6.801/866

Affine Structure from Motion

T. Darrell

[Read F&P Ch. 12.0, 12.2, 12.3, 12.4]

"Affine geometry is, roughly speaking, what is left after all ability to measure lengths, areas, angles, etc. has been removed from Euclidean geometry. The concept of parallelism remains, however, as well as the ability to measure the ratio of distances between collinear points."

[Snapper and Troyer, 1989]

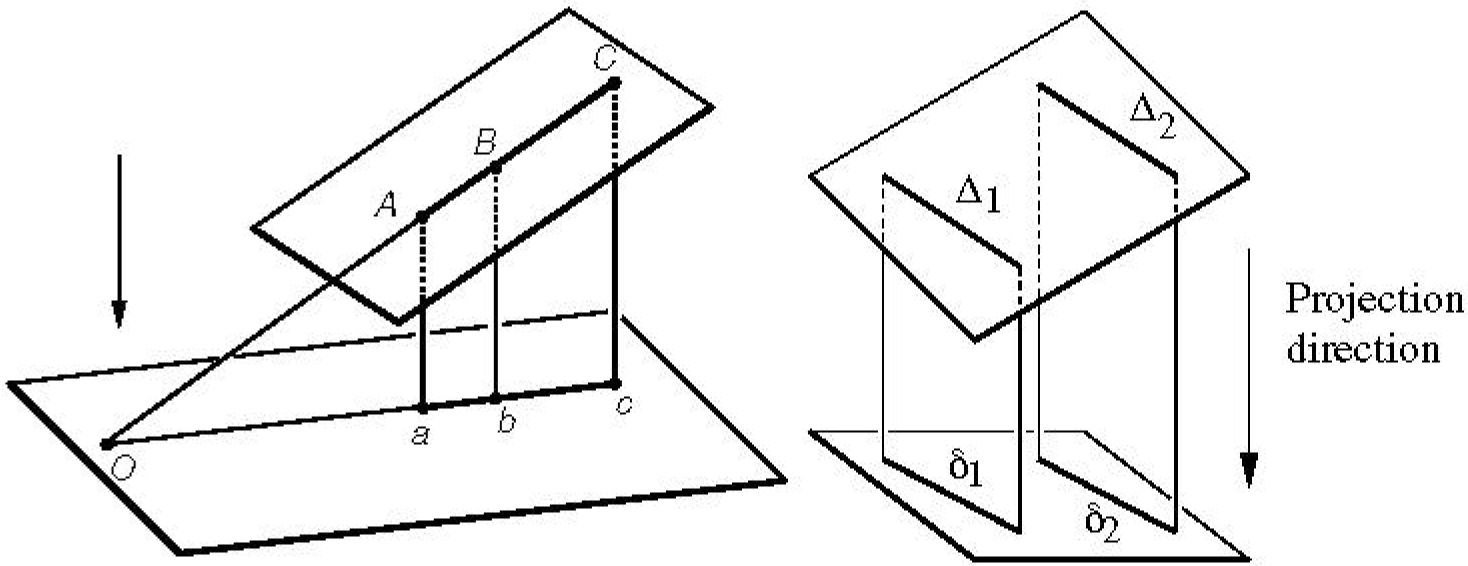


FIGURE 13.2: Parallel projection preserves: (left) the ratio of signed distances between collinear points and (right) the parallelism of lines.

We will ignore the correspondence problem in the rest of this chapter, assuming that the projections of n points have been matched across m pictures.

Tracked feature j in camera i: $\boldsymbol{p}_{ij}$

Affine camera

$$\boldsymbol{p}_{ij} = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} = \mathcal{A}_i \boldsymbol{P}_j + \boldsymbol{b}_i$$

$$\boldsymbol{p}_{ij} = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} = \mathcal{A}_i \boldsymbol{P}_j + \boldsymbol{b}_i$$

We define affine structure from motion as the problem of estimating the m 2×4 matrices

$$\mathcal{M}_i = \begin{pmatrix} \mathcal{A}_i & \boldsymbol{b}_i \end{pmatrix}$$

and the n positions P_j of the points P_j in some fixed coordinate system from the mn image correspondences p_{ij}

$$\boldsymbol{p}_{ij} = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} = \mathcal{A}_i \boldsymbol{P}_j + \boldsymbol{b}_i$$

This equation provides 2mn constraints on the 8m+3n unknown coefficients defining the matrices M_i and the point positions P_j.

Fortunately, 2mn is greater than 8m+3n for large enough values of m and n…

But, the solution is ambiguous…

If M_i and P_j are solutions to

$$\boldsymbol{p}_{ij} = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} = \mathcal{A}_i \boldsymbol{P}_j + \boldsymbol{b}_i$$

then 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 Q is an arbitrary affine transformation matrix, that is,

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

where C is a non-singular 3 × 3 matrix and d is a vector in R3. In other words, ***any solution of the affine structure-from-motion problem can only defined up to an affine transformation ambiguity.***

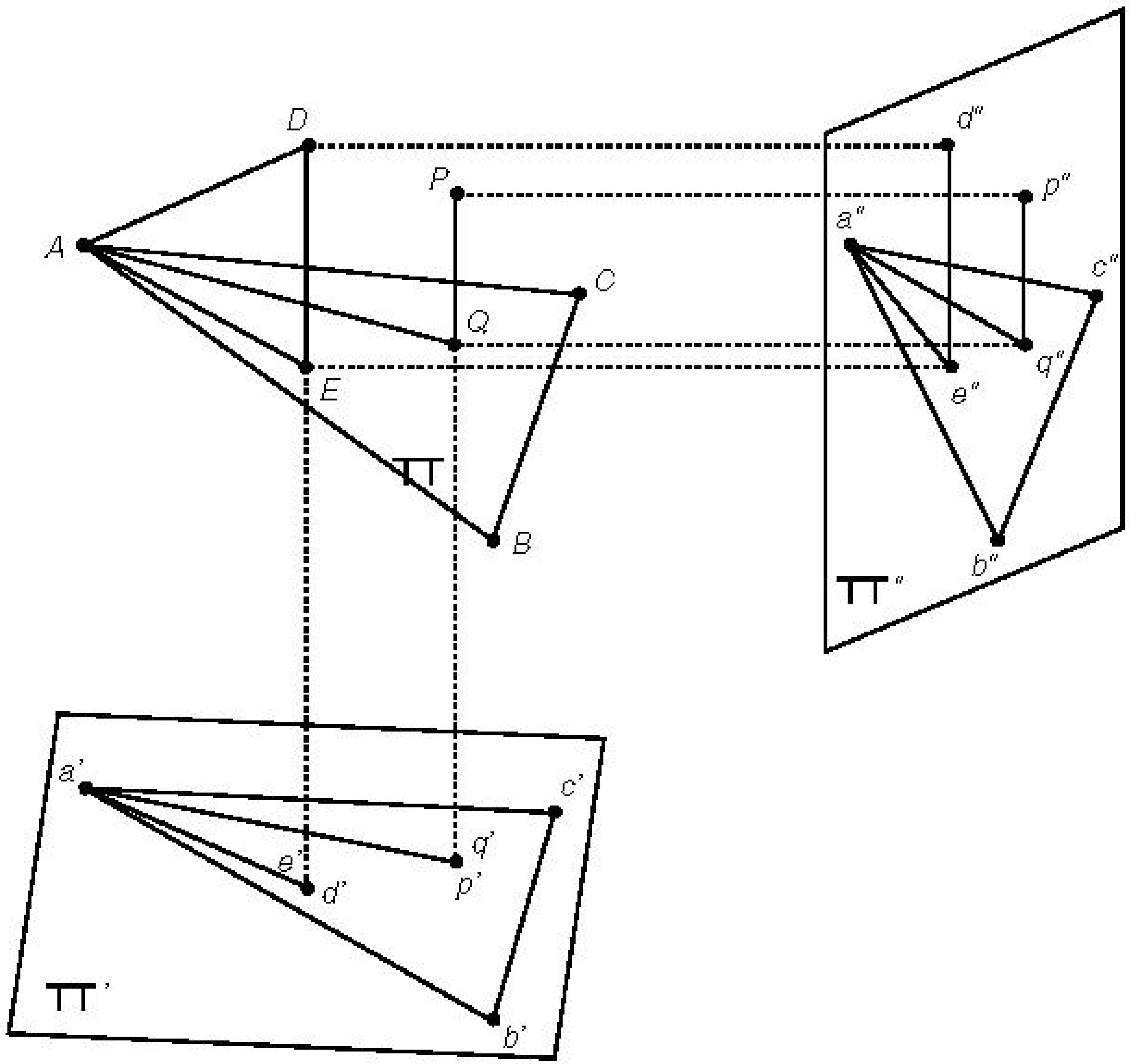
Affine Structure from Motion

- Two views
  - Geometric Approach: infer affine shape (then recover affine projection matricies if needed)
  - Algebraic Approach: estimate projection matricies (then determine position of scene points)
- Sequence
  - Factorization Approach

