Performance evaluation of the five-point relative pose with emphasis on planar scenes

Siniša Šegvić, Gerald Schweighofer and Axel Pinz

Vision-based Measurement Group,

Institute of Electrical Measurement and Measurement Signal Processing,

Graz University of Technology

Relative pose (relative orientation): the mutual position of the two cameras imaging a common scene

- 3D rotation + translation up to scale (5 DOF)
- absolute scale can not be recovered by monocular vision

Relative pose (relative orientation): the mutual position of the two cameras imaging a common scene

- 3D rotation + translation up to scale (5 DOF)
- absolute scale can not be recovered by monocular vision
- important **building block** in structure and motion estimation

Relative pose (relative orientation): the mutual position of the two cameras imaging a common scene

- 3D rotation + translation up to scale (5 DOF)
- absolute scale can not be recovered by monocular vision
- important **building block** in structure and motion estimation

Applications:

- autonomous navigation and/or mapping
- offline and online 3D modelling
- augmented reality
- compression
- automated inspection

We address **performance evaluation** of the novel 5pt algorithm

- 5pt algorithm performance on **planar scenes**
- comparison with homography (planar, near-planar)
- comparison with **conditioned** 8pt algorithms (near-planar)

We address **performance evaluation** of the novel 5pt algorithm

- 5pt algorithm performance on **planar scenes**
- comparison with homography (planar, near-planar)
- comparison with **conditioned** 8pt algorithms (near-planar)

Contents:

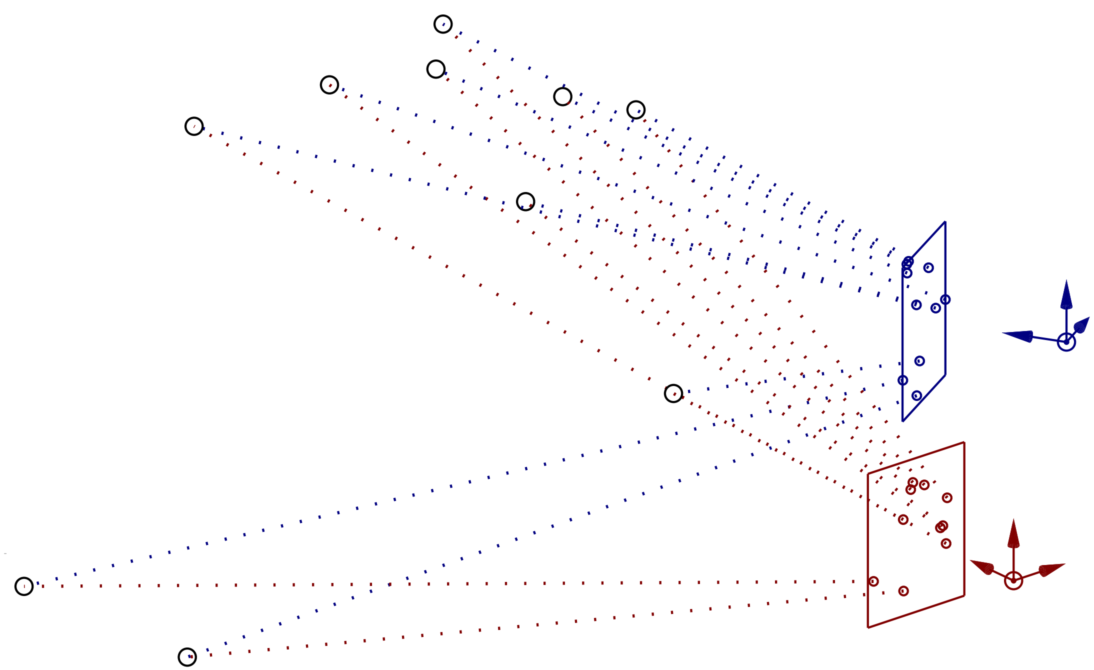
- The problem description
- The three considered algorithms
- Experimental setup
- Results
- Conclusion

The relative pose is recovered from **image correspondences**:

- many correspondence approaches, all seek a compromise between genuine matches and outliers
- the main approaches: **wide-baseline matching**, **tracking**
- the **subpixel** matching accuracy essential

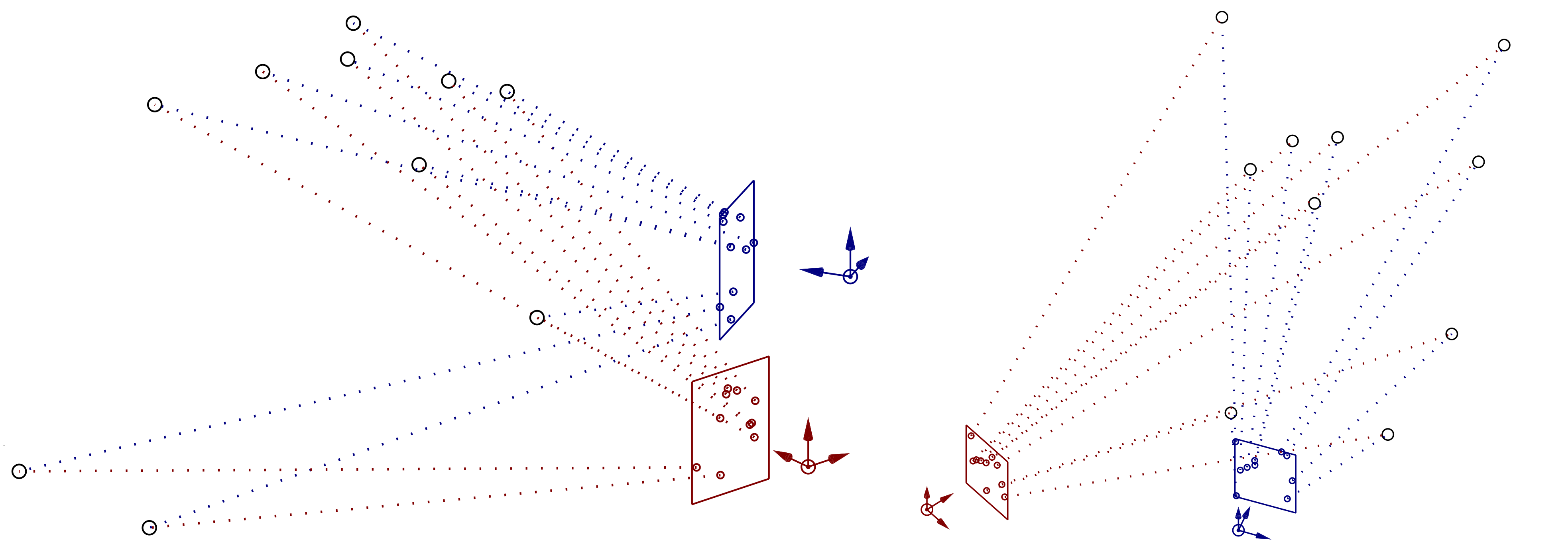
The relative pose is recovered from **image correspondences**:

- many correspondence approaches, all seek a compromise between genuine matches and outliers
- the main approaches: **wide-baseline matching**, **tracking**
- the **subpixel** matching accuracy essential



The relative pose is recovered from **image correspondences**:

- many correspondence approaches, all seek a compromise between genuine matches and outliers
- the main approaches: **wide-baseline matching**, **tracking**
- the **subpixel** matching accuracy essential



Three main contexts:

- **minimal** case, with exact solutions (RANSAC loop)
- **overconstrained** case: optimizing an algebraic criterion (closed-form re-estimation on the set of inliers)
- **iterative refinement**: optimizing a nonlinear criterion (robust ML solution, may imply recovering structure as well)

Three main contexts:

- [CF] **minimal** case, with exact solutions (RANSAC loop)
- [CF] **overconstrained** case: optimizing an algebraic criterion (closed-form re-estimation on the set of inliers)
- **iterative refinement**: optimizing a nonlinear criterion (robust ML solution, may imply recovering structure as well)

Three main contexts:

- [CF] **minimal** case, with exact solutions (RANSAC loop)
- [CF] **overconstrained** case: optimizing an algebraic criterion (closed-form re-estimation on the set of inliers)
- **iterative refinement**: optimizing a nonlinear criterion (robust ML solution, may imply recovering structure as well)

What can be recovered in **closed-form** from two views?

- the **essential** matrix† (epipolar geometry)
  $\mathbf{q}_{i\mathrm{B}}^{\top} \cdot \mathbf{E} \cdot \mathbf{q}_{i\mathrm{A}} = 0$ ($\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$, decomposition unique)
- the **homography** matrix‡ (geometry of a planar scene)
  $\mathbf{H} \cdot \mathbf{q}_{i\mathrm{A}} \sim \mathbf{q}_{i\mathrm{B}}$ ($\mathbf{H} \sim \mathbf{R} + \frac{1}{d}\mathbf{T} \cdot \mathbf{n}^{\top}$, decomposition not unique)
- the affine epipolar geometry, affine homography (not considered here)

The **eight point** (8pt) algorithm:

- recovers the essential matrix as a solution to the homogeneous linear system $\mathbf{A}_{n\times 9}\cdot\mathbf{e}=\mathbf{0}$
- requires at least **8** correspondences in general position
- badly conditioned by default (forward bias), can be improved in the overconstrained case
- does not work with planes: “wrong” matrices satisfy the epipolar constraint.

