Multi-View Geometry and Computational Photography Using Non-Classical Cameras

Amit Agrawal and Srikumar Ramalingam

Mitsubishi Electric Research Labs (MERL)

Schedule

9:15 Introduction and Geometry of Rays

9:45 Modeling Non-Classical Cameras

10:30 Break

10:45 Calibration and Multi-View-Geometry

11:30 Break

11:45 Synthesis/Reconstructions

12:30 Future Directions and Discussions

Imaging Models

Single viewpoint cameras:

- Perspective projection plus radial or decentering distortion
  - imaging model well-known…
  - again, interior orientation (intrinsic parameters) allows to perform **forward projection and back-projection**
  - calibration approaches:
    - plumbline calibration: use images of straight line patterns to estimate “non-perspective” parameters
    - calibration with control points: compute all parameters of the model using bundle adjustment

Imaging Models

Single viewpoint cameras:

- Fisheyes
  - several models have been proposed (ad hoc or derived from actual lens designs)
  - e.g. equi-angular model (existence of distortion center and optical axis such that distance of image point to distortion center is proportional to angle between projection ray and optical axis)

Imaging Models

Catadioptric systems (camera + mirror):

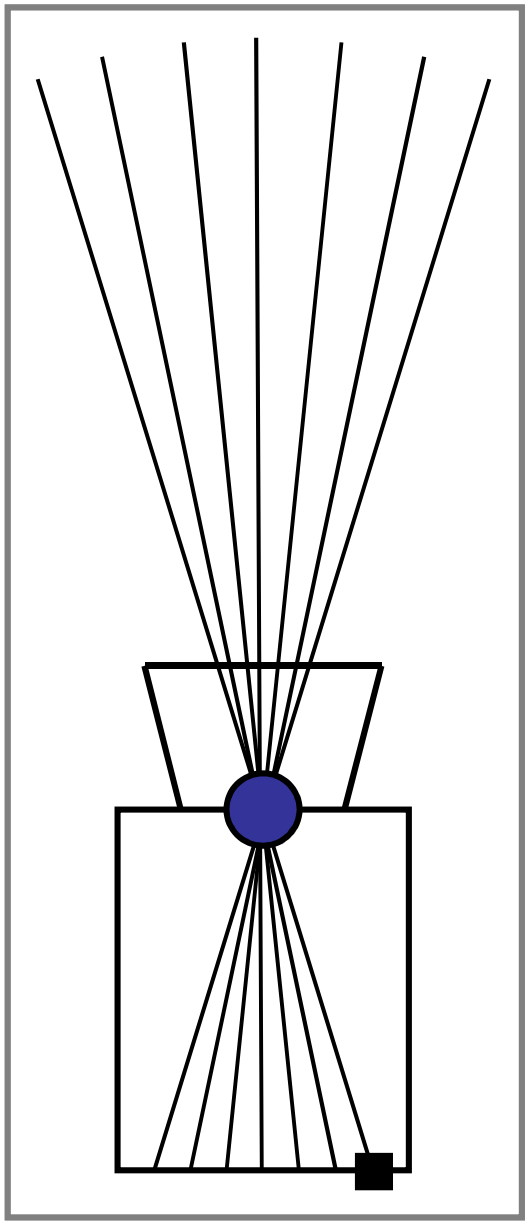
- Knowledge of mirror shape and position relative to camera, together with camera's interior orientation, allows to perform back-projection

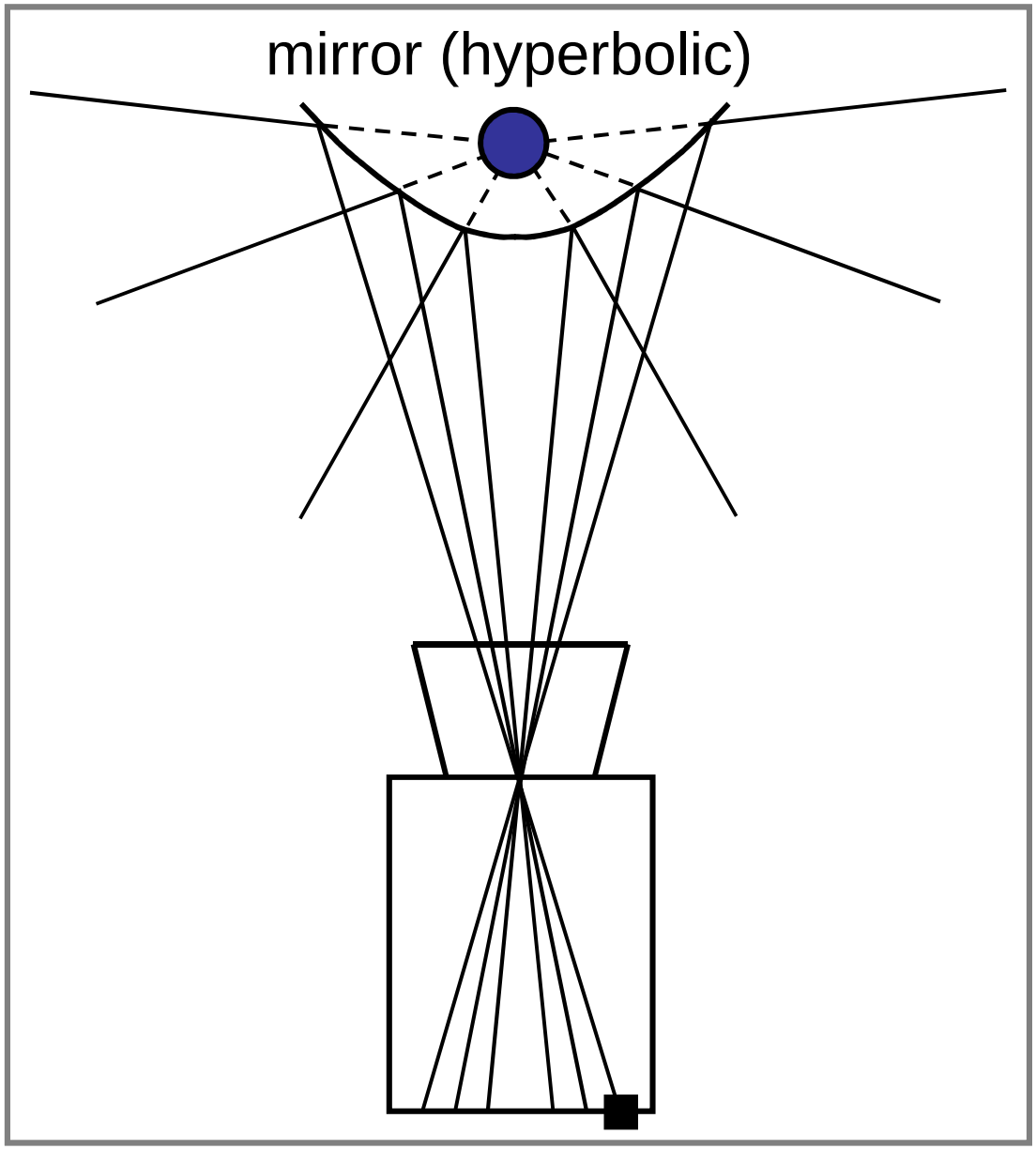
Imaging Models

Back to single viewpoint cameras:

- **Central** catadioptric systems
  - with appropriate mirror shape and position, system has a single effective viewpoint (cf. next slide)
  - practically relevant: parabolic mirror + orthographic camera, hyperbolic mirror + perspective camera
  - various imaging models have been proposed:
    - models whose parameters represent correlations between mirror shape/position and interior orientation of camera
    - unifying models for all types of central catadioptric cameras
  - calibration approaches:
    - plumbline approaches (sometimes with closed-form solutions)
    - calibration with control points: compute all parameters of the model using bundle adjustment

Imaging Models

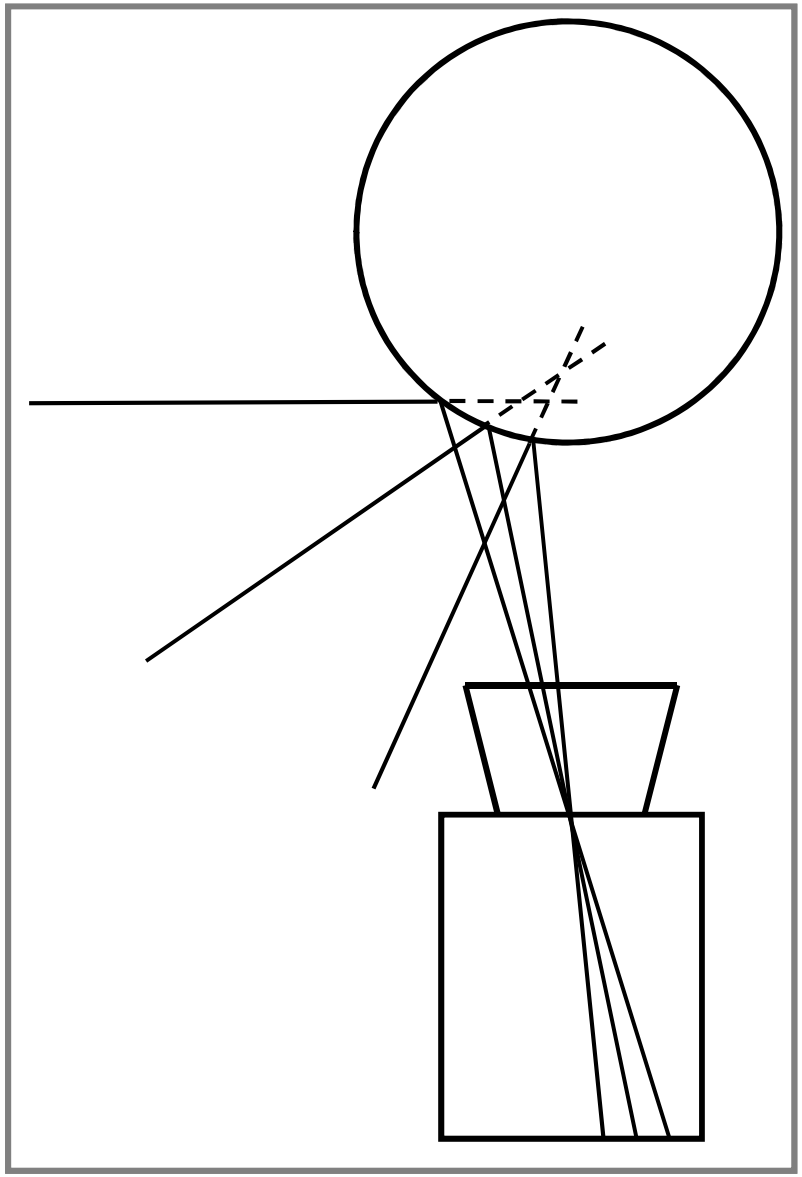




Imaging Models

Non-single viewpoint cameras:

- **Non-central** catadioptric systems
  - spheres, cones or any non-quadric mirrors give non-central system: projection rays do not intersect in a single point
  - calibration approaches have been developed for individual systems
  - example:
    - mirror that leads to equi-angular imaging model



Imaging Models

All above imaging models are subsumed by the following **generic imaging model**:

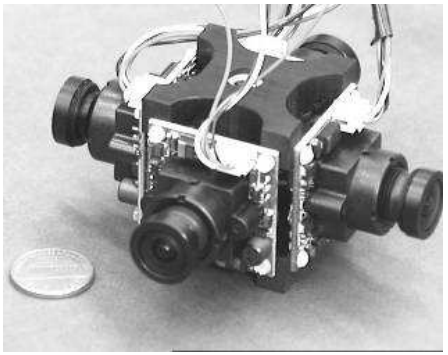
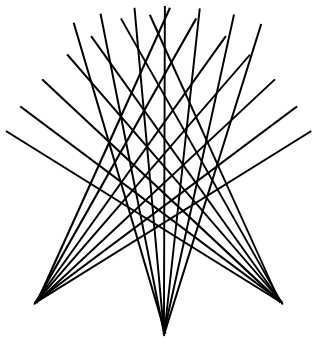
- A pixel “watches along” one viewing ray
- Camera model is lookup table, containing for each pixel the coordinates of the associated ray

Calibration = computation of all these rays

Imaging Models

Comments on the generic imaging model:

- is idealized (in reality, a pixel sees more than a line)
- more complete model, including radiometric properties, is used by Grossberg and Nayar (ICCV 2001)
- other sampling than pixel-wise is possible (e.g. sub-pixel)
- conceptually, allows to consider a stereo or multi-camera system as a single camera: union of their pixels and associated rays

Contents

- Introduction
- General imaging models
- Non-parametric calibration and distortion correction
- Non-parametric self-calibration
- Structure-from-motion

Non-parametric calibration — Basic idea

Input: images of calibration objects

Goal: compute projection ray for each pixel, in some 3D coordinate system

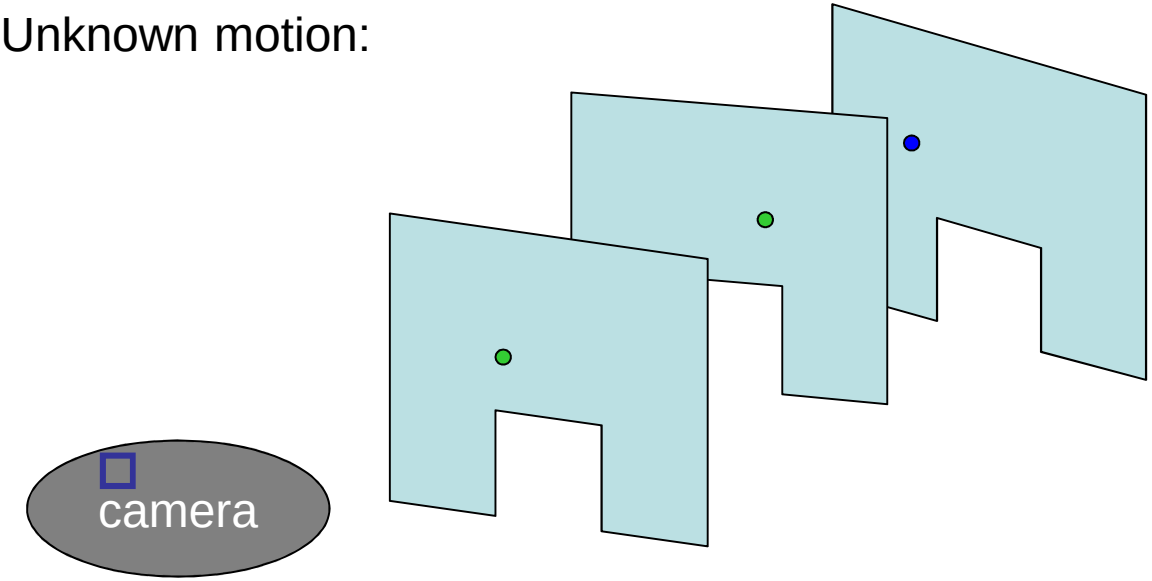
- General approach applicable for non-central cameras
- Variants for special cases (central and axial cameras)

Non-parametric calibration

General approach

[Sturm-Ramalingam-ECCV'04]

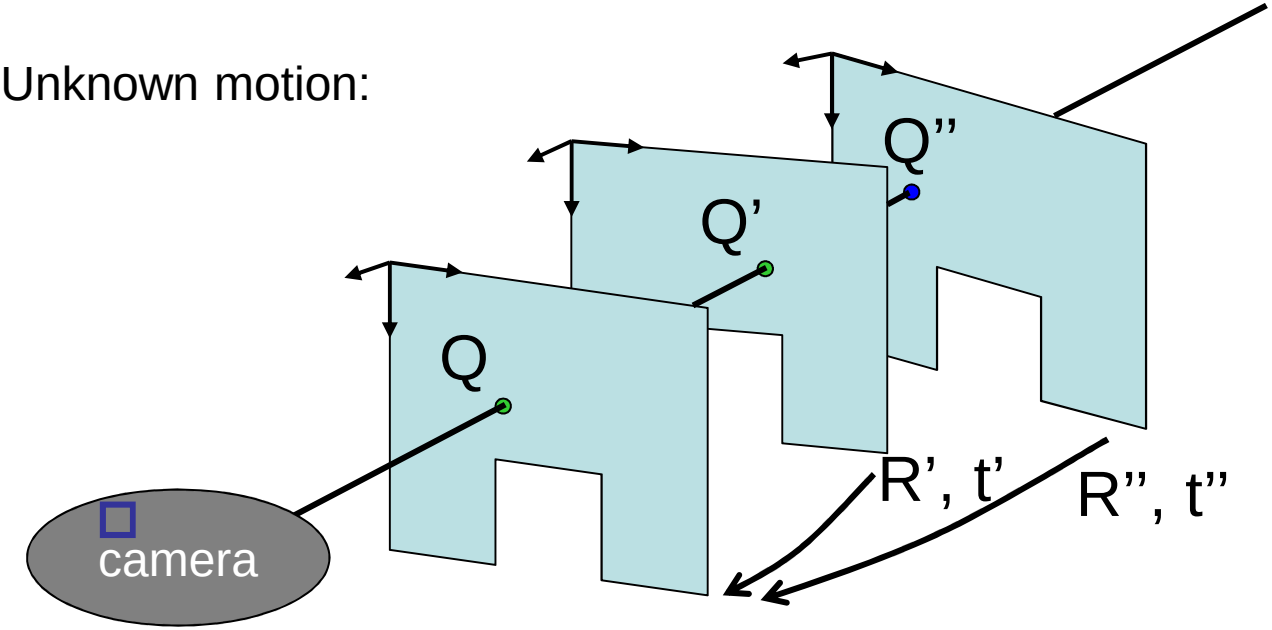
Unknown motion:



Non-parametric calibration — General approach

[Sturm-Ramalingam-ECCV'04]

Unknown motion:

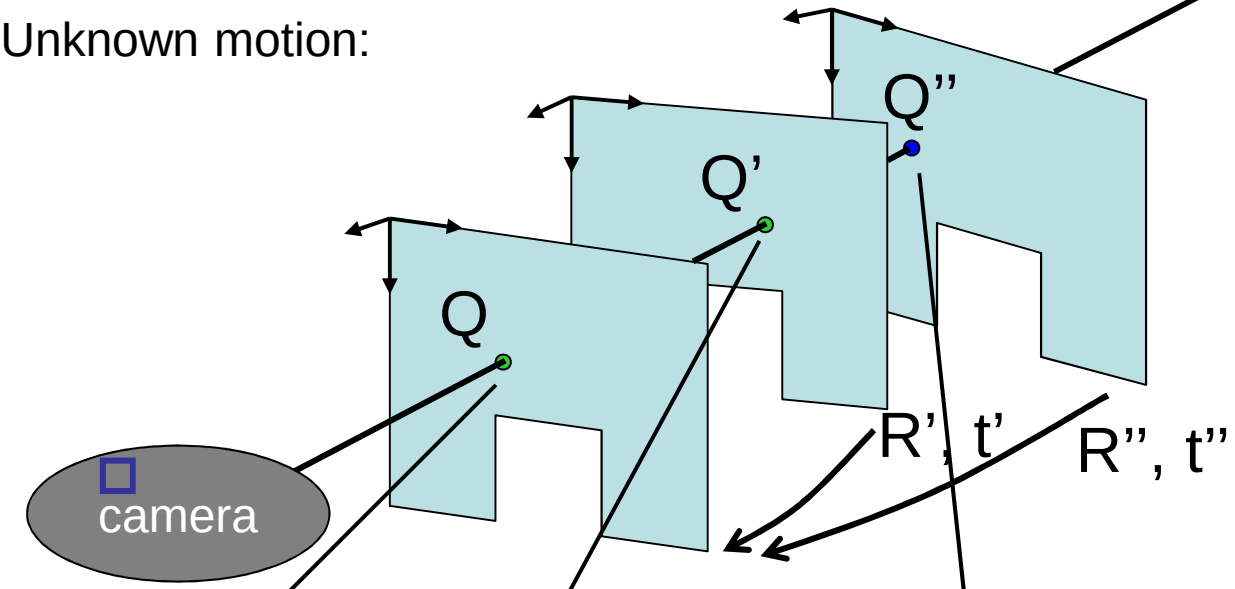


Estimate motions that make points collinear

Non-parametric calibration — General approach

[Sturm-Ramalingam-ECCV'04]

