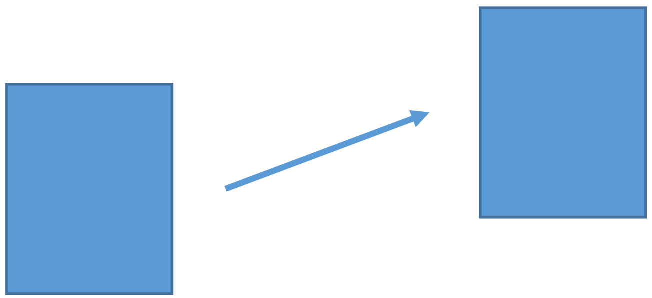
Image Transformations

Translation

$$\mathbf{x} \mapsto \mathbf{x} + \mathbf{t} \quad \text{where} \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^2$$



Can be written in homogeneous coordinates as

$$\begin{aligned} \mathbf{x}' &= \mathbf{T}\,\hat{\mathbf{x}} \\ &= [\mathbf{I} \mid \mathbf{t}] \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \end{aligned}$$

where $\hat{\mathbf{x}}$ represents $\mathbf{x}$ in homogeneous coordinates.

This can be written entirely in homogeneous coordinates as

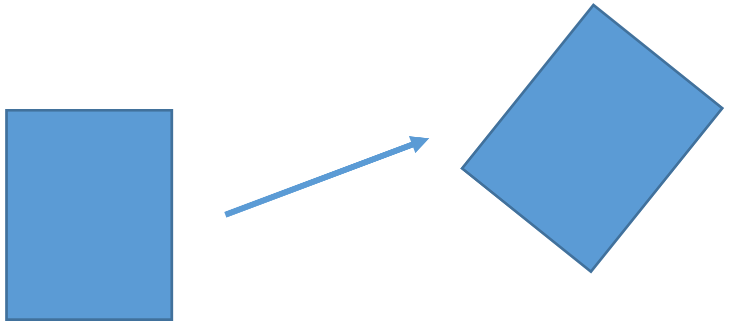
$$\hat{\mathbf{x}}' = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Rotation and translation

$$\mathbf{x} \mapsto \mathbf{R}\mathbf{x} + \mathbf{t} \quad \text{where} \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^2$$

and

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} .$$



Can be written in homogeneous coordinates as

$$\begin{aligned} \mathbf{x}' &= \mathbf{T}\,\hat{\mathbf{x}} \\ &= [\mathbf{R} \mid \mathbf{t}] \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} . \end{aligned}$$

This can be written entirely in homogeneous coordinates as

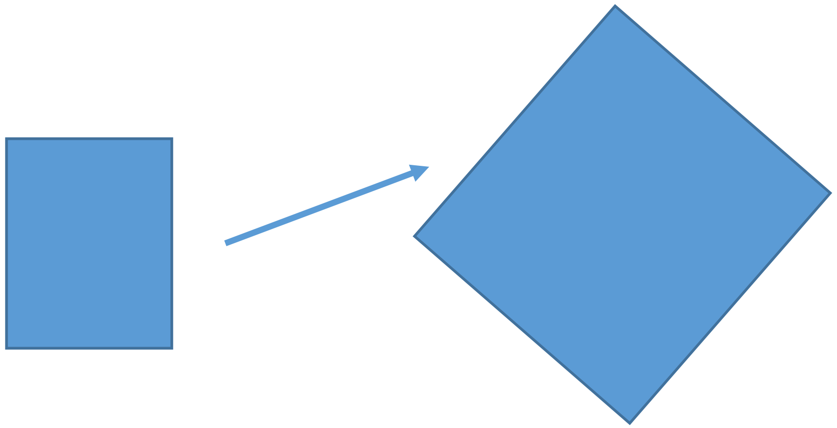
$$\hat{\mathbf{x}}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Rotation, translation and scaling

$$\mathbf{x} \mapsto s(\mathbf{R}\mathbf{x} + \mathbf{t})$$

Can be written in homogeneous coordinates as

$$\begin{aligned} \mathbf{x}' &= \mathbf{T}\,\hat{\mathbf{x}} \\ &= s[\mathbf{R} \mid \mathbf{t}] \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \ . \end{aligned}$$



This can be written entirely in homogeneous coordinates as

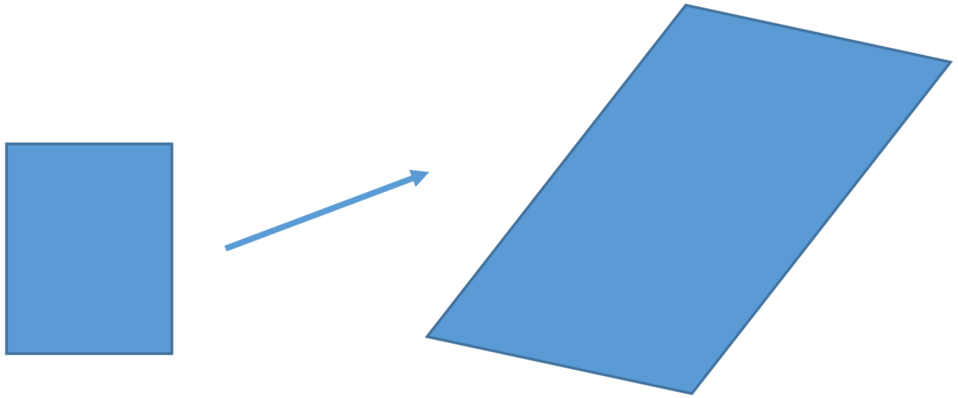
$$\hat{\mathbf{x}}' = s \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Affine transformation

$$\mathbf{x} \mapsto \mathbf{A}\mathbf{x} + \mathbf{t}\ .$$

Can be written in homogeneous coordinates as

$$\begin{aligned}\mathbf{x}' &= \mathbf{T}\,\hat{\mathbf{x}} \\ &= [\mathbf{A} \mid \mathbf{t}] \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}\ .\end{aligned}$$



This can be written entirely in homogeneous coordinates as

$$\hat{\mathbf{x}}' = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Preserving the plane at infinity

We note

(i) All these transformations *preserve the plane at infinity*.

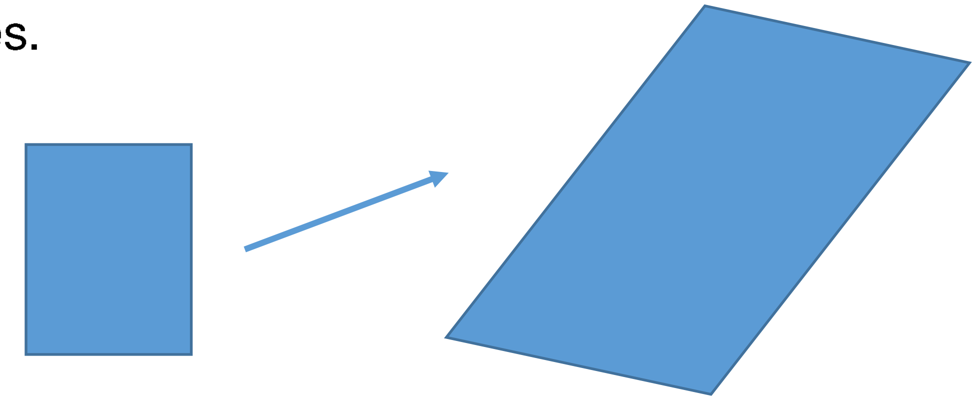
(ii) Plane at infinity is the set of points with homogeneous coordinates $\mathbf{x} = (x, y, 0)^\top$.

(iii) Verify:

$$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ 0 \end{pmatrix}$$

(iv) They map parallel lines to parallel lines.

(v) They are transformations of $\mathbb{R}^2$.



Preserving the plane at infinity

We note

(i) All these transformations *preserve the plane at infinity*.

(ii) Plane at infinity is the set of points with homogeneous coordinates $\mathbf{x} = (x, y, 0)^\top$.

(iii) Verify:

$$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ 0 \end{pmatrix}$$

(iv) They map parallel lines to parallel lines.

(v) They are transformations of $\mathbb{R}^2$.

Notation

(i) From now on, we assume that all points are given in homogenous coordinates

(ii) Points (in homogeneous coordinates) are denoted simply by $\mathbf{x}$, and not by $\hat{\mathbf{x}}$.

Homographies

(i) Affine (infinity-preserving) mappings are sometimes not enough to capture the transformations that apply to images.

(a) Views taken from different viewpoints.

(b) Correspondences between world-plane and image.

(ii) We need a transformation:

$$\begin{aligned}\mathbf{x}' &= \mathbf{H}\mathbf{x} \\ &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}\end{aligned}$$

(iii) In non-homogeneous coordinates:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{13}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{13}}$$

(iv) They are transformations of P^2 (the projective plane).

Homographies

Homographies used in image stitching.

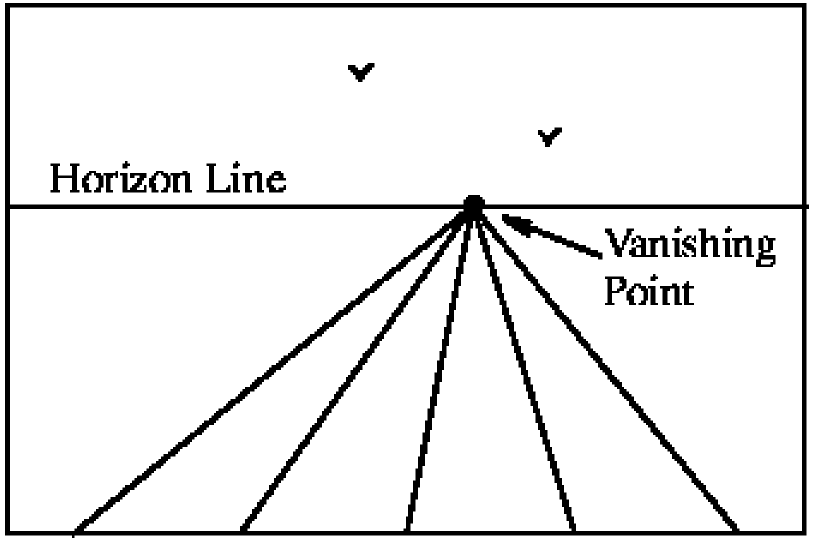
Homographies applied to images to create mosaic.

Note the projective distortion of the images.

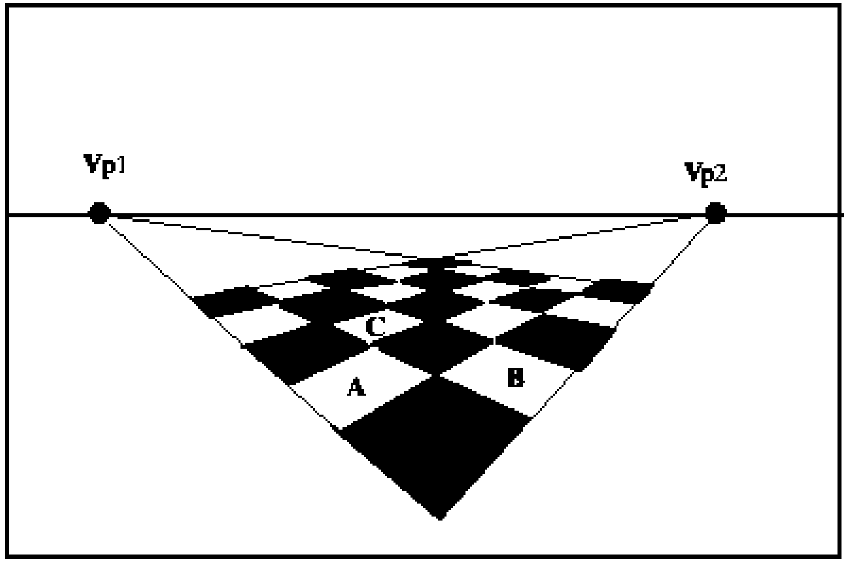
Every image stitching and mosaicing program uses homographies to match images.

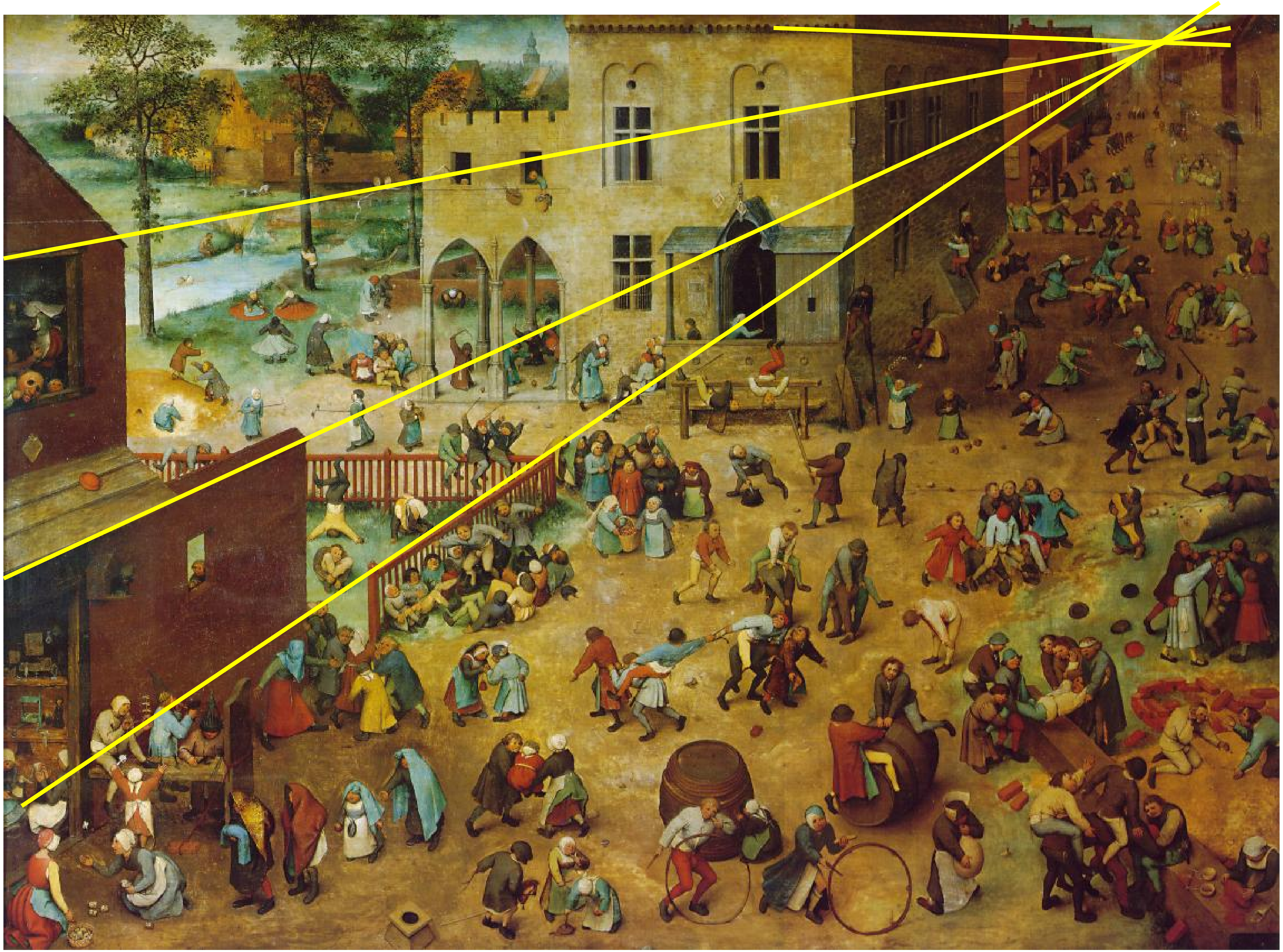
Vanishing lines

- Lines parallel to a scene plane are imaged with their vanishing point on the plane's vanishing line



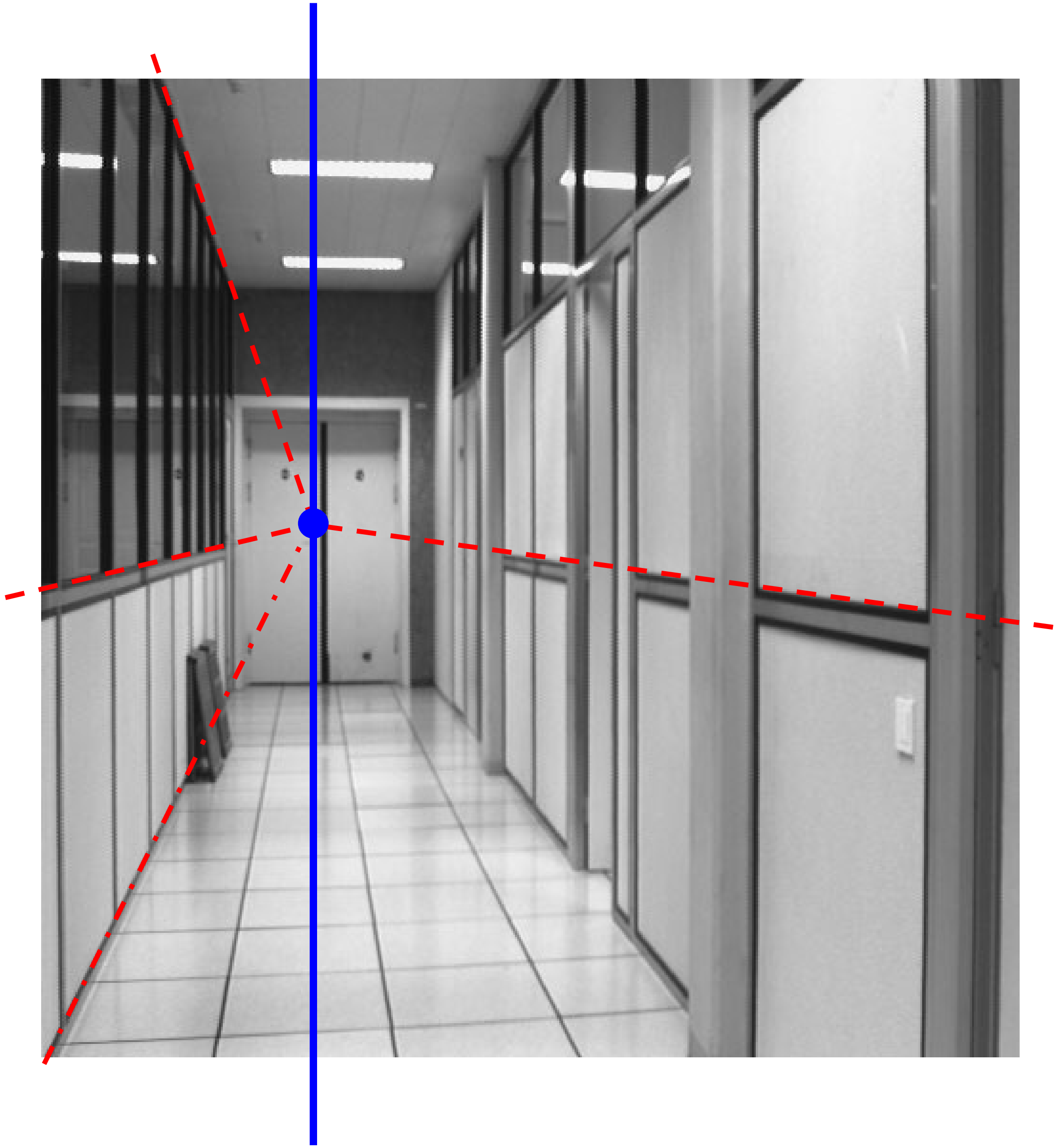
- Parallel planes in the scene have the same vanishing line



