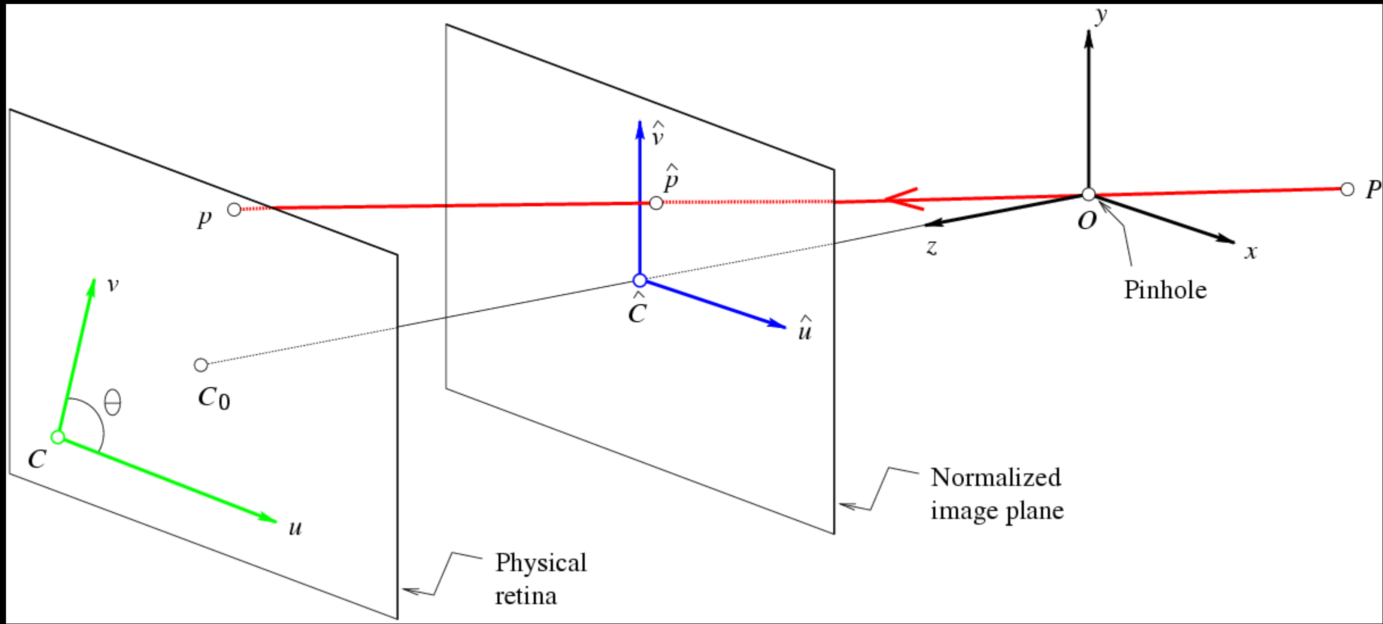
Euclidean cameras and strong (Euclidean) calibration

- Linear least-squares methods
- Linear calibration
- Degenerate point configurations
- Analytical photogrammetry
- A quick detour through stereopsis

Planches :

- http://www.di.ens.fr/~ponce/geomvis/lect2.pptx
- http://www.di.ens.fr/~ponce/geomvis/lect2.pdf

The Intrinsic Parameters of a Camera



Calibration Matrix

$$\boldsymbol{p} = \mathcal{K}\hat{\boldsymbol{p}}, \quad \text{where} \quad \boldsymbol{p} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad \text{and} \quad \mathcal{K} \stackrel{\text{def}}{=} \begin{pmatrix} \alpha & -\alpha \cot\theta & u_0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

The Perspective Projection Equation

$$\boldsymbol{p} = \frac{1}{z}\mathcal{M}\boldsymbol{P}, \quad \text{where} \quad \mathcal{M} \stackrel{\text{def}}{=} \begin{pmatrix} \mathcal{K} & \mathbf{0} \end{pmatrix}$$

The Extrinsic Parameters of a Camera

- When the camera frame (C) is different from the world frame (W),

$$\begin{pmatrix} {}^{C}P \\ 1 \end{pmatrix} = \begin{pmatrix} {}^{C}_{W}\mathcal{R} & {}^{C}O_{W} \\ \mathbf{0}^{T} & 1 \end{pmatrix} \begin{pmatrix} {}^{W}P \\ 1 \end{pmatrix}.$$

- Thus,

$$\boxed{\boldsymbol{p} = \frac{1}{z}\mathcal{M}\boldsymbol{P},} \quad \text{where} \quad \begin{cases} \mathcal{M} = \mathcal{K}\begin{pmatrix} \mathcal{R} & \boldsymbol{t} \end{pmatrix}, \\ \mathcal{R} = {}^{C}_{W}\mathcal{R}, \\ \boldsymbol{t} = {}^{C}O_{W}, \\ \boldsymbol{P} = \begin{pmatrix} {}^{W}P \\ 1 \end{pmatrix}. \end{cases}$$

$$\boxed{p \approx M\,P}$$

- Note: z is *not* independent of $\mathcal{M}$ and $\boldsymbol{P}$:

$$\mathcal{M} = \begin{pmatrix} \boldsymbol{m}_1^T \\ \boldsymbol{m}_2^T \\ \boldsymbol{m}_3^T \end{pmatrix} \Longrightarrow z = \boldsymbol{m}_3 \cdot \boldsymbol{P}, \quad \text{or} \quad \begin{cases} u = \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}, \\ v = \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}. \end{cases}$$

Explicit Form of the Projection Matrix

$$\mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}$$

Note: If $\mathcal{M} = (\,\mathcal{A} \quad \boldsymbol{b}\,)$ then $|\boldsymbol{a}_3| = 1$.

Replacing $\mathcal{M}$ by $\lambda\mathcal{M}$ in

$$\begin{cases} u = \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}} \\ v = \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}} \end{cases}$$

does not change u and v.

M is only defined up to scale in this setting !!

Linear Camera Calibration

Given n points $P_1, \ldots, P_n$ with *known* positions and their images $p_1, \ldots, p_n$

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \\ \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \end{pmatrix} \iff \begin{pmatrix} \boldsymbol{m}_1 - u_i \boldsymbol{m}_3 \\ \boldsymbol{m}_2 - v_i \boldsymbol{m}_3 \end{pmatrix} \boldsymbol{P}_i = 0$$

$$\mathcal{P}\boldsymbol{m} = 0 \text{ with } \mathcal{P} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1^T & \boldsymbol{0}^T & -u_1 \boldsymbol{P}_1^T \\ \boldsymbol{0}^T & \boldsymbol{P}_1^T & -v_1 \boldsymbol{P}_1^T \\ \ldots & \ldots & \ldots \\ \boldsymbol{P}_n^T & \boldsymbol{0}^T & -u_n \boldsymbol{P}_n^T \\ \boldsymbol{0}^T & \boldsymbol{P}_n^T & -v_n \boldsymbol{P}_n^T \end{pmatrix} \text{ and } \boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix} = 0$$

How do you solve overconstrained homogeneous linear equations ??

$$E = |\mathcal{U}\boldsymbol{x}|^2 = \boldsymbol{x}^T(\mathcal{U}^T\mathcal{U})\boldsymbol{x}$$

- Orthonormal basis of eigenvectors: $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_q$.
- Associated eigenvalues: $0 \leq \lambda_1 \leq \ldots \leq \lambda_q$.
- Any vector can be written as

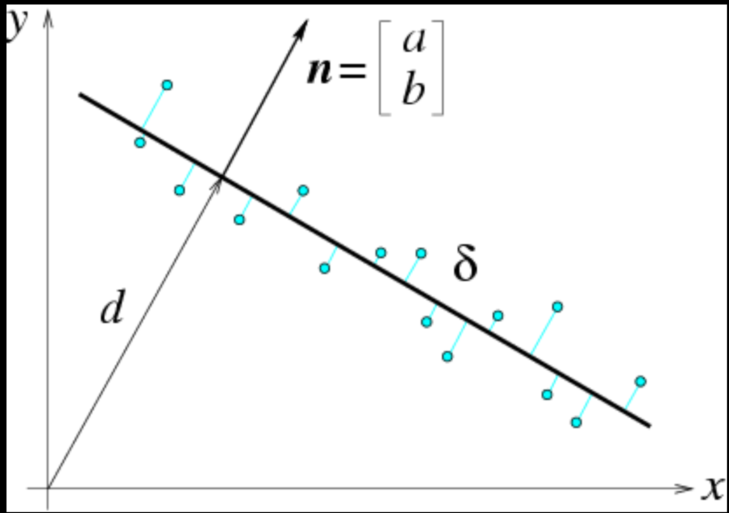
$$\boldsymbol{x} = \mu_1 \boldsymbol{e}_1 + \ldots + \mu_q \boldsymbol{e}_q$$

for some μ_i $(i = 1, \ldots, q)$ such that $\mu_1^2 + \ldots + \mu_q^2 = 1$.

$$\begin{aligned} E(\boldsymbol{x}) - E(\boldsymbol{e}_1) &= \boldsymbol{x}^T(U^TU)\boldsymbol{x} - \boldsymbol{e}_1^T(U^TU)\boldsymbol{e}_1 \\ &= \lambda_1\mu_1^2 + \ldots + \lambda_q\mu_q^2 - \lambda_1 \\ &\geq \lambda_1(\mu_1^2 + \ldots + \mu_q^2 - 1) = 0 \end{aligned}$$

$\boxed{\text{The solution is } \boldsymbol{e}_1}$.

Example: Line Fitting



Problem: minimize

$$E(a,b,d) = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

with respect to (*a*,*b*,*d*).

- Minimize *E* with respect to *d*:

$$\frac{\partial E}{\partial d} = 0 \Longrightarrow d = \sum_{i=1}^{n} \frac{ax_i + by_i}{n} = a\bar{x} + b\bar{y}$$

- Minimize *E* with respect to *a*,*b*:

$$E = \sum_{i=1}^{n} [a(x_i - \bar{x}) + b(y_i - \bar{y})]^2 = |\mathcal{U}\boldsymbol{n}|^2$$ where $$\mathcal{U} = \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \dots & \dots \\ x_n - \bar{x} & y_n - \bar{y} \end{pmatrix}$$

- Done !!

Note:

$$\mathcal{U}^T\mathcal{U} = \begin{pmatrix} \sum_{i=1}^{n} x_i^2 - n\bar{x}^2 & \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} \\ \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} & \sum_{i=1}^{n} y_i^2 - n\bar{y}^2 \end{pmatrix}$$

- Matrix of second moments of inertia

- Axis of least inertia

Linear Camera Calibration

Given n points $P_1, \ldots, P_n$ with *known* positions and their images $p_1, \ldots, p_n$

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \\ \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \end{pmatrix} \Longleftrightarrow \begin{pmatrix} \boldsymbol{m}_1 - u_i \boldsymbol{m}_3 \\ \boldsymbol{m}_2 - v_i \boldsymbol{m}_3 \end{pmatrix} \boldsymbol{P}_i = 0$$

Linear least squares for $n > 5$!

$$\boxed{\mathcal{P}\boldsymbol{m} = 0} \text{ with } \mathcal{P} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1^T & \boldsymbol{0}^T & -u_1 \boldsymbol{P}_1^T \\ \boldsymbol{0}^T & \boldsymbol{P}_1^T & -v_1 \boldsymbol{P}_1^T \\ \ldots & \ldots & \ldots \\ \boldsymbol{P}_n^T & \boldsymbol{0}^T & -u_n \boldsymbol{P}_n^T \\ \boldsymbol{0}^T & \boldsymbol{P}_n^T & -v_n \boldsymbol{P}_n^T \end{pmatrix} \text{ and } \boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix} = 0$$

Once *M* is known, you still got to recover the intrinsic and extrinsic parameters !!!

This is a decomposition problem, not an estimation problem.

$$\rho\, \mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}$$

- Intrinsic parameters
- Extrinsic parameters

Degenerate Point Configurations

Are there other solutions besides *M* ??

$$\mathbf{0} = \mathcal{P}\boldsymbol{l} = \begin{pmatrix} \boldsymbol{P}_1^T & \mathbf{0}^T & -u_1\boldsymbol{P}_1^T \\ \mathbf{0}^T & \boldsymbol{P}_1^T & -v_1\boldsymbol{P}_1^T \\ \dots & \dots & \dots \\ \boldsymbol{P}_n^T & \mathbf{0}^T & -u_n\boldsymbol{P}_n^T \\ \mathbf{0}^T & \boldsymbol{P}_n^T & -v_n\boldsymbol{P}_n^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \boldsymbol{\nu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{P}_1^T\boldsymbol{\lambda} - u_1\boldsymbol{P}_1^T\boldsymbol{\nu} \\ \boldsymbol{P}_1^T\boldsymbol{\mu} - v_1\boldsymbol{P}_1^T\boldsymbol{\nu} \\ \dots \\ \boldsymbol{P}_n^T\boldsymbol{\lambda} - u_n\boldsymbol{P}_n^T\boldsymbol{\nu} \\ \boldsymbol{P}_n^T\boldsymbol{\mu} - v_n\boldsymbol{P}_n^T\boldsymbol{\nu} \end{pmatrix}$$

$$\begin{cases} \boldsymbol{P}_i^T\boldsymbol{\lambda} - \dfrac{\boldsymbol{m}_1^T\boldsymbol{P}_i}{\boldsymbol{m}_3^T\boldsymbol{P}_i}\boldsymbol{P}_i^T\boldsymbol{\nu} = 0 \\ \boldsymbol{P}_i^T\boldsymbol{\mu} - \dfrac{\boldsymbol{m}_2^T\boldsymbol{P}_i}{\boldsymbol{m}_3^T\boldsymbol{P}_i}\boldsymbol{P}_i^T\boldsymbol{\nu} = 0 \end{cases} \Rightarrow \begin{cases} \boldsymbol{P}_i^T(\boldsymbol{\lambda}\boldsymbol{m}_3^T - \boldsymbol{m}_1\boldsymbol{\nu}^T)\boldsymbol{P}_i = 0 \\ \boldsymbol{P}_i^T(\boldsymbol{\mu}\boldsymbol{m}_3^T - \boldsymbol{m}_2\boldsymbol{\nu}^T)\boldsymbol{P}_i = 0 \end{cases}$$

- Coplanar points: $(\lambda,\mu,\nu) = (\Pi,0,0)$ or $(0,\Pi,0)$ or $(0,0,\Pi)$
- Points lying on the intersection curve of two quadric surfaces = straight line + twisted cubic

Does not happen for 6 or more random points!

