



MIT CSAIL

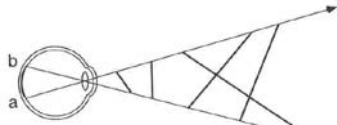
6.869: Advances in Computer Vision

MIT
COMPUTER
VISION

Lecture 13

Shape from X

Depth Perception: The inverse problem



Monocular cues to depth

- **Absolute depth cues:** (assuming known camera parameters) these cues provide information about the absolute depth between the observer and elements of the scene
- **Relative depth cues:** provide relative information about depth between elements in the scene (this point is twice as far at that point, ...)

Relative depth cues



Simple and powerful cue, but hard to make it work in practice...

Atmospheric perspective

- Based on the effect of air on the color and visual acuity of objects at various distances from the observer.
- Consequences:
 - Distant objects appear bluer
 - Distant objects have lower contrast.



Atmospheric perspective

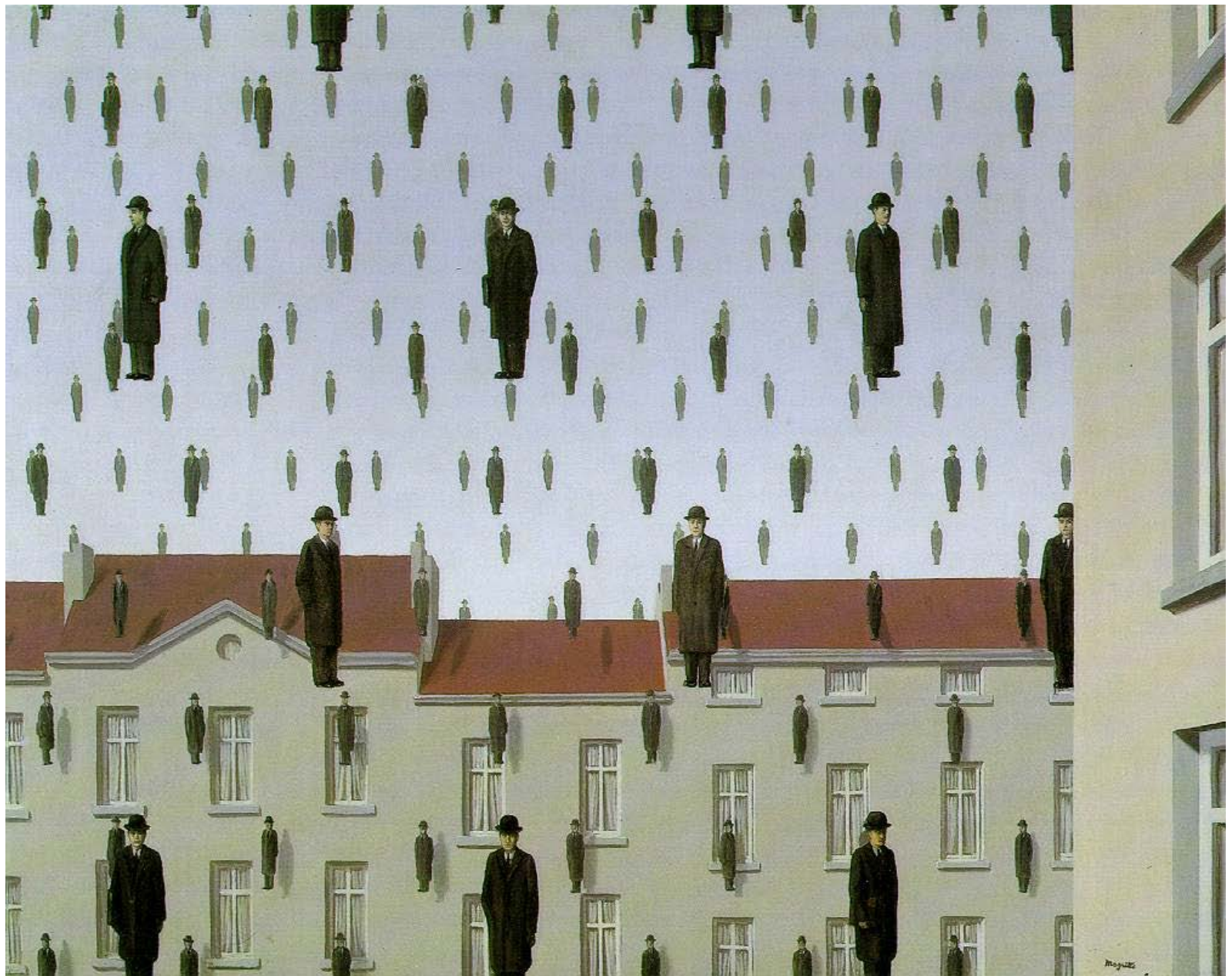




[Claude Lorrain](#) (artist)

French, 1600 - 1682

Landscape with Ruins, Pastoral Figures, and Trees, 1643/1655



[Golconde Rene Magritte]

Shadows



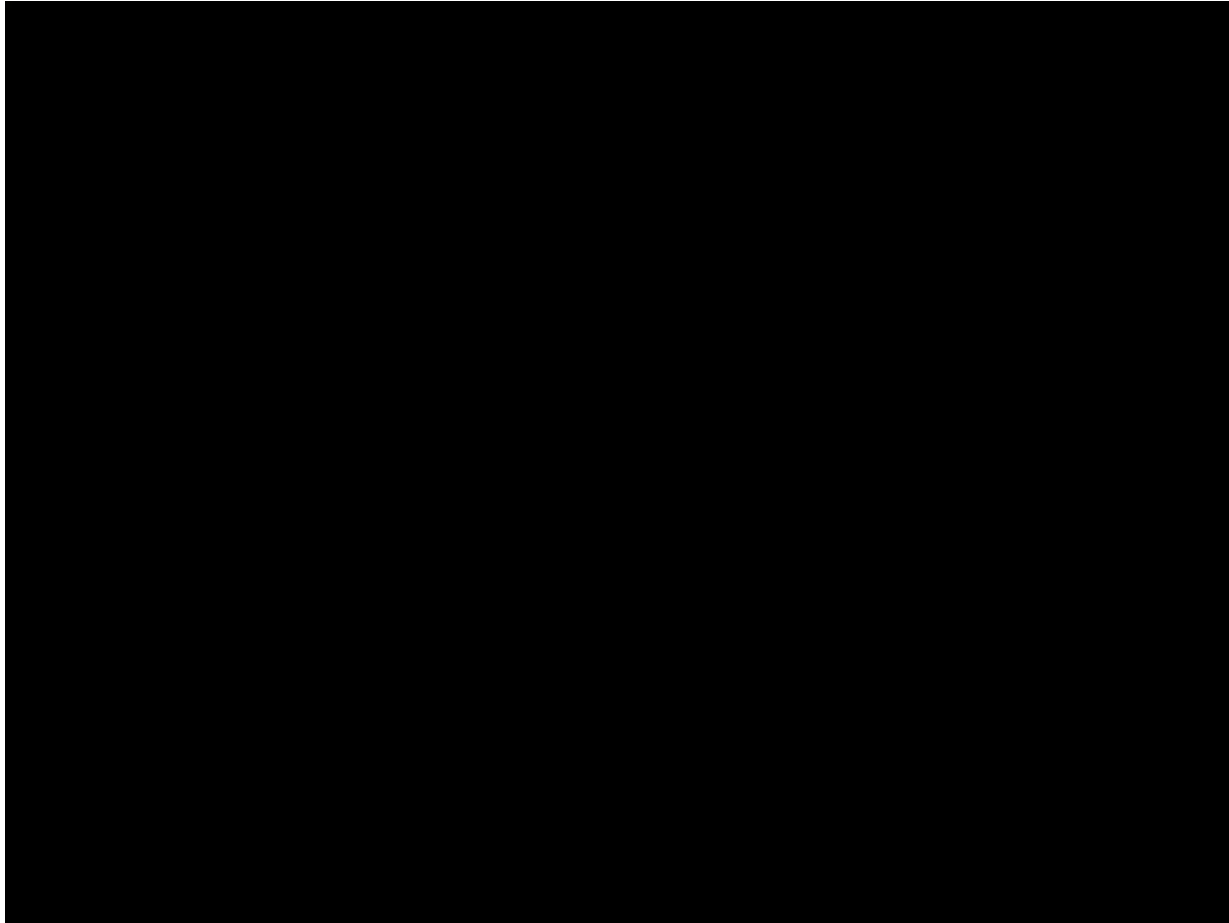
Cornell CS569 Spring 2008

Lecture 8 • 3

Shadows

video

Shadows



Linear perspective

Linear Perspective

Based on the apparent convergence of parallel lines to common vanishing points with increasing distance from the observer.

(Gibson : “perspective order”)

In Gibson’s term, perspective is a characteristic of the visual field rather than the visual world. It approximates how we see (the retinal image) rather than what we see, the objects in the world.

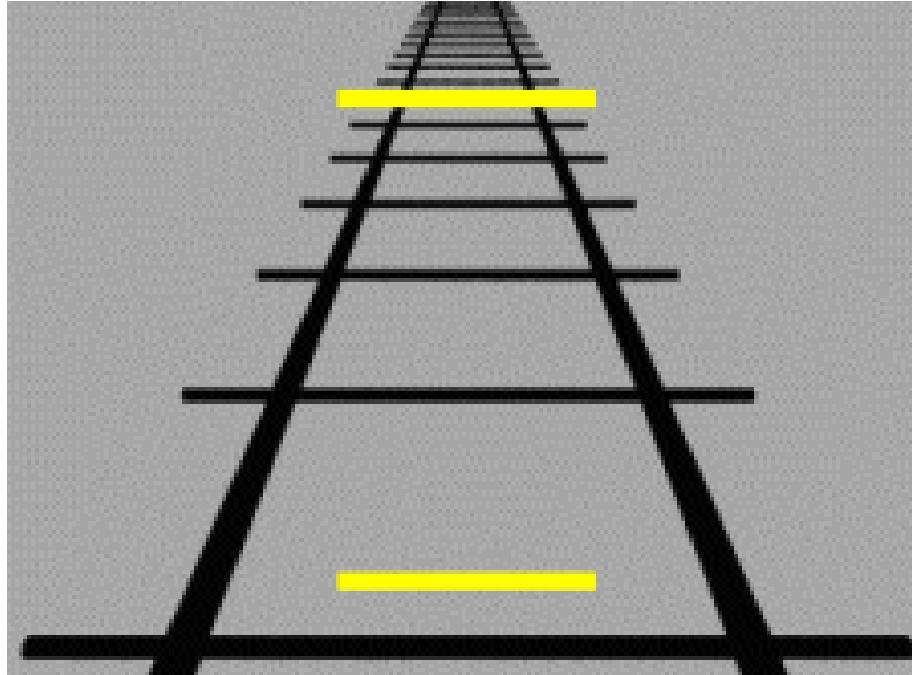
Perspective : a representation that is specific to one individual, in one position in space and one moment in time (a powerful immediacy).

Is perspective a universal fact of the visual retinal image ? Or is perspective something that is learned ?



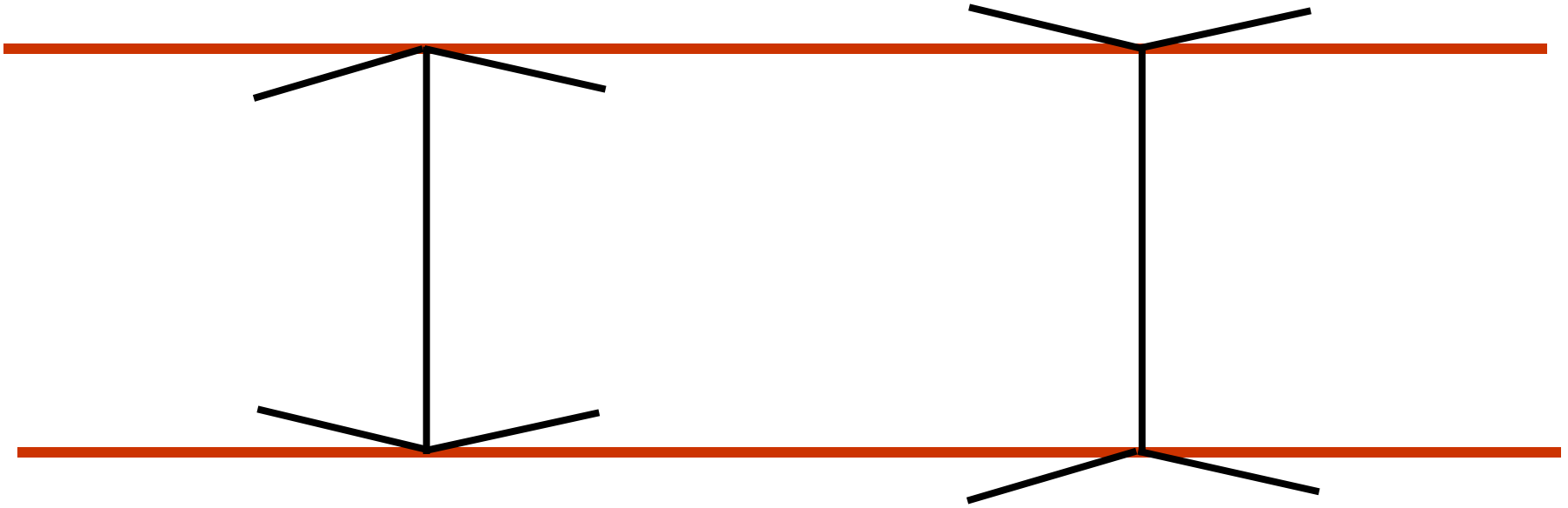
Simple and powerful cue, and easy to make it work in practice...

Linear Perspective



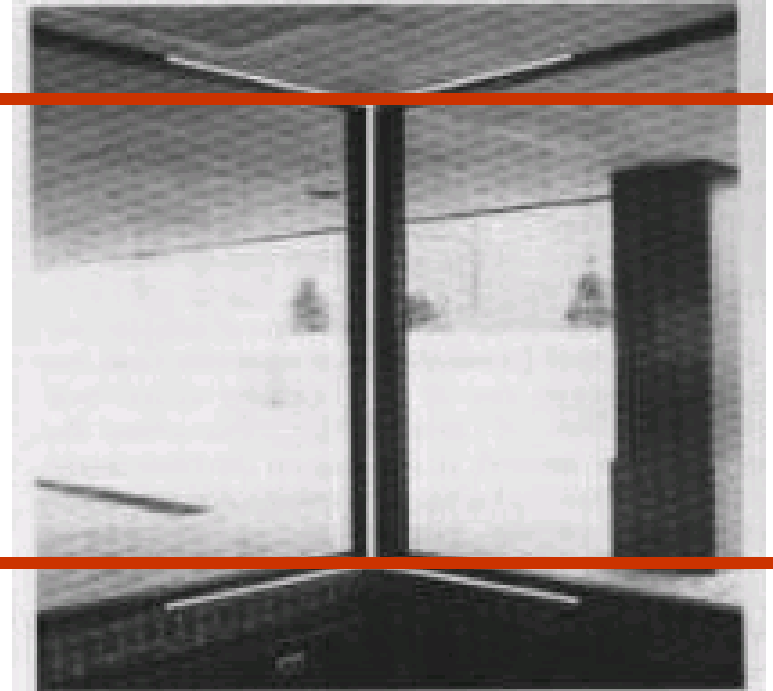
Ponzo's illusion

Linear Perspective



Muller-Lyer
1889

Linear Perspective

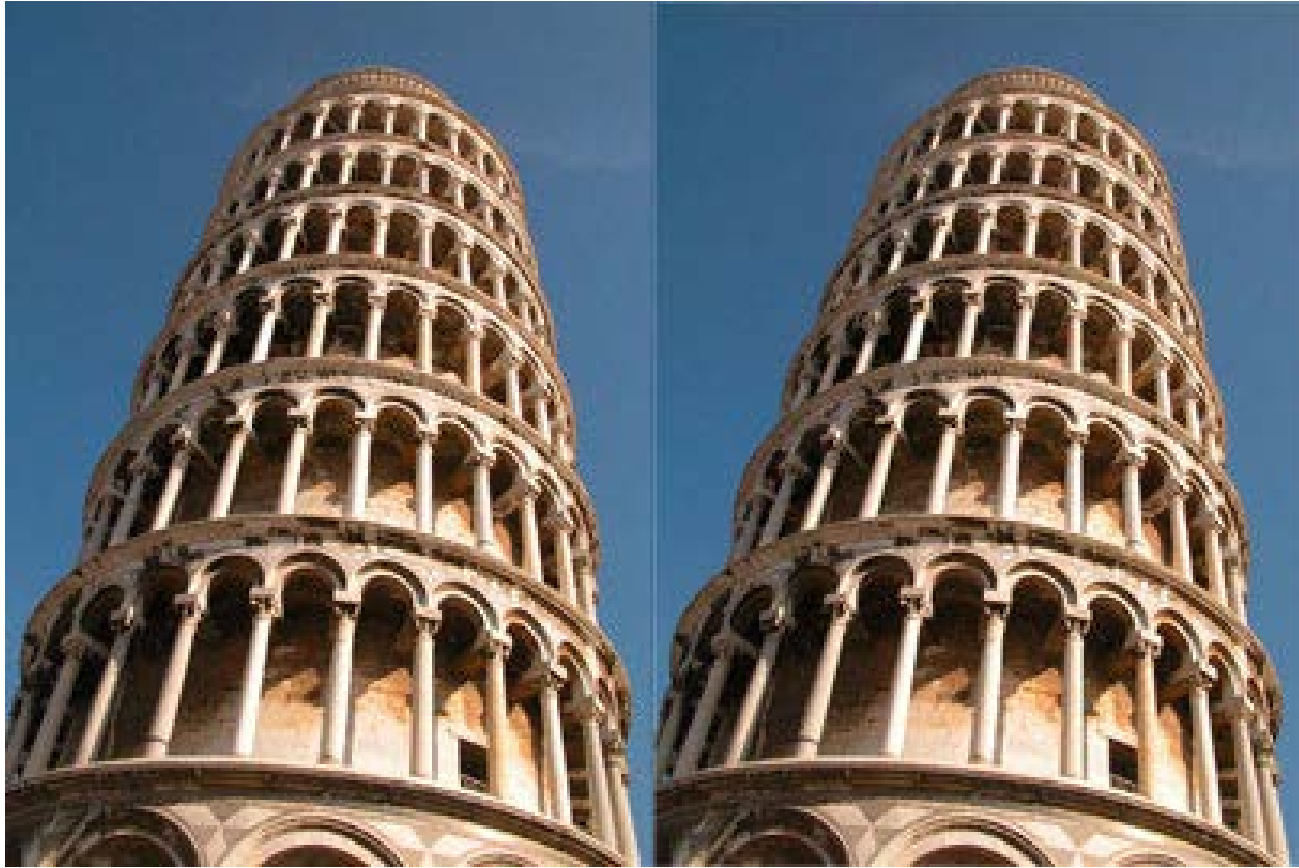


Muller-Lyer
1889

Linear Perspective

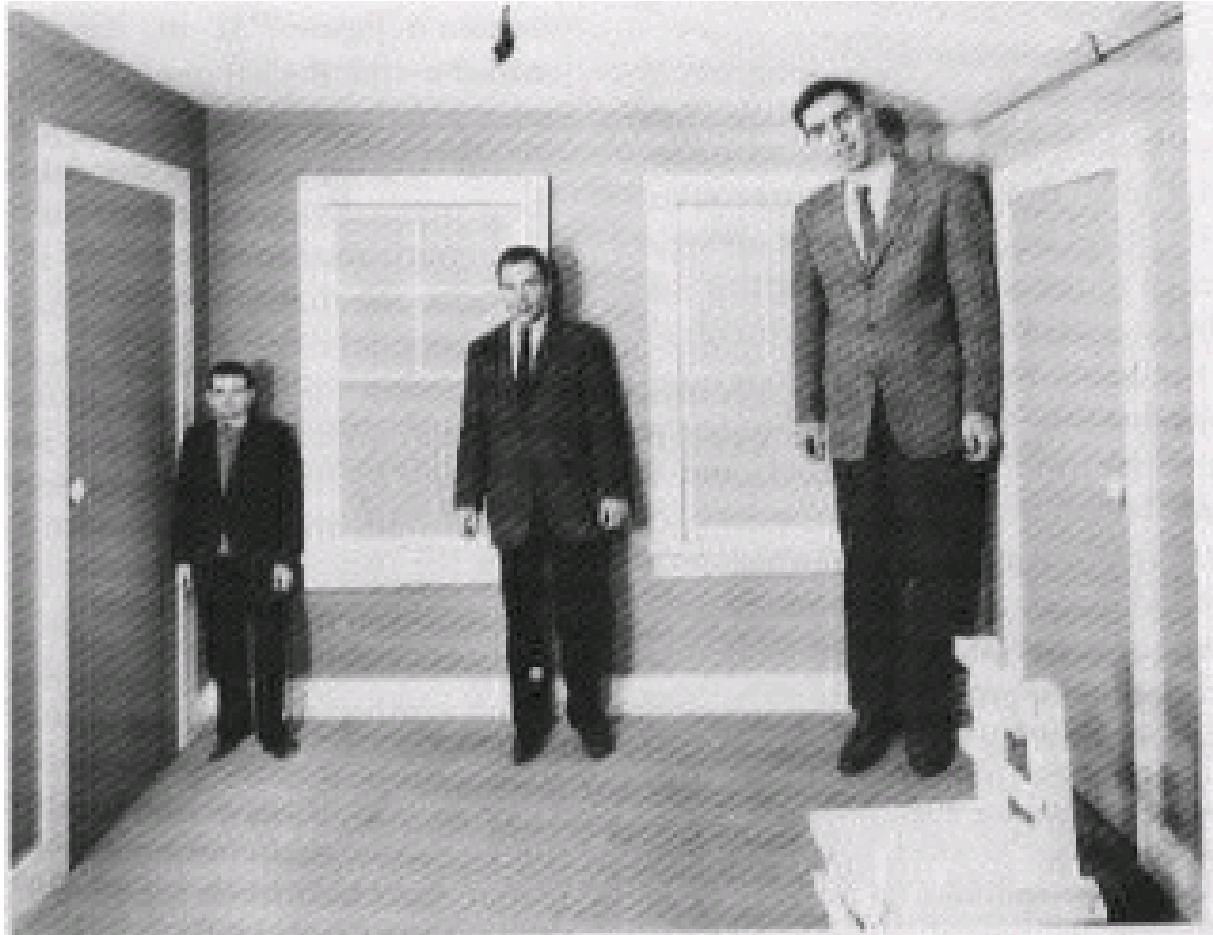
Muller-Lyer
1889

The two Towers of Pisa



Frederick Kingdom, Ali Yoonessi and Elena Gheorghiu of McGill Vision Research unit.

