Algorithms for Augmented Reality

3D Pose Estimation

by

Sebastian Grembowietz

2004-11-19

index

- introduction
- 3d pose estimation
  - techniques in general
  - how many points?
  - different models of projection
    - orthographic projection
    - [weak] perspective projection
  - two algorithms in detail:
    - POSIT (DeMenthon & Davis 1994)
    - Linear Algorithms (Quan & Lan 1999)

my task

- what is 3d pose estimation?
  - pose = position and orientation of an object (6DOF)
  - pose estimation = getting the pose of an object from a 2d image (e.g. from a CCD)
- and what is it good for?
  - mixing reality and virtual reality
- some examples?

  → videos

my task

- given
  - calibrated camera
  - model with feature-points
  - corresponding points on the screen (image-plane)
- wanted
  - rotation and translation of the model

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} =$$

techniques

- algebraic algorithms
  - positive aspects:
    - speed
  - negative aspects:
    - poor noise filtering
    - numeric instabilities
- Linear 4-Point Algorithm (Quan & Lan)

techniques

- optimizing (iterizing) algorithms
  - positive aspects:
    - numericly stable
  - negative aspects:
    - result depends on initial guess
    - divergence
- Levenberg-Marquardt (next Hauptseminar)

techniques

- hybrid algorithms
  - try to combine positive aspects of both algebraic algorithms and optimizing algorithms:
    - numeric stability
    - robust algorithm, i.e. noise does not harm much
    - speed
- POSIT (DeMenthon & Davis)

features used

- points
  - *Quan & Lan* (Linear N-Point Camera Pose Determination, **[QuanLan1999]**)
  - *DeMenthon & Davis* (Model-Based Object Pose in 25 Lines of Code **[DeMenthonDavis1994]**)
  - *Lowe* (Fitting Parameterized Three-Dimensional Models to Images **[Lowe1991]**)
  - *Yuan* (A General Phogrammetric Solution for the Determining Object Position and Orientation **[Yuan1989]**)

features used

- lines
  - *Dhome et al* (Determination of the Attitude of 3D Objects from Single Perspective View **[Dhome1989]**)
  - *Lowe* (Perceptual Organization and Visual Recognition **[Lowe1985]**)
- surface
  - *Rosenhahn et al.* (Pose Estimation of 3D Free-form Contours **[Rosenhahn2002]**)
  - *Nevatia & Ulupinar* (Perception of 3-D Surfaces from 2-D Contours **[NevatiaUlupinar1993]**)

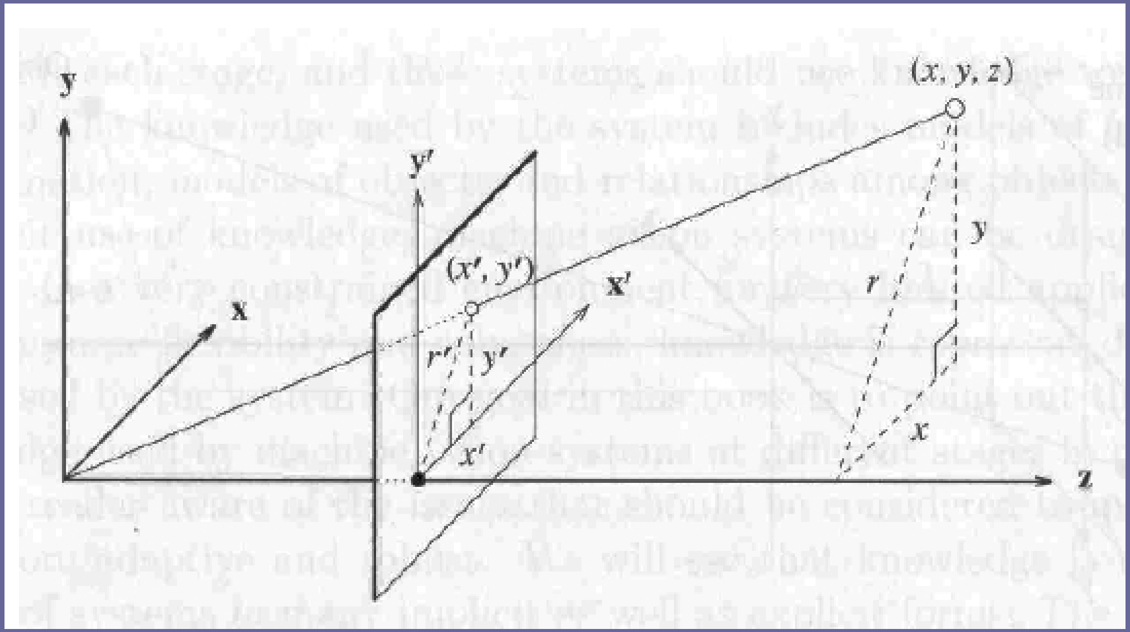
perspective

- perspective projection

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ fZ \\ Z \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ f \\ 1 \end{bmatrix}$$

f - focal length

  - vanishing points
  - vanishing lines (horizon)

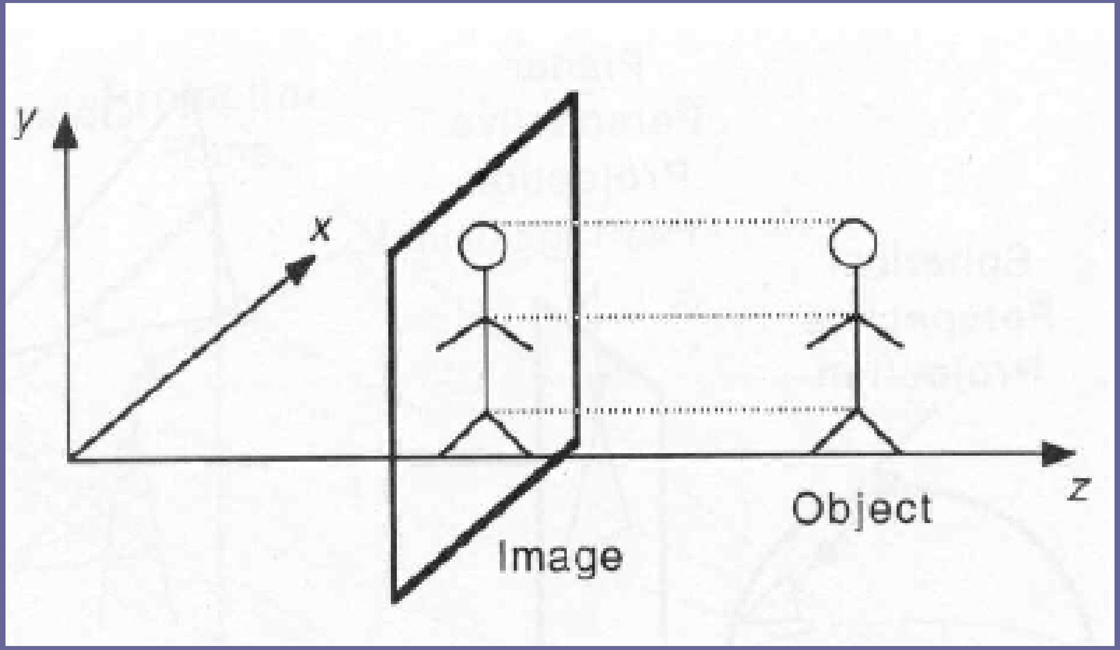


orthographic

- orthographic projection

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & f \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ f \\ 1 \end{bmatrix}$$