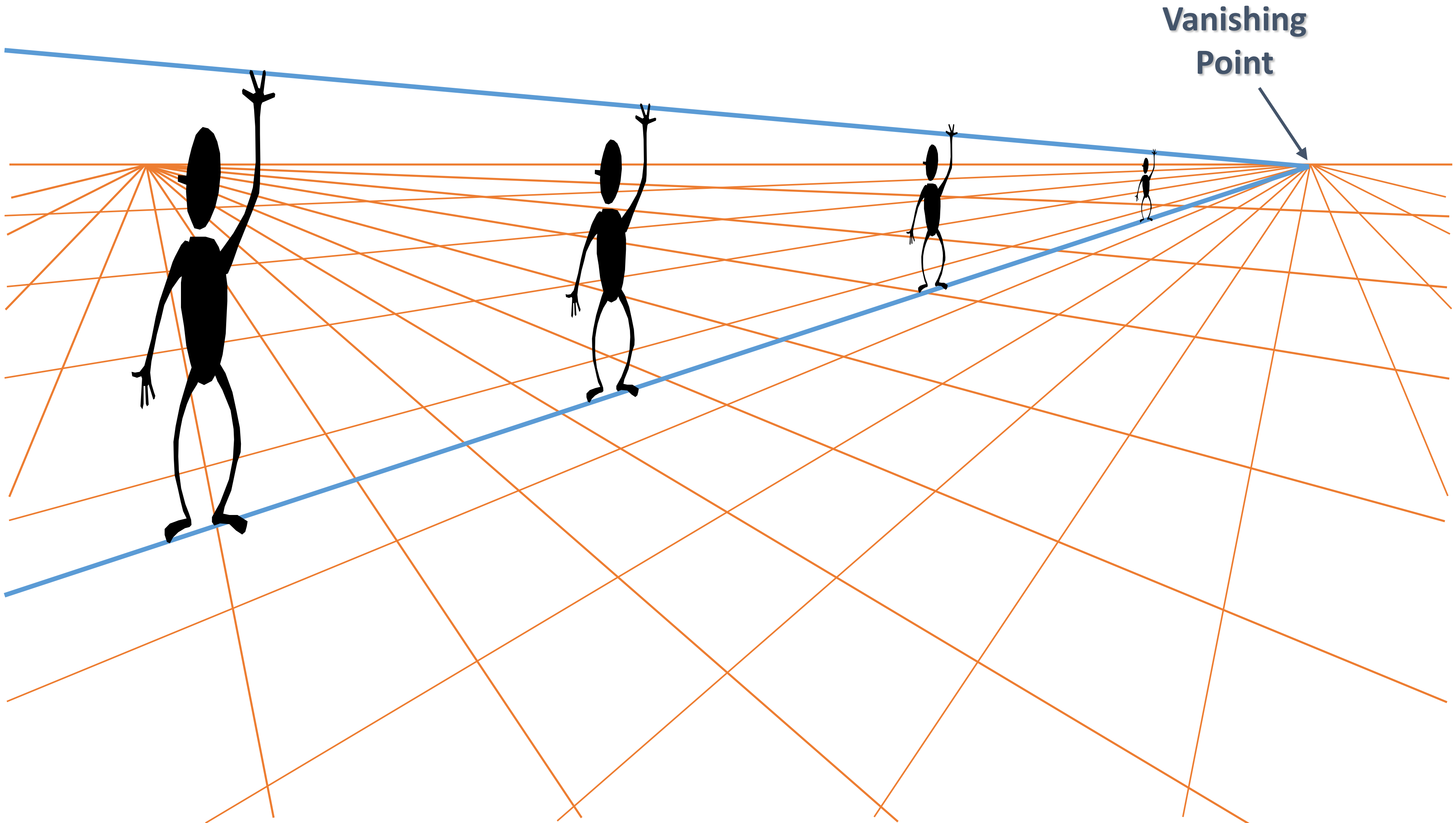


Example

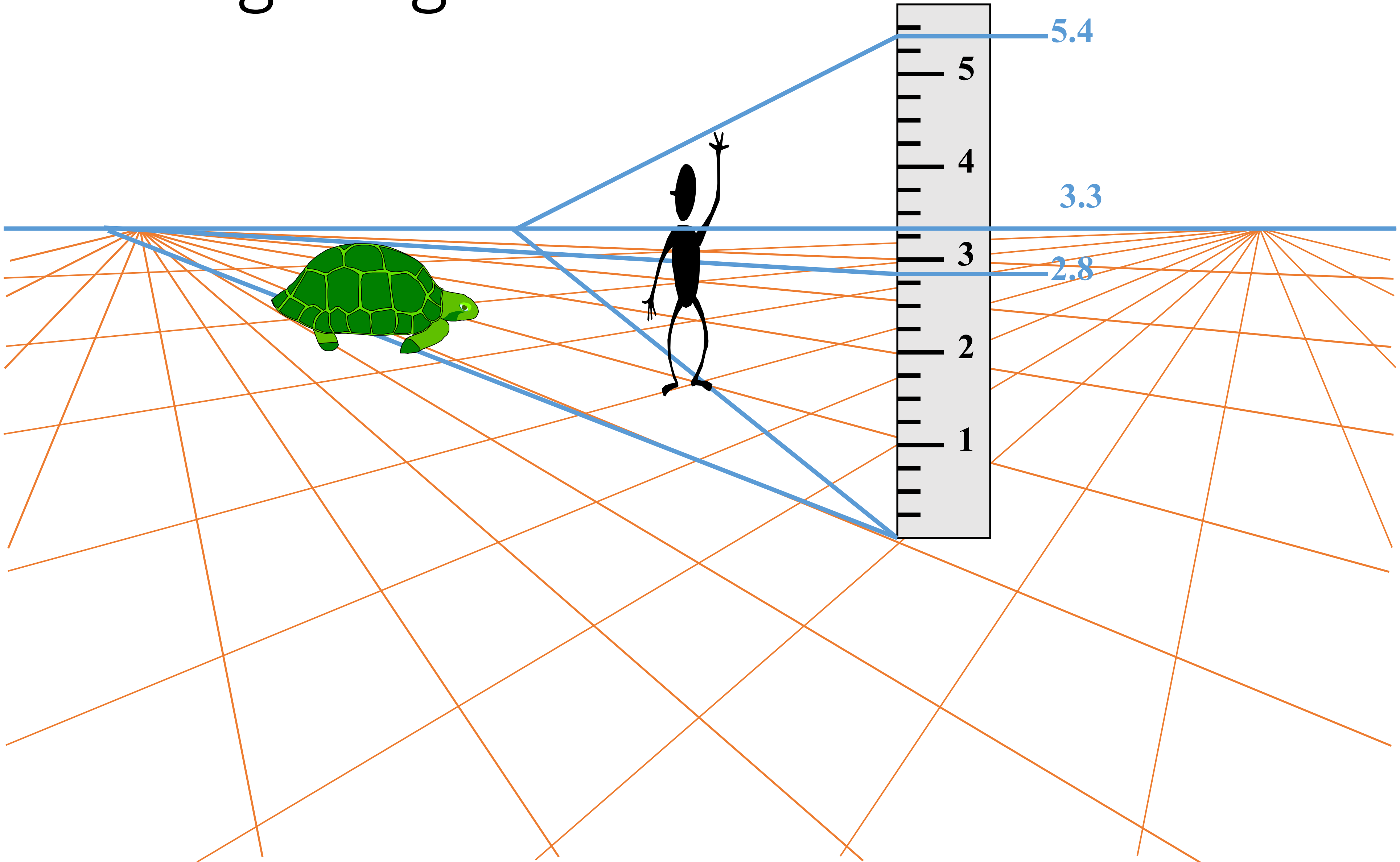
vanishing line for side wall planes



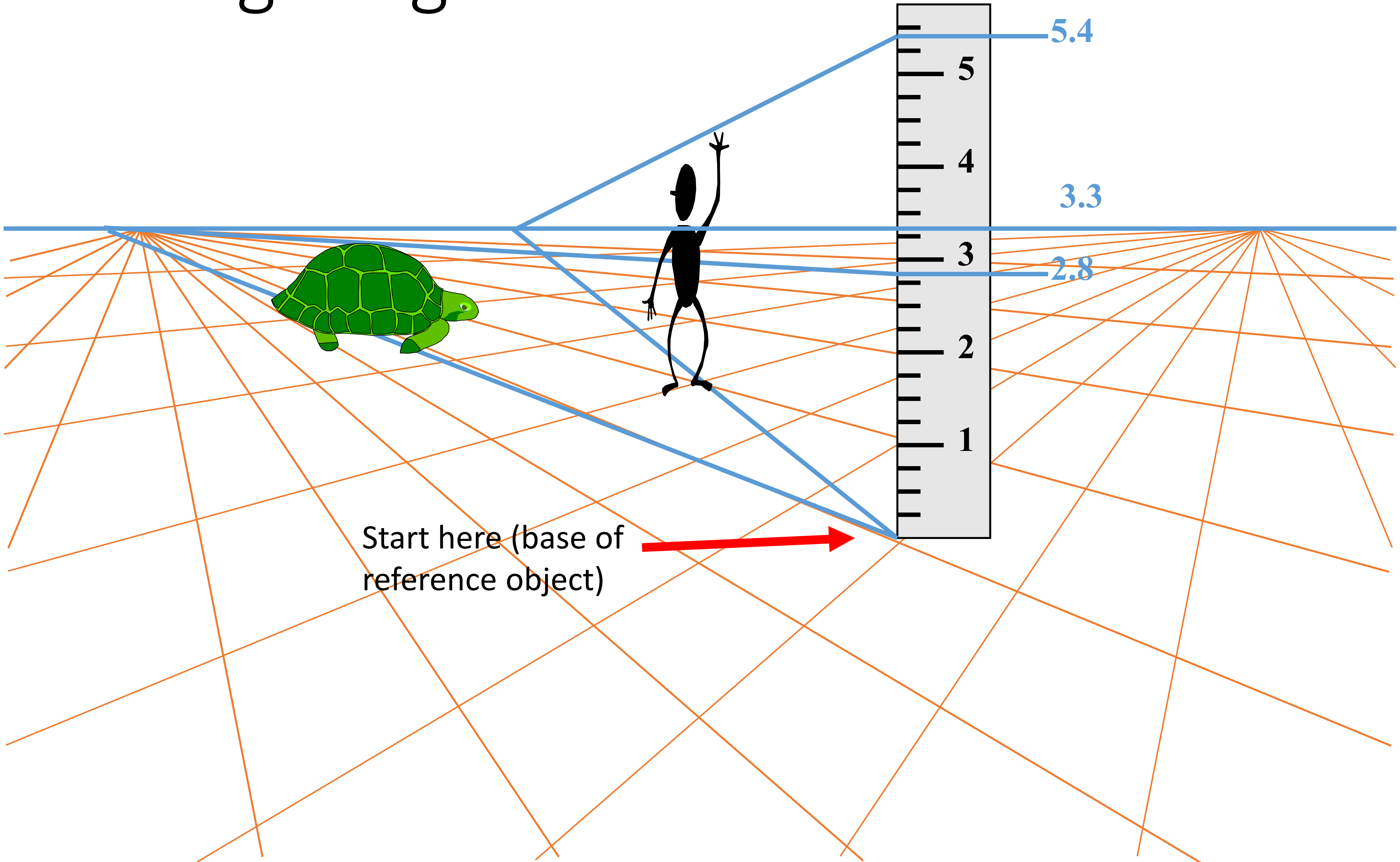
Vanishing Points



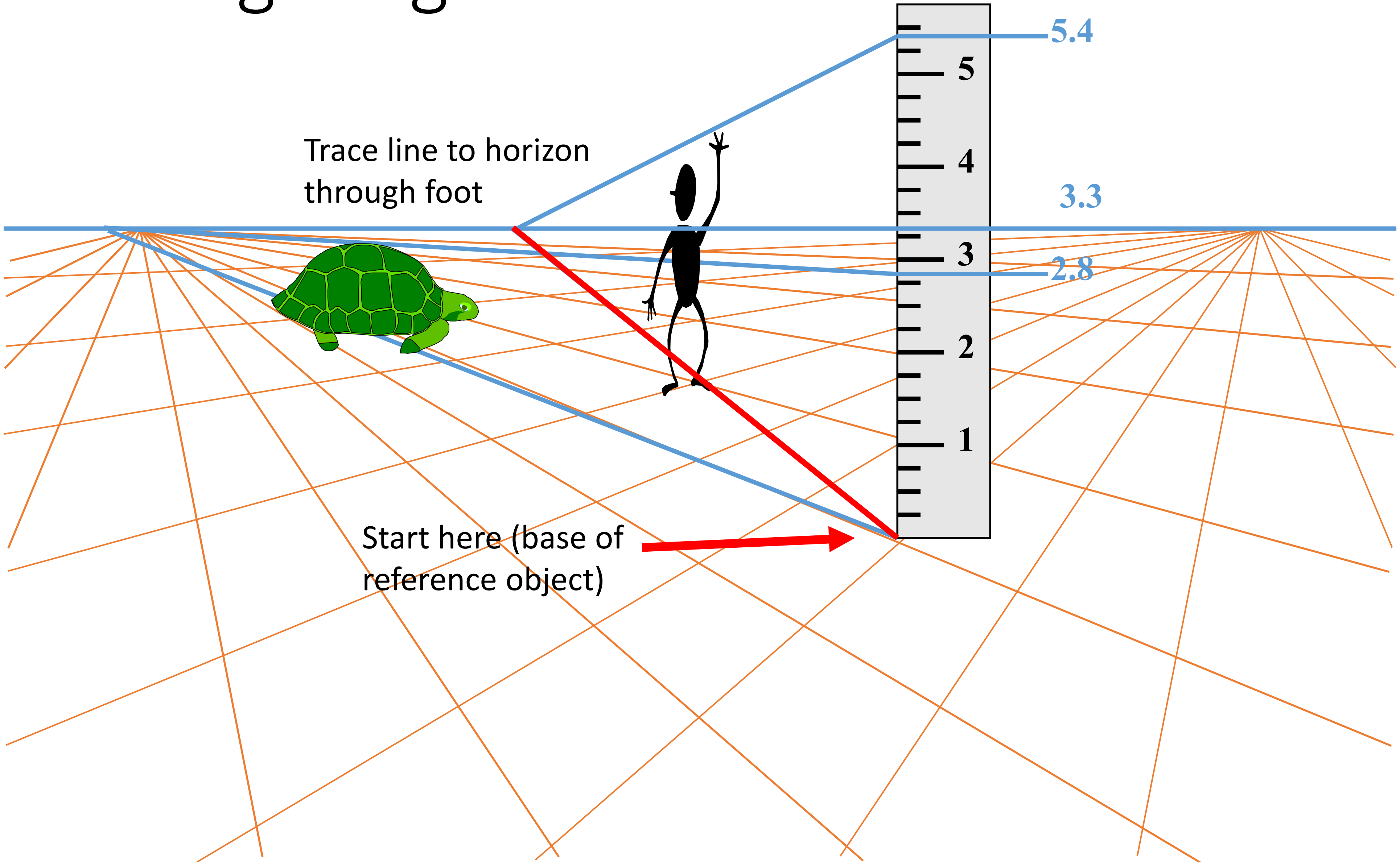
Measuring Height



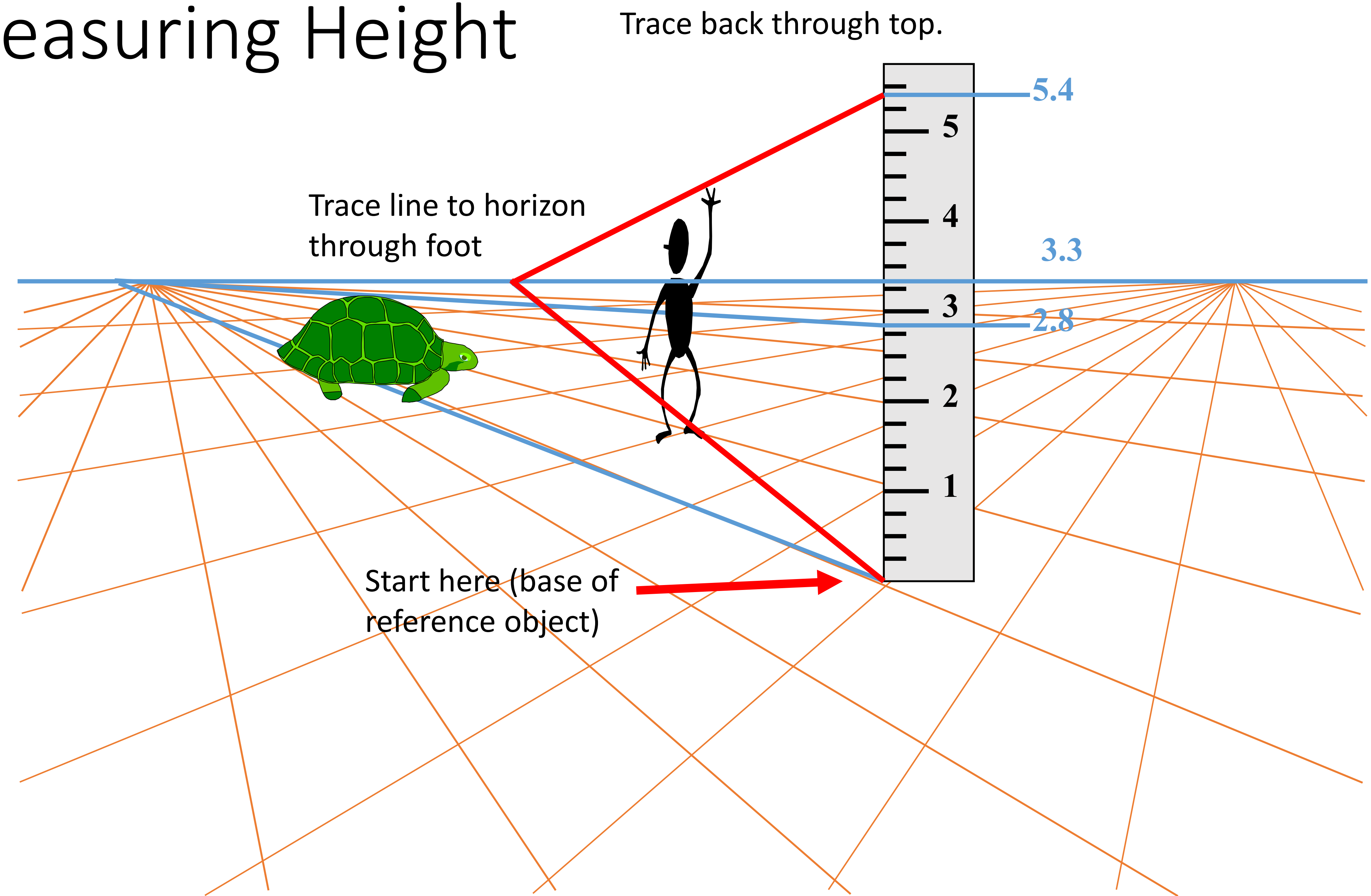
Measuring Height



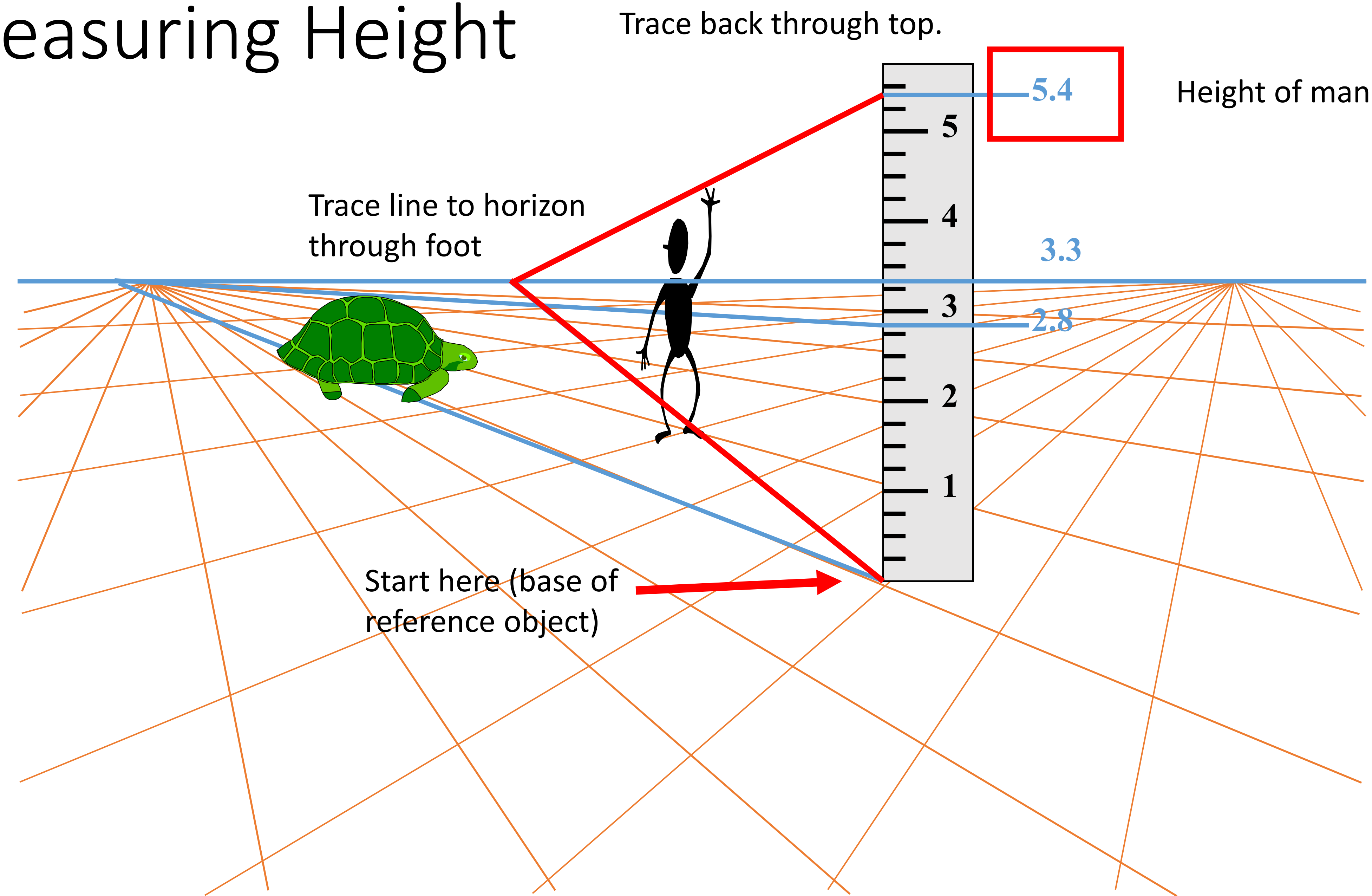
Measuring Height



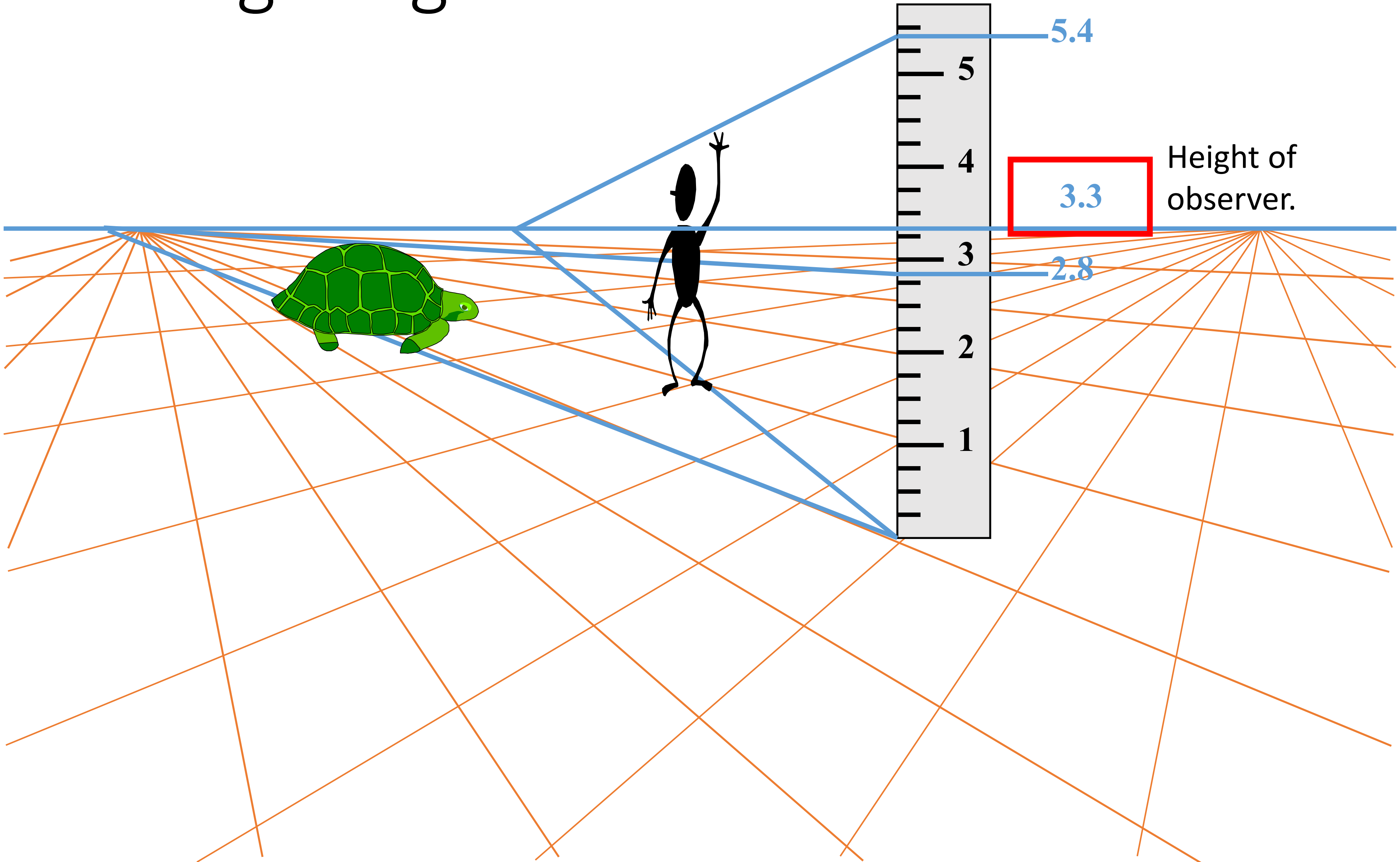
Measuring Height



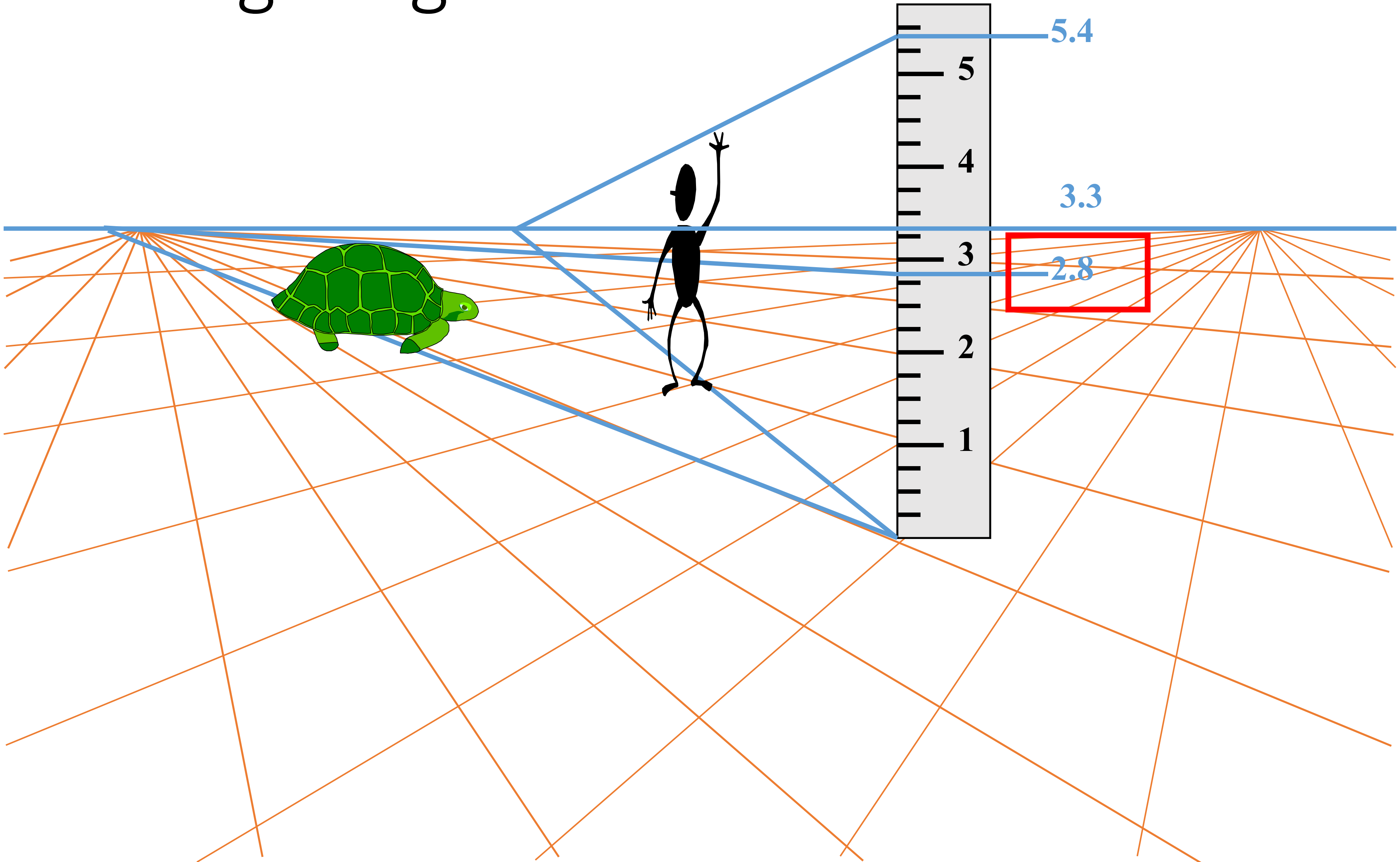
Measuring Height



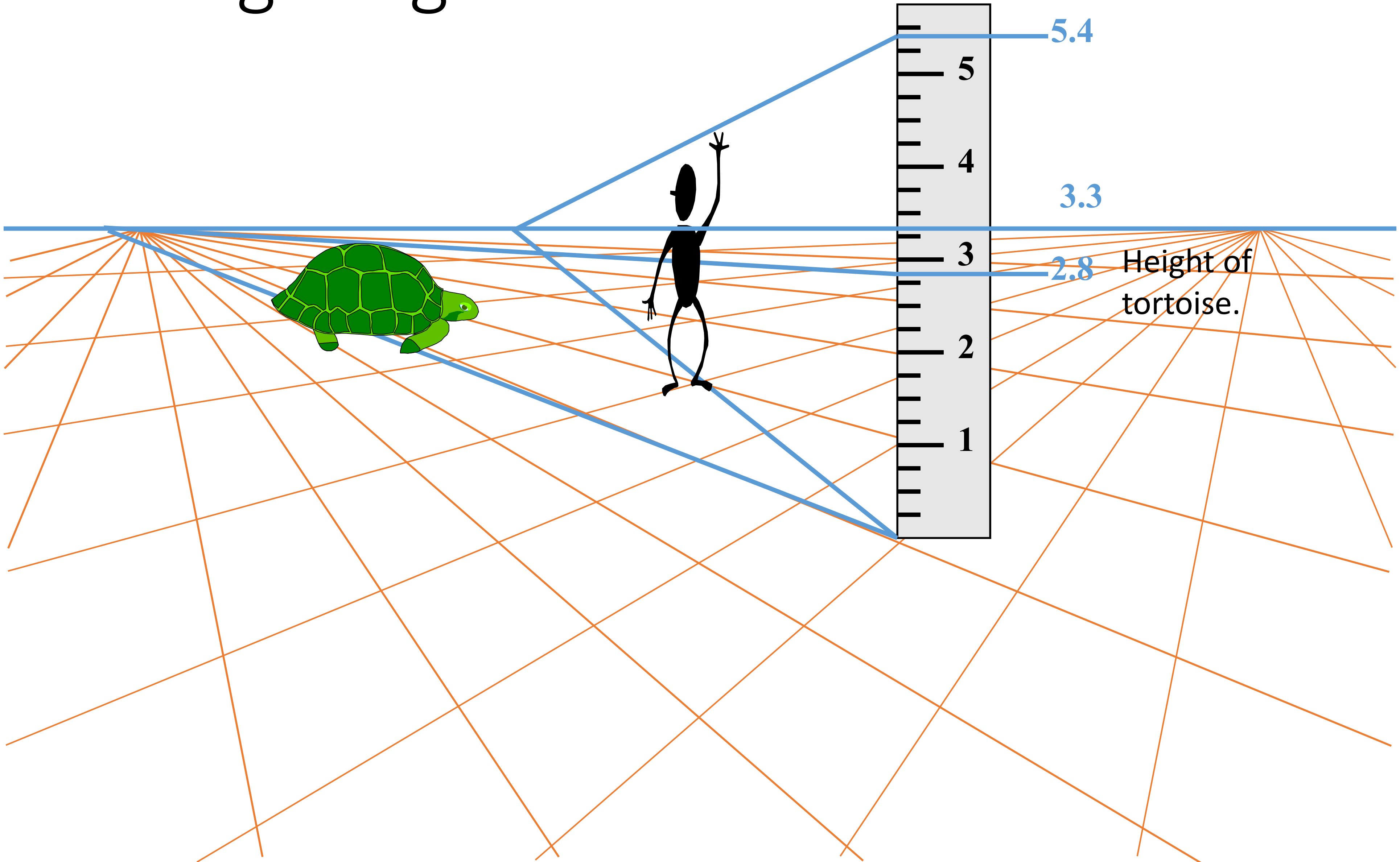
Measuring Height



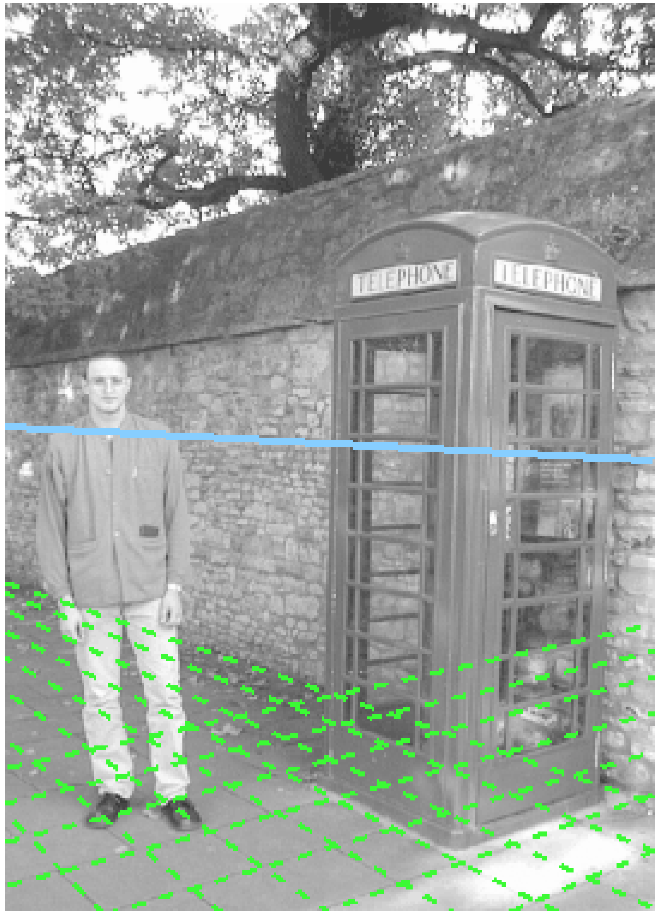
Measuring Height



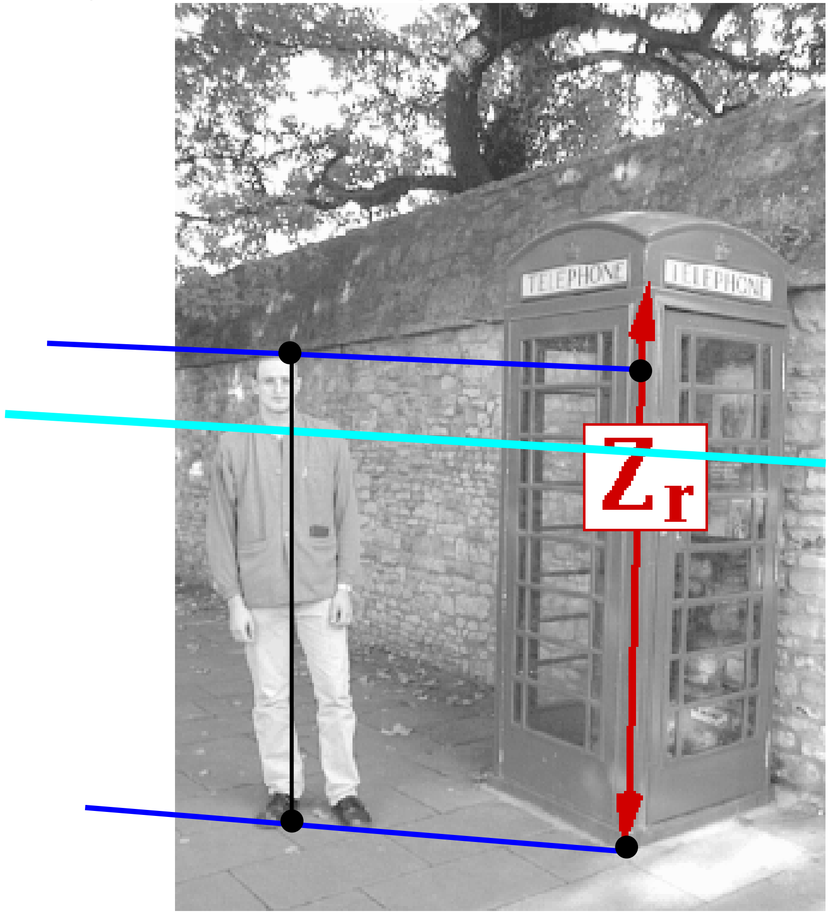
Measuring Height

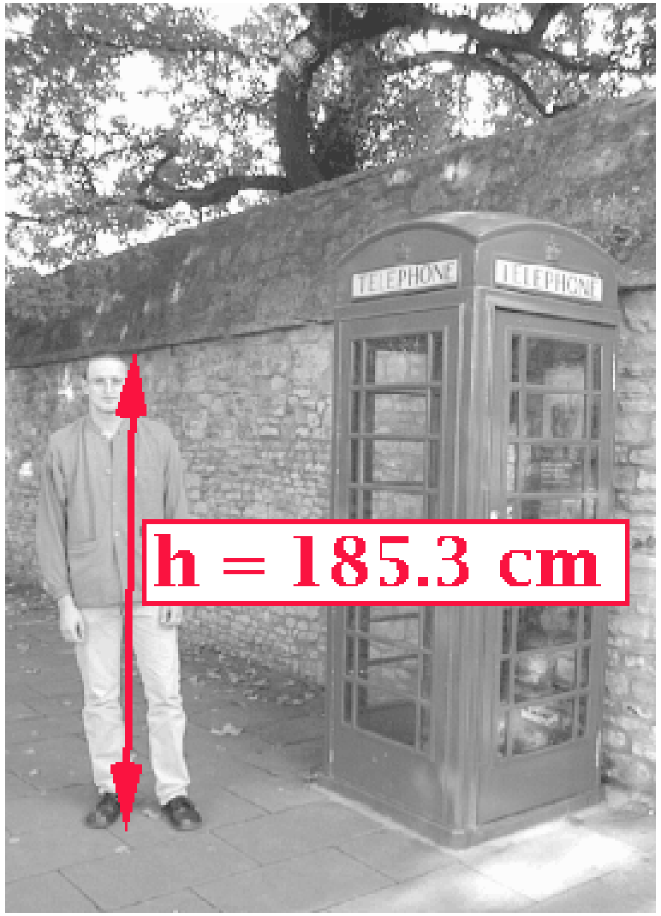


Example – height measurement

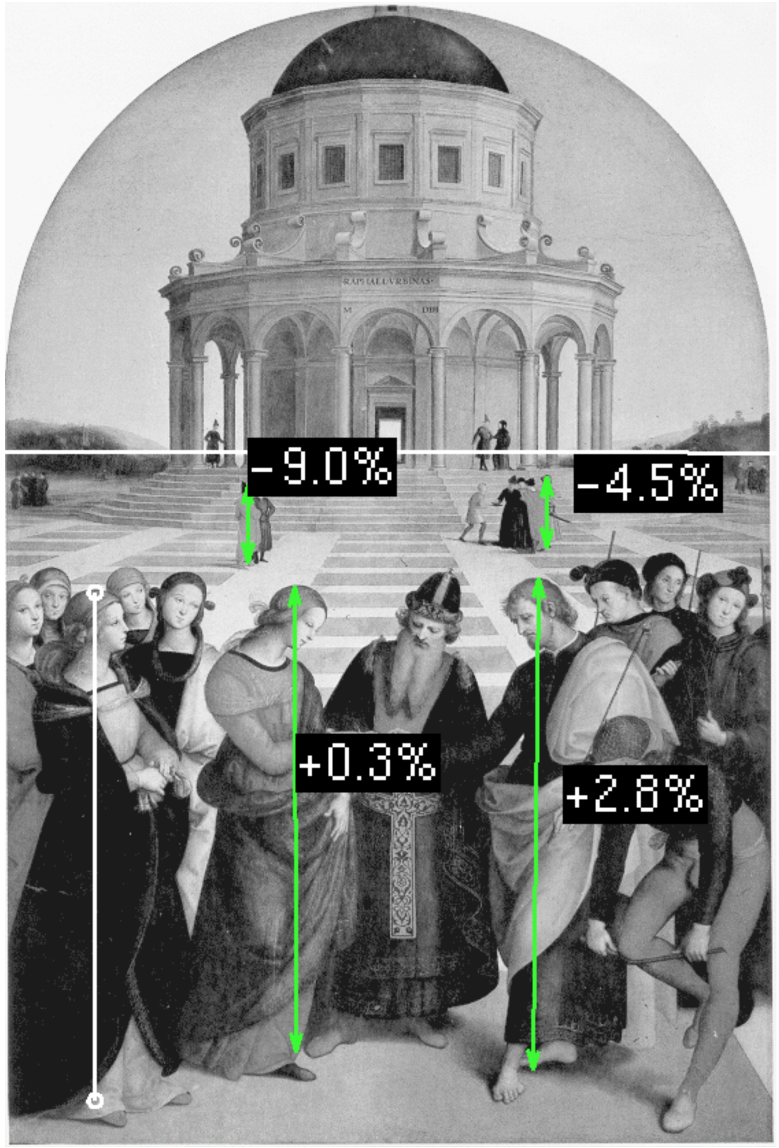


Example height measurement continued

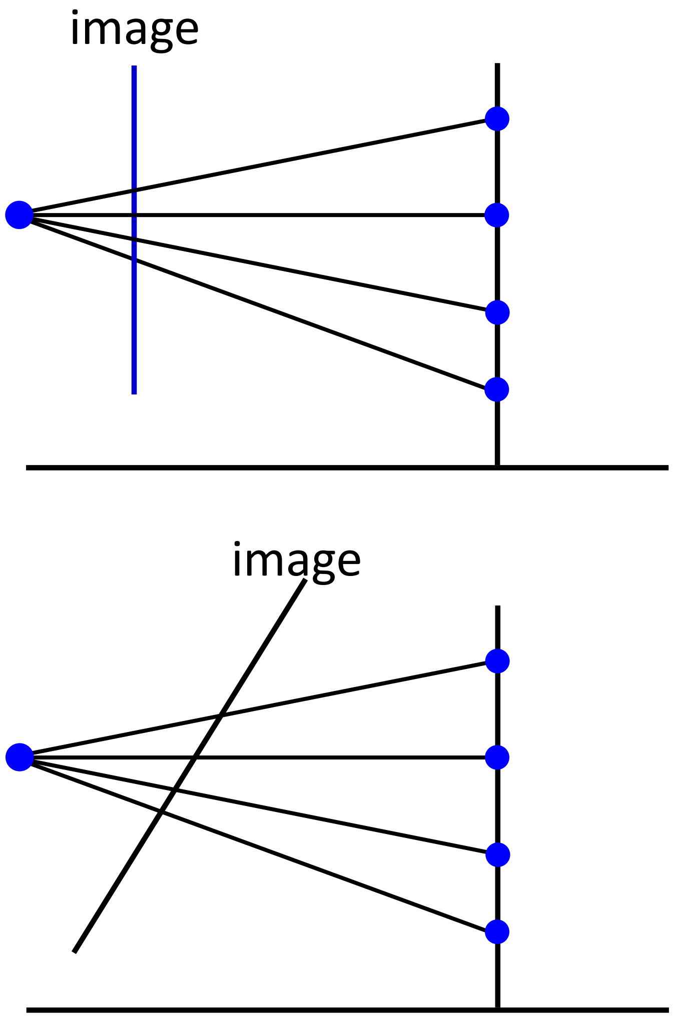




Relative Heights



The problem with lines not parallel to the image plane



Foreshortening.