The strength of linear perspective



3D percept is driven by the scene, which imposes its ruling to the objects

Manhattan assumption

Application of the statistics of edges: Manhattan World



Many scenes of man-made environments are laid out on a 3-D “Manhattan” grid.

This 3-D structure imposes statistical regularities on the edges, and hence the image gradients, in the image.

These regularities allow us to infer the viewer orientation relative to the Manhattan grid and to detect targets *unaligned* to the grid.

Bayesian Model of Manhattan World

Evidence for line edges -- x, y, z or random lines -- provided by the image gradient. Prior on occurrence of these edges.

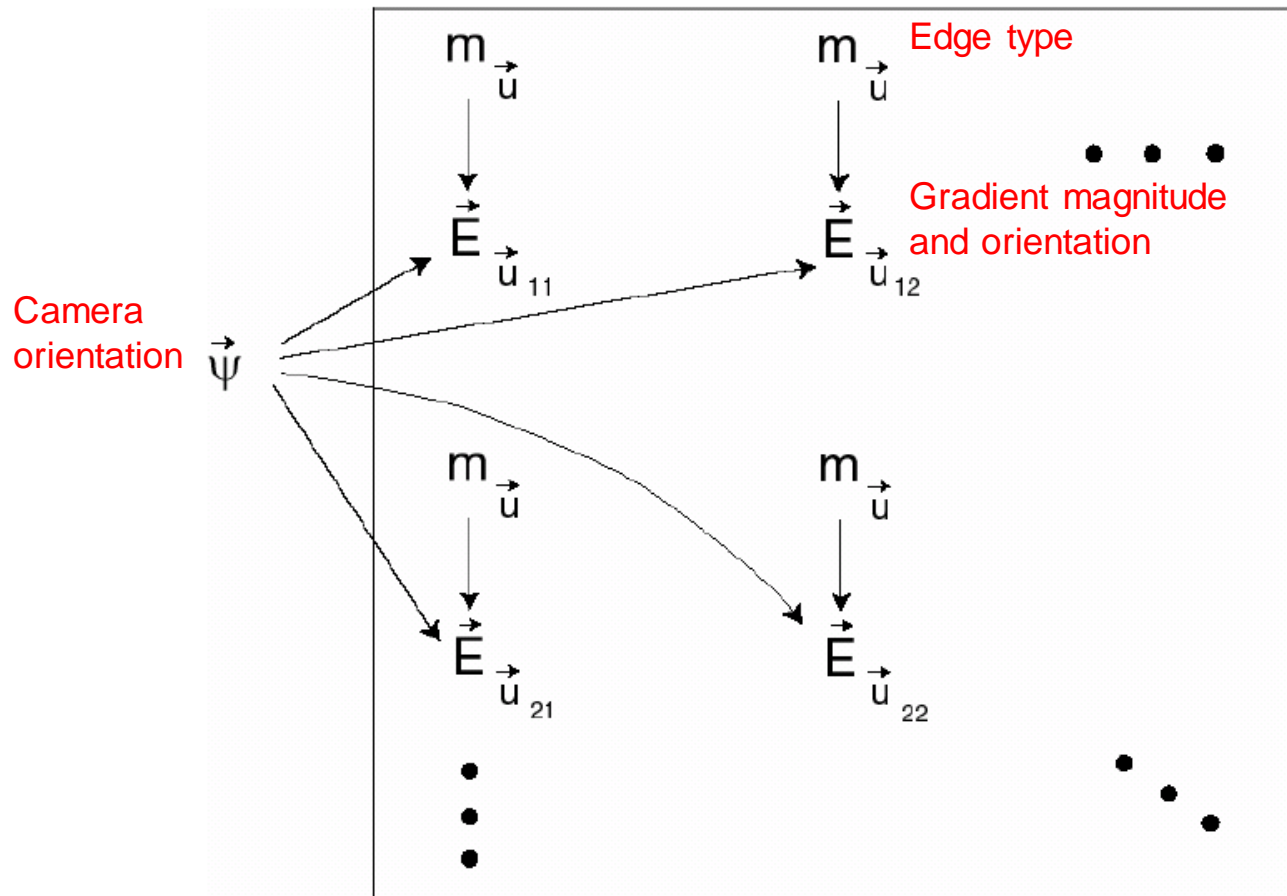
Image gradient magnitude provides evidence for presence or absence of edges, using P_{on} and P_{off} distributions.

Image gradient direction provides information about edge *orientations*.

Hidden assignment variables: at each pixel, is there an x, y, z or random line, or no edge at all?

If we knew this assignment at each pixel, and the camera orientation Ψ , we could predict likely values of image gradient magnitude and direction, $\vec{E}_{\vec{u}} = (E_{\vec{u}}, \phi_{\vec{u}})$

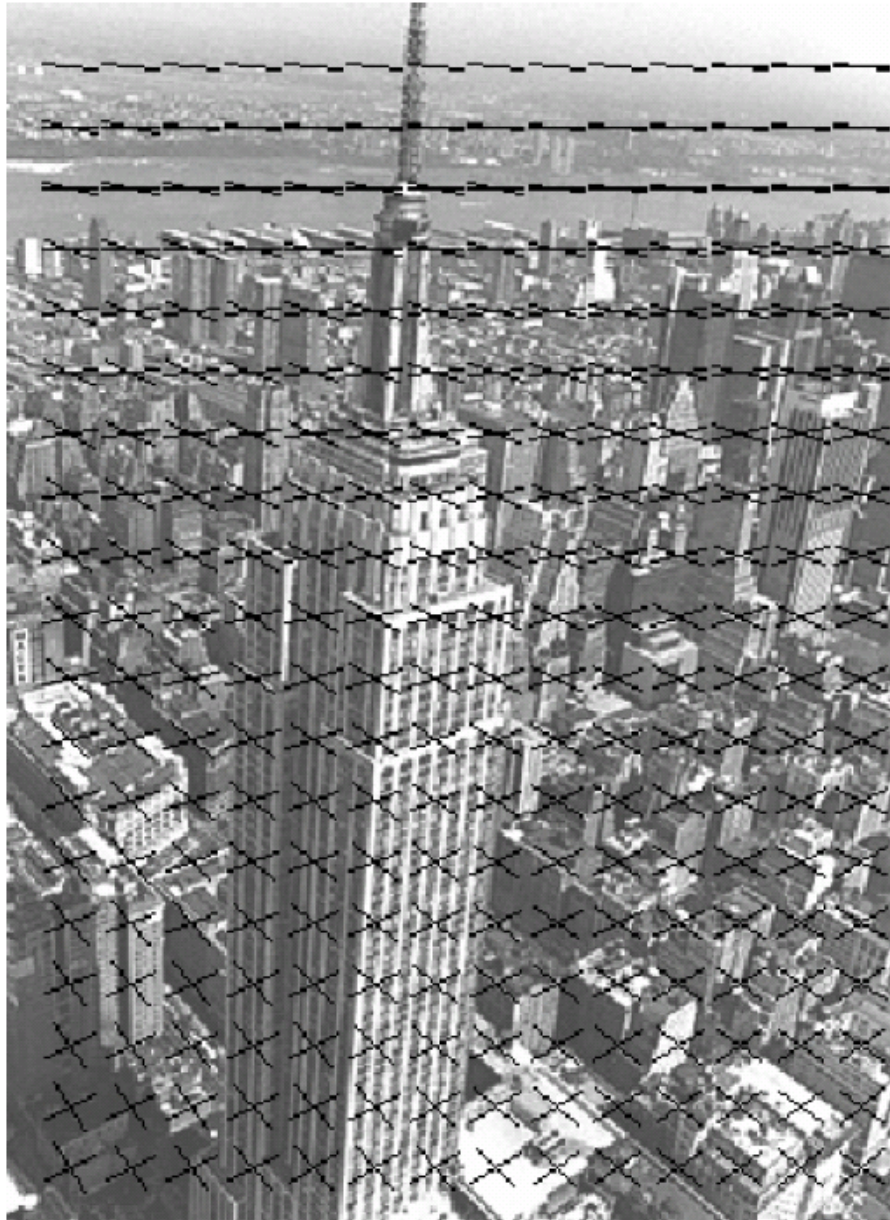
Evidence over all pixels: Bayes net of full Bayesian model



Box represents entire image, with an image gradient vector and assignment variable at each pixel location \vec{u}_{ij}

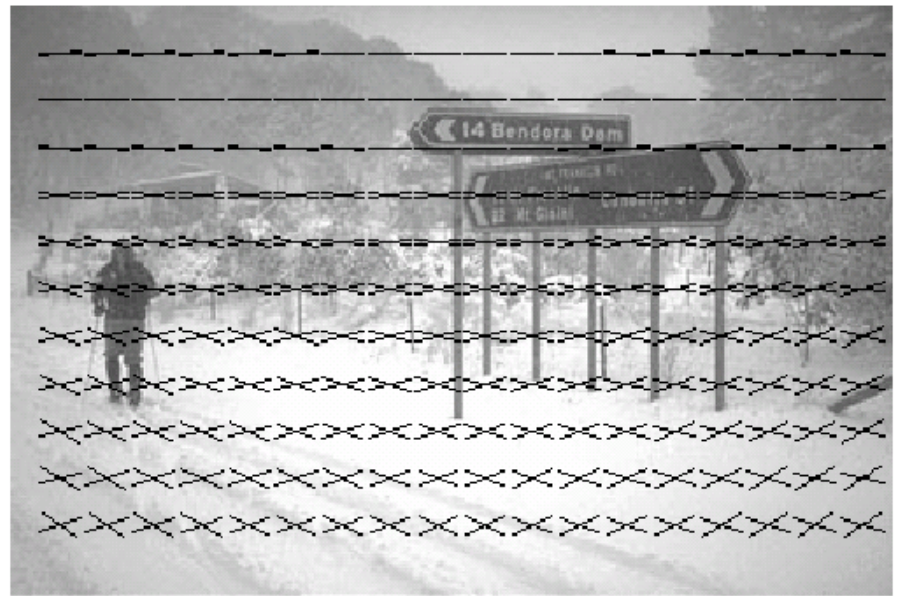
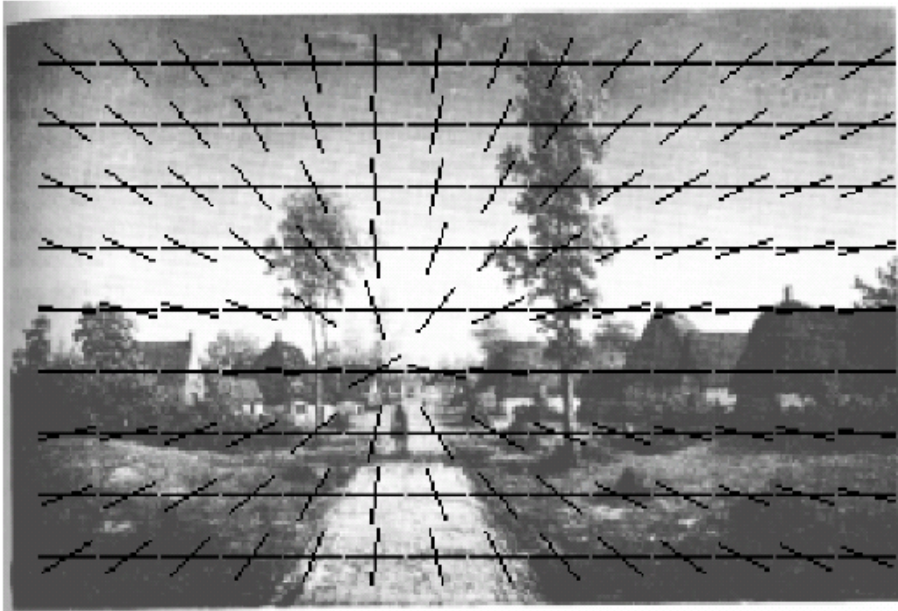
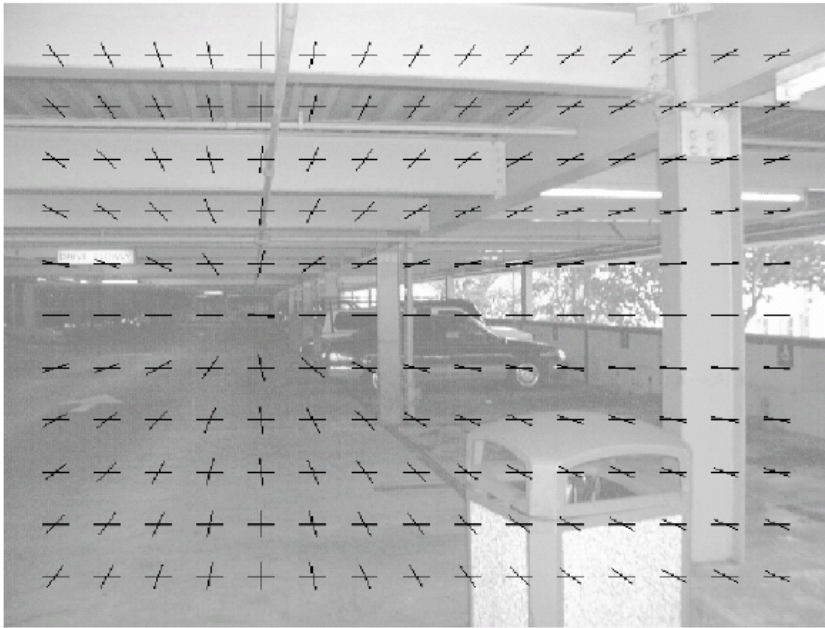
Structure of net graphically illustrates assumption of conditional independence across pixels.

Experimental Results



Estimate of *most probable camera orientation given image*, rendered in terms of the corresponding orientations of x and y lines (drawn in black).

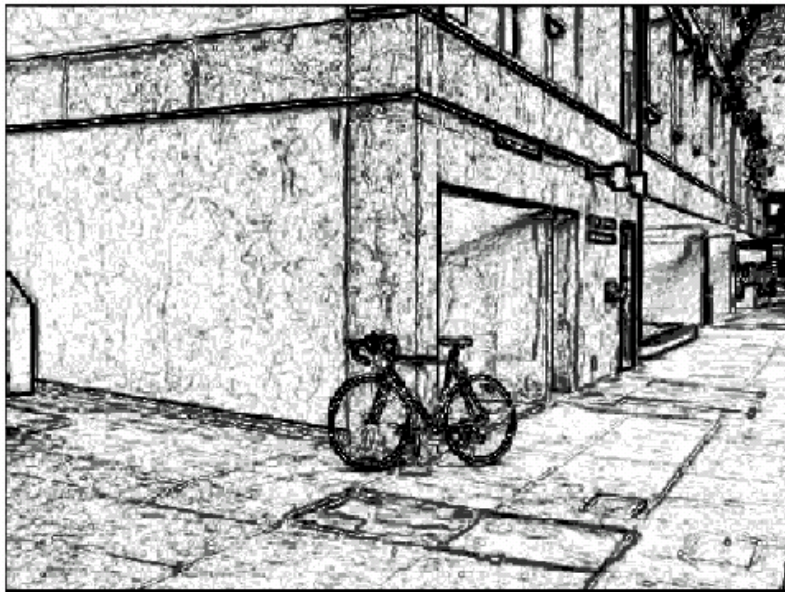
Note how the x lines align with the sides of buildings that are visible and facing left. The y lines align with the other visible sides facing right.



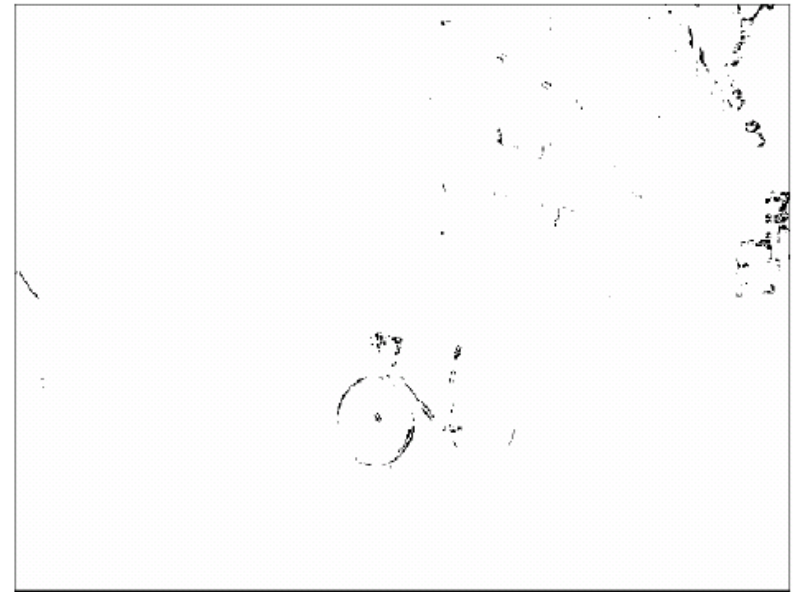
Outlier detection

Slide by James Coughlan

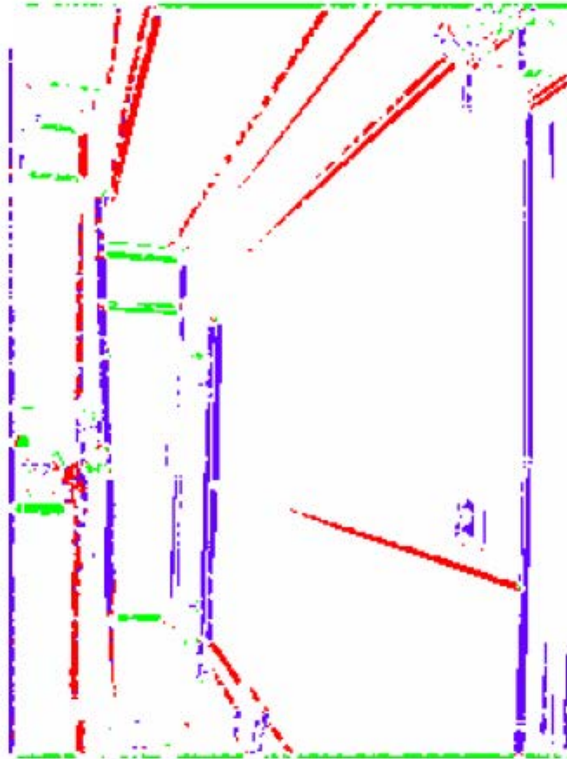
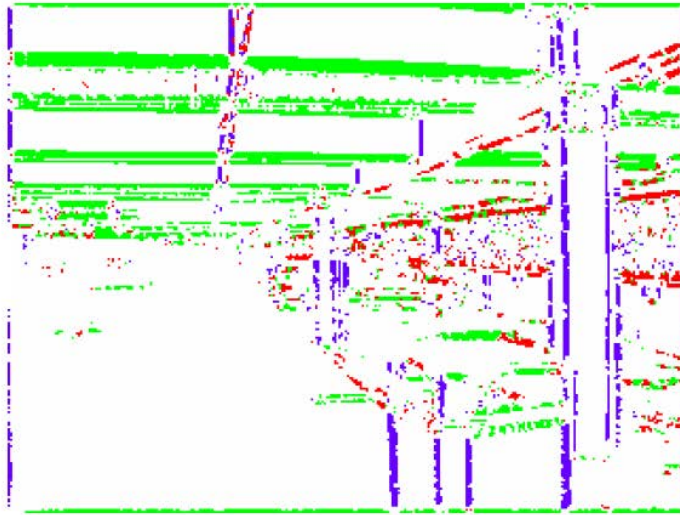
Input image:



$\text{Log}(P_{\text{on}}/P_{\text{off}})$



Outliers detected

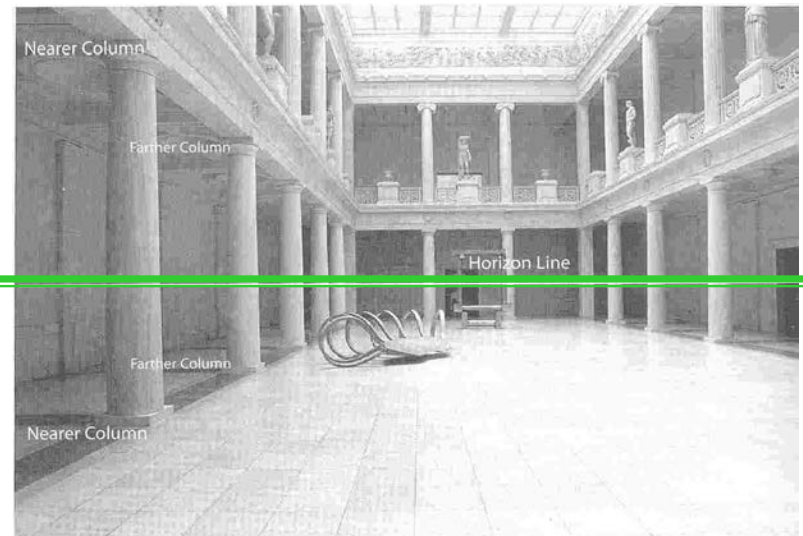


x lines in **red**
y lines in **green**
z lines in **blue**

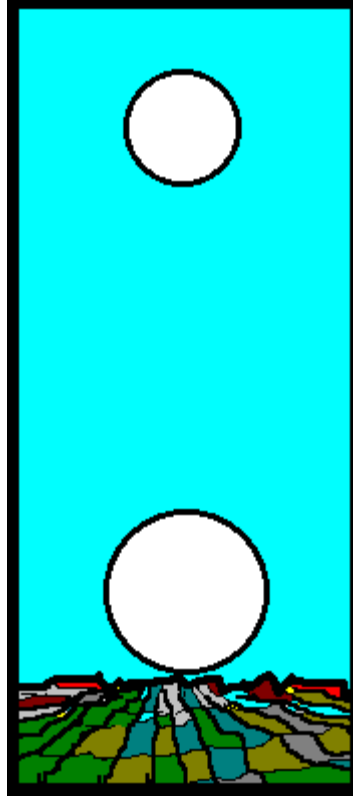
The importance of the horizon line

Distance from the horizon line

- Based on the tendency of objects to appear nearer the horizon line with greater distance to the horizon.
- Objects approach the horizon line with greater distance from the viewer. The base of a nearer column will appear lower against its background floor and further from the horizon line. Conversely, the base of a more distant column will appear higher against the same floor, and thus nearer to the horizon line.



