Computer Vision on Rolling Shutter Cameras

PART V: Structure from Motion

Per-Erik Forssén, Erik Ringaby, Johan Hedborg

Computer Vision Laboratory
Dept. of Electrical Engineering
Linköping University

CVPR 2012
Providence, Rhode Island
June 16-21, 2012

Linköping University
INSTITUTE OF TECHNOLOGY

Tutorial overview

1:30-2:00pm	Introduction	Per-Erik
2:00-2:15pm	Rolling Shutter Geometry	Per-Erik
2:15-3:00pm	Rectification and Stabilisation	Erik
3:00-3:30pm	Break	
3:30-3:45pm	Rolling Shutter and the Kinect	Erik
3:45-4:30pm	**Structure from Motion**	**Johan**

Overview

- Previous work
- Rotation camera model
- Rotation and translation camera model

Previous Work

Geometric Models of Rolling-Shutter Cameras
Christopher Geyer, Marci Meingast, and Shankar Sastry
OMNIVIS 2005

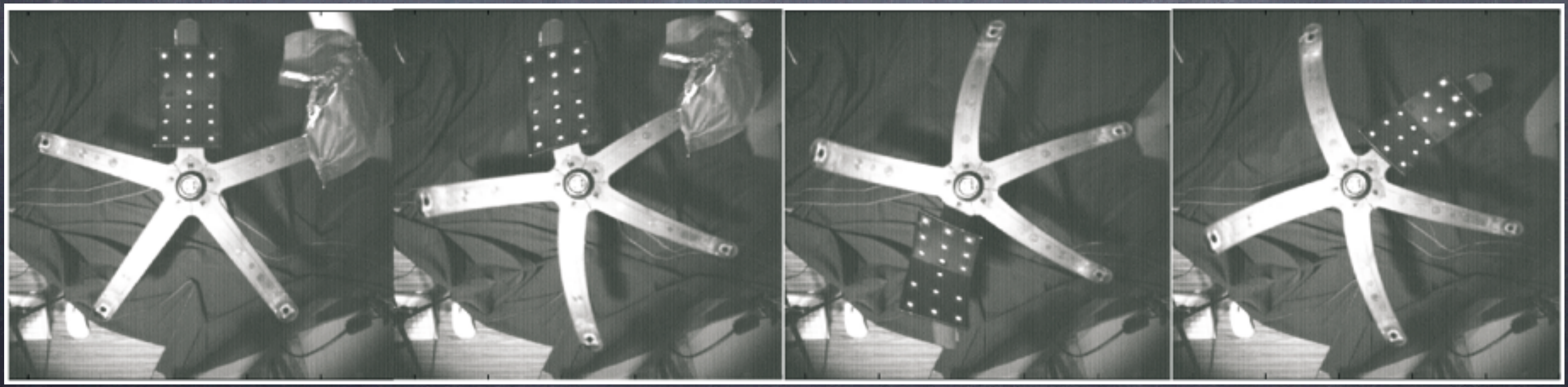
- Did rolling-shutter SfM on synthetic data under the assumption of fronto-parallel motion

Previous Work

Simultaneous Object Pose and Velocity Computation Using a Single View from a Rolling Shutter Camera

Omar Ait-Aider, Nicolas Andreff, Jean Marc Lavest and Philippe Martinet, ECCV 2006

- Solved the perspective-n-point problem for rolling shutter cameras, one frame only



Previous Work

Parallel Tracking and Mapping on a camera phone

Georg Klein and David Murray

ISMAR 2009

- Ported PTAM (parallel tracking and mapping) to the CMOS camera of the Iphone 3G
- System Initialization: find planar structure and do homography estimation
- Then a rolling shutter aware perspective-n-point method

Previous Work

Parallel Tracking and Mapping on a camera phone

Georg Klein and David Murray

ISMAR 2009

- Solve the velocity of the camera
- Then compensate for the RS-distortion

$$\begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_n \end{bmatrix} \dot{\mu} = \begin{bmatrix} \dot{\mathbf{m}}_1 \\ \vdots \\ \dot{\mathbf{m}}_n \end{bmatrix}, \mathbf{m}'_i = \mathbf{m}_i - \mathbf{J}_i \dot{\mu} \delta t$$

Classical Structure from Motion (Video data)

- Point correspondences : FAST + KLT with cross-checking
- Incremental Bundle Adjustment:

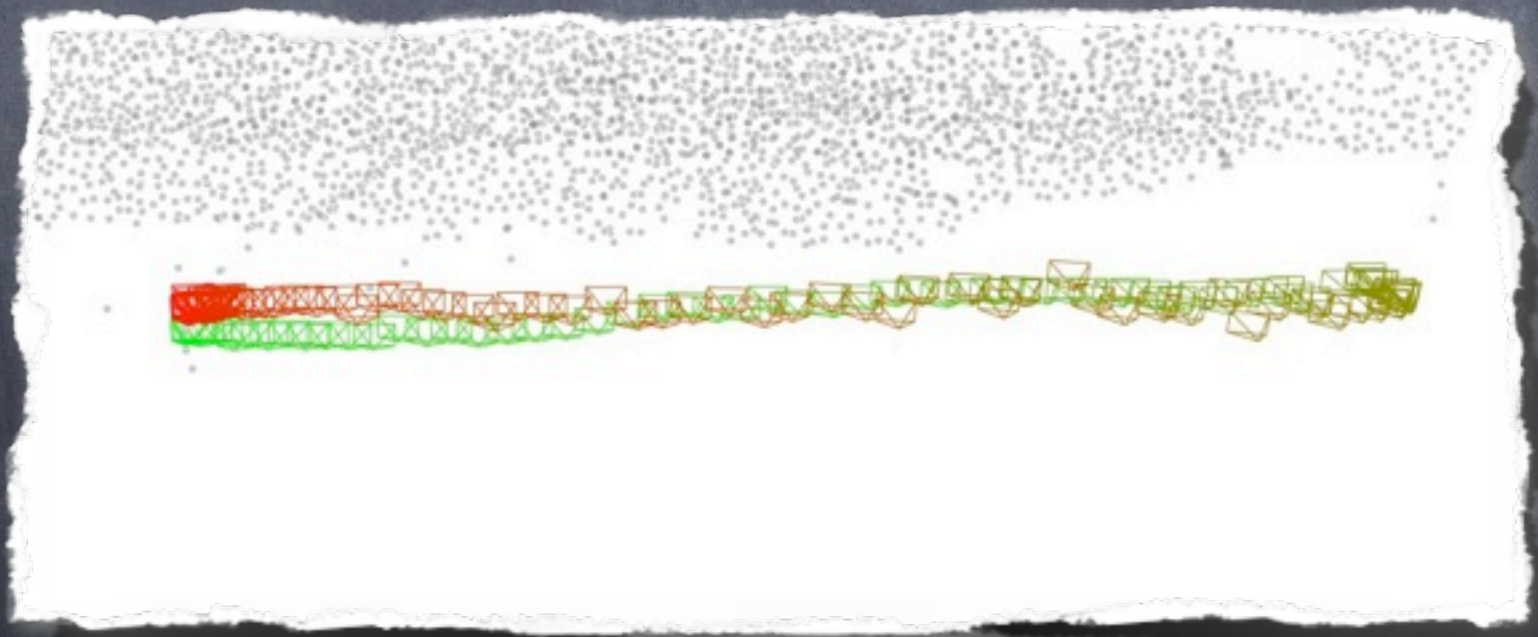
Estimate initial pose & triangulate 3D model

Add camera with perspective-n-point and points with triangulation

Do Bundle Adjustment

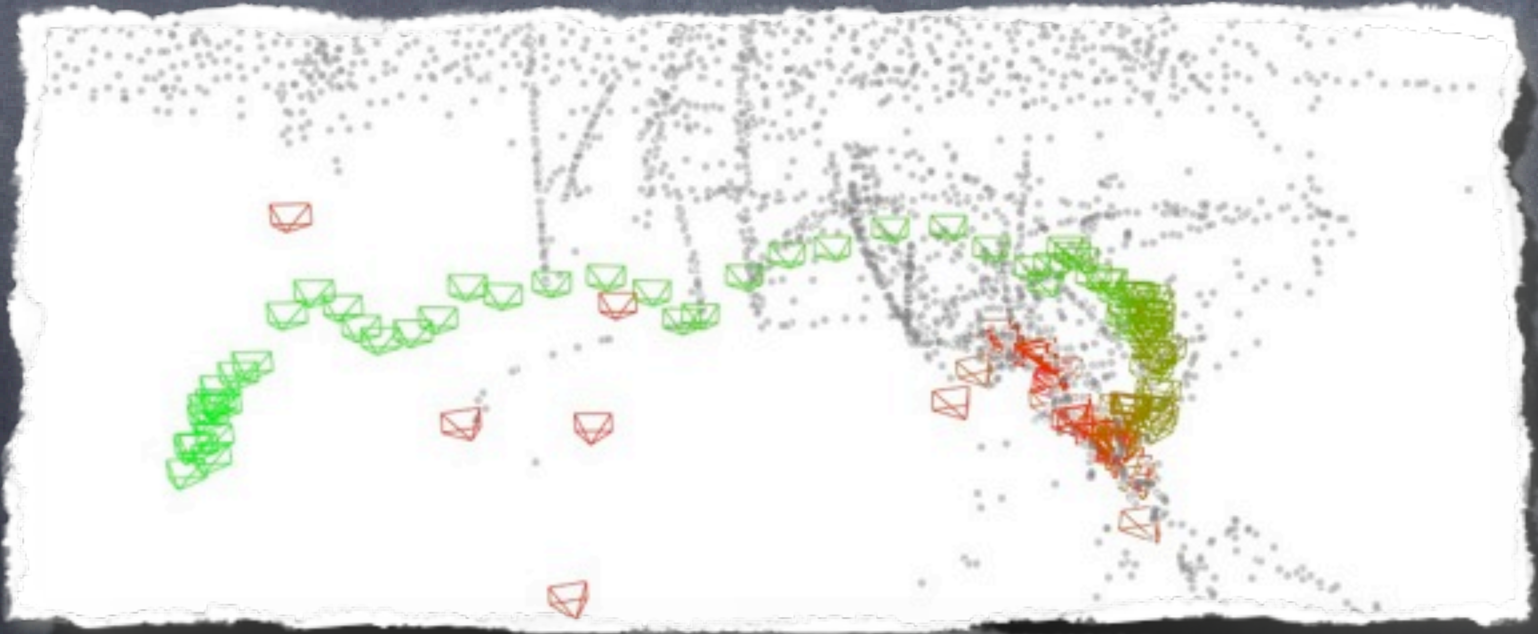
Classical Structure from Motion (Video data)

- Point correspondences : FAST + KLT with cross-checking
- Incremental Bundle Adjustment:



Classical Structure from Motion (Video data)

- Point correspondences : FAST + KLT with cross-checking
- Incremental Bundle Adjustment:
- iPhone 4 camera 720p 30Hz



A pure rotation camera model approach

- Using the rotation based correction previously shown in this tutorial

Structure and Motion Estimation from Rolling Shutter Video
Johan Hedborg, Erik Ringaby, Per-Erik Forssén, Michael Felsberg
ICCV workshop 2011

Initial camera poses and structure

- Tracking interest points for three or more views
- Compensate for lens distortion and intrinsic parameters
- Estimate camera rotations and rectify points for the views
- Estimate an essential matrix and then the relative poses for 3 of the views (with RANSAC)
- Triangulate points

