

Reproducible Research, IPOL, and Satellite Stereo Images

UPF – DTIC SEMINAR
Barcelona, 9th Jan 2014

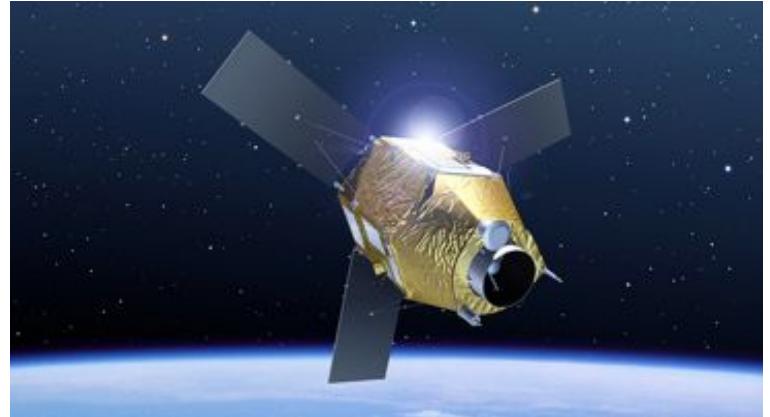
Gabriele Facciolo
CMLA, ENS Cachan, FR

Plan



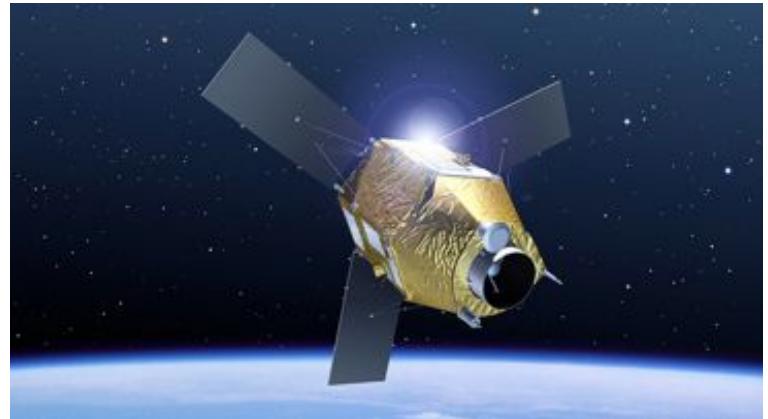
Some slides courtesy of N. Limare and E. Meinhardt

Plan



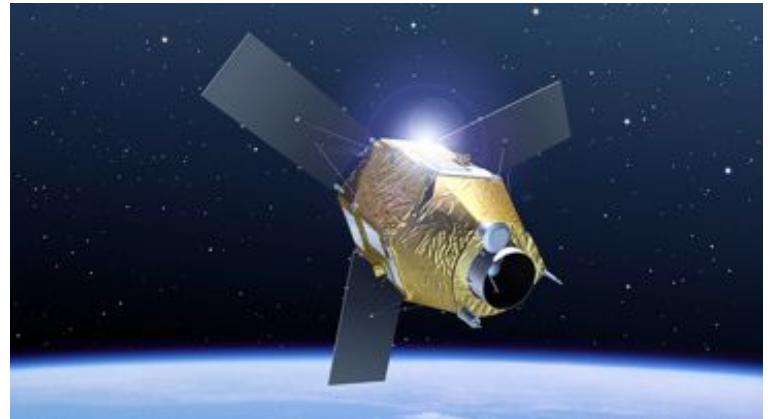
Some slides courtesy of N. Limare and E. Meinhardt

Plan



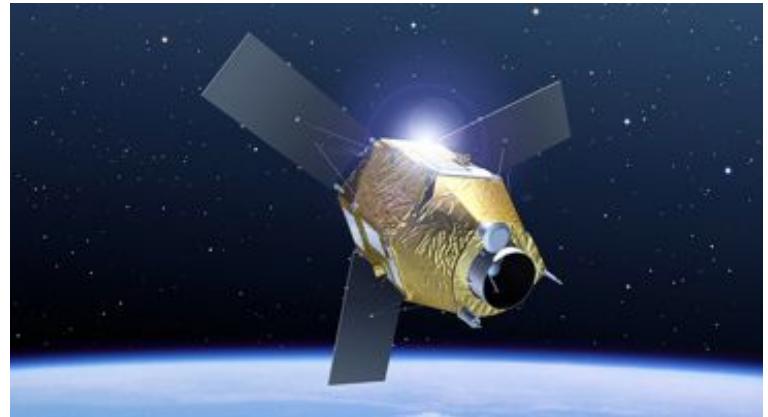
©1999 20TH CENTURY FOX FILM CORP.

Plan



©1999 20TH CENTURY FOX FILM CORP.

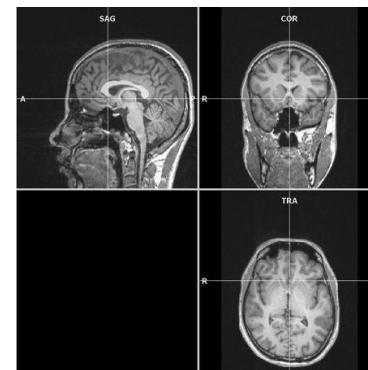
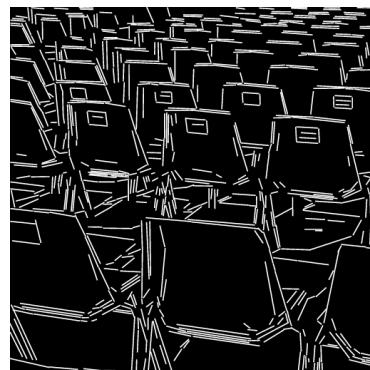
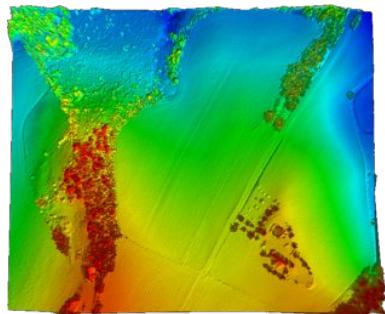
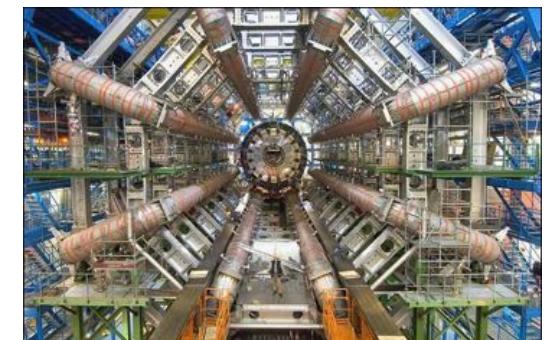
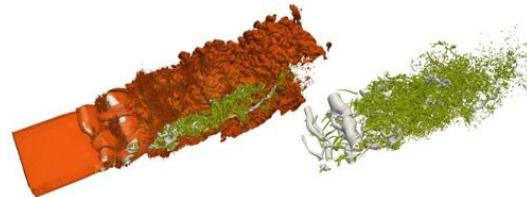
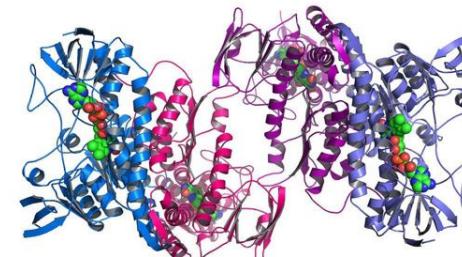
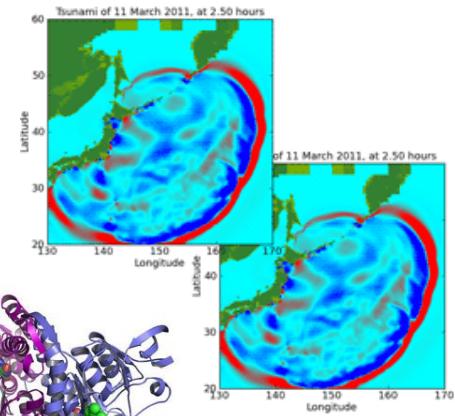
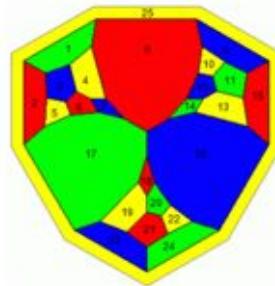
Plan



©1999 20TH CENTURY FOX FILM CORP.

Software Everywhere

- physics
- econometrics
- signal processing
- chemistry
- archeology
- medicine
- climate & weather
- geophysics
- ...



Software vs Science?

Research software is not like general-purpose equipment:
telescopes, supercomputers, compilers, libraries



Research software is made by scientists for scientists,
specialized for an experimental process.

Most of the research software is:

- not released
- not published, not reviewed, not cited
- not completely specified
- ... and often buggy

Reproducible Research

Research is reproducible if other researchers can independently obtain the same results from the published material.

- Theoretical sciences have proofs
- Experimental sciences have procedures
- Computational sciences have ...
 - insufficient descriptions
 - missing parameters
 - missing pre/post processing steps
 - missing data



refs: Claerbout 1992, Donoho 1995, Stodden, Vandewalle

Research must be Reproducible

Scientific method (1200 - 1800)



Roger Bacon, Francis Bacon,
Galileo Galilei, Robert Boyle,
René Descartes, ...