Moon illusion

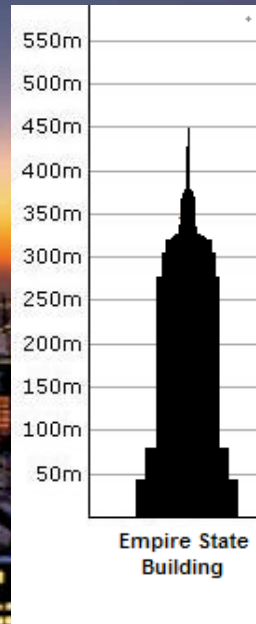


Relative height

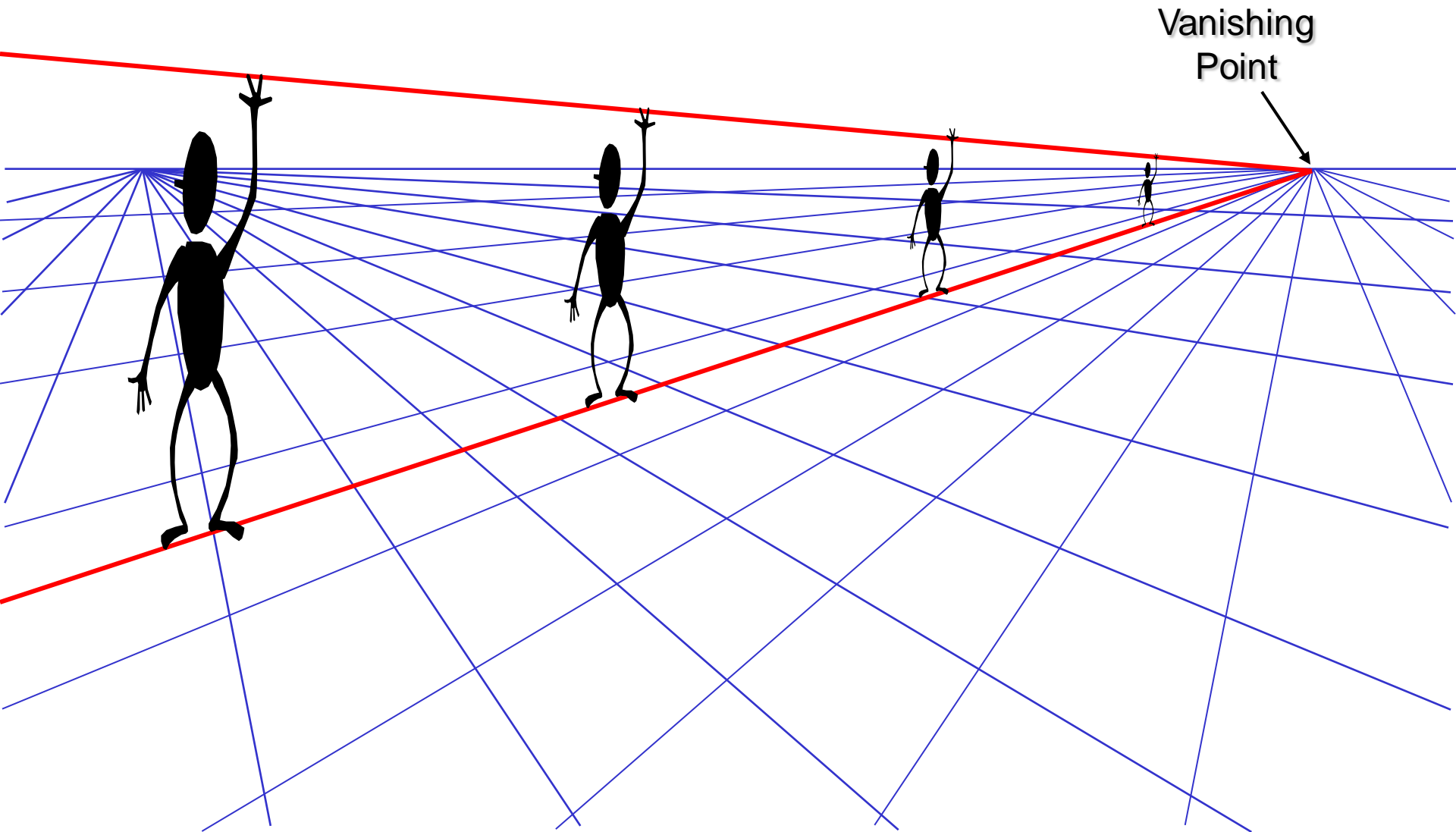
The object closer to the horizon is perceived as farther away, and the object further from the horizon is perceived as closer

If you know camera parameters: height of the camera, then we know real depth

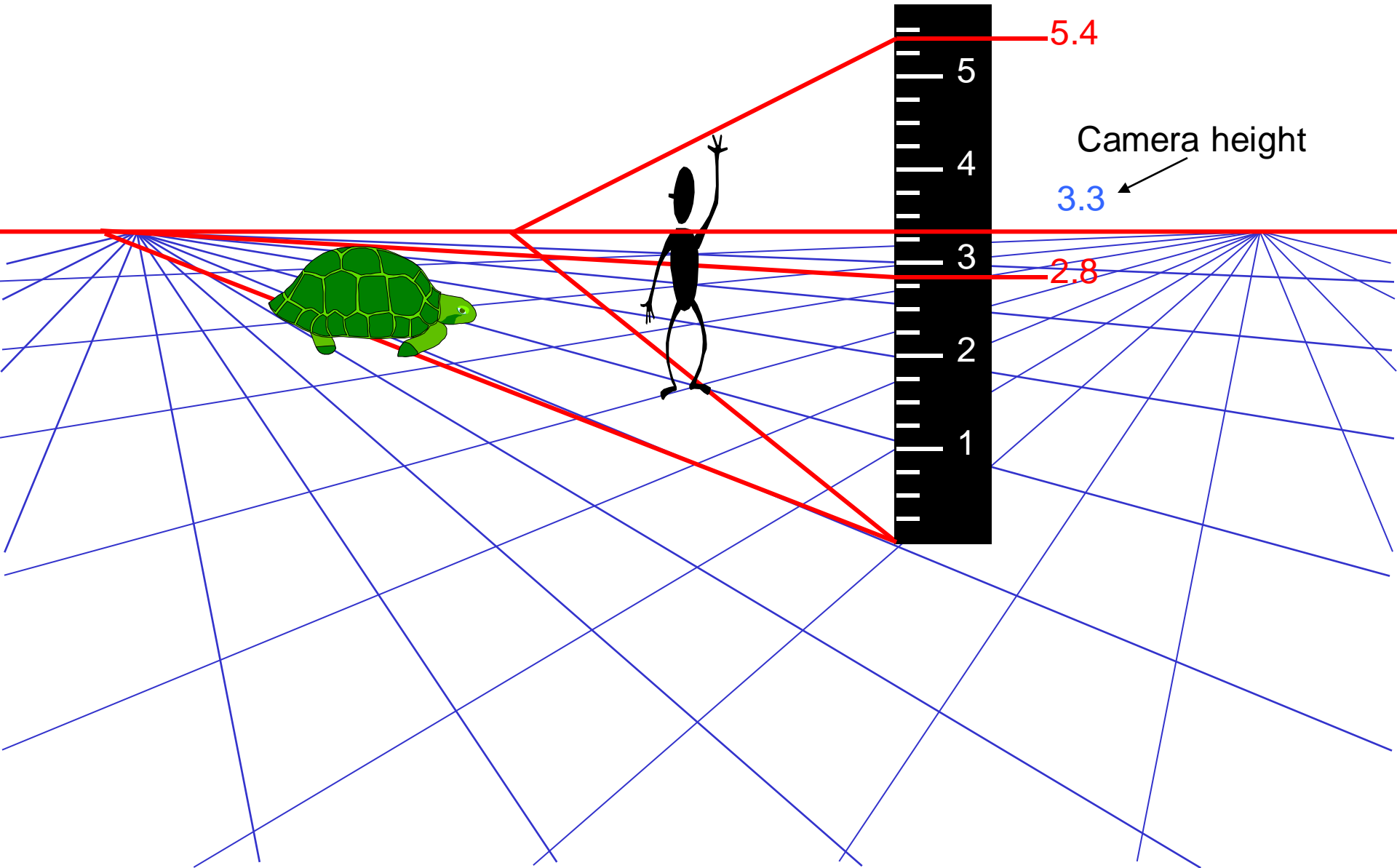
At which elevation has been taken this picture?



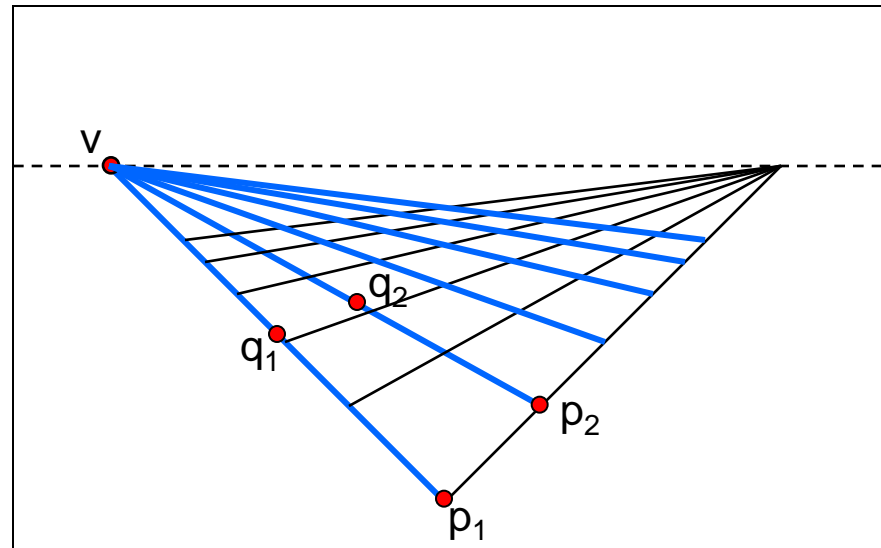
Comparing heights



Measuring height



Computing vanishing points (from lines)



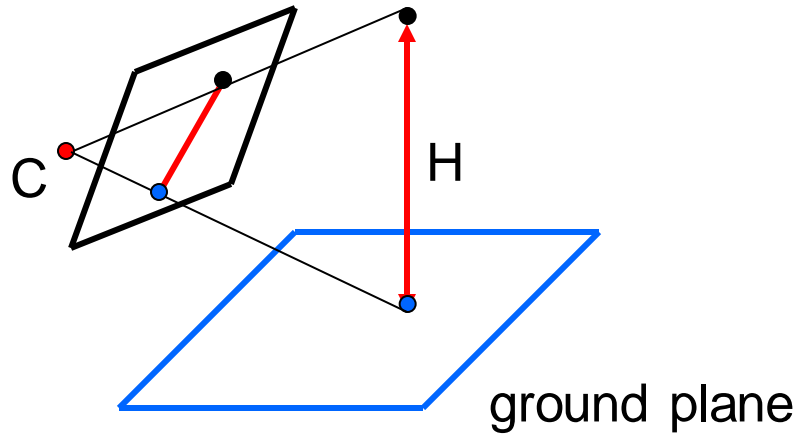
Intersect p_1q_1 with p_2q_2

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

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>

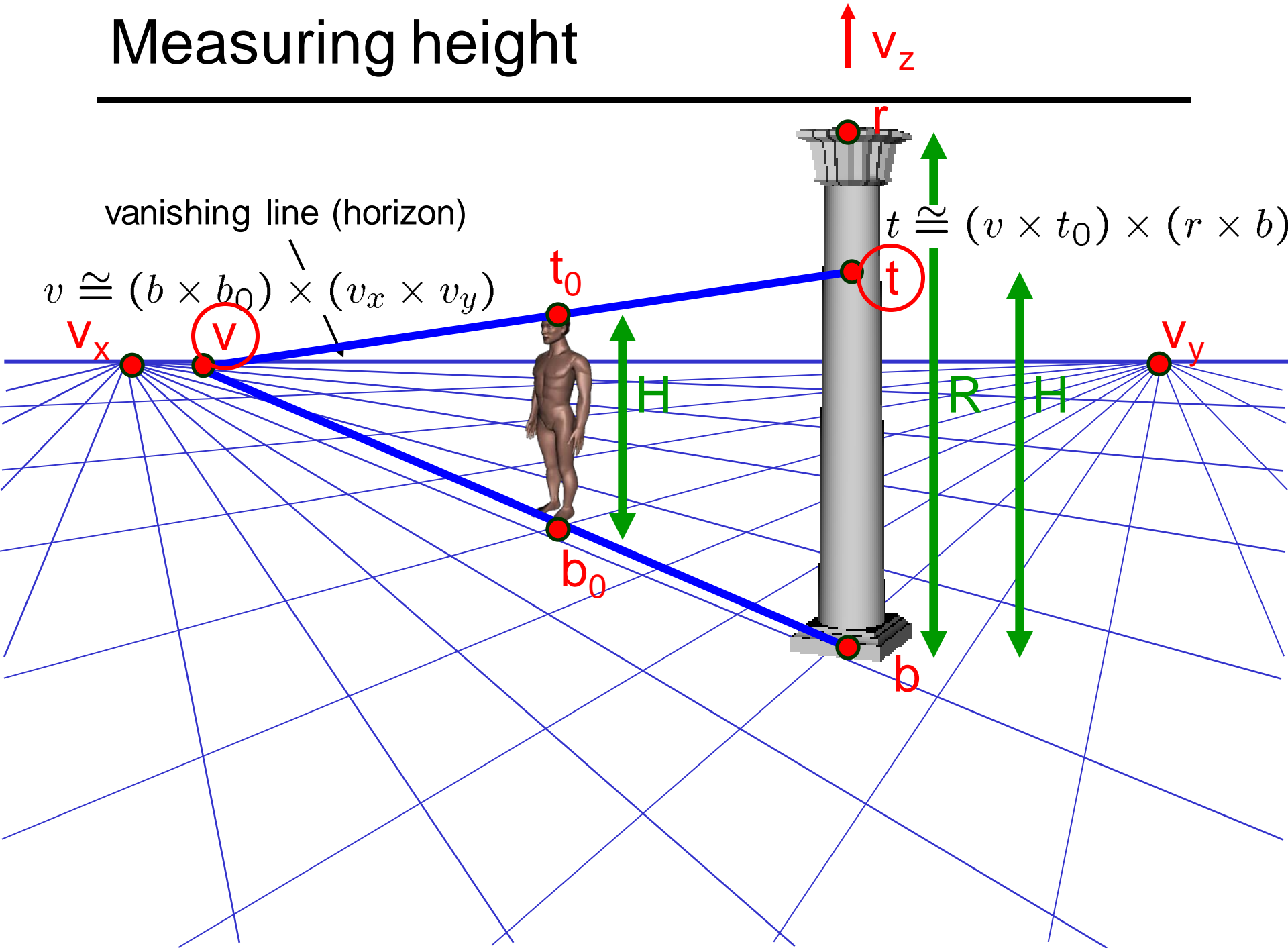
Measuring height without a ruler



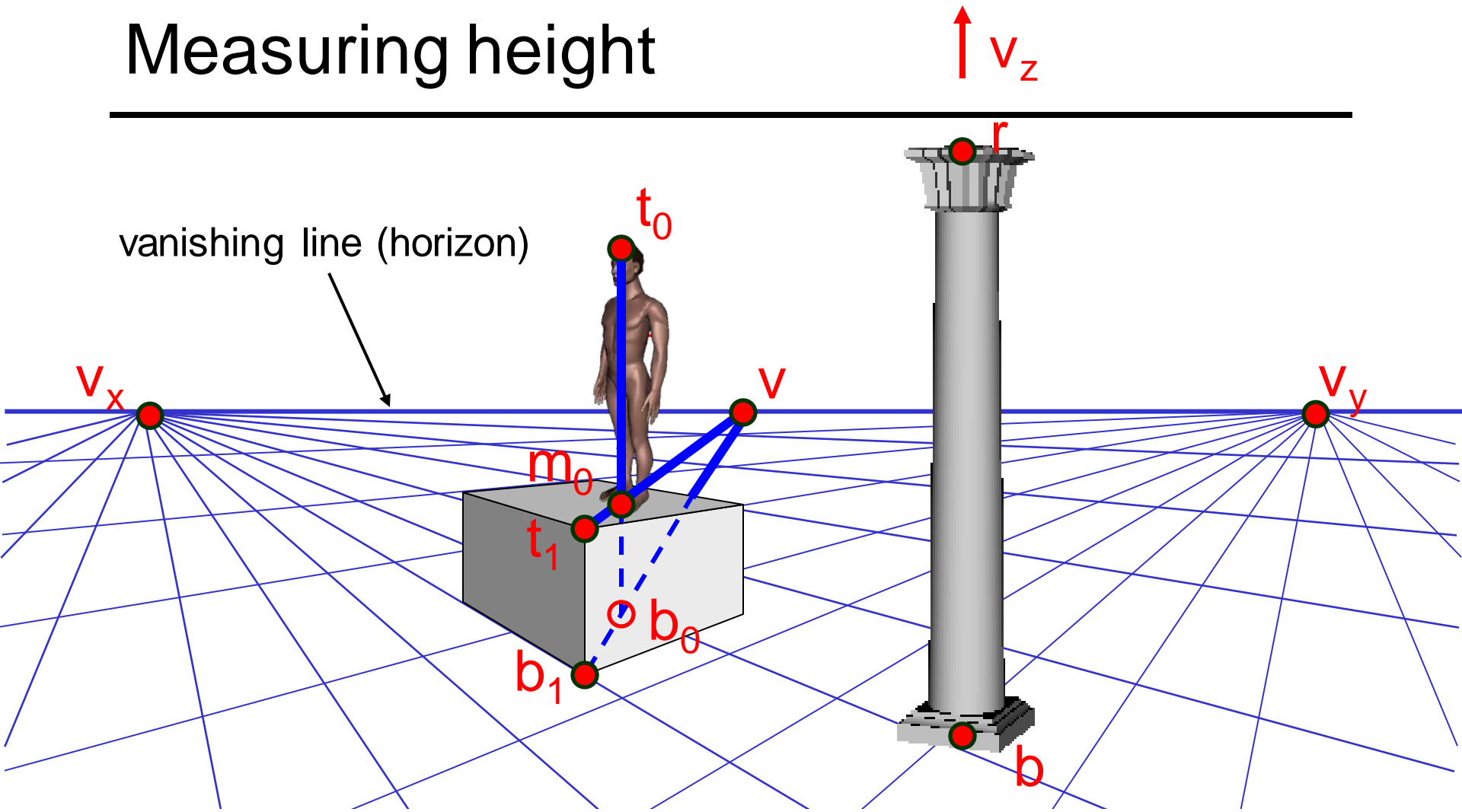
Compute H from image measurements

- Need more than vanishing points to do this

Measuring height



Measuring height



What if the point on the ground plane b_0 is not known?

- Here the guy is standing on the box
- Use one side of the box to help find b_0 as shown above

What if v_z is not infinity?

