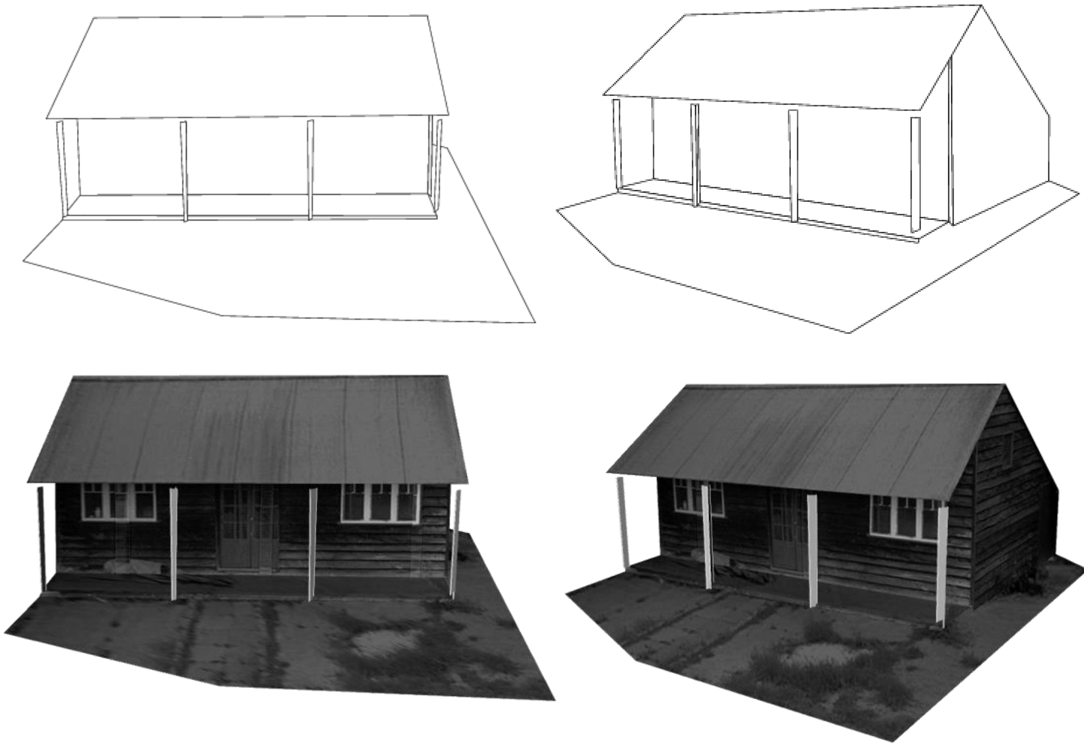
Single-View Geometry

EECS 442 – Prof. David Fouhey

Winter 2019, University of Michigan

http://web.eecs.umich.edu/~fouhey/teaching/EECS442_W19/

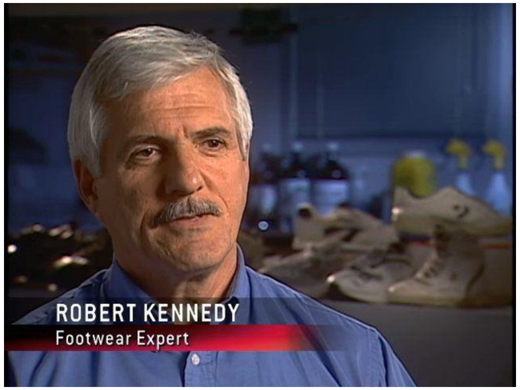
Application: Single-view modeling



A. Criminisi, I. Reid, and A. Zisserman,
Single View Metrology, IJCV 2000

Application: Measuring Height

Application: Measuring Height

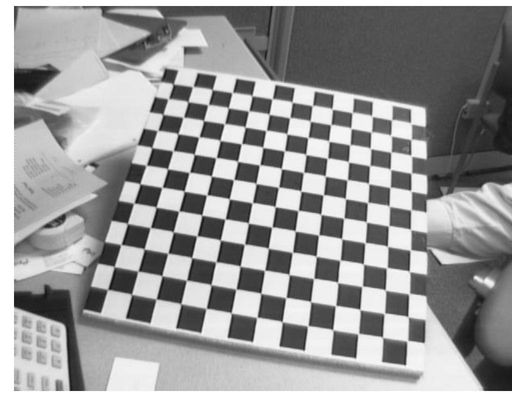
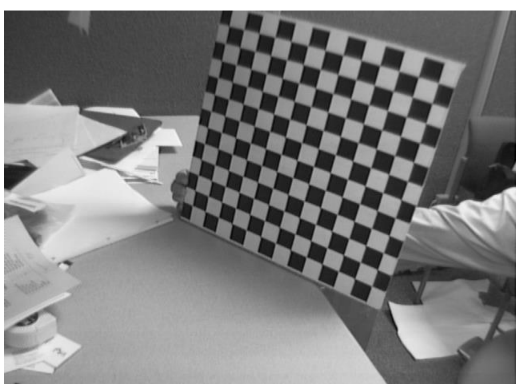


- CSI before CSI
- Covered criminal cases talking to random scientists (e.g., footwear experts)
- How do you tell how tall someone is if they’re not kind enough to stand next to a ruler?

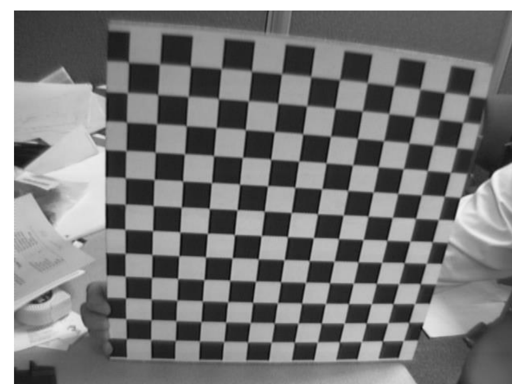
Application: Camera Calibration

- Calibration a HUGE pain



...



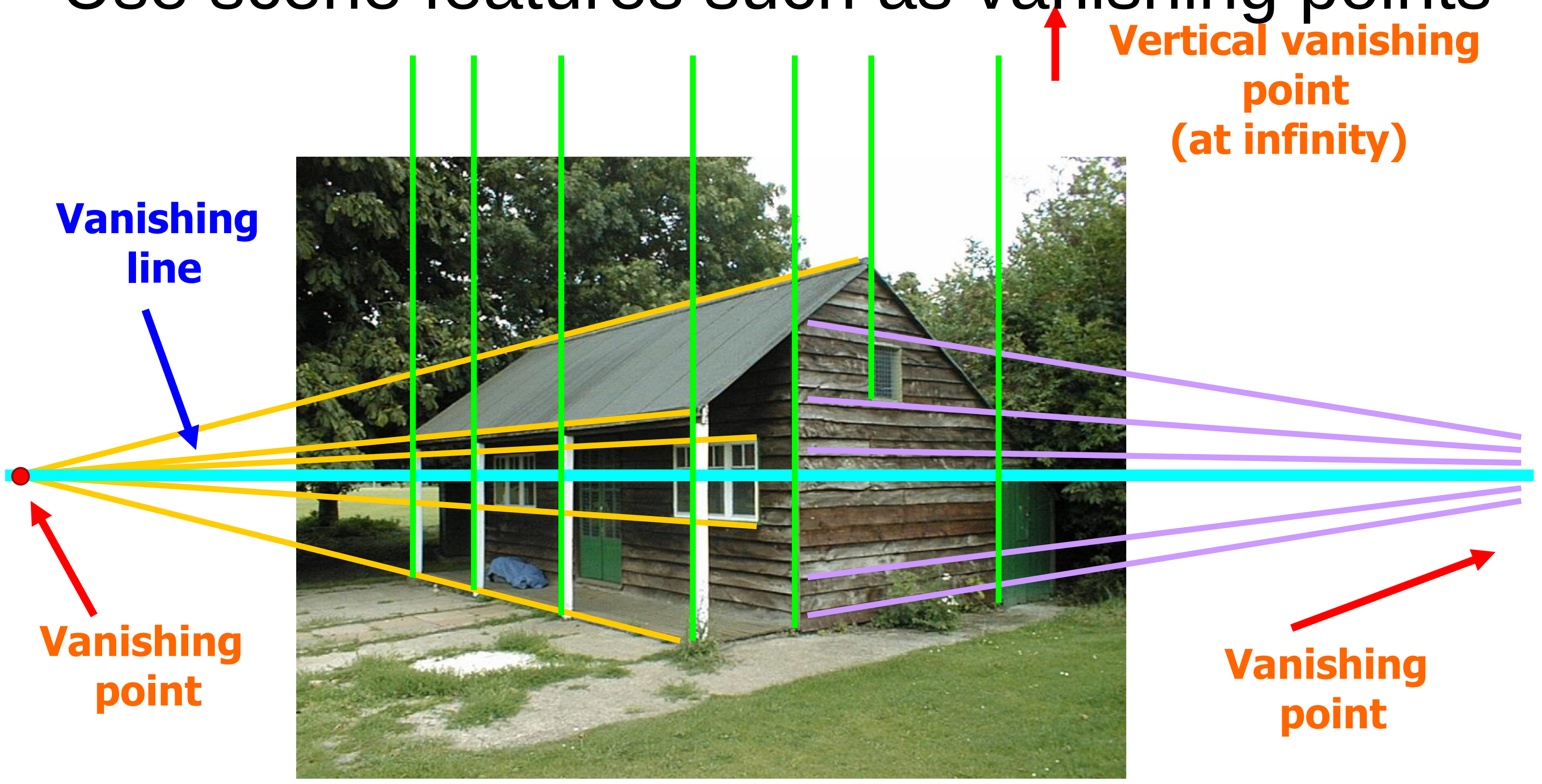
Application: Camera Calibration

- What if 3D coordinates are unknown?
- Use scene features such as vanishing points

Slide from Efros, Photo from Criminisi

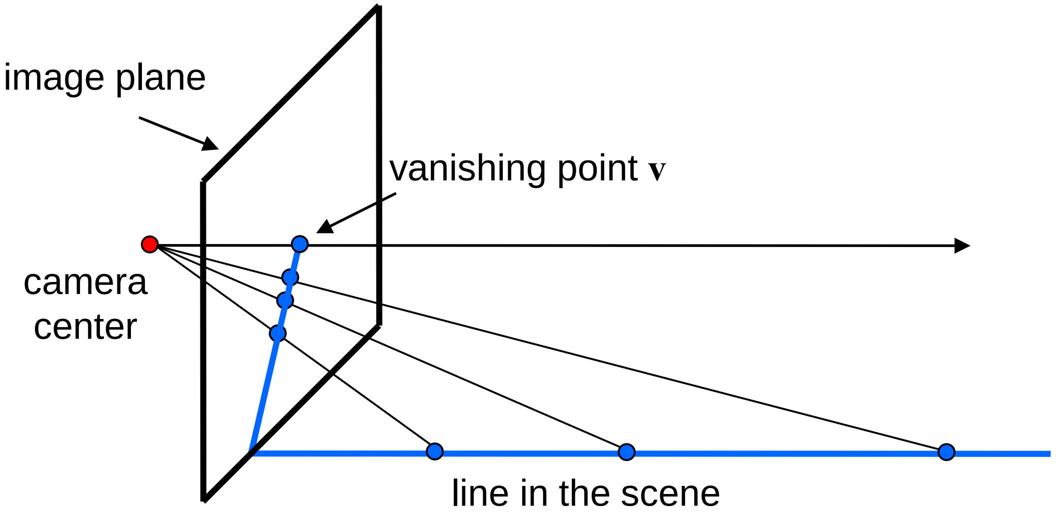
Camera calibration revisited

- What if 3D coordinates are unknown?
- Use scene features such as vanishing points



Slide from Efros, Photo from Criminisi

Recall: Vanishing points



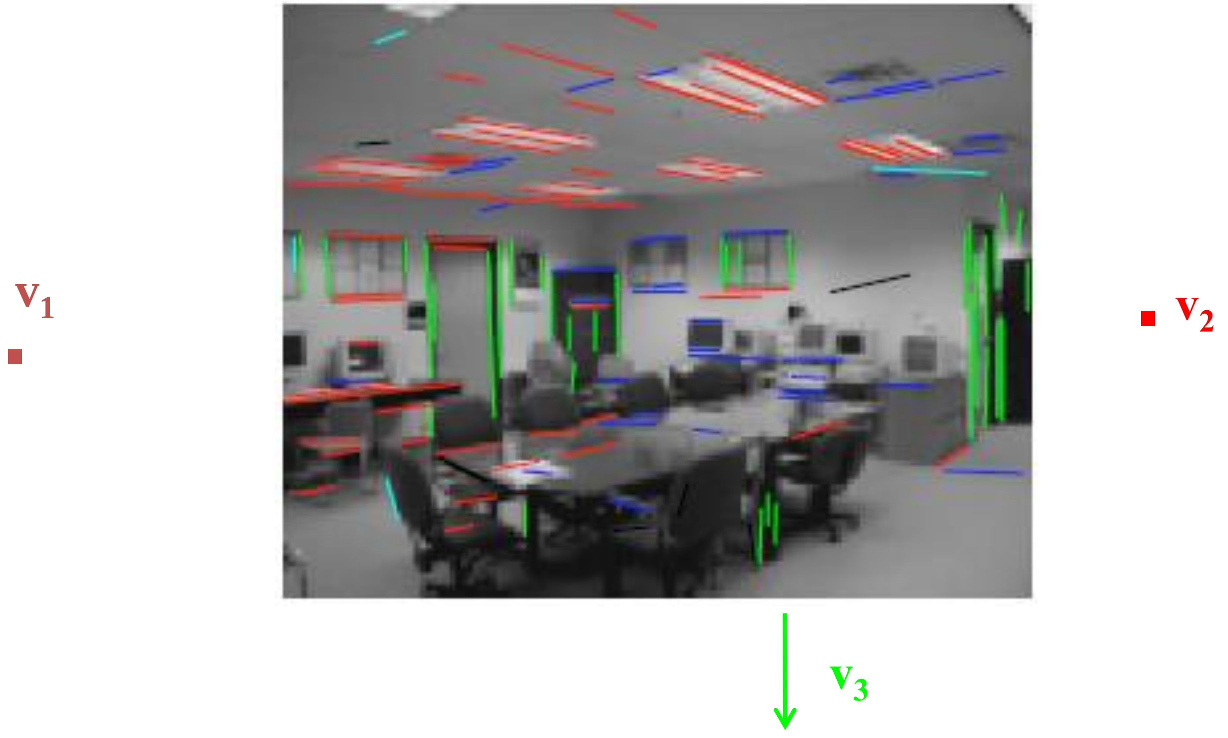
- All lines having the same *direction* share the same vanishing point

Calibration from vanishing points

Consider a scene with 3 orthogonal directions

$\mathbf{v_1}$, $\mathbf{v_2}$ are *finite* vps, $\mathbf{v_3}$ *infinite* vp

Want to align world coordinates with directions



Calibration from vanishing points

$$\boldsymbol{P}_{3x4} \equiv [\, \boldsymbol{p}_1 \ \boldsymbol{p}_2 \ \boldsymbol{p}_3 \ \boldsymbol{p}_4 \,]$$

It turns out that

$\boldsymbol{p}_1 \equiv \boldsymbol{P}\,[1,0,0,0]^T$ VP in X direction

$\boldsymbol{p}_2 \equiv \boldsymbol{P}\,[0,1,0,0]^T$ VP in Y direction

$\boldsymbol{p}_3 \equiv \boldsymbol{P}\,[0,0,1,0]^T$ VP in Z direction

$\boldsymbol{p}_4 \equiv \boldsymbol{P}\,[0,0,0,1]^T$ Projection of origin

Note the usual $\equiv$ (i.e., all of this is up to scale) as well as the 0 for the vps

Calibration from vanishing points

- Let's align the world coordinate system with the three orthogonal vanishing directions:

$$\boldsymbol{e_1} = \begin{bmatrix}1\\0\\0\end{bmatrix} \quad \boldsymbol{e_2} = \begin{bmatrix}0\\1\\0\end{bmatrix} \quad \boldsymbol{e_3} = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$\lambda \boldsymbol{v_i} = \boldsymbol{K}[\boldsymbol{R}, \boldsymbol{t}] \begin{bmatrix}\boldsymbol{e_i}\\0\end{bmatrix}$$

$$\lambda \boldsymbol{v_i} = \boldsymbol{KRe}_i$$ Drop the t

$$\boldsymbol{R^{-1}K^{-1}}\lambda \boldsymbol{v}_i = \boldsymbol{e}_i$$ Inverses

Calibration from vanishing points

So $\boldsymbol{e_i} = \boldsymbol{R^{-1}K^{-1}}\lambda \boldsymbol{v}_i$, but who cares?

What are some properties of axes?