→ science **must be reproduced**

Reproducible research (1990 -)



Jon Claerbout, David Donoho,
Serguei Fomel, Randy
Leveque, Davis Bailey, Victoria
Stodden, Juliana Freire, ...

→ the science is in the **software, data and process**

The article is a mere teaser



David Donoho
Stanford

*An article about computational science in a scientific publication is not the scholarship itself, it is merely **advertising** of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.*

Traditional Articles (in image processing)

Description of a method/algorithm with some results. The **code** used for the experiments is **rarely made public**

We can:

- Read the formulas
- Believe the results

We can't:

- Verify/reproduce the results
- View the images in detail
- Try with new data



Why not to distribute the research code?

- Maintain advantage w.r.t. other researchers
- Ashamed of the code
(no time for cleanup & documentation)
- Prevent incorrect use
(by choosing wrong parameters)
- ???

Revisit Objectives of Publishing Articles



Picture of a rare utopian community
in the act of sharing their research code

Revisit Objectives of Publishing Articles



Picture of a rare utopian community
in the act of sharing their research code



Revisit Objectives of Publishing Articles



Picture of a rare utopian community
in the act of sharing their research code

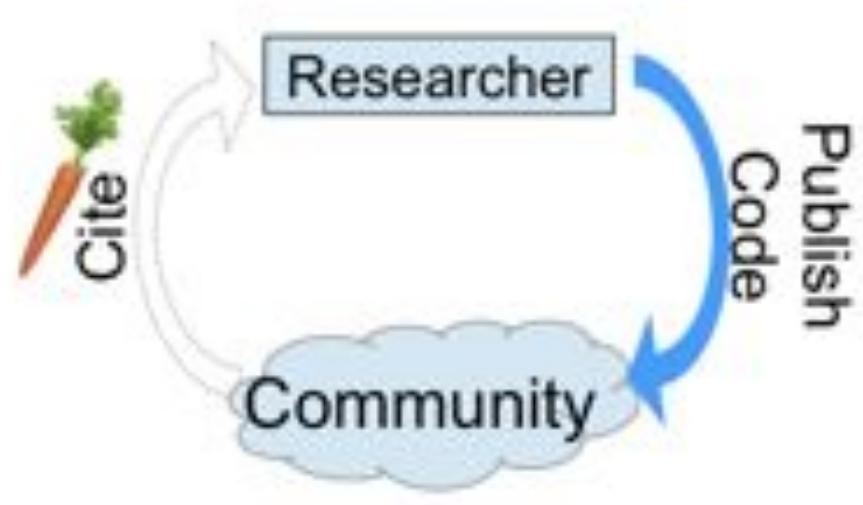


KEY to lure researchers into
sharing their code

Revisit Objectives of Publishing Articles



Traditional Research Articles



Source Code

Step 1: Make the code
a publication by itself

Revisit Objectives of Publishing Articles



Traditional Research Articles



Source Code

Step 2: Guide community
to cite implementations

IPOL : Image Processing On Line



<http://ipol.im/>

*“IPOL is a **research journal of image processing and image analysis**. Each article contains a text describing an **algorithm** and **source code**, with an **online demonstration** facility and an **archive** of online experiments. The text and source code are peer-reviewed and the demonstration is controlled. IPOL follows the **Open Access** and **Reproducible Research** models.”*

For every article, the **implementation** is:

- **Reviewed** and Published
- License GPL or BSD
- Following Software Guidelines for correctness, portability, documentation

IPOL : Image Processing On Line

GOAL:
**provide a reference implementations
of image processing algorithms**

- IPOL is a journal (running since 2011):
ISSN; DOI; Int'l editorial committee; ...
- Partnership with a SIAM
- IPOL exists because no other journal did it
- IPOL publishes algorithms, not software. Implementations
are here to provide the full details and to run the algorithm

IPOL Article Components

1. Algorithm **description (PDF)**
2. Implementation source **code**
3. **Web demo interface**
4. Public **archive** with original test data

The screenshot shows the IPOL Journal - Image Processing On Line website. The main title is "LSD: a Line Segment Detector" by Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Below the title, there are links for "article", "demo", and "archive". A green bar at the bottom contains publication details: "published: 2012-03-24", "reference: Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall, LSD: a Line Segment Detector, Image Processing On Line, vol. 2012, http://dx.doi.org/10.5201/ipol.2012.gjmr-lsd". The abstract discusses a linear-time Line Segment Detector. Below the abstract, there are download links for "full text manuscript" (PDF 554K) and "source code" (ZIP). A preview section is shown below, with a note that loading takes a few seconds. The footer includes the IPOL logo and copyright information.

The screenshot shows the same article page as above, but with specific components highlighted: "includes some further improvement over", "Download", "full text manuscript: PDF (554K)", "source code: ZIP", and "Preview".

IPOL Article Components

1. Algorithm **description (PDF)**
2. Implementation source **code**
3. **Web demo** interface
4. Public **archive** with original test data

LSD: a Line Segment Detector

[article](#) [demo](#) [archive](#)

Please cite the reference article if you publish results obtained with this online demo.

Select Data

Click on an image to use it as the algorithm input.

Chairs image Le Pree LSD molecule Noise

Upload Data

Upload your own image files to use as the algorithm input.

No file chosen

Images larger than 100000000 pixels will be resized. Upload size is limited to 2MByte per image file and 10MByte for the whole uploaded set. TIFF, JPEG, PNG, GIF, PNM (and other standard formats) are supported. The uploaded files may be re-used for further analysis. Only upload suitable images. See the copyright and legal conditions for details.

[feeds](#) • [sitemap](#) • [contact](#) • [privacy policy](#) • ISSN: 2105-1232 • DOI: 10.5201/Ipol
supported by CMLA, ENS Cachan • DMI, Universitat de les Illes Balears • Fine, Universidad de la República • Gandi.net
© 2009-2012, IPOL Image Processing On Line & the authors

LSD: a Line Segment Detector

[article](#) [demo](#) [archive](#)

Please cite the reference article if you publish results obtained with this online demo.

Run again? new image different subimage

Result

847 Line Segments were detected. The algorithm ran in 0.27s.

You can download the result in [EPS](#) format, in [SVG](#) format, or an [ASCII](#) file (see description below).



Test with new data and explore parameters without compiling

IPOL Article Components

1. Algorithm **description** (PDF)
2. Implementation source **code**
3. **Web demo** interface
4. Public **archive** with original test data

LSD: a Line Segment Detector

article demo archive

Please cite the reference article if you publish results obtained with this online demo.

6954 public archives of online experiments with original images since 2009/04/27 08:11.
This archive is not moderated. In case of copyright infringement or similar problem, please contact us to request the removal of some images. Some archived content may be deleted by the editorial board for size matters, inadequate content, user requests, or other reasons.

pages: << < > >> - 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104
105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142
143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218
219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256
257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294
295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332
333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370
371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408
409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446
447 [448]

key 35G93AB22D7BF8CAB0314C29E286F55B
date 2013/04/02 09:54
LSD Version 1.6 of November 11, 2011, compiled
version Dec 4 2012 15:49:05
run time 0.151010036469
(s)
files output.txt output.eps output.svg

Images

key 6ED5D2C294A071F7CEB88D7663847DEC
date 2013/04/02 11:07
LSD Version 1.6 of November 11, 2011, compiled
version Dec 4 2012 15:49:05
run time 0.235600948334
(s)
files output.txt output.eps output.svg

Images

lsd Version 1.6 of November 11, 2011, compiled
version Dec 4 2012 15:49:05
run time (s) 0.104063034058
files output.txt output.eps output.svg

run time (s) 0.534783124824
files output.txt output.eps output.svg

Importance of the online demo



CODE

Importance of the online demo



CODE



CODE + DEMO



- on-line demo facilitates experimentation
- and leads to a stricter verification of the claims

Importance of the online demo



CODE



CODE+DEMO+ARCHIVE



- on-line demo facilitates experimentation
- and leads to a stricter verification of the claims

IPOL Usage Stats

- 45 articles published with code and demo since 2011
- 20 articles in preparation
- 109 citations (cf. Google Scholar)
- 2012: 125k visits, 13k code/data downloads
- 2012: 50k demo runs, 30k archived runs on original data



Challenges

- Effort to prepare the code for publication isn't negligible
- Community must learn to cite the implementations



Follow-up to...

IPOL wouldn't be possible without the support and trust of the authors, reviewers and editors who contributed to it. Lots of help from Paris, Palma, Barcelona, Montevideo, Durham, ...