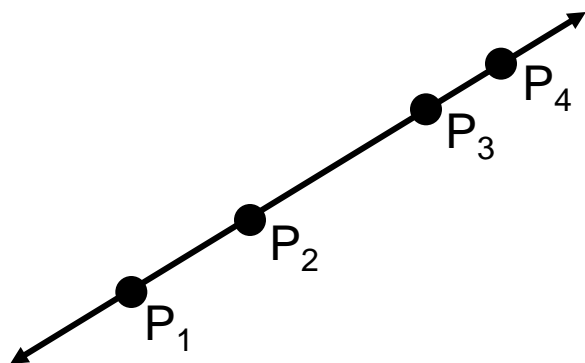


The cross ratio

A Projective Invariant

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

The cross-ratio of 4 collinear points



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

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

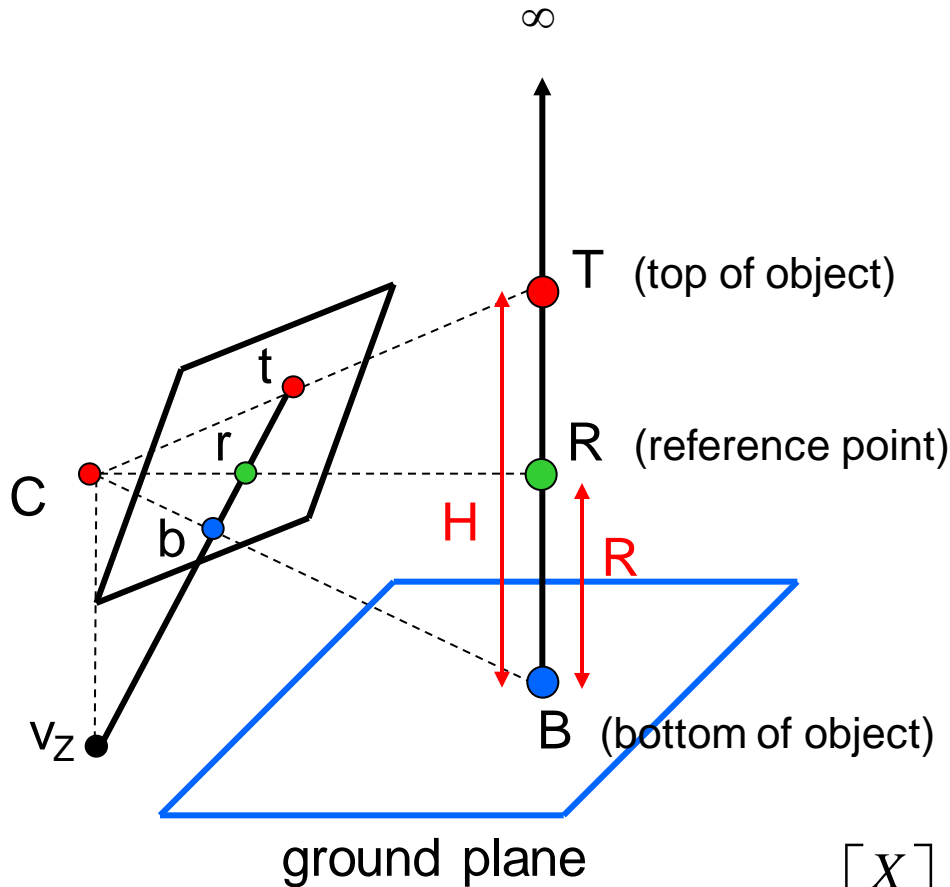
Can permute the point ordering

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

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

This is the fundamental invariant of projective geometry

Measuring height



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

scene cross ratio

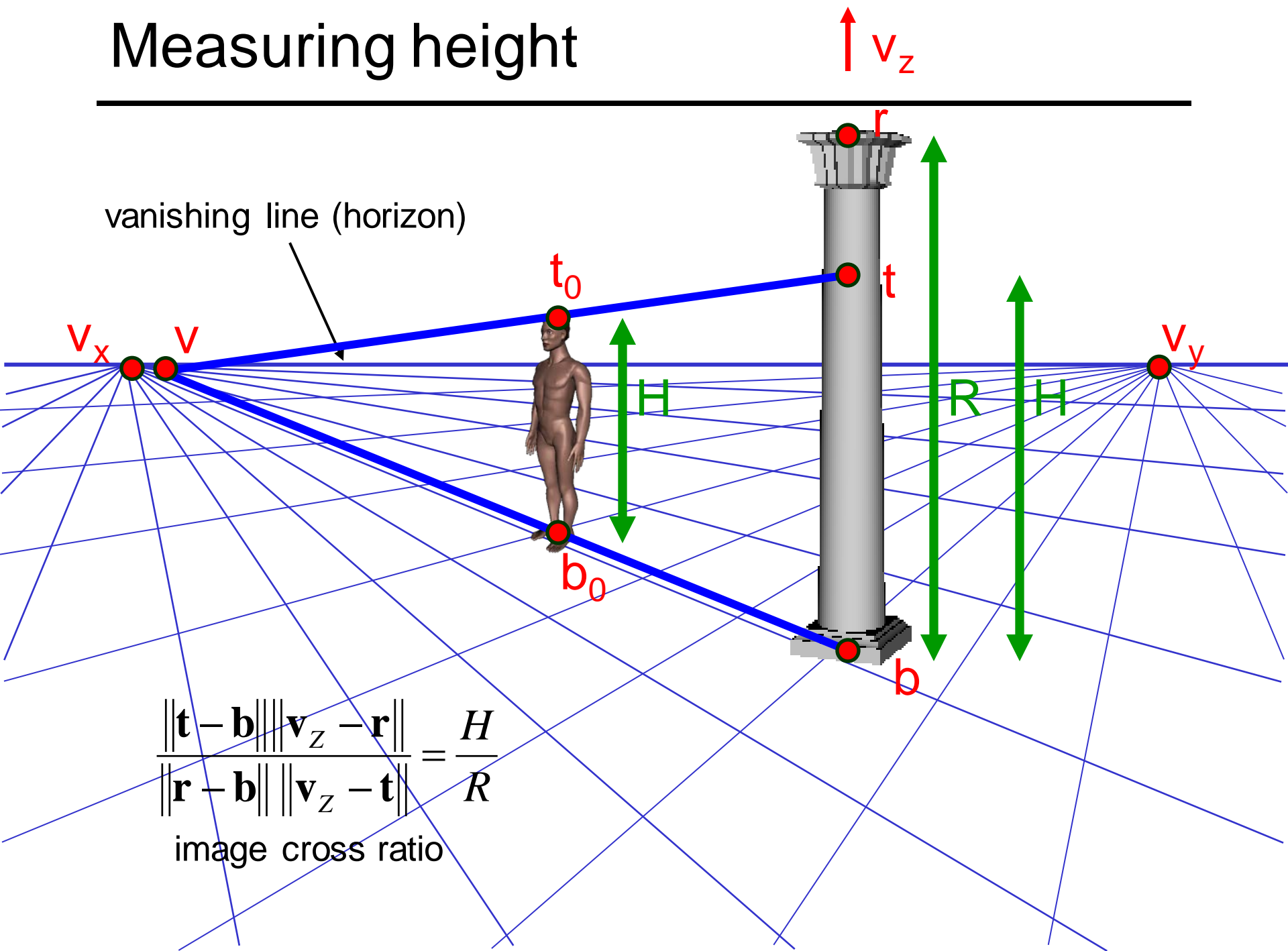
$$\frac{\|t - b\| \|v_z - r\|}{\|r - b\| \|v_z - t\|} = \frac{H}{R}$$

image cross ratio

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

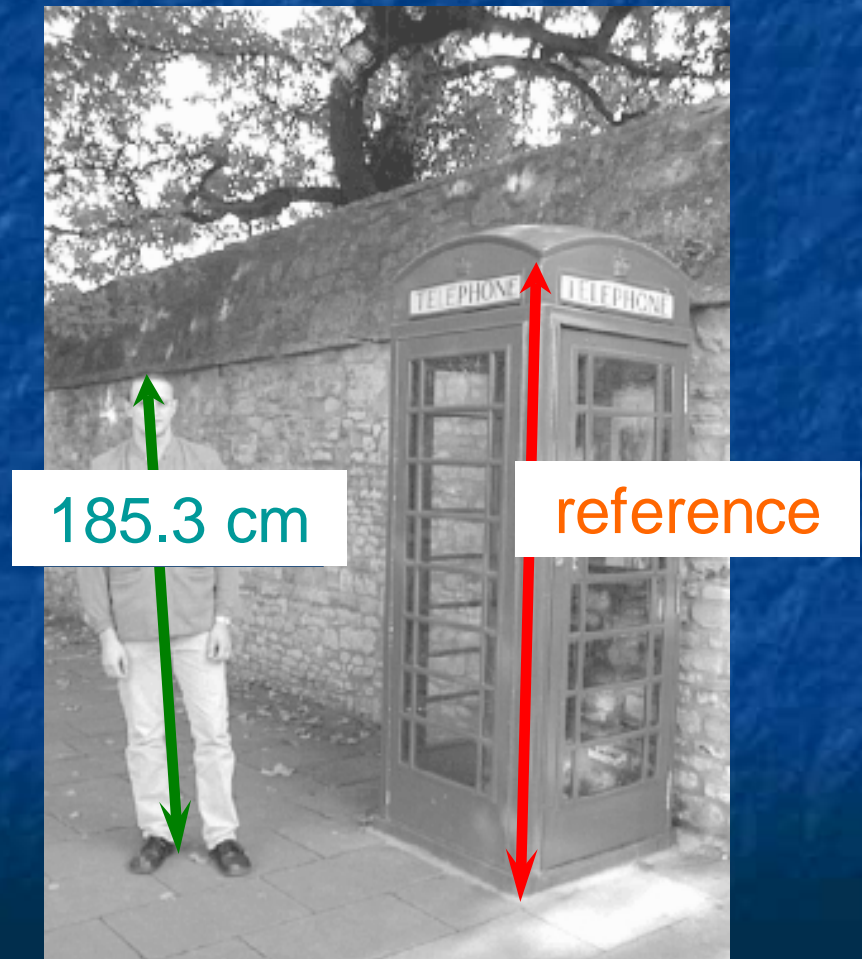
image points as $\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Measuring height



Measuring heights in real photos

Problem: How tall is this person?



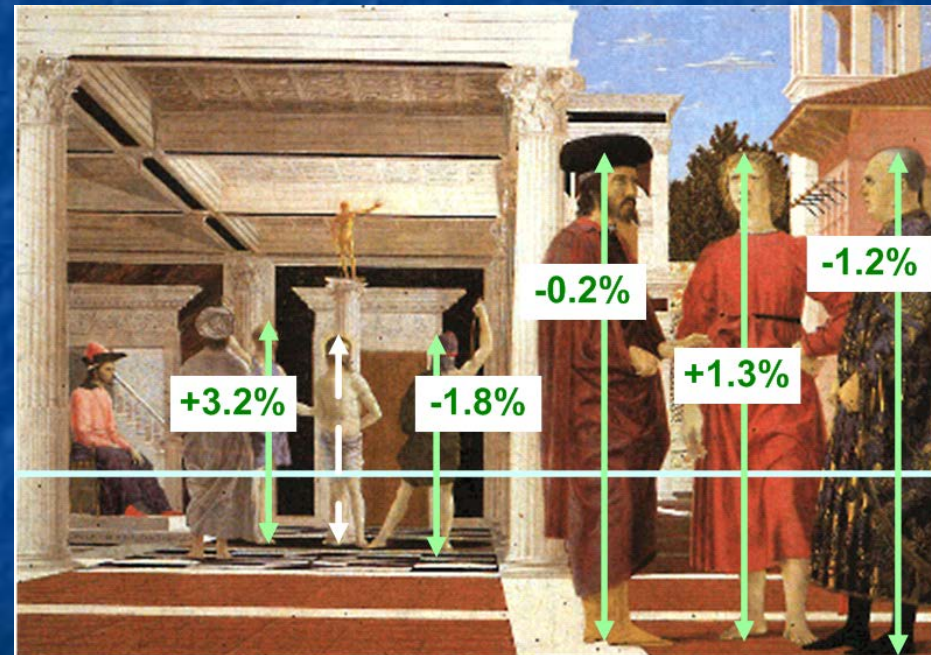
Assessing geometric accuracy

Problem:

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



Piero della Francesca,
Flagellazione di Cristo,
c.1460, Urbino

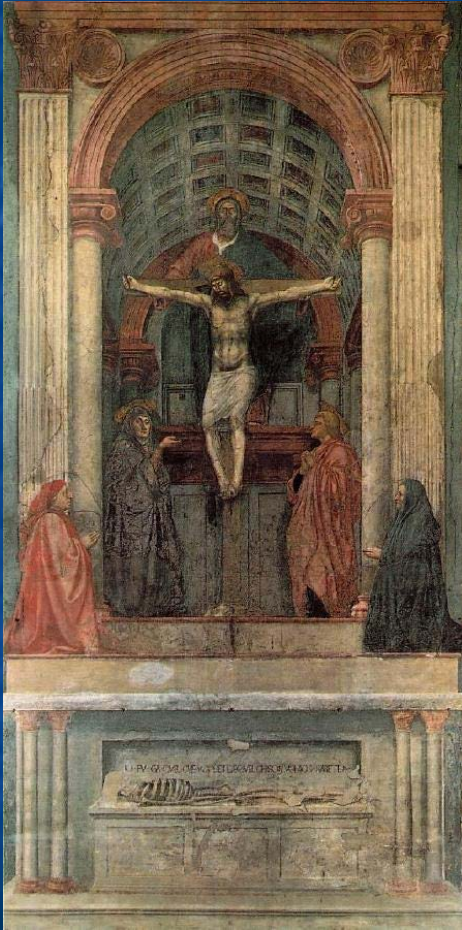


Measuring relative heights

Single-View Metrology

Complete 3D reconstructions from
single views

Example: The Virtual Trinity



Masaccio, *Trinita'*,
1426, Florence

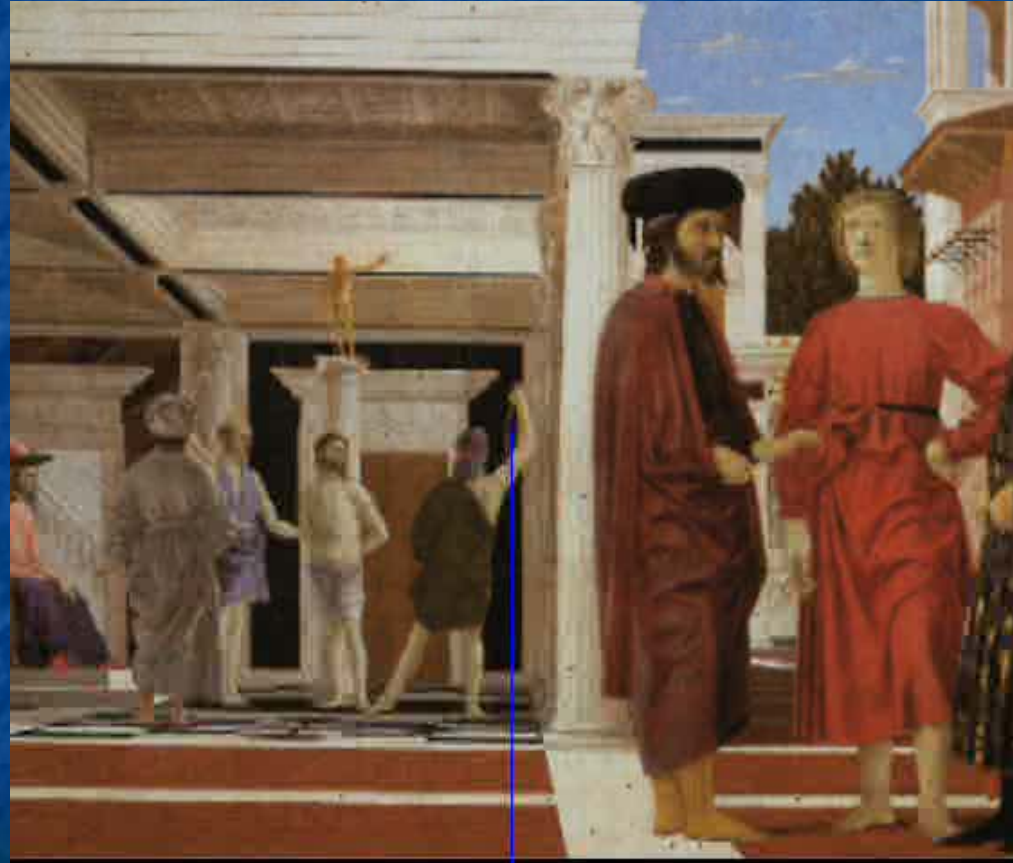


Complete 3D reconstruction

Example: The Virtual Flagellation



Piero della Francesca,
Flagellazione di Cristo,
c.1460, Urbino



Complete 3D reconstruction

Example: The Virtual St. Jerome



Henry V Steenwick,
St. Jerome in His Study,
1630, The Netherlands



Complete 3D reconstruction

Example: The Virtual Music Lesson



J. Vermeer,
The Music Lesson,
1665, London



Complete 3D reconstruction

Example: A Virtual Museum @ Microsoft

A dive into the paintings third dimension



The museum

Diving into the paintings



The Trinity
Masaccio



Flagellation
P. della Francesca



St Jerome
H. Steinwick



The Music Lesson
J. Vermeer

References

Antonio Criminisi

**Accurate Visual Reconstruction
from Single and Multiple
Uncalibrated Images**

(Springer-Verlag)

ISBN: 1-85233-468-1

look for it on amazon.com !

www.research.microsoft.com/~antcrim



Texture Gradient

Texture Gradient

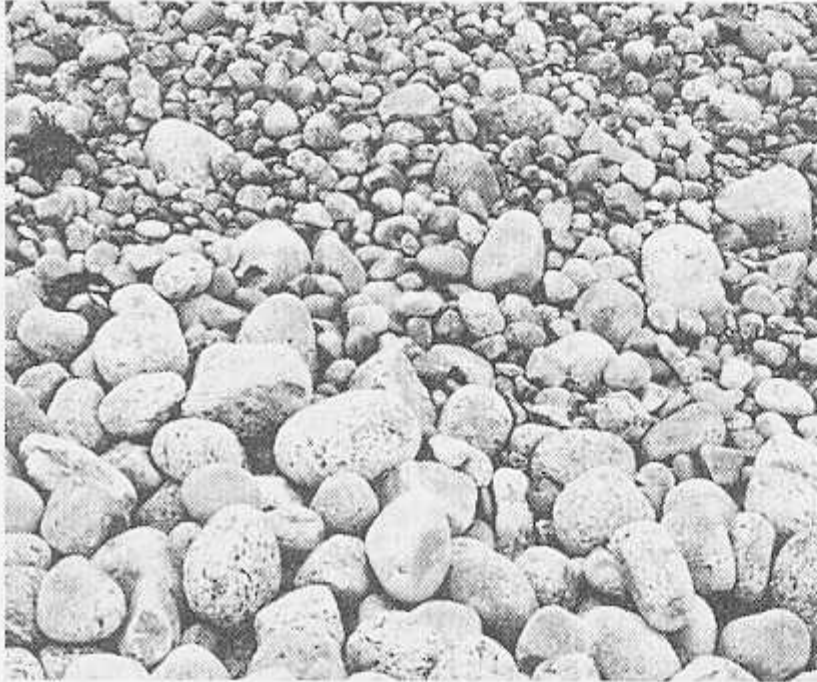


FIGURE 8.27

Texture gradients provide information about depth. (Frank Siteman/Stock, Boston.)

© Frank Siteman/Stock Boston

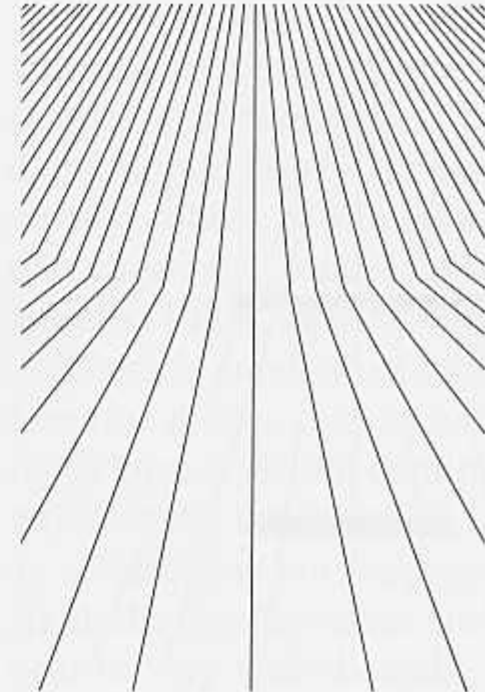


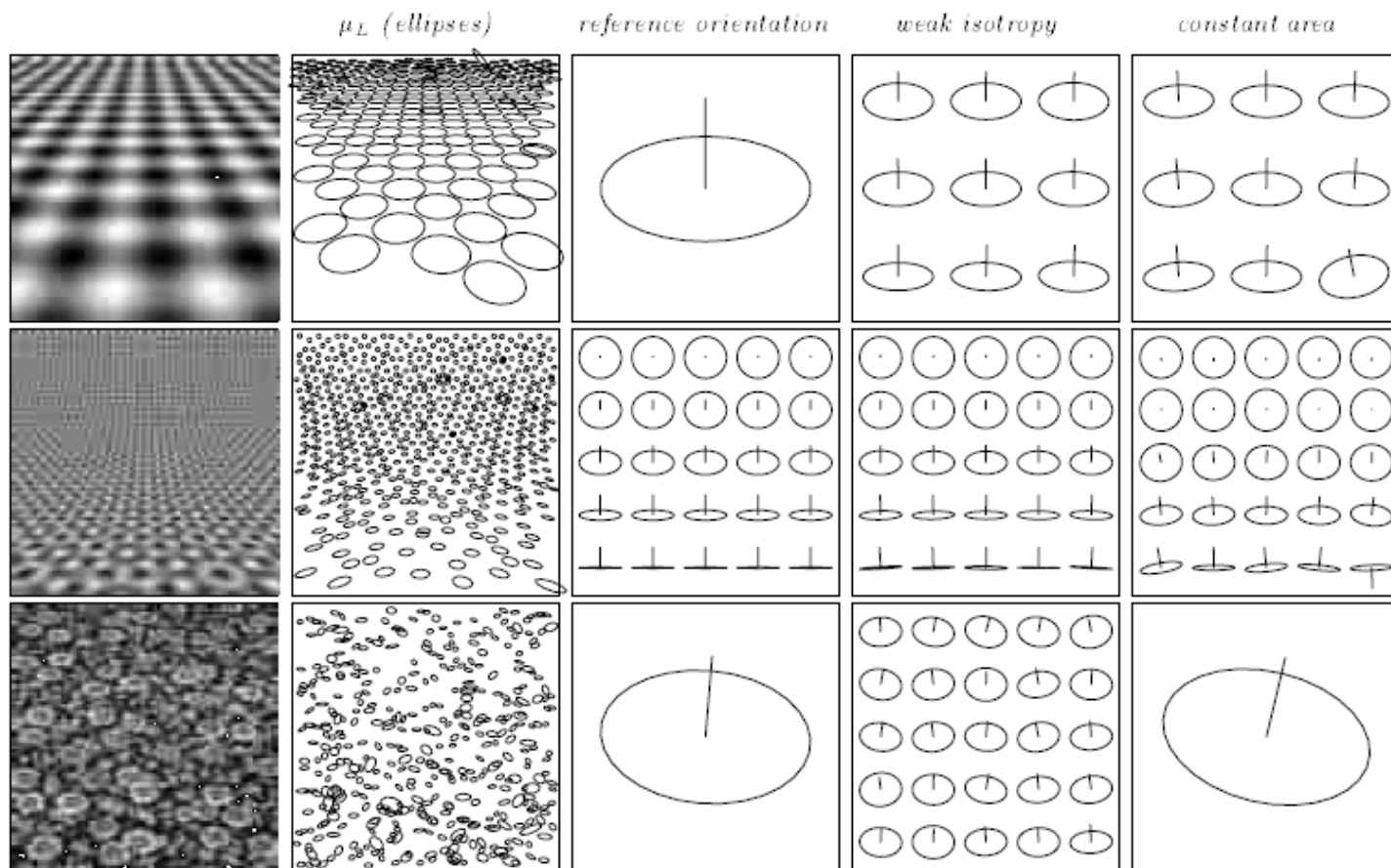
FIGURE 8.28

Texture discontinuity signals the presence of a corner.

