The City Scanning Project :

Validation and parallel algorithms

Olivier Koch,
Visitor from March to August 2002

Advisor : Seth Teller

Contents

1/ Project overview

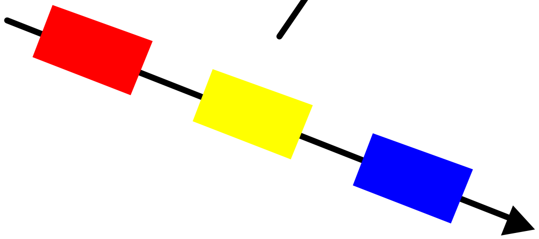
2/ code validation

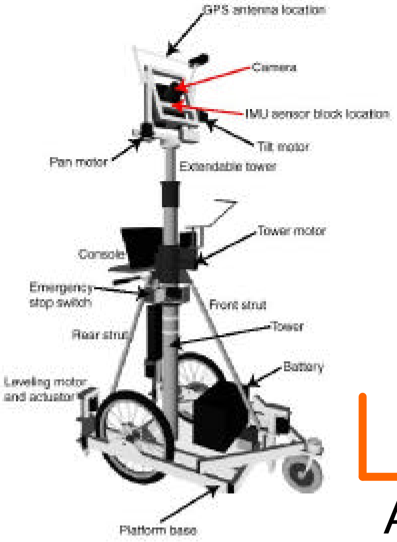
3/ parallel algorithms

4/ a few ideas

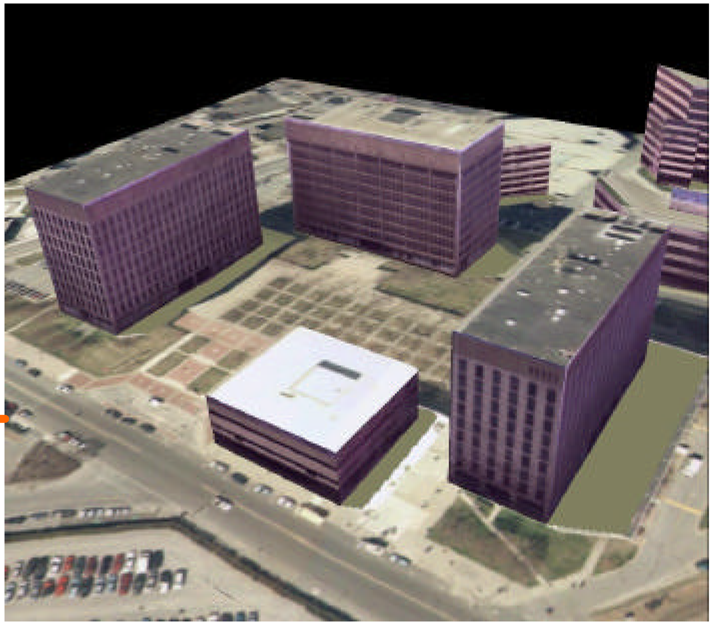
Project overview

Reconstruct a realistic 3D model of an urban environment from calibrated real-world images