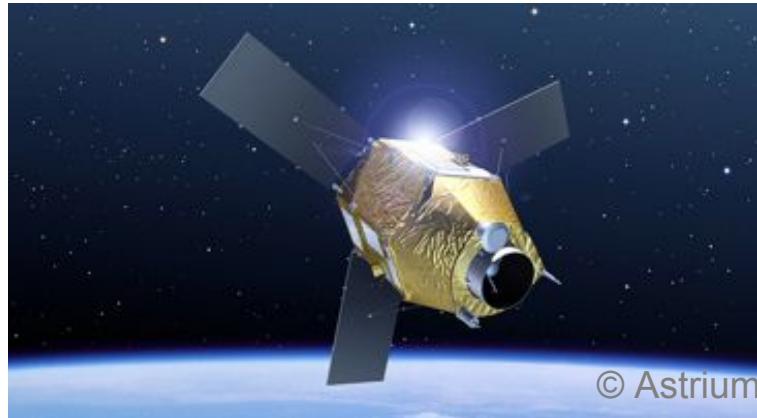


<http://ipol.im/>
edit@ipol.im
discuss@list.ipol.im
@ipol_journal





Satellite stereo images



© Astrium



S2P: satellite stereo pipeline

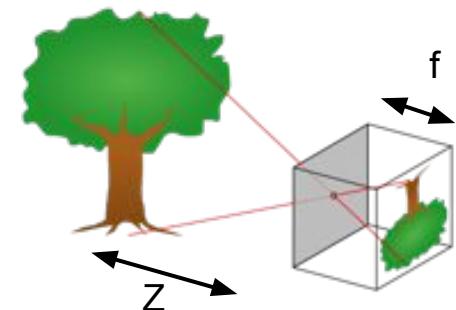
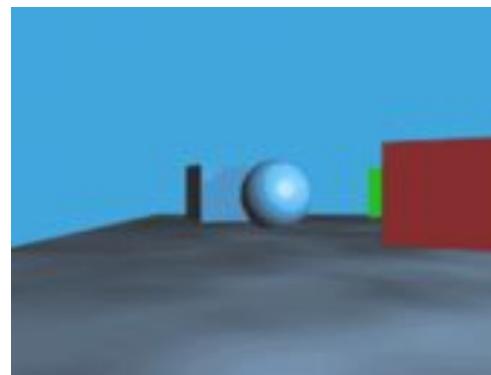
C. De Franchis, E. Meinhardt, J-M Morel, J. Michel, GF

Some slides courtesy of C. De Franchis

Stereoscopy in one slide

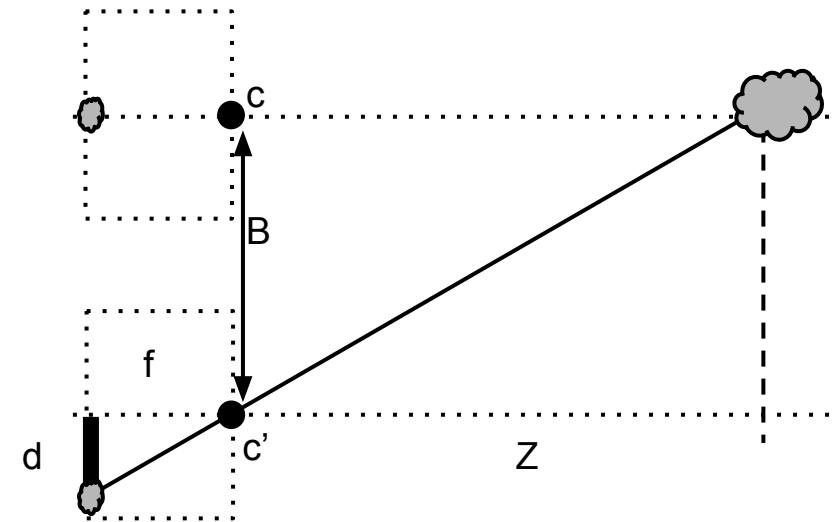
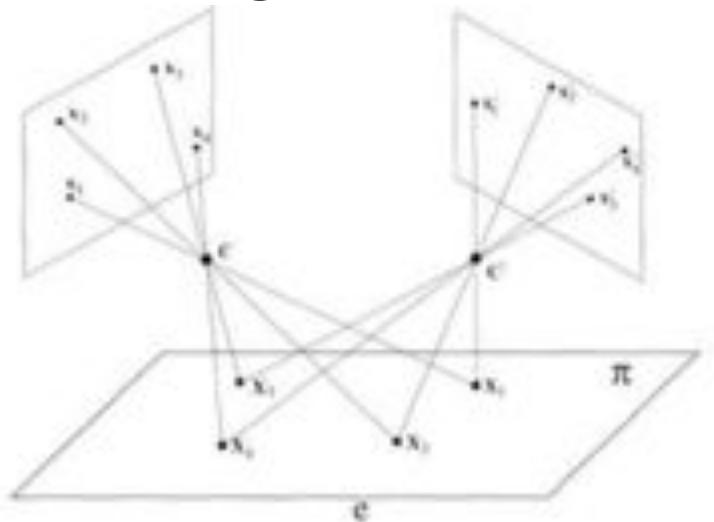
Depth Z is inversely proportional to the parallax movement d

$$Z = B \cdot f / d$$



Projective camera

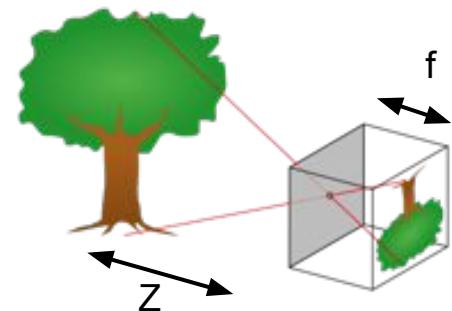
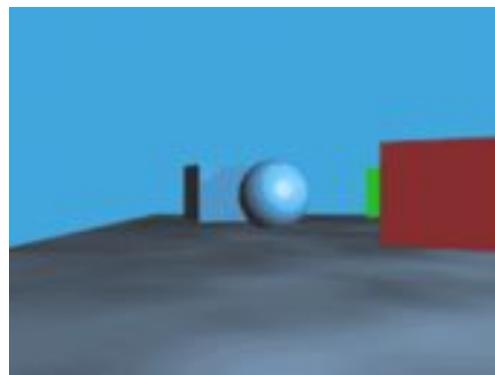
Two images



Stereoscopy in one slide

Depth Z is inversely proportional to the parallax movement d

$$Z = B \cdot f / d$$



Projective camera

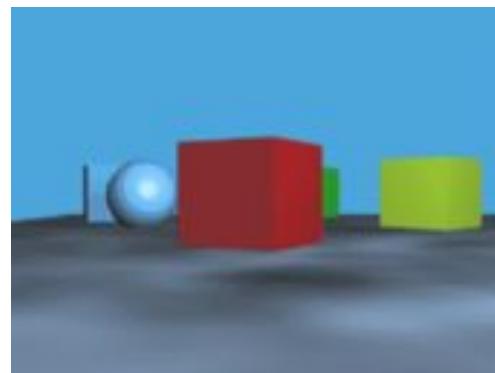
Two images



© Jan Cech

Stereoscopy in one slide

Depth Z is inversely proportional to the parallax movement d

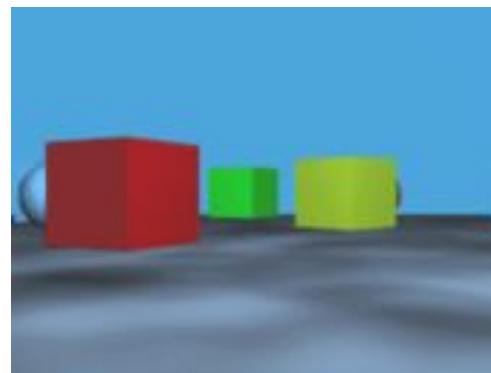


Estimated disparity (parallax movement)

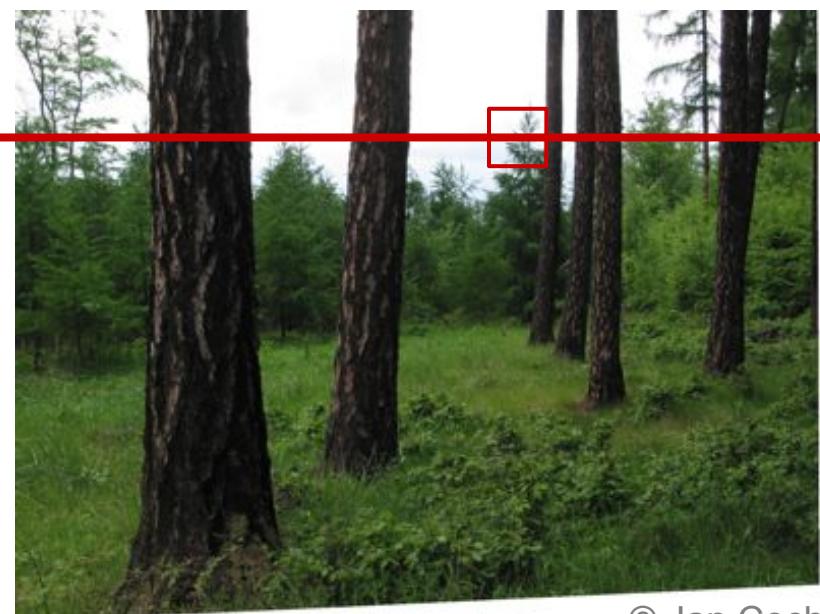
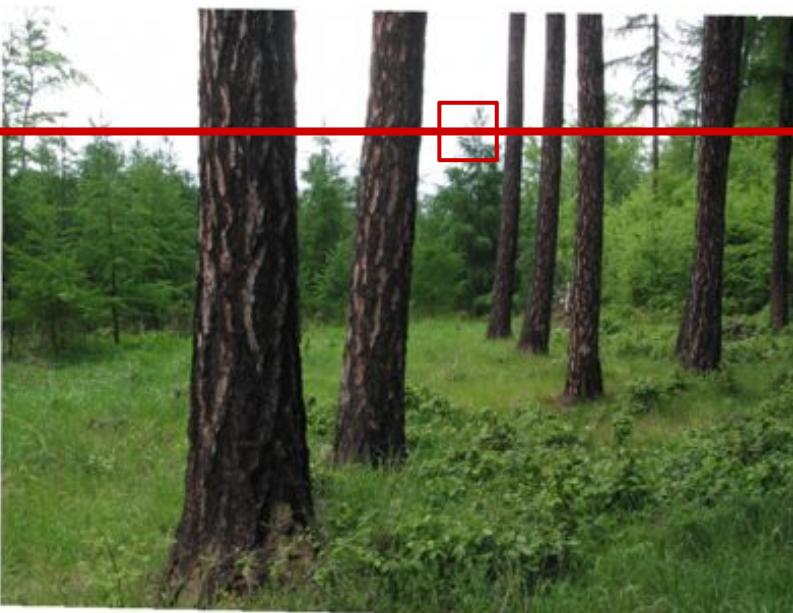


