Generic stereoscopic tools for planetary topography

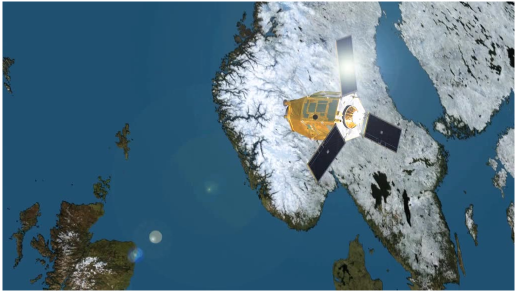
Frédéric Schmidt (GEOPS - UPSud), **Carlo de Franchis**, Jean-Michel Morel (CMLA - ENS Paris-Saclay)

09 November 2016

Centre de Mathématiques et de Leurs Applications

The Pléiades Earth observation satellites

- Pléiades 1A launched in december 2011
- Orbit at 694 km
- Swath width: 20 km
- Ground Sampling Distance (GSD): 70 cm / pix
- Quasi-simultaneous stereo acquisitions → 3D models

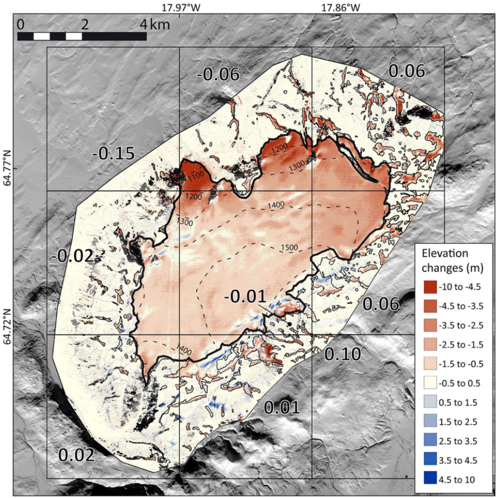


Vidéo produced by CNES, copyright ©CNES 2011

Why 3D digital models?

They are an essential tool for:

- large-scale measurements:
  - snow height on glaciers [Berthier et al. 2014]
  - forests evolution [Gumbricht 2012]
  - assessment after natural disasters [Yésou et al. 2015]
- change detection [Chaabouni-Chouayakh et al. 2010]
- cartography (orthorectification) [Leprince et al. 2007]
- more generally, image comparison



Elevation differences on the Tungnafellsjökull Ice Cap

Why 3D digital models?

They are an essential tool for:

- large-scale measurements:
  - snow height on glaciers [Berthier et al. 2014]
  - forests evolution [Gumbricht 2012]
  - assessment after natural disasters [Yésou et al. 2015]
- change detection [Chaabouni-Chouayakh et al. 2010]
- cartography (orthorectification) [Leprince et al. 2007]
- more generally, image comparison

Bassies (Pyrénées), 2015-03-11

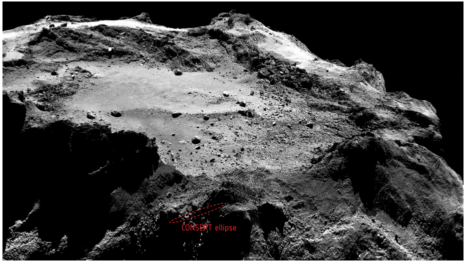
Why 3D digital models?

They are an essential tool for:

- large-scale measurements:
  - snow height on glaciers [Berthier et al. 2014]
  - forests evolution [Gumbricht 2012]
  - assessment after natural disasters [Yésou et al. 2015]
- change detection [Chaabouni-Chouayakh et al. 2010]
- cartography (orthorectification) [Leprince et al. 2007]
- more generally, image comparison

Bassies (Pyrénées), 2014-10-26

Why 3D digital models? For Rosetta!



Philae search area on comet 67P/Churyumov-Gerasimenko

Philae final landing site

How to compute 3D digital models?

Active methods:

- Kinect
- Lidar
- Synthetic Aperture Radar (SAR)

Passive image-based methods:

- (multi-view) stereo
- structure from motion
- photogrammetry
- computer vision. . .

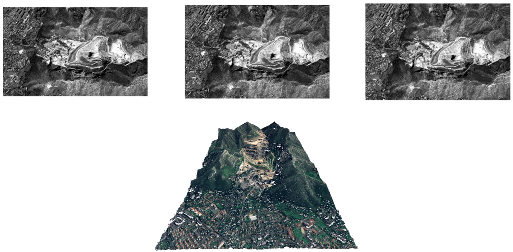
How to compute 3D digital models?

Active methods:

- Kinect
- Lidar
- Synthetic Aperture Radar (SAR)

Passive image-based methods:

- (multi-view) stereo
- structure from motion
- photogrammetry
- computer vision. . .

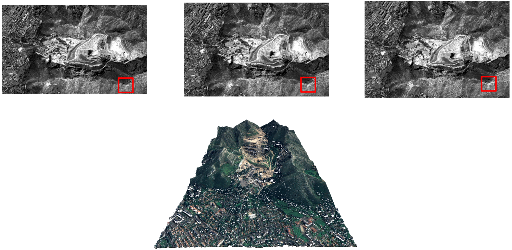


3D reconstruction from images

General principle:

- find corresponding pixels
- intersect the back-projected 3D lines

Need a camera model, and its parameters.