Affine Structure from Motion Theorem

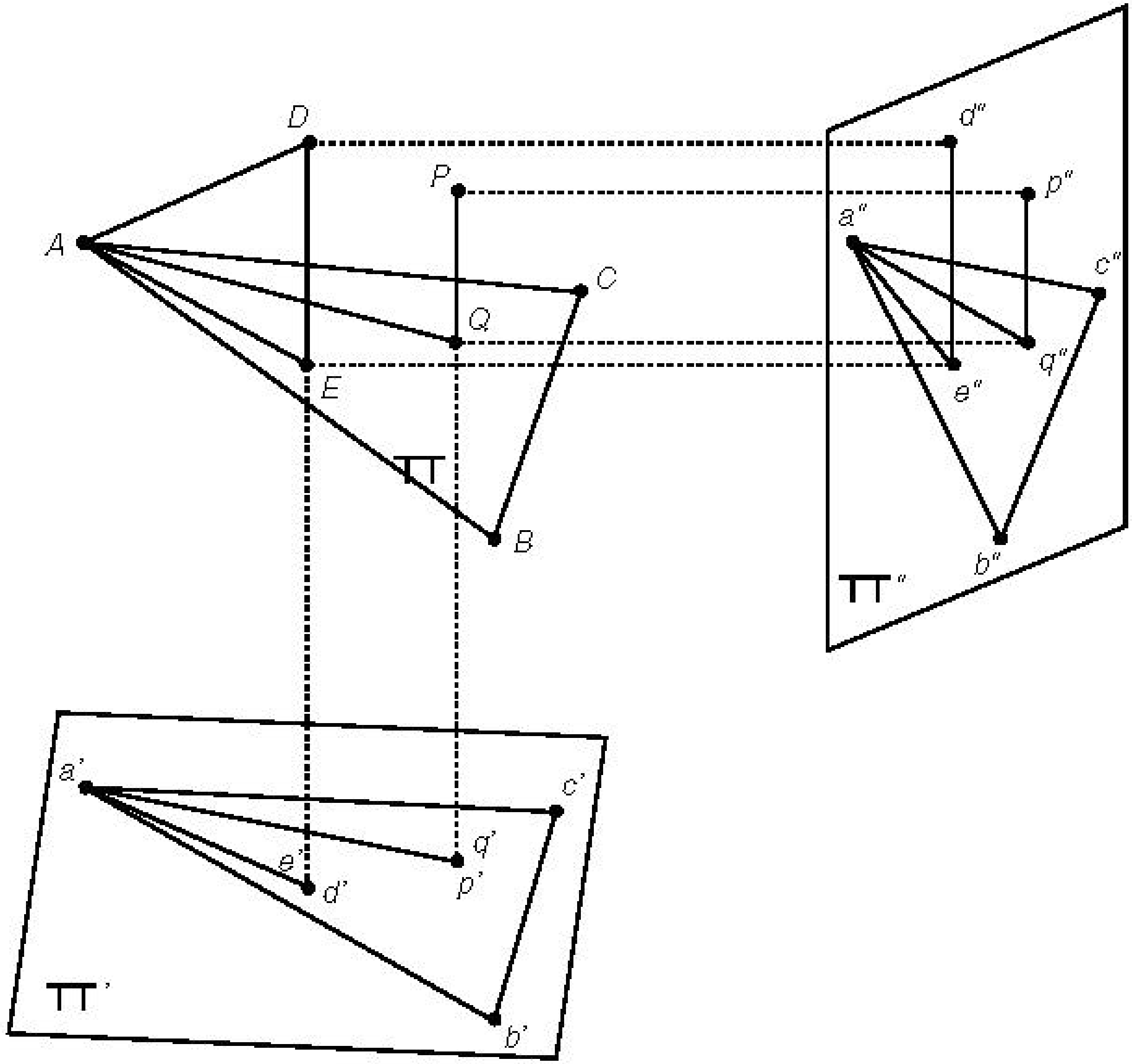
Two affine views of four non co-planar points are sufficient to compute the affine coordinate of any other point P.

[Koenderink and Van Doorn, 1990]



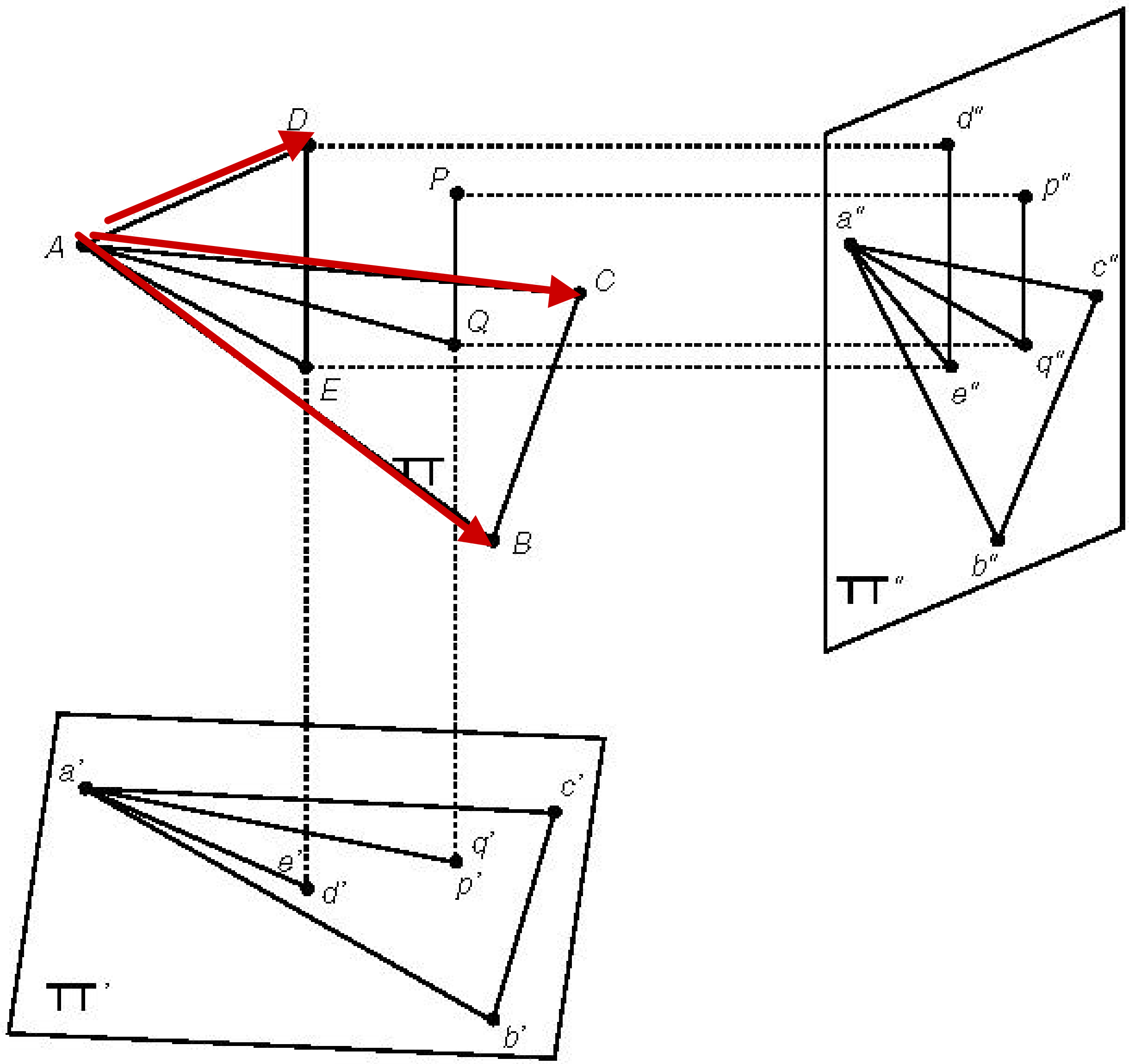
D
P
A
C
Q
E
B
TT
d"
p"
a"
c"
q"
e"
b"
TT"
a'
c'
q'
e'
d'
p'
b'
TT'

Two affine views of four points are sufficient to compute the affine coordinate of any other point P…



