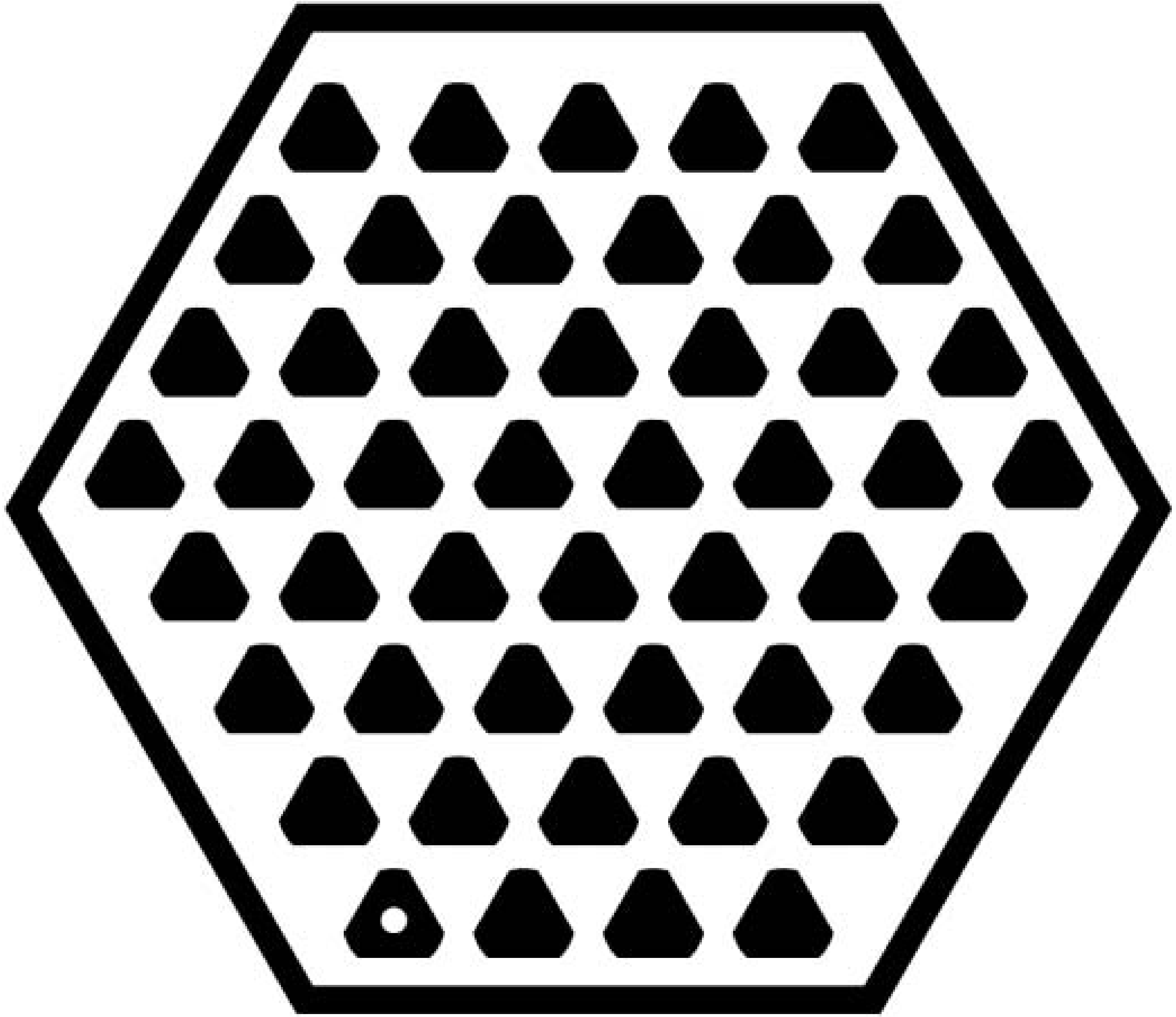
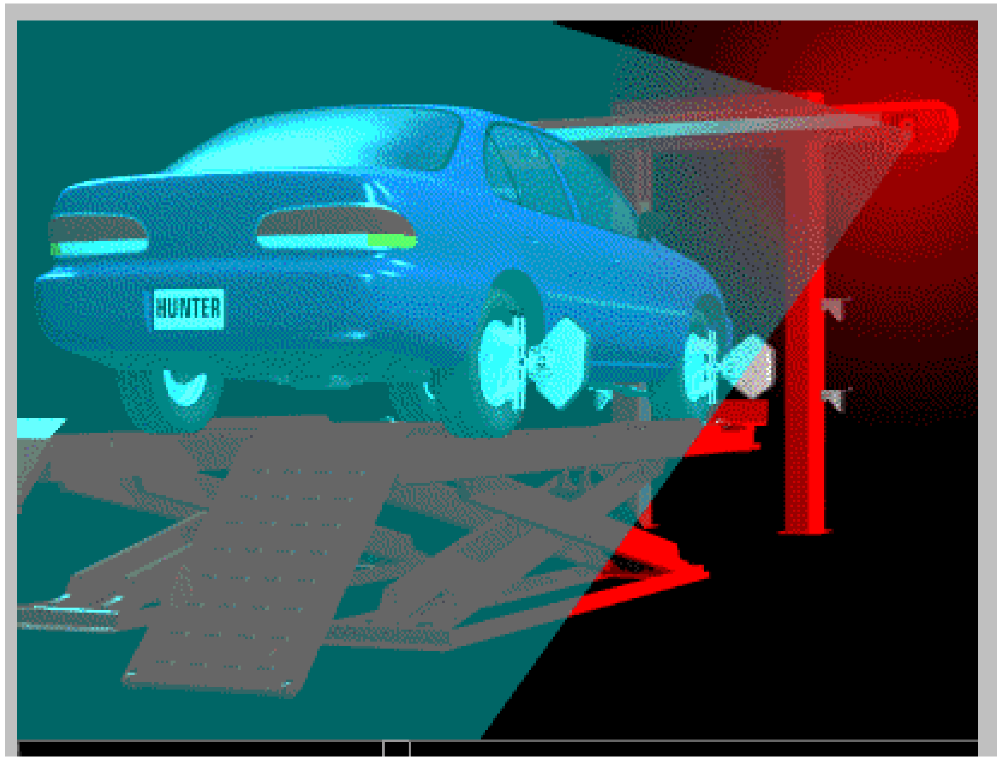
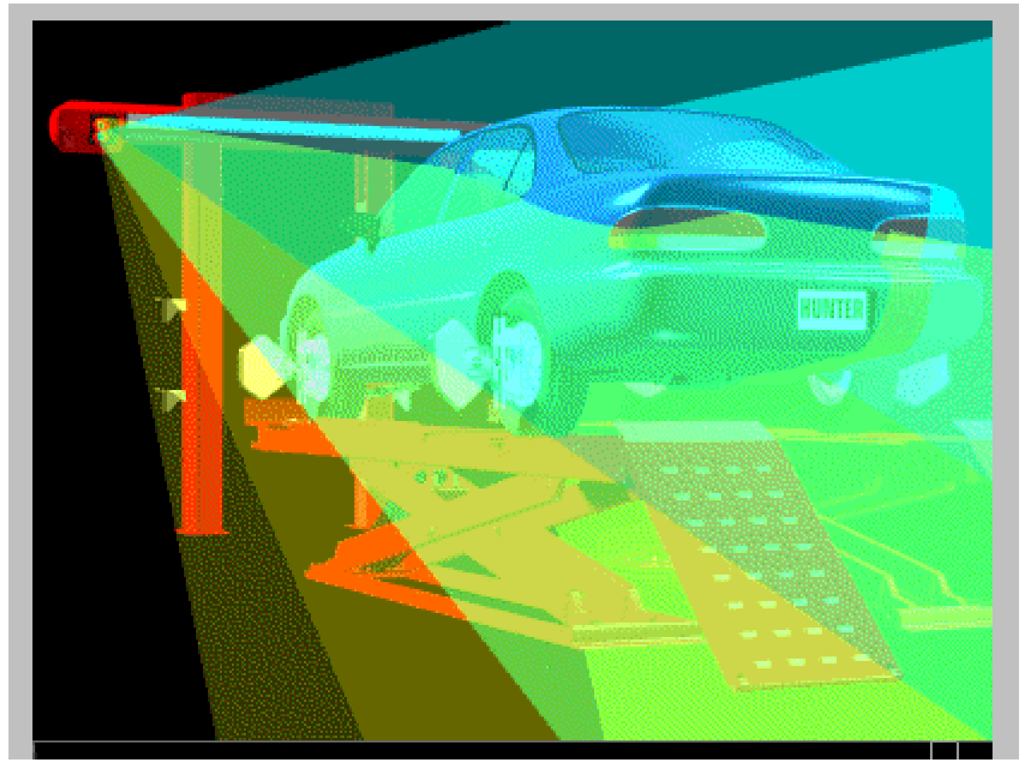