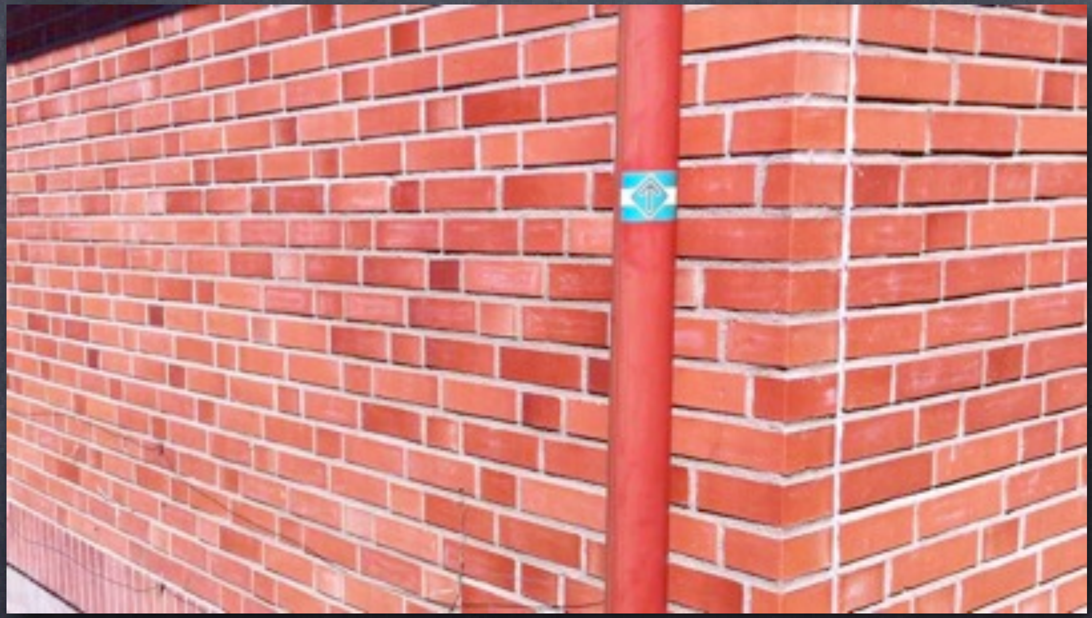
Structure from motion from RS video

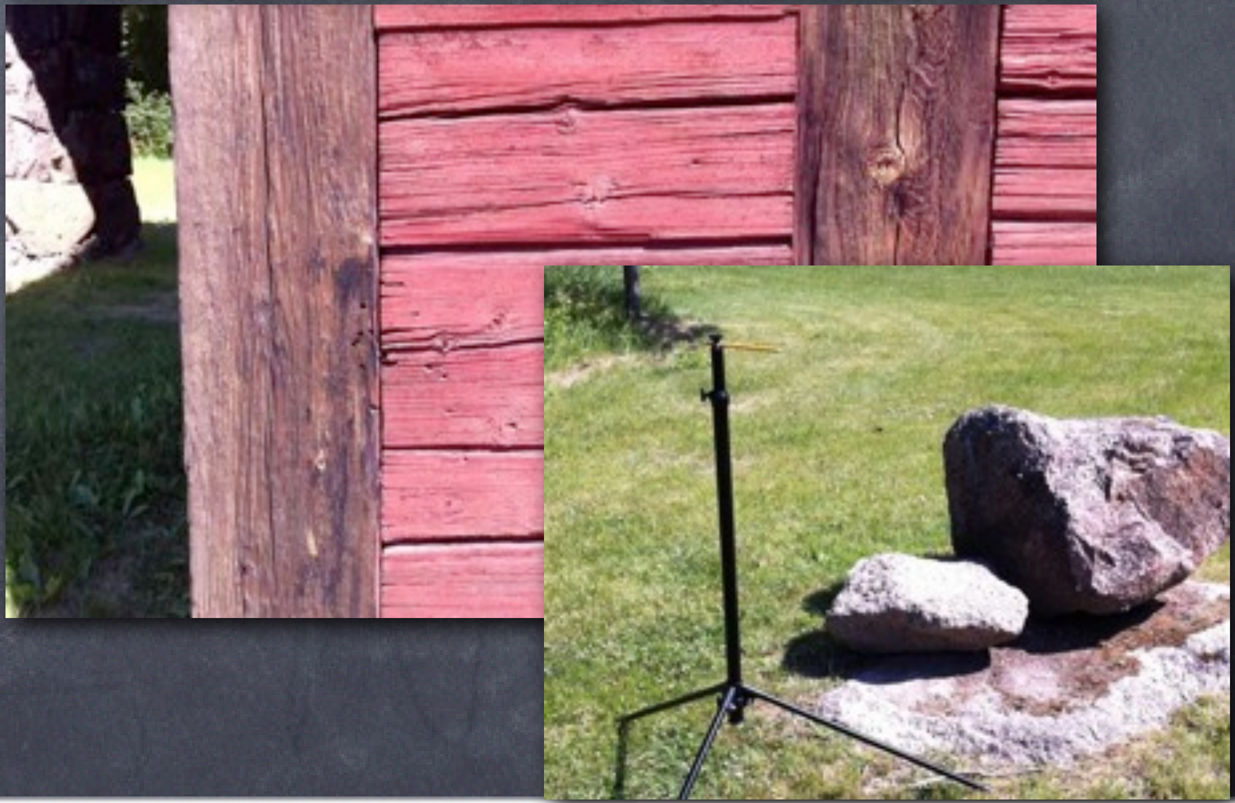
- Track interest points for a number of consecutive views + lens and intrinsic correction
- Rectify the newly tracked points
- Find new camera poses with perspective-n-point
- Triangulate new points
- Perform bundle adjustment

Alternative pre-rectification

- Rectification done by off the shelf software (here Deshaker)
- Easy and works OK, but...
- Lens distortion and intrinsic correction a lot harder
- Slower : 2x optical flow and image warping

