CIS 5800

Machine Perception

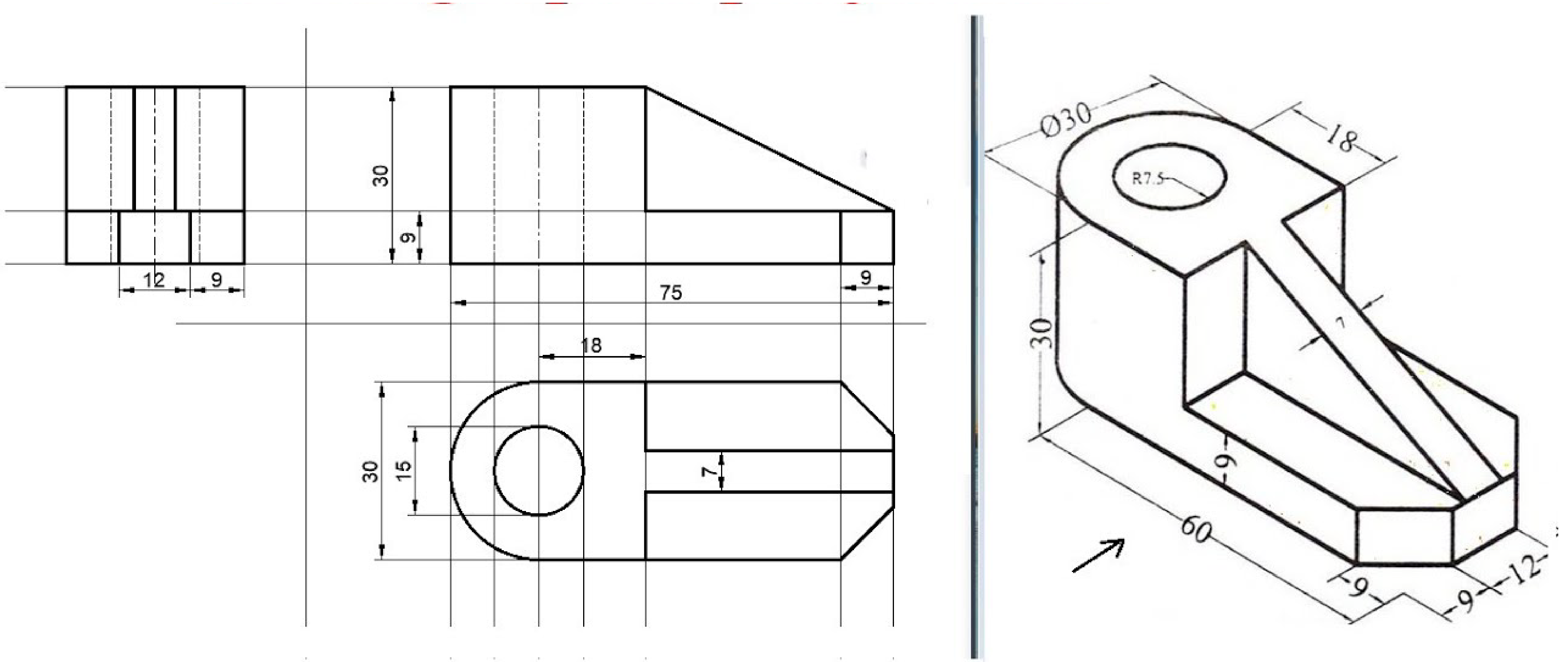
Instructor: Lingjie Liu

Lec 8: Feb 14, 2024

Administrivia

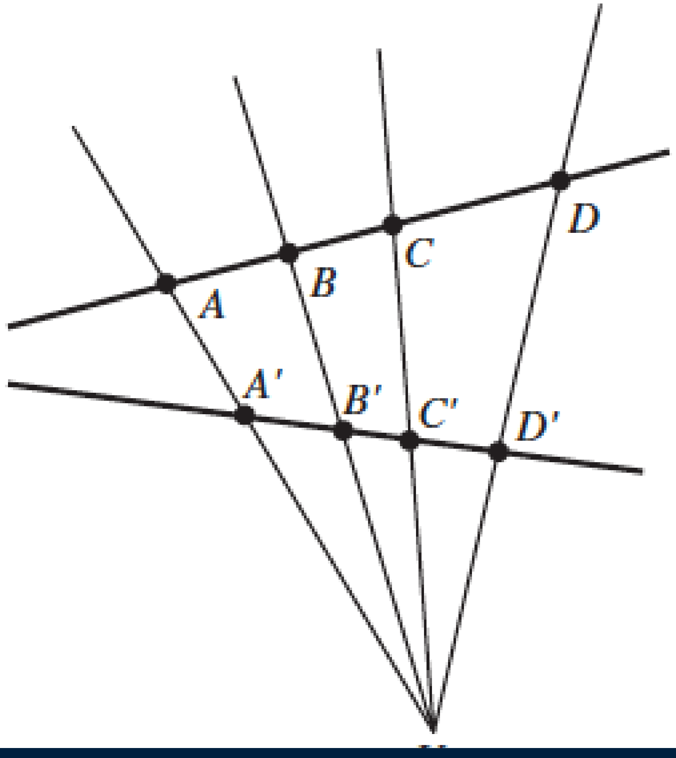
- HW1, due TODAY.
- HW2, will release this Thursday.

Aside: Engineers Use "Orthographic" Projections



- Image formed by rays orthogonal to the image plane, hence "orthographic". Field of view limited by size of image plane.
- In perspective projection which happens in eyes and cameras, rays pass through a camera center. Much larger field of view.

Recap: Cross Ratios of Collinear Points

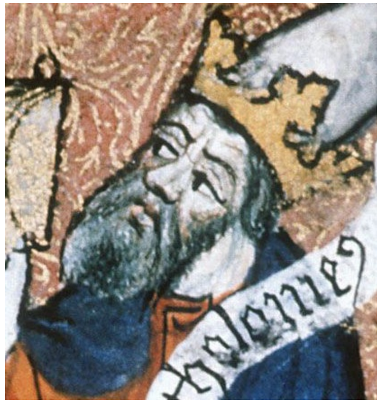


Given four points A, B, C, D, we define the cross-ratio of their distances as

$$CR(A, B, C, D) = \frac{AC}{AD} : \frac{BC}{BD}.$$

$CR(A, B, C, D)$ remains invariant under projective transformations

$$\frac{AC}{AD} : \frac{BC}{BD} = \frac{A'C'}{A'D'} : \frac{B'C'}{B'D'}$$



Pappus (290-350 AD)

Denote $\mathbf{x}'_i = (x'_i, 1)^T$ and $\mathbf{x}_i = (x_i, 1)^T$. Suppose $\mathbf{x}'_i = H\mathbf{x}_i$. Note $|H| = \det(H)$. Then

$$x'_i - x'_j = |\mathbf{x}'_i \ \ \mathbf{x}'_j| = |\lambda_i H\mathbf{x}_i \ \ \lambda_j H\mathbf{x}_j| = \lambda_i\lambda_j |H(\mathbf{x}_i \ \ \mathbf{x}_j)|$$

$$= \lambda_i\lambda_j |H||\mathbf{x}_i \ \ \mathbf{x}_j| = \lambda_i\lambda_j \det(H)(x_i - x_j)$$

