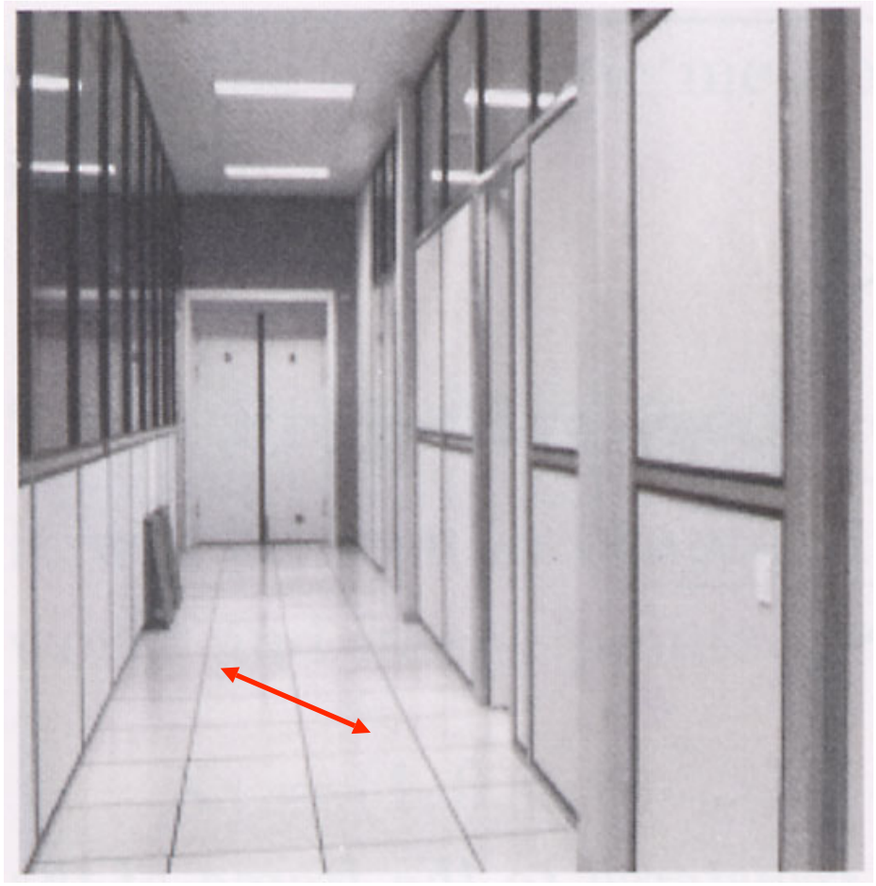


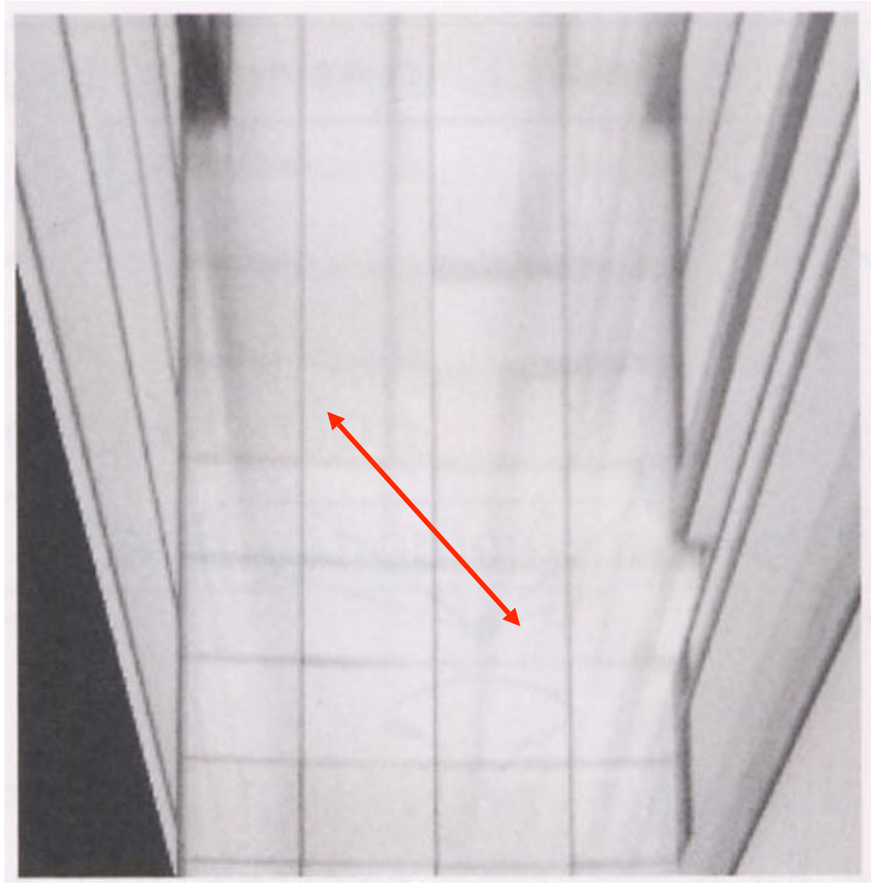
Measurements on planes



Approach: unwarp, then measure

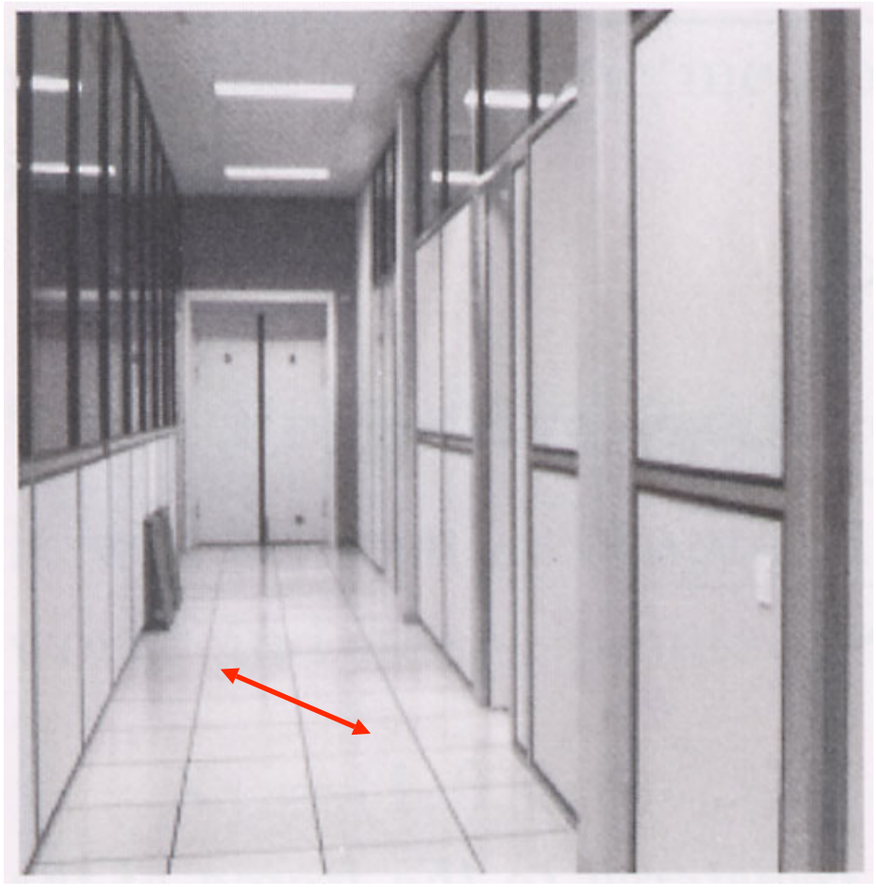
Measurements on planes

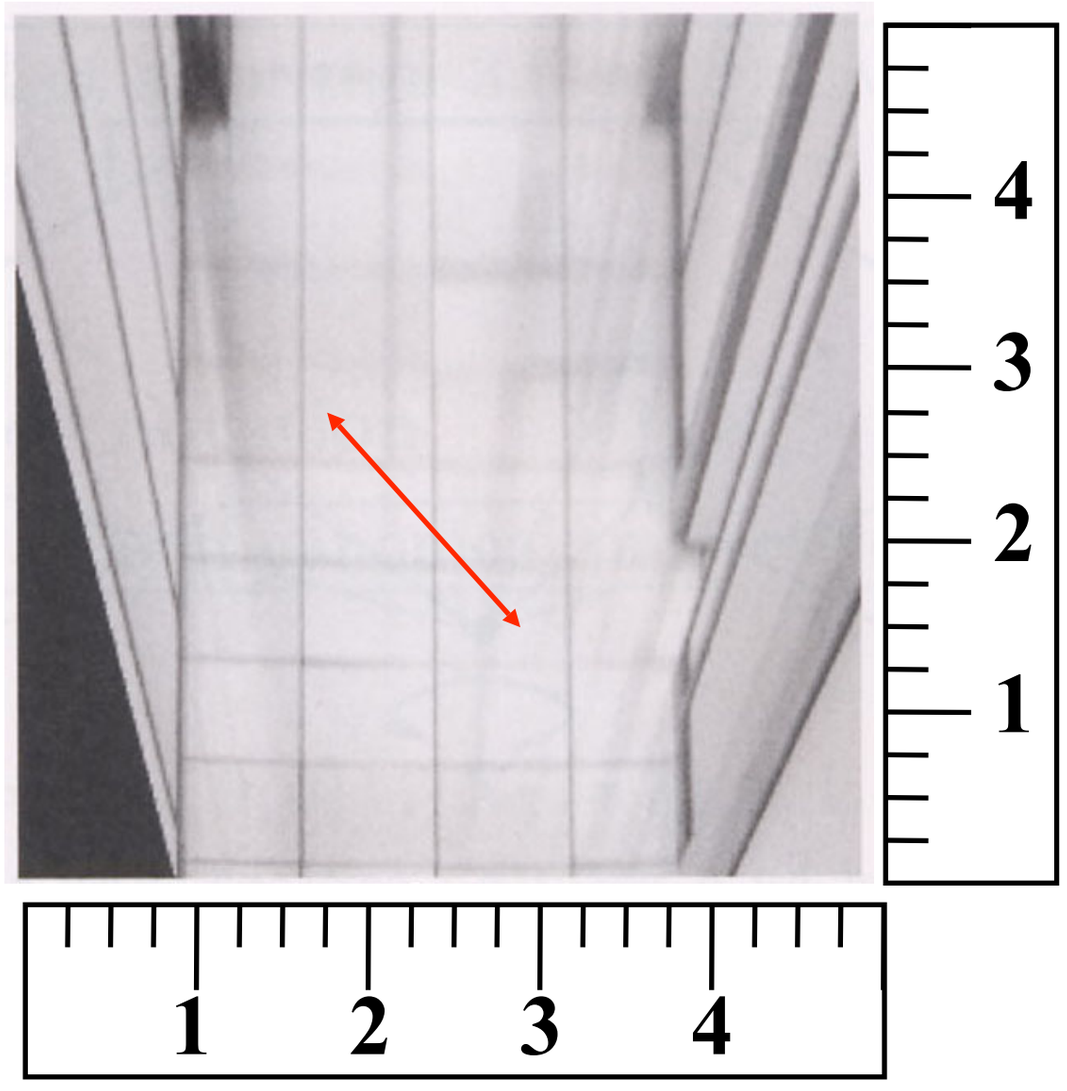




Approach: unwarp, then measure

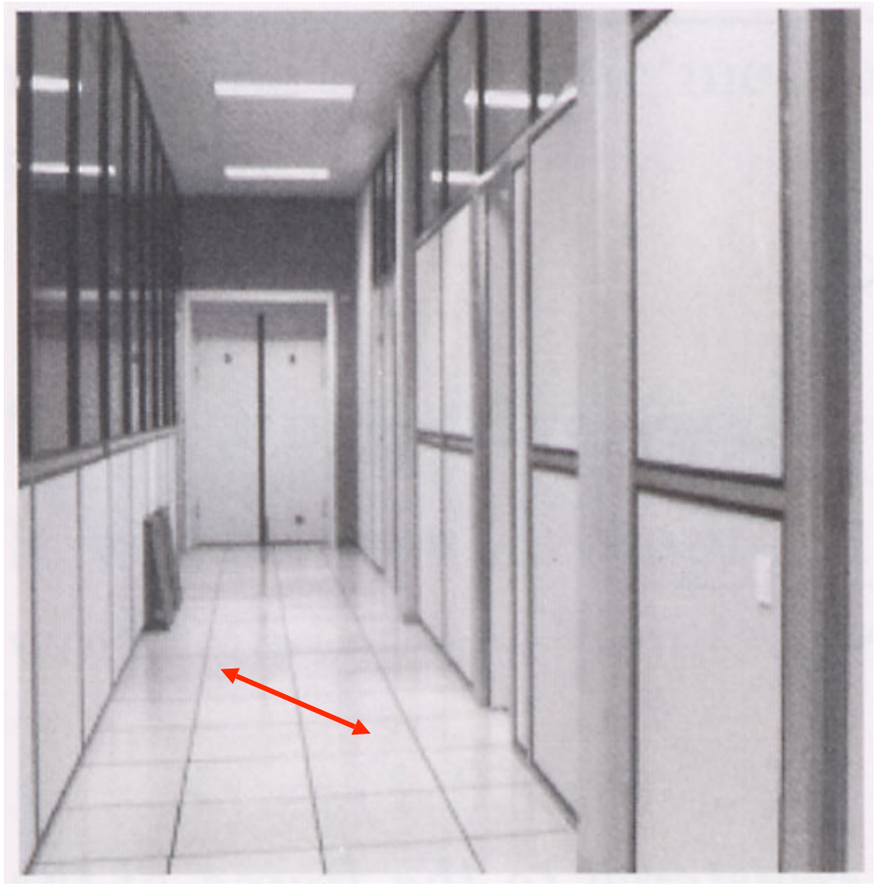
Measurements on planes

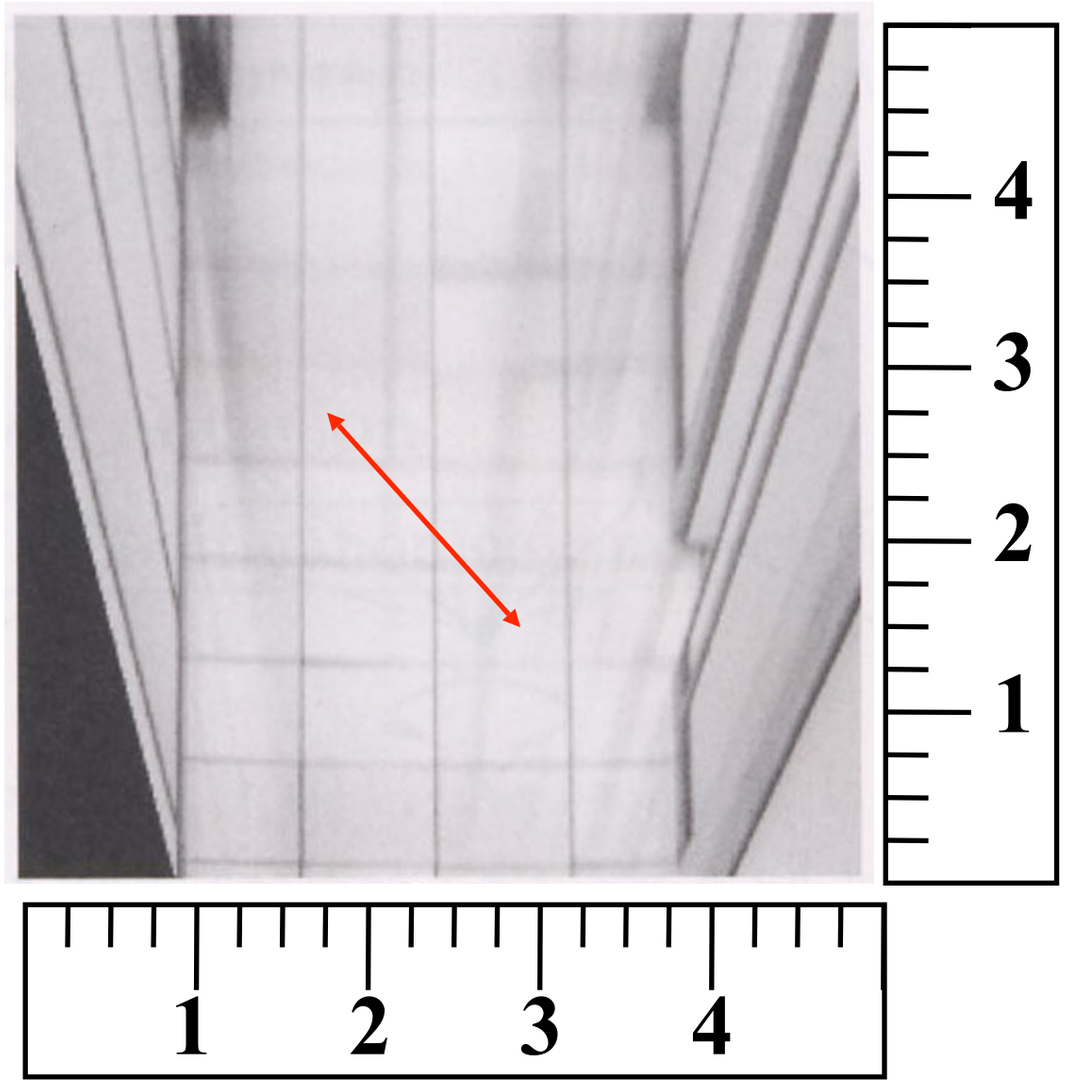




Approach: unwarp, then measure

Measurements on planes

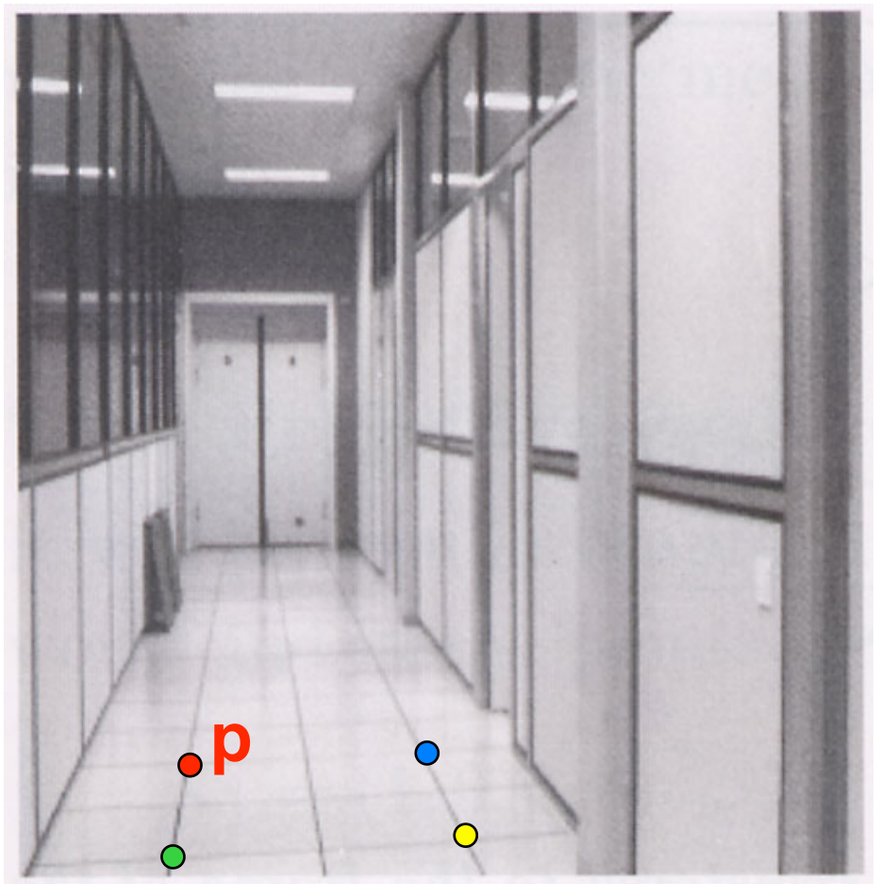


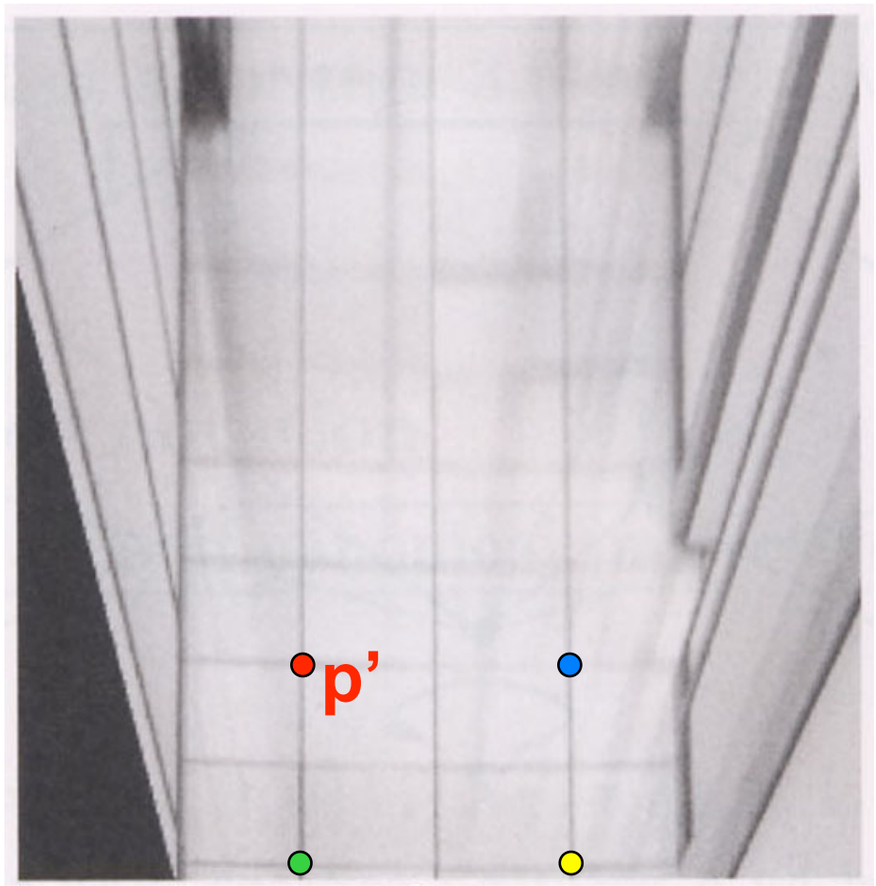


Approach: unwarp, then measure

What kind of warp is this?

Unwarp ground plane





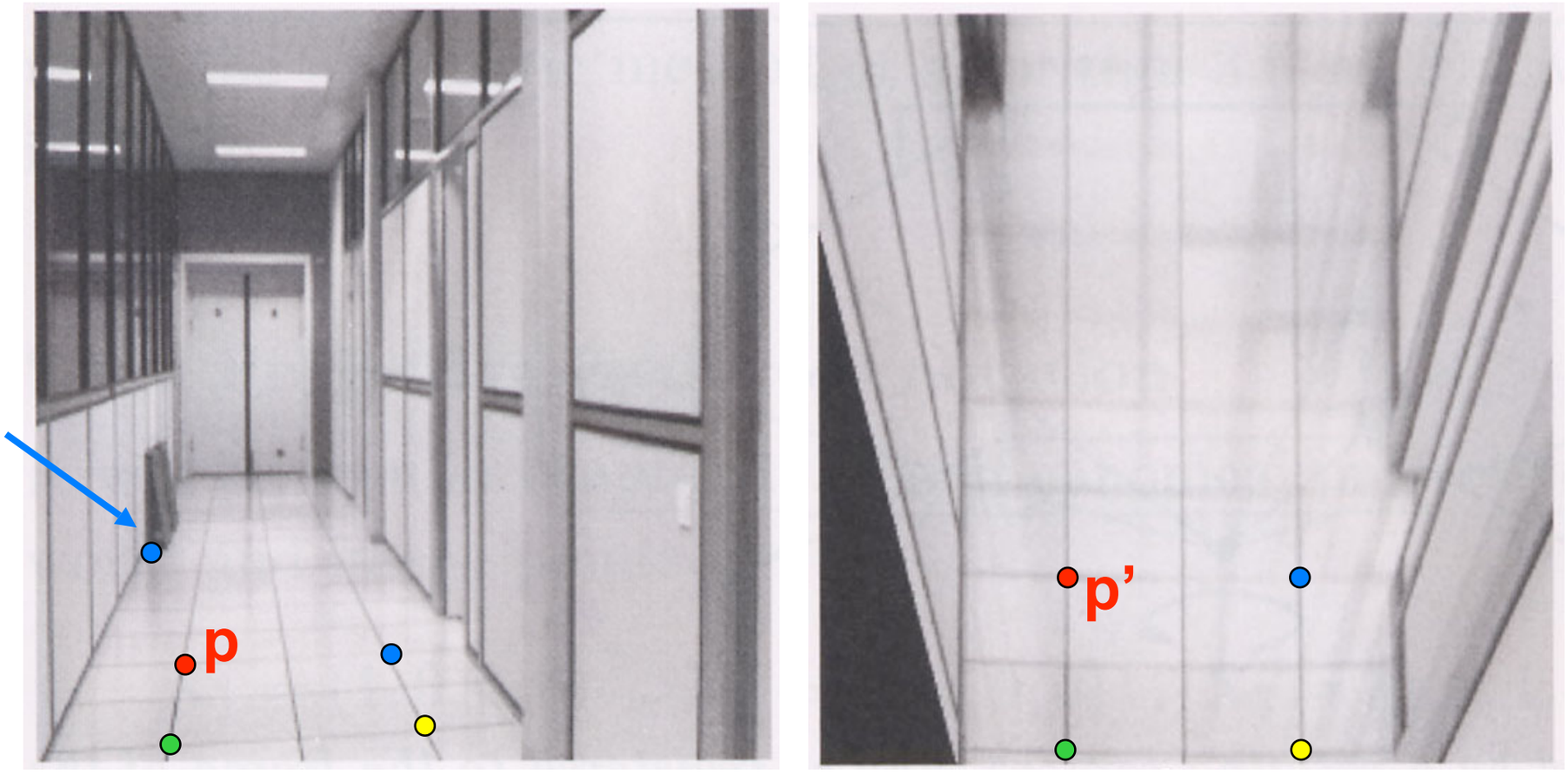
Our old friend – the homography

Need 4 reference points with world coordinates

p = (x,y)

p’ = (X,Y,0)

Unwarp ground plane



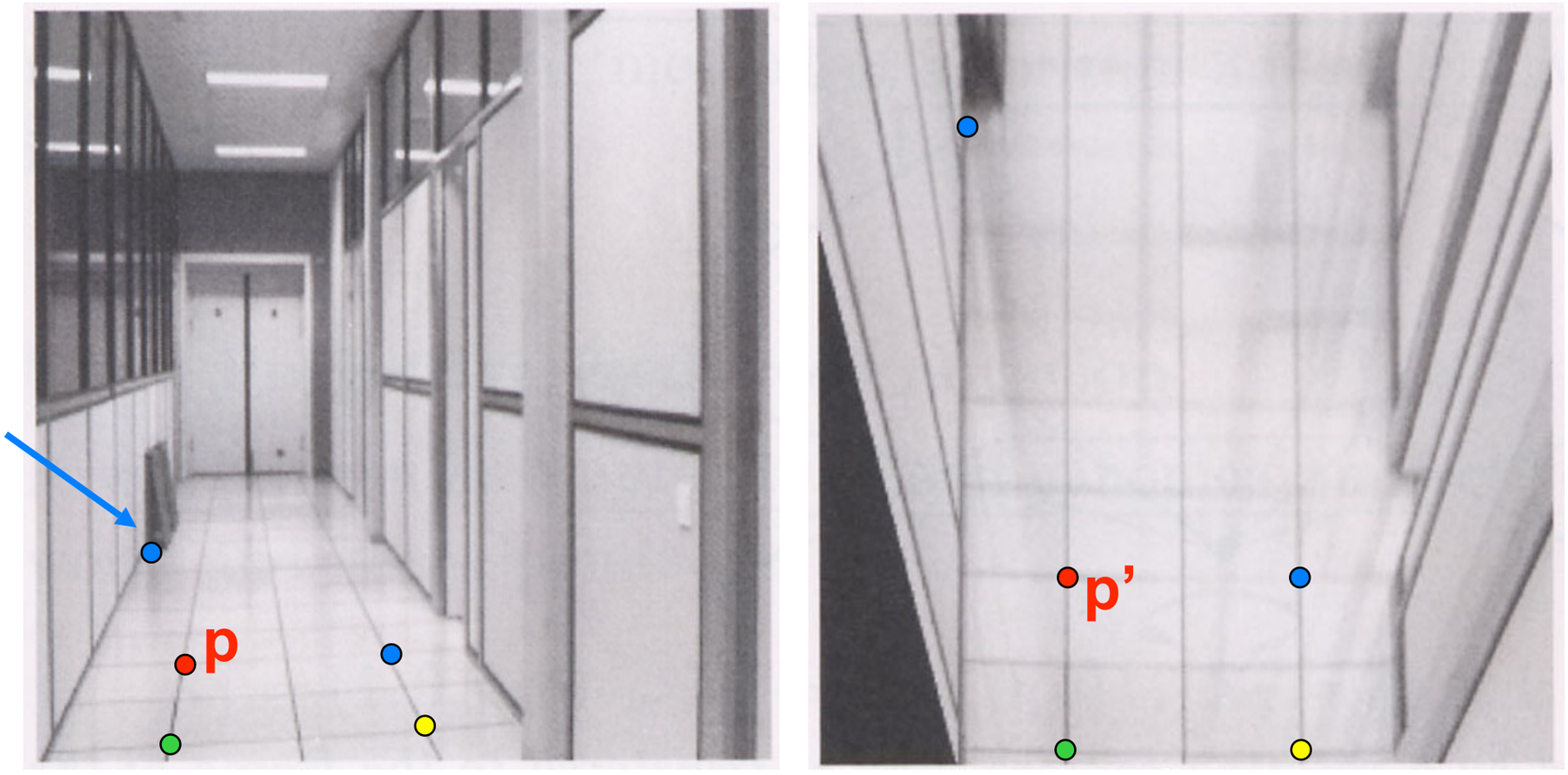
Our old friend – the homography

Need 4 reference points with world coordinates

p = (x,y)

p’ = (X,Y,0)