Unknown motion:



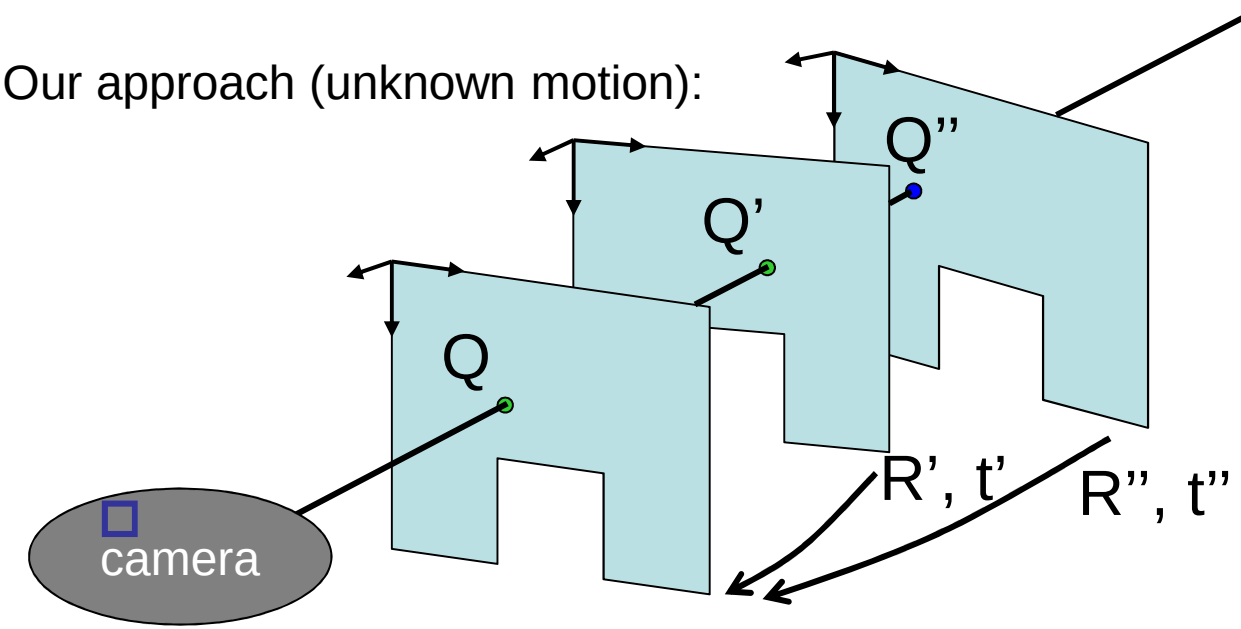
Estimate motions that make points collinear

$$\begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{pmatrix} \quad \left(R' \begin{pmatrix} Q'_1 \\ Q'_2 \\ Q'_3 \\ Q'_4 \end{pmatrix} + t' Q'_4 \right) \quad \left(R'' \begin{pmatrix} Q''_1 \\ Q''_2 \\ Q''_3 \\ Q''_4 \end{pmatrix} + t'' Q''_4 \right)$$

Non-parametric calibration — General approach

[Sturm-Ramalingam-ECCV'04]

Our approach (unknown motion):



Estimate motions that make points collinear

$$\begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{pmatrix} \quad R' \begin{pmatrix} Q'_1 \\ Q'_2 \\ Q'_3 \\ Q'_4 \end{pmatrix} + t' Q'_4 \quad R'' \begin{pmatrix} Q''_1 \\ Q''_2 \\ Q''_3 \\ Q''_4 \end{pmatrix} + t'' Q''_4 \Bigg)_{4x3} \longrightarrow \text{rank} < 3$$

$$\begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{pmatrix} \quad R' \begin{pmatrix} Q'_1 \\ Q'_2 \\ Q'_3 \end{pmatrix} + t' \, Q'_4 \quad R'' \begin{pmatrix} Q''_1 \\ Q''_2 \\ Q''_3 \end{pmatrix} + t'' \, Q''_4 \Bigg)_{4x3} \longrightarrow \text{rank} < 3$$

$$\begin{pmatrix} Q_1 & R'\begin{pmatrix} Q'_1 \\ Q'_2 \\ Q'_3 \end{pmatrix} + t' Q'_4 & R''\begin{pmatrix} Q''_1 \\ Q''_2 \\ Q''_3 \end{pmatrix} + t'' Q''_4 \\ Q_2 & & \\ Q_3 & & \\ Q_4 & Q'_4 & Q''_4 \end{pmatrix}_{4x3} \longrightarrow \text{rank} < 3$$

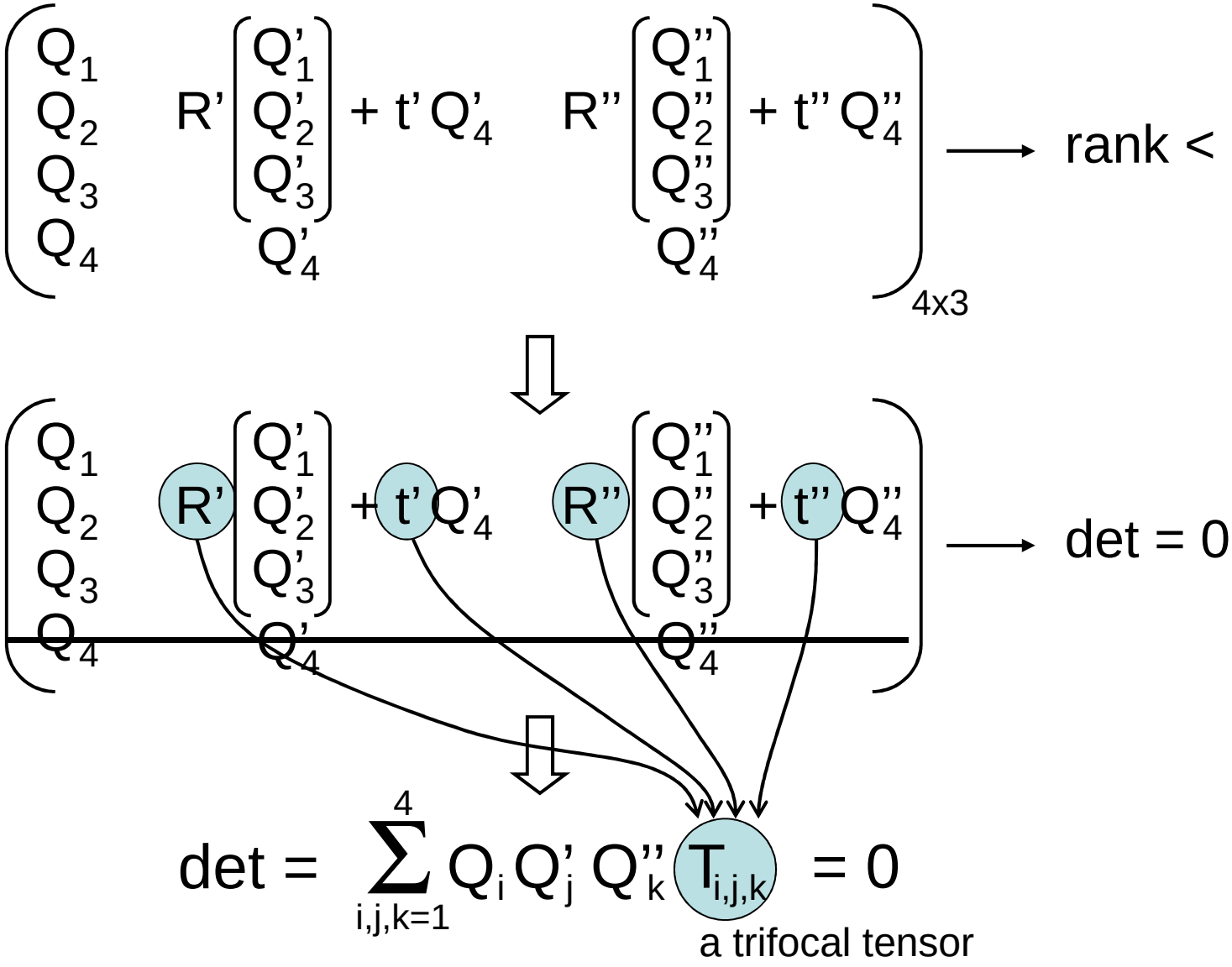
$$\Downarrow$$

$$\begin{pmatrix} Q_1 & R'\begin{pmatrix} Q'_1 \\ Q'_2 \\ Q'_3 \end{pmatrix} + t' Q'_4 & R''\begin{pmatrix} Q''_1 \\ Q''_2 \\ Q''_3 \end{pmatrix} + t'' Q''_4 \\ Q_2 & & \\ Q_3 & & \\ \hline Q_4 & Q'_4 & Q''_4 \end{pmatrix} \longrightarrow \det = 0$$

$$\Downarrow$$

$$\det = \sum_{i,j,k=1}^{4} Q_i \, Q'_j \, Q''_k \, T_{i,j,k} = 0$$

a trifocal tensor


Q1 Q2 Q3 Q4
R' Q'1 Q'2 Q'3 Q'4 + t' Q'4
R'' Q''1 Q''2 Q''3 Q''4 + t'' Q''4
4x3
rank < 3
R' + t' Q'4
R'' + t'' Q''4
det = 0
det = Σ i,j,k=1 4 Qi Q'j Q''k Ti,j,k = 0
a trifocal tensor

4 such tensors exist, striking out one row in turn:

$$\begin{pmatrix} Q_1 & R' \begin{pmatrix} Q'_1 \\ Q'_2 \\ Q'_3 \end{pmatrix} + t' Q'_4 & R'' \begin{pmatrix} Q''_1 \\ Q''_2 \\ Q''_3 \end{pmatrix} + t'' Q''_4 \\ Q_2 \\ Q_3 \\ \cancel{Q_4} & Q'_4 & Q''_4 \end{pmatrix}_{4x3} \longrightarrow \det = 0$$

Each one has a particular structure, see the following slide for two examples

Non-parametric calibration

General approach

i	C_i	V_i	W_i
1	$Q_1 Q_1' Q_4''$	0	R_{31}'
2	$Q_1 Q_2' Q_4''$	0	R_{32}'
3	$Q_1 Q_3' Q_4''$	0	R_{33}'
4	$Q_1 Q_4' Q_1''$	0	$-R_{31}''$
5	$Q_1 Q_4' Q_2''$	0	$-R_{32}''$
6	$Q_1 Q_4' Q_3''$	0	$-R_{33}''$
7	$Q_1 Q_4' Q_4''$	0	$t_3' - t_3''$
8	$Q_2 Q_1' Q_4''$	R_{31}'	0
9	$Q_2 Q_2' Q_4''$	R_{32}'	0
10	$Q_2 Q_3' Q_4''$	R_{33}'	0
11	$Q_2 Q_4' Q_1''$	$-R_{31}''$	0
12	$Q_2 Q_4' Q_2''$	$-R_{32}''$	0
13	$Q_2 Q_4' Q_3''$	$-R_{33}''$	0
14	$Q_2 Q_4' Q_4''$	$t_3' - t_3''$	0
15	$Q_3 Q_1' Q_4''$	$-R_{21}'$	$-R_{11}'$
16	$Q_3 Q_2' Q_4''$	$-R_{22}'$	$-R_{12}'$
17	$Q_3 Q_3' Q_4''$	$-R_{23}'$	$-R_{13}'$
18	$Q_3 Q_4' Q_1''$	R_{21}''	R_{11}''
19	$Q_3 Q_4' Q_2''$	R_{22}''	R_{12}''
20	$Q_3 Q_4' Q_3''$	R_{23}''	R_{13}''
21	$Q_3 Q_4' Q_4''$	$t_2'' - t_2'$	$t_1'' - t_1'$
22	$Q_4 Q_1' Q_1''$	$R_{21}' R_{31}'' - R_{21}'' R_{31}'$	$R_{11}' R_{31}'' - R_{11}'' R_{31}'$
23	$Q_4 Q_1' Q_2''$	$R_{21}' R_{32}'' - R_{22}'' R_{31}'$	$R_{11}' R_{32}'' - R_{12}'' R_{31}'$
24	$Q_4 Q_1' Q_3''$	$R_{21}' R_{33}'' - R_{23}'' R_{31}'$	$R_{11}' R_{33}'' - R_{13}'' R_{31}'$
25	$Q_4 Q_1' Q_4''$	$R_{21}' t_3'' - R_{31}' t_2''$	$R_{11}' t_3'' - R_{31}' t_1''$
26	$Q_4 Q_2' Q_1''$	$R_{22}' R_{31}'' - R_{21}'' R_{32}'$	$R_{12}' R_{31}'' - R_{11}'' R_{32}'$
27	$Q_4 Q_2' Q_2''$	$R_{22}' R_{32}'' - R_{22}'' R_{32}'$	$R_{12}' R_{32}'' - R_{12}'' R_{32}'$
28	$Q_4 Q_2' Q_3''$	$R_{22}' R_{33}'' - R_{23}'' R_{32}'$	$R_{12}' R_{33}'' - R_{13}'' R_{32}'$
29	$Q_4 Q_2' Q_4''$	$R_{22}' t_3'' - R_{32}' t_2''$	$R_{12}' t_3'' - R_{32} t_1''$
30	$Q_4 Q_3' Q_1''$	$R_{23}' R_{31}'' - R_{21}'' R_{33}'$	$R_{13}' R_{31}'' - R_{11}'' R_{33}'$
31	$Q_4 Q_3' Q_2''$	$R_{23}' R_{32}'' - R_{22}'' R_{33}'$	$R_{13}' R_{32}'' - R_{12}'' R_{33}'$
32	$Q_4 Q_3' Q_3''$	$R_{23}' R_{33}'' - R_{23}'' R_{33}'$	$R_{13}' R_{33}'' - R_{13}'' R_{33}'$
33	$Q_4 Q_3' Q_4''$	$R_{23}' t_3'' - R_{33}' t_2''$	$R_{13}' t_3'' - R_{33}' t_1''$
34	$Q_4 Q_4' Q_1''$	$R_{31}'' t_2' - R_{21}'' t_3'$	$R_{31}'' t_1' - R_{11}'' t_3'$
35	$Q_4 Q_4' Q_2''$	$R_{32}'' t_2' - R_{22}'' t_3'$	$R_{32}'' t_1' - R_{12}'' t_3'$
36	$Q_4 Q_4' Q_3''$	$R_{33}'' t_2' - R_{23}'' t_3'$	$R_{33}'' t_1' - R_{13}'' t_3'$
37	$Q_4 Q_4' Q_4''$	$t_2' t_3'' - t_3' t_2''$	$t_1' t_3'' - t_1'' t_3'$

Calibration algorithm:

(1) Take images of calibration object in different poses

(2) 2D-3D matching (pixels to points on object)

(3) Estimation of tensors, based on linear equations

$$\sum_{i,j,k=1}^{4} Q_i \, Q'_j \, Q''_k \; T_{i,j,k} = 0$$

and taking into account the tensors' structure (e.g. coefficients that are zero)

(4) Extraction of motion parameters from tensors:

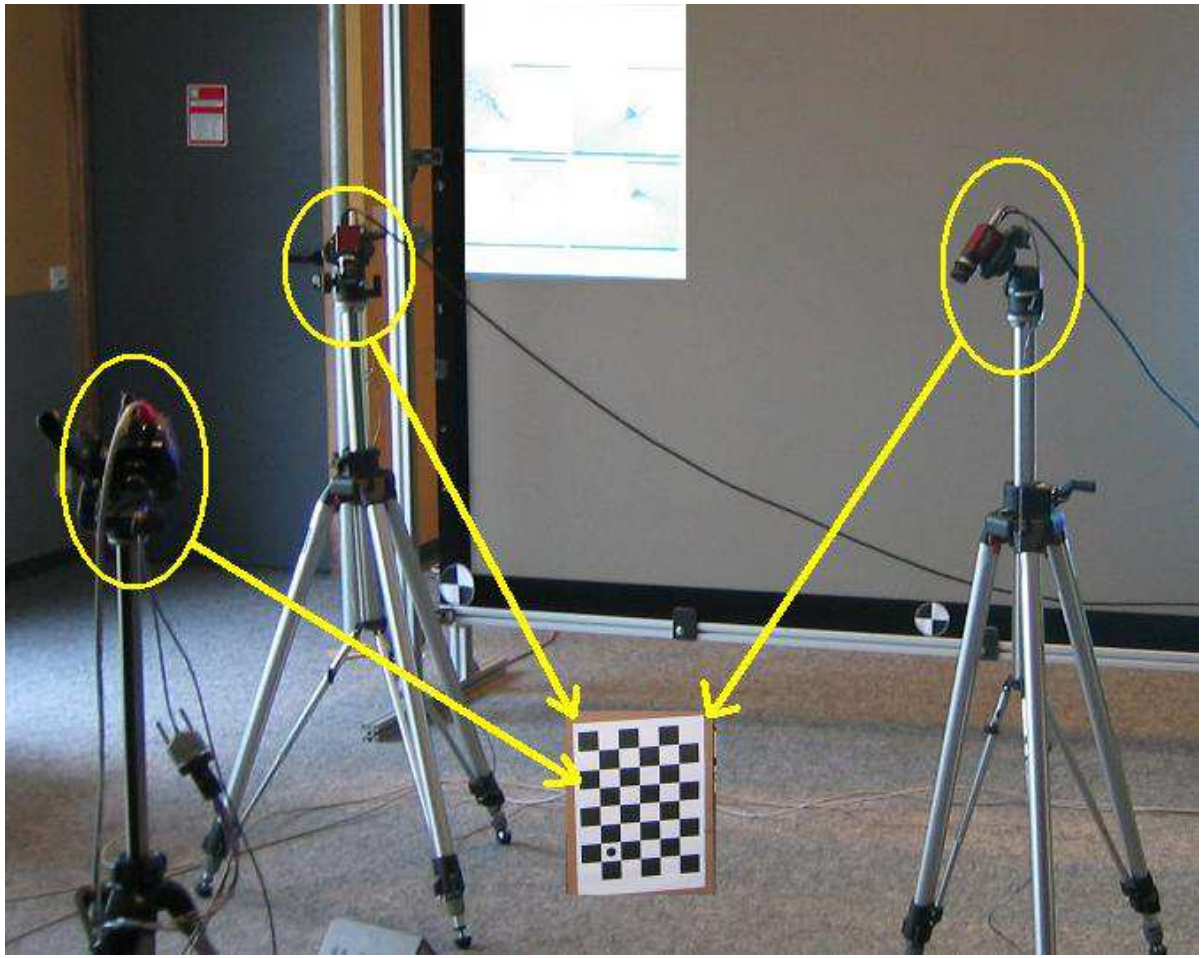
- some can be directly read off (some rotation coefficients, cf. previous slide)
- others can be computed using orthonormality constraints on R' and R''