Perspective Projection Describes Image Formation

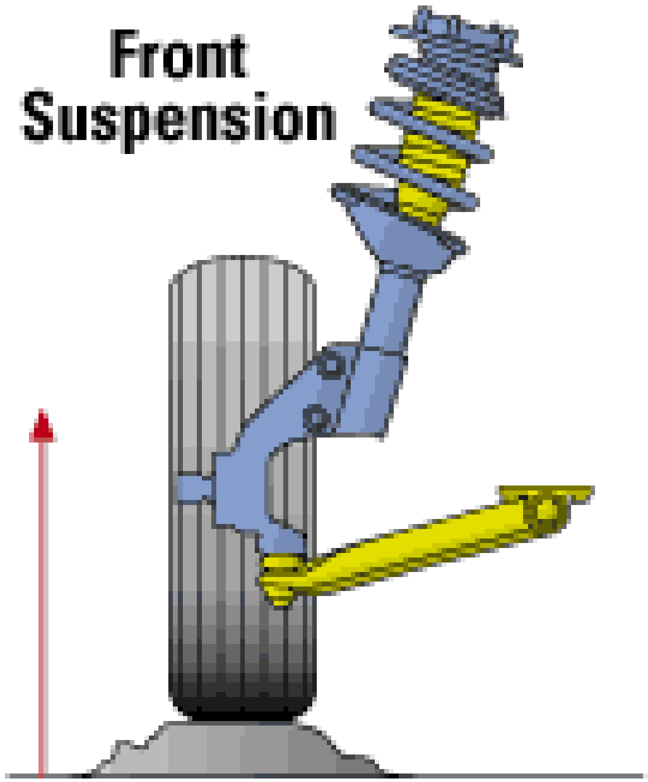
Berthold K.P. Horn

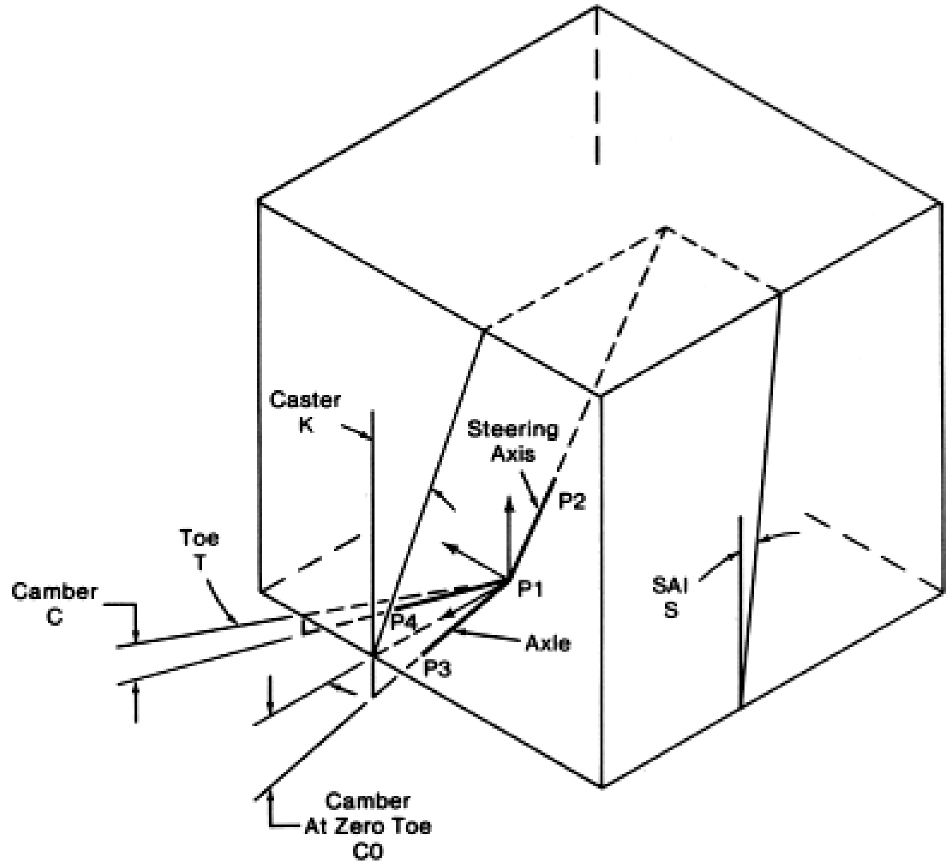




Wheel Alignment:

Camber, Caster, Toe-In, SAI, …





Camber: angle between axle and horizontal plane.

Toe: angle between projection of axle on horizontal plane and a perpendicular to thrust line.

Caster: angle in side elevation between steering axis and vertical.

SAI: angle in frontal elevation between steering axis and vertical.