Unwarp ground plane



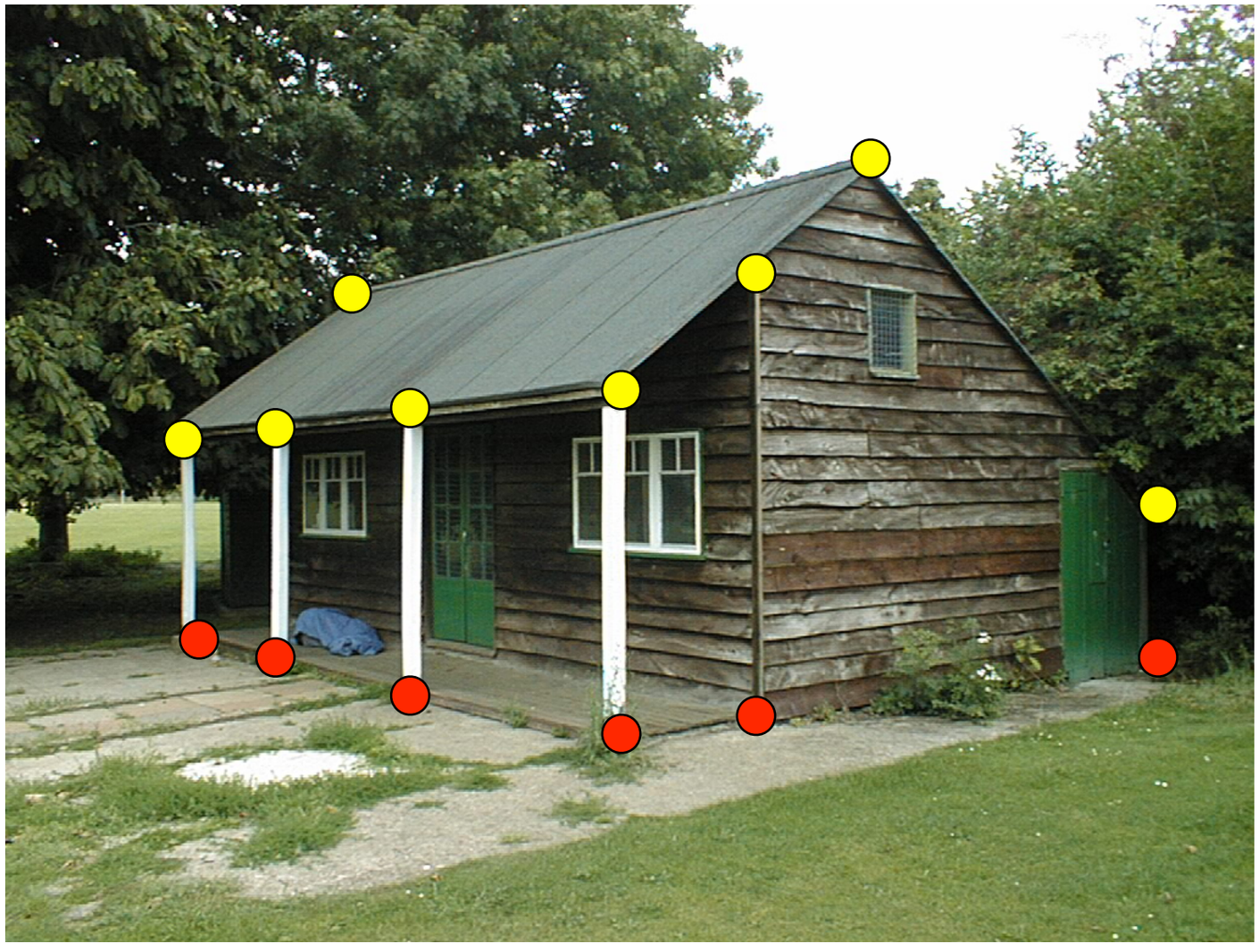
Our old friend – the homography

Need 4 reference points with world coordinates

p = (x,y)

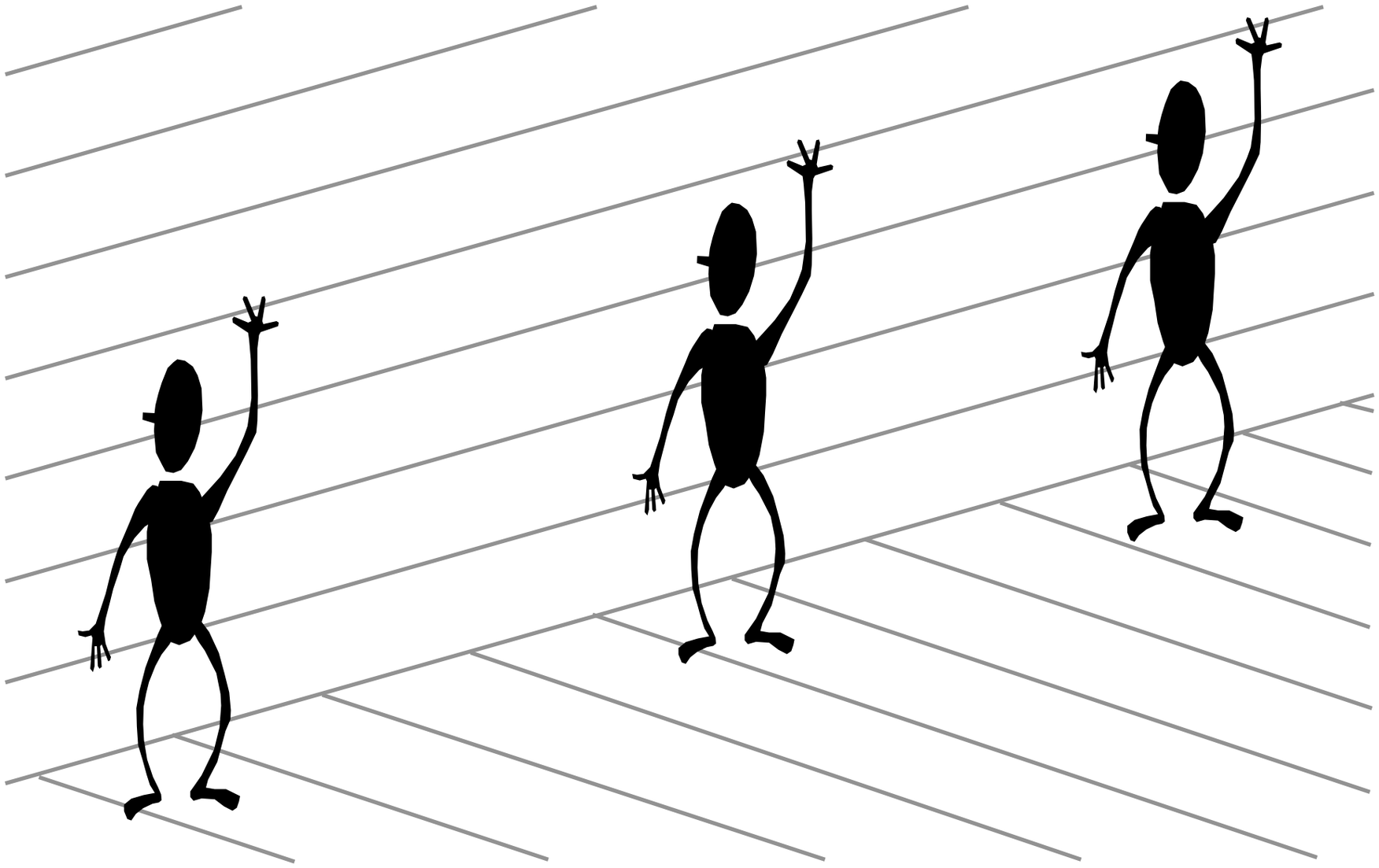
p’ = (X,Y,0)

Finding world coordinates (X,Y,Z)