Note: C

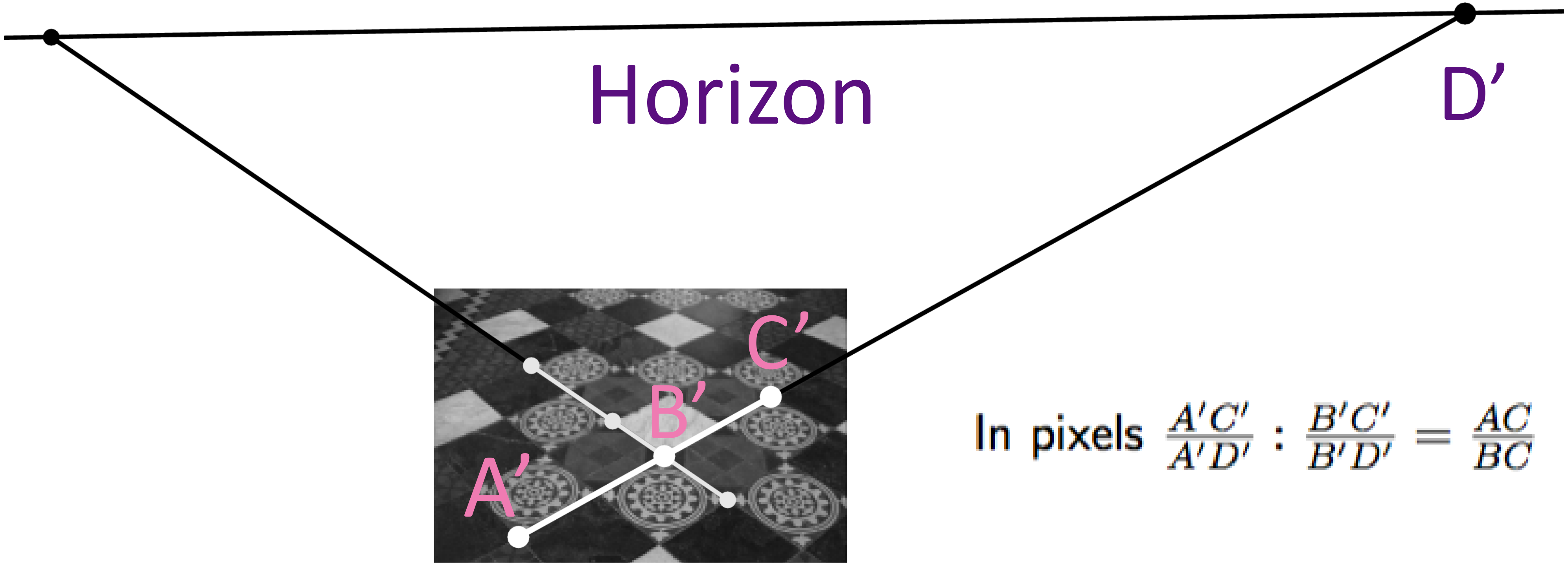
but it is

e.g., CR

Girard Desargues (1591-1661)

Brannan et al

Recap: Cross ratios for metrology



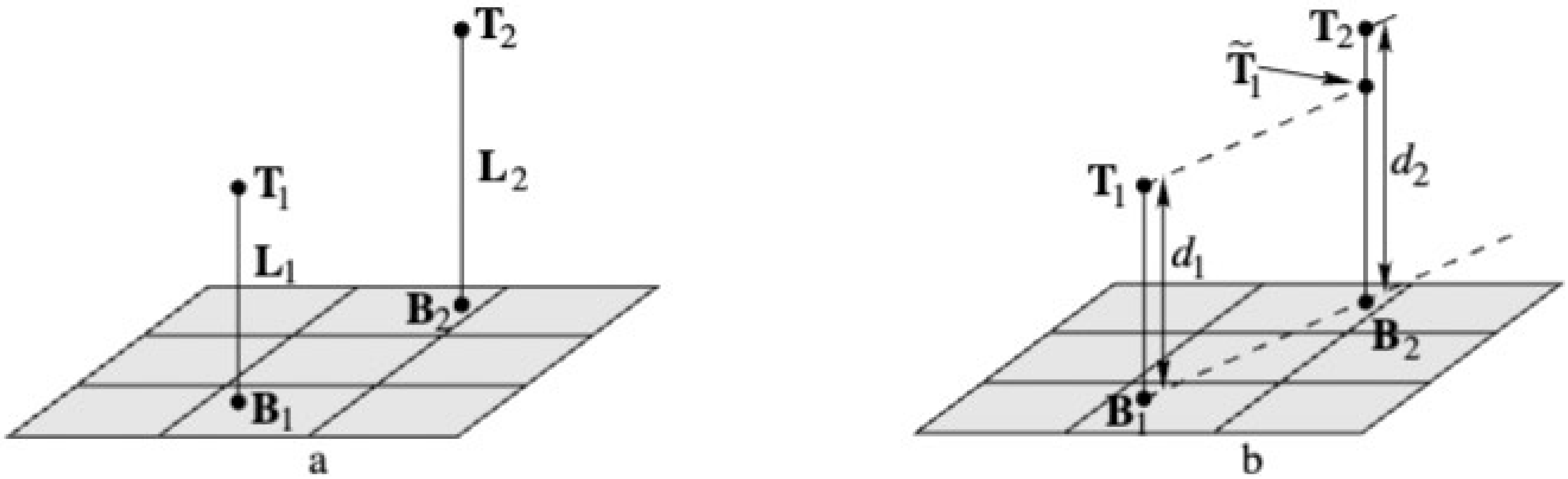
In pixels $\frac{A'C'}{A'D'} : \frac{B'C'}{B'D'} = \frac{AC}{BC}$

When a point D is at infinity, the cross-ratio becomes a ratio !

$\frac{AC}{AD} : \frac{BC}{BD} = \frac{AC}{BC}$ (Think $\frac{AC}{\infty} : \frac{BC}{\infty} = \frac{AC}{BC} \times \frac{\infty}{\infty} = \frac{AC}{BC}$)

Recap: Length transfer in 3D

- In the real 3-D world, you can compare one object with known length to another to “transfer” its length. This is what you do with a ruler, for example.



How to do this in an image?

ZH Sec 8.7

Recap: “Fixing pose from homography”

- $K^{-1}H$ is supposed to be our estimate of $[r_1, r_2, T]$, but there are no guarantees that r_1 and r_2 are in fact orthonormal, as they should be.

- So we construct the “Procrustes” problem of finding the “closest valid rotation matrix R” to $K^{-1}H$

We find the orthogonal matrix R that is the closest to $\begin{pmatrix} h_1' & h_2' & h_1' \times h_2' \end{pmatrix}$:

$$\arg\min_{R \in SO(3)} \| R - \begin{pmatrix} h_1' & h_2' & h_1' \times h_2' \end{pmatrix} \|_F^2$$

Recap: Kabsch algorithm for Procrustes

$$\boxed{\arg\min_{R \in SO(3)} \| R - \begin{pmatrix} h_1' & h_2' & h_1' \times h_2' \end{pmatrix} \|_F^2}$$

If the SVD of

$$\begin{pmatrix} h_1' & h_2' & h_1' \times h_2' \end{pmatrix} = USV^T$$

then the solution is

$$R = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{pmatrix} V^T$$

The diagonal matrix is inserted to guarantee that $\det(R) = 1$.

To find the translation : $T = h_3' / \|h_1'\|$

(In case original columns were not even unit norm)

proof in supp readings- Kabsch-Algorithm-RT-from-H-proof.pdf. We will also prove it in the next class.