Stereoscopy in one slide

Depth Z is inversely proportional to the parallax movement d

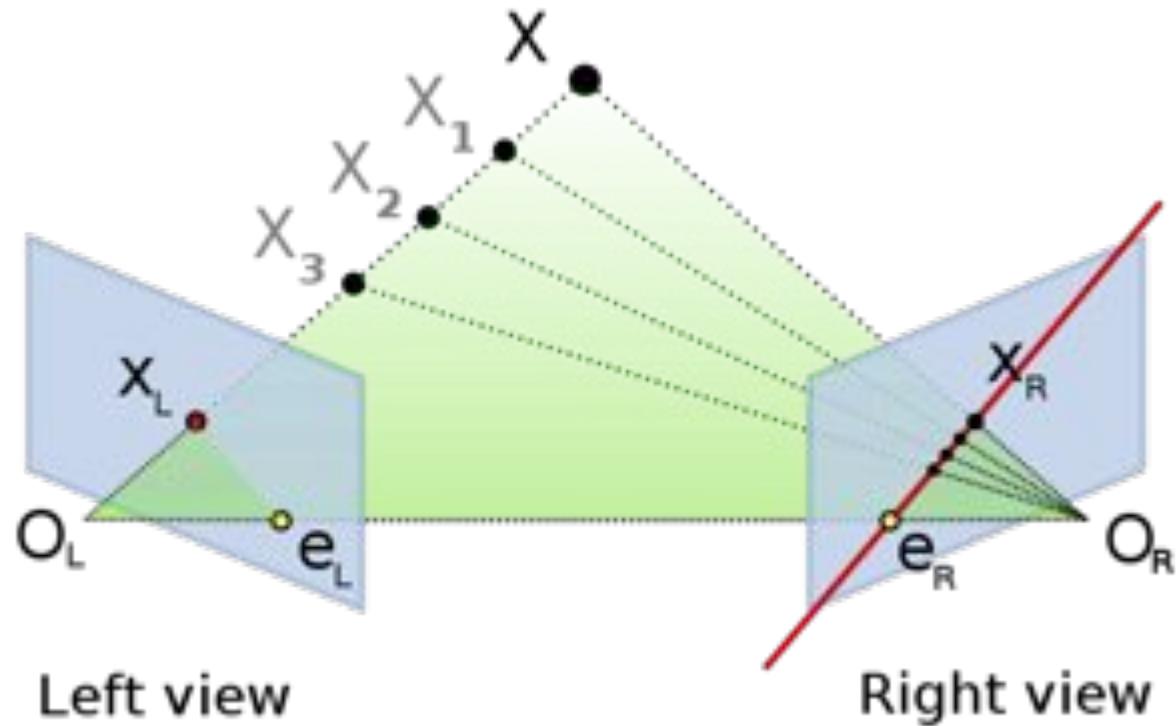
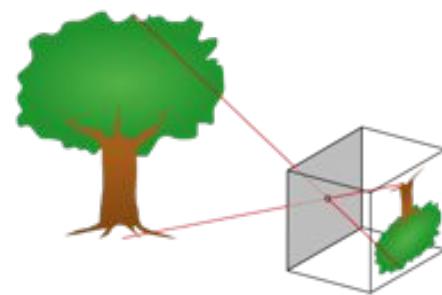


The movement is always “horizontal” for rectified images



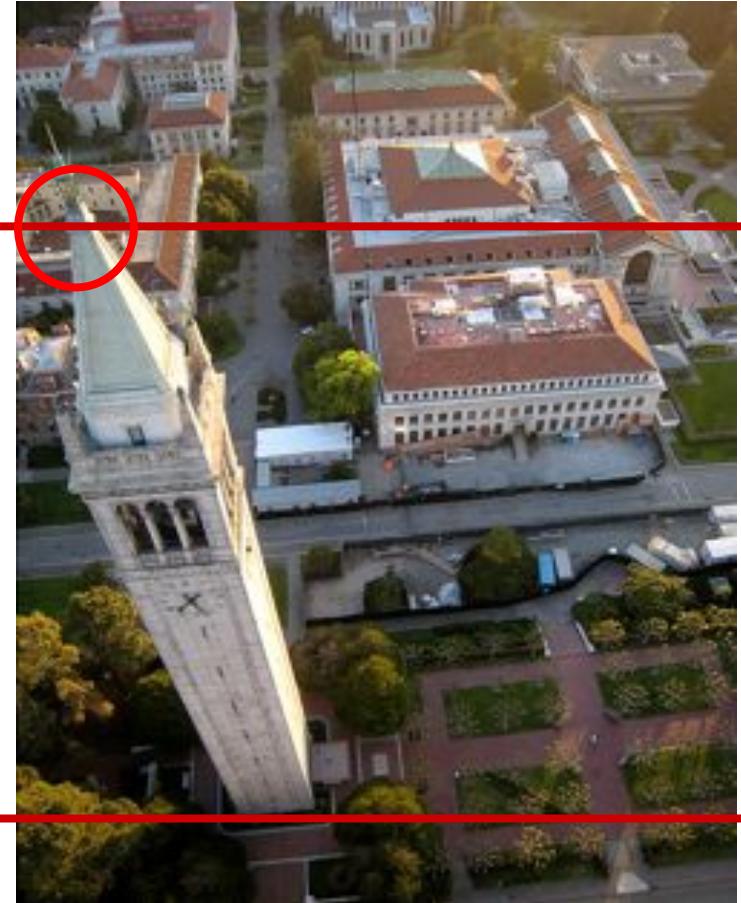
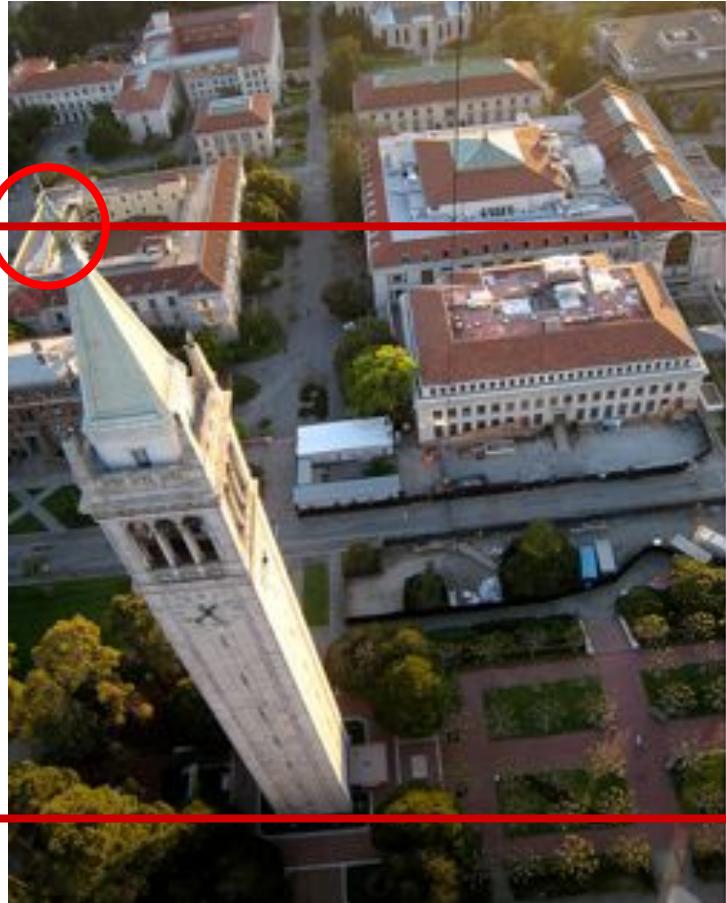
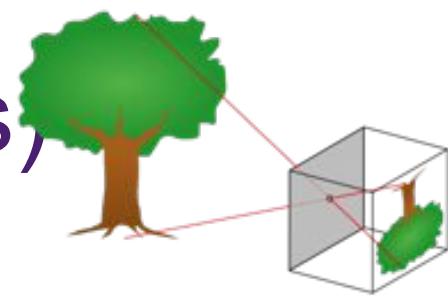
© Jan Cech

Epipolar lines



The correspondent in the right view of a point X_L lies on the line that passes through e_R

Rectification (projective cameras)



These images can be rectified

Epipolar lines



These images can be rectified

Rectification (projective cameras)