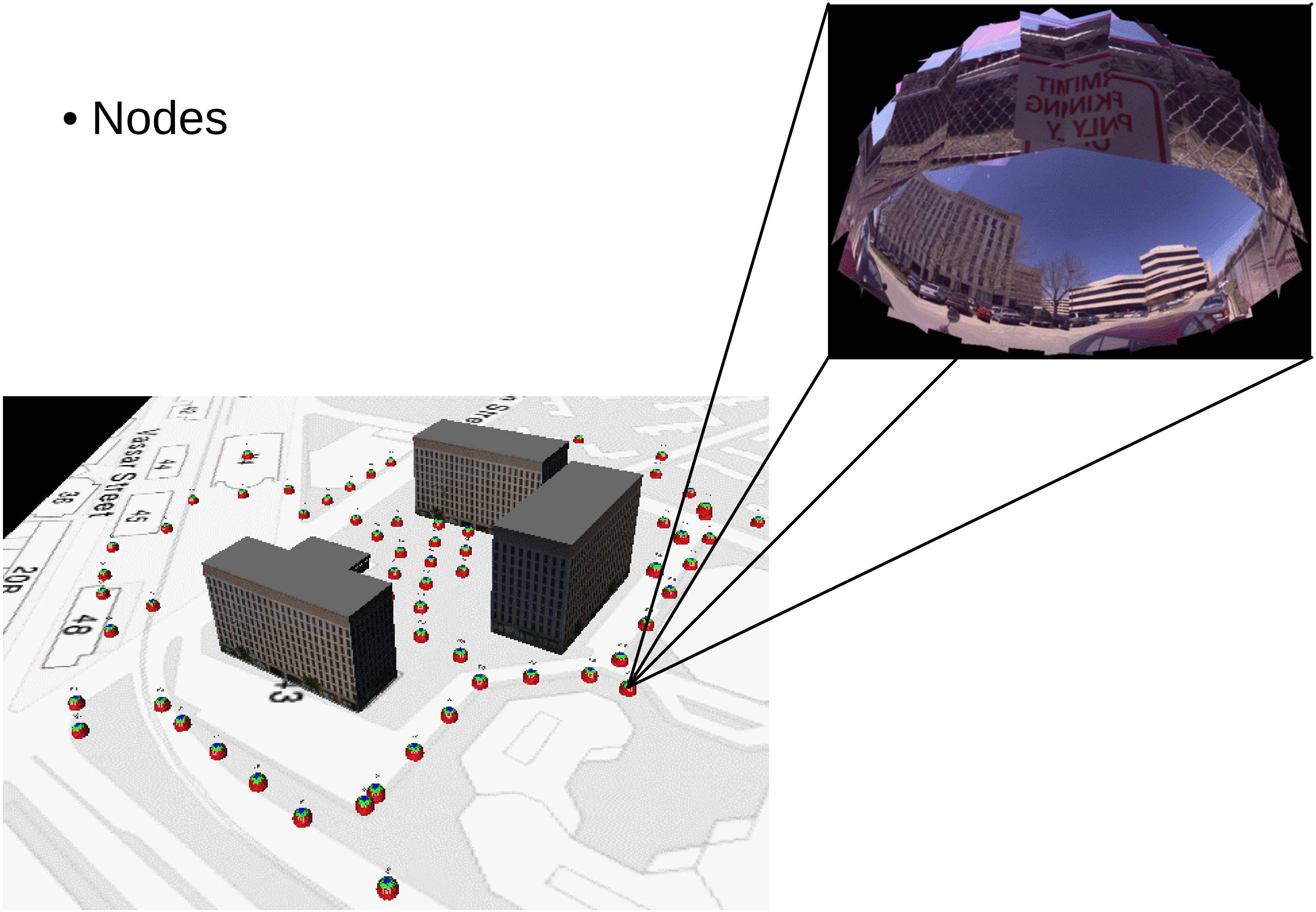
Argus



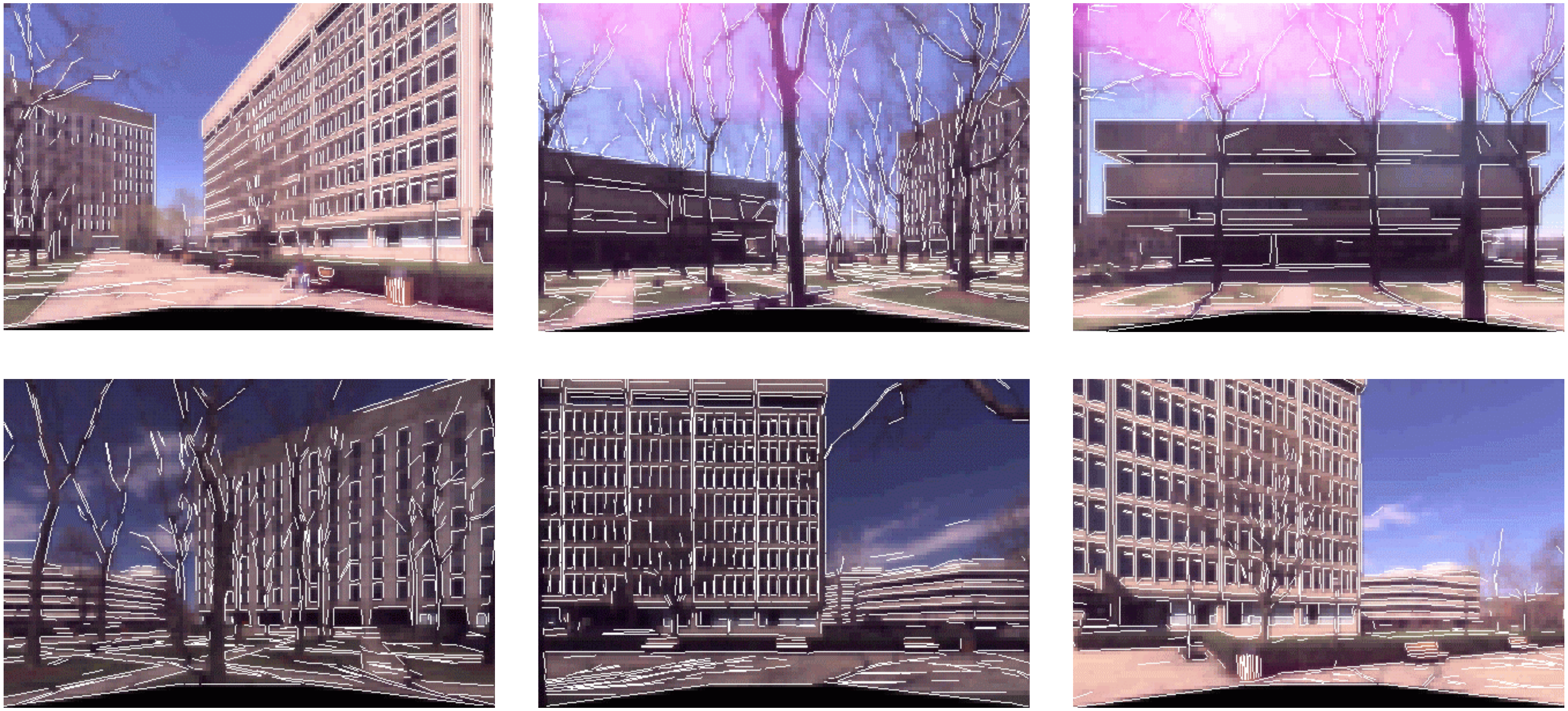
Technology Square

Project overview