Planar Target

Mounted Rigidly on Rim

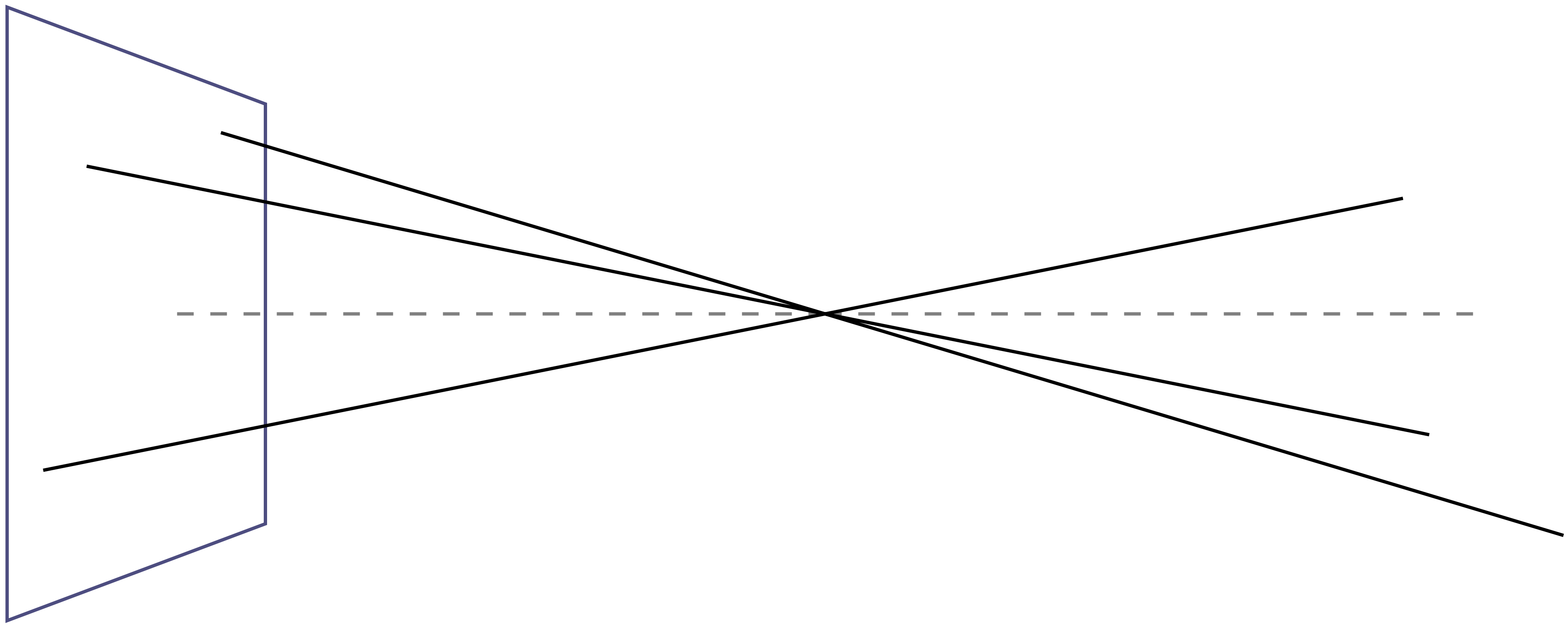
CCD Cameras and LED Illumination

Determining Rolling Axis

Determining Steering Axis

Photogrammetric Problems:

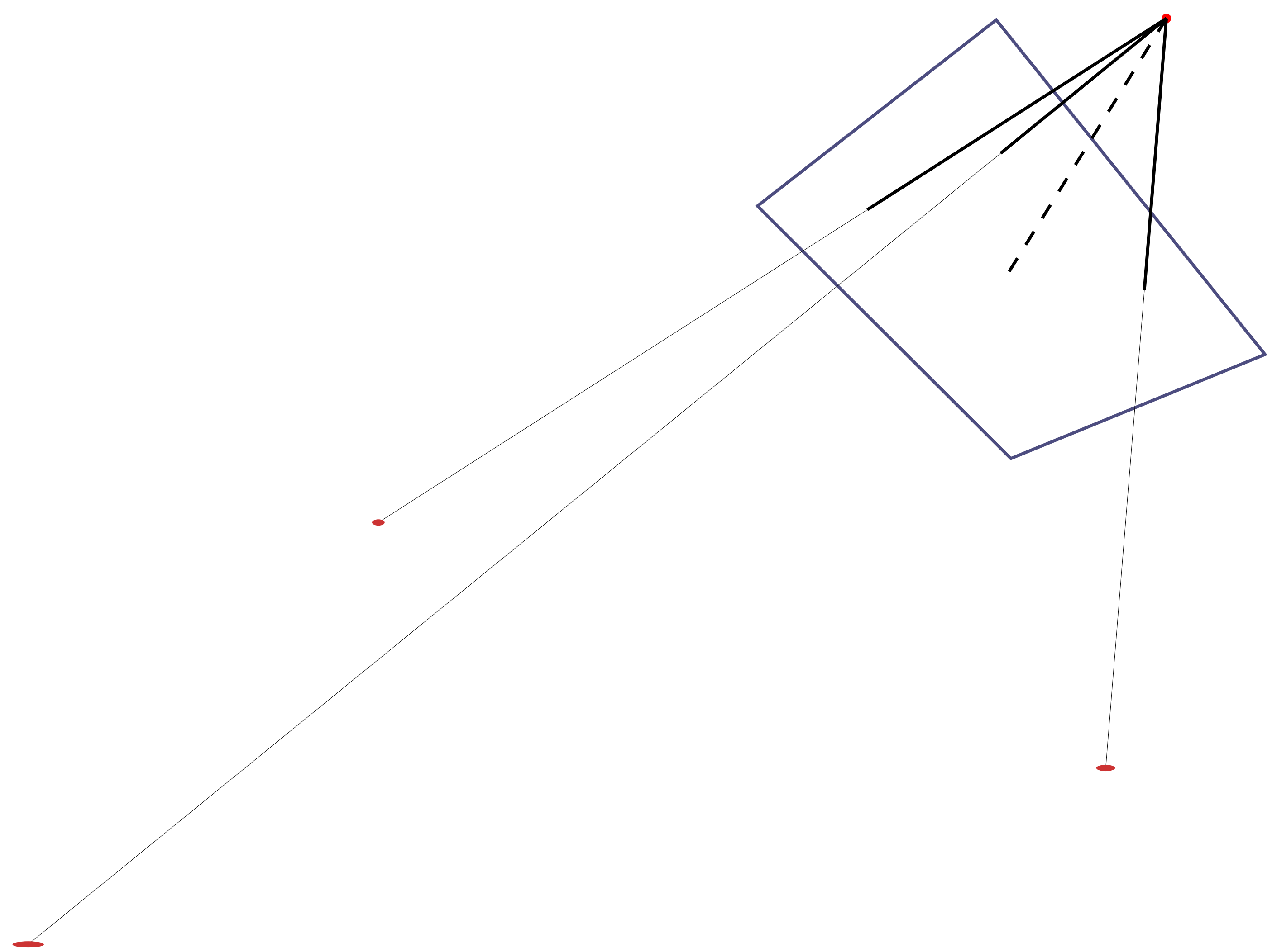
- **Interior Orientation (2D ↔ 3D)**
- **Exterior Orientation (3D ↔ 2D)**
- **Absolute Orientation (3D ↔ 3D)**
- **Relative Orientation (2D ↔ 2D)**



Interior Orientation:

Principal Point + Principal Distance

Center of Projection: (u_o, v_o, f)



Exterior Orientation:

Rotation + Translation of Camera: R and **t**

Rigid body transformation:

rotation and translation —
of object coordinate system into camera coordinate system.

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = R \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} + \begin{pmatrix} x_o \\ y_o \\ z_o \end{pmatrix} \tag{1}$$

R is a 3×3 orthonormal matrix that represents rotation,
while $\mathbf{t} = (x_o, y_o, z_o)^T$ represents translation.

Constraints:

Orthonormality: $R^T R = I$
(six independent second order constraints).

Rotation rather than reflection: $\det(R) = +1$
(one third order constraint).

Perspective projection:

Camera coordinate system (3D) into image coordinate system (2D) (*) (**).

$$\begin{aligned} u &= f\,(x_c/z_c) + u_o \\ v &= f\,(y_c/z_c) + v_o \end{aligned} \tag{2}$$

f is principal distance — "effective focal length"

while $(u_o, v_o)^T$ is principal point — "image center"

Interior orientation of the camera is given by $(u_o, v_o, f)^T$.

(*) Camera coordinate system origin is at center of projection.

(**) Camera coordinate system is aligned with image axes.

Planar object:

Let $z_t = 0$ in the plane of the object.

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x_t \\ y_t \\ 0 \end{pmatrix} + \begin{pmatrix} x_o \\ y_o \\ z_o \end{pmatrix} \tag{3}$$

Absorbing the translation $(x_o, y_o, z_o)^T$ into the 3×3 matrix, and dividing the third component by f we get:

$$\begin{pmatrix} x_c \\ y_c \\ z_c/f \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & x_o \\ r_{21} & r_{22} & y_o \\ r_{31}/f & r_{32}/f & z_o/f \end{pmatrix} \begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix} \tag{4}$$

Now perspective projection (eq. 2) gives (if we let $k = z_c/f$):

$$\begin{pmatrix} k(u - u_o) \\ k(v - v_o) \\ k \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & x_o \\ r_{21} & r_{22} & y_o \\ r_{31}/f & r_{32}/f & z_o/f \end{pmatrix} \begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix} \tag{5}$$

(Odd form makes it easier to match result with projective geometry formulation).