Analytical Photogrammetry

Given n points $P_1, \ldots, P_n$ with *known* positions and their images $p_1, \ldots, p_n$

Find $\boldsymbol{i}$ and $\boldsymbol{e}$ such that

$$\sum_{i=1}^{n} \left[\left(u_i - \frac{\boldsymbol{m}_1(\boldsymbol{i}, \boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i}, \boldsymbol{e}) \cdot \boldsymbol{P}_i} \right)^2 + \left(v_i - \frac{\boldsymbol{m}_2(\boldsymbol{i}, \boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i}, \boldsymbol{e}) \cdot \boldsymbol{P}_i} \right)^2 \right] \quad \text{is minimized}$$

Non-Linear Least-Squares Methods

- Newton
- Gauss-Newton
- Levenberg-Marquardt

Iterative, quadratically convergent in favorable situations

Triangulation

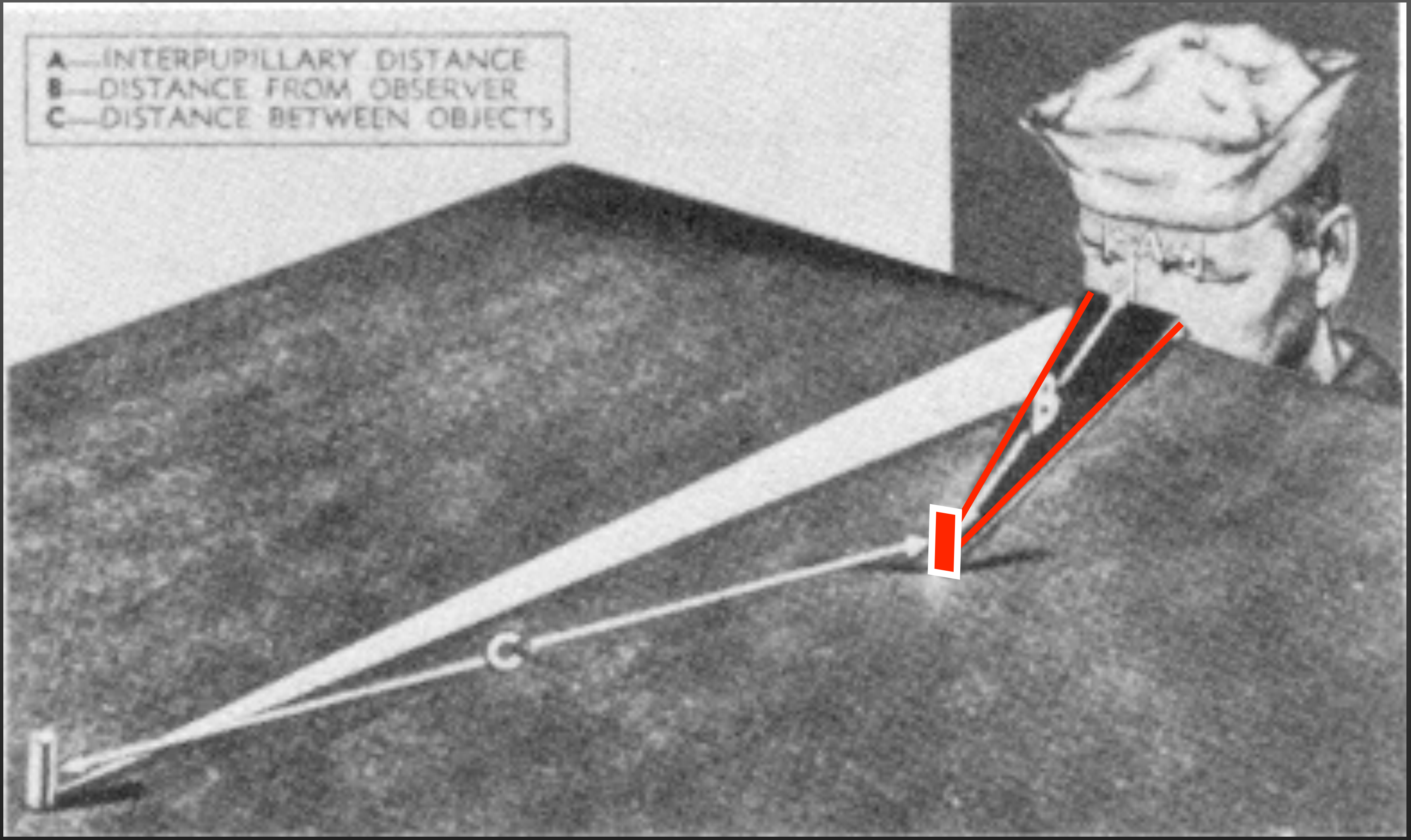


Figure extraite de "US Navy Manual of Basic Optics and Optical Instruments", Bureau of Naval Personnel. Reprinted by Dover Publications, Inc., 1969.

Triangulation

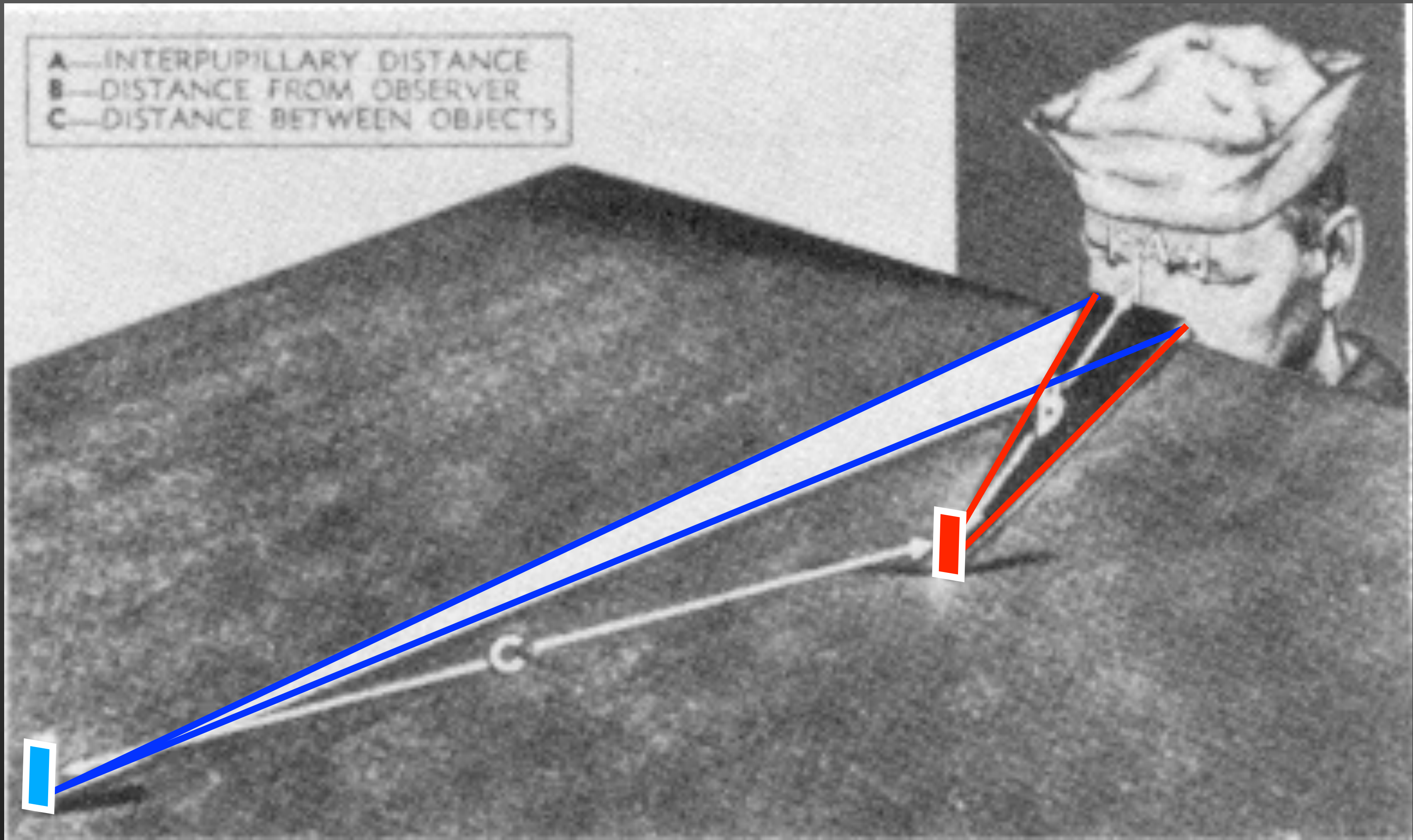


Figure extraite de "US Navy Manual of Basic Optics and Optical Instruments", Bureau of Naval Personnel. Reprinted by Dover Publications, Inc., 1969.

Why movies look "flat" on TV

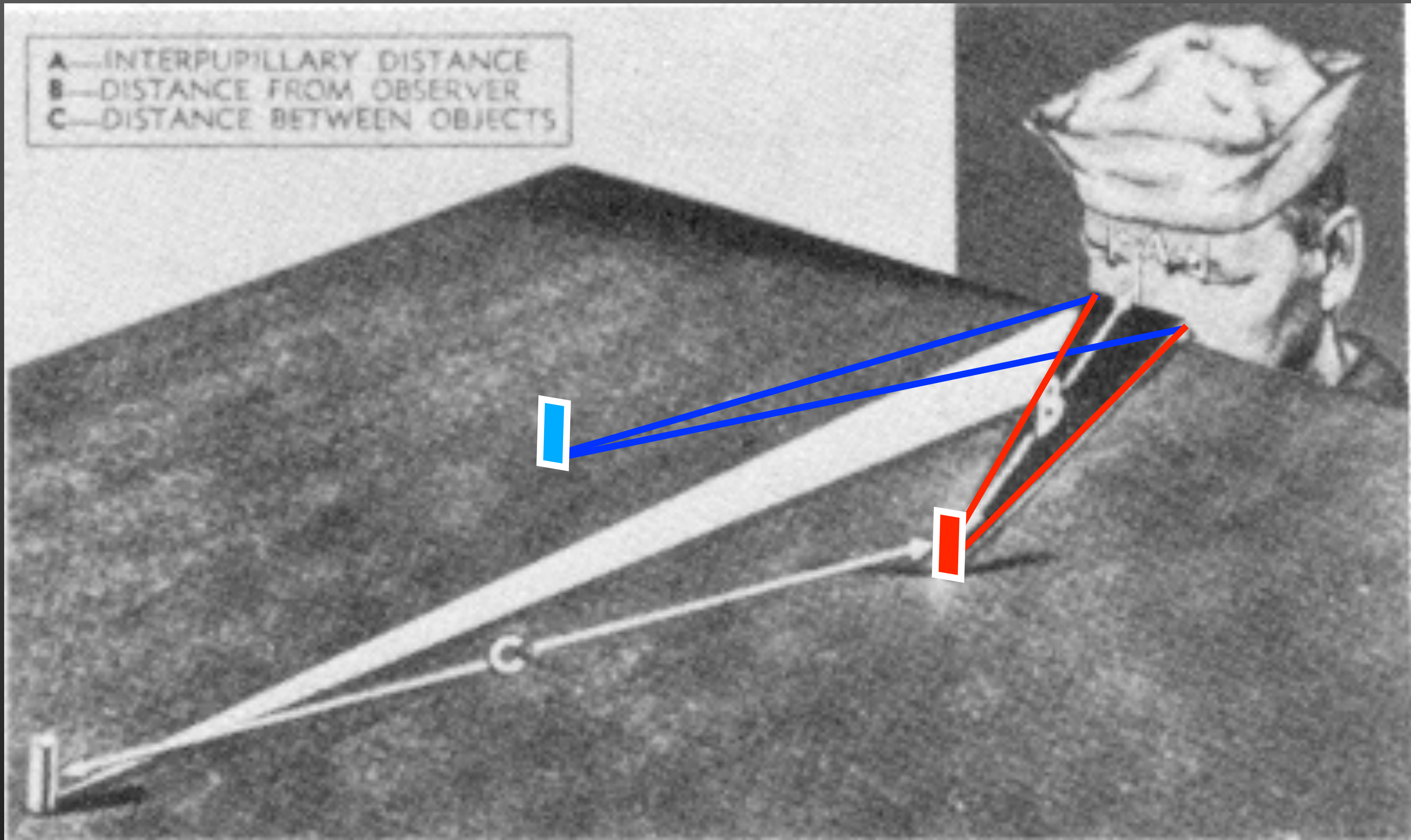


Figure extraite de "US Navy Manual of Basic Optics and Optical Instruments", Bureau of Naval Personnel. Reprinted by Dover Publications, Inc., 1969.