Pinhole camera model: projective mapping from 3D space to 2D images plane, represented by a 3×4 matrix

$$\mathtt{P} = \mathtt{KR}[\mathtt{I}|-\mathtt{C}]$$

Many names: pinhole, frame, conic, projective...

[Marr and Poggio 1976] [Hartley and Zisserman 2004]

3D reconstruction from images

General principle:

- find corresponding pixels
- intersect the back-projected 3D lines

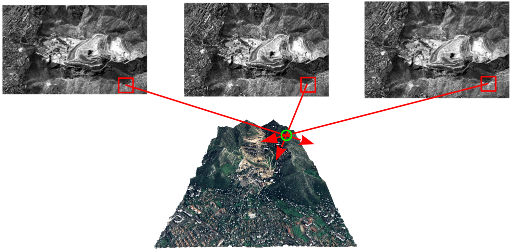
Need a camera model, and its parameters.

Pinhole camera model: projective mapping from 3D space to 2D images plane, represented by a 3×4 matrix

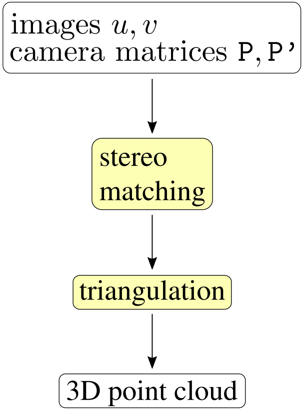
$$\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I}|-\mathtt{C}]$$

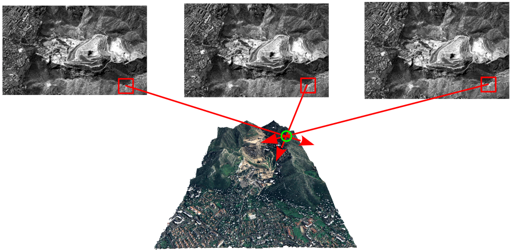
Many names: pinhole, frame, conic, projective. . .

[Marr and Poggio 1976] [Hartley and Zisserman 2004]