  - “just drop Z”
  - size preserves
  - angles preserve



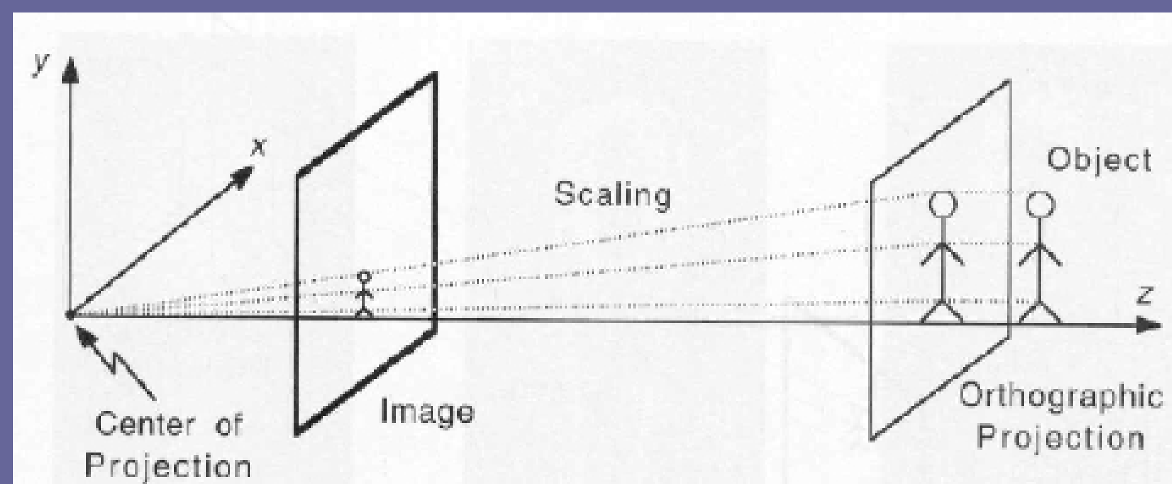
weak perspective

- weak perspective projection

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & f \; k \\ 0 & 0 & 0 & k \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ f \; k \\ k \end{bmatrix}$$

k - constant

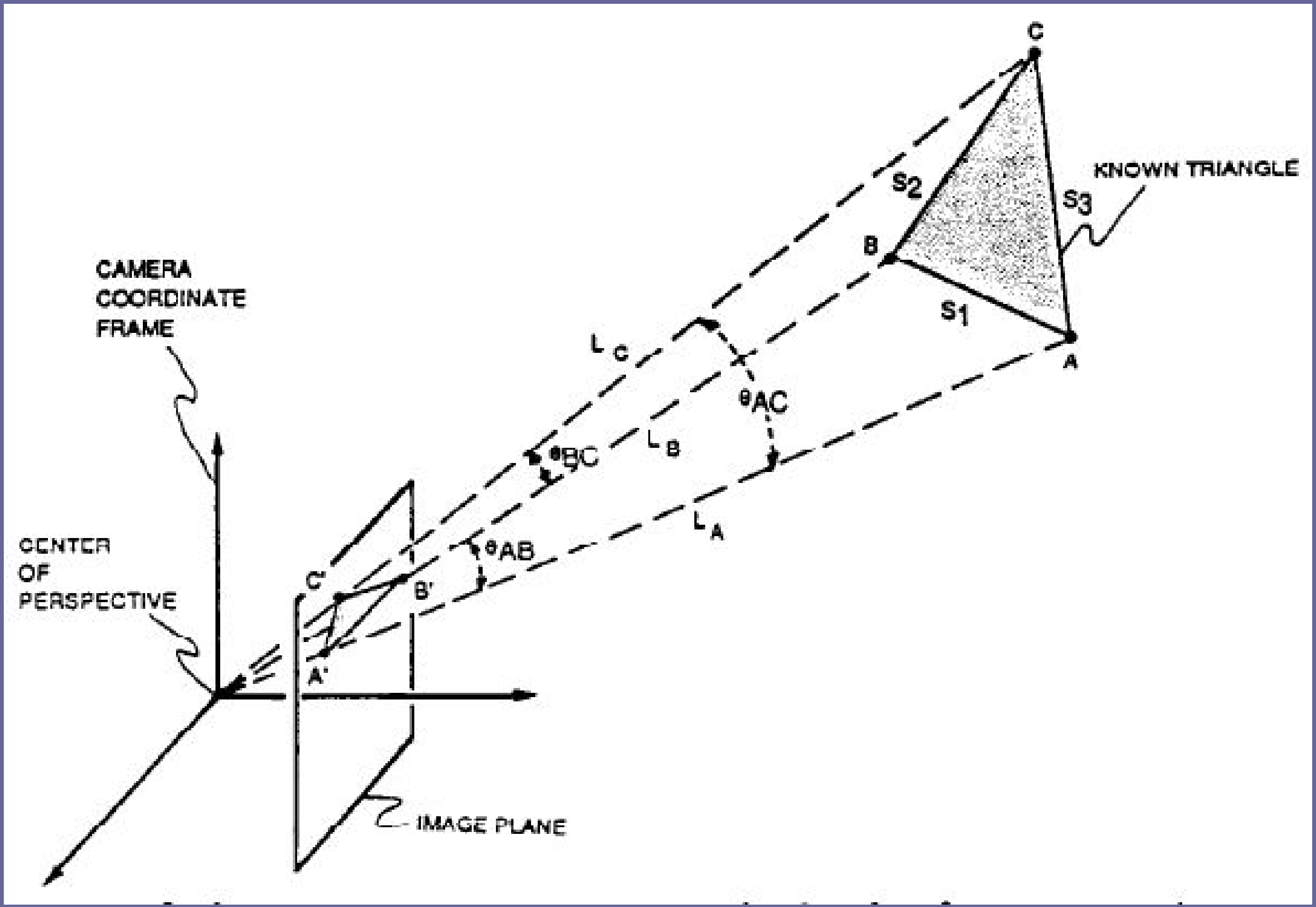
  - fast
  - approximation of perspective projection



3d Pose Estimation

- discovered 1795 by Lacroix (inspired by Monge [Smith2003])
- solved first algebraicly by Grunert in 1841 [Haralick1991]
- Fischer and Bolles coined the term "perspective three-point problem" in 1981 [FB1981]

- are three points enough? 6DOF from 3 Points?