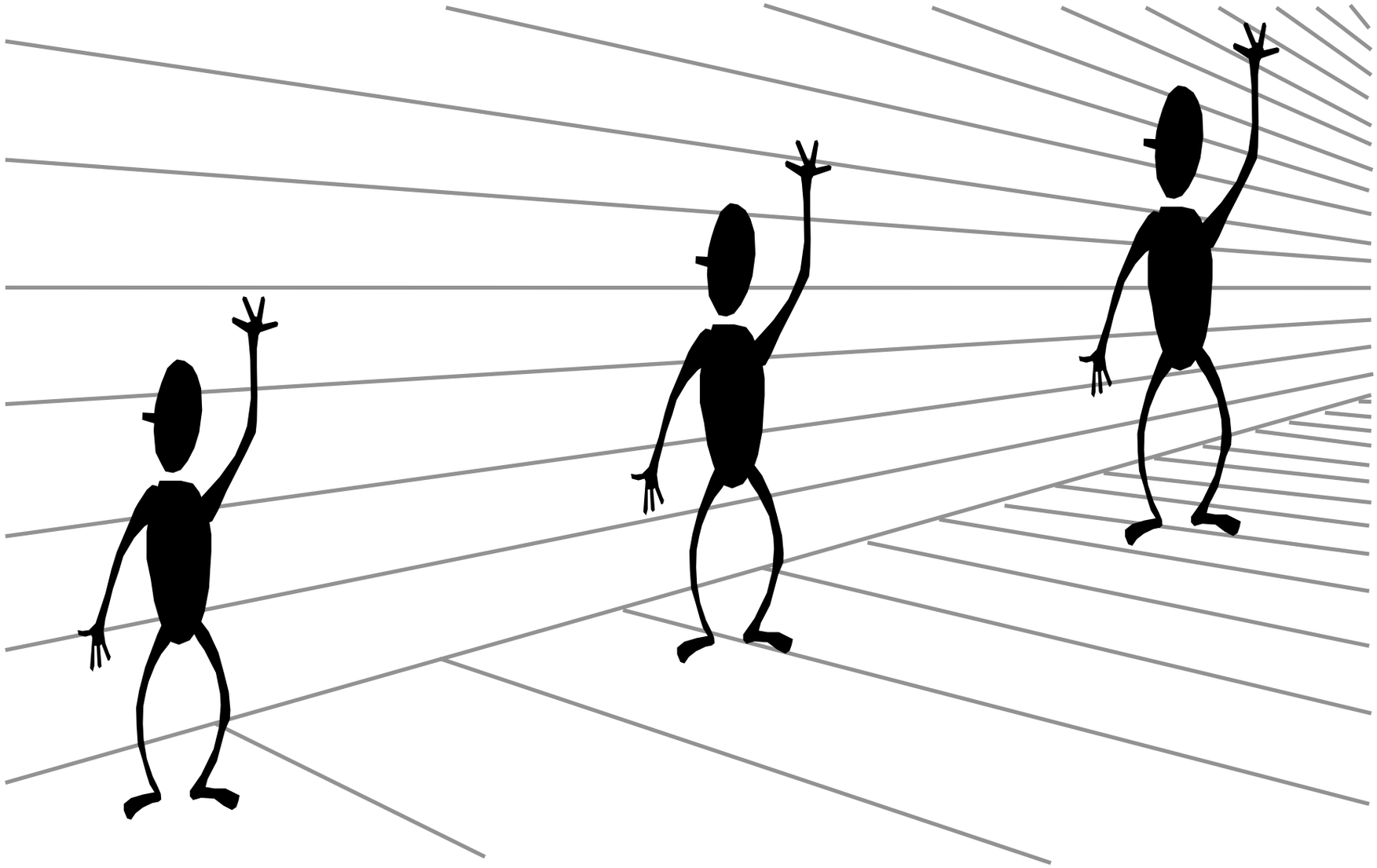


1. Define the ground plane (Z=0)
2. Compute points (X,Y,0) on that plane
3. Compute the *heights* Z of all other points

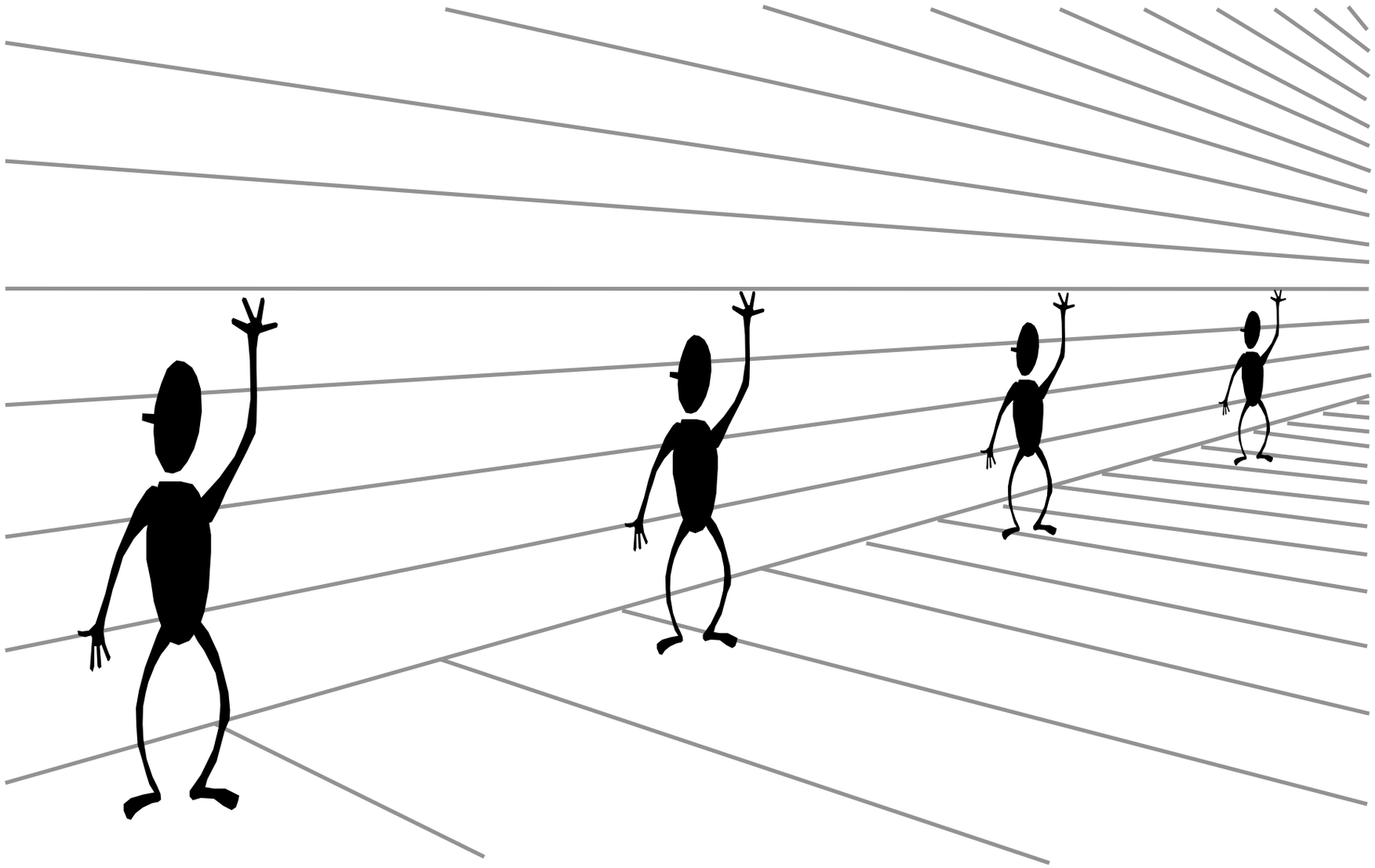
Comparing heights



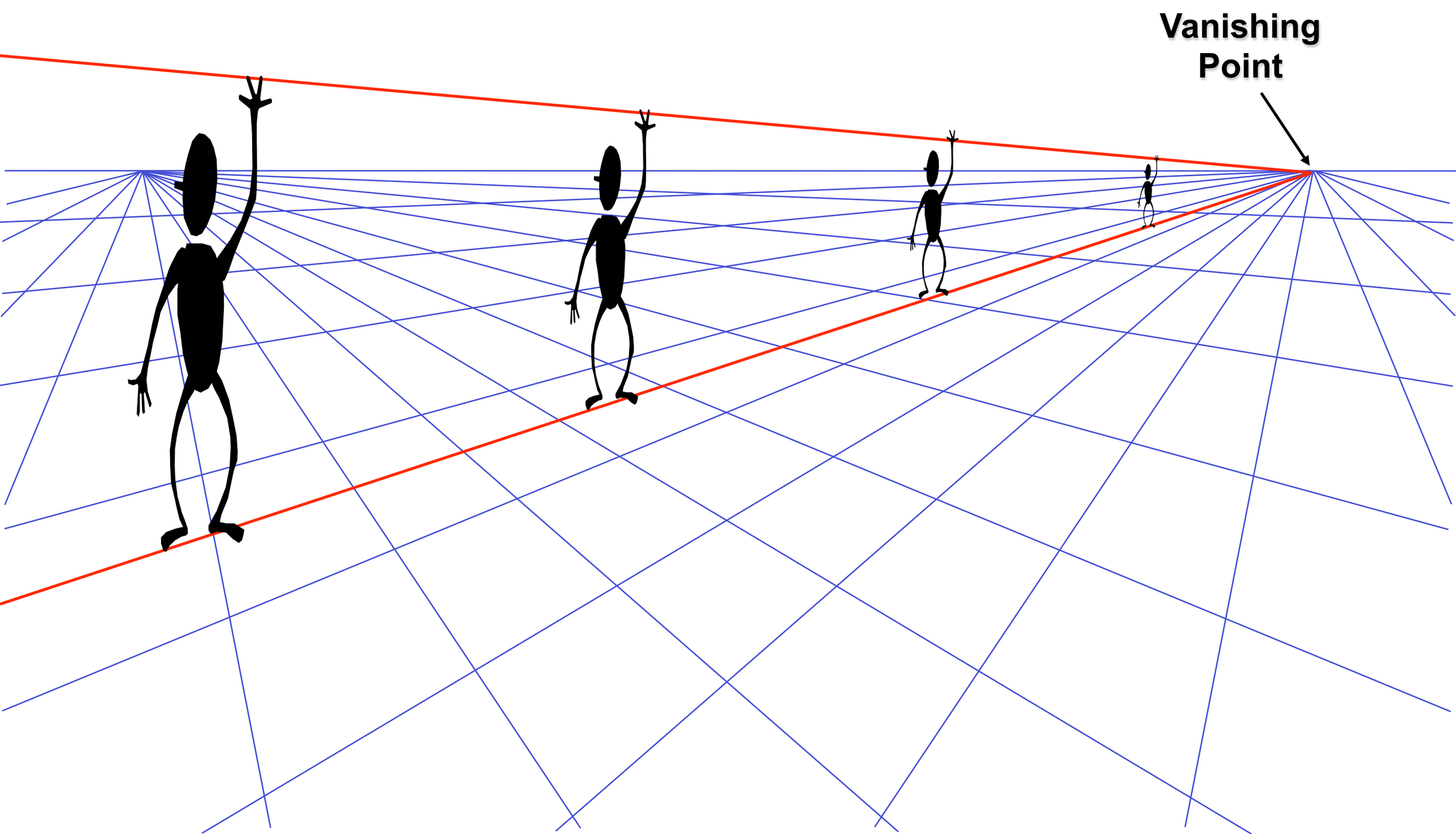
Perspective cues



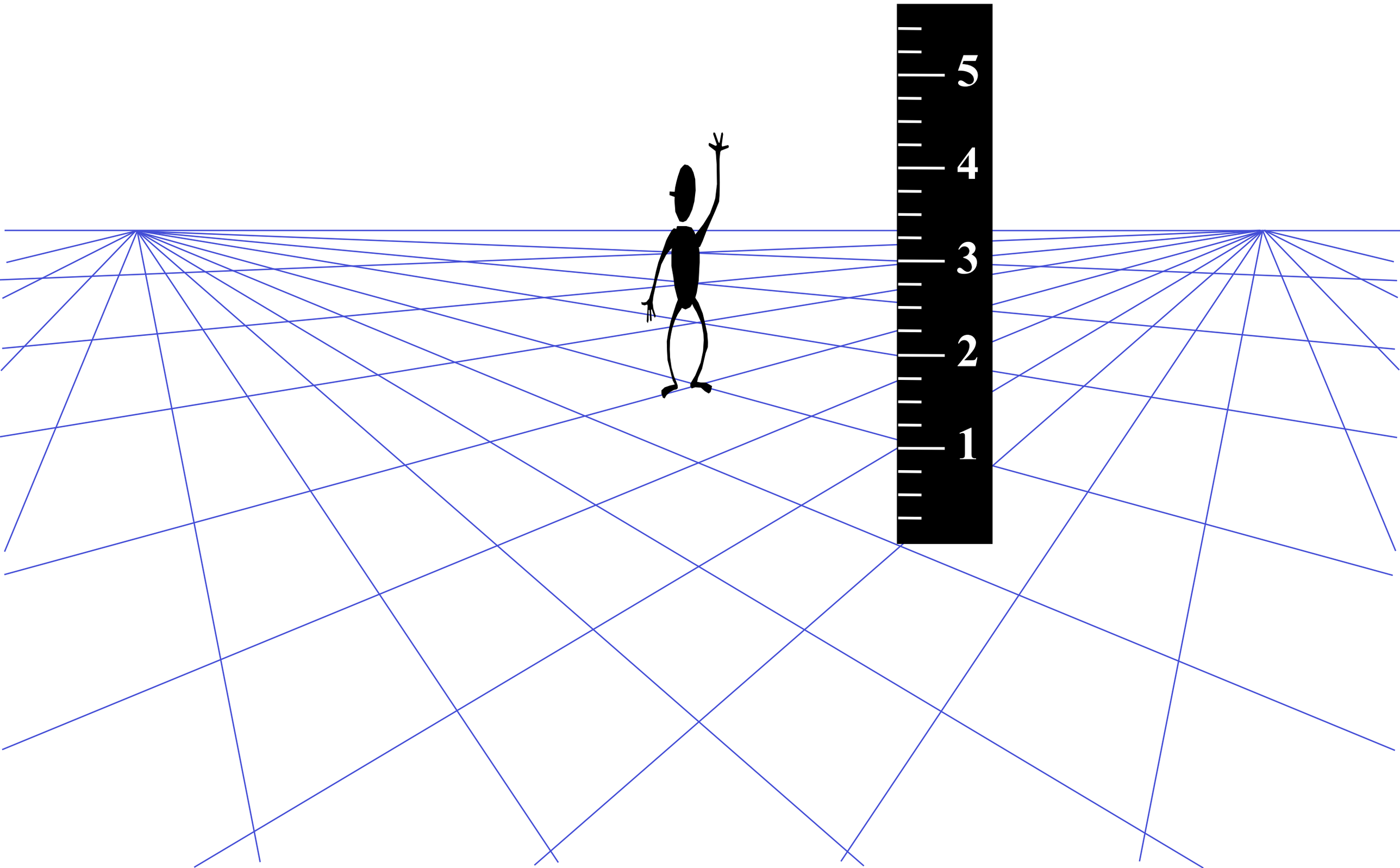
Perspective cues



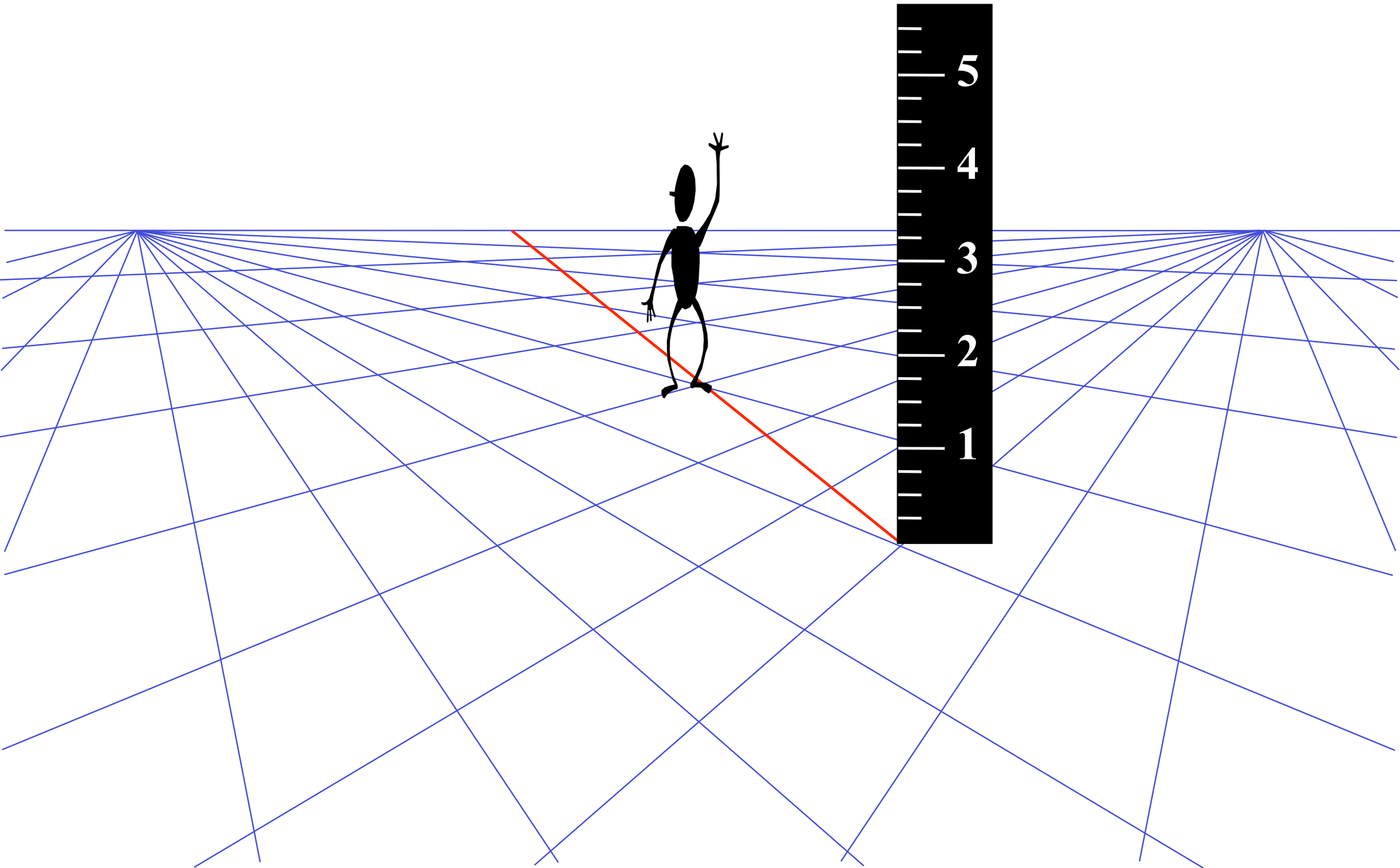
Comparing heights



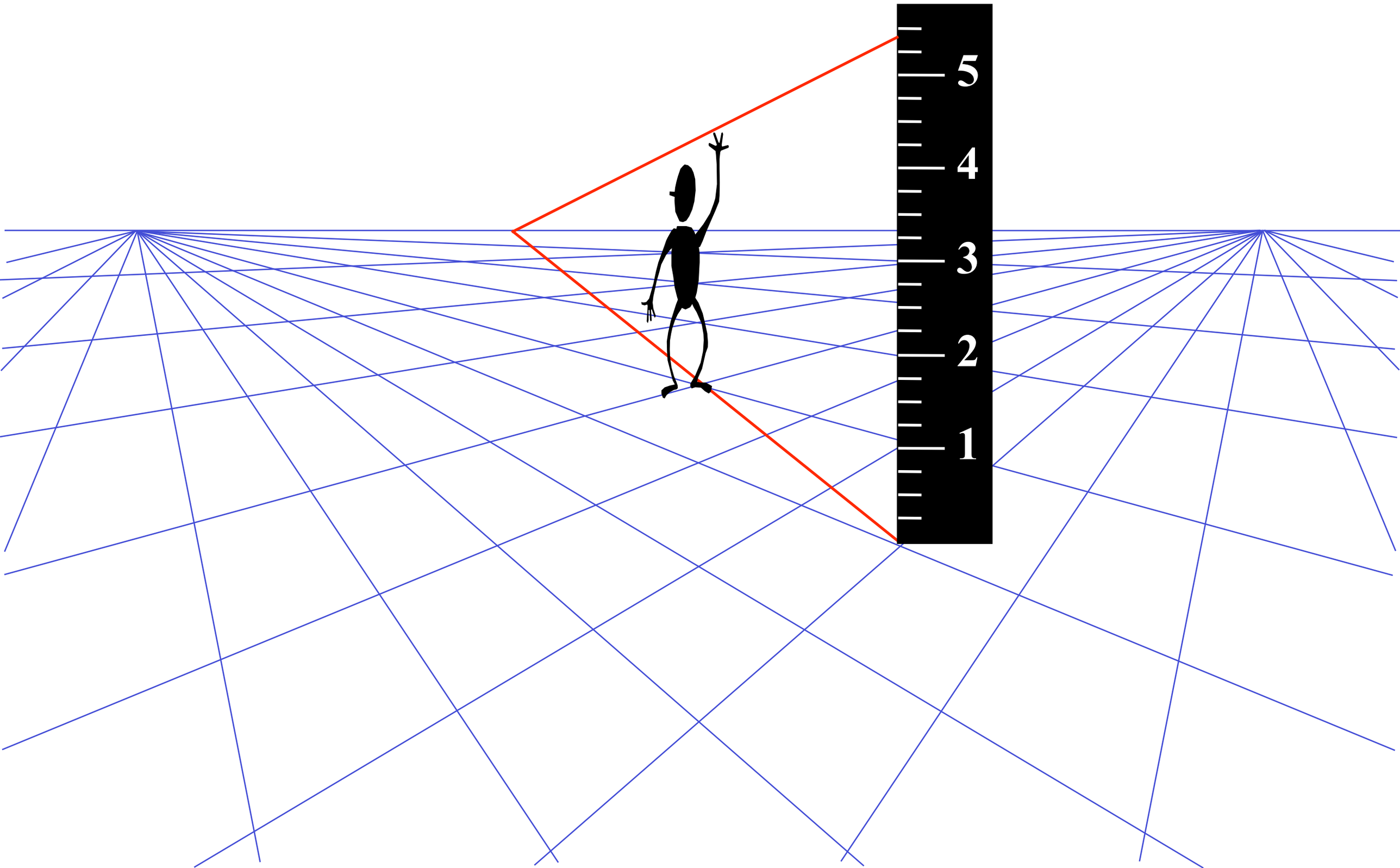
Measuring height



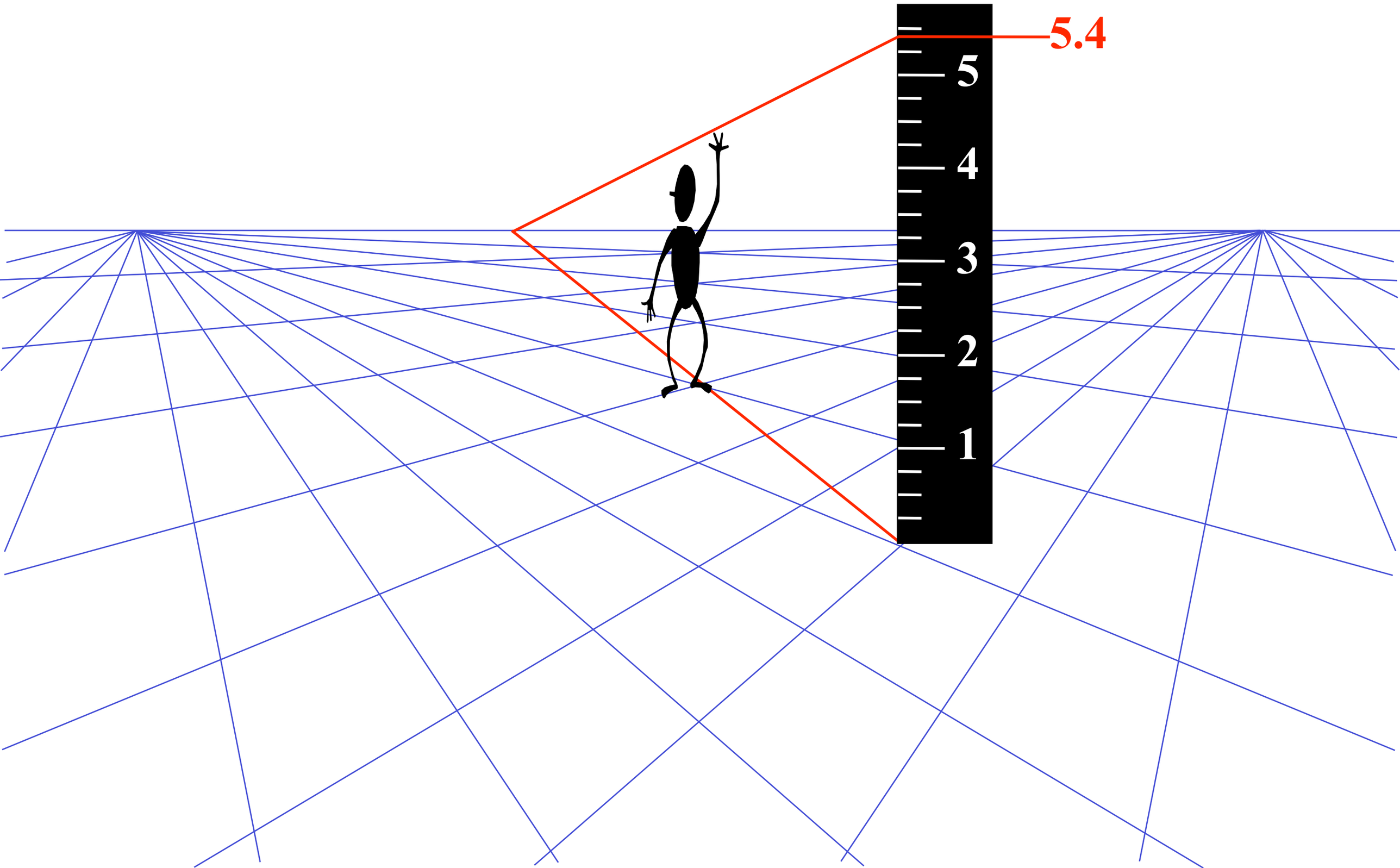
Measuring height



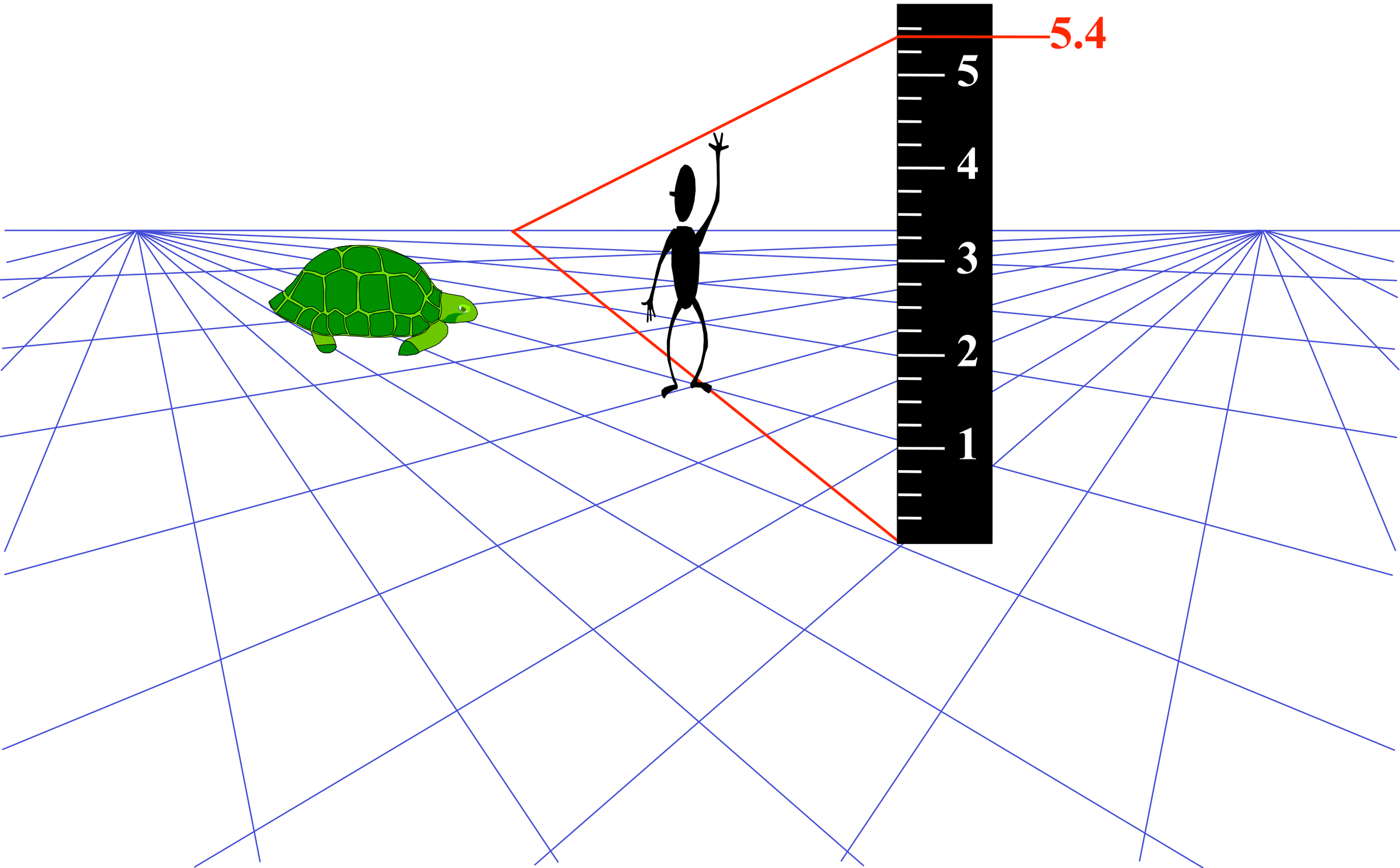
Measuring height



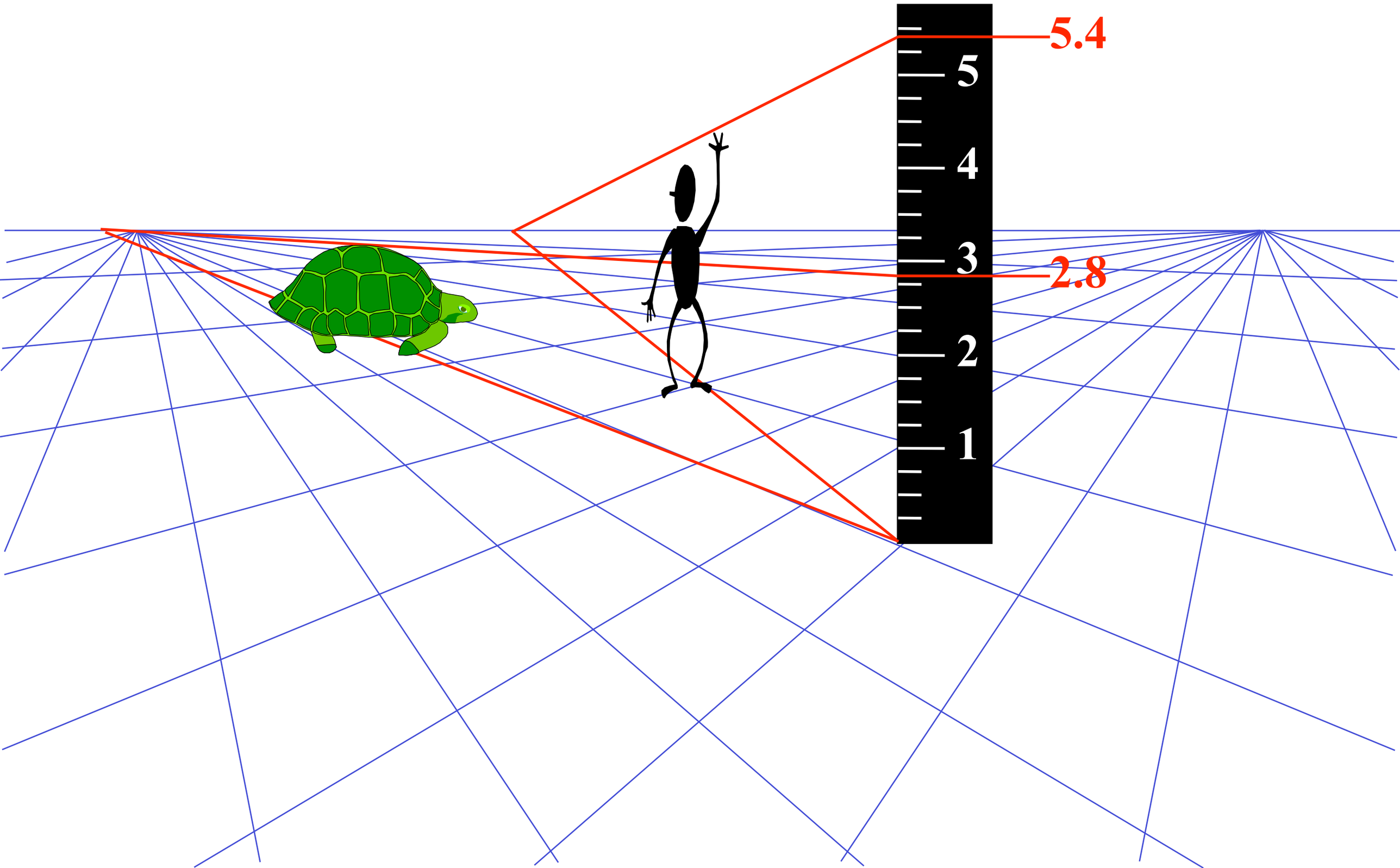
Measuring height



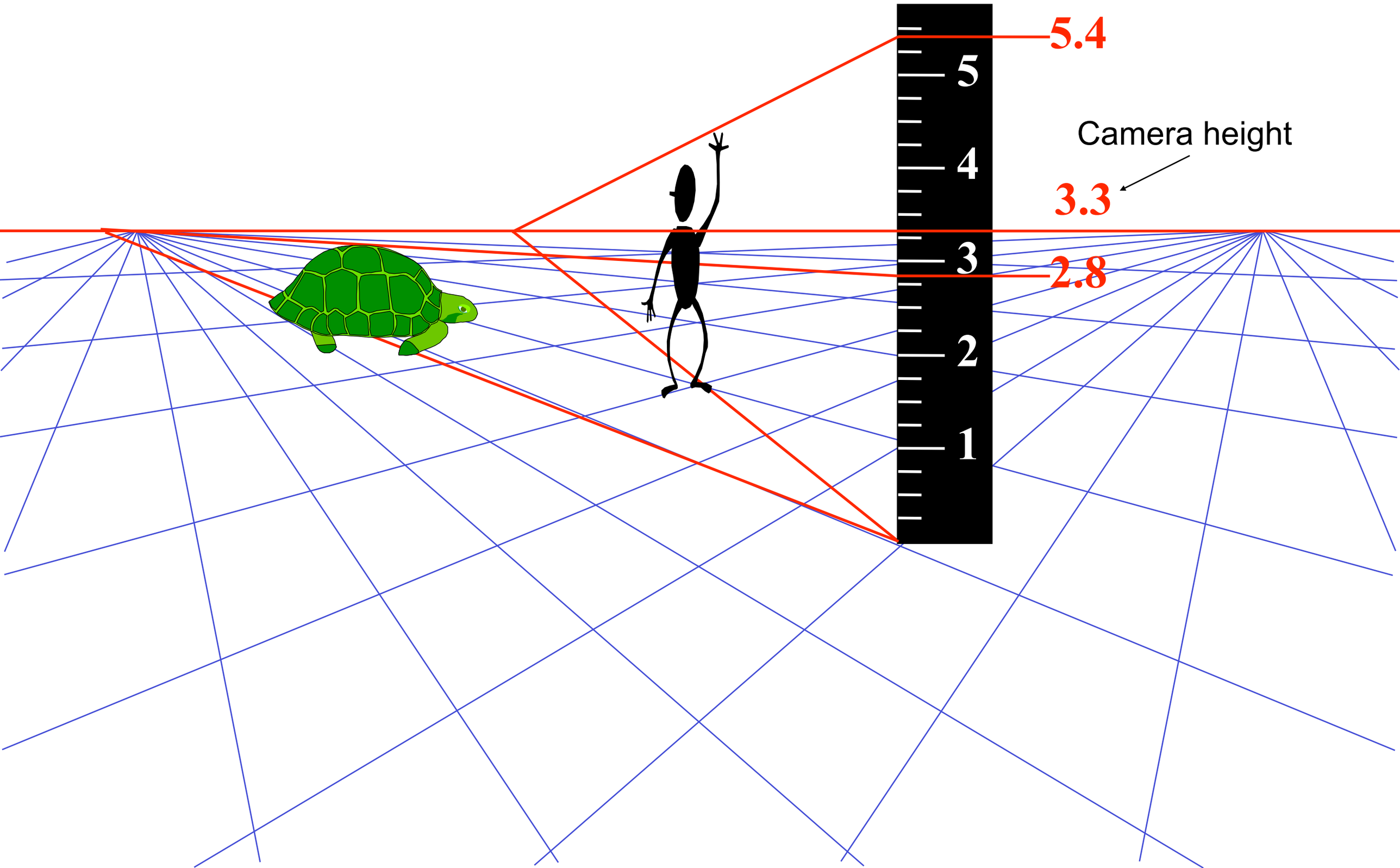
Measuring height



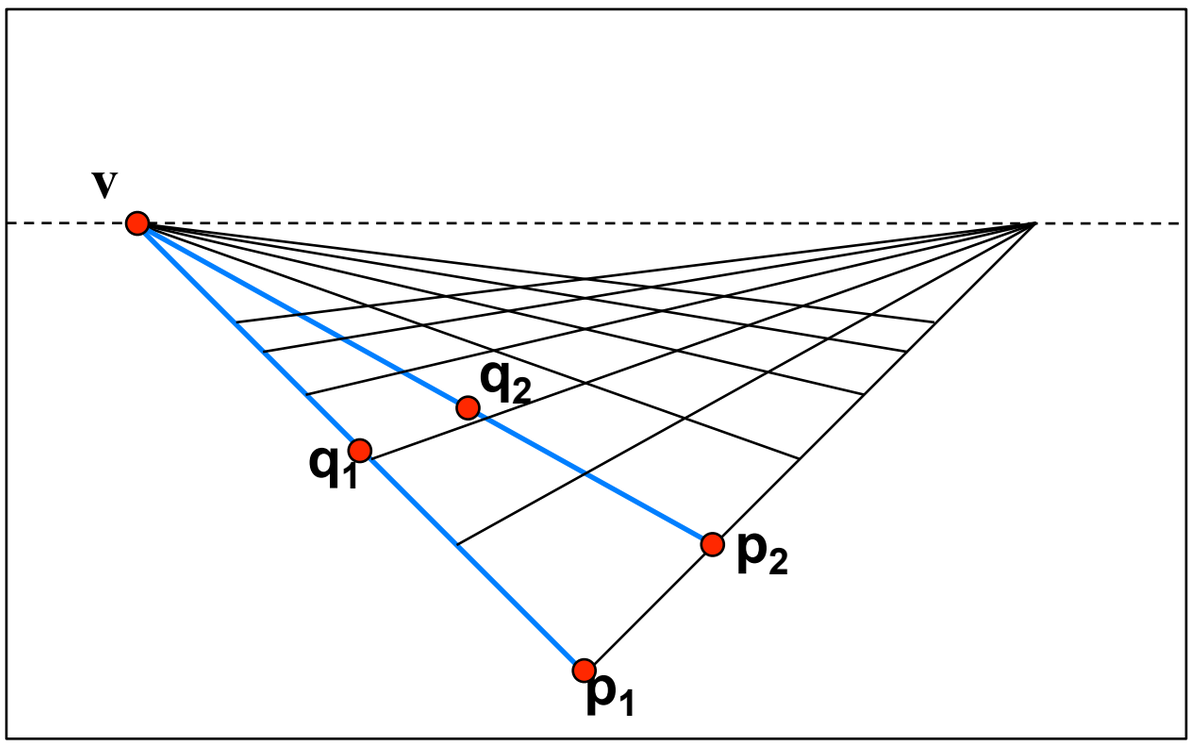
Measuring height



Measuring height

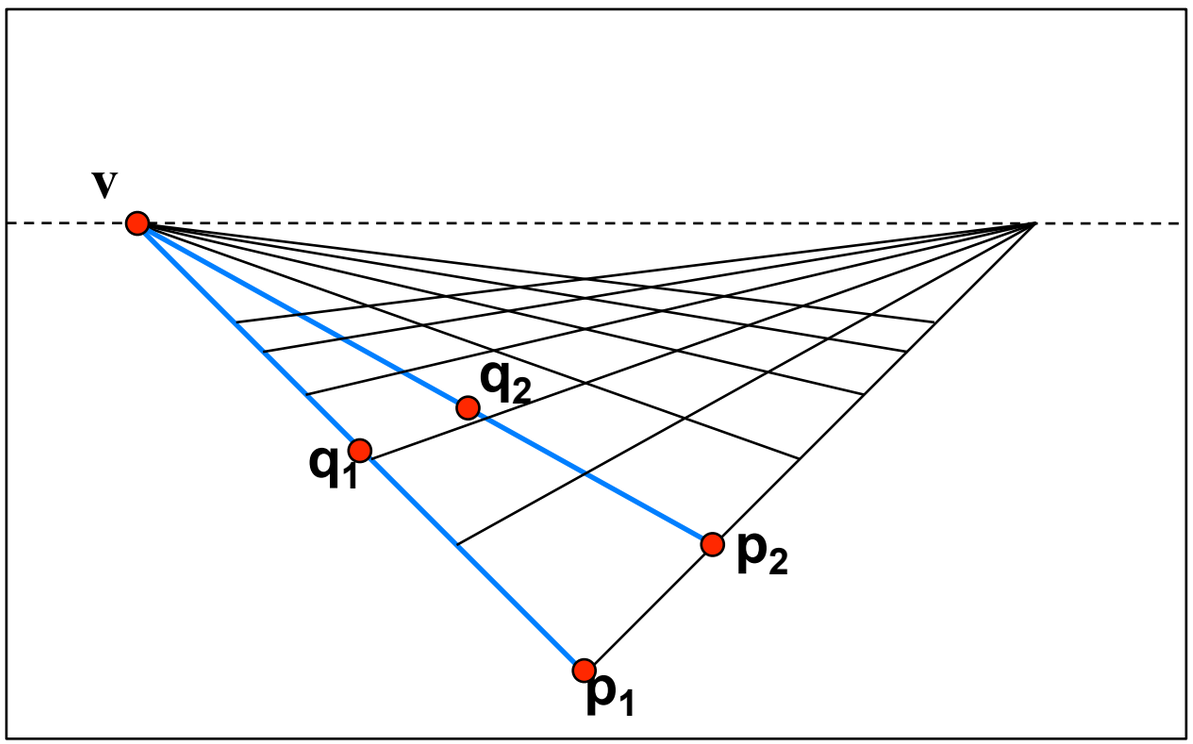


Computing vanishing points (from lines)



Intersect p_1q_1 with p_2q_2

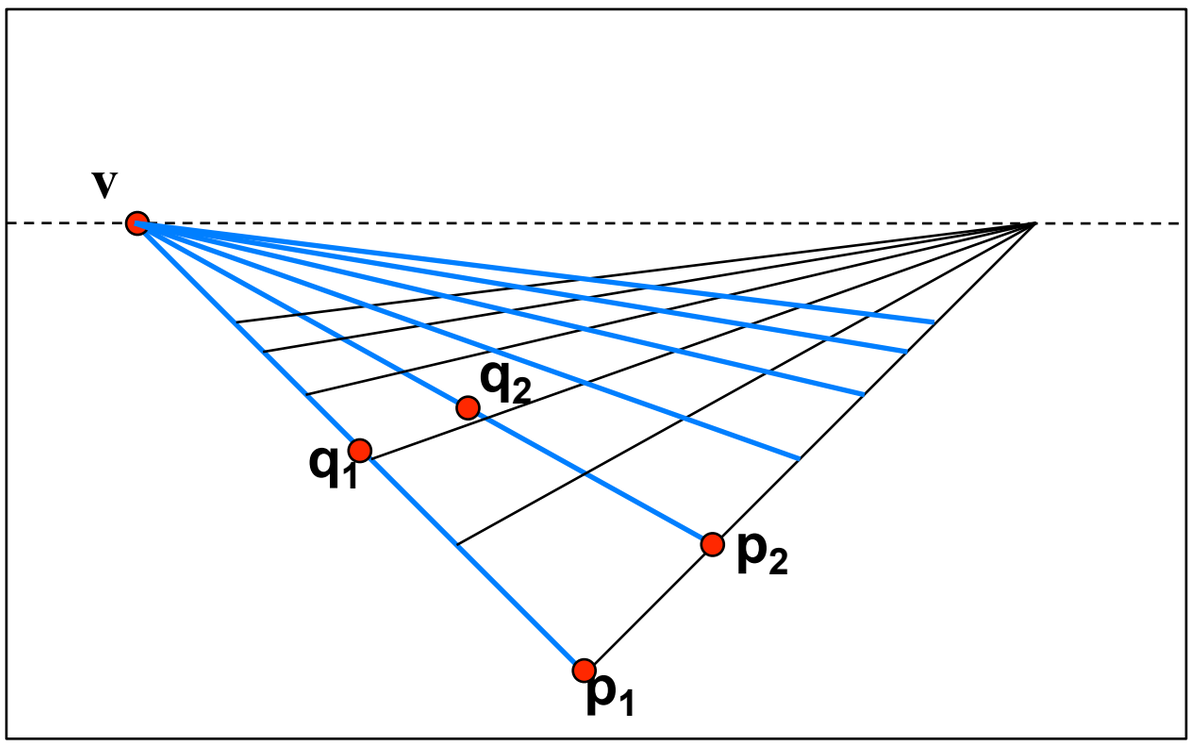
Computing vanishing points (from lines)



Intersect p_1q_1 with p_2q_2

$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

Computing vanishing points (from lines)



Intersect p_1q_1 with p_2q_2

$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

Least squares version

- Better to use more than two lines and compute the “closest” point of intersection
- See notes by Bob Collins for one good way of doing this:
  - http://www-2.cs.cmu.edu/~ph/869/www/notes/vanishing.txt