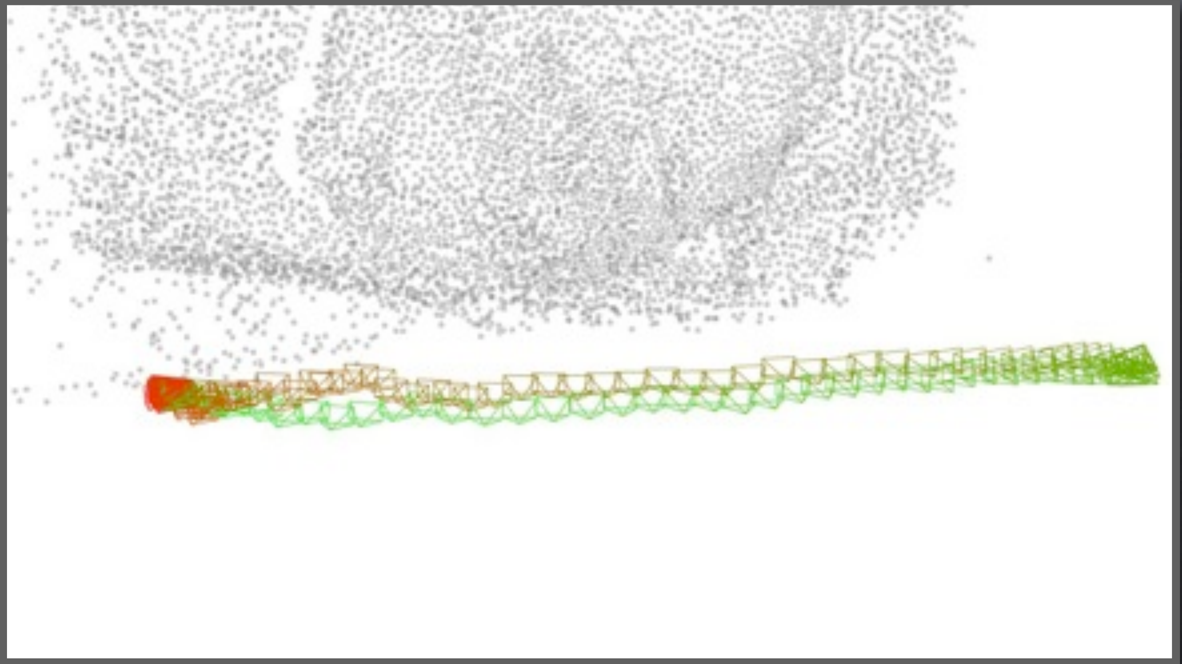
Dataset

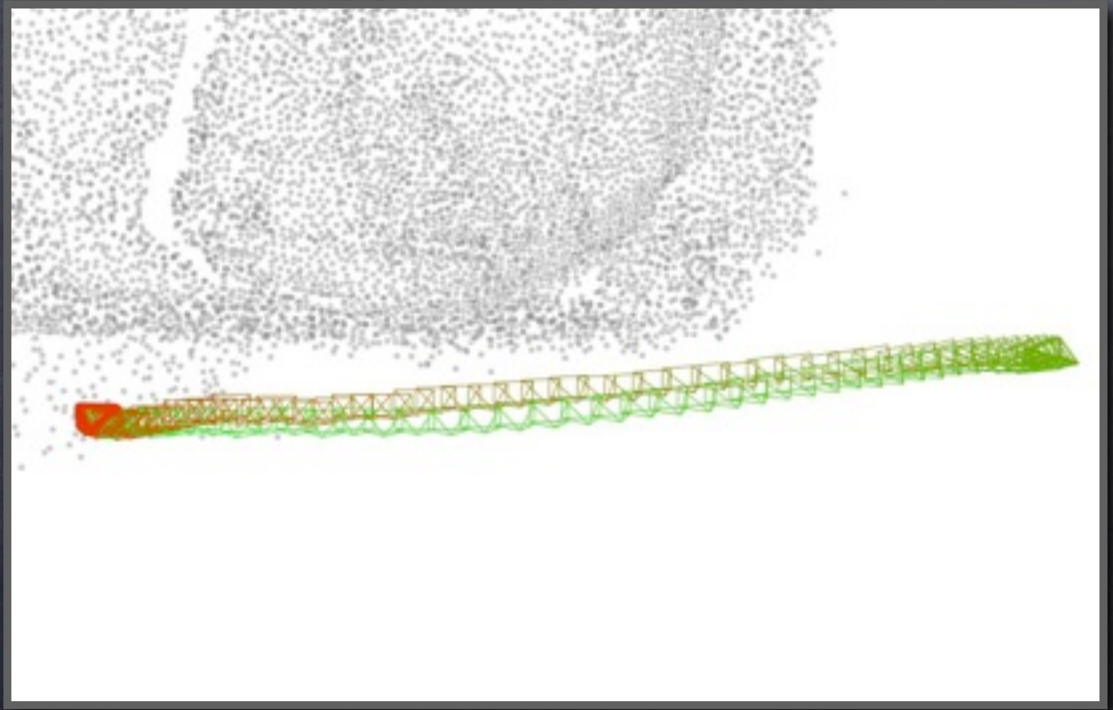


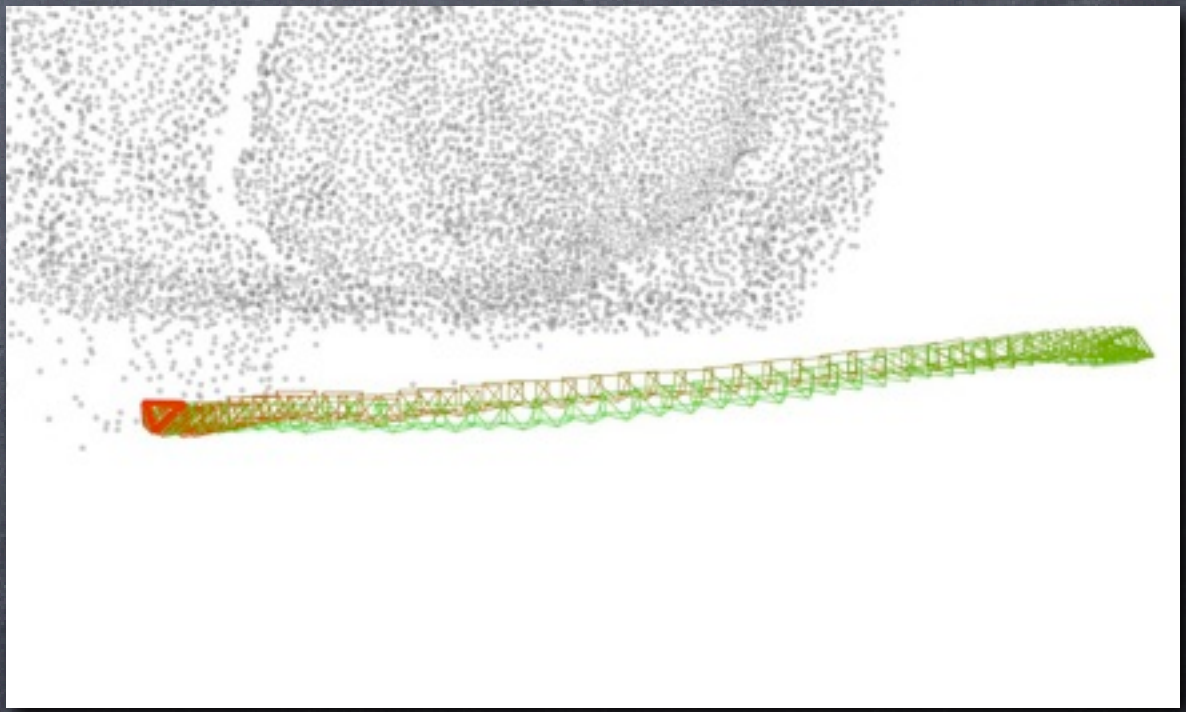


Result

Global Shutter SaM	0,0070
Deshaker	0,0063
Our	**0,0052**

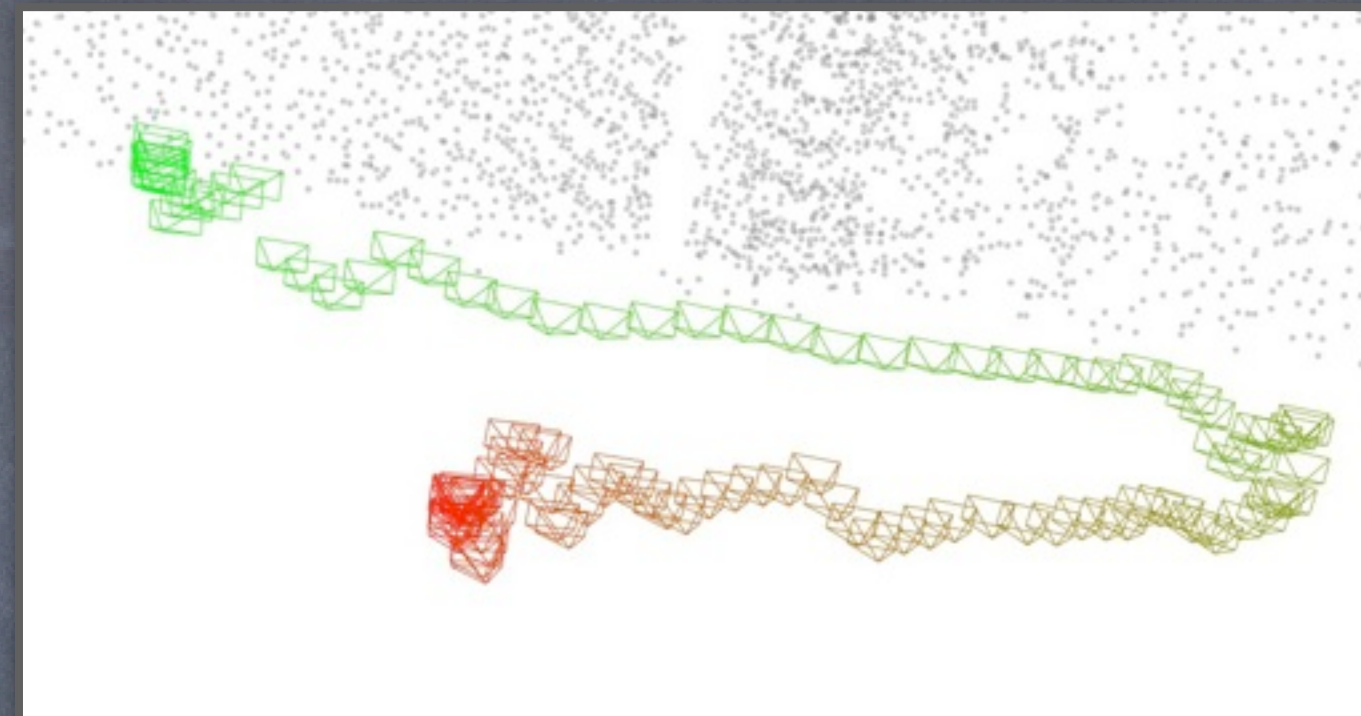


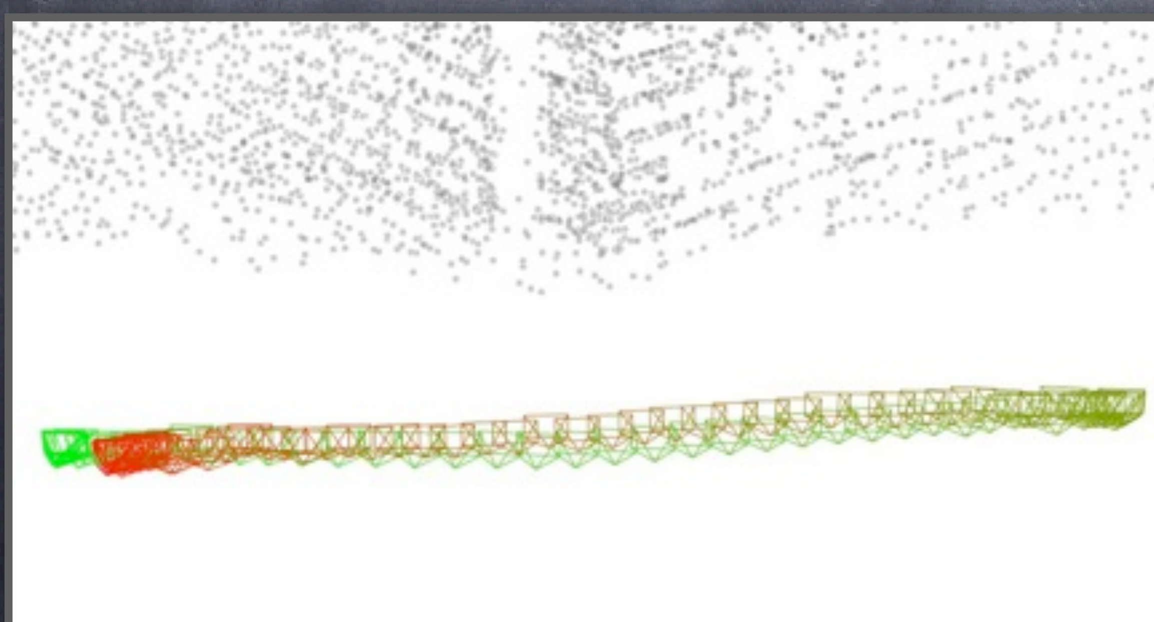


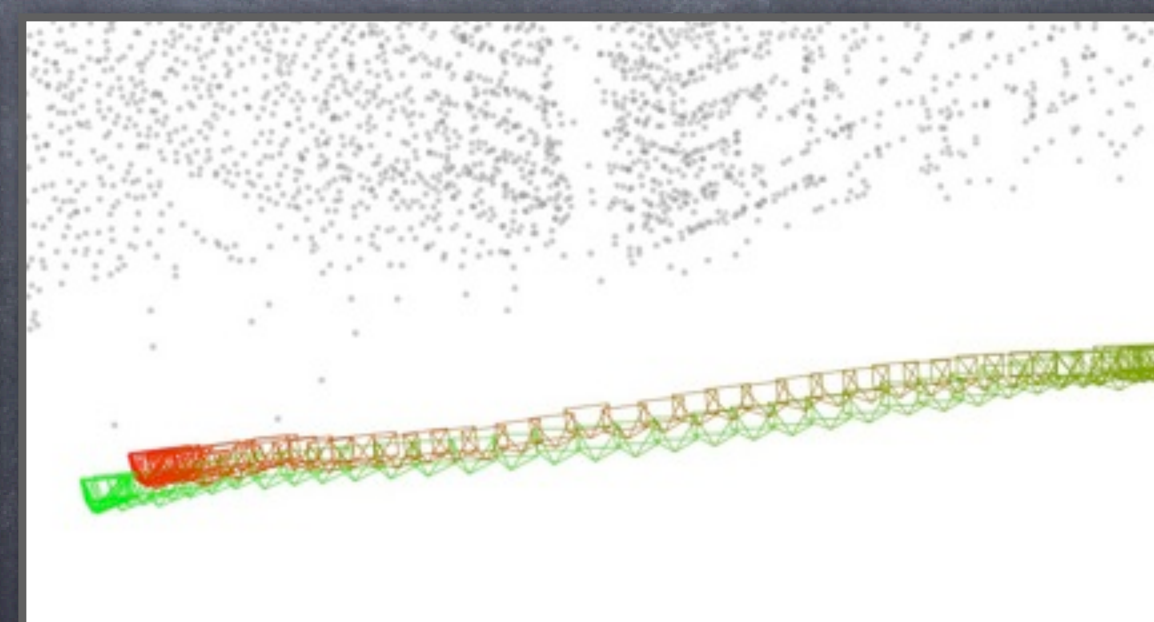


Result

Global Shutter SaM	0,3636
Deshaker	**0,0540**
Our	0,0650

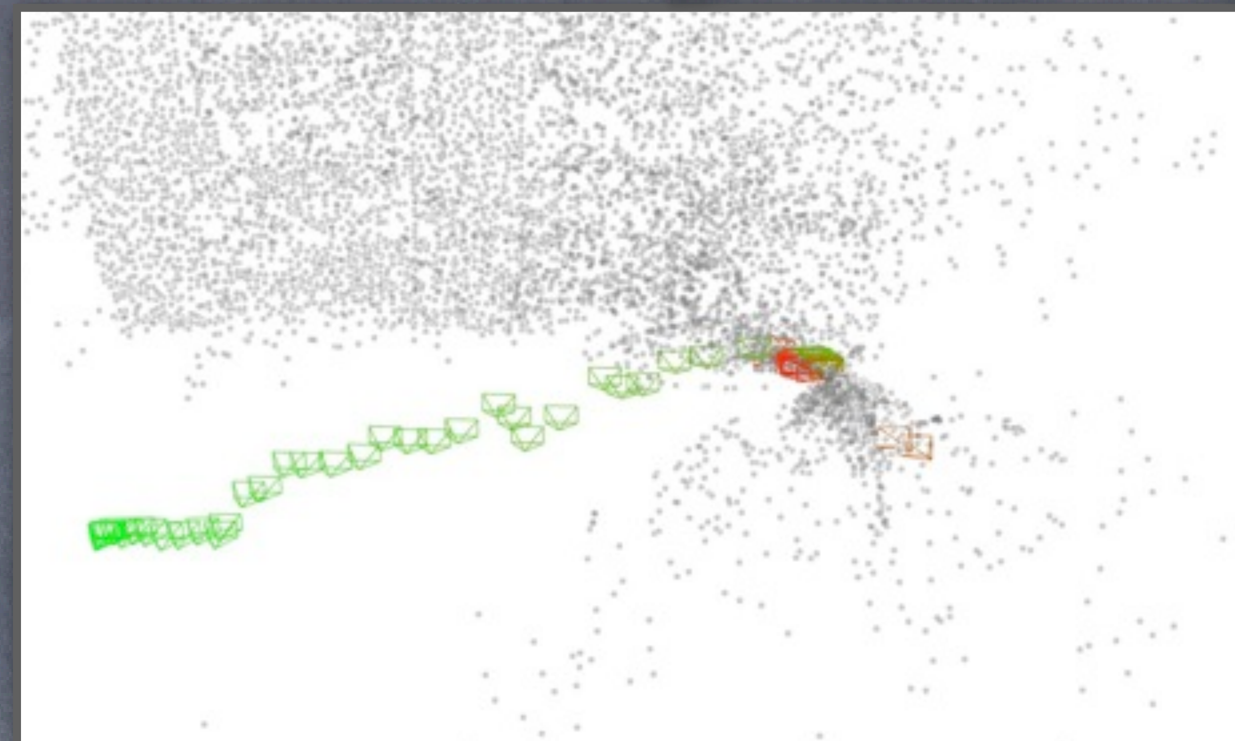


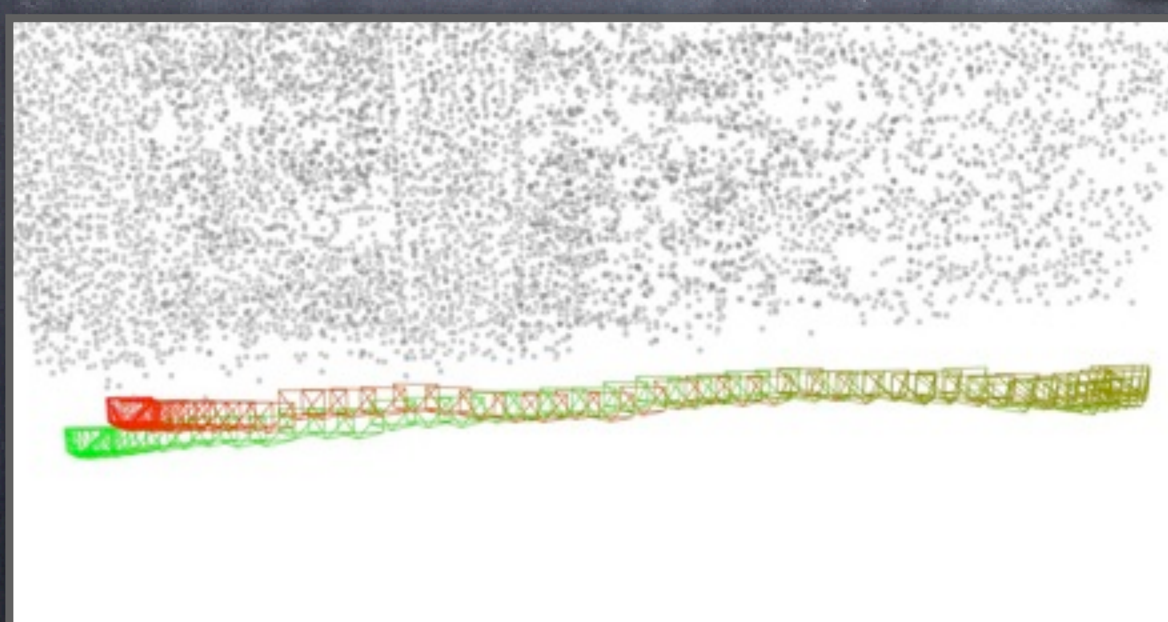


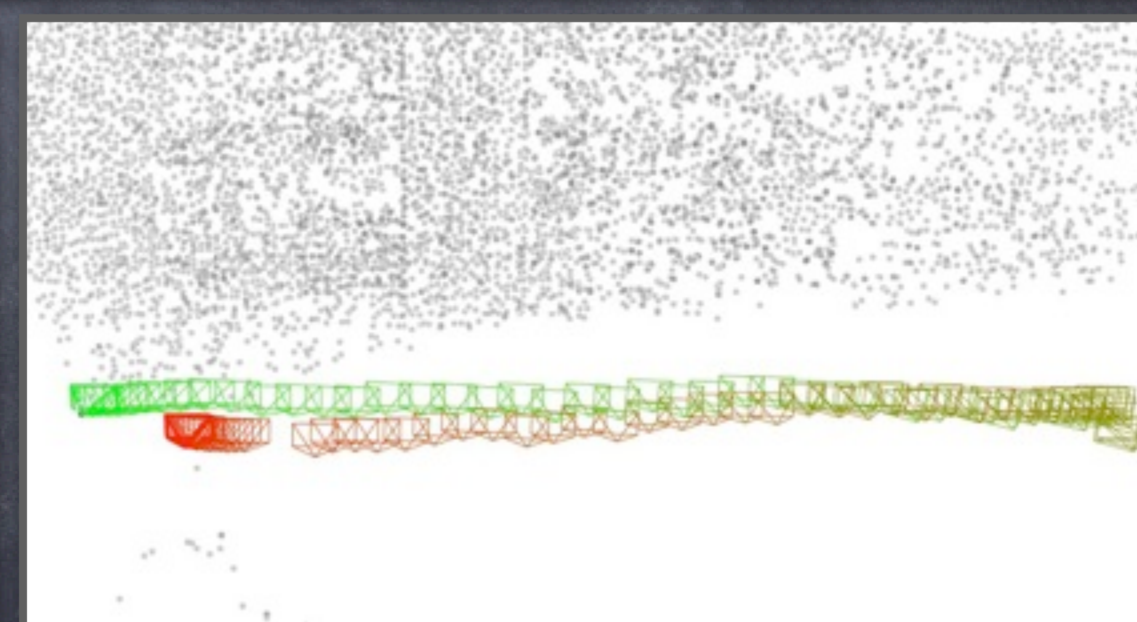


Result

Global Shutter SaM	0,9855
Deshaker	0,1000
Our	**0,0583**

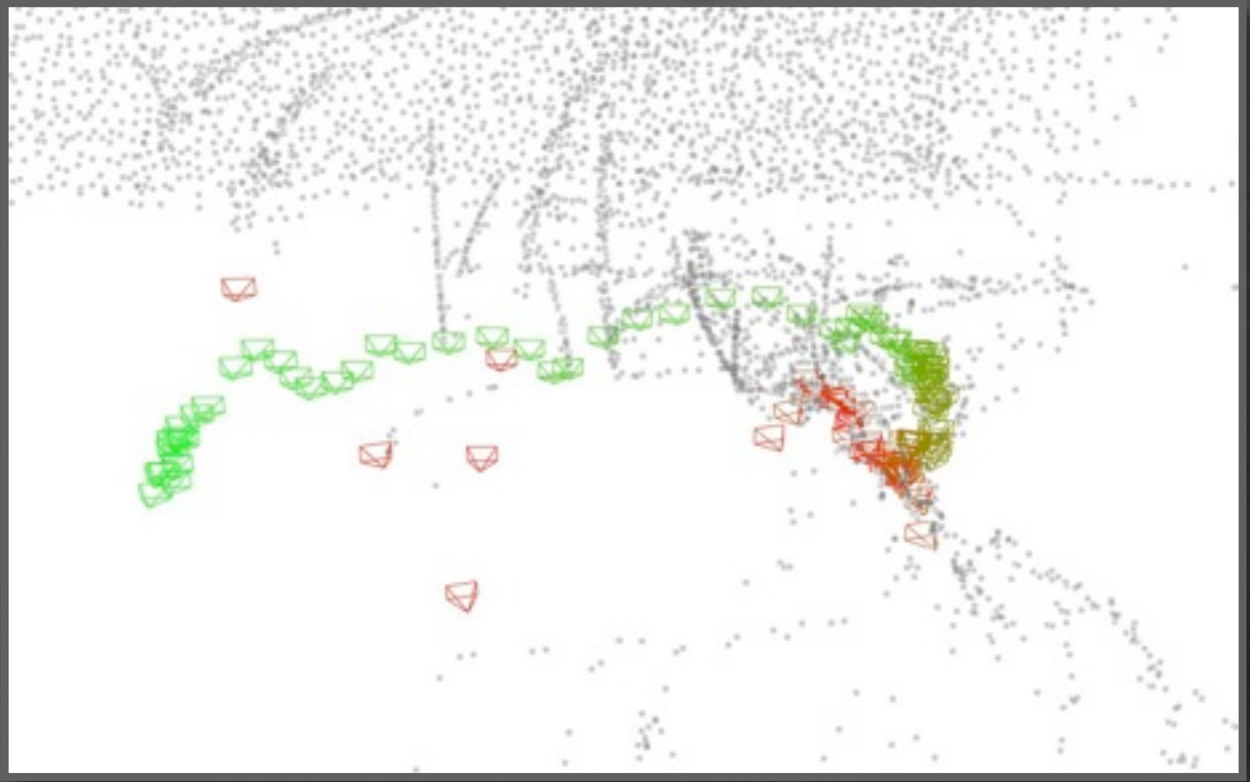


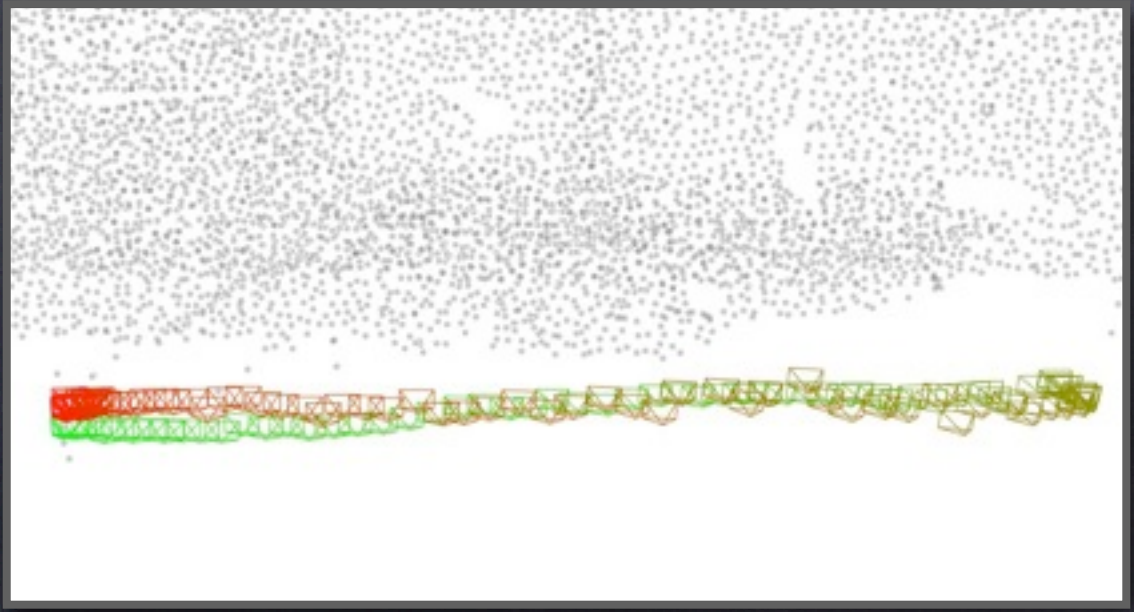


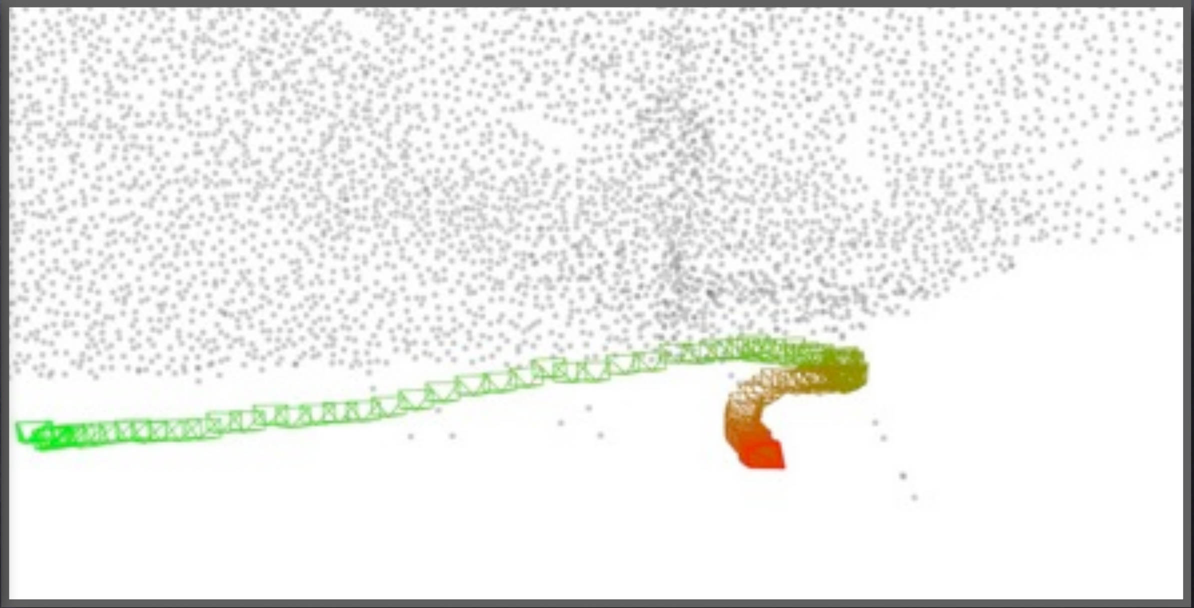


Result

Global Shutter SaM	0,9677