Depends on the tilt of the camera relative to the horizon.

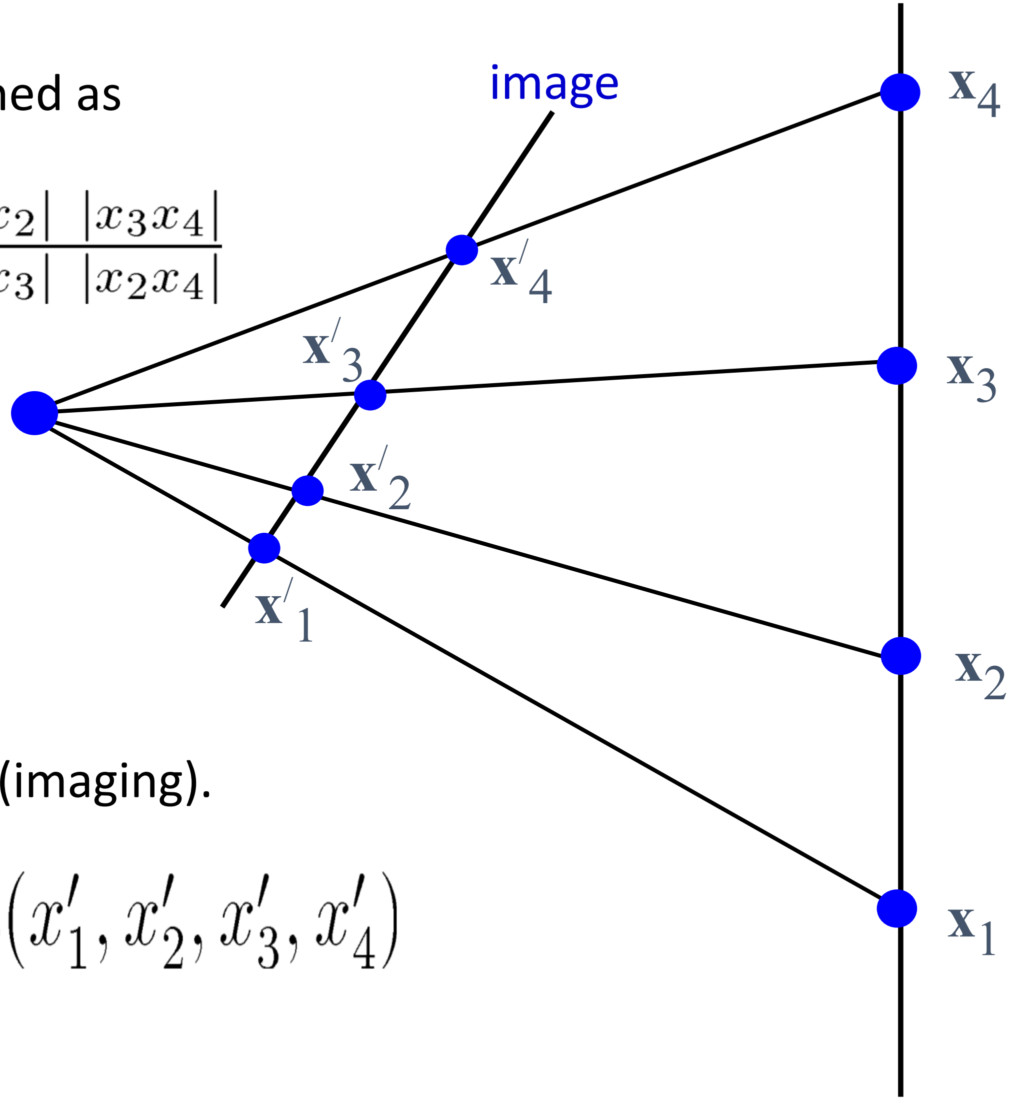
Computed using the cross-ratio perspective invariant and the vertical vanishing point.

The cross ratio

Cross ratio of 4 points on a line is defined as

$$\mathrm{Cross}(x_1, x_2, x_3, x_4) = \frac{|x_1 x_2|\ |x_3 x_4|}{|x_1 x_3|\ |x_2 x_4|}$$

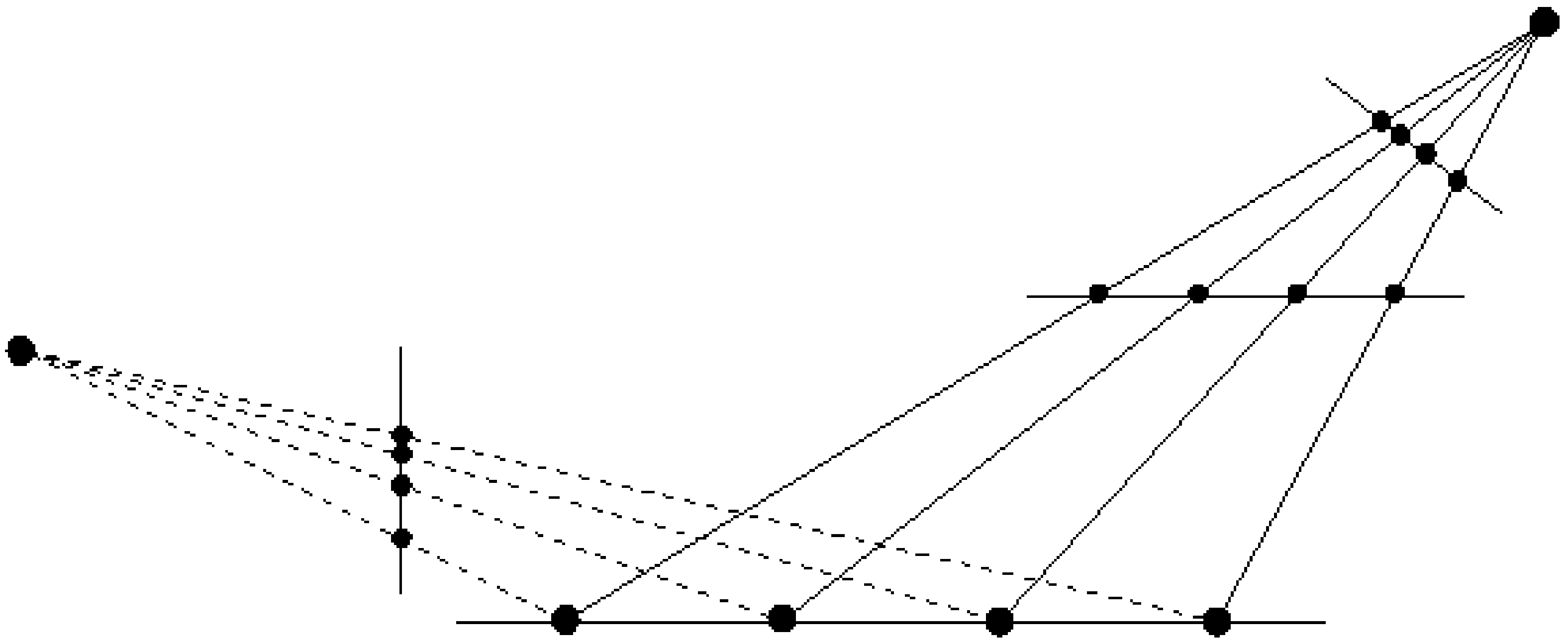
where $|x_i x_j| = \mathrm{distance}(x_i, x_j)$



Cross ration is invariant to projection (imaging).

$$\mathrm{Cross}(x_1, x_2, x_3, x_4) = \mathrm{Cross}(x'_1, x'_2, x'_3, x'_4)$$

more cross ratios



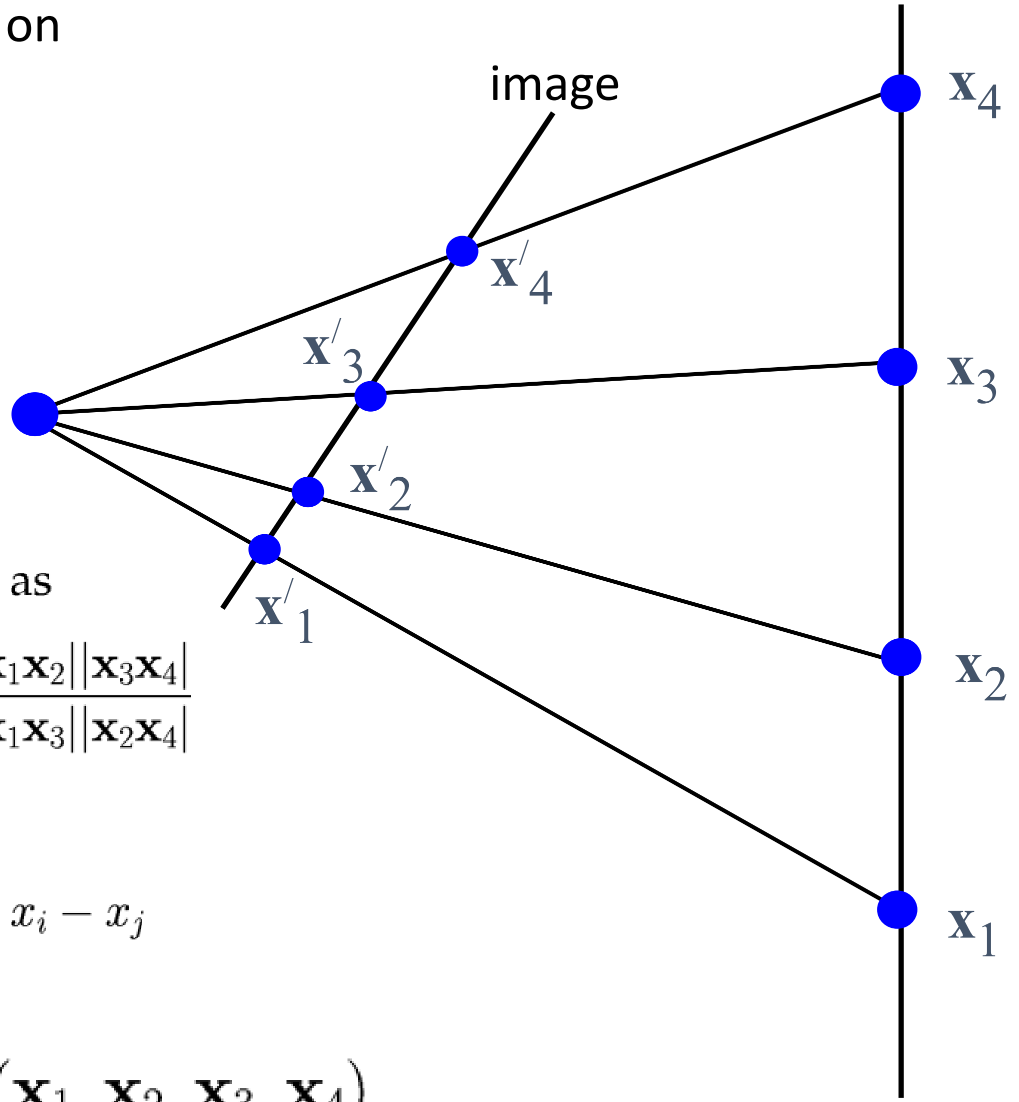
The cross-ratio has the same value for all of these sets of four collinear points

The cross ratio

Under perspective projection, points on a line are related by

$$\mathbf{x}' = \mathtt{H}_{2\times 2}\,\mathbf{x}$$

$$\begin{pmatrix} x' \\ 1 \end{pmatrix} = \lambda \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix}$$



Given 4 points $\mathbf{x}_i$ the *cross ratio* is defined as

$$\mathrm{Cross}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = \frac{|\mathbf{x}_1\mathbf{x}_2||\mathbf{x}_3\mathbf{x}_4|}{|\mathbf{x}_1\mathbf{x}_3||\mathbf{x}_2\mathbf{x}_4|}$$

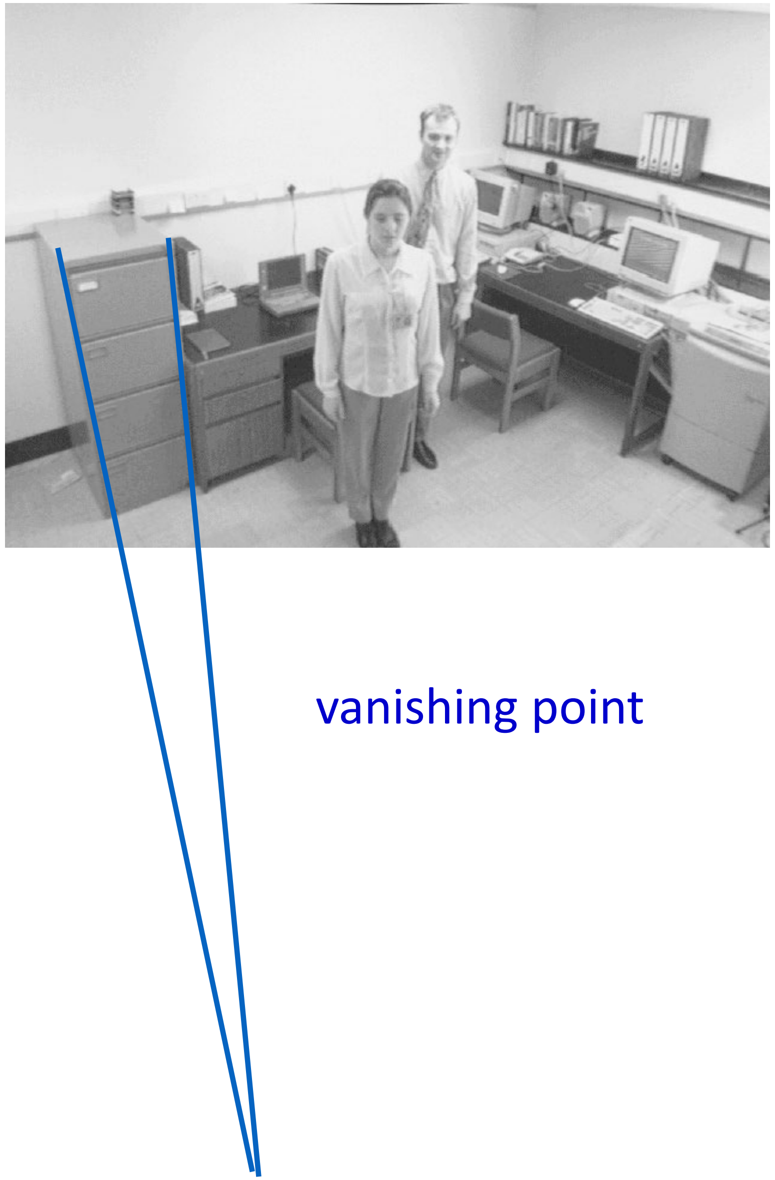
where

$$|\mathbf{x}_i\mathbf{x}_j| = \det \begin{bmatrix} x_i & x_j \\ 1 & 1 \end{bmatrix} = x_i - x_j$$

$$\mathrm{Cross}(\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3, \mathbf{x}'_4) = \mathrm{Cross}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$$

Example

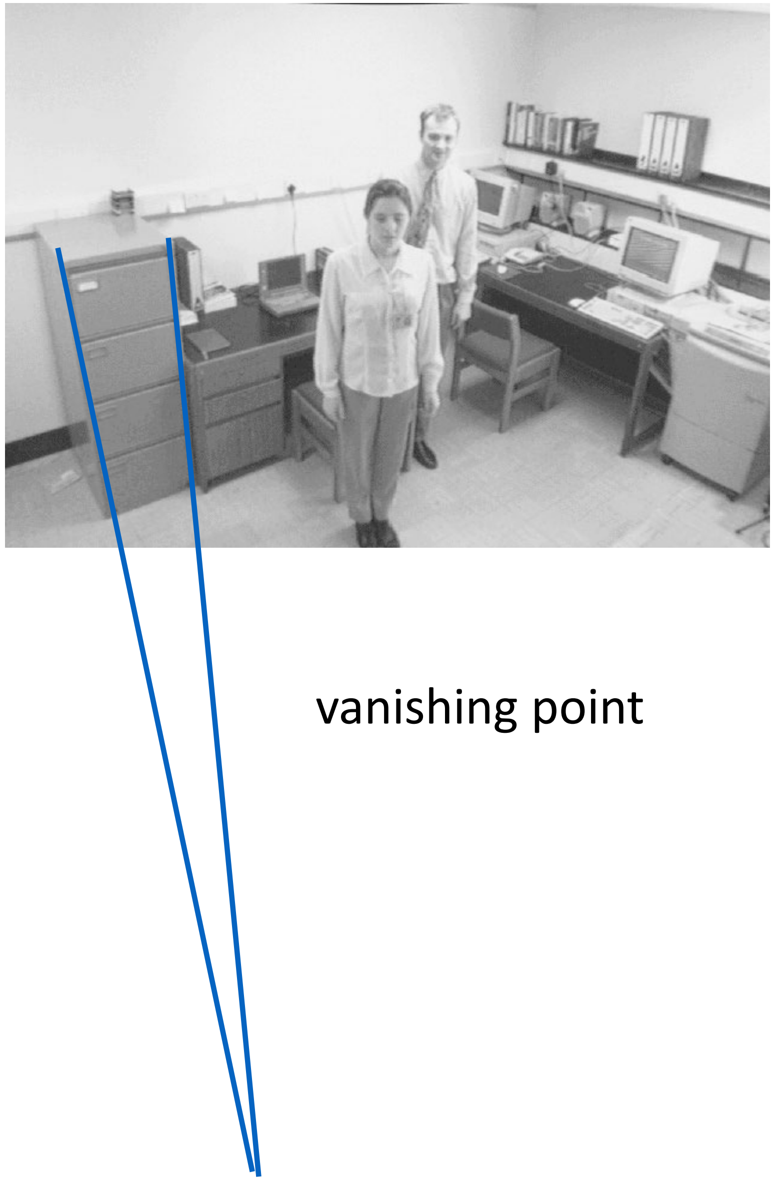
vanishing line



vanishing point


189.7 cm
169.8 cm

vanishing line



vanishing point


189.7 cm
169.8 cm

Case of Terrick Nooner
Arkansas, 1993.

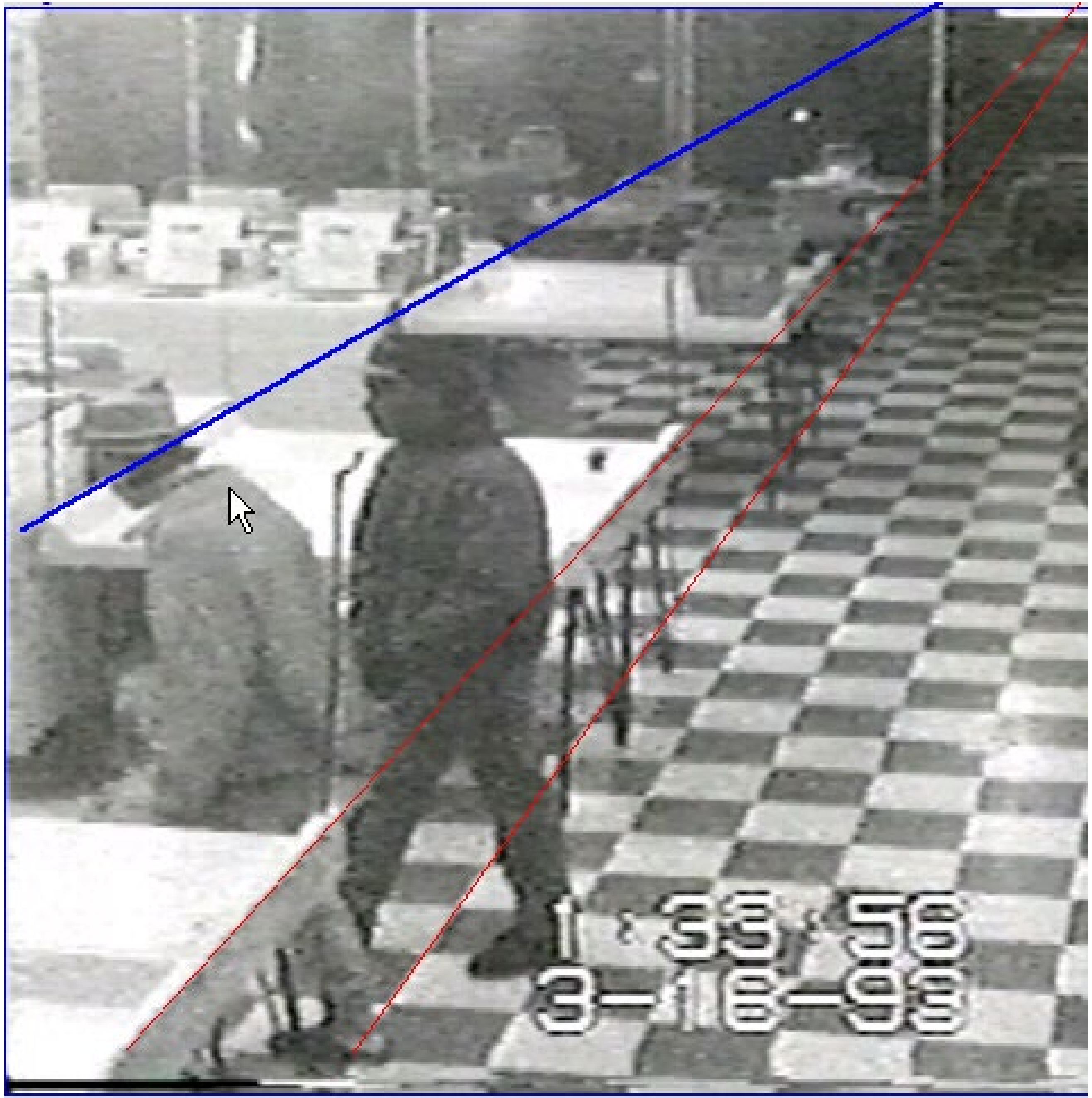
Image 55 – digitized with 9:8
aspect ratio