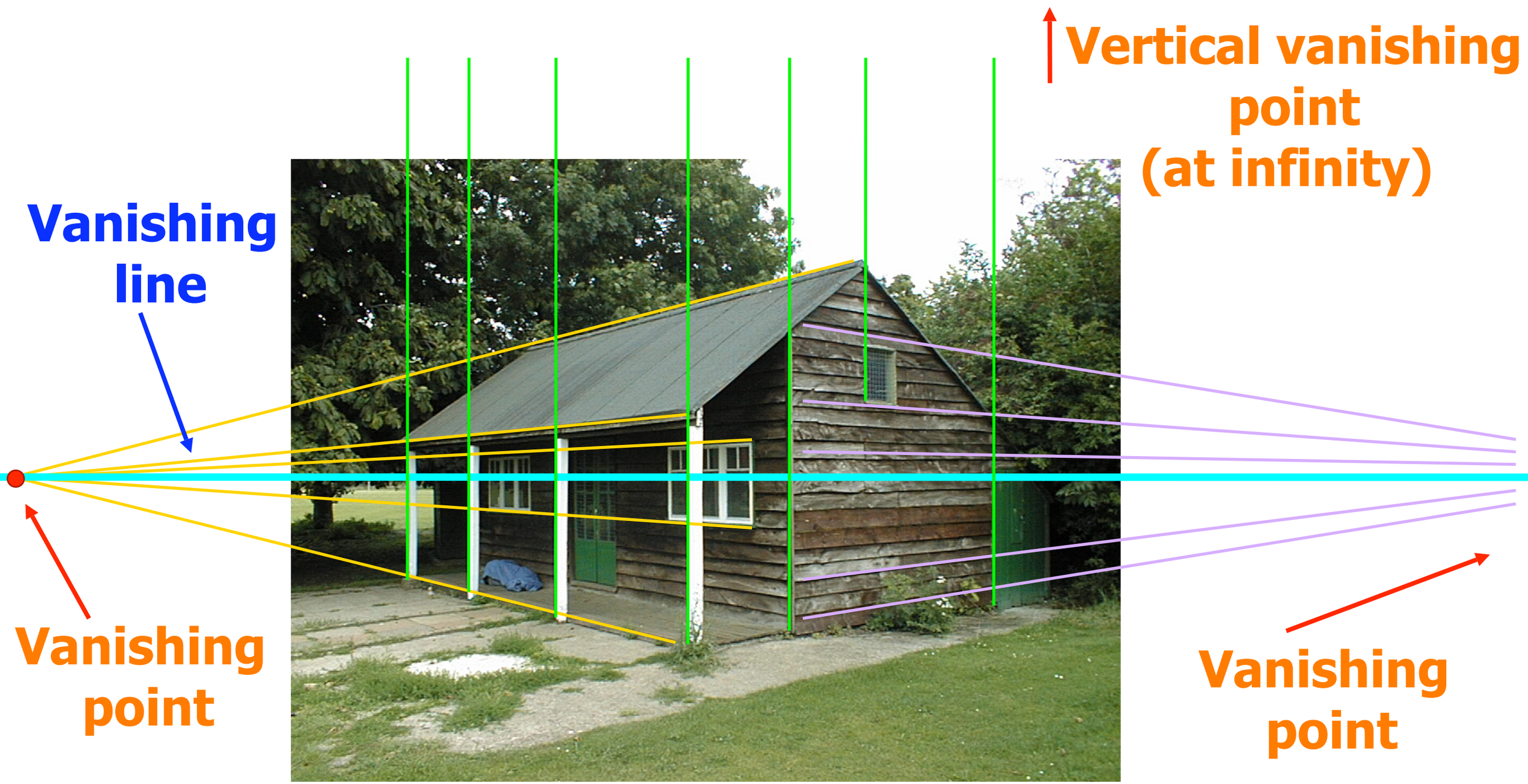
Criminisi ’99

Criminisi ’99

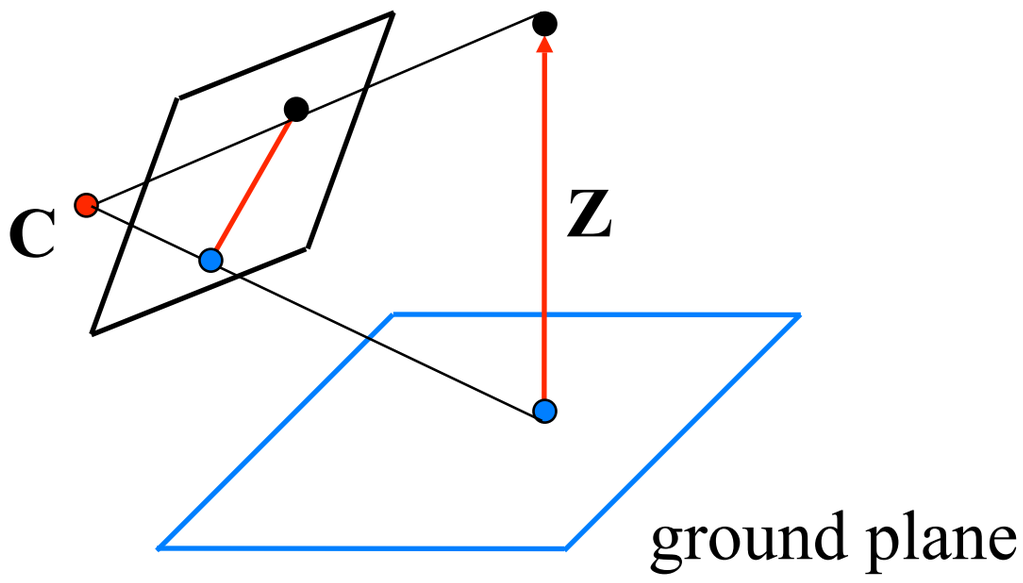
Criminisi ’99

Criminisi ’99

Criminisi ’99



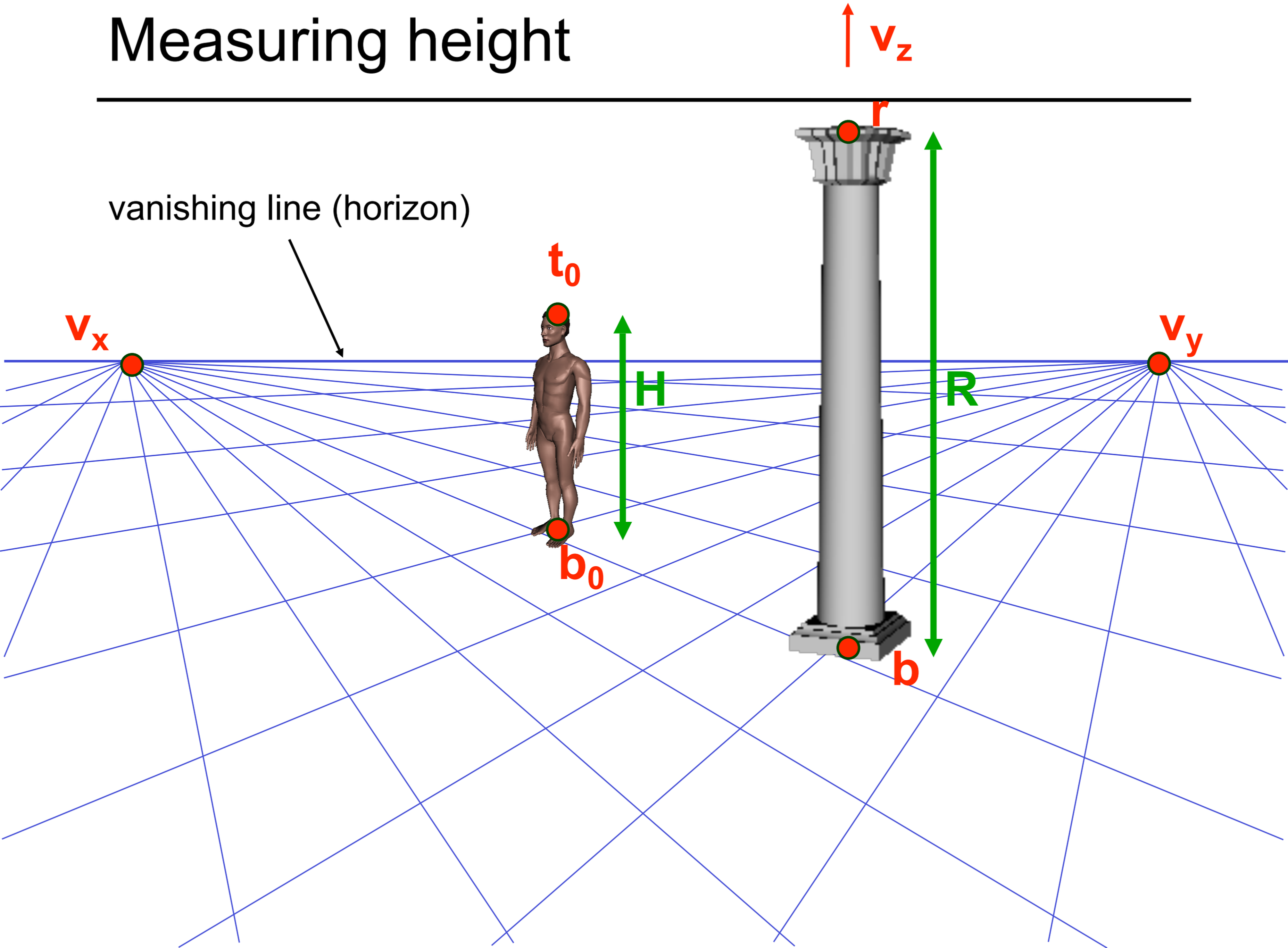
Measuring height without a ruler



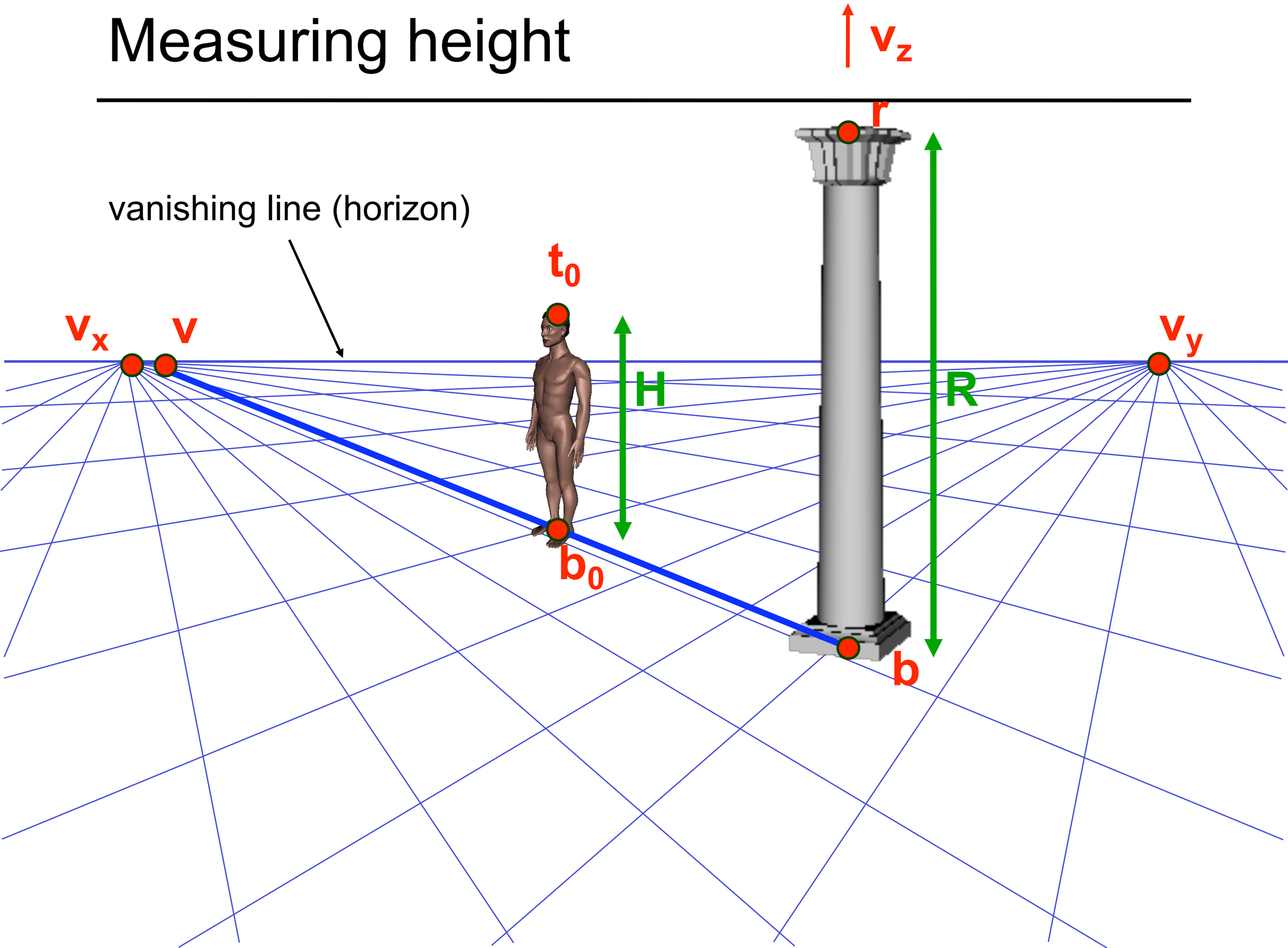
Compute Z from image measurements

- Need more than vanishing points to do this

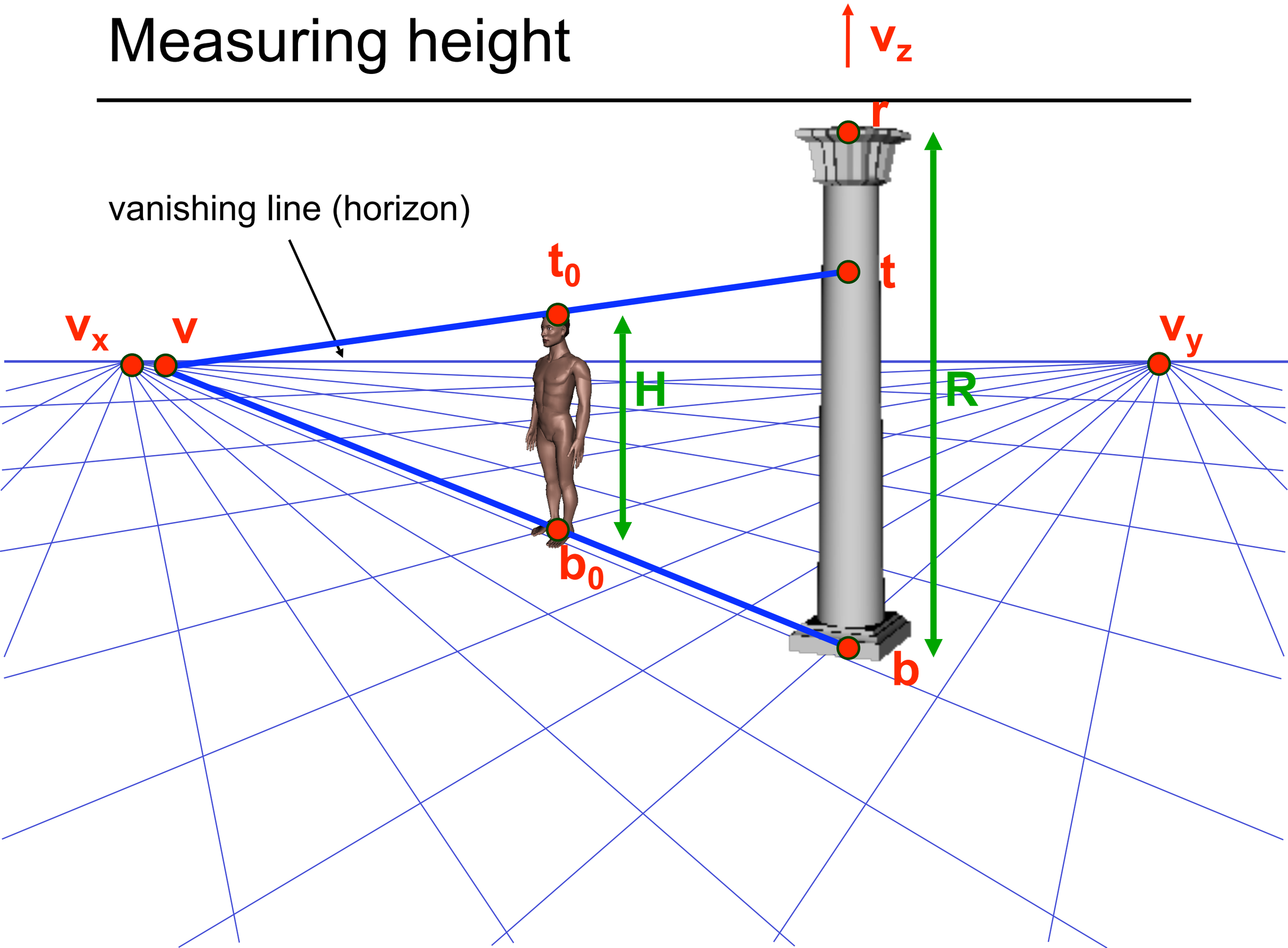
Measuring height



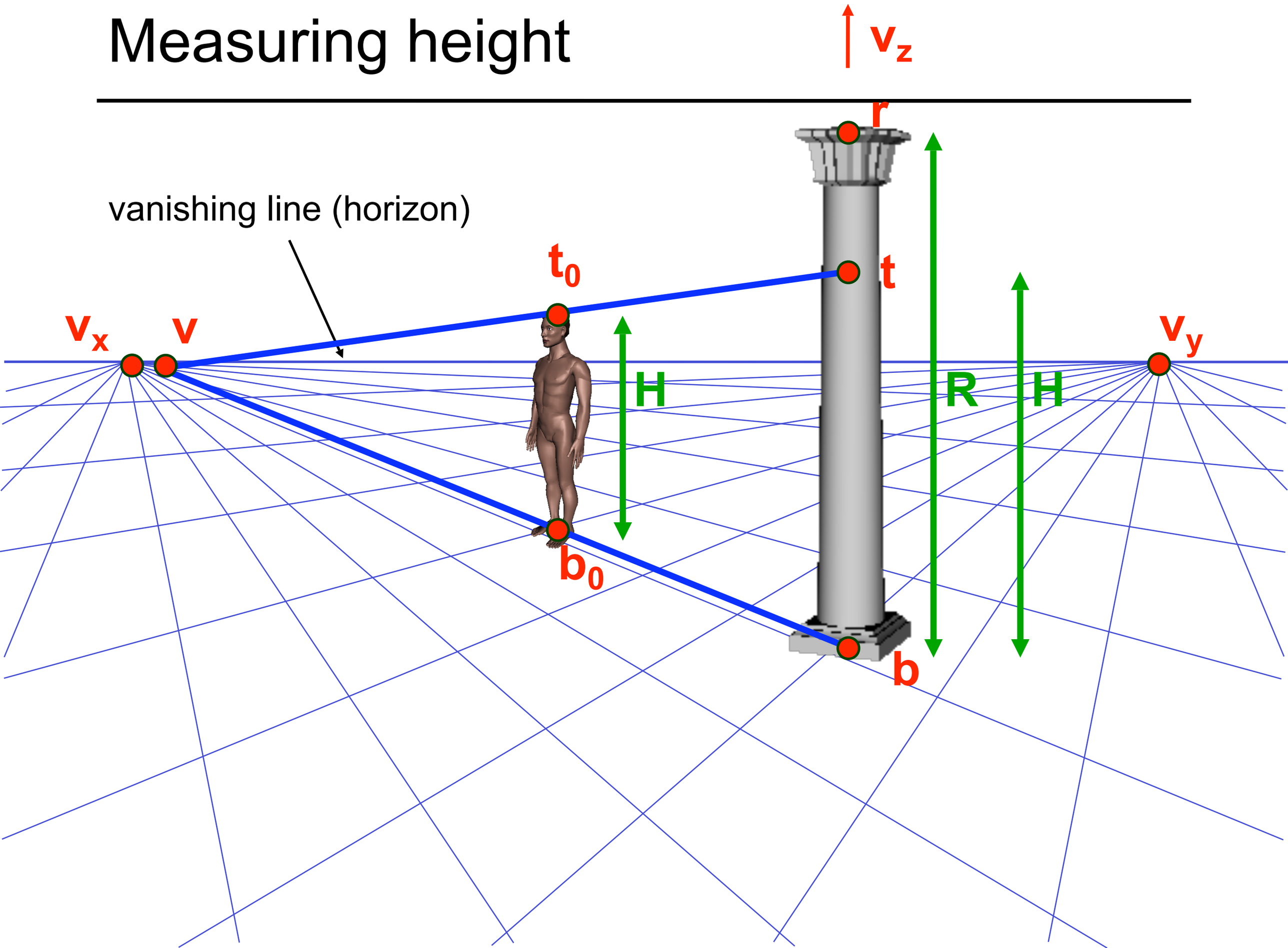
Measuring height



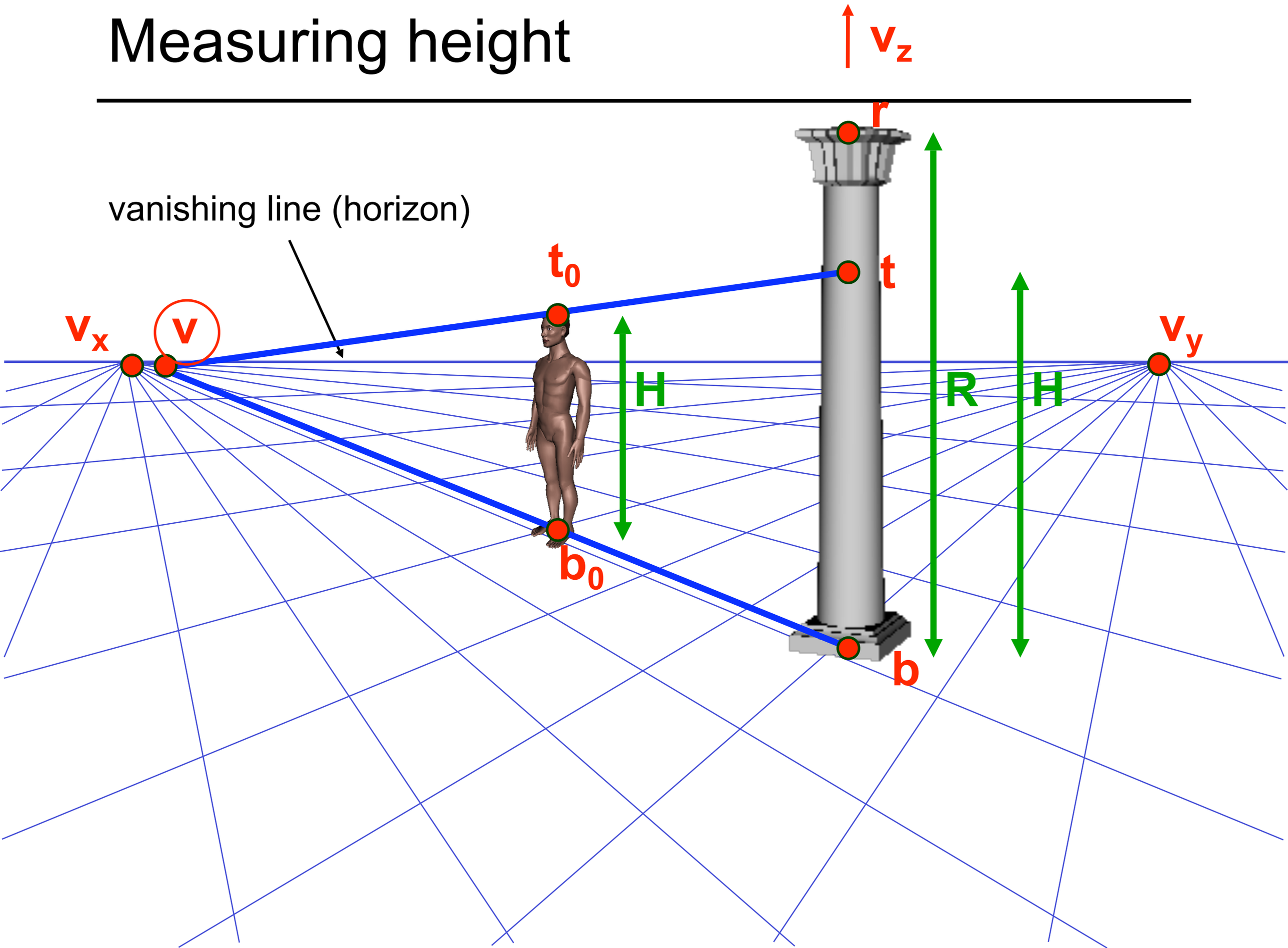
Measuring height



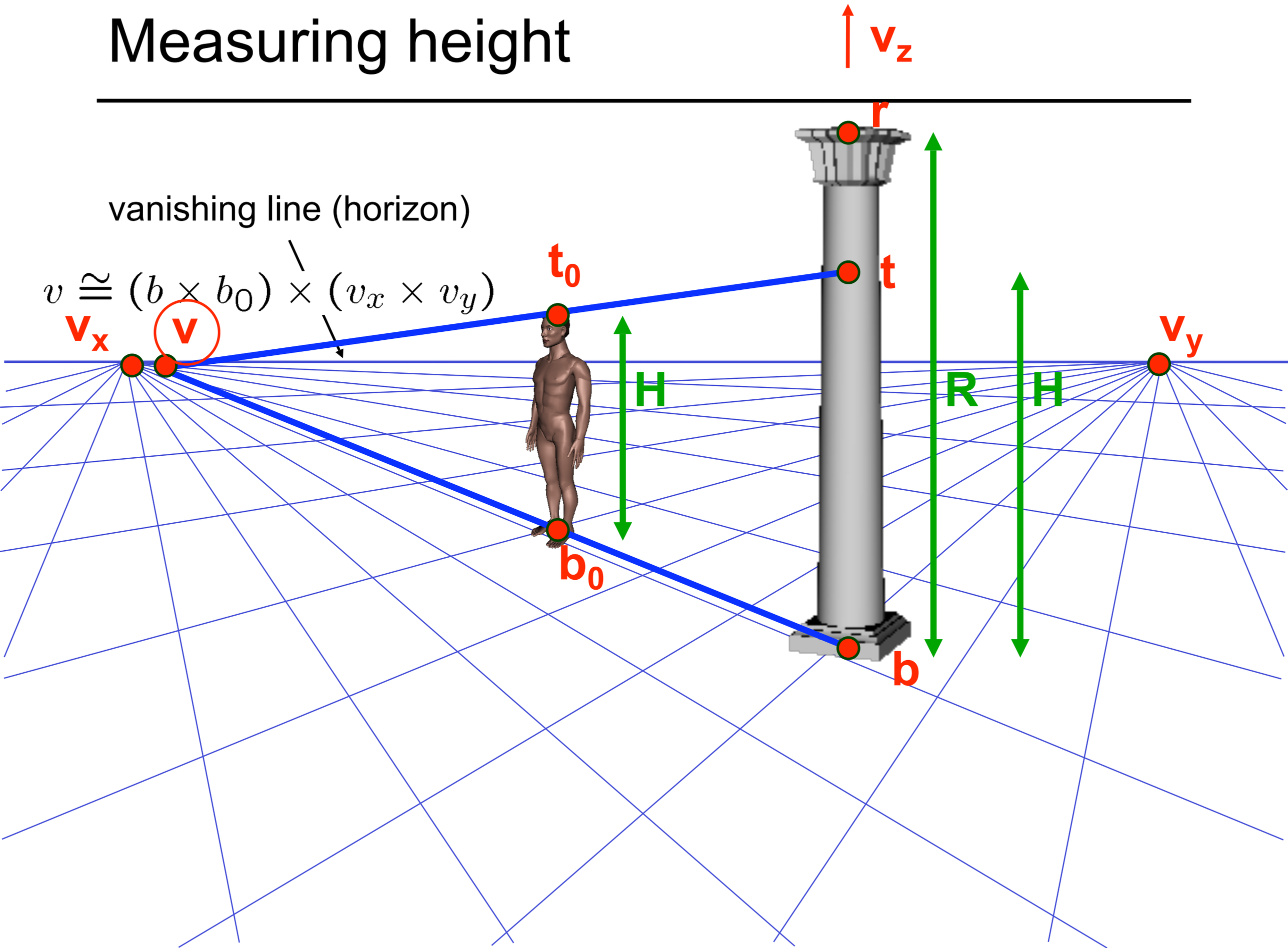
Measuring height



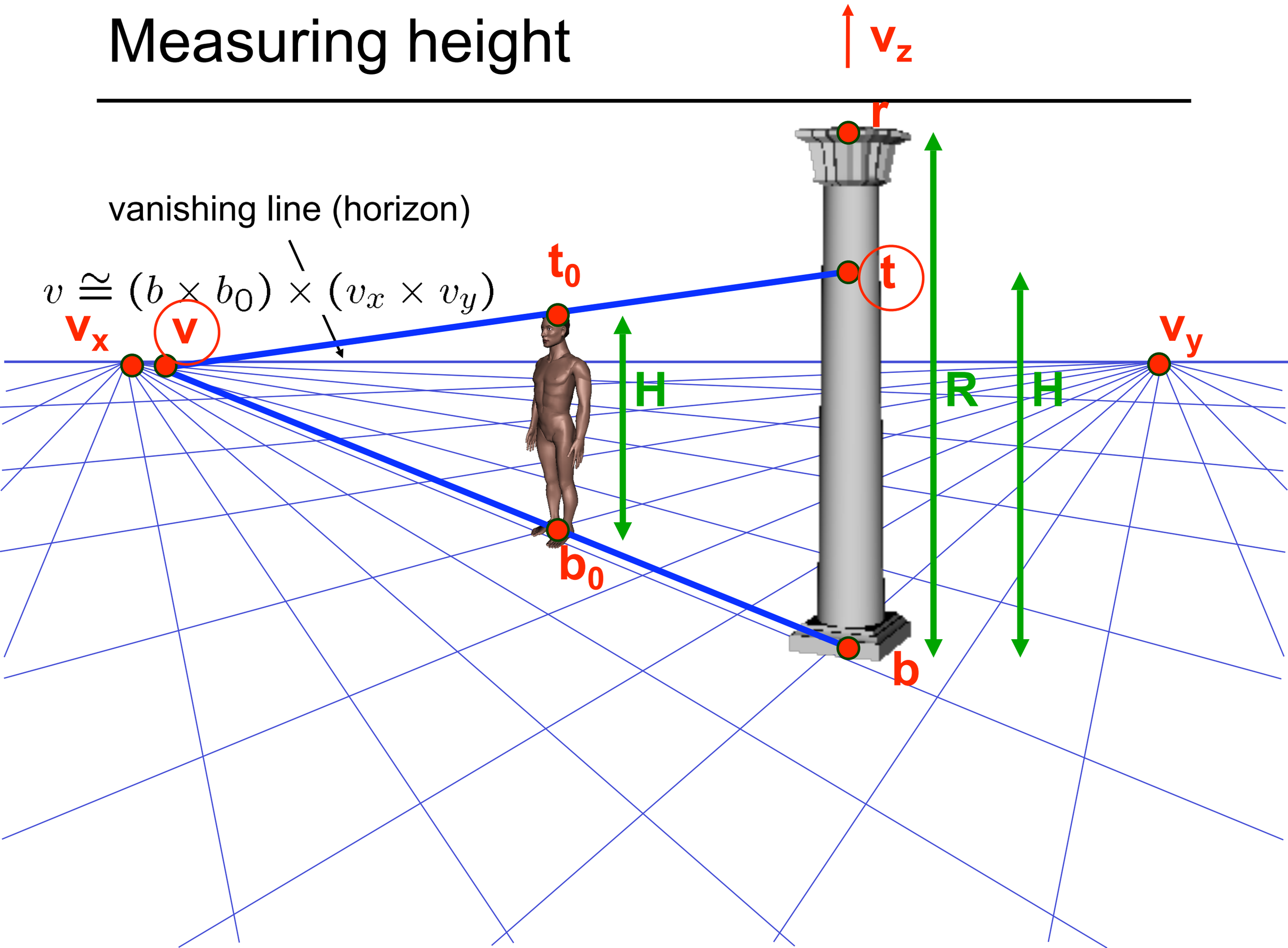
Measuring height



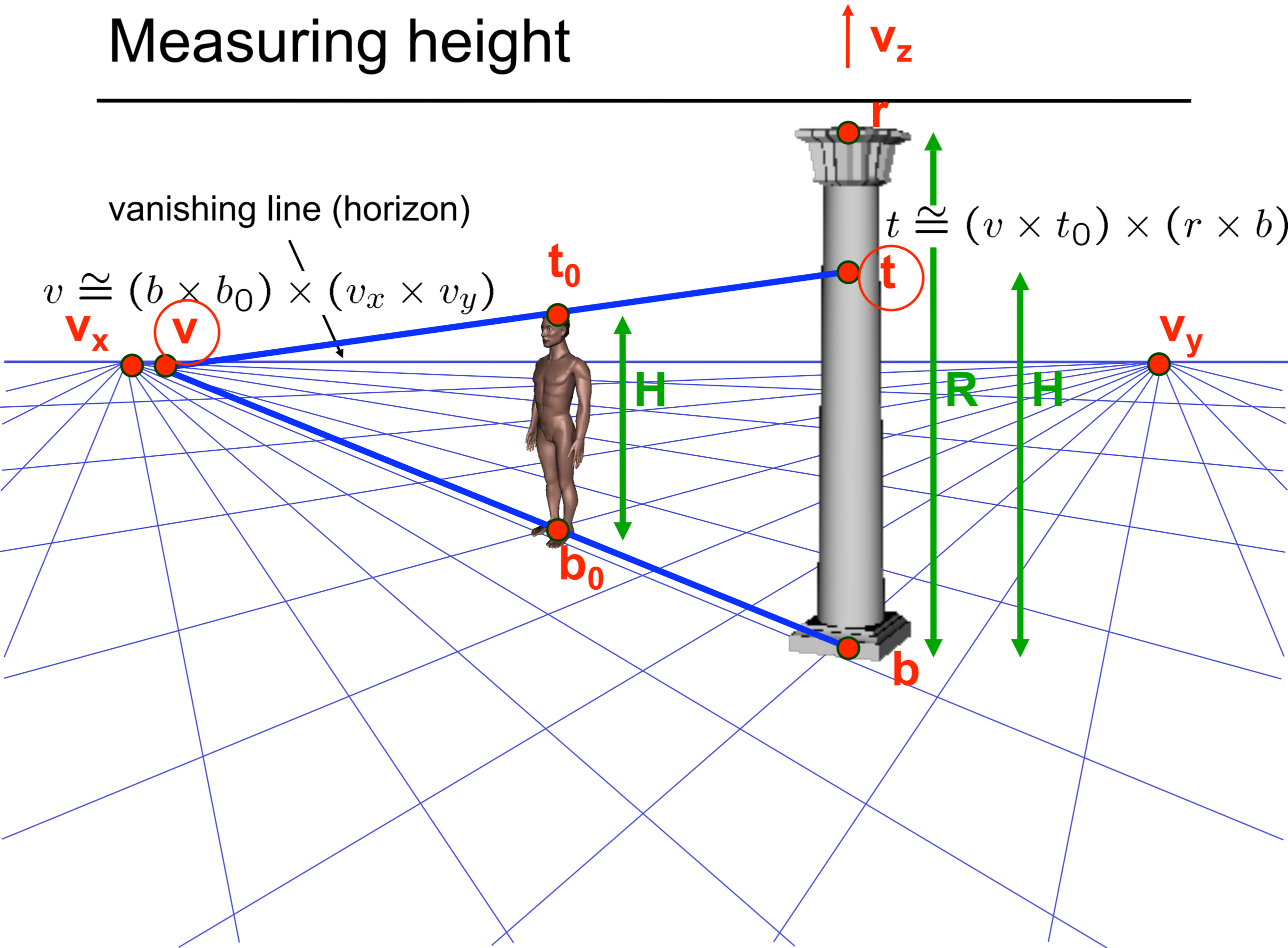
Measuring height



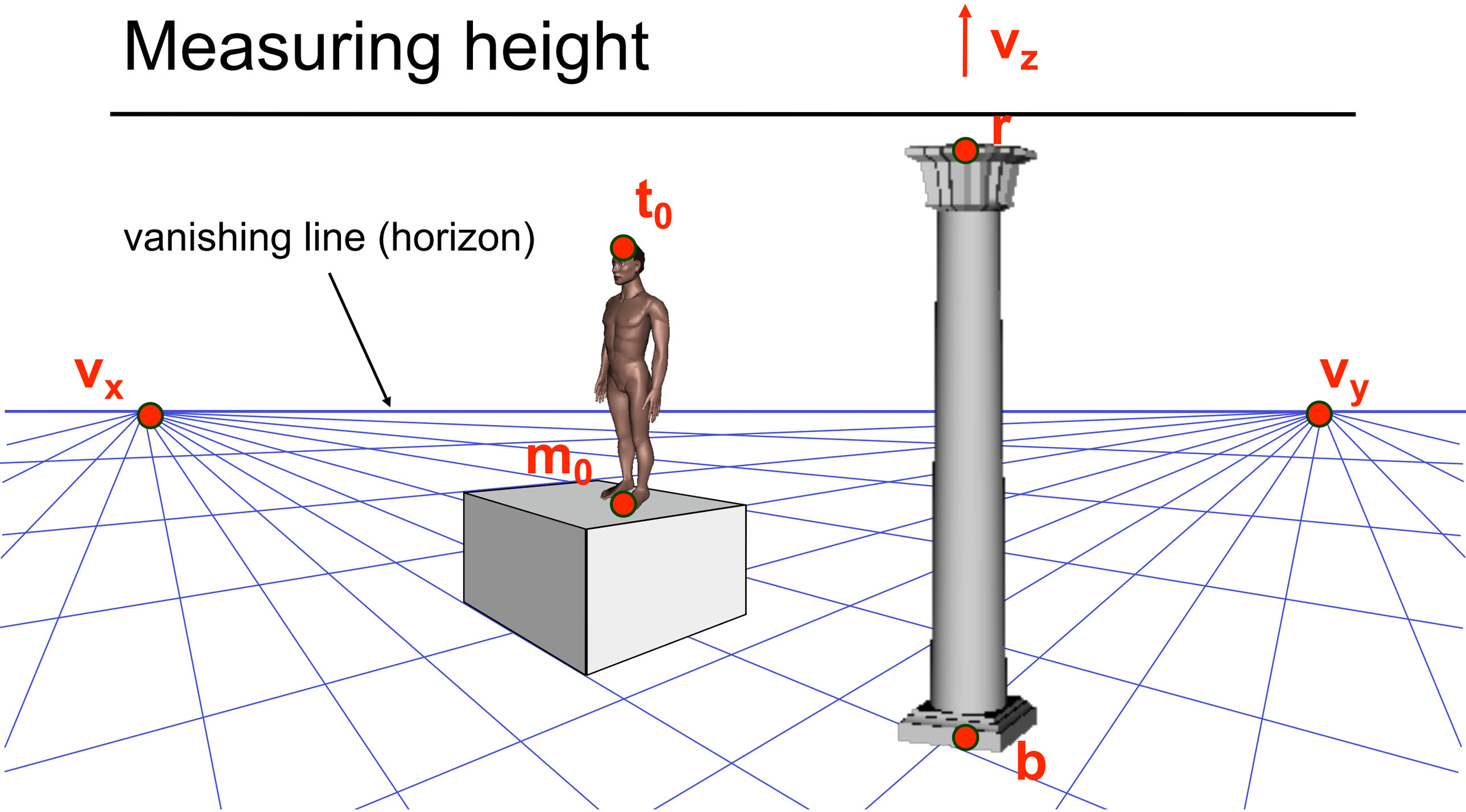
Measuring height



Measuring height



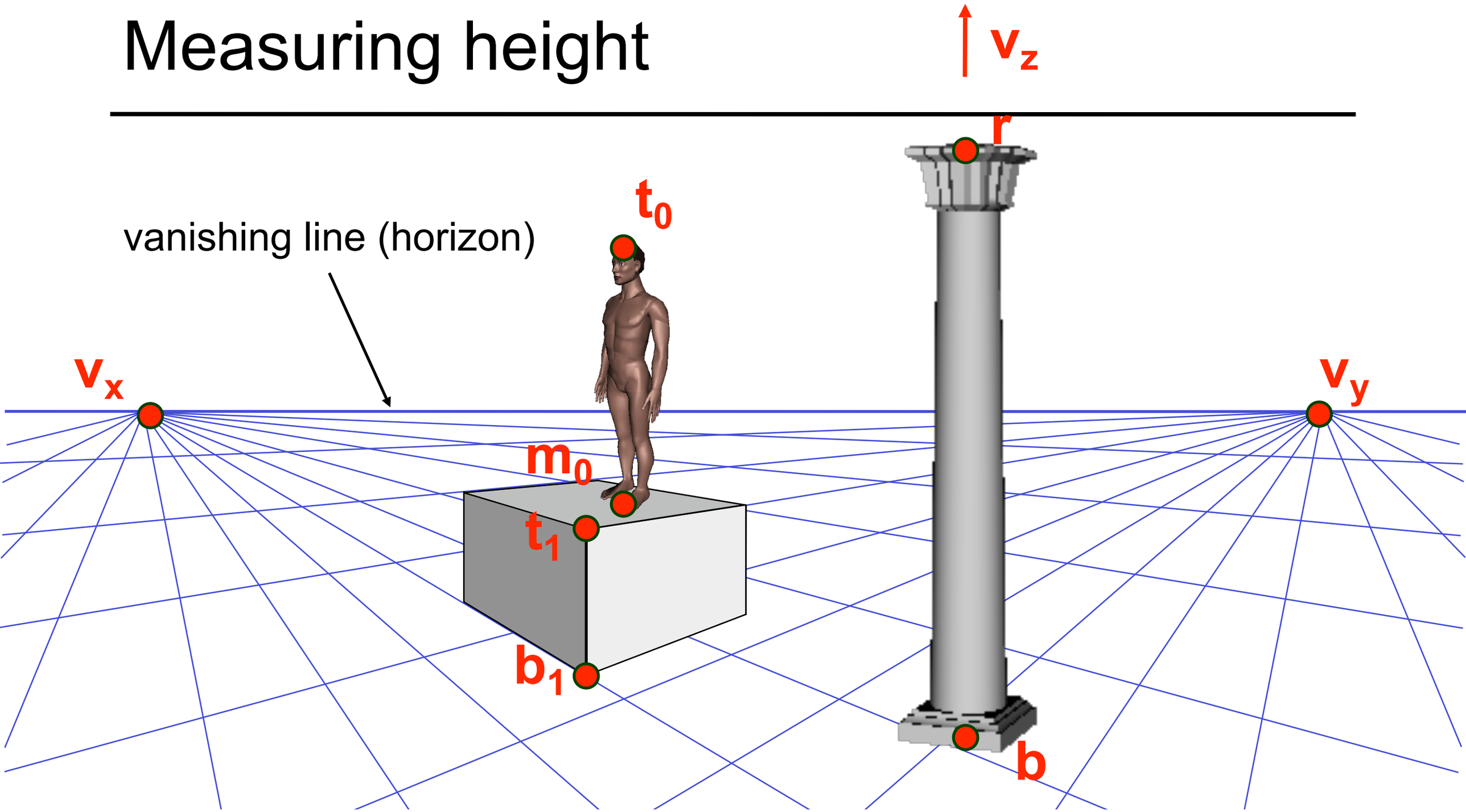
Measuring height



What if the point on the ground plane $\mathbf{b_0}$ is not known?