Two projective images now rectified

Book: Multiple View Geometry, Hartley Zisserman

Stereoscopy summary



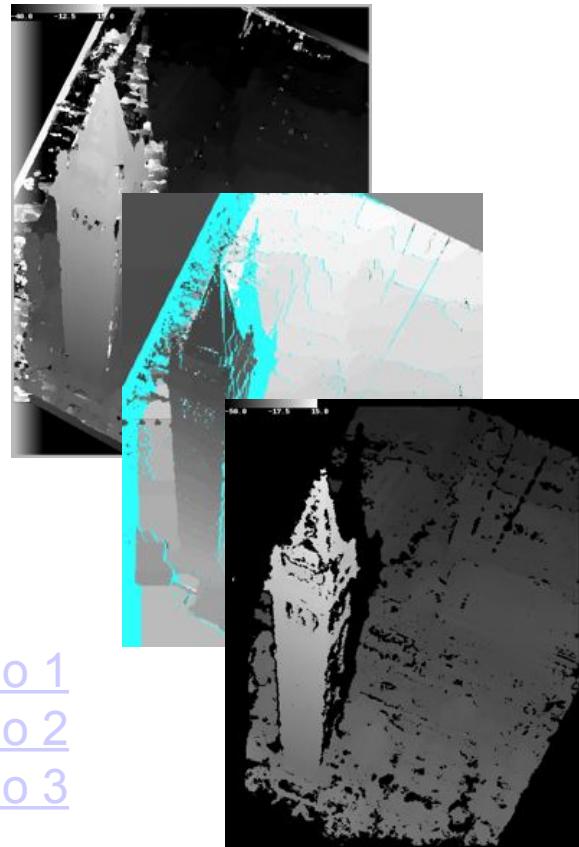
Rectification

[IPOL demo](#)



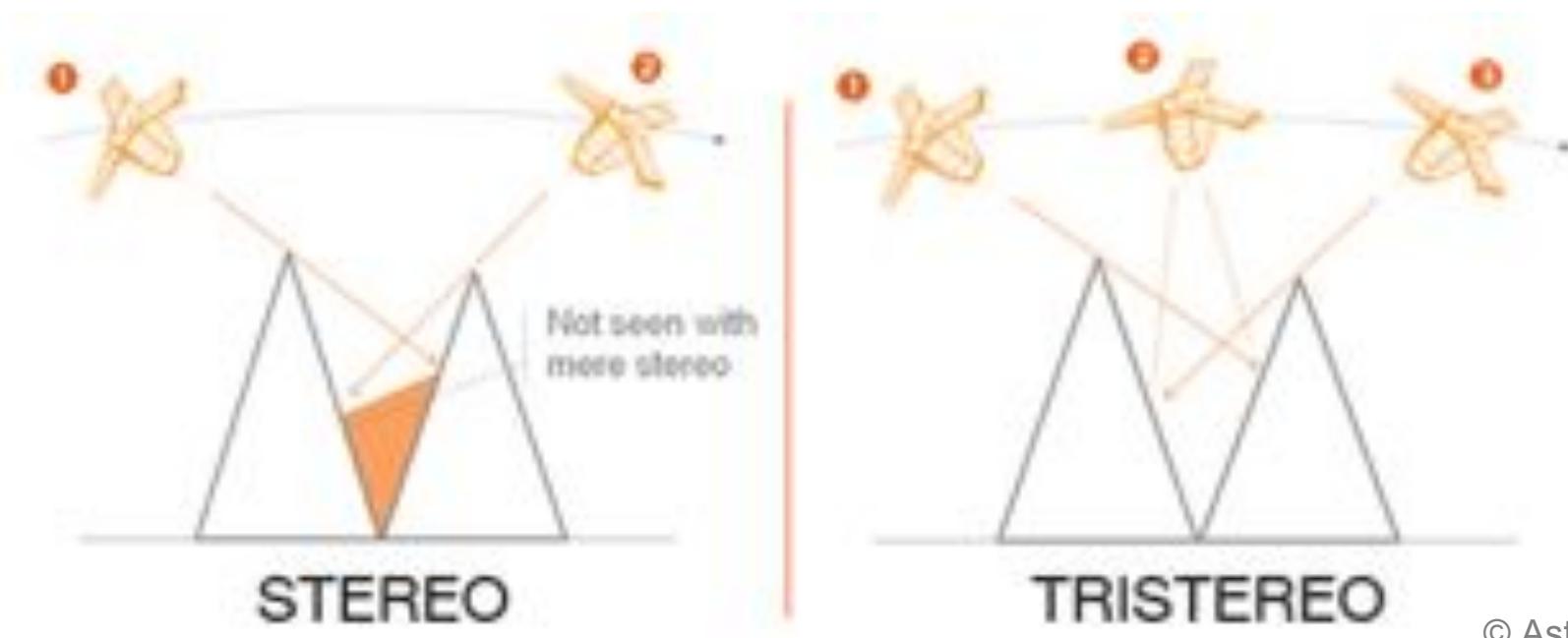
Match

[IPOL demo 1](#)
[IPOL demo 2](#)
[IPOL demo 3](#)



Pleiades satellite

- Altitude: 700km
- Sensor: **pushbroom** 70cm GSD
- Images: panchromatic **40k x 40k**, RGBI 10k x 10k
- Complete **calibration data** (co-localization functions)
- **Agility**: stereo, tristereo and more



© Astrium

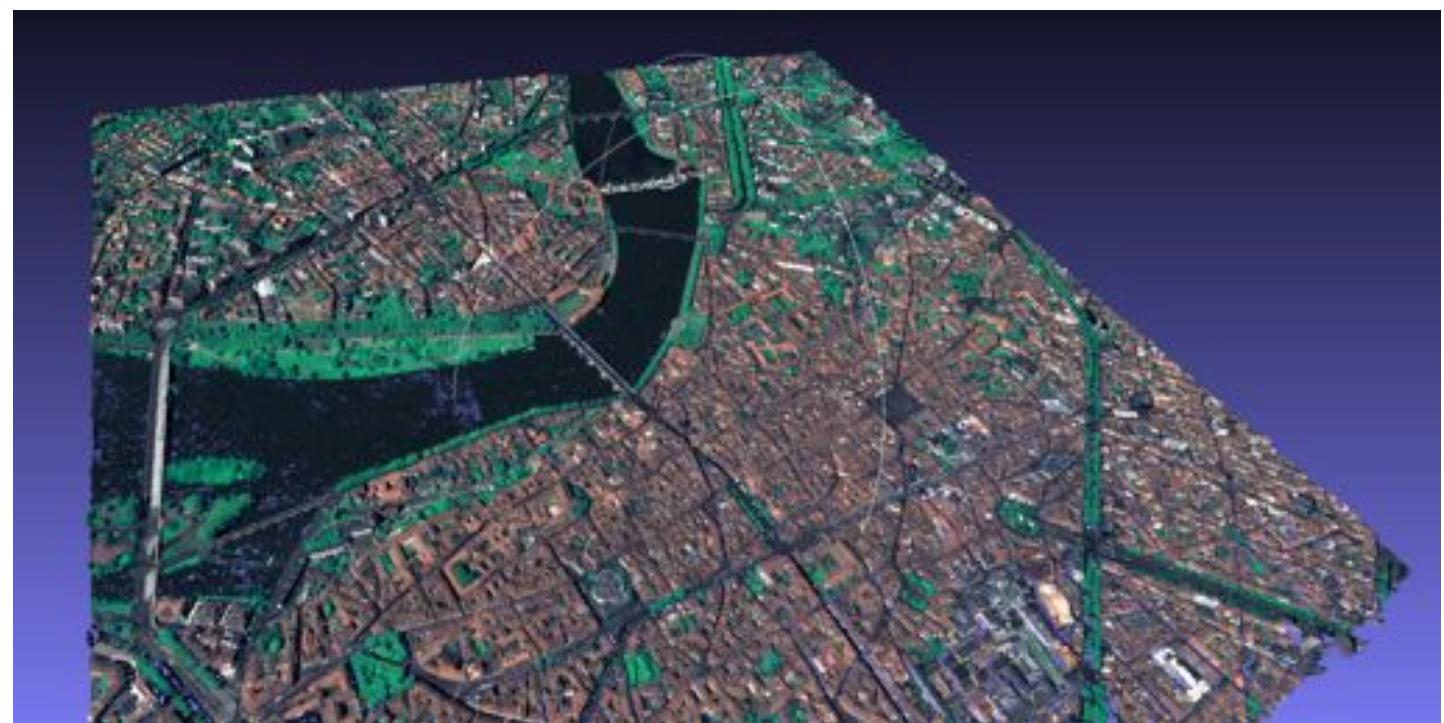
Agility:

Wait!
Intensity
isn't constant!