(5) Put calibration grids in same 3D coordinate system

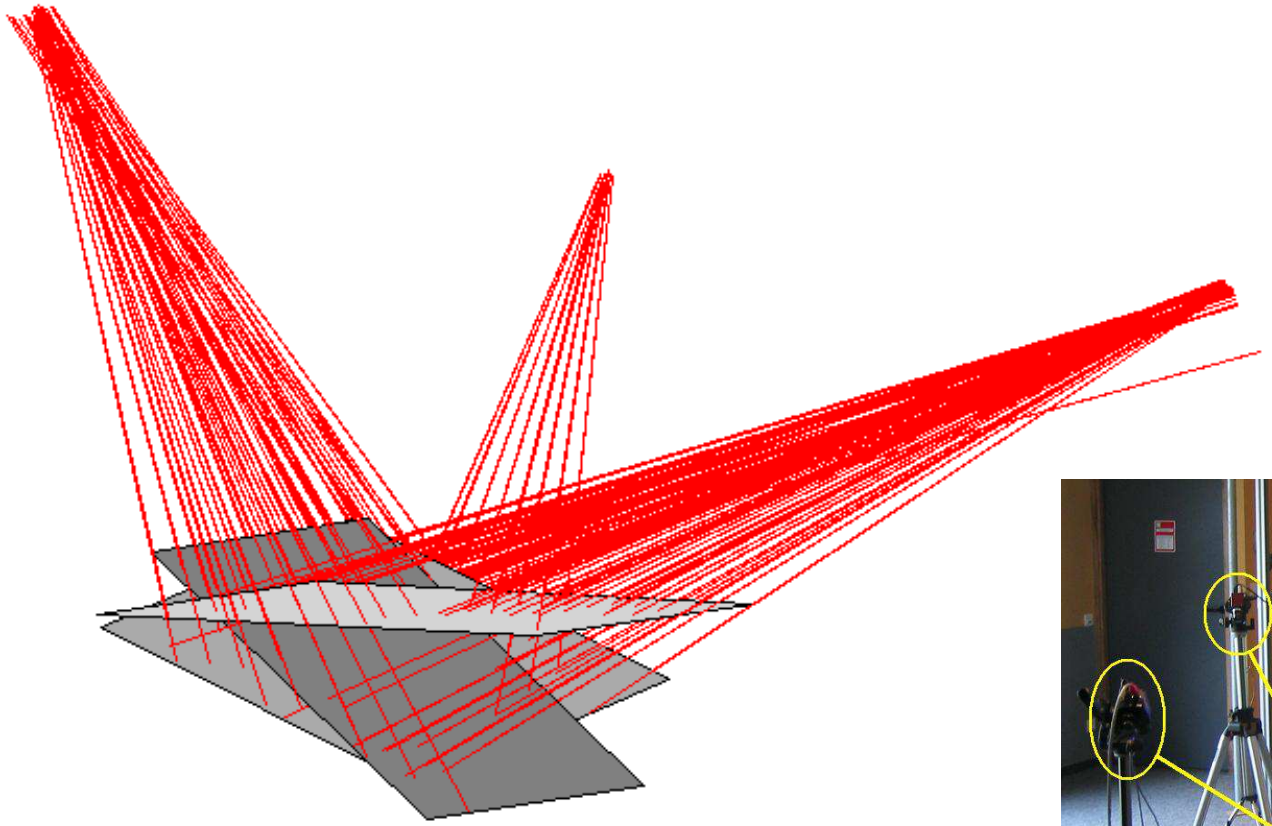
(6) Compute projection rays: for each pixel join the associated calibration points

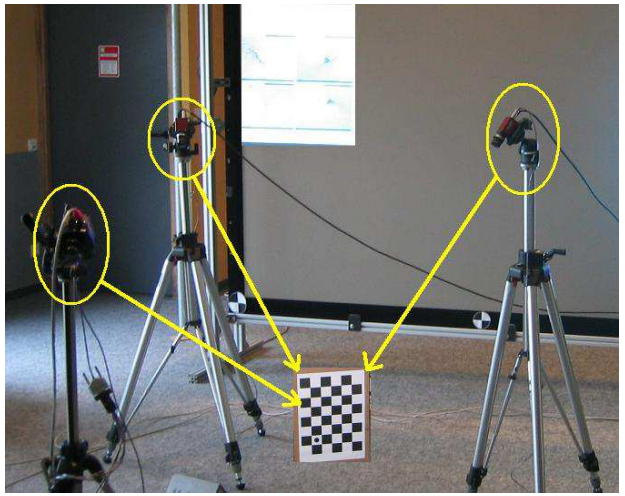
(7) Bundle adjustment

Results for non-central camera
(multi-camera system, considered as single non-central camera):

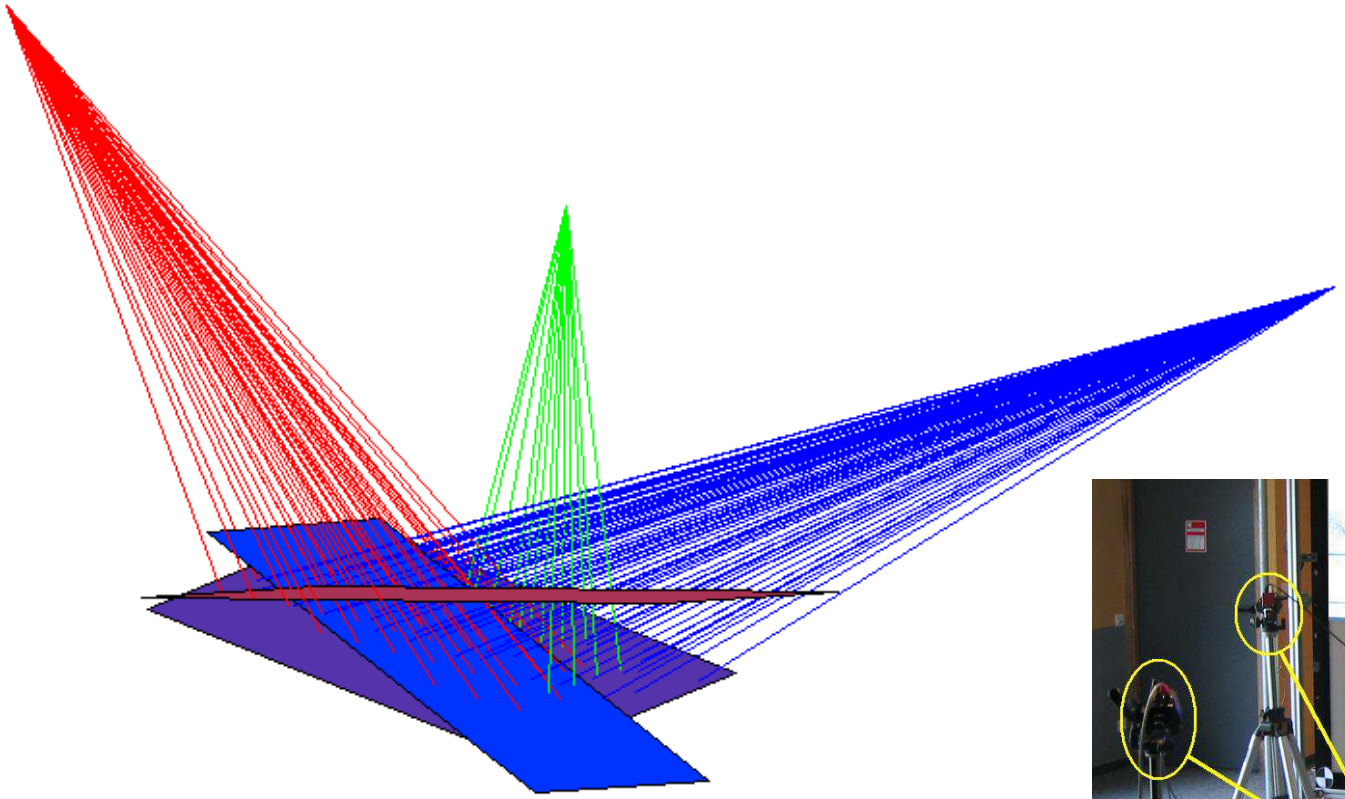


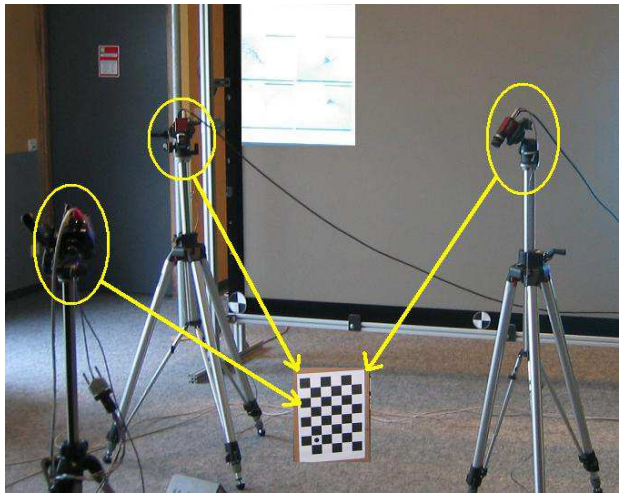
Results for non-central camera:





Results for non-central camera: after constraining rays into central clusters





Intermediate discussion:

- the approach is designed for 3D calibration objects

  → ***variant for using* planar *calibration objects (see next)***

- this approach uses *exactly* 3 images
  - only pixels covered by all 3 images of the calibration grid are calibrated
    → especially with large field of view, difficult to calibrate whole image
  - results may not be highly accurate

  → ***methods for using multiple images (see later)***

- the approach allows to calibrate non-central cameras!
- BUT: if used with images acquired by central camera
  - tensors are not computed uniquely (linear equation system of too low rank)
    → calibration fails

  → ***variant of the approach for central cameras and a few other special cases (see later)***

Using planar calibration grids: $Q_3 = Q'_3 = Q''_3 = 0$

$$\begin{pmatrix} Q_1 \\ Q_2 \\ 0 \\ Q_4 \end{pmatrix} \quad R' \begin{pmatrix} Q'_1 \\ Q'_2 \\ 0 \\ Q'_4 \end{pmatrix} + t' Q'_4 \quad R'' \begin{pmatrix} Q''_1 \\ Q''_2 \\ 0 \\ Q''_4 \end{pmatrix} + t'' Q''_4 \Bigg)_{4x3} \longrightarrow \text{rank} < 3$$

- Tensors are different
- Extraction of motion parameters is more complicated, but possible

Non-parametric calibration — Using multiple images

Using multiple images:

- Idea:

  (1) Initial calibration using 3 images and above approach

  (2) Consider an additional image:

  - Compute pose of calibration grid using already available calibration information
  - Extend the calibration to pixels covered by the additional grid

  (3) Repeat (2) for all images. Then, bundle adjustment.

- Also:

  Possibility of performing initial calibration using multiple images if the regions covered by the grids mutually overlap

[Ramalingam-etal-CVPR'05]

Non-parametric calibration

If the calibration approach is used with images acquired by a central camera, then tensors are not computed uniquely (linear equation system of too low rank)

→ calibration fails

We thus consider a hierarchy of generic imaging models:

- Non-central
- Axial (non-central, but all projection rays touch a line, the **camera axis**)
  - linear push-broom camera
  - catadioptric system using a spherical mirror
- Central (all projection rays go through a single point, the **optical center**)

Calibration approaches for these three models have been developed

Non-parametric calibration Approach for central model

Approach for central model:

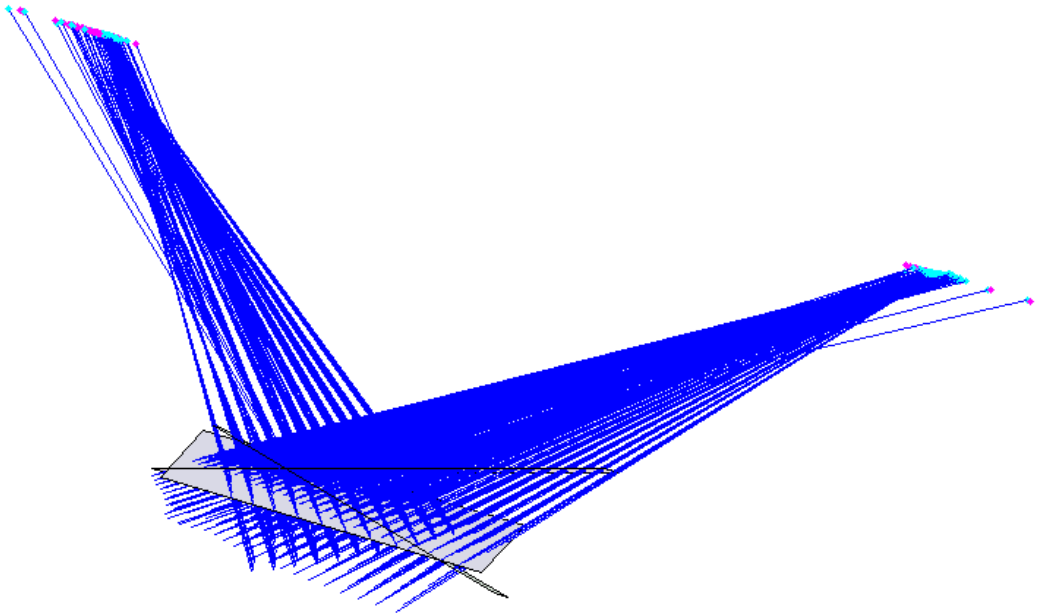
- Two images are sufficient (if 3D calibration object)
- Introduce coordinates of optical center C as unknowns
- Constraint: collinearity of optical center and two calibration points

$$\begin{pmatrix} C_1 & Q_1 & & \\ C_2 & Q_2 & R' \begin{pmatrix} Q'_1 \\ Q'_2 \\ Q'_3 \end{pmatrix} + t' & Q'_4 \\ C_3 & Q_3 & & \\ C_4 & Q_4 & & \end{pmatrix}_{4x3} \longrightarrow \text{rank} < 3$$

- Gives rise to yet another set of tensors
- Extraction of motion parameters and optical center from tensors

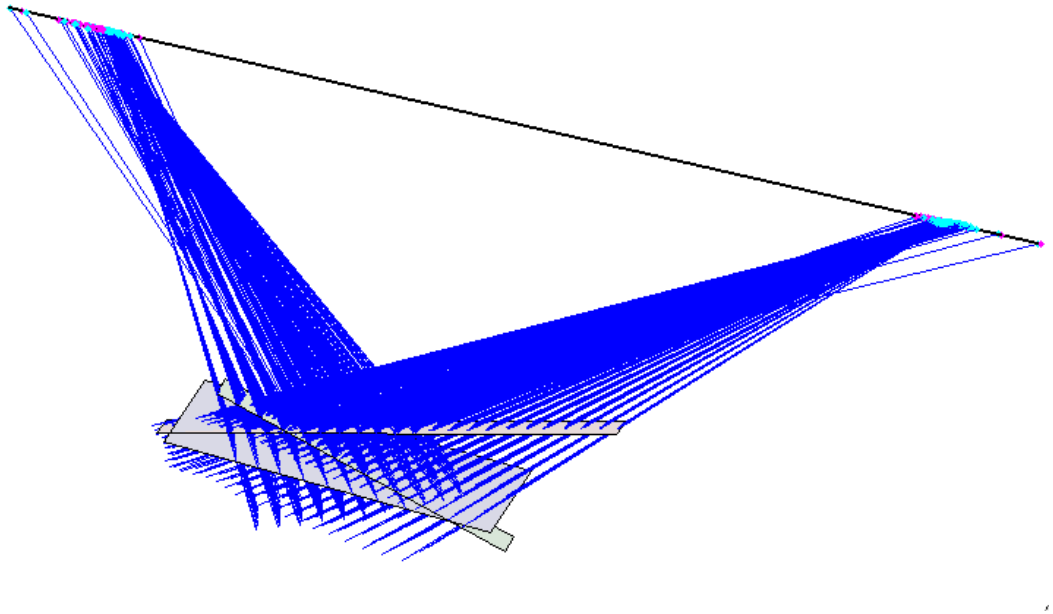
Similar approach for axial camera model (not shown here) [Ramalingam-etal-ACCV'06]

Results for axial camera model
(for a stereo system, considered as single axial camera):



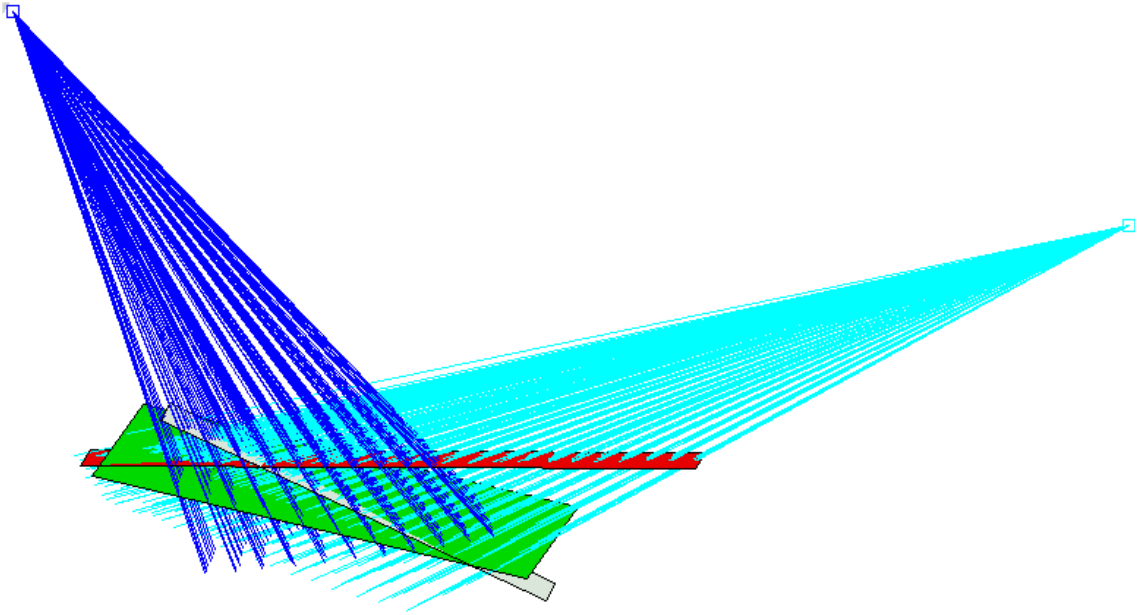
Results for axial camera model
(for a stereo system, considered as single axial camera):

After constraining rays to cut a single axis



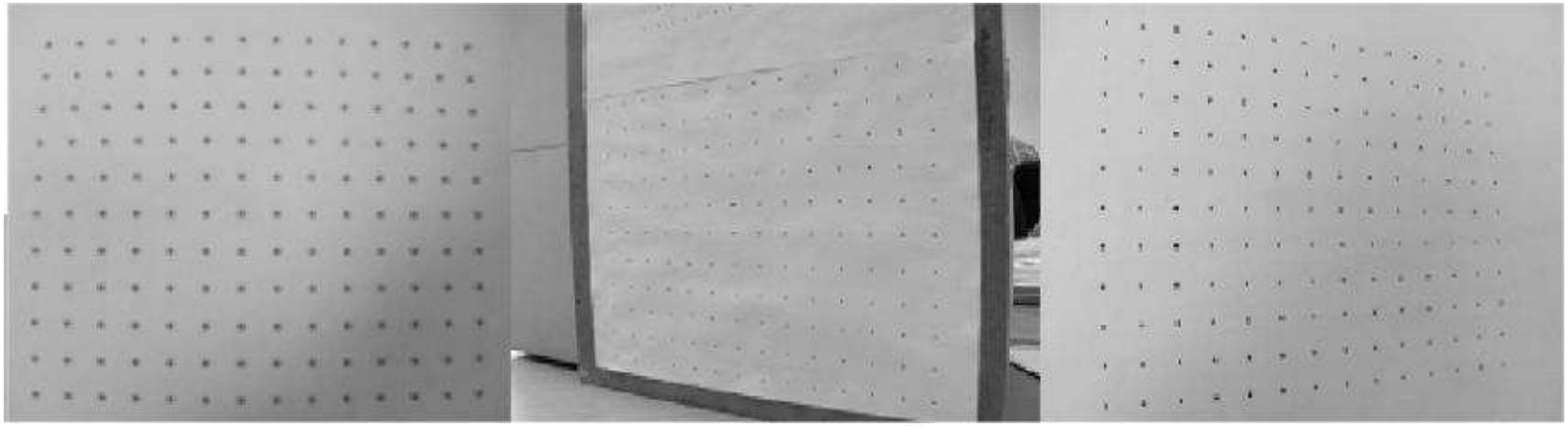
Results for axial camera model
(for a stereo system, considered as single axial camera):