Know $\boldsymbol{e_i^T e_j} = 0$ for $i \neq j$, so K, R have to satisfy

$$\left(\boldsymbol{R^{-1}K^{-1}}\lambda_j \boldsymbol{v_j}\right)^{\boldsymbol{T}}\left(\boldsymbol{R^{-1}K^{-1}}\lambda_i \boldsymbol{v}_i\right) = \boldsymbol{0}$$

$$\left(\boldsymbol{R^T K^{-1}}\lambda_j \boldsymbol{v_j}\right)^{\boldsymbol{T}}\left(\boldsymbol{R^T K^{-1}}\lambda_i \boldsymbol{v}_i\right) = \boldsymbol{0} \qquad R^{-1} = R^T$$

$$\lambda_i \lambda_j \left(\boldsymbol{R^T K^{-1}} \boldsymbol{v_j}\right)^{\boldsymbol{T}}\left(\boldsymbol{R^T K^{-1}} \boldsymbol{v}_i\right) = \boldsymbol{0} \qquad \text{Move scalars}$$

$$\boldsymbol{v_j K^{-T} R R^T K^{-1} v_i} = \boldsymbol{0} \qquad \text{Clean up}$$

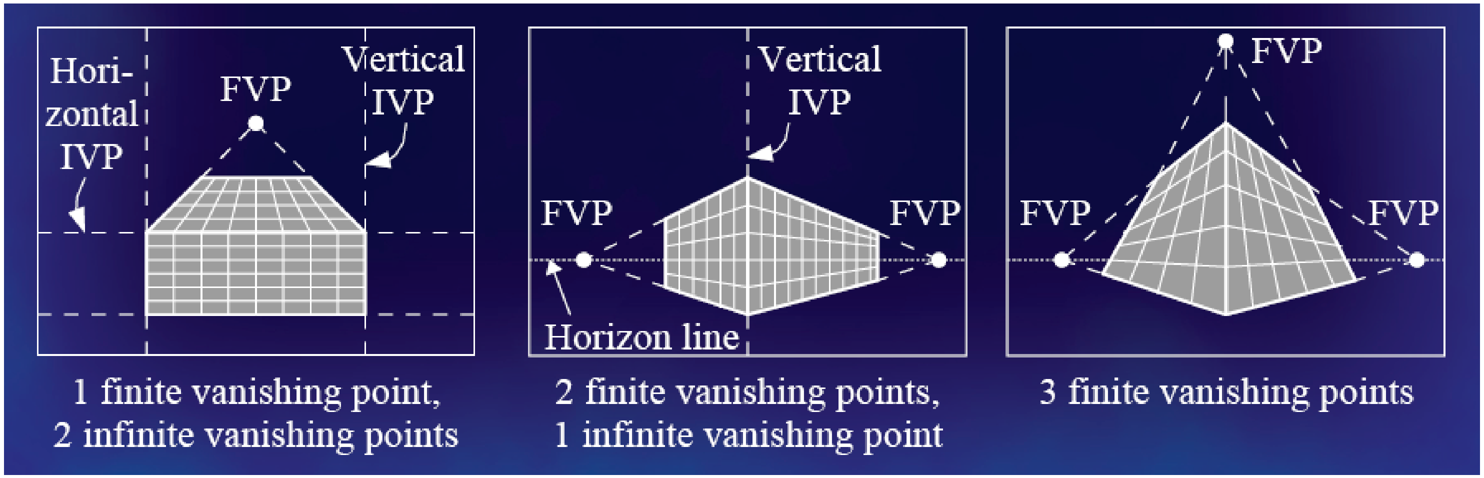
$$\boldsymbol{v_j K^{-T} K^{-1} v_i} = \boldsymbol{0} \qquad RR^T = I$$

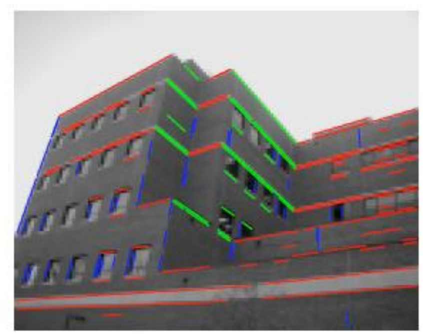
Calibration from vanishing points

- Intrinsics (focal length f, principal point u_0,v_0) have to ensure that the rays corresponding to supposedly orthogonal vanishing points are orthogonal

$$\boldsymbol{v_j K^{-T} K^{-1} v_i = 0}$$

Calibration from vanishing points



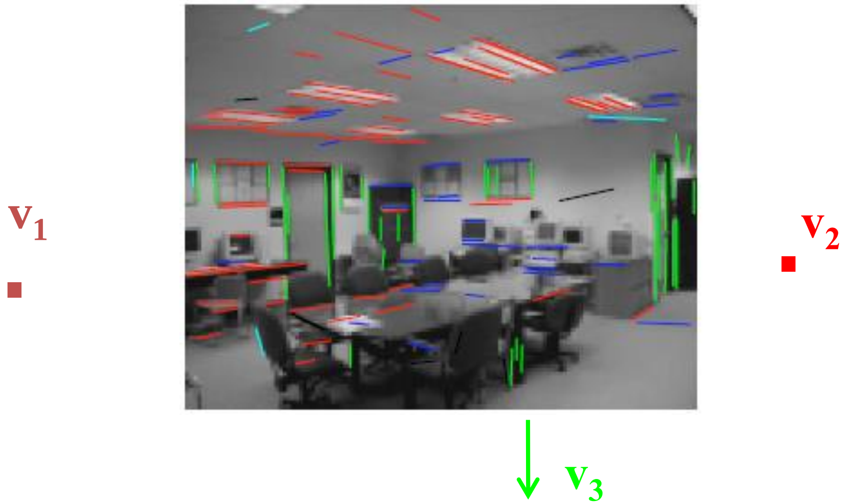


Cannot recover focal length, principal point is the third vanishing point

Can solve for focal length, principal point

Directions and vanishing points

Given vanishing point $\boldsymbol{v}$ camera calibration $\boldsymbol{K}$: $\boldsymbol{K}^{-1}\boldsymbol{v}$ is direction corresponding to that vanishing point.



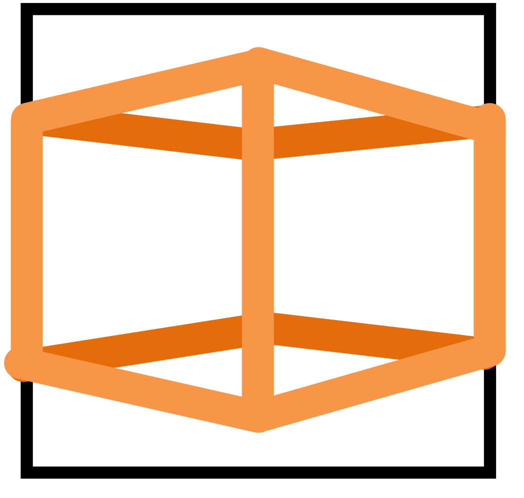
$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \boldsymbol{v_3}$$

$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 10^{10} \\ 1 \end{bmatrix}$$

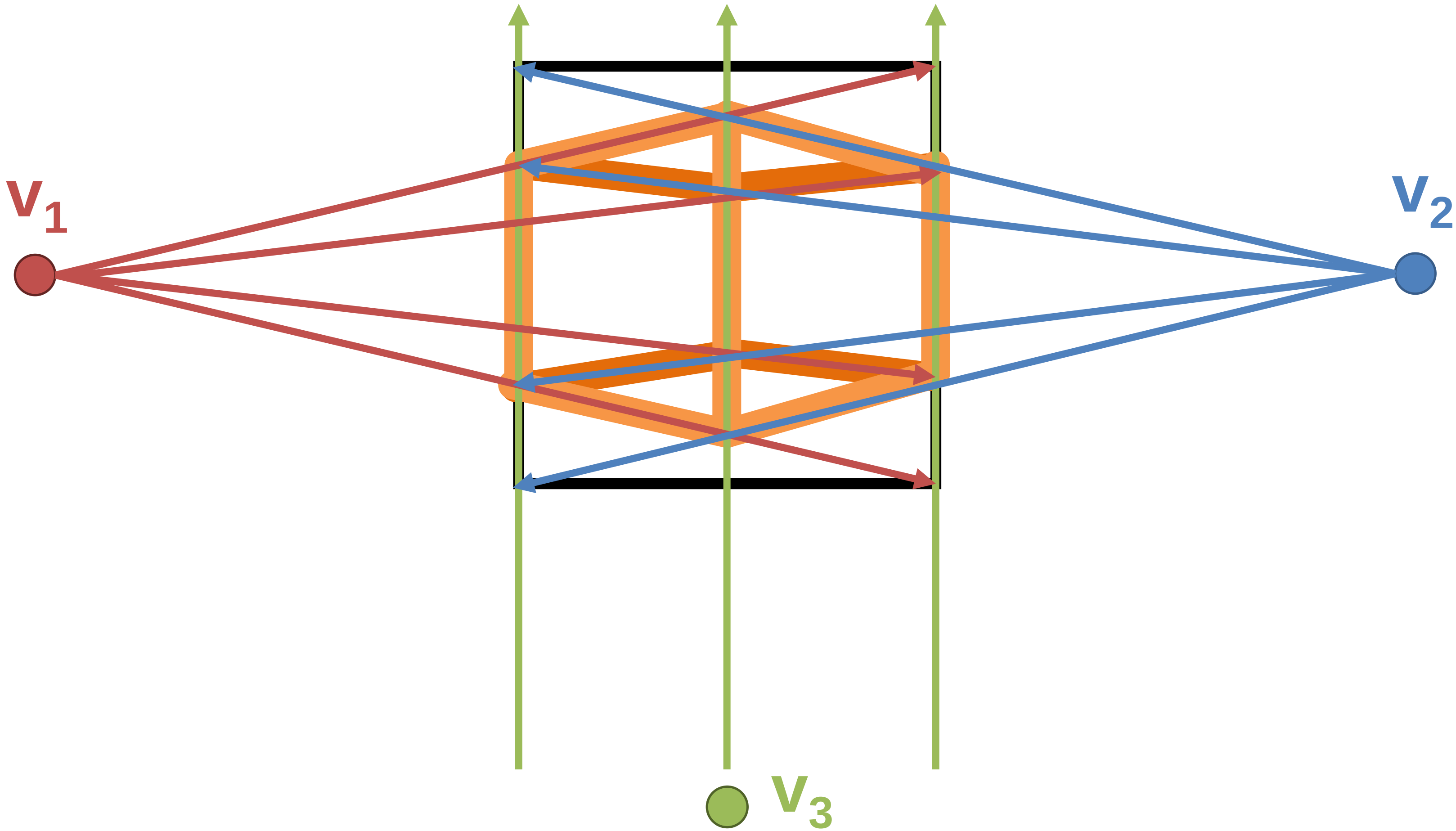
$$\begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 10^{10} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 10^{10}/f \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 10^{6}/f \\ 1 \end{bmatrix} \rightarrow \approx \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Directions and vanishing points

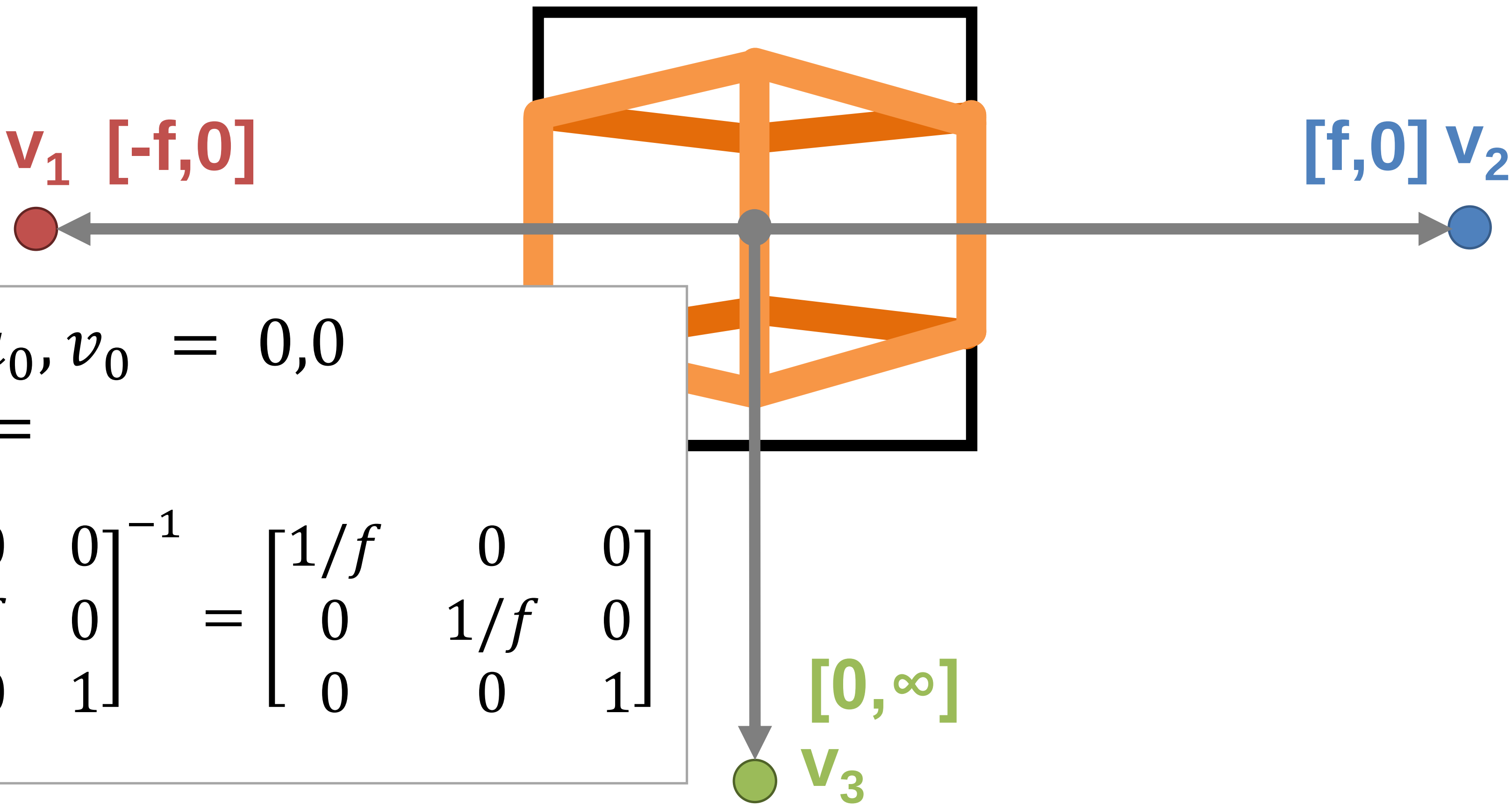


Directions and vanishing points



Directions and vanishing points

If $\boldsymbol{v}$ vanishing point, and $\boldsymbol{K}$ the camera intrinsics, $\boldsymbol{K}^{-1}\boldsymbol{v}$ is the corresponding direction.



Set $u_0, v_0 = 0{,}0$

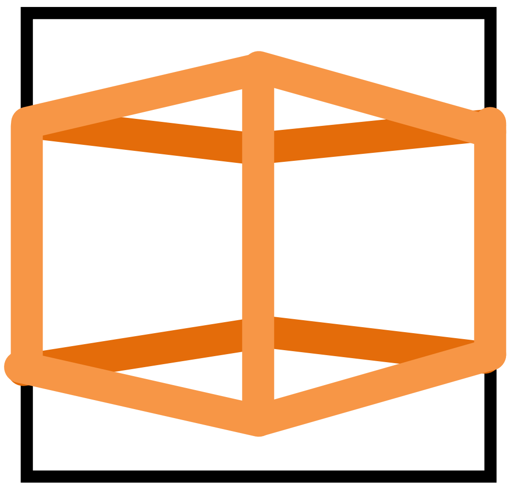
$$K^{-1} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Directions and vanishing points

If $\boldsymbol{v}$ vanishing point, and $\boldsymbol{K}$ the camera intrinsics, $\boldsymbol{K}^{-1}\boldsymbol{v}$ is the corresponding direction.

v_1 [-f,0]

$K^{-1}v_1 = [-1,0,1]$



[f,0] v_2

$K^{-1}v_2 = [1,0,1]$

$$K^{-1} = \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$[0,\infty]$ $K^{-1}v_3 = [0,\infty,1]$

v_3

Rotation from vanishing points

Know that $\lambda_i \boldsymbol{v_i} = \boldsymbol{KRe_i}$ and have **K**, but want **R**

So: $\lambda \boldsymbol{K^{-1}} \boldsymbol{v}_i = \boldsymbol{Re_i}$

What does $\boldsymbol{Re_i}$ look like?

$$\boldsymbol{Re_1} = [\boldsymbol{r_1} \quad \boldsymbol{r_2} \quad \boldsymbol{r_3}] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \boldsymbol{r_1}$$

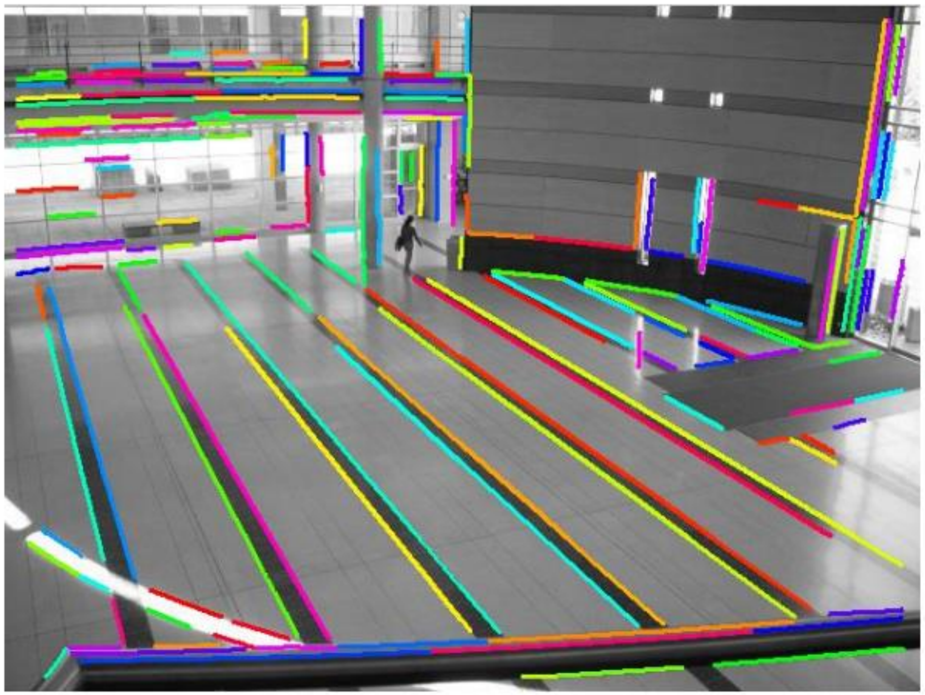
The ith column of R is a scaled version of

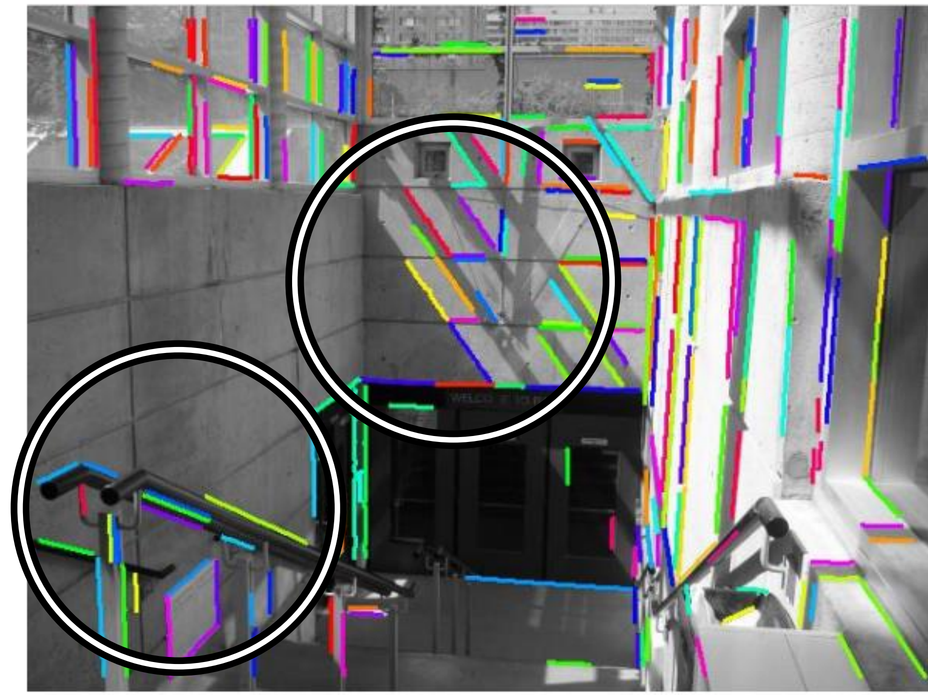
$\boldsymbol{r_i} = \lambda \boldsymbol{K^{-1}} \boldsymbol{v_i}$

Calibration from vanishing points

- Solve for K (focal length, principal point) using 3 orthogonal vanishing points
- Get rotation directly from vanishing points once calibration matrix known
- Pros:
  - Could be totally automatic!
- Cons:
  - Need 3 vanishing points, estimated accurately, but with at least two finite!