Homogeneous Representation:

Homogeneous representation of point in plane uses three numbers [Wylie 70].

$$(u, v, w)^T$$

Actual planar coordinates are obtained by dividing first two elements by third:

$$x = u/w \quad \text{and} \quad y = v/w$$

Representation *not* unique since $(ku, kv, kw)^T$ corresponds to same point.

A 3×3 matrix T can represent a homogeneous transformation
from the object plane to the image plane.

$$\begin{pmatrix} ku \\ kv \\ k \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix} \tag{6}$$

Matching Homogeneous Transformation:

We can match

$$\begin{pmatrix} k(u-u_o) \\ k(v-v_o) \\ k \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & x_o \\ r_{21} & r_{22} & y_o \\ r_{31}/f & r_{32}/f & z_o/f \end{pmatrix} \begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix}, \tag{5}$$

with

$$\begin{pmatrix} ku \\ kv \\ k \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix} \tag{6}$$

provided

$$\begin{aligned} t_{11} &= r_{11}, \quad t_{12} = r_{12}, \quad t_{13} = x_o \\ t_{21} &= r_{21}, \quad t_{22} = r_{22}, \quad t_{23} = y_o \\ t_{31} &= r_{31}/f, \quad t_{32} = r_{32}/f, \quad t_{33} = z_o/f \end{aligned} \tag{7}$$

(In *addition*, measurements in the image must be made in a coordinate system with the origin at the principal point, so that u_o, $v_o = 0$.)

Constraints on Transform:

T same as kT (Scale factor ambiguity). Pick $t_{33} = 1$ (leaves 8 DOF).

T *must* satisfy two non-linear constraints

$$t_{11}t_{12} + t_{21}t_{22} + f^2 t_{31}t_{32} = 0 \tag{8}$$

and

$$t_{11}t_{11} + t_{21}t_{21} + f^2 t_{31}t_{31} = t_{12}t_{12} + t_{22}t_{22} + f^2 t_{32}t_{32} \tag{9}$$

(from the orthonormality of R).

Two non-linear constraints reduce DOF from 8 to 6 —

rotation has 3 DOF and translation has 3 DOF.

Projective Transformation versus Perspective Projection:

Given real perspective projection, one can always compute a matrix T.

But it is *not* possible to go in the other direction —
for most matrices T, there is no corresponding perspective projection.

Only a subset of measure of zero of homogeneous transformations T
allow physical interpretation as rigid body motion and perspective projection.

An arbitrary 3×3 matrix T will *not* satisfy the two non-linear constraints.

Disallowed Mappings:

Mappings allowed by projective geometry but not by perspective projection —

skewing and anisotropic scaling

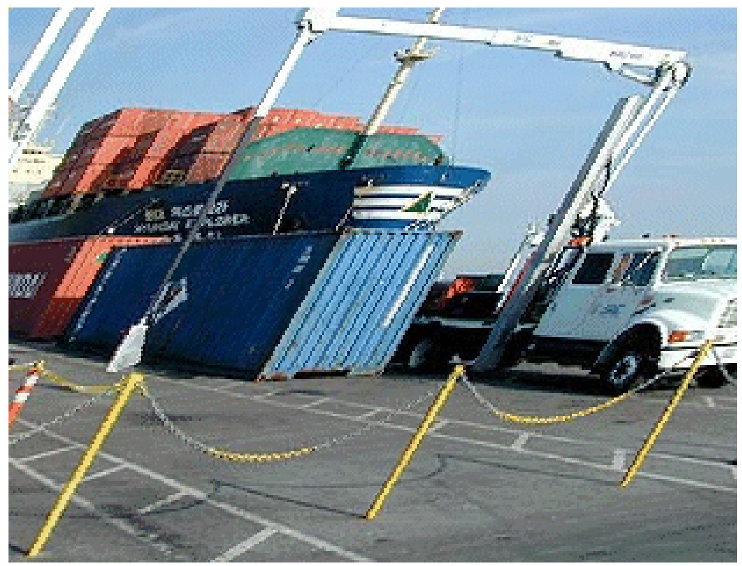
Distort a normal perspective image by the additional operations

$$\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 1 & 0 \\ 0 & k \end{pmatrix}$$

result is not an “image” —
from *any* position with *any* camera orientation.

Yet such distortions merely transform matrix T —
thus *are* allowed in projective geometry.

Disallowed Mappings:



Picture of a Picture:

Is there a physical imaging situation that corresponds to homogeneous transformation by an arbitrary matrix T?

> The transformations of perspective geometry correspond to taking a perspective image of a perspective image.

In this case, the overall transformation need *not* satisfy the non-linear constraints.

But, we are interested in “direct” images, *not* pictures taken of a picture!

Finding T:

Correspondence between $(x_i, y_i, 1)^T$ and $(ku_i, kv_i, k)^T$ (Horn 86):

$$x_i t_{11} + y_i t_{12} + t_{13} - x_i u_i t_{31} - y_i u_i t_{32} - u_i t_{33} = 0$$
$$x_i t_{21} + y_i t_{22} + t_{23} - x_i v_i t_{31} - y_i v_i t_{32} - v_i t_{33} = 0$$

Four correspondences yield system of 8 homogeneous equations:

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 & -u_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -y_2u_2 & -u_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3u_3 & -y_3u_3 & -u_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4u_4 & -y_4u_4 & -u_4 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1v_1 & -y_1v_1 & -v_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2v_2 & -y_2v_2 & -v_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3v_3 & -y_3v_3 & -v_3 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4v_4 & -y_4v_4 & -v_4 \end{pmatrix} \begin{pmatrix} t_{11} \\ t_{12} \\ t_{13} \\ t_{21} \\ t_{22} \\ t_{23} \\ t_{31} \\ t_{32} \\ t_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

T has 9 elements, but only 8 DOF, since any multiple of T describes same mapping.

Arbitrarily pick value for one element of T, say $t_{33} = 1$.

Leaves 8 *non-homogeneous* equations in 8 unknown.

Recovering Orientation using Vanishing Points

Consider

$$(\alpha a, \alpha b, 1)^T$$

as $\alpha \rightarrow \infty$. That is, $(a, b, 0)^T$.

The vanishing point for line with direction $(a, b)^T$ in object plane is

$$\begin{aligned} u &= (t_{11}a + t_{12}b)/(t_{31}a + t_{32}b) \\ v &= (t_{21}a + t_{22}b)/(t_{31}a + t_{32}b) \end{aligned} \tag{10}$$

The line from the COP $(0, 0, 0)^T$, to vanishing point in image plane $(u, v, f)^T$, is parallel (in 3D) to line on object.

Apply to x- and y-axes of object, and from the two vanishing points find directions of the two axes in camera coordinate system.

Recovering Orientation using Vanishing Points

Vanishing point for x-axis is $(1, 0, 0)^T$ in object coordinate system.

$$T(1, 0, 0)^T = (t_{11}, t_{21}, t_{31})^T$$

Vanishing point for y-axis is $(0, 1, 0)^T$ in object coordinate system.

$$T(0, 1, 0)^T = (t_{12}, t_{22}, t_{32})^T$$

These correspond to $(t_{11}/t_{31}, t_{21}/t_{31})^T$ and $(t_{12}/t_{32}, t_{22}/t_{32})^T$ in the image.

Connect COP $(0, 0, 0)^T$, to the vanishing points in the image plane

$$\begin{aligned} \mathbf{x} &= (t_{11}, t_{21}, f t_{31})^T, \\ \mathbf{y} &= (t_{12}, t_{22}, f t_{32})^T. \end{aligned} \tag{11}$$

Recovering Rotation

Make unit vectors in the direction of the x- and y-axes of the object plane.

$$\hat{\mathbf{x}} = \mathbf{x} / \|\mathbf{x}\| \quad \text{and} \quad \hat{\mathbf{y}} = \mathbf{y} / \|\mathbf{y}\|$$

So if f is known, directions of object coordinate axes (in camera coordinates) can be found from first two columns of T.