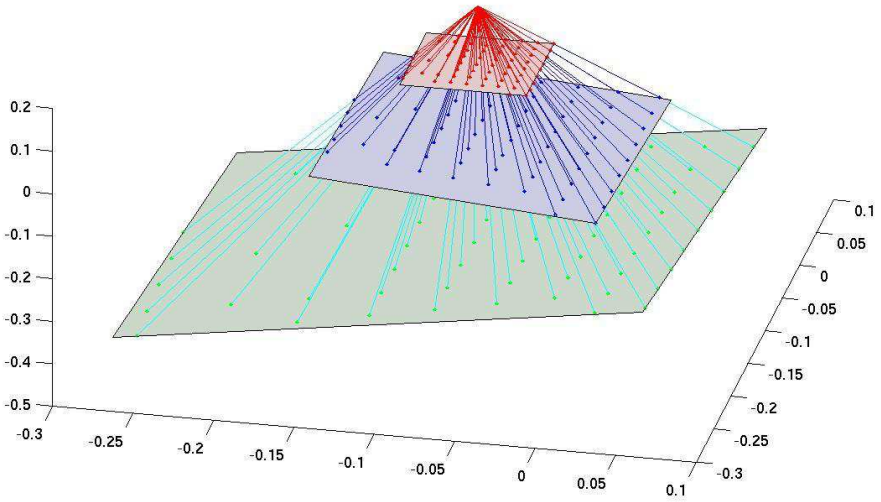
After constraining rays into central clusters

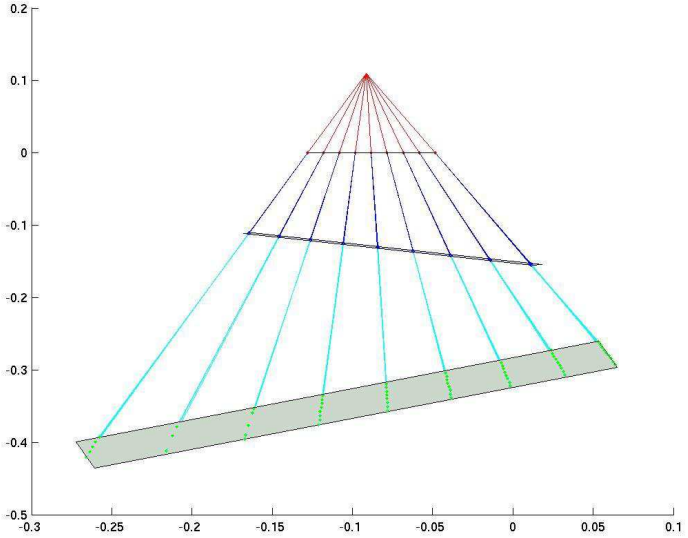


Non-parametric calibration Approach for central model

Central model applied on pinhole camera with slight radial distortion



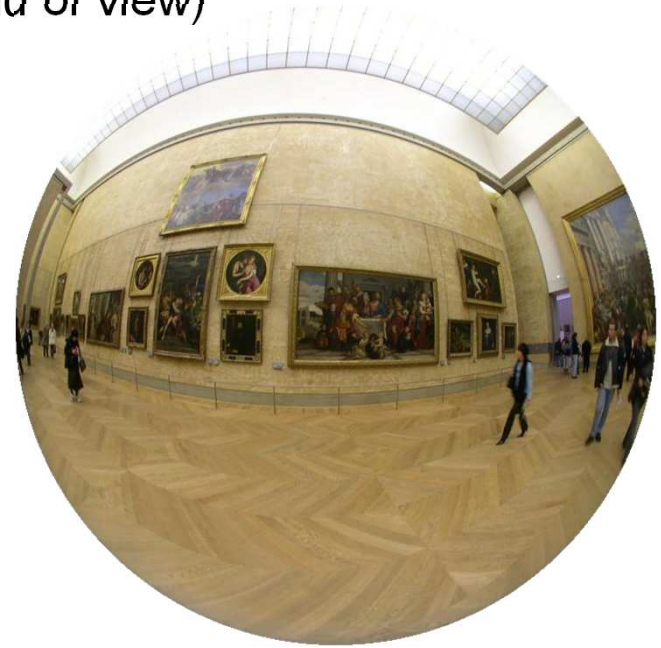




Non-parametric calibration

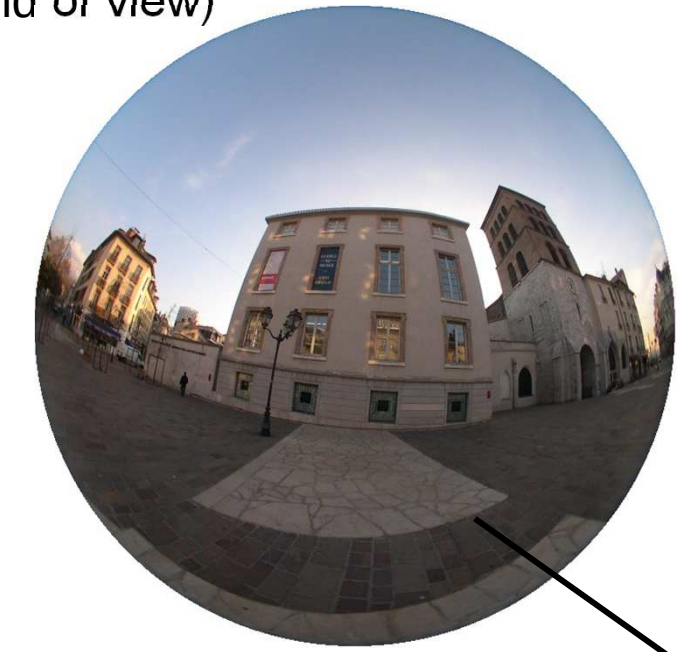
Approach for central model

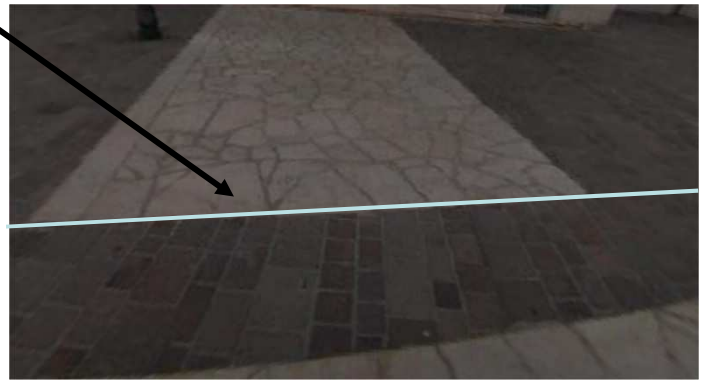
Results for fisheye camera
(183 field of view)



Non-parametric calibration Approach for central model

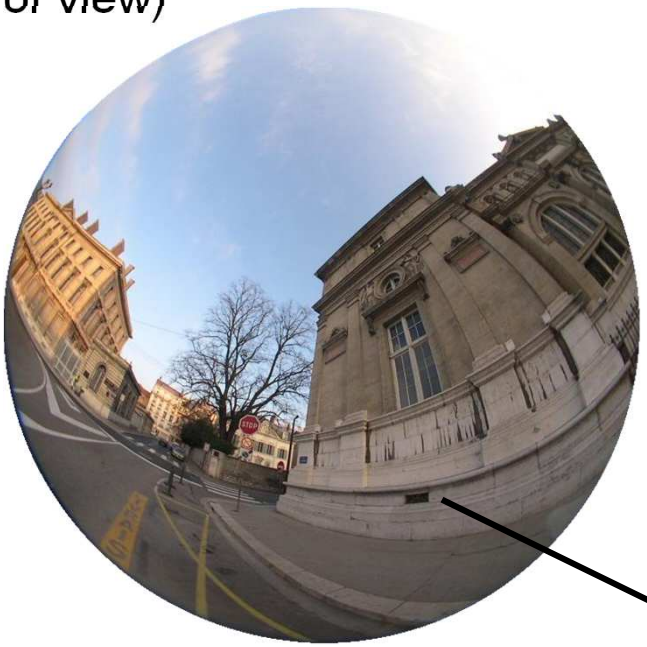
Results for fisheye camera
(183 field of view)





Non-parametric calibration Approach for central model

Results for fisheye camera
(183 field of view)



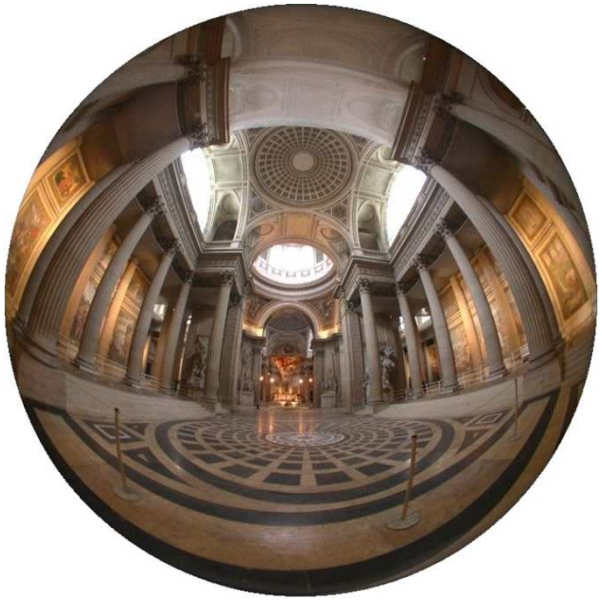
Distortion correction:

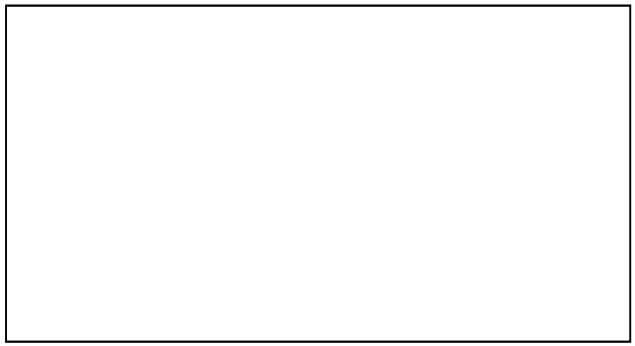
- Classical approach is based on analytical relationship between distorted and undistorted image coordinates (based on parametric calibration model)

- Generalization: approach for non-parametric calibration

General distortion correction approach (for central cameras):

- Input: image and calibration information (projection rays for all pixels)
- Idea:
  - attribute pixels' color to their projection rays in 3D
  - define some plane in 3D
  - cut all projection rays: at each intersection point, paint a dot of the ray's color
  - the painted plane shows a distortion corrected image



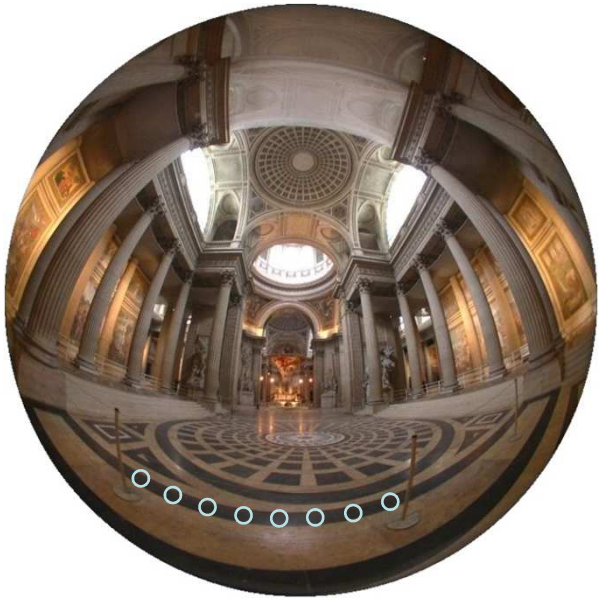


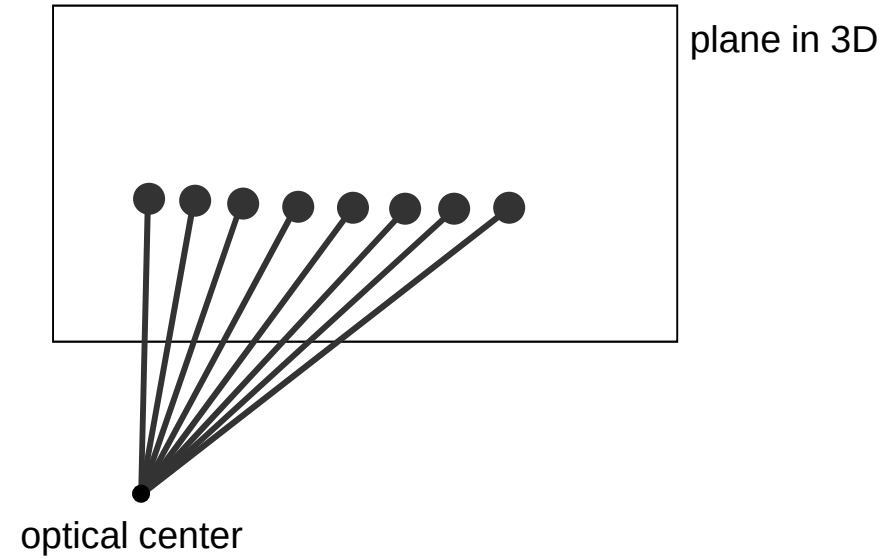
plane in 3D

optical center

General distortion correction approach (for central cameras):

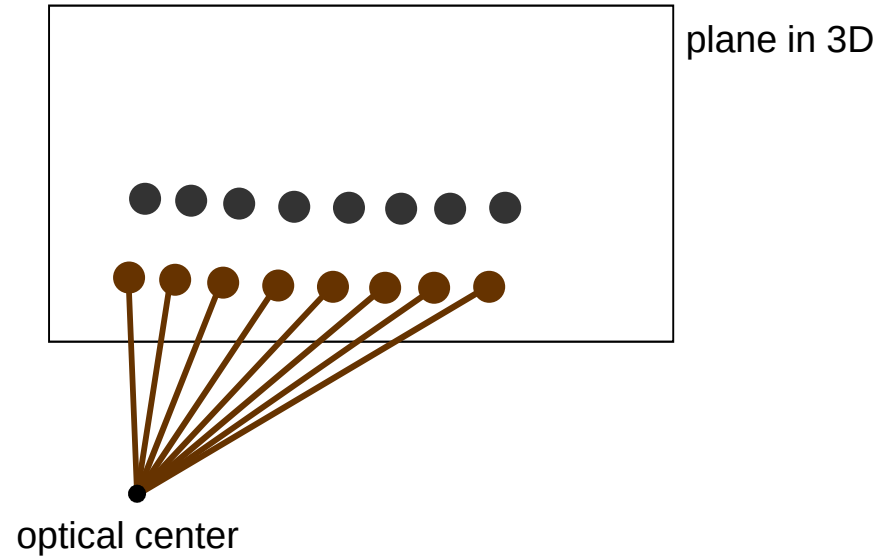
- Input: image and calibration information (projection rays for all pixels)
- Idea:
  - attribute pixels' color to their projection rays in 3D
  - define some plane in 3D
  - cut all projection rays: at each intersection point, paint a dot of the ray's color
  - the painted plane shows a distortion corrected image





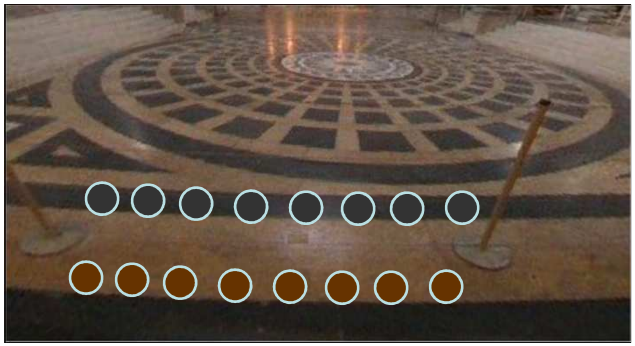
General distortion correction approach (for central cameras):

- Input: image and calibration information (projection rays for all pixels)
- Idea:
  - attribute pixels' color to their projection rays in 3D
  - define some plane in 3D
  - cut all projection rays: at each intersection point, paint a dot of the ray's color
  - the painted plane shows a distortion corrected image



General distortion correction approach (for central cameras):

- Input: image and calibration information (projection rays for all pixels)
- Idea:
  - attribute pixels' color to their projection rays in 3D
  - define some plane in 3D
  - cut all projection rays: at each intersection point, paint a dot of the ray's color
  - the painted plane shows a distortion corrected image



plane in 3D

optical center

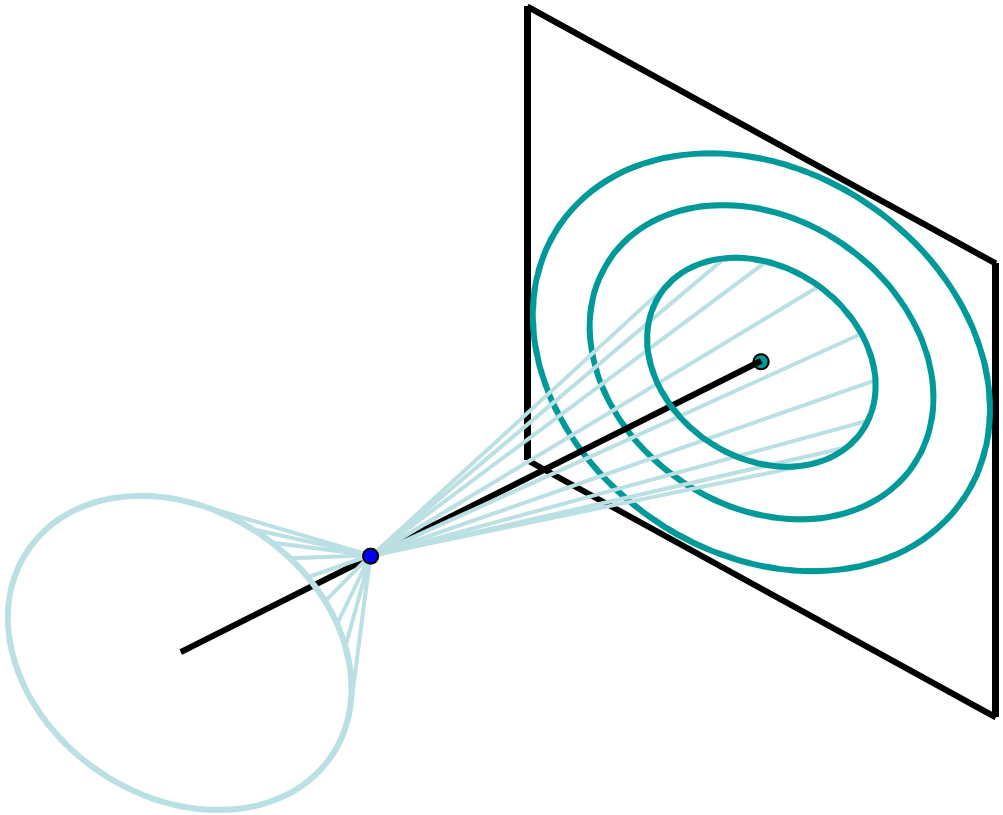
- General approach that allows to calibrate any camera
- Variants for central and axial camera modes
- Variants for using planar or 3D calibration objects

- How about stability?
  - Possible overfitting when calibrating “not very non-central cameras” with the general approach (result may be worse than with the central approach)
  - Stability depends on:
    - amount of “non-centrality”
    - number of images
    - accuracy of matches
  - If unstable:
    use more images, regularization, assumption of radial symmetry, …

- Here, pixel-wise discretization of camera model
- Any other discretization (sub-pixel or super-pixel) is possible
- Trade-off between
  - potential accuracy of calibration (the finer the discretization, the better)
  - potential instability (the finer the discretization, the more unknowns…)

Non-parametric calibration Radially symmetric cameras

Interesting special case: radially symmetric cameras



Calibration:

Computation of distortion center and

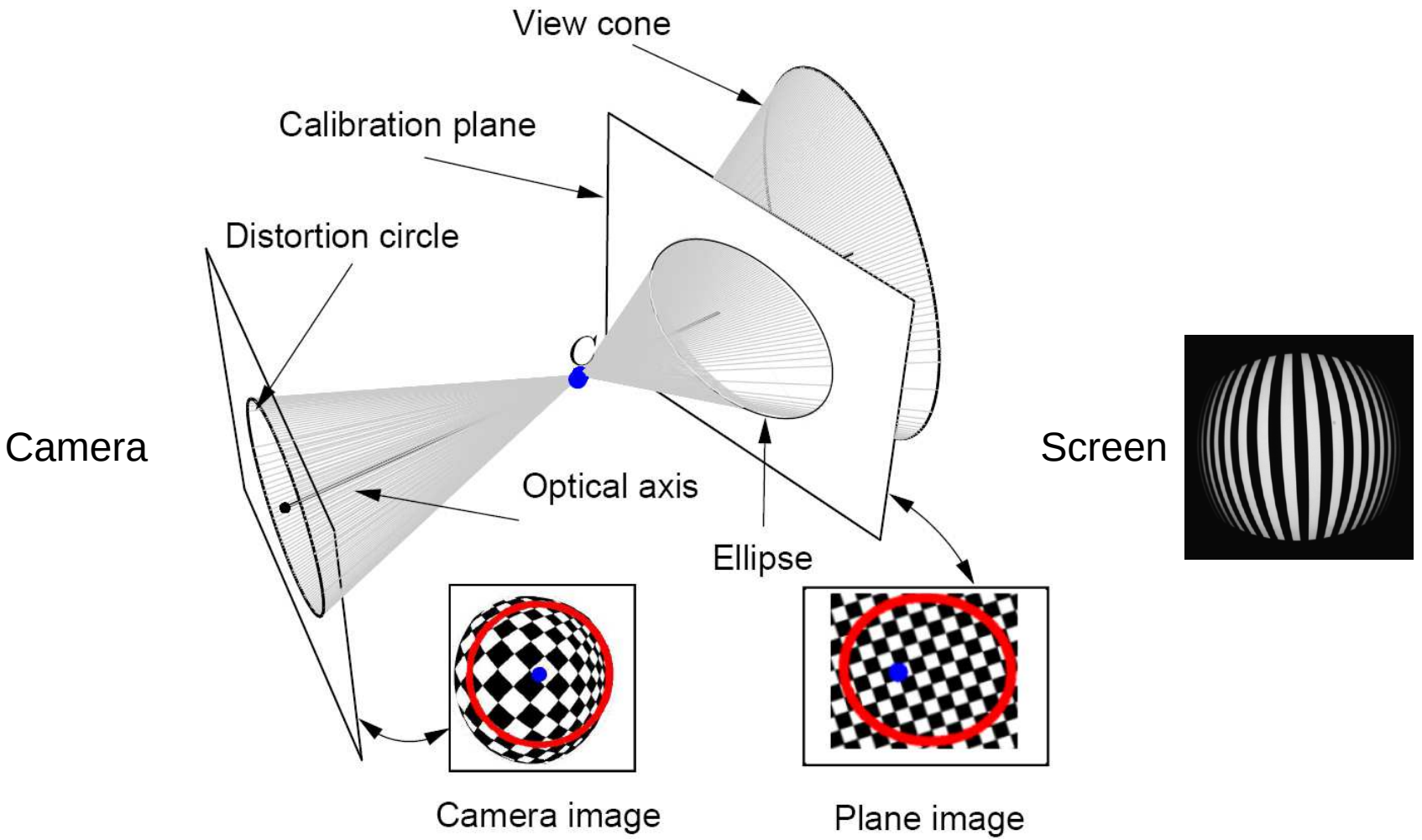
distortion function: *radius → view angle / focal length*

Note: each distortion circle ≡ perspective camera

[Tardif-Sturm-OMNIVIS'05]

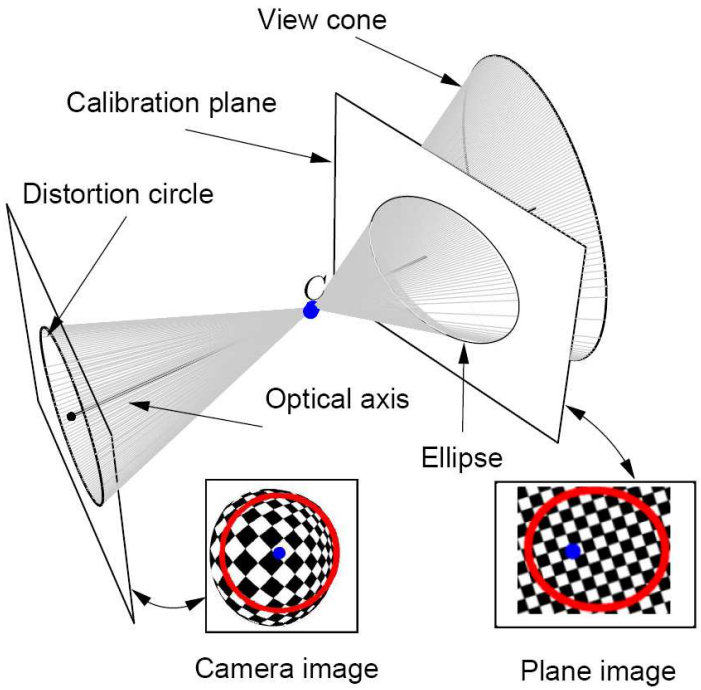
Non-parametric calibration Radially symmetric cameras

Calibration



Non-parametric calibration Radially symmetric cameras

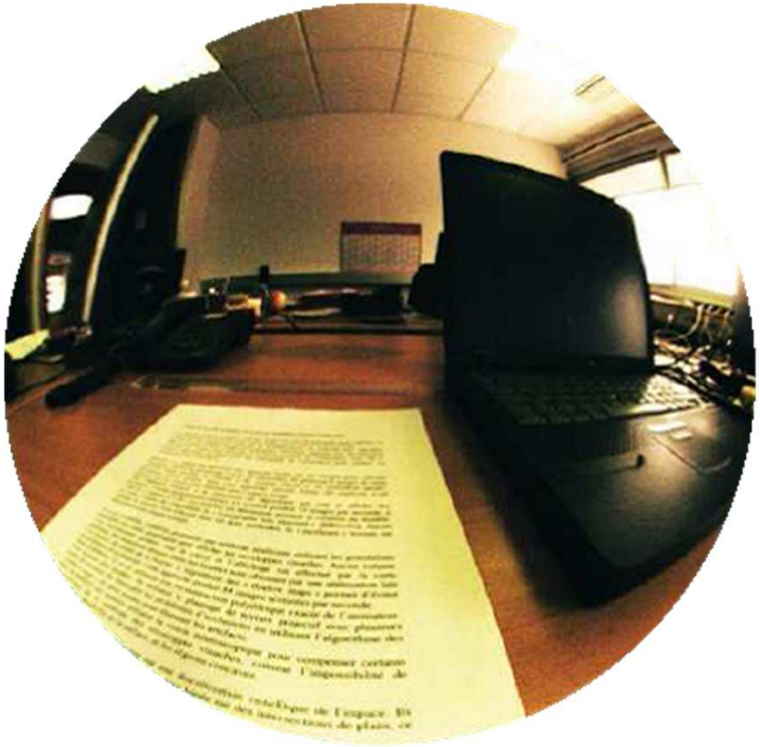
Calibration



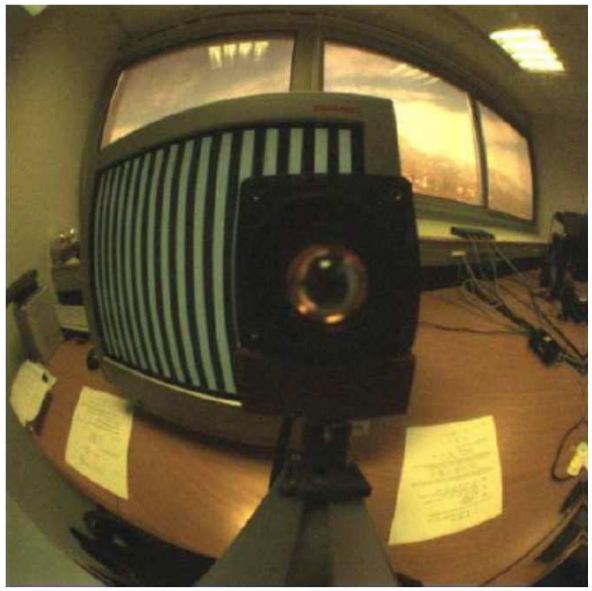
(1) For each distortion circle:
- compute homography screen ↔ image
- run classical plane-based calibration [Zhang'99, Sturm'99]

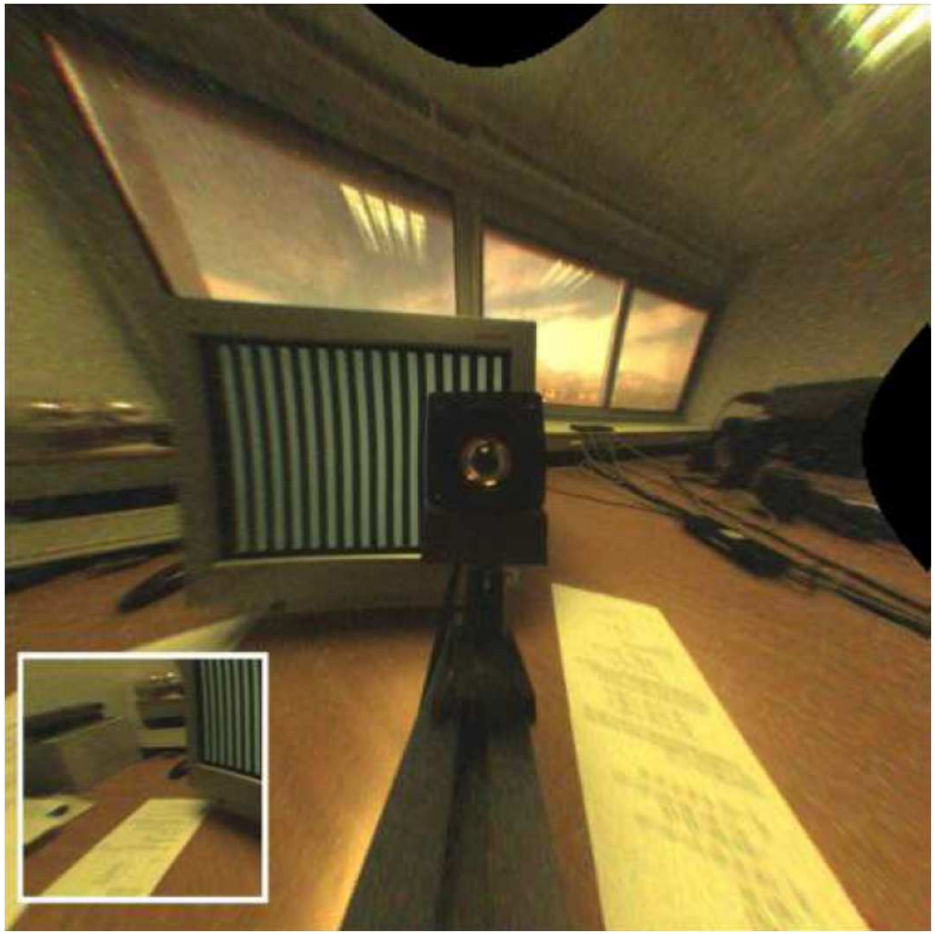
(2) Bundle adjustment over all distortion circles

Result of distortion correction for fisheye



Result for homemade “Christmas camera”





Non-parametric calibration Radially symmetric cameras

Discussion:

- Effective calibration approach for general radial distortion
- Handles field of view larger than 180° !
- Approaches for both, central and non-central cameras
- A single image is sufficient (but for stability, more images should be used)

Other recent work:

- Epipolar geometry of radially symmetric cameras [Barreto-Daniilidis-ICCV'05]
- Multi-view geometry and self-calibration of radial cameras [Thirthala-Pollefeys-ICCV'05]
- Self-calibration from two or more views of an arbitrary scene plane [Tardif-etal-ECCV'06]
- Direct method for computation of distortion center [Hartley-Kang-ICCV'05]

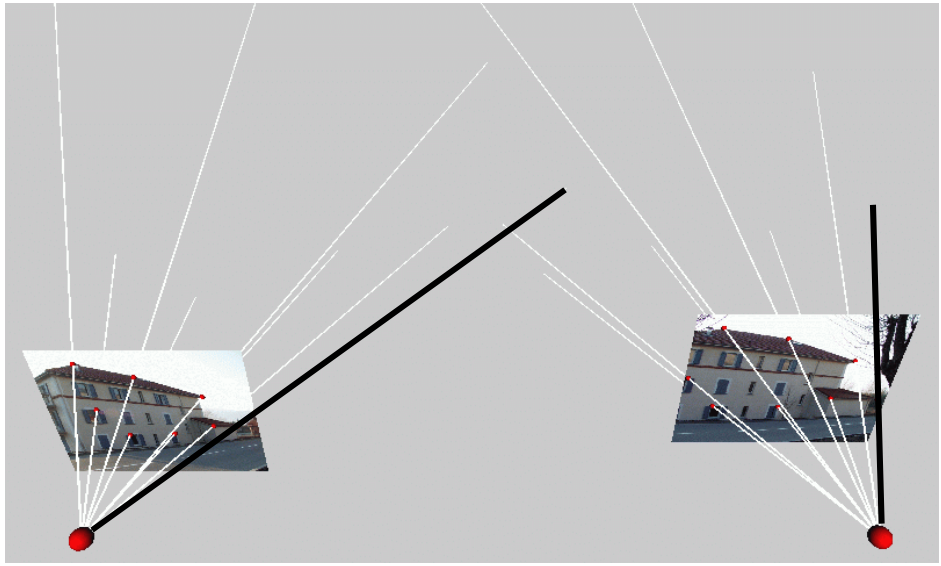
Outline

- Introduction and Imaging Systems
- Non-parametric calibration
- On Distortions
- Structure-from-motion
- Mirror design

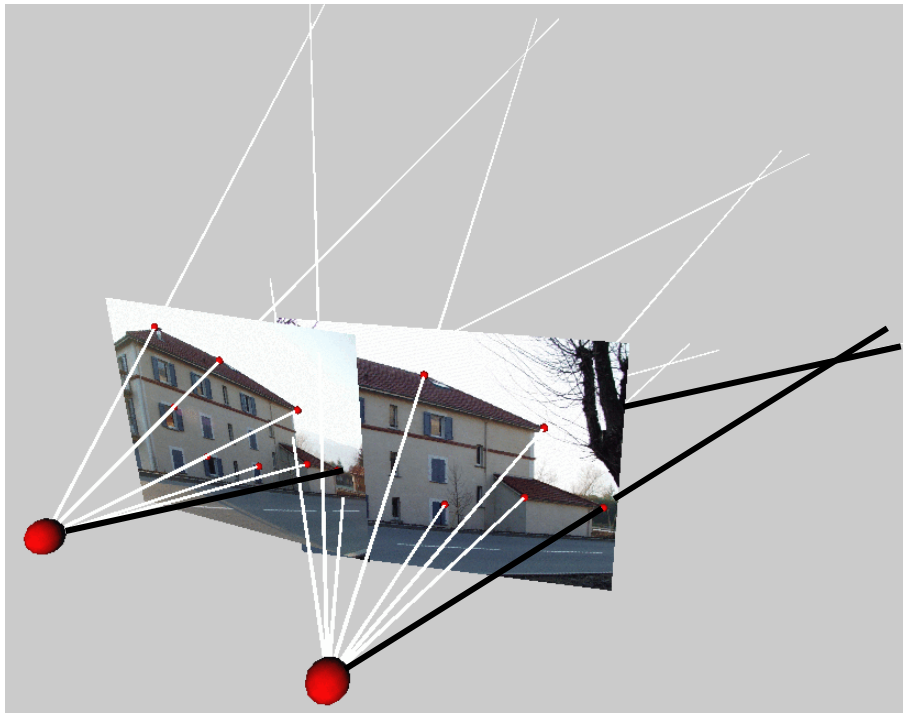
Motivation:

- Many different SfM algorithms (pose, motion, triangulation, …) exist, for different camera types
- But, in principle, if calibrated cameras are considered, one single approach for each SfM problem is sufficient, for all camera types

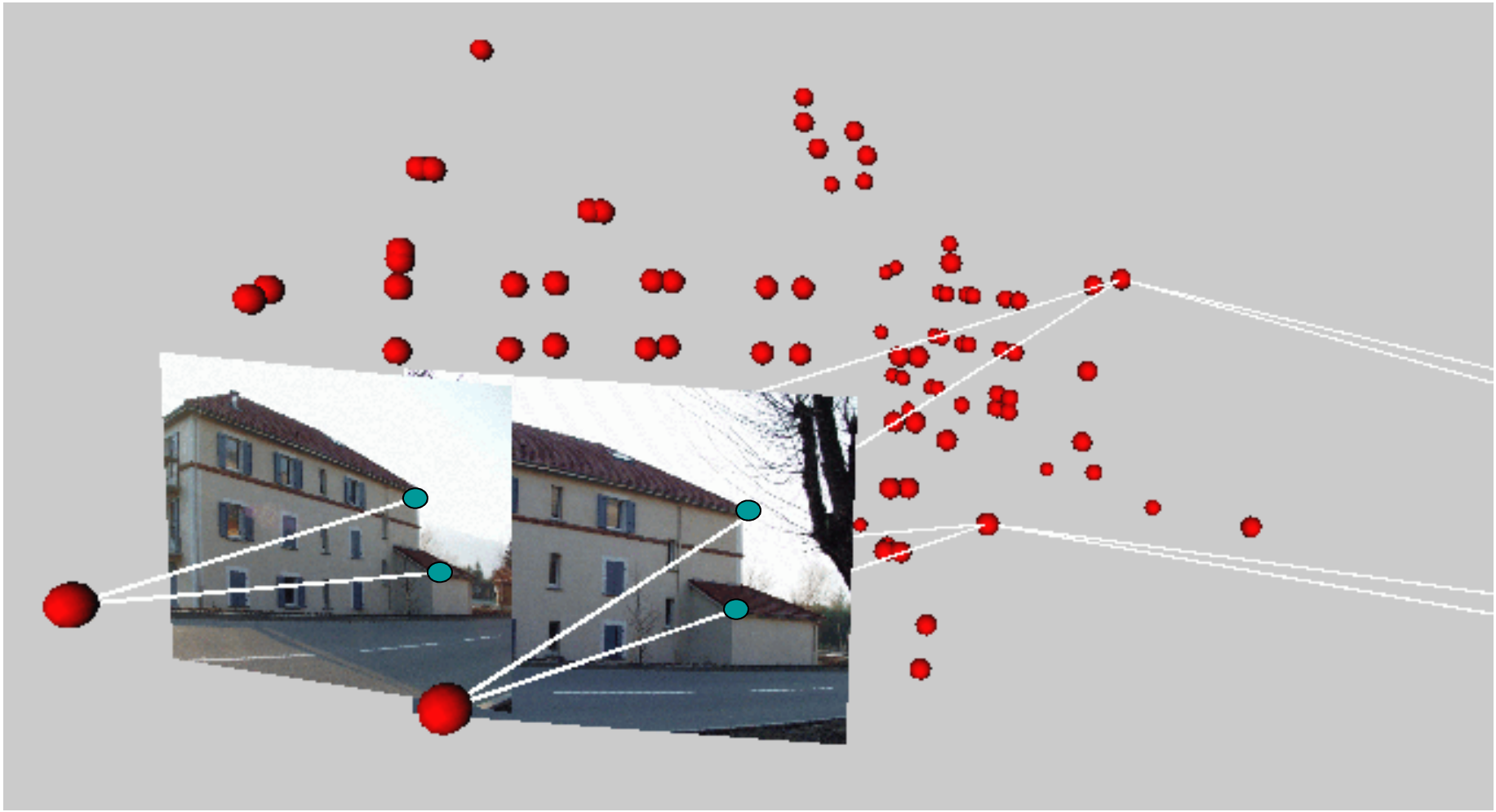
Calibration: determine, for each pixel, the corresponding line of sight (“projection ray”)