A Witkin. Recovering Surface Shape and Orientation from Texture (1981)



Texture Gradient



Texture Gradient

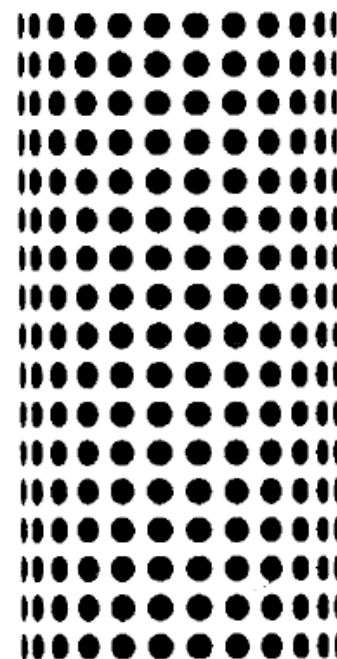
- Filter outputs
- Textons

Shape from Texture Using Local Spectral Moments

Boaz J. Super, *Member, IEEE*, and Alan C. Bovik, *Senior Member, IEEE*

Abstract—We present a non-feature-based solution to the problem of computing the shape of curved surfaces from texture information. First, the use of local spatial-frequency spectra and their moments to describe texture is discussed and motivated. A new, more accurate method for measuring the local spatial-frequency moments of an image texture using Gabor elementary functions and their derivatives is presented. Also described is a technique for separating shading from texture information, which makes the shape-from-texture algorithm robust to the shading effects found in real imagery. Second, a detailed model for the projection of local spectra and spectral moments of any surface reflectance patterns (not just textures) is developed. Third, the conditions under which the projection model can be solved for the orientation of the surface at each point are explored. Unlike earlier non-feature-based, curved surface shape-from-texture approaches, the assumption that the surface texture is isotropic is not required; surface texture homogeneity can be assumed instead. The algorithm's ability to operate on anisotropic and non-deterministic textures, and on both smooth- and rough-textured surfaces, is demonstrated.

Index Items—Shape from texture, shape recovery, surface orientation, moments, wavelet, spatial frequency, Gabor functions, texture, projection.



PLANAR SURFACE ORIENTATION FROM TEXTURE SPATIAL FREQUENCIES

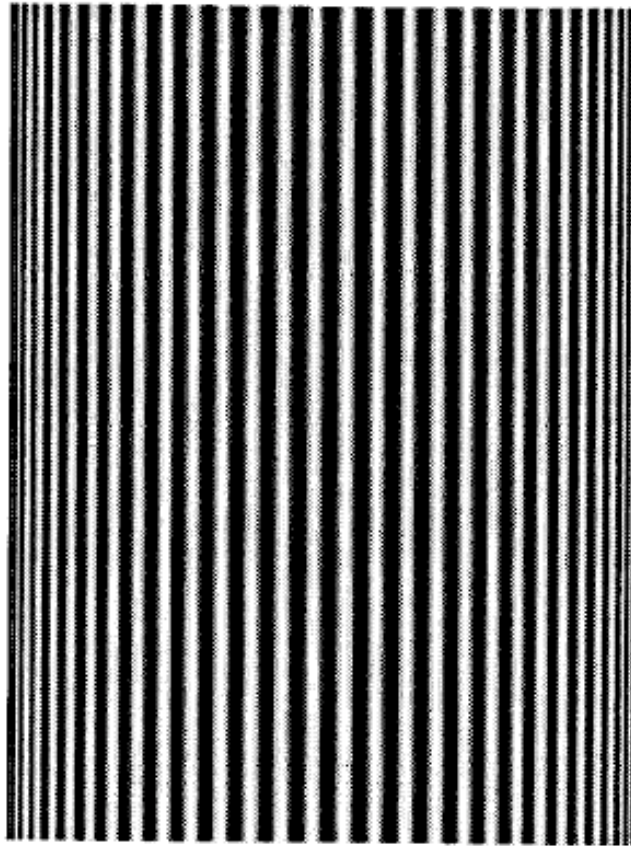
BOAZ J. SUPER*† and ALAN C. BOVIK‡

Assumptions:

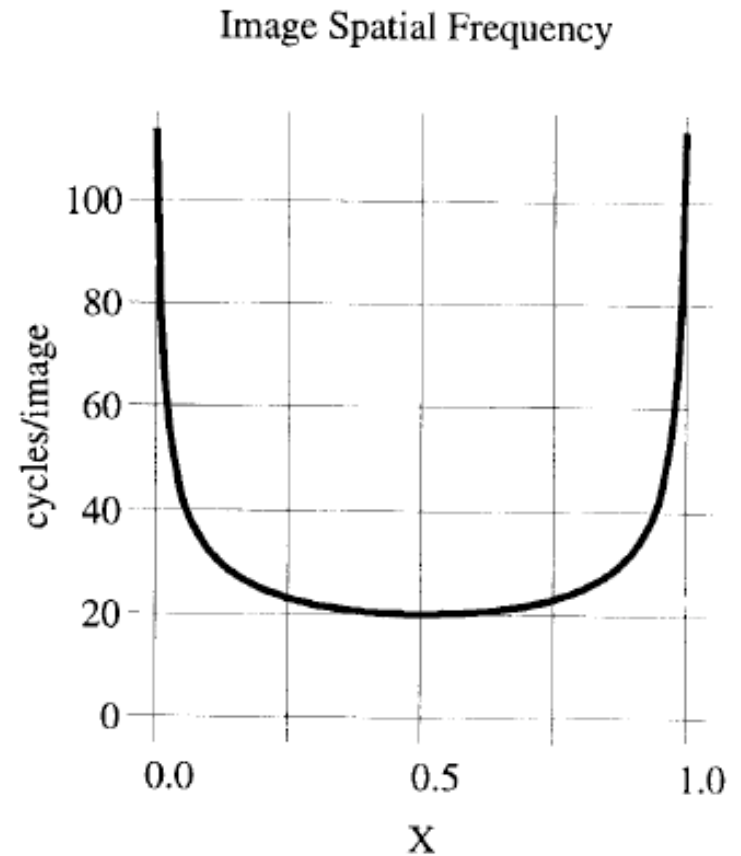
- Smooth closed surface
- Homogeneous texture
- (sometimes, isotropic texture)

Texture description

Use filter outputs to measure local spatial frequency.



(a)



(b)

Fig. 2. (a) Cylinder with sinusoidal grating texture. (b) Horizontal component of image spatial frequency on center cross-section of (a).

Texture projection

Assume orthographic projection.

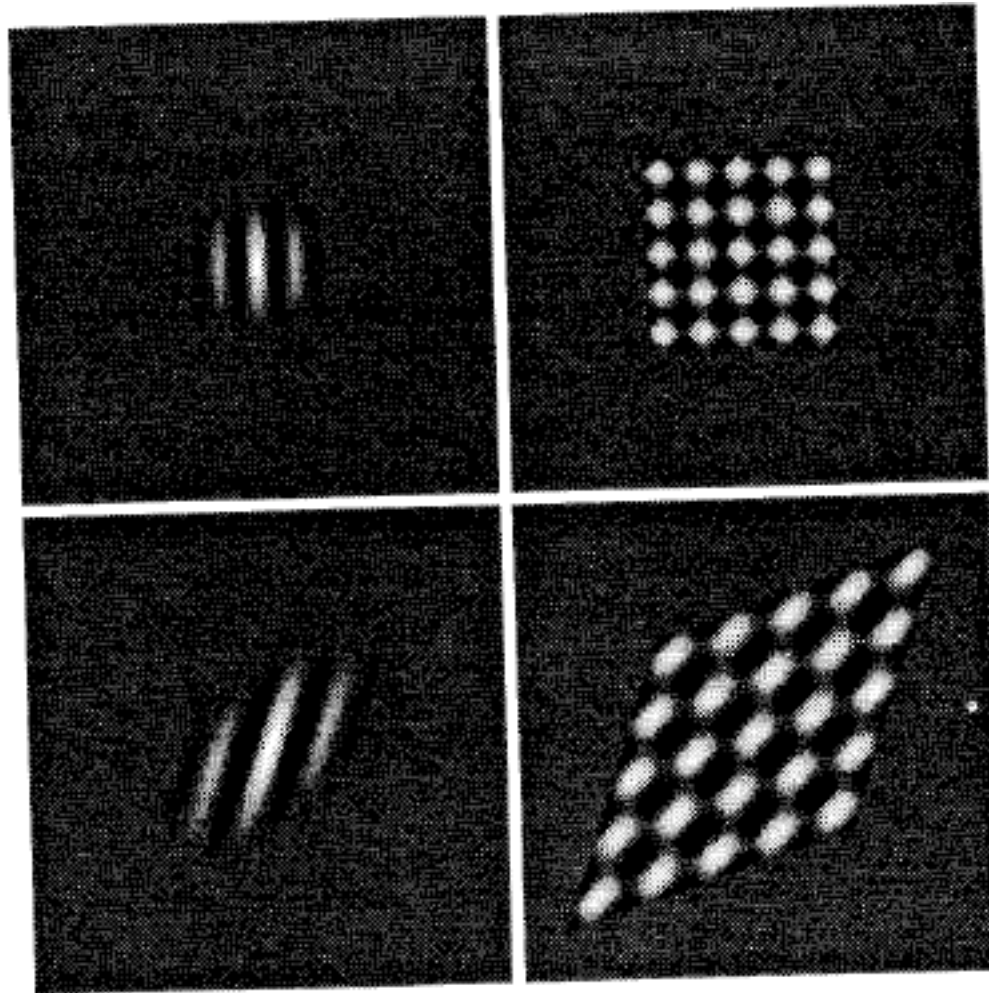
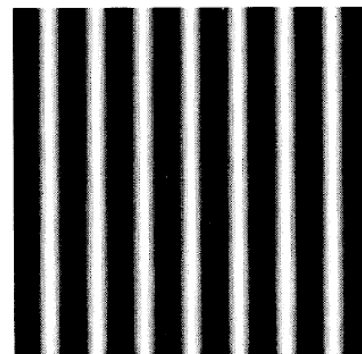
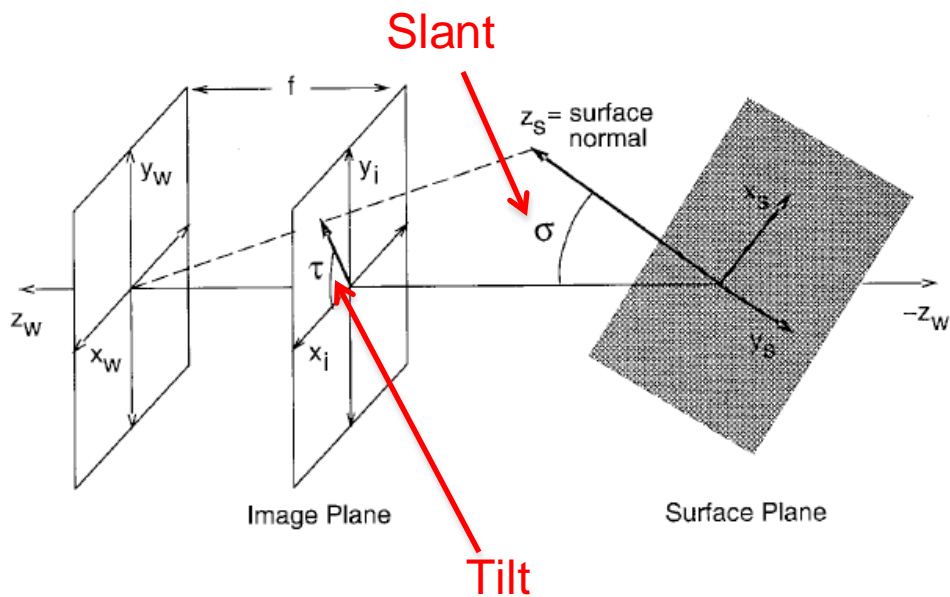
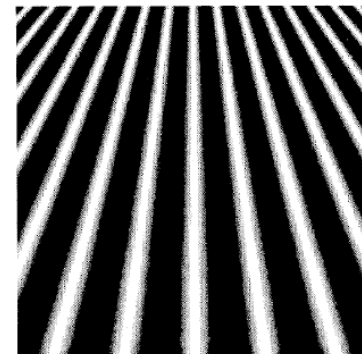


Fig. 5. Top row: real part of Gabor filter with radial frequency of 12 cycles/image, and a texture patch. Bottom row: back-projections of Gabor filter and texture patch onto a plane with orientation $(\sigma, \tau) = (60^\circ, 45^\circ)$.

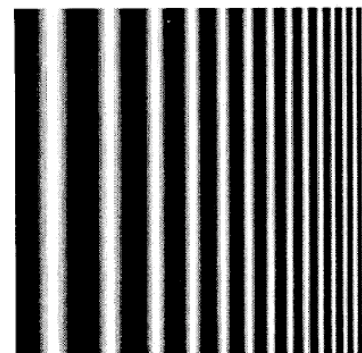
Slant and tilt



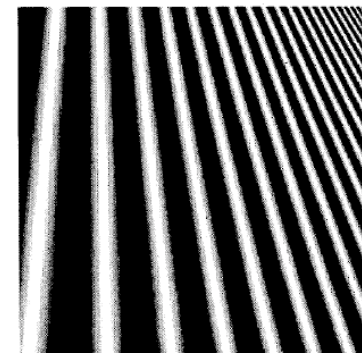
$\sigma = 0^\circ$



$\sigma = 45^\circ, \tau = 90^\circ$



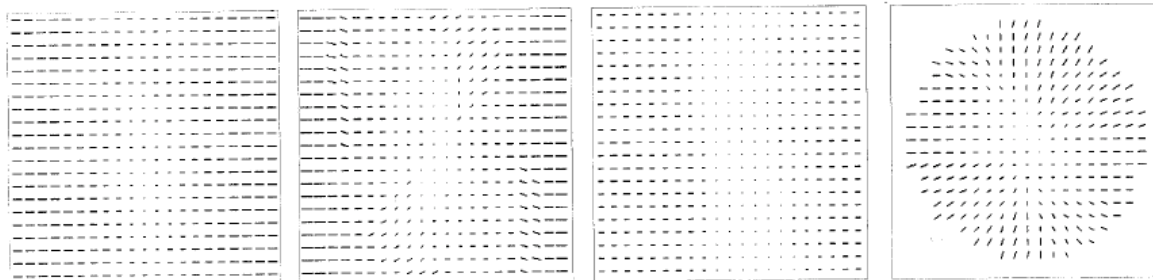
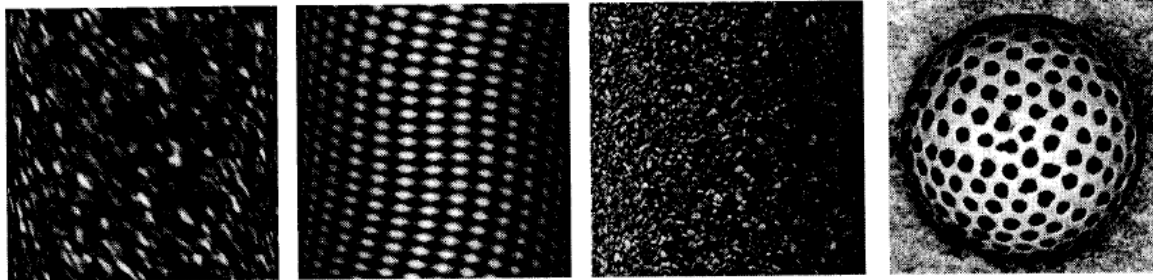
$\sigma = 45^\circ, \tau = 0^\circ$



$\sigma = 45^\circ, \tau = 45^\circ$

Box 1. Summary of algorithm

1. Convolve the image with Gabor functions and their partial derivatives, and smooth the filter output amplitudes (to reduce noise) by convolving them with a Gaussian.
2. Select the Gabor filter h_k with the largest amplitude output at each point.
3. Compute the (signed) instantaneous frequency $\mathbf{u}_i(\mathbf{x}_i)$ at each point using equation (6).
4. Sample (σ, τ) -space, backprojecting $\mathbf{u}_i(\mathbf{x}_i)$ to compute $\mathbf{u}_s(\mathbf{x}_s)$ using equation (20). Compute the variance $V_{\sigma, \tau}$ of $\mathbf{u}_s(\mathbf{x}_s)$. Coarse-to-fine sampling in multiple stages may be used.
5. Output the values of (σ, τ) for which $V_{\sigma, \tau}$ is a minimum.



(a)

(b)

(c)

(d)

Recovering shape and irradiance maps from rich dense texture fields

Anthony Lobay and D.A. Forsyth

CVPR 04

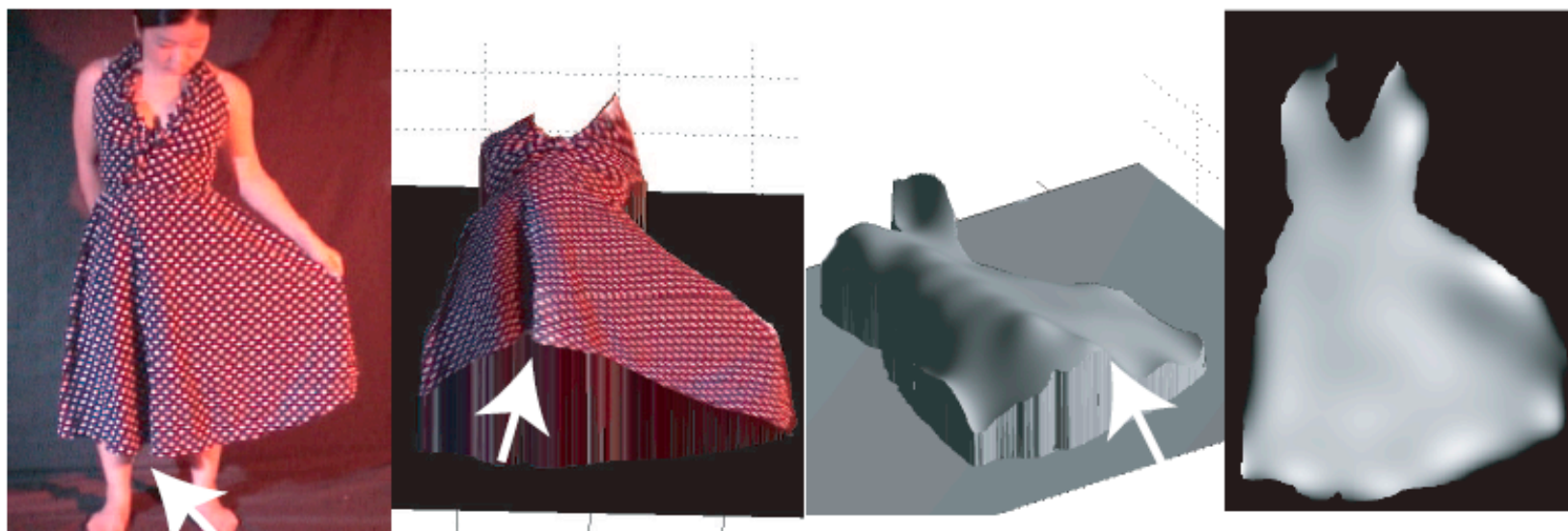


Figure 3: *On the left, a view of a model in a spotted dress. In the center left, a textured view of the reconstruction obtained using our method. This reconstruction used 1200 texton instances, in 8 clusters. Note the relatively fine detail that was obtained by the reconstruction, including the two main folds in the skirt (indicated with arrows). Typically, rendering texture on top of the view produces a better looking surface, so we show the surface without texturing on the center right; arrows indicate the reconstructed folds in the geometry. Notice that the fold in the skirt is well represented. The smoothing term is generally good at resolving normal ambiguities, but patches of surface that are not well connected to the main body can be subjected to a concave-convex ambiguity, as has happened to part of the skirt's bodice here. On the right, the irradiance map estimated using our method.*

Texture description

Non-occluded textons, and approximated as flat.



The two pieces of the solution

If we knew the transformations

- We can find the textons
- We can find the local intensity contrast

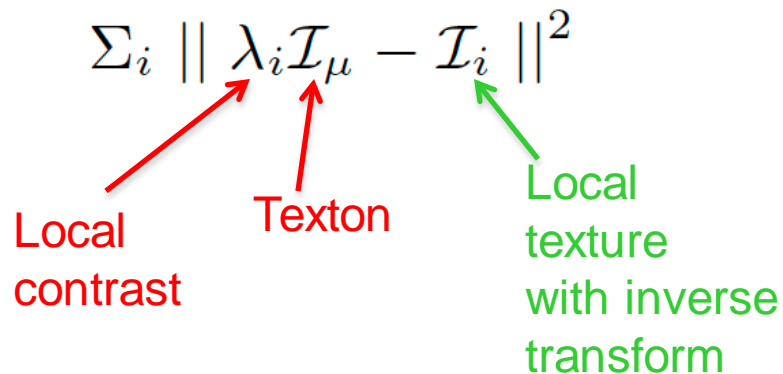
By minimization of:

$$\sum_i \|\lambda_i \mathcal{I}_\mu - \mathcal{I}_i\|^2$$

Local contrast

Texton

Local texture with inverse transform

A diagram illustrating the minimization equation. The equation is $\sum_i \|\lambda_i \mathcal{I}_\mu - \mathcal{I}_i\|^2$. Three arrows point from labels below to terms in the equation: a red arrow from 'Local contrast' to λ_i , a red arrow from 'Texton' to \mathcal{I}_μ , and a green arrow from 'Local texture with inverse transform' to \mathcal{I}_i .

If we knew the texton and contrast

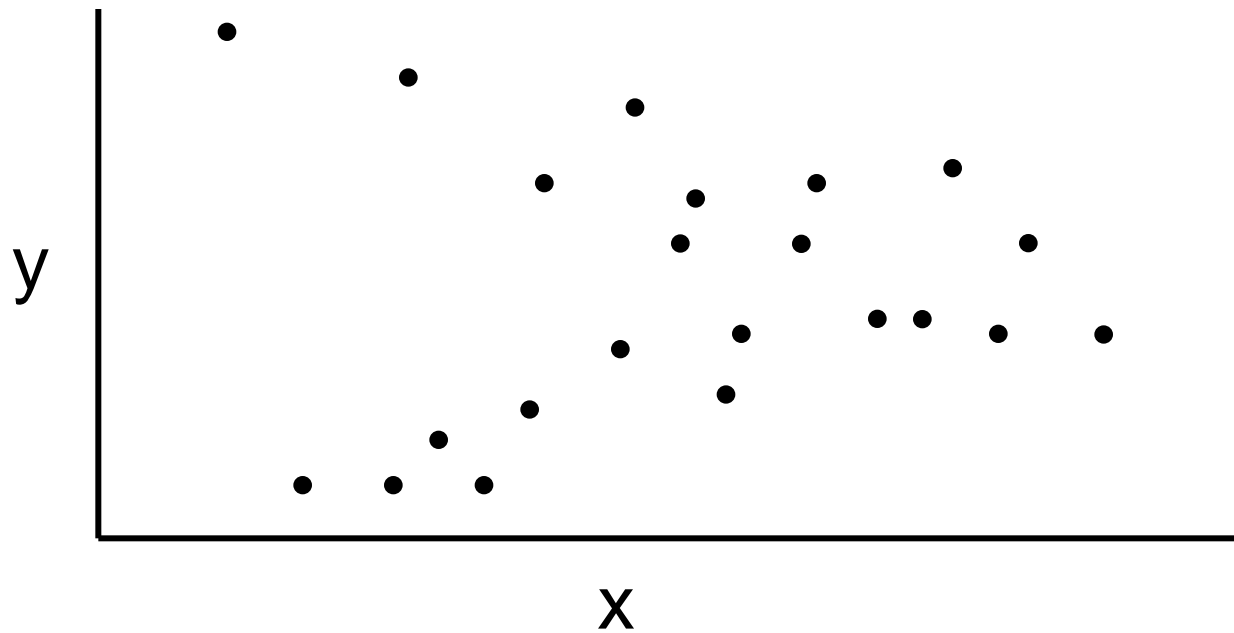
- Recover the transformation by transforming the texton to match each local patch.