Proof in class notes, optional

References: The Kabsch papers

Acta Cryst. (1976). A**32**, 922

A solution for the best rotation to relate two sets of vectors. By WOLFGANG KABSCH, *Max-Planck-Institut für Medizinische Forschung*, 6900 *Heidelberg, Jahnstrasse* 29, *Germany* (*BRD*)

(*Received* 23 *February* 1976; *accepted* 12 *April* 1976)

A simple procedure is derived which determines a best rotation of a given vector set into a second vector set by minimizing the weighted sum of squared deviations. The method is generalized for any given metric constraint on the transformation.

In various crystallographic situations the problem arises of finding a best rotation to fit a given atomic arrangement to approximately measured coordinates. Examples have been given by McLachlan (1972) and Diamond (1976). Diamond determines the best unconstrained transformation between the two sets of coordinates and factorizes it into a symmetric and an orthogonal matrix. McLachlan finds a best rotation between the two sets of coordinates by an iterative process. The analysis below shows that a direct solution is also possible, despite the non-linear character of the problem.

Let $\mathbf{x}_n$ and $\mathbf{y}_n$ ($n=1,2,\ldots,N$) be two given vector sets and w_n the weight corresponding to each pair $\mathbf{x}_n,\mathbf{y}_n$. The problem is then to find an orthogonal matrix $\mathsf{U}=(u_{ij})$ which minimizes the function

$$E=\tfrac{1}{2}\sum_n w_n(\mathsf{U}\mathbf{x}_n-\mathbf{y}_n)^2 \tag{1}$$

subject to the constraints

$$\sum_k u_{ki}u_{kj}-\delta_{ij}=0 \tag{2}$$

where the δ_{ij} are the elements of the unit matrix. A translation, if admitted, can always be removed from the problem by shifting the centroids of the vector sets to the origin.

Introducing a symmetric matrix $\mathsf{L}=(l_{ij})$ of Lagrange multipliers an auxiliary function (see, for example, Brand, 1958)

$$F=\tfrac{1}{2}\sum_{i,j} l_{ij}\Big(\sum_k u_{ki}u_{kj}-\delta_{ij}\Big) \tag{3}$$

is constructed and added to E to form the Lagrangian function

$$G=E+F\,. \tag{4}$$

Since for each different condition (2) an independent number l_{ij} is available, the constrained minimum of E is now included among the free minima of G. A free minimum of G can only occur where

$$\frac{\partial G}{\partial u_{ij}}=\sum_k u_{ik}\Big(\sum_n w_n x_{nk}x_{nj}+l_{kj}\Big)-\sum_n w_n y_{ni}x_{nj}=0 \tag{5}$$

and

$$\frac{\partial^2 G}{\partial u_{mk}\partial u_{ij}}=\delta_{mi}\Big(\sum_n w_n x_{nk}x_{nj}+l_{kj}\Big) \tag{6}$$

are the elements of a positive definite matrix. x_{nk} and y_{nk} are the kth components of the vectors $\mathbf{x}_n$ and $\mathbf{y}_n$.

Let

$$r_{ij}=\sum_n w_n y_{ni}x_{nj} \tag{7}$$

and

$$s_{ij}=\sum_n w_n x_{ni}x_{nj} \tag{8}$$

be the elements of a matrix $\mathsf{R}=(r_{ij})$ and a symmetric matrix $\mathsf{S}=(s_{ij})$, respectively. For $i=m=1$ from equation (6), a minimum of the Lagrangian function G requires that $\mathsf{S}+\mathsf{L}$ is positive definite, and – by rewriting equation (5) – that

$$\mathsf{U}\,.\,(\mathsf{S}+\mathsf{L})=\mathsf{R}\,. \tag{9}$$

The problem is now to find a symmetric matrix L of Lagrange multipliers such that U is orthogonal. If both sides of (9) are multiplied by their transposed matrices, the unknown orthogonal matrix U can be eliminated:

$$\begin{aligned}\mathsf{U}\widetilde{(\mathsf{S}+\mathsf{L})}\mathsf{U}(\mathsf{S}+\mathsf{L})&=\widetilde{(\mathsf{S}+\mathsf{L})}\tilde{\mathsf{U}}\mathsf{U}(\mathsf{S}+\mathsf{L})\\&=(\mathsf{S}+\mathsf{L})\,(\mathsf{S}+\mathsf{L})=\tilde{\mathsf{R}}\mathsf{R}\,.\end{aligned} \tag{10}$$

Since $\tilde{\mathsf{R}}\mathsf{R}$ is a symmetric positive definite matrix the positive eigenvalues μ_k and the corresponding eigenvectors $\mathbf{a}_k$ can be found by well established procedures. Since $\mathsf{S}+\mathsf{L}$ is symmetric and positive definite also, it is evident from (10) that it must have the same normalized eigenvectors $\mathbf{a}_k$ and the positive eigenvalues $\sqrt{\mu_k}$.

It can be easily verified that the Lagrange multipliers are then

$$l_{ij}=\sum_k \sqrt{\mu_k}\qquad a_{ki}a_{kj}-s_{ij} \tag{11}$$

where a_{ki} denotes the ith component of $\mathbf{a}_k$. The effect of the orthogonal matrix U on these eigenvectors $\mathbf{a}_k$ is determined from (9) and defines unit vectors $\mathbf{b}_k$ as

$$\mathbf{b}_k=\mathsf{U}\,.\,\mathbf{a}_k=\frac{1}{\sqrt{\mu_k}}\,\mathsf{U}(\mathsf{S}+\mathsf{L})\mathbf{a}_k=\frac{1}{\sqrt{\mu_k}}\,\mathsf{R}\mathbf{a}_k\,. \tag{12}$$

The orthogonal matrix U is finally constructed as

$$u_{ij}=\sum_k b_{ki}a_{kj} \tag{13}$$

and the problem to find the constraint minimum of the function E is solved.

Sometimes it may happen that all of the vectors $\mathbf{x}_n$ or $\mathbf{y}_n$ lie in a plane. Then one of the eigenvalues of $\tilde{\mathsf{R}}\mathsf{R}$, *e.g.* μ_3, will be zero. In this case a complete set of vectors $\mathbf{a}_k,\mathbf{b}_k$ is constructed by setting

$$\mathbf{a}_3=\mathbf{a}_1\times\mathbf{a}_2\qquad \mathbf{b}_3=\mathbf{b}_1\times\mathbf{b}_2\,. \tag{14}$$

Note that the procedure described in this article can be easily extended to vector spaces of higher dimensions.

It is possible also to replace the constraints of equation (2) by the more general constraints

$$\tilde{\mathsf{U}}\mathsf{U}=\mathsf{M}\,, \tag{15}$$

where M is a symmetric and positive definite matrix. If B is any specific solution of (15), it is easy to prove that all possible other solutions U of that equation can be written as

$$\mathsf{U}=\mathsf{V}\,.\,\mathsf{B} \tag{16}$$

with an orthogonal matrix V. If the initial vector set $\mathbf{x}_n$ is transformed into $\mathbf{x}'_n=\mathsf{B}\mathbf{x}_n$ then this problem is reduced to minimizing $E'=\frac{1}{2}\sum_n w_n(\mathsf{V}\mathbf{x}'_n-\mathbf{y}_n)^2$ with the constraint $\tilde{\mathsf{V}}\mathsf{V}=1$.