Deshaker	0,9218
Our	**0,0286**







The full translation and rotation camera model approach

- The next step
- Avoids error accumulation
- Better use of trajectory information

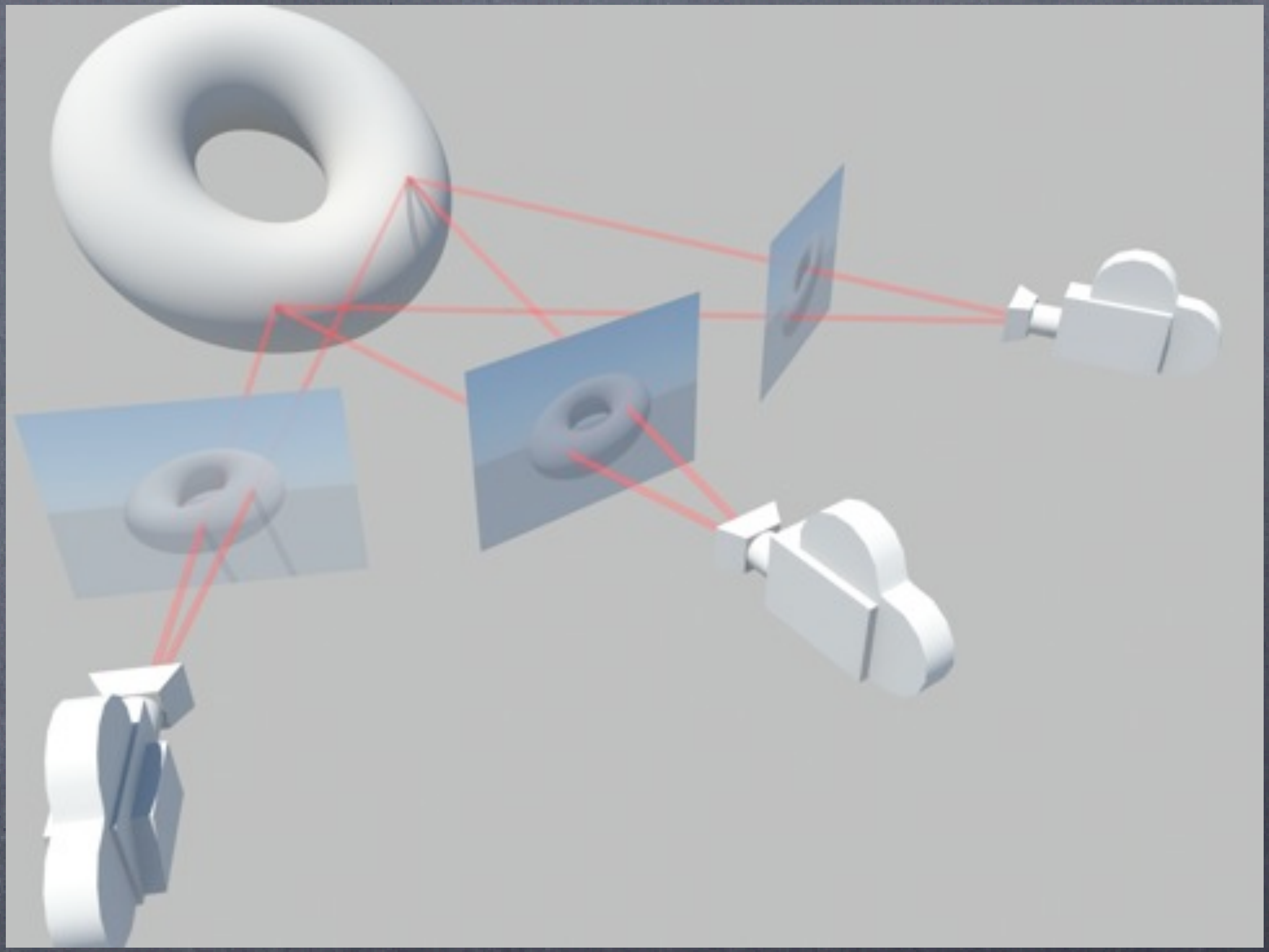
Rolling Shutter Bundle Adjustment

Johan Hedborg, Per-Erik Forssén, Michael Felsberg, Erik Ringaby

CVPR 2012

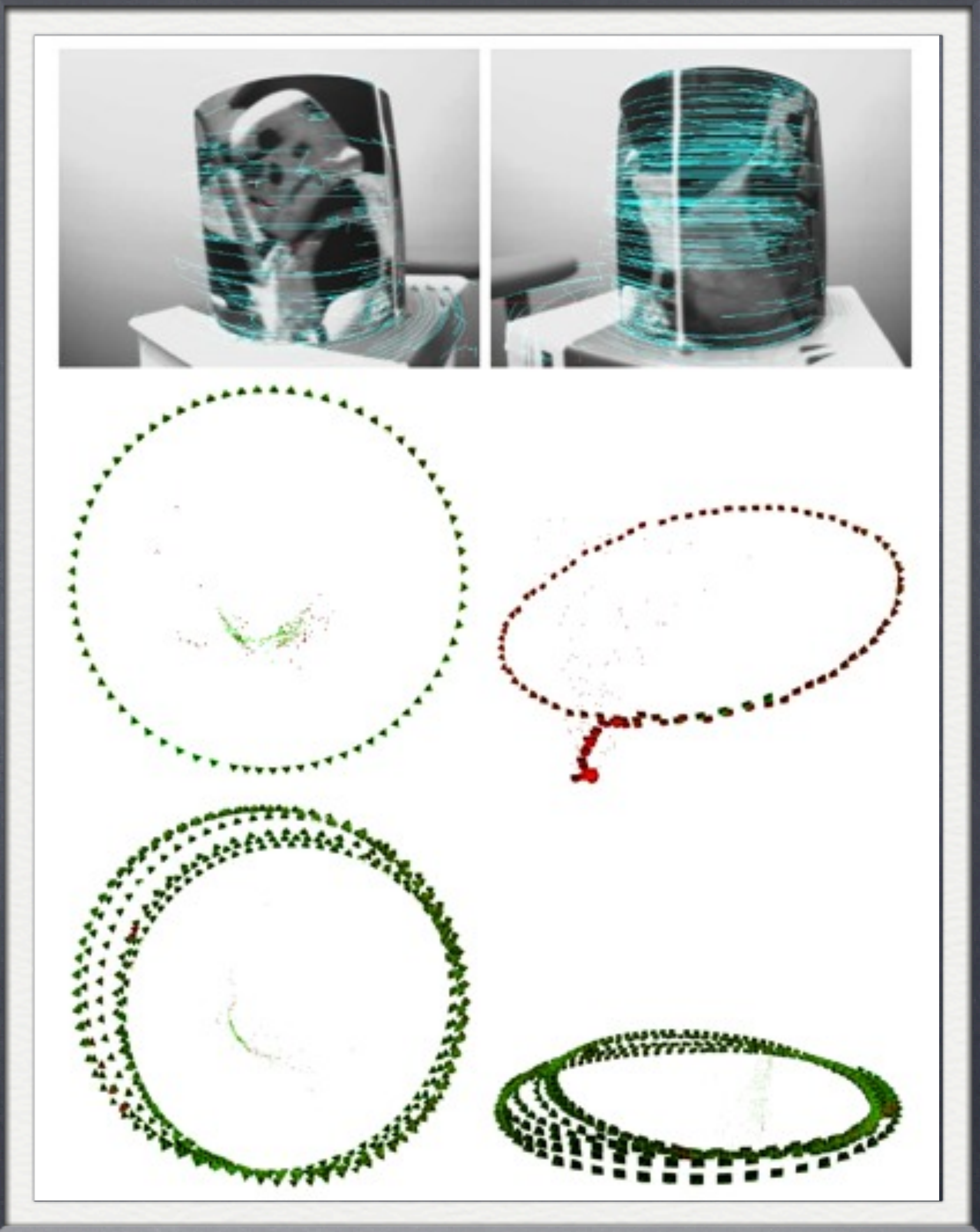
Bundle Adjustment

- Optimize over :
  cameras + structure
- By minimizing :
  distance from
  observed points
  to re-projected
  structure
- To achieve :
  as good camera poses
  and structure
  as possible



BA, more than refinement

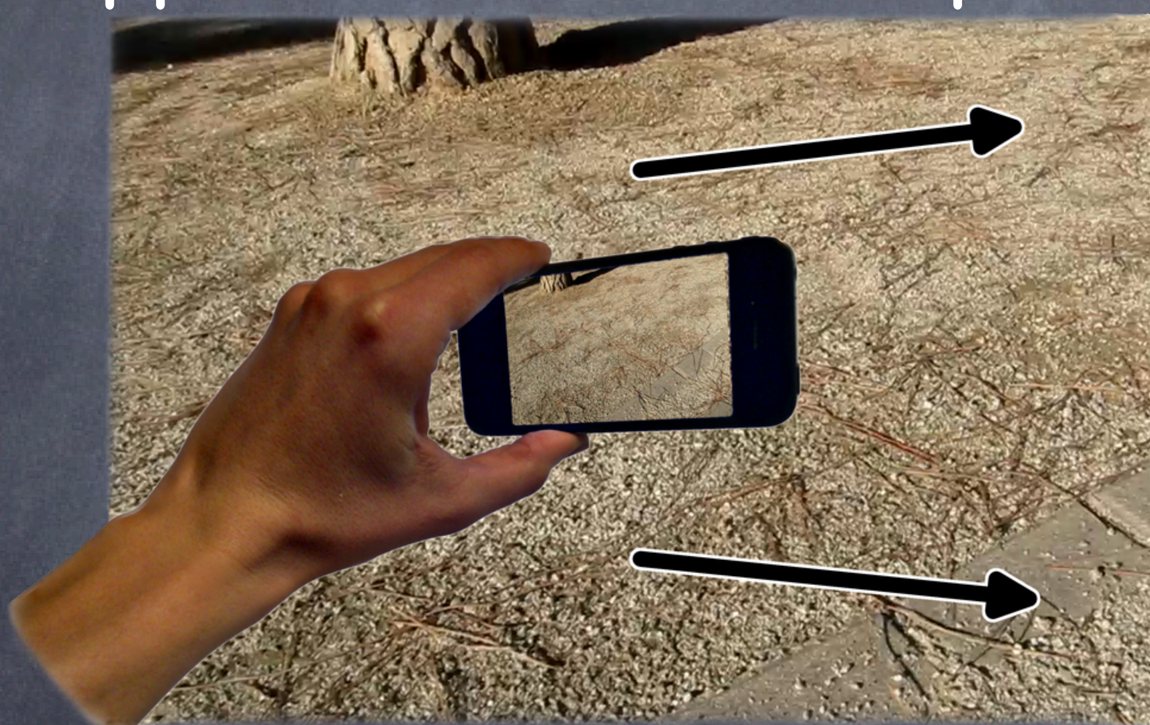
- Sequential BA
- Enabler for certain problems
- Figure shows windowed BA, win size : 20 frames