Reason enough
to avoid moving
too much

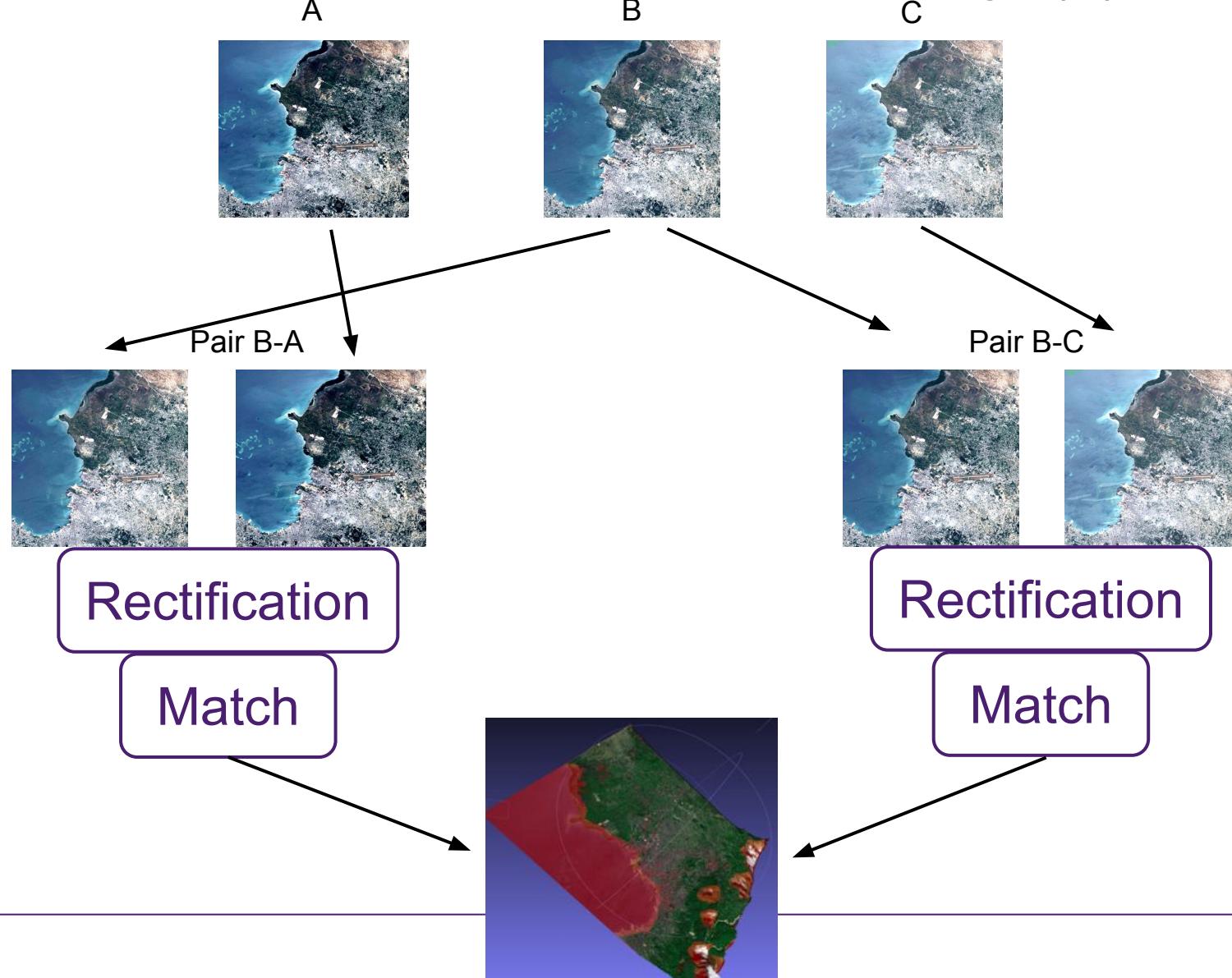


S2P: Tristereo chain

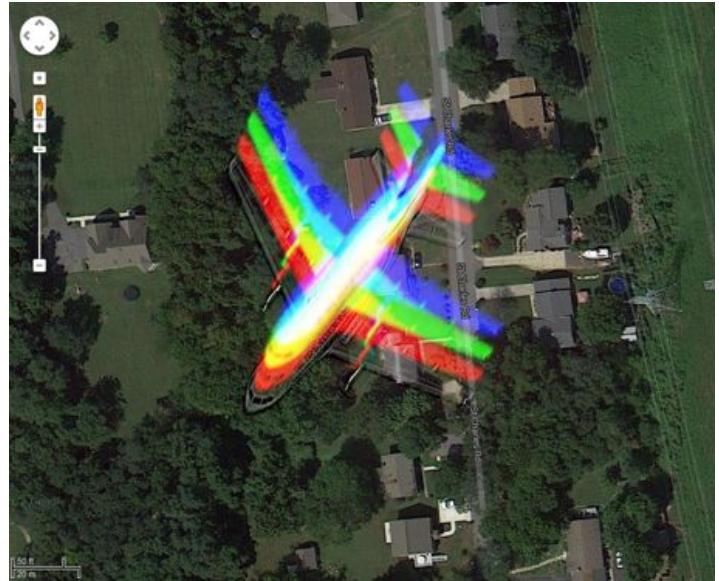
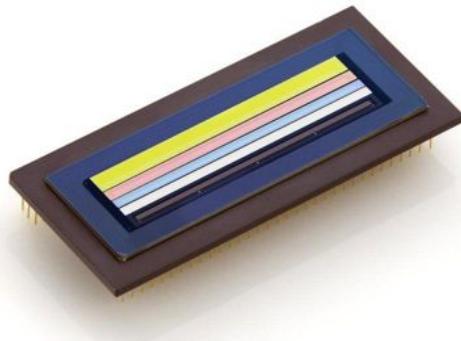
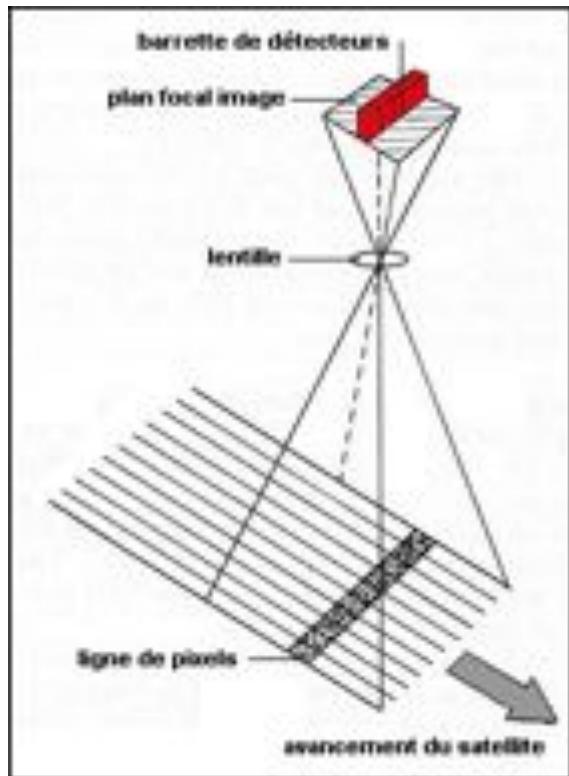


S2P: Tristereo chain overview

Extract two pairs with the same reference image (B)



Pushbroom isn't Pinhole



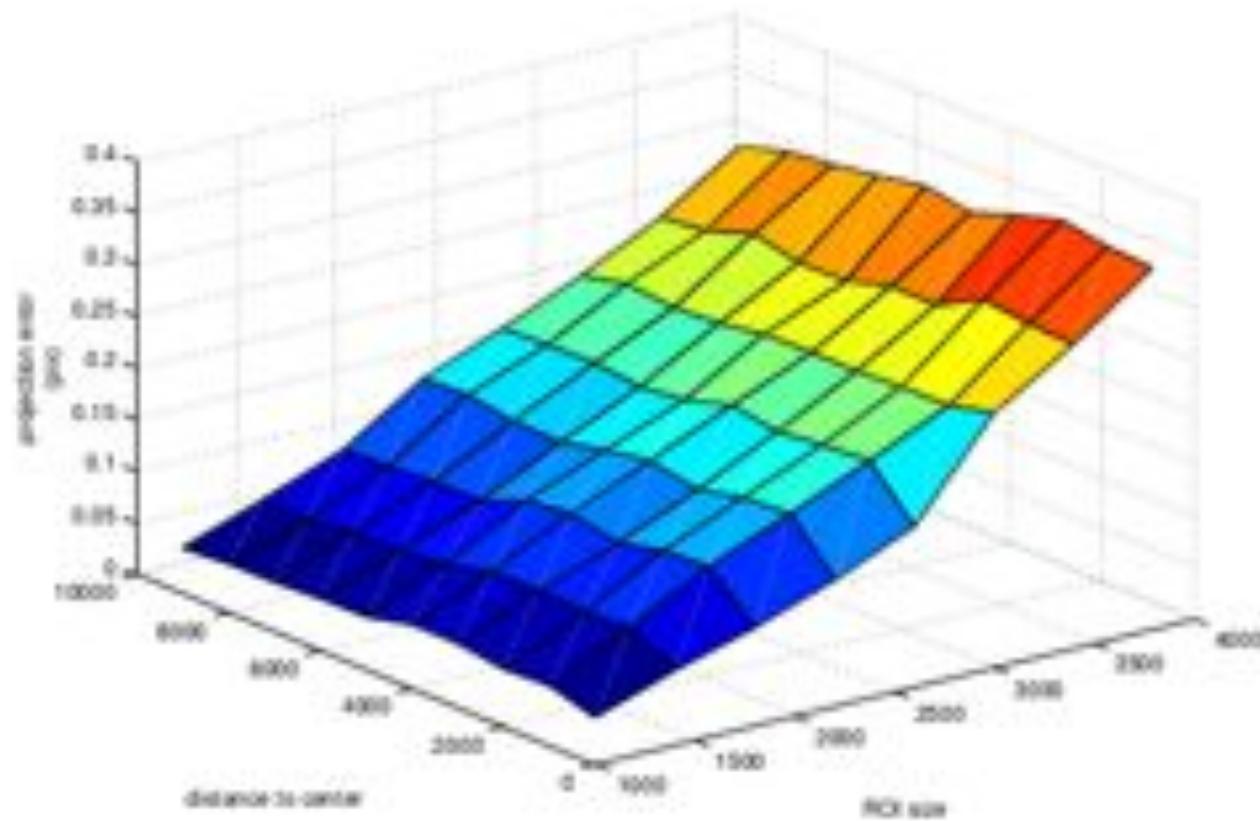
[The mystery of rainbow airplanes](#)

No epipolar rectification for pushbroom

“Linear Pushbroom Cameras”, Hartley and Gupta ‘97

Pushbroom isn't Pinhole

However for small subimages the geometry is almost pinhole
We use tiles of ~ 1000x1000