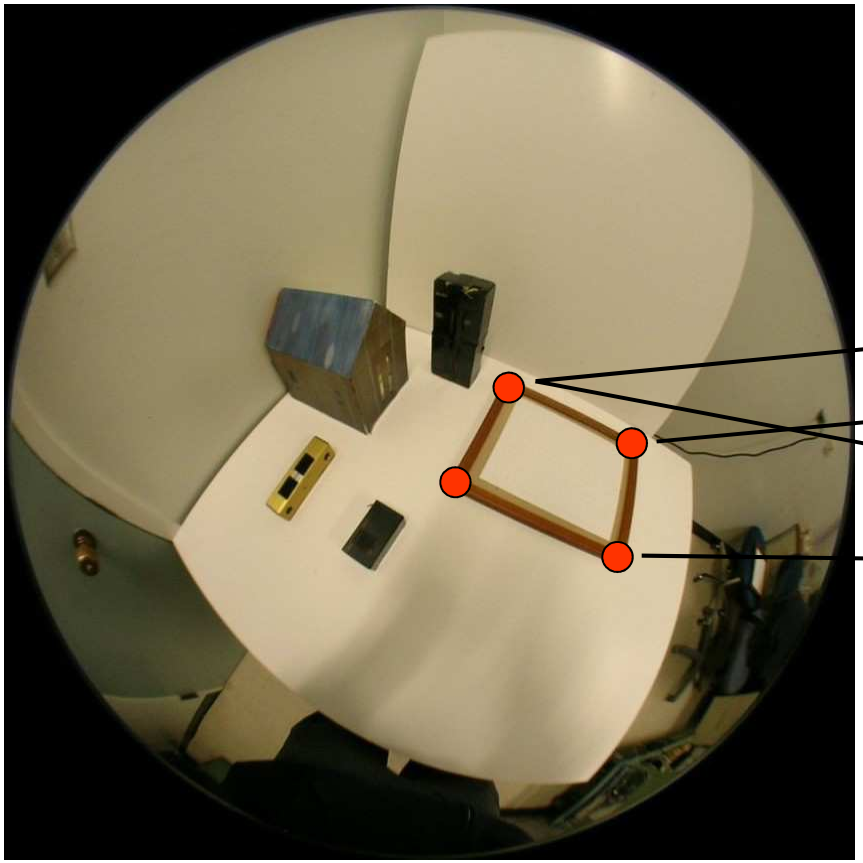
Motion estimation: compute motion such that matching rays intersect

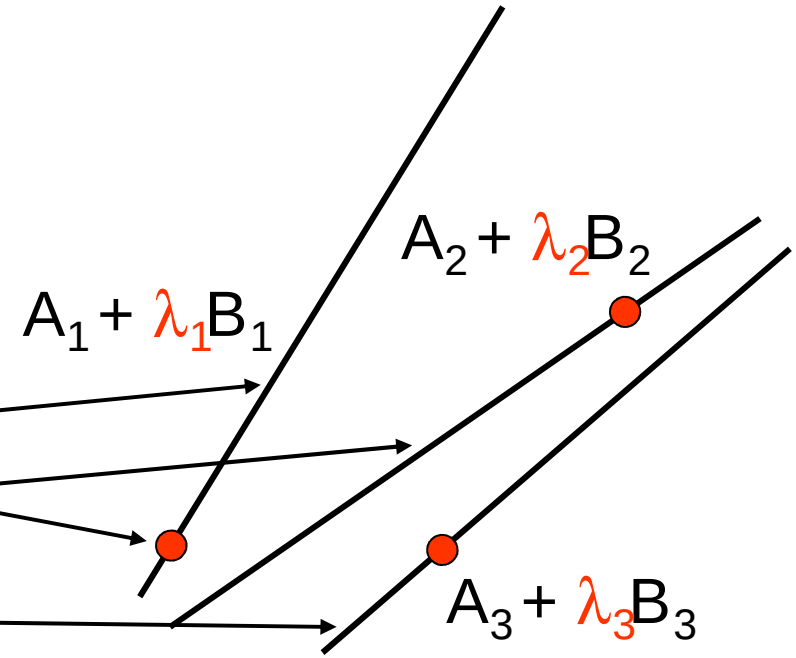


Triangulation / 3D Reconstruction



Pose estimation of known object



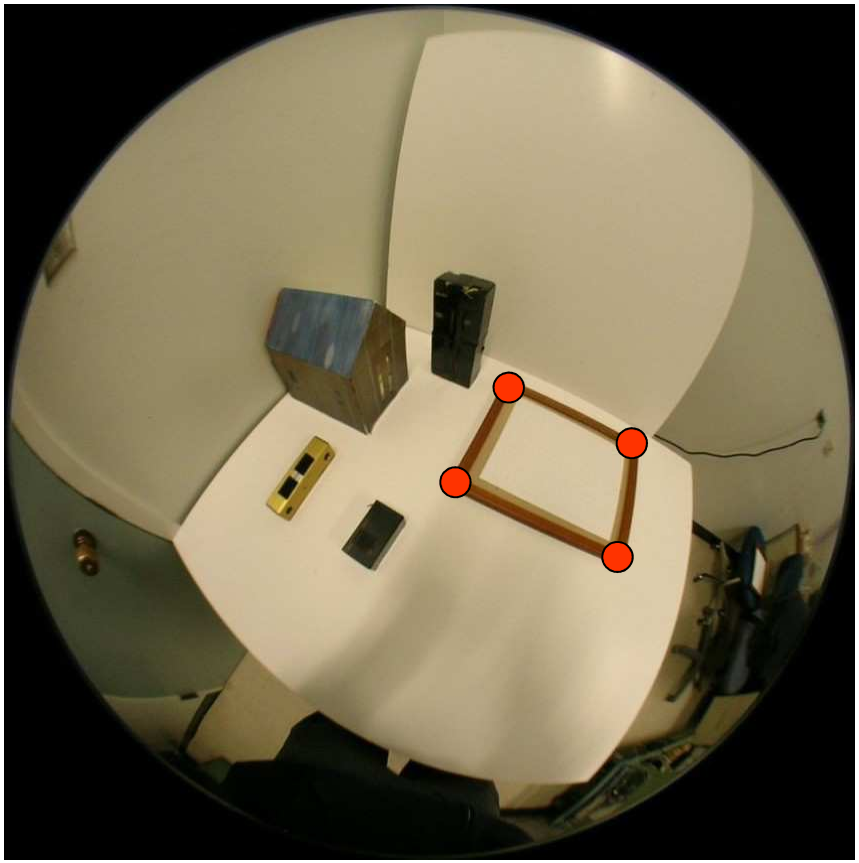


$$dist^2(A_1 + \lambda_1 B_1 , A_2 + \lambda_2 B_2) = d_{12}^2$$

$$dist^2(A_1 + \lambda_1 B_1 , A_3 + \lambda_3 B_3) = d_{13}^2$$

$$dist^2(A_2 + \lambda_2 B_2 , A_3 + \lambda_3 B_3) = d_{23}^2$$

Pose estimation of known object



- 3 quadratic equations:
  up to 8 solutions
- Central camera: solutions come in mirrored pairs (for a solution in front of the camera, another one behind exists too)
- Non-central camera: no such simple symmetry exists
- With 4 points, unique solution
  in general

[Chen-Chang-PAMI'04,Nistér-CVPR'04,Ramalingam-etal-OMNIVIS'04]

Structure-from-motion

Motion estimation

Motion estimation: unknown scene

- Pixel matches gives rise to ray matches
- Represent rays using Plücker coordinates
- Displacement for Plücker coordinates:

$$L'_1 = \begin{pmatrix} R & 0 \\ -[t]_x R & R \end{pmatrix}_{6x6} L_1$$

- Rays intersect if

$$L_2^T \begin{pmatrix} 0 & Id \\ Id & 0 \end{pmatrix}_{6x6} L'_1 = 0$$

→ Essential matrix

$$E = \begin{pmatrix} -[t]_x R & R \\ R & 0 \end{pmatrix}_{6x6}$$

$$L_2^T \, E \, L_1 = 0$$

Motion estimation

Motion estimation:

(1) Estimation of E (possible using linear equations: minimum 17 matches)

(2) Extraction of R and t from E (simple)

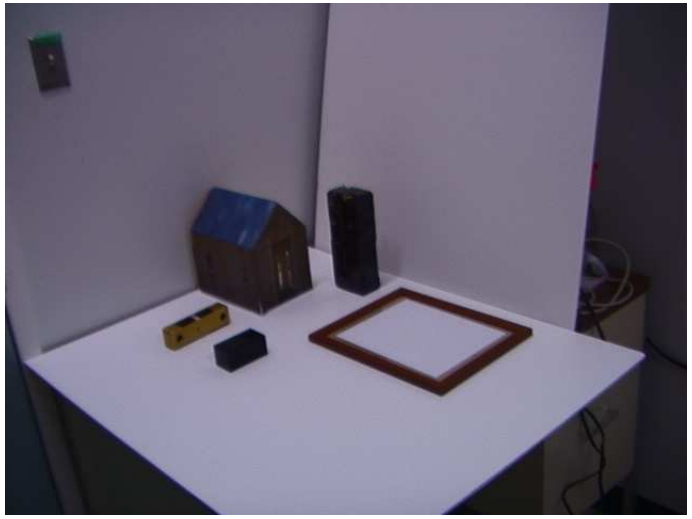
Note: scale of motion can be estimated if non-central cameras!
(but may be unreliable if cameras not very non-central)

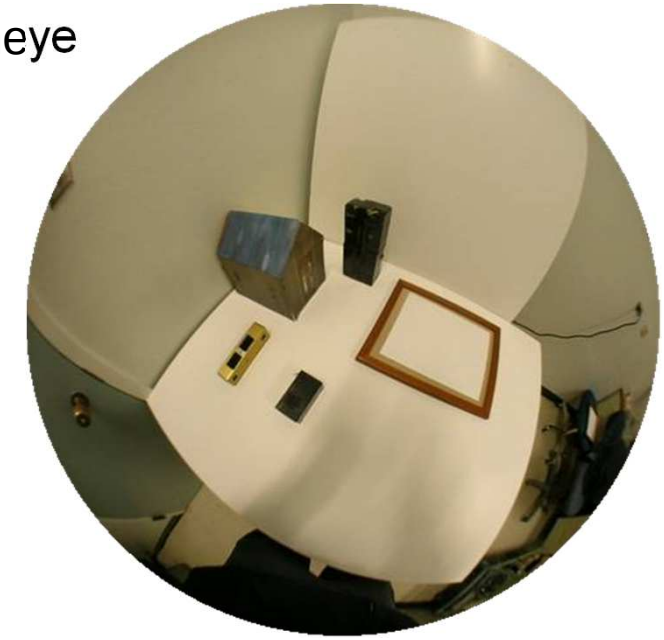
Variants for: axial, x-slit, central cameras

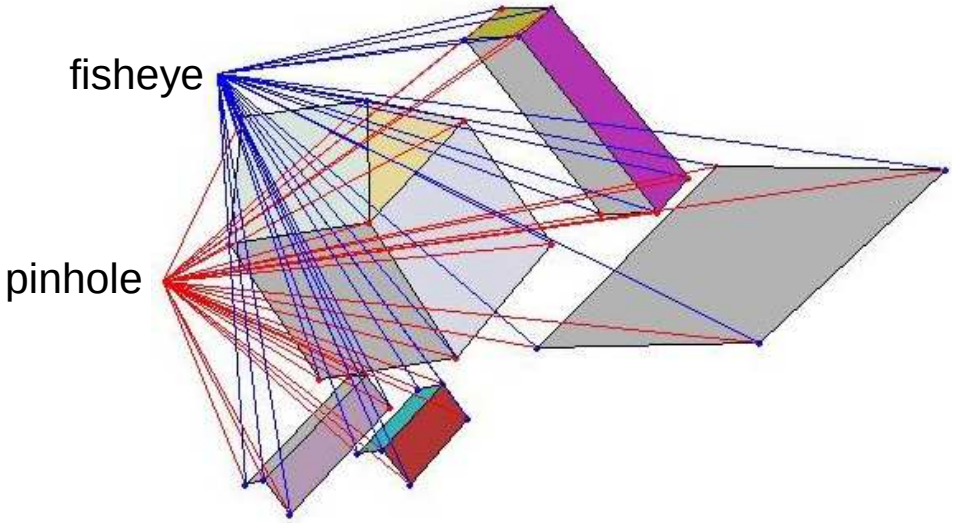
[Pless-CVPR'03,Sturm-etal-Bookchapter'06]

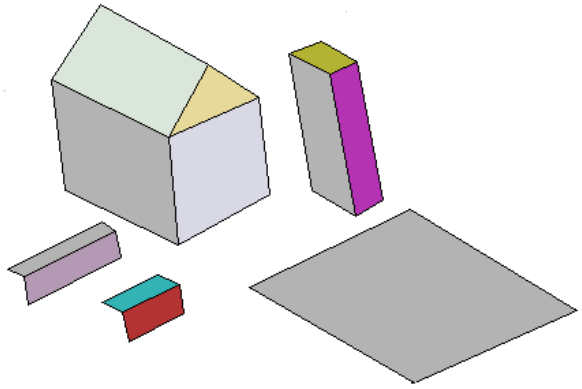
Structure-from-motion 3D reconstruction

Motion estimation and 3D from pinhole+fisheye









Structure-from-motion 3D reconstruction

Motion estimation and 3D from pinhole+fisheye

Perspective epipolar geometry:

- Epipolar line of a pixel *p* computed via the fundamental matrix: $v=Fp$

Such a parametric epipolar geometry exists for some omnidirectional cameras, e.g. para-catadioptric ones

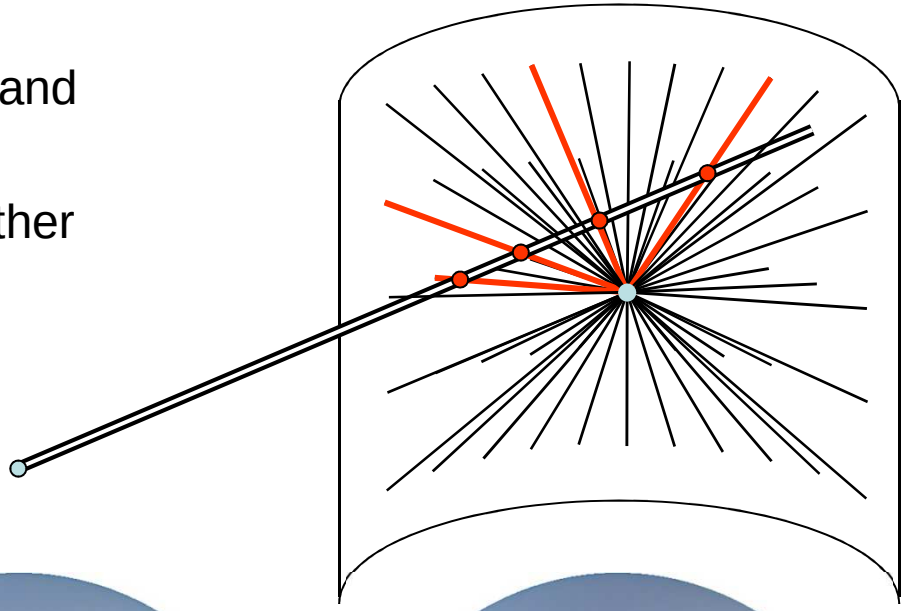
It also exists between cameras of different types, e.g. a stereo pair consisting of a perspective and a para-catadioptric camera

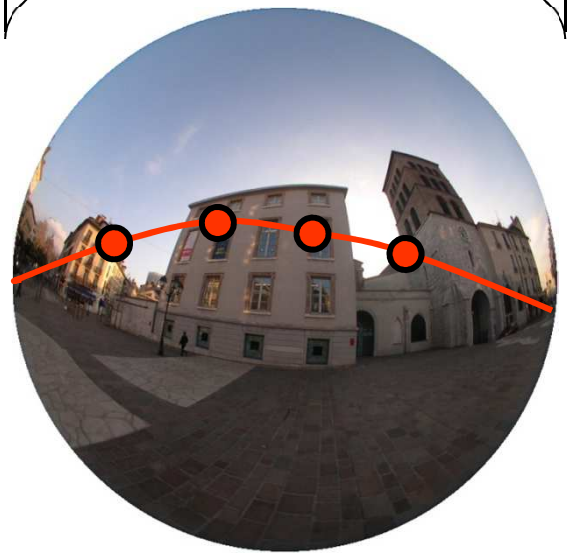
[Svoboda-etal-ECCV'98,Feldman-et-al-ICCV'05,Sturm-OMNIVIS'02]

Non-parametric epipolar geometry:

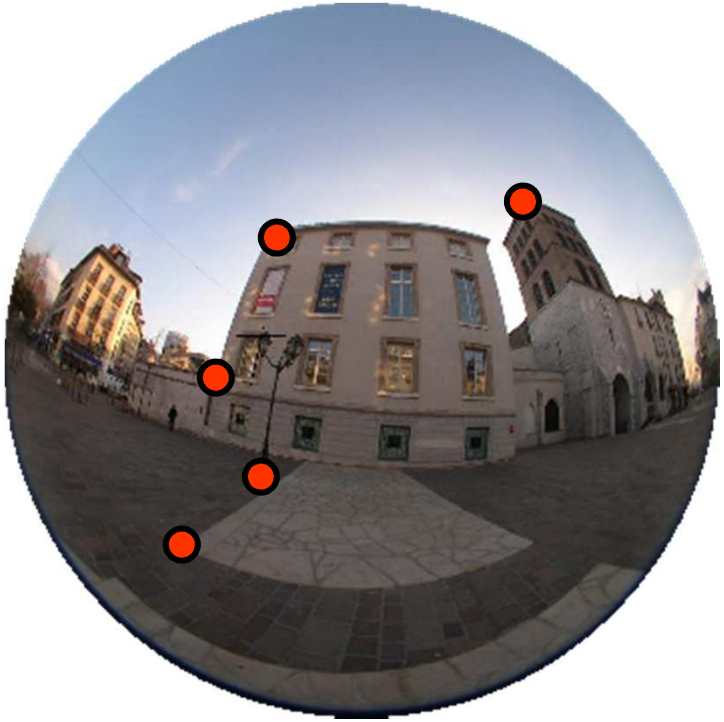
- Consider a pixel in one image and the associated projection ray
- Determine projection rays of other camera that cut that ray
- The associated pixels form an “epipolar curve”

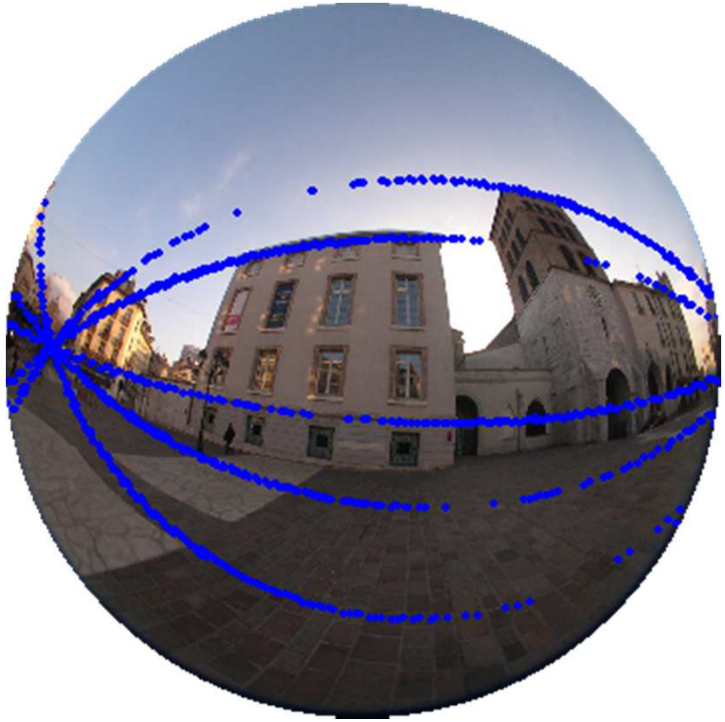
Here: illustration with central cameras, but concept is applicable to whatever camera, i.e. also non-central ones



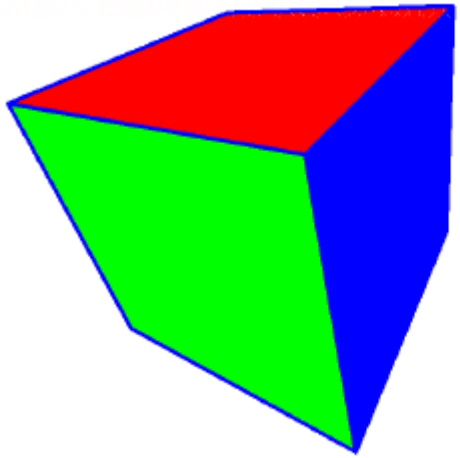


Non-parametric epipolar geometry:

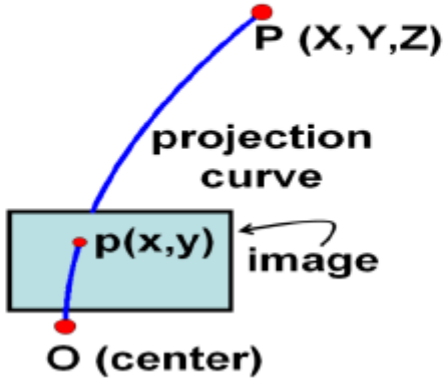




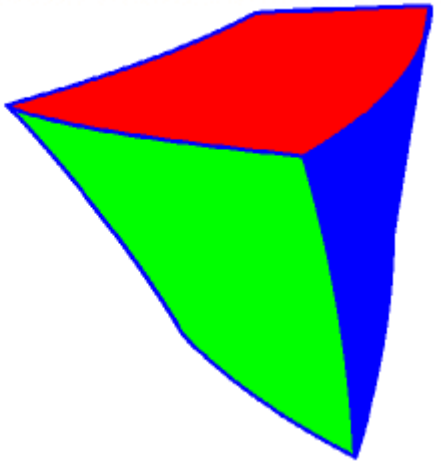
Beyond Generic Imaging Model: Piece-wise linear model (PLM)



A cube observed by a pinhole camera



A PCM where every pixel observes light in a parametric curve.