Expectation Maximization (EM): a solution to chicken-and-egg problems

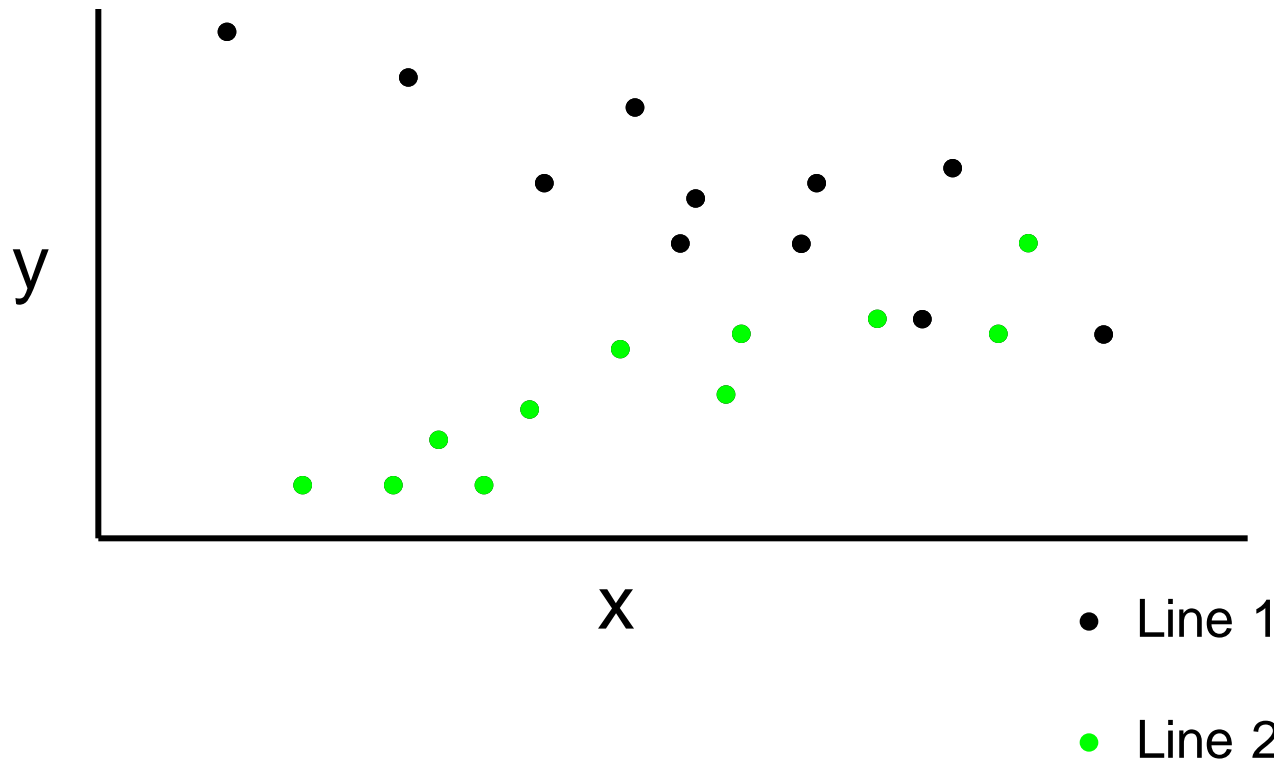


Model fitting example

Fitting two lines to observed data



Fitting two lines: on the one hand...



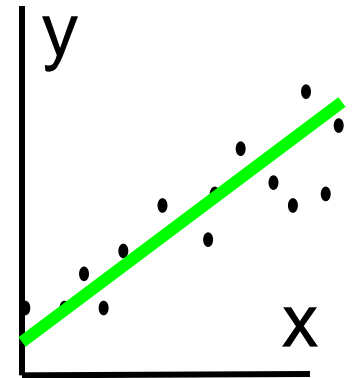
If we knew which points went with which lines, we'd be back at the single line-fitting problem, twice.

Maximum likelihood estimation for the slope of a single line

data: $(X_n, Y_n), n = 1 \dots, N$

model: $Y = aX + w$

where $w \sim N(\mu = 0, \sigma = 1)$.



Data likelihood for point n:

$$P(X_n, Y_n | a) = c \exp[-(Y_n - aX_n)^2 / 2]$$

Maximum likelihood estimate:

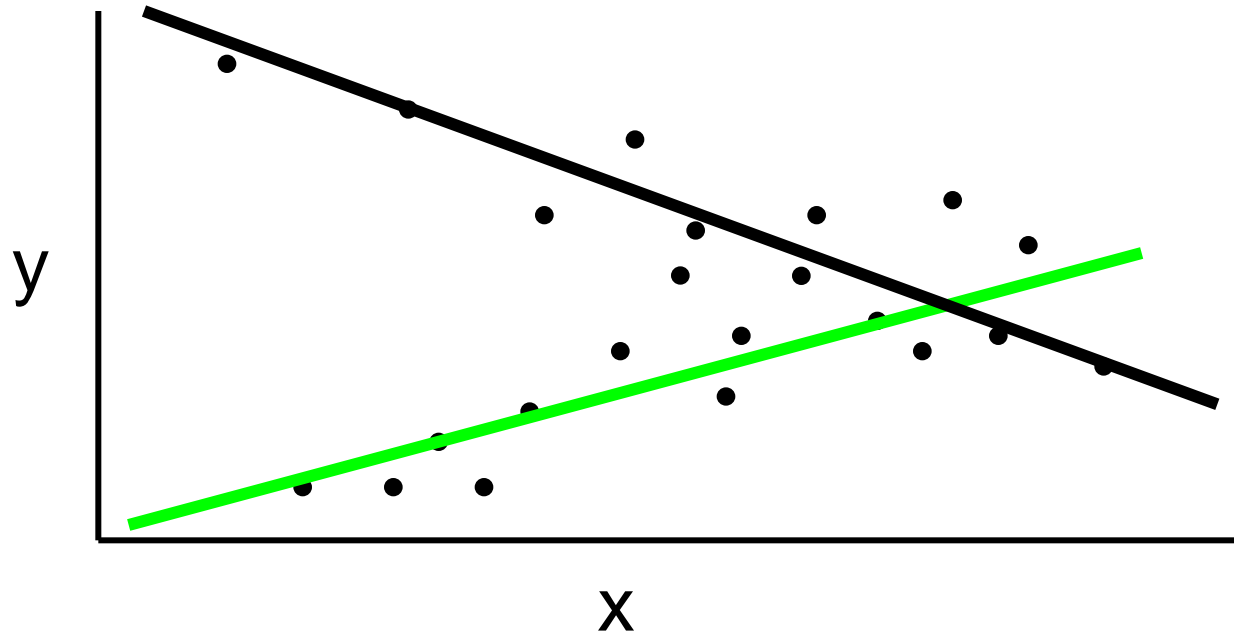
$$\hat{a} = \arg \max_a p(Y_1, \dots, Y_n | a) = \arg \max_a \sum_n -d(Y_n; a)^2 / 2$$

$$\text{where } d(Y_n; a) = |Y_n - aX_n|$$

gives regression formula

$$\hat{a} = \frac{\sum_n Y_n X_n}{\sum_n X_n^2}$$

Fitting two lines, on the other hand...



We could figure out the probability that any point came from either line if we just knew the two equations for the two lines.

MLE with hidden/latent variables: Expectation Maximisation

General problem:

$$y = (Y_1, \dots, Y_N); \quad \theta = (a_1, a_2); \quad z = (z_1, \dots, z_N)$$

data parameters hidden variables

For MLE, want to maximise the log likelihood

The sum over z inside the log gives a complicated expression for the ML solution.

$$\begin{aligned} \hat{\theta} &= \arg \max_{\theta} \log p(y|\theta) \\ &= \arg \max_{\theta} \log \sum_z p(y, z|\theta) \end{aligned}$$

Maximizing the log likelihood of the data

if you knew the z_n labels for each sample n :

$$\hat{\theta} = \operatorname{argmax}_{\theta} \sum_n \delta(z_n = 1) \log p(y_n | z_n = 1, \theta) + \delta(z_n = 2) \log p(y_n | z_n = 2, \theta)$$

Maximizing the log likelihood of the data

if you knew the z_n labels for each sample n :

$$\hat{\theta} = \operatorname{argmax}_{\theta} \sum_n \delta(z_n = 1) \log p(y_n | z_n = 1, \theta) + \delta(z_n = 2) \log p(y_n | z_n = 2, \theta)$$

In the EM algorithm, we replace those known labels with their expectation under the current algorithm parameters. So

$$E[\delta(z_n = i)] = p(z_n = i | y, \theta_{old})$$

Call that quantity $= \alpha_i(n)$

$$\propto p(y | z_n = i, \theta_{old}) \propto e^{-(y_n - a_i x_n)^2 / 2}$$

Maximizing gives

And then for the estimate of the line parameters, we have

$$\hat{\theta} = \operatorname{argmin}_{\theta} \sum_n \alpha_1(n)(y_n - a_1 x_n)^2 + \alpha_2(n)(y_n - a_2 x_n)^2$$

and maximising that gives

$$\hat{a}_i = \frac{\sum_n \alpha_i(n) y_n x_n}{\sum_n \alpha_i(n) x_n^2}$$

EM fitting to two lines

with

$$\alpha_i(n) \propto e^{-(y_n - a_i x_n)^2 / 2}$$

and

$$\alpha_1(n) + \alpha_2(n) = 1$$

“E-step”

↑
repeat

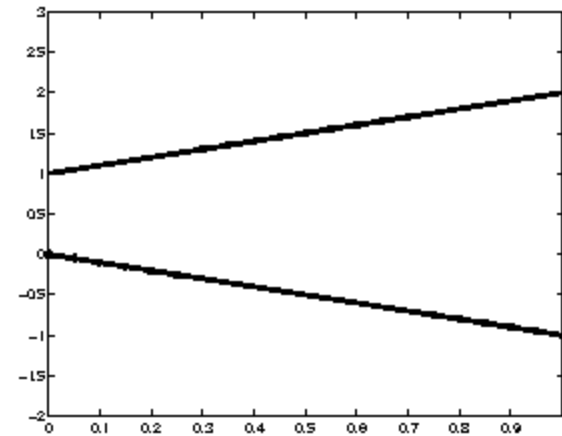
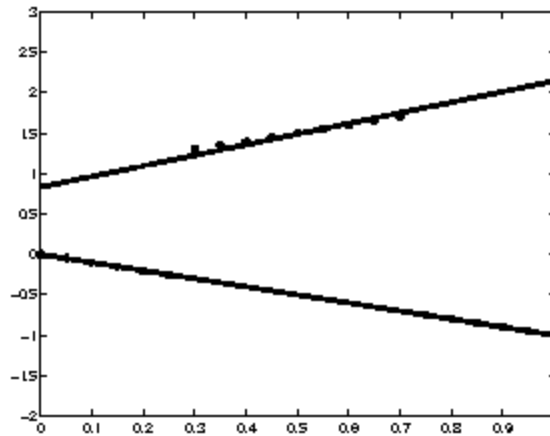
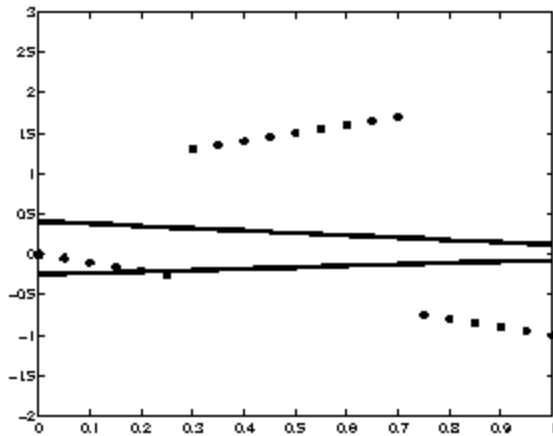
Regression becomes:

$$\hat{a}_i = \frac{\sum_n \alpha_i(n) y_n x_n}{\sum_n \alpha_i(n) x_n^2}$$

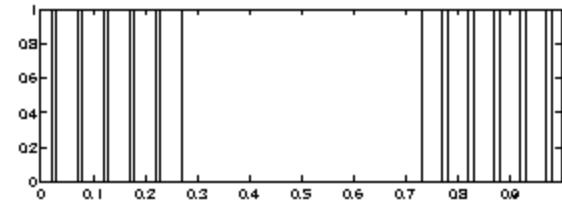
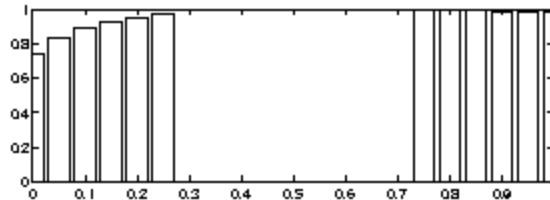
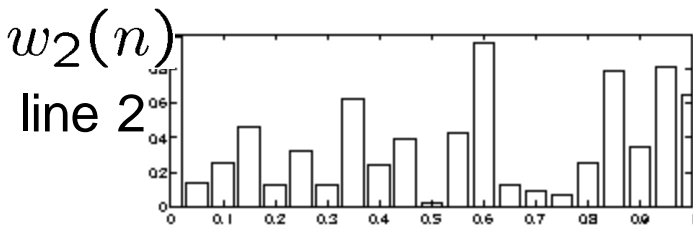
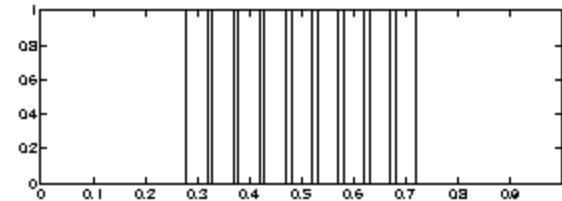
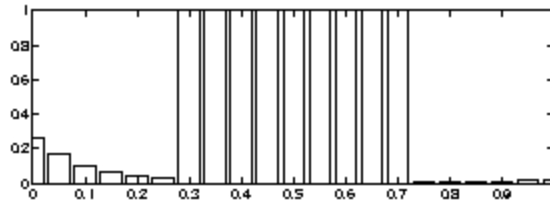
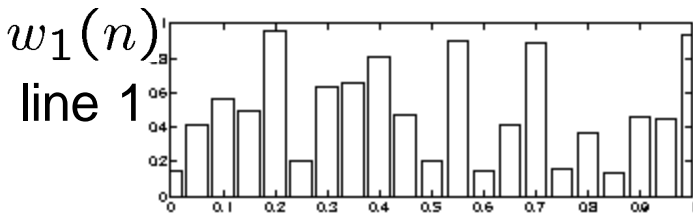
↓
“M-step”

Experiments: EM fitting to two lines

(from a tutorial by Yair Weiss, <http://www.cs.huji.ac.il/~yweiss/tutorials.html>)



Line weights



Iteration

1

2

74

3

EM

$$\frac{1}{2\sigma_{im}^2} \sum_i (\| \lambda_i \mathcal{I}_\mu - \mathcal{T}_i^{-1} \mathcal{I} \|^2 \delta_i) + \sum_i (1 - \delta_i) K + \frac{1}{2\sigma_{light}^2} (\lambda_i - 1)^2 + L$$

Find interest points



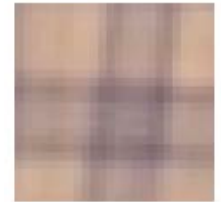
EM iterations



1



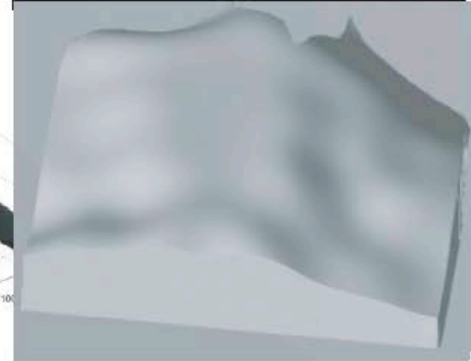
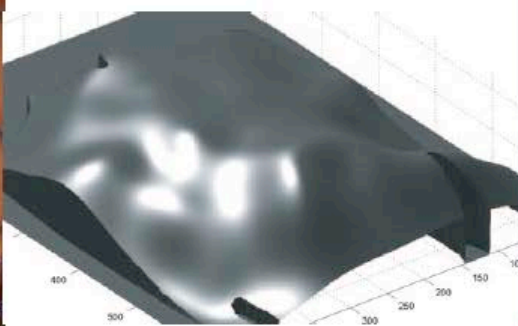
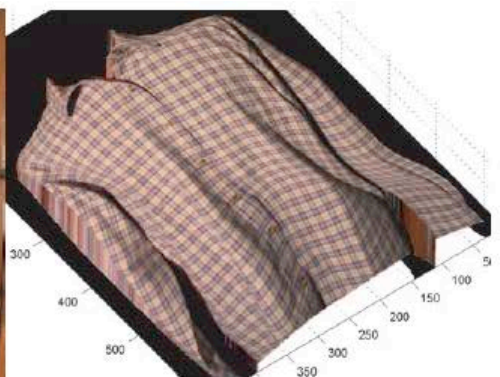
5



10

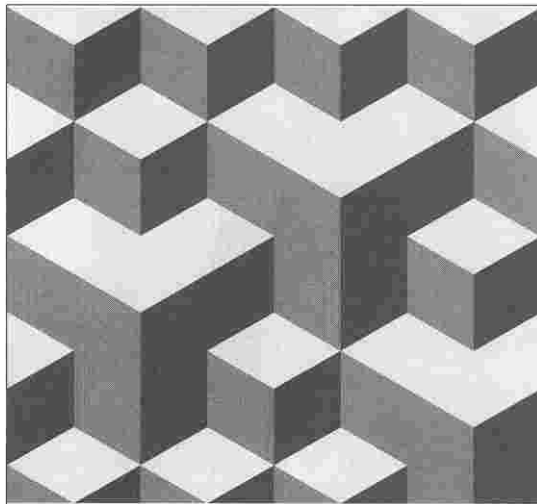
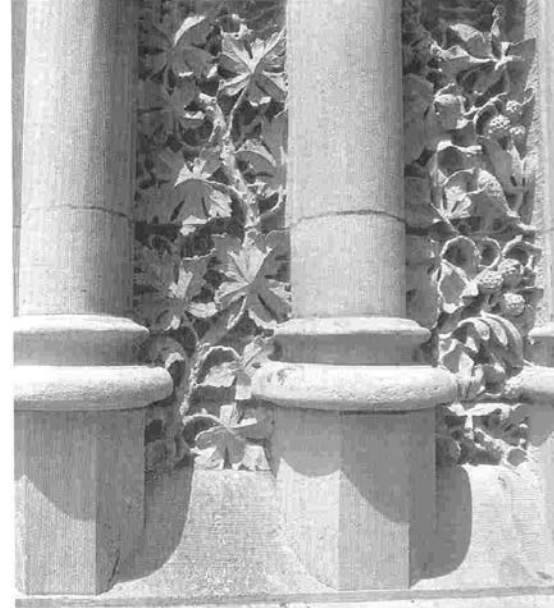


20



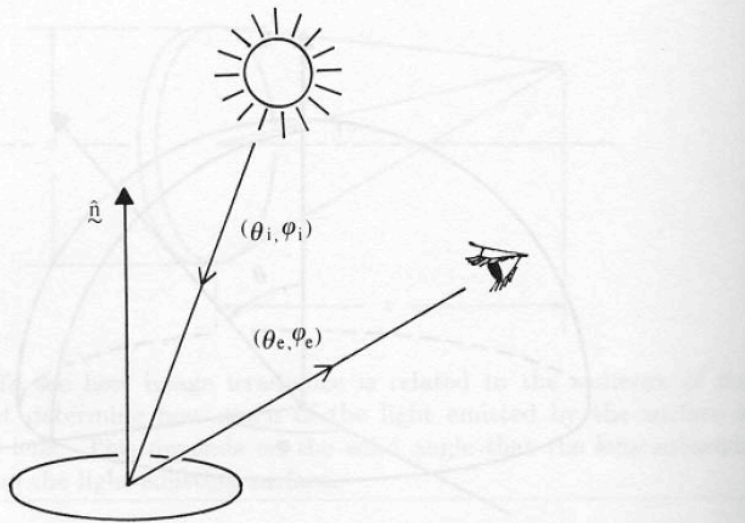
Shading

- Based on 3 dimensional modeling of objects in light, shade and shadows.



- Perception of depth through shading alone is always subject to the concave/convex inversion. The pattern shown can be perceived as stairs receding towards the top and lighted from above, or as an overhanging structure lighted from below.