Why movies look "flat" on TV

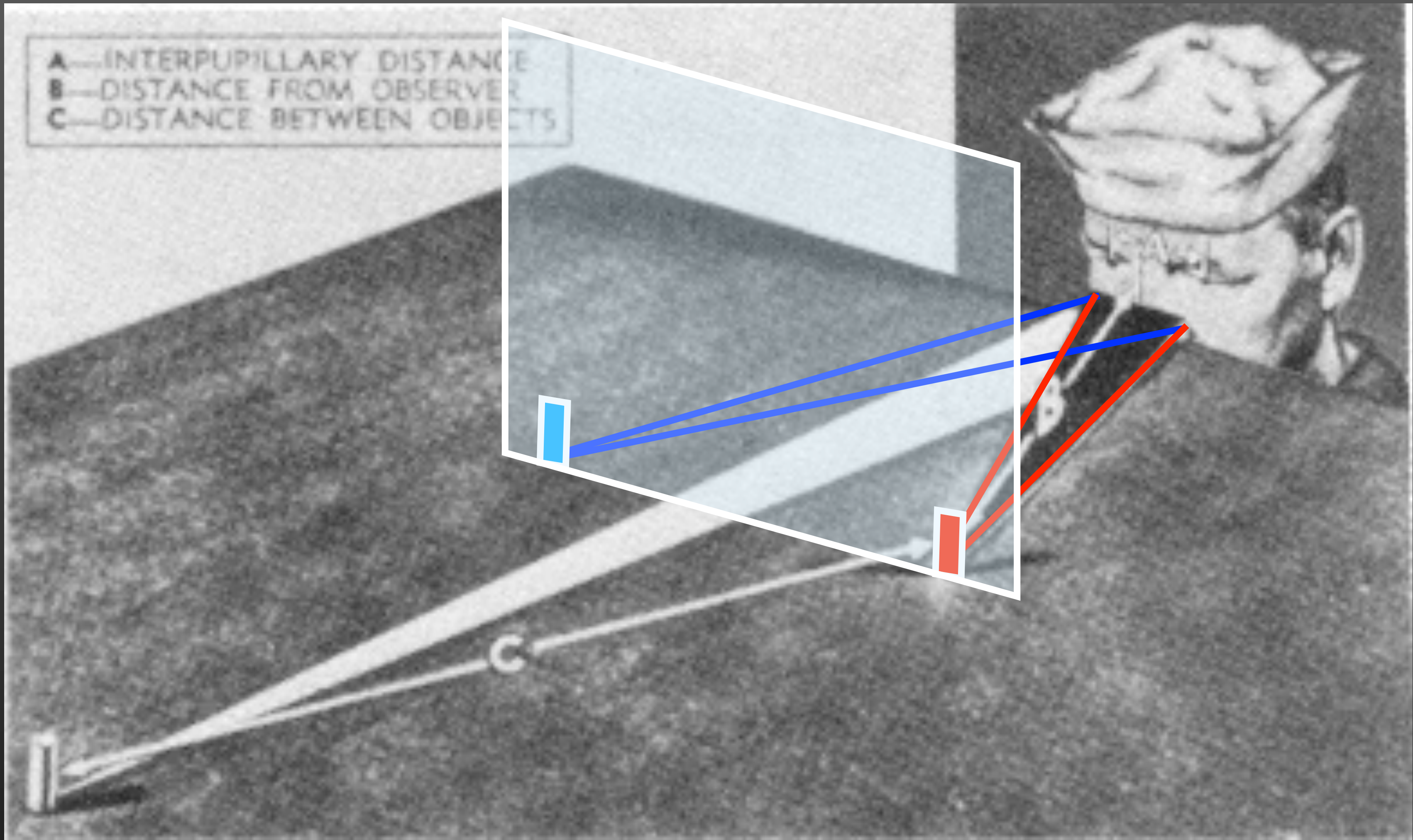
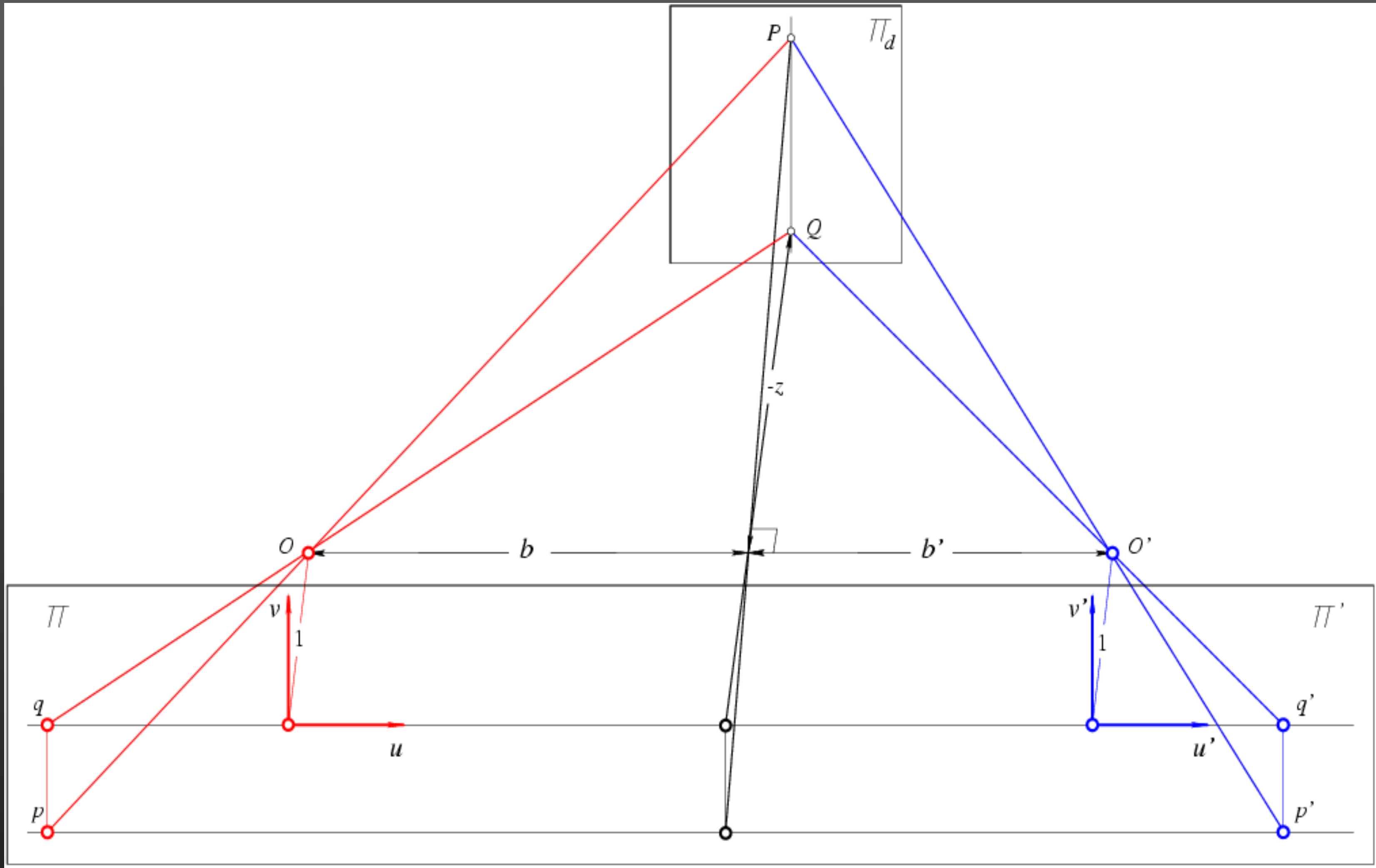


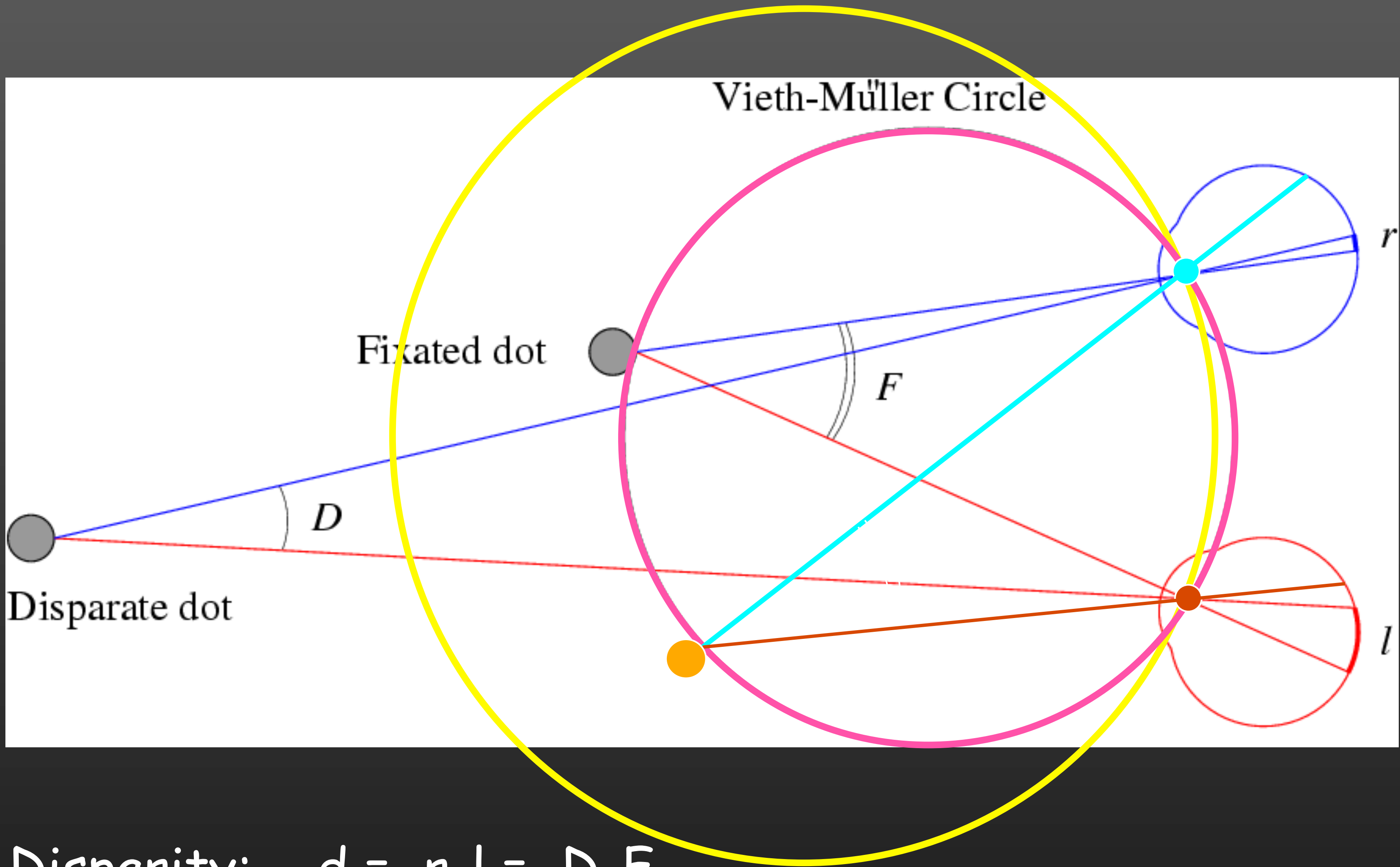
Figure extraite de "US Navy Manual of Basic Optics and Optical Instruments", Bureau of Naval Personnel. Reprinted by Dover Publications, Inc., 1969.

Reconstruction from Rectified Images



Disparity: d=u'-u. → Depth: z = -B/d.

Triangulation for human eyes



Disparity: d = r-l = D-F.

d<0

In 3D, the horopter.

Triangulation for "uncalibrated" human eyes

What if F is not known?

Helmholtz (1909):

- There is evidence showing that the vergence angles cannot be measured precisely.
- People get fooled by bas-relief sculptures.
- There is an analytical explanation for this.
- Relative depth can be judged accurately.