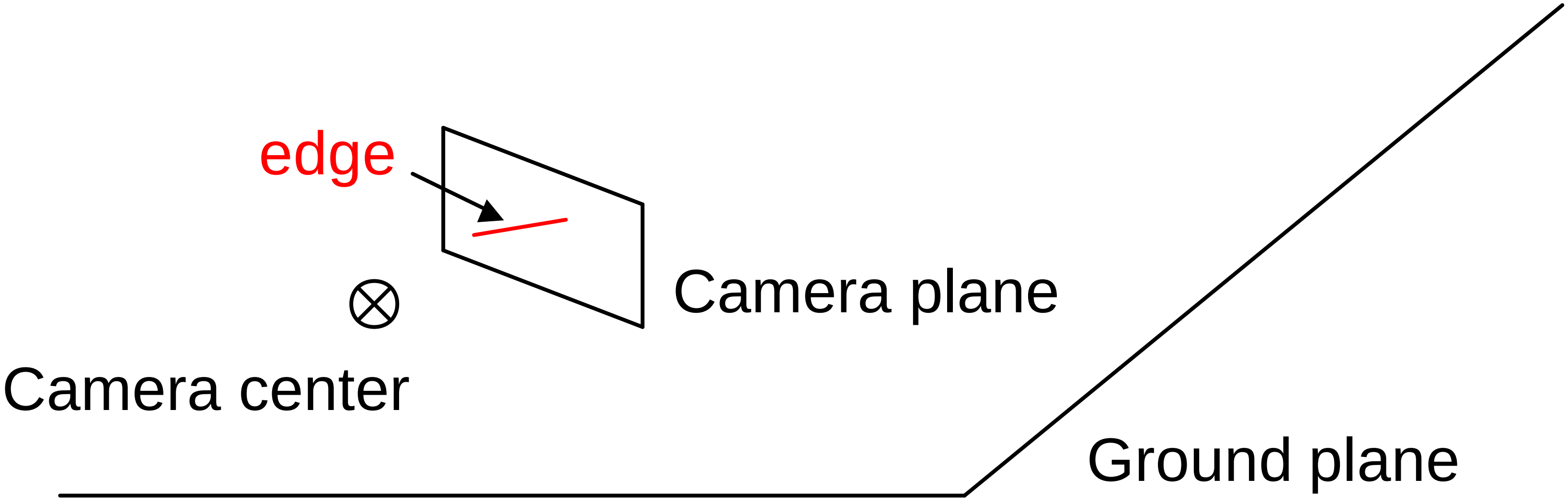
- Nodes



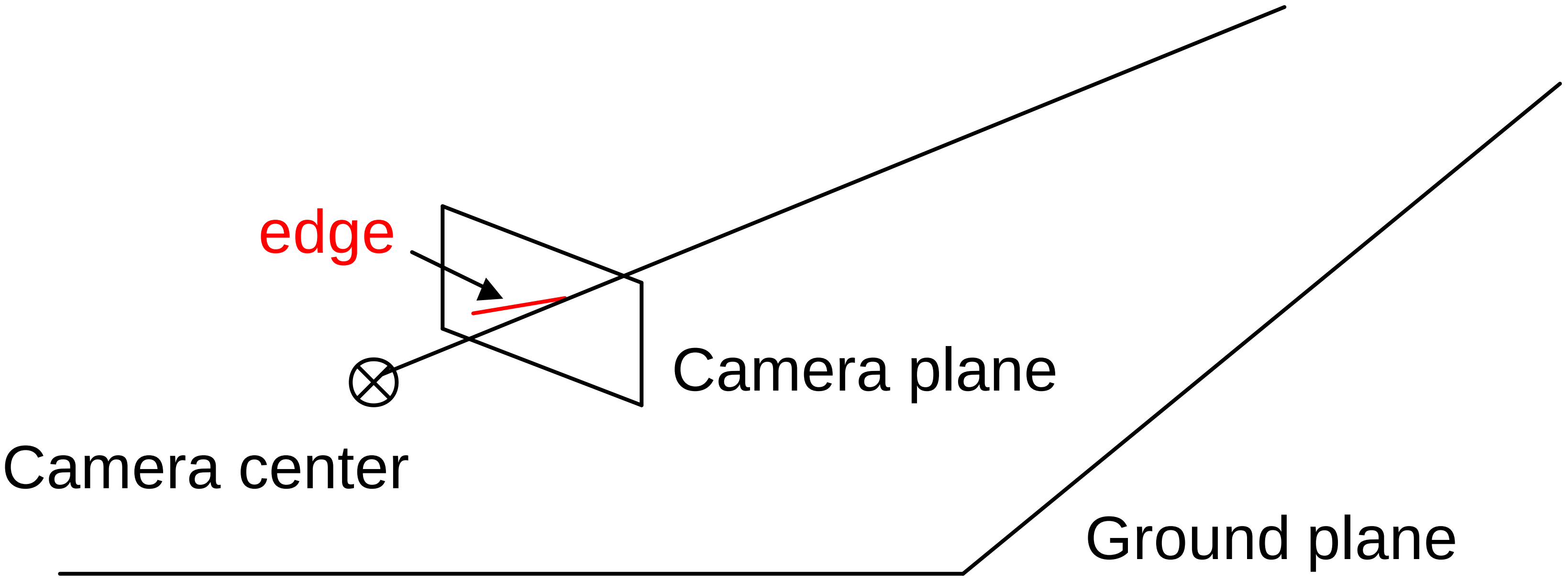
- Edge detection