The same cube observed by a PLM, where each pixel observes light traveling in a conic curve.

Problem Statement

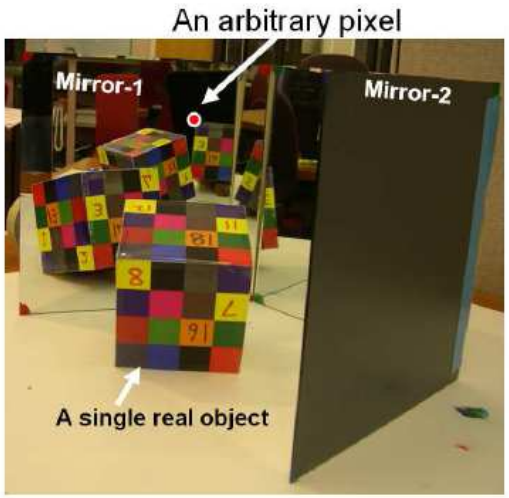
Assumption:

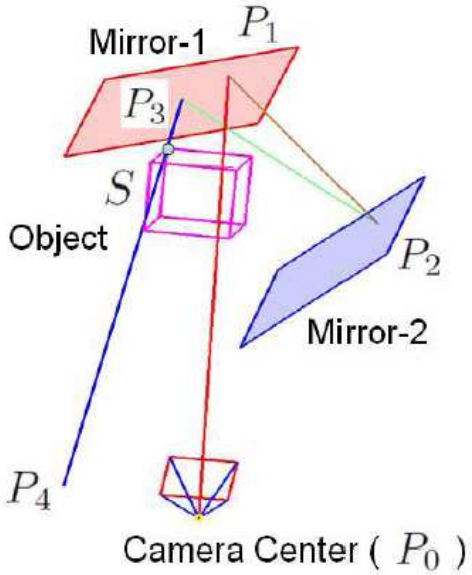
The camera is calibrated and every pixel is associated with a light-path.

Goal:

Given the light-paths for every pixel, can you extend the classical pose estimation and motion estimation algorithms for such camera models?

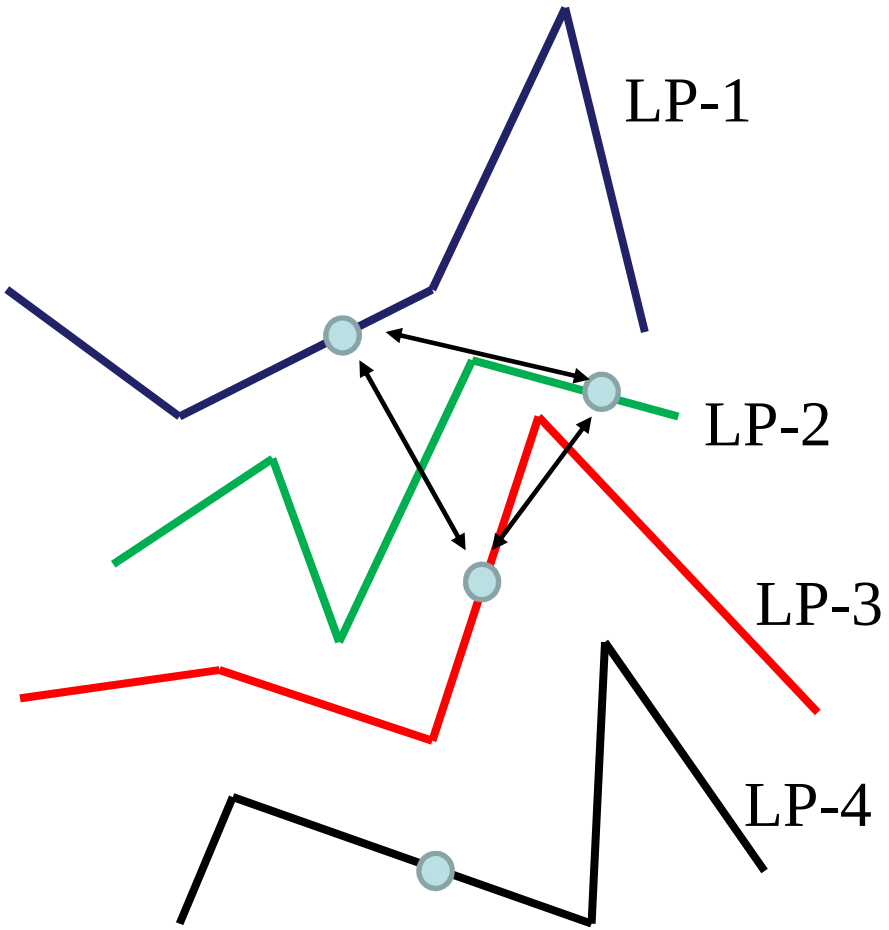
Problem Statement (Continued.)





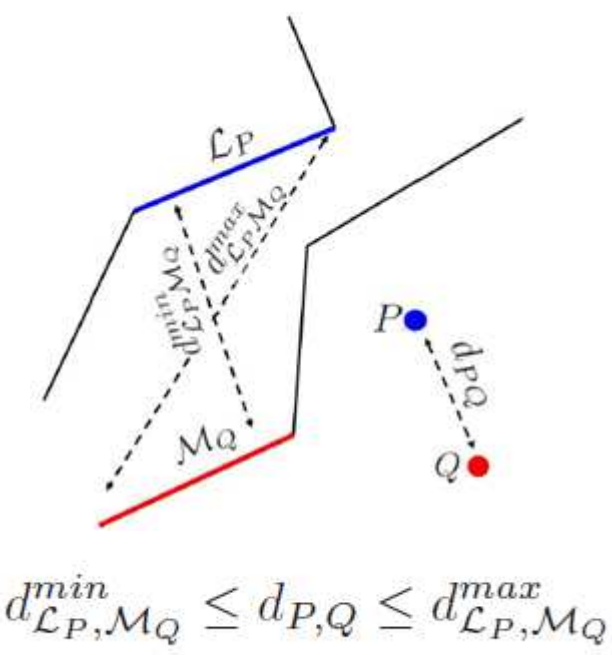
- Can you extend the pose and motion estimation algorithms without knowing explicit correspondence between the individual segment and the 3D point (For example, we do not know that the scene point S lies on the fourth segment of the light-path from the camera center.).

Basic Idea behind our pose estimation algorithm.

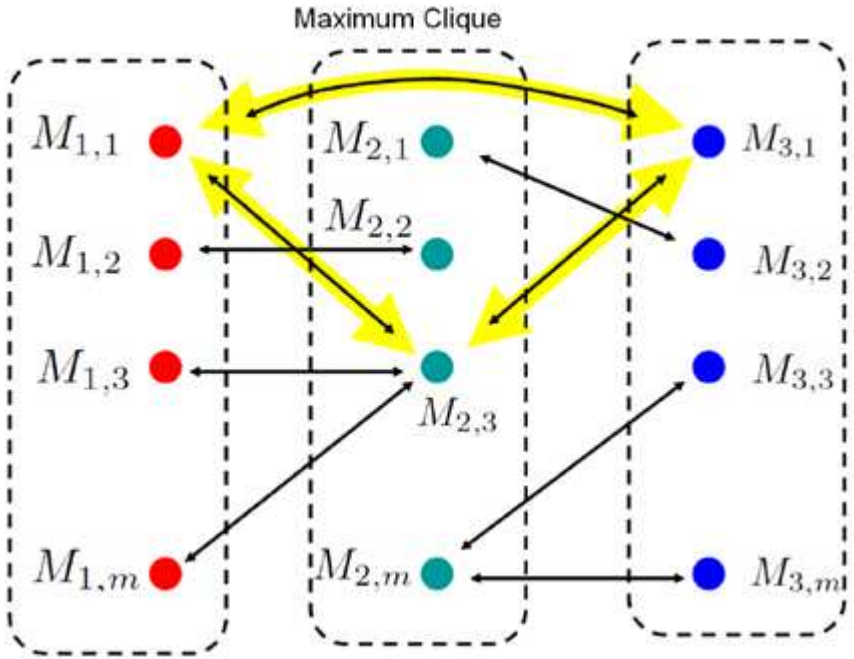


- Given 3D points and corresponding light-paths, our goal is to find a transformation (R,T) such that every 3D point lies on at least one segment of its corresponding light-path.
- This problem can be solved in a RANSAC framework with a brute-force search. This is very difficult when the number of segments is very large.

Pairwise Cheirality Constraint for efficient pose estimation

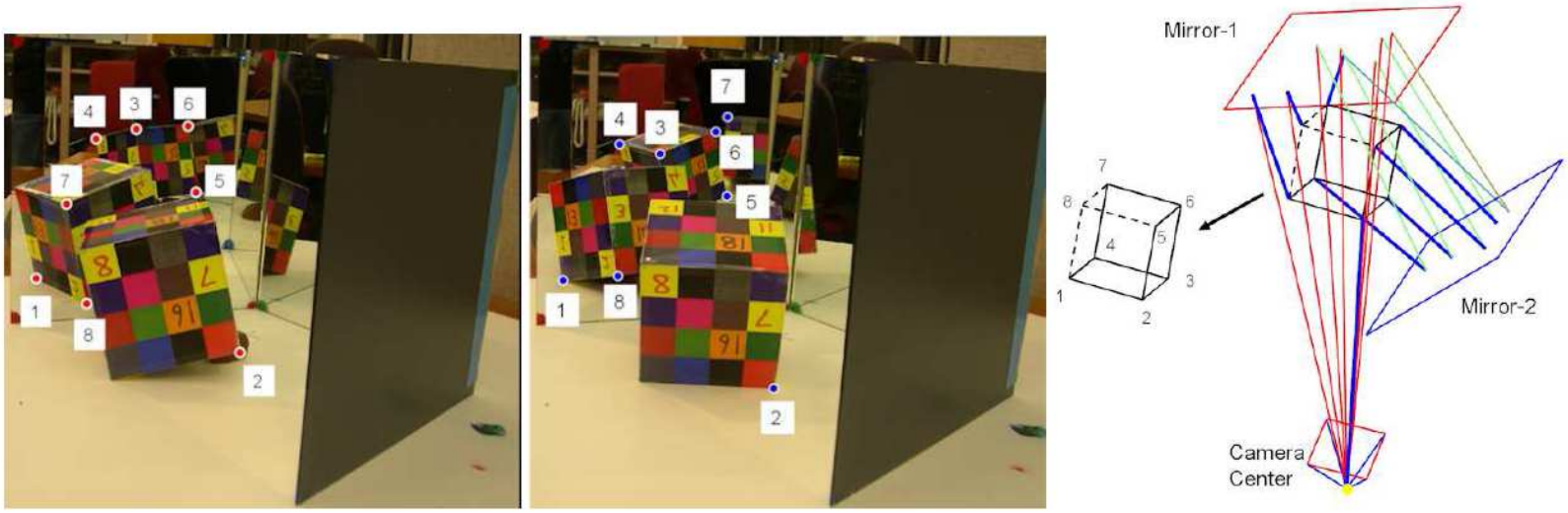


Pairwise cheirality constraint which does not allow a particular pairwise assignment of 3D points to line segments based on the minimum and maximum distance between the line segments.



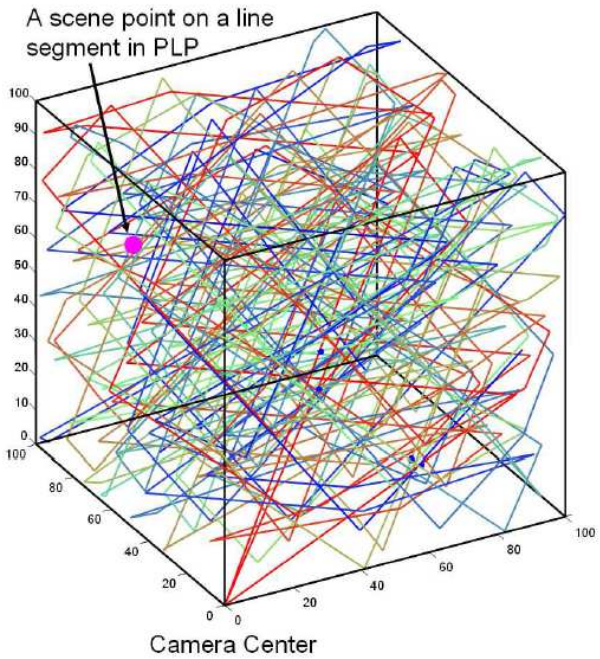
The number of candidate solutions for the 3D point – 3D segment assignments are given by all the maximum cliques of a tri-partite graph.

Motion Estimation Algorithm

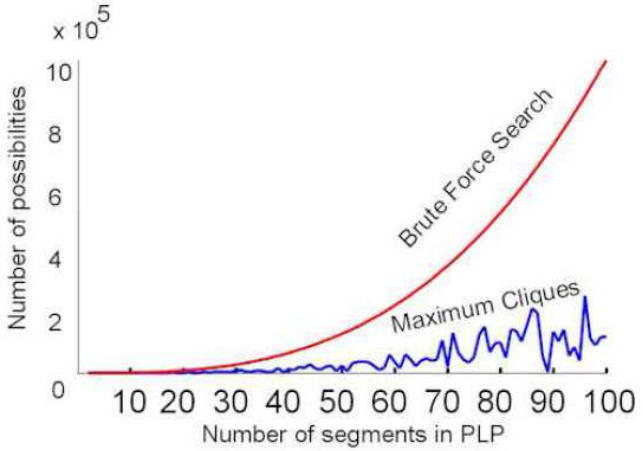


- Given two images and correspondences, the goal is compute the motion of the 3D object in the scene.
- The correspondences between the exact segment of the light-path is not given. This makes the problem very difficult to solve using a brute-force method.
- Similar to the pose estimation problem, we introduce a pair-wise cheirality constraint that reduces the search space significantly.

Simulations for pose estimation

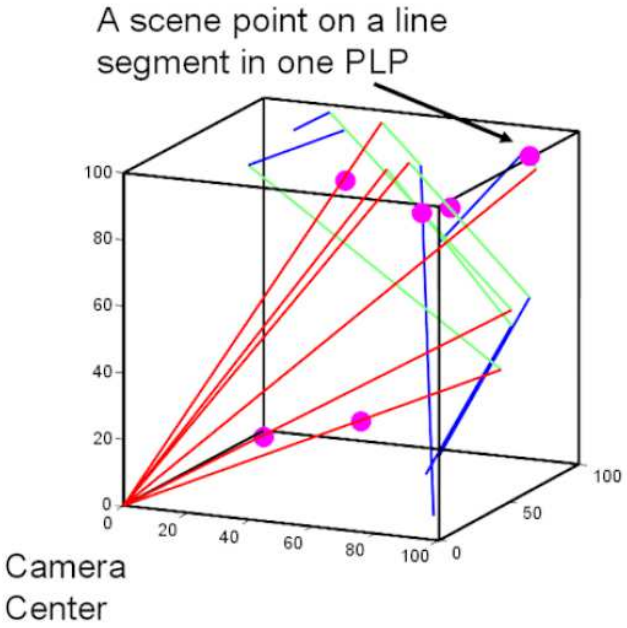


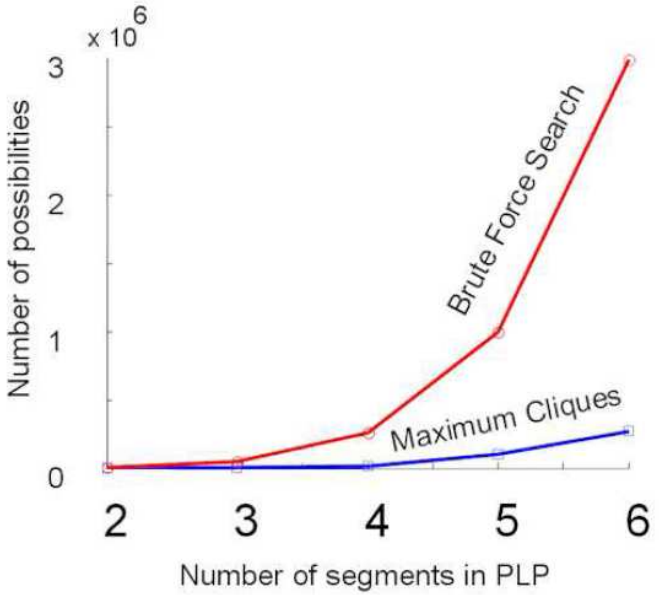
- A single light-path is traced inside a mirror-walled cube for up to 100 segments.
- The light-path is color-coded from red to blue (the initial segments are red in color and the final ones are blue in color).



- The number of candidate matches that need to be tested.
- Note that the number of candidates given by the enumeration of maximum cliques is much less than the brute-force search.

Simulations for Motion Estimation

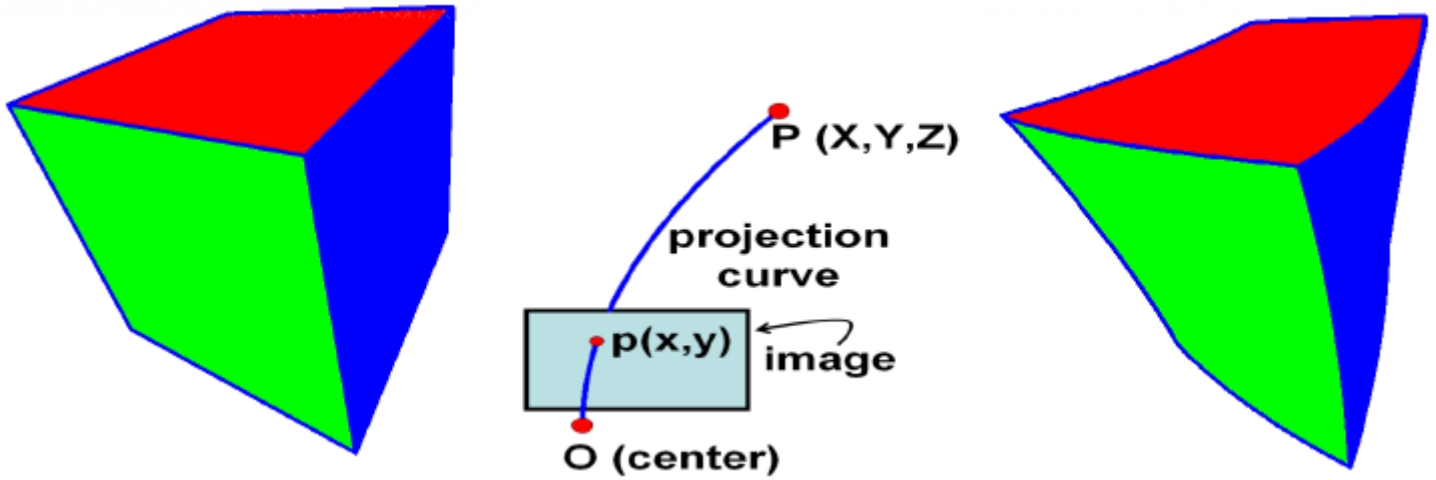




- We show 6 light-paths each containing 3 segments.