Finding Vanishing Points



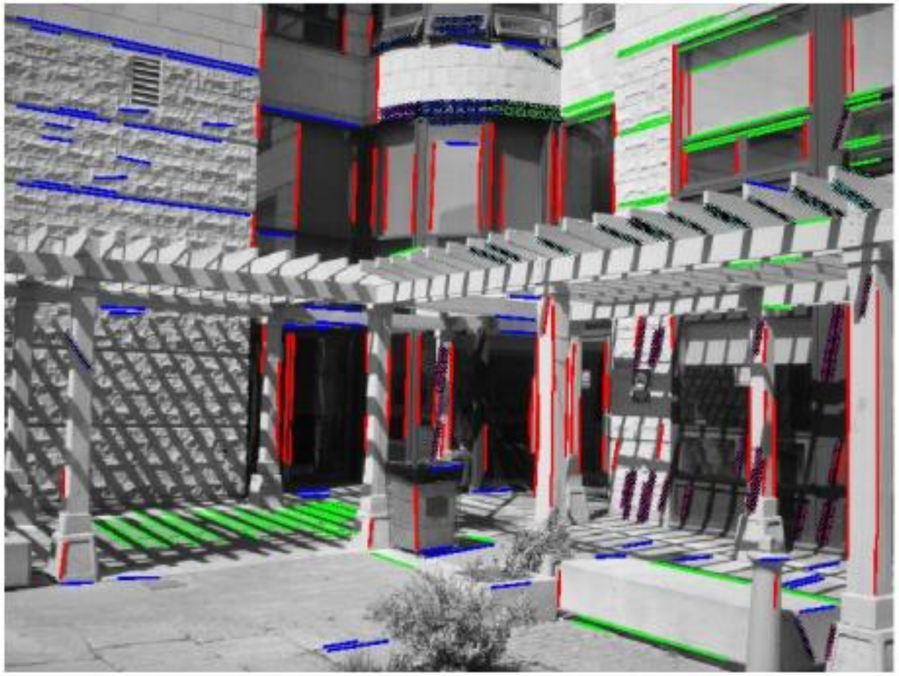


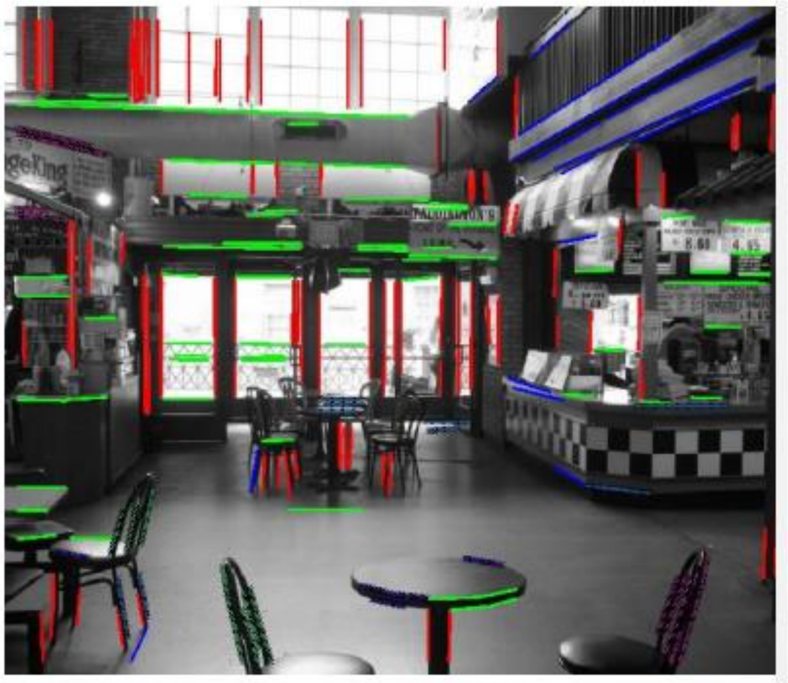
What might go wrong with the circled points?

Finding Vanishing Points

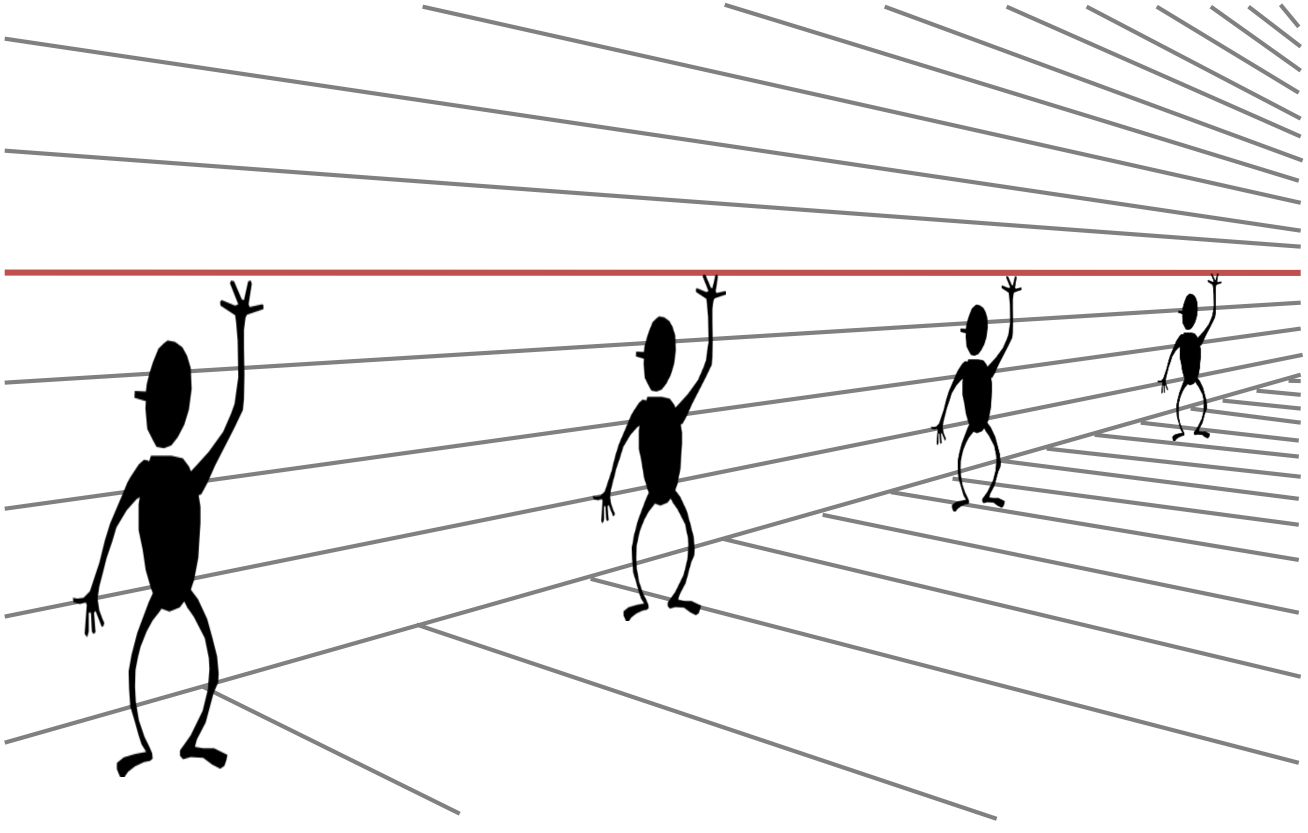
- Find edges $E = \{e_1, \dots, e_n\}$
- All $\binom{n}{2}$ intersections of edges $v_{ij} = e_i \times e_j$ are potential vanishing points
- Try all triplets of popular vanishing points, check if the camera's focal length, principal point "make sense"
- **What are some options for this?**

Finding Vanishing Points



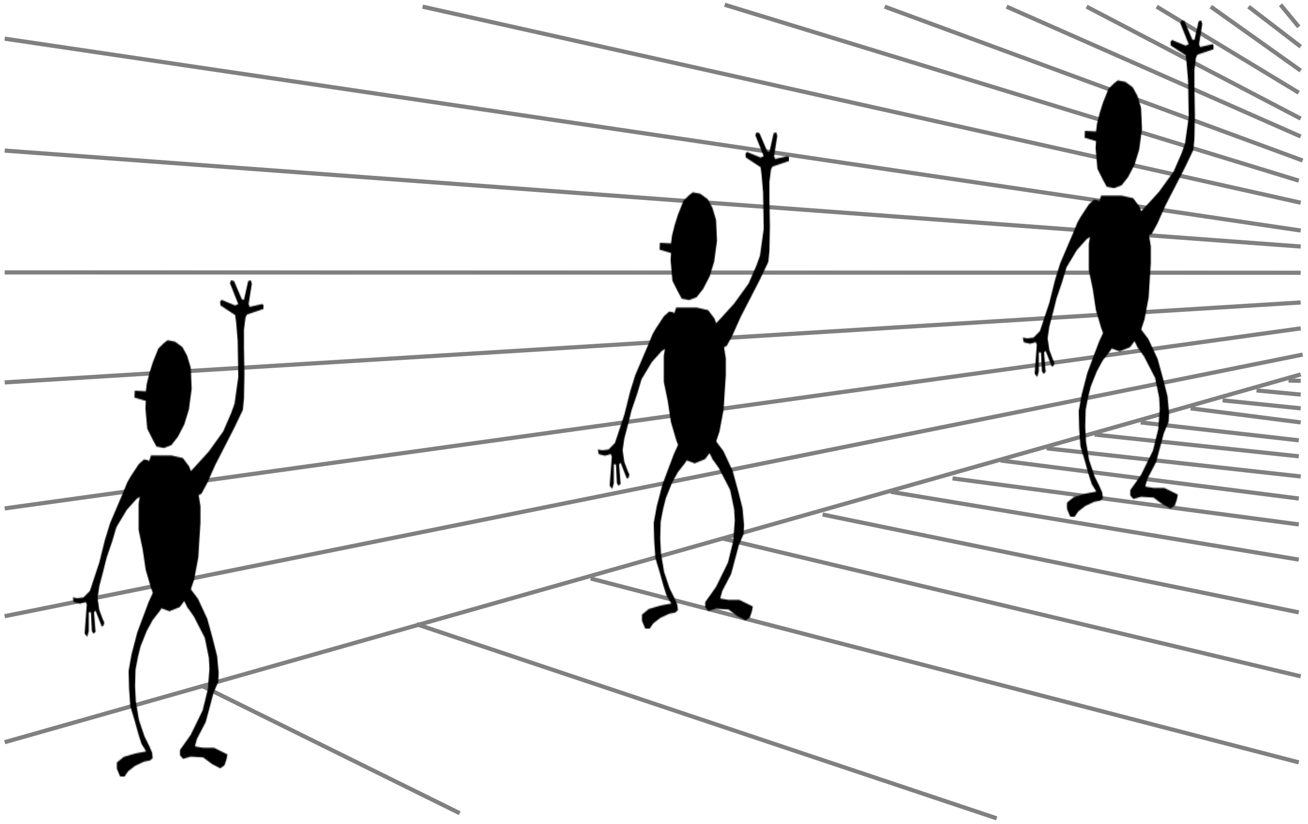


Measuring height



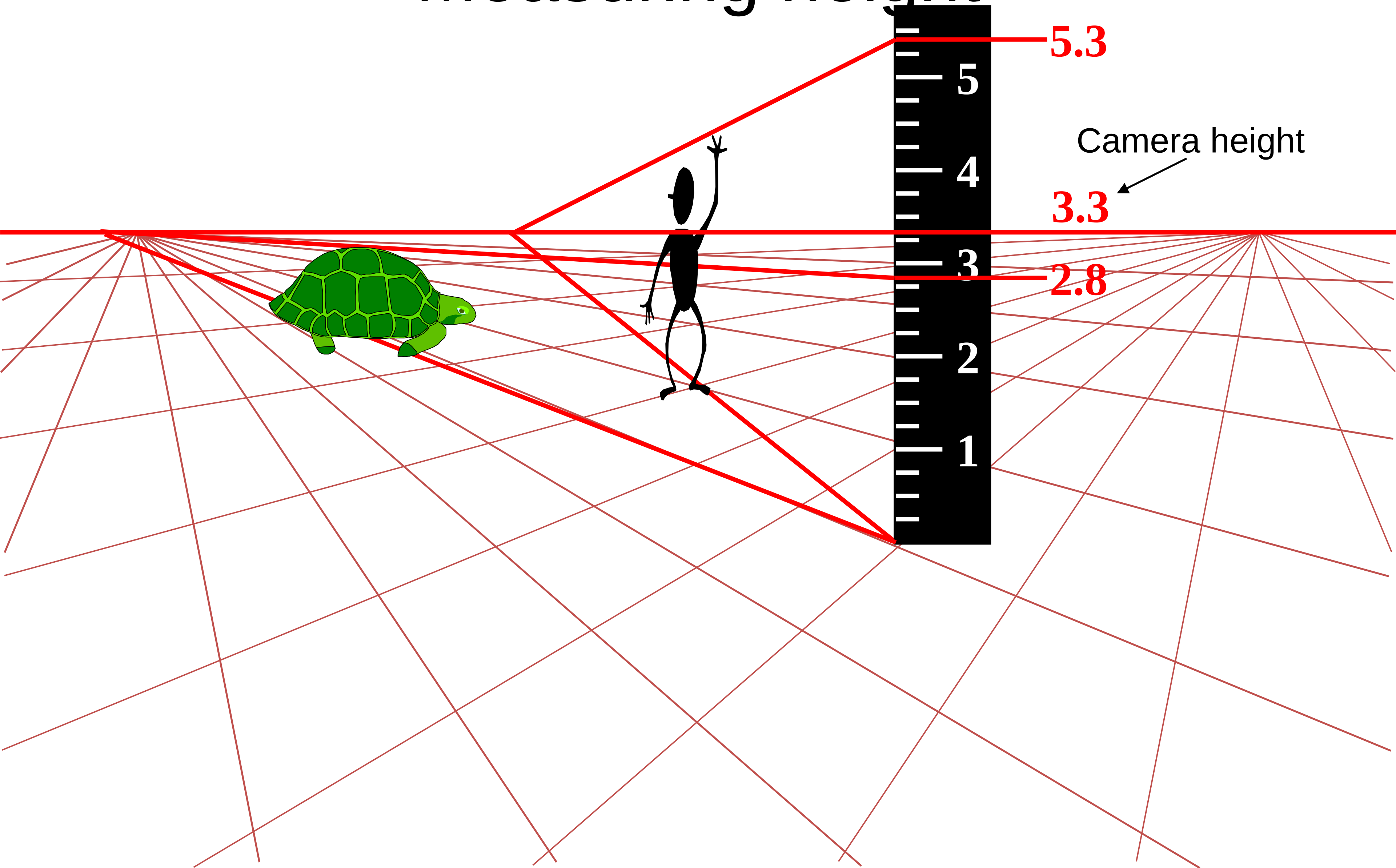
Slide by Steve Seitz

Measuring height

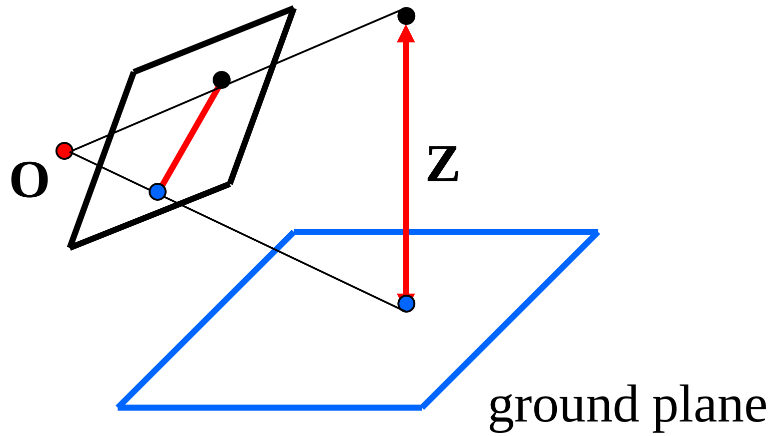


Slide by Steve Seitz

Measuring height



Measuring height without a ruler



Compute Z from image measurements

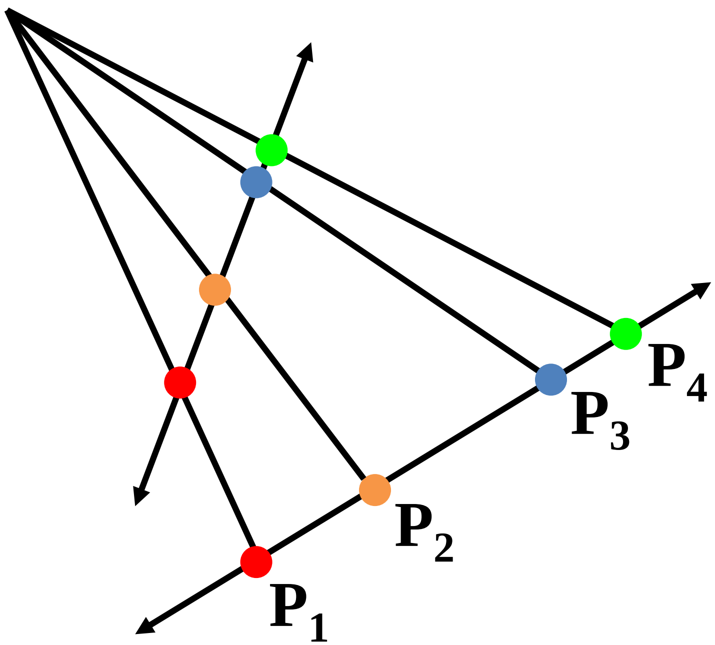
- Need more than vanishing points to do this