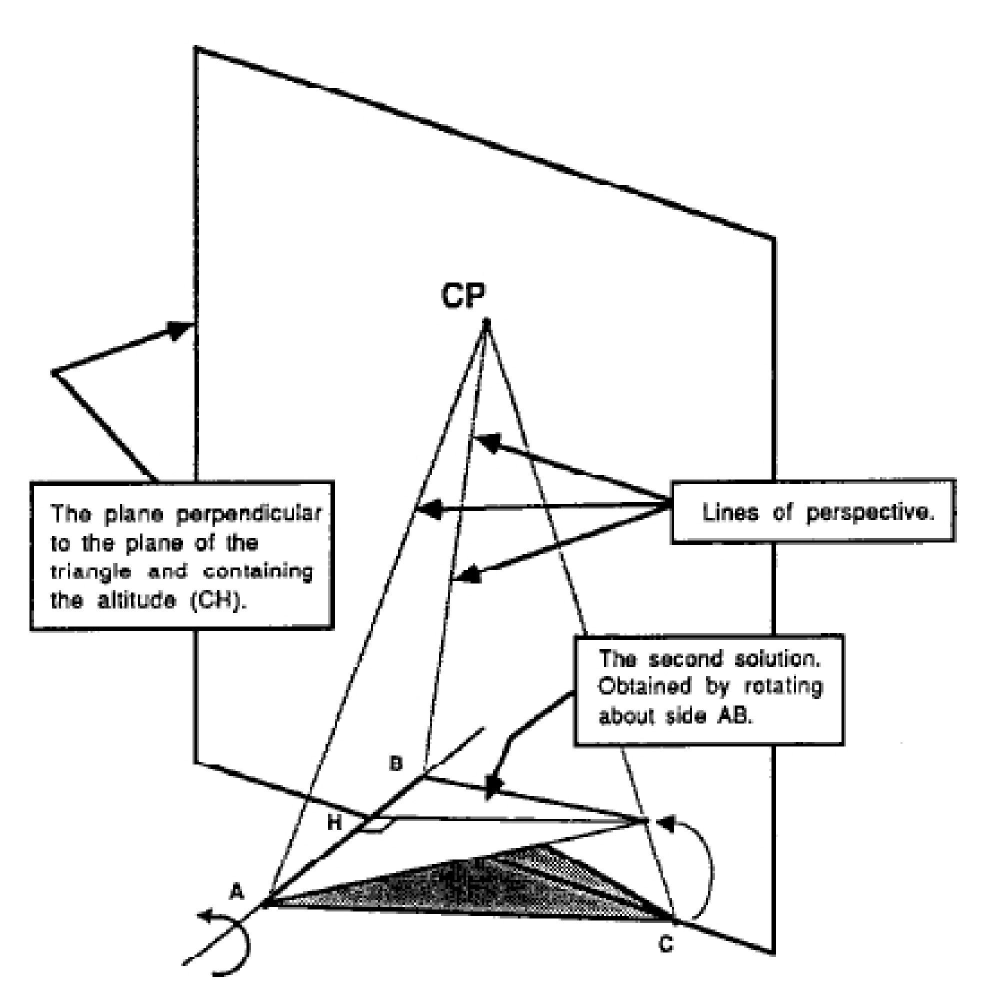


- NO!

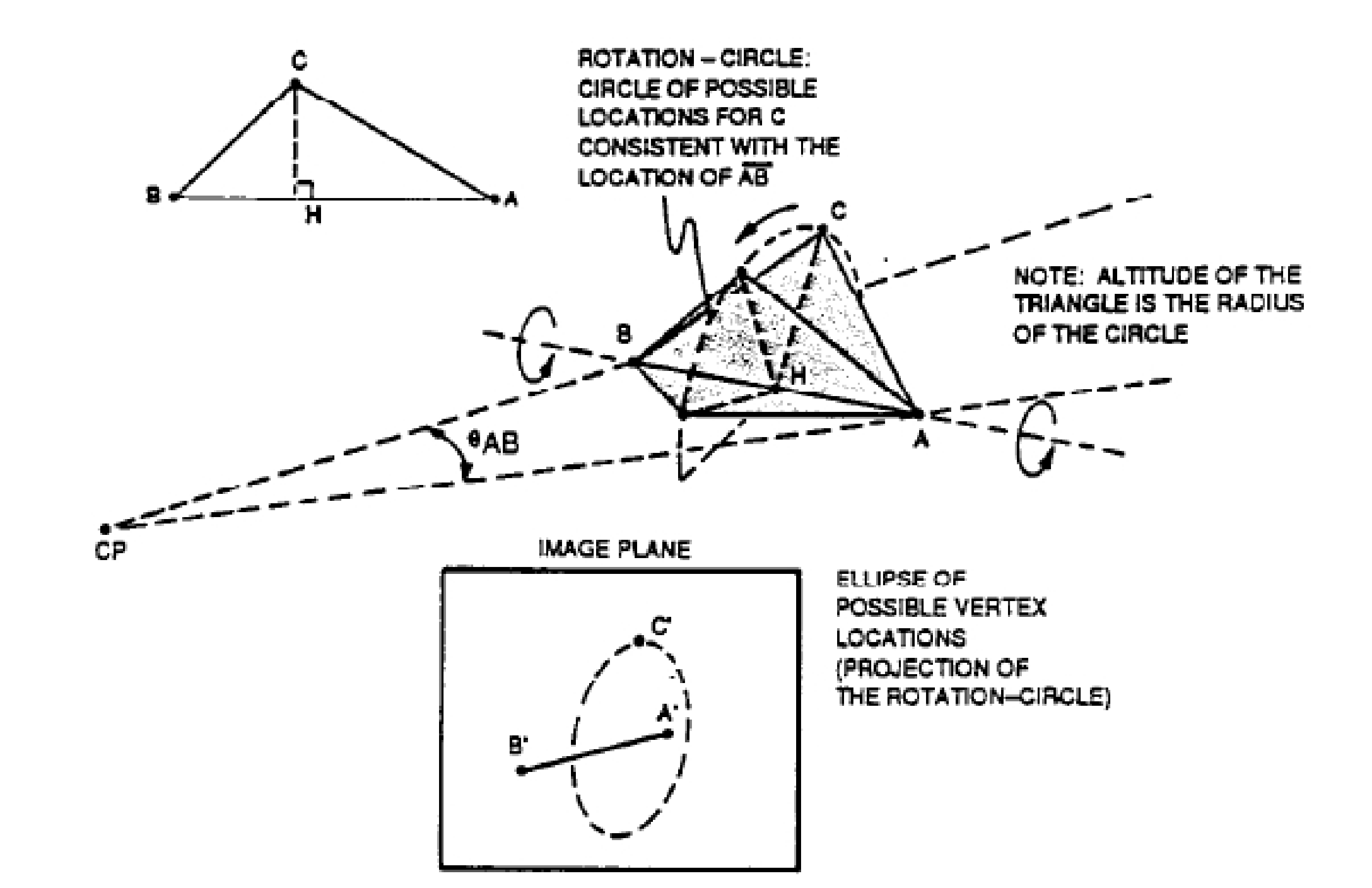
critical configurations:

- rotated leg solutions

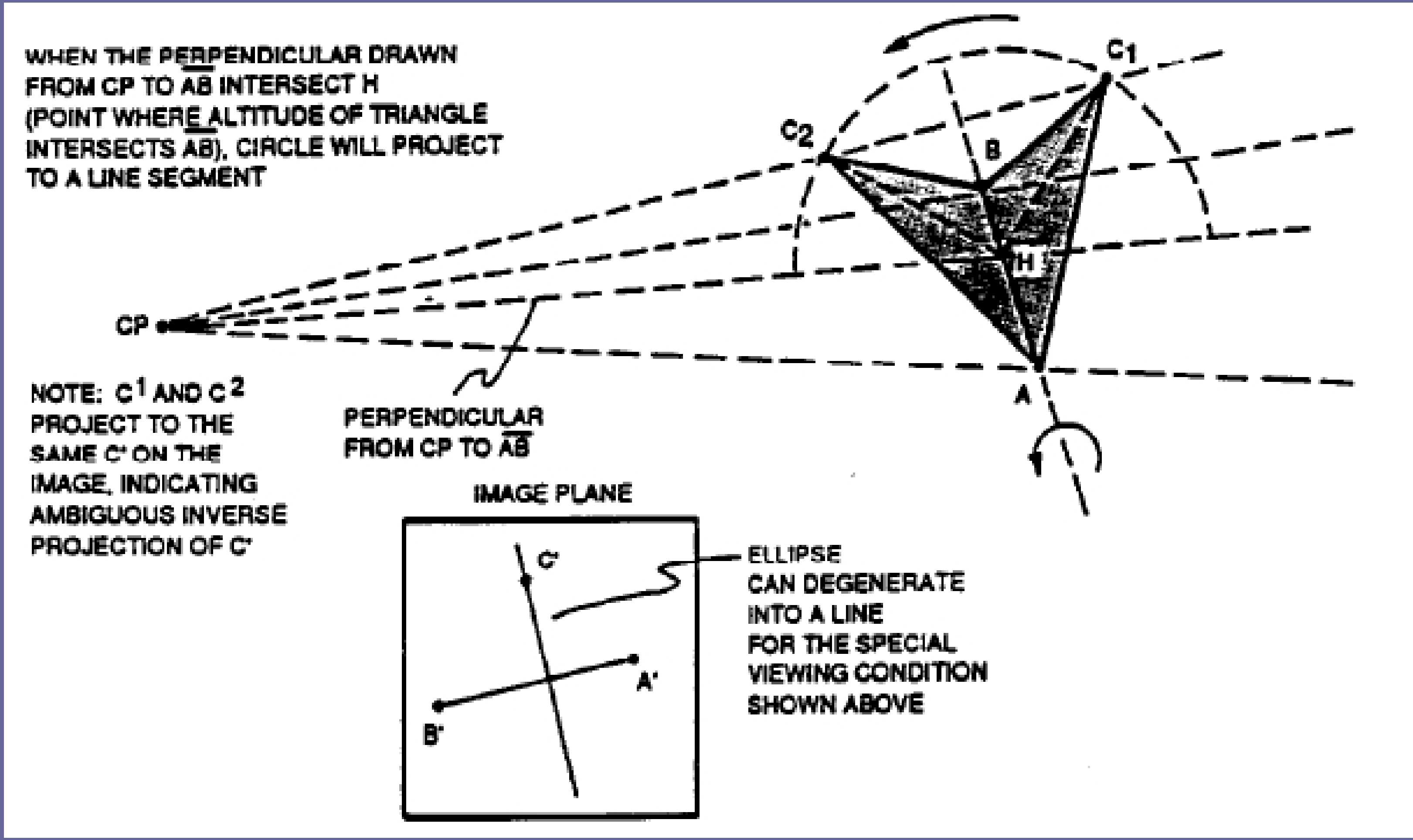
⇒ 2 solutions!