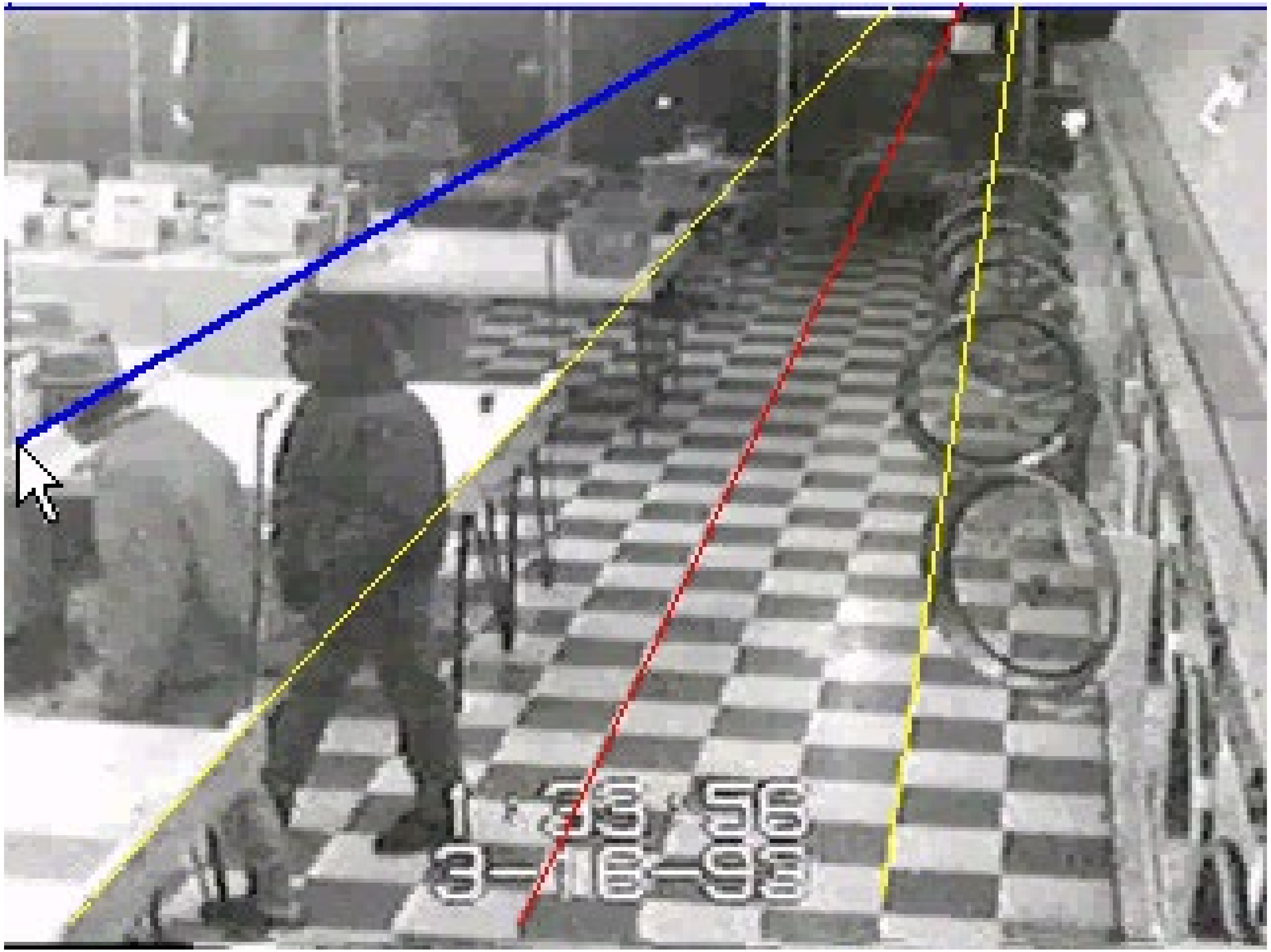
Image 56 – most useful image

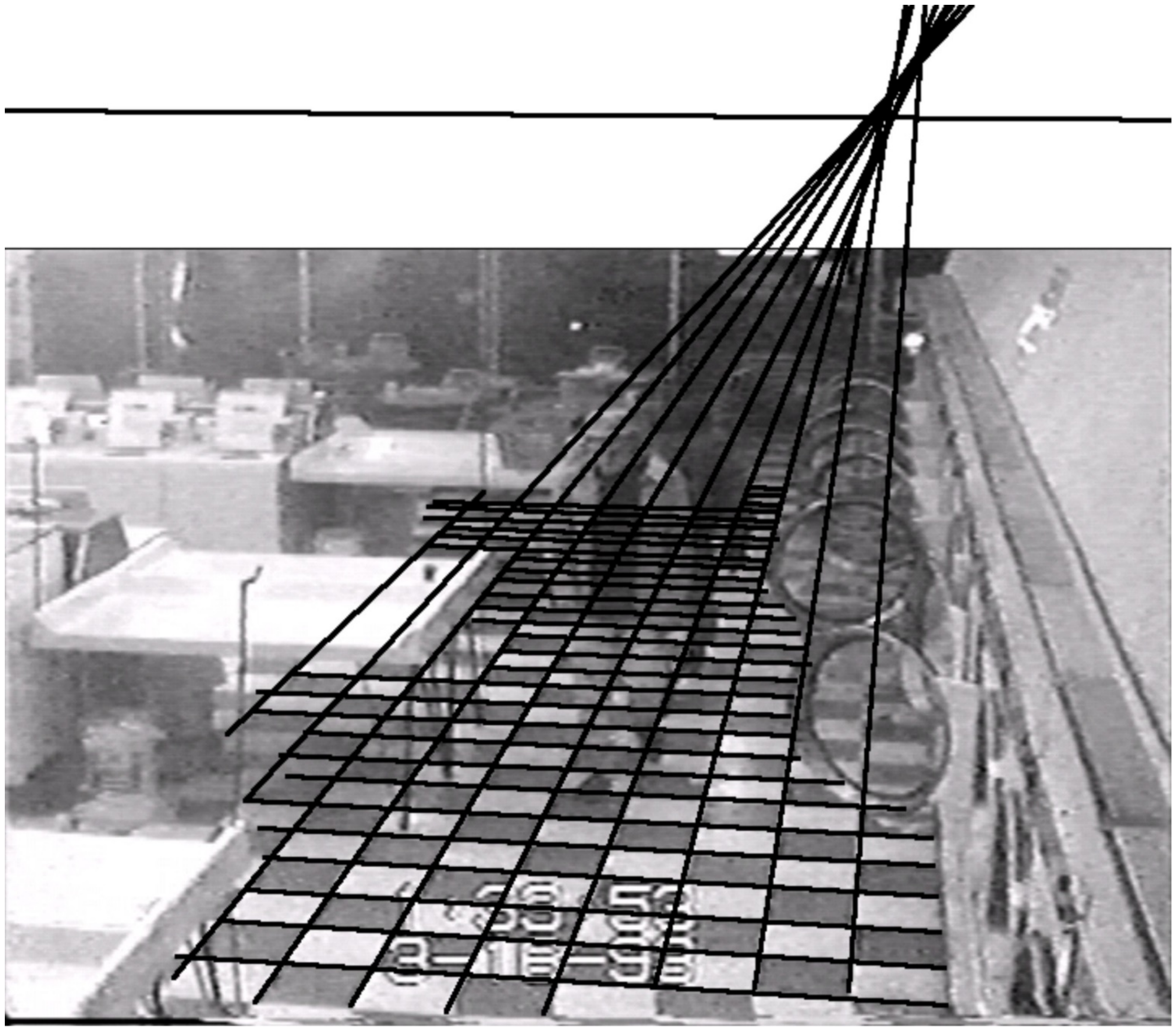
Image 57
almost identical with image 56

Images digitized with 1:1 aspect ratio pixels

Vanishing Lines

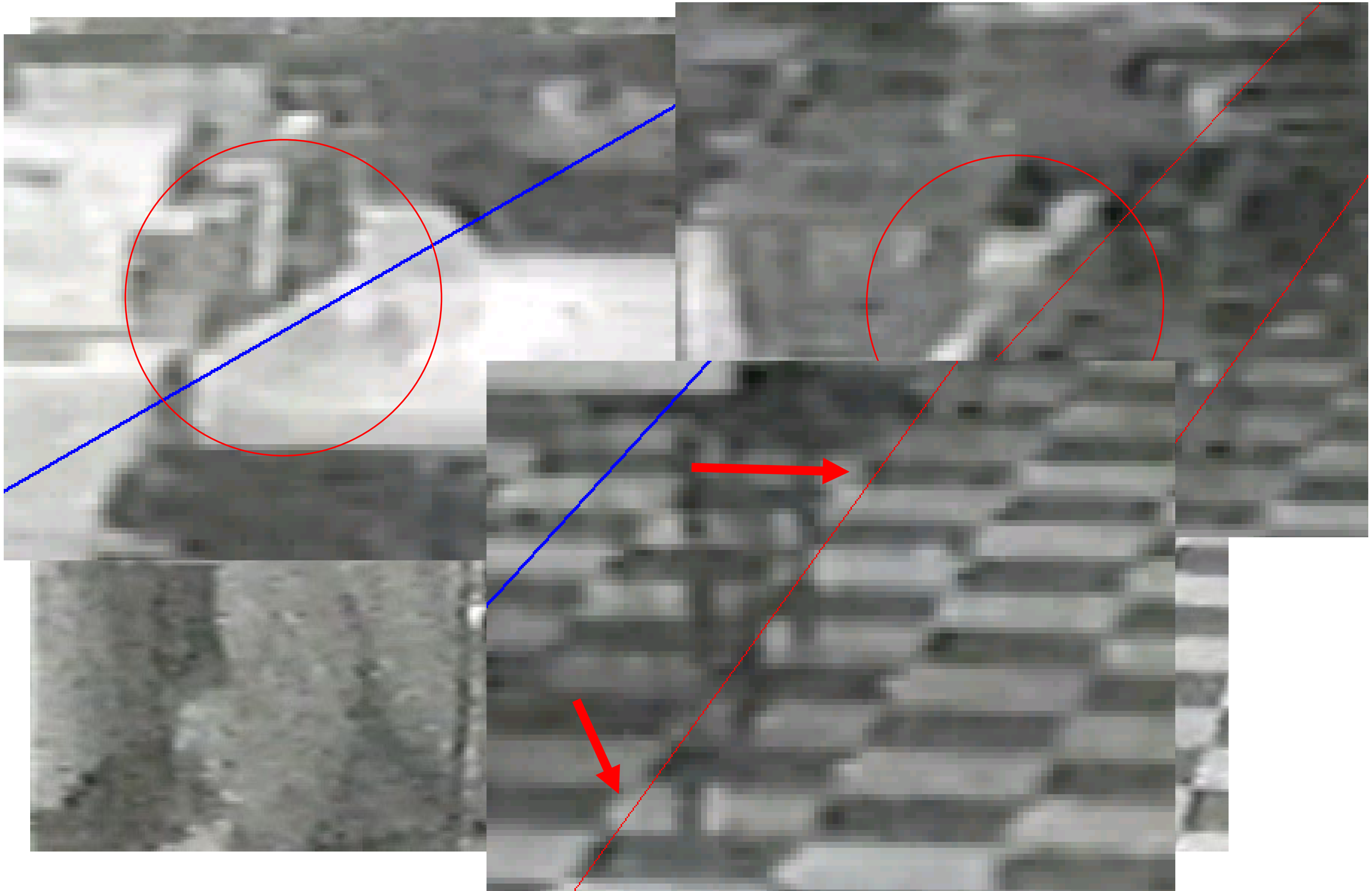

1:33:56
3-16-93


1:33:56
3-16-93



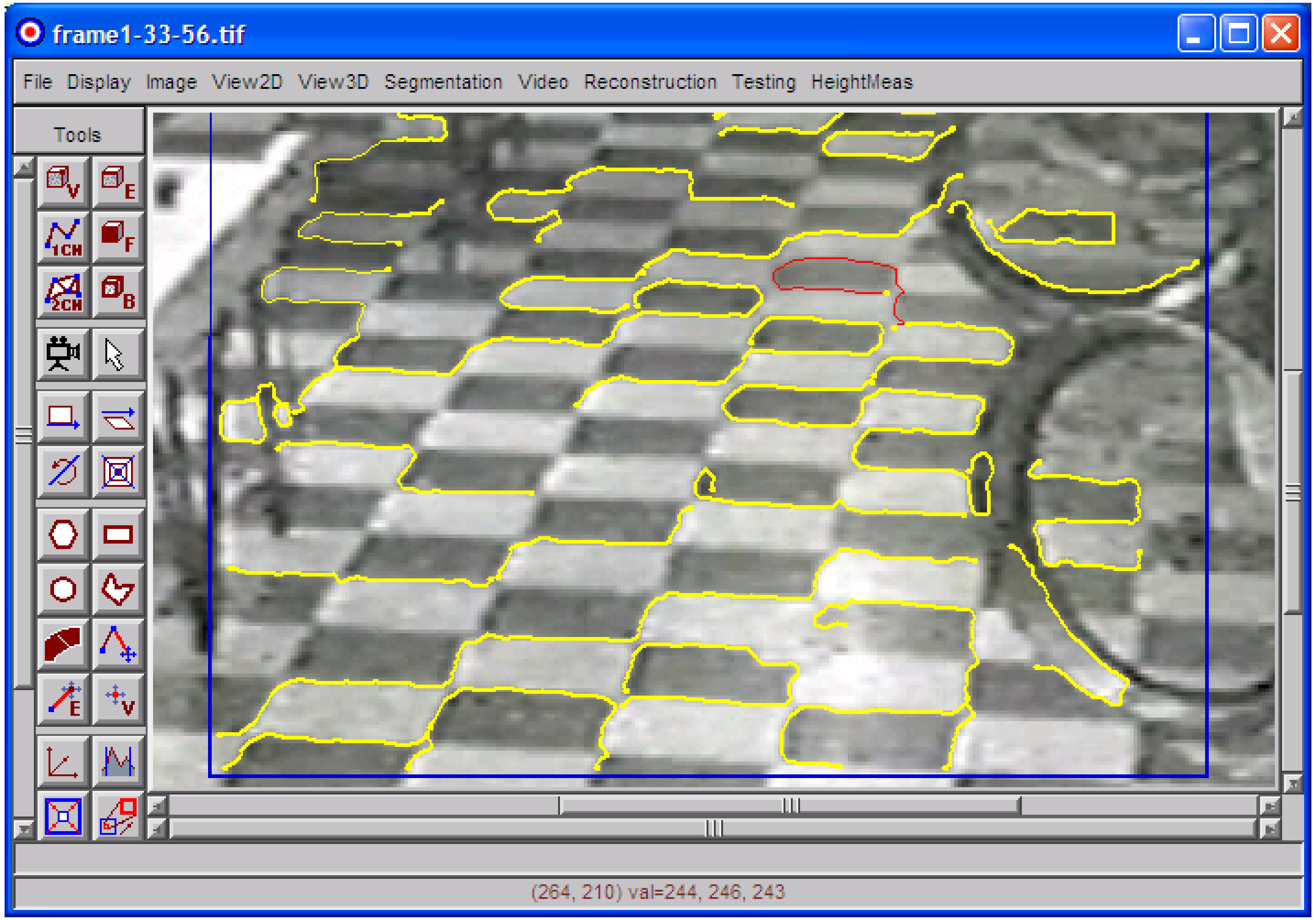
Results of hand drawn lines (uncorrected)

Effect of mistake by 10 pixels in the horizon
(equivalent to 1.1 inches increase in subject height)

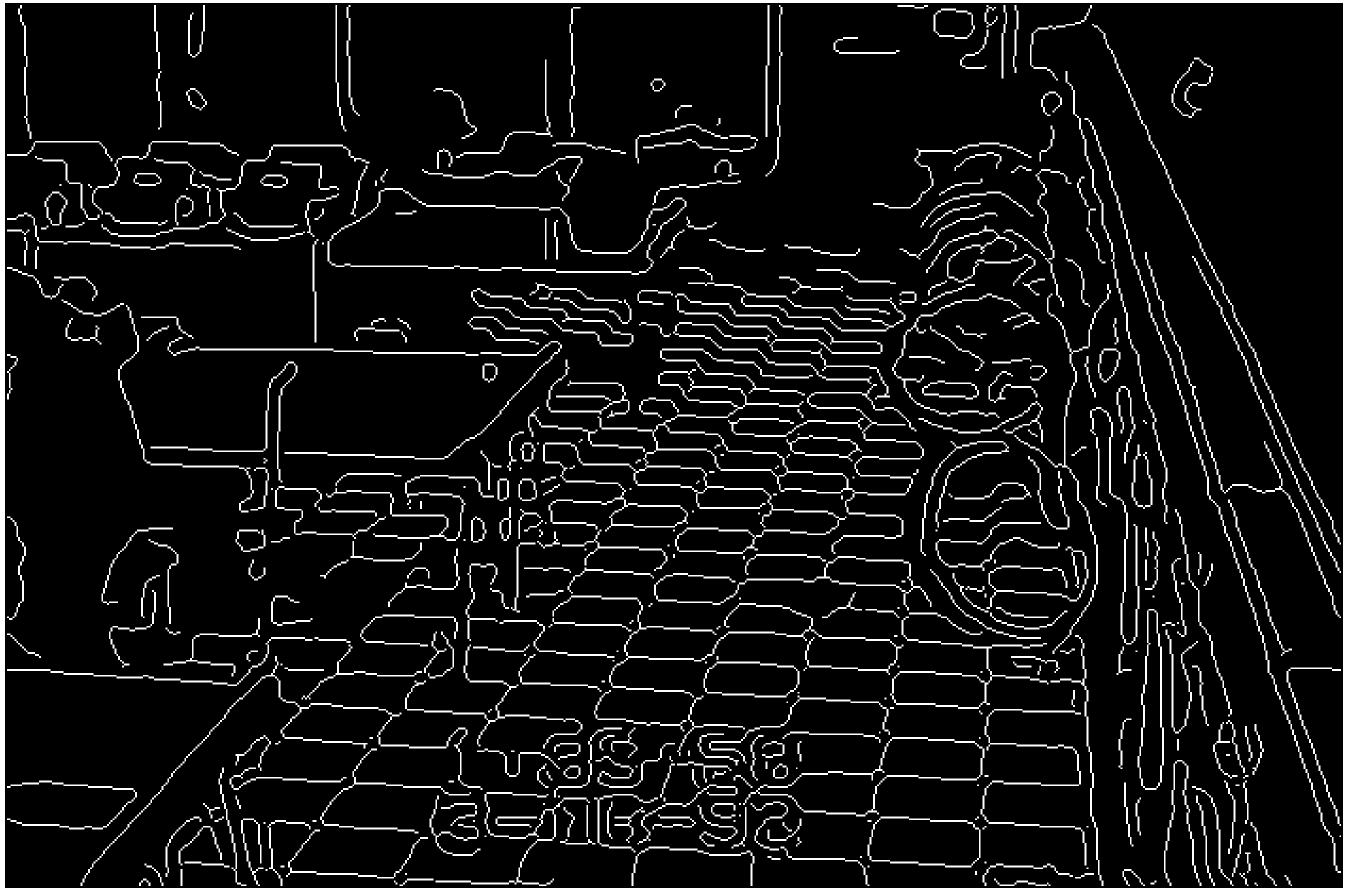


Automatic extraction of lines on the floor

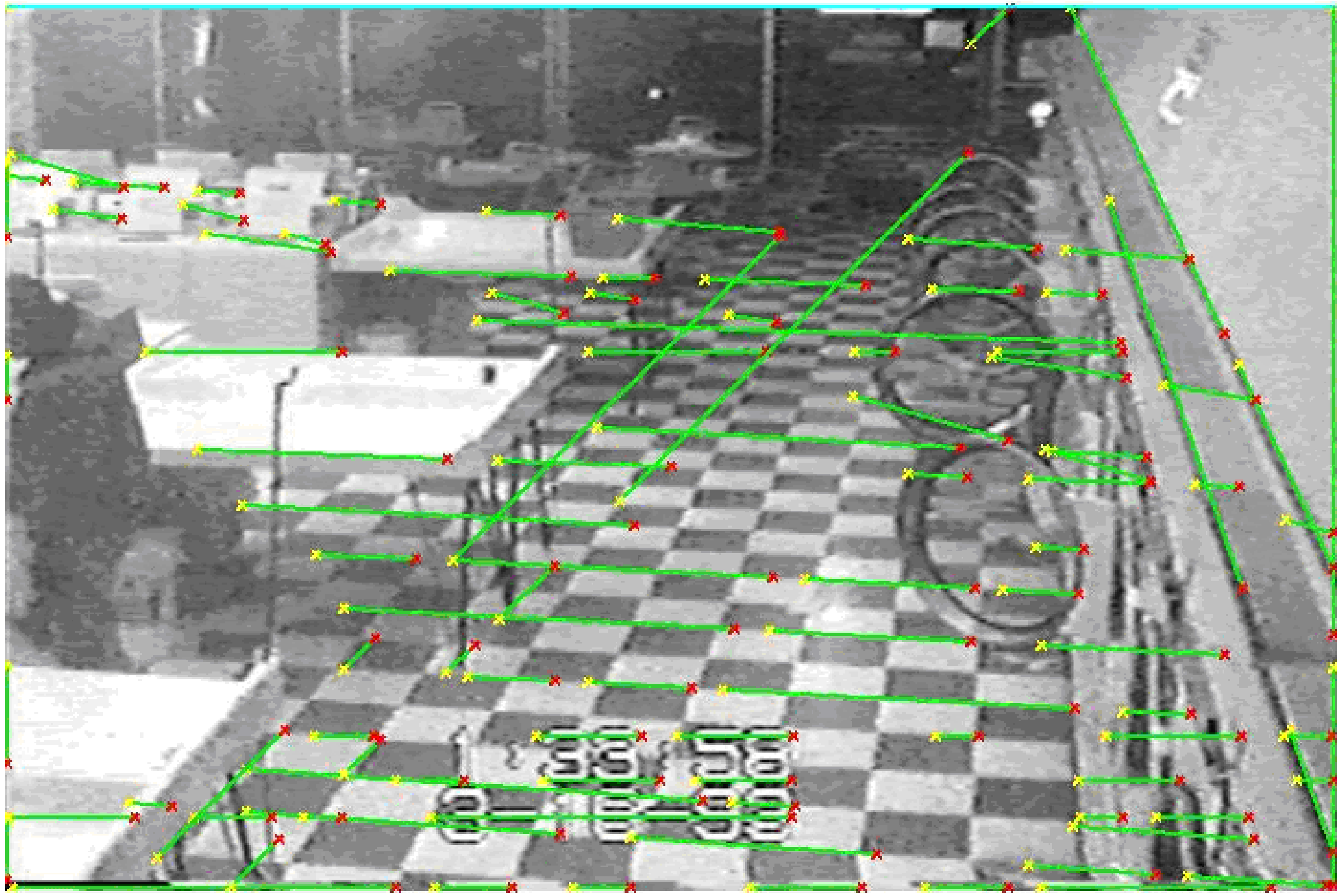
Edges detected using Criminisi tool (OVM)



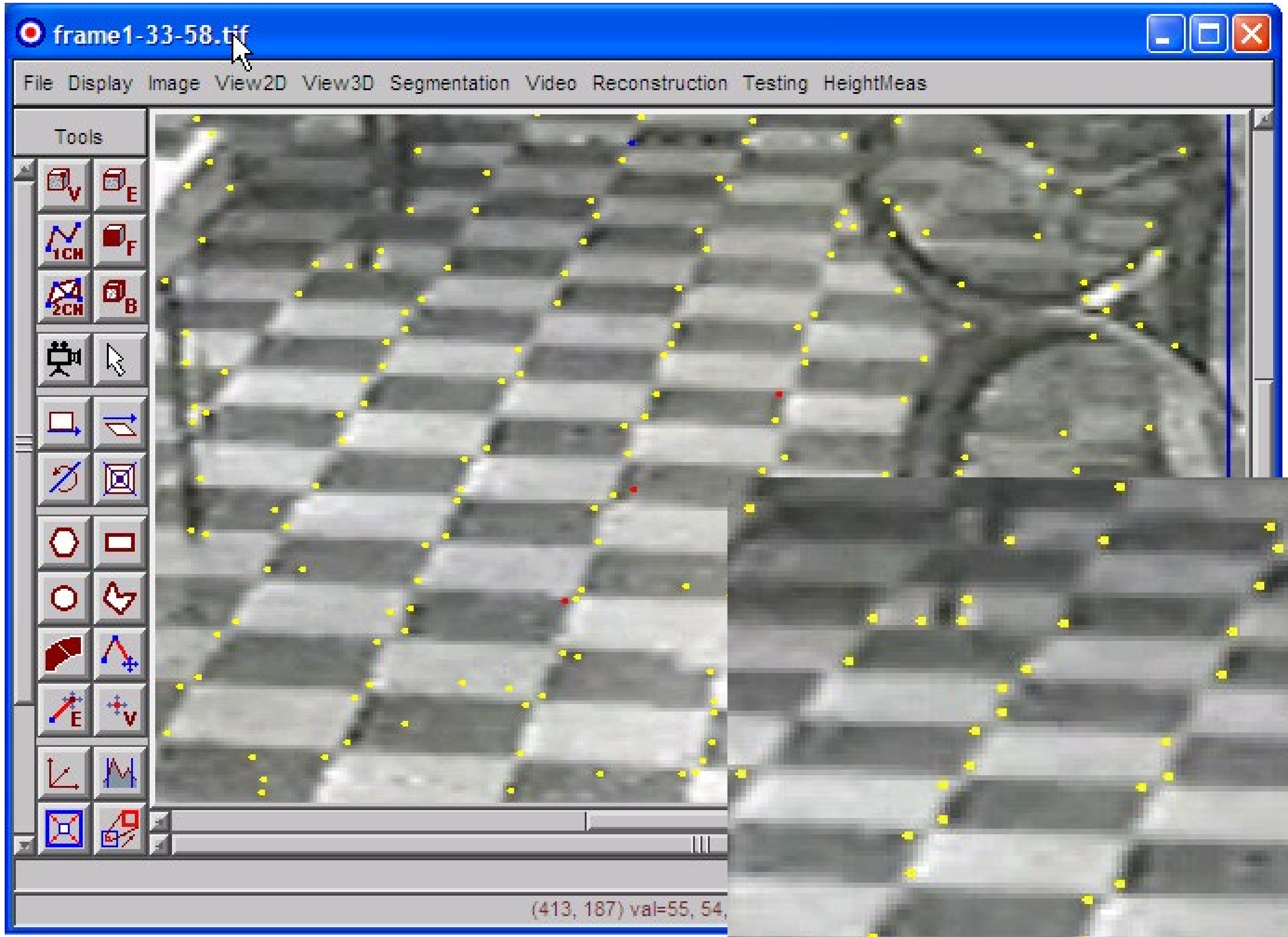
Canny edges from Targetjr

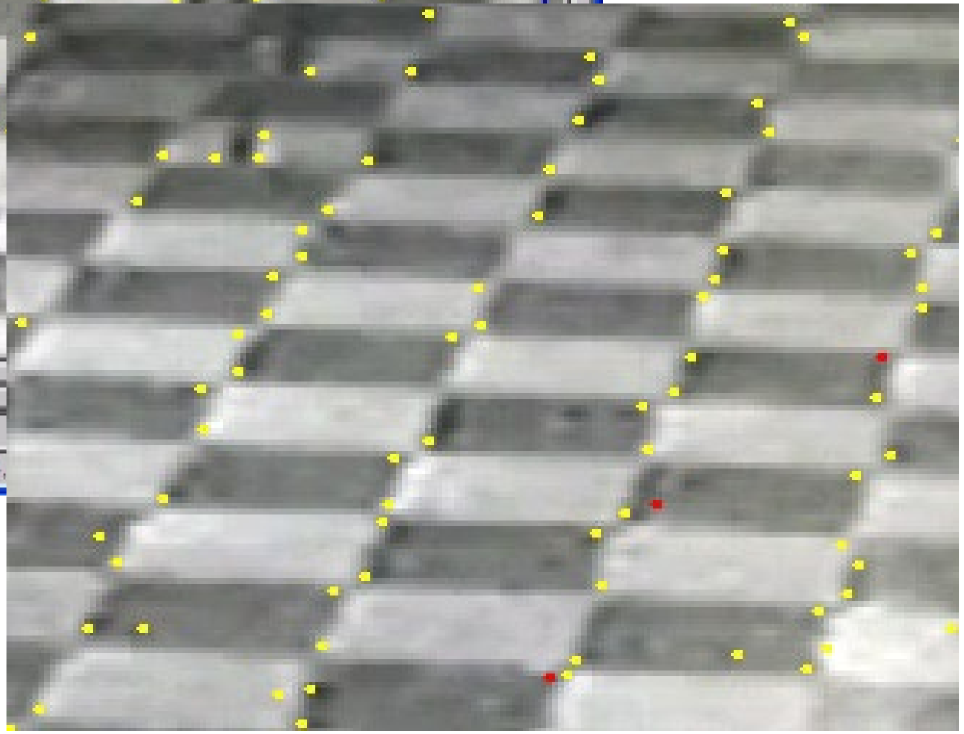


Result of Canny edge detection (Matlab)



Hough line detection on edge image

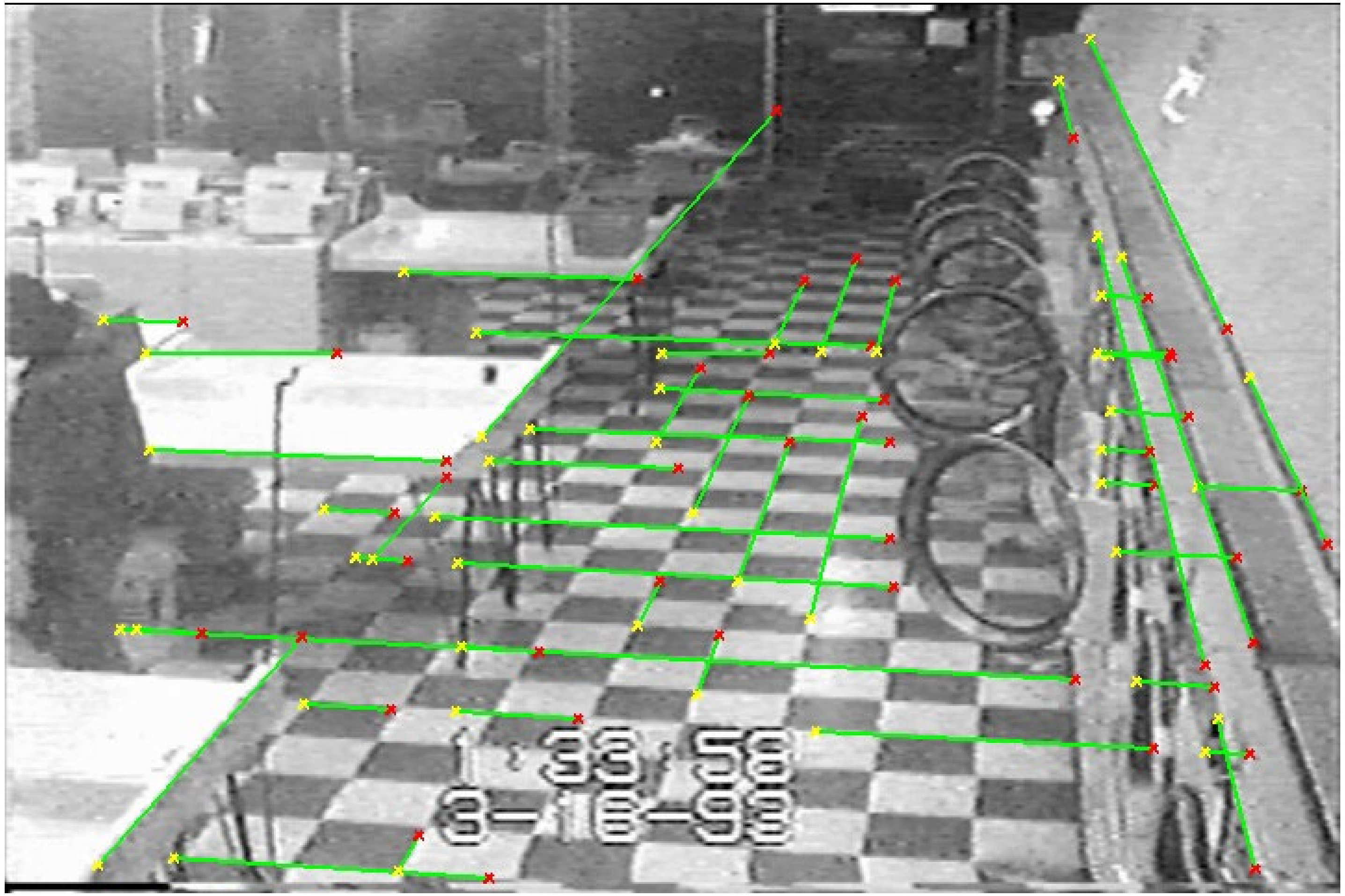




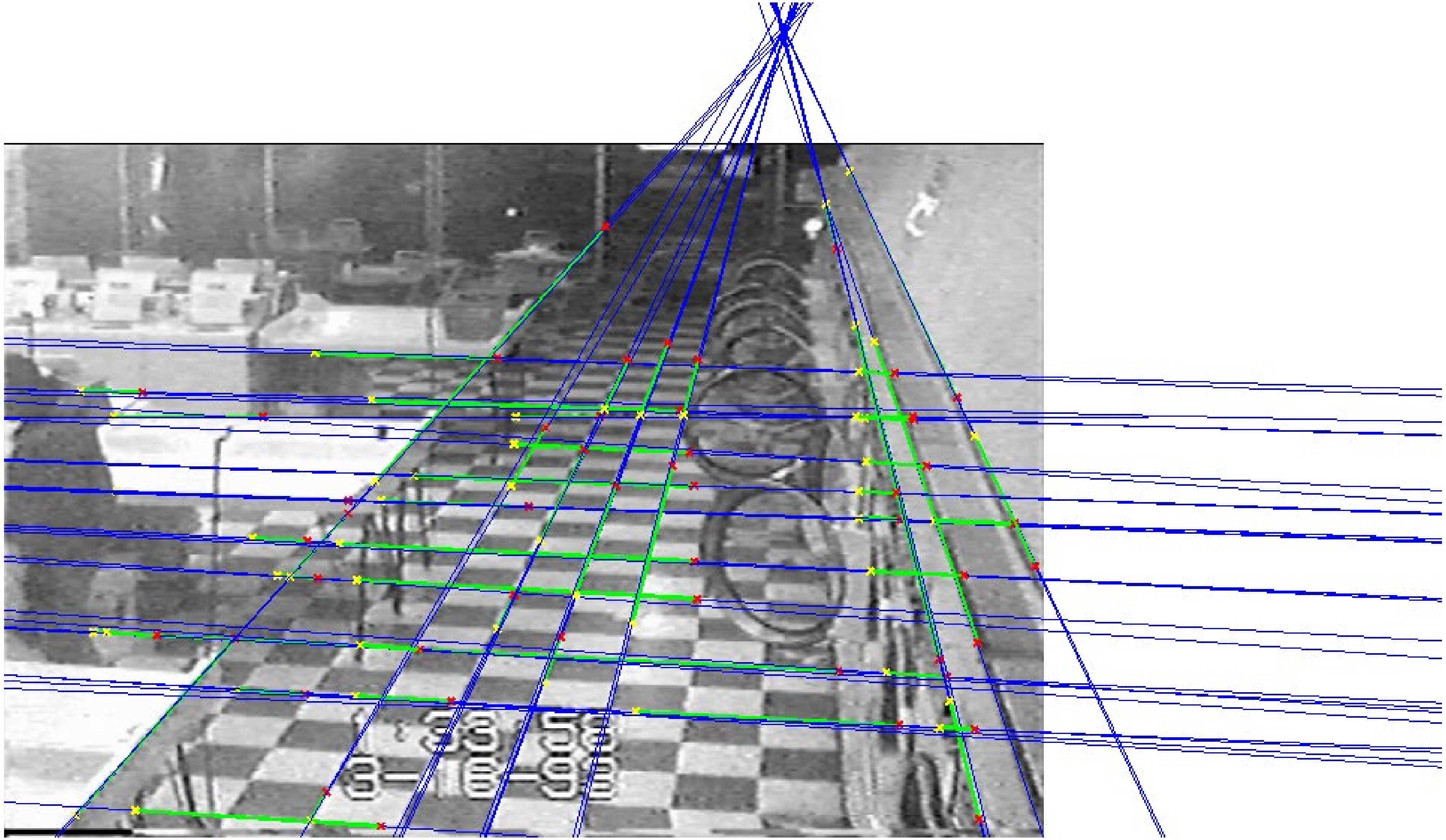
Harris Corner detector results

Corners detected using

Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support” (by X.C. He and N.H.C. Yung). ICPR 2004