The **five point** algorithm:

- epipolar geometry + the “calibrated” constraint:
  $2 \cdot \mathbf{E}\mathbf{E}^T\mathbf{E} - trace(\mathbf{E}\mathbf{E}^T)\mathbf{E} = 0$
  - operates on matrices E_i obtained as the lowest four null-vectors of $\mathbf{A}_{n \times 9}$
  - the linear combination $\mathbf{E} = a \cdot \mathbf{E_6} + b \cdot \mathbf{E_7} + c \cdot \mathbf{E_8} + d \cdot \mathbf{E_9}$ plugged into the calibrated constraint
  - the resulting cubic system solved for a, b, c, d
  - up to ten solutions (needs disambiguation)

The **five point** algorithm:

- epipolar geometry + the “calibrated” constraint:
  $2 \cdot \mathbf{E}\mathbf{E}^T\mathbf{E} - trace(\mathbf{E}\mathbf{E}^T)\mathbf{E} = 0$
  - operates on matrices E_i obtained as the lowest four null-vectors of $\mathbf{A}_{n \times 9}$
  - the linear combination $\mathbf{E} = a \cdot \mathbf{E_6} + b \cdot \mathbf{E_7} + c \cdot \mathbf{E_8} + d \cdot \mathbf{E_9}$ plugged into the calibrated constraint
  - the resulting cubic system solved for a, b, c, d
  - up to ten solutions (needs disambiguation)
- can operate with only five correspondences
- very good results in minimal cases (5 + 1 points)
- can operate on planar scenes
  (but not with the plane at inifinity!)

The **linear recovery** of the homography:

- requires **4** or more correpondences, well conditioned

The **linear recovery** of the homography:

- requires **4** or more correpondences, well conditioned

The issue of **planar ambiguity**:

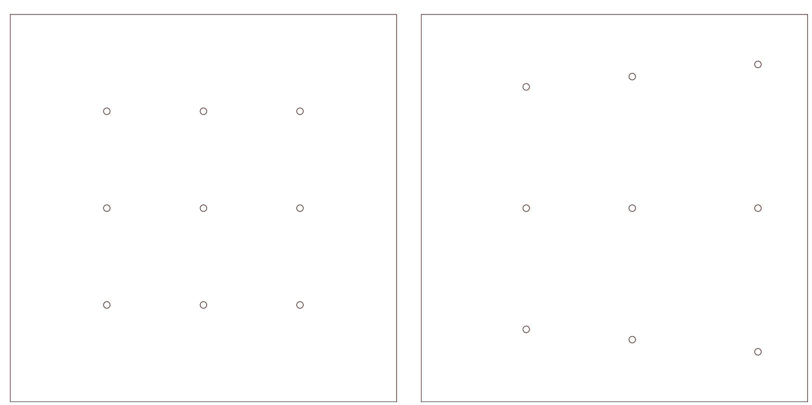
- each homography gives rise to **8** motion hypoheses
- the visibility constraint eliminates 6 or 7 of the 8
- the ambiguity arises when all imaged points are closer to one of the two cameras!

The **linear recovery** of the homography:

- requires **4** or more correpondences, well conditioned

The issue of **planar ambiguity**:

- each homography gives rise to **8** motion hypoheses
- the visibility constraint eliminates 6 or 7 of the 8
- the ambiguity arises when all imaged points are closer to one of the two cameras!

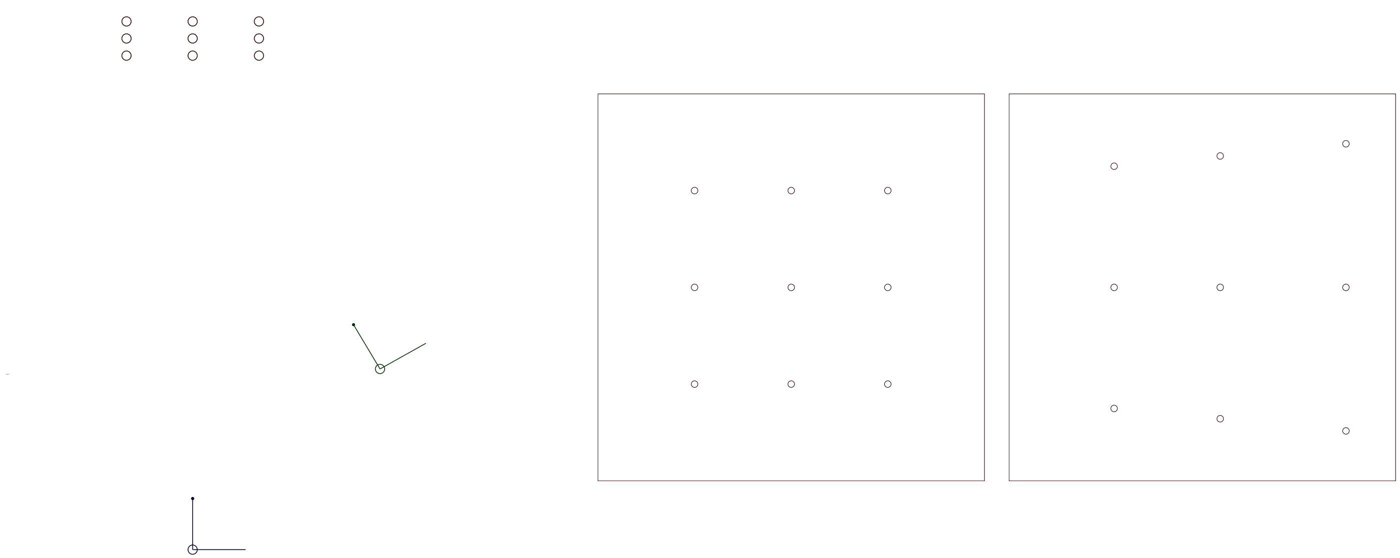


The **linear recovery** of the homography:

- requires **4** or more correpondences, well conditioned

The issue of **planar ambiguity**:

- each homography gives rise to **8** motion hypoheses
- the visibility constraint eliminates 6 or 7 of the 8
- the ambiguity arises when all imaged points are closer to one of the two cameras!

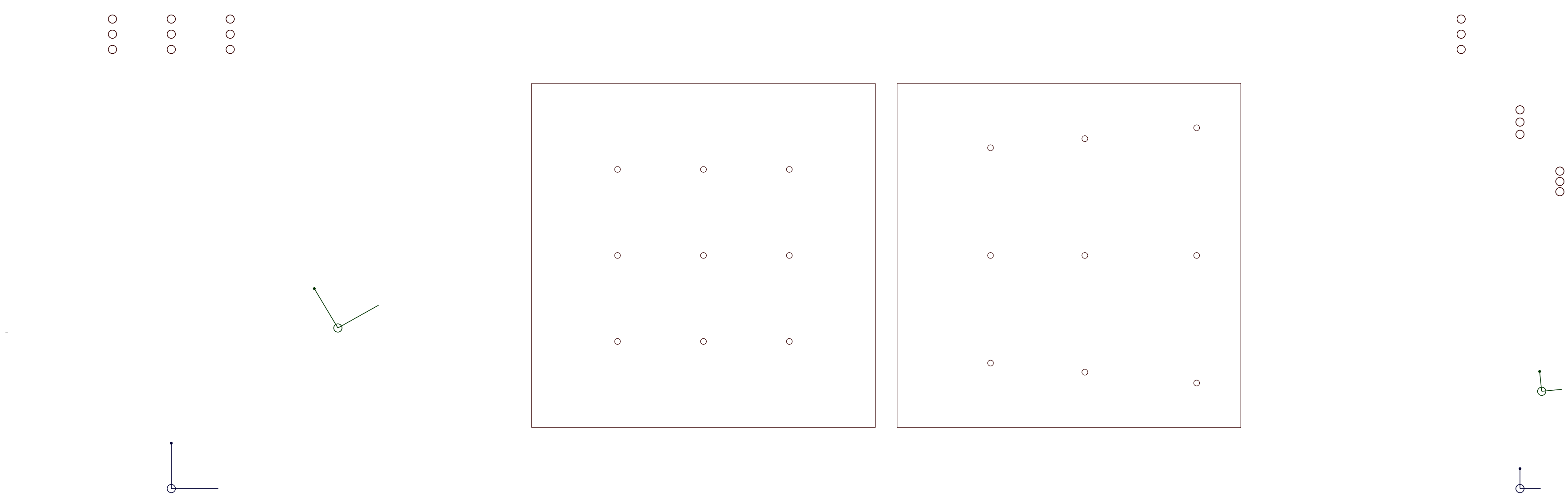


The **linear recovery** of the homography:

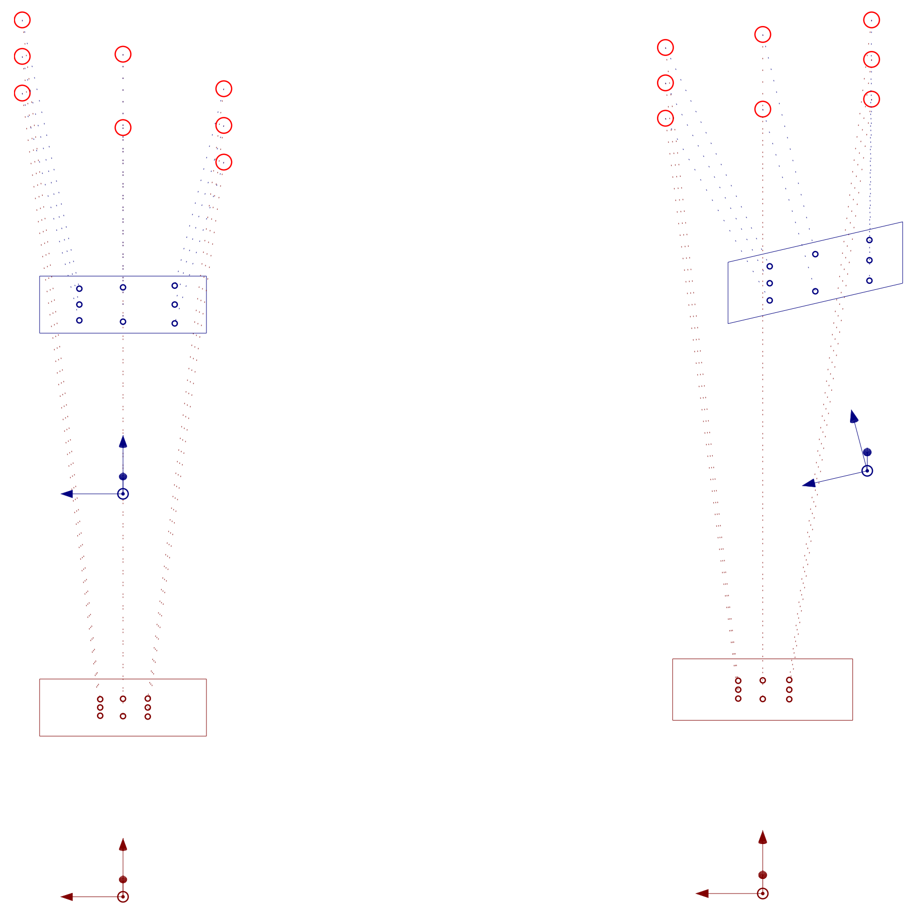
- requires **4** or more correpondences, well conditioned

The issue of **planar ambiguity**:

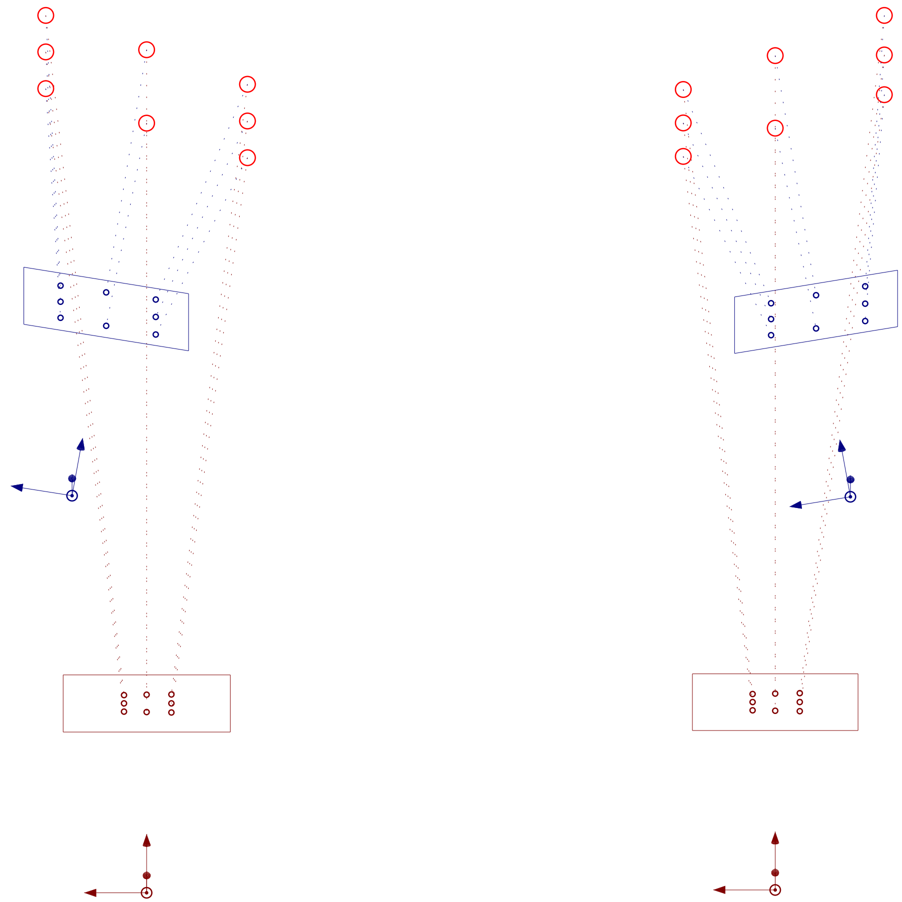
- each homography gives rise to **8** motion hypoheses
- the visibility constraint eliminates 6 or 7 of the 8
- the ambiguity arises when all imaged points are closer to one of the two cameras!



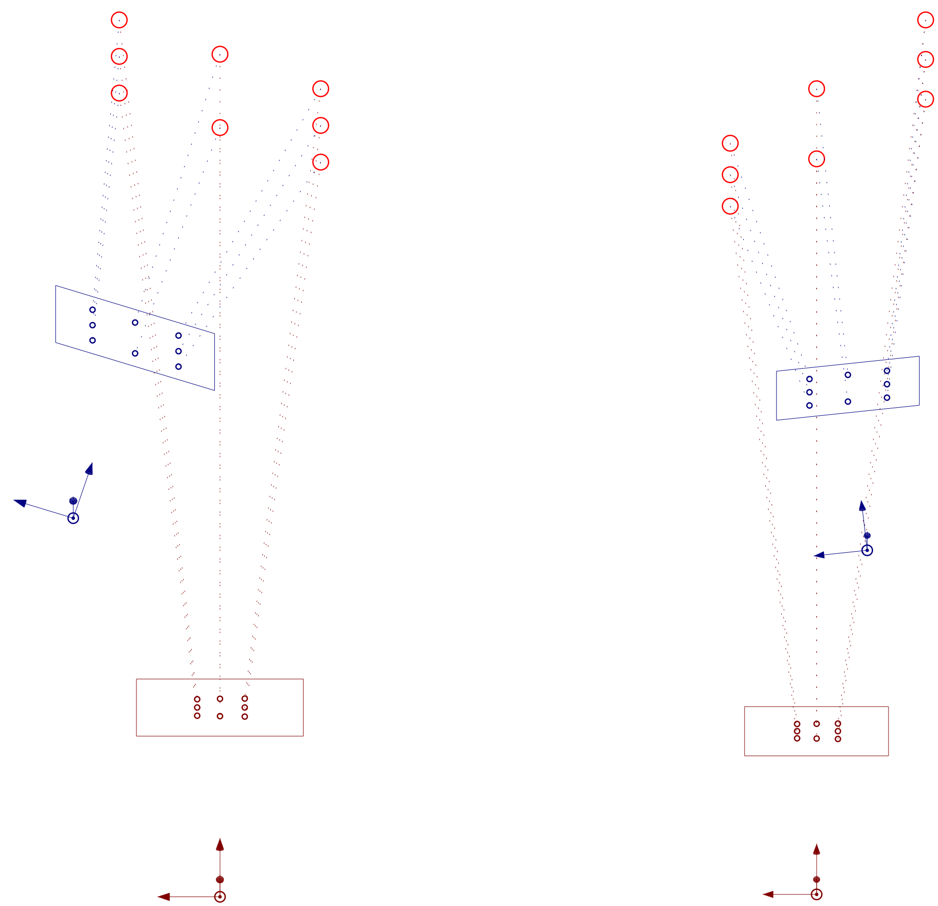
θ=00°:



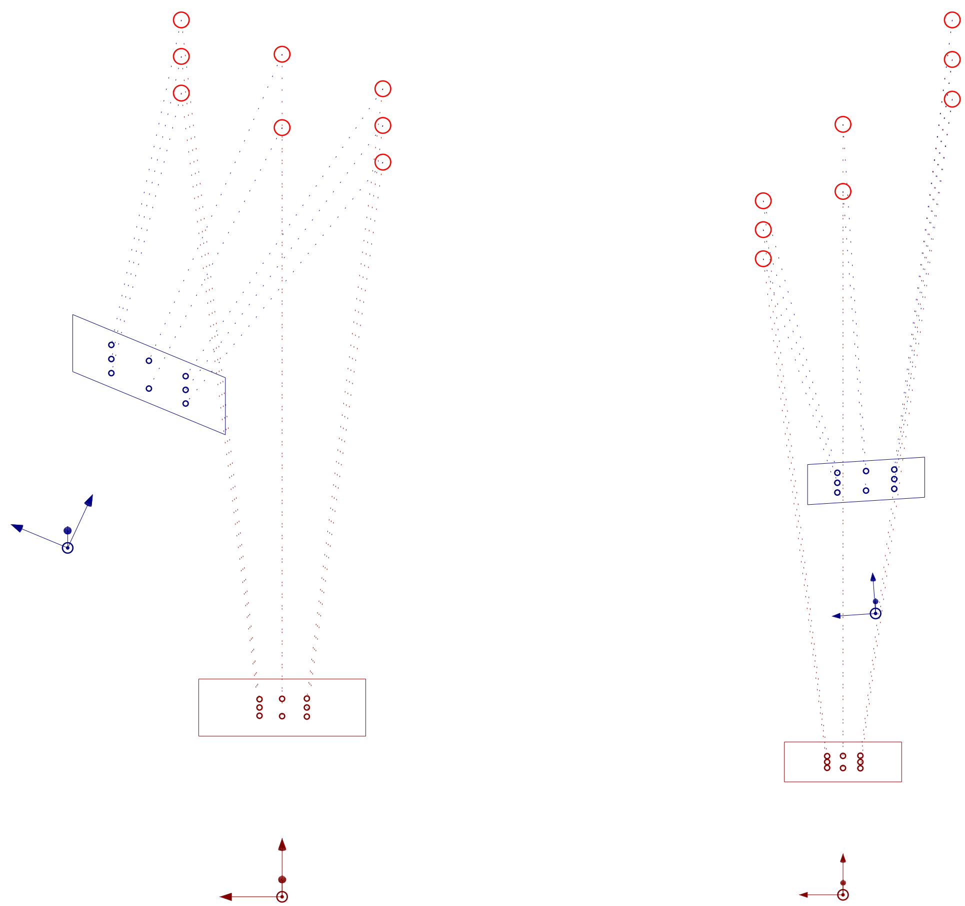
$\theta = 10^{\circ}$:



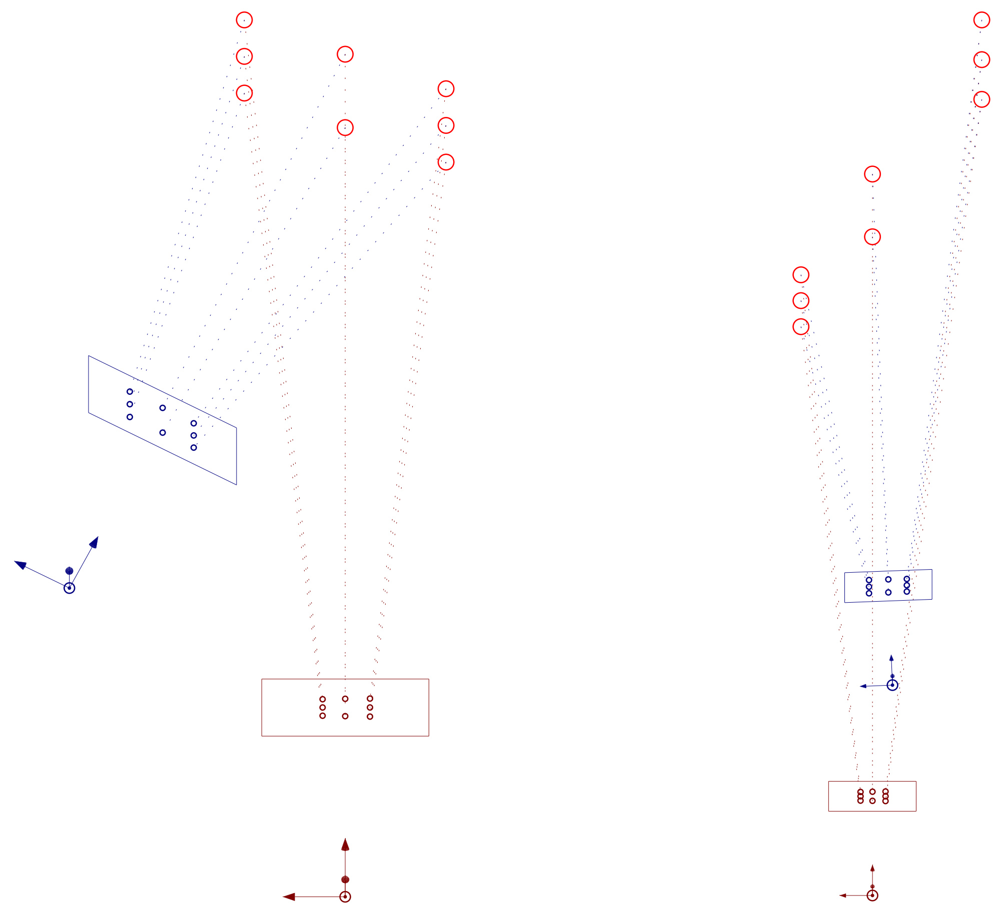
θ=20°:



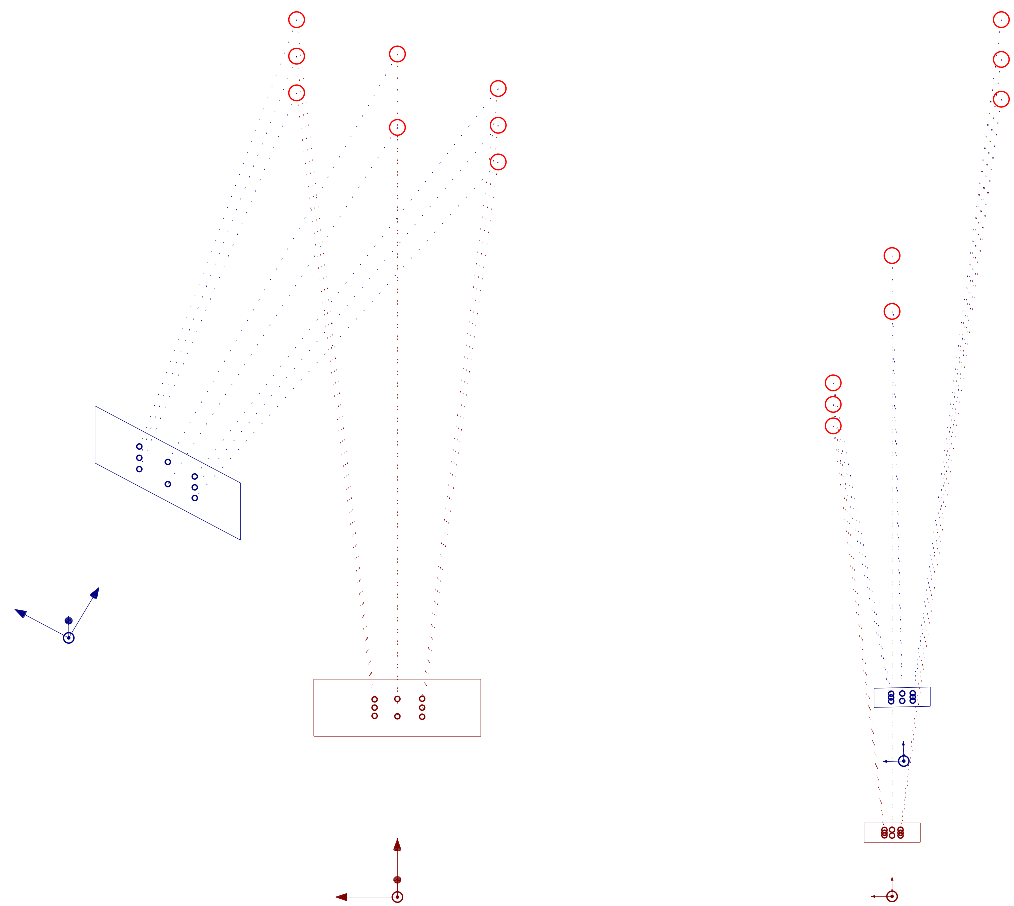
θ=30°:



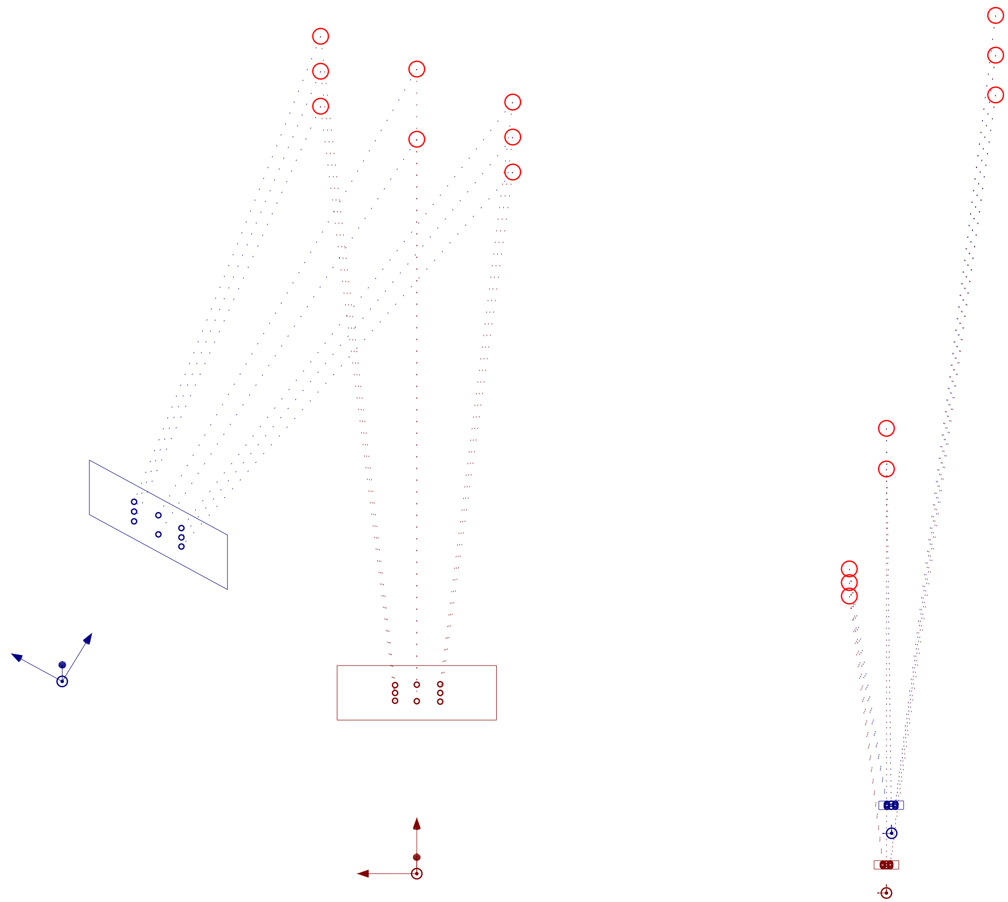
θ=40°:



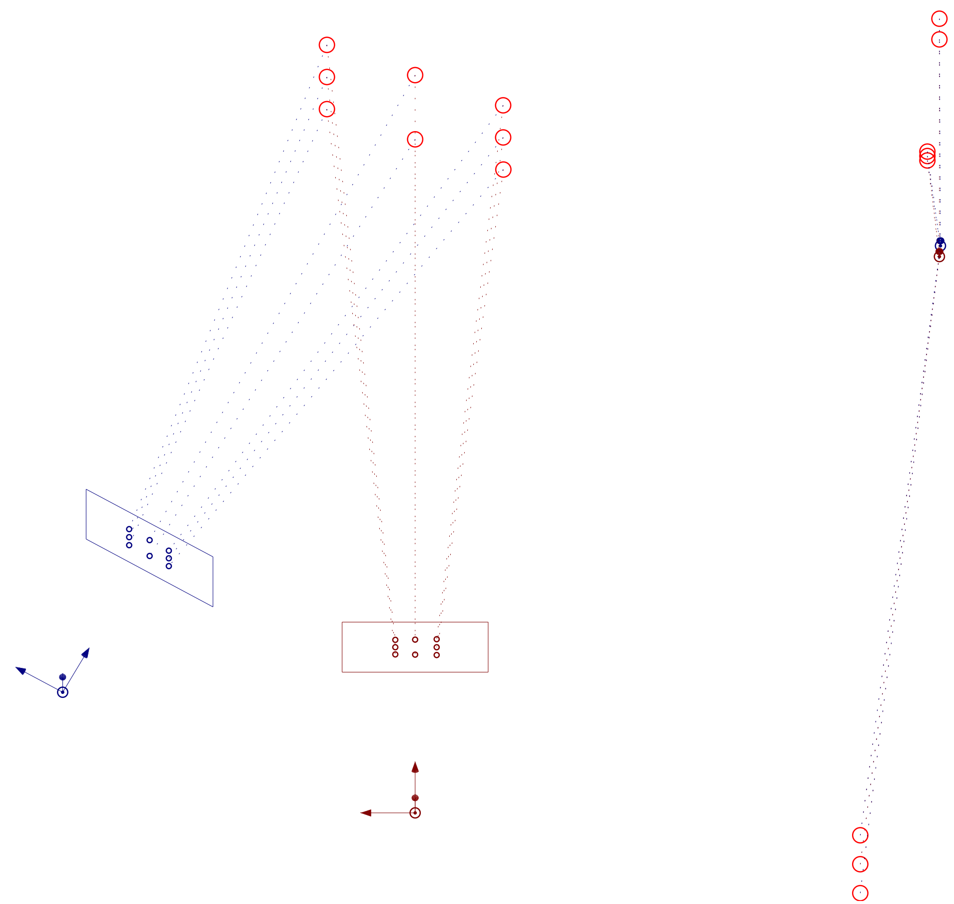
$\theta = 50^{\circ}$:



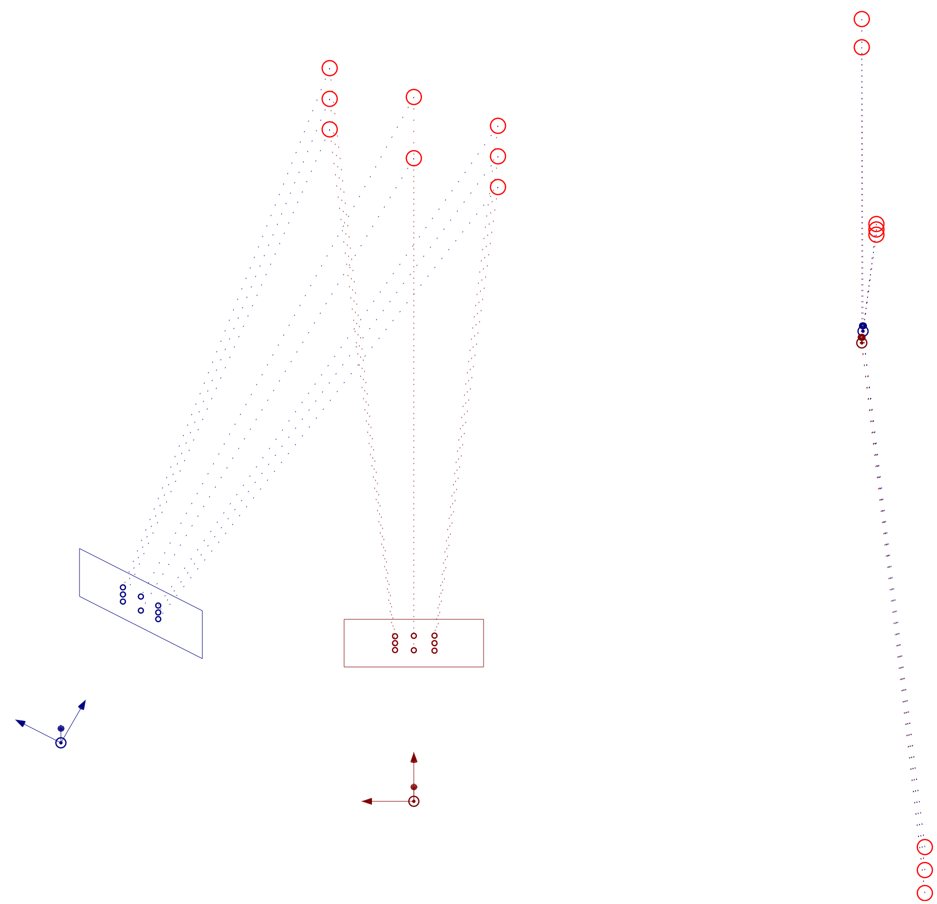
θ=60°:



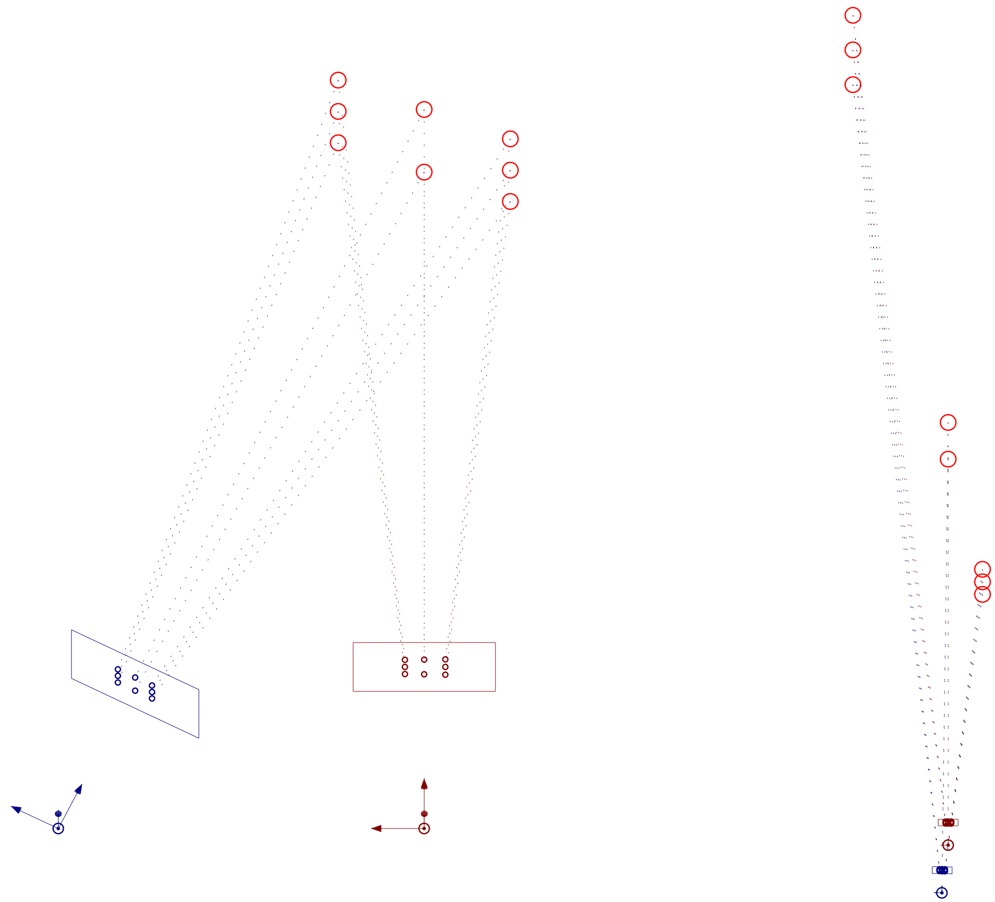
θ=70°:



θ=80°:



θ=90°:



Improving the **numeric conditioning** of the 8pt algorithm:

- the standard 8pt algorithm:
  $\min |\mathbf{A} \cdot \mathbf{e}|$, subject to $|\mathbf{e}| = 1$
- in the overconstrained case, the choice of $\mathbf{W_L}$ and $\mathbf{W_R}$ below dramatically affects the solution:
  $\mathbf{W_L} \cdot \mathbf{A} \cdot \mathbf{W_R} \cdot \mathbf{e}' = 0$, where $\mathbf{e}' = \mathbf{W_R}^{-1} \cdot \mathbf{e}$

Improving the **numeric conditioning** of the 8pt algorithm:

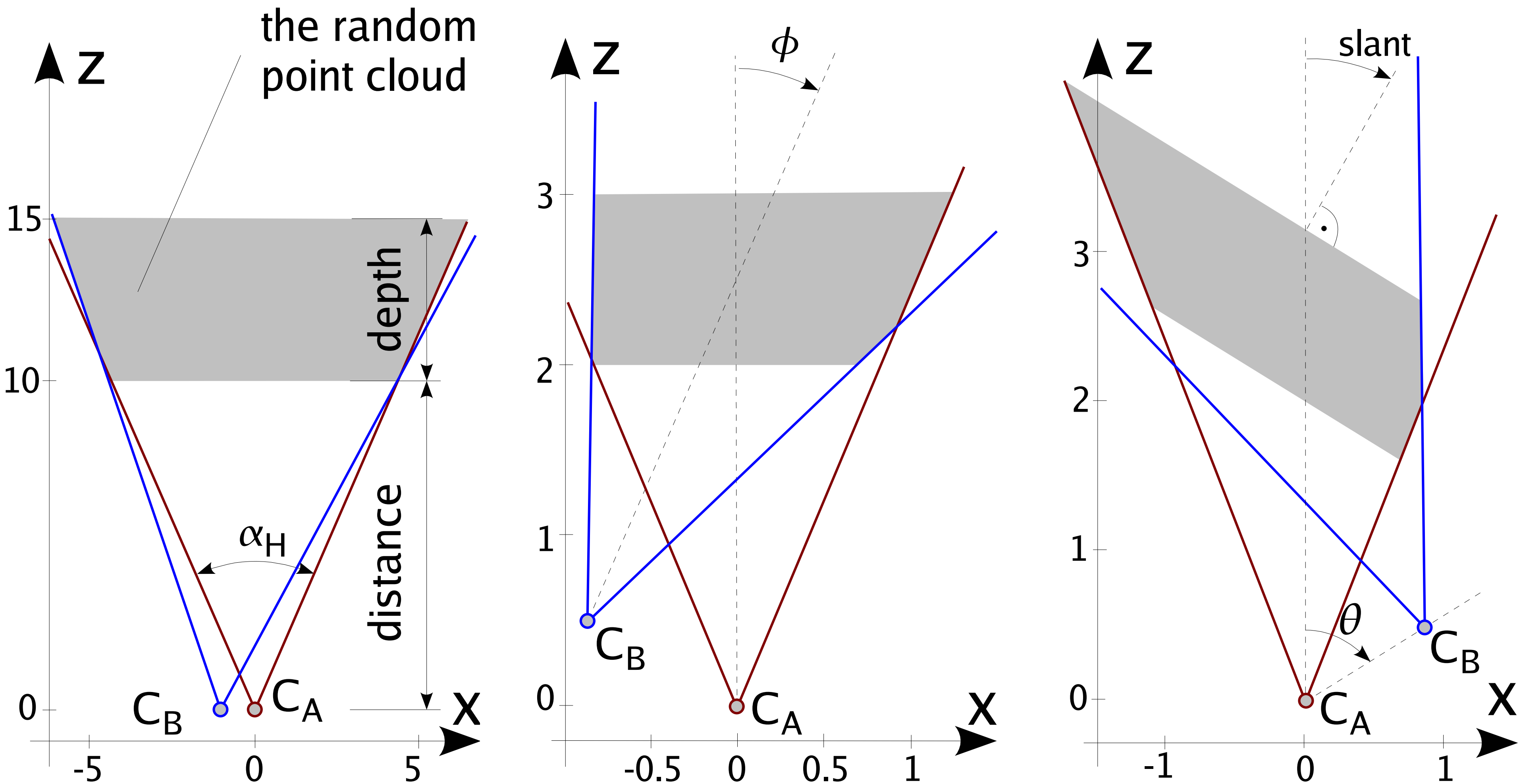
- the standard 8pt algorithm:
  $\min |\mathbf{A} \cdot \mathbf{e}|$, subject to $|\mathbf{e}| = 1$
- in the overconstrained case, the choice of $\mathbf{W_L}$ and $\mathbf{W_R}$ below dramatically affects the solution:
  $\mathbf{W_L} \cdot \mathbf{A} \cdot \mathbf{W_R} \cdot \mathbf{e}' = 0 \,,$ where $\mathbf{e}' = \mathbf{W_R}^{-1} \cdot \mathbf{e}$
- how to choose $\mathbf{W_L}$ and $\mathbf{W_R}$ (equilibrate the system)?
  $\rightarrow$ Mühlich provides a convincing recipe for $\mathbf{W_R}$

Improving the **numeric conditioning** of the 8pt algorithm:

- the standard 8pt algorithm:
  $\min |\mathbf{A} \cdot \mathbf{e}|$, subject to $|\mathbf{e}| = 1$
- in the overconstrained case, the choice of $\mathbf{W_L}$ and $\mathbf{W_R}$ below dramatically affects the solution:
  $\mathbf{W_L} \cdot \mathbf{A} \cdot \mathbf{W_R} \cdot \mathbf{e}' = 0\,,$ where $\mathbf{e}' = \mathbf{W_R}^{-1} \cdot \mathbf{e}$
- how to choose $\mathbf{W_L}$ and $\mathbf{W_R}$ (equilibrate the system)?
  $\rightarrow$ Mühlich provides a convincing recipe for $\mathbf{W_R}$
- Hartley's normalization recovers $\mathbf{E}' = \mathbf{T_2}^{-\top}\mathbf{E}\mathbf{T_1}^{-1}$ relating the transformed points $\mathbf{q}'_{ik} = \mathbf{T_k}\mathbf{q}_{ik}, k = \mathrm{A, B}$
- normalization is a proper subset of right equilibration.