I would like to thank Dr K. C. Holmes for reading the manuscript.

References

DIAMOND, R. (1976). *Acta Cryst.* A**32**, 1–10.
MCLACHLAN, A. D. (1972). *Acta Cryst.* A**28**, 656–657.
BRAND, L. (1958). *Advanced Calculus*, pp. 194–198. New York: John Wiley.

Acta Cryst. (1978). A**34**, 827–828

A discussion of the solution for the best rotation to relate two sets of vectors. BY W. KABSCH, *Max-Planck-Institut für Medizinische Forschung, Abteilung Biophysik, Jahnstrasse* 29, 6900 *Heidelberg, Federal Republic of Germany*

(*Received* 3 *April* 1978; *accepted* 13 *April* 1978)

A method is discussed for obtaining the best proper rotation to relate two sets of vectors.

The simple procedure for obtaining the best rotation to relate two sets of vectors described in an earlier paper (Kabsch, 1976) has been used in processing oscillation films (Kabsch, 1977), for the determination of non-crystallographic symmetry elements (Kabsch, Gast, Schulz & Leberman, 1977), and for a comparison of macromolecules. In the last application an improper rotation was sometimes obtained from the procedure (Nyburg & Yuen, 1977). The purpose of this communication is to show how a best proper rotation can always be obtained from the procedure.

Let $\mathbf{x}_n$ and $\mathbf{y}_n$ ($n=1,\ldots,N$) be two given vector sets and w_n the weights corresponding to each pair $\mathbf{x}_n$, $\mathbf{y}_n$. All possible orthogonal matrices $\mathbf{U}$ for which the function

$$E=\tfrac{1}{2}\sum_n w_n(\mathbf{U}\mathbf{x}_n-\mathbf{y}_n)^2 \tag{1}$$

has an extremal point must obey [see equation (9) of Kabsch, 1976]

$$\mathbf{U}(\mathbf{S}+\mathbf{L})=\mathbf{R}. \tag{2}$$

Writing x_{nk} and y_{nk} for the kth components of the vectors $\mathbf{x}_n$ and $\mathbf{y}_n$ the matrices $\mathbf{R}$ and $\mathbf{S}$ are defined as

$$\mathbf{R}=(r_{ij})=\Big(\sum_n w_n y_{ni}x_{nj}\Big) \tag{3}$$

$$\mathbf{S}=(s_{ij})=\Big(\sum_n w_n x_{ni}x_{nj}\Big). \tag{4}$$

$\mathbf{L}=(l_{ij})$ is a symmetric matrix of Lagrange multipliers which is determined from the equation

$$(\mathbf{S}+\mathbf{L})(\mathbf{S}+\mathbf{L})=\tilde{\mathbf{R}}\mathbf{R}. \tag{5}$$

$\tilde{\mathbf{R}}\mathbf{R}$ is a known symmetric positive definite matrix whose positive eigenvalues μ_k and eigenvectors $\mathbf{a}_k$ can be determined by standard procedures. The general solution of (5) is of the form

$$(\mathbf{S}+\mathbf{L})=(s_{ij}+l_{ij})=\Big(\sum_k a_{ki}a_{kj}\,.\,\sigma_k\sqrt{\mu_k}\Big), \tag{6}$$

where a_{ki} denotes the ith component of $\mathbf{a}_k$ and the arbitrary quantities σ_k can only assume the values ± 1. If an eigenvalue μ_k is degenerate the eigenvector $\mathbf{a}_k$ of $\tilde{\mathbf{R}}\mathbf{R}$ cannot be determined uniquely. However, $\mathbf{S}+\mathbf{L}$ will not be affected by this ambiguity if all its eigenvalues of the magnitude $\sqrt{\mu_k}$ have identical signs. The final construction of all orthogonal matrices $\mathbf{U}=(u_{ij})$ for which E assumes an extremal point is given by

$$u_{ij}=\sum_k b_{ki}a_{kj}, \tag{7}$$

where b_{ki} is the ith component of the vector

$$\mathbf{b}_k=\mathbf{U}\mathbf{a}_k=\mathbf{U}(\mathbf{S}+\mathbf{L})\mathbf{a}_k/(\sigma_k\sqrt{\mu_k})=\mathbf{R}\mathbf{a}_k/(\sigma_k\sqrt{\mu_k}). \tag{8}$$

The residual E at each extremal point is

$$\begin{aligned}E&=\tfrac{1}{2}\sum_n w_n(\mathbf{U}\mathbf{x}_n-\mathbf{y}_n)^2=\tfrac{1}{2}\sum_n w_n(\mathbf{x}_n^2+\mathbf{y}_n^2)\\&\quad-\sum_n w_n\mathbf{y}_n\,.\,(\mathbf{U}\mathbf{x}_n)\\&=\tfrac{1}{2}\sum_n w_n(\mathbf{x}_n^2+\mathbf{y}_n^2)-\sum_n w_n\Big[\sum_k(\mathbf{b}_k\,.\,\mathbf{y}_n)(\mathbf{x}_n\,.\,\mathbf{a}_k)\Big]\\&=\tfrac{1}{2}\sum_n w_n(\mathbf{x}_n^2+\mathbf{y}_n^2)-\sum_k\mathbf{b}_k\,.\,(\mathbf{R}\mathbf{a}_k)\\&=\tfrac{1}{2}\sum_n w_n(\mathbf{x}_n^2+\mathbf{y}_n^2)-\sum_k\sigma_k\sqrt{\mu_k}.\end{aligned} \tag{9}$$

The maximum of E is obtained if all σ_k are -1. The minimum of E is obtained if all σ_k are $+1$, which agrees with the result of Kabsch (1976).

It has also been shown in Kabsch (1976) that $\mathbf{S}+\mathbf{L}$ must be positive definite at the minimum of E. Hence, from (2) the determinants of the two matrices, $\mathbf{U}$ and $\mathbf{R}$, must have the same signs.

In the case that $\det(\mathbf{R})>0$, the orthogonal matrix $\mathbf{U}$ corresponding to the minimum of E will be a proper rotation. In the case that $\det(\mathbf{R})<0$, an improper rotation will be obtained at the minimum of E (Nyburg & Yuen, 1977). From (9), the smallest residual E corresponding to a best true rotation is then obtained if $\sigma_1=\sigma_2=+1$ and $\sigma_3=-1$ assuming that μ_3 is the smallest eigenvalue of $\tilde{\mathbf{R}}\mathbf{R}$ (three-dimensional vector space). Note that if the smallest eigenvalue is degenerate a best rotation cannot be determined uniquely in the case $\det(\mathbf{R})<0$.

Finally, it might be worth mentioning that this procedure can be generalized to find a best unitary matrix to relate two sets of vectors in the complex finite-dimensional vector space.

Summarizing the above results, the following procedure for obtaining a best proper rotation in a three-dimensional vector space is suggested.

(*a*) Remove any translation between the two given vector sets $\mathbf{x}_n$, $\mathbf{y}_n$ and determine $E_0=\frac{1}{2}\sum_n w_n(\mathbf{x}_n^2+\mathbf{y}_n^2)$ and $\mathbf{R}$.

(*b*) Form $\tilde{\mathbf{R}}\mathbf{R}$, determine eigenvalues μ_k and the mutually orthogonal eigenvectors $\mathbf{a}_k$ and sort so that $\mu_1\ge\mu_2\ge\mu_3$. Set $\mathbf{a}_3=\mathbf{a}_1\times\mathbf{a}_2$ to be sure to have a right-handed system.