- assume edges are horizontal
- determine corresponding orientation in 3D world

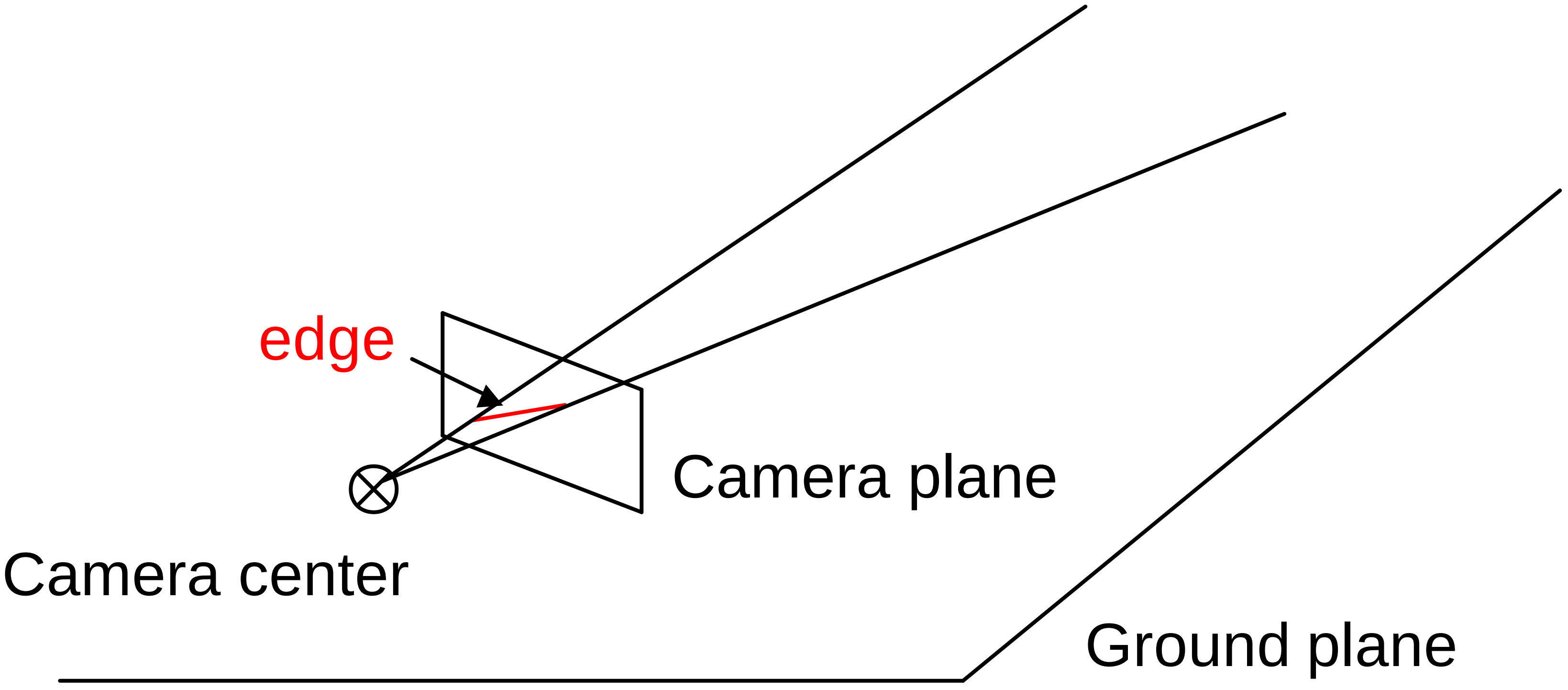


- assume edges are horizontal