The artificial **experimental setup**:

- planar motion along a unit circle:
  1 DOF rotation (ϕ) + 1 DOF translation (θ)
  around the common y axis
- the target point cloud instantiated between two planes (`distance`, `depth`, `slant`)
- i.i.d. Gaussian noise σ expressed in pixels of a 384×288 image



$(-5^\circ, 90^\circ, 10, 5, 0^\circ)$ $(-23^\circ, 60^\circ, 2, 1, 0^\circ)$ $(23^\circ, -60^\circ, 2, 1, -30^\circ)$

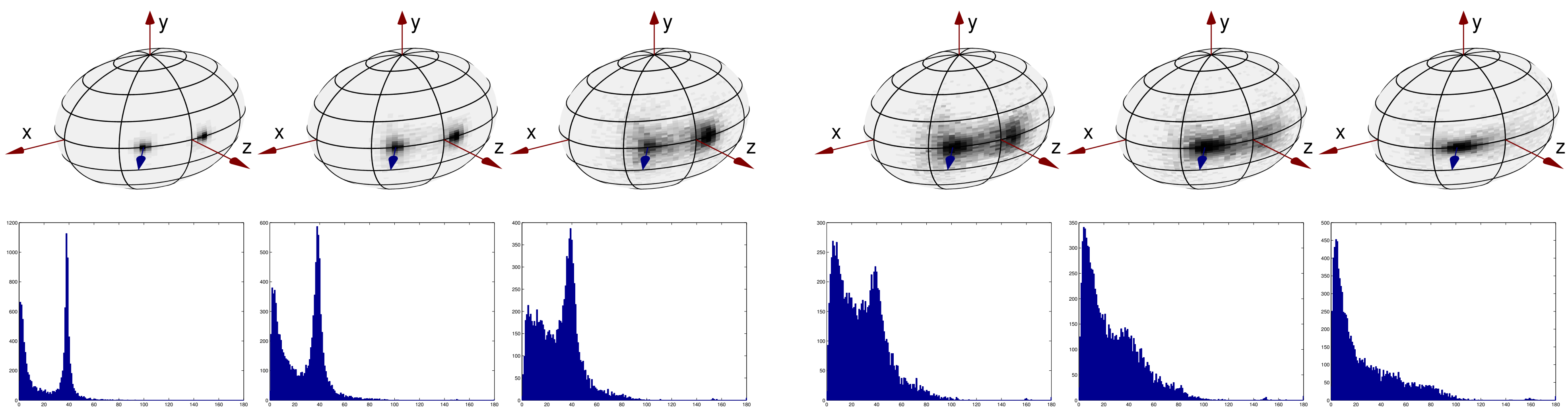
$(\phi, \theta, \texttt{distance}, \texttt{depth}, \texttt{slant})$

Experimental design:

- we look at the distribution of the angular error in the recovered **epipole**, $\Delta\mathrm{t} := \measuredangle(\mathrm{t}, \hat{\mathrm{t}})$, for n=10000
- $q_1\{\Delta\mathrm{t}\}$ (**minimal**), $med\{\Delta\mathrm{t}\}$ (**overconstrained**)
- the experiments were performed in
  - Matlab (prototype, 3D figures)
  - C++ with a little help from Python (production)
- used 5pt implementations by the original authors (Matlab) and from the library VW34 from Oxford (C++)

The `5pt(6)` algorithm and the **planar scenes**:

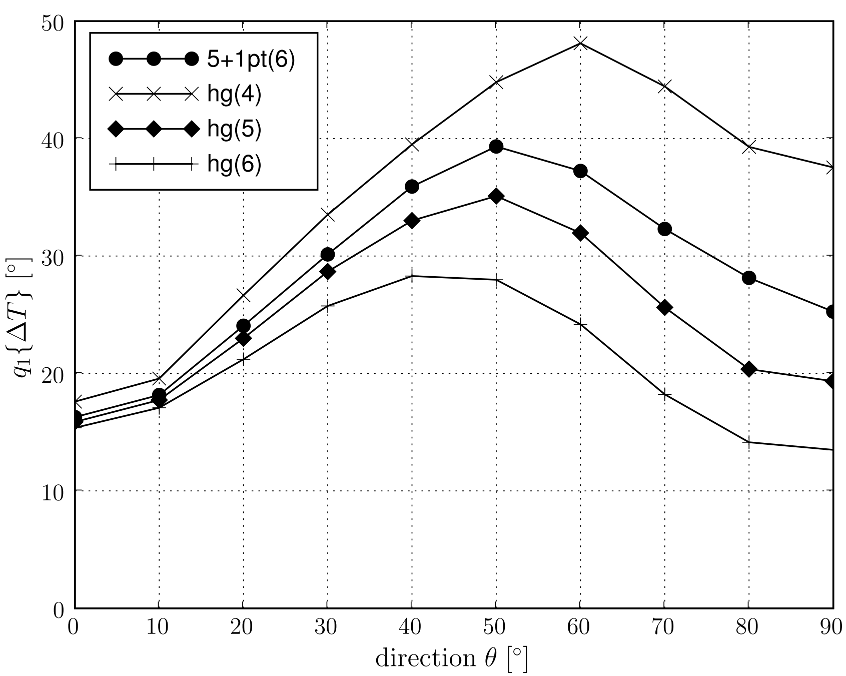
- frequency distributions of t (top), and $\Delta\mathrm{t}$ (bottom)
- the **unlabeled arrow** denotes $\hat{\mathrm{t}}$
- in the presence of ambiguity, both solutions are recovered (preference may be present!)

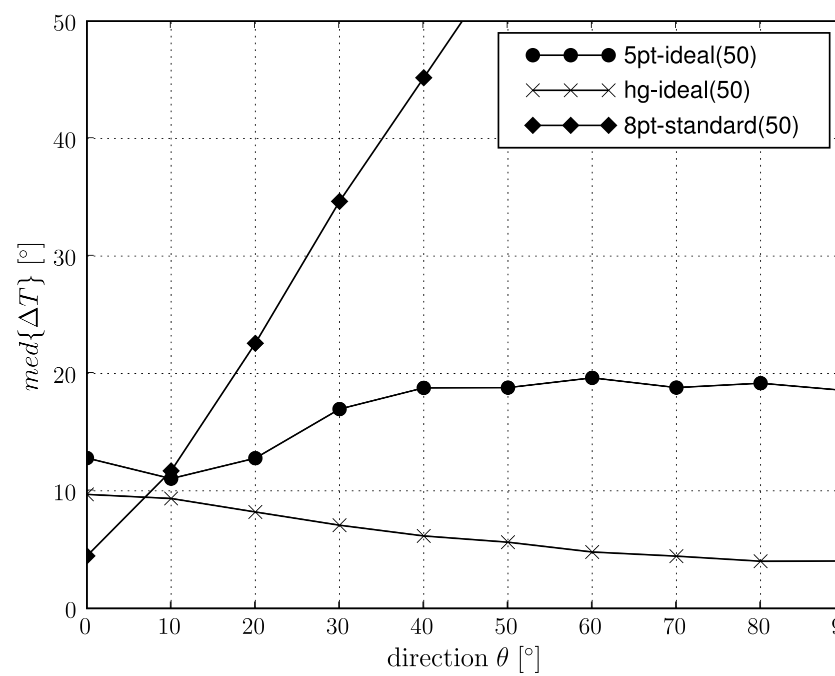


Left: `depth`=0, σ=(0.05,0.1,0.2); Right: `depth`=(1,2,5), σ=0.2
θ=150°, `slant`=10°

5pt algorithm vs. homography (`5pt` vs. `hg`) for **planar scenes**:

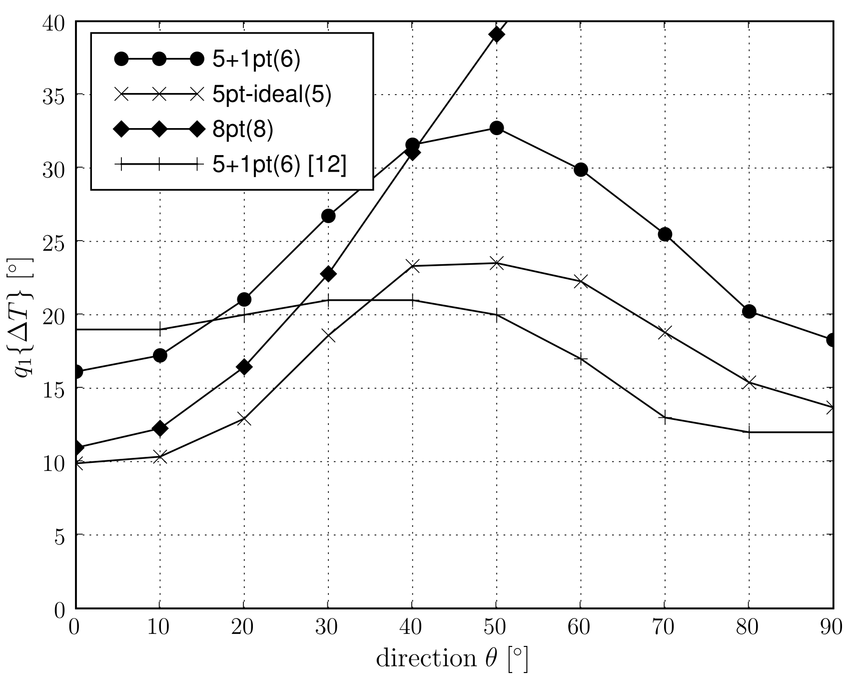
- minimal (left), and overconstrained cases (right)
- makes sense to compare: `5pt(6)` vs `hg(6)` (and `5pt-ideal(5)` vs `hg-ideal(5)`)
- the homography is better in minimal cases, and even more better in the overconstrained cases