Movies look "flat" on TV

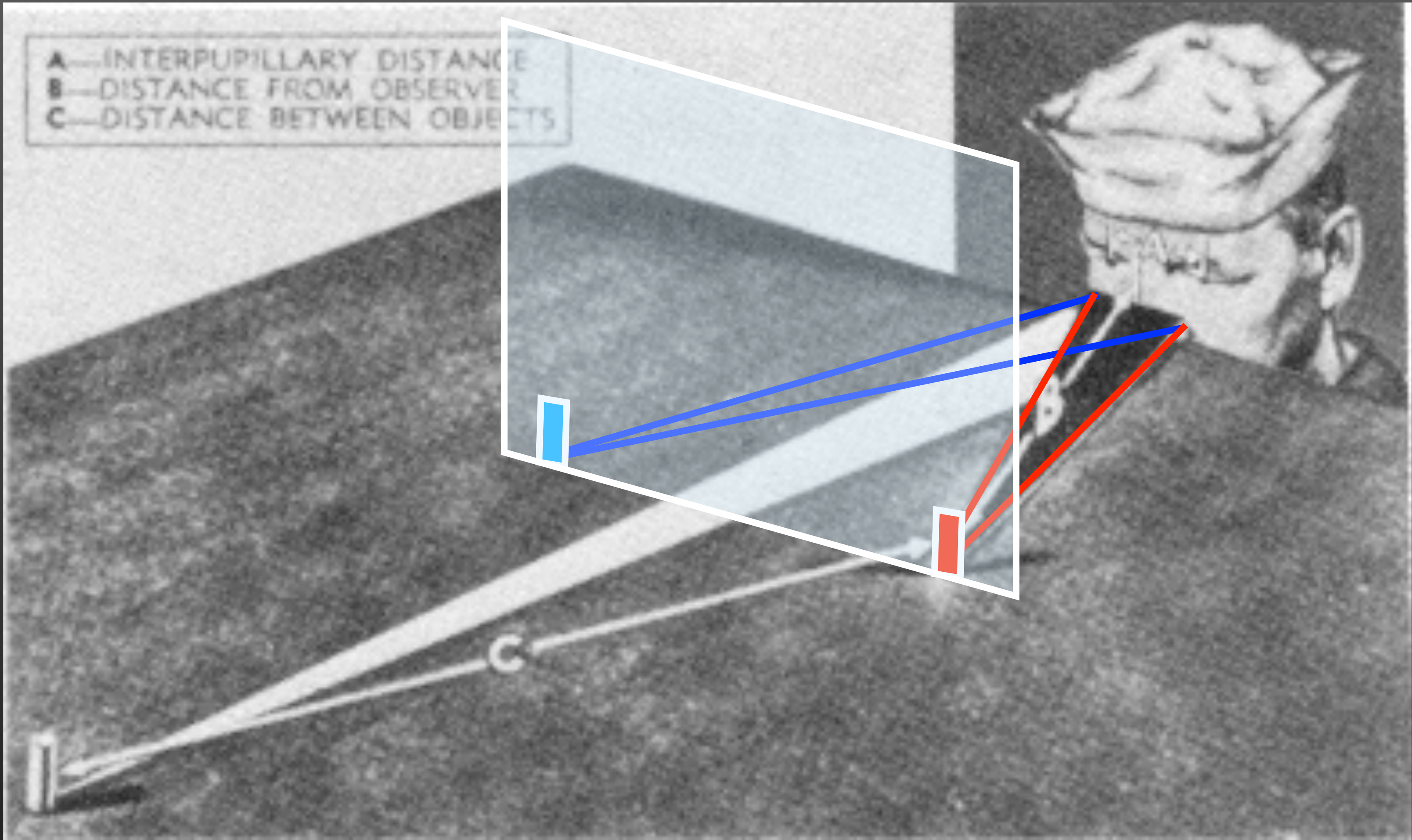


Figure extraite de "US Navy Manual of Basic Optics and Optical Instruments", Bureau of Naval Personnel. Reprinted by Dover Publications, Inc., 1969.

This is why people make 3D movies

Courtesy of Steve Seitz

But do we really need two eyes to "see in 3D" ?

Jan J. Koenderink
Univ. de Delft, NL

But do we really need two eyes to "see in 3D" ?

Figure from "Pictorial Relief", J.J. Koenderink, Phil. Trans. R. Soc. Lond. A (1998) 356, 1071-1086.

Comment "sonder" notre perception de l'orientation d'une surface.

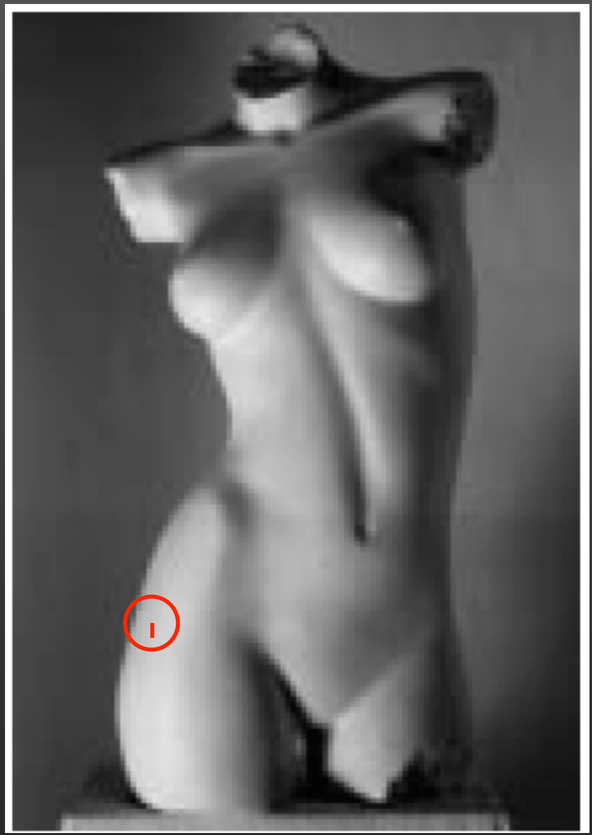


Figure from "Pictorial Relief", J.J. Koenderink, Phil. Trans. R. Soc. Lond. A (1998) 356, 1071-1086.

How to “probe” our perception of surface orientation

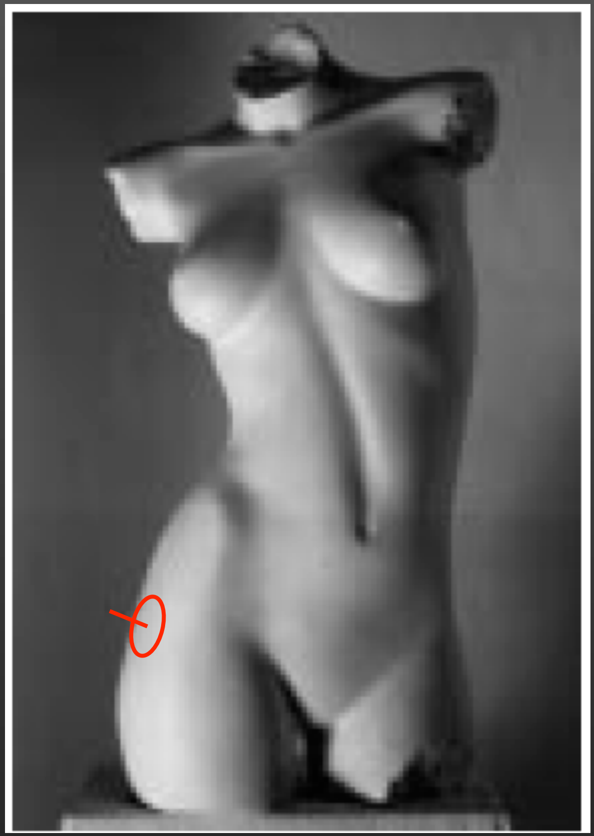


Figure from “Pictorial Relief”, J.J. Koenderink, Phil. Trans. R. Soc. Lond. A (1998) 356, 1071-1086.

How to “probe” our perception of surface orientation

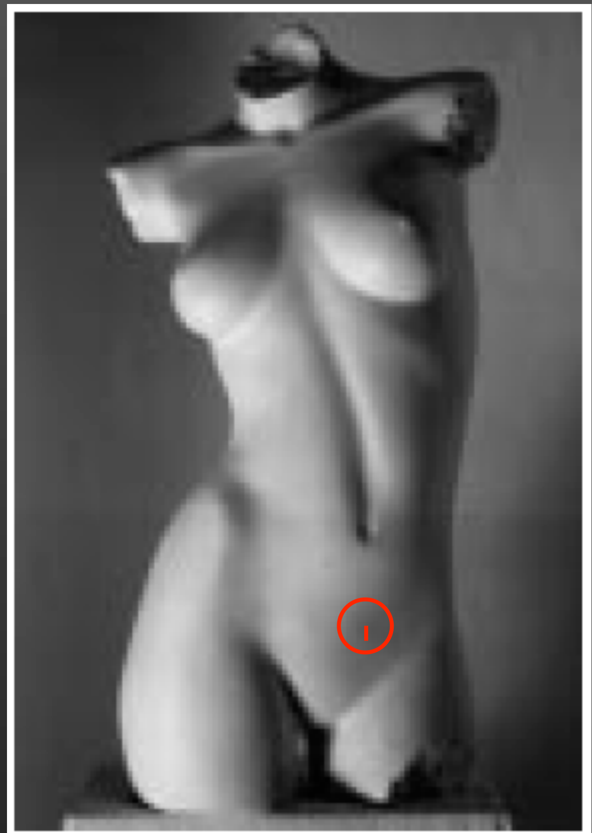


Figure from “Pictorial Relief”, J.J. Koenderink, Phil. Trans. R. Soc. Lond. A (1998) 356, 1071-1086.

How to “probe” our perception of surface orientation

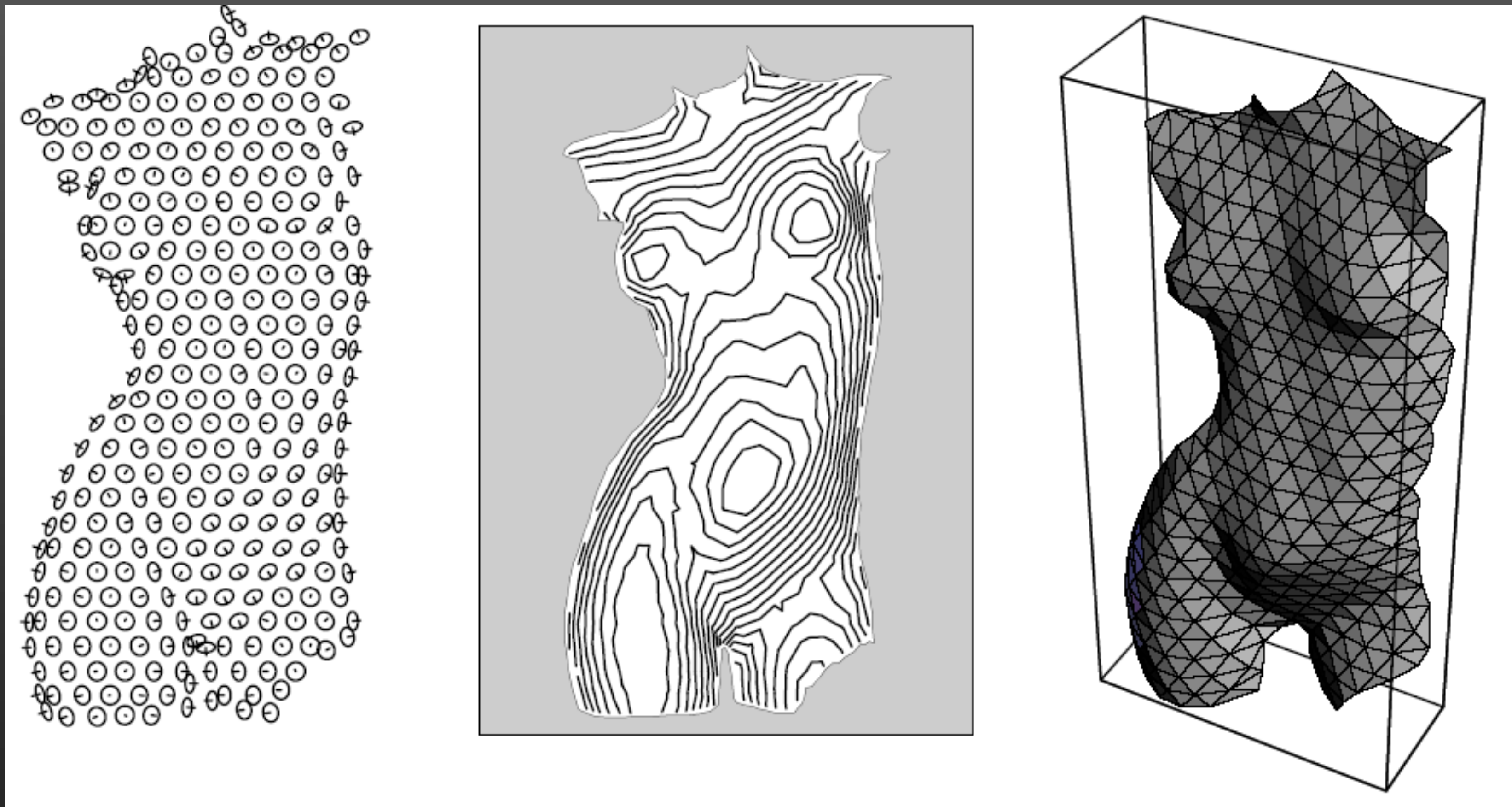


Figure from “Pictorial Relief”, J.J. Koenderink, Phil. Trans. R. Soc. Lond. A (1998) 356, 1071-1086.