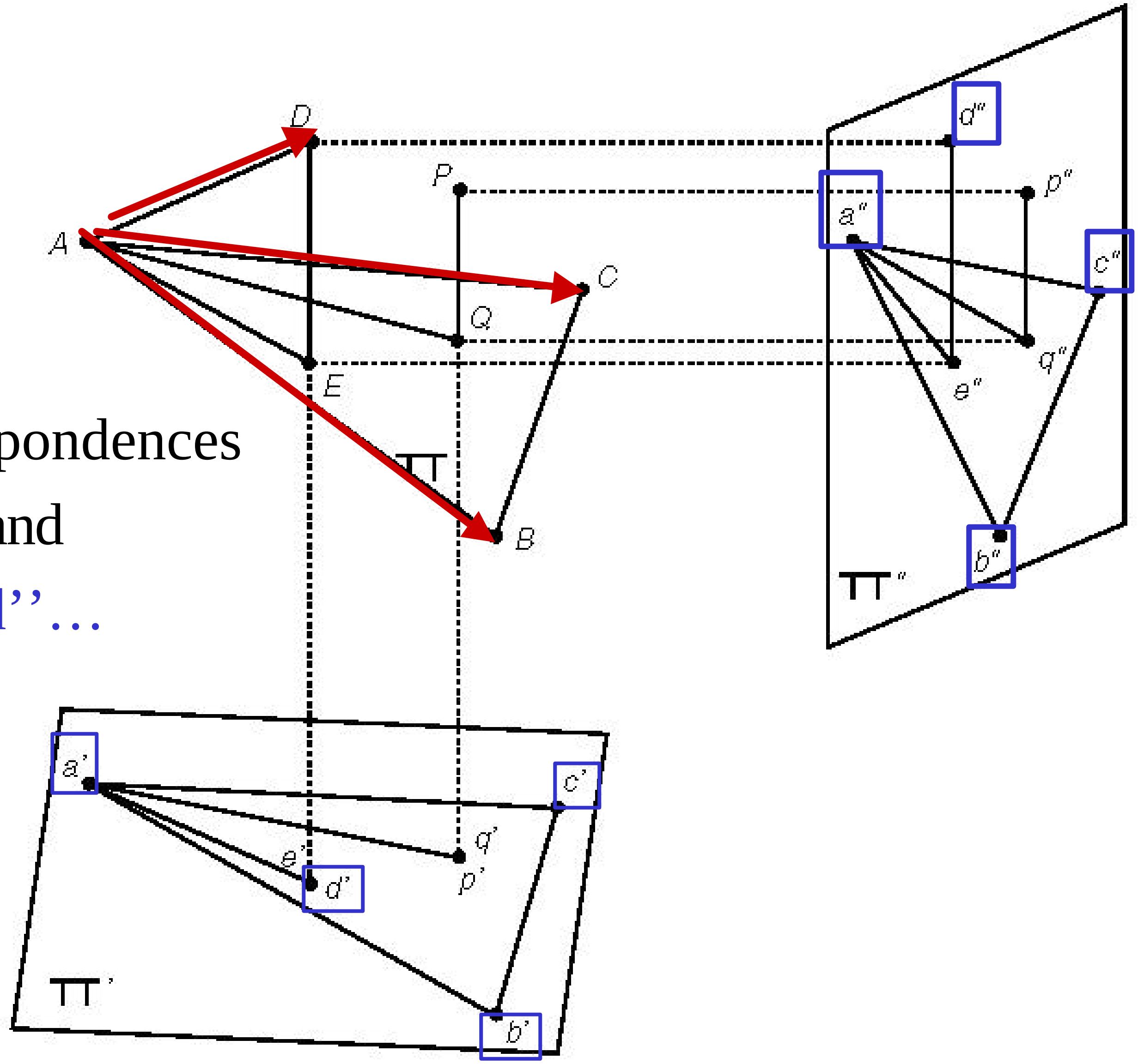
Given Affine Basis (A,B,C,D)

(e.g., A=(0,0,0), B=(0,0,1), C=(0,1,0), D=(1,0,0))



Given Affine Basis (A,B,C,D)

(e.g., A=(0,0,0), B=(0,0,1), C=(0,1,0), D=(1,0,0))



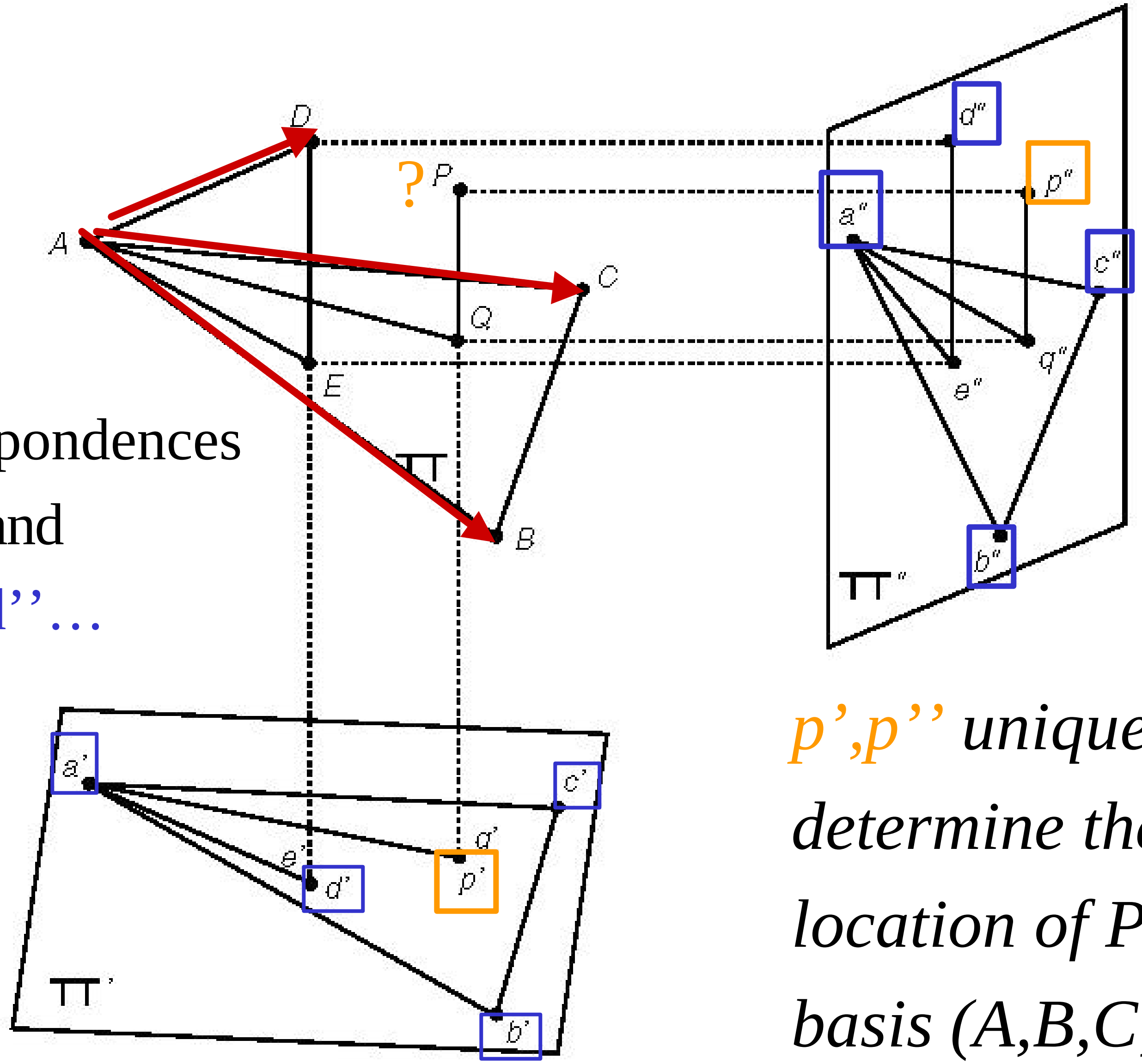
And correspondences

a',b',c',d' and

a'',b'',c'',d''…

Given Affine Basis (A,B,C,D)

(e.g., A=(0,0,0), B=(0,0,1), C=(0,1,0), D=(1,0,0))