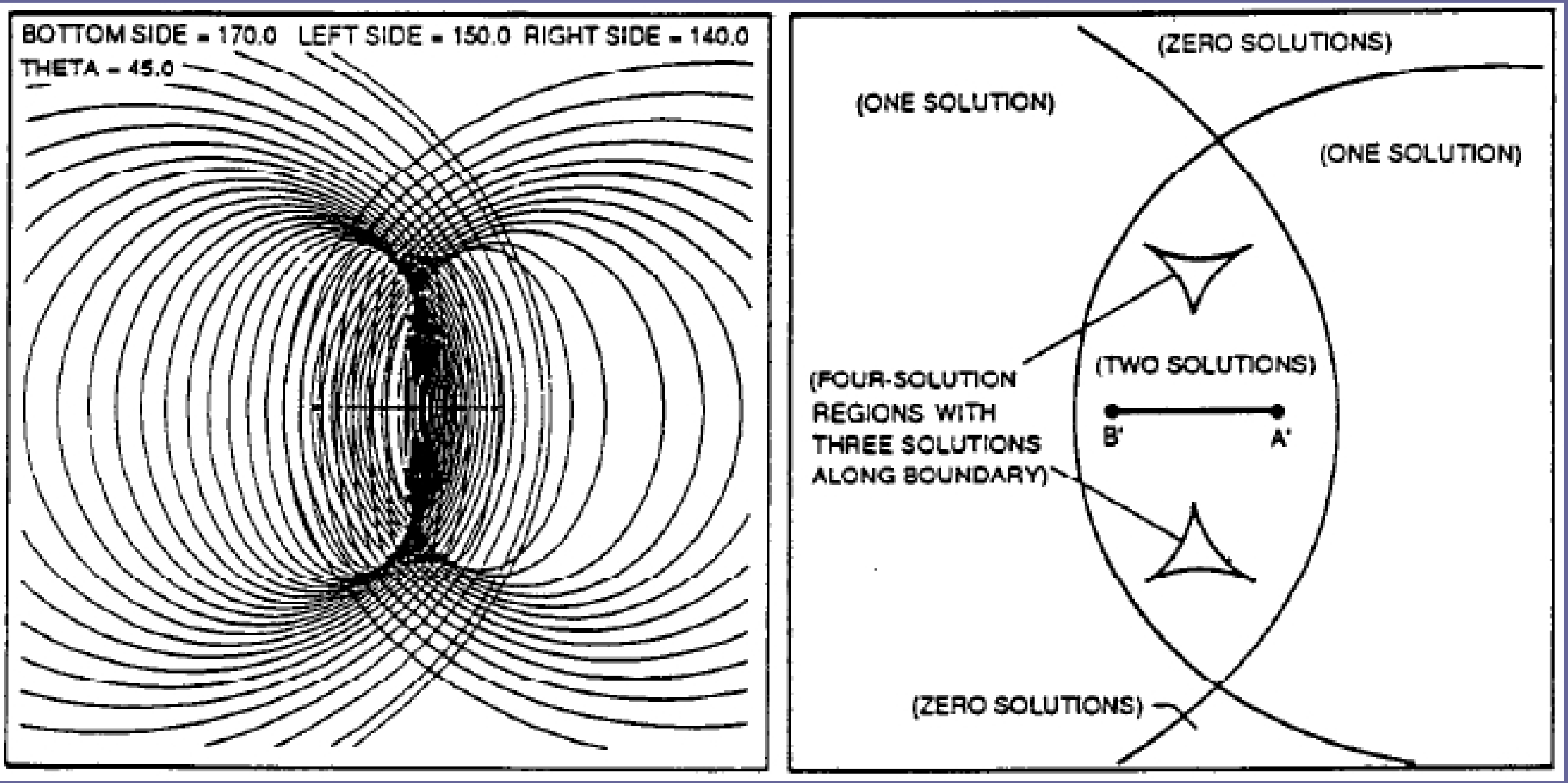
- rotation circle in general



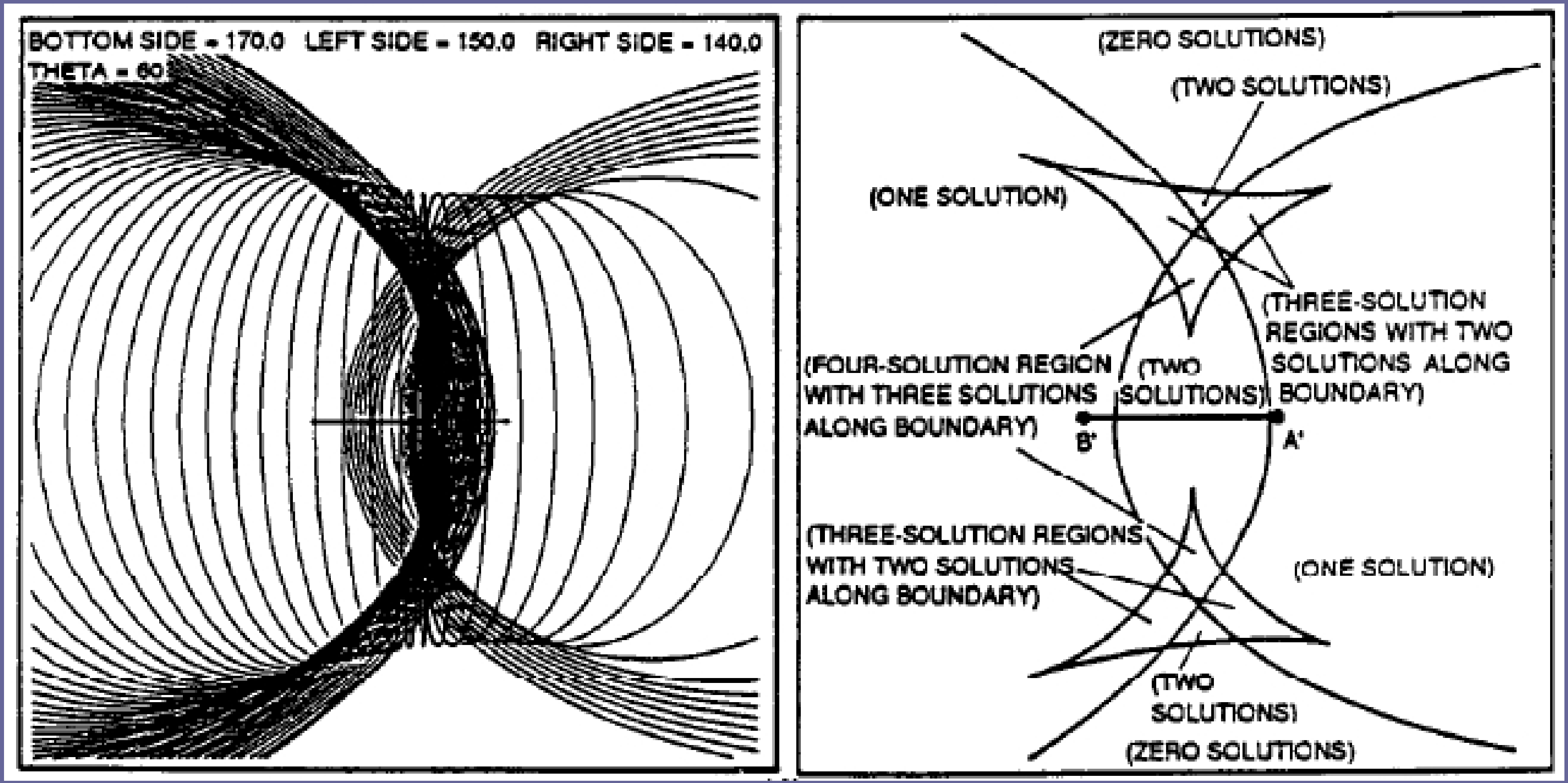
- rotation circle projected to line (2 solutions)



- swept rotation circles



- another swept rotation circle



- summary p3p:

Three points generate up to four possible solutions which can **not** be ignored in general.
[Wolfe1991]

Note: This is also called fourfold ambiguity.

3d Pose Estimation

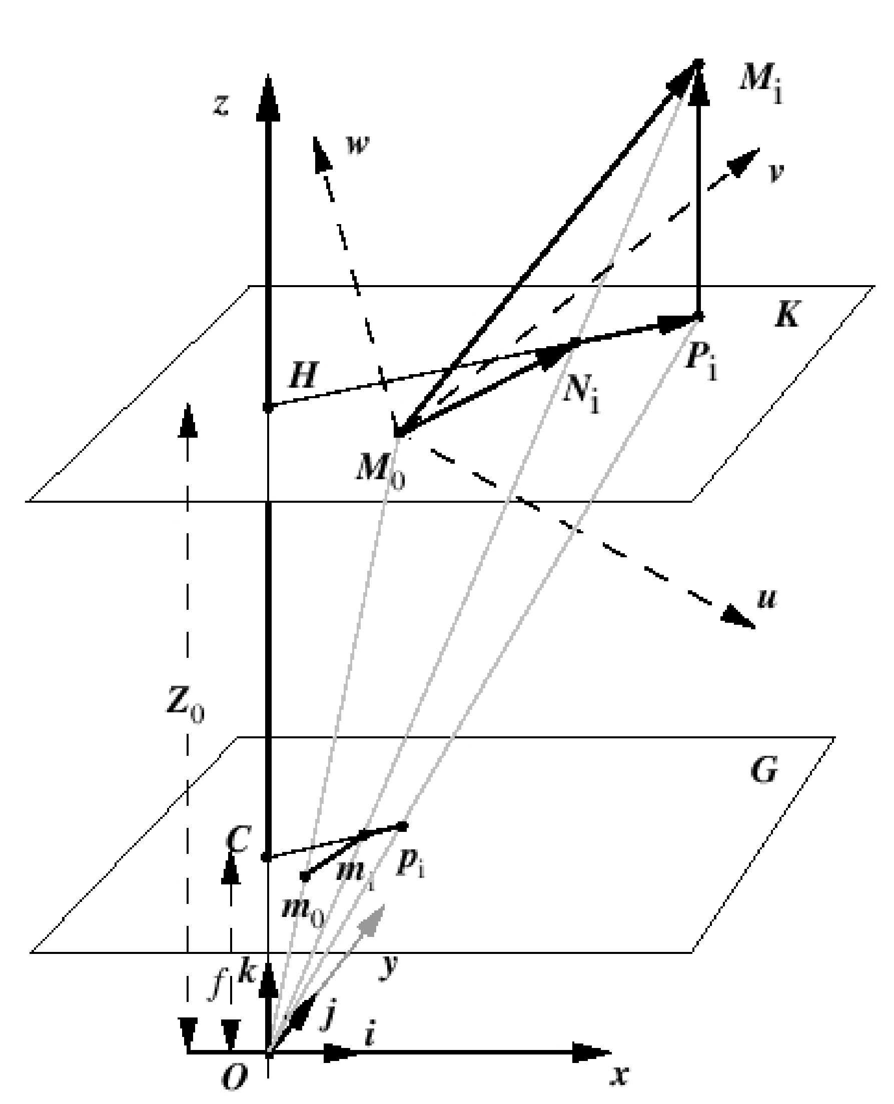
- use four or more points to determine pose
- straight-forward approach (4p):
  - extract four triangles out of the four points, this gives you 16 solutions at maximum, then merge these and you have a pose.
  - new problem: Merging results (finding the common root) can be very difficult and expensive
- other algorithms for four points:
  - POSIT (optimizing algorithm)
  - Linear Algorithms (algebraic algorithm)