Zeiss's synopter (1907)

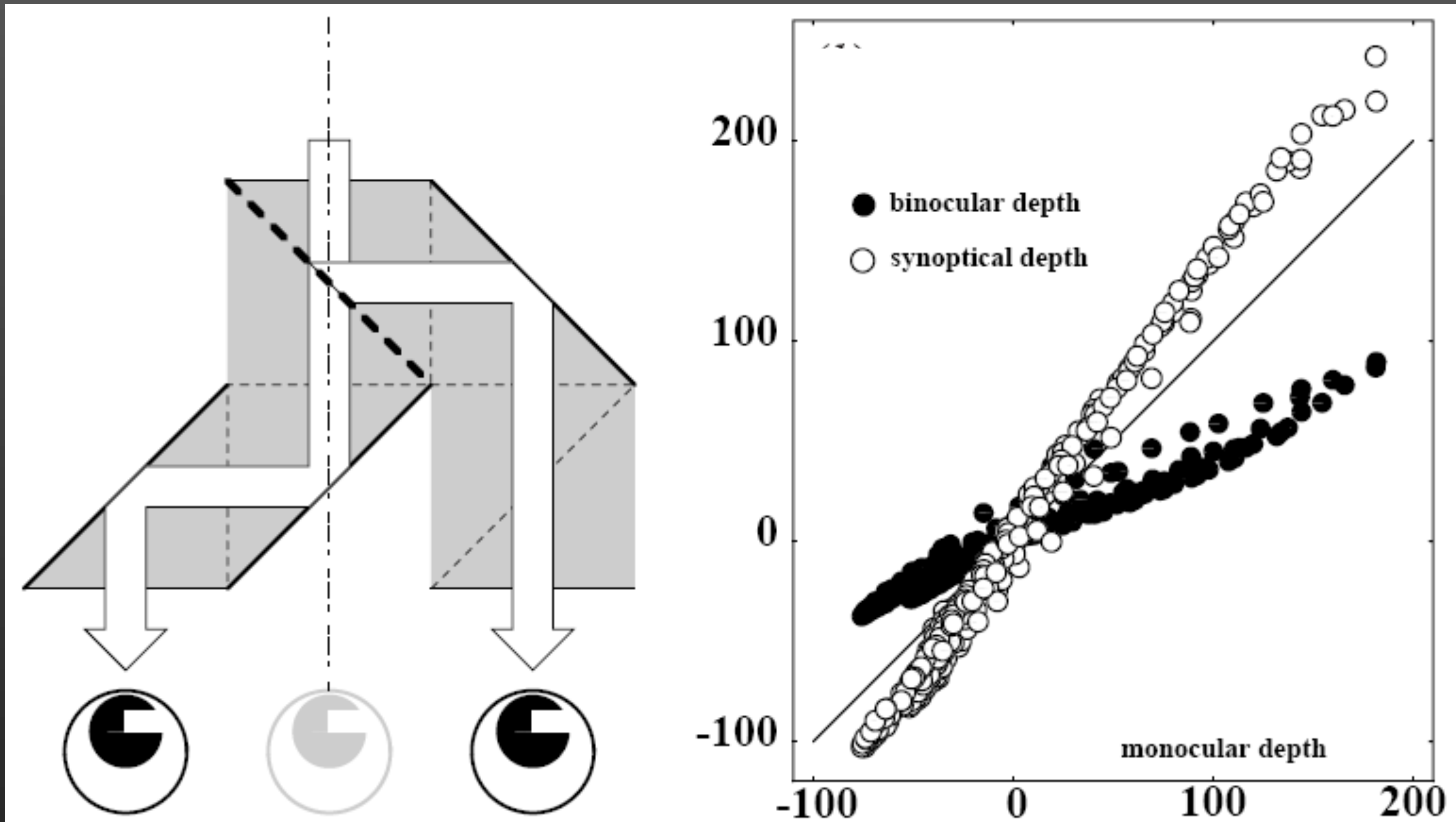


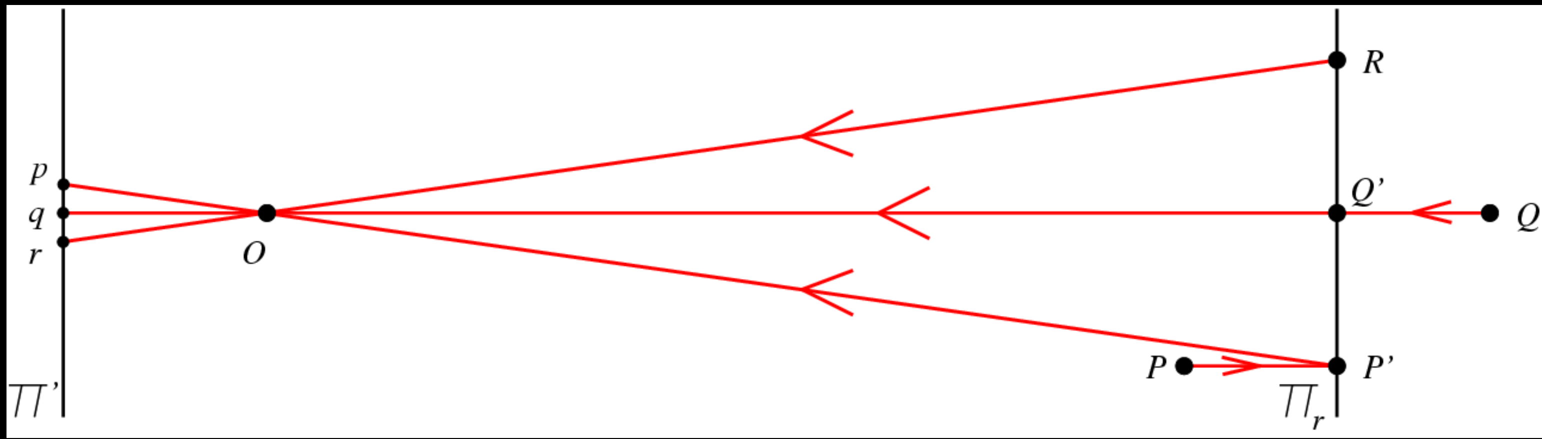
Figure from "Pictorial Relief", J.J. Koenderink, Phil. Trans. R. Soc. Lond. A (1998) 356, 1071-1086.

Affine cameras

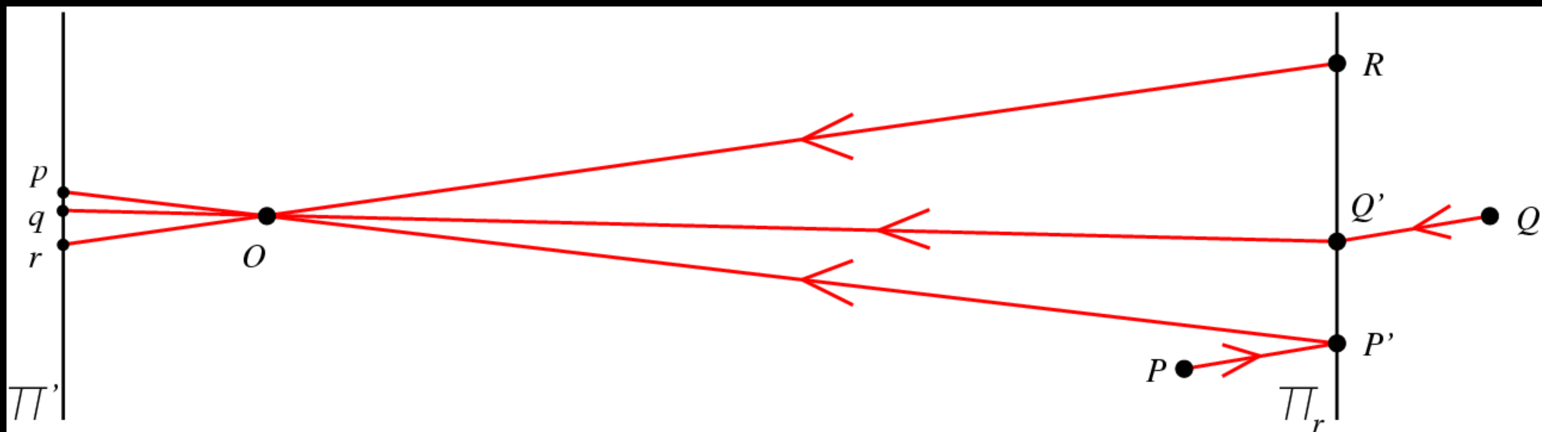
- Affine cameras
- Elements of affine geometry
- Affine structure from motion
- Two-view affine geometry
- Affine SFM revisited

Affine Cameras

Weak-Perspective Projection

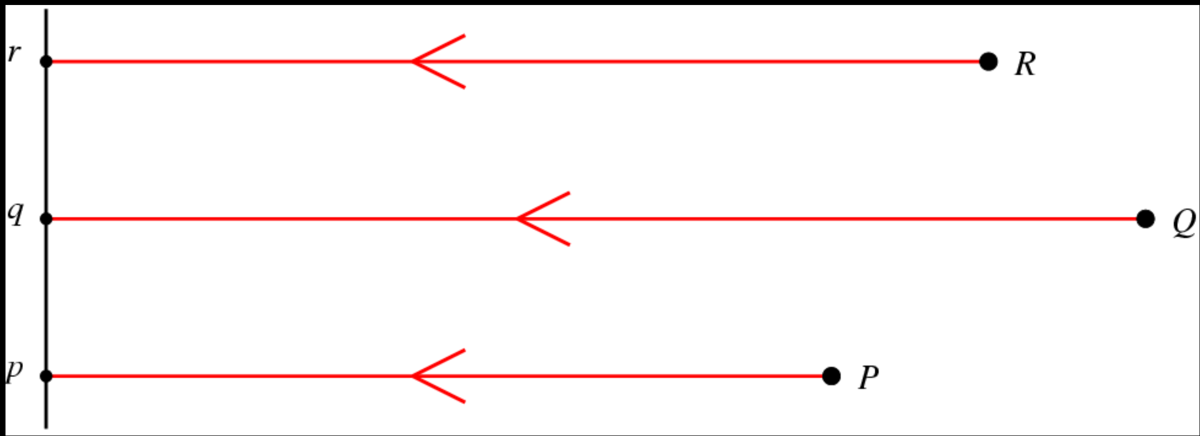


Paraperspective Projection

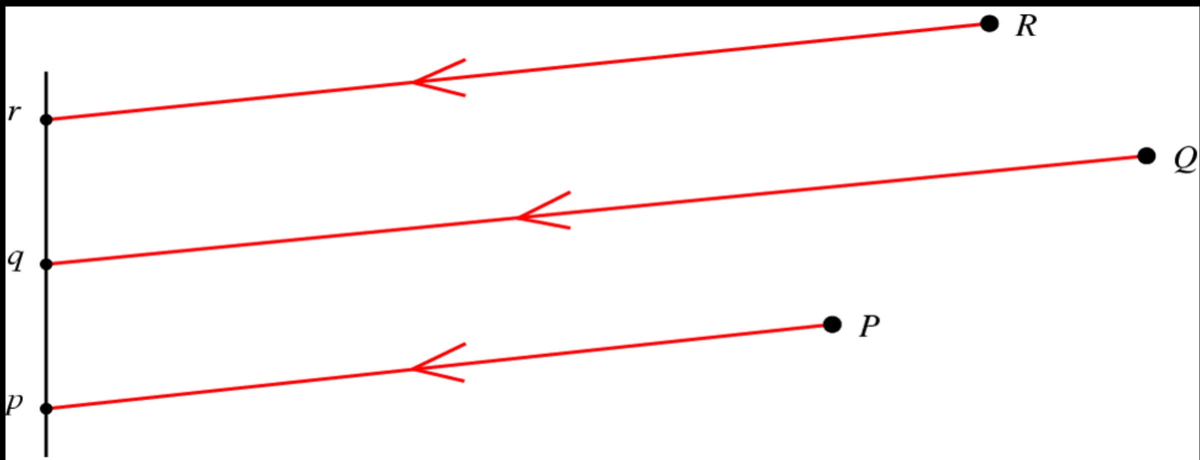


More Affine Cameras

Orthographic Projection



Parallel Projection



Weak-Perspective Projection Model

$$\boldsymbol{p} = \frac{1}{z_r}\mathcal{M}\boldsymbol{P}$$ ($\boldsymbol{p}$ and $\boldsymbol{P}$ are in homogeneous coordinates)

$$p = M\,\boldsymbol{P}$$ ($\boldsymbol{P}$ is in homogeneous coordinates)

$$p = A\,\boldsymbol{P} + \boldsymbol{b}$$ (neither $\boldsymbol{p}$ nor $\boldsymbol{P}$ is in hom. coordinates)

Definition: A 2x4 matrix $M = [A\ b]$, where A is a rank-2 2x3 matrix, is called an affine projection matrix.

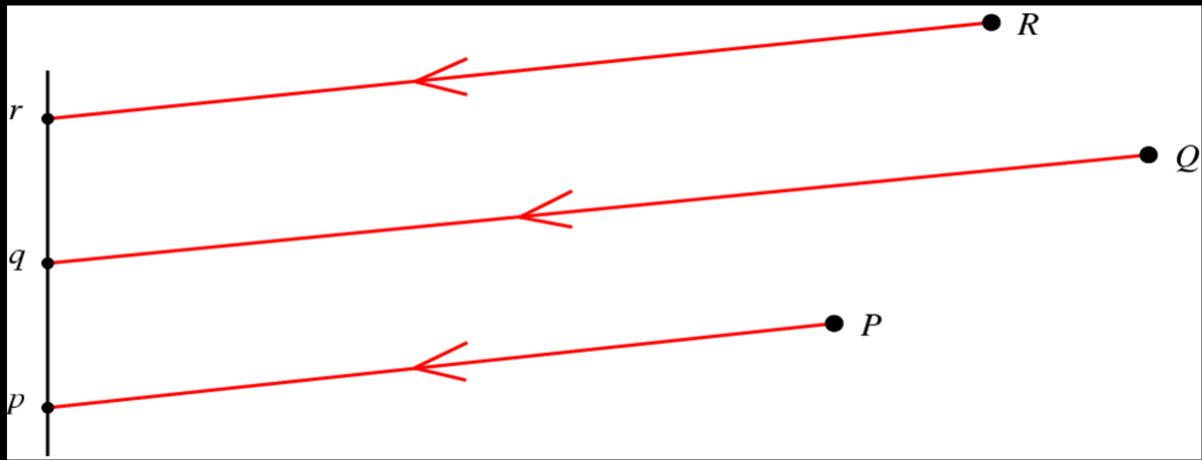
Theorem: All affine projection models can be represented by affine projection matrices.