Construct triad using

$$\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}}$$

This leads to

$$R^T = \begin{pmatrix} \hat{\mathbf{x}}^T \\ \hat{\mathbf{y}}^T \\ \hat{\mathbf{z}}^T \end{pmatrix} \quad \text{or} \quad R = \begin{pmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \end{pmatrix} \tag{12}$$

Recovering Direction of Translation

Homogeneous coordinates of the origin on the object plane: $(0, 0, 1)^T$

$$T(0, 0, 1)^T = (t_{13}, t_{23}, t_{33})^T$$

Image of origin is at $(t_{13}/t_{33}, t_{23}/t_{33})^T$.

Connecting origin to this point yields vector parallel to

$$\mathbf{t} = (t_{13}, t_{23}, f t_{33})^T. \qquad (13)$$

This is the *direction* of the translational vector to the object origin obtained directly from last column of T.

Recovering Translation

The z component of translation is (f/M) —
where M is lateral magnification.

Need line in object plane that is parallel to the image plane —
take cross-product of $\hat{\mathbf{z}}$ found above and $(0,0,1)^T$:

$$(z_1, z_2, z_3)^T \times (0,0,1)^T = (-z_2, z_1, 0)^T$$

This point on object is imaged at $T(-z_2, z_1, 0)^T$

Find length of line l_i from there to image of origin, at $(t_{13}/t_{33}, t_{23}/t_{33})^T$

Divide by $l_t = \sqrt{z_1^2 + z_2^2}$ to find $M = l_i/l_t$.

Finally, $\mathbf{t} = M(t_{13}, t_{23}, f t_{33})^T/(f t_{33})$

Analysis

In general T does not correspond to rigid body motion and perspective projection.

Why were we able to calculate rigid body motion R and $\mathbf{t}$?

We selectively neglected "inconvenient" information in T!

Now construct T' based on recovered R and $\mathbf{t}$.

In general, $T' \neq T$!

Even worse, $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, as constructed, need not be orthogonal.

In general R, as constructed, is not a rotation!

For R to be orthonormal, $(t_{11}, t_{21}, f t_{31})^T \cdot (t_{12}, t_{22}, f t_{32})^T = 0$.

Least squares estimate of T from data does not ensure this.

Can find "nearest" orthogonal $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, but...

Same story for the other non-linear constraint on elements of T.

Sensitivity to Error in Measurements.

With perfect measurements, T has desired properties.

What is error in R and $\mathbf{t}$ as function of error in measurements?

Difficult to find analytically because of complex steps.

Error in rotation: Angle of $\delta R = R_1^T R_2$.

Monte Carlo simulation

Results:

Sensitivity depends on FOV.

Sensitivity depends on orientation of planar target.

Sensitivity depends on choice of origin!

Two Step Optimization

(1) Find linear transformation T that best fits the data.

(2) Find rigid body transformation R and $\mathbf{t}$ "nearest" to T.

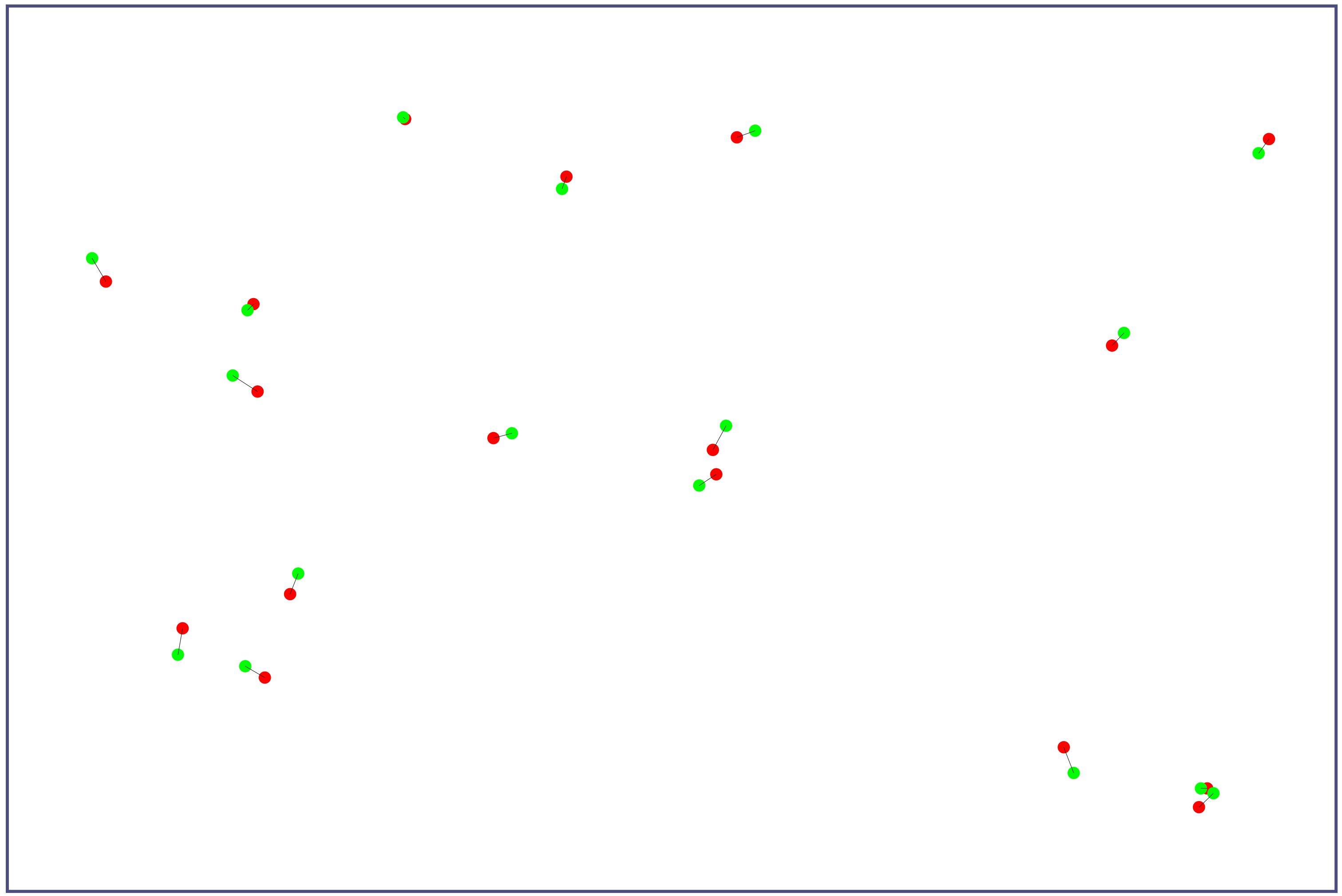
One Step Optimization

Adjust R and $\mathbf{t}$ directly —
to minimize sum of squares of errors in image position.

The results are different.

Least squares adjustment to find a best fit T in two-step method
minimizes quantities without any direct physical significance.

Can be interpreted as weighted sum of image errors —
but with arbitrary weights.



Minimize Image Errors:

- Observed Points
- Predicted Points — based on R and **t**

Minimizing the Image Projection Error

Properly model rigid body motion and perspective projection.

Error is image plane distance between predicted and observed features.

Non-linear problem. No closed form solution.

Find R and $\mathbf{t}$ that minimizes sum of squares of image errors.

Easy optimization — if good representation for rotation is chosen.

Unit quaternions (equivalently Pauli Spin matrices, Euler *parameters*, etc.)

Newton-Raphson method is then sufficient.

Seven unknowns ($\mathring{q}$ and $\mathbf{t}$) and one constraint ($\mathring{q} \cdot \mathring{q} = 1$).

Comparing Error Sensitivity

Sample Experiment:

Target: Four features in 168 mm by 168 mm pattern
Distance: 1600 mm
Focal length: 18mm
Pixel size: 8.4 μ by 8.4 μ
Object Plane Orientation: normal 60° from optical axis
Error in image plane measurements: 0.05 pixel (1/20)

Monte Carlo simulation.