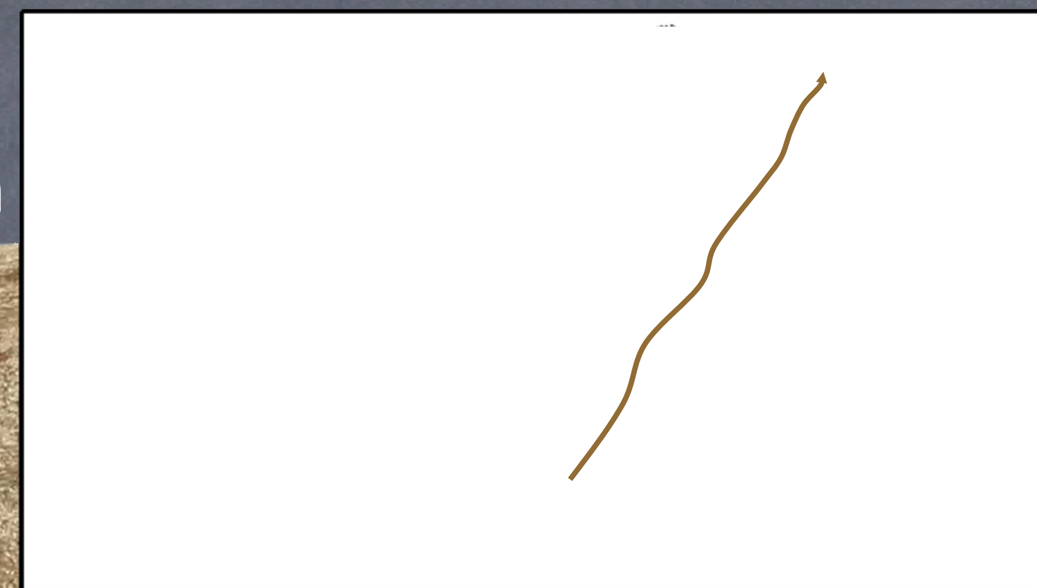


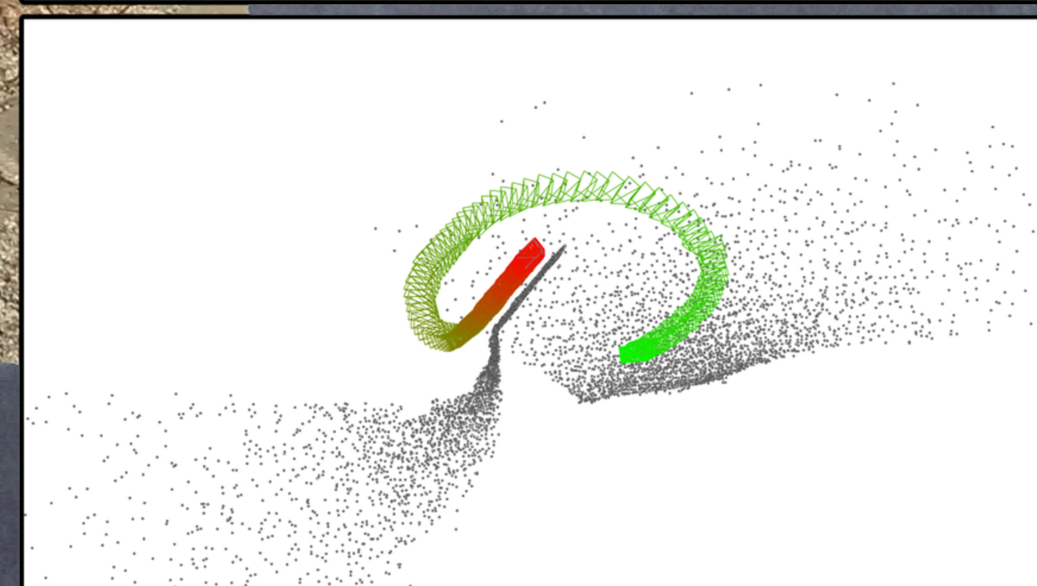
BUNDLE ADJUSTMENT RULES
Chris Engels, Henrik Stewénius, David Nistér, PCV 2006

Classical BA on Rolling Shutter images

Approximate camera path

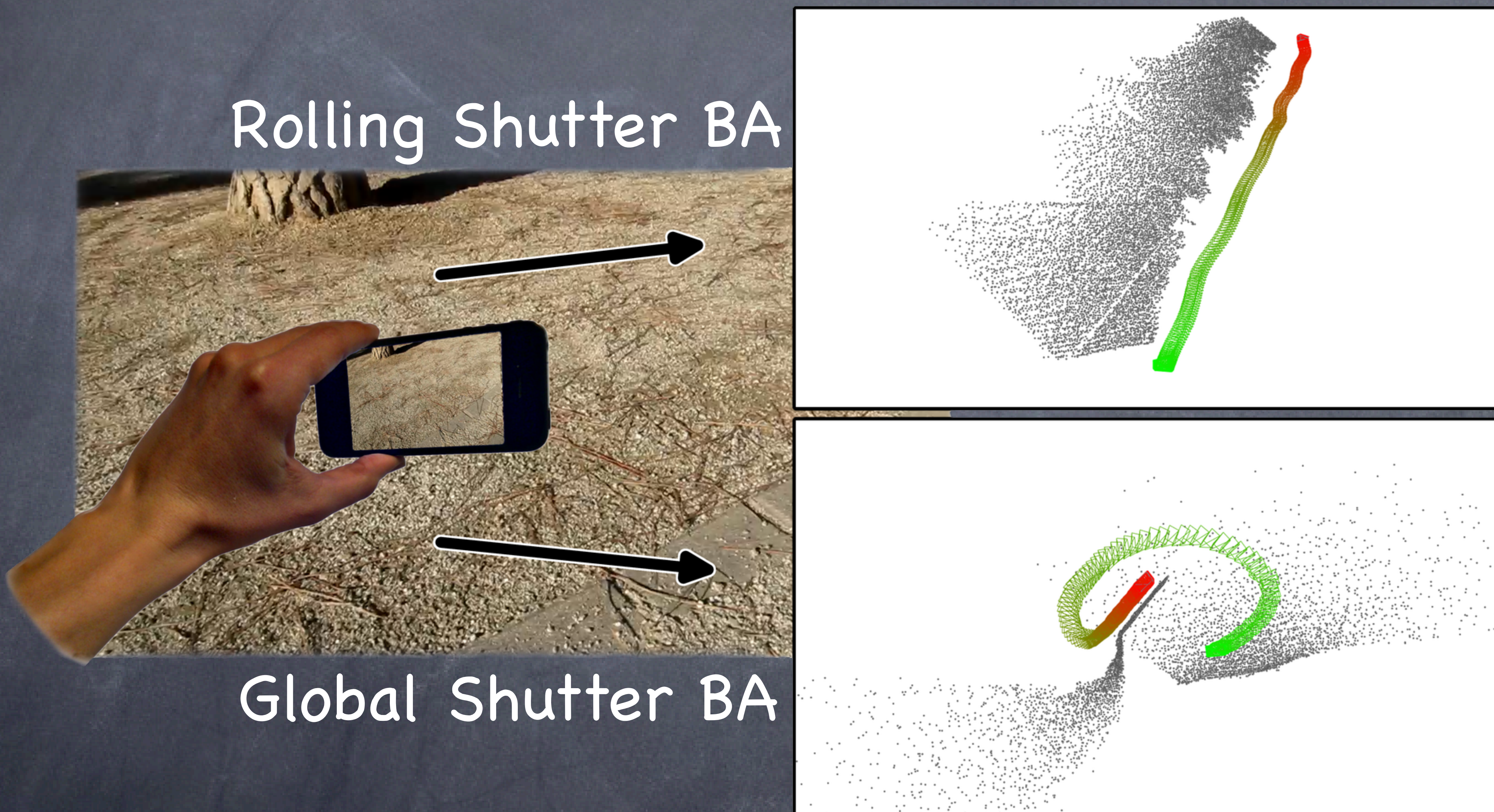






Global Shutter BA

Classical BA on Rolling Shutter images



Cost Function, BA

Cameras: $\mathbf{C}_j(y),\ j = 1..J$. 3D points: $\mathbf{X}_k,\ k = 1..K$

$$\mathbf{C}_j = [\mathbf{R}_j^T | -\mathbf{R}_j^T \mathbf{t}_j]$$

$$\text{proj}(\mathbf{C}_j, \mathbf{X}_k) \text{ where proj} : (\Re^{3\times 4}, \Re^3) \to \Re^2$$

$$\min_{\mathbf{C},\mathbf{X}} \frac{1}{2} \sum_{j=1}^{J} \sum_{k \in \mathcal{V}_j} ||\mathbf{p}_{j,k} - \text{proj}(\mathbf{C}_j, \mathbf{X}_k)||_2^2$$

$\mathcal{V}_j$ is the set of visible points in camera j

Cost Function, RSBA

Cameras: $\mathbf{C}_j(y),\ j = 1..J$. 3D points: $\mathbf{X}_k,\ k = 1..K$

$$\mathbf{C}_j(y) = [\mathbf{R}^T_{j,j+1}(y)| - \mathbf{R}^T_{j,j+1}(y)\mathbf{t}_{j,j+1}(y)]$$

$$\text{proj}(\mathbf{C}_j(y), \mathbf{X}_k) \text{ where proj} : (\Re^{3\times 4}, \Re^3) \to \Re^2$$

$$\min_{\mathbf{C},\mathbf{X}} \frac{1}{2} \sum_{j=1}^{J} \sum_{k \in \mathcal{V}_j} ||\mathbf{p}_{j,k} - \text{proj}(\mathbf{C}_j(y_k), \mathbf{X}_k)||_2^2$$

$\mathcal{V}_j$ is the set of visible points in camera j

Optimization, Levenberg-Marquardt

begin
  $k := 0; \quad \nu := 2; \quad \mathbf{x} := \mathbf{x}_0$
  $\mathbf{A} := \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}); \quad \mathbf{g} := \mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x})$
  found $:= (\|\mathbf{g}\|_\infty \le \varepsilon_1); \quad \mu := \tau * \max\{a_{ii}\}$
  while (**not** *found*) **and** $(k < k_{\max})$
    $k := k+1;$ Solve $(\mathbf{A} + \mu \mathbf{I})\mathbf{h}_{\text{lm}} = -\mathbf{g}$
    if $\|\mathbf{h}_{\text{lm}}\| \le \varepsilon_2(\|\mathbf{x}\| + \varepsilon_2)$
      found := **true**
    else
      $\mathbf{x}_{\text{new}} := \mathbf{x} + \mathbf{h}_{\text{lm}}$
      $\varrho := (F(\mathbf{x}) - F(\mathbf{x}_{\text{new}}))/(L(\mathbf{0}) - L(\mathbf{h}_{\text{lm}}))$
      if $\varrho > 0$
        $\mathbf{x} := \mathbf{x}_{\text{new}}$
        $\mathbf{A} := \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}); \quad \mathbf{g} := \mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x})$
        found $:= (\|\mathbf{g}\|_\infty \le \varepsilon_1)$
        $\mu := \mu * \max\{\frac{1}{3}, 1 - (2\varrho - 1)^3\}; \quad \nu := 2$
      else
        $\mu := \mu * \nu; \quad \nu := 2 * \nu$
end