Projective invariant

- We need to use a *projective invariant*: a quantity that does not change under projective transformations (including perspective projection)

Projective invariant

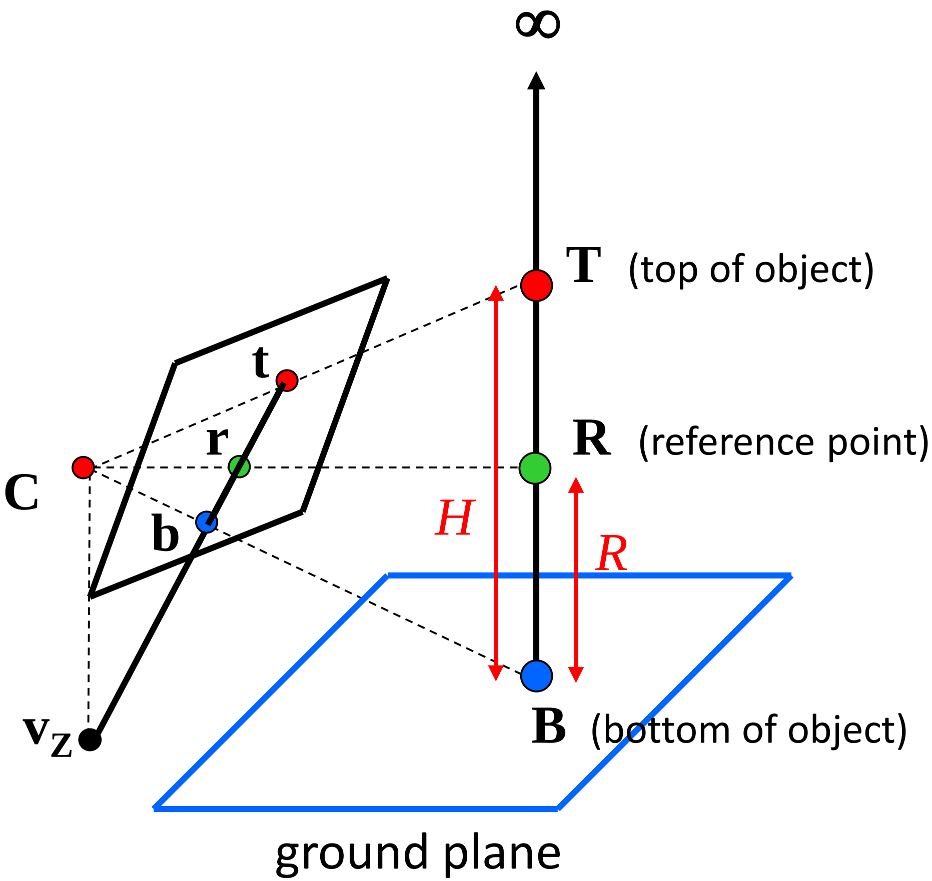
- We need to use a *projective invariant*: a quantity that does not change under projective transformations (including perspective projection)
- The cross-ratio of four points:



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

This is one of the cross-ratios (can reorder arbitrarily)

Measuring height



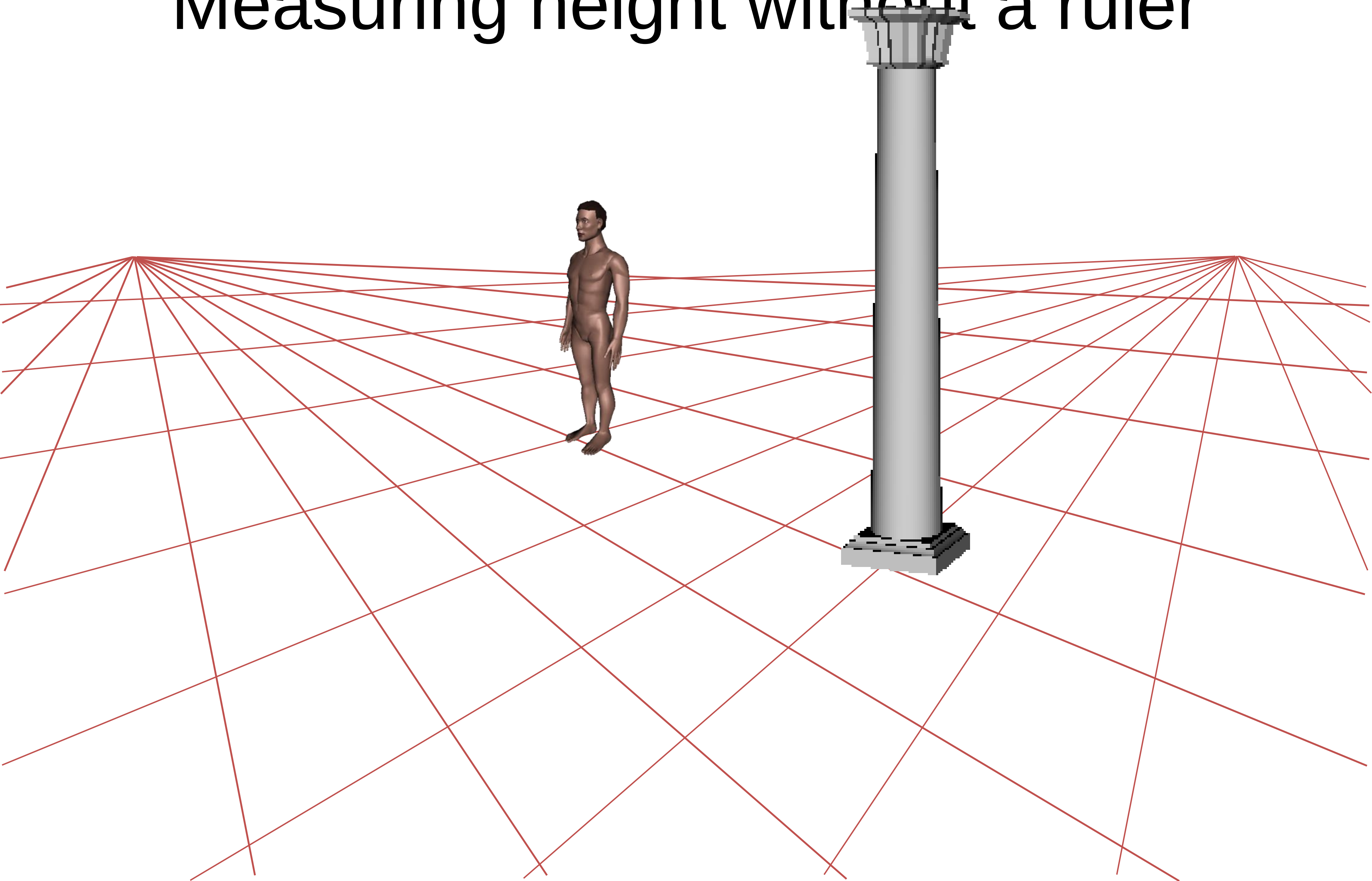
$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|} = \frac{H}{R}$$

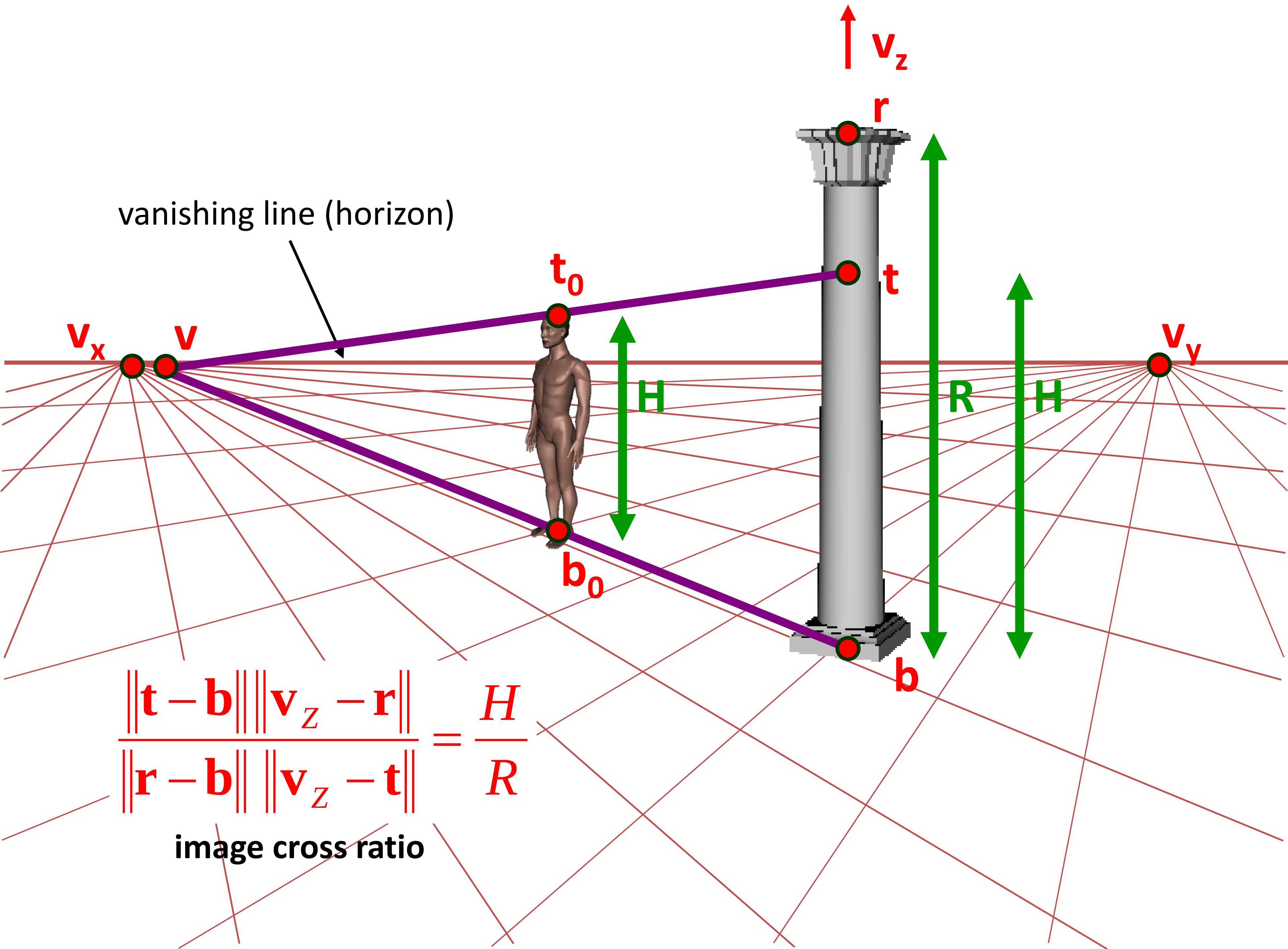
scene cross ratio

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

image cross ratio

Measuring height without a ruler



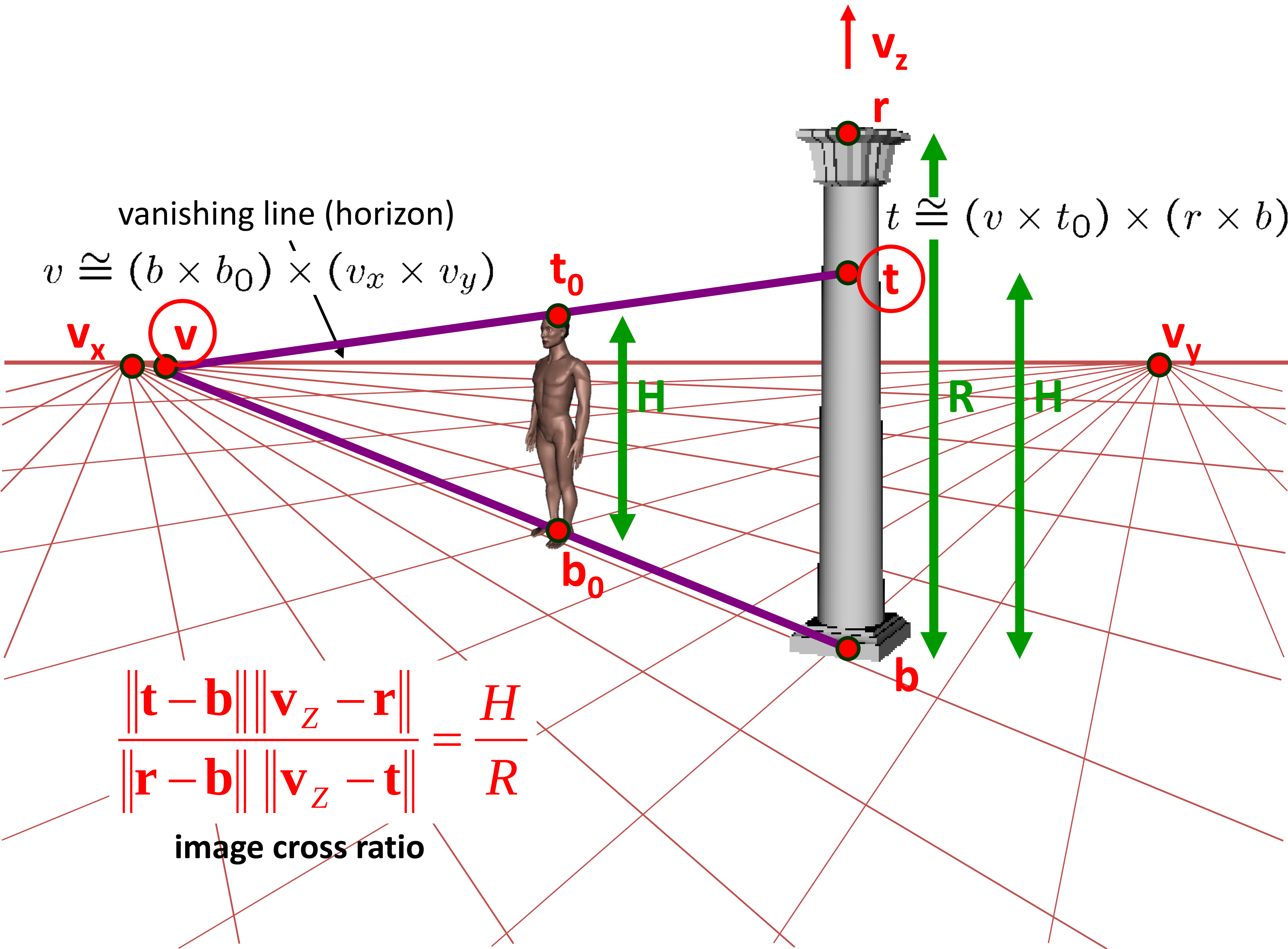


$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$

image cross ratio

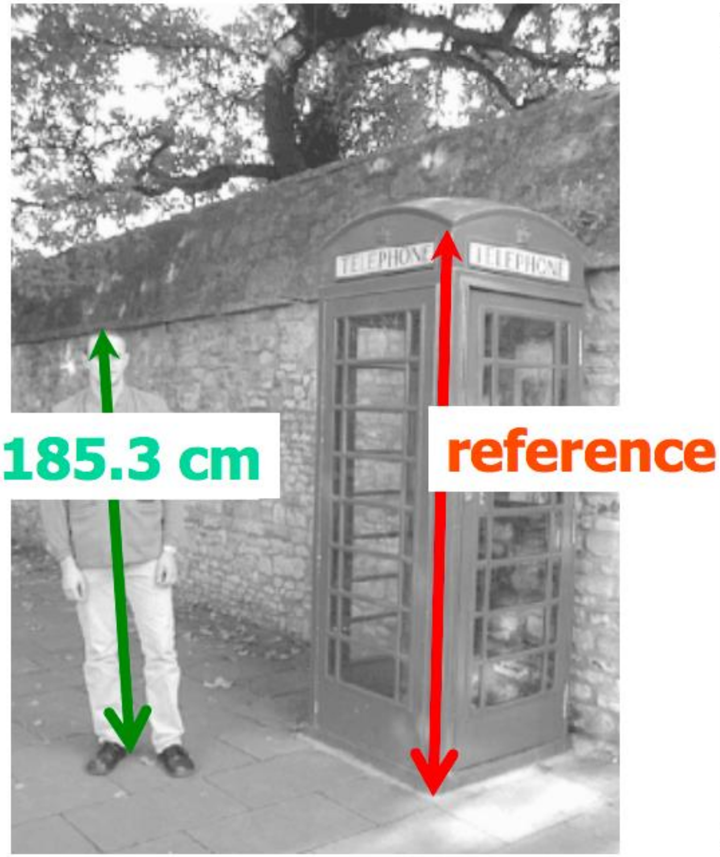
Remember This?