Reflectance map



Horn, 1986

Figure 10-7. The bidirectional reflectance distribution function is the ratio of the radiance of the surface patch as viewed from the direction (θ_e, ϕ_e) to the irradiance resulting from illumination from the direction (θ_i, ϕ_i) .

$$BRDF = f(\theta_i, \phi_i, \theta_e, \phi_e) = \frac{L(\theta_e, \phi_e)}{E(\theta_i, \phi_i)}$$

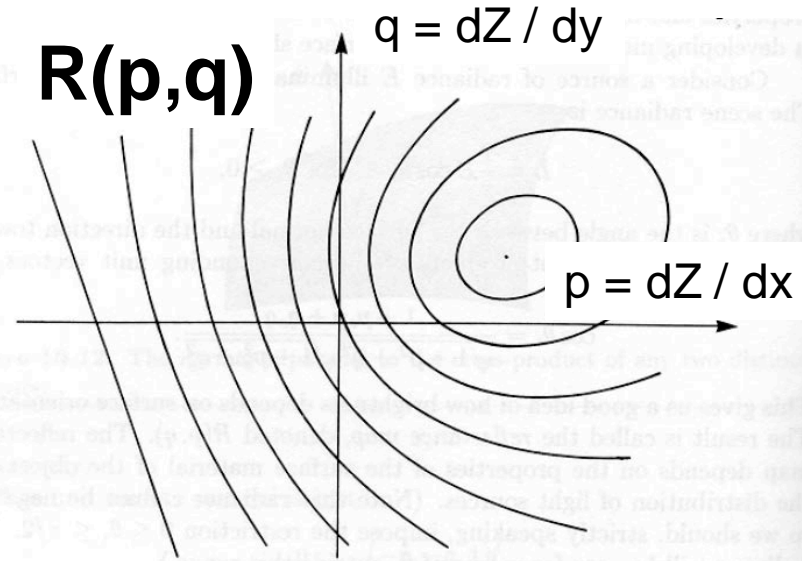


Figure 10-13. The reflectance map is a plot of brightness as a function of surface orientation. Here it is shown as a contour map in gradient space. In the case of a Lambertian surface under point-source illumination, the contours turn out to be nested conic sections. The maximum of $R(p, q)$ occurs at the point $(p, q) = (p_s, q_s)$, found inside the nested conic sections, while $R(p, q) = 0$ all along the line on the left side of the contour map.

Linear shape from shading

Lambertian point source

$$R(p, q) = k \frac{1 + p_s p + q_s q}{\sqrt{1 + p^2 + q^2} \sqrt{1 + p_s^2 + q_s^2}}$$

1st order Taylor

series about

$p=q=0$

$$\approx k_2 + \left. \frac{\partial R(p, q)}{\partial p} \right|_{p=0, q=0} p + \left. \frac{\partial R(p, q)}{\partial q} \right|_{p=0, q=0} q$$

$$= k_2 (1 + p_s p + q_s q)$$

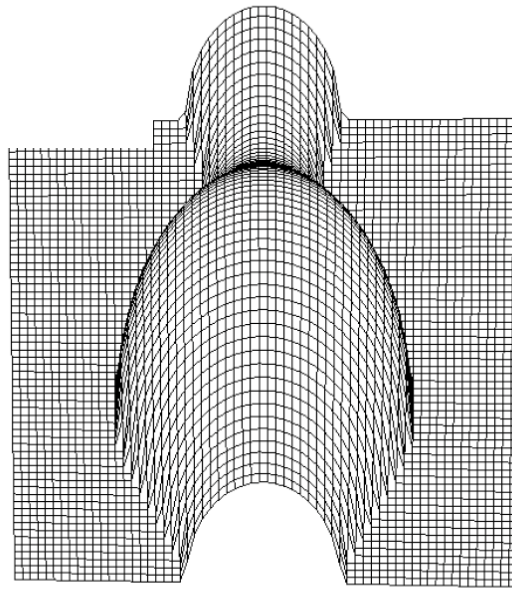
A close form solution can be obtained using the Fourier transform (Pentland 88)

$$\frac{\partial}{\partial x} Z(x, y) \longleftrightarrow F_Z(\omega_1, \omega_2) (-i\omega_1)$$

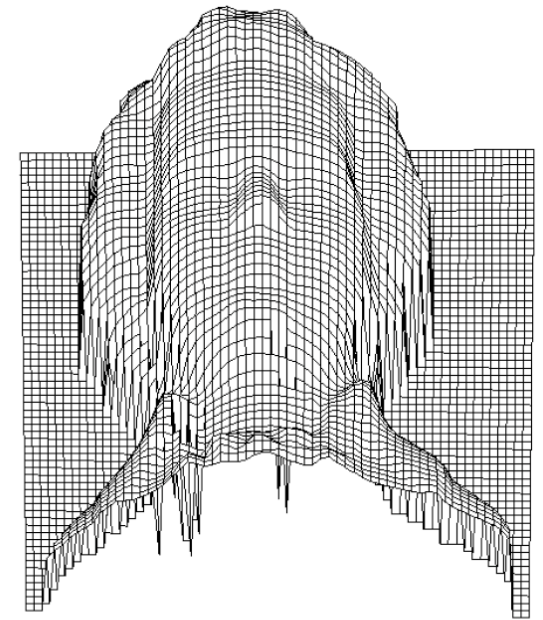
Shape from Shading: A Survey

Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah

Ground truth

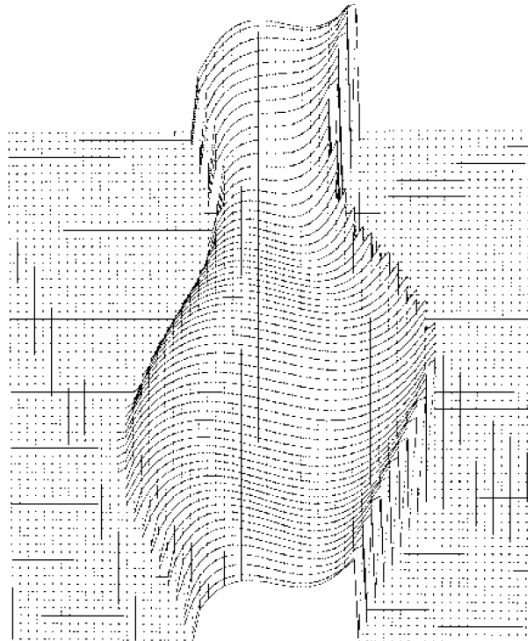


(a)

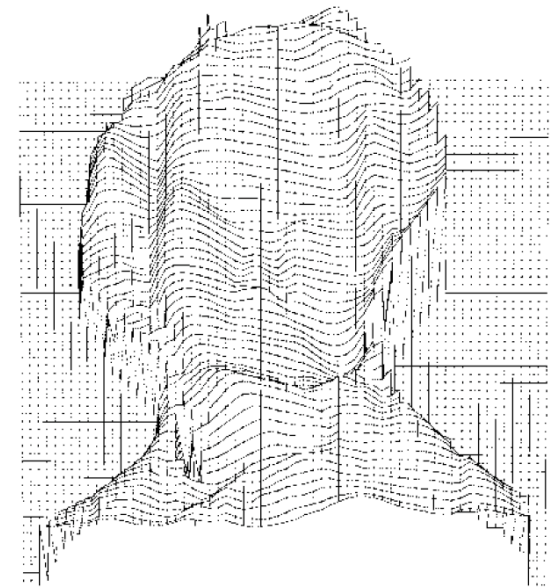


(b)

Linear shape
from shading



(a)



(b)

Learning based methods

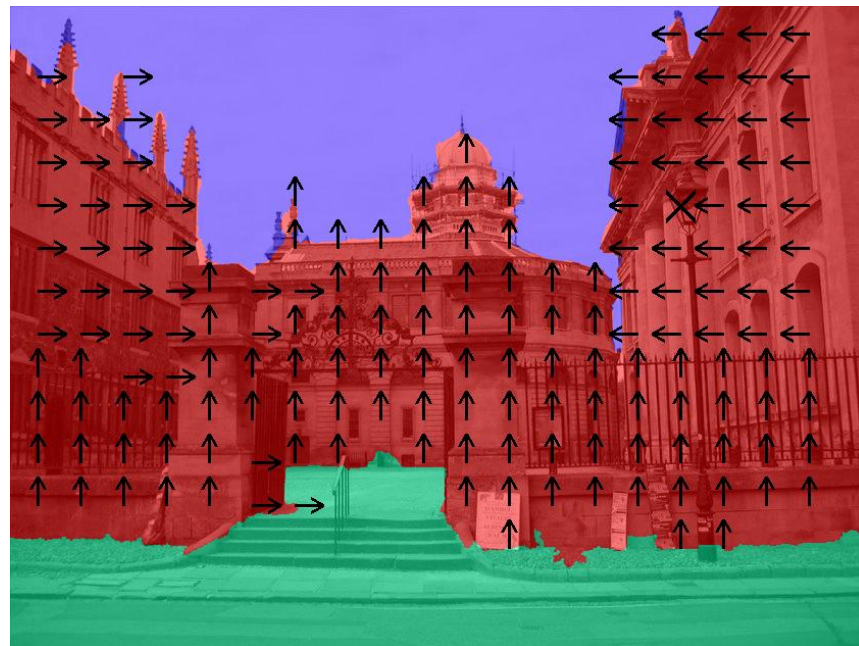
- User recognition to learn structure of the world from labeled examples



...



Label Geometric Classes



- **Goal:** learn labeling of image into 7 Geometric Classes:
- **Support (ground)**
- **Vertical**
 - Planar: facing **Left** (\leftarrow), **Center** (\uparrow), **Right** (\rightarrow)
 - Non-planar: **Solid** (X), **Porous** or wiry (O)
- **Sky**

What cues to use?



Vanishing points, lines



Color, texture, image location



Texture gradient Slides by Efros

Dataset very general



The General Case (outdoors)

- Typical outdoor photograph off the Web
 - Got 300 images using Google Image Search keyboards: “outdoor”, “scenery”, “urban”, etc.
- Certainly not random samples from world
 - 100% horizontal horizon
 - 97% pixels belong to 3 classes -- ground, sky, vertical (gravity)
 - Camera axis usually parallel to ground plane
- Still very general dataset!

Let's use many weak cues

- Material
- Image Location
- Perspective

SURFACE CUES
Location and Shape L1. Location: normalized x and y, mean L2. Location: norm. x and y, 10 th and 90 th pctl L3. Location: norm. y wrt estimated horizon, 10 th , 90 th pctl L4. Location: whether segment is above, below, or straddles estimated horizon L5. Shape: number of superpixels in segment L6. Shape: normalized area in image
Color C1. RGB values: mean C2. HSV values: C1 in HSV space C3. Hue: histogram (5 bins) C4. Saturation: histogram (3 bins)
Texture T1. LM filters: mean abs response (15 filters) T2. LM filters: hist. of maximum responses (15 bins)
Perspective P1. Long Lines: (num line pixels)/sqrt(area) P2. Long Lines: % of nearly parallel pairs of lines P3. Line Intersections: hist. over 8 orientations, entropy P4. Line Intersections: % right of center P5. Line Intersections: % above center P6. Line Intersections: % far from center at 8 orientations P7. Line Intersections: % very far from center at 8 orientations P8. Vanishing Points: (num line pixels with vertical VP membership)/sqrt(area) P9. Vanishing Points: (num line pixels with horizontal VP membership)/sqrt(area) P10. Vanishing Points: percent of total line pixels with vertical VP membership P11. Vanishing Points: x-pos of horizontal VP - segment center (0 if none) P12. Vanishing Points: y-pos of highest/lowest vertical VP wrt segment center P13. Vanishing Points: segment bounds wrt horizontal VP P14. Gradient: x, y center of gradient mag. wrt. image center

Need Spatial Support

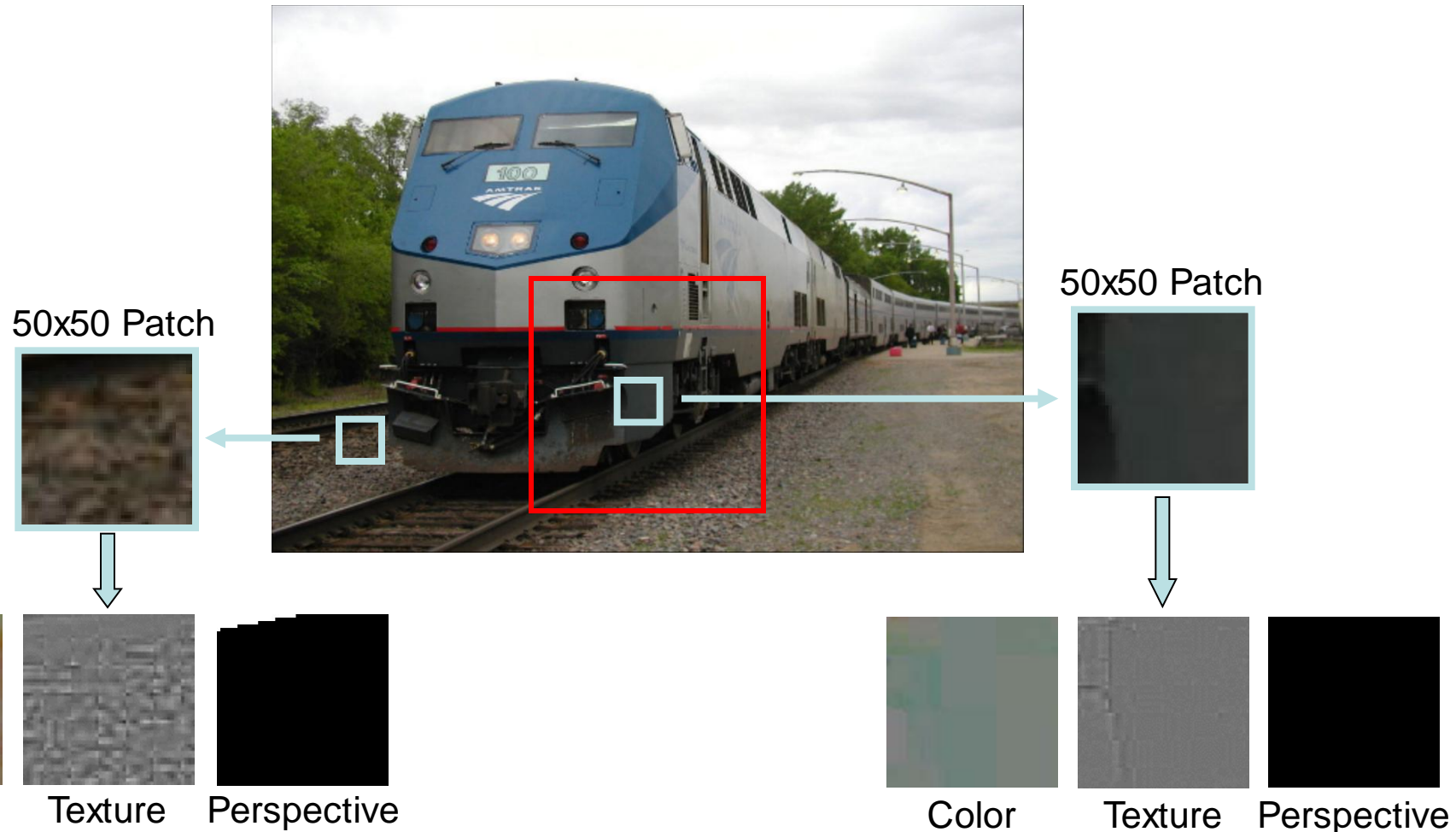
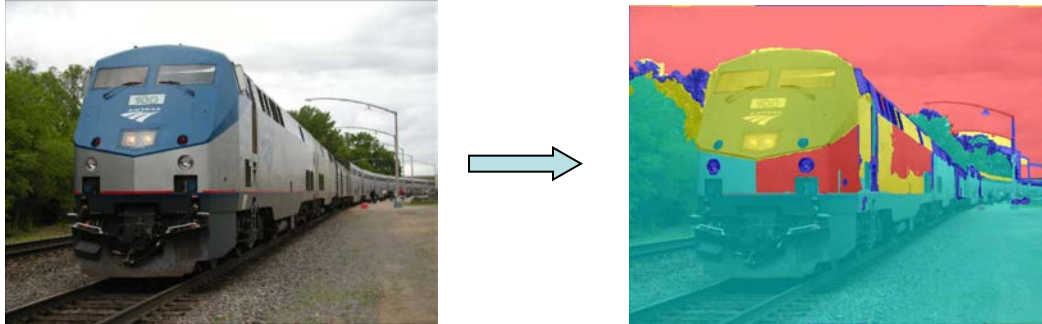


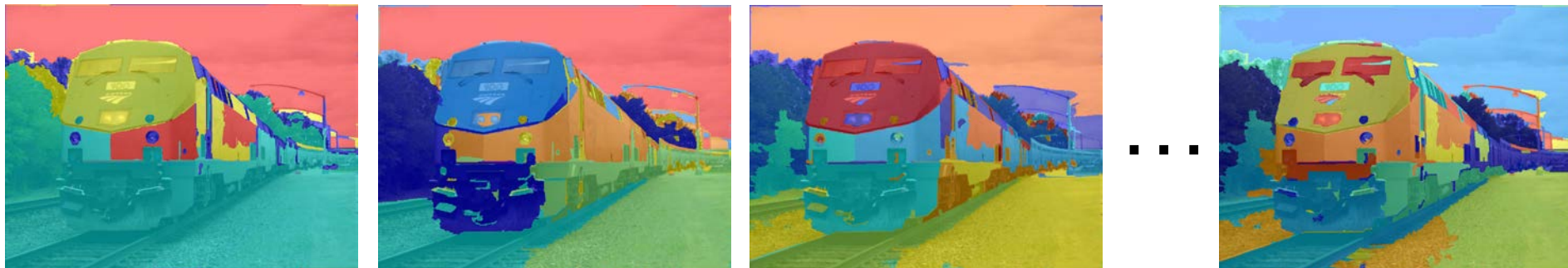
Image Segmentation

- Naïve Idea #1: segment the image



– Chicken & Egg problem

- Naïve Idea #2: multiple segmentations



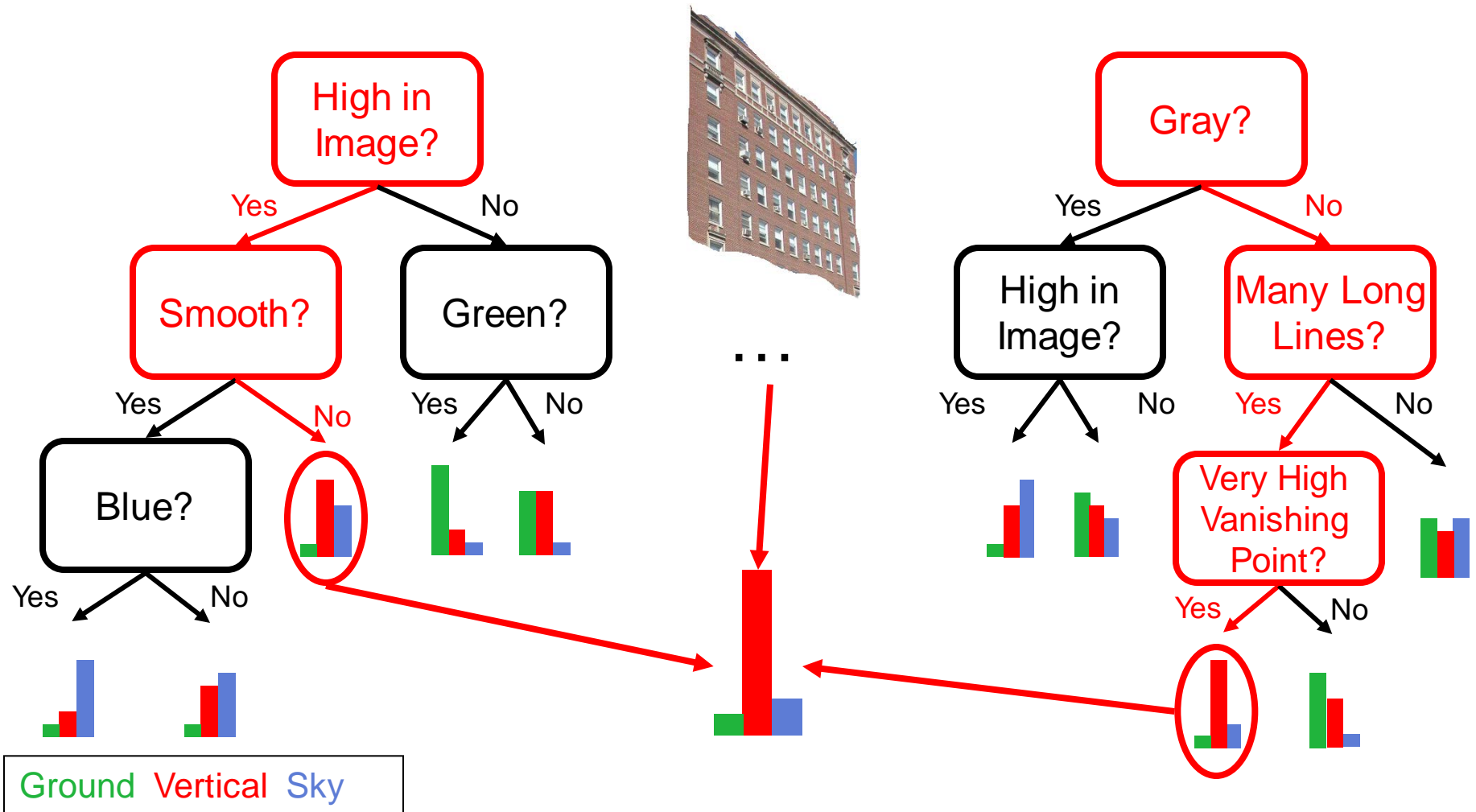
– Decide later which segments are good

Estimating surfaces from segments

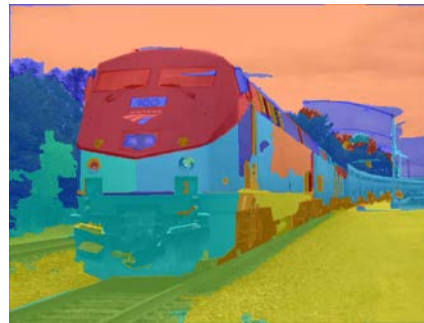
- We want to know:
 - Is this a good (coherent) segment?
 $P(\text{good segment} \mid \text{data})$
 - If so, what is the surface label?
 $P(\text{label} \mid \text{good segment}, \text{data})$
- *Learn* these likelihoods from training images
 - we use Boosted Decision Trees



Boosted Decision Trees



Labeling Segments



For each segment:

- Get $P(\text{good segment} \mid \text{data}) P(\text{label} \mid \text{good segment}, \text{data})$