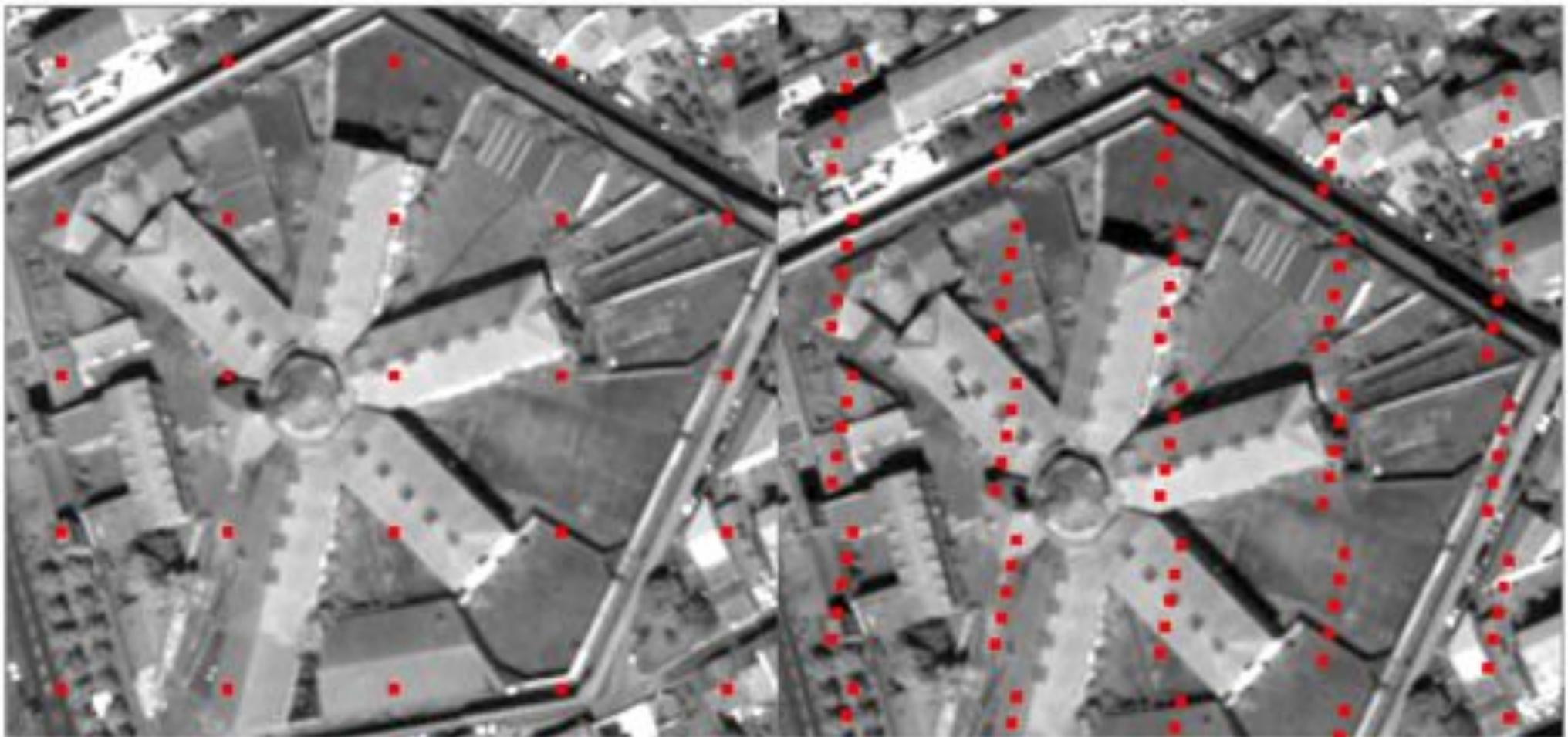
Degenerate rectification cases

Rectification using keypoint matches may fail because overall the scene is flat



Use calibration data to rectify

Rectification using virtual matches extracted from the provided calibration data (RPC co-localization functions)



Use calibration data to rectify

Rectify using virtual matches extracted from the
provided calibration data (RPC colocalization functions)



Use calibration data to rectify

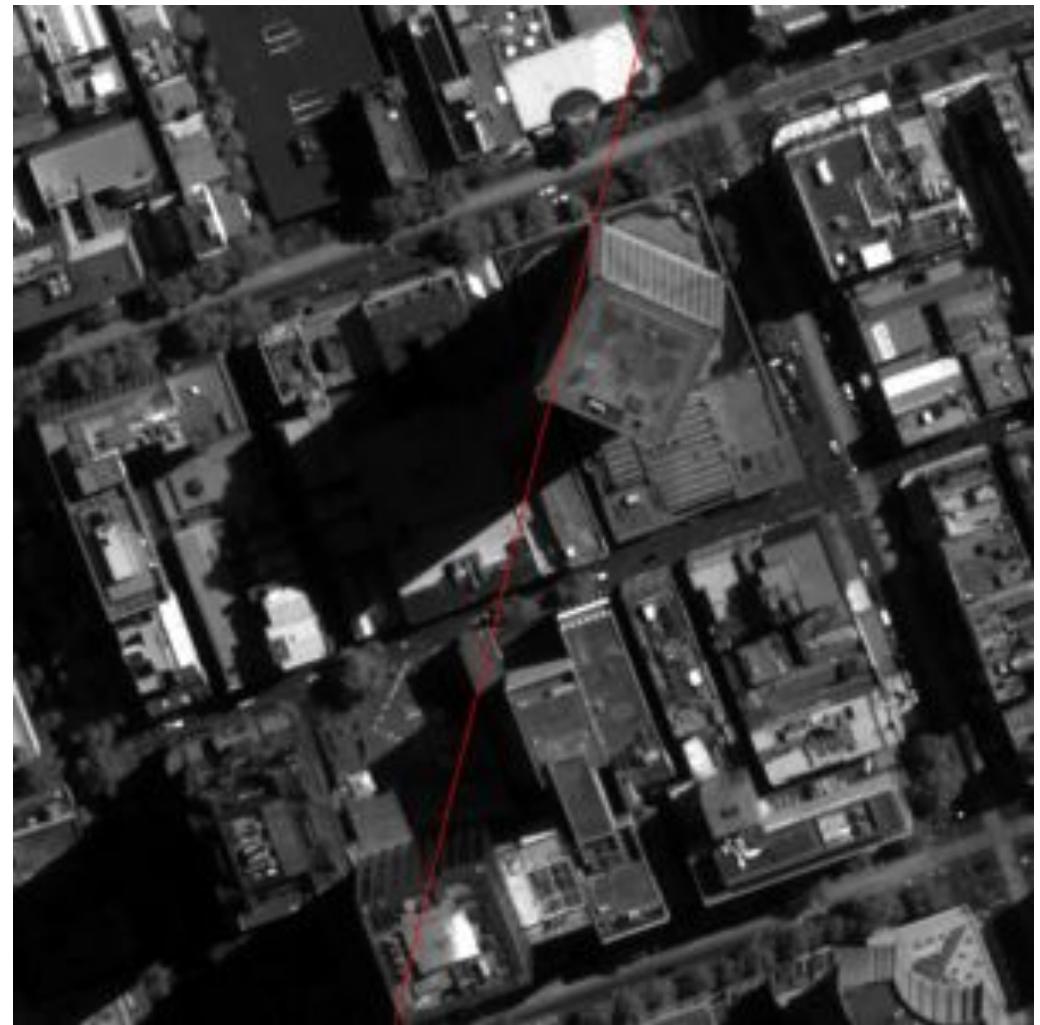
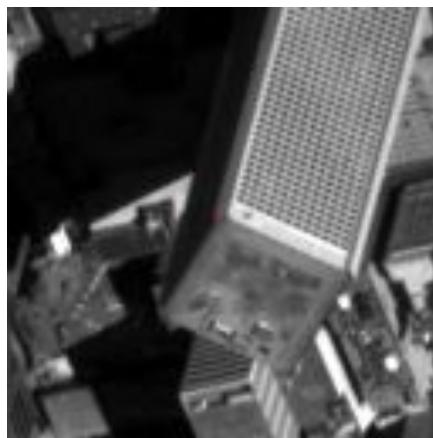
Rectify using virtual matches extracted from the
provided calibration data (RPC colocalization functions)