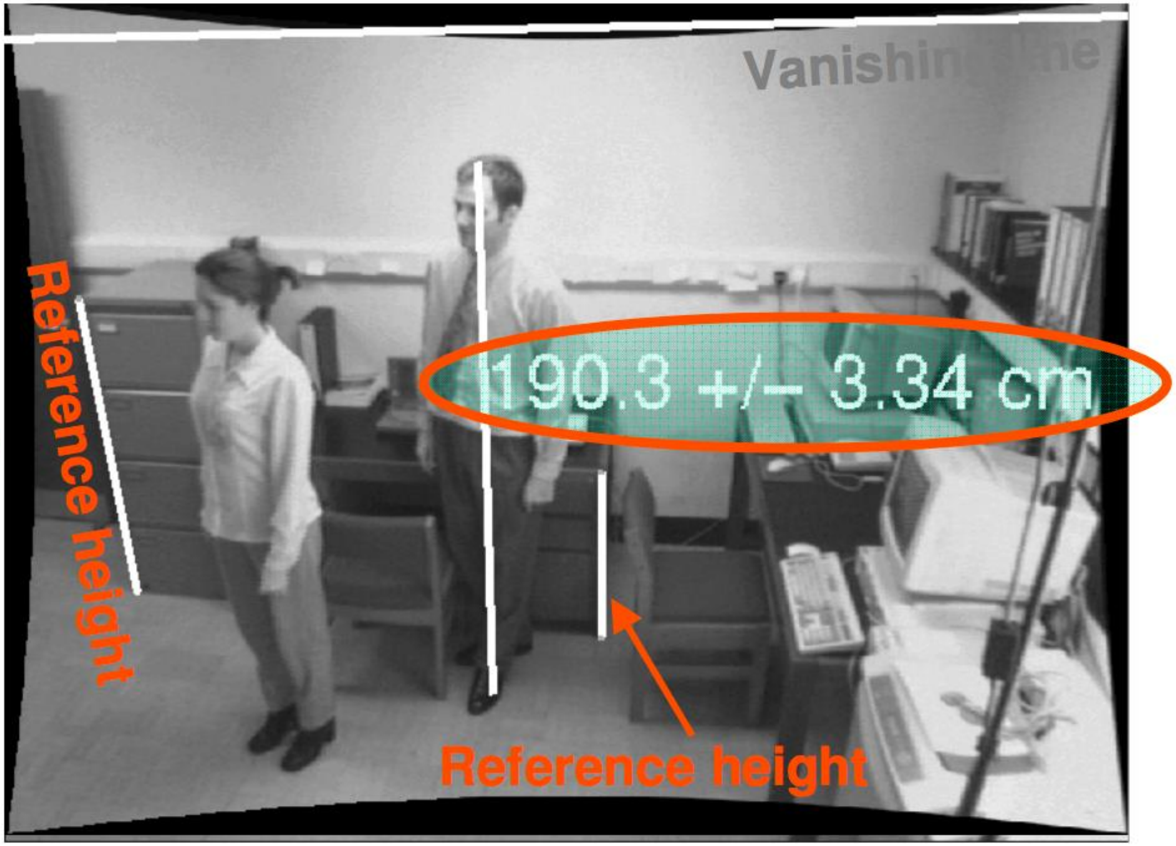
- Line equation: $ax + by + c = 0$
- Vector form: $\boldsymbol{l}^T \boldsymbol{p} = 0$, $\boldsymbol{l} = [a, b, c]$, $\mathbf{p} = [x, y, 1]$
- Line through two points?
  - $\boldsymbol{l} = \boldsymbol{p_1} \times \boldsymbol{p_2}$
- Intersection of two lines?
  - $\boldsymbol{p} = \boldsymbol{l_1} \times \boldsymbol{l_2}$
- Intersection of two parallel lines is at infinity



$\mathbf{v_z}$
r
vanishing line (horizon)
$t \cong (v \times t_0) \times (r \times b)$
$v \cong (b \times b_0) \times (v_x \times v_y)$
$\mathbf{t_0}$
t
$\mathbf{v_x}$
v
$\mathbf{v_y}$
H
R
H
$\mathbf{b_0}$
b
$$\frac{\|\mathbf{t}-\mathbf{b}\|\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$
image cross ratio

Examples





A. Criminisi, I. Reid, and A. Zisserman, [Single View Metrology](), IJCV 2000
Figure from [UPenn CIS580 slides]()

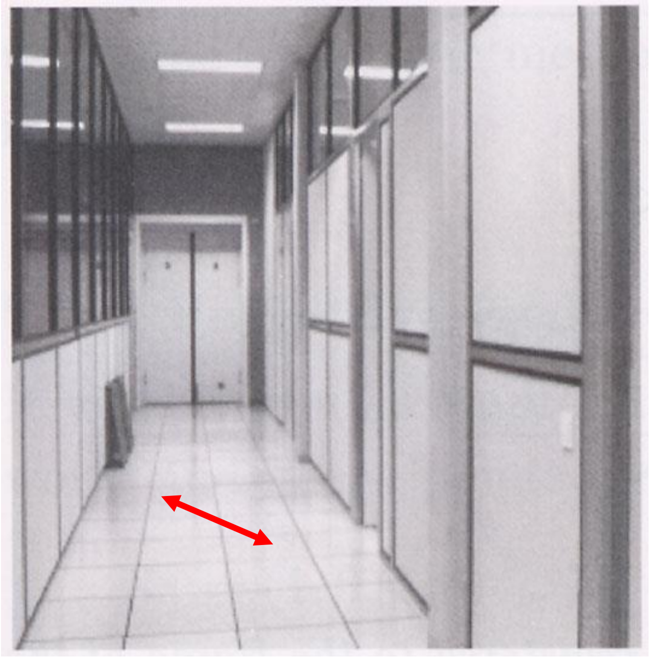
Another example

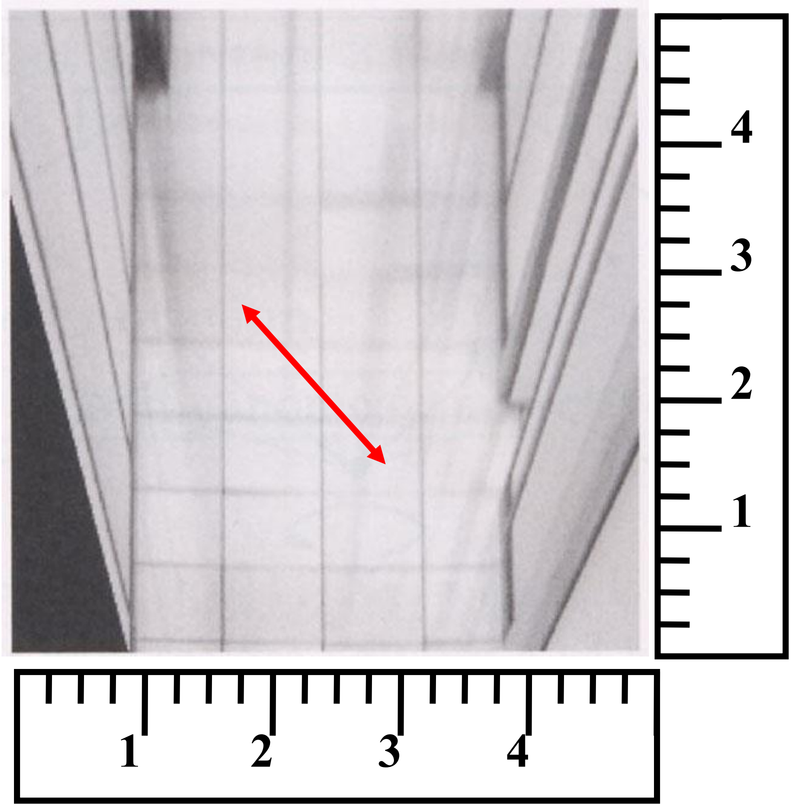
- Are the heights of the two groups of people consistent with one another?

Piero della Francesca*, Flagellation,* ca. 1455

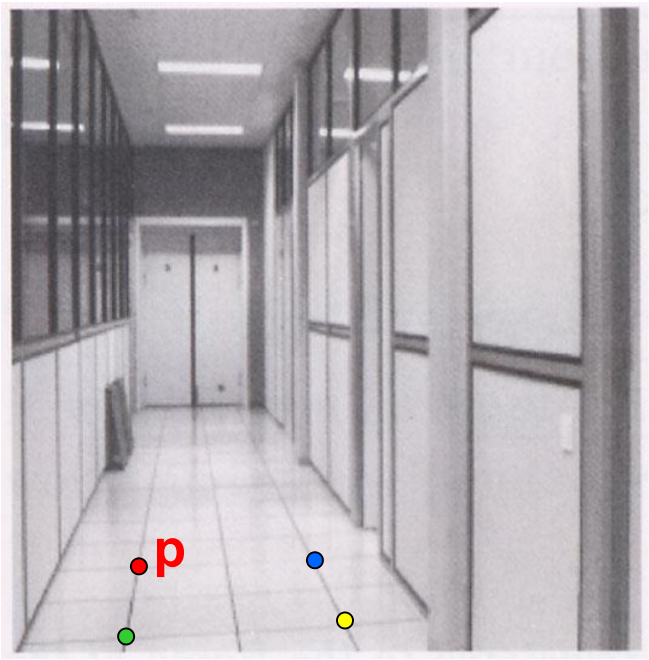
A. Criminisi, M. Kemp, and A. Zisserman, Bringing Pictorial Space to Life: computer techniques for the analysis of paintings,
Proc. Computers and the History of Art, 2002

Measurements on planes





Measurements on planes



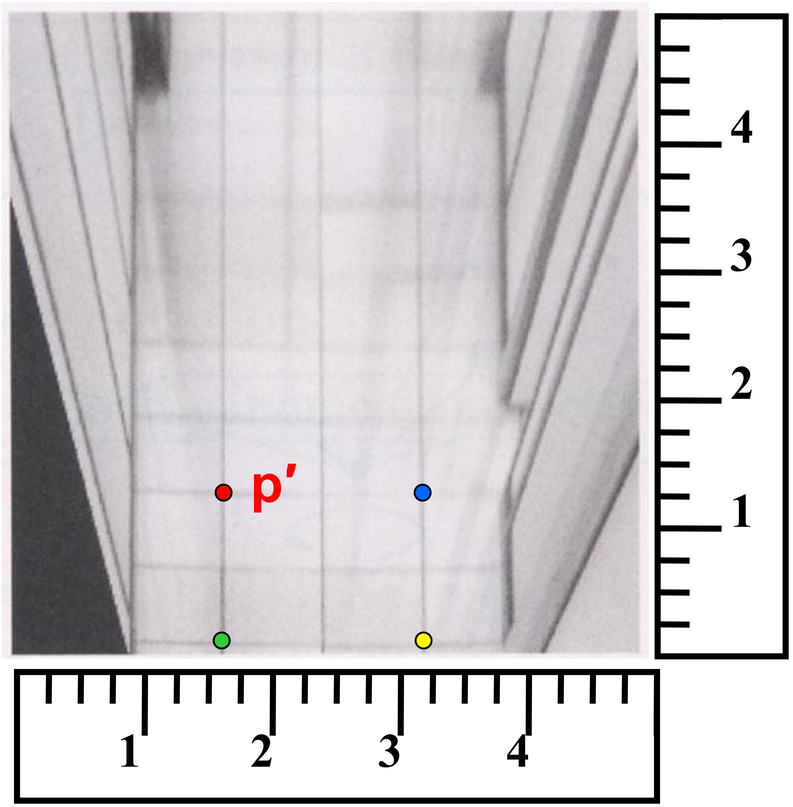
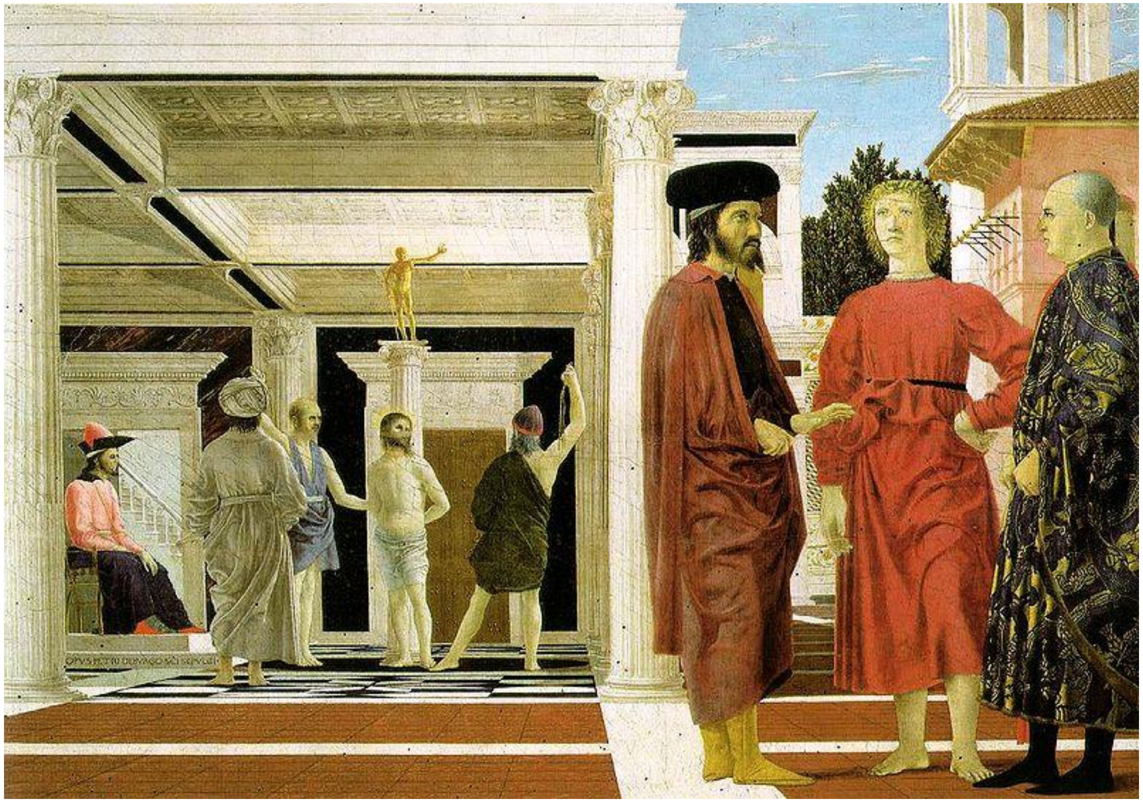
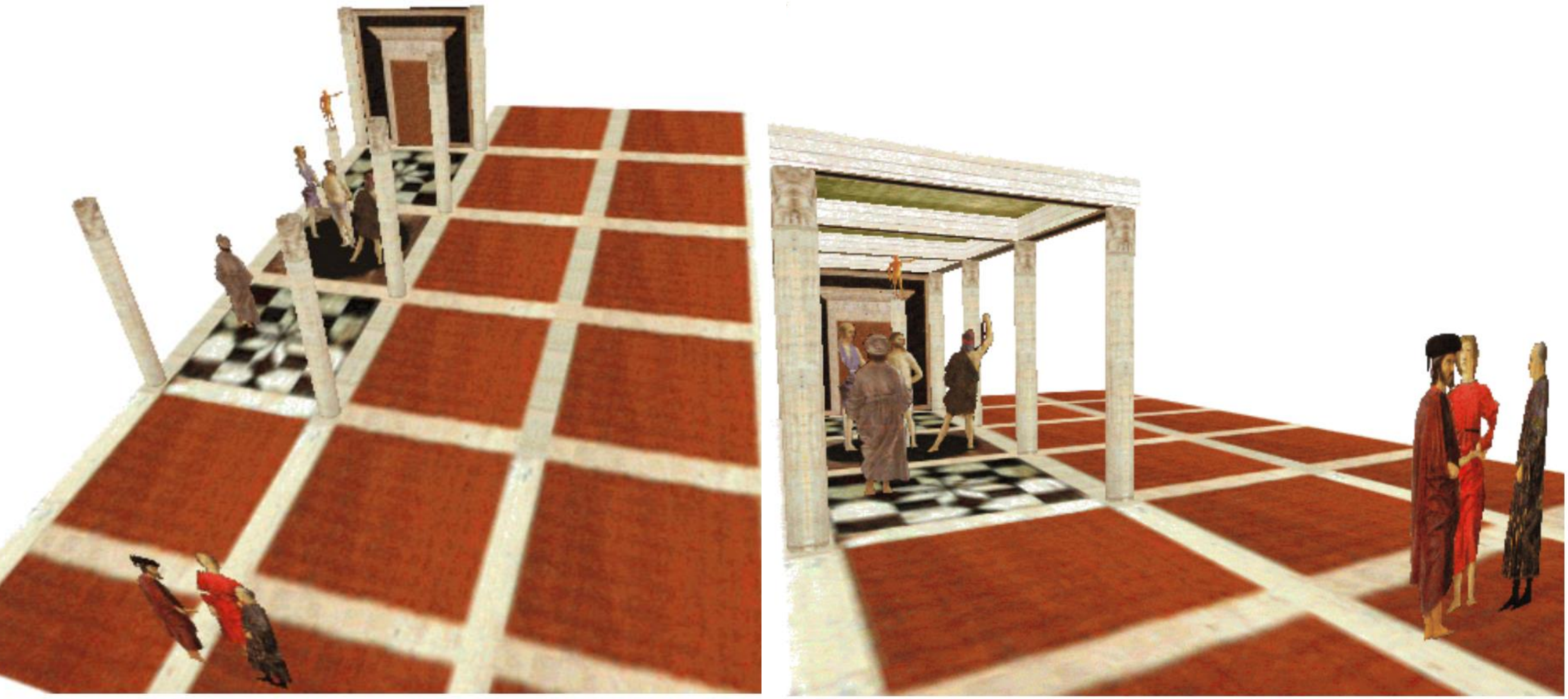