(*c*) Determine $\mathbf{R}\mathbf{a}_k$ ($k=1,2,3$), normalize the first two vectors to obtain $\mathbf{b}_1$, $\mathbf{b}_2$ and set $\mathbf{b}_3=\mathbf{b}_1\times\mathbf{b}_2$. This will also take care of the case $\mu_2>\mu_3=0$.

(*d*) Form $\mathbf{U}$ according to (7) to obtain the best rotation. Set $\sigma_3=-1$ if $\mathbf{b}_3.(\mathbf{R}\mathbf{a}_3)<0$, otherwise $\sigma_3=+1$. The residual error is then $E=E_0-\sqrt{\mu_1}-\sqrt{\mu_2}-\sigma_3\sqrt{\mu_3}$.

References

KABSCH, W. (1976). *Acta Cryst.* A**32**, 922–923.
KABSCH, W. (1977). *J. Appl. Cryst.* **10**, 426–429.
KABSCH, W., GAST, W. H., SCHULZ, G. E. & LEBERMAN, R. (1977). *J. Mol. Biol.* **117**, 999–1012.
NYBURG, S. C. & YUEN, P. S. (1977). Private communication.

Kabsch '76: A solution for the best rotation to relate two sets of vectors

Kabsch '78: A discussion of the solution for the best rotation to relate two sets of vectors

Recap: Full Kabsch algorithm for finding pose via homography

1. Find H up to a scale factor from the point coorrespondences

2. Compute $H' = K^{-1}H$. Let H''s columns be $(h_1' \;\; h_2' \;\; h_3')$

3. Minimize

$$\| (h_1' \;\; h_2' \;\; h_3') - \lambda \;\; (r_1 \;\; r_2 \;\; T) \;\; \|_F$$

w.r.t. $\lambda \in \mathbb{R}, r_1, r_2, T \in \mathbb{R}^3$
s.t. $r_1^T r_2 = 0$ and $\|r_1\| = \|r_2\| = 1$

Let

$$(h_1' \;\; h_2') = U_{3x2} \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix} V_{2x2}^T.$$

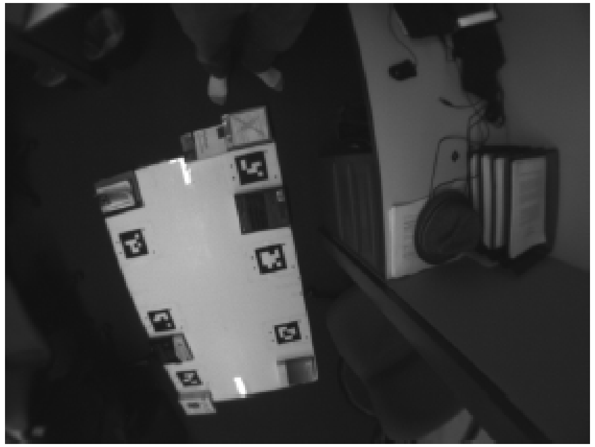
Then

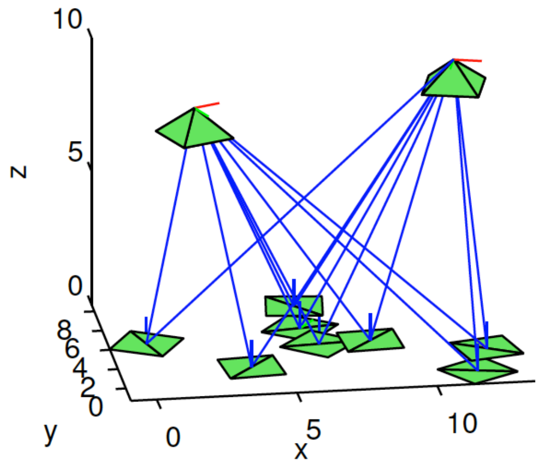
$$\begin{pmatrix} r_1 & r_2 \end{pmatrix} = U_{3x2} V_{2x2}^T \quad \text{and} \quad \lambda = \frac{s_1 + s_2}{2}$$

Alternative to running Kabsch including the 3rd column $c = h_1' \times h_2'$ as on last slide

4. $T = c/\lambda$ and $R = \begin{pmatrix} r_1 & r_2 & r_1 \times r_2 \end{pmatrix}$. Scale R to have determinant 1 if needed.

So now, we finally have valid camera pose/extrinsics!





Recap: Pose From Homography

Computing the homography can tell us how the camera (and therefore, e.g. a robot attached to the camera) is oriented w.r.t. to a world plane! (assuming known K)

Q: Where do you get $\boldsymbol{r}_3$ from though? A: $\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2$

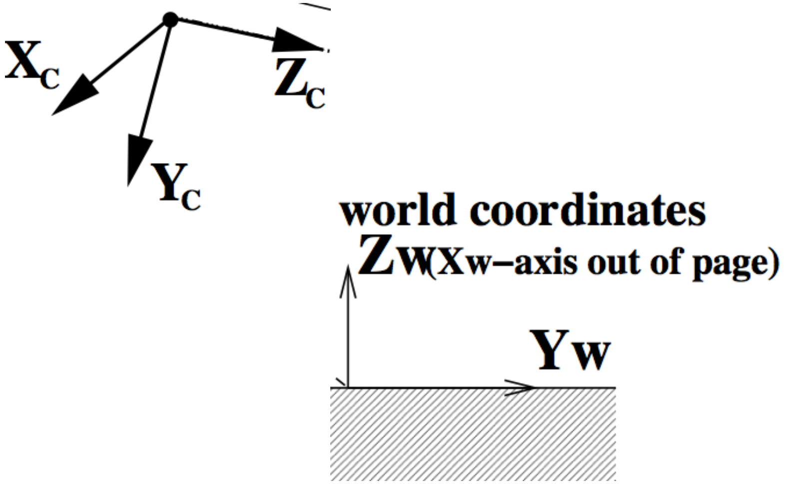
Recall the projection from world to camera

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim K \begin{pmatrix} r_1 & r_2 & r_3 & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix}$$

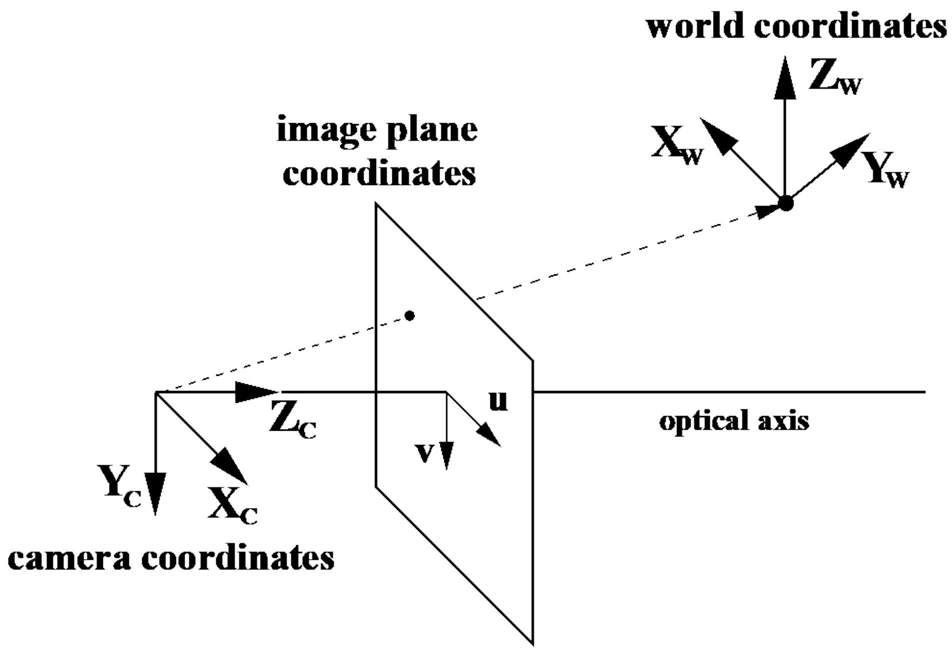
and assume that all points in the world lie in the ground plane $Z = 0$.