Projective geometry (PG) error is 1.1°.
Noise gain 22 degrees / pixel.

Perspective projection (PP) method error is 0.04°.
Noise gain 0.8 degrees / pixel.

Projective Geometry based method more than an order of magnitude worse.

With *three* points, PP method error is still only 0.06°.

With more points, errors decrease in proportion to $\approx 1/\sqrt{N}$

Projective Geometry based method bad under variety of conditions tested.

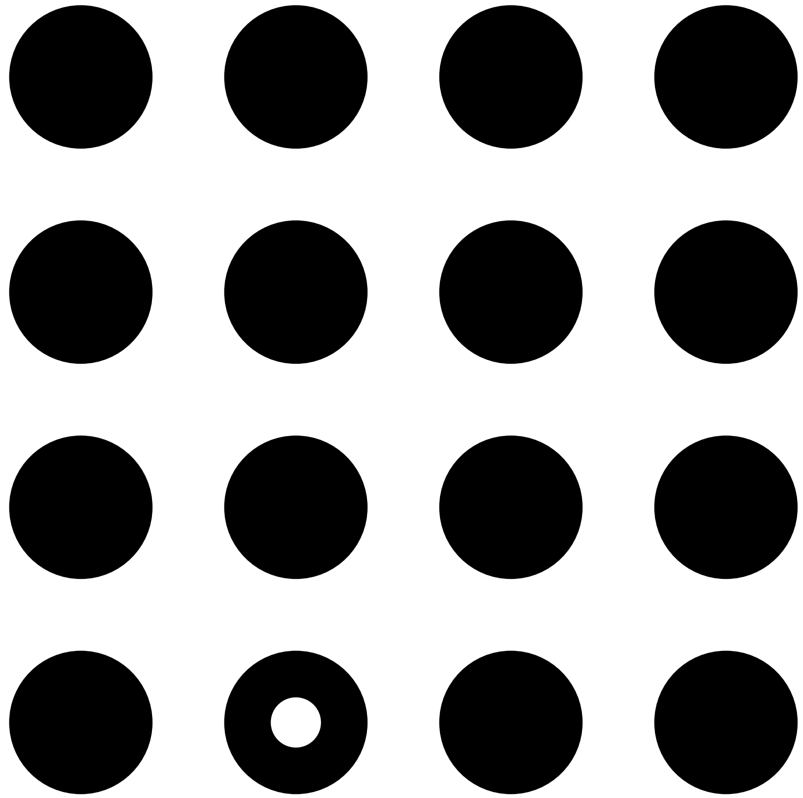


Image Analyst Pattern

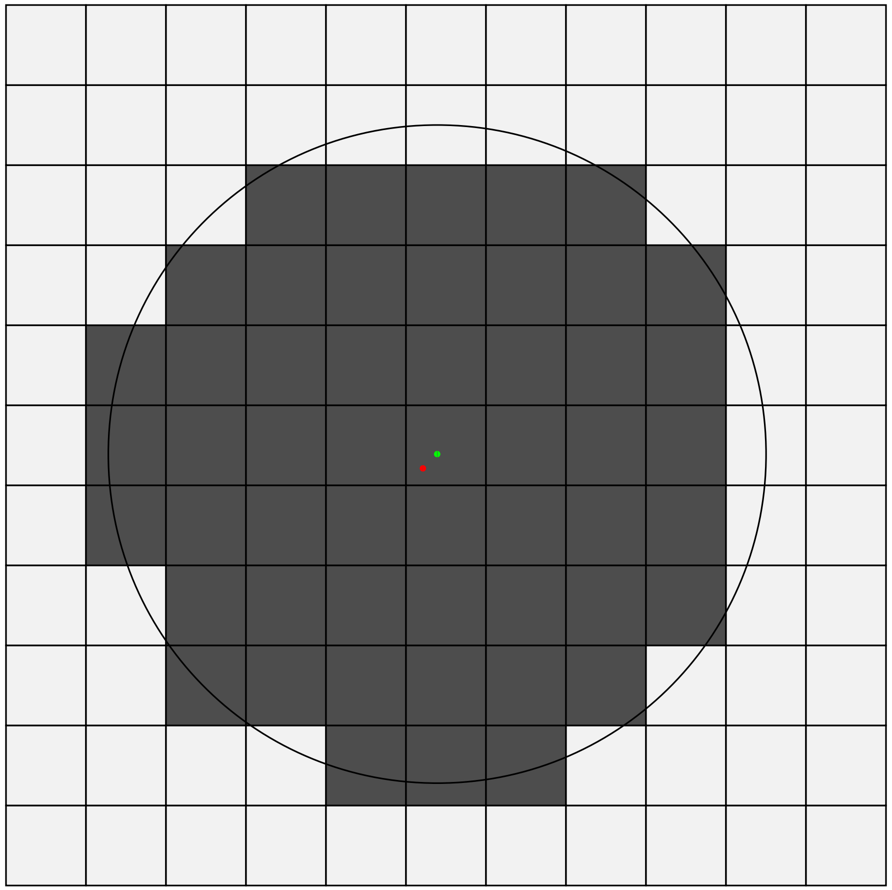
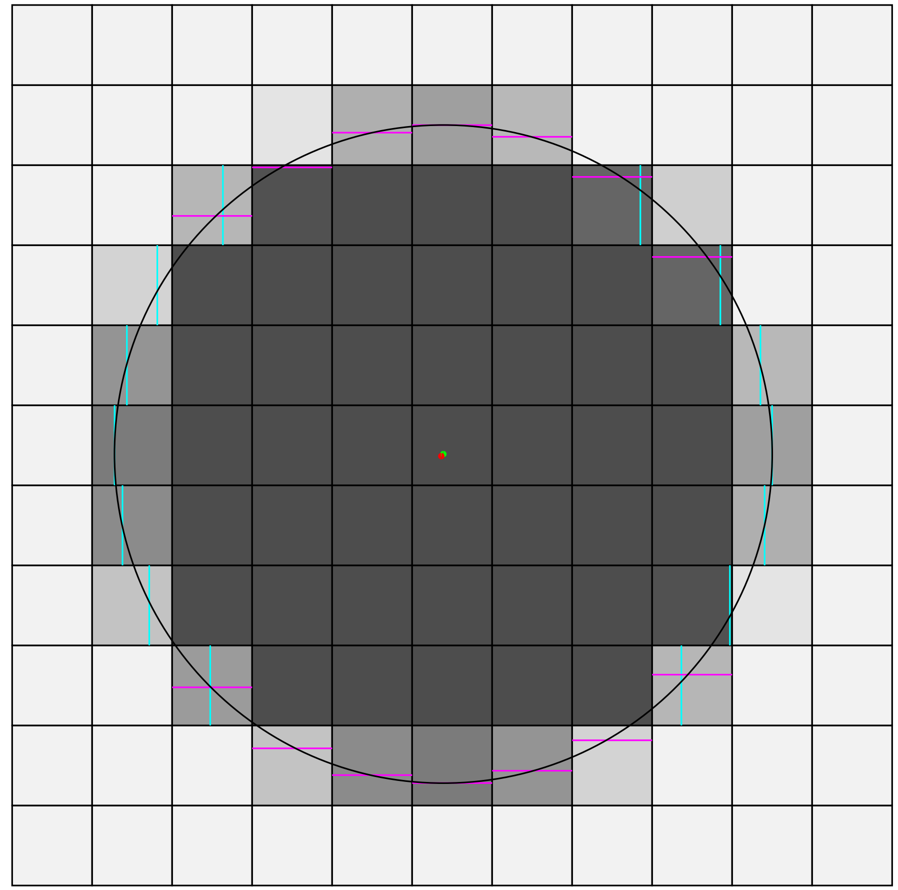