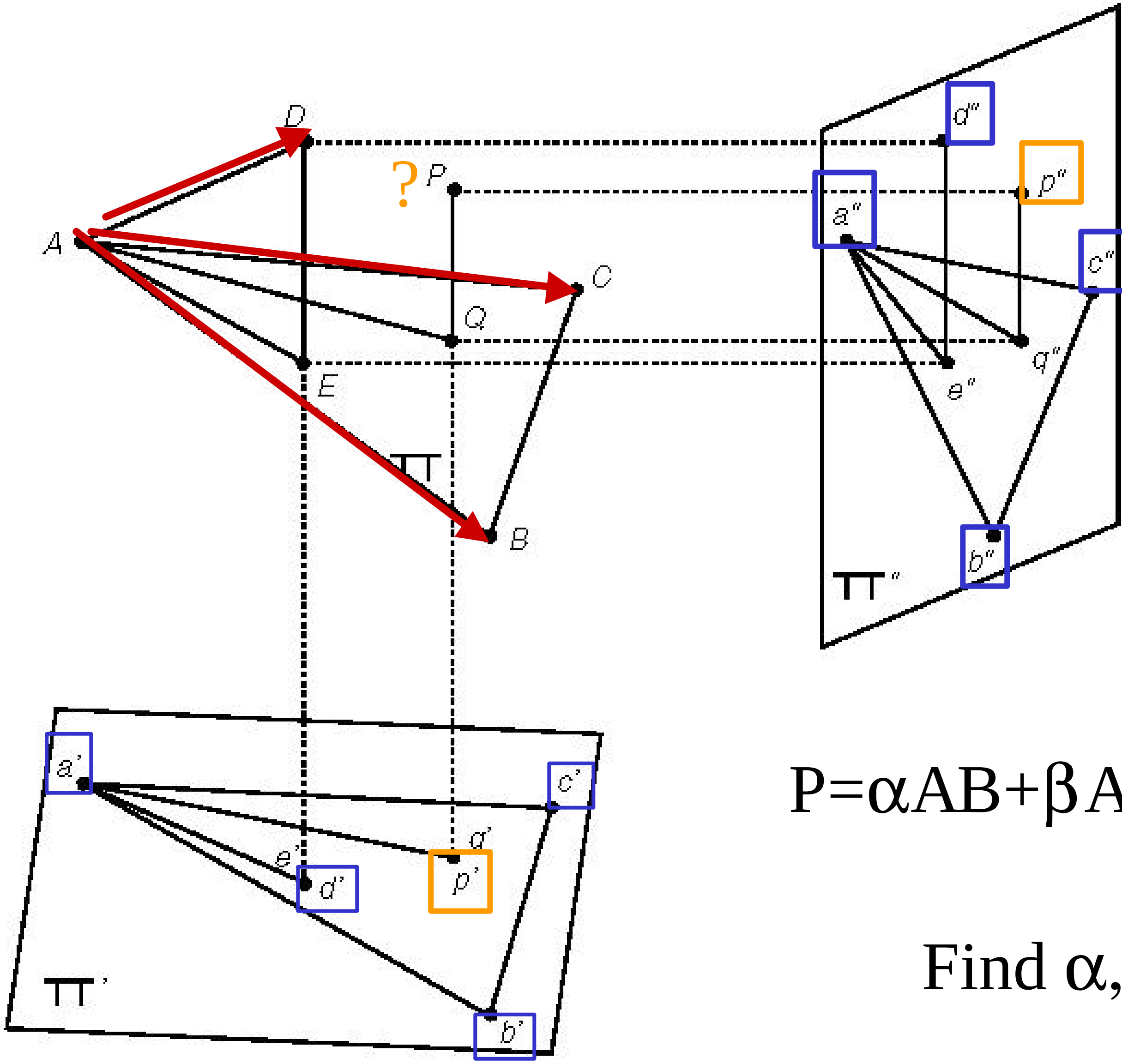
And correspondences

a’,b’,c’,d’ and

a’’,b’’,c’’,d’’…

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)!

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…



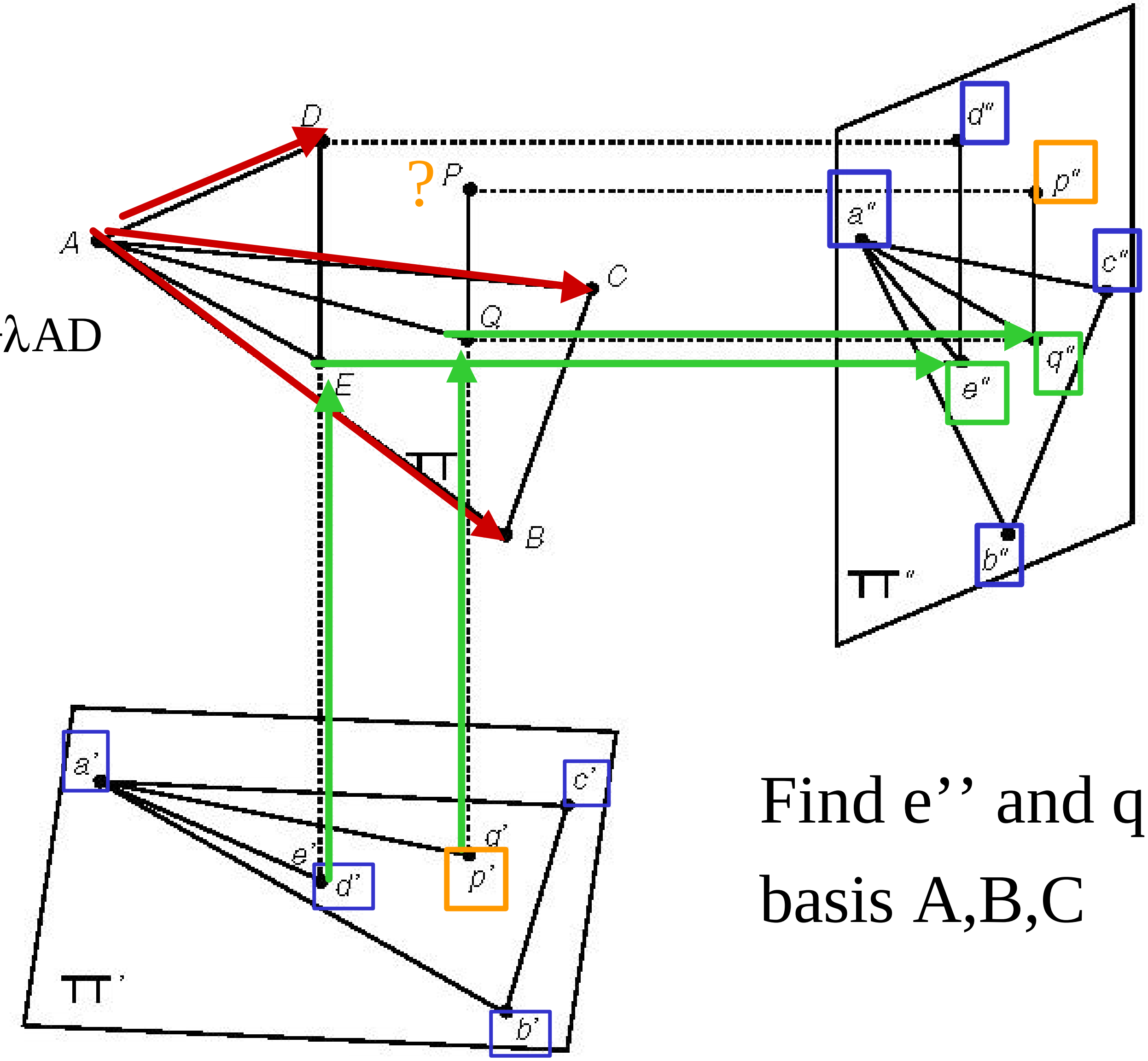
$$P = \alpha AB + \beta AC + \lambda AD$$

Find α, β, λ ?