POSIT

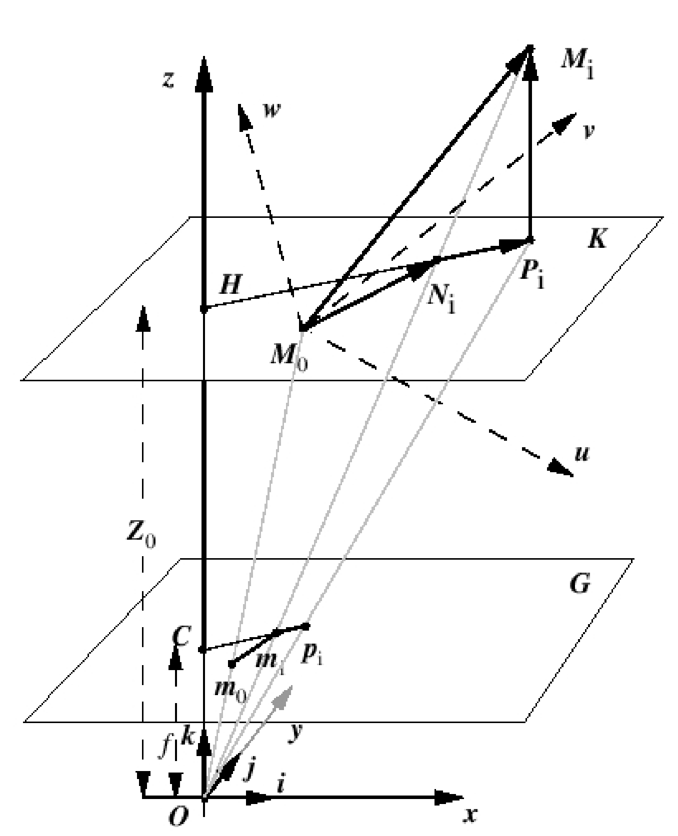
- Authors: Daniel F. DeMenthon and Larry S. Davis in 1992
- characteristics:
  - optimizing algorithm
  - uses weak perspective projection
  - does **not** require initial pose estimate
  - inexpensive in its iteration loop
  - can be written in 25 lines of code in Mathematica (as the title says)

3d Pose Estimation

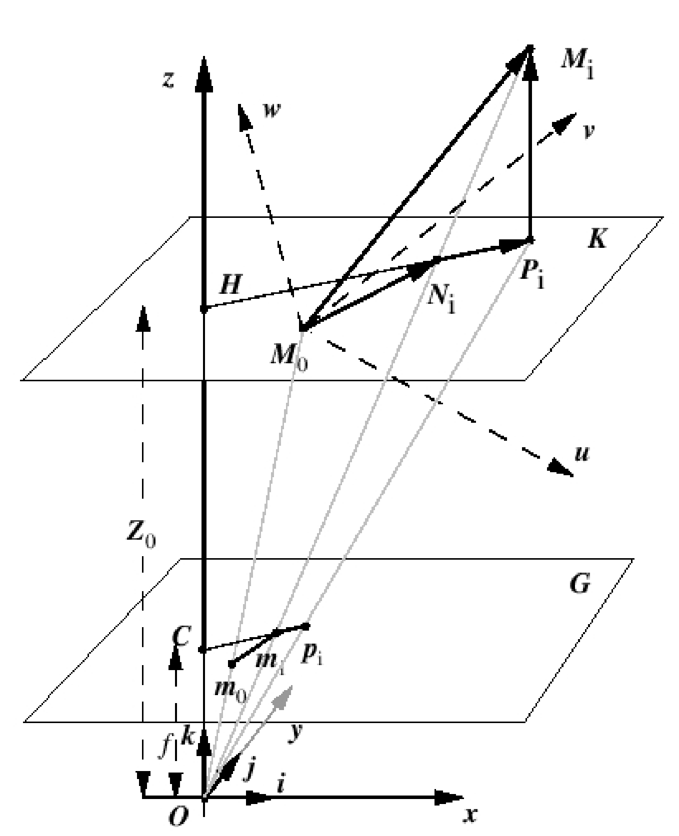
- all M_x belong to the model
- all m_x belong to the image plane G
- plane K parallel to G at distance Z_0
- M_i projects perspectively on N_i (on K) and m_i (on G)
- M_i projects orthographically on P_i
- P_i projects perspectively on p_i



- Outline of the algorithm
  - POS (pose from orthography and scaling)
    - compute an object-matrix depending on model-coordinates M_i only
      - Note: Object-Matrix will be calculated only once
    - compute a rotation-matrix from the image-points m_i
    - get translation from rotation-matrix by normalizing first or second row



- POSIT (***POS*** with ***it***erations)
  - shift the feature-points M_i^* from the object pose just found to the lines of sight
    (where they would belong if the pose was correct)
  - do a POS (starting with a new rotation matrix) on the image-points m_i^* of the shifted feature points M_i^*



- POSIT usually takes four to five iterations until it's done

- a little math...

$$R = \begin{bmatrix} i_u & i_v & i_w \\ j_u & j_v & j_w \\ k_u & k_v & k_w \end{bmatrix}$$