Solving the Normal equations

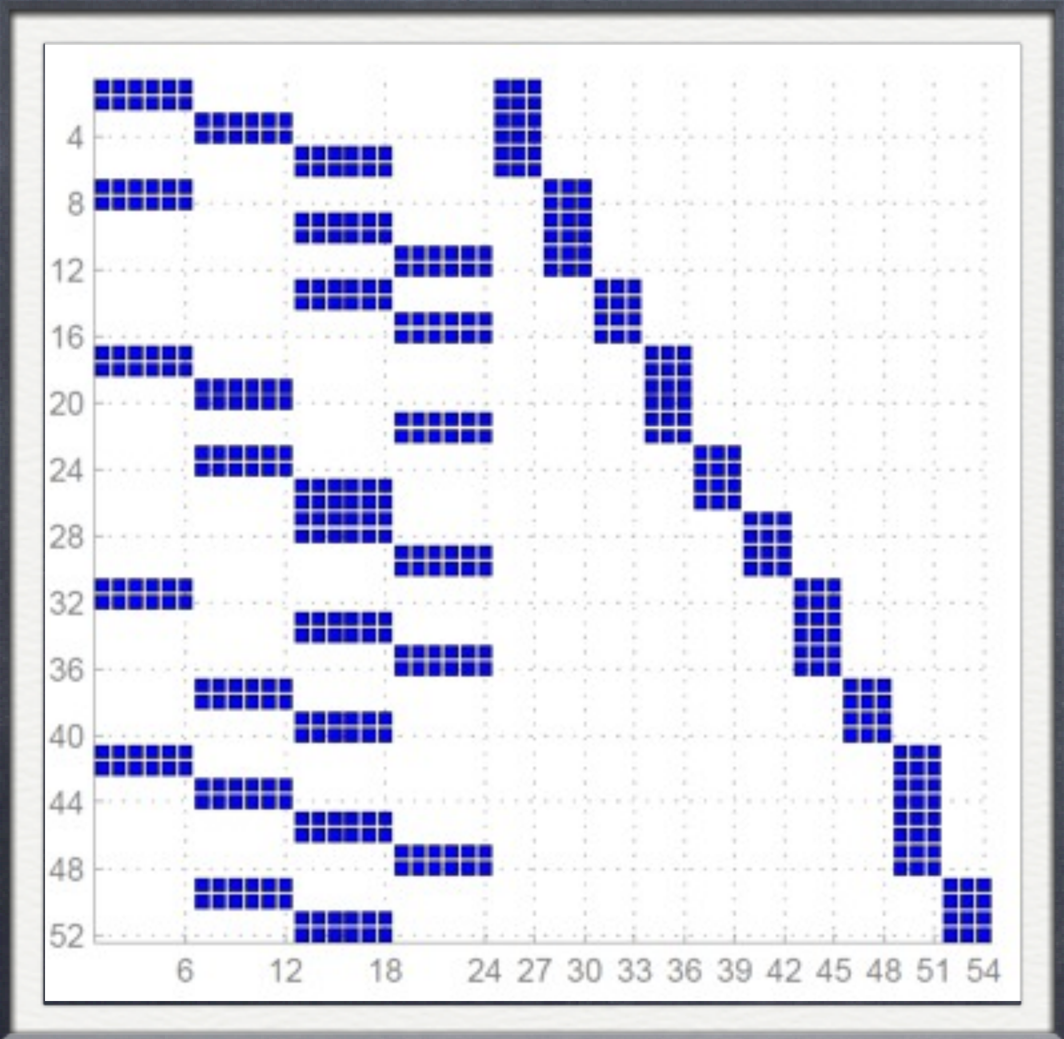
Time and memory complexity

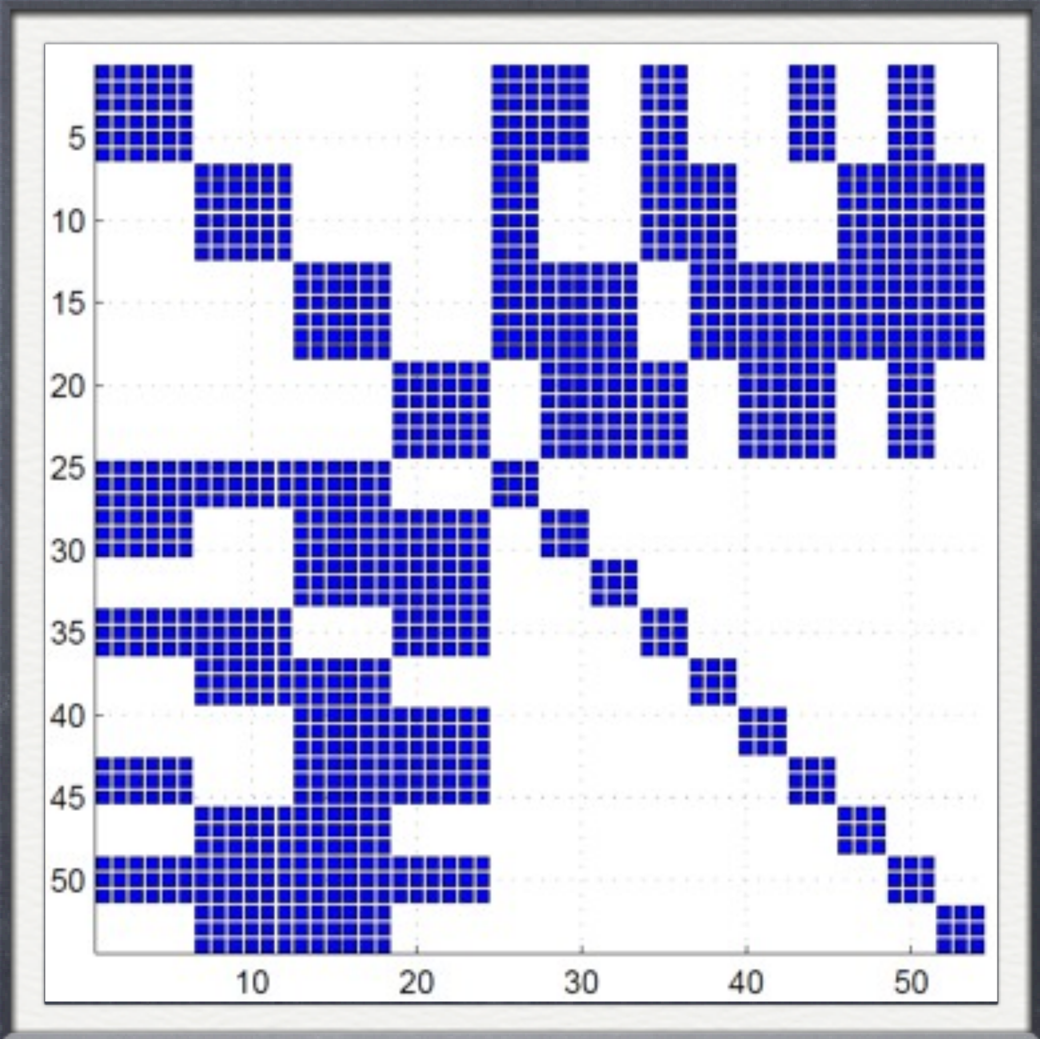
$$(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda\,\mathrm{diag}(\mathbf{J}^{\mathrm{T}}\mathbf{J}))\Delta\mathbf{x} = -\mathbf{J}^{\mathrm{T}}\mathbf{r}(\mathbf{x}_k)$$

- Small problem 200 camera, 20K 3D points
- Calibrated camera 3+3 dof, 3D point 3 dof
- 6x200+3x20K ➠
  J^TJ is 61200x61200 matrix
- Size 30 TB!

Jacobian J and $J^T J$

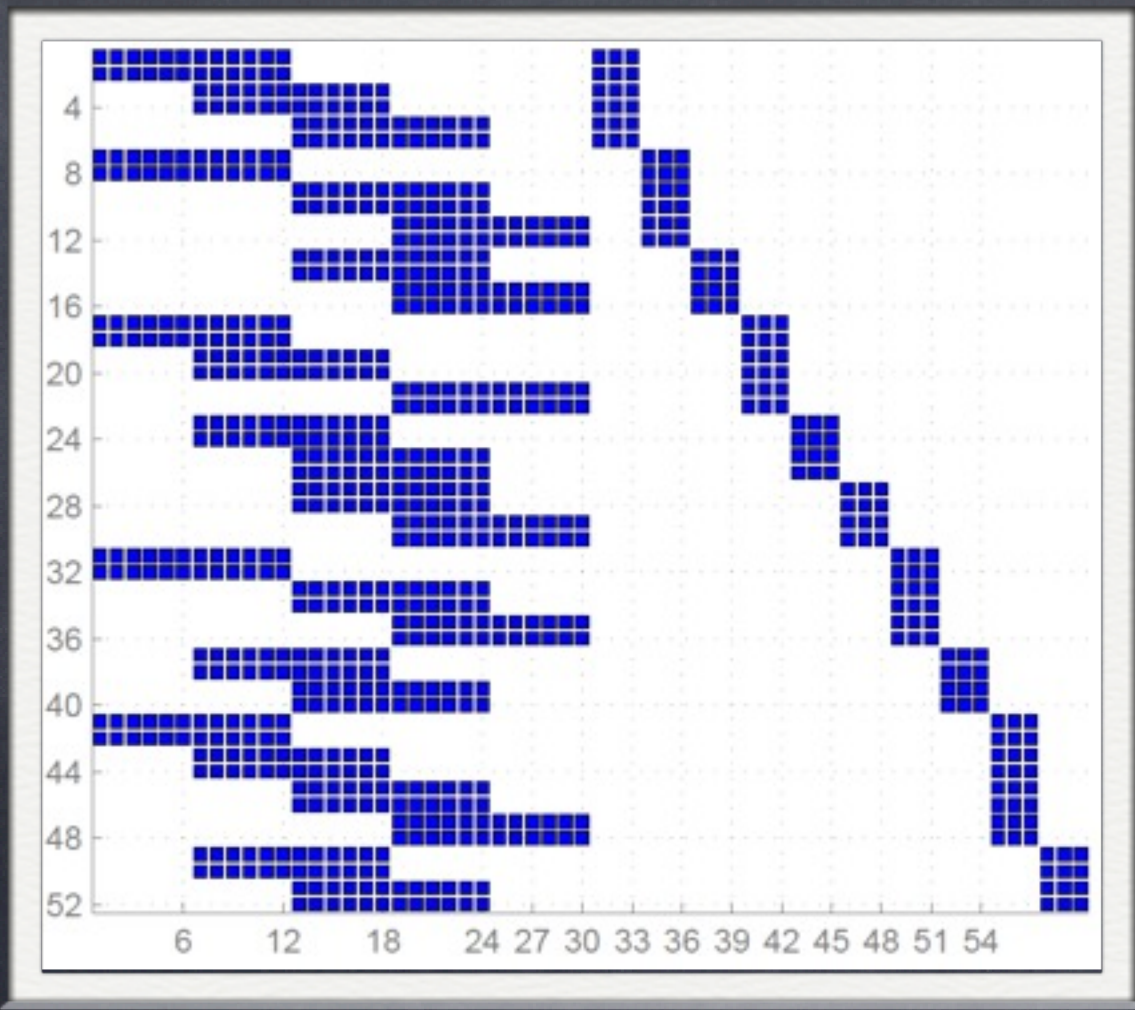
$$\min_{\mathbf{C},\mathbf{X}} \frac{1}{2} \sum_{j=1}^{J} \sum_{k \in \mathcal{V}_j} \|\mathbf{p}_{j,k} - \text{proj}(\mathbf{C}_j, \mathbf{X}_k)\|_2^2$$

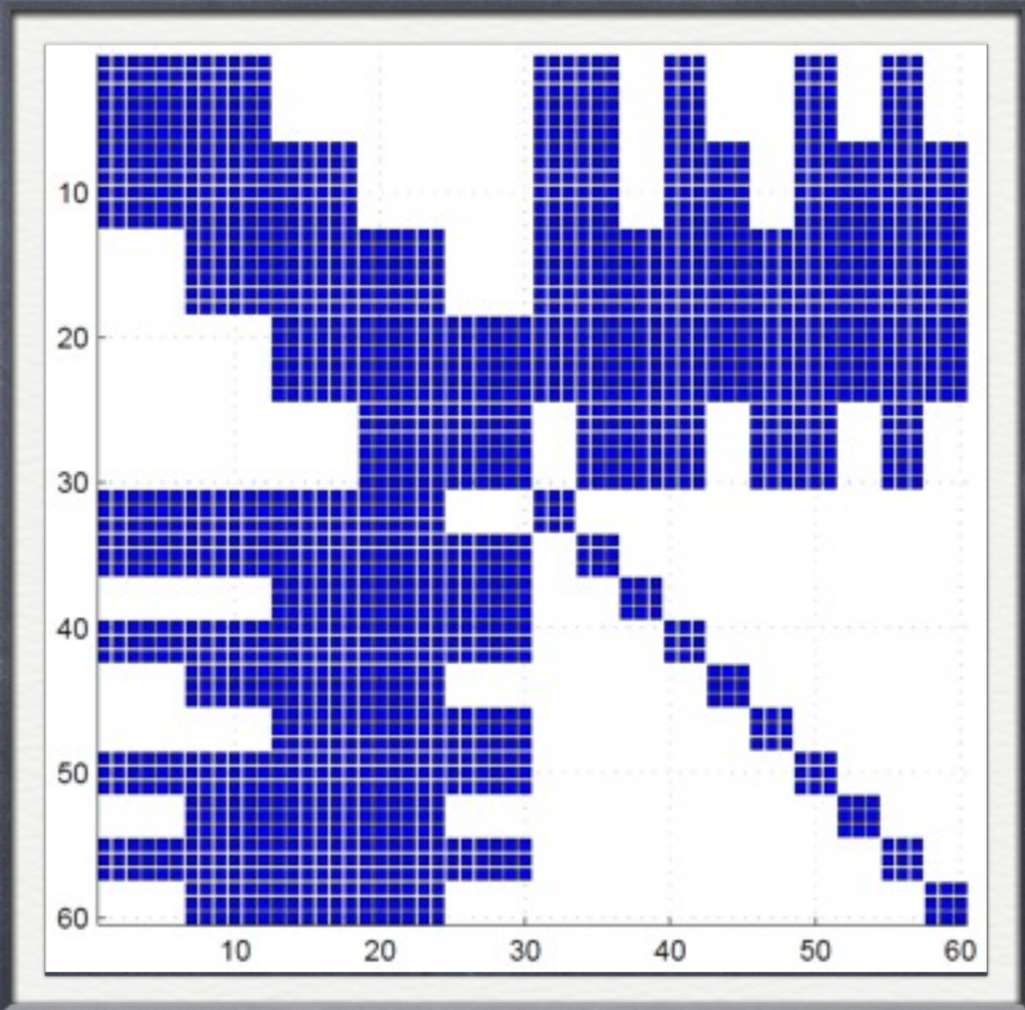




Jacobian J and $J^T J$ for RS

$$\min_{\mathbf{C},\mathbf{X}} \frac{1}{2} \sum_{j=1}^{J} \sum_{k \in \mathcal{V}_j} ||\mathbf{p}_{j,k} - \text{proj}(\mathbf{C}_j(y_k), \mathbf{X}_k)||_2^2$$





Solving the normal equations

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda\,\mathrm{diag}(\mathbf{J}^{\mathrm{T}}\mathbf{J}))\Delta\mathbf{x} = -\mathbf{J}^{\mathrm{T}}\mathbf{r}(\mathbf{x}_k)\,, \tag{7}$$

$$\begin{bmatrix} \mathbf{J}_c^{\mathrm{T}}\mathbf{J}_c & \mathbf{J}_c^{\mathrm{T}}\mathbf{J}_m \\ \mathbf{J}_m^{\mathrm{T}}\mathbf{J}_c & \mathbf{J}_m^{\mathrm{T}}\mathbf{J}_m \end{bmatrix} + \lambda\,\mathrm{diag}\begin{bmatrix} \mathbf{J}_c^{\mathrm{T}}\mathbf{J}_c & 0 \\ 0 & \mathbf{J}_m^{\mathrm{T}}\mathbf{J}_m \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^{\mathrm{T}} & \mathbf{V} \end{bmatrix}. \tag{8}$$

The normal equations (7) now read

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^{\mathrm{T}} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{c} \\ \Delta\mathbf{m} \end{bmatrix} = -\begin{bmatrix} \mathbf{J}_c^{\mathrm{T}} \\ \mathbf{J}_m^{\mathrm{T}} \end{bmatrix} \mathbf{r}\,. \tag{9}$$