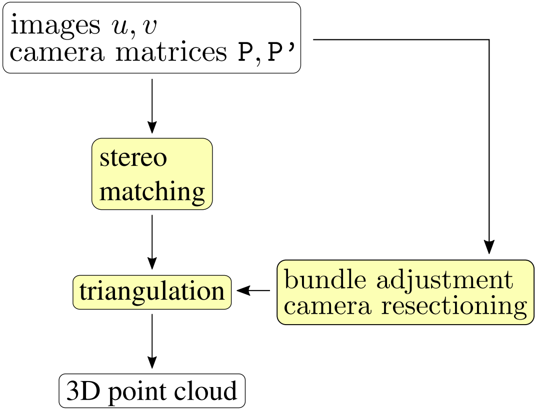


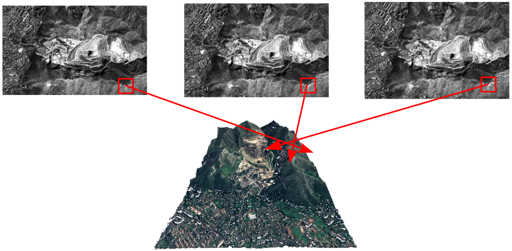
Baseline 3D reconstruction algorithm



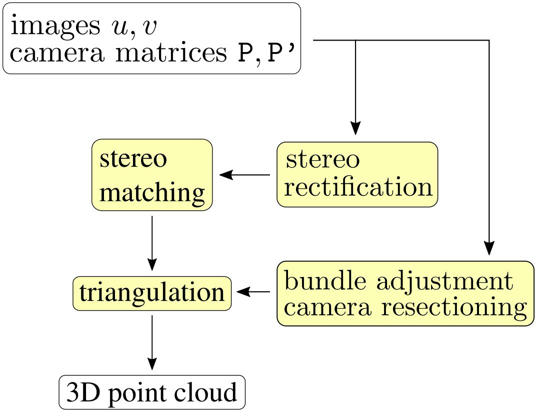


Baseline 3D reconstruction algorithm





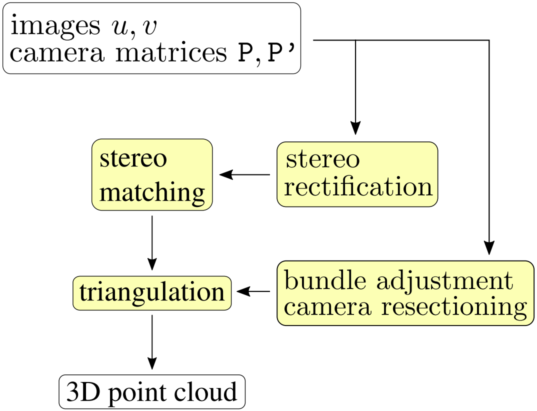
Baseline 3D reconstruction algorithm

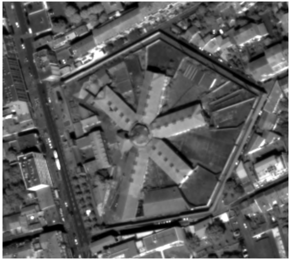


input images

rectified images

Baseline 3D reconstruction algorithm





input images

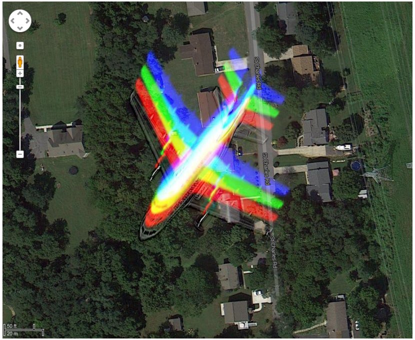
rectified images

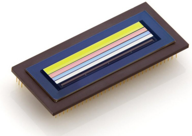
Pushbroom cameras

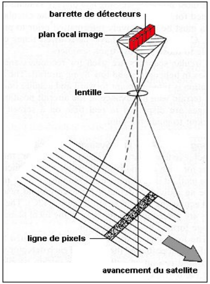
The cameras used on satellites are *pushbroom*, not pinhole:

- image lines, and color channels, are acquired sequentially
- images are **huge**: 40k × 40k pixels
- most of the computer vision and image processing literature deals with pinhole cameras.

Goal: fill the gap between computer vision and remote sensing



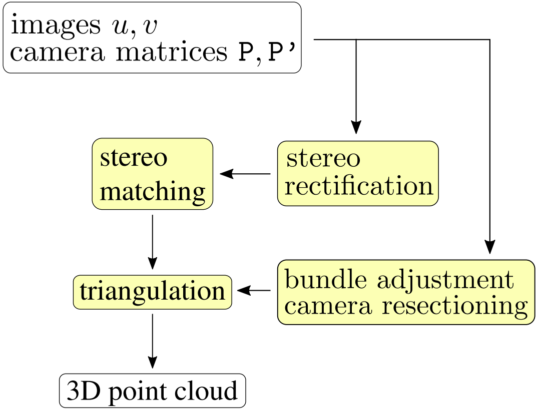




Pushbroom cameras

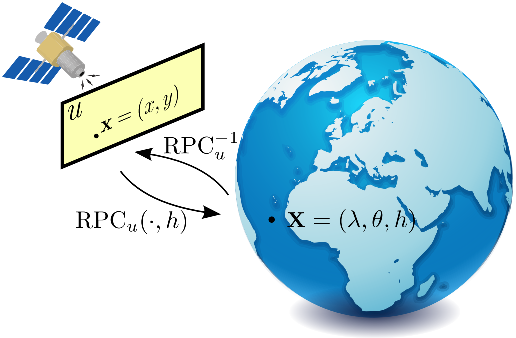
Camera modeling is more complex:

camera matrix P (12 coefficients) $\longrightarrow$ rational polynomial functions (RPC) (170 coefficients)



The Rational Polynomial Camera Model

- For end-users, image vendors provide a **localization** function. It is as a **Rational Polynomial Function** with degree 3.
- Its inverse, with respect to $\mathbf{x}$, is given as well.

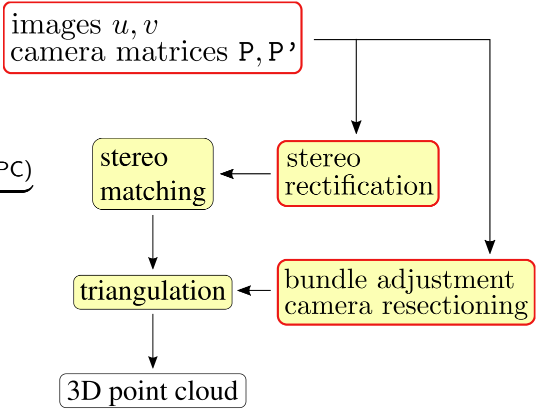


Pushbroom cameras

- camera modeling is more complex:

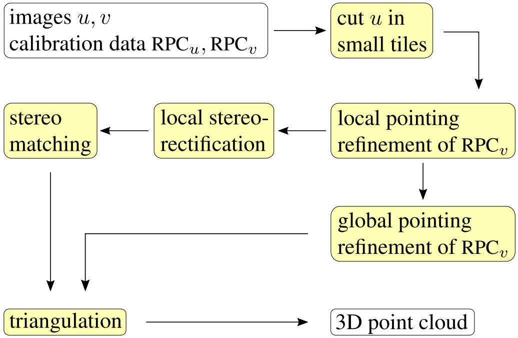
$\underbrace{\text{camera matrix P}}_{\text{12 coefficients}} \longrightarrow \underbrace{\text{rational polynomial functions (RPC)}}_{\text{170 coefficients}}$