- Here the guy is standing on the box
- Use one side of the box to help find $\mathbf{b_0}$ as shown above

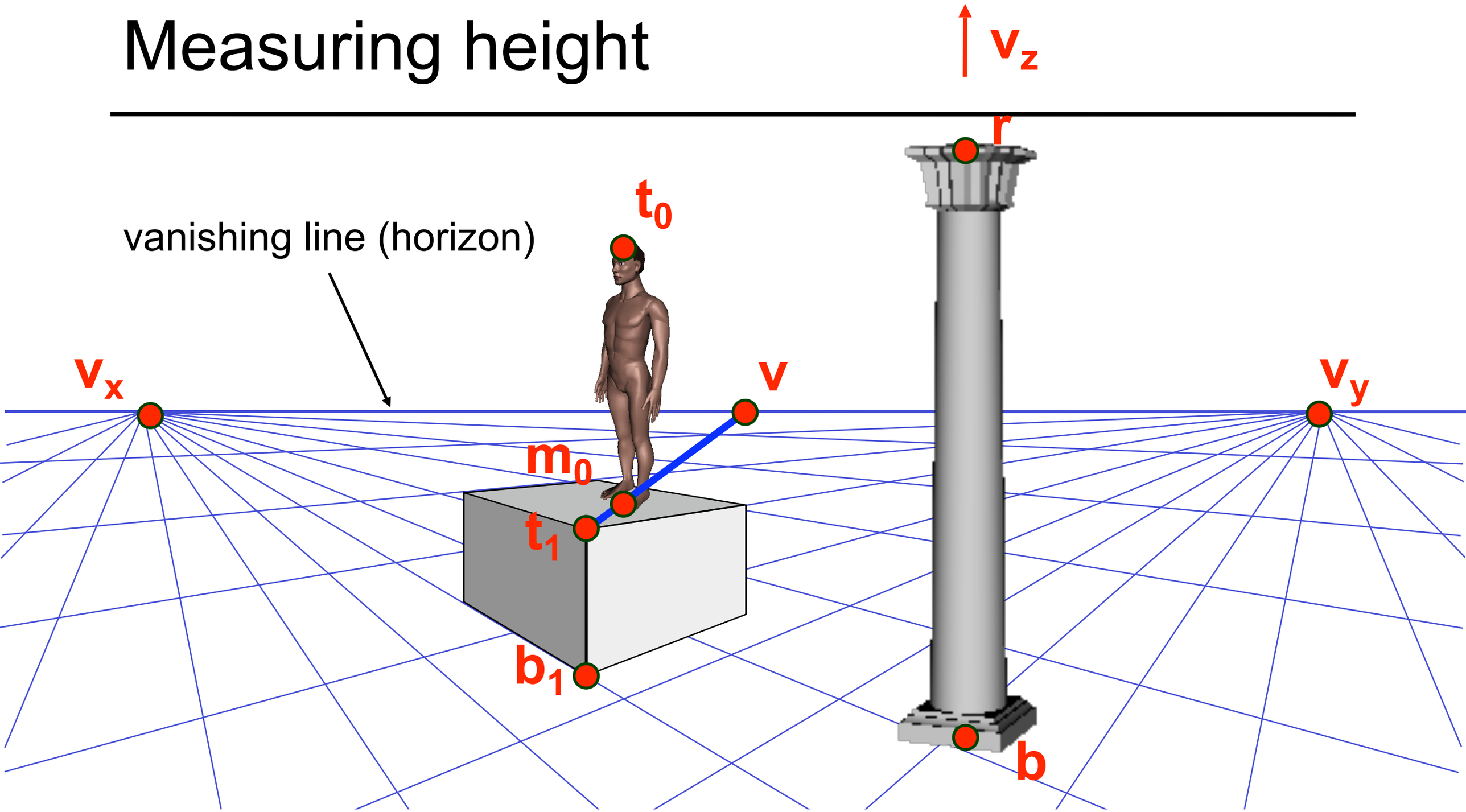
Measuring height



What if the point on the ground plane $\mathbf{b_0}$ is not known?

- Here the guy is standing on the box
- Use one side of the box to help find $\mathbf{b_0}$ as shown above

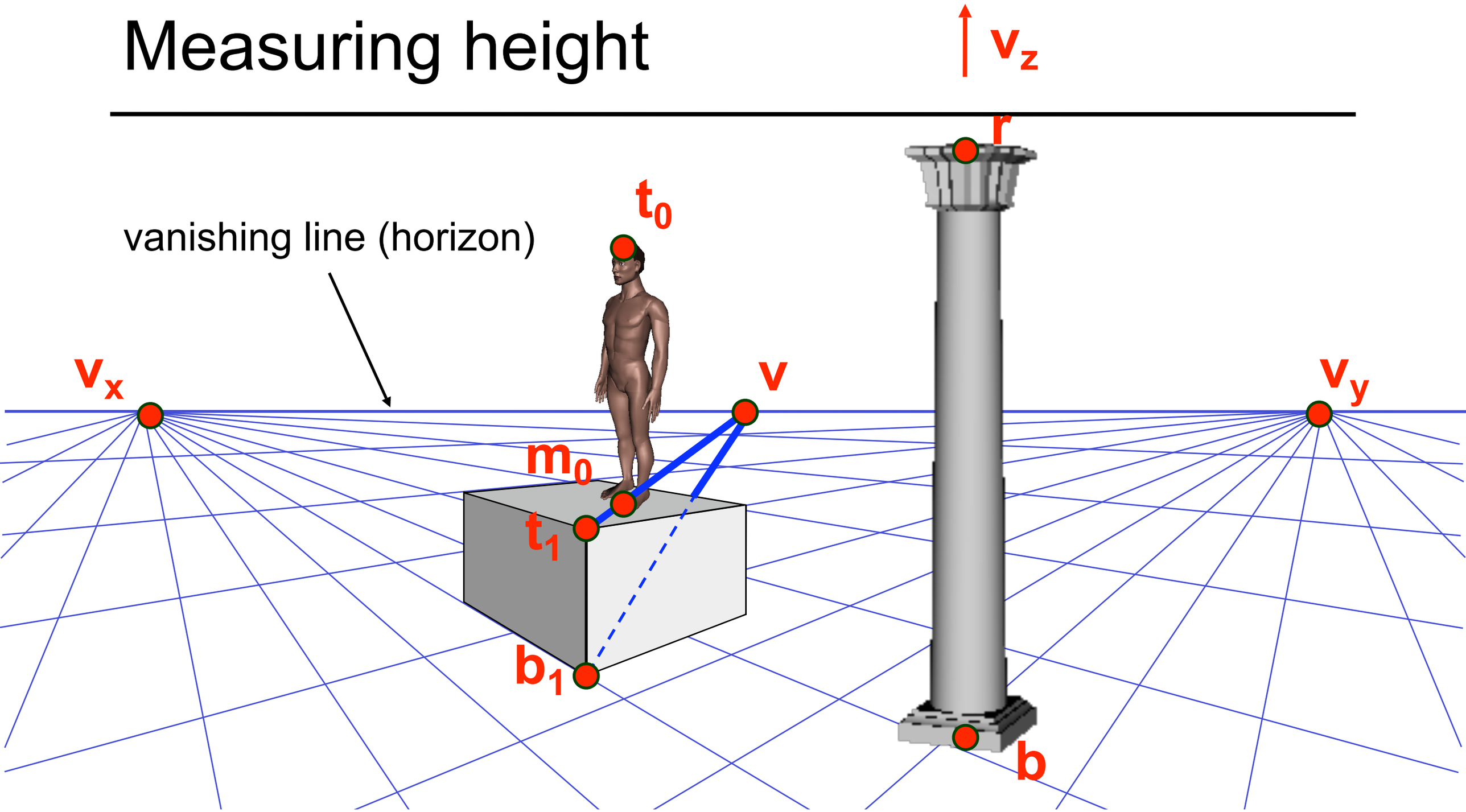
Measuring height



What if the point on the ground plane $\mathbf{b_0}$ is not known?

- Here the guy is standing on the box
- Use one side of the box to help find $\mathbf{b_0}$ as shown above

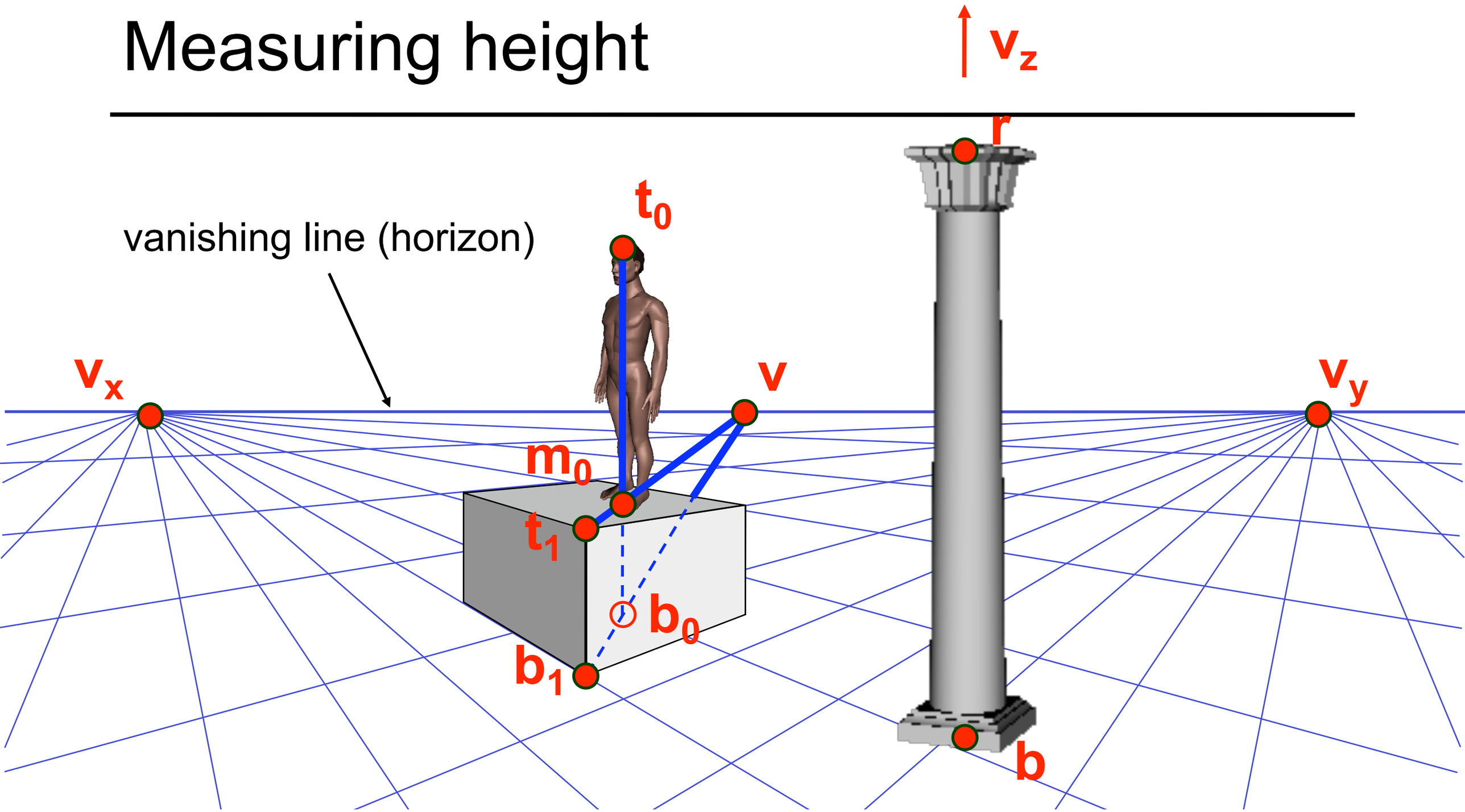
Measuring height



What if the point on the ground plane $\mathbf{b_0}$ is not known?

- Here the guy is standing on the box
- Use one side of the box to help find $\mathbf{b_0}$ as shown above

Measuring height



What if the point on the ground plane $\mathbf{b_0}$ is not known?

- Here the guy is standing on the box
- Use one side of the box to help find $\mathbf{b_0}$ as shown above

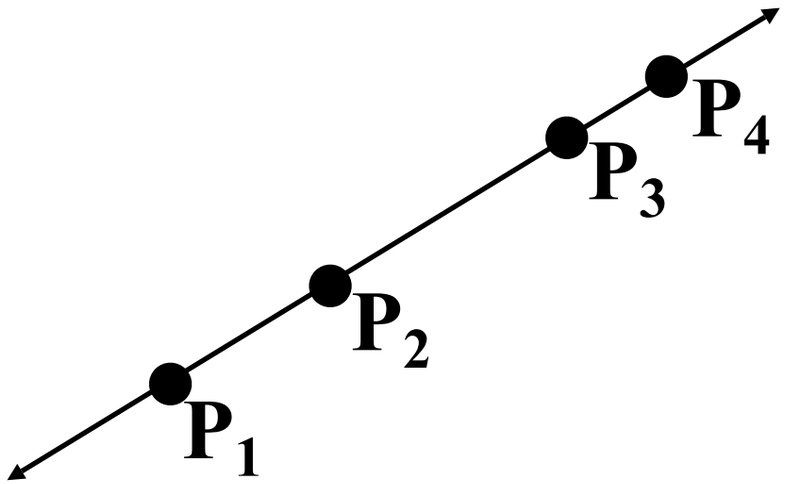
What if v_z is not infinity?

The cross ratio

A Projective Invariant

- Something that does not change under projective transformations (including perspective projection)

The cross-ratio of 4 collinear points



$$\frac{\|\mathbf{P}_3 - \mathbf{P}_1\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_3 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_1\|}$$

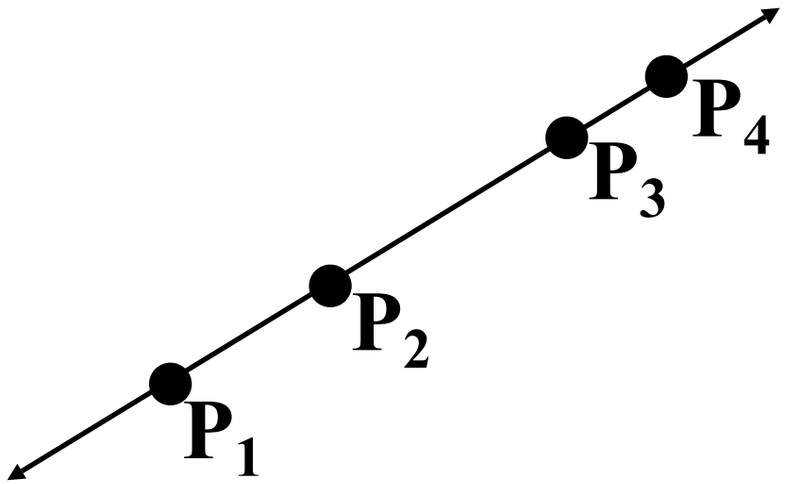
$$\mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

The cross ratio

A Projective Invariant

- Something that does not change under projective transformations (including perspective projection)

The cross-ratio of 4 collinear points



$$\frac{\|\mathbf{P}_3 - \mathbf{P}_1\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_3 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_1\|}$$

$$\mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Can permute the point ordering

- 4! = 24 different orders (but only 6 distinct values)

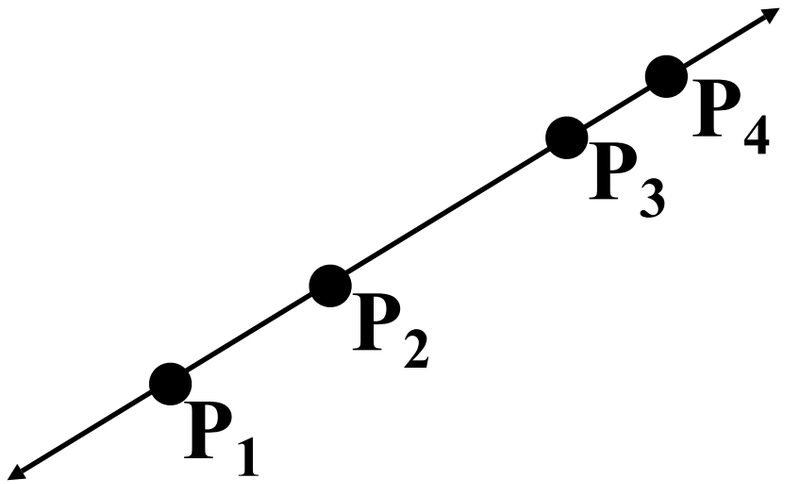
This is the fundamental invariant of projective geometry

The cross ratio

A Projective Invariant

- Something that does not change under projective transformations (including perspective projection)

The cross-ratio of 4 collinear points



$$\frac{\|\mathbf{P}_3 - \mathbf{P}_1\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_3 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_1\|}$$

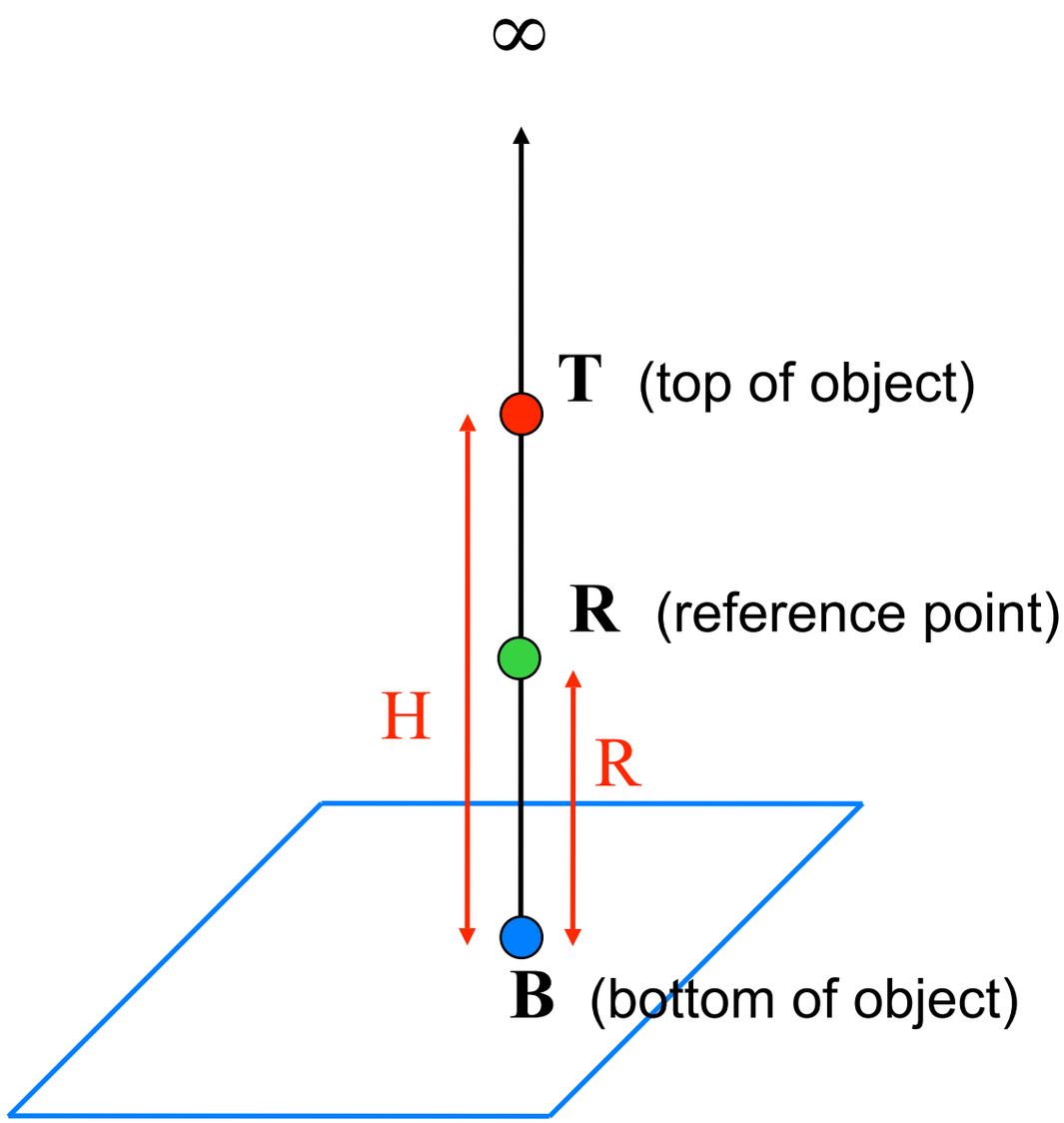
$$\mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Can permute the point ordering $\dfrac{\|\mathbf{P}_1 - \mathbf{P}_3\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_1 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_3\|}$

- 4! = 24 different orders (but only 6 distinct values)

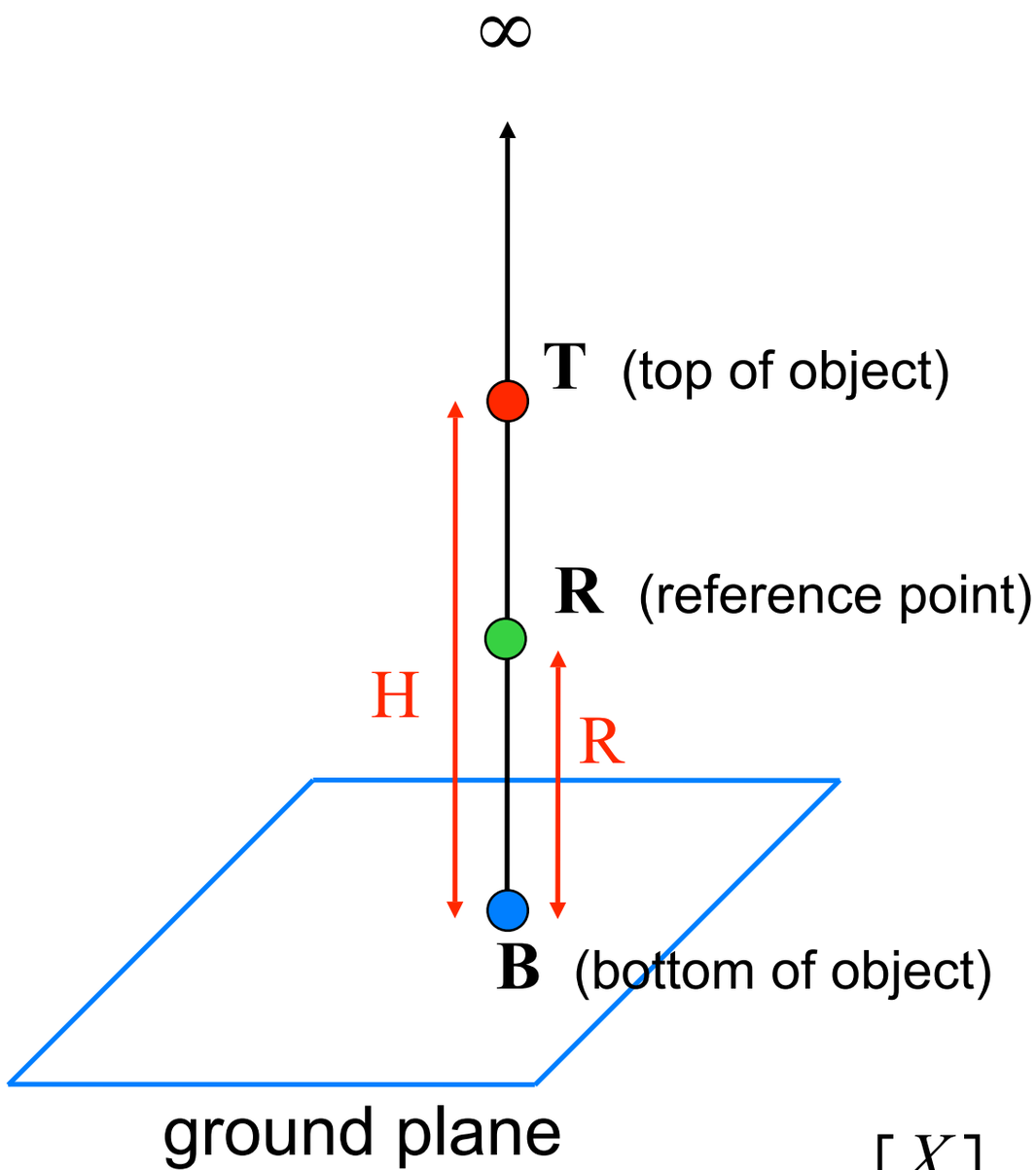
This is the fundamental invariant of projective geometry

Measuring height



scene points represented as $\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$ image points as $\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Measuring height