- Note that the number of candidates given by the enumeration of maximum cliques is much less than the brute-force search.

Parametric Curve Modeling



- Every pixel is associated with a parametric curve represented in the following form:

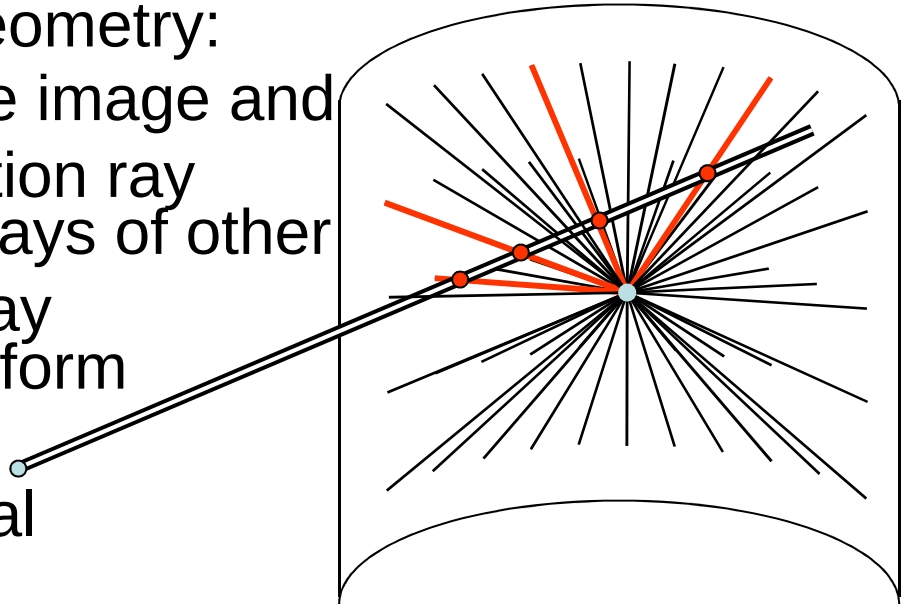
$$\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} at^2 + bt \\ ct \\ t \end{pmatrix}$$

- Pose and motion estimation algorithms can be analytically solved.
- The solutions are based on solving non-linear set of polynomial equations.

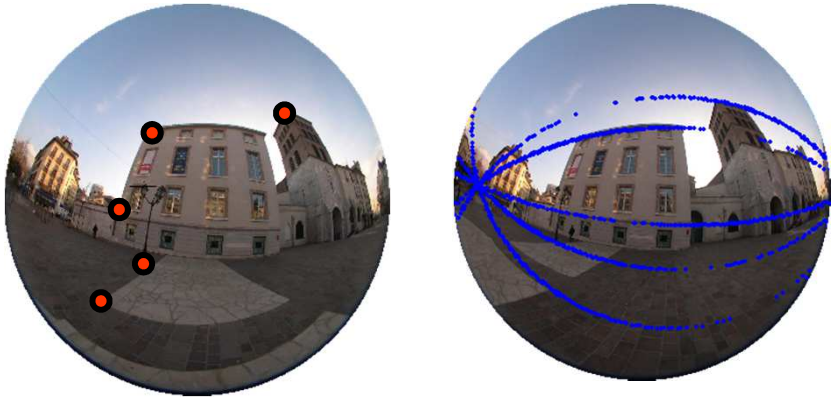
Epipolar geometry for generalized cameras

Non-parametric epipolar geometry:

- Consider a pixel in one image and the associated projection ray
- Determine projection rays of other camera that cut that ray
- The associated pixels form an “epipolar curve”

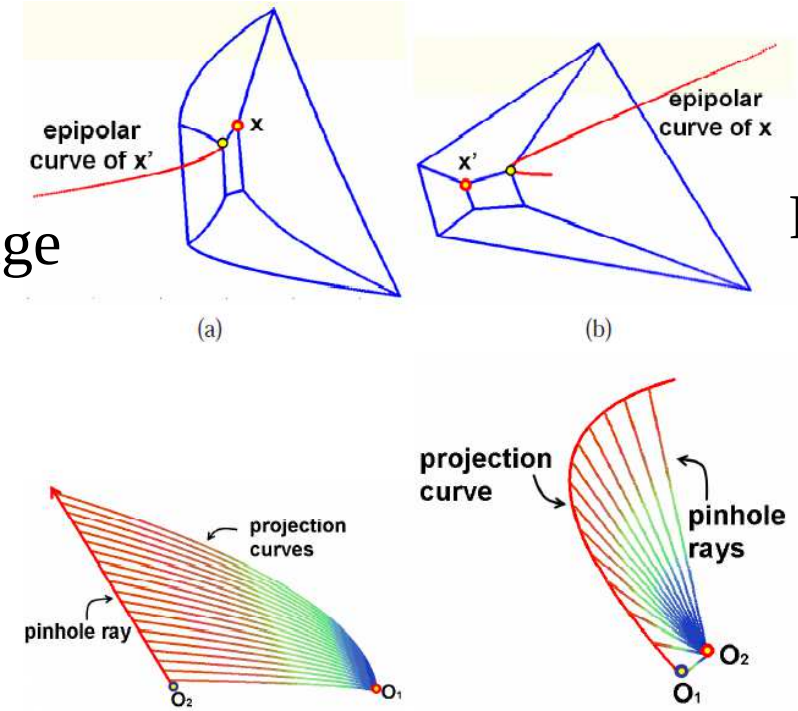


Here: illustration with central cameras, but concept is applicable to whatever camera, i.e. also non-central ones



Epipolar Geometry in PCM

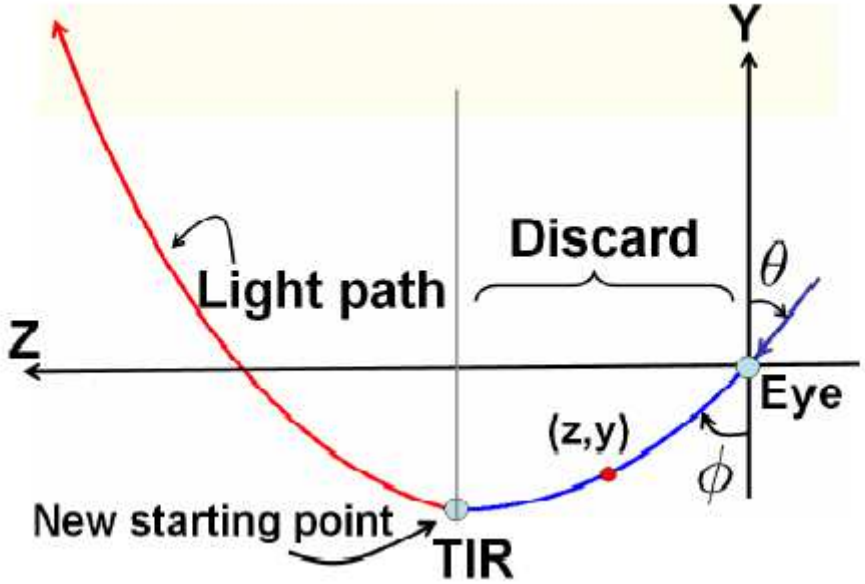
A PCM Image



Perspective Image

- Synthesized images of the inside of a cube using a pinhole camera and a PCM respectively.
- Epipolar curve on the PCM image is shown. The curve is obtained by collecting the pixels whose projection curves intersect a straight line in space.

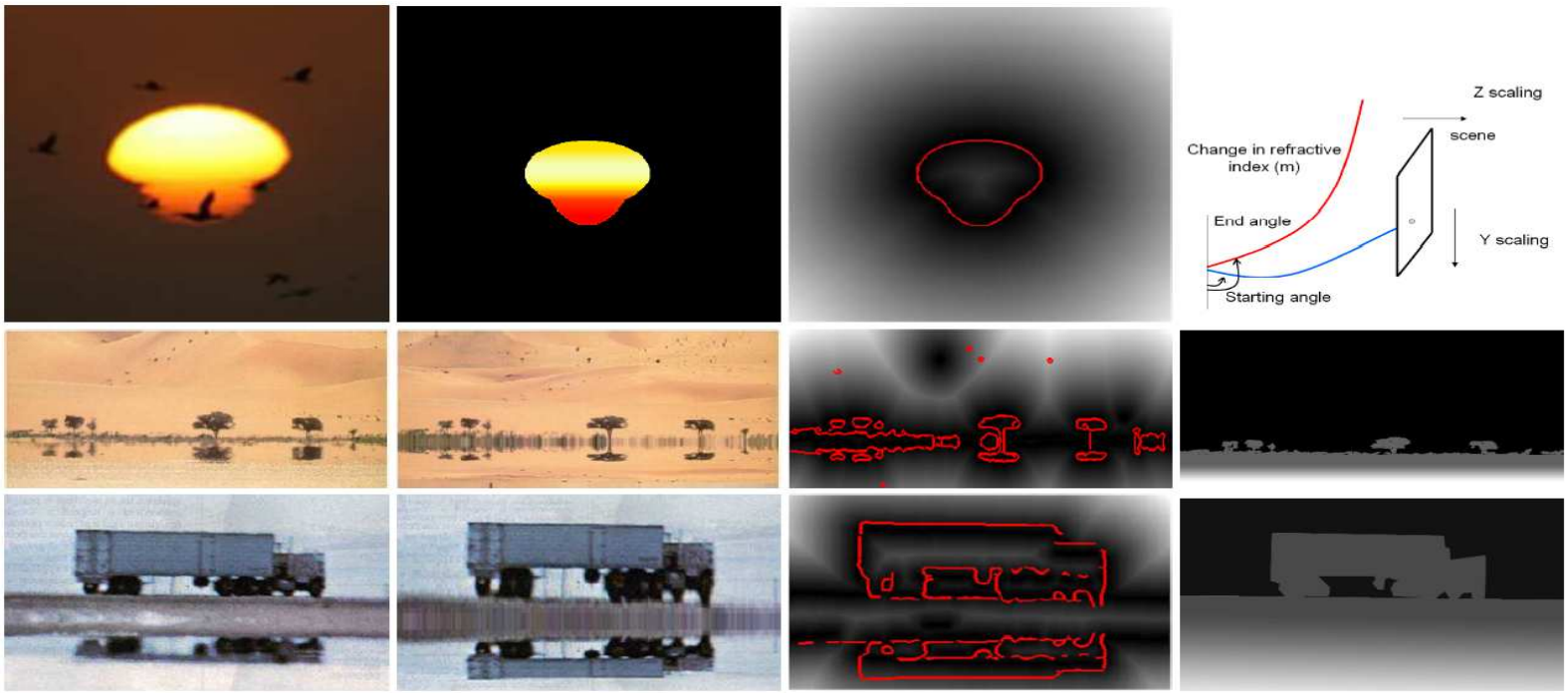
Physics based modeling of Mirages



- The path of a single light ray during the formation of mirages. Note that the ray undergoes total internal reflection (TIR) in the air.
- Using Taylor approximation, we model the projection ray in a mirage as a conic.

$$\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} f \\ dt + e \\ at^2 + bt + c \end{pmatrix}$$

Synthesize and Compare Algorithm



Real images of Mirages

Synthesized images of mirages

Chamfer Matching

Estimation of planes in the scene

- The scene is modeled as a set of vertical and horizontal planes.
- We search for the parameters by synthesizing mirages and comparing them with real ones.

Multi-view geometry for perspective images:

- Consider points (or other features) in images
- Which geometric constraints exist that tell if points are potential matches?
  - 2 images: epipolar geometry (fundamental/essential matrix)

$$\mathbf{q}_2^T \mathbf{E}\, \mathbf{q}_1 = 0$$

  - 3 or 4 images: trifocal and quadrifocal tensors

$$\sum_{i_1=1}^{3} \sum_{i_2=1}^{3} \cdots \sum_{i_n=1}^{3} q_{1,i_1} q_{2,i_2} \cdots q_{n,i_n} T_{i_1,i_2,\cdots,i_n} = 0$$

Multi-view geometry for generic imaging model:

- Constraints between *projection rays*

$$\sum_{i_1=1}^{6} \sum_{i_2=1}^{6} \cdots \sum_{i_n=1}^{6} L_{1,i_1} L_{2,i_2} \cdots L_{n,i_n} T_{i_1,i_2,\cdots,i_n} = 0$$

Perspective multi-view geometry:

- Consider points $\mathbf{q}_i$ in *n* images with projection matrices P_i
- They are potential matches if scalars λ_i and a 3D point $\mathbf{Q}$ exist with:

$$\lambda_i \mathbf{q}_i = \mathsf{P}_i \mathbf{Q}, \quad \forall i = 1 \cdots n$$

- This can be written as:

$$\underbrace{\begin{pmatrix} \mathsf{P}_1 & \mathbf{q}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathsf{P}_2 & \mathbf{0} & \mathbf{q}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathsf{P}_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{q}_n \end{pmatrix}}_{\mathsf{M}} \begin{pmatrix} \mathbf{Q} \\ -\lambda_1 \\ -\lambda_2 \\ \vdots \\ -\lambda_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- Existence of null-vector implies rank-deficiency of M
- M is of size 3n 4+n
  → all submatrices (4+n) × (4+n) have zero determinant

$$\underbrace{\begin{pmatrix} \mathsf{P}_1 & \mathbf{q}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathsf{P}_2 & \mathbf{0} & \mathbf{q}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathsf{P}_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{q}_n \end{pmatrix}}_{\mathsf{M}}$$

- Determinants of submatrices can be written as:

$$\sum_{i_1=1}^{3} \sum_{i_2=1}^{3} \cdots \sum_{i_n=1}^{3} q_{1,i_1} q_{2,i_2} \cdots q_{n,i_n} T_{i_1,i_2,\cdots,i_n} = 0$$

- where: matching tensors $\mathbf{T}$ depend exactly on the projection matrices P_i
  - *n* = 2: fundamental (essential) matrix
  - *n* = 3: trifocal tensors
  - *n* = 4: quadrifocal tensors
- Uses of matching tensors:
  - Matching constraints
  - Useful for motion estimation from image correspondences

Multi-view geometry for generic imaging model:

- Projection rays are represented by Plücker coordinates:
  - let $\mathbf{A}$ and $\mathbf{B}$ be any 2 points on a 3D line
  - Plücker coordinates can be defined as:

$$\mathbf{L} = \begin{pmatrix} A_4B_1 - A_1B_4 \\ A_4B_2 - A_2B_4 \\ A_4B_3 - A_3B_4 \\ A_3B_2 - A_2B_3 \\ A_1B_3 - A_3B_1 \\ A_2B_1 - A_1B_2 \end{pmatrix}$$

  - they are independent of the choice of $\mathbf{A}$ and $\mathbf{B}$

[Sturm-CVPR'05]

- Consider projection rays $\mathbf{L}_i$ for *n* calibrated cameras
- For the moment, parameterize rays by two points $\mathbf{A}_i$ and $\mathbf{B}_i$ each.
- Pose of cameras is parameterized as

$$\mathsf{P}_i = \begin{pmatrix} \mathsf{R}_i & \mathbf{t}_i \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}$$

- Rays are potential matches if scalars λ_i and μ_i and a 3D point $\mathbf{Q}$ exist with:

$$\lambda_i \mathbf{A}_i + \mu_i \mathbf{B}_i = \mathsf{P}_i \mathbf{Q}, \quad \forall i = 1 \cdots n$$

- Rays are potential matches if scalars λ_i and μ_i and a 3D point $\mathbf{Q}$ exist with:

$$\lambda_i \mathbf{A}_i + \mu_i \mathbf{B}_i = \mathsf{P}_i \mathbf{Q}, \quad \forall i = 1 \cdots n$$

- This can be written as:

$$\underbrace{\begin{pmatrix} \mathsf{P}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \cdots & \mathbf{0} & \mathbf{0} \\ \mathsf{P}_2 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathsf{P}_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_n & \mathbf{B}_n \end{pmatrix}}_{\mathsf{M}} \begin{pmatrix} \mathbf{Q} \\ -\lambda_1 \\ -\mu_1 \\ \vdots \\ -\lambda_n \\ -\mu_n \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$$

- Existence of null-vector implies rank-deficiency of M
- M is of size 4n 4+2n
  → all submatrices (4+2n) × (4+2n) have zero determinant

$$\underbrace{\begin{pmatrix} \mathsf{P}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \cdots & \mathbf{0} & \mathbf{0} \\ \mathsf{P}_2 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathsf{P}_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_n & \mathbf{B}_n \end{pmatrix}}_{\mathsf{M}}$$

- When developping determinants of submatrices, coordinates of points $\mathbf{A}_i$ and $\mathbf{B}_i$ appear in terms of this form:

$$A_{i,j}B_{i,k} - A_{i,k}B_{i,j}$$

$\rightarrow$ Plücker coordinates of $\mathbf{L}_i$

- We obtain matching constraints of the form:

$$\sum_{i_1=1}^{6}\sum_{i_2=1}^{6}\cdots\sum_{i_n=1}^{6} L_{1,i_1}L_{2,i_2}\cdots L_{n,i_n}T_{i_1,i_2,\cdots,i_n} = 0$$

- Matching tensors $\mathbf{T}$ depend on pose matrices P_i

- Like for perspective images, matching tensors exist for 2, 3, and 4 cameras
- Example: two views

$$\mathrm{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & A_{1,1} & B_{1,1} & 0 & 0 \\ 0 & 1 & 0 & 0 & A_{1,2} & B_{1,2} & 0 & 0 \\ 0 & 0 & 1 & 0 & A_{1,3} & B_{1,3} & 0 & 0 \\ 0 & 0 & 0 & 1 & A_{1,4} & B_{1,4} & 0 & 0 \\ R_{11} & R_{12} & R_{13} & t_1 & 0 & 0 & A_{2,1} & B_{2,1} \\ R_{21} & R_{22} & R_{23} & t_2 & 0 & 0 & A_{2,2} & B_{2,2} \\ R_{31} & R_{32} & R_{33} & t_3 & 0 & 0 & A_{2,3} & B_{2,3} \\ 0 & 0 & 0 & 1 & 0 & 0 & A_{2,4} & B_{2,4} \end{pmatrix}$$

of size 8x8

M is rank-deficient, thus singular

$\rightarrow$ matching constraint is: $\det \mathrm{M} = \mathrm{L}_2^{\mathrm{T}} \underbrace{\begin{pmatrix} -[\mathrm{t}]_x \mathrm{R} & \mathrm{R} \\ \mathrm{R} & 0 \end{pmatrix}}_{\text{essential matrix}} \mathrm{L}_1 = 0$

- Matching tensors for non-central cameras are of size 6×6×…
- Reduced parameterizations exist:
  - Axial cameras: 5×5×…
  - X-slit cameras: 4×4×…
  - Central cameras: 3×3×…
- Matching tensors between cameras of different types are straightforward, e.g.:
  - Essential matrix of a non-central and a central camera: 6×3

Summary for structure-from-motion:

- When calibrated cameras are considered, an SfM problem (pose, motion, …) can be solved with one and the same algorithm, whatever the type of camera
- But: results are not optimal (e.g. in the sense of reprojection errors)
  → methods are useful for embedding in RANSAC, but should be followed by bundle adjustment if good accuracy required
- Extension of structure-from-motion theory from perspective to general camera model
- Some missing pieces, e.g. matching tensors for line images