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…

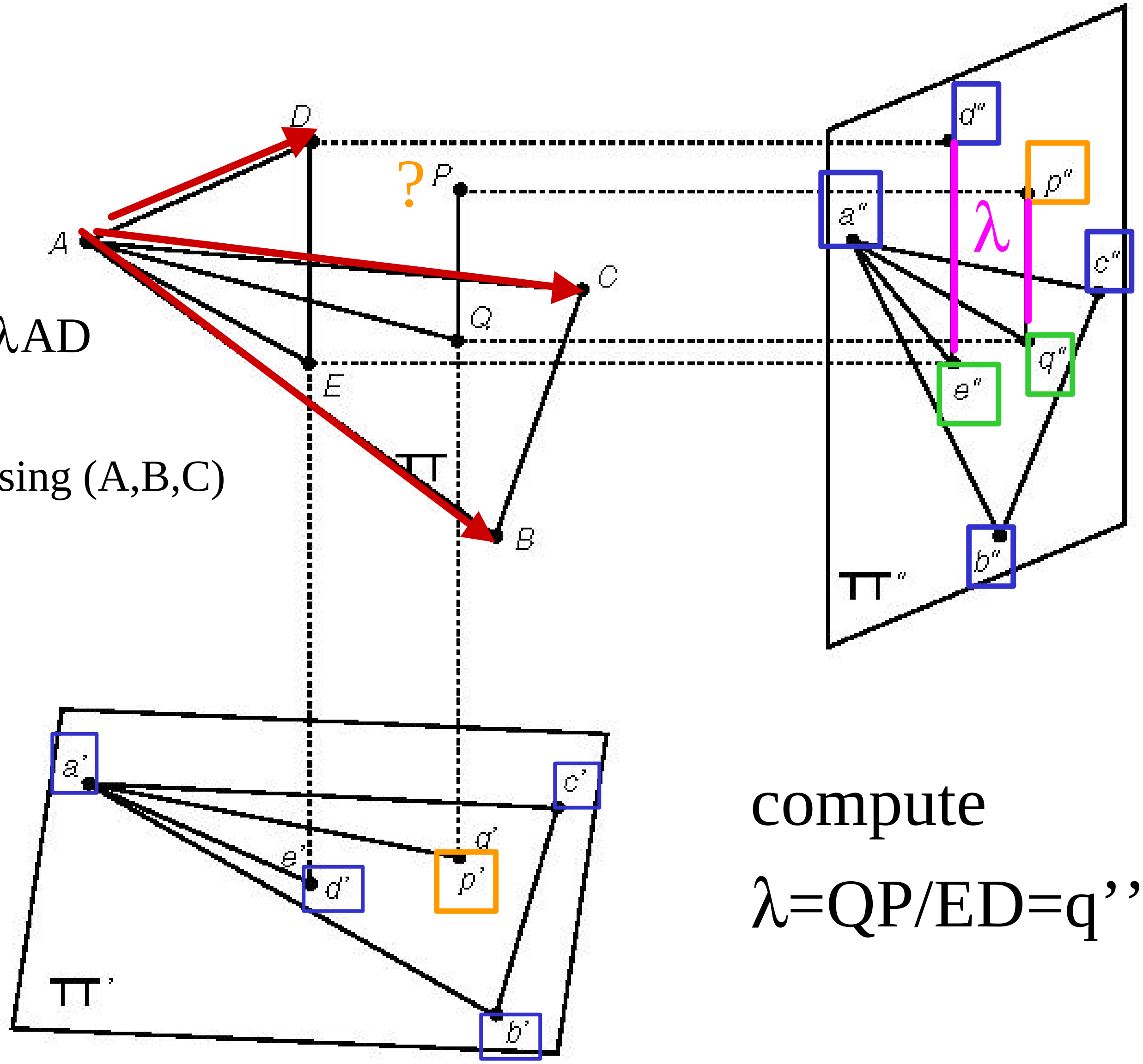
P=αAB+βAC+λAD

Find α,β,λ ?



Find e’’ and q’’ using basis A,B,C

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…



$P = \alpha AB + \beta AC + \lambda AD$

Find α, β, λ ?

Find e’’ and q’’ using (A,B,C)

compute

$\lambda = QP/ED = q''p''/e''d''$

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…

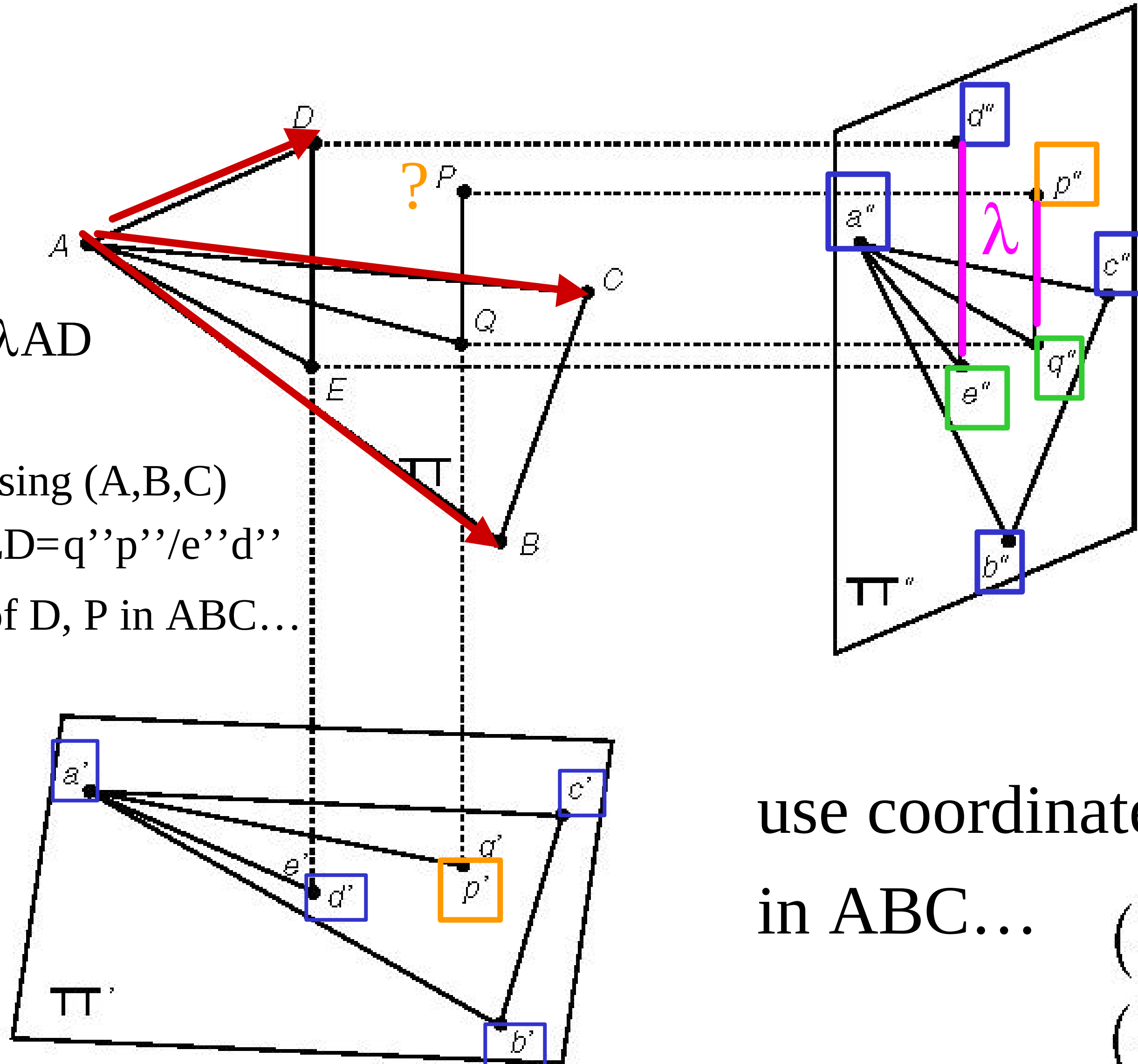
P=αAB+βAC+λAD

Find α,β,λ ?

Find e’’ and q’’ using (A,B,C)

Compute λ=QP/ED=q’’p’’/e’’d’’