Then the transformation reads

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim \underbrace{K \begin{pmatrix} r_1 & r_2 & T \end{pmatrix}}_{\substack{\text{The planar homography} \\ H:\, \mathbb{P}^2 \to \mathbb{P}^2}} \begin{pmatrix} X \\ Y \\ W \end{pmatrix}$$



Recap: From world to camera: Euclidean transformation



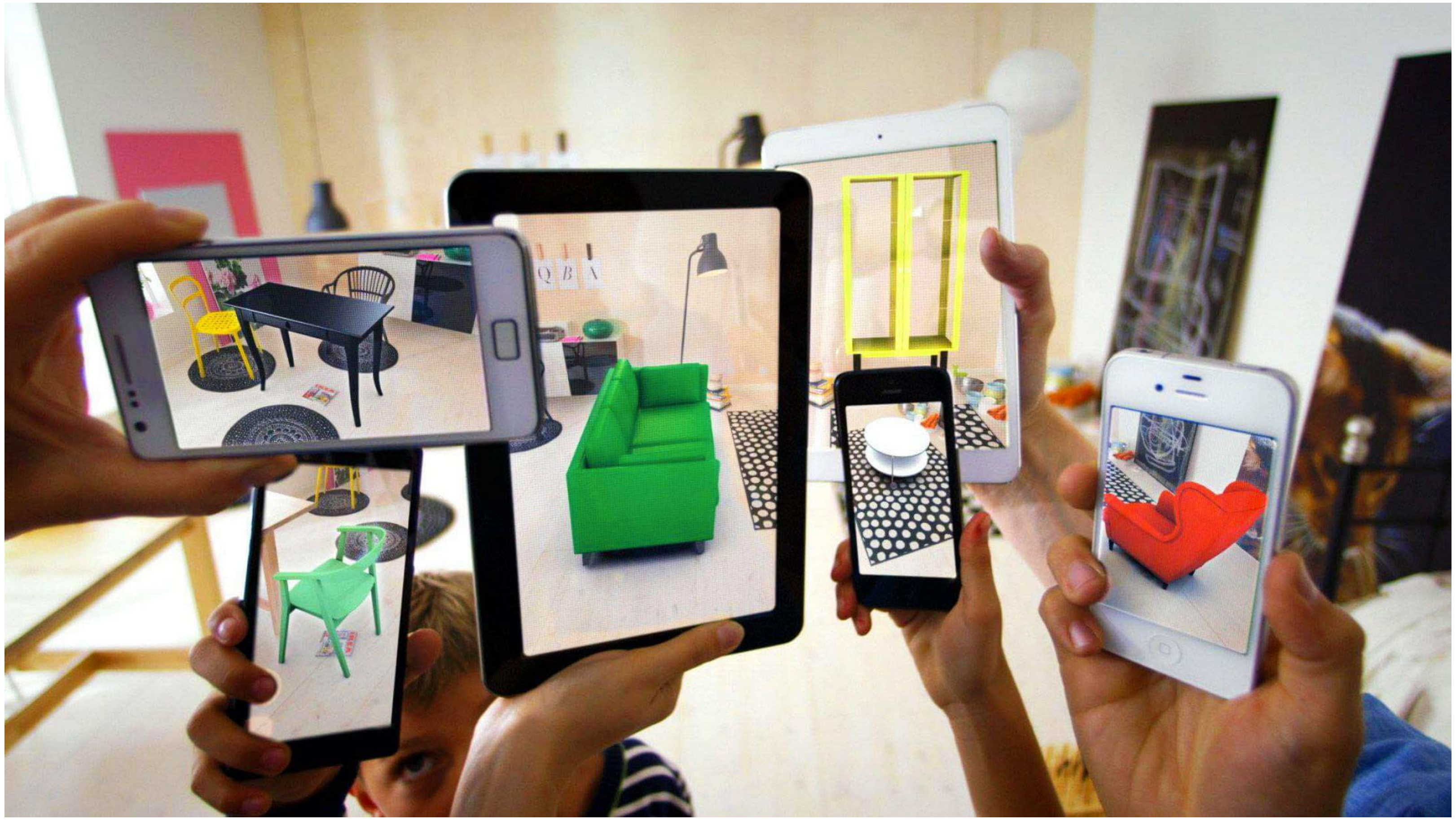
What do R and t mean exactly?
R denotes the rotation of the world axes w.r.t. camera axes
= *inverse* rotation of the camera axes w.r.t. world axes.

What about $\boldsymbol{t}$?
If $\boldsymbol{a}$ were the translation of the world origin from the camera origin, then $R(x + a)$ would be the camera coords of a world point x. i.e. $Rx + Ra$. So $\boldsymbol{t}$ here is actually R times the translation of world origin from camera origin.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R_{3\times3} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + t = \begin{bmatrix} R_{3\times3} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix} \boldsymbol{X_w}$$

Applying extrinsics (and intrinsics) for 3D shape projection

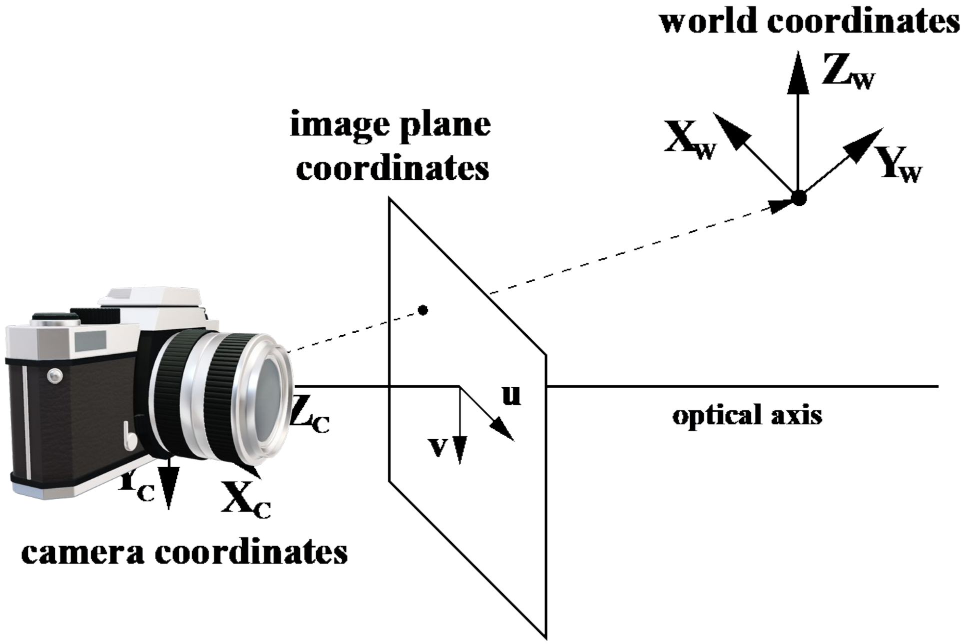
Can do AR-style projection of a 3D object onto the world plane once the full extrinsics and intrinsics are known! **You will do this in HW2.**



IKEA App, image from WIRED.