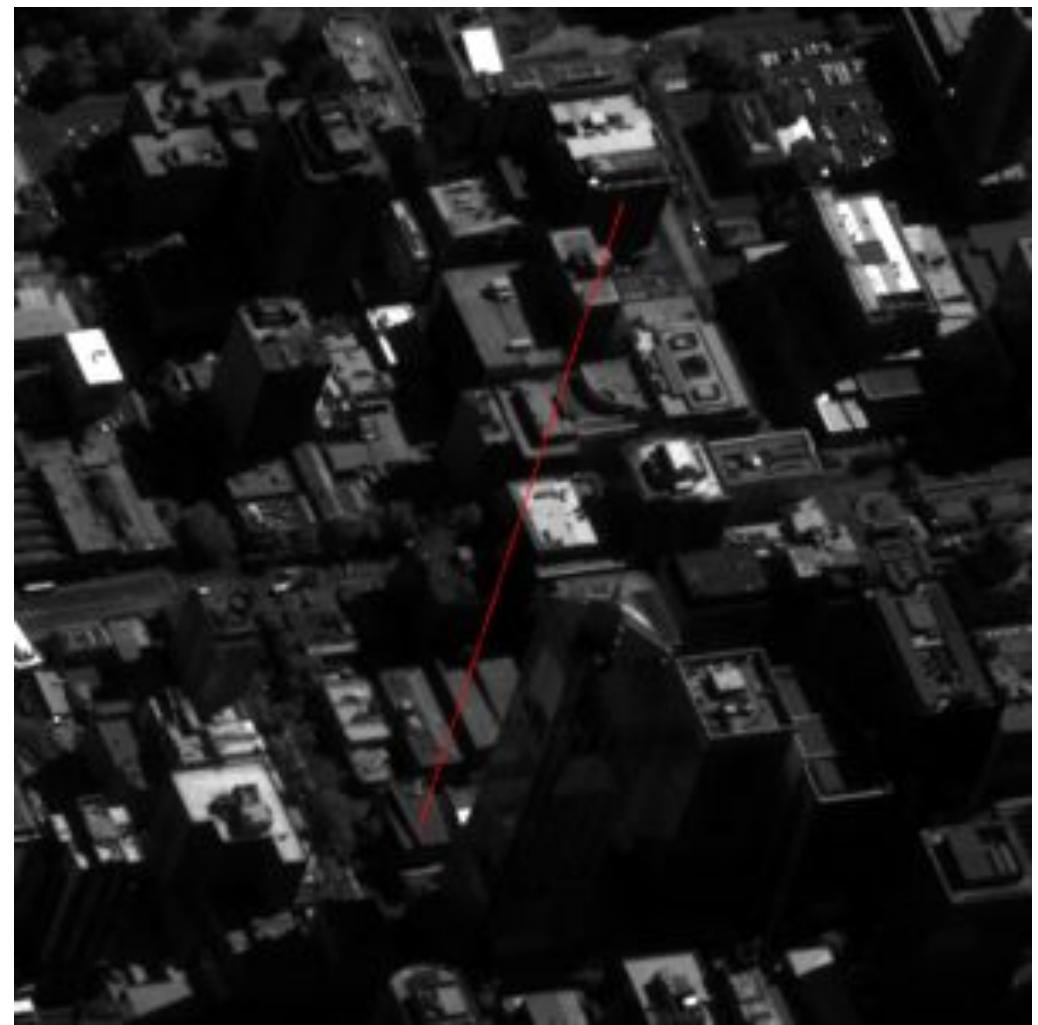
...

It's moving
vertically,
isn't it?

Errors in the colocalization functions



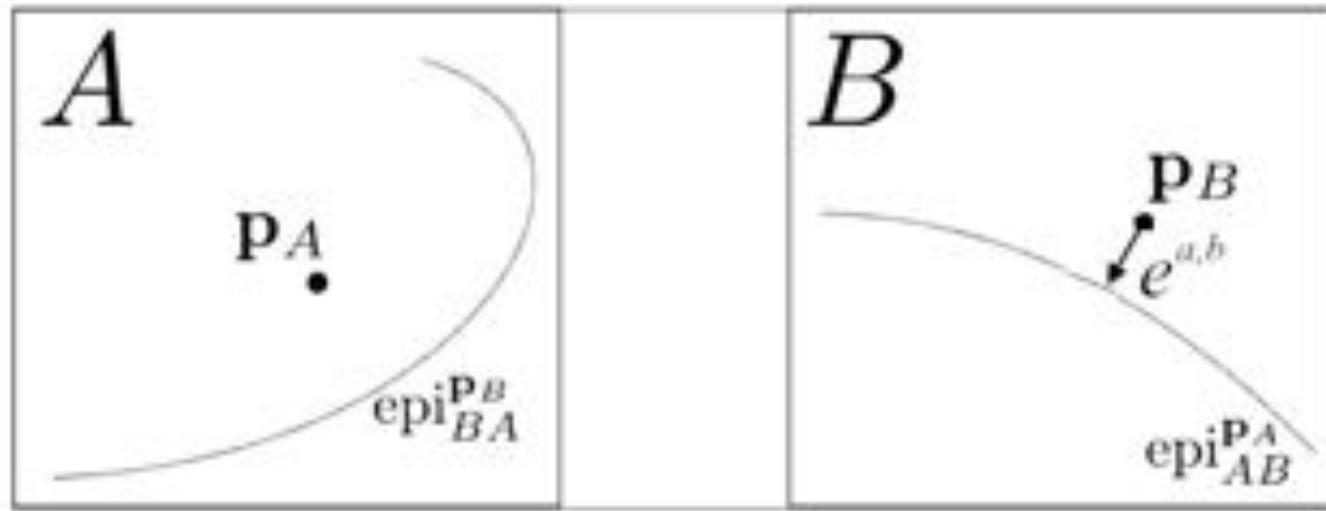
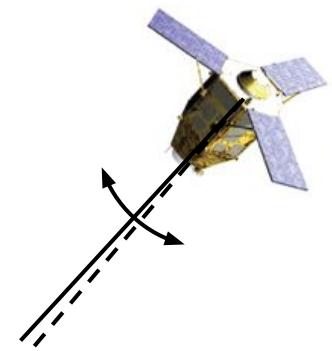
Errors in the colocalization functions



Pointing error

Imprecise orientation of the satellite

Note that 1×10^{-6} rad = 1 pixel



Quantify:

- Use keypoints pairs p_A, p_B extracted from the images
- Compute the epipolars using the calibration function
- The error e is the distance to the predicted epipolars

Pointing error correction

- Correct one of the images by an affinity A_b
- Optimize A_b to minimize

$$f(A_b) = \sum_{k=1}^m e_k^{a,b} + 0.01 |h_k^{a,b} - h_k^{a,b_0}|^2$$

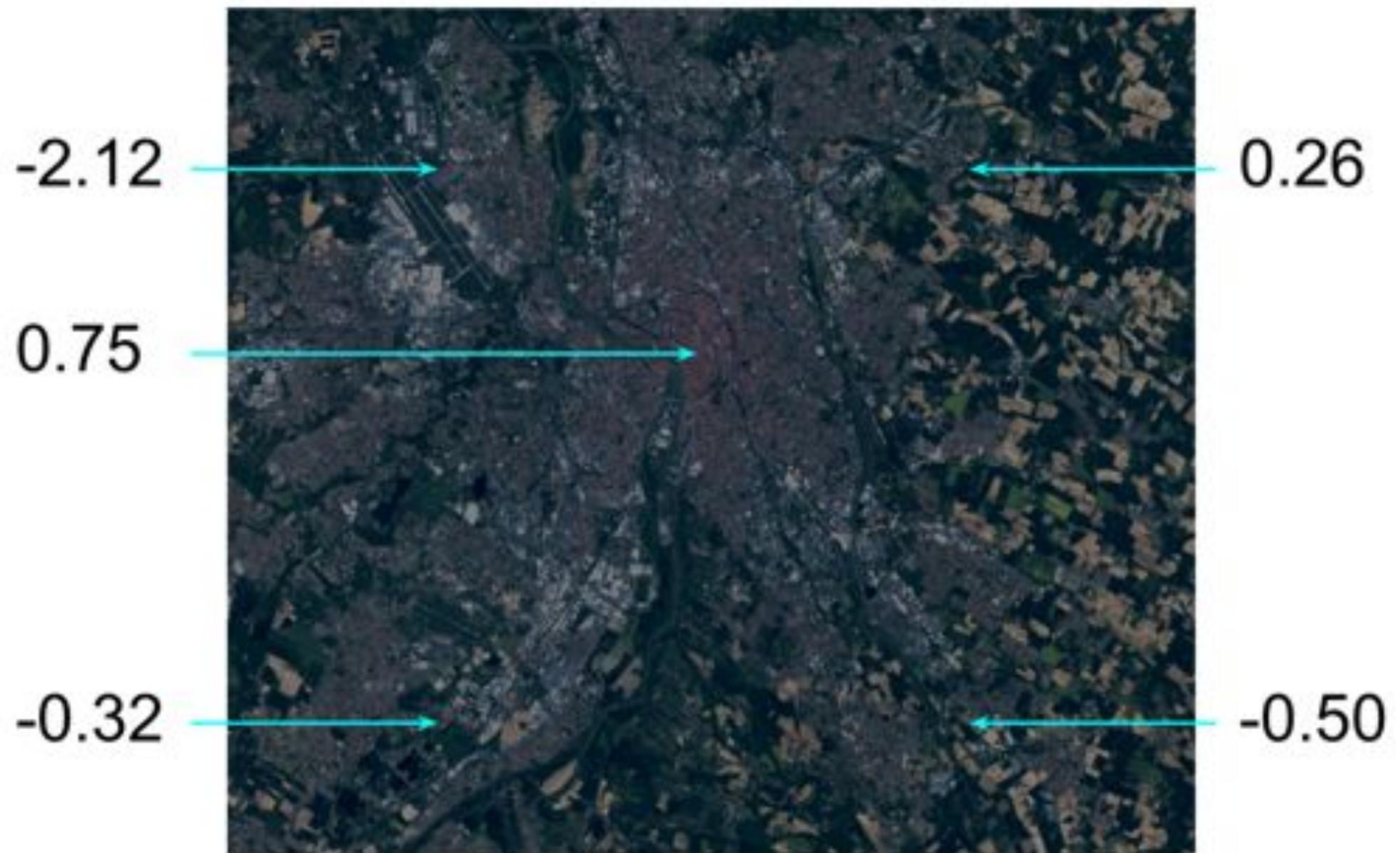
Rectification after pointing correction



Rectification after pointing correction



Pointing error across the image



Online demo

IPOL Demo

Results:

- [Cliffs](#)
- [Houses](#)
- [Airbus](#)
- [Stadium](#)