- bundle adjustment is more complex
- epipolar rectification is not possible



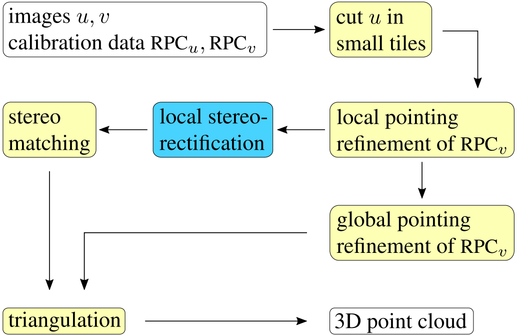
Outline of the algorithm

1. Epipolar rectification for pushbroom images
2. Local correction of the pointing error
3. Stereo matching
4. Triangulation



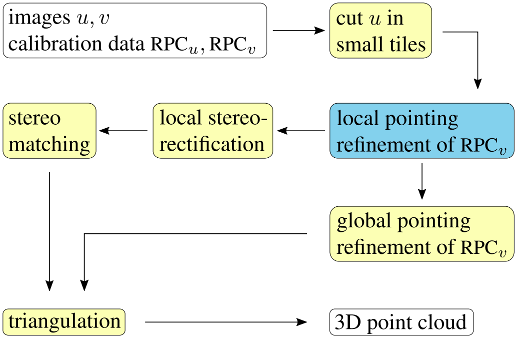
Outline of the algorithm

1. Epipolar rectification for pushbroom images
2. Local correction of the pointing error
3. Stereo matching
4. Triangulation



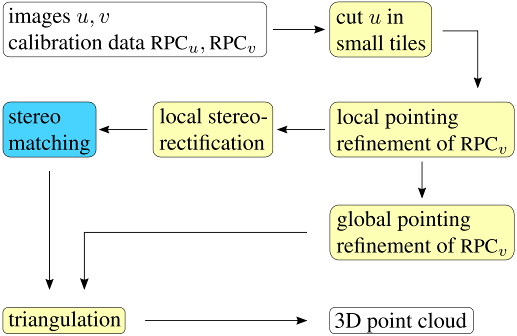
Outline of the algorithm

1. Epipolar rectification for pushbroom images
2. Local correction of the pointing error
3. Stereo matching
4. Triangulation



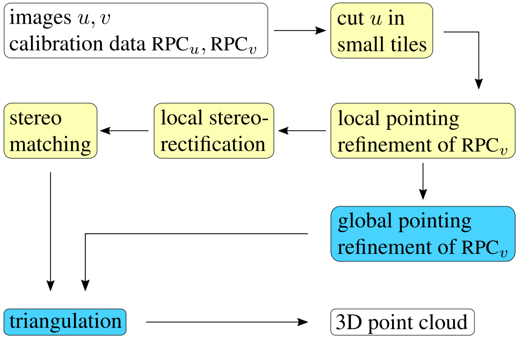
Outline of the algorithm

1. Epipolar rectification for pushbroom images
2. Local correction of the pointing error
3. Stereo matching
4. Triangulation

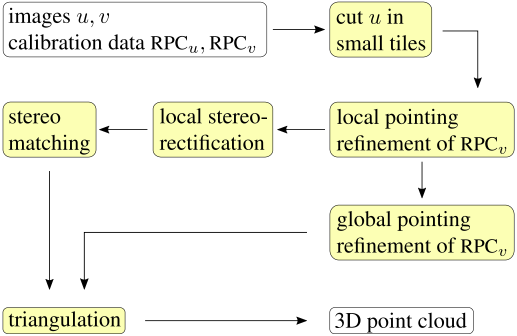


Outline of the algorithm

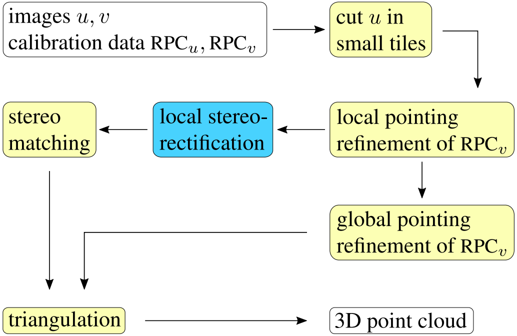
1. Epipolar rectification for pushbroom images
2. Local correction of the pointing error
3. Stereo matching
4. Triangulation



1. Epipolar rectification of pushbroom images



1. Epipolar rectification of pushbroom images

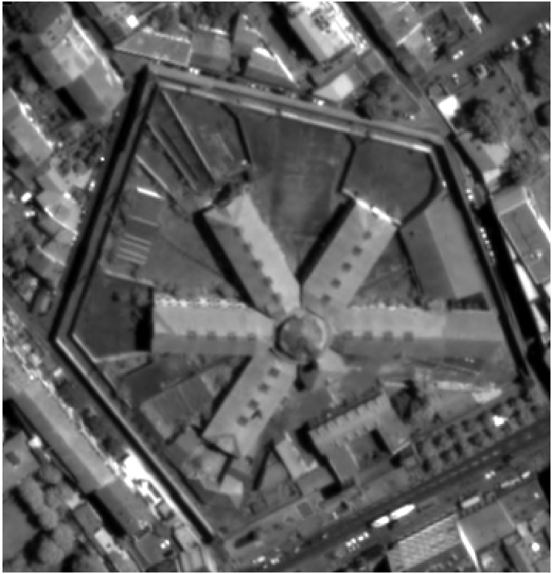


Epipolar rectification: what is it?