- determine corresponding orientation in 3D world



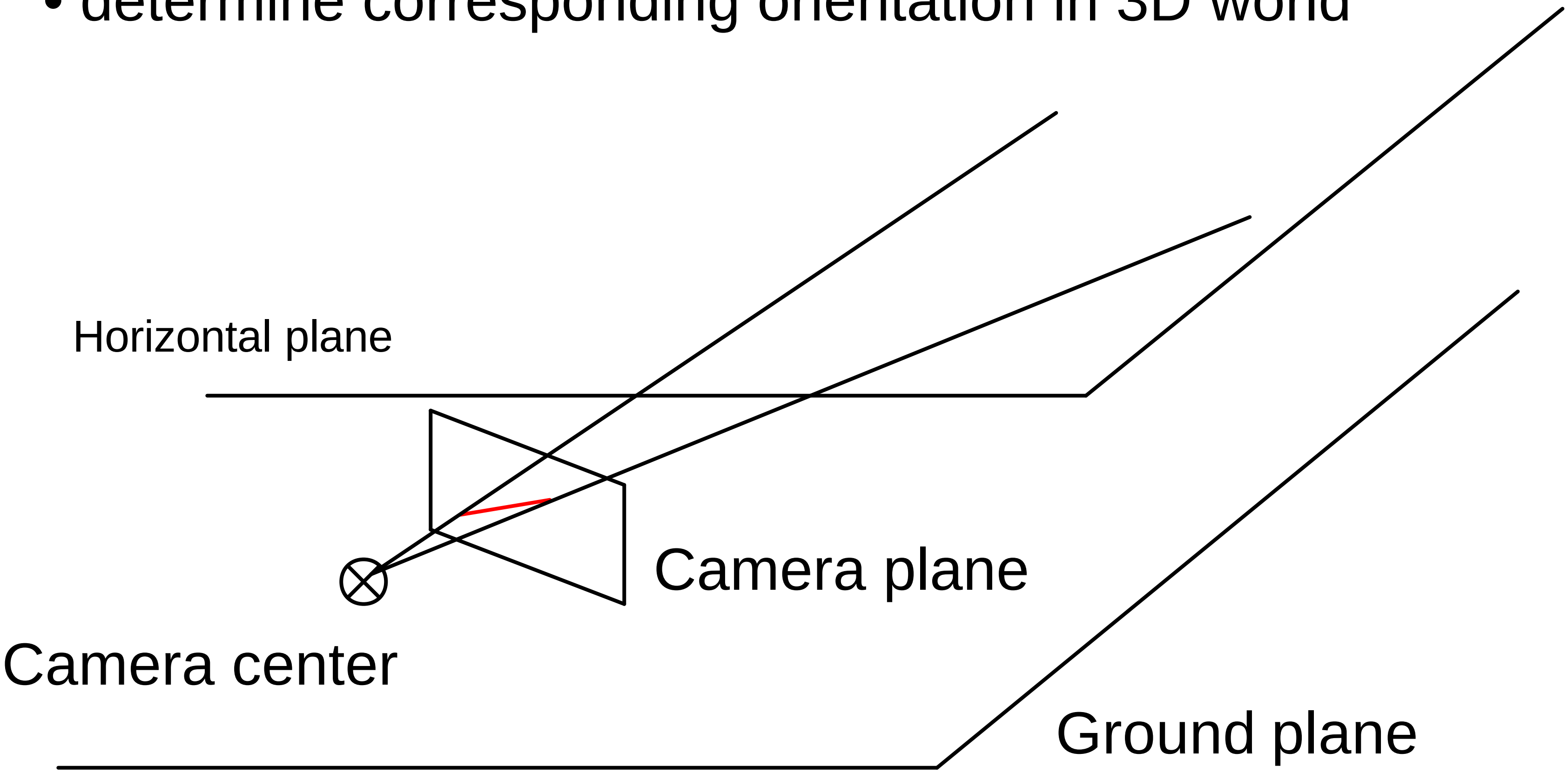
Project overview