Application of pose: projecting a solid shape into the world

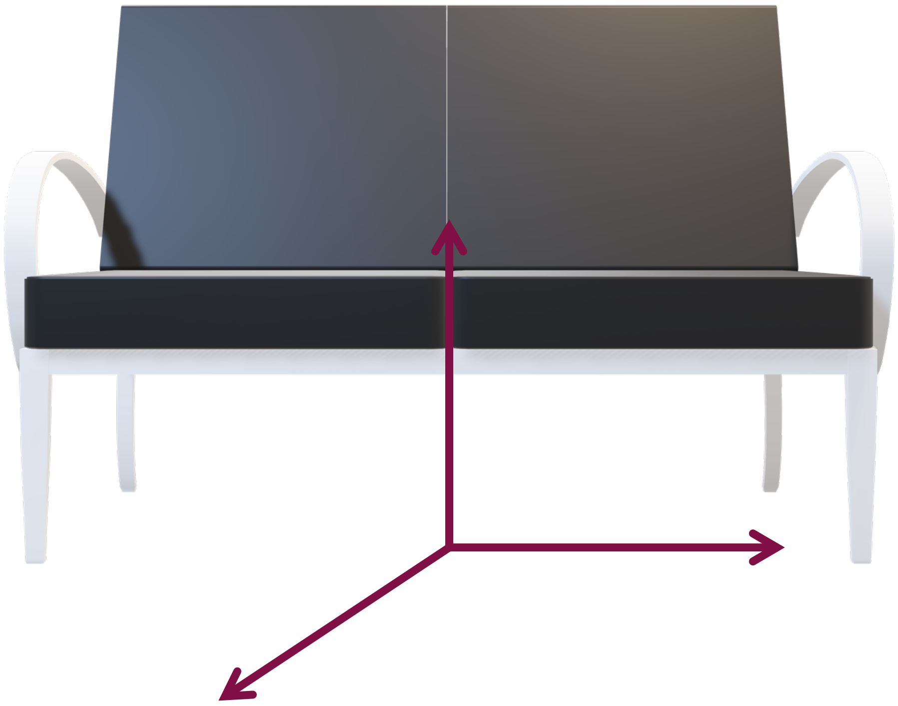
- Our normal projection equations tell us how world points in world coordinates project onto a camera, given camera pose (R, T) and intrinsics K



$$\boldsymbol{x} \sim K[R|\boldsymbol{t}]\boldsymbol{X_w}$$

Application of pose: projecting a solid shape into the world

- Suppose the shape is expressed by the positions of points X_s in a "shape-coordinate system"

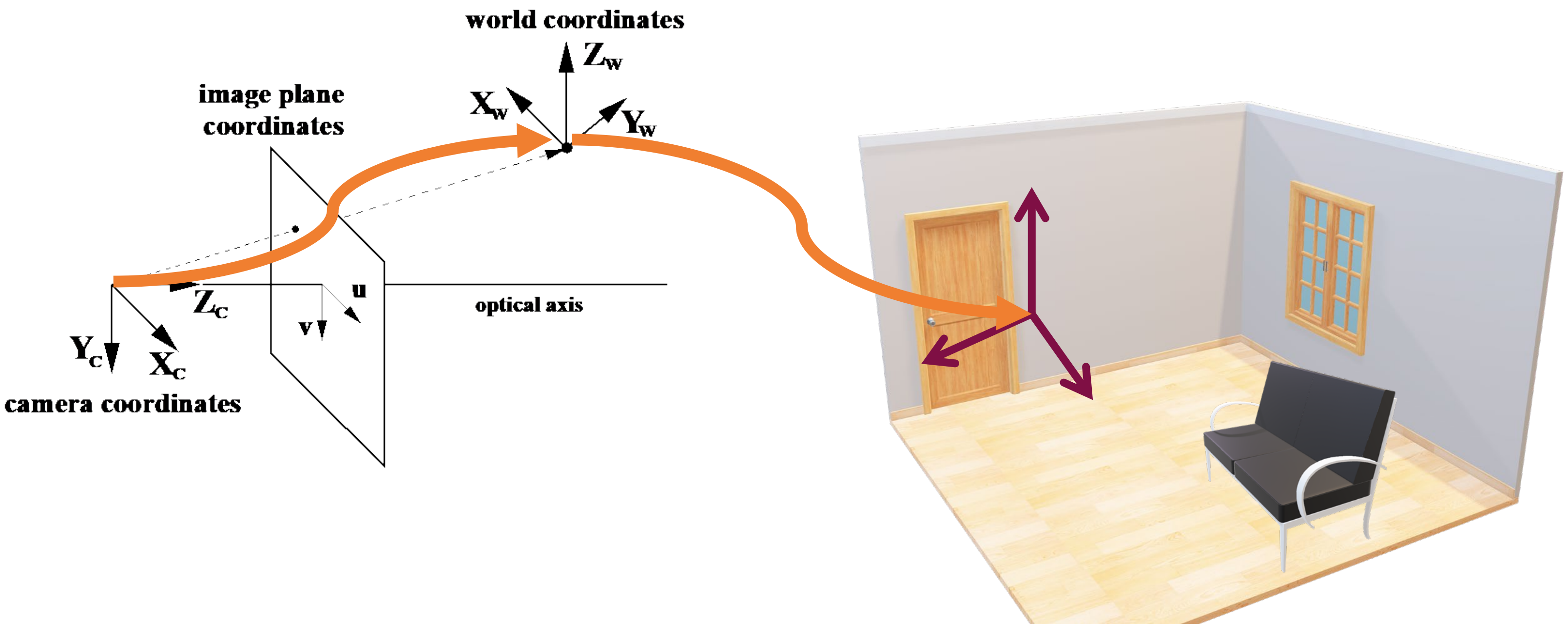


Coordinate system attached to the object

Application of pose: projecting a solid shape into the world

- First find R_{sw}, t_{sw} that convert object-centric coordinates X_s into world-centric coordinates $\boldsymbol{X}_w = R_{sw}\boldsymbol{X}_s + \boldsymbol{t}_{sw}$ to place the object at the right place in the world. (Think: what do R_{sw} and t_{sw} mean exactly?)