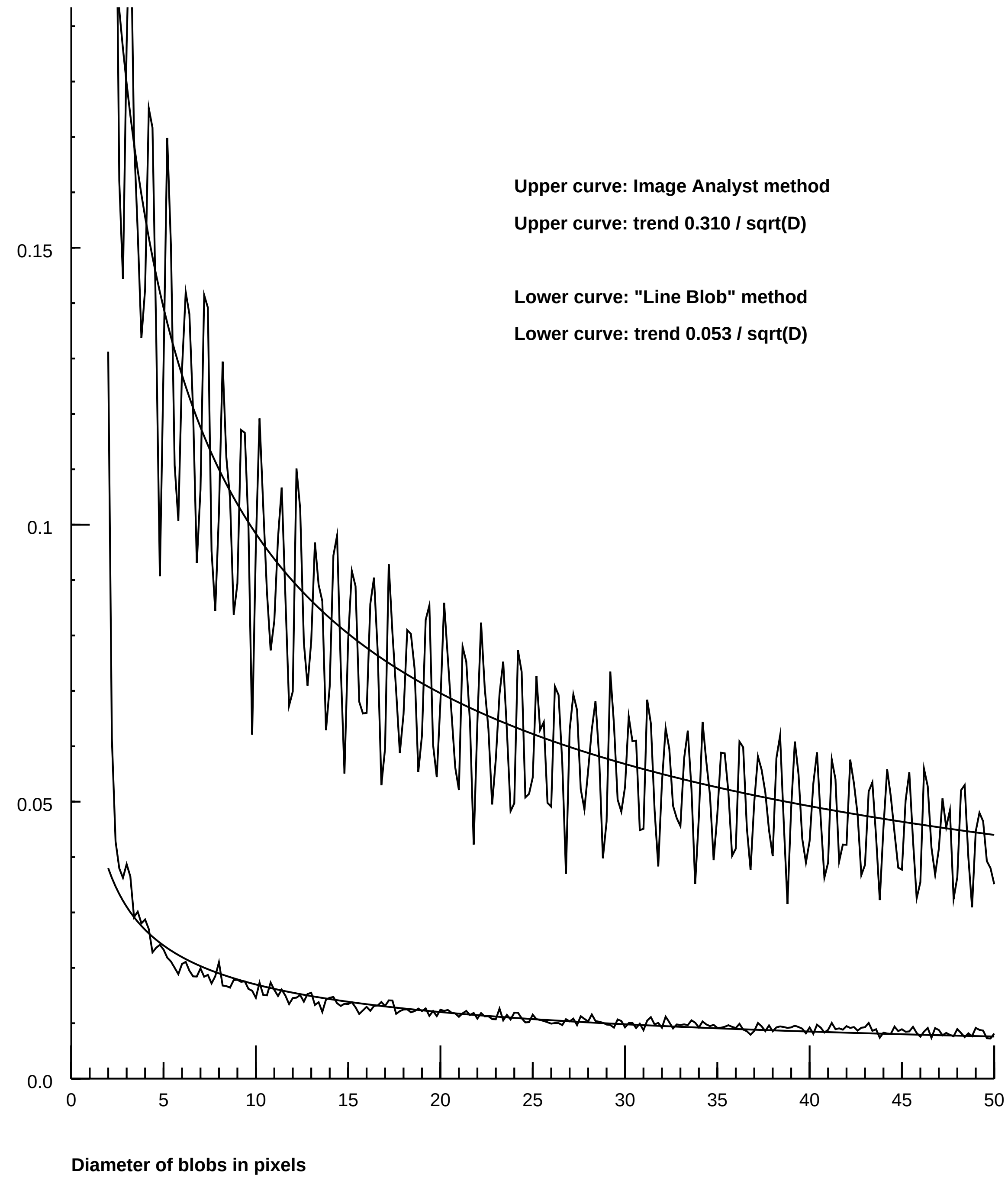
Image Analyst Centroiding Method

- Green dot: True centroid of original disk
- Red dot: Centroid of thresholded binary image



Line Blob Centroiding Method

- Green dot: True centroid of original disk
- Red dot: Line blob estimate of centroid



Upper curve: Image Analyst method
Upper curve: trend 0.310 / sqrt(D)
Lower curve: "Line Blob" method
Lower curve: trend 0.053 / sqrt(D)
0.15
0.1
0.05
0.0
0
5
10
15
20
25
30
35
40
45
50
Diameter of blobs in pixels

Main Algorithm (patent disclosure)

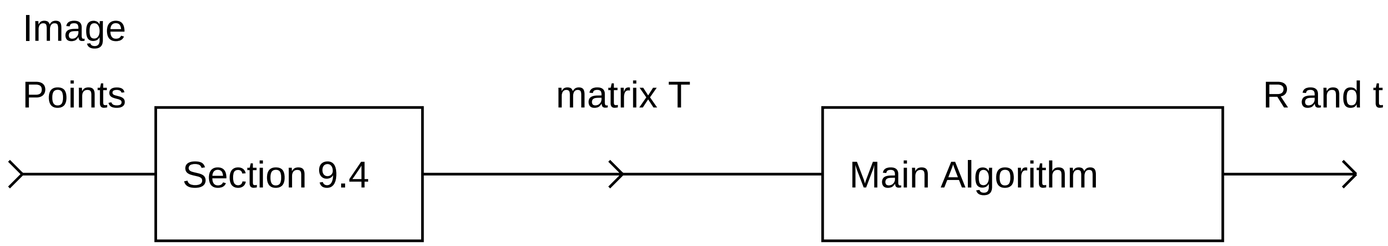
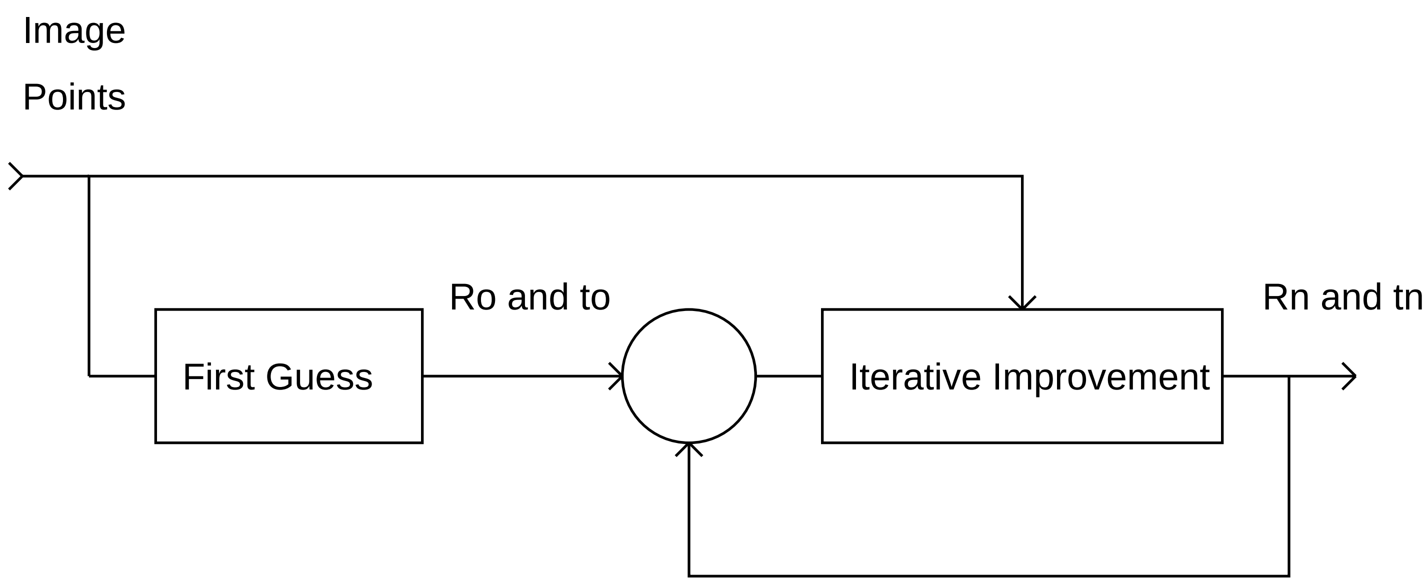
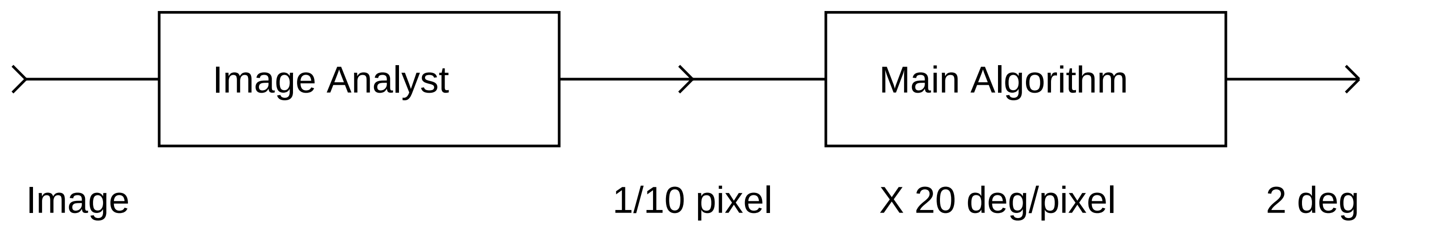