Image rectification: example



Piero della Francesca*, Flagellation,* ca. 1455

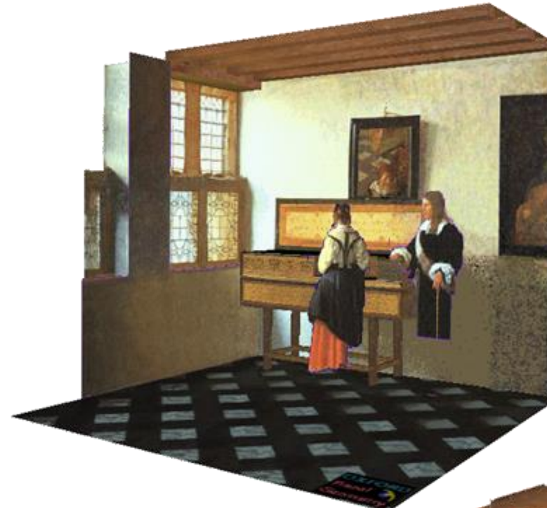
Application: 3D modeling from a single image



A. Criminisi, M. Kemp, and A. Zisserman, [Bringing Pictorial Space to Life: computer techniques for the analysis of paintings](), *Proc. Computers and the History of Art*, 2002

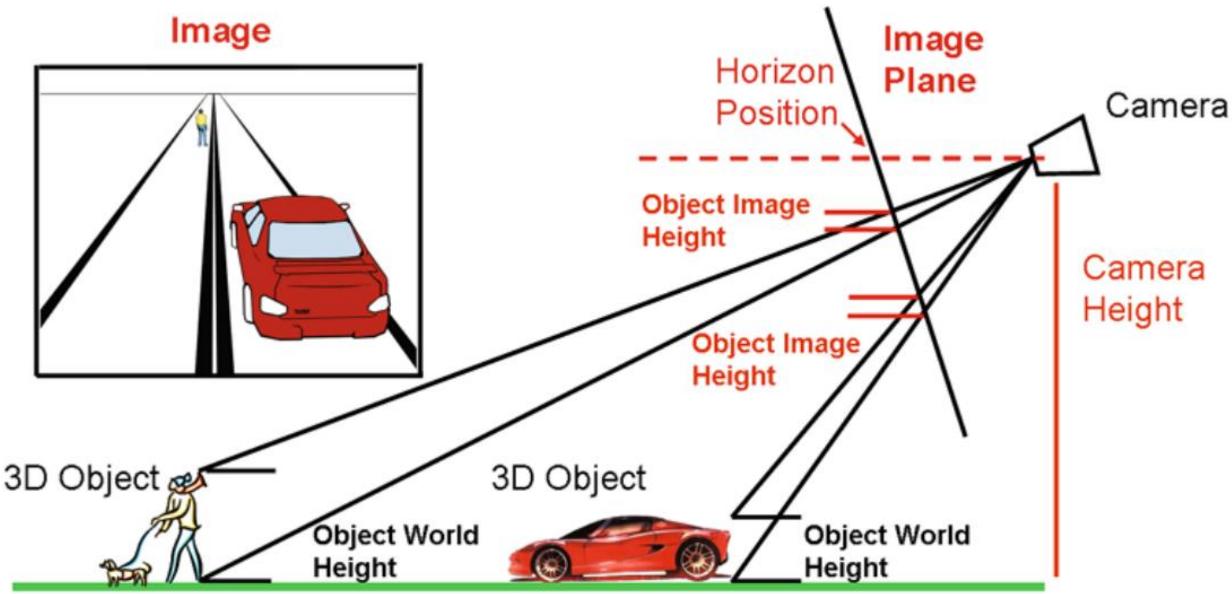
Application: 3D modeling from a single image

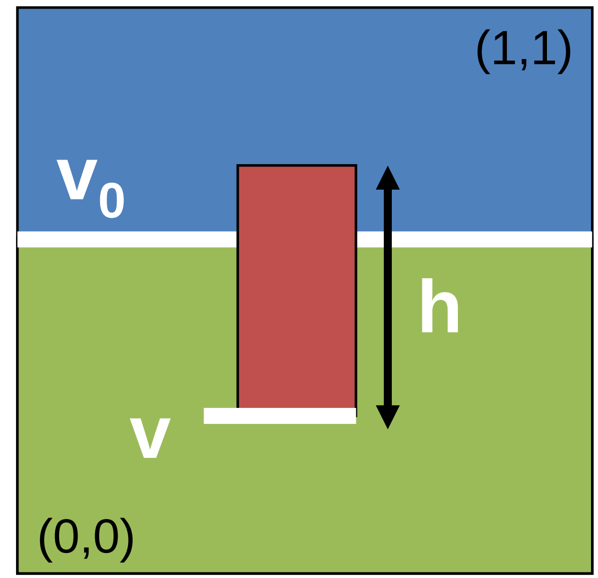
J. Vermeer, *Music Lesson*, 1662



A. Criminisi, M. Kemp, and A. Zisserman,[Bringing Pictorial Space to Life: computer techniques for the analysis of paintings](),
Proc. Computers and the History of Art, 2002

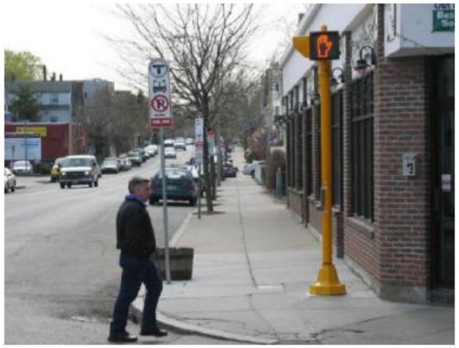
Application: Object Detection





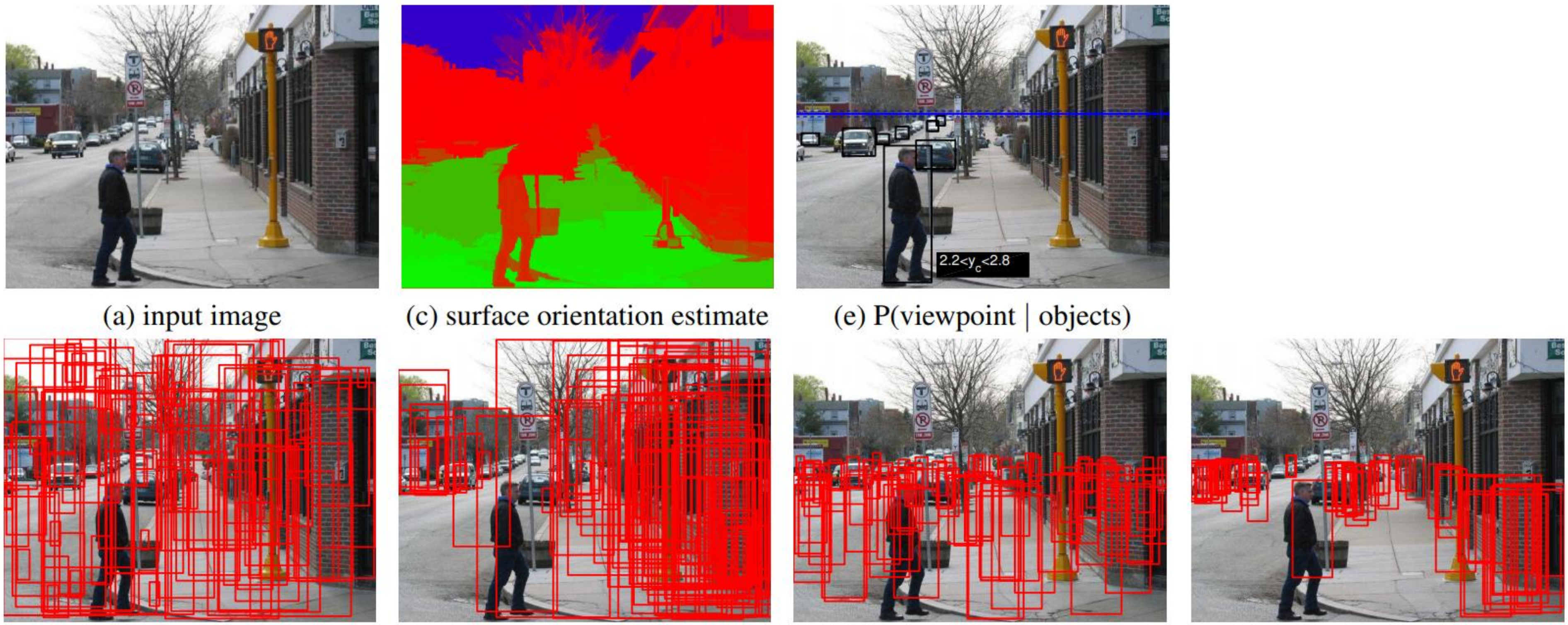
"Reasonable" approximation: $y_{object} \approx \frac{h y_{camera}}{v_0 - v}$

Application: Object detection



(a) input image

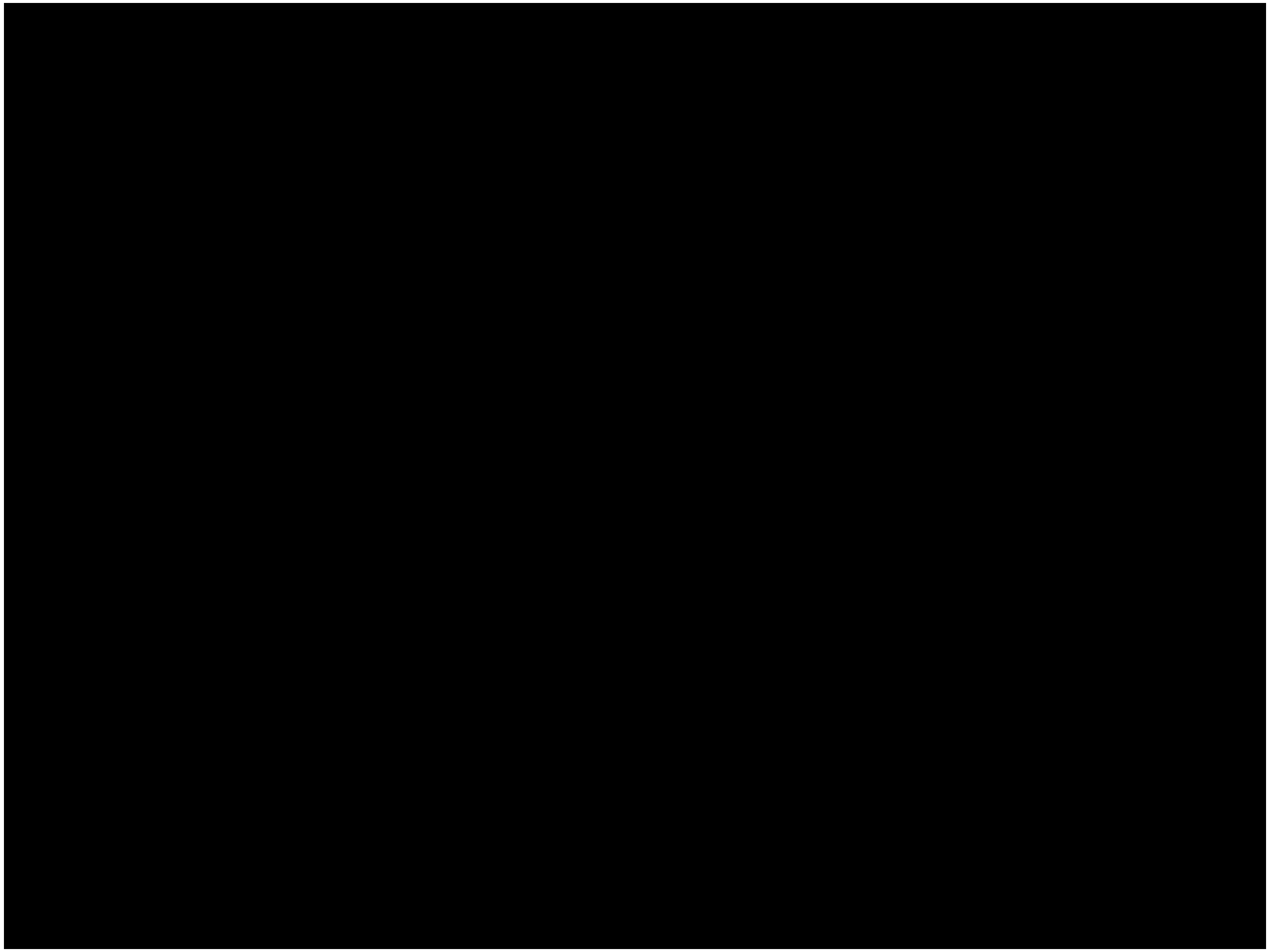
Application: Object detection



(a) input image (c) surface orientation estimate (e) P(viewpoint | objects)

(b) P(person) = uniform (d) P(person | geometry) (f) P(person | viewpoint) (g) P(person|viewpoint,geometry)

Application: Image Editing



K. Karsch and V. Hedau and D. Forsyth and D. Hoiem, Rendering Synthetic Objects into Legacy Photographs, *SIGGRAPH Asia* 2011

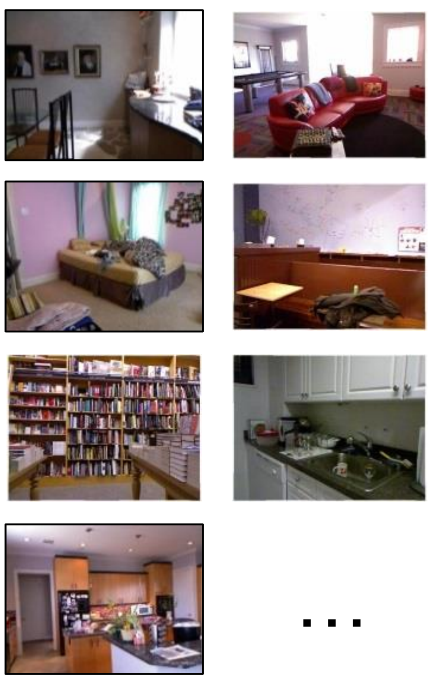
Application: Estimating Layout

V. Hedau, D. Hoiem, D. Forsyth
Recovering the spatial layout of cluttered rooms ICCV 2009

Unsupervised Learning

Can we learn 3D simply from regularities?

Image
Collection



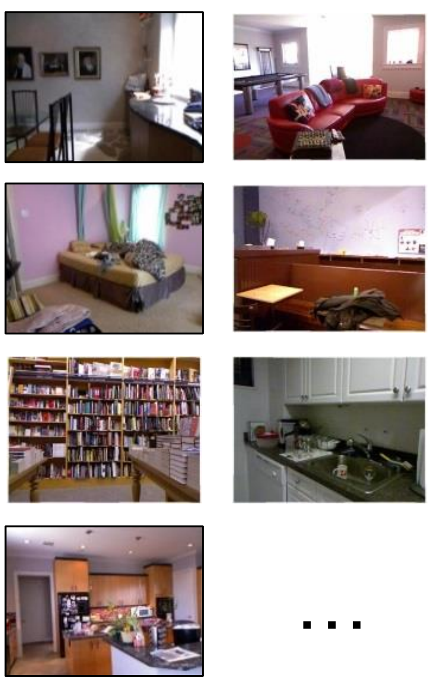
…

D.F. Fouhey, W. Hussain, A. Gupta, M. Hebert. *Single Image 3D without a Single 3D Image.* ICCV 2015.

Unsupervised Learning

Can we learn 3D simply from regularities?

Image Collection



\+

Tools From Geometry

Vanishing Points

D.F. Fouhey, W. Hussain, A. Gupta, M. Hebert. *Single Image 3D without a Single 3D Image.* ICCV 2015.

Unsupervised Learning

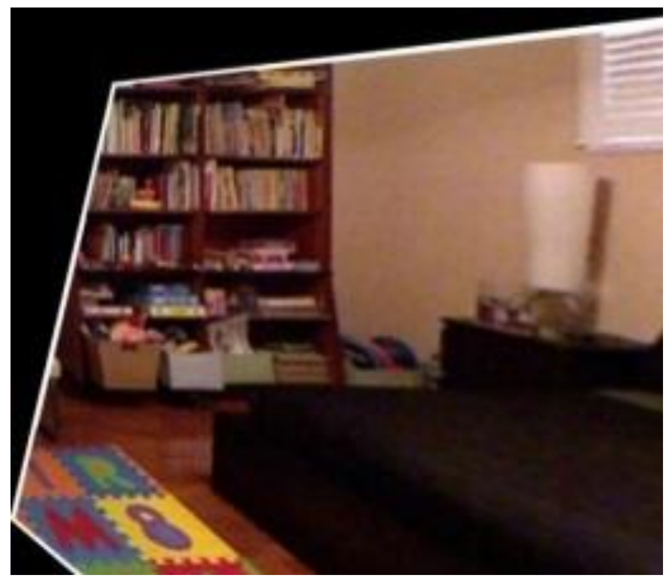
Can we learn 3D simply from regularities?

Image Collection

...