  - i_u, i_v, i_w are in object coordinate system (M_0u, M_0v, M_0w)
  - once i and j are computed, k is obtained by taking the cross product of i and j
  - if Z_0 (depth of M_0) is found, we can determine M_0 because of translation vector T is aligned with Om_0 (namely $T = Z_0/f * Om_0$) (f = focal length)

- more math...
  - image points of **weak** perspective projection:
    - $x_i' = f\,X_i / Z_0 \quad y_i' = f\,Y_i / Z_0$
  - image points of [strong] perspective projection:
    - $x_i = f\,X_i / Z_i \quad y_i = f\,Y_i / Z_i$
  - these can be combined to:
    - $x_i' = f\,X_0 / Z_0 + f\,(X_i - X_0) / Z_0 = x_0 + s\,(X_i - X_0)$
    - $y_i' = y_0 + s\,(Y_i - Y_0)$ *[$s = f / Z_0$]*

- ...more math...
  - now i and j from the rotation matrix and Z_0 from M_0 are combined with M_0M_i and coordinates x_i and y_i from image points m_0 to m_i
  - $M_0M_i * (f / Z_0) i = x_i (1 + \varepsilon_i) - x_0$ Equation 1
  - $M_0M_i * (f / Z_0) j = y_i (1 + \varepsilon_i) - y_0$ Equation 2
    - where $\varepsilon_i = (1 / Z_0) M_0M_i*k$

- the end is near:
  - We create linear systems out of equations 1 and 2, substitute (f / Z_0) i by I and (f / Z_0) j by J, using the feature points we have (at least four points).
  - These linear equations can be solved via singular value decomposition (SVD); we get j and i from normalizing I and J.
  - This way we also get Z_0 as f, the focal length, is known.

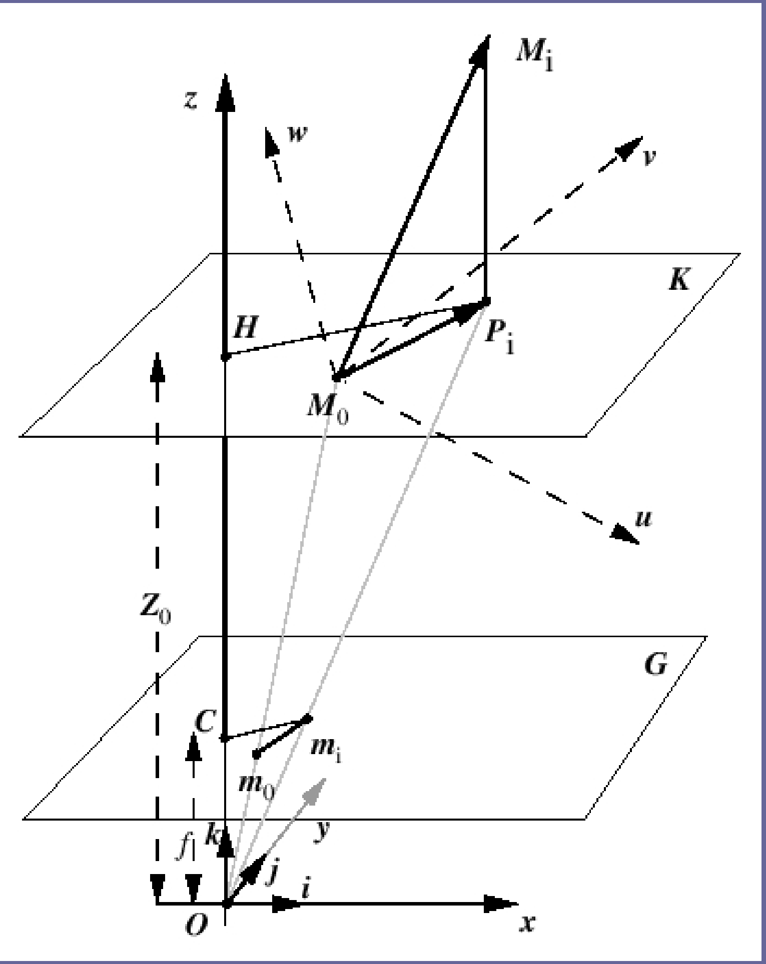
POSIT

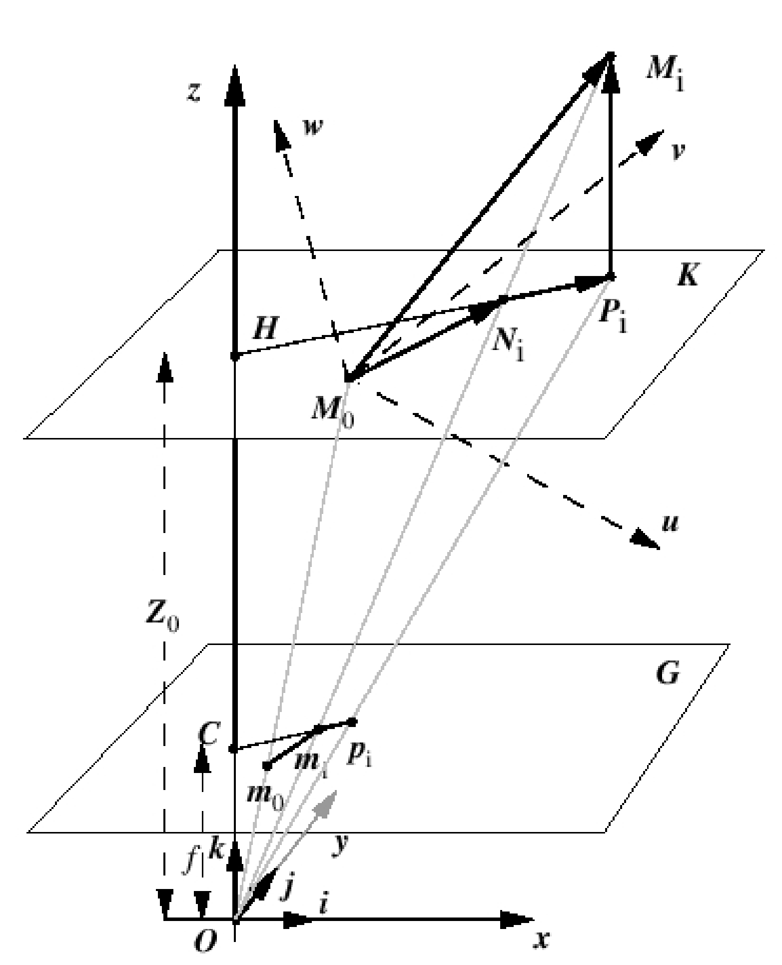
- last words to POSIT
  - we start setting $\varepsilon_i = 0$, assuming scaled orthographic image points and perspective image points coincide

    Note: when we do tracking, we best use the ε_i from the last image

  - Once we know i, j and Z_0, we can compute a better ε_i, and this leads to better values for i, j and Z_0 in the next iteration.