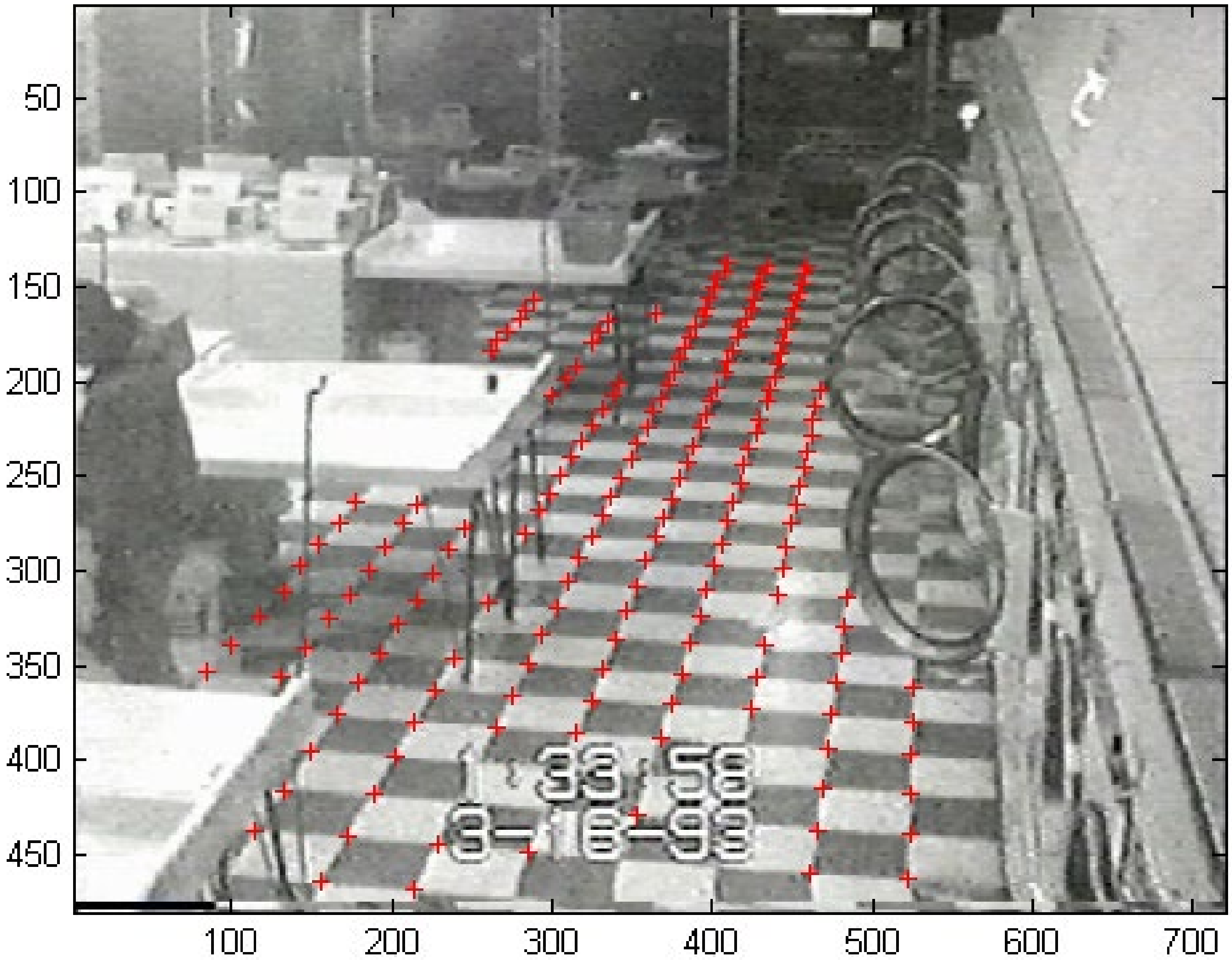
Result of Hough transform on lines and points

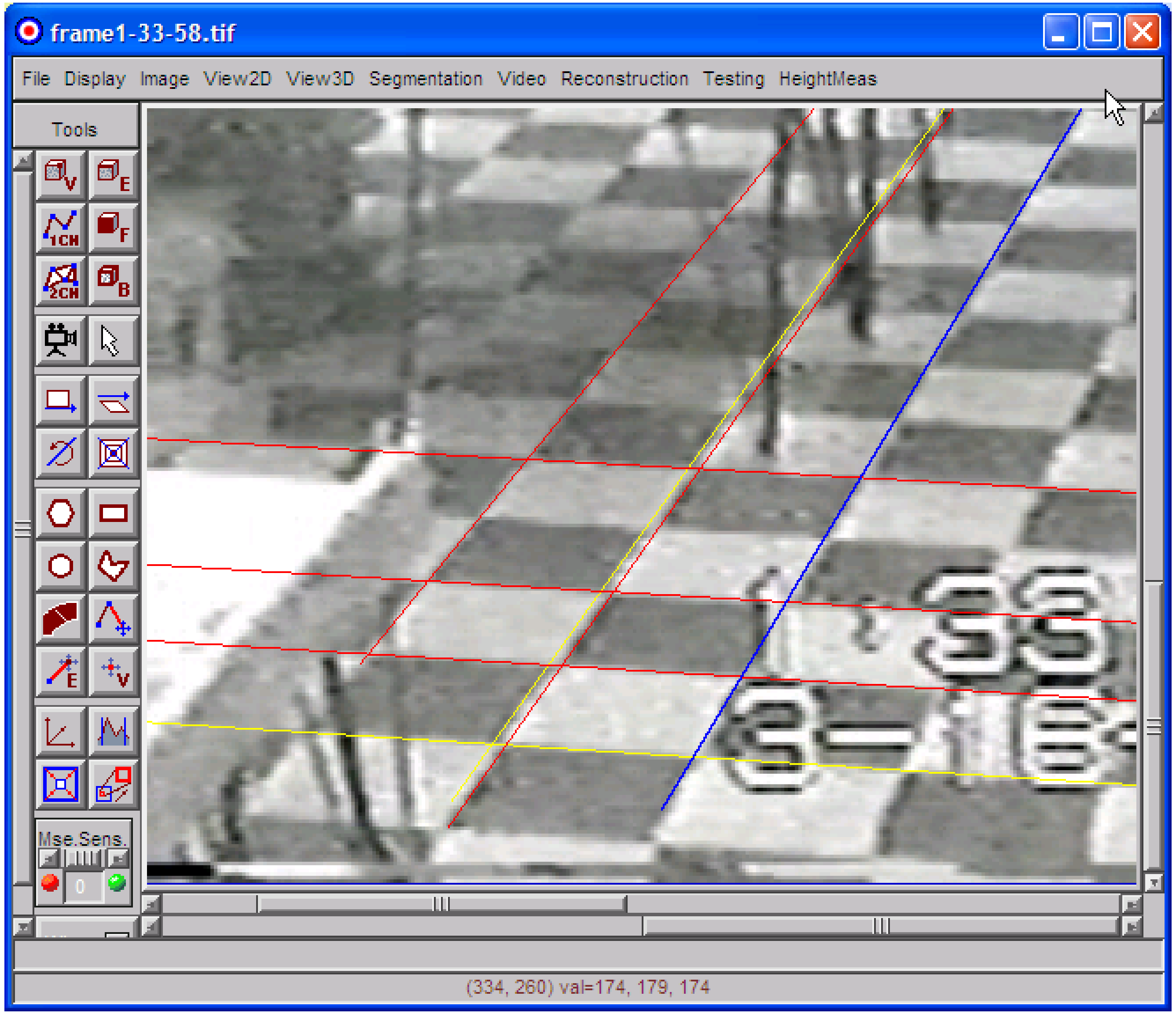


Best achieved result using automatic tools

Effect of 20 pixels error in horizon (equivalent to 2.2 inches in subject height estimate)

Large-scale bundle adjustment.
Fit a model to the extracted Harris corners.


50
100
150
200
250
300
350
400
450
100
200
300
400
500
600
700


frame1-33-58.tif
File Display Image View2D View3D Segmentation Video Reconstruction Testing HeightMeas
Tools
Mse.Sens.
(334, 260) val=174, 179, 174

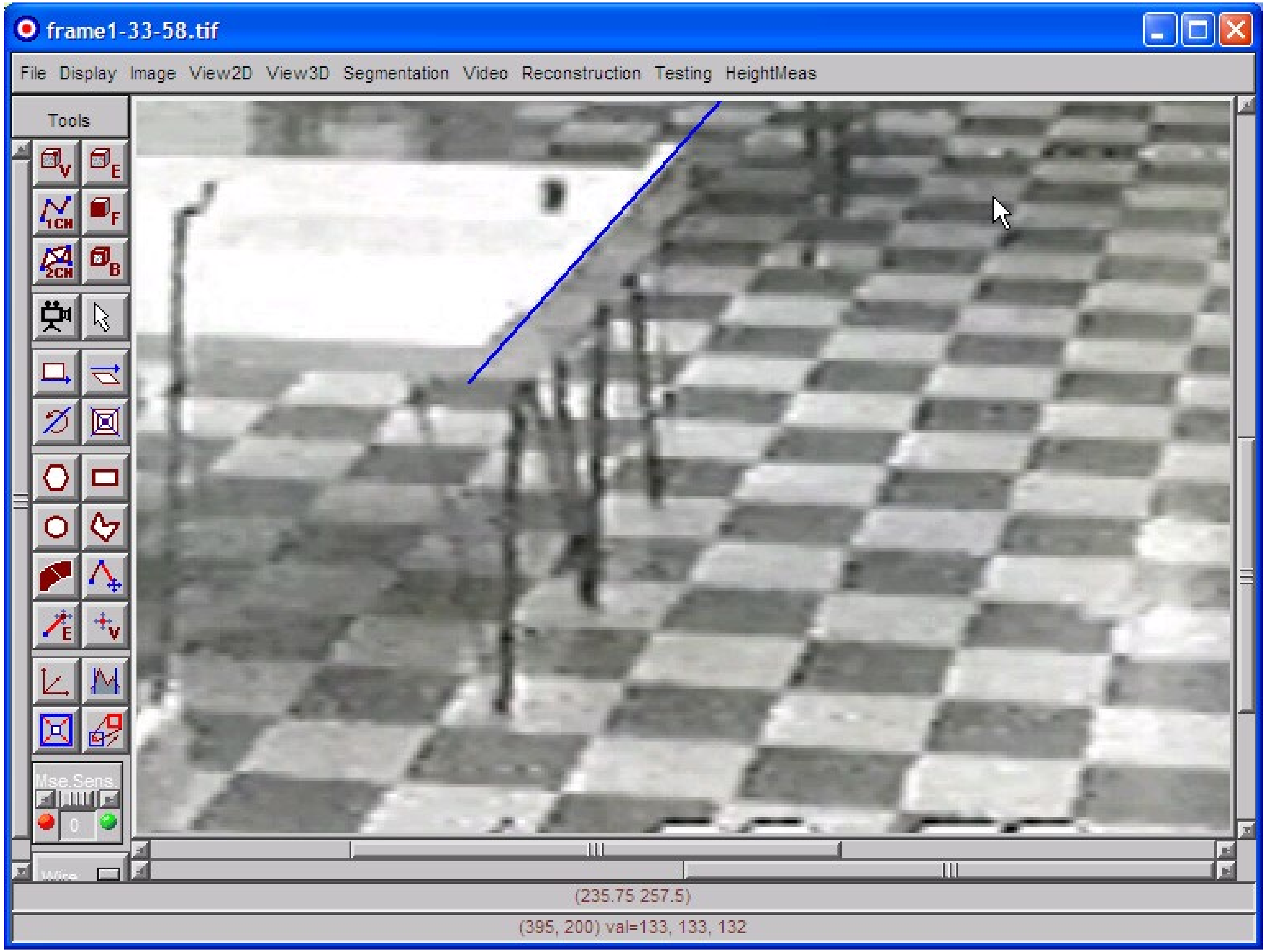
Error analysis of the horizon measurement

- Horizon is estimated by fitting model to all the measurements at once.
    - Standard paradigm: fit model to data.
- Model is a checkerboard grid representing lines on the tiled floor.
- Data is the marked lines in the image.
- Model fits the data with an average (RMS) error of ~0.75 pixels.
- Accuracy of the fit (standard deviation of error) gives a measure of the accuracy of the model.

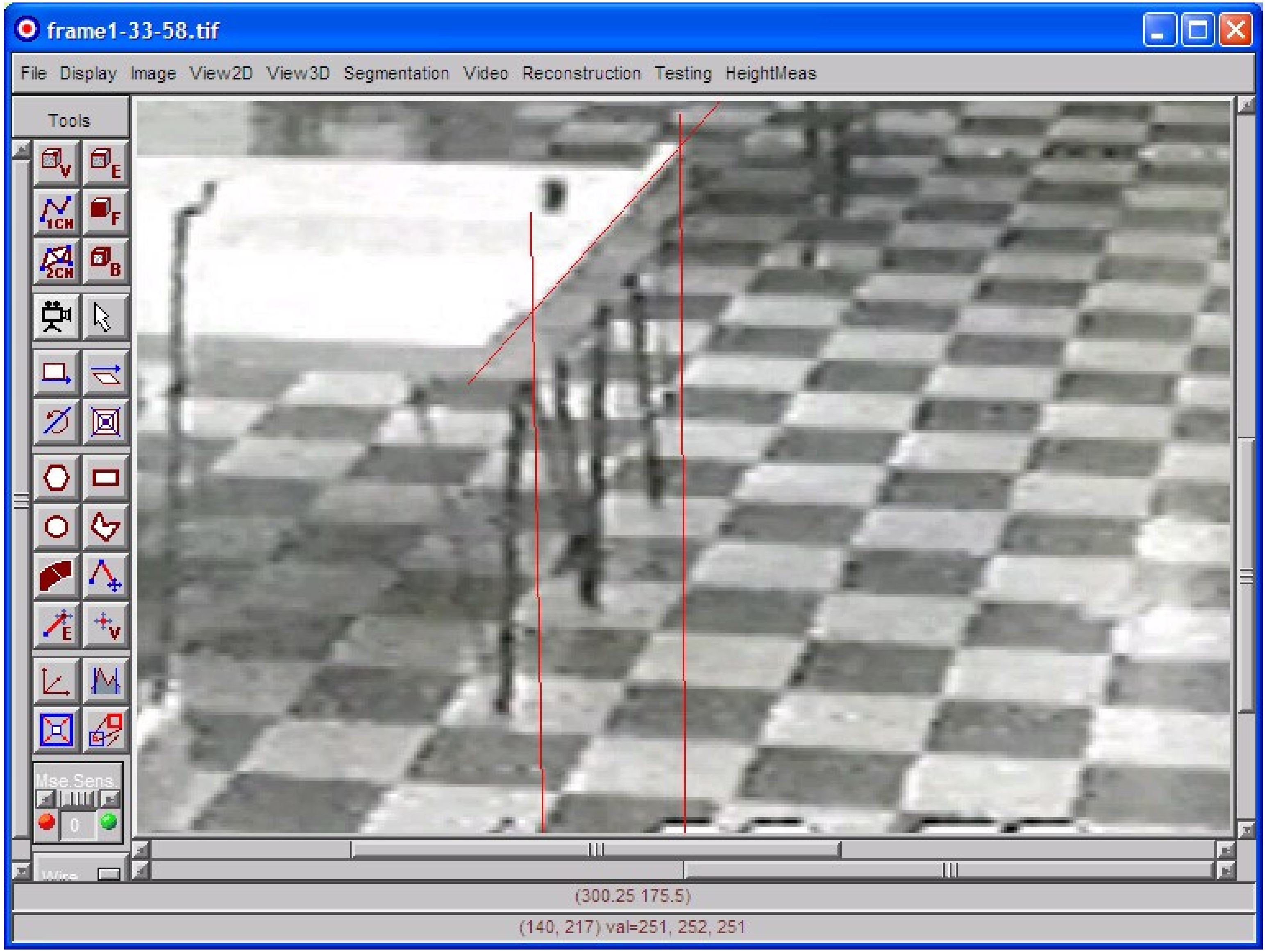
Image	RMS error
53	0.79
54	0.87
55	0.75
56	0.64
57	0.80
58	0.73

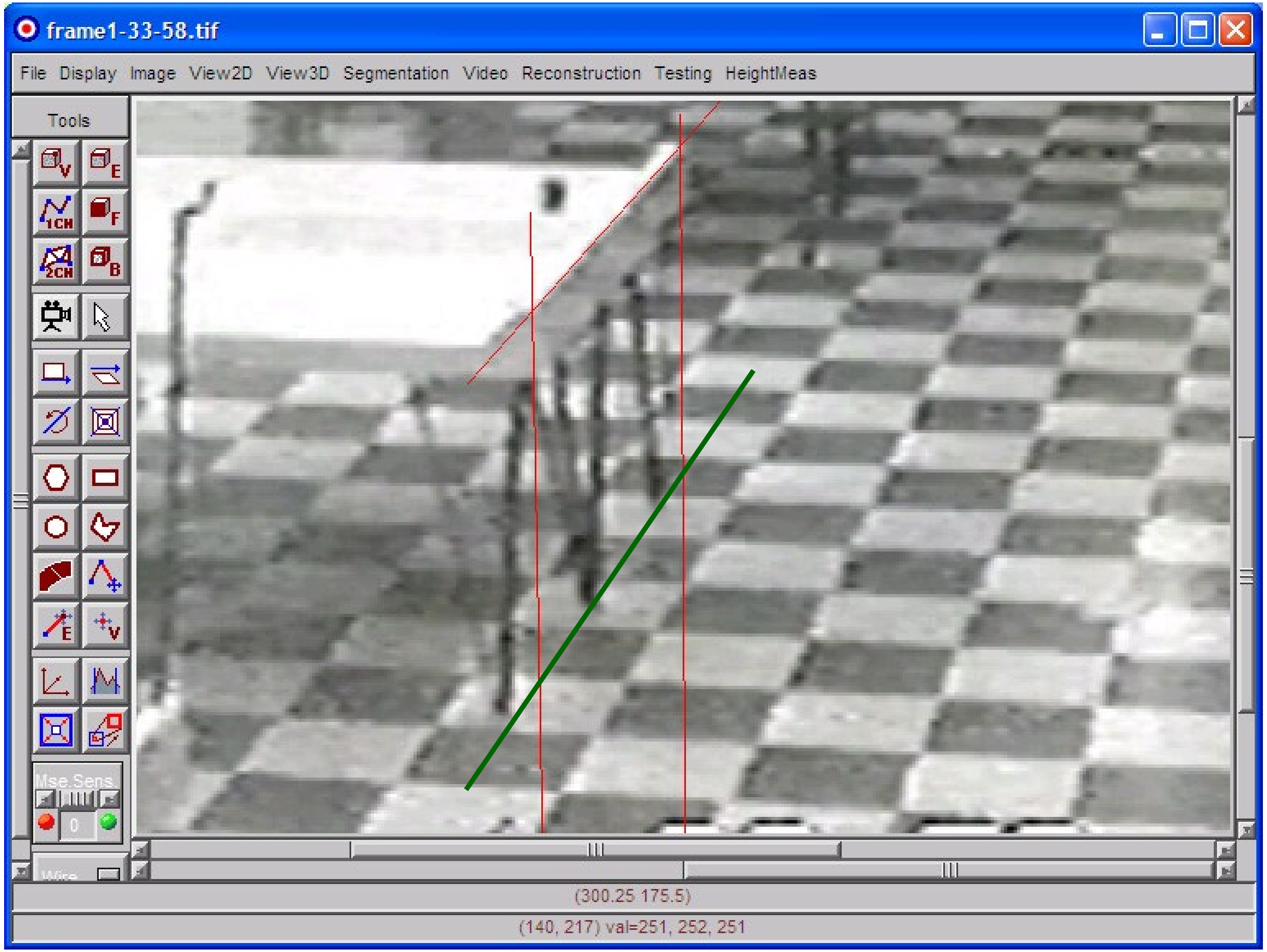
Height of reference object.

Using Table as a Reference

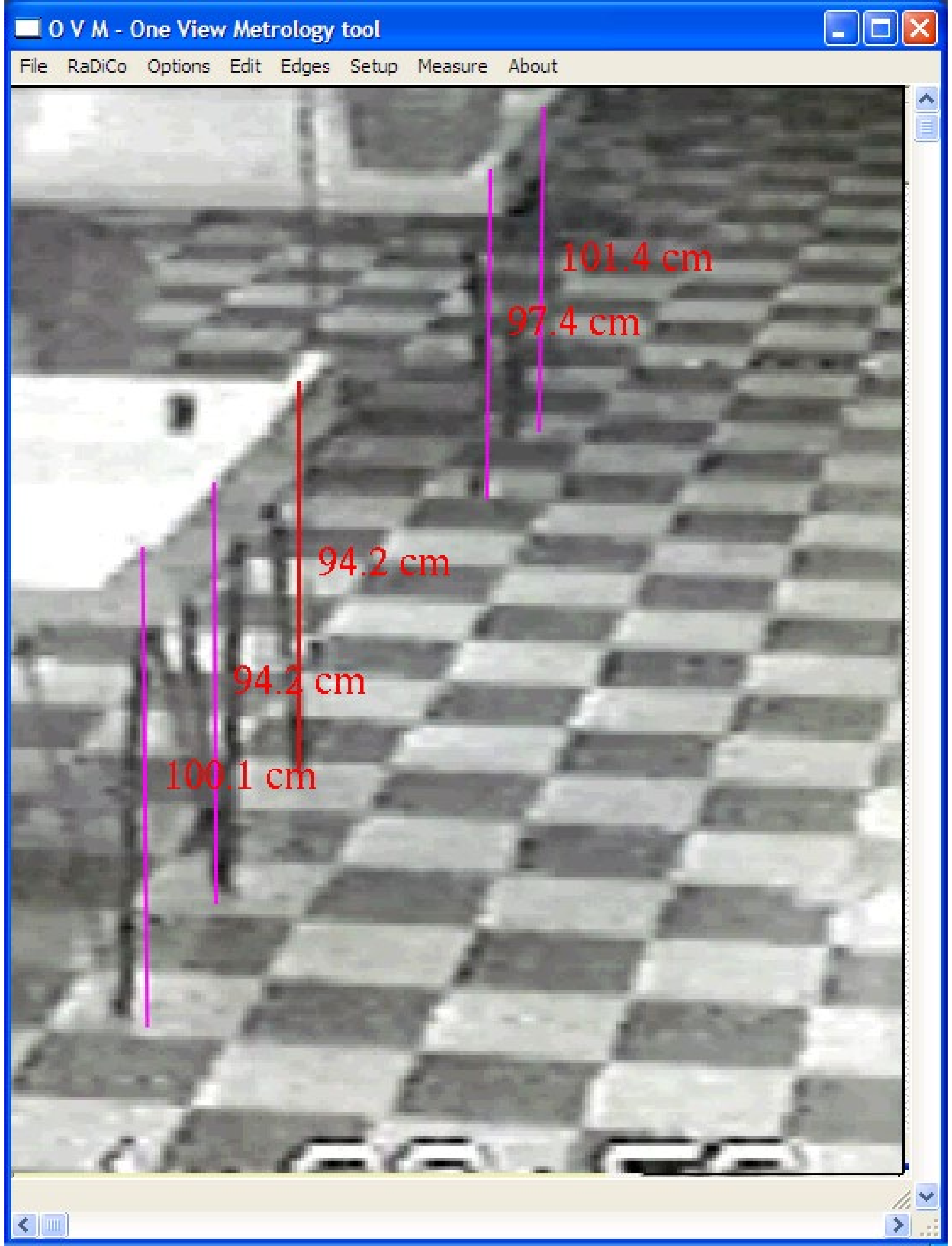


Draw the line of the top of the table

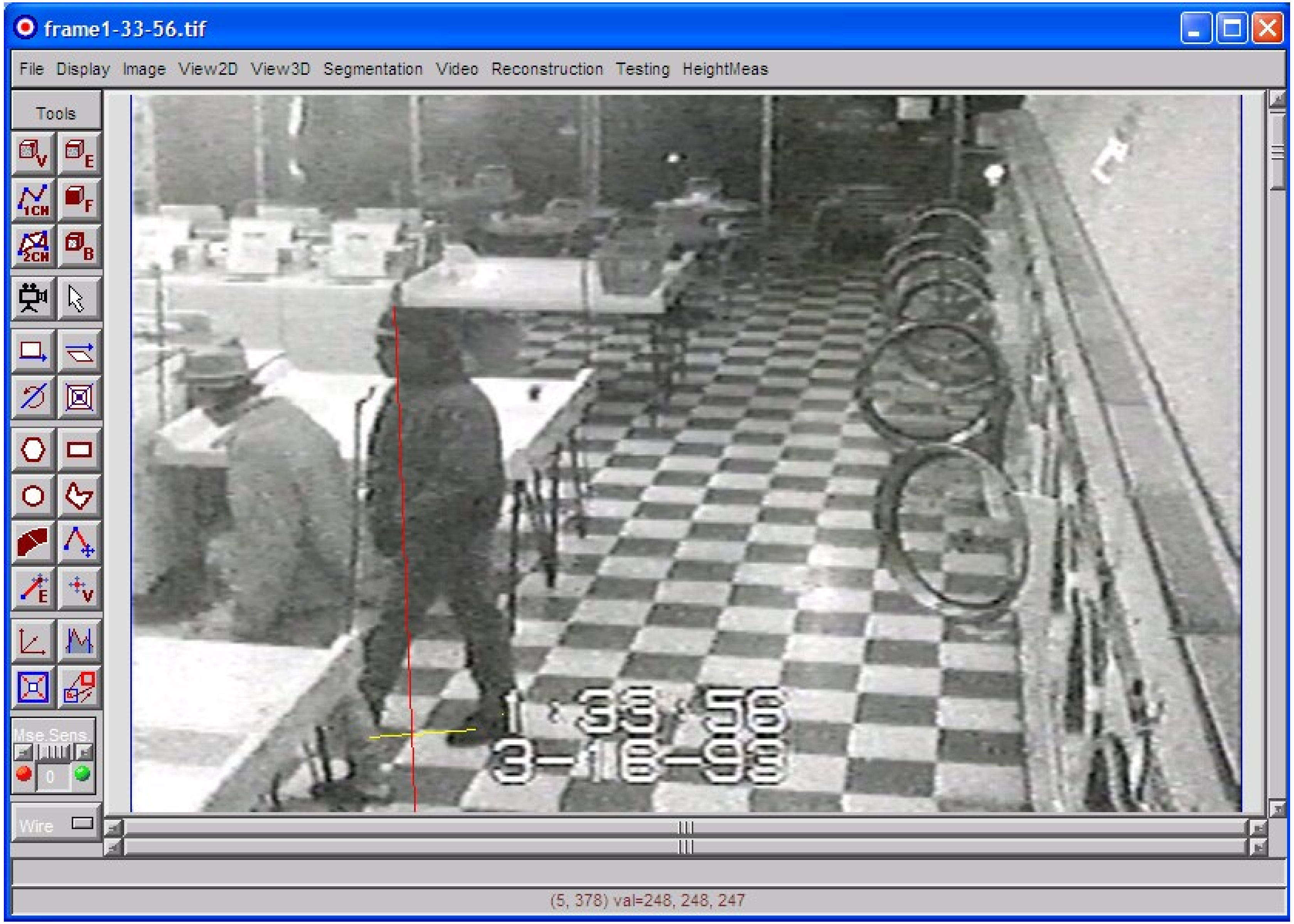

frame1-33-58.tif
File Display Image View2D View3D Segmentation Video Reconstruction Testing HeightMeas
Tools
Mse.Sens.
(300.25 175.5)
(140, 217) val=251, 252, 251