Process of **resampling** the images in such a way that depth variations cause **apparent motion** in the the **horizontal** direction only.

Epipolar rectification: what is it?

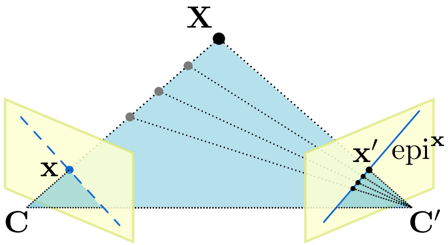
Process of **resampling** the images in such a way that depth variations cause **apparent motion** in the **horizontal** direction only.



Pinhole cameras

- $\mathbf{C}, \mathbf{C}'$ and $\mathbf{x}$ define a plane, called the **epipolar plane**.
- Its intersection with the second image is the **epipolar line** of $\mathbf{x}$, denoted by $\mathrm{epi}^{\mathbf{x}}$.
- All the $\mathbf{x}' \in \mathrm{epi}^{\mathbf{x}}$ share the same epipolar plane, hence the **same** epipolar line in the first image.

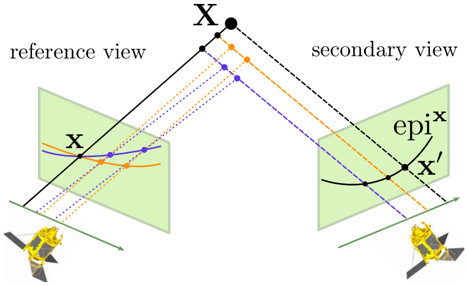
Conclusion: there is a one-to-one correspondence between epipolar lines.



Pushbroom cameras

- Satellite cameras are not **pinhole**, but **pushbroom**.
- As the camera center moves, the epipolar plane becomes a **doubly ruled surface**, namely a **hyperbolic paraboloid**.
- Epipolar lines become **curves**, still denoted by $\mathrm{epi}^{\mathbf{x}}$.
- All the $\mathbf{x}' \in \mathrm{epi}^{\mathbf{x}}$ have a **different** epipolar surface, hence a **different** epipolar line in the first image.

Conclusion: there is **no** one-to-one correspondence between epipolar curves.



Epipolar rectification: why and how

Why epipolar rectification:

- To reduce the exploration from 2D to 1D
- It is just an intermediate step

Then it could be done **locally**. Let's try to **approximate** the pushbroom model with a pinhole on **small image tiles**.

How to do epipolar rectification:

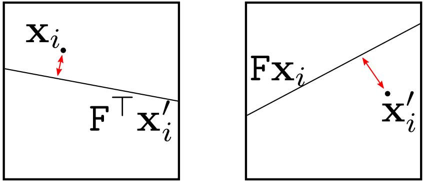
1. Find keypoint matches $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ with SIFT [Lowe 2004, Rey Otero 2014]
2. Estimate the fundamental matrix $\mathtt{F}$ [Hartley and Zisserman 2004]

$$\mathbf{x}_i'^{\top} \mathbf{F} \mathbf{x}_i = 0$$

3. Estimate resampling homographies $\mathtt{H}$ and $\mathtt{H}'$ [Loop Zhang 1999]

$$\mathtt{F} = \mathtt{H}'^{\top} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \mathtt{H}$$

Results

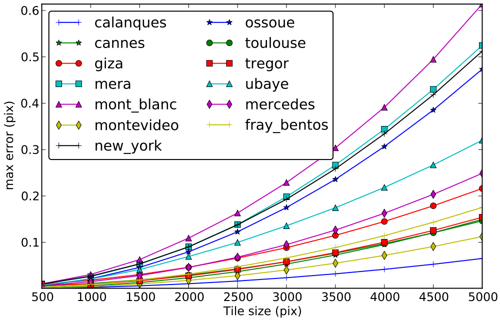


To evaluate the method, measure the **epipolar error**

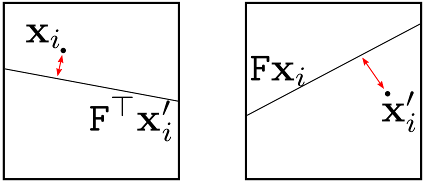
$$\max_{i \in \{1,\dots,n\}} \max\{d(\mathbf{x}'_i, \mathrm{F}\mathbf{x}_i), d(\mathbf{x}_i, \mathrm{F}^\top \mathbf{x}'_i)\},$$

where $d(\mathbf{x}', \mathrm{F}^\top \mathbf{x})$ is the **vertical disparity**:

$$d(\mathbf{x}', \mathrm{F}\mathbf{x}) = \frac{|\mathbf{x}'^\top \mathrm{F}\mathbf{x}|}{\sqrt{(\mathrm{F}_1^\top \mathbf{x})^2 + (\mathrm{F}_2^\top \mathbf{x})^2}}$$