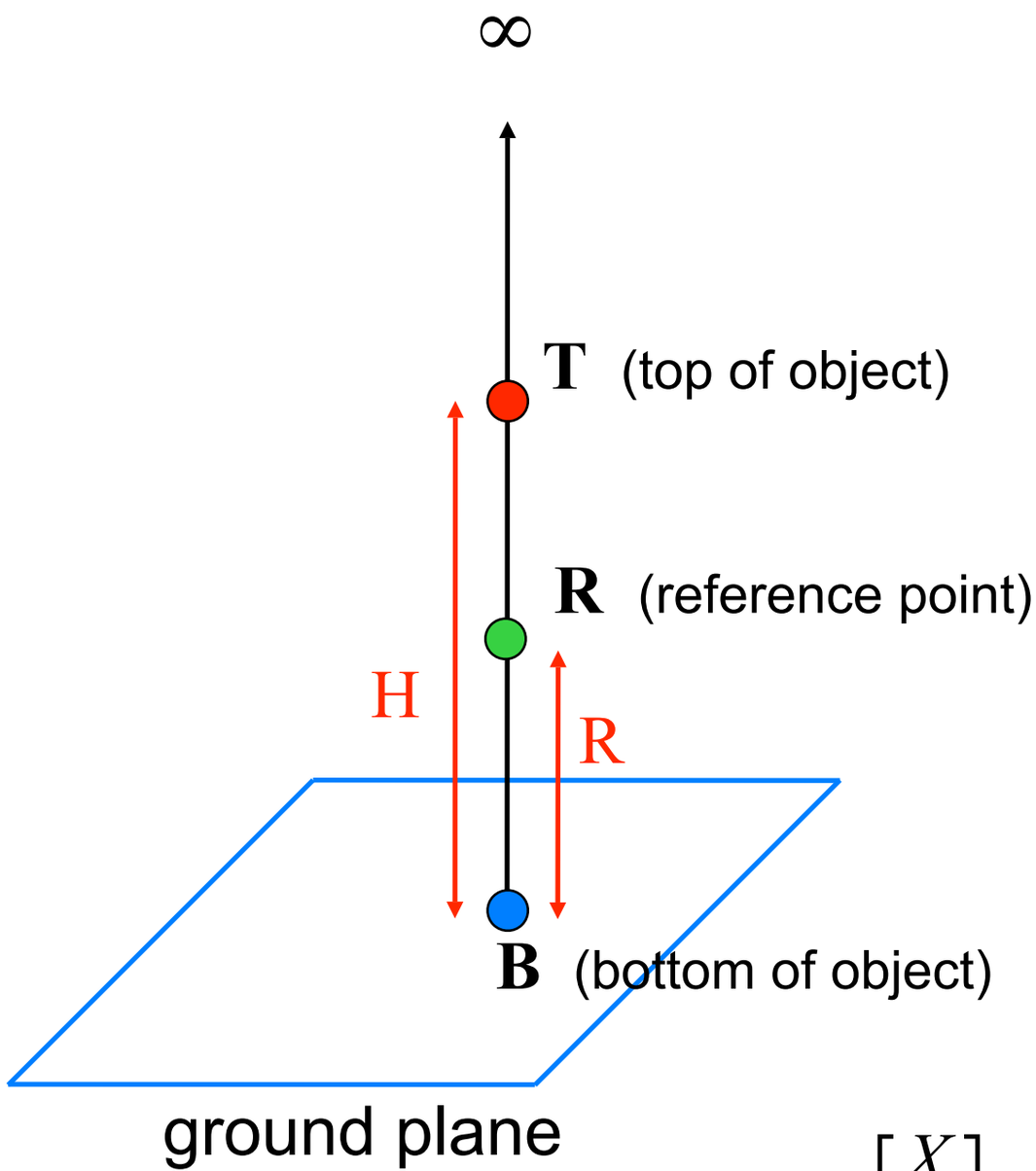


$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}$$

scene cross ratio

scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

Measuring height

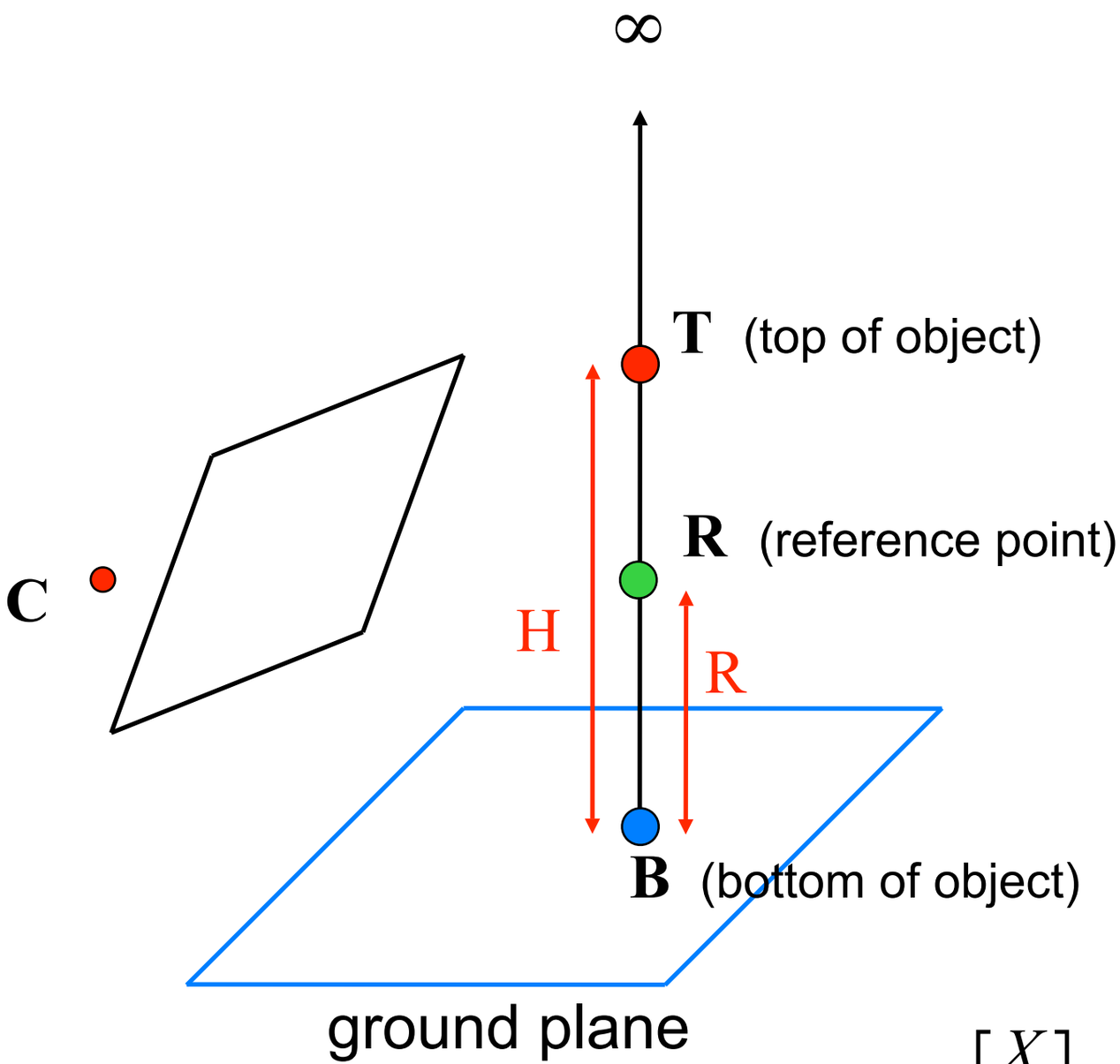


$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

scene cross ratio

scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

Measuring height

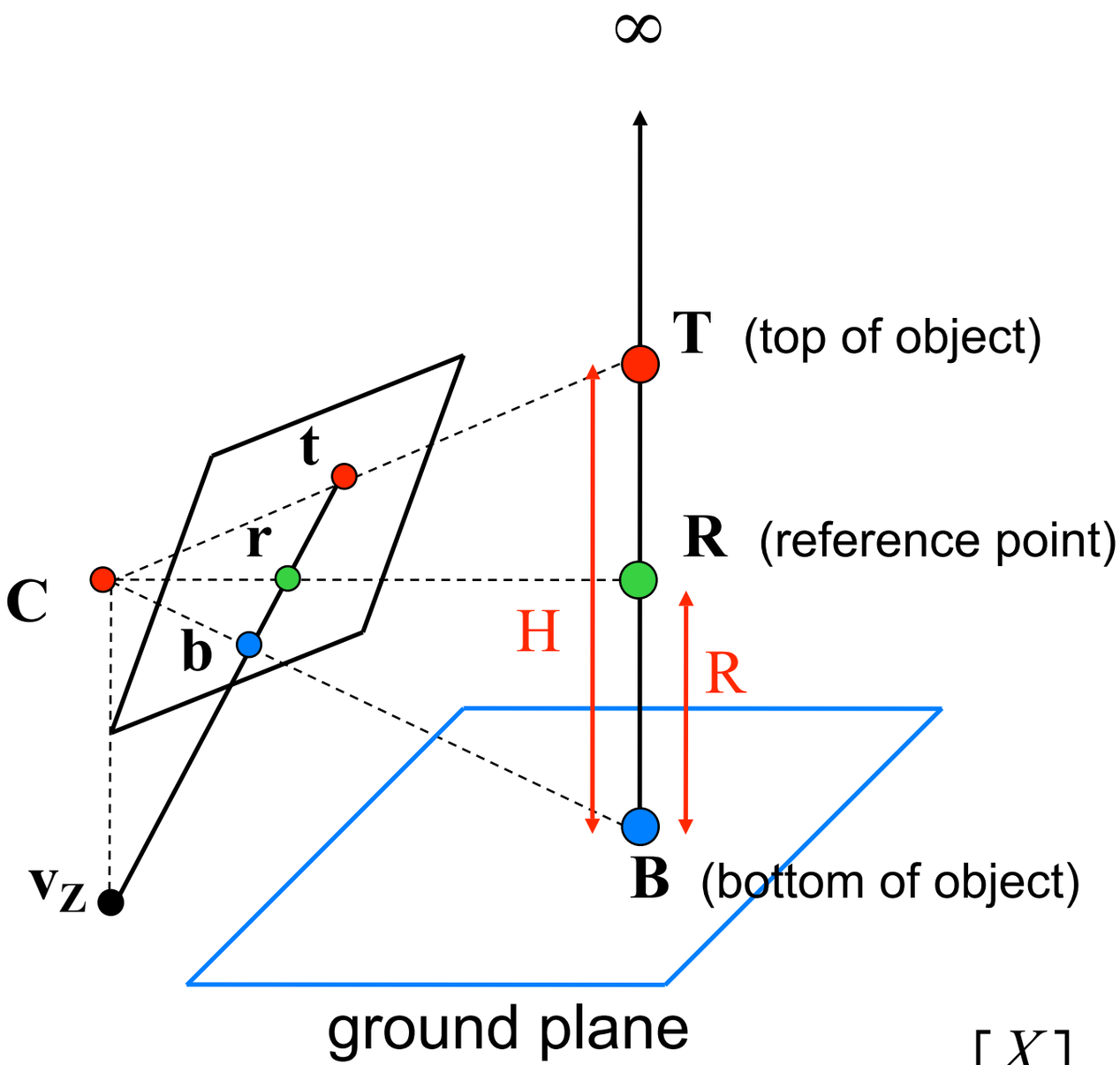


$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

scene cross ratio

scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

Measuring height

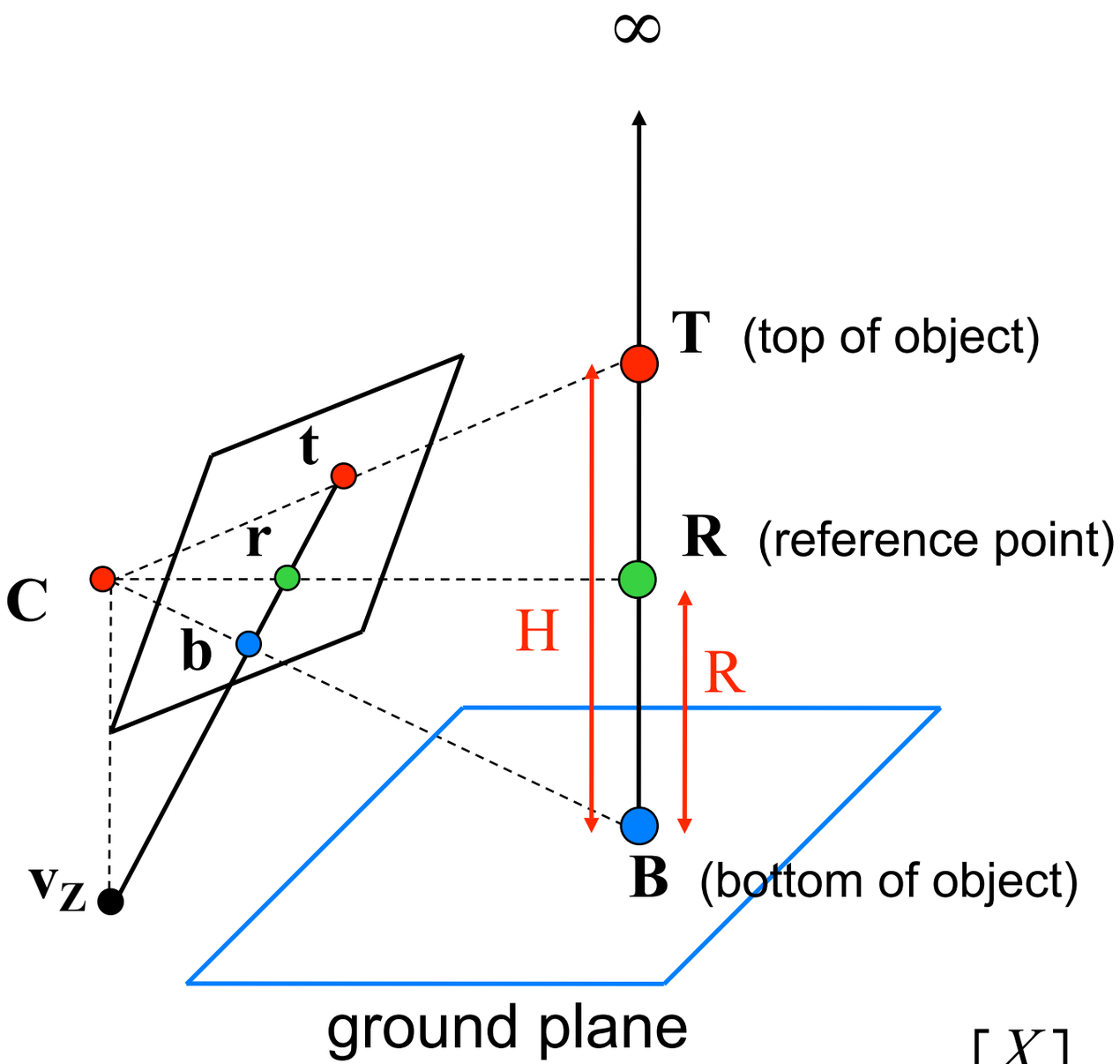


$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

scene cross ratio

scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

Measuring height



$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

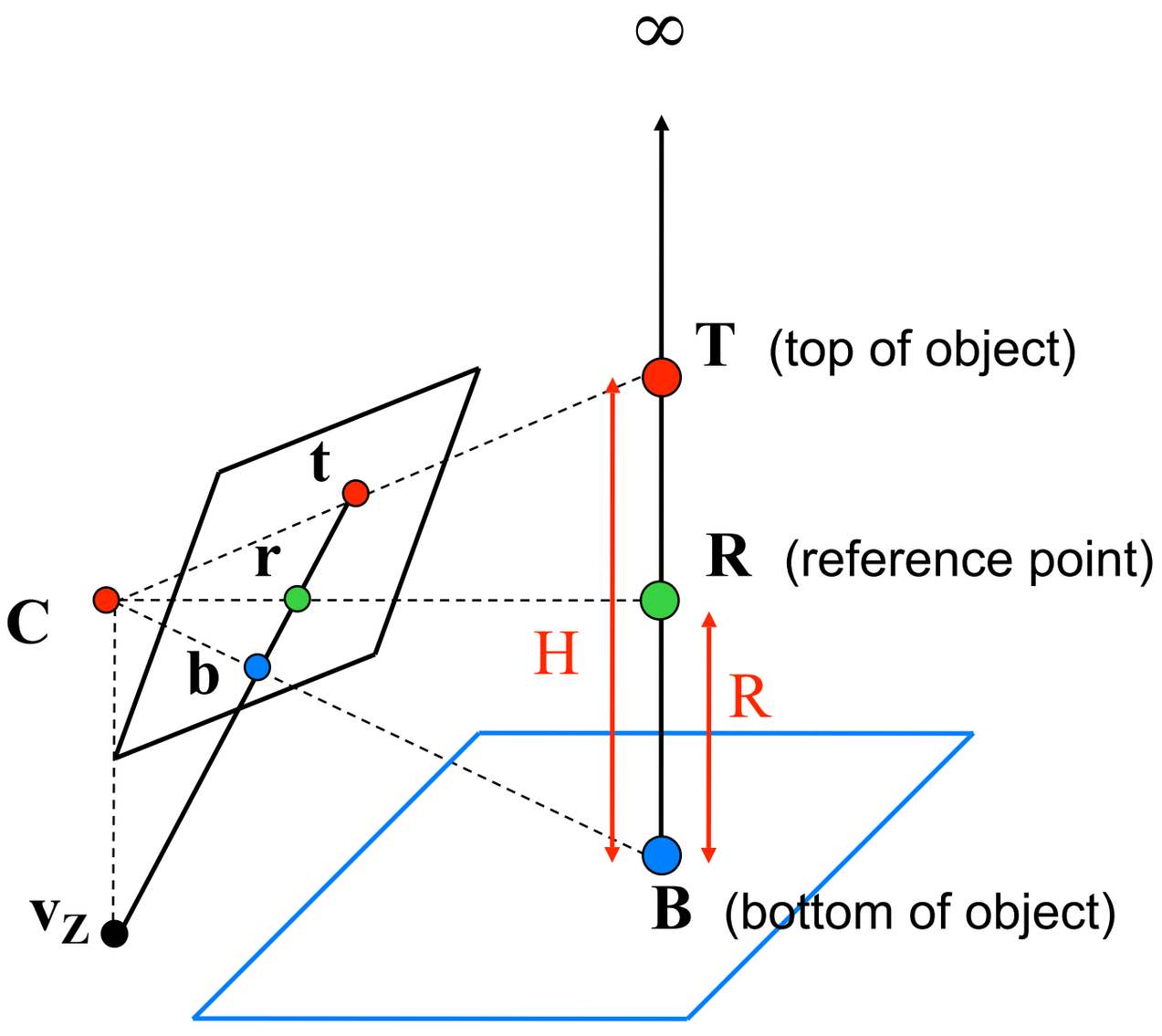
scene cross ratio

$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}$$

image cross ratio

scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

Measuring height



$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

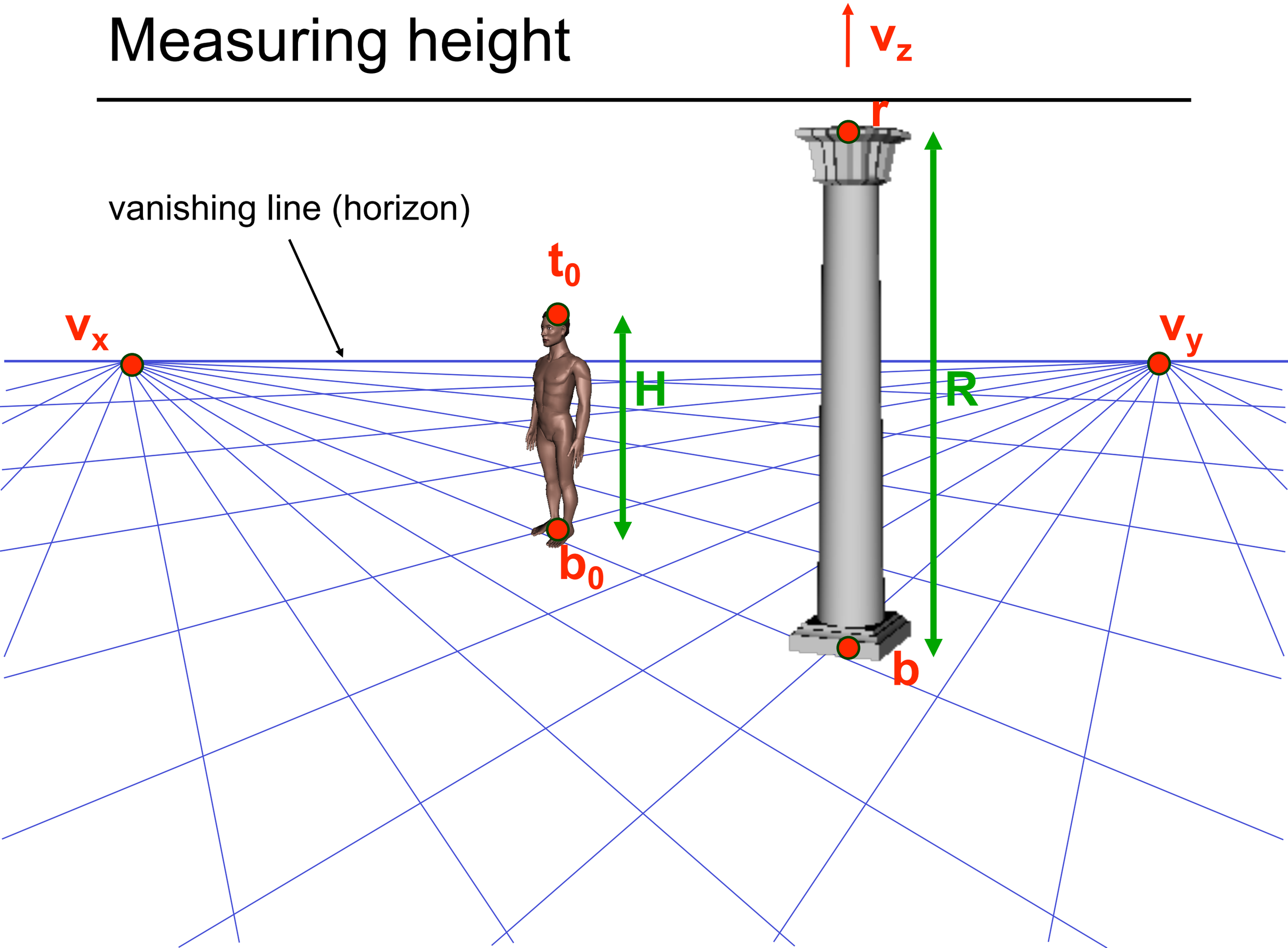
scene cross ratio

$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$

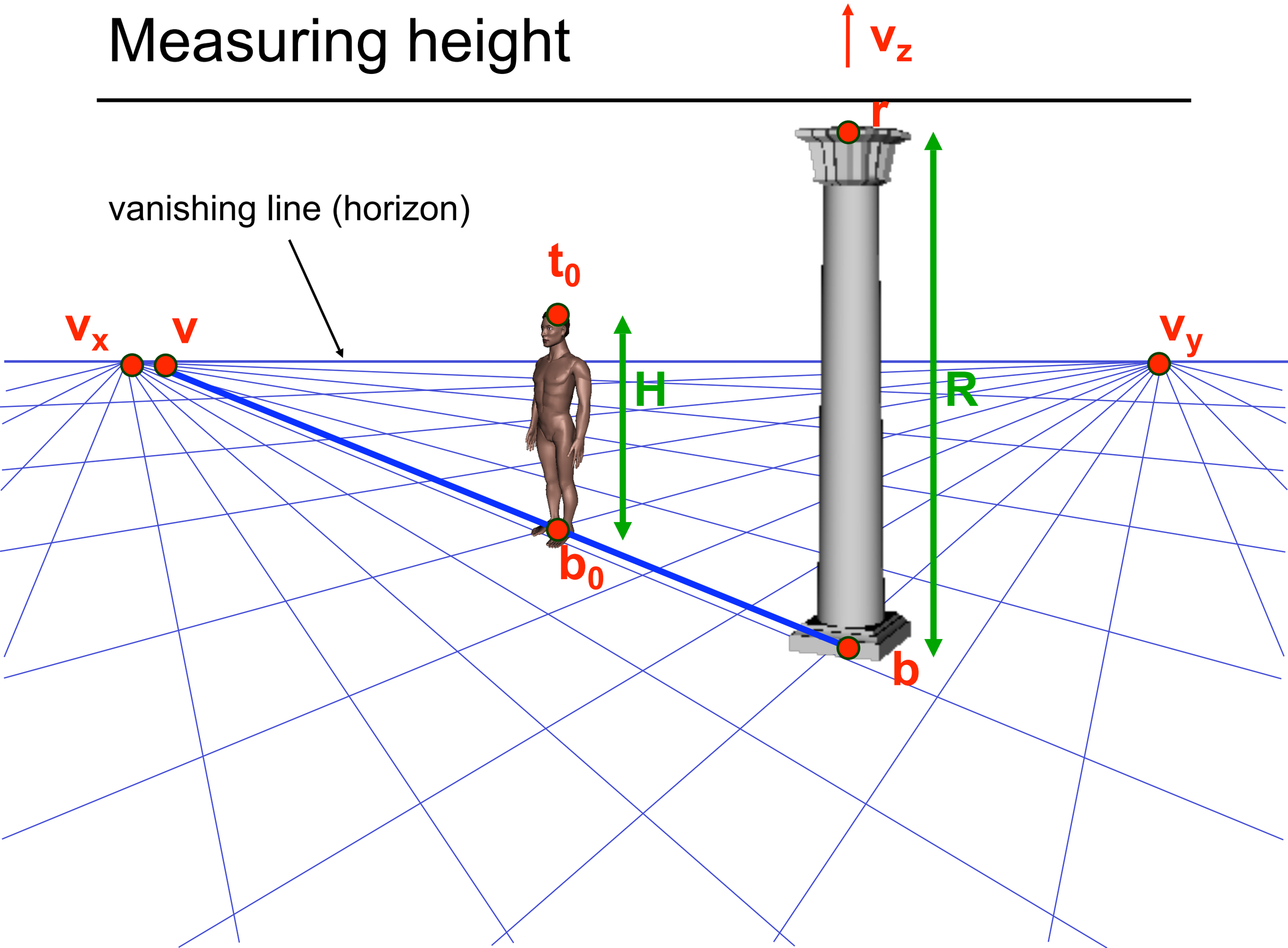
image cross ratio

scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

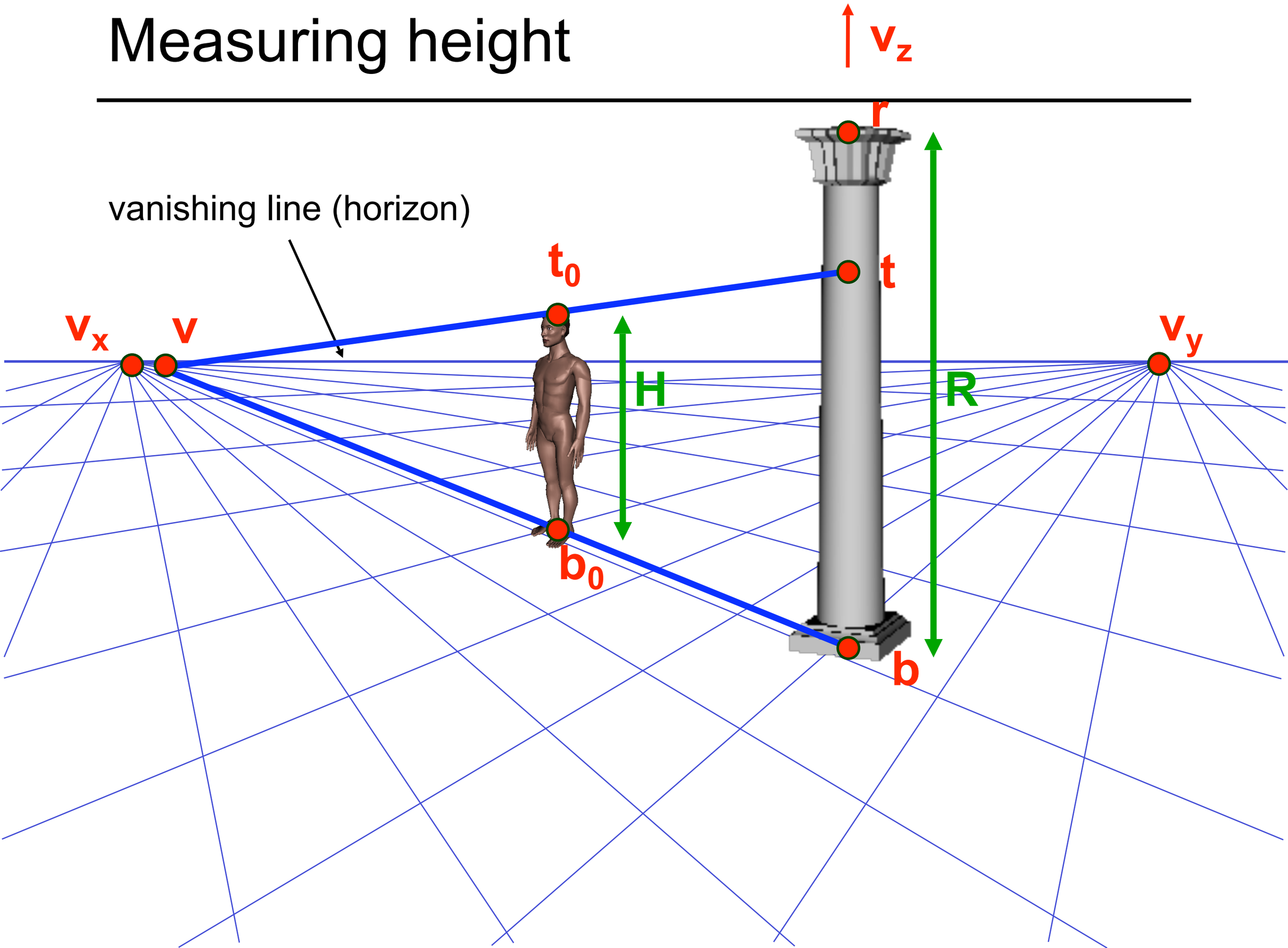
Measuring height



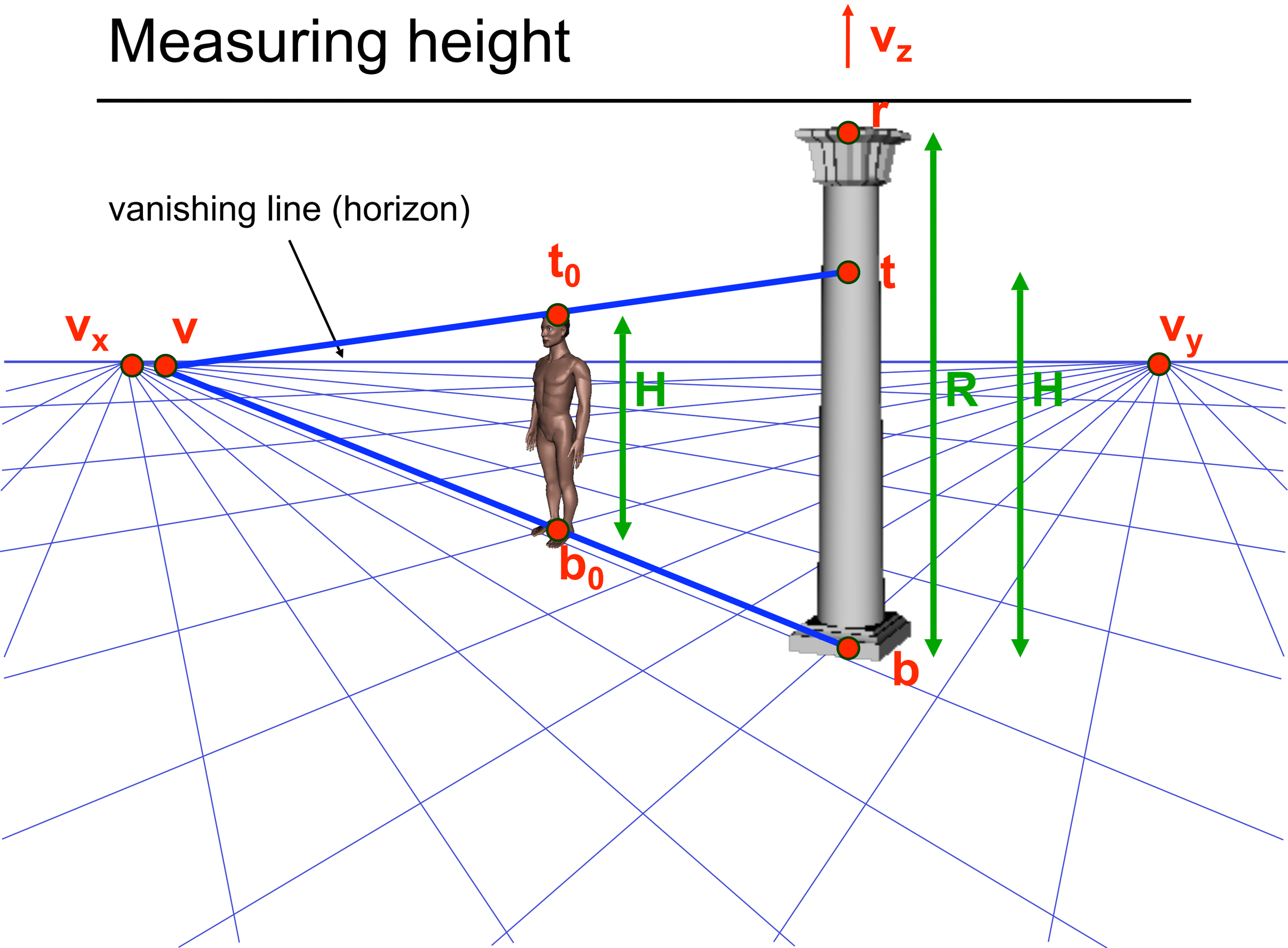
Measuring height



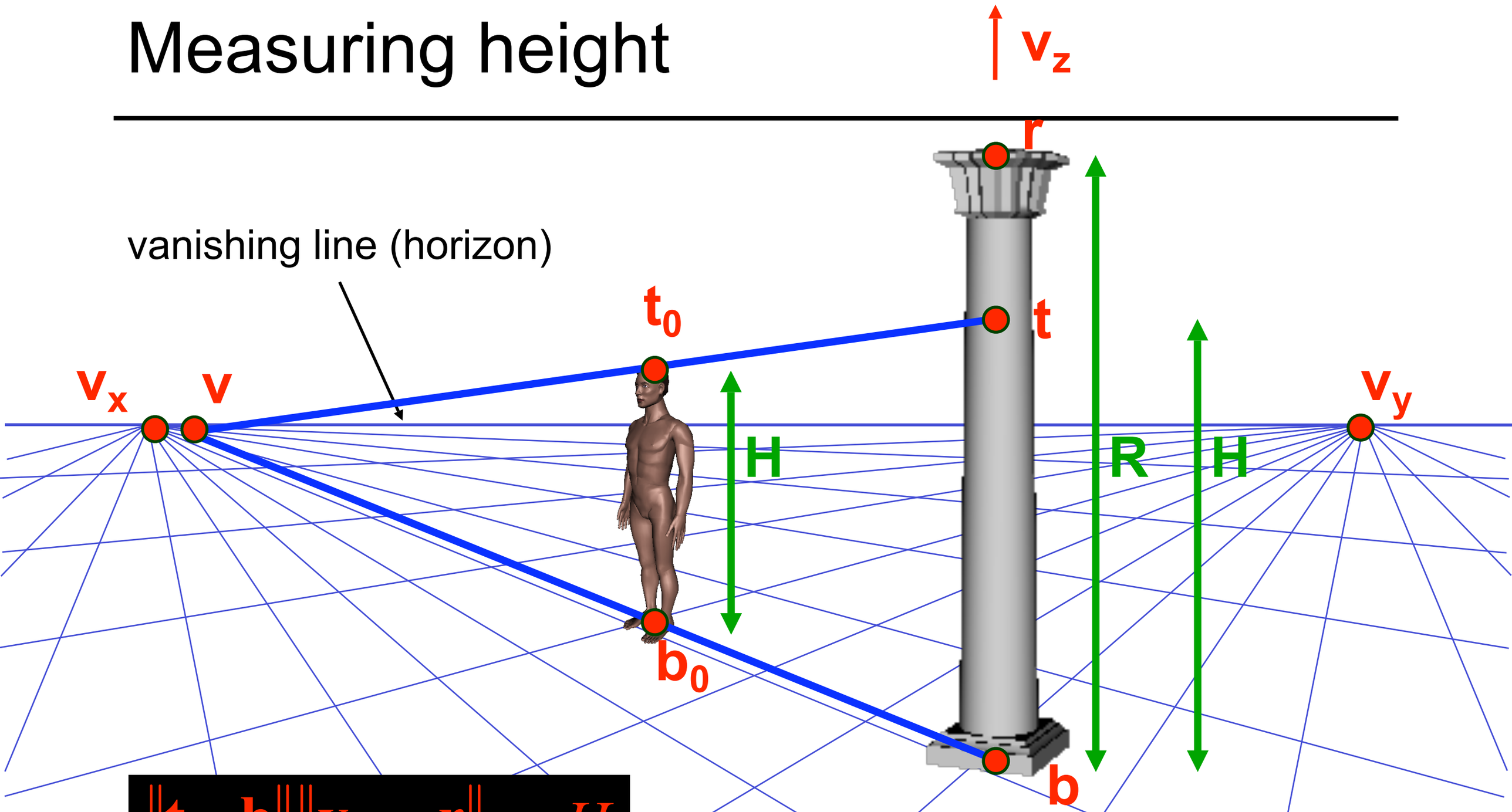
Measuring height



Measuring height



Measuring height

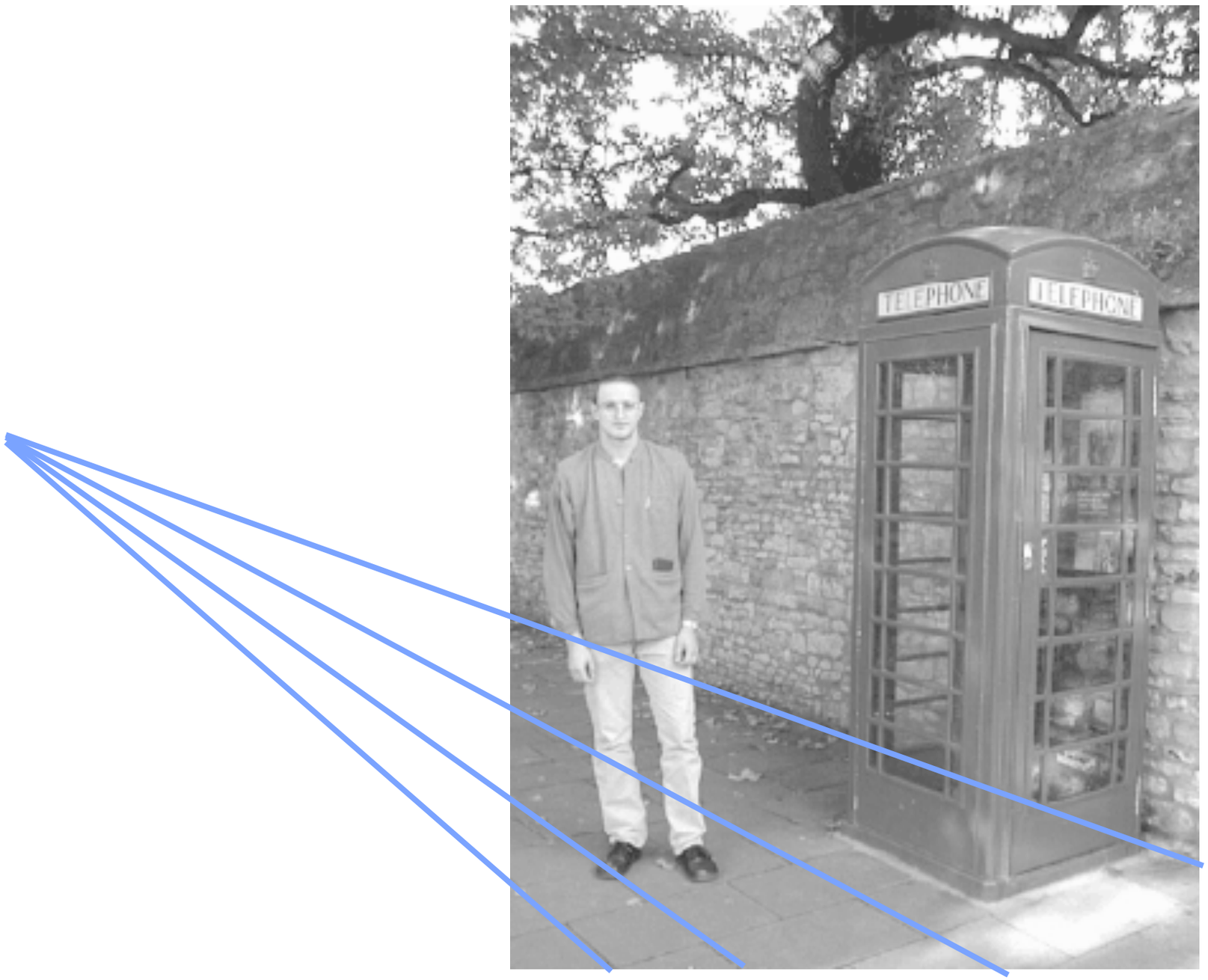


$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_z-\mathbf{t}\|}=\frac{H}{R}$$

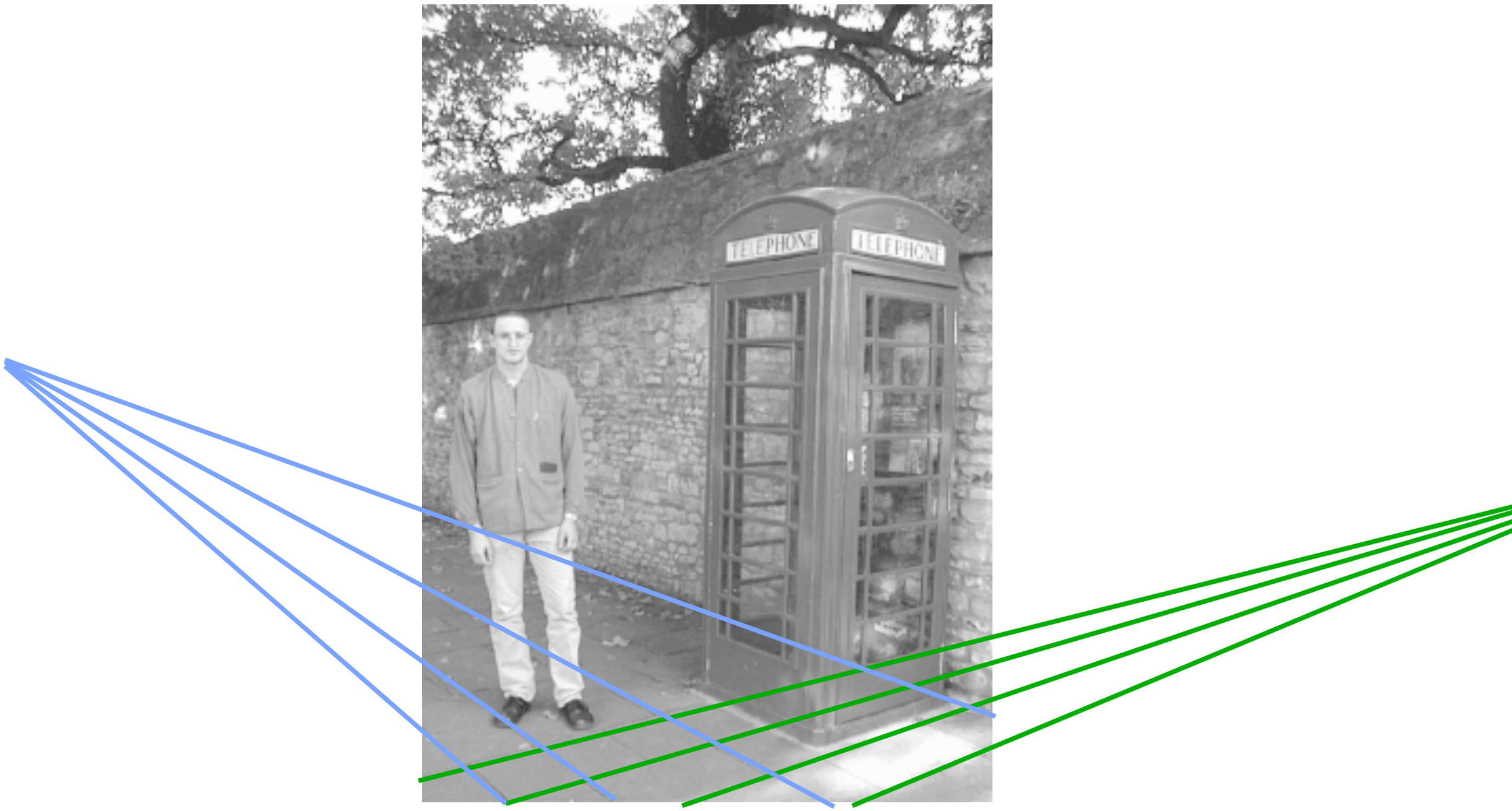
image cross ratio

Measuring heights of people

Measuring heights of people

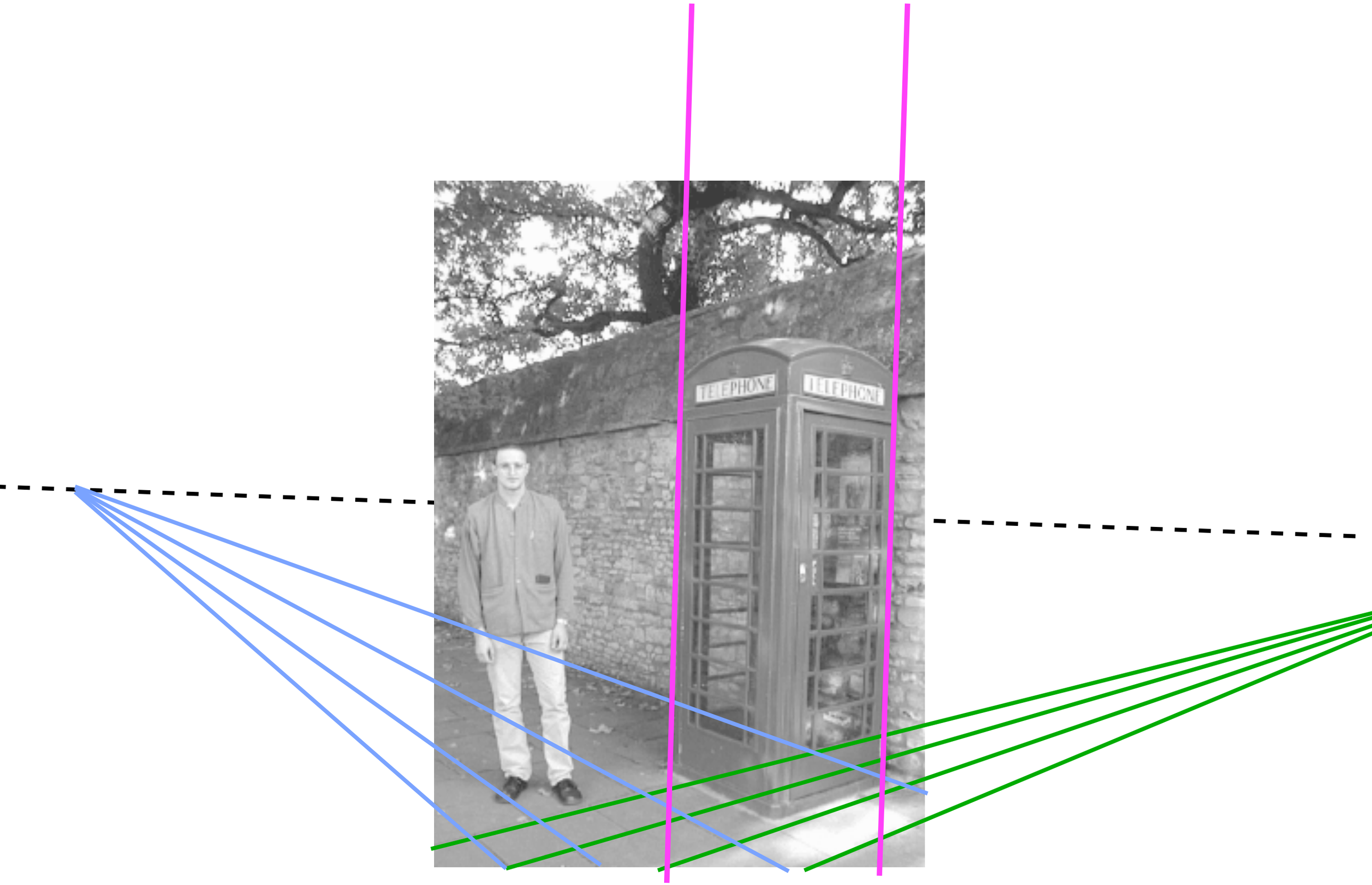


Measuring heights of people

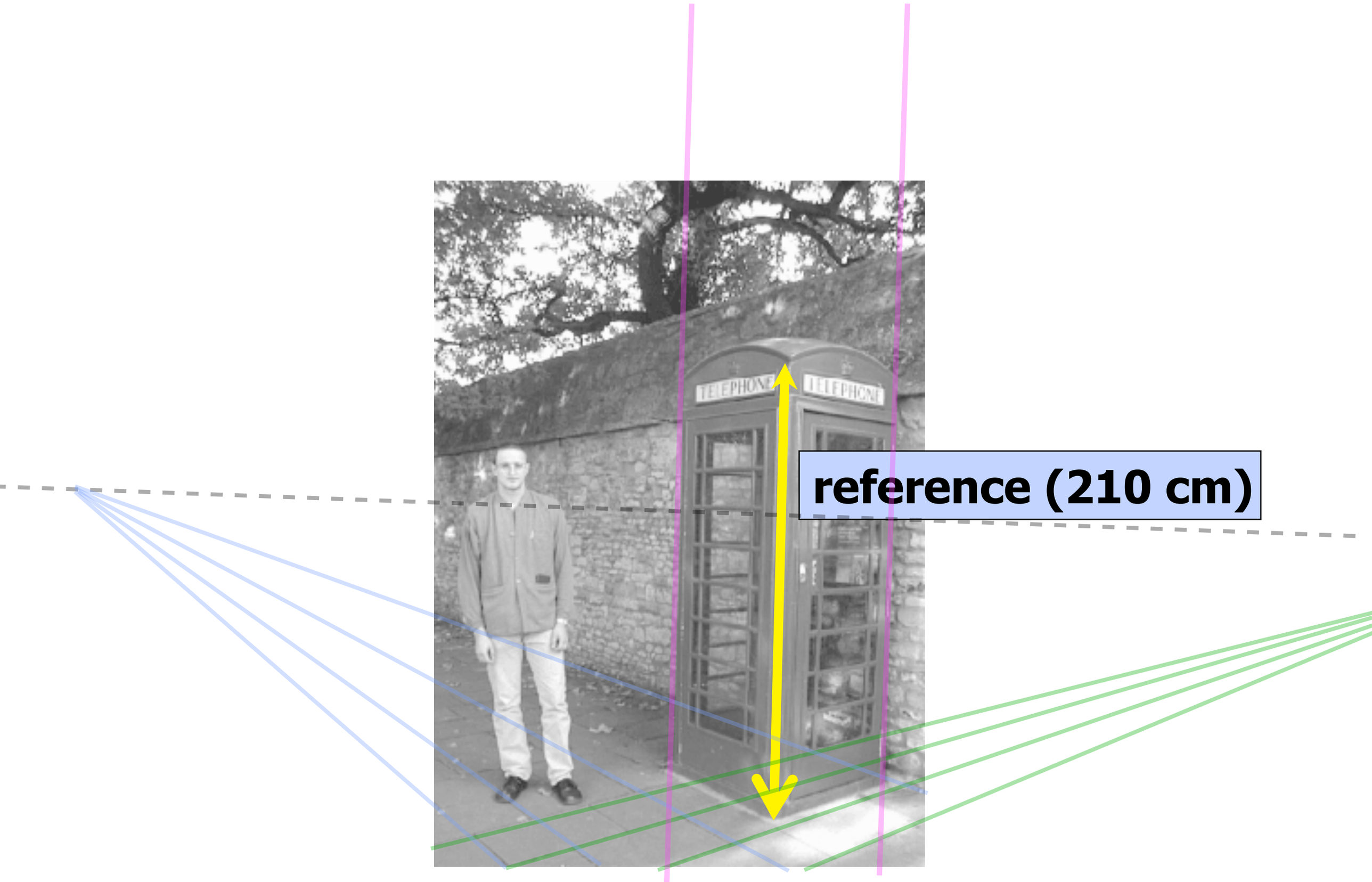


Measuring heights of people

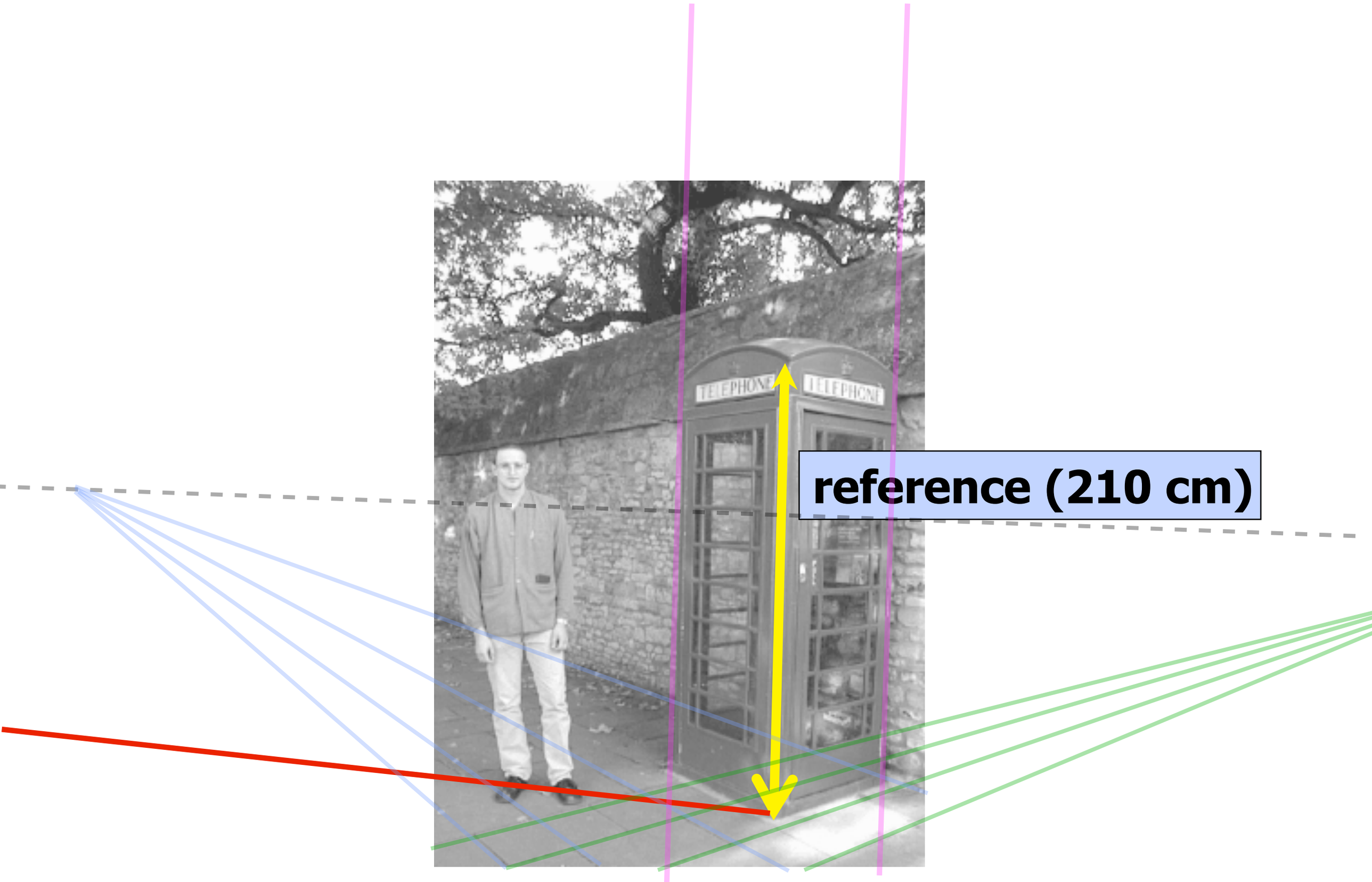
Measuring heights of people



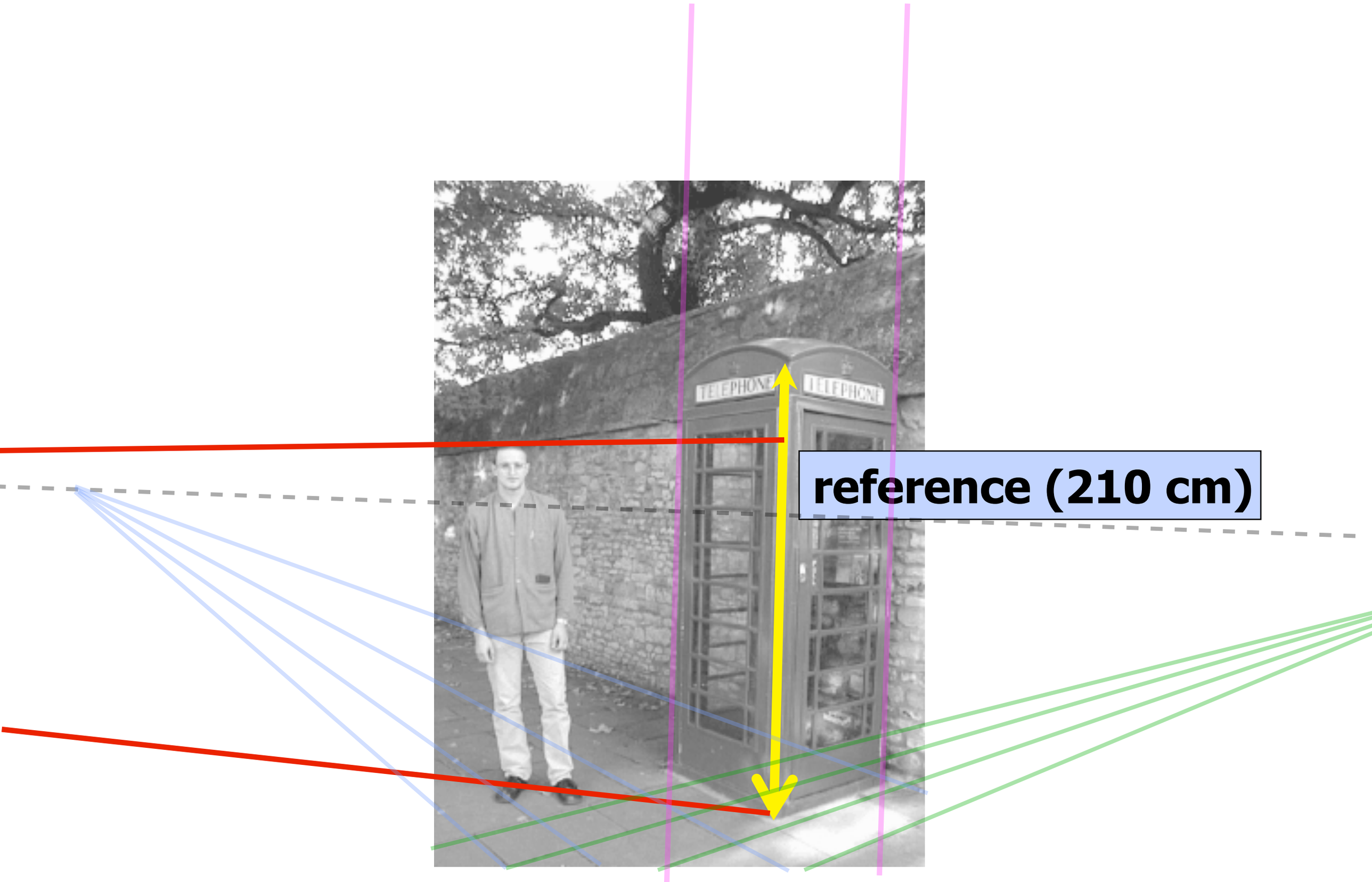
Measuring heights of people



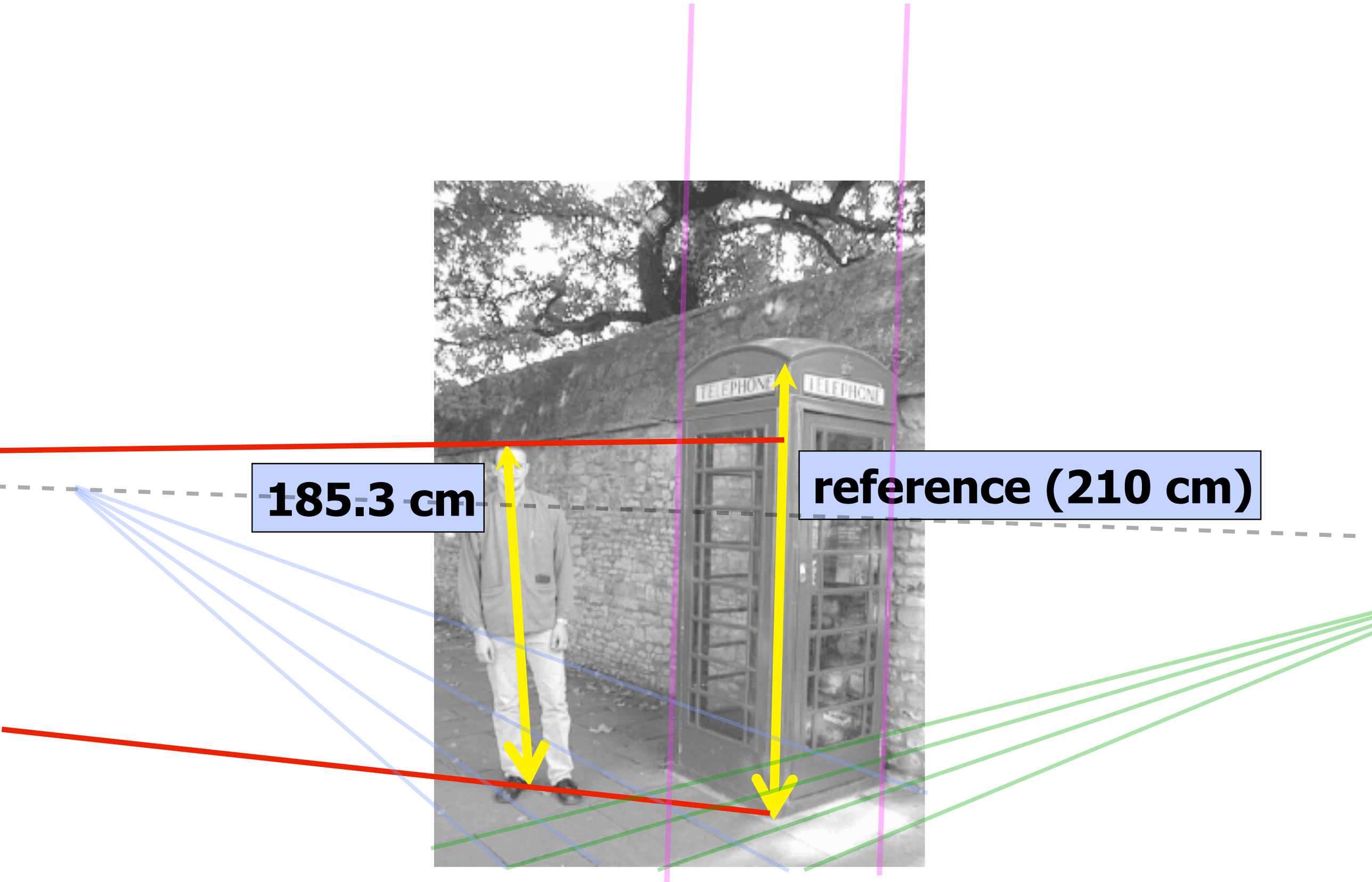
Measuring heights of people



Measuring heights of people



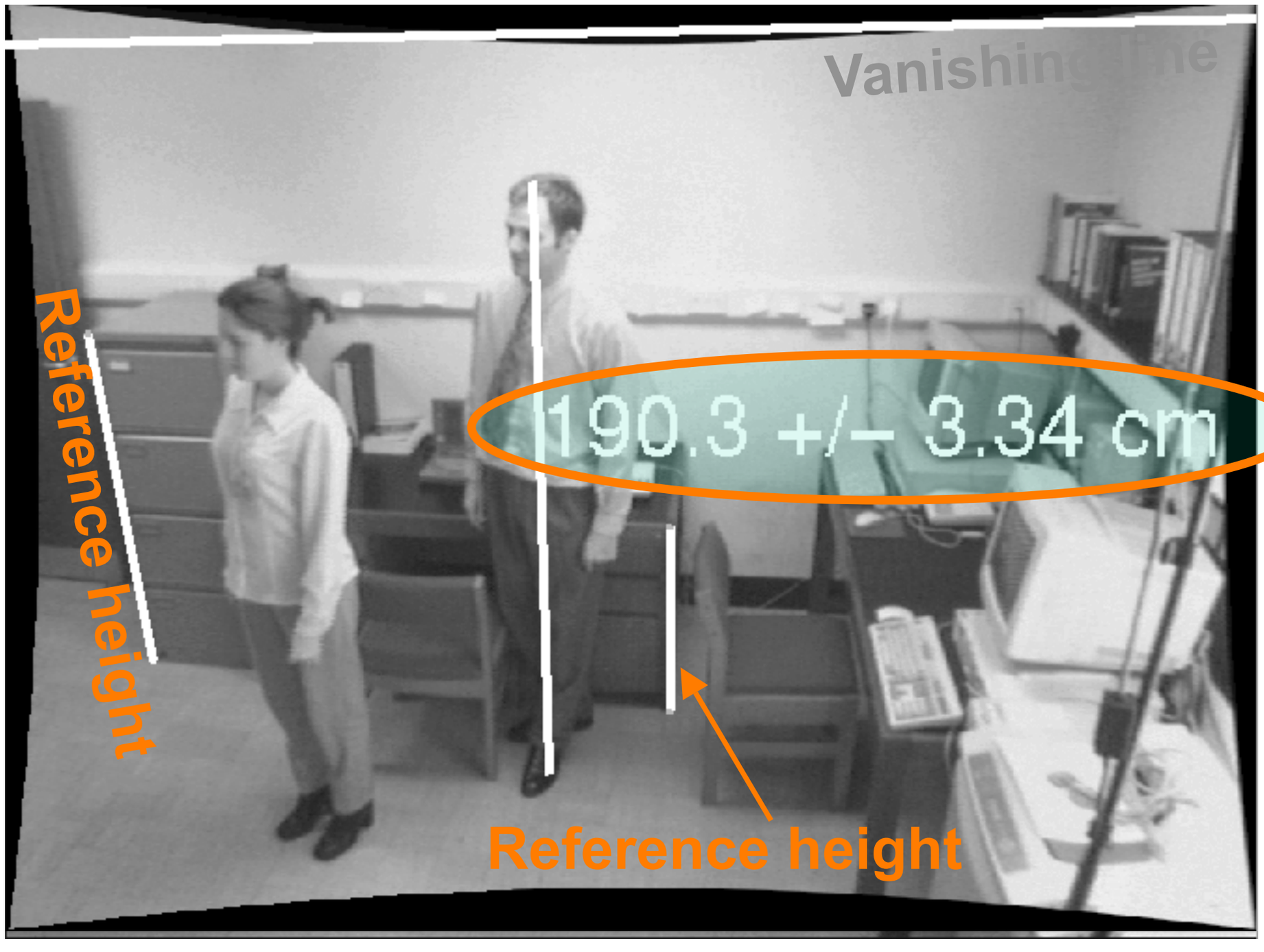
Measuring heights of people



Forensic Science: measuring heights of suspects

Forensic Science: measuring heights of suspects

Forensic Science: measuring heights of suspects



Assessing geometric accuracy

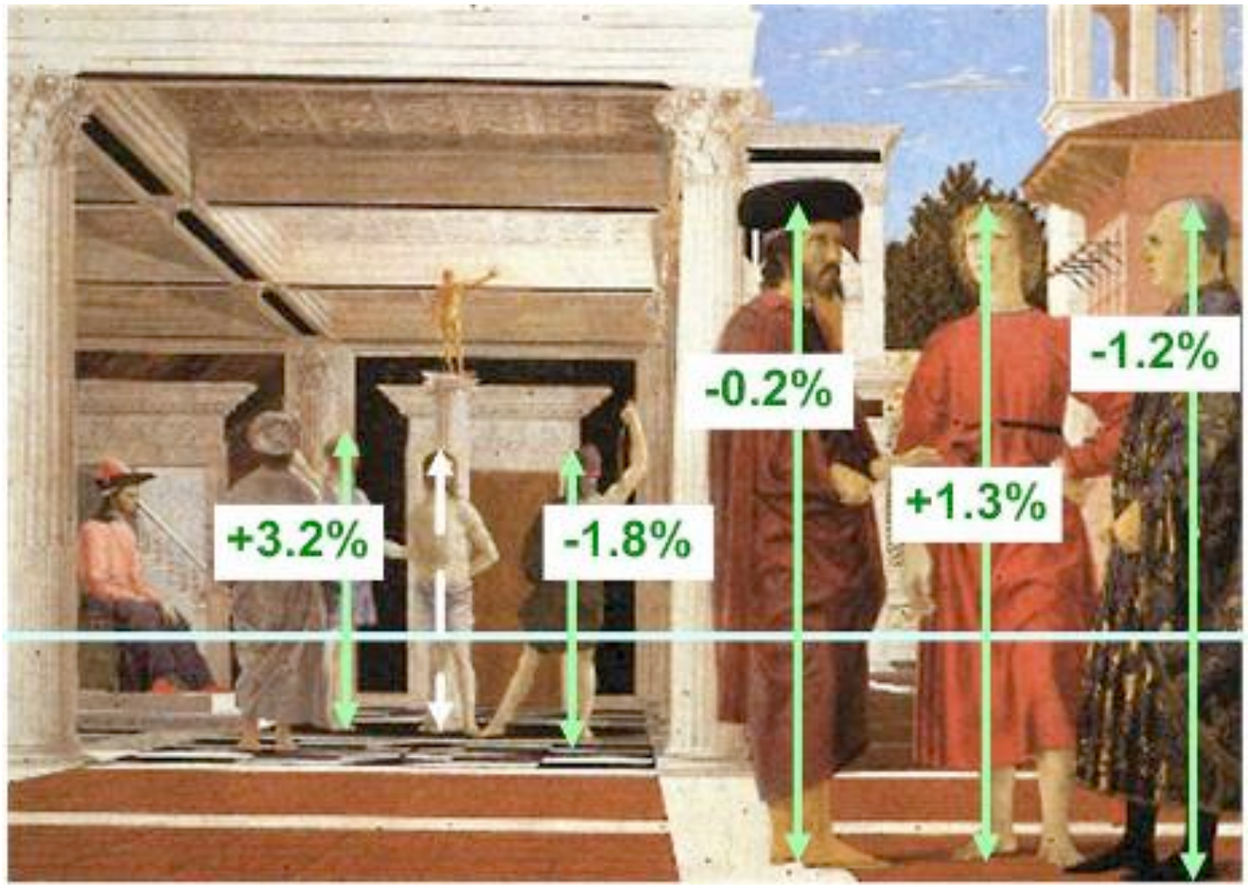
Are the heights of the 2 groups of people consistent with each other?