- assume edges are horizontal
- determine corresponding orientation in 3D world



Project overview

- assume edges are horizontal
- determine corresponding orientation in 3D world



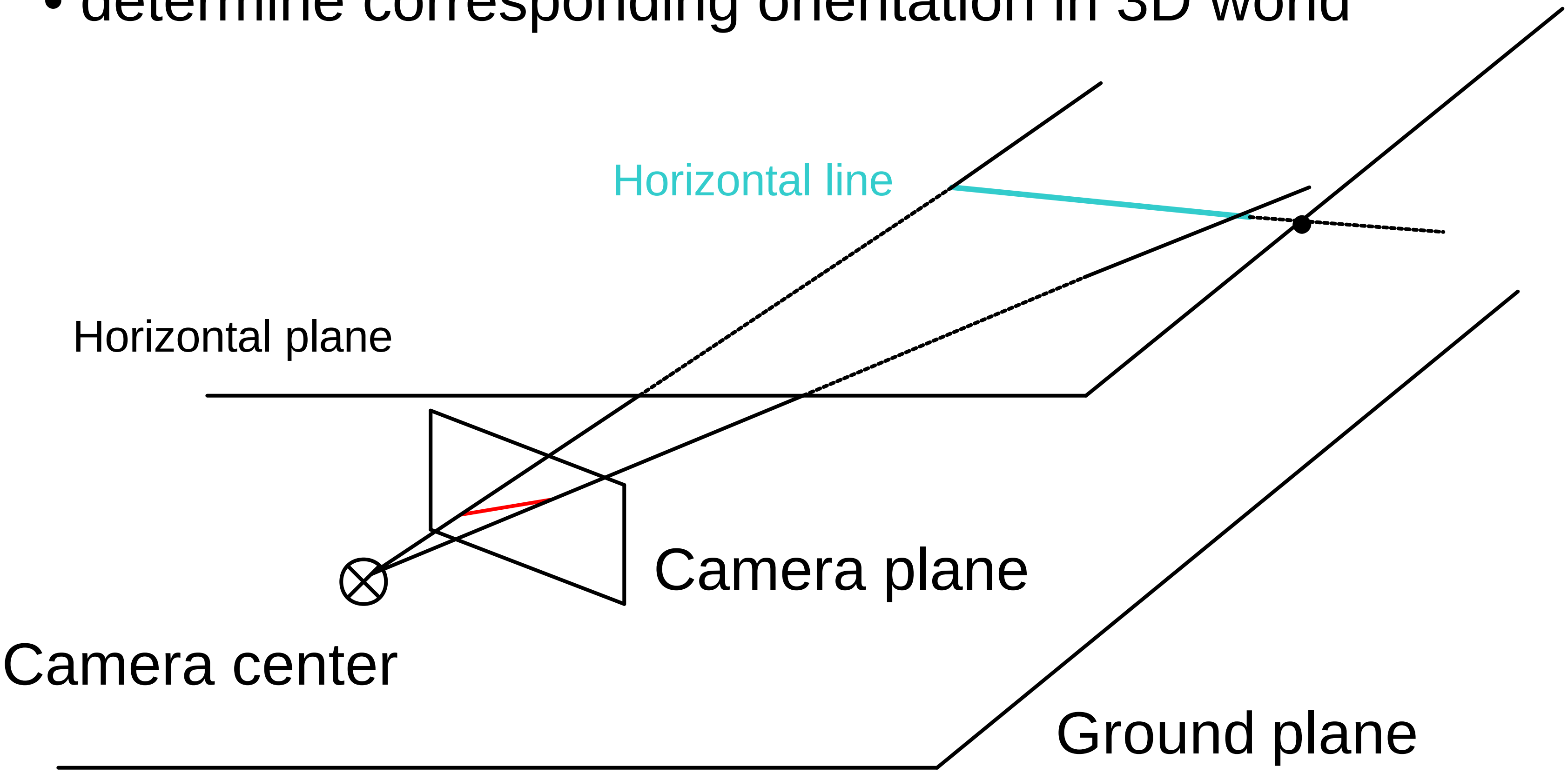
Horizontal plane

Camera plane