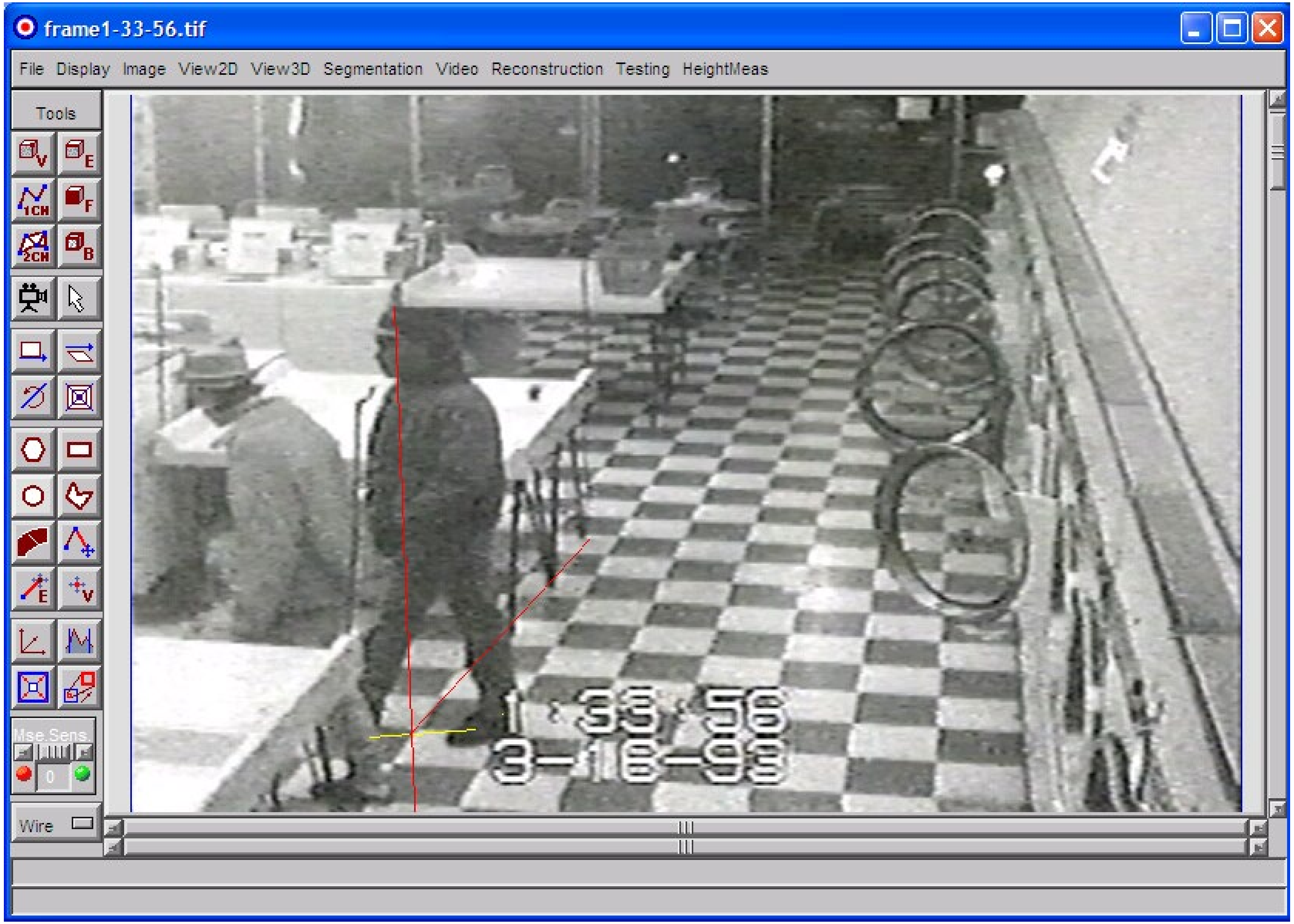


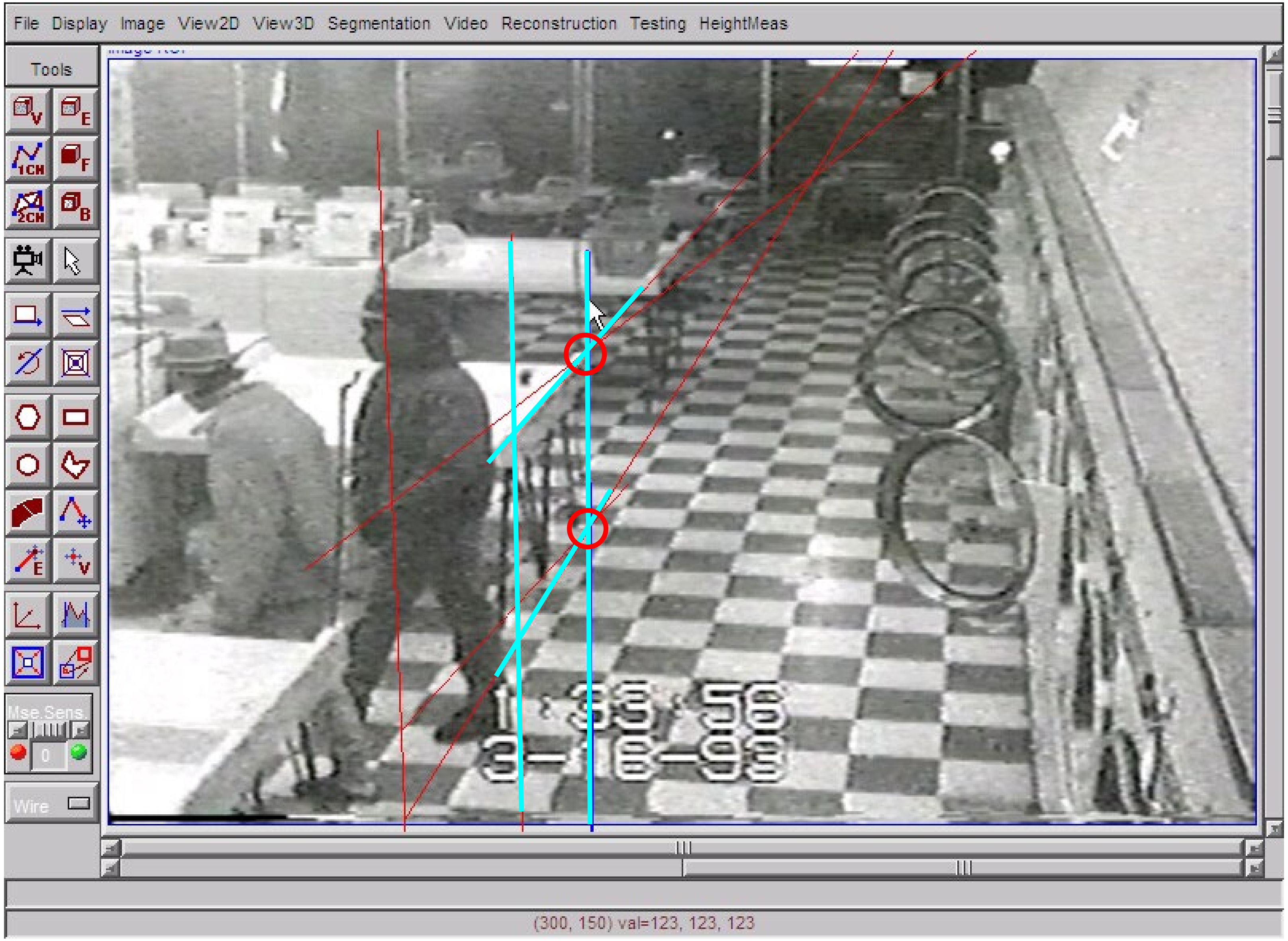
Drop perpendiculars from each of the corners

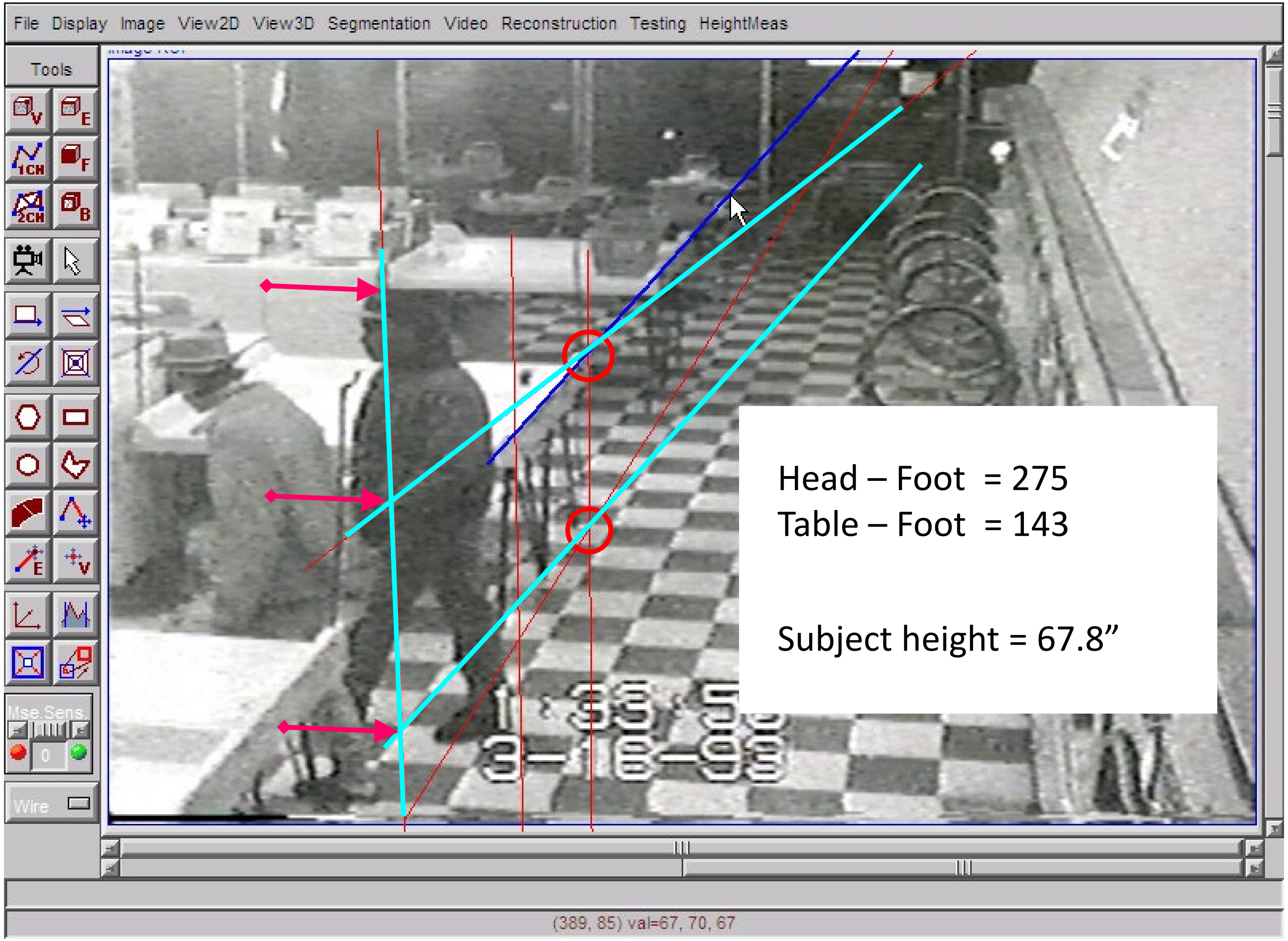

O V M - One View Metrology tool
File RaDiCo Options Edit Edges Setup Measure About
101.4 cm
97.4 cm
94.2 cm
94.2 cm
100.1 cm

Height of the subject


frame1-33-56.tif
File Display Image View2D View3D Segmentation Video Reconstruction Testing HeightMeas
Tools
Mse.Sens.
Wire
1:33:56
3-16-93
(5, 378) val=248, 248, 247


frame1-33-56.tif
File Display Image View2D View3D Segmentation Video Reconstruction Testing HeightMeas
Tools
Mse.Sens.
0
Wire
1:33:56
3-16-93


File Display Image View2D View3D Segmentation Video Reconstruction Testing HeightMeas
Tools
Mse.Sens
Wire
11:33:56
3-16-93
(300, 150) val=123, 123, 123

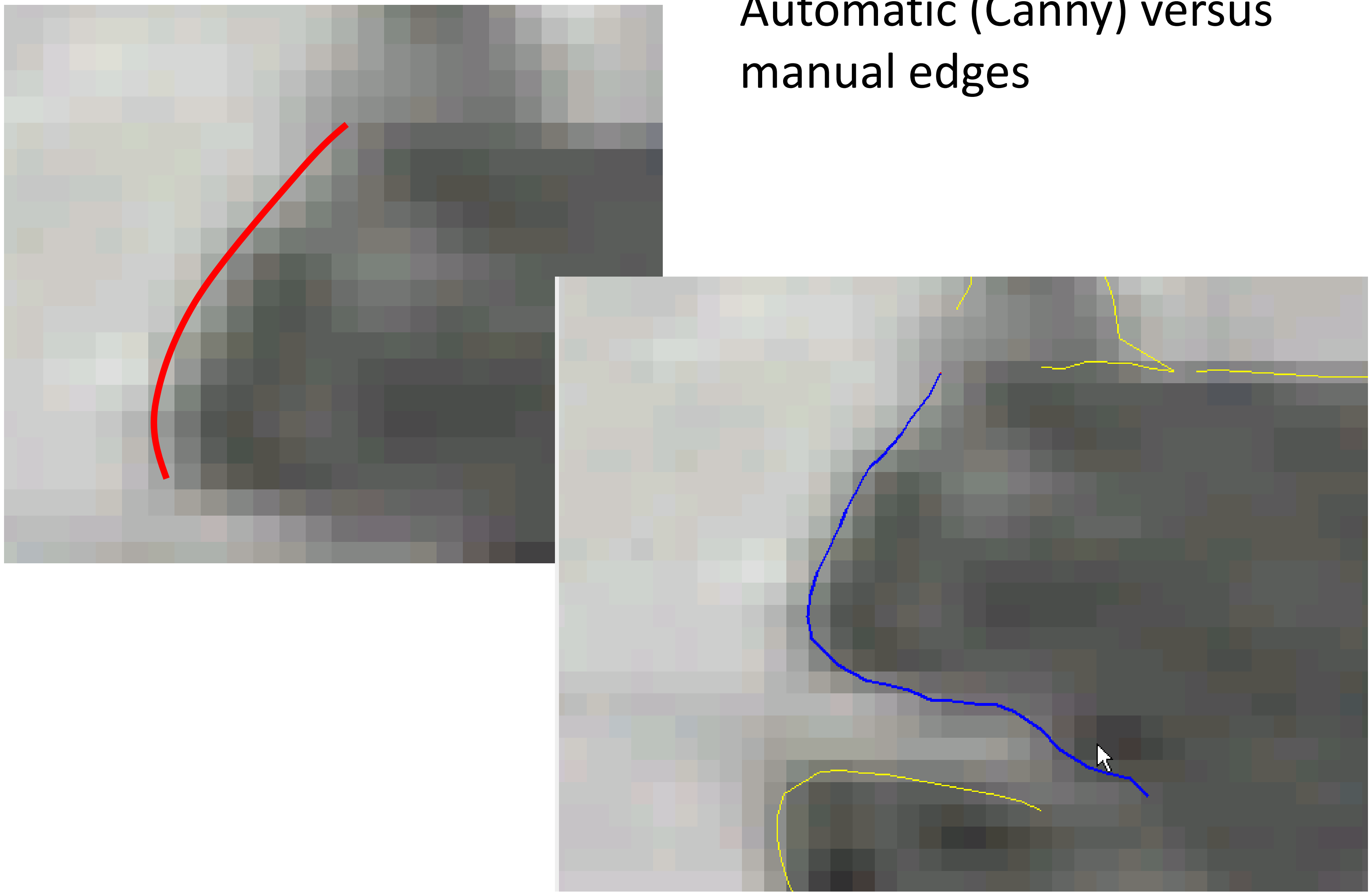


Head – Foot = 275
Table – Foot = 143

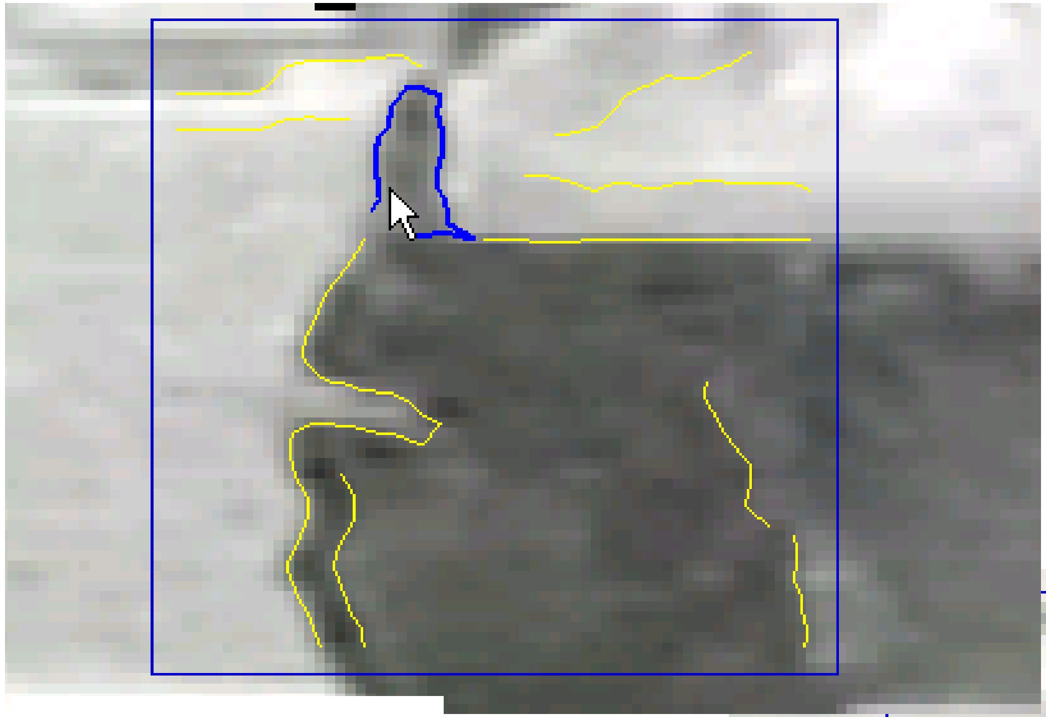
Subject height = 67.8”

More on automatic methods.

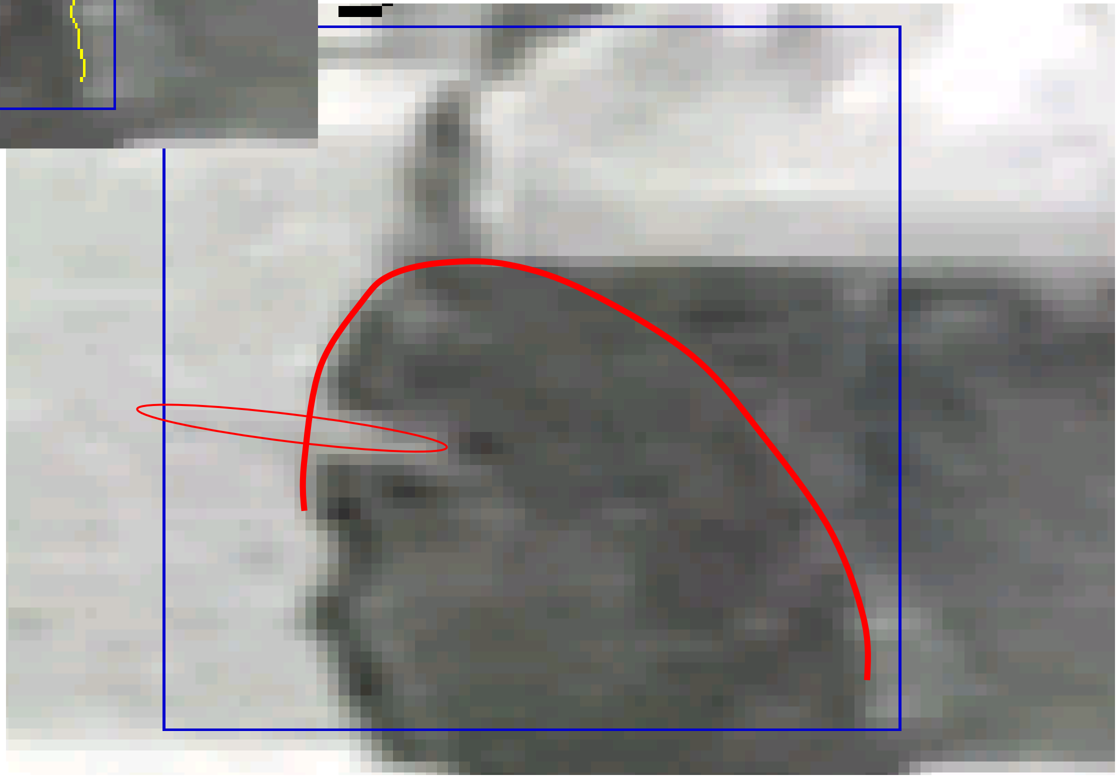
Automatic (Canny) versus manual edges



More on automatic edge detection.



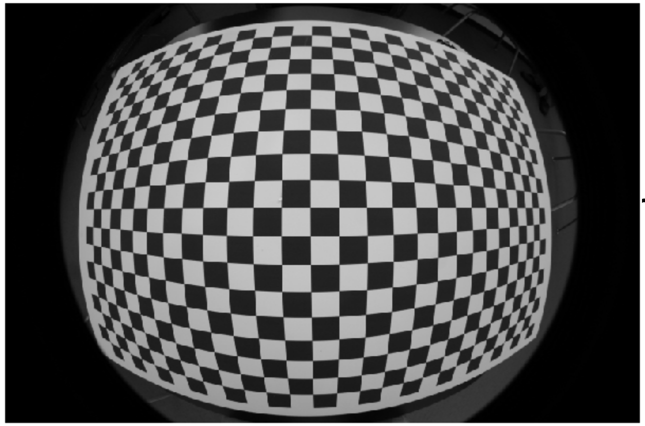
Canny edges



Radial Distortion

Radial Distortion.

- Points in the image are distorted (moved) radially from the centre of the image.
- Straight lines become curved.



rrel

Corrected image

Radial distortion algorithm:

Hartley-Kang, ICCV 2007, IEEE Transactions on PAMI, 2007

- Uses images of a checkered calibration grid

- Computes directly a distortion correction curve: distorted vs undistorted radius.

- Method is ideally suited to these images because the tiled floor acts as a calibration pattern.

Moderate barrel distortion

Corrected image

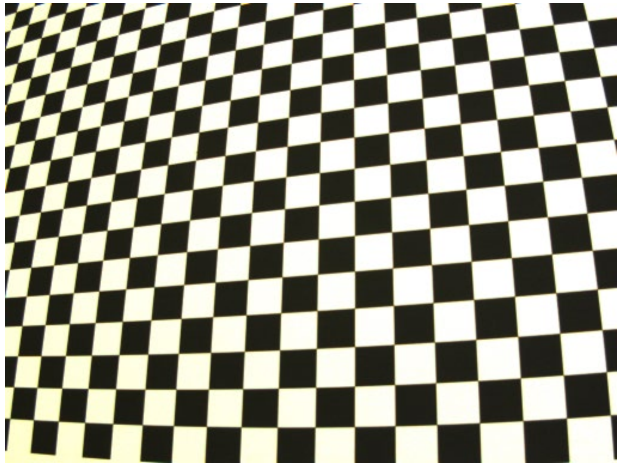
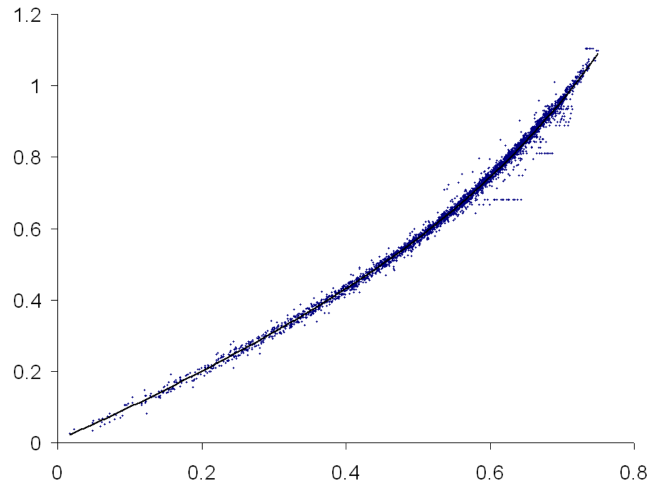
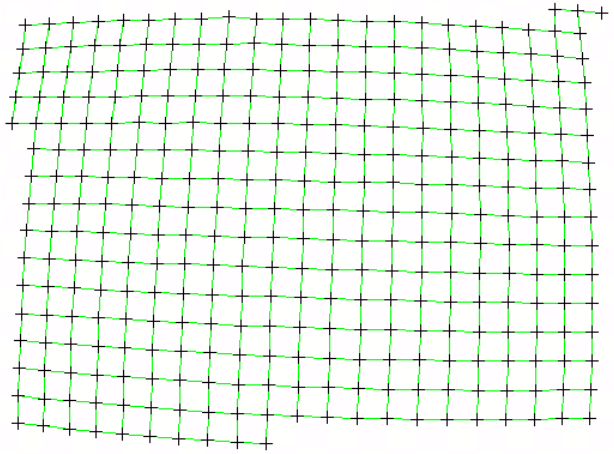
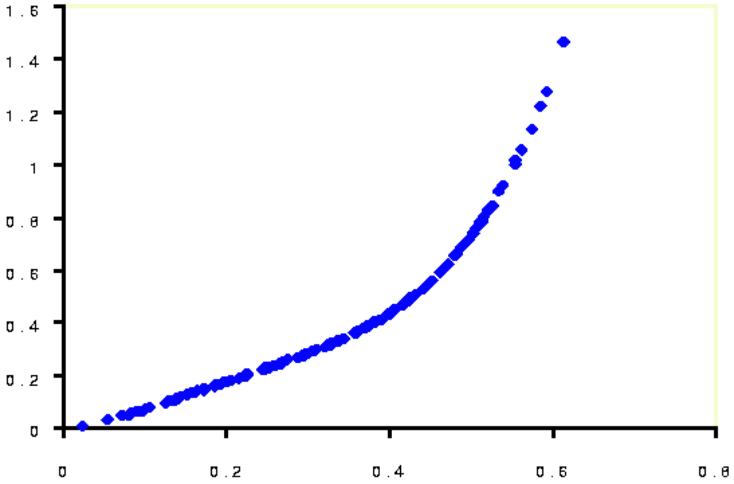


Image of calibration grid



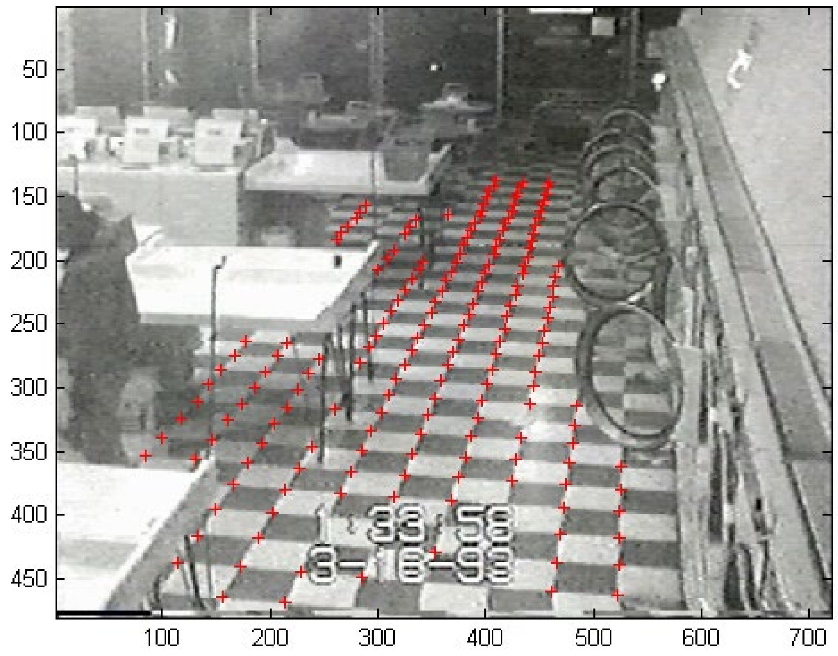


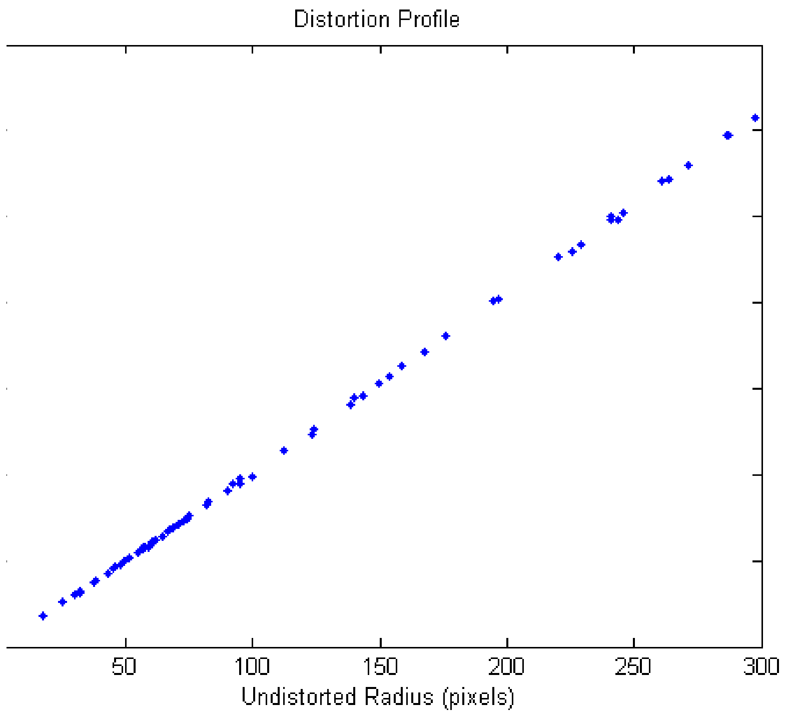
Extracted edges of grid



Sample distortion curves

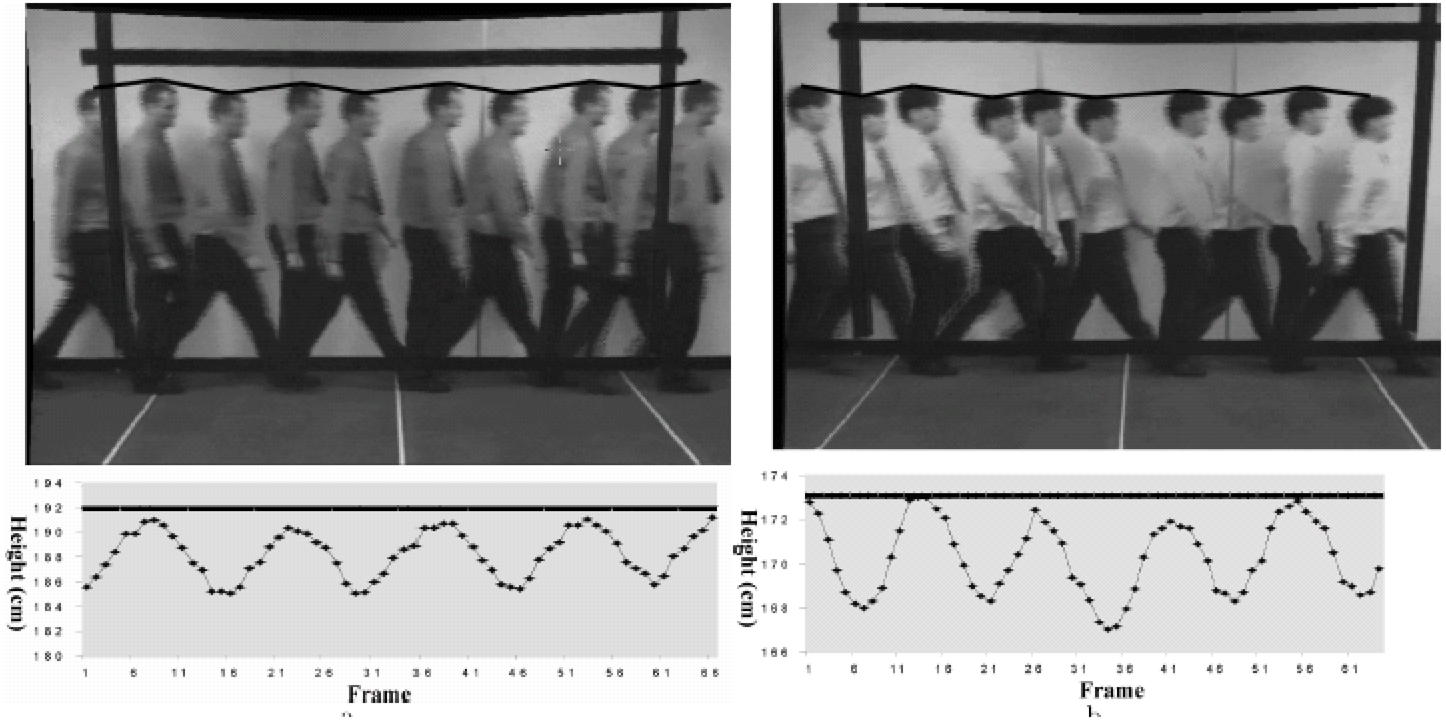
Radial distortion calibration of current image





Conclusion:

- The distortion curve is a straight line.
- There is **no significant radial distortion** in this image.