+

Tools From Geometry



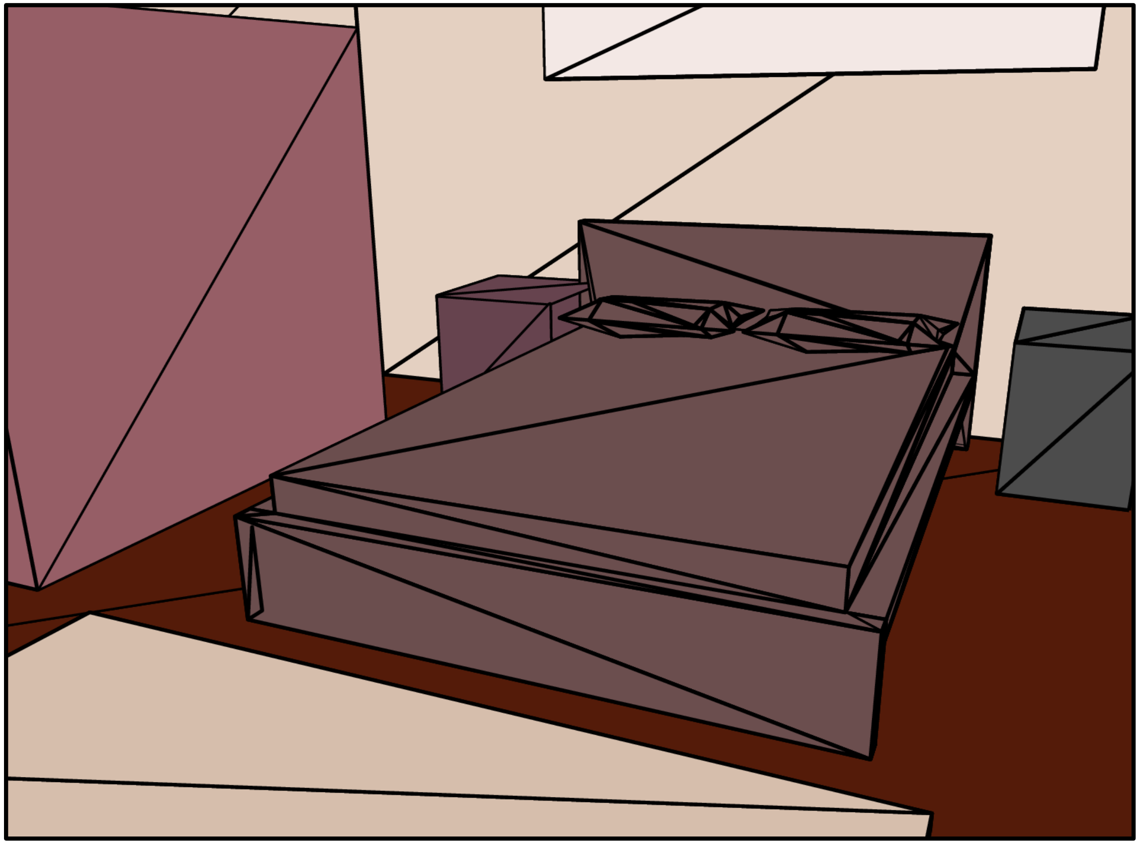
Fronto-Parallel Image

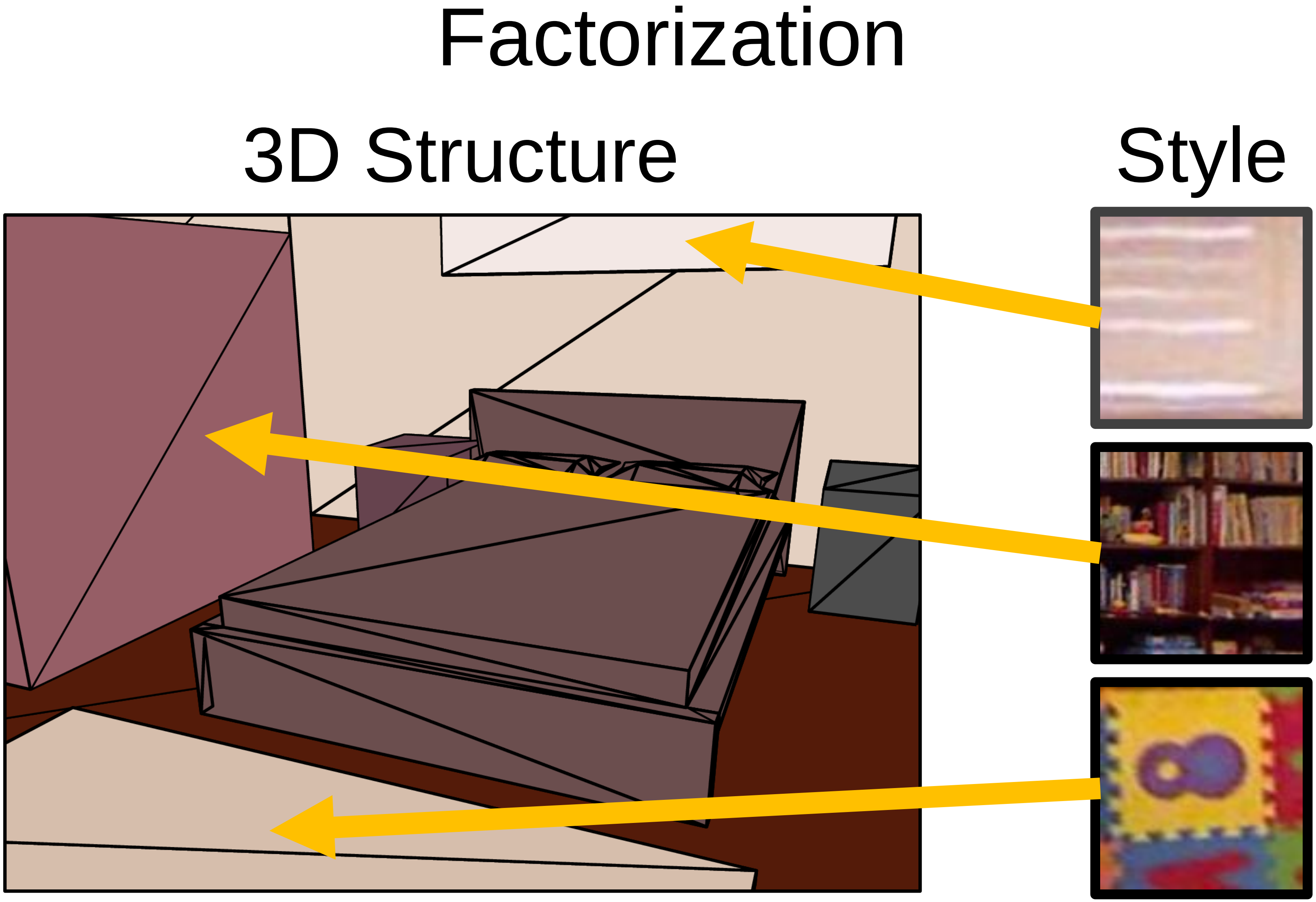
D.F. Fouhey, W. Hussain, A. Gupta, M. Hebert. *Single Image 3D without a Single 3D Image.* ICCV 2015.