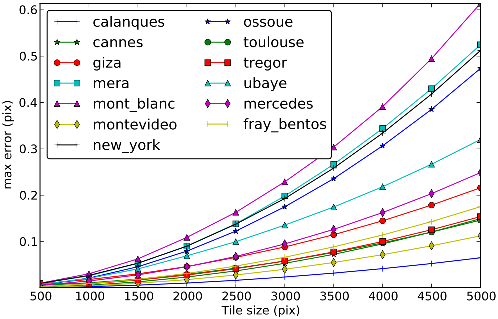


Results



Conclusion:

- After epipolar rectification, the maximal error w.r.t true camera model (RPC) is only 0.05 pixel!
- Working with small tiles (1000×1000 pixels) permits to do the usual epipolar rectification with enough accuracy for stereo matching.



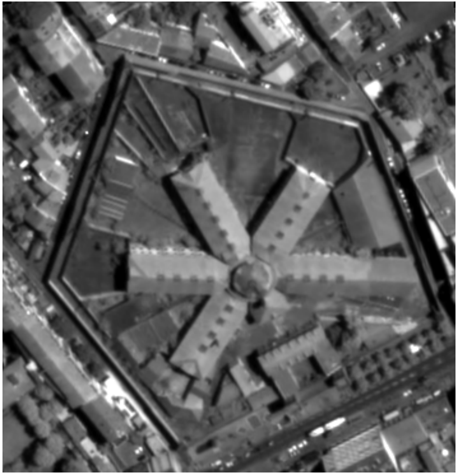
Results

epipolar rectification from keypoints

rectification from RPC

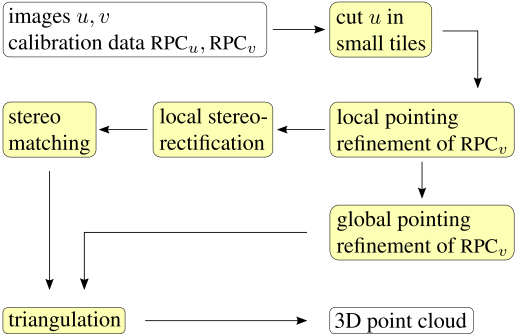
Results

epipolar rectification from keypoints

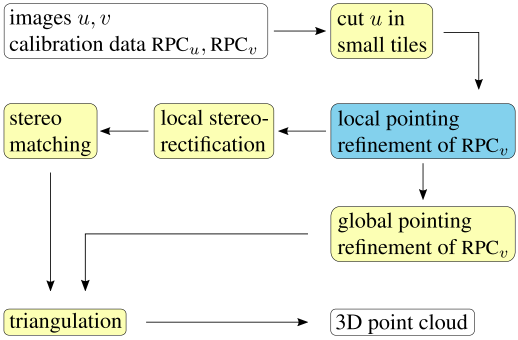


rectification from RPC

2. Local correction of the pointing error



2. Local correction of the pointing error



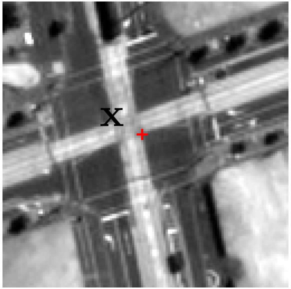
The relative pointing error

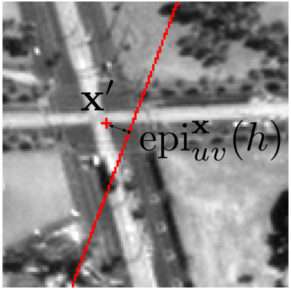
Due to **attitude measurement** inaccuracies, the RPC functions may contain an **error of a few pixels**.

Given two corresponding points $\mathbf{x} \leftrightarrow \mathbf{x}'$, the epipolar curve

$$\mathrm{epi}_{uv}^{\mathbf{x}} : h \mapsto \mathsf{RPC}_v^{-1}(\mathsf{RPC}_u(\mathbf{x}, h), h)$$

may not pass through $\mathbf{x}'$.



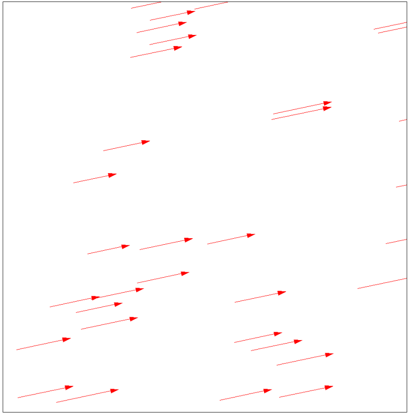


On small tiles

- epipolar curves can be considered as **parallel lines**
- we observed that the pointing error is mostly a **constant offset**

Hence, given a set of **keypoint matches** (obtained with SIFT [Rey Otero 14]), the error is corrected with a translation of the second image:

$$\mathrm{T}^\star = \arg\min_{\mathrm{T}} \frac{1}{N} \sum_{i=1}^{N} d(\mathrm{T}\mathbf{x}'_i, \mathrm{epi}^{\mathbf{x}_i}_{u,v}(\mathbf{R}))$$