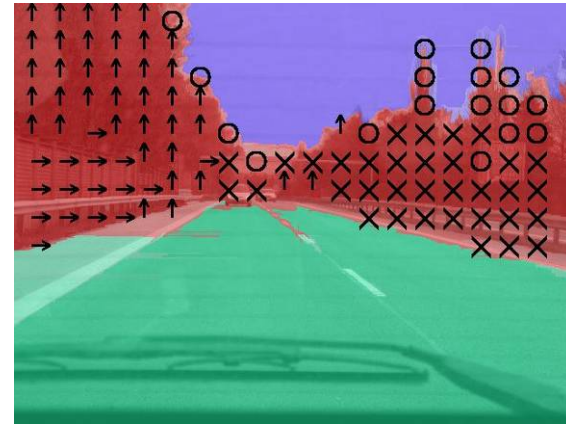
Image Labeling

Labeled Segmentations



Labeled Pixels

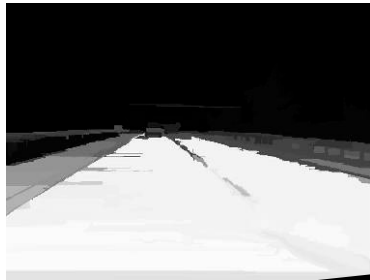
No Hard Decisions



Support

Vertical

Sky



V-Left

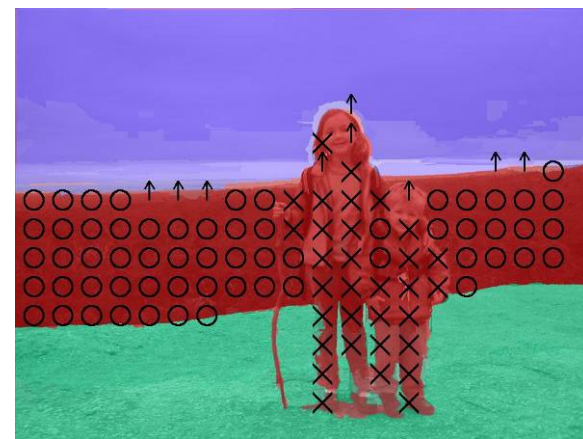
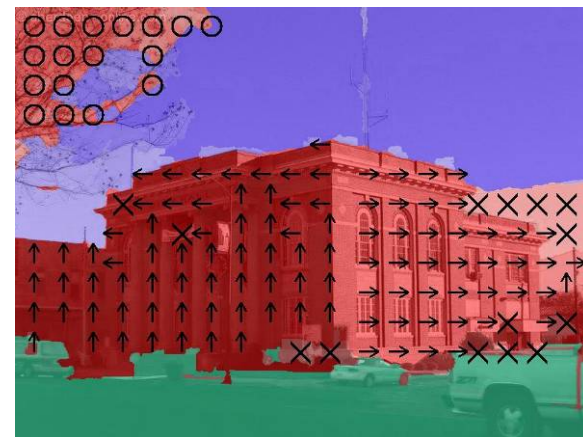
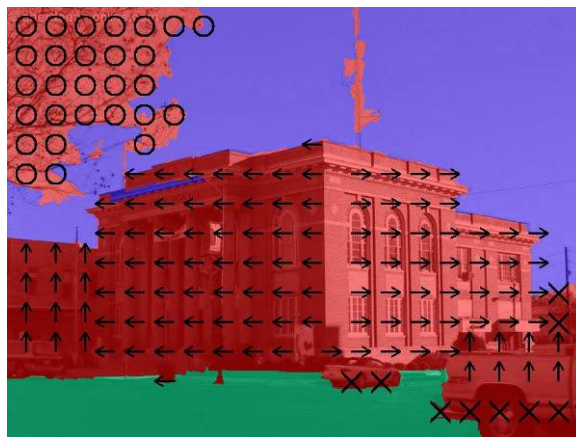
V-Center

V-Right

V-Porous

V-Solid

Labeling Results



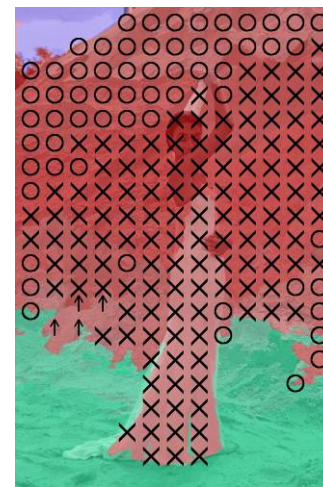
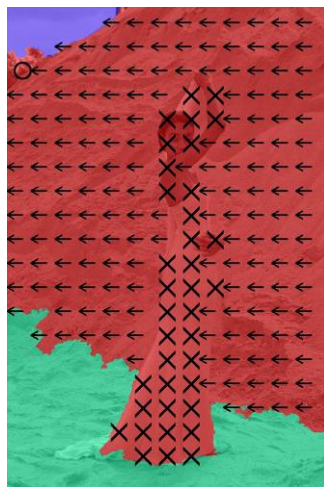
Input image

Ground Truth

Our Result

Slides by Efros

Labeling Results



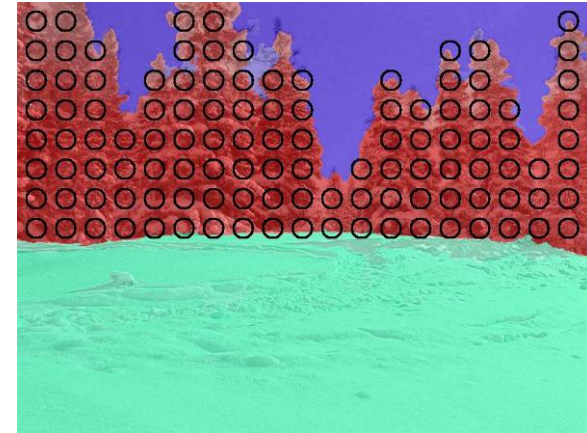
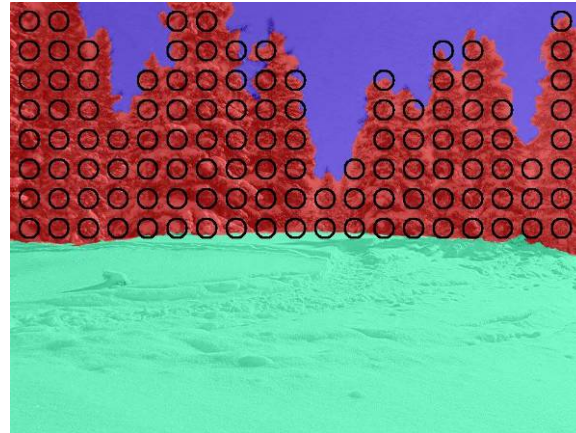
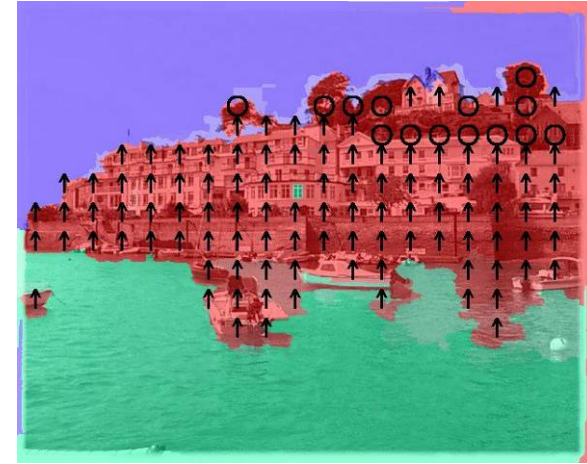
Input image

Ground Truth

Our Result

Slides by Efros

Labeling Results



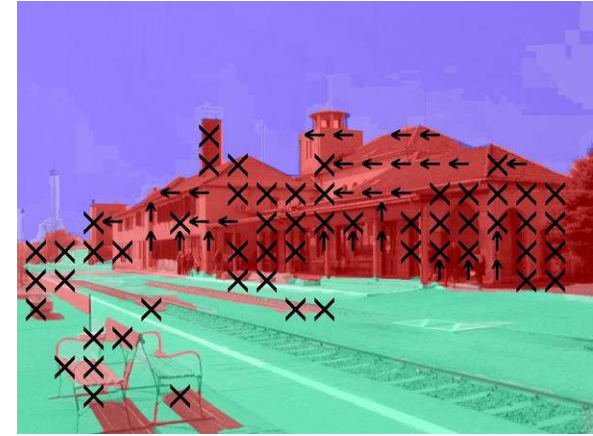
Input image

Ground Truth

Our Result

Slides by Efros

Labeling Results



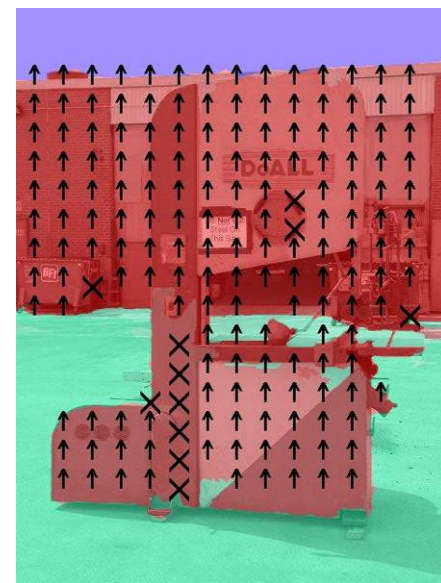
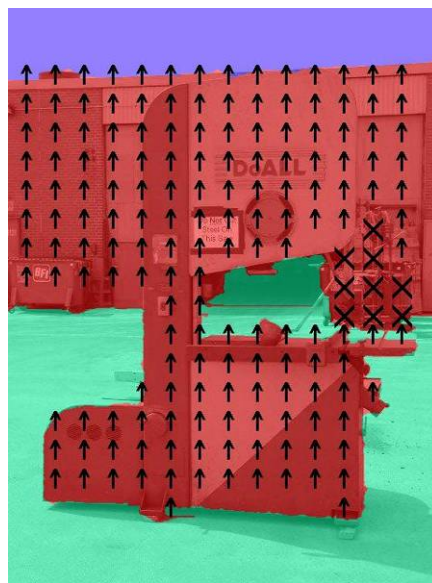
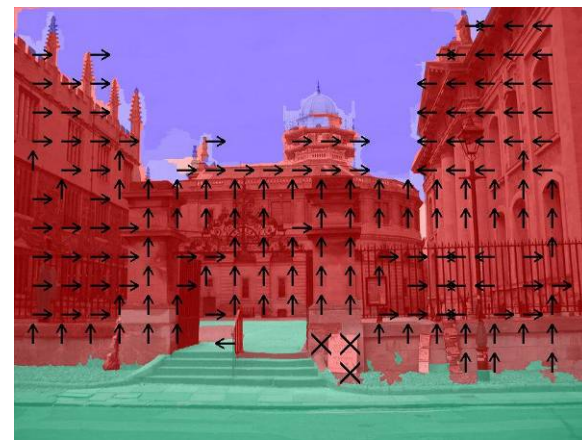
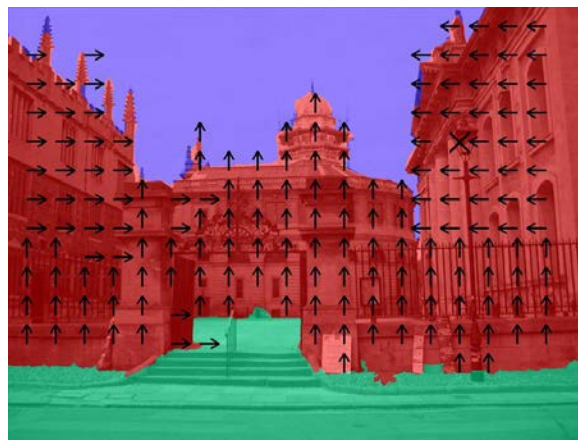
Input image

Ground Truth

Our Result

Slides by Efros

Labeling Results



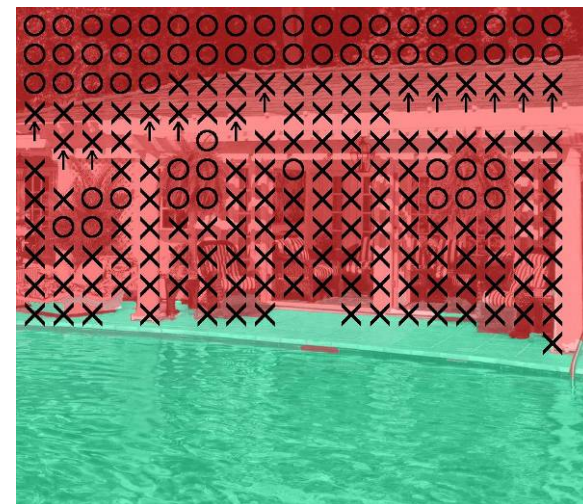
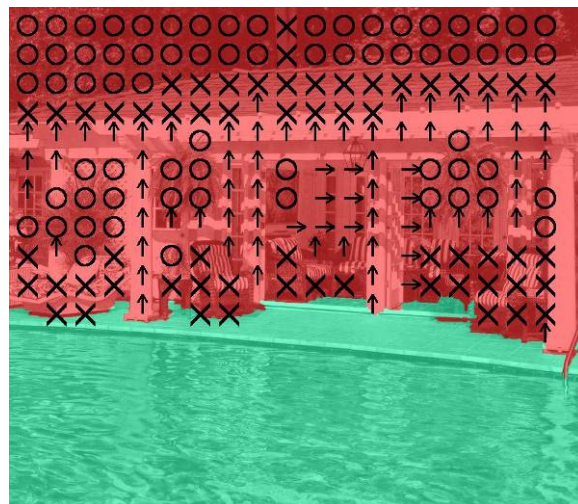
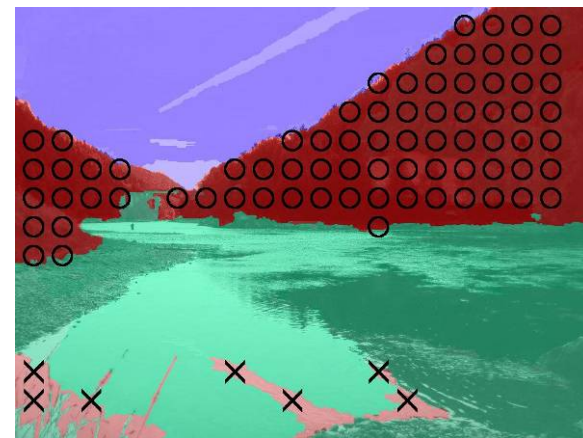
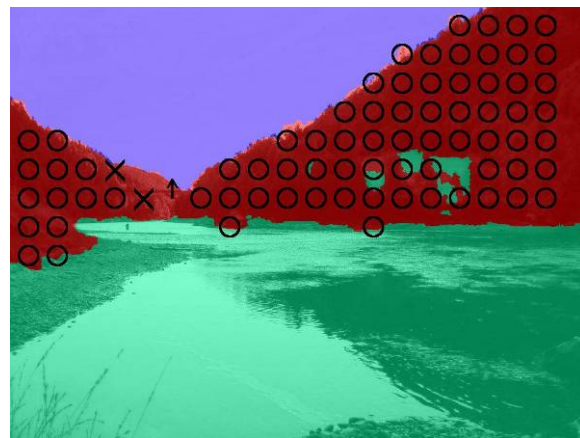
Input image

Ground Truth

Our Result

Slides by Efros

Labeling Results

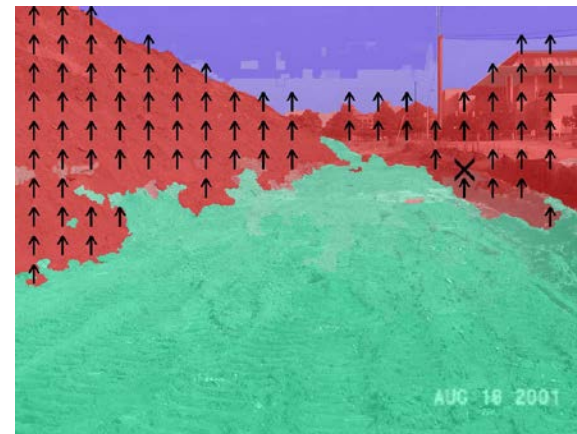
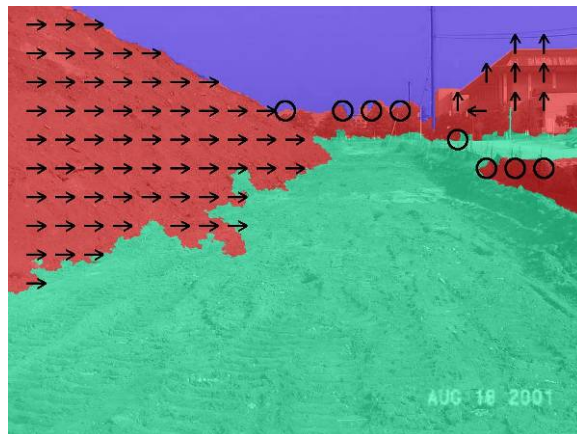


Input image

Ground Truth

Our Result

Labeling Results

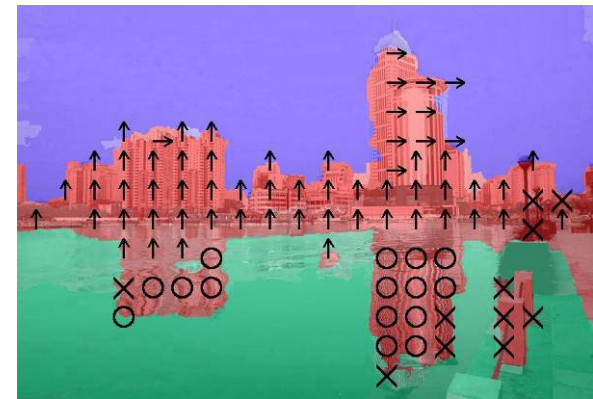
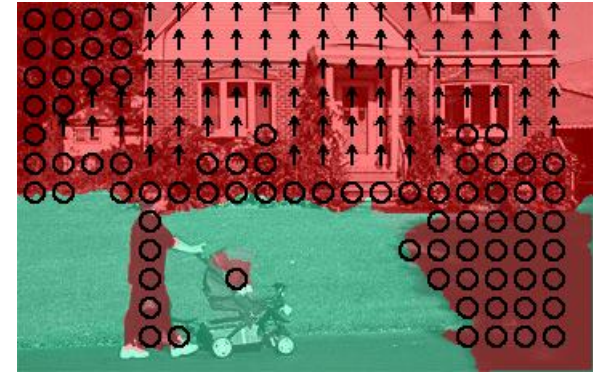


Input image

Ground Truth

Our Result

Some Failures



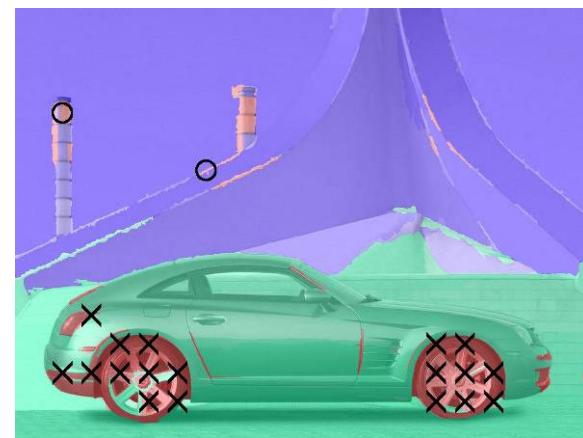
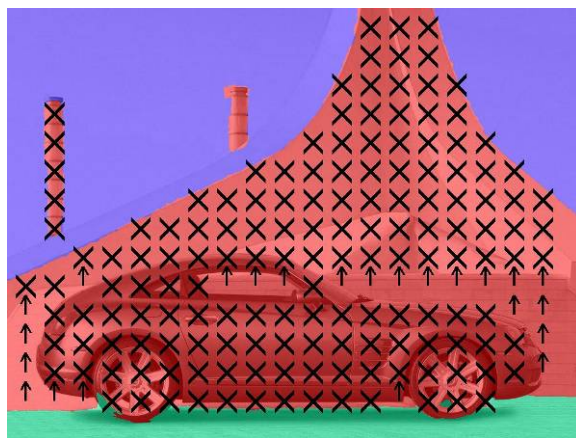
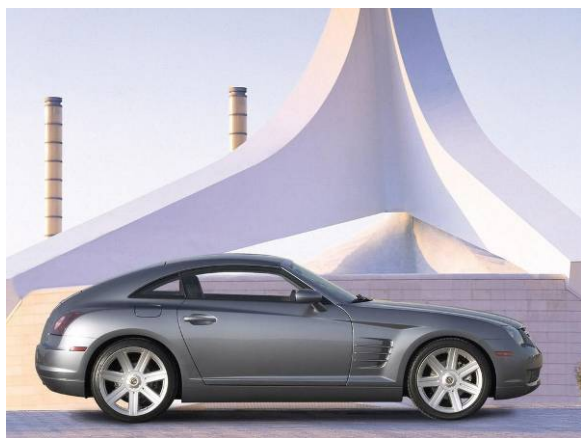
Input image

Ground Truth

Our Result

Slides by Efros

Catastrophic Failures



Input image

Ground Truth

Our Result

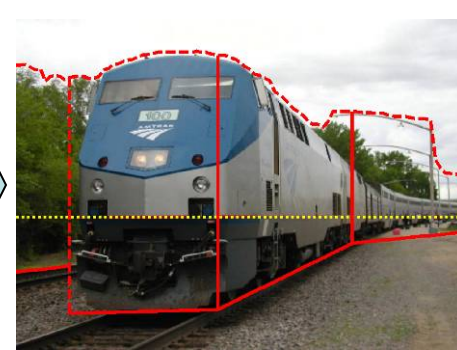
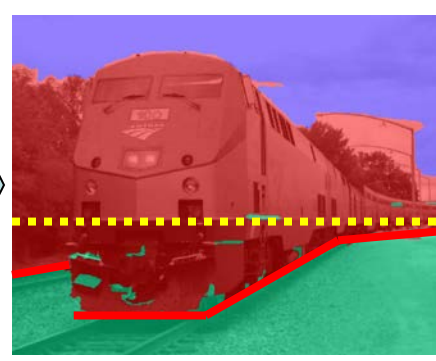
Automatic Photo Popup

Labeled Image

Fit Ground-Vertical
Boundary with Line
Segments

Form Segments
into Polylines

Cut and Fold



Final Pop-up Model