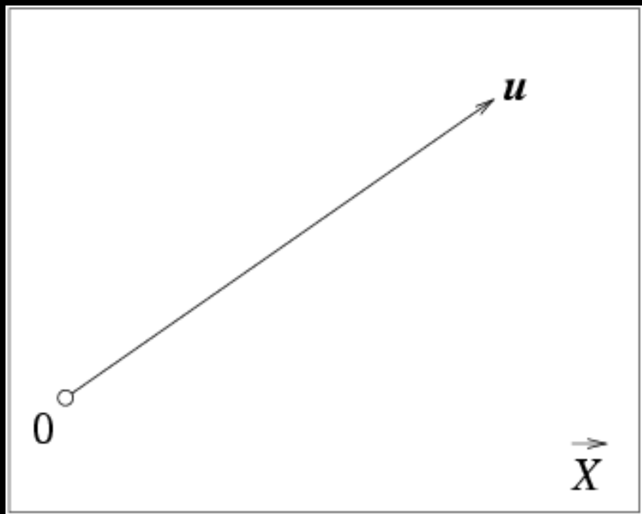
General form of the weak-perspective projection equation:

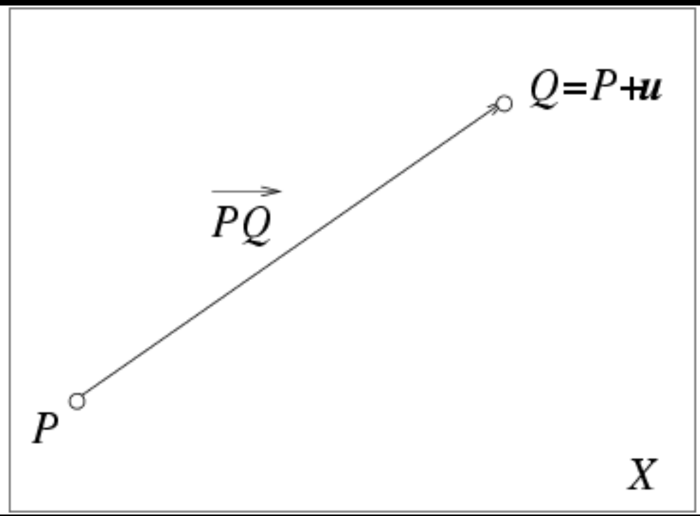
$$\mathbf{M} = \frac{1}{z_r}\begin{bmatrix} k & s \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \mathbf{R}_2 & \mathbf{t}_2 \end{bmatrix} \qquad (1)$$

Theorem: An affine projection matrix can be written uniquely (up to a sign amibguity) as a weak perspective projection matrix as defined by (1).

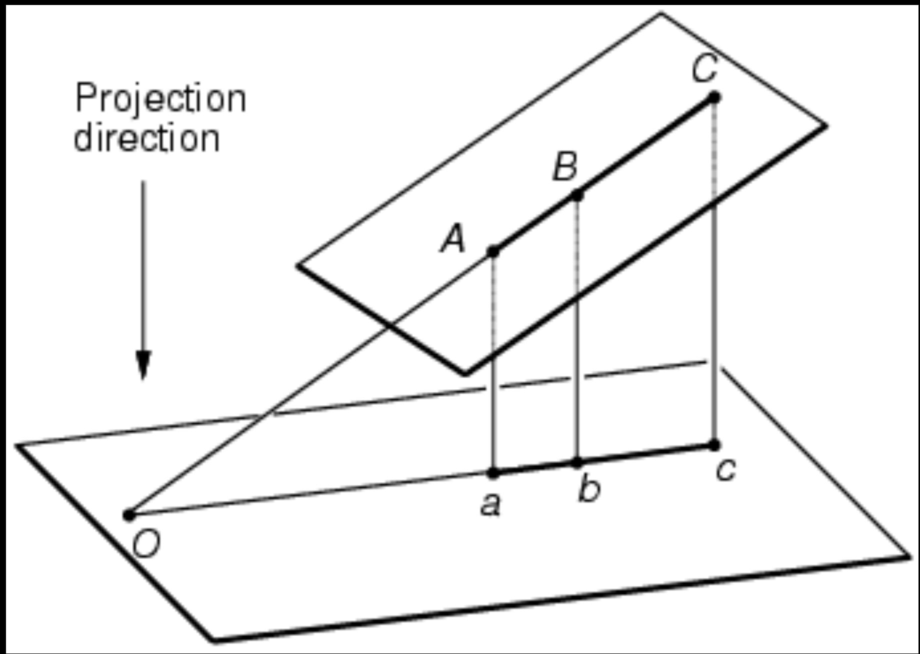
Affine cameras and affine geometry

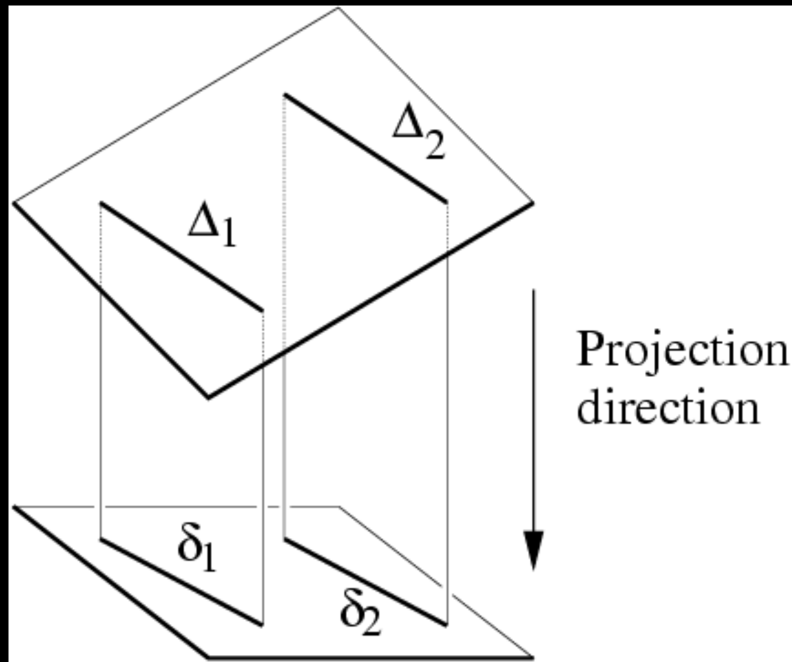




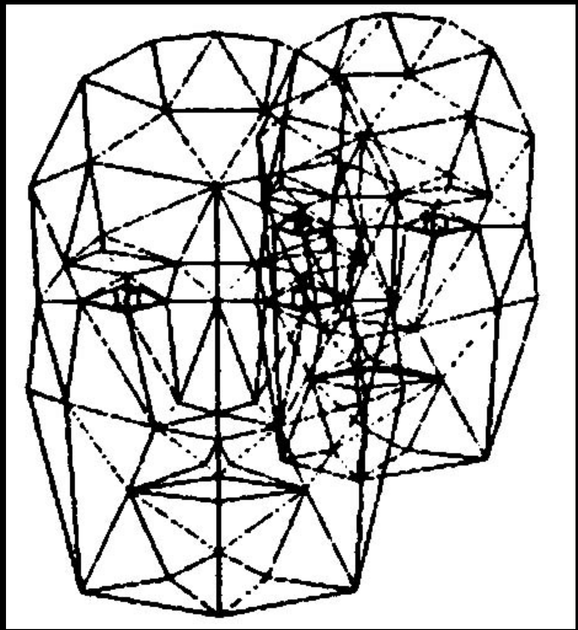


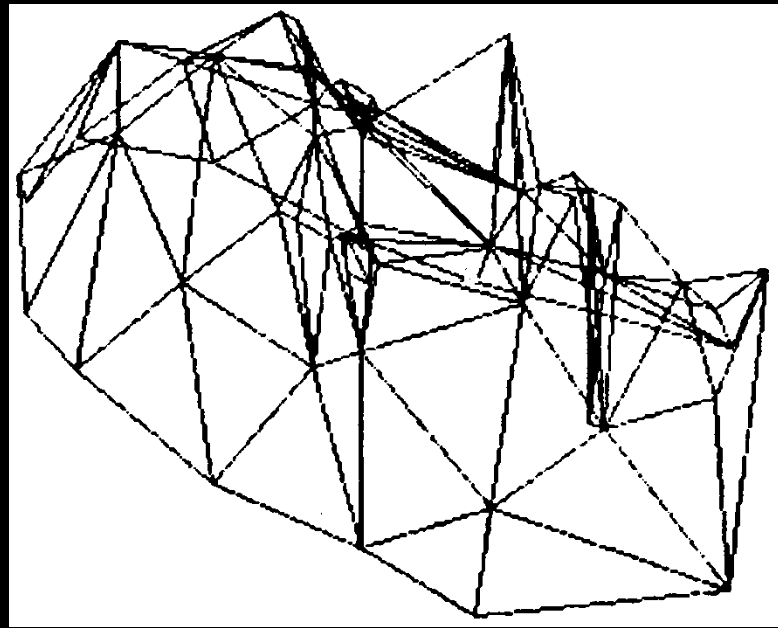
Affine projections induce affine transformations from planes onto their images.





Affine Structure from Motion





Reprinted with permission from "Affine Structure from Motion," by J.J. (Koenderink and A.J.Van Doorn, Journal of the Optical Society of America A, 8:377-385 (1990). © 1990 Optical Society of America.

Given *m* pictures of *n* points, can we recover
- the three-dimensional configuration of these points? (structure)
- the camera configurations? (motion)

The Affine Structure-from-Motion Problem

Given *m* images of *n* fixed points P_j we can write

$$\boldsymbol{p}_{ij} = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} = \mathcal{A}_i \boldsymbol{P}_j + \boldsymbol{b}_i \quad \text{for} \quad i = 1, \ldots, m \quad \text{and} \quad j = 1, \ldots, n.$$

Problem: estimate the *m* 2x4 matrices M_i and the *n* positions P_j from the *mn* correspondences $\boldsymbol{p}_{ij}$.

2*mn* equations in 8*m*+3*n* unknowns

Overconstrained problem, that can be solved using (non-linear) least squares!

The Affine Ambiguity of Affine SFM

When the intrinsic and extrinsic parameters are unknown

If M_i and P_j are solutions,

$$\boldsymbol{p}_{ij} = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} = (\mathcal{M}_i \mathcal{Q}) \left(\mathcal{Q}^{-1} \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} \right) = \mathcal{M}'_i \begin{pmatrix} \boldsymbol{P}'_j \\ 1 \end{pmatrix}$$

So are M_i' and P_j' where

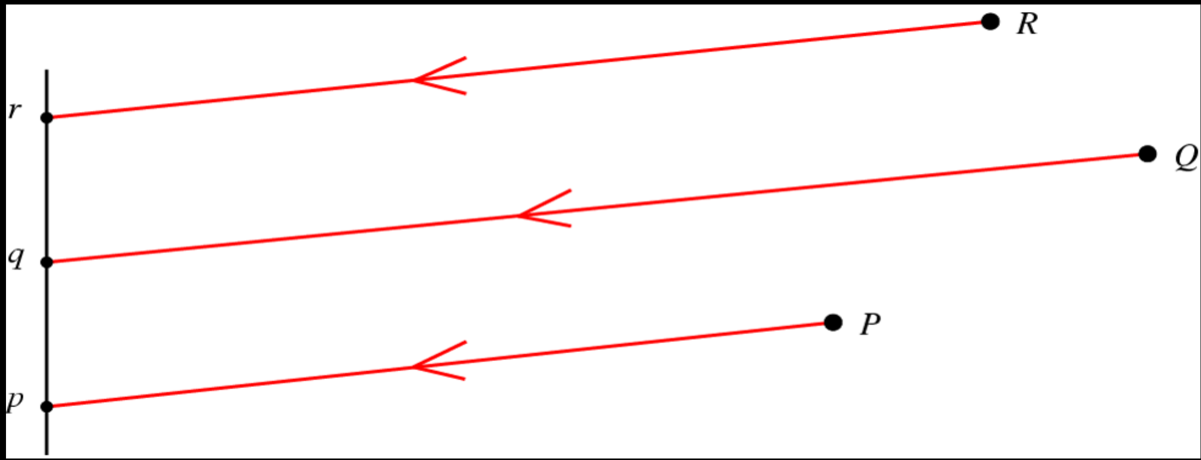
$$\mathcal{M}'_i = \mathcal{M}_i \mathcal{Q} \quad \text{and} \quad \begin{pmatrix} \boldsymbol{P}'_j \\ 1 \end{pmatrix} = \mathcal{Q}^{-1} \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix}$$

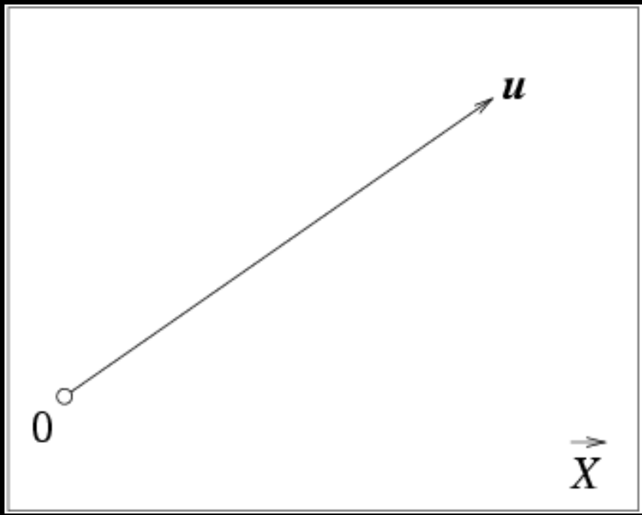
and

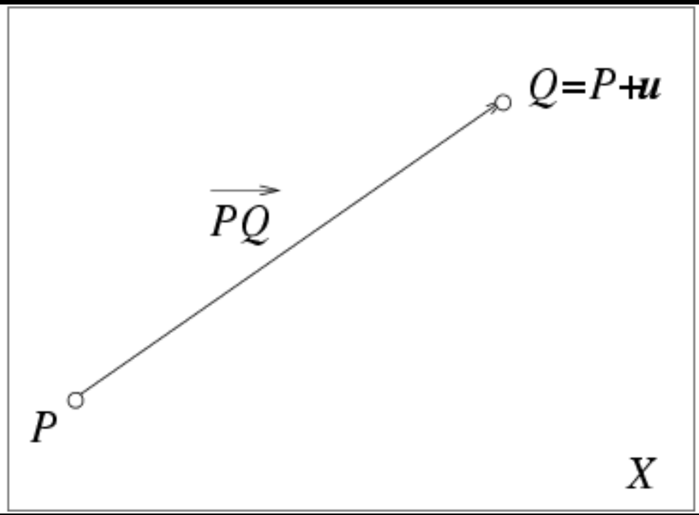
$$\mathcal{Q} = \begin{pmatrix} \mathcal{C} & \boldsymbol{d} \\ \boldsymbol{0}^T & 1 \end{pmatrix} \quad \text{with} \quad \mathcal{Q}^{-1} = \begin{pmatrix} \mathcal{C}^{-1} & -\mathcal{C}^{-1}\boldsymbol{d} \\ \boldsymbol{0}^T & 1 \end{pmatrix}$$

Q is an affine transformation.