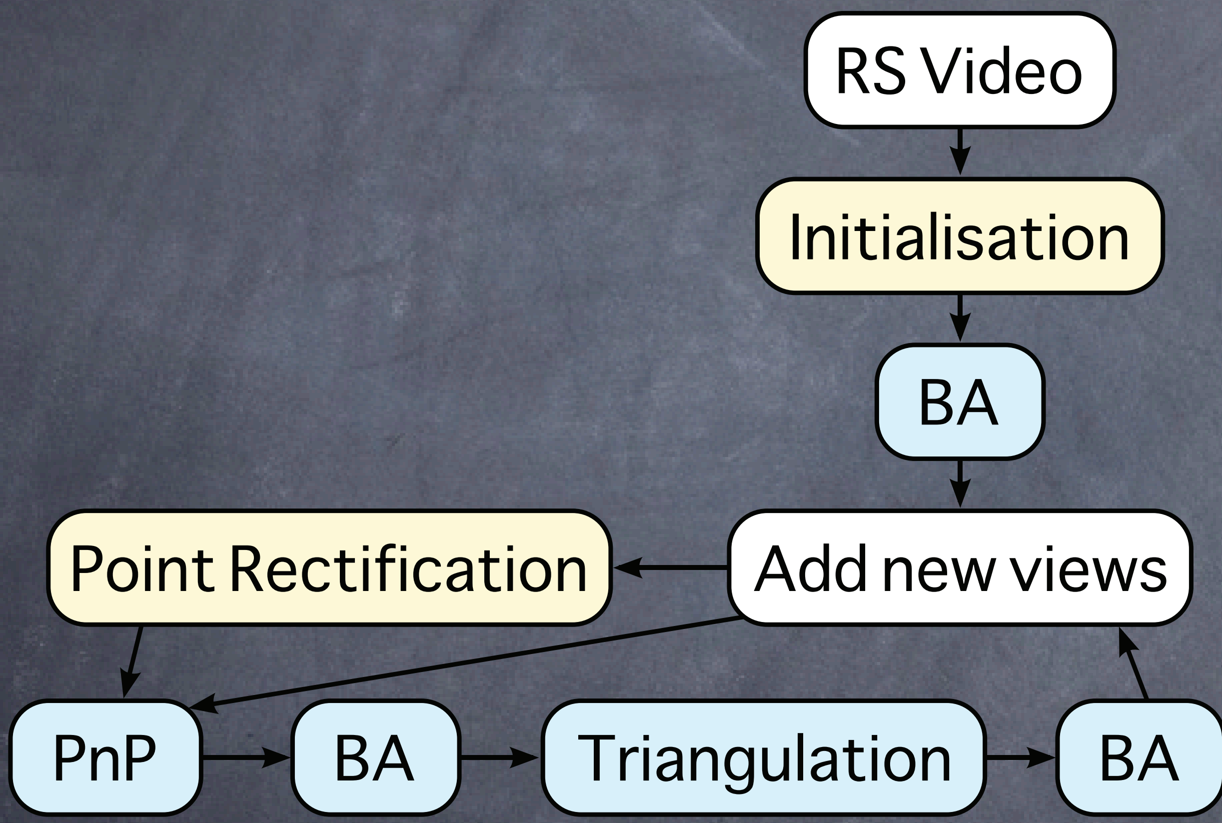
The camera parameter update can now be computed separately by elimination

$$\left(\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^{\mathrm{T}}\right)\Delta\mathbf{c} = \left(\mathbf{W}\mathbf{V}^{-1}\mathbf{J}_m^{\mathrm{T}} - \mathbf{J}_c^{\mathrm{T}}\right)\mathbf{r}\,. \tag{10}$$

Once we have the camera update, the update for the 3D points is obtained as:

$$\Delta\mathbf{m} = -\mathbf{V}^{-1}(\mathbf{J}_m^{\mathrm{T}}\mathbf{r} + \mathbf{W}^{\mathrm{T}}\Delta\mathbf{c})\,. \tag{11}$$

RS structure from motion pipeline



"Classical" SFM

Hedborg et al. ICCV workshop, 2011

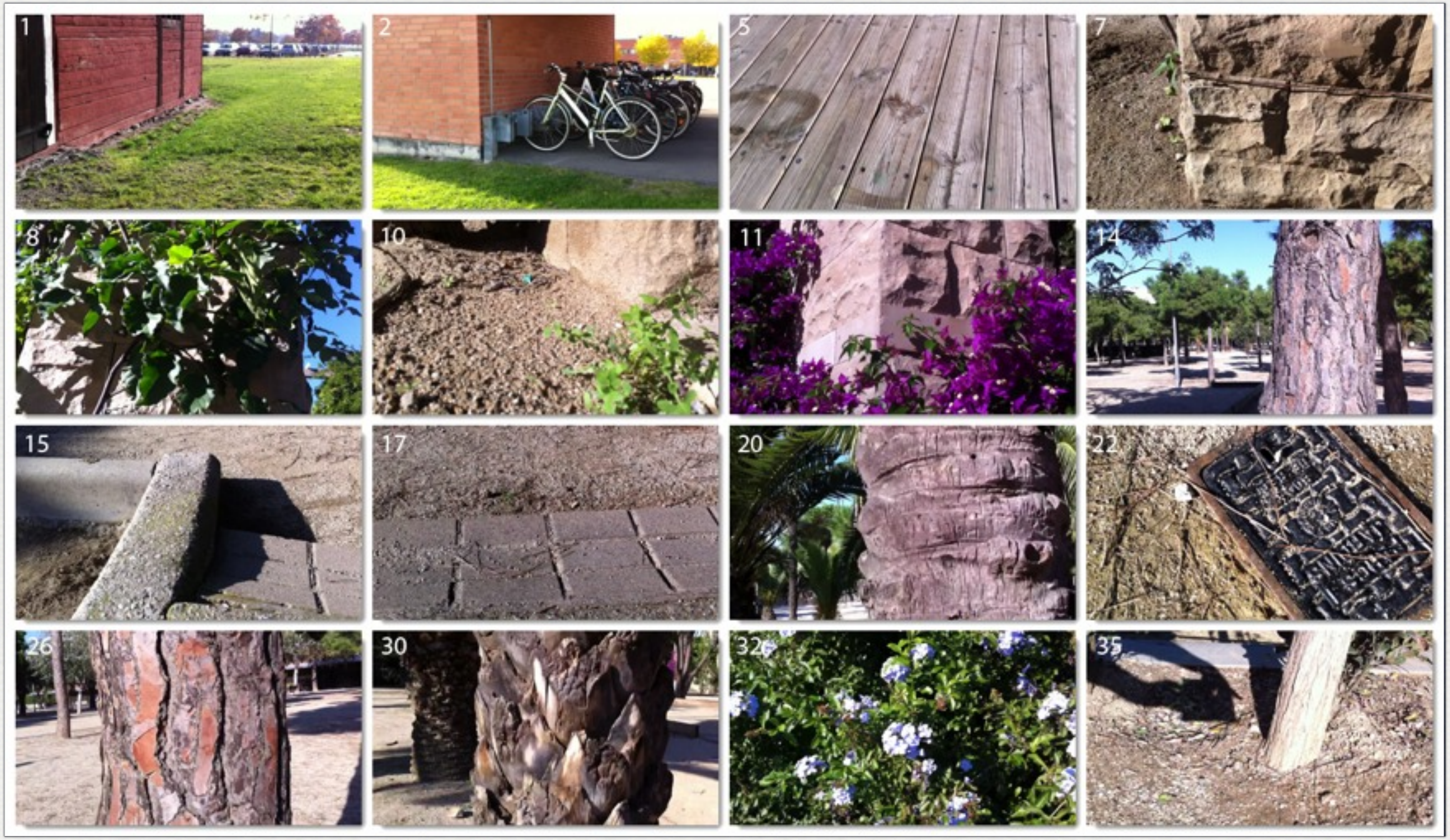
Rolling Shutter SfM

The Rig

"Ground truth":
Canon S95, prosumer
large GS sensor
Wide angle
➡ error : ~0.2 pix +
Smooth trajectories
Without regularization

RS camera :
iPhone 4, 720p
rolling shutter camera
~30ms read-out speed
➡ error : ~0.6 pix

Large Evaluation Set



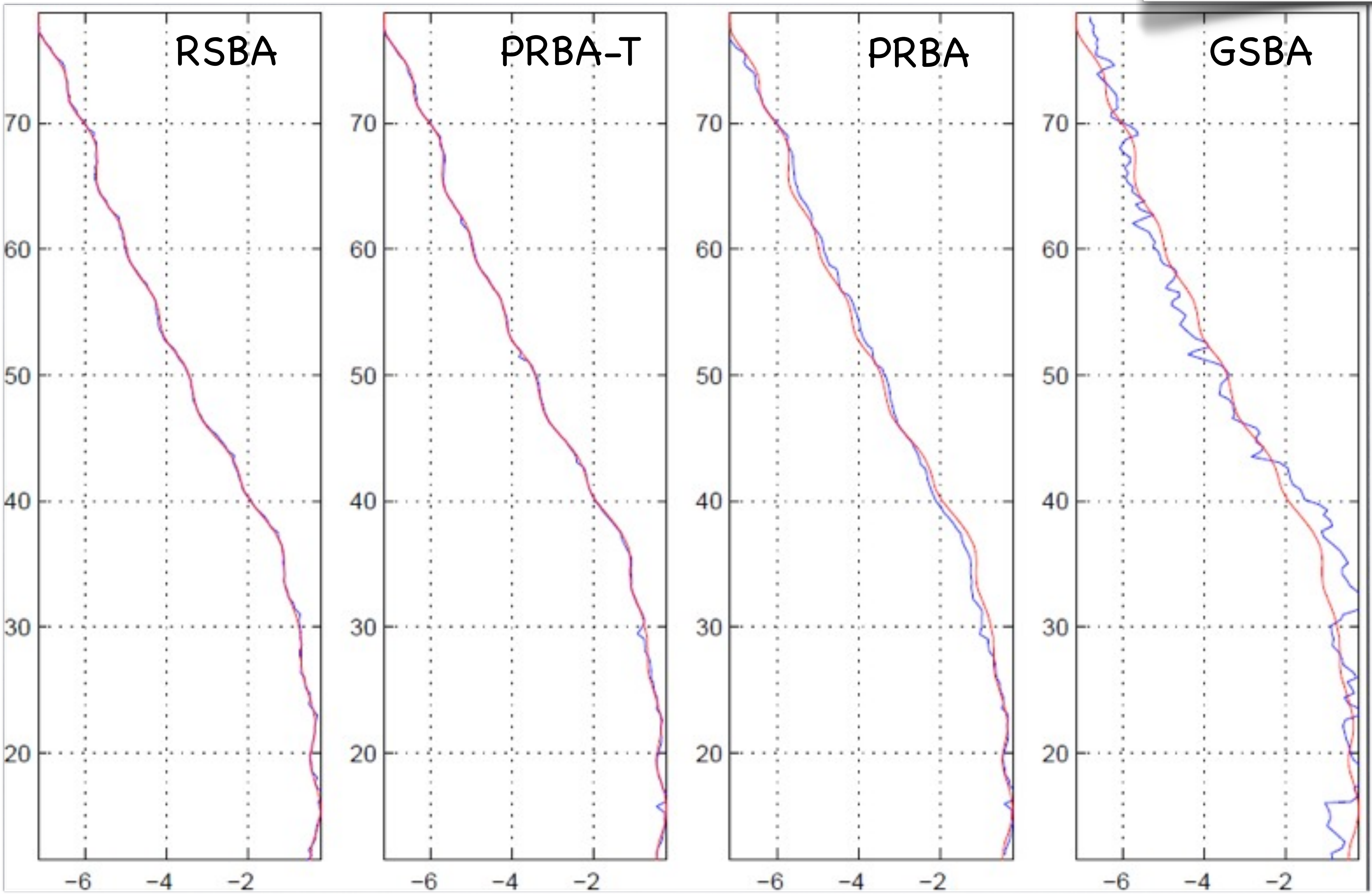
Compared methods

1. Rolling shutter bundle adjustment (RSBA)
2. Same as (3) but with an outlier scheme (PRBA-T)
3. Rolling shutter compensated images for which classical Bundle Adjustment where run (Hedborg at al. ICCV workshop, 2011) (PRBA)
4. Classical Bundle Adjustment (GSBA)