Error vectors on a tile of size 1000×1000 pixels

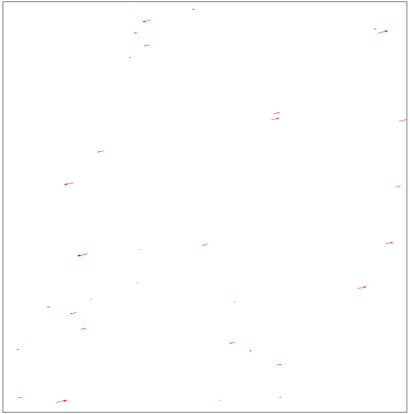
[Rey Otero 14] Ives Rey Otero and Mauricio Delbracio, Anatomy of the SIFT Method, Image Processing On Line, 4 (2014)

On small tiles

- epipolar curves can be considered as **parallel lines**
- we observed that the pointing error is mostly a **constant offset**

Hence, given a set of **keypoint matches** (obtained with SIFT [Rey Otero 14]), the error is corrected with a translation of the second image:

$$\mathrm{T}^\star = \arg\min_{\mathrm{T}} \frac{1}{N} \sum_{i=1}^{N} d(\mathrm{T}\mathbf{x}'_i, \mathrm{epi}^{\mathbf{x}_i}_{u,v}(\mathbf{R}))$$



Error vectors on a tile of size 1000×1000 pixels

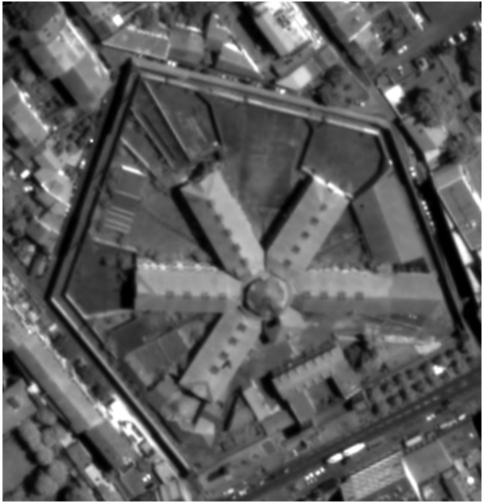
[Rey Otero 14] Ives Rey Otero and Mauricio Delbracio, Anatomy of the SIFT Method, Image Processing On Line, 4 (2014)

Local correction of the relative pointing error

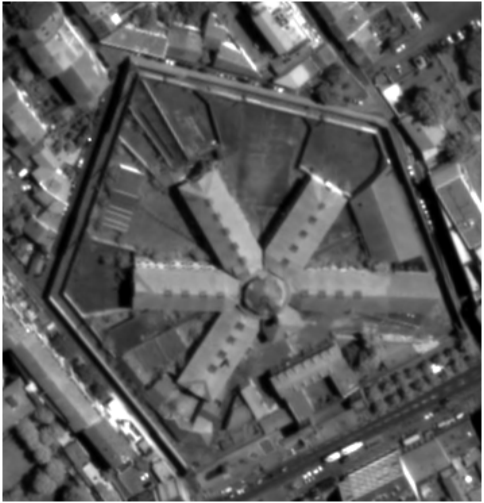
before

after

Local correction of the relative pointing error

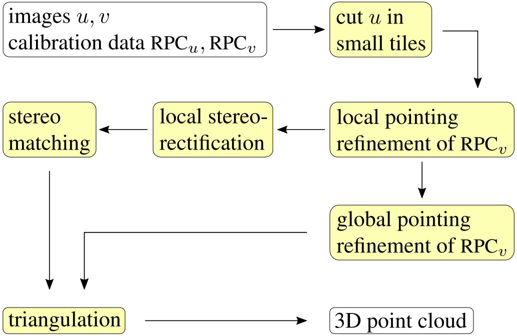


before

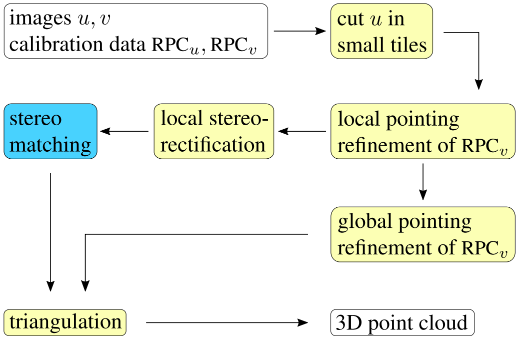


after

3. Stereo matching



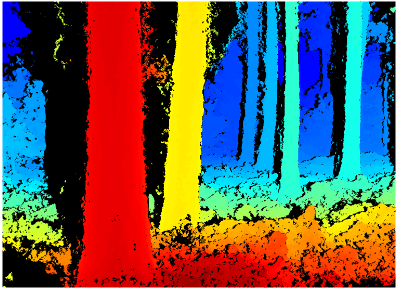
3. Stereo matching



Stereo Matching

Problem: for each 3D point visible in the first image, find its location in the second image (if not occluded).

input: rectified image pair

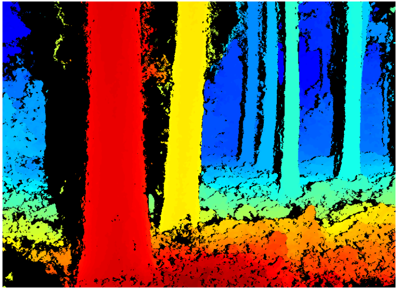


output: disparity map

Stereo Matching

Problem: for each 3D point visible in the first image, find its location in the second image (if not occluded).