Affine cameras and affine geometry







Affine Spaces: (Semi-Formal) Definition

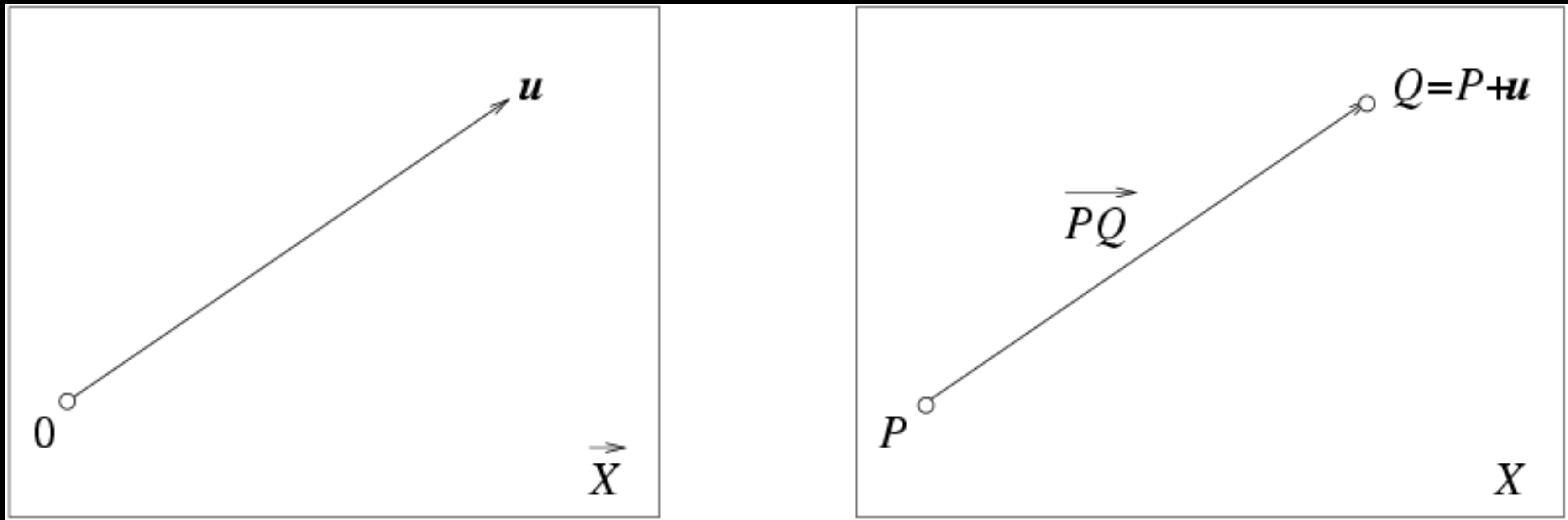
X set of points
$\vec{X}$ underlying vector space
ϕ action of the additive group of $\vec{X}$ on X

ϕ maps elements $\boldsymbol{u}$ of $\vec{X}$ onto bijections $\phi_{\boldsymbol{u}} : X \to X$ such that

$$\begin{array}{lll} \forall P \in X & \phi_{\mathbf{0}}(P) = P & \\ \forall P \in X & \forall \boldsymbol{u}, \boldsymbol{v} \in X & \phi_{\boldsymbol{u}+\boldsymbol{v}}(P) = \phi_{\boldsymbol{u}}(\phi_{\boldsymbol{v}}(P)) \\ \forall P, Q \in X & \exists! \boldsymbol{u} \in \vec{X} & \phi_{\boldsymbol{u}}(P) = Q \end{array}$$

$$P + \boldsymbol{u} \stackrel{\text{def}}{=} \phi_{\boldsymbol{u}}(P)$$
$$\overrightarrow{PQ} \equiv Q - P \stackrel{\text{def}}{=} \boldsymbol{u} \quad \text{such that} \quad \phi_{\boldsymbol{u}}(P) = Q$$

Example: R^2 as an Affine Space



$$P + \boldsymbol{u} \stackrel{\text{def}}{=} \phi_{\boldsymbol{u}}(P)$$

$$\overrightarrow{PQ} \equiv Q - P \stackrel{\text{def}}{=} \boldsymbol{u} \quad \text{such that} \quad \phi_{\boldsymbol{u}}(P) = Q$$

In General

The notation

$$P + \boldsymbol{u} \stackrel{\text{def}}{=} \phi_{\boldsymbol{u}}(P)$$
$$\overrightarrow{PQ} \equiv Q - P \stackrel{\text{def}}{=} \boldsymbol{u} \quad \text{such that} \quad \phi_{\boldsymbol{u}}(P) = Q$$

is justified by the fact that choosing some origin *O* in *X* allows us to identify the point *P* with the vector $\overrightarrow{OP}$.

$$\begin{cases} Q = P + \overrightarrow{PQ}, \\ Q - P = \overrightarrow{PQ}, \end{cases} \iff \begin{cases} \overrightarrow{OQ} = \overrightarrow{OP} + \overrightarrow{PQ}, \\ \overrightarrow{OQ} - \overrightarrow{OP} = \overrightarrow{PQ}. \end{cases}$$

NOTE: P+$\boldsymbol{u}$ and Q-P are defined independently of O!!

Barycentric Combinations

- Can we add points? $\cancel{R=P+Q}$ NO!

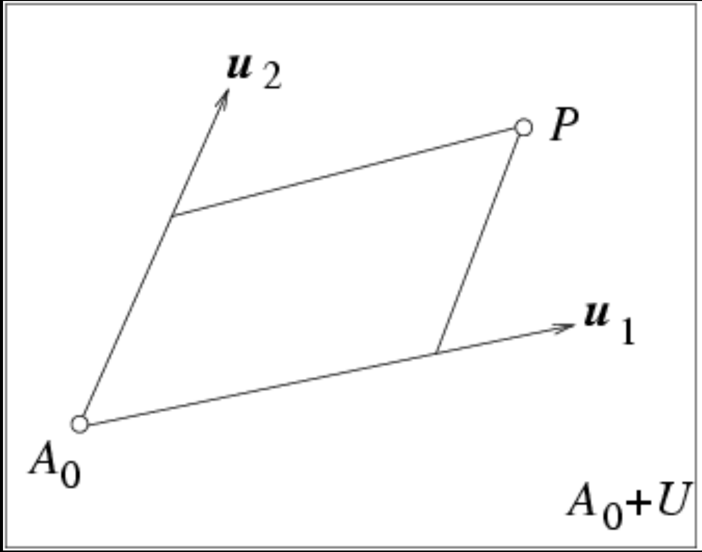
- But, when $\boxed{\alpha_0 + \alpha_1 + \ldots + \alpha_m = 1}$ we can define

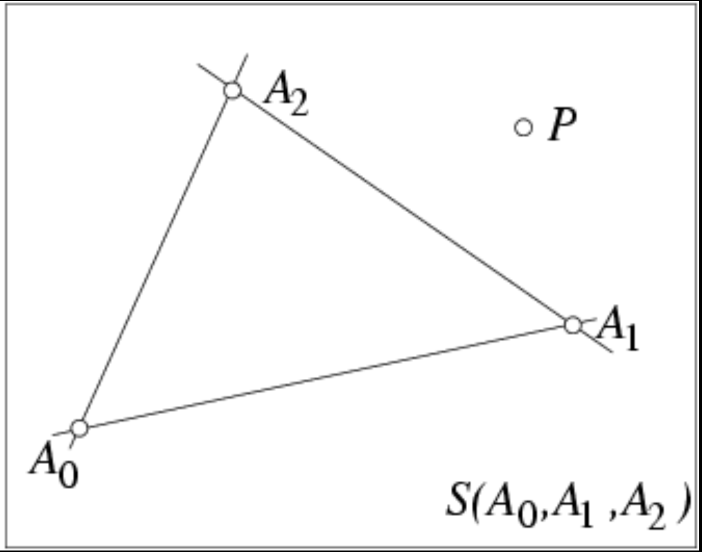
$$\boxed{\sum_{i=0}^{m} \alpha_i A_i \stackrel{\text{def}}{=} A_j + \sum_{i=0, i \neq j}^{m} \alpha_i (A_i - A_j)}$$

- Note:

$$\boxed{\sum_{i=0}^{m} \alpha_i \overrightarrow{OA_i} = \overrightarrow{OA_j} + \sum_{i=0, i \neq j}^{m} \alpha_i (\overrightarrow{OA_i} - \overrightarrow{OA_j})}$$

Affine Subspaces





$$O + U \stackrel{\text{def}}{=} \{O + \boldsymbol{u},\ \boldsymbol{u} \in U\}$$

$$S(A_0, A_1 \ldots, A_m) = \{\sum_{i=0}^{m} \alpha_i A_i,\ \alpha_0 + \ldots + \alpha_m = 1\}$$

Affine Coordinates

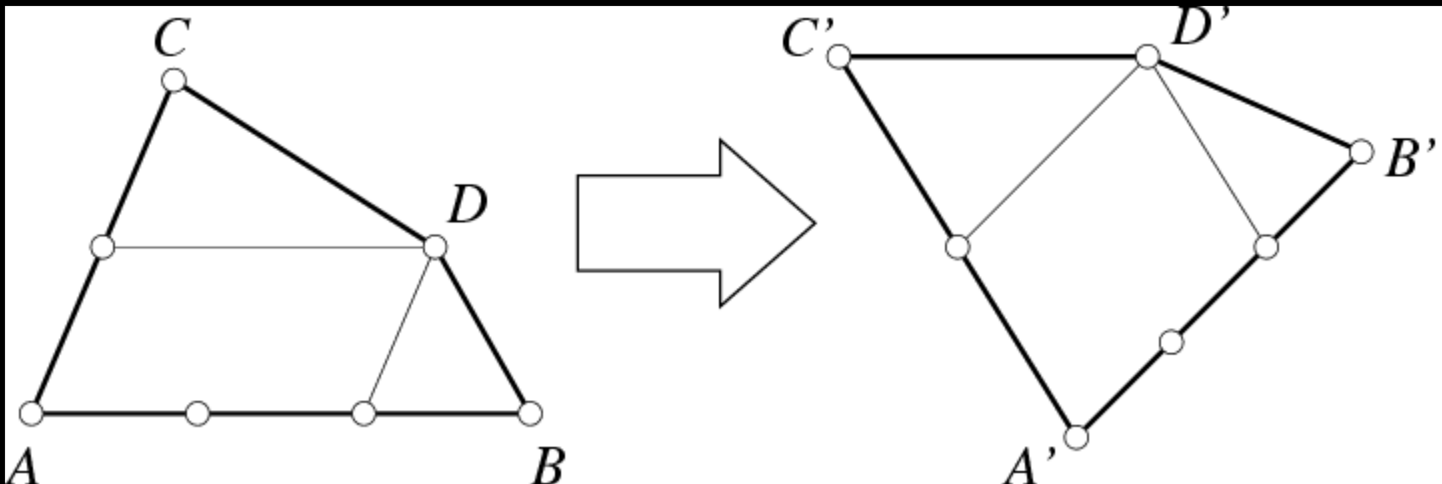
- Coordinate system for U : $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)$
- Coordinate system for $Y = A_0 + U$: $(A_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)$
- Affine coordinates: $P = A_0 + \alpha_1 \boldsymbol{u}_1 + \ldots + \alpha_m \boldsymbol{u}_m$
- Coordinate system for Y : $(A_0, A_1, \ldots, A_m)$
- Barycentric coordinates: $P = \alpha_0 A_0 + \alpha_1 A_1 + \ldots + \alpha_m A_m$

$$
\begin{aligned}
P &= \boxed{\alpha_0 A_0 + \alpha_1 A_1 + \ldots + \alpha_m A_m} \\
&= \boxed{A_0 + \alpha_1 (A_1 - A_0) + \ldots + \alpha_m (A_m - A_0)}
\end{aligned}
$$

Affine Transformations

Bijections from X to Y that:

- map m-dimensional subspaces of X onto m-dimensional subspaces of Y;
- map parallel subspaces onto parallel subspaces; and
- preserve affine (or barycentric) coordinates.