Height (cm)
194
192
190
188
186
184
182
180
1 6 11 16 21 26 31 36 41 46 51 56 61 66
Frame
Height (cm)
174
172
170
168
166
1 6 11 16 21 26 31 36 41 46 51 56 61
Frame

Back to image transformations

3D transformations

(i) Translations

$$\mathbf{x}' = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}$$

(ii) Rotation and translation

$$\mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}$$

(iii) Affine transformation

$$\mathbf{x}' = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}$$

(iv) Projective transformation

$$\mathbf{x}' = \mathbf{H}\mathbf{x}$$

3D rotations - Euler angles / rotation matrices

A 3D rotation is represented by a 3×3 matrix (in non-homogeneous coordinates).

$$\mathbf{R} = \mathbf{R}_z(\psi)\mathbf{R}_y(\phi)\mathbf{R}_x(\theta) \ .$$

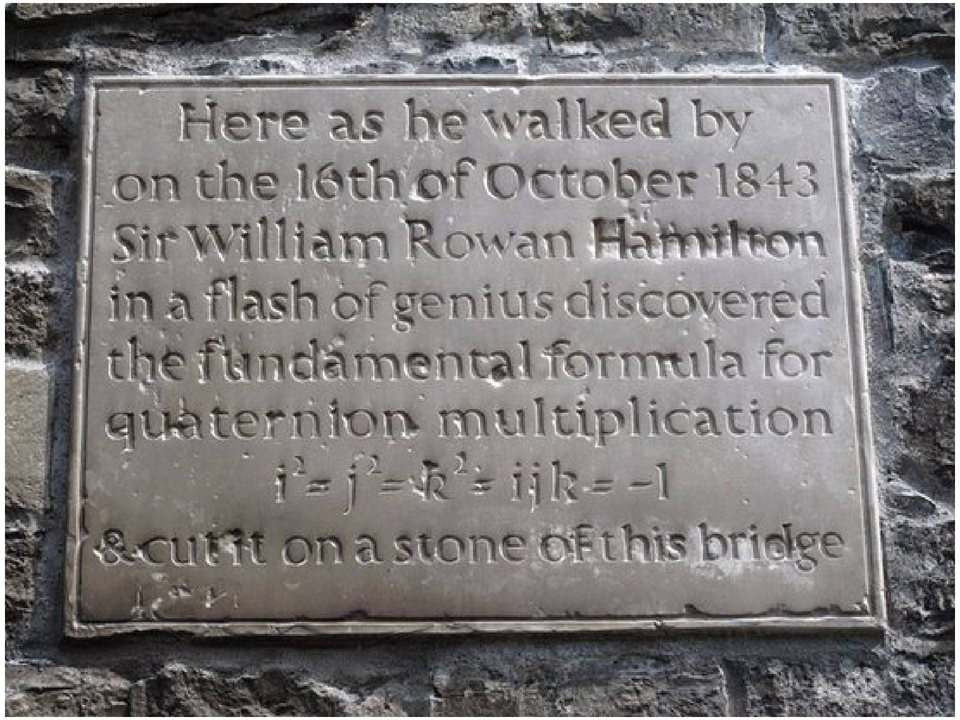
where

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_y(\phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}$$

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

William Rowan Hamilton (1805-1865)
The discoverer of quaternions.



Here as he walked by
on the 16th of October 1843
Sir William Rowan Hamilton
in a flash of genius discovered
the fundamental formula for
quaternion multiplication
$i^2 = j^2 = k^2 = ijk = -1$
& cut it on a stone of this bridge.

3D rotations - quaternions

A 3D rotation may be represented by a unit quaternion. A *quaternion* is a 4-vector

$$\mathbf{q} = (q_0, q_1, q_2, q_3)$$

sometimes written as

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \ .$$

where

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \ .$$

Otherwise stated:

$$\begin{aligned}
\mathbf{ij} &= \mathbf{k} \\
\mathbf{ji} &= -\mathbf{k} \\
\mathbf{jk} &= \mathbf{i} \\
\mathbf{kj} &= -\mathbf{i} \\
\mathbf{ki} &= \mathbf{j} \\
\mathbf{ik} &= -\mathbf{j}
\end{aligned}$$

quaternions and rotations

The quaternion $\mathbf{q} = (q_0, q_1, q_2, q_3)$ with

$$\|\mathbf{q}\|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

represents a rotation through angle

$$\theta = 2\arccos(q_0)$$

about the axis

$$\mathbf{v} = \frac{(q_1, q_2, q_3)}{\sqrt{q_1^2 + q_2^2 + q_3^2}}\ . \qquad (*)$$

The other way round: a rotation through angle θ about a unit axis $=(v_1, v_2, v_3)$ is represented by the quaternion

$$\mathbf{q} = \cos(\theta/2) + \sin(\theta/2)(v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k})\ .$$

Question: What happens to $(*)$ when $q_1 = q_2 = q_3 = 0$?

More notes on quaternions

(i) A quaternion $\mathbf{q}$ and its negative, $-\mathbf{q}$ represent the same rotation.

(ii) Given $\mathbf{q}_1$ and $\mathbf{q}_2$ and their corresponding rotations: $\mathbf{R}(\mathbf{q}_1)$ and $\mathbf{R}(\mathbf{q}_2)$, then the product quaternion $\mathbf{q}_1\mathbf{q}_2$ represents the rotation

$$\mathbf{R}(\mathbf{q}_1\mathbf{q}_2) = \mathbf{R}(\mathbf{q}_1)\mathbf{R}(\mathbf{q}_2) \ .$$

(iii) The *conjugate* of a quaternion $\mathbf{q} = (q_0, q_1, q_2, q_3)$ is the quaternion $\bar{\mathbf{q}} = (q_0, -q_1, -q_2, -q_3)$.

(iv) For a unit quaternion $\mathbf{q}$, we have $\mathbf{q}\bar{\mathbf{q}} = 1$, so we can also write $\bar{\mathbf{q}} = \mathbf{q}^{-1}$. Exercise: For an arbitrary quaternion (not unit) show that $\mathbf{q}\bar{\mathbf{q}} = \|\mathbf{q}\|^2$.

(v) A *vector quaternion* is one for which $\mathbf{q}_0 = 0$.

(vi) A quaternion $\mathbf{q}$ rotates the vector quaternion $\mathbf{p}$ to the vector (quaternion) $\mathbf{q}\mathbf{p}\bar{\mathbf{q}}$. Exercise: if $\mathbf{p}$ is a vector quaternion, then $\mathbf{q}\mathbf{p}\mathbf{q}^{-1}$ is also a vector quaternion.

Exercise: If $\mathbf{q}$ and $\mathbf{p}$ are two quaternions, and $\mathbf{p}\mathbf{q} = 1$, show $\mathbf{q}\mathbf{p} = 1$.

Quaternion and matrix representation

The rotation matrix $\mathbf{R}(\mathbf{q})$ corresponding to a unit quaternion $\mathbf{q} = (q_r, q_i, q_j, q_k)$ is given by

$$\mathbf{R} = \begin{bmatrix} 1 - 2s(q_j^2 + q_k^2) & 2s(q_i q_j - q_k q_r) & 2s(q_i q_k + q_j q_r) \\ 2s(q_i q_j + q_k q_r) & 1 - 2s(q_i^2 + q_k^2) & 2s(q_j q_k - q_i q_r) \\ 2s(q_i q_k - q_j q_r) & 2s(q_j q_k + q_i q_r) & 1 - 2s(q_i^2 + q_j^2) \end{bmatrix}$$

Source:
https://en.wikipedia.org/w/index.php?title=Quaternions_and_spatial_rotation

Reading assignment

(i) Read the Wikipedia page

Source:
https://en.wikipedia.org/w/index.php?title=Quaternions_and_spatial_rotation

(ii) Read the paper:

Vol. 4, No. 4/April 1987/J. Opt. Soc. Am. A 629

Closed-form solution of absolute orientation using unit quaternions

Berthold K. P. Horn

Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, Hawaii 96720

Received August 6, 1986; accepted November 25, 1986

Finding the relationship between two coordinate systems using pairs of measurements of the coordinates of a number of points in both systems is a classic photogrammetric task. It finds applications in stereophotogrammetry and in robotics. I present here a closed-form solution to the least-squares problem for three or more points. Currently various empirical, graphical, and numerical iterative methods are in use. Derivation of the solution is simplified by use of unit quaternions to represent rotation. I emphasize a symmetry property that a solution to this problem ought to possess. The best translational offset is the difference between the centroid of the coordinates in one system and the rotated and scaled centroid of the coordinates in the other system. The best scale is equal to the ratio of the root-mean-square deviations of the coordinates in the two systems from their respective centroids. These exact results are to be preferred to approximate methods based on measurements of a few selected points. The unit quaternion representing the best rotation is the eigenvector associated with the most positive eigenvalue of a symmetric 4 × 4 matrix. The elements of this matrix are combinations of sums of products of corresponding coordinates of the points.

Definition. A $3D$ rotation matrix is a matrix $\mathtt{R}$ such that

$$\mathtt{R}^{-1} = \mathtt{R}^{\top}$$
$$\det(\mathtt{R}) = 1$$

In other words, a $3D$ rotation is an orthogonal matrix with determinant 1.

Theorem(Euler's rotation theorem) Any 3D rotation is a rotation about some axis. Points lying on this axis are unchanged.

Exercise. Any real matrix such that $\mathtt{R}^{-1} = \mathtt{R}^{\top}$ satisfies $\det(\mathtt{R}) = \pm 1$.

Angle-axis representation of rotations

A rotation through the angle θ about a unit axis $\mathbf{v} \in \mathbb{R}^3$ is represented by the vector

$$\mathbf{r} = \theta \mathbf{v}$$

Conversely, a vector $\mathbf{r}$ represents the rotation through angle $\|\mathbf{r}\|$ about the axis $\mathbf{r}/\|\mathbf{r}\|$ (unit axis).

Question: What happens when $\mathbf{r} = \mathbf{0}$?

Advantage over quaternions:

(i) The angle-axis representation is a minimal representation (3 parameters) whereas the quaternion representation needs 4 parameters.

(ii) Quaternion representation needs to be normalized to have norm 1.

Disadvantage of angle-axis rotation:

(i) The angle-axis representation works best for rotations close to the identity (small rotations).

(ii) For other rotations, a rotation should be represented as a delta with respect to a reference rotation.

$$\mathbf{R} = \mathbf{R}_0 \cdot \delta\mathbf{R}(\mathbf{v})$$

Some important notation.

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

This definition is so that the cross-product becomes a matrix-vector product. Thus,

$$\mathbf{t} \times \mathbf{x} = [\mathbf{t}]_\times \mathbf{x} \ .$$

Exercise: Verify this.

Rodriguez formula

How to transform between angle-axis representation and rotation matrix.

Result A4.6. *The matrix $e^{[\mathbf{t}]_\times}$ is a rotation matrix representing a rotation through an angle $\|\mathbf{t}\|$ about the axis represented by the vector $\mathbf{t}$.*

This representation of a rotation is called the *angle-axis* representation.

We may write a specific formula for the rotation matrix corresponding to $e^{[\mathbf{t}]_\times}$. We observe that $[\mathbf{t}]_\times^3 = -\|\mathbf{t}\|^2 \, [\mathbf{t}]_\times = -\|\mathbf{t}\|^3 \, [\hat{\mathbf{t}}]_\times$, where $\hat{\mathbf{t}}$ represents a unit vector in the direction $\mathbf{t}$. Then, with $\mathrm{sinc}(\theta)$ representing $\sin(\theta)/\theta$, we have

$$\begin{aligned}
e^{[\mathbf{t}]_\times} &= \mathtt{I} + [\mathbf{t}]_\times + [\mathbf{t}]_\times^2/2! + [\mathbf{t}]_\times^3/3! + [\mathbf{t}]_\times^4/4! + \dots \\
&= \mathtt{I} + \|\mathbf{t}\| \, [\hat{\mathbf{t}}]_\times + \|\mathbf{t}\|^2 \, [\hat{\mathbf{t}}]_\times^2/2! - \|\mathbf{t}\|^3 \, [\hat{\mathbf{t}}]_\times/3! - \|\mathbf{t}\|^4 \, [\hat{\mathbf{t}}]_\times^2/4! + \dots \\
&= \mathtt{I} + \sin\|\mathbf{t}\| \, [\hat{\mathbf{t}}]_\times + (1 - \cos\|\mathbf{t}\|) \, [\hat{\mathbf{t}}]_\times^2 \\
&= \mathtt{I} + \mathrm{sinc}\|\mathbf{t}\| \, [\mathbf{t}]_\times + \frac{1 - \cos\|\mathbf{t}\|}{\|\mathbf{t}\|^2} \, [\mathbf{t}]_\times^2 \qquad \text{(A4.9)} \\
&= \cos\|\mathbf{t}\|\mathtt{I} + \mathrm{sinc}\|\mathbf{t}\| \, [\mathbf{t}]_\times + \frac{1 - \cos\|\mathbf{t}\|}{\|\mathbf{t}\|^2} \, \mathbf{t}\mathbf{t}^\mathsf{T}
\end{aligned}$$

where the last line follows from the identity $[\mathbf{t}]_\times^2 = \mathbf{t}\mathbf{t}^\mathsf{T} - \|\mathbf{t}\|^2\mathtt{I}$.

If $\mathbf{t}$ is written as $\mathbf{t} = \theta\hat{\mathbf{t}}$, where $\hat{\mathbf{t}}$ is a unit vector in the direction of the axis, and $\theta = \|\mathbf{t}\|$ is the angle of rotation, then (A4.9) is equivalent to the Rodrigues formula for a rotation matrix:

$$\mathtt{R}(\theta, \hat{\mathbf{t}}) = \mathtt{I} + \sin\theta[\hat{\mathbf{t}}]_\times + (1 - \cos\theta)[\hat{\mathbf{t}}]_\times^2 \tag{A4.12}$$

(i) Extraction of the axis and rotation angle from a rotation matrix $\mathtt{R}$ is just a little tricky. The unit rotation axis $\mathbf{v}$ can be found as the eigenvector corresponding to the unit eigenvalue – that is by solving $(\mathtt{R} - \mathtt{I})\mathbf{v} = \mathbf{0}$. Next, it is easily seen from (A4.9) that the rotation angle ϕ satisfies

$$\begin{aligned} 2\cos(\phi) &= (\mathrm{trace}(\mathtt{R}) - 1) \\ 2\sin(\phi)\mathbf{v} &= (\mathtt{R}_{32} - \mathtt{R}_{23}, \mathtt{R}_{13} - \mathtt{R}_{31}, \mathtt{R}_{21} - \mathtt{R}_{12})^{\mathsf{T}}. \end{aligned} \tag{A4.10}$$

Writing this second equation as $2\sin(\phi)\mathbf{v} = \hat{\mathbf{v}}$, we can then compute $2\sin(\phi) = \mathbf{v}^{\mathsf{T}}\hat{\mathbf{v}}$. Now, the angle ϕ can be computed from $\sin(\phi)$ and $\cos(\phi)$ using a two-argument $\arctan$ function (such as the C-language function atan2(y, x)).

It has often been written that ϕ can be computed directly from (A4.10) using $\arccos$ or $\arcsin$. However, this method is not numerically accurate, and fails to find the axis when $\phi = \pi$.

(ii) To apply a rotation $\mathtt{R}(\mathbf{t})$ to some vector $\mathbf{x}$, it is not necessary to construct the

matrix representation of t. In fact

$$\begin{aligned}\mathtt{R}(\mathbf{t})\mathbf{x} &= \left(\mathtt{I} + \operatorname{sinc}\|\mathbf{t}\|[\mathbf{t}]_\times + \frac{1-\cos\|\mathbf{t}\|}{\|\mathbf{t}\|^2}[\mathbf{t}]_\times^2\right)\mathbf{x} \\ &= \mathbf{x} + \operatorname{sinc}\|\mathbf{t}\|\mathbf{t}\times\mathbf{x} + \frac{1-\cos\|\mathbf{t}\|}{\|\mathbf{t}\|^2}\mathbf{t}\times(\mathbf{t}\times\mathbf{x}) \qquad \text{(A4.11)}\end{aligned}$$

Reconstruction and Video Enhancement

Photo Tourism

Exploring photo collections in 3D

Noah Snavely Steven M. Seitz Richard Szeliski
University of Washington *Microsoft Research*

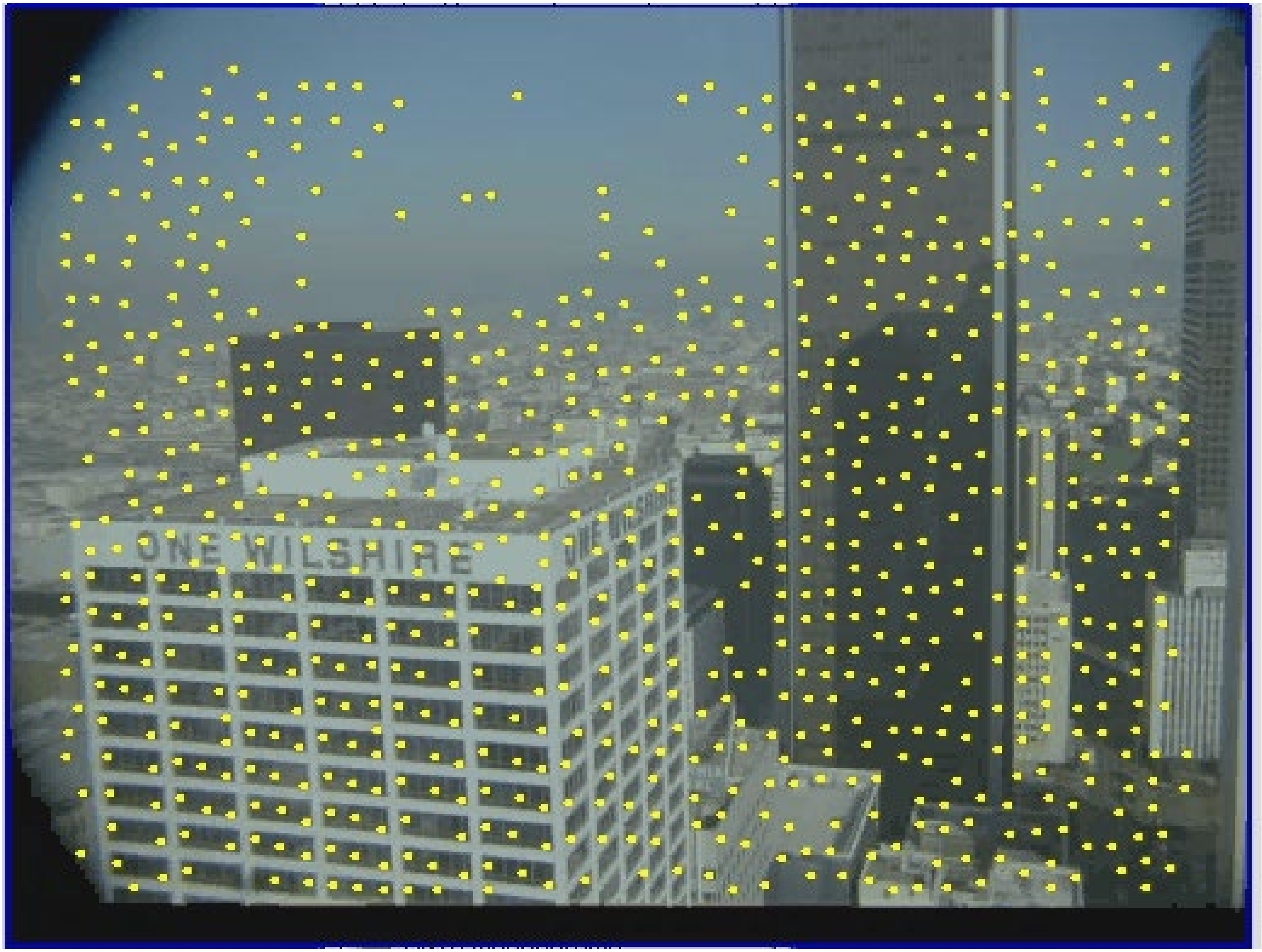
SIGGRAPH 2006

Image sequence of Wilshire Boulevard, LA

Courtesy Oxford Visual Geometry group

Steps of reconstruction:

1. Tracking of points in the video
2. Weeding out bad tracks
3. Projective reconstruction
4. Self calibration / Euclidean reconstruction
5. Model building

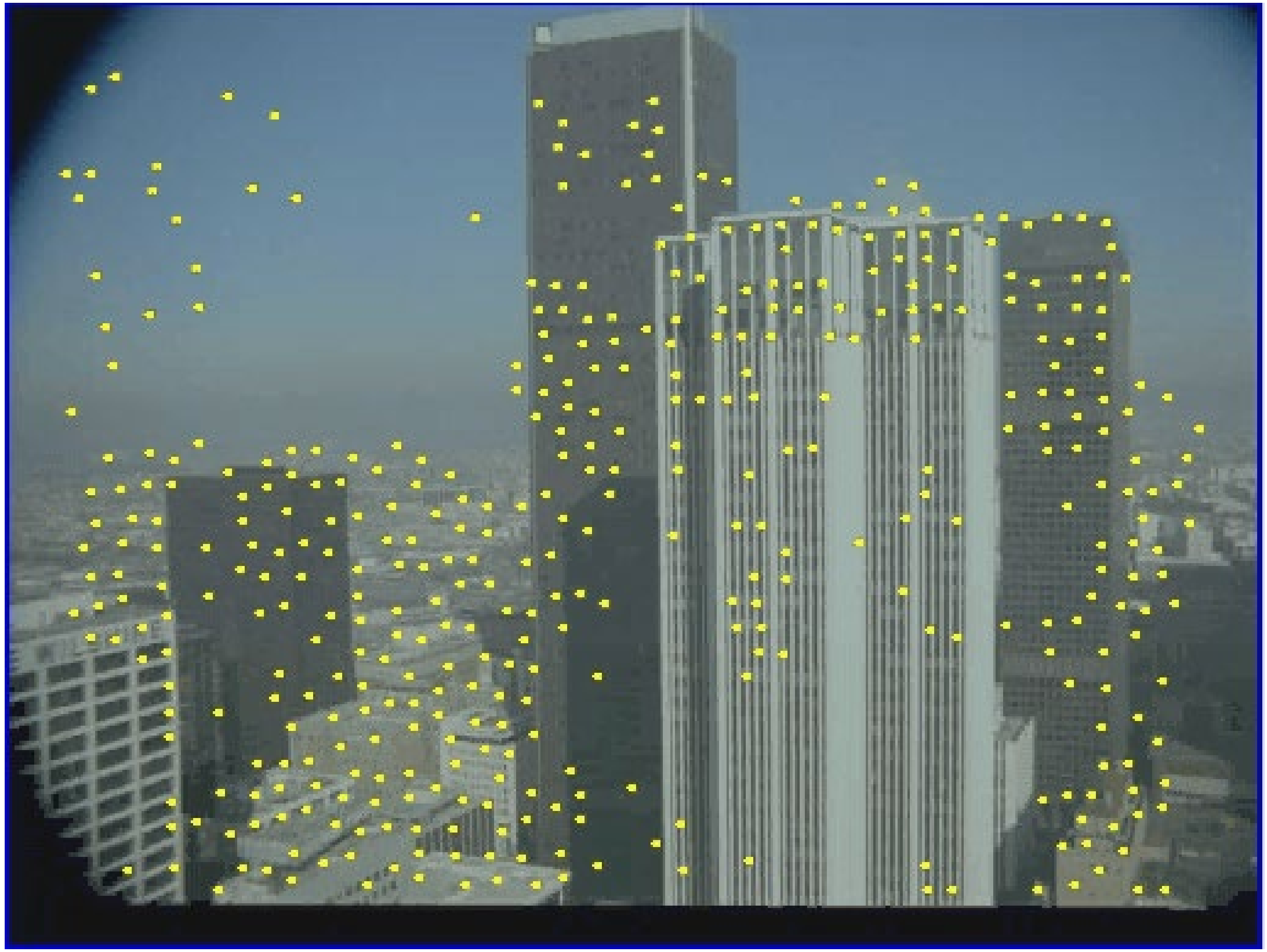


Unweeded points

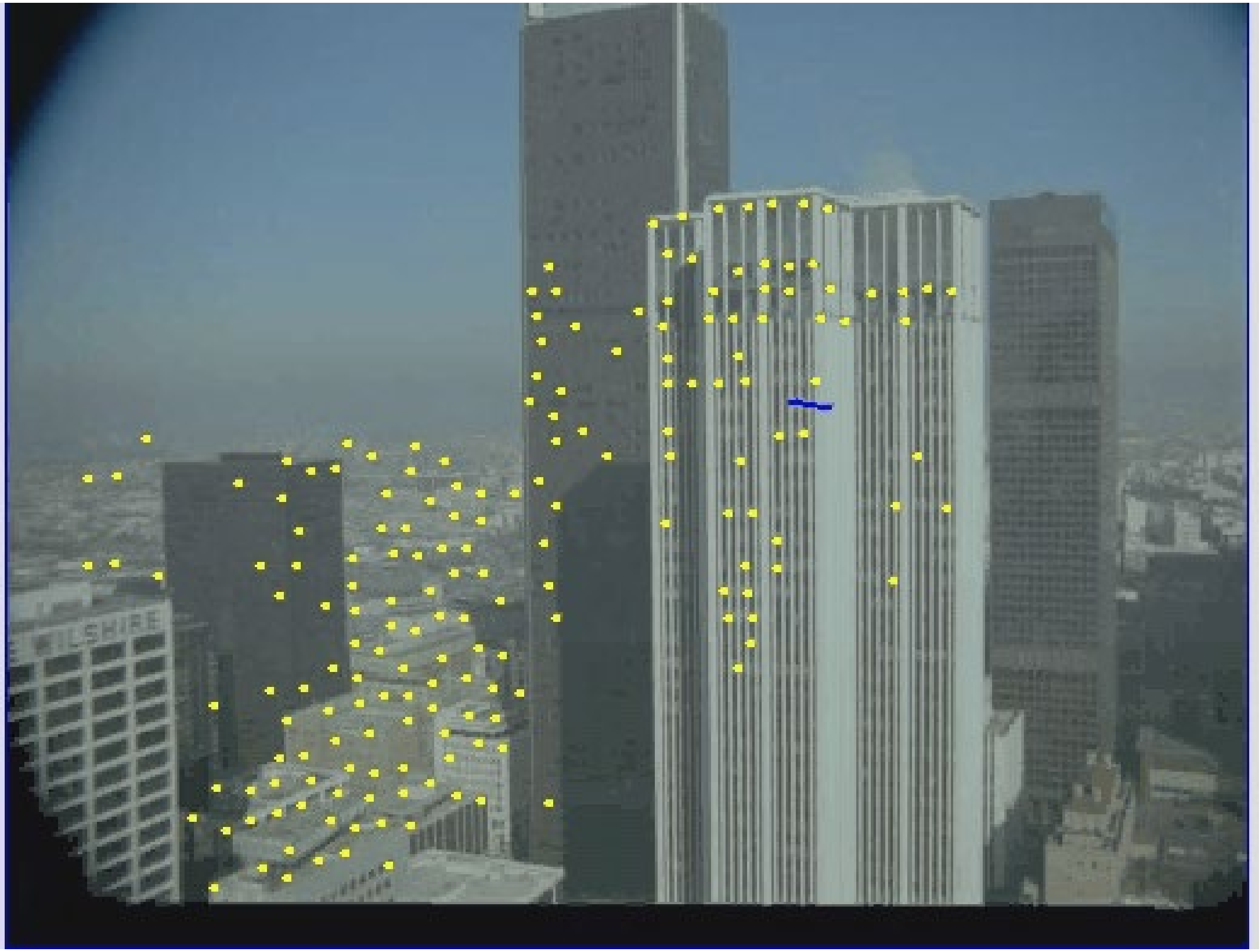
Tracking

First step is to weed the points

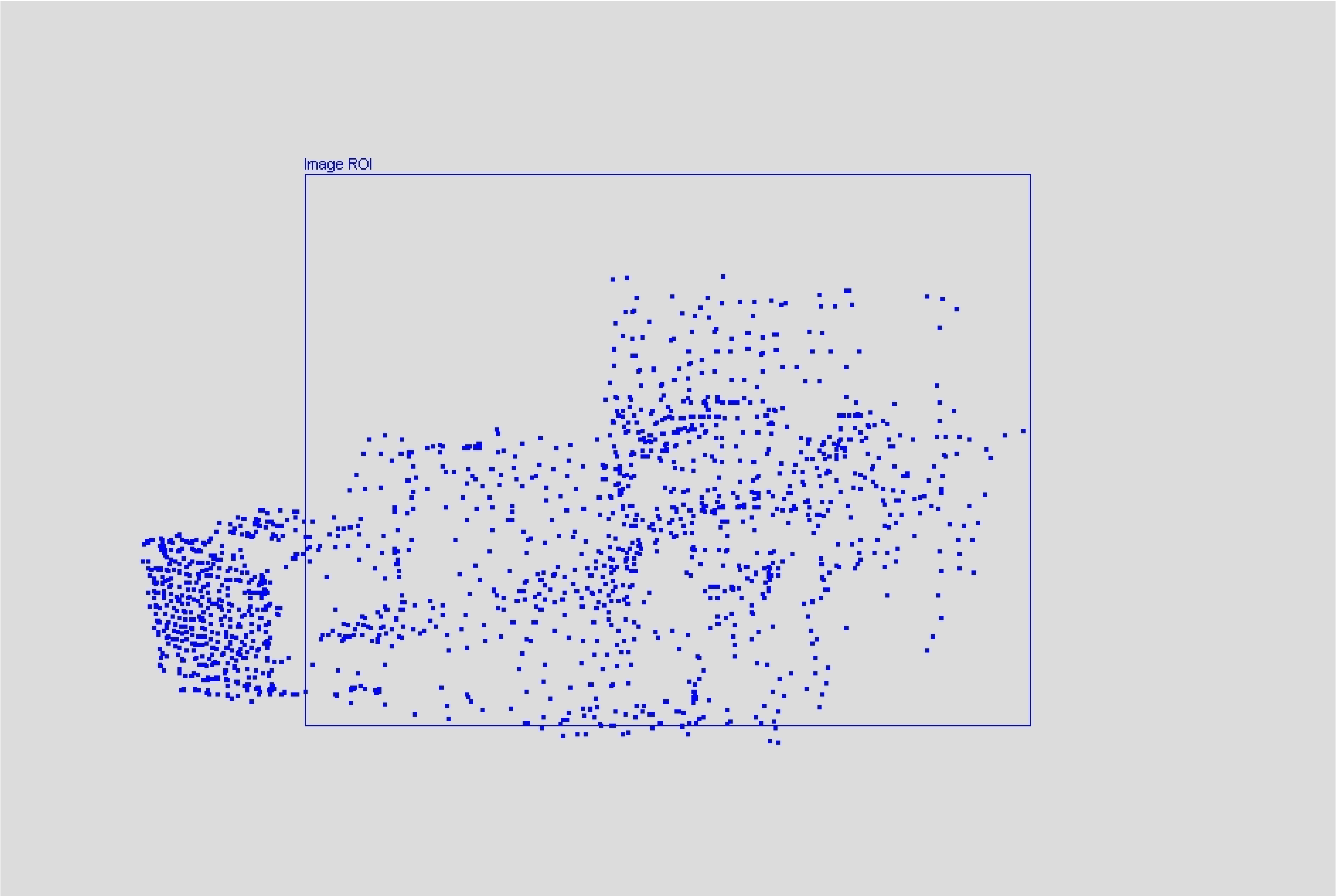
1. Produces lots of bad matches
2. Must be weeded out using a weeding program