- Picture of situation with $\varepsilon_i = 0$ and real situation

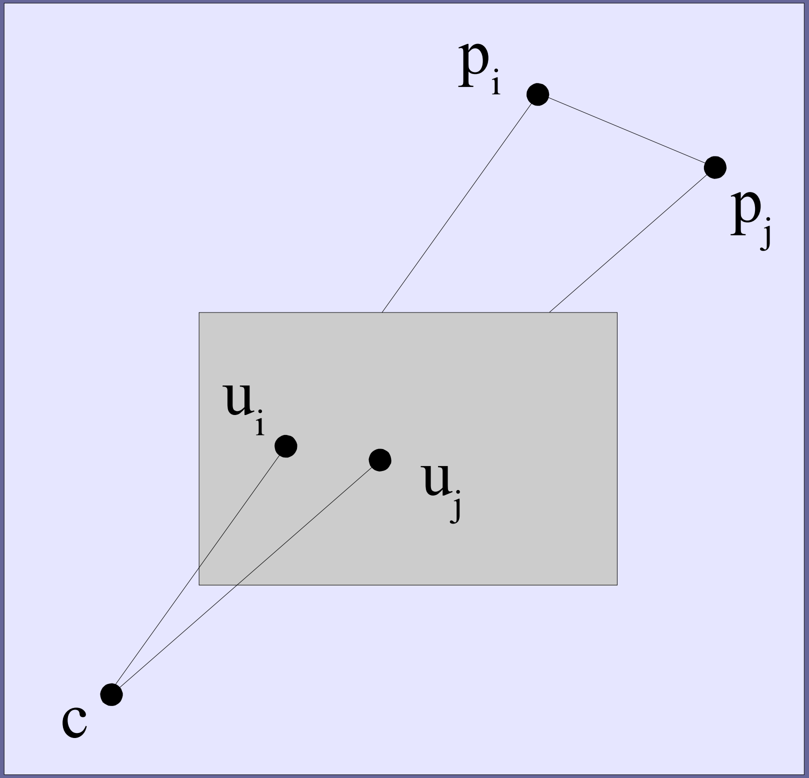




linear algorithms

- Authors:
  Long Quan and Zhongdan Lan in 1999
- characteristics:
  - family of linear algebraic algorithms
  - uses 4, 5 and up to n-points
  - uses redundancy of the fourth point to directly create a unique solution

- p3p revisited:
  - we have correspondences between u_x and p_x
  - we know the angle $u_j u_i c$
  - we know $\| p_i - p_j \| = d_{ij}$
  - the following quadratic equation is valid:



$$d_{ij}^2 = x_i^2 + x_j^2 - 2\, x_i x_j \cos(u_j u_i c)$$ Equation 1

- from equation 1 we can go to

$$f_{ij}\,(x_i, x_j) = x_i^{\,2} + x_j^{\,2} - 2\,x_i x_j \cos(u_j u_i c) - d_{ij}^{\,2}$$ Equation 2

- for n=3 points used, we get the following system:

$f_{12}(x_1, x_2) = 0$

$f_{13}(x_1, x_3) = 0$

$f_{23}(x_2, x_3) = 0$

- by elimination of first x_3 and then x_2, we get:

$$g(x) = a_5 x^4 + a_4 x^3 + a_3 x^2 + a_2 x^1 + a_1 = 0 \quad (x := x_1^{\,2})$$ Equation 3

linear algorithms

- Equation 3 gives as previously shown at most four different solutions...

- We “solve” the ambiguity by adding another point (so now we are in p4p)

- The fourth point gives us

$$g(x) = a_5 x^4 + a_4 x^3 + a_3 x^2 + a_2 x^1 + a_1 = 0$$
$$g(x) = a_{5'} x^4 + a_{4'} x^3 + a_{3'} x^2 + a_{2'} x^1 + a_{1'} = 0$$
$$g(x) = a_{5''} x^4 + a_{4''} x^3 + a_{3''} x^2 + a_{2''} x^1 + a_{1''} = 0$$

which can be written in matrix form

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_{1'} & a_{2'} & a_{3'} & a_{4'} & a_{5'} \\ a_{1''} & a_{2''} & a_{3''} & a_{4''} & a_{5''} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{bmatrix} = A_{3x5} t_5 = 0$$

Equation 4

- Quan and Lan solved equation 4 by applying SVD on it, then using a nonlinear constraint and again applying SVD
- After that, we have t_5 and can easily get the x:

  $x = t_1/t_0$ or t_2/t_1 or t_3/t_2 or t_4/t_3

  or an average of these values.
- For obtaining x_i we now just have to square-root x :

  $x_i = x^{1/2}$ (as x_i is positive this is nonambiguous)

linear algorithms

- p5+p:
  - given five or more point-correspondences, the linear algorithms introduced by Long Quan and Zhongdan Lan need only one SVD to determine x_i

P7P

- Tsai's approach
  - uses seven points
  - very useful if camera is not calibrated
  - see Hauptseminar on camera calibration

Thank you for your attention

Any questions?

resources

- **[FB1981]**: Fischer and Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Graphics and Image Processing, vol. 24, no. 6, pp.381-395, June 1981
- **[Lowe1991]**: Fitting Parameterized Three-Dimensional Models to Images, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no.5, pp129-142, May 1991
- **[QuanLan1999]**: Linear N-Point Camera Pose Determination, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 21, no. 7, July 1999
- **[DeMenthonDavis1994]**: Model-Based Object Pose in 25 Lines of Code, International Journal of Computer Vision, vol. 15, no. 1 pp. 123-141, May 1995

resources

- **[Rosenhahn2002]:** Rosenhahn, Perwass and Sommer : Pose Estimation of 3D Free-form Contours.Technical Report 0207,University Kiel. 2002
  - **more Rosenhahn papers**: *http://www.ks.informatik.uni-kiel.de/~bro/MyHomepage/publications.html*
- **[NevatiaUlupinar1993]:** Perception of 3-D Surfaces from 2-D Contours, PAMI(15), No. 1, January 1993, pp. 3-18.
  - **other publication from Nevatia et al**: *http://iris.usc.edu/Outlines/Papers.html*
- **[Dhome1989]**: Determination of the Attitude of 3D Objects from Single Perspective View, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 12, pp. 1265-1278, Decembre 1989

resources

- **[Yuan1989]**: A General Phogrammetric Solution for the Determining Object Position and Orientation, IEEE Trans. Robotics and Automation, vol. 5, no.2, pp.129-142, April 1989
- **[Lowe1985]**: Perceptual Organization and Visual Recognition, Norwell, Mass.: Kluwer Academic Publishers, 1985
- **[Haralick1991]**: Analysis and Solutions of the Three Point Perspective Pose Estimation Problem, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.592-598, Maui, Hawaii, 1991
- **[Smith2003]**: David Eugene Smith, History of Modern Mathematics, Mathematical Monographs No. 1 (taken from Project Gutenberg's History of MM, *www2.cddc.vt.edu/gutenberg/etext05/hsmmt10p.pdf*)
- **[Wolf1991]**: The Perspective View of Three Points, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 13, no. 1, January 1991