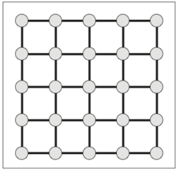
input: rectified image pair



output: disparity map

Stereo Matching

The problem is modeled as the **minimization** of an energy defined on the **image graph**:

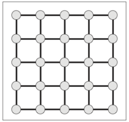


$$E(\mathsf{D}) = \underbrace{\sum_{\mathbf{p}\in\mathcal{V}} C(\mathbf{p}, \mathsf{D}_\mathbf{p})}_{\substack{\text{data term:} \\ \text{AD, NCC, Census}\dots}} + \underbrace{\sum_{(\mathbf{p},\mathbf{q})\in\mathcal{E}} V(\mathsf{D}_\mathbf{p}, \mathsf{D}_\mathbf{q})}_{\substack{\text{regularity term:} \\ \text{imposes smoothness} \\ \text{on the edges of the} \\ \text{image graph, } e.g. \\ V(d,d')=|d-d'|}}$$

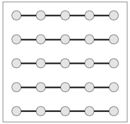
Problem: on 2D image graphs, minimizing E is NP-hard.

Two kinds of **approximations** are used to solve the minimization problem:

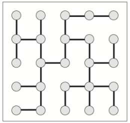
1. Compute a **local minimum**:
   - Refine low resolution result: coarse-to-fine, filtering, . . .
   - FastPD [Komodakis and Tziritas 07]
   - Block Coordinate Descent [Chen and Koltun 14]
2. **Modify** the problem:
   - Dynamic Programming (DP) on trees [Veksler 05, Bleyer 08]
   - Semi-Global Matching (SGM) [Hirschmüller 05]



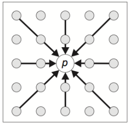
4-connected image graph



DP optimization [Baker & Binford 81]

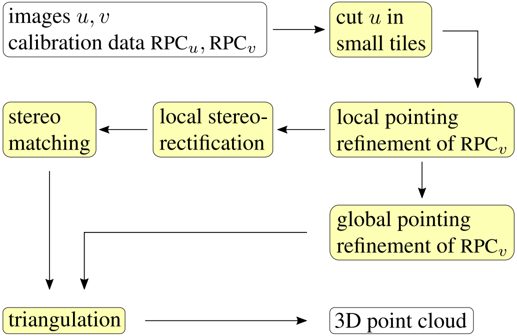


DP on a tree [Veksler 05]

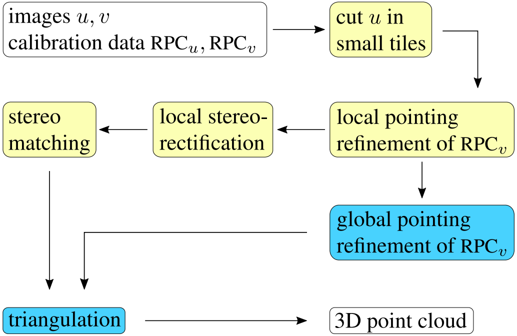


SGM [Hirschmüller 05]

4. Satellite Stereo Pipeline: S2P



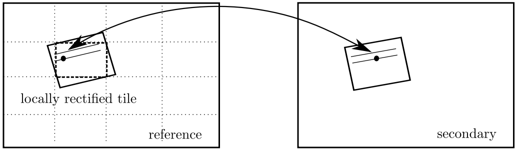
4. Satellite Stereo Pipeline: S2P



Triangulation

Triangulation requires two things:

- **point matches**: transported back from the tiles ✓
- **cameras parameters**: they were refined tilewise ✗



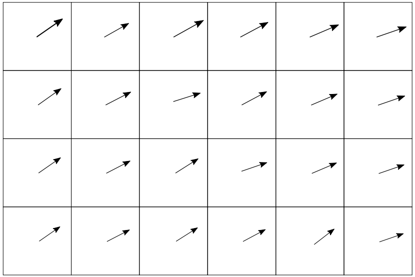
Thus a unique **global** (affine) refinement is estimated from the local translations.

Triangulation

Triangulation requires two things:

- **point matches**: transported back from the tiles ✓
- **cameras parameters**: they were refined tilewise ✗

Thus a unique **global** (affine) refinement is estimated from the local translations.



S2P implementation

Source code on github:

`https://github.com/carlodef/s2p`

Online demo on IPOL:

`http://dev.ipol.im/~carlo/s2p`