Use coordinates of D, P in ABC…

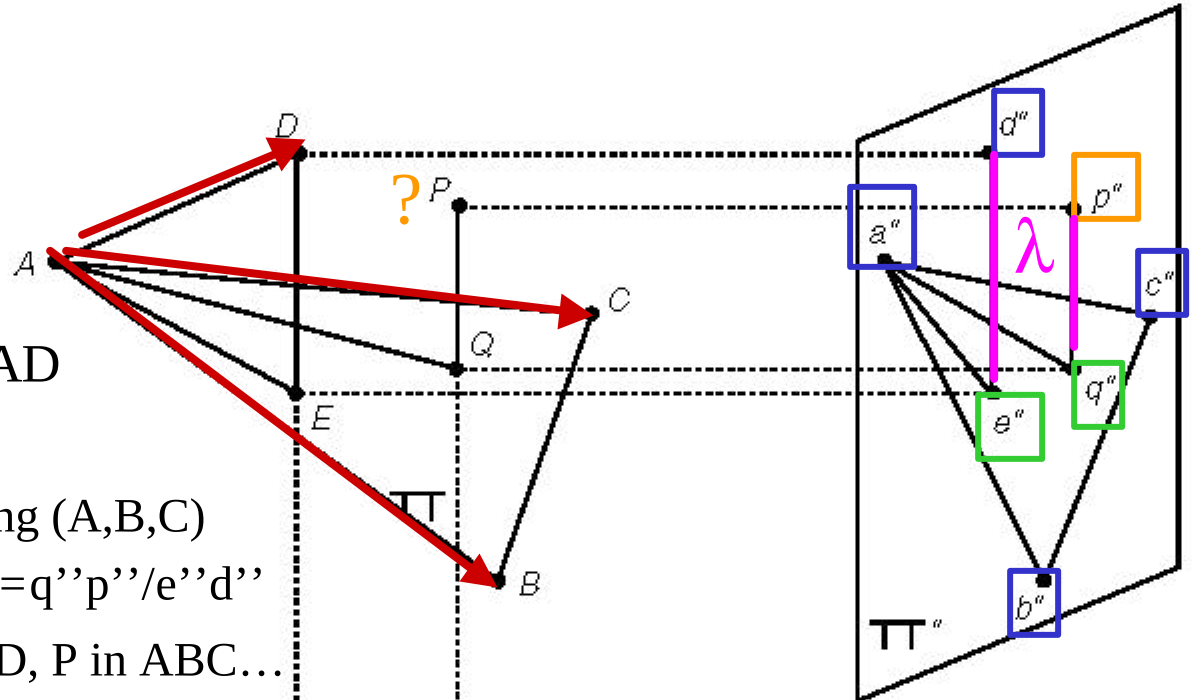


use coordinates of D, P in ABC…

$$(\alpha_{d'}, \beta_{d'})$$

$$(\alpha_{p'}, \beta_{p'})$$

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…



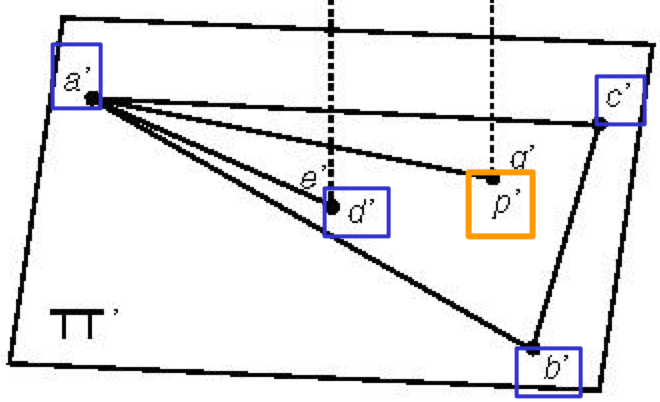
P=αAB+βAC+λAD

Find α,β,λ ?

Find e’’ and q’’ using (A,B,C)

Compute λ=QP/ED=q’’p’’/e’’d’’

Use coordinates of D, P in ABC…



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

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…

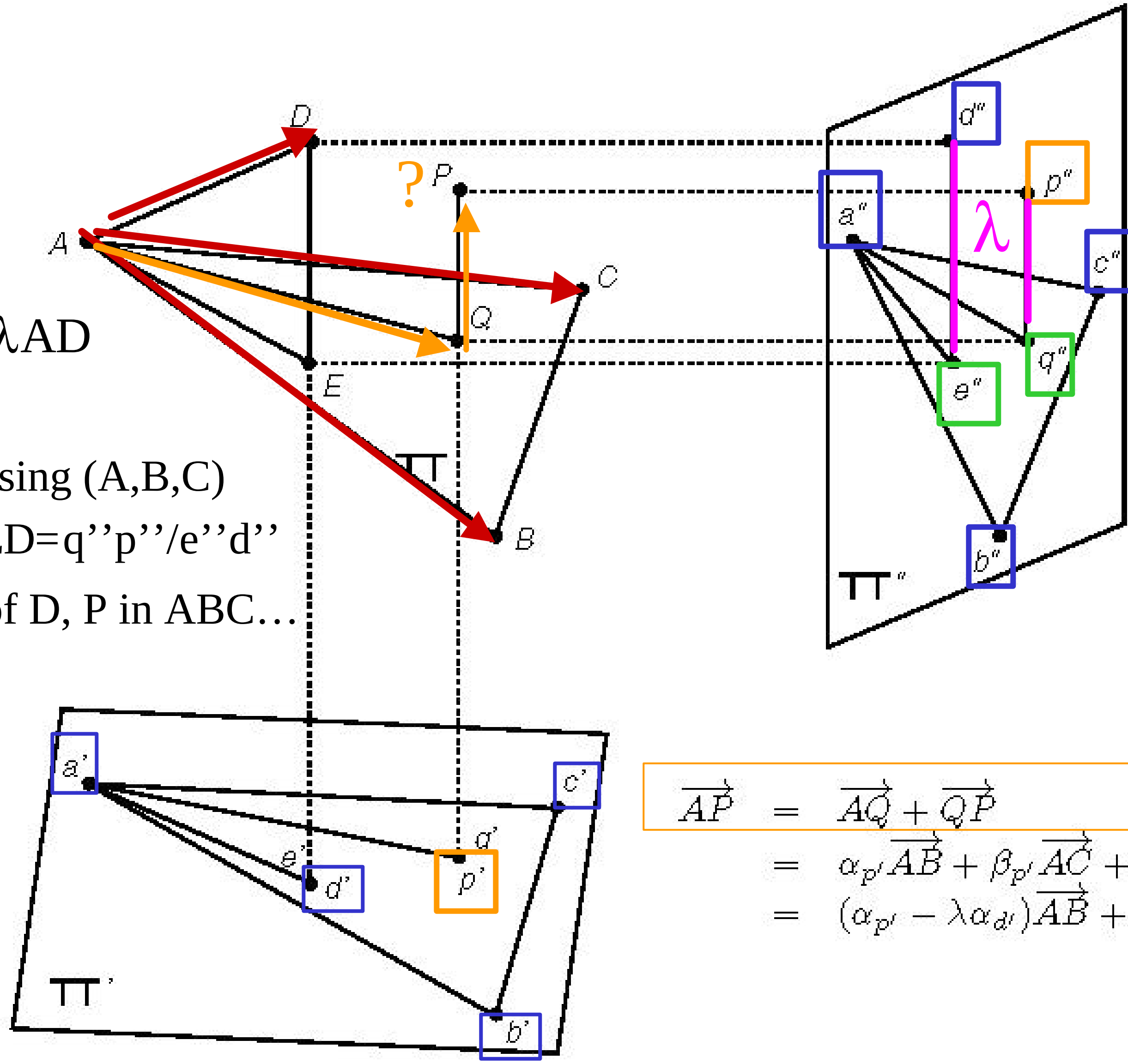
$P=\alpha AB+\beta AC+\lambda AD$

Find α,β,λ ?

Find e’’ and q’’ using (A,B,C)