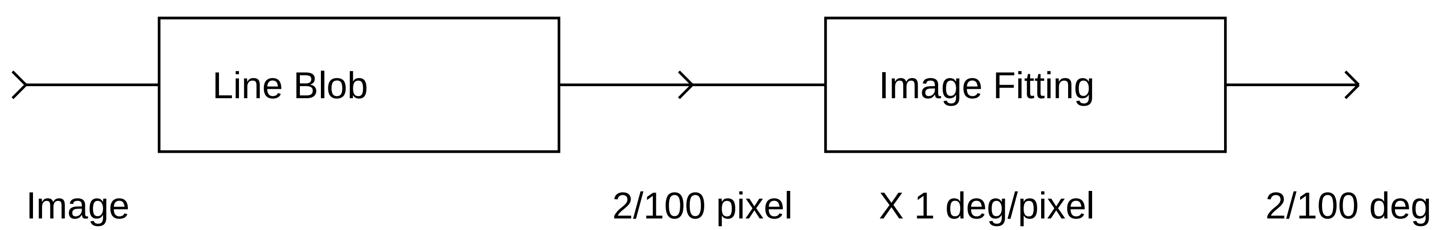
Image Fitting (working implementation)



Patent Disclosure



Working Implementation



Summary

Methods based on projective geometry are fundamentally different from methods based on perspective projection;

Methods based on projective geometry yield a transformation matrix T that in general does *not* correspond to a physical imaging situation — that is, rotation, translation, and perspective projection;

Optimization methods based on actual physical imaging situation produce *considerably* more accurate results.

Additional Notes

One *can* make very accurate measurements using images;

One needs to carefully design the target to get high accuracy;

One needs to carefully design the target to get reliability.

References:

Absolute Orientation:

"Closed Form Solution of Absolute Orientation using Unit Quaternions,"
Journal of the Optical Society A, Vol. 4, No. 4, pp. 629–642, Apr. 1987.

Relative Orientation:

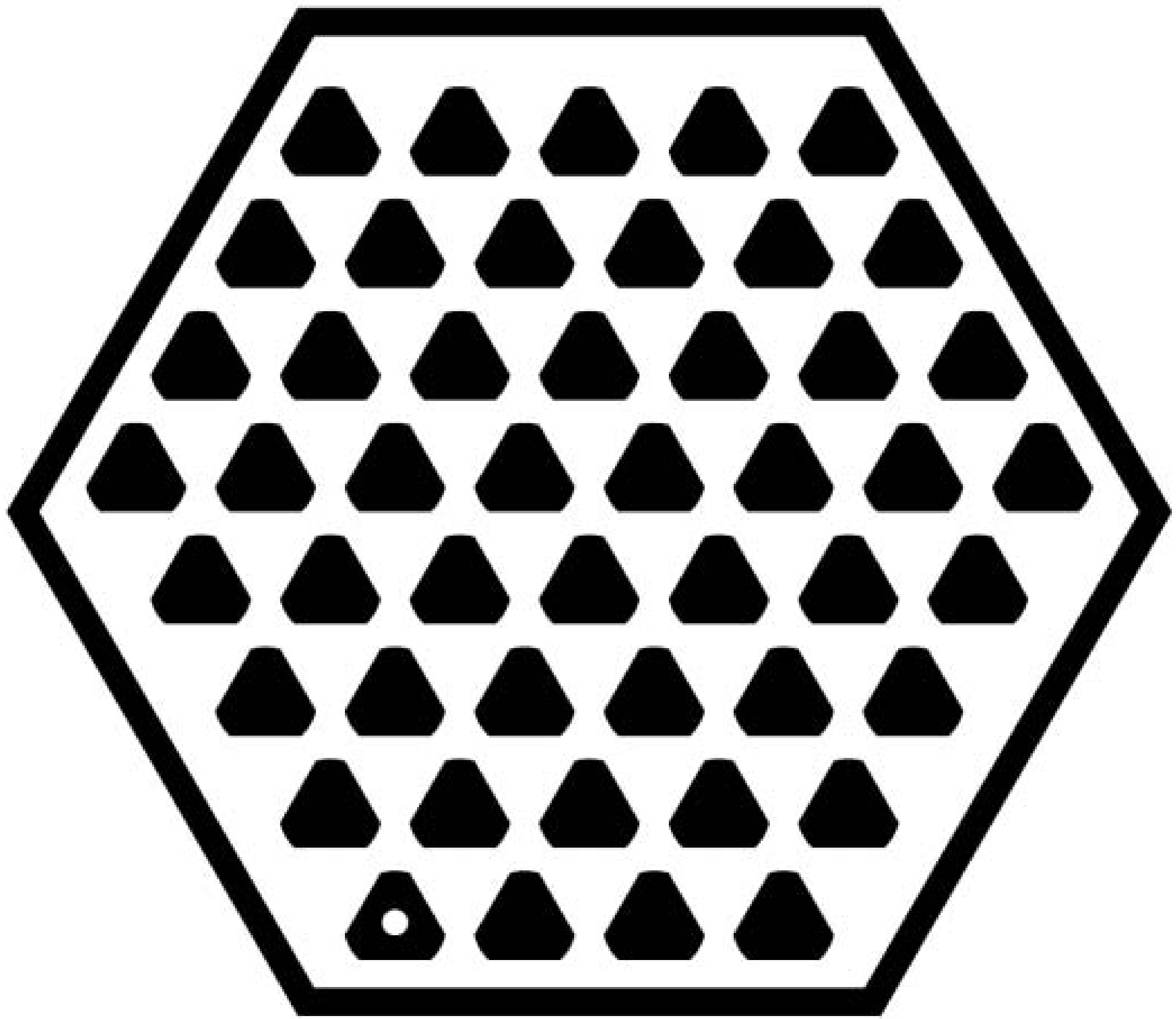
"Relative Orientation Revisited,"
Journal of the Optical Society of America, A, Vol. 8, pp. 1630–1638, Oct. 1991.

Interior Orientation / Exterior Orientation:

"Tsai's Camera Calibration Method Revisited,"
on `http:\\www.ai.mit.edu\people\bkph`

Detailed Example:

"Projective Geometry Considered Harmful,"
on `http:\\www.ai.mit.edu\people\bkph`



Projective Geometry Considered Harmful

Berthold K.P. Horn