Factorization

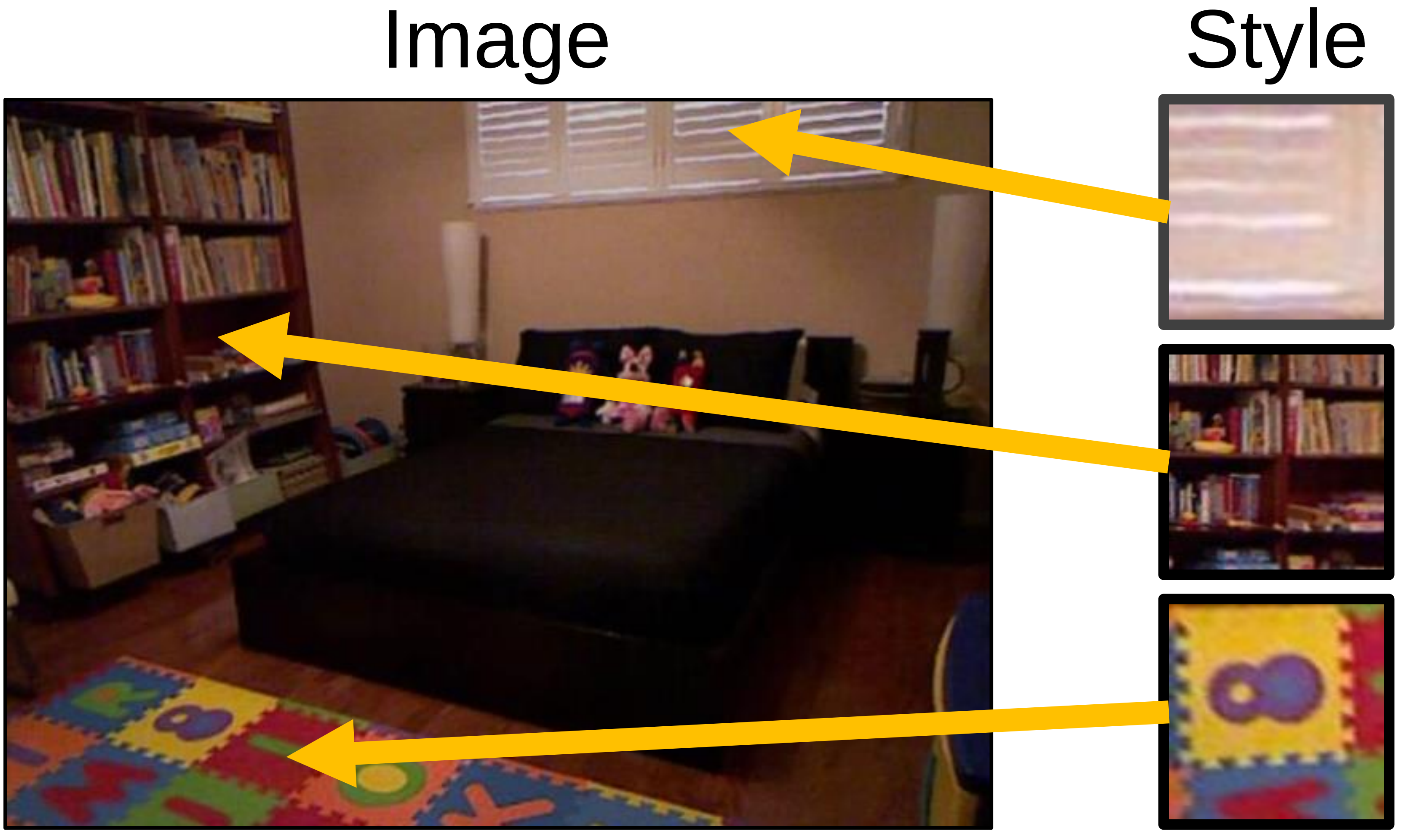
Factorization

3D Structure



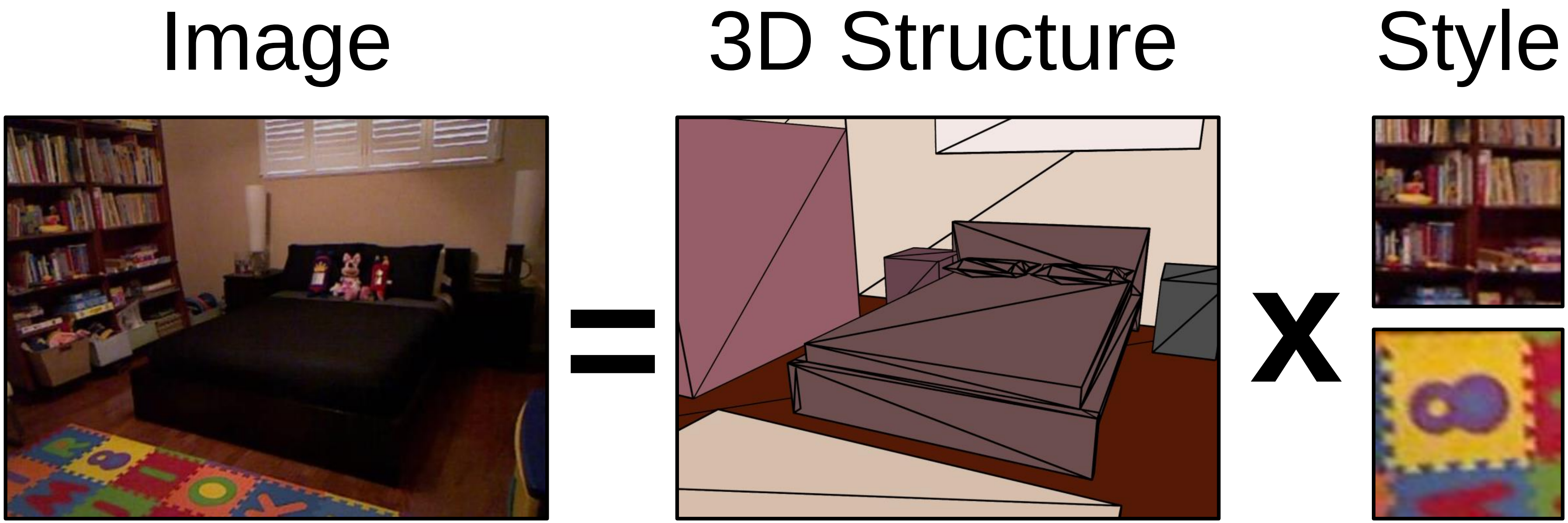

Factorization
3D Structure
Style

Factorization



Style Elements

Factorization

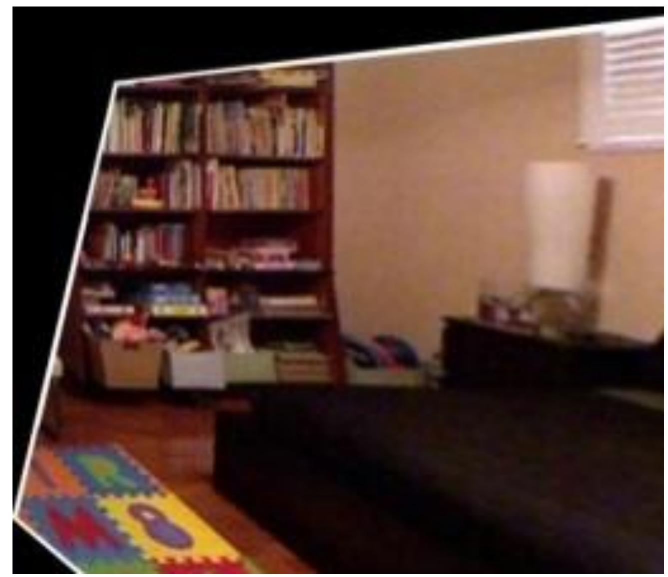


Solving for Style

Image

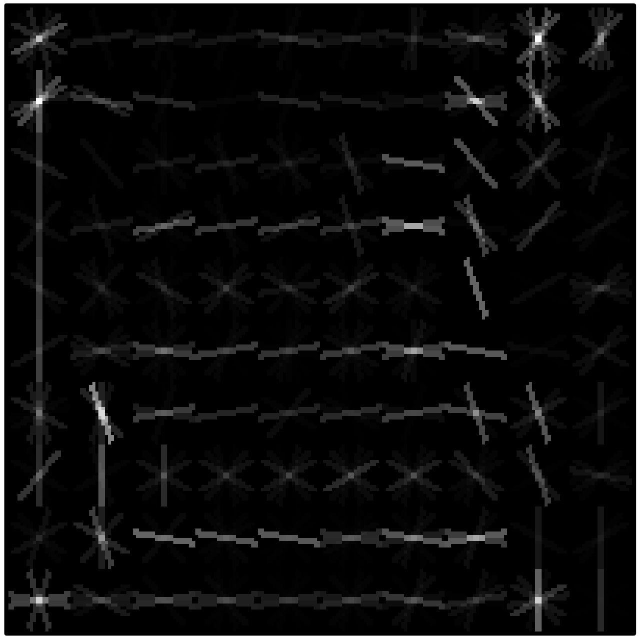
Vanishing
Points

Style



Fronto-Parallel
Image

Solving for 3D Structure



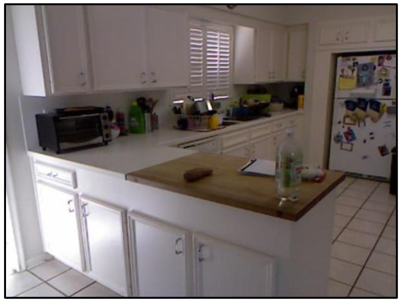
Style Element

Input Image

HOG, Dalal and Triggs ’05; ELDA from Hariharan et al. ‘12

Solving for 3D Structure

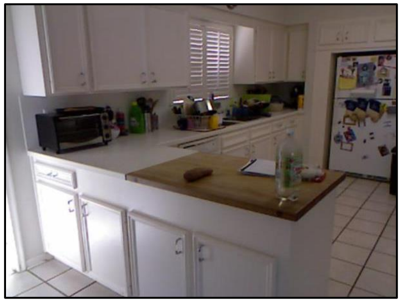
Style
Element



Input
Image

Solving for 3D Structure

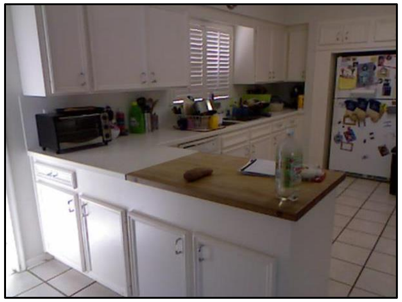
Style Element



Input Image

Solving for 3D Structure

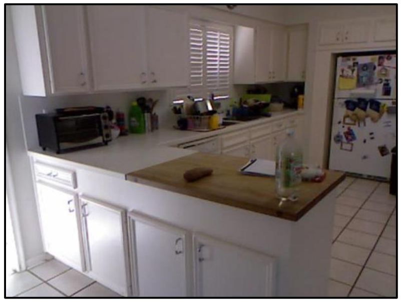
Style Element



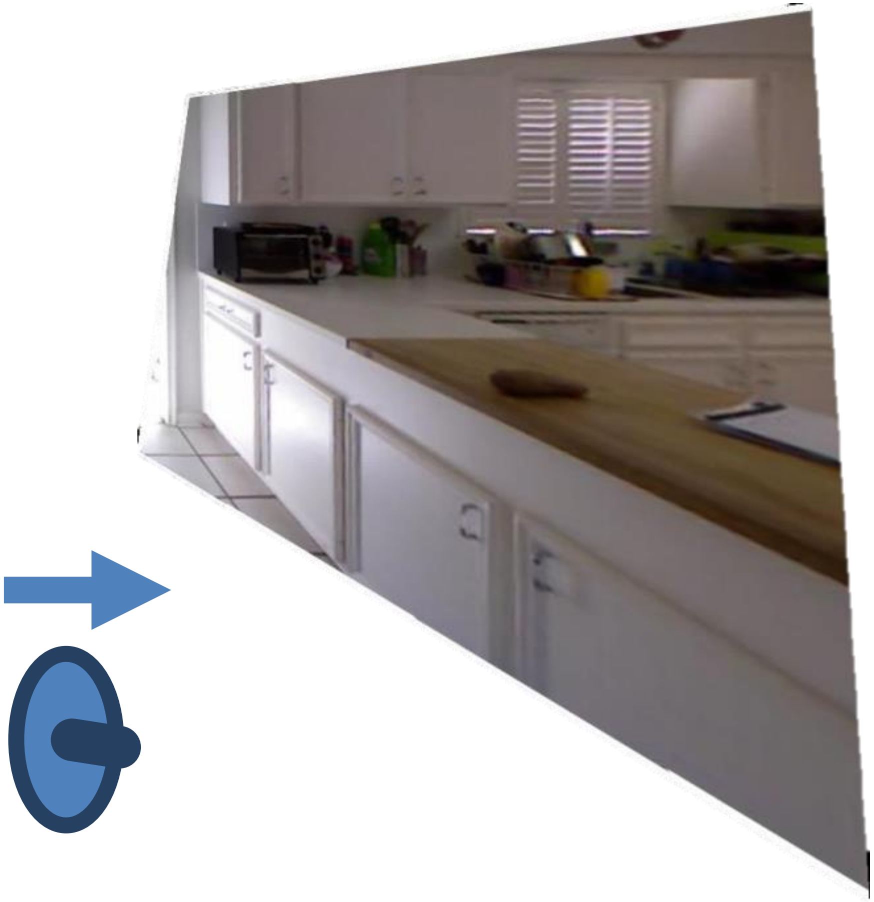
Input Image

Solving for 3D Structure

Style Element

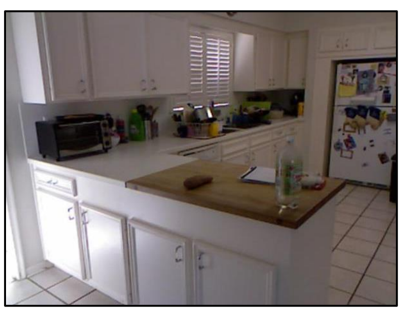


Input Image



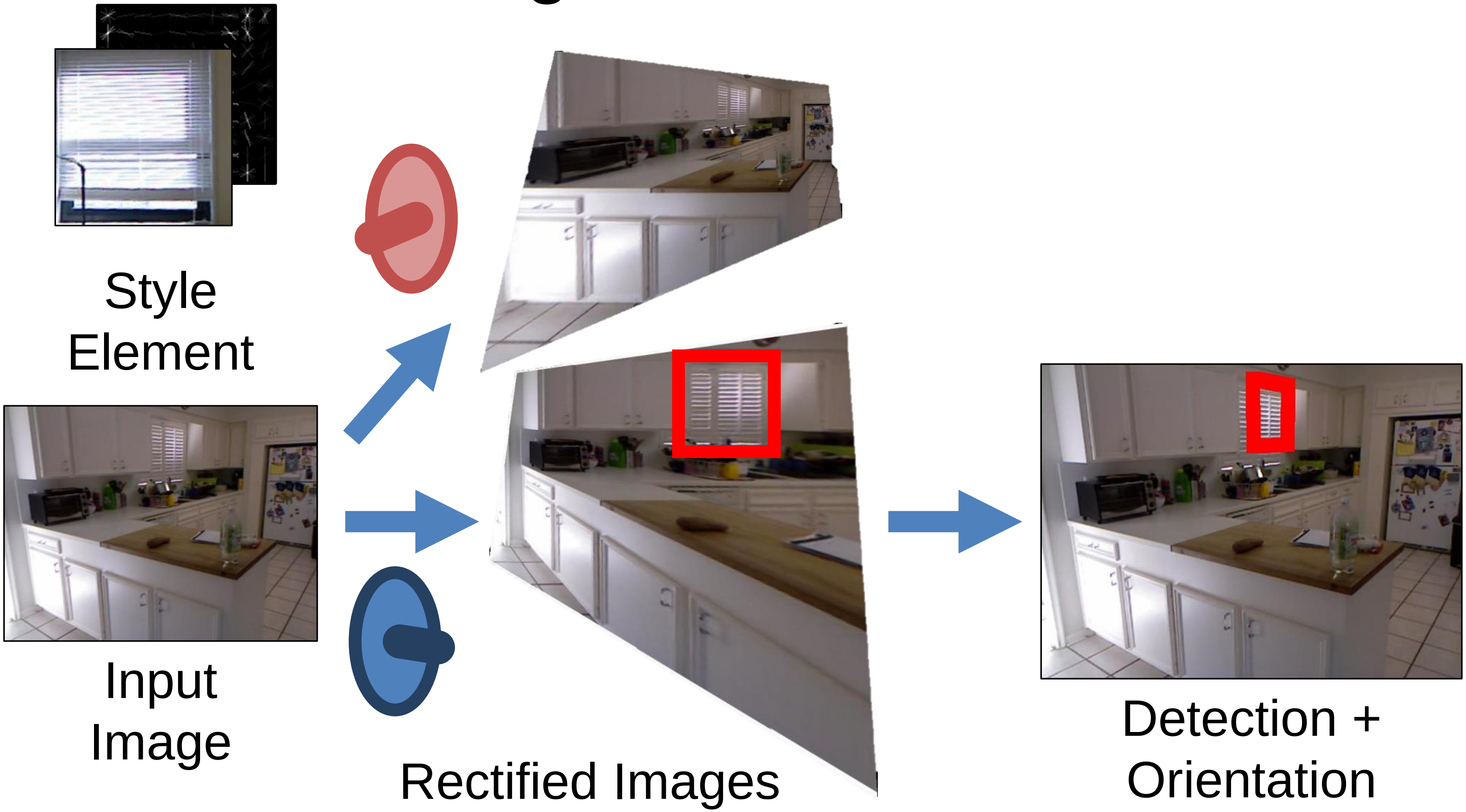
Solving for 3D Structure

Style
Element



Input
Image

Solving for 3D Structure



Solving for 3D over a Dataset

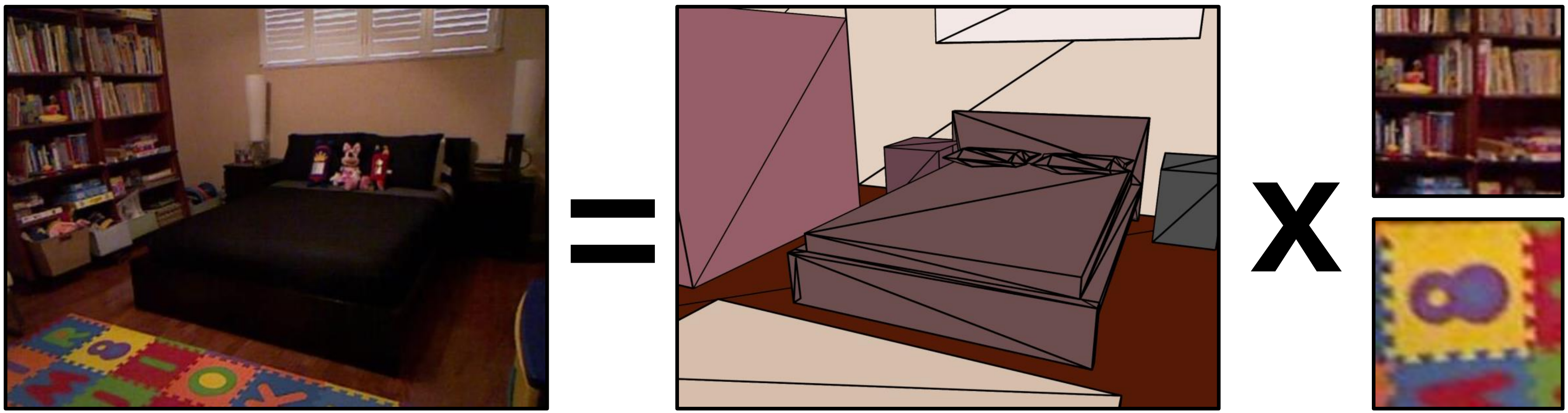
Style Element

Set of Images

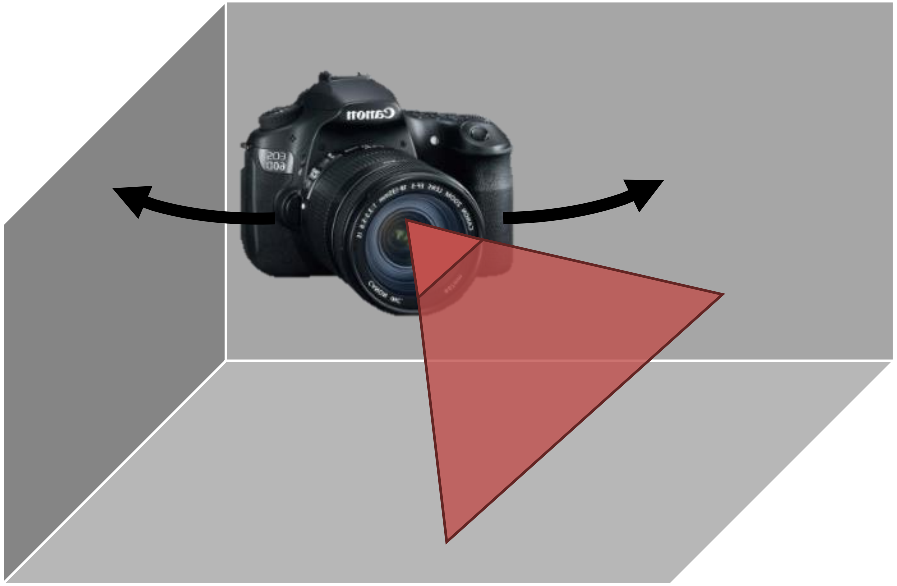
Detection + Orientation

Factorization

Image 3D Structure Style



Prior



On average: 3D structure is a camera inside a box, rotated uniformly

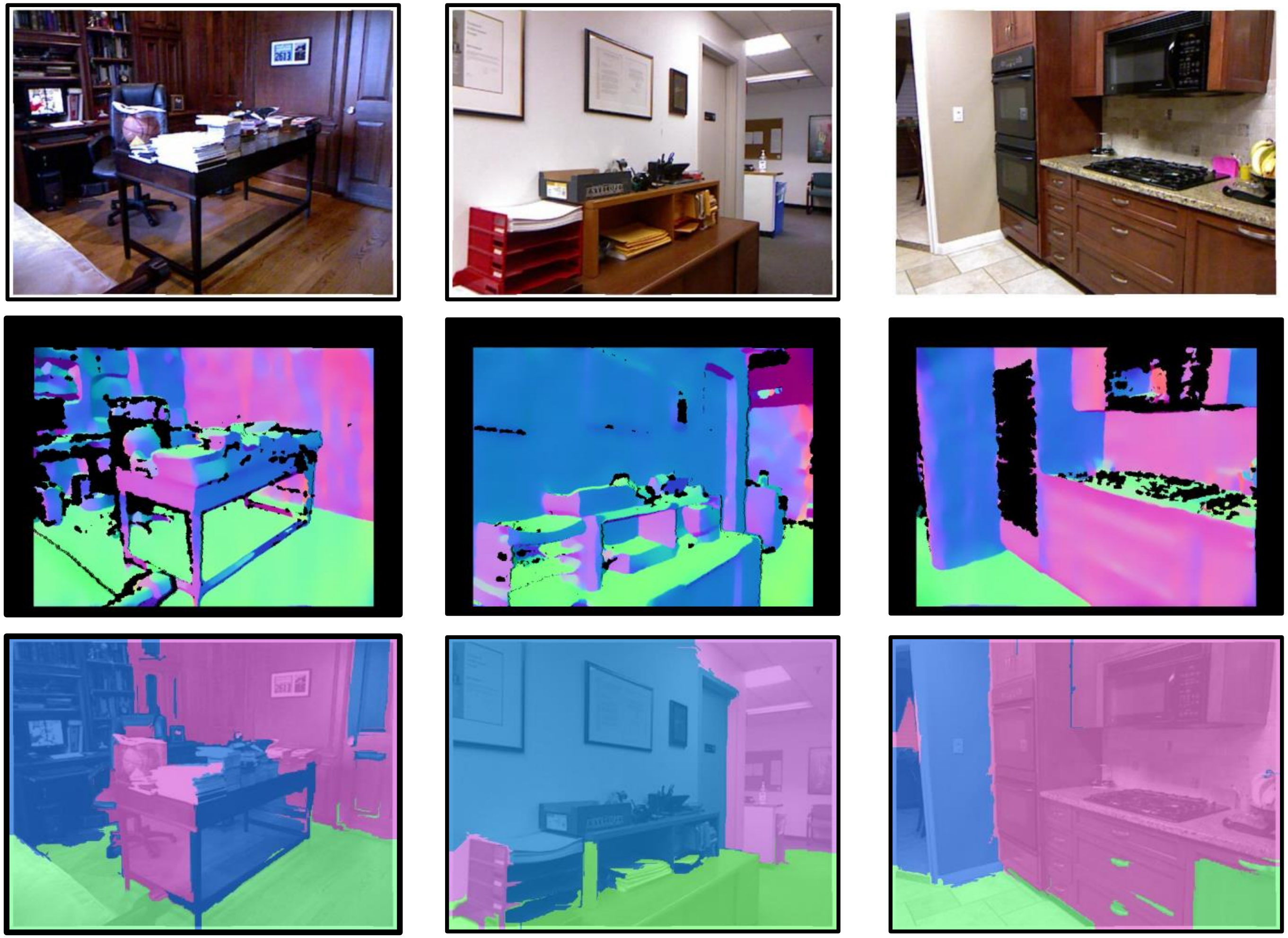
Discovered Style Elements

Results

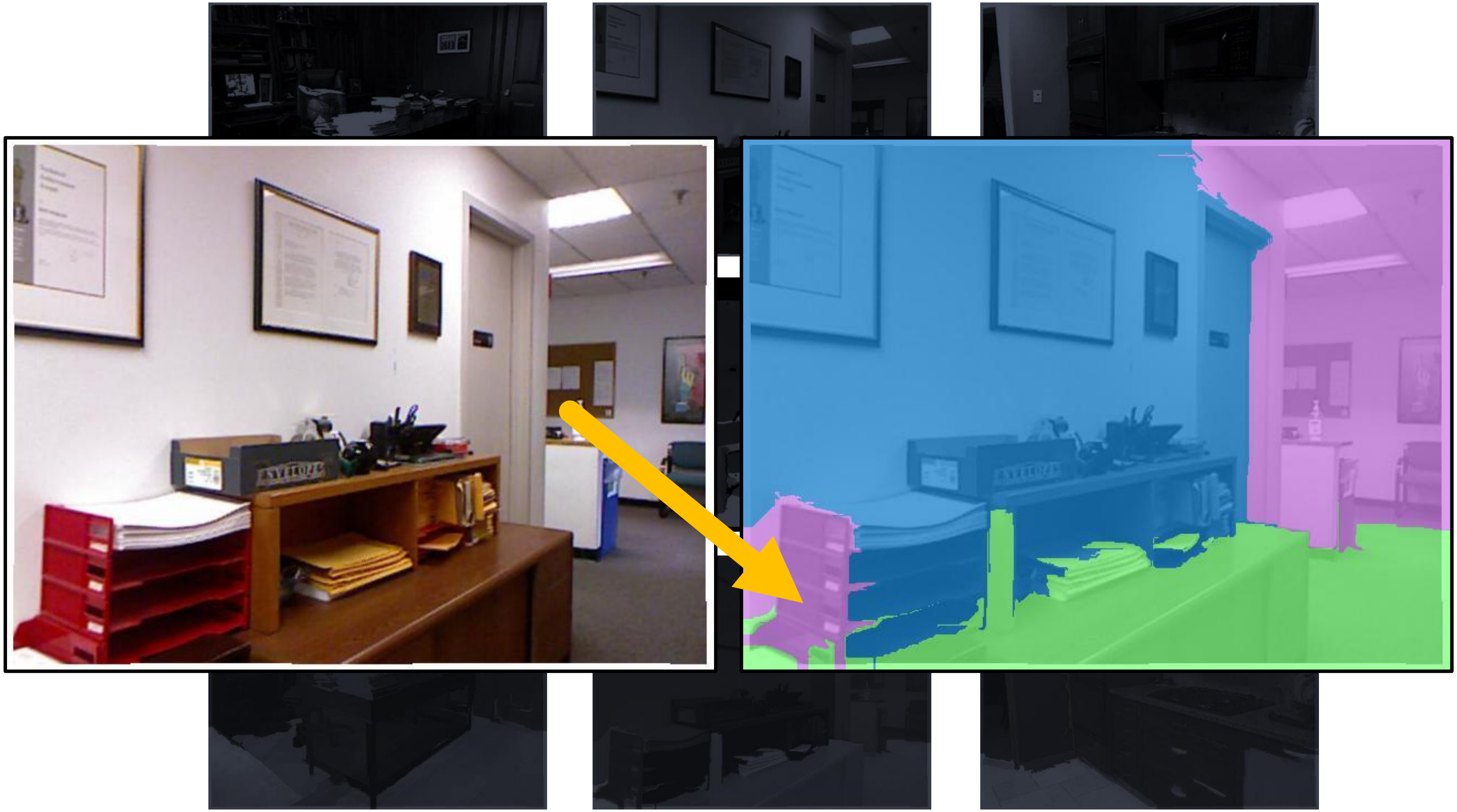
Input

GT

Output



Results



Scaling Up To The World

RGBD Datasets

What about?

Places-205, Zhou et al. NIPS 2014

Style Learned from Internet

Automatically Discovered Style Elements

Supermarket

Laundromat

Museum

Locker Room

Places-205, Zhou et al. NIPS 2014

Learning from the Internet