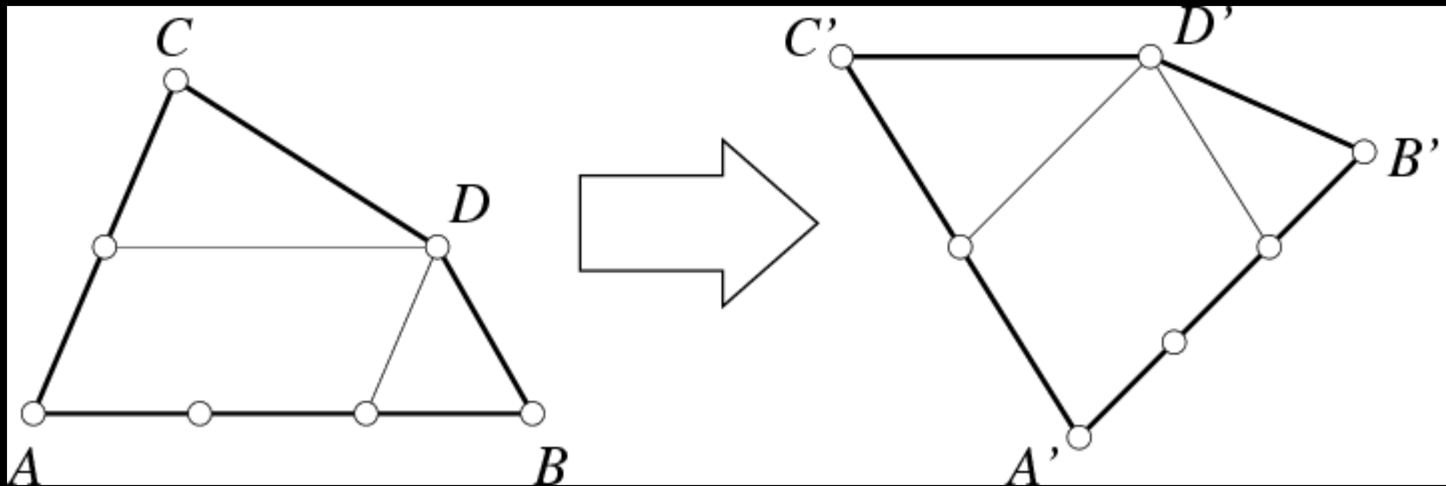


In E^3 they are combinations of rigid transformations, non-uniform scalings and shears.

Affine Transformations

Bijections from X to Y that:

- map lines of X onto lines of Y; and
- preserve the ratios of signed lengths of line segments.



In E^3 they are combinations of rigid transformations, non-uniform scalings and shears.

Affine Transformations II

• Given two affine spaces X and Y of dimension m, and two coordinate frames *(A)* and *(B)* for these spaces, there exists a unique affine transformation mapping *(A)* onto *(B)*.

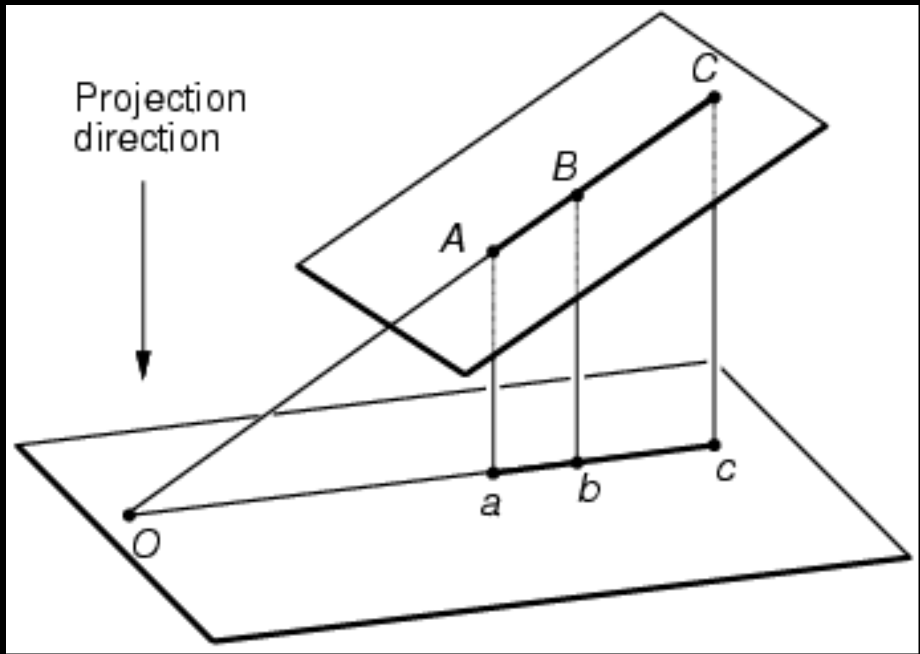
• Given an affine transformation from X to Y, one can always write:

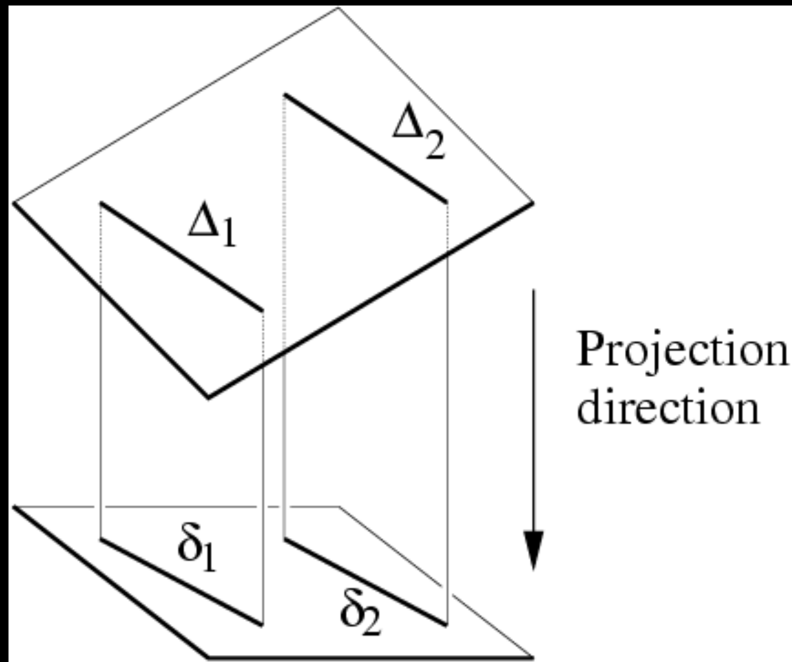
$$\boxed{\psi(P) = \psi(O) + \vec{\psi}(P - O)}$$

• When coordinate frames have been chosen for X and Y, this translates into:

$$\boxed{\psi(\boldsymbol{P}) = \boldsymbol{d} + \mathcal{C}\boldsymbol{P} = \mathcal{C}\boldsymbol{P} + \boldsymbol{d}}$$

Affine projections induce affine transformations from planes onto their images.





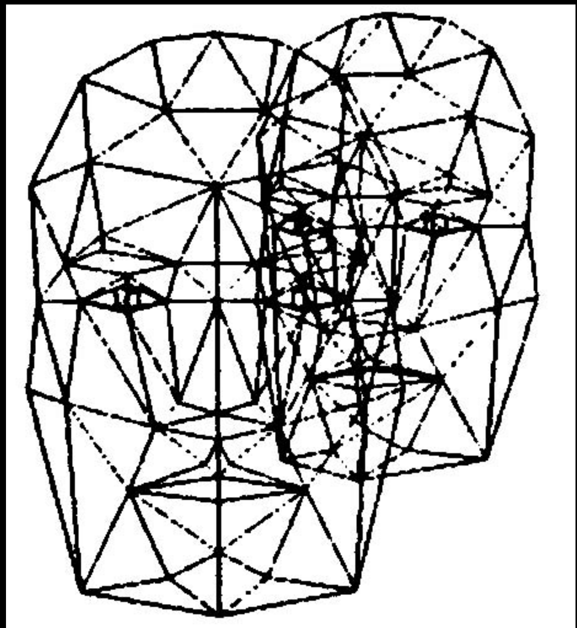
Affine Shape

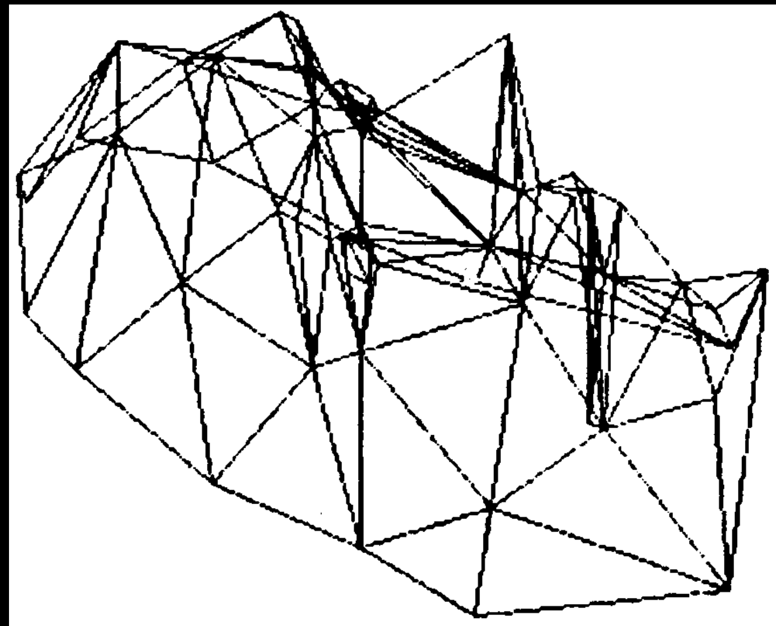
Two point sets S and S' in some affine space X are affinely equivalent when there exists an affine transformation $\psi: X \rightarrow X$ such that $X' = \psi(X)$.

Affine structure from motion = affine shape recovery.

= recovery of the corresponding motion equivalence classes.

Affine Structure from Motion





Reprinted with permission from “Affine Structure from Motion,” by J.J. (Koenderink and A.J.Van Doorn, Journal of the Optical Society of America A, 8:377-385 (1990). © 1990 Optical Society of America.

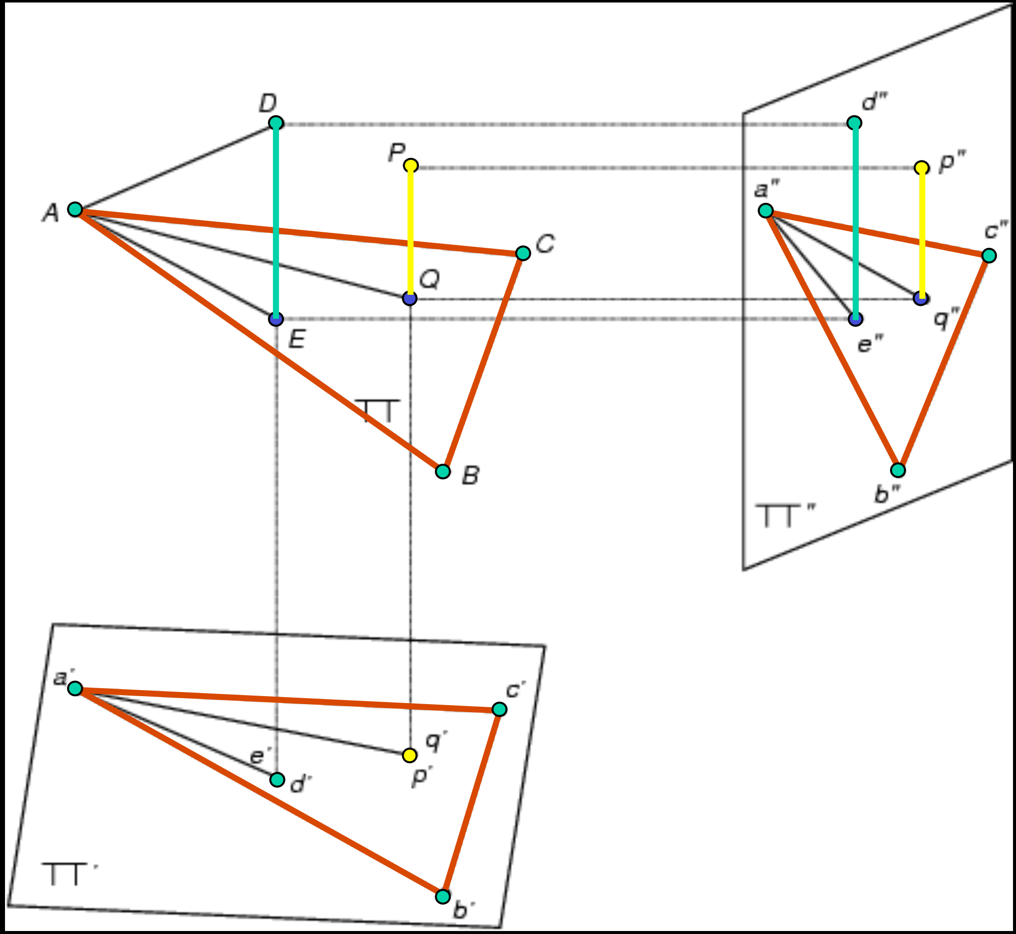
Given *m* pictures of *n* points, can we recover

- the three-dimensional configuration of these points? (structure)
- the camera configurations? (motion)

Geometric affine scene reconstruction from two images (Koenderink and Van Doorn, 1991).

$$\begin{cases} \alpha_{q'} = \alpha_Q \\ \beta_{q'} = \beta_Q \end{cases}$$

$$\lambda = \frac{\overline{q''p''}}{\overline{e''d''}} = \frac{\overline{QP}}{\overline{ED}}$$



$$\begin{aligned}
\overrightarrow{AP} &= \overrightarrow{AQ} + \overrightarrow{QP} \\
&= \alpha_{p'}\overrightarrow{AB} + \beta_{p'}\overrightarrow{AC} + \lambda\overrightarrow{ED} \\
&= (\alpha_{p'} - \lambda\alpha_{d'})\overrightarrow{AB} + (\beta_{p'} - \lambda\beta_{d'})\overrightarrow{AC} + \lambda\overrightarrow{AD}
\end{aligned}$$