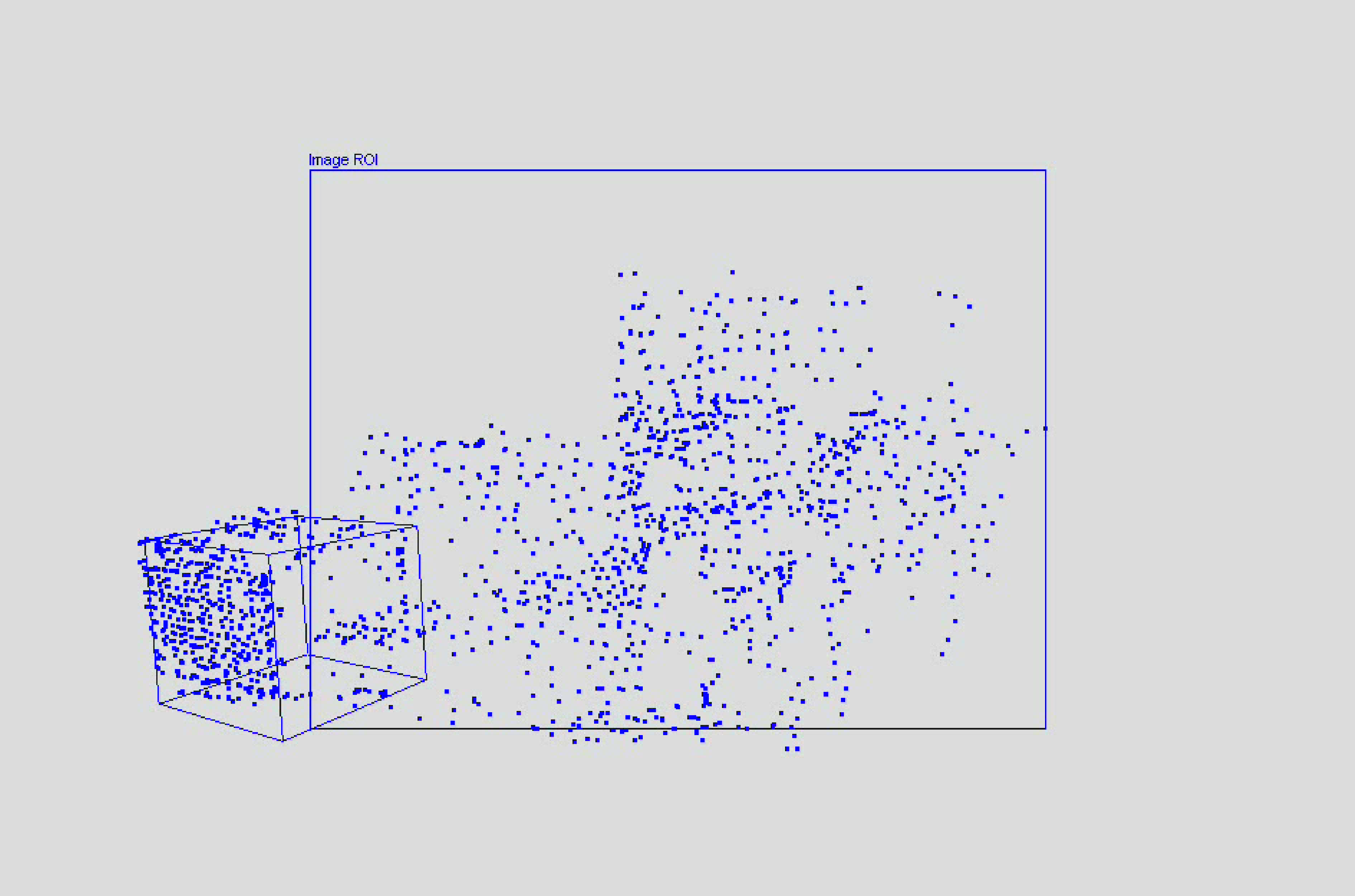
Outliers removed using the fundamental matrix



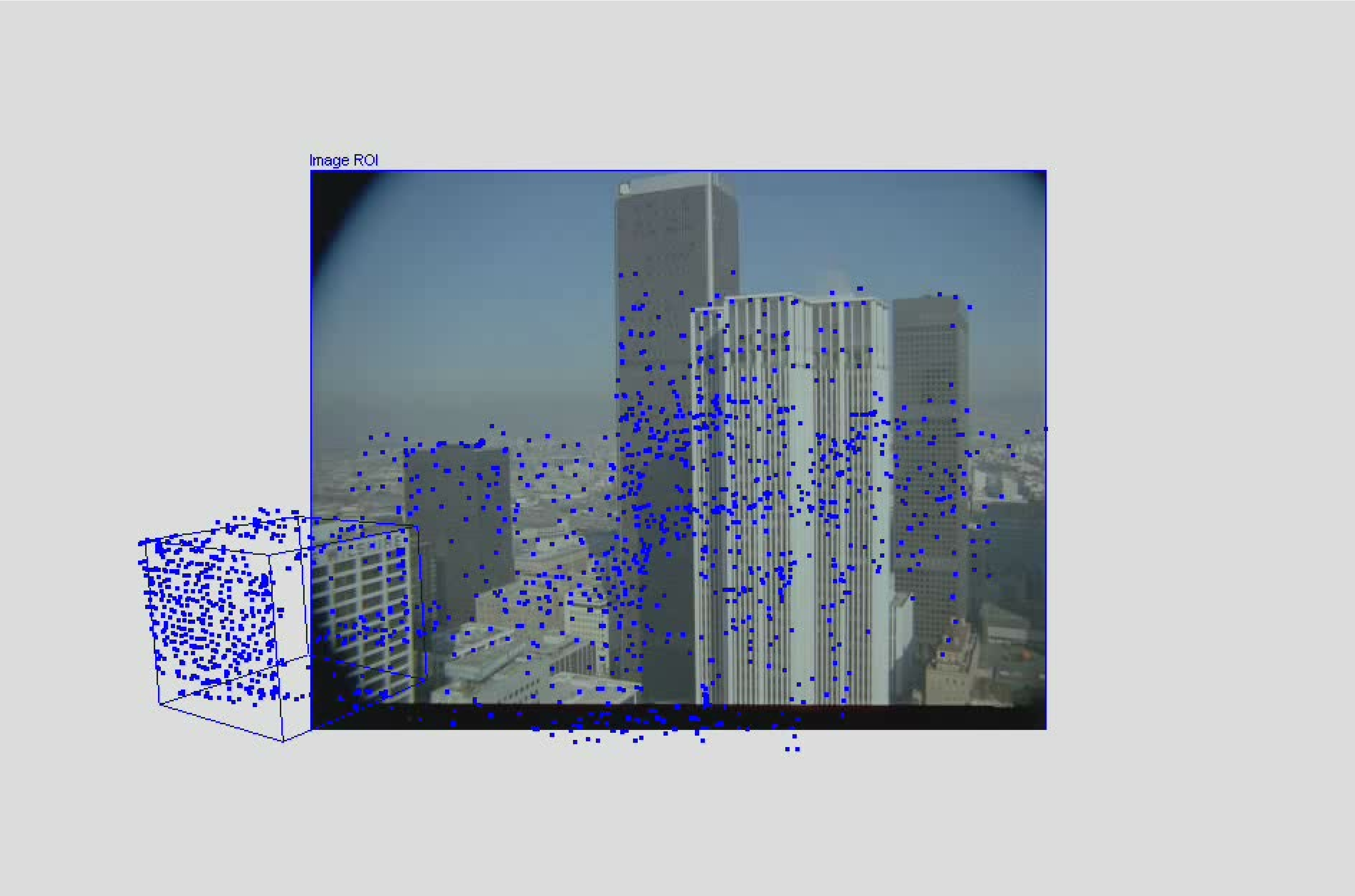
Outliers removed using the trifocal tensor



Results of reconstruction of points



Rectangular reference frame



Rectangular reference frame

What is the mathematics behind this?

Projective Geometry

Felix Klein: Erlangen Program (1872)

View of geometry as an analysis of what is invariant under different groups of transforms.

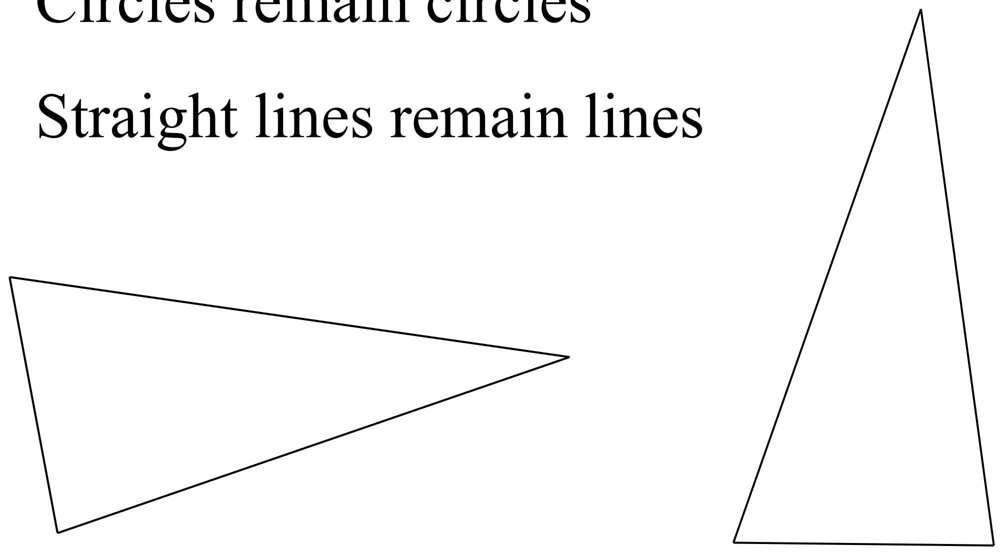
i.e. What is unchanged about the plane when it undergoes a particular type of transformation (motion, distortion, change).

Euclidean Geometry

The geometry of rigid transformations (congruence).

Invariants are:

- Angles, distances, areas
- Circles remain circles
- Straight lines remain lines



Concepts of Euclidean Geometry

- Circle
- Length
- Area
- Congruent triangles
- Parallelism
- Angles

All these are invariant (unchanged) under rigid transformations.

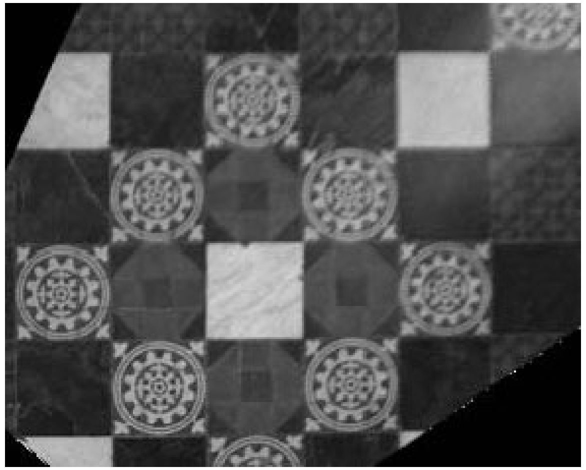
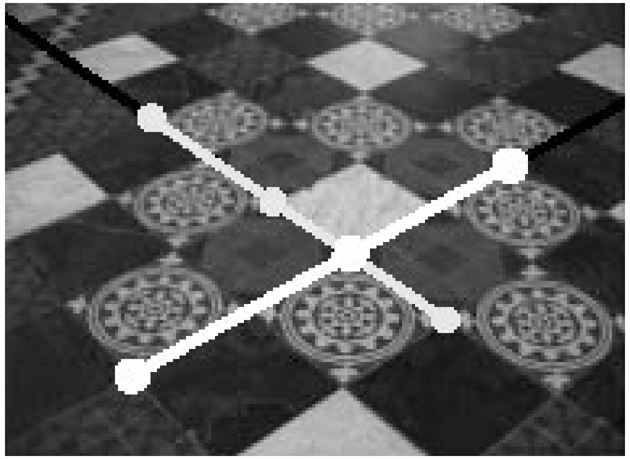
Projective Geometry

The geometry of projective transforms.

What transformations does a plane undergo when you take an image of it with a pinhole camera?




Parallel lines are not preserved

Neither are angles, areas, distance-ratios, lengths

“Second Order” curves.

- Circles are not preserved under projective transformations.
- They can be seen as ellipses (including circles), parabolas, hyperbolas, (second degree curves).
- The right concept in projective geometry is a “conic”.
- Conics are preserved by projective transformations.

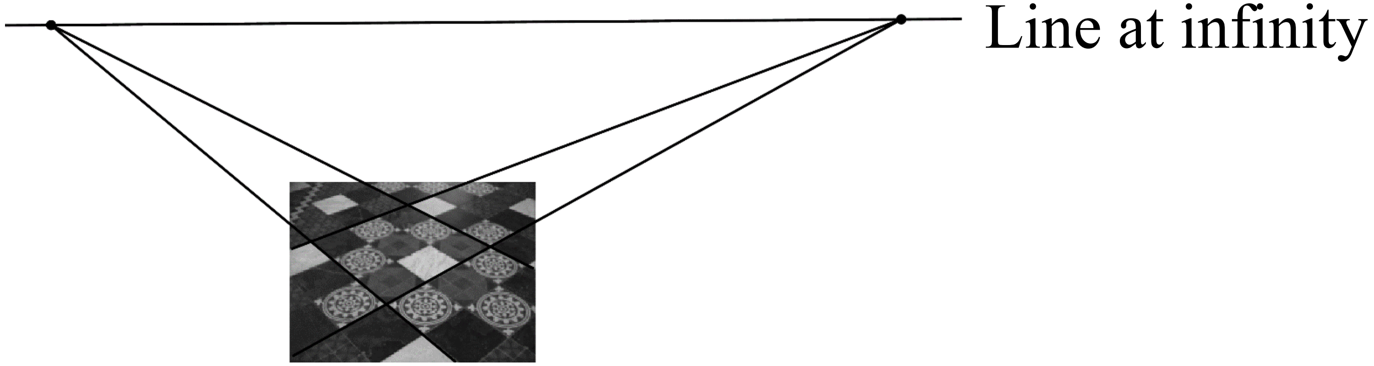
Fisheye image

Projective Image

Circles are projected as ellipses

The Projective Plane $\mathcal{P}^2$

Extend the Euclidean Plane by adding a line at infinity.



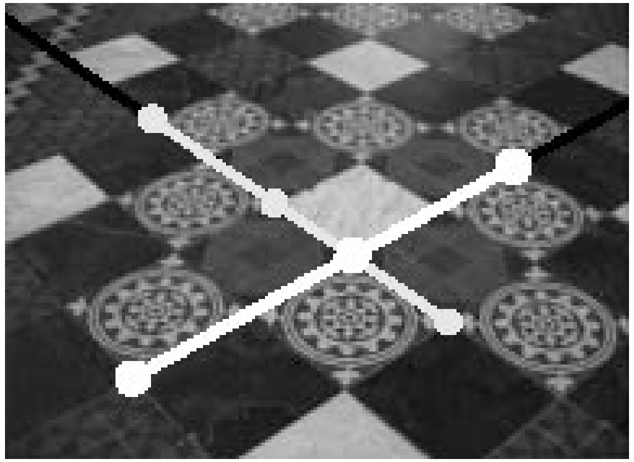
Any two lines meet in exactly one point

Projective Transformation

Any mapping of the projective plane to itself that preserves lines is called a “projective transformation”, or “collineation”, or “homography”

Mapping: A mapping from points in the plane to points in the plane.

Preserves lines: If three points lie on a line, then their corresponding points under the mapping lie on a line.



Homogeneous coordinates

- A point (x, y) on a plane is represented by a 3-vector $(x, y, 1)^\top$.
- All multiples of $(x, y, 1)^\top$ represent the same point. Thus

$$(x, y, 1)^\top = (2x, 2y, 2)^\top = \dots = (kx, ky, k)^\top$$

  for any non-zero k.
- An arbitrary 3-vector $(x, y, w)^\top$ represents the point $(x/w, y/w)$
- Points with $w = 0$ represent points on the "plane at infinity"

Projective space, $\mathcal{P}^2$ and $\mathcal{P}^3$

- The set of all (equivalence classes of) homogeneous $(n+1)$-vectors is known as "projective n-space", $\mathcal{P}^n$.
- Points with last coordinate equal to zero are called "points at infinity".
- Thus, $\mathcal{P}^2 = R^2 \cup \{\text{line at infinity}\}$
- $\mathcal{P}^3 = R^3 \cup \{\text{plane at infinity}\}$

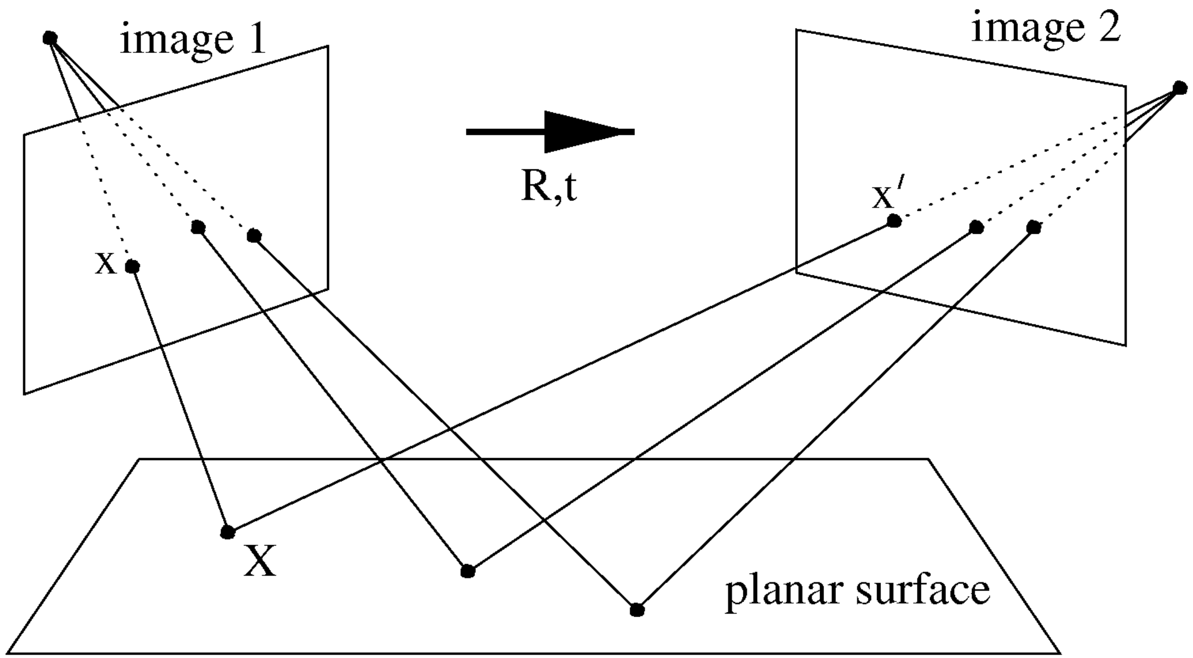
Homography between planes

Definition *Any point-to-point mapping from* $\mathcal{P}^2$ *to* $\mathcal{P}^2$ *that takes lines to lines is called a homography.*

Other names : collineation, projectivity, projective transform.

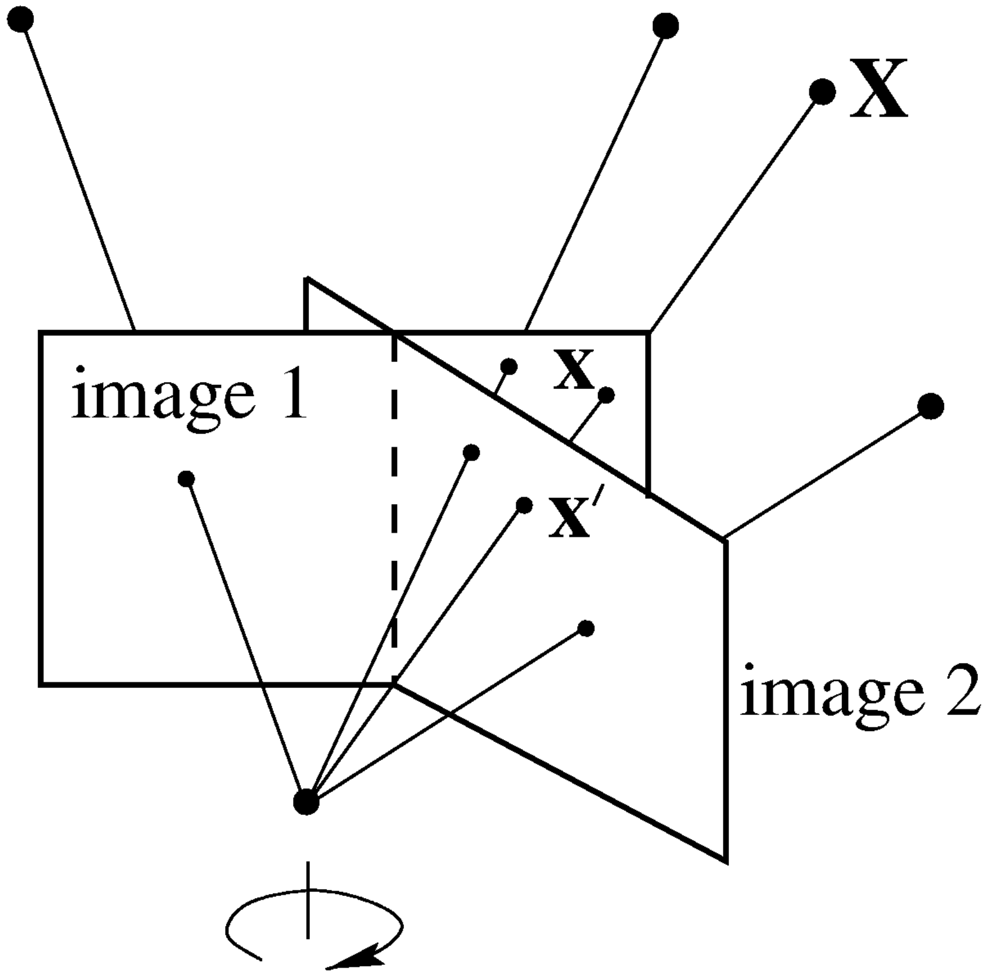
How do 2D homographies arise

- Between a plane in the world and its image with a perspective camera.
- Between two images of a plane.
- Between two images of the world taken with a rotating (but not moving) camera.



Two images of a plane are related by a homography

Images of floor, related by homographies



Images taken with a rotating camera.

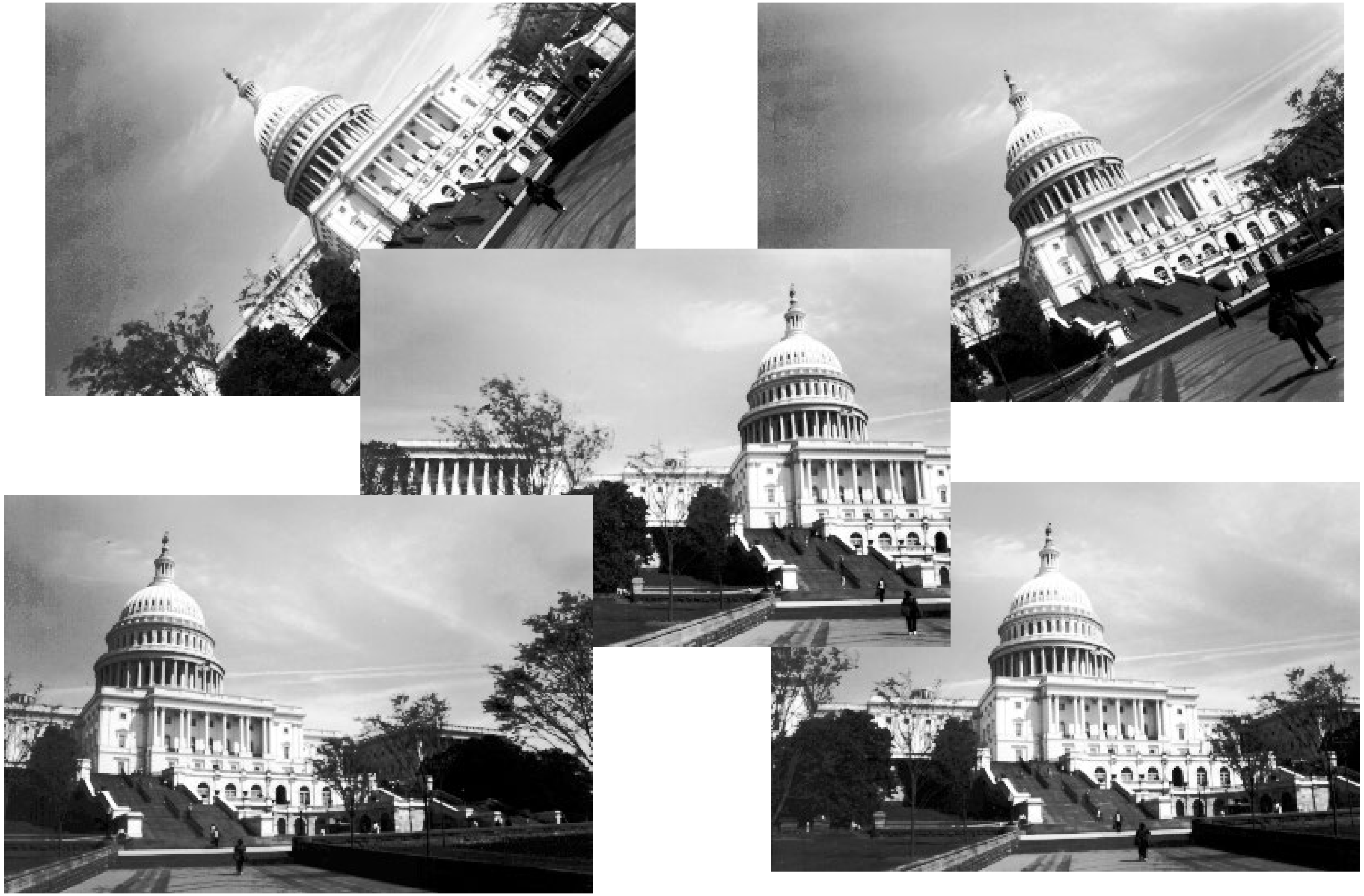
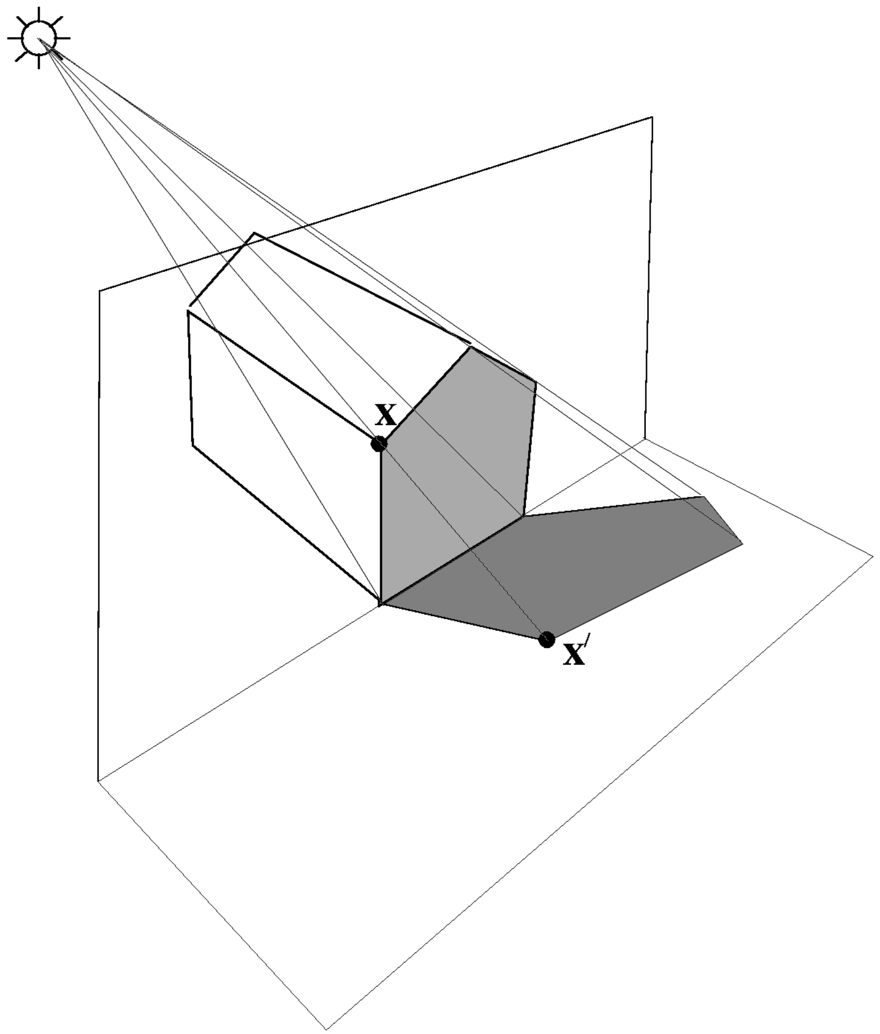


Image taken from the same location

Mosaicking by homographic warping



Images of a planar object and its shadow.

Projective Transformations

A projective transformation is a mapping

$$(x, y) \mapsto (x', y')$$

where

$$\begin{aligned} x' &= \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ y' &= \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \end{aligned}$$

Homography in non-homogeneous coordinates

- In non-homogeneous coordinates, homography is written as follows:

$$
\begin{aligned}
x' &= \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\
y' &= \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}
\end{aligned}
$$

- In homogeneous coordinates we write:

$$\mathrm{x}' = \mathrm{Hx}$$

Homogeneous coordinates

- A point (x, y) on a plane is represented by a 3-vector $(x, y, 1)^\top$.
- All multiples of $(x, y, 1)^\top$ represent the same point. Thus

  $$(x, y, 1)^\top = (2x, 2y, 2)^\top = ... = (kx, ky, k)^\top$$

  for any non-zero k.
- An arbitrary 3-vector $(x, y, w)^\top$ represents the point $(x/w, y/w)$
- Points with $w = 0$ represent points on the "plane at infinity"

Rational Numbers (fractions)

- A rational number is represented by a pair of integers, (p/q).
- Two rational numbers are the same if the pairs differ by a constant multiple. Thus

$$(p/q) \approx (2p/2q) \approx (kp/kq)$$

- Finite rational numbers are (p/q) with $q \neq 0$.
- The fraction (p/q) with $q = 0$ represents ∞.

Algebraic formulation of a homography

- Points in the plane are represented by homogeneous coordinates $\mathbf{x} = (x, y, w)^\top$
- Homography is represented by a 3×3 matrix $\mathtt{H}$

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{23} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}$$

- Thus, homography is just a linear transformation on homogeneous coordinates.

Thus, a projective transformation is represented by an *invertible* matrix

$$\mathtt{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

Here is an exercise:

Theorem 1. *A mapping of $\mathcal{P}^2 \rightarrow \mathcal{P}^2$ preserves lines if and only if it is of the form given above.*

Computation of 2D homography

- Homography $\mathtt{H}$ has 8 degrees of freedom (9 entries, but scale irrelevant).
- Each point provides two constraints on $\mathtt{H}$.
- Thus 4 point matches are required to compute $\mathtt{H}$.
- With more than 8 point matches, least-squares techniques are used.
- Computation of homographies is reliable and easy.

Tracking planes – original video

Stabilization via homographies