Compute $\lambda = QP/ED = q''p''/e''d''$

Use coordinates of D, P in ABC…



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

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…

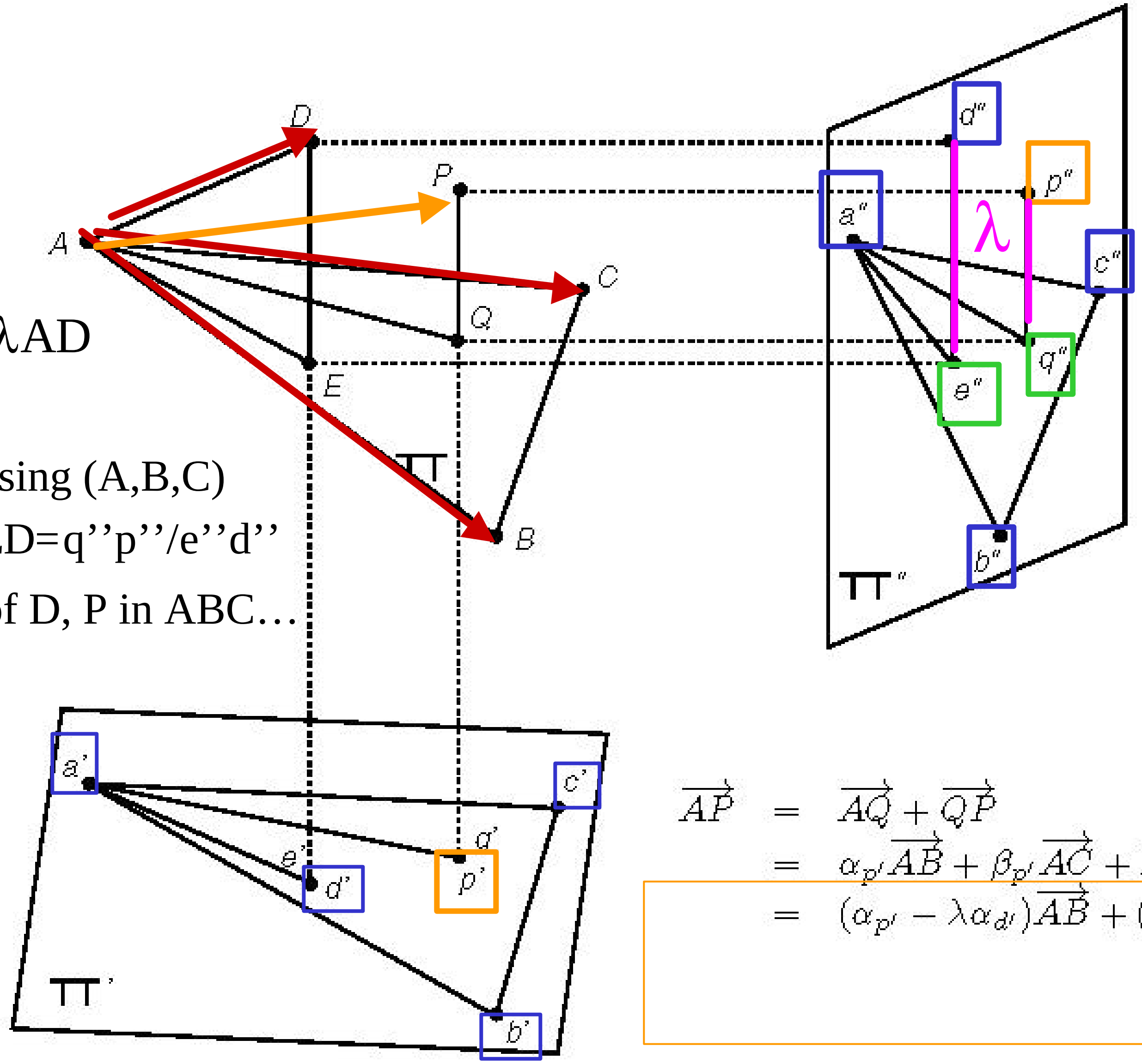
$P = \alpha AB + \beta AC + \lambda AD$

Find α, β, λ ?

Find e’’ and q’’ using (A,B,C)

Compute $\lambda = QP/ED = q''p''/e''d''$

Use coordinates of D, P in ABC…



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

Geometric Approach

p’,p’’ uniquely determined the location of P in the basis (A,B,C,D)

AP was expressed using weighted combination of AB, AC, AD

Weights were determined by a’,a’’,b’,b’’,c’,c’’,d’,d’’,p’,p’’.

Affine Structure from Motion

- Two views
  - Geometric Approach: infer affine shape (then recover affine projection matricies if needed)
  - Algebraic Approach: estimate projection matricies (then determine position of scene points)
- Sequence
  - Factorization Approach

Algebraic approach

3-d P satisfies two affine views:

$$\begin{aligned} \boldsymbol{p} &= \mathcal{A}\boldsymbol{P} + \boldsymbol{b}, \\ \boldsymbol{p}' &= \mathcal{A}'\boldsymbol{P} + \boldsymbol{b}', \end{aligned}$$

$$\begin{pmatrix} \mathcal{A} & \boldsymbol{p} - \boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}' - \boldsymbol{b}' \end{pmatrix} \begin{pmatrix} \boldsymbol{P} \\ -1 \end{pmatrix} = \mathbf{0}.$$

$$\mathrm{Det} \begin{pmatrix} \mathcal{A} & \boldsymbol{p} - \boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}' - \boldsymbol{b}' \end{pmatrix} = 0$$

$$\mathrm{Det}\begin{pmatrix} \mathcal{A} & \boldsymbol{p}-\boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}'-\boldsymbol{b}' \end{pmatrix} = 0$$

But any affine transform of A is equally good…

$$\mathrm{Det}\begin{pmatrix} \mathcal{A}\mathcal{C} & \boldsymbol{p}-\mathcal{A}\boldsymbol{d}-\boldsymbol{b} \\ \mathcal{A}'\mathcal{C} & \boldsymbol{p}'-\mathcal{A}'\boldsymbol{d}-\boldsymbol{b}' \end{pmatrix} = 0$$

for any affine transform

$$\mathcal{Q} = \begin{pmatrix} \mathcal{C} & \boldsymbol{d} \\ \mathbf{0}^T & 1 \end{pmatrix}$$

$$\mathrm{Det}\begin{pmatrix} \mathcal{A}\mathcal{C} & \boldsymbol{p} - \mathcal{A}\boldsymbol{d} - \boldsymbol{b} \\ \mathcal{A}'\mathcal{C} & \boldsymbol{p}' - \mathcal{A}'\boldsymbol{d} - \boldsymbol{b}' \end{pmatrix} = 0 \qquad \mathcal{Q} = \begin{pmatrix} \mathcal{C} & \boldsymbol{d} \\ \boldsymbol{0}^T & 1 \end{pmatrix}$$

Let's pick a special C, d…

$$\mathcal{C} = \mathcal{S}^{-1} \qquad \mathcal{S} = \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \boldsymbol{a}_1'^T \end{pmatrix} \qquad \boldsymbol{r} = \begin{pmatrix} b_1 \\ b_2 \\ b_1' \end{pmatrix}$$

$$\boldsymbol{d} = -\mathcal{S}^{-1}\boldsymbol{r}.$$

which is equivalent to choosing cannonical affine projection matrices

$$\tilde{\mathcal{M}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad \tilde{\mathcal{M}}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix}$$

and our determinant becomes very simple:

$$\mathrm{Det}\begin{pmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & u' \\ a & b & c & v'-d \end{pmatrix} = au - bv + cu' + v' - d = 0$$

a,b,c,d can be estimated using least squares with a sufficient number of points. Then P can be recovered with:

$$\begin{pmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & u' \\ a & b & c & v'-d \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{P}} \\ -1 \end{pmatrix} = 0$$

Affine Structure from Motion

- Two views
  - Geometric Approach: infer affine shape (then recover affine projection matricies if needed)
  - Algebraic Approach: estimate projection matricies (then determine position of scene points)
- Sequence
  - Factorization Approach

Factorization Approach

Consider a sequence of affine cameras….

$$\boldsymbol{p}_i = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P} \\ 1 \end{pmatrix} = \mathcal{A}_i \boldsymbol{P} + \boldsymbol{b}_i$$

Stack affine projection equations:

$$\boldsymbol{q} = \boldsymbol{r} + \mathcal{A}\boldsymbol{P}$$

$$\boldsymbol{q} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{p}_1 \\ \dots \\ \boldsymbol{p}_m \end{pmatrix}, \quad \boldsymbol{r} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{b}_1 \\ \dots \\ \boldsymbol{b}_m \end{pmatrix} \quad \text{and} \quad \mathcal{A} \stackrel{\text{def}}{=} \begin{pmatrix} \mathcal{A}_1 \\ \dots \\ \mathcal{A}_m \end{pmatrix}$$

$$\boldsymbol{q} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{p}_1 \\ \cdots \\ \boldsymbol{p}_m \end{pmatrix}, \quad \boldsymbol{r} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{b}_1 \\ \cdots \\ \boldsymbol{b}_m \end{pmatrix} \quad \text{and} \quad \mathcal{A} \stackrel{\text{def}}{=} \begin{pmatrix} \mathcal{A}_1 \\ \cdots \\ \mathcal{A}_m \end{pmatrix}$$

Form the (2m+1)n data matrix where each column is the observed data from one point:

$$\mathcal{D} = \begin{pmatrix} \boldsymbol{q}_1 & \cdots & \boldsymbol{q}_n \\ 1 & \cdots & 1 \end{pmatrix}$$

$$\boldsymbol{q} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{p}_1 \\ \dots \\ \boldsymbol{p}_m \end{pmatrix}, \quad \boldsymbol{r} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{b}_1 \\ \dots \\ \boldsymbol{b}_m \end{pmatrix} \quad \text{and} \quad \mathcal{A} \stackrel{\text{def}}{=} \begin{pmatrix} \mathcal{A}_1 \\ \dots \\ \mathcal{A}_m \end{pmatrix}$$

Form the (2m+1)n data matrix where each column is the observed data from one point:

$$\mathcal{D} = \begin{pmatrix} \boldsymbol{q}_1 & \dots & \boldsymbol{q}_n \\ 1 & \dots & 1 \end{pmatrix}$$