- Then just render the object points at $K[R|\boldsymbol{t}]\boldsymbol{X}_w$



Pose from Point Correspondences, the Perspective N Point Problem (PnP)

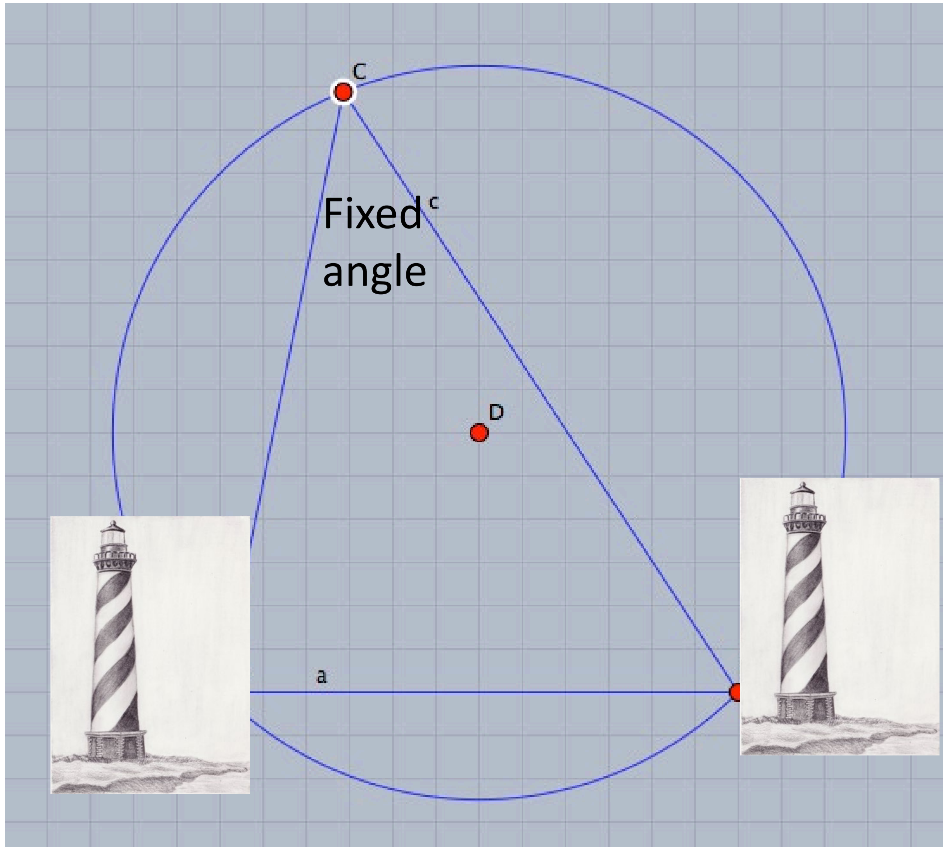
Localization by observing known 3D points from the world?

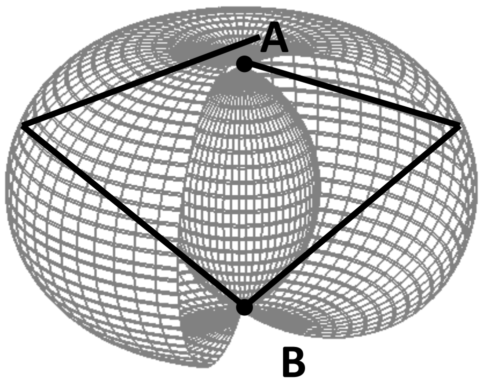
A real problem for autonomous cars, for example!
GPS: ~ a few feet accuracy. Just not good enough.

Instead, autonomous cars rely on 3D maps of the world to localize!

Navigation with "bearings" from 2 points

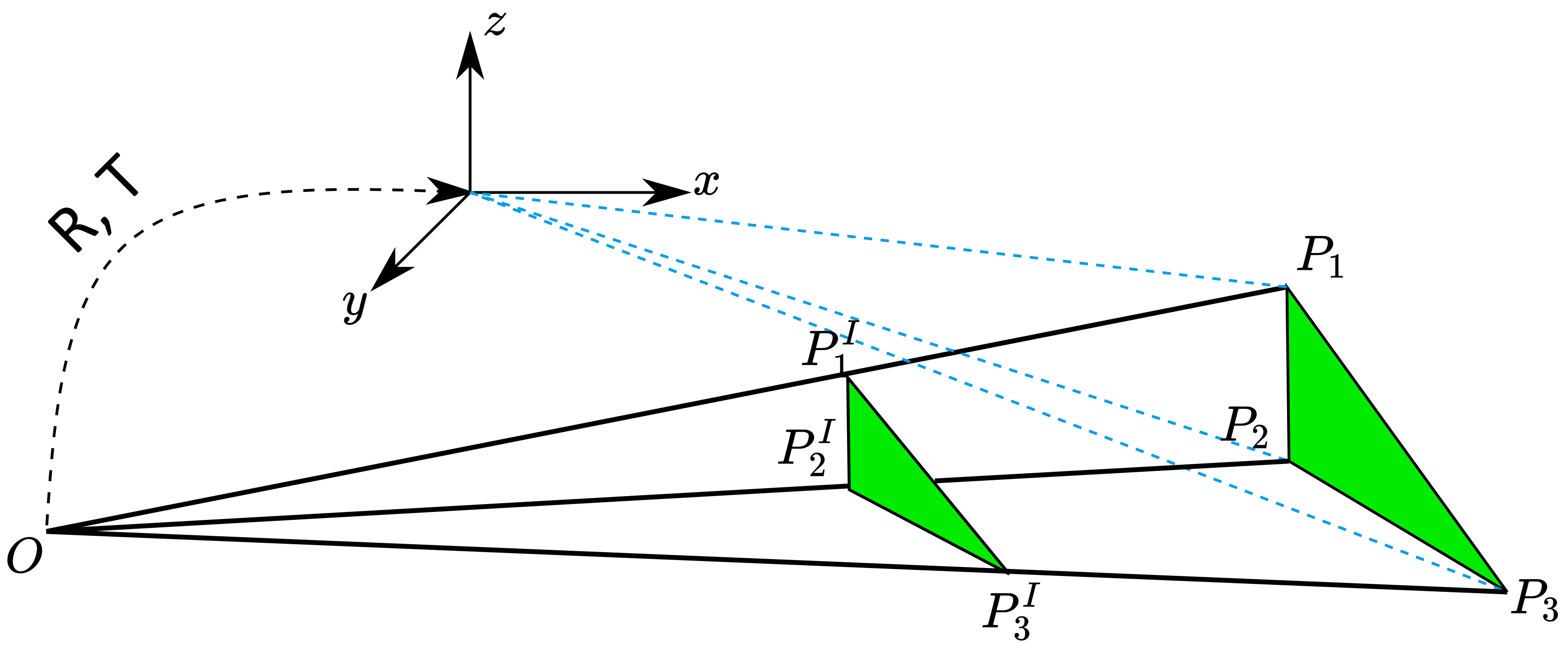
If I observe two lighthouses being some fixed angle θ apart, where am I?





Two points are not sufficient.
Camera could be anywhere on an upper semicircle in 2D.
Worse in 3D: anywhere on a toroid in 3D. Need more points!

The *Perspective* 3-Point Problem



- Given the point correspondences, find camera pose R, T

What are the differences from 4-Point Algorithm?

Jingnan Shi, https://jingnanshi.com/blog/pnp_minimal.html

Simplified 3-Point Problem w. 3D Camera Coordinates

A triangle's world 3D coordinates $P_i \in \mathbb{R}^3$ are known, and its camera-centric 3D coordinates $P_i^c \in \mathbb{R}^3$ are known

The 3D->3D 3-Point Problem: Find camera pose R, T such that

$$P_1^c = RP_1 + T$$
$$P_2^c = RP_2 + T$$
$$P_3^c = RP_3 + T$$

Full camera coordinates may come from depth cameras, but otherwise, we only have pixel coordinates.

Plan: starting from only pixel coordinates, first reduce the problem to 3D->3D.