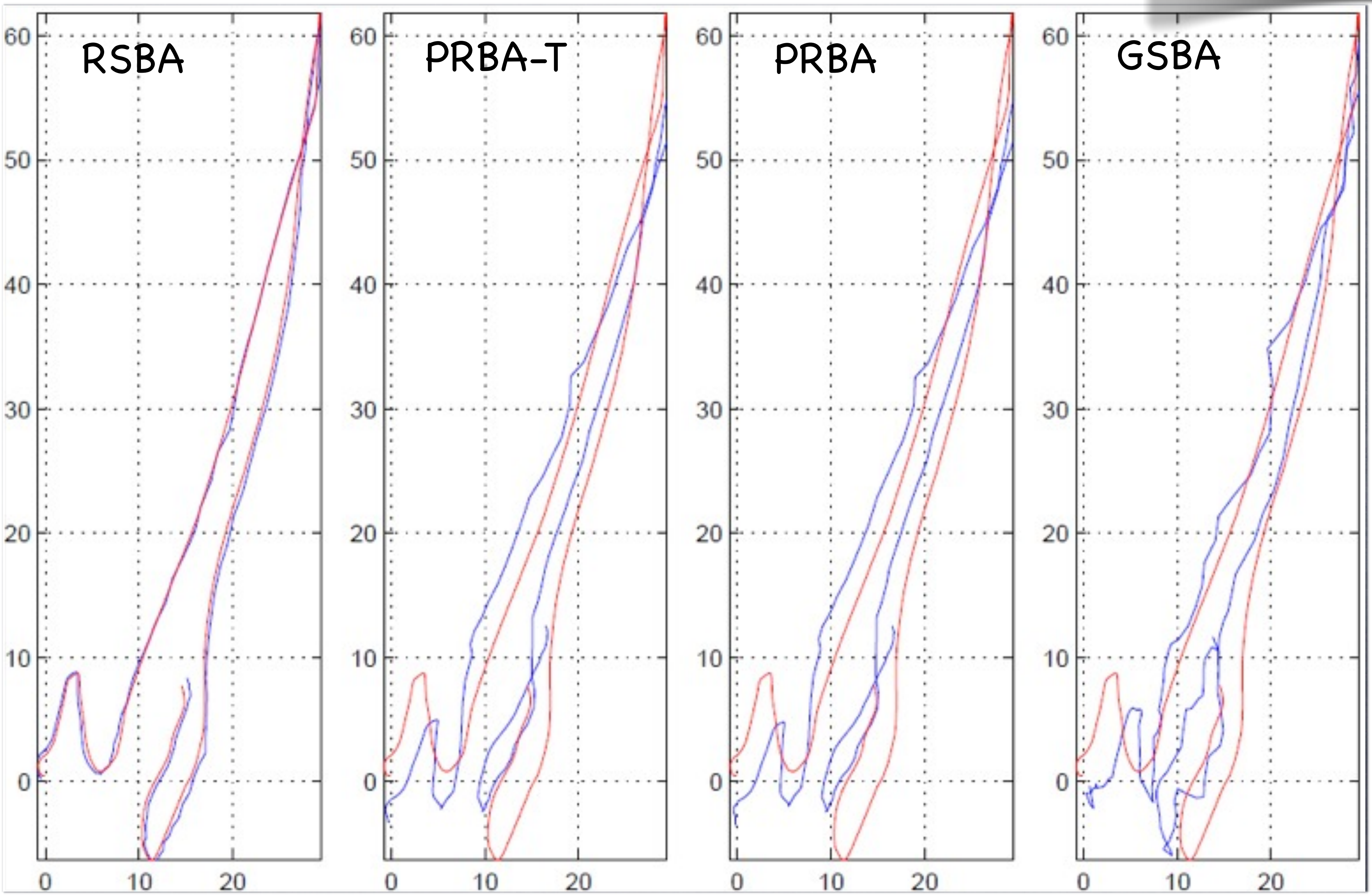
Scene 1

Scene 22

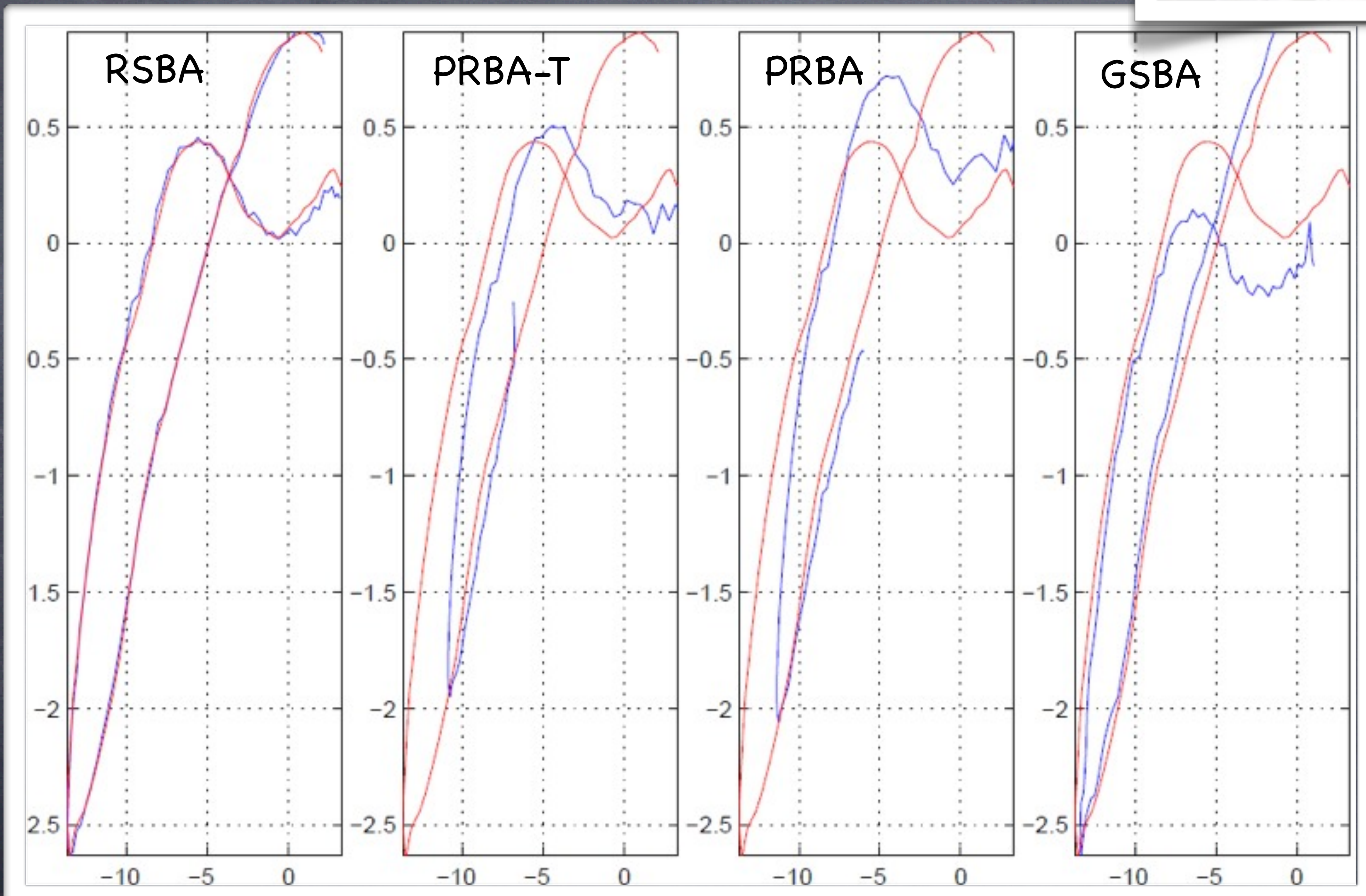
Results Scene1



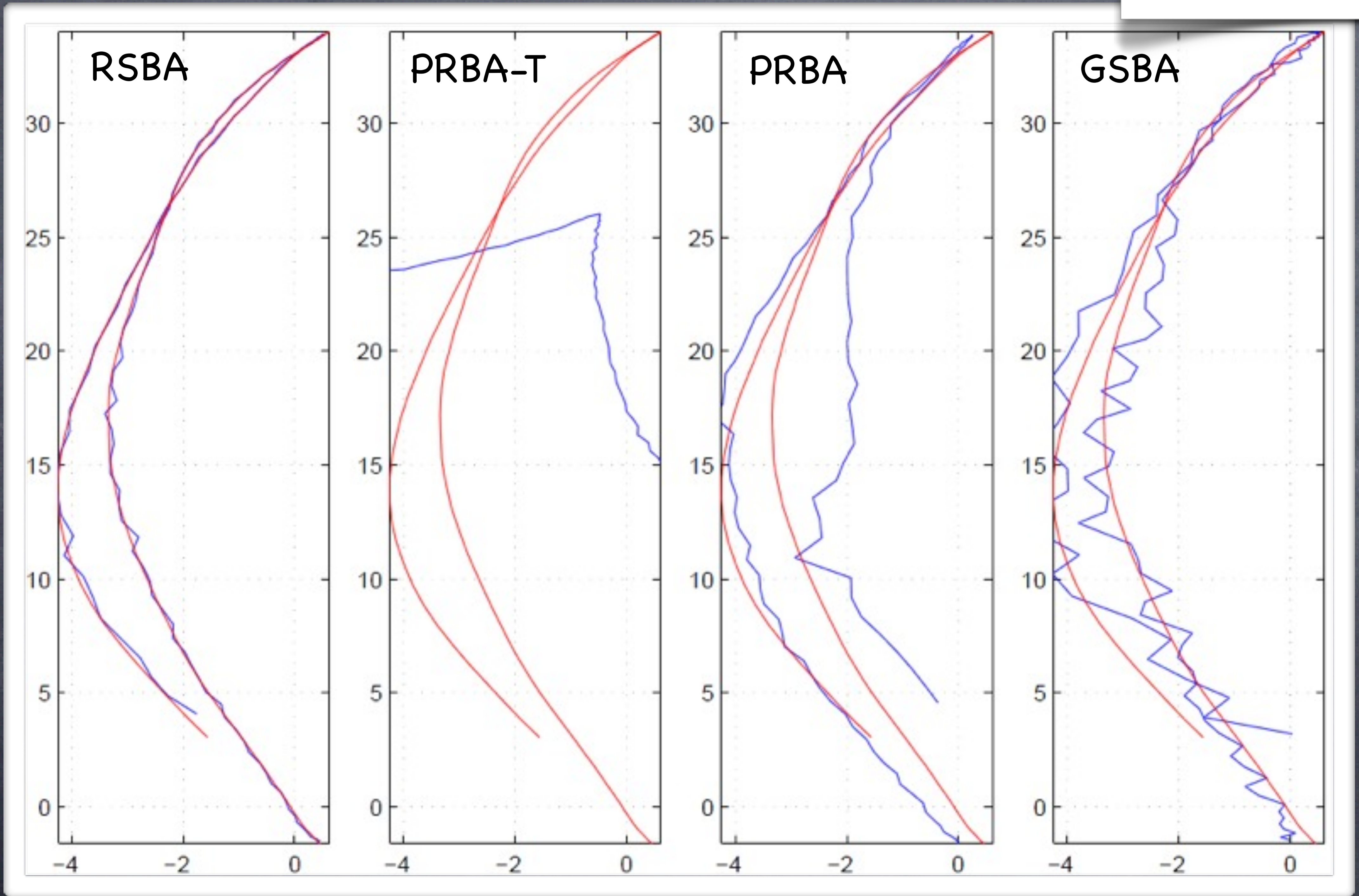
Results Scene8



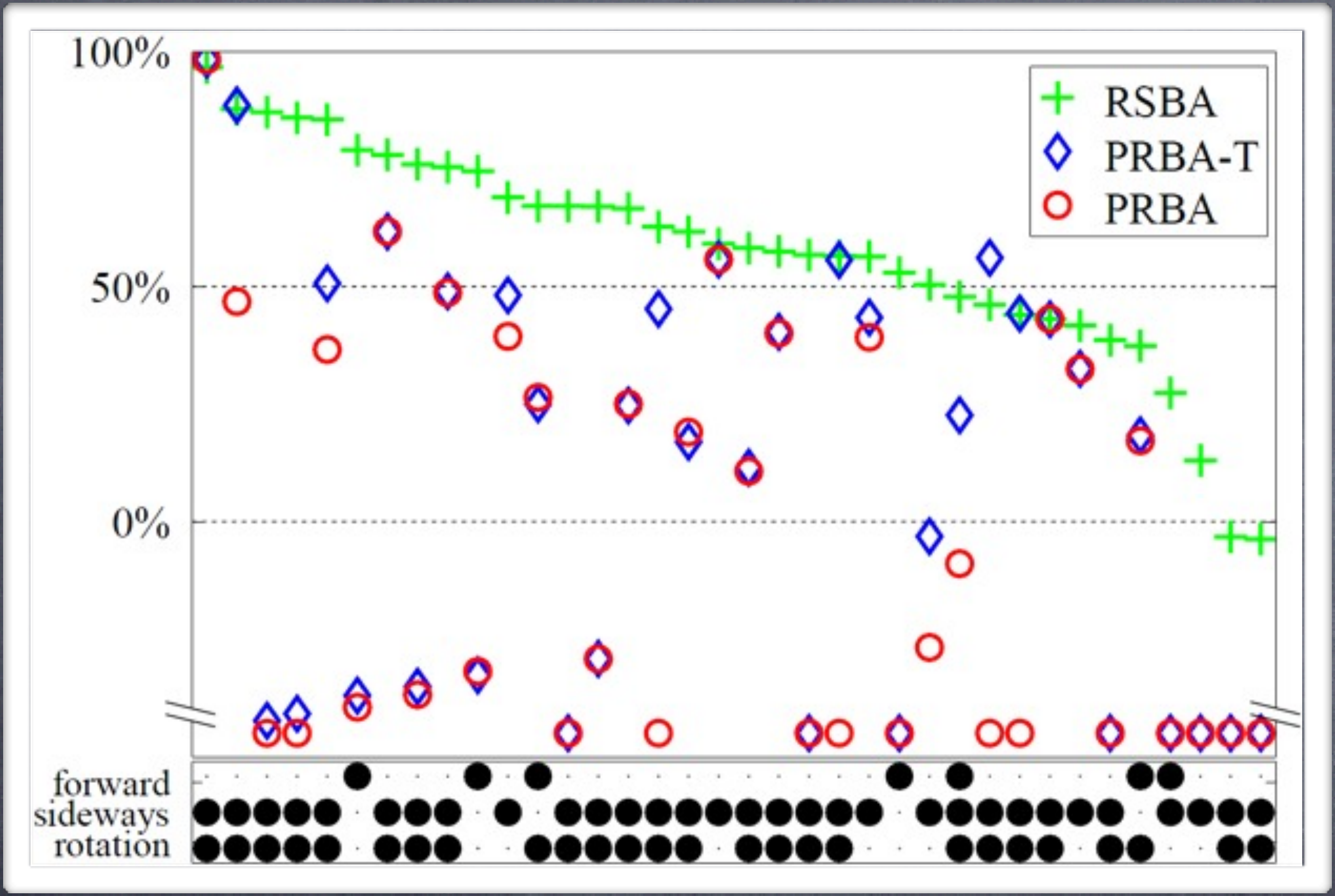
Results Scene20



Results Scene22



Compared to Classical BA



Relative Improvements

Translation error area of the surface between the camera trajectories

Timings

- GSBA : 221 cameras, 13777 3D points, 5 it. to conv. (intel W3520, one core)
  4:91 sec

- RSBA : 221 cameras, 14320 3D points, 4 it. to conv. (intel W3520 , one core)
  8:37 sec

- Smaller scene 74 cameras , RSBA : 2:36 sec for 5 602 GSBA took 1:49 sec for 5 577 points

Summary

- Classical Bundle Adjustment on rolling shutter video is brittle
- Rotation compensation works well in certain situations
- The speed penalty is moderate when using RSBA
- Rolling shutter aware structure from motion is possible

References

- Geometric Models of Rolling-Shutter Cameras
  Christopher Geyer, Marci Meingast, and Shankar Sastry, OMNIVIS 2005

- Simultaneous Object Pose and Velocity Computation Using a Single View from a Rolling Shutter Camera
  Omar Ait-Aider, Nicolas Andreff, Jean Marc Lavest and Philippe Martinet, ECCV 2006

- Parallel Tracking and Mapping on a camera phone
  Georg Klein and David Murray, ISMAR 2009

- Structure and Motion Estimation from Rolling Shutter Video
  Johan Hedborg, Erik Ringaby, Per-Erik Forssén, Michael Felsberg
  ICCV workshop 2011

- Rolling Shutter Bundle Adjustment
  Johan Hedborg, Per-Erik Forssén, Michael Felsberg, Erik Ringaby,CVPR 2012

- Bundle Adjustment Rules
  Chris Engels, Henrik Stewénius, David Nistér, PCV 2006