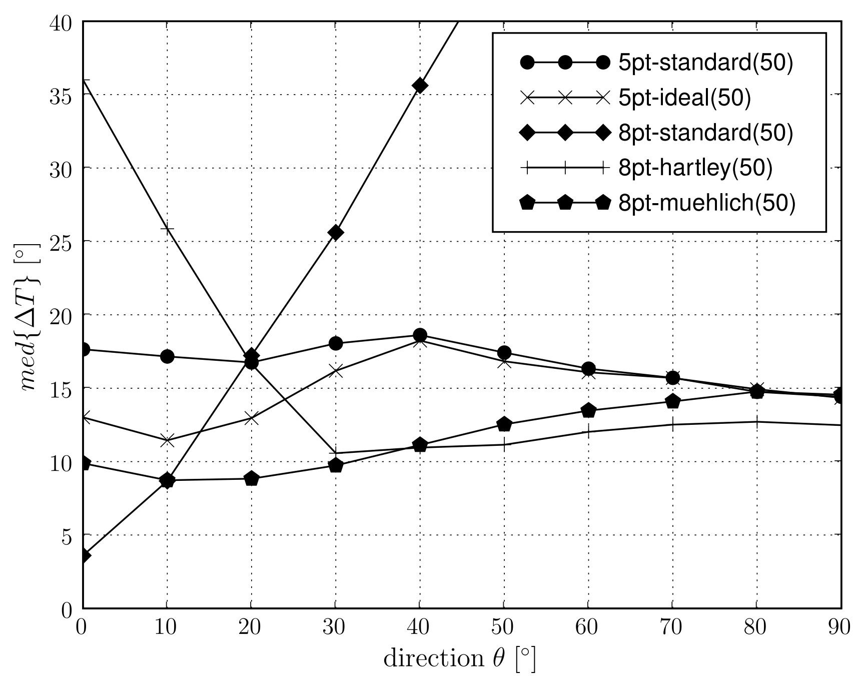




5pt **vs.** 8pt for **3D scenes** (depth=5):

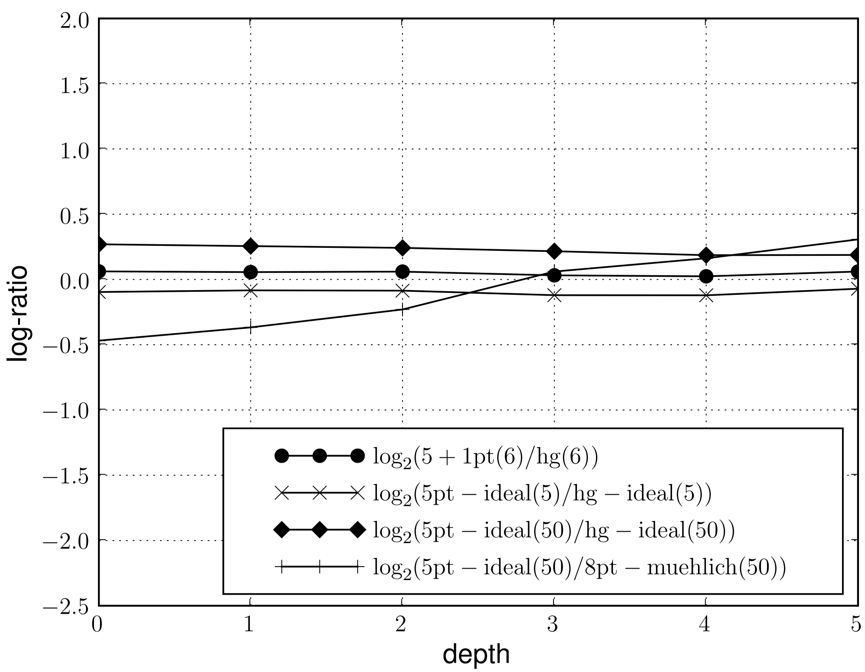
- minimal (left), and overconstrained cases (right)
- 5pt(6) beats 8pt(8) (with less information!)
- in the default overconstrained case 8pt-muehlich is better (this depends on sample size, depth, distance, σ, α_H)



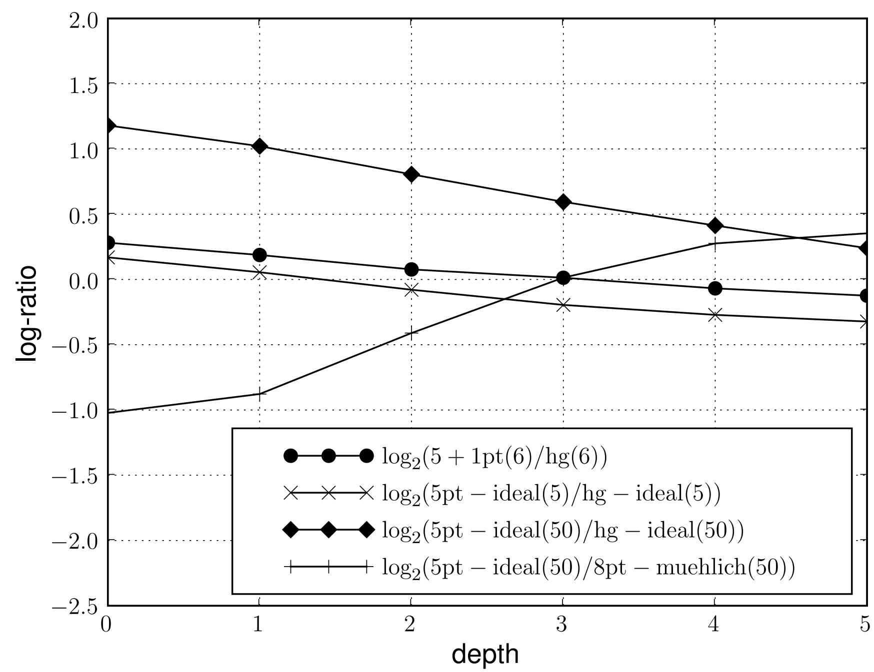


`5pt` vs. `8pt` vs. `hg` for **near-planar scenes**:

- log-ratio of $\{q_1, \text{med}\}$ against the depth, θ=0°, 45°, 90°
- `hg` and `5pt` level-off between `depth`=2 and `depth`=4
- in the overconstrained cases, `5pt` is never the best option



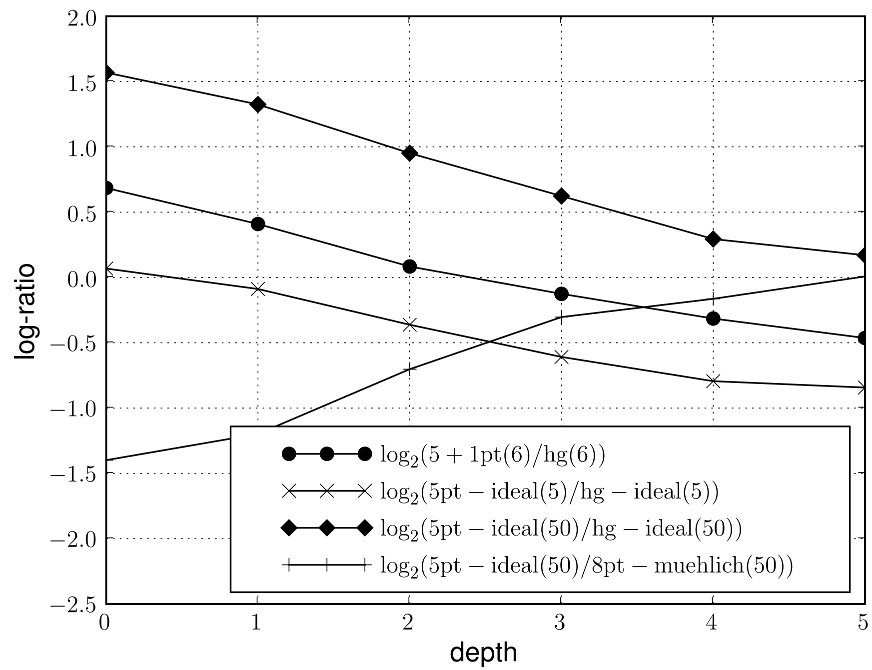
θ=0°



θ=45°

`5pt` vs. `8pt` vs. `hg` for **near-planar scenes** (cont.):

- log-ratio of the accuracy against the depth



θ=90°

The addressed **issues**:

- "planar degradation" of the 5pt algorithm
- comparison `5pt` vs `hg` (planar, near-planar scenes)
- comparison `5pt` vs conditioned `8pt` (near-planar, 3D scenes)

The addressed **issues**:

- "planar degradation" of the 5pt algorithm
- comparison `5pt` vs `hg` (planar, near-planar scenes)
- comparison `5pt` vs conditioned `8pt` (near-planar, 3D scenes)

Conclusions:

- `5pt` is usually not a method of choice in the overconstrained cases (planar and 3D)
- `5pt` is the best option in minimal 3D cases
- `5pt` is a viable option in a minimal planar case, but `hg` scores better
- Model selection required for best results