***Flagellation*,**
Piero della Francesca

Assessing geometric accuracy

Are the heights of the 2 groups of people consistent with each other?

***Flagellation*,**
Piero della Francesca



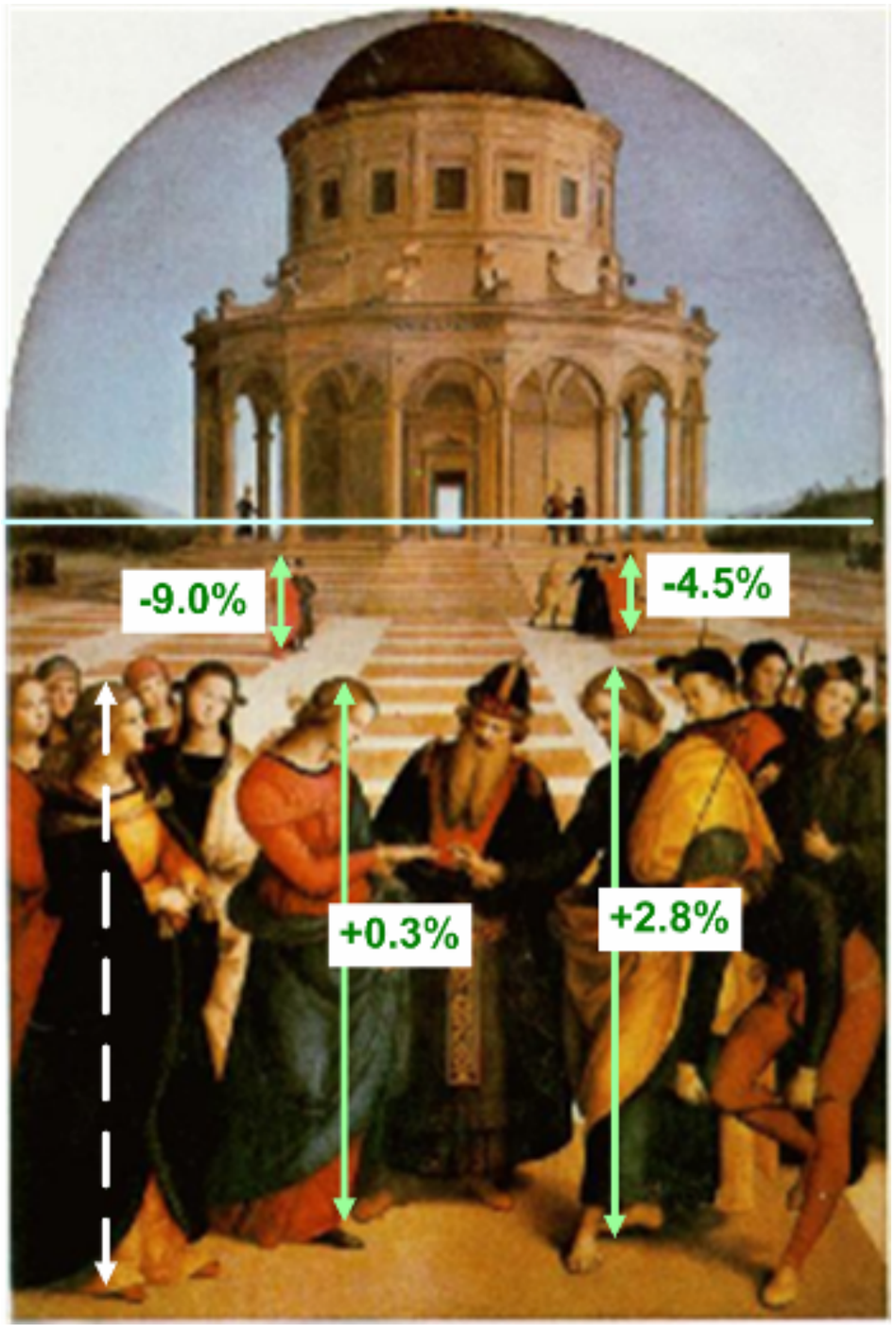
Estimated relative heights

Assessing geometric accuracy

***The Marriage of the Virgin*,**
Raphael

Assessing geometric accuracy

***The Marriage of the Virgin*, Raphael**



Estimated relative heights

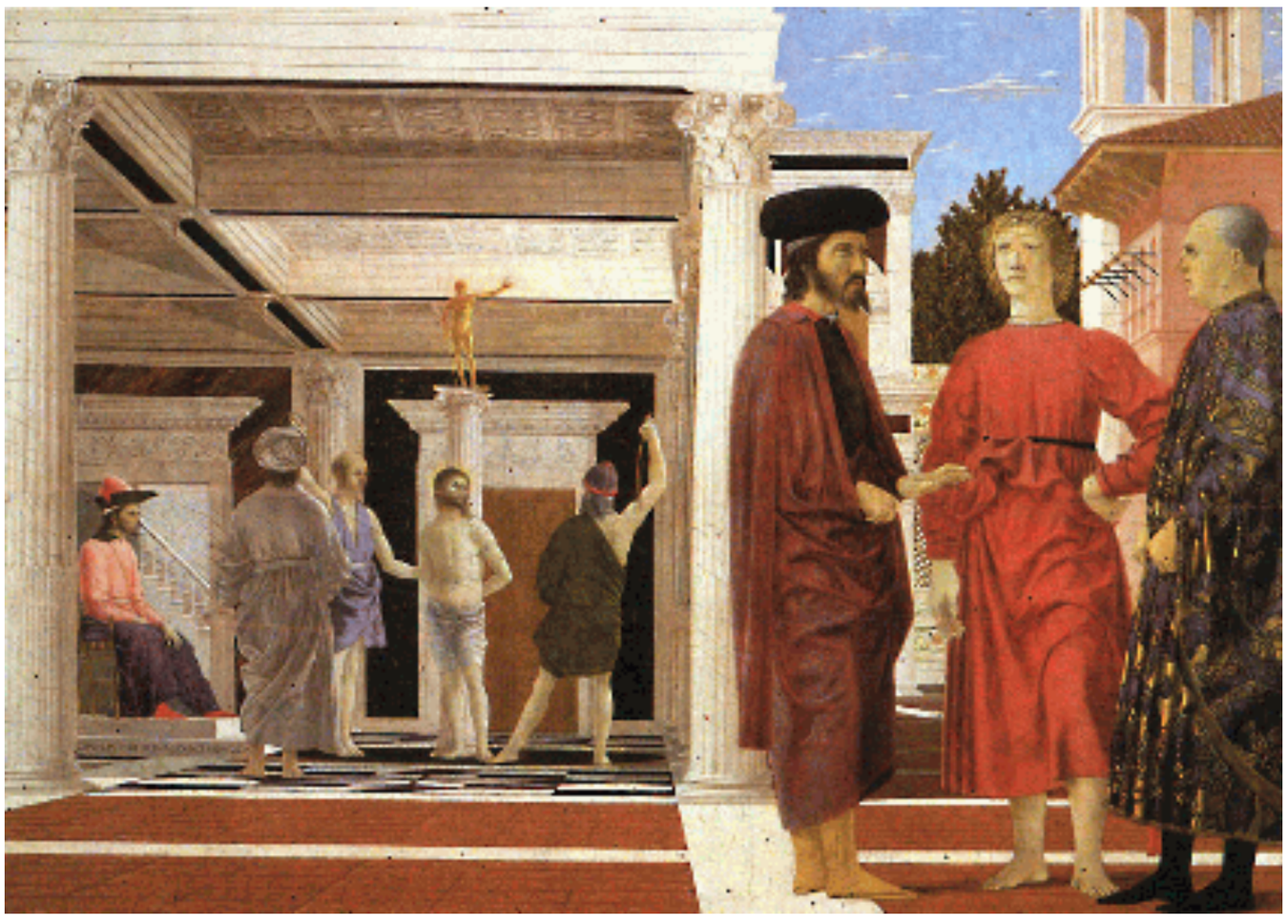
Criminisi et al., ICCV 99

Complete approach

- Load in an image
- Click on lines parallel to X axis
  - repeat for Y, Z axes
- Compute vanishing points
- Specify 3D and 2D positions of 4 points on reference plane
- Compute homography H
- Specify a reference height
- Compute 3D positions of several points
- Create a 3D model from these points
- Extract texture maps
  - Cut out objects
  - Fill in holes
- Output a VRML model

Interactive silhouette cut-out

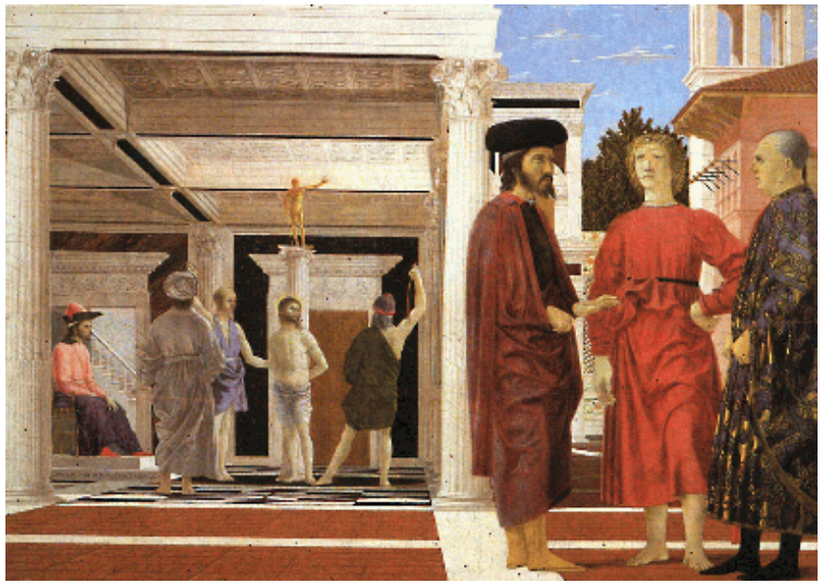
Interactive silhouette cut-out

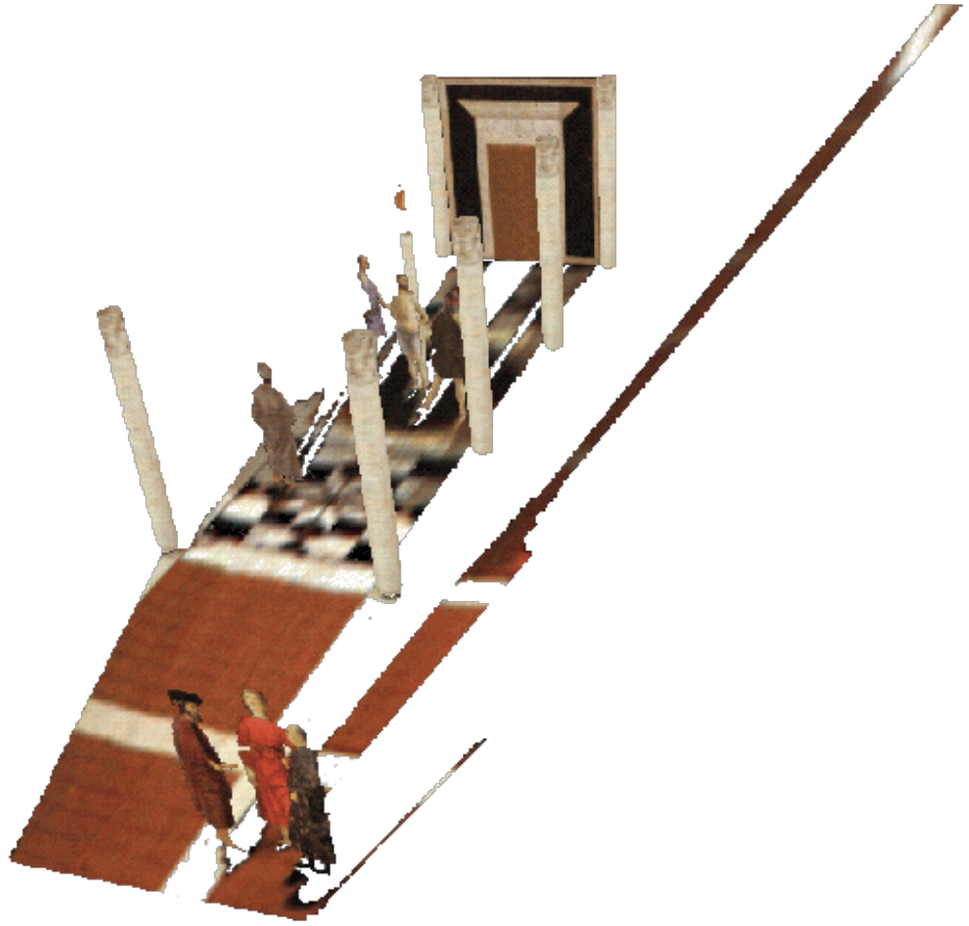


Interactive silhouette cut-out

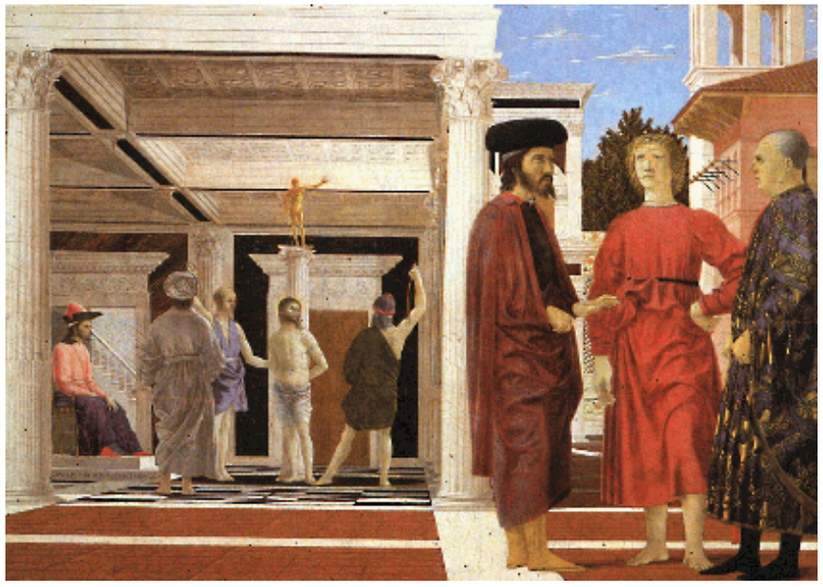
Occlusion filling

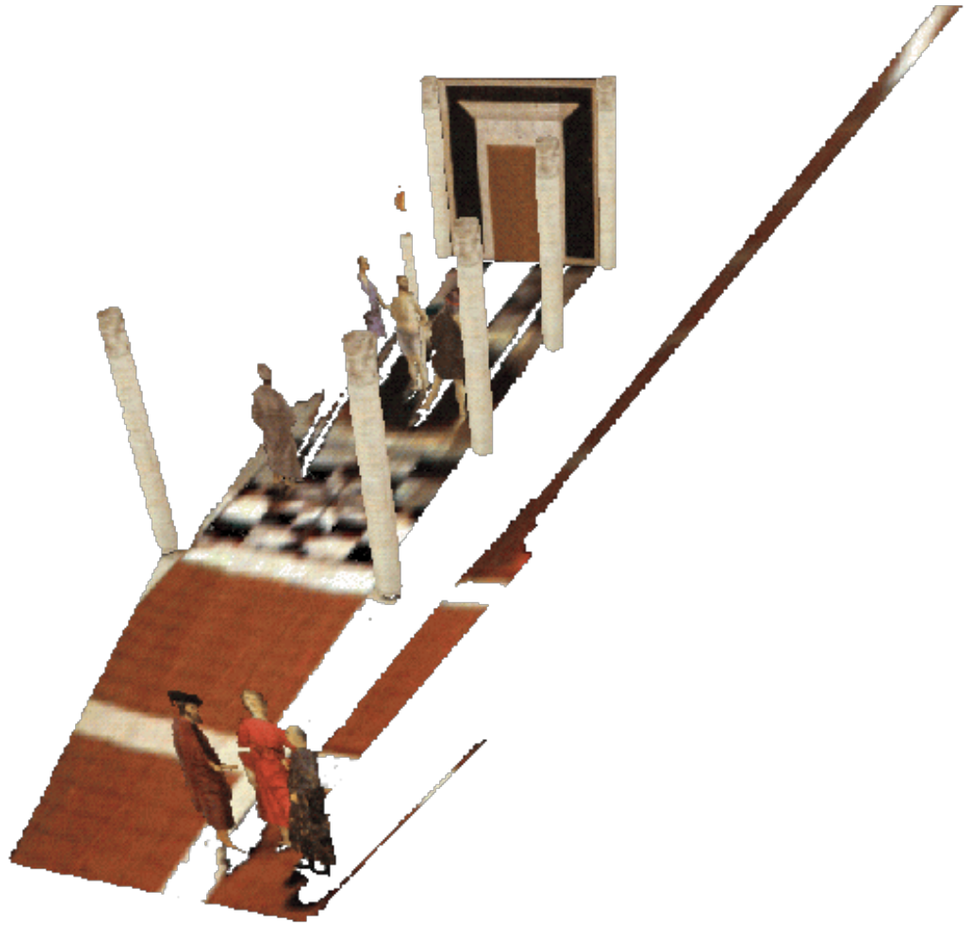
Occlusion filling





Occlusion filling





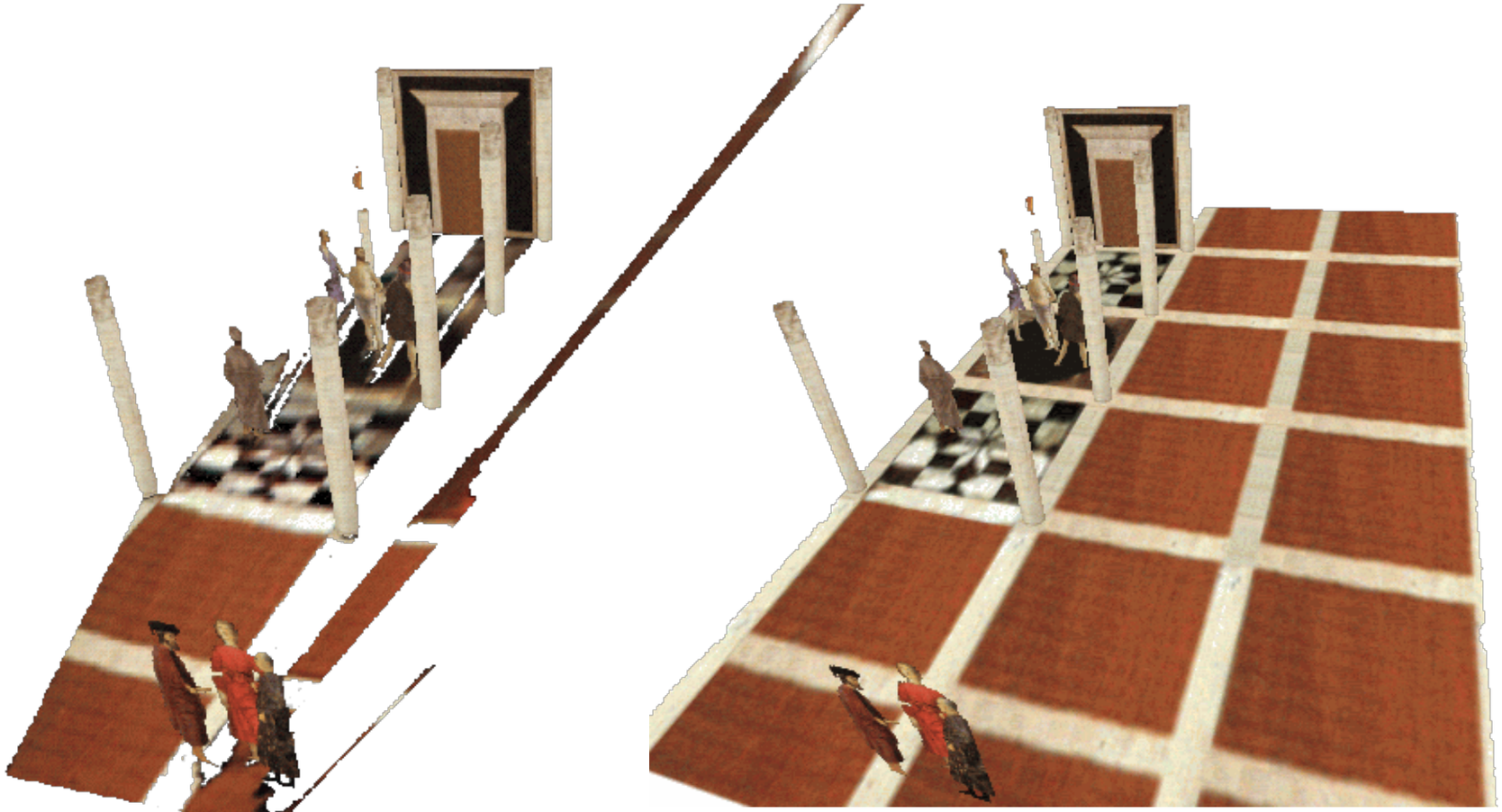
Geometric filling by exploiting:

- symmetries
- repeated regular patterns

Texture synthesis

- repeated stochastic patterns

Occlusion filling



Geometric filling by exploiting:

- symmetries
- repeated regular patterns

Texture synthesis

- repeated stochastic patterns

Complete 3D reconstruction

Complete 3D reconstruction

Single image

Complete 3D reconstruction

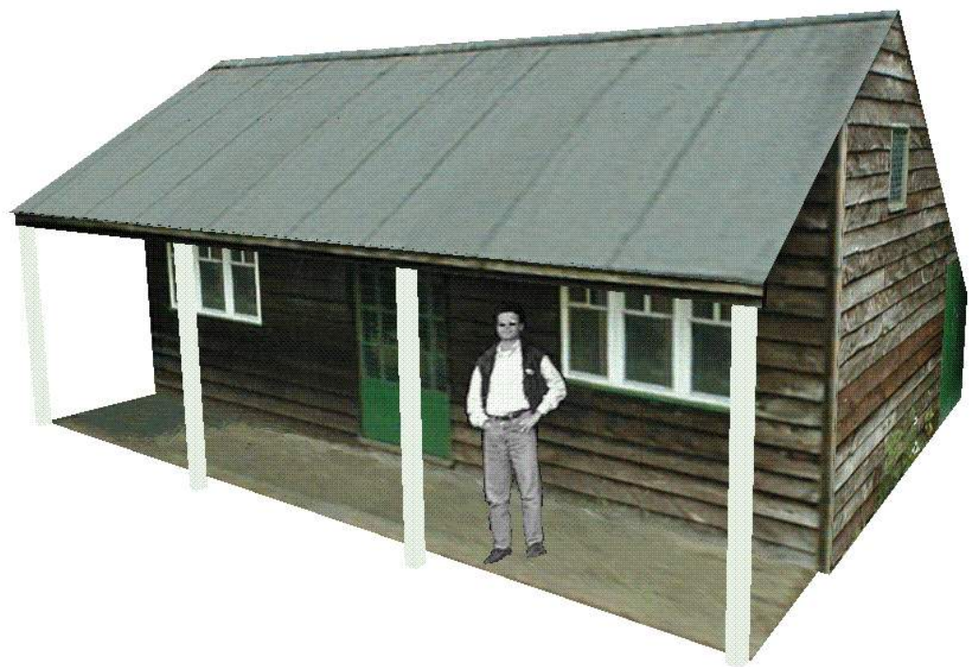
Single View algorithms

Single image

- **Planar measurements**
- **Height measurements**
- **Automatic vanishing point/line computation**
- **Interactive segmentation**
- **Occlusion filling**
- **Object placement in 3D model**

Complete 3D reconstruction

Single View algorithms



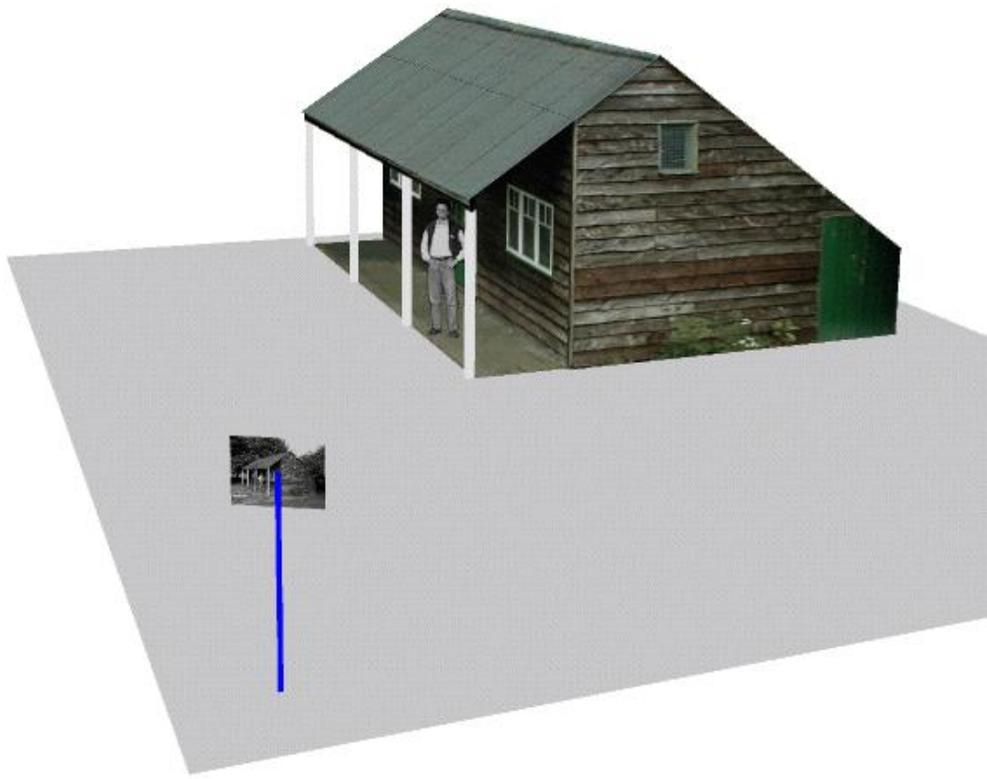
Single image

- **Planar measurements**
- **Height measurements**
- **Automatic vanishing point/line computation**
- **Interactive segmentation**
- **Occlusion filling**
- **Object placement in 3D model**

3D model

Complete 3D reconstruction

Single View algorithms



Single image

- **Planar measurements**
- **Height measurements**
- **Automatic vanishing point/line computation**
- **Interactive segmentation**
- **Occlusion filling**
- **Object placement in 3D model**

3D model


The Virtual Museum
A. Criminisi @ Microsoft, 2002

The camera never lies ?

Of course, your assumptions about reality might be wrong!

The Ames Room

Some Related Techniques

Image-Based Modeling and Photo Editing

- Mok et al., SIGGRAPH 2001
- http://graphics.csail.mit.edu/ibedit/

Single View Modeling of Free-Form Scenes

- Zhang et al., CVPR 2001
- http://grail.cs.washington.edu/projects/svm/

Tour Into The Picture

- Anjyo et al., SIGGRAPH 1997
- http://koigakubo.hitachi.co.jp/little/DL_TipE.html