Since

$$\boldsymbol{q} = \boldsymbol{r} + \mathcal{A}\boldsymbol{P}$$

then

Rank(D) <= 4

With an appropriate choice of origin (e.g., first point, centriod),

$$\boldsymbol{p}_i = \mathcal{A}_i \boldsymbol{P} \qquad\qquad \boldsymbol{q} = \mathcal{A}\boldsymbol{P},$$

and the data matrix becomes:

$$\mathcal{D} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{q}_1 & \ldots & \boldsymbol{q}_n \end{pmatrix} = \mathcal{A}\mathcal{P}$$

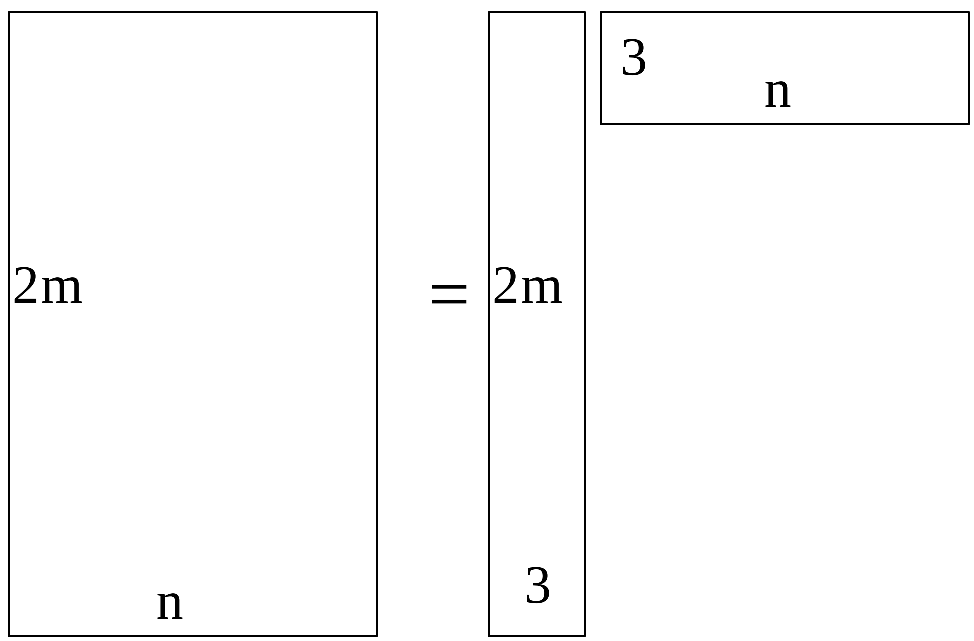
$$\mathcal{P} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1 & \ldots & \boldsymbol{P}_n \end{pmatrix}.$$

Rank of Object-relative Data Matrix

D = A P

Data-Matrix = Affine-Motions x 3-d-Points

(2m x n) = (2m x 3) x (3 x n)



D is now rank 3

Factorization algorithm

Given a data matrix,

find Motion (A) and Shape (P) matrices that generate that data…

Tomasi and Kanade Factorization algorithm (1992):

Use Singular Value Decomposition to factor D into appropriately sized A and P.

SVD

Technique: Singular Value Decomposition Let $\mathcal{A}$ be an $m \times n$ matrix, with $m \geq n$, then $\mathcal{A}$ can always be written as

$$\mathcal{A} = \mathcal{U}\mathcal{W}\mathcal{V}^T,$$

where:

- $\mathcal{U}$ is an $m \times n$ column-orthogonal matrix, i.e., $\mathcal{U}^T\mathcal{U} = \mathrm{Id}_m$,
- $\mathcal{W}$ is a diagonal matrix whose diagonal entries w_i $(i = 1, \ldots, n)$ are the singular values of $\mathcal{A}$ with $w_1 \geq w_2 \geq \ldots \geq w_n \geq 0$,
- and $\mathcal{V}$ is an $n \times n$ orthogonal matrix, i.e., $\mathcal{V}^T\mathcal{V} = \mathcal{V}\mathcal{V}^T = \mathrm{Id}_n$.

The SVD of a matrix can also be used to characterize matrices that are rank-deficient: suppose that $\mathcal{A}$ has rank $p < n$, then the matrices $\mathcal{U}$, $\mathcal{W}$, and $\mathcal{V}$ can be written as

$$\mathcal{U} = \begin{array}{|c|c|}\hline \mathcal{U}_p & \mathcal{U}_{n-p} \\ \hline\end{array} \quad \mathcal{W} = \begin{array}{|c|c|}\hline \mathcal{W}_p & 0 \\ \hline 0 & 0 \\ \hline\end{array} \quad \text{and} \quad \mathcal{V}^T = \begin{array}{|c|}\hline \mathcal{V}_p^T \\ \hline \mathcal{V}_{n-p}^T \\ \hline\end{array},$$

Factorization algorithm

1. Compute the singular value decomposition $\mathcal{D} = \mathcal{U}\mathcal{W}\mathcal{V}^T$.

2. Construct the matrices $\mathcal{U}_3$, $\mathcal{V}_3$, and $\mathcal{W}_3$ formed by the three leftmost columns of the matrices $\mathcal{U}$ and $\mathcal{V}$, and the corresponding 3×3 sub-matrix of $\mathcal{W}$.

3. Define

$$\mathcal{A}_0 = \mathcal{U}_3 \quad \text{and} \quad \mathcal{P}_0 = \mathcal{W}_3\mathcal{V}_3^T;$$

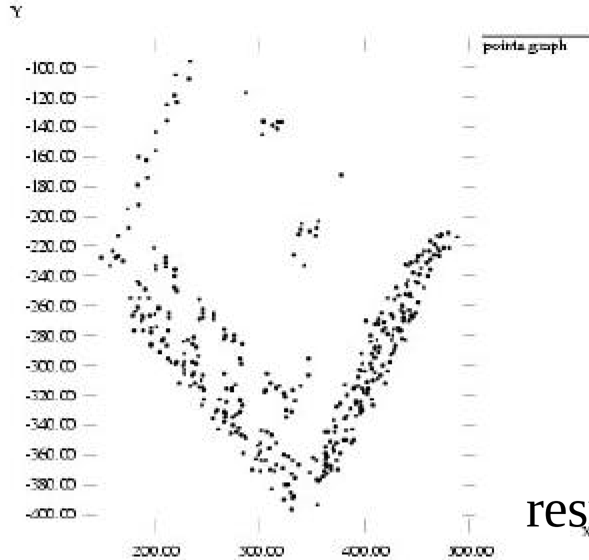
the $2m \times 3$ matrix $\mathcal{A}_0$ is an estimate of the camera motion, and the $3 \times n$ matrix $\mathcal{P}_0$ is an estimate of the scene structure.

Factorization algorithm

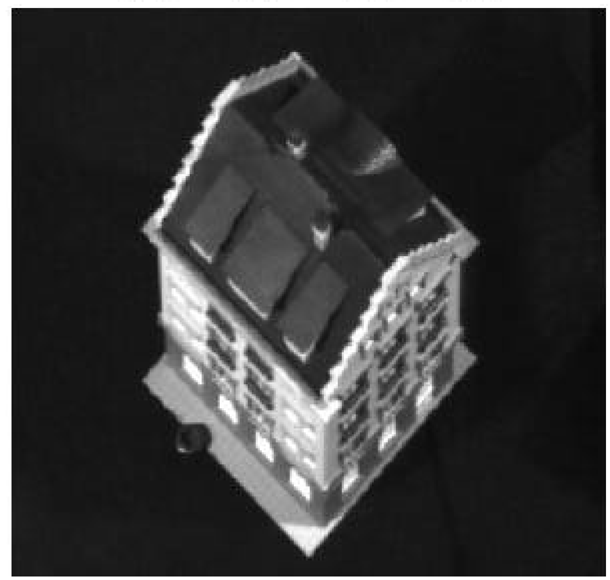
1

60

Input



result



comparision

Factorization algorithm

Can perform *Euclidean upgrade* to estimate metric quantities...

- Of all the family of affine solutions, find the one that obeys calibration constraints.

Extensions to basic algorithm:

- sparse data
- multiple motions
- projective cameras (later)

Affine Structure from Motion

- Two views
  - Geometric Approach: infer affine shape (then recover affine projection matrices if needed)
  - Algebraic Approach: estimate projection matrices (then determine position of scene points)
- Sequence
  - Factorization Approach

[Most Figures from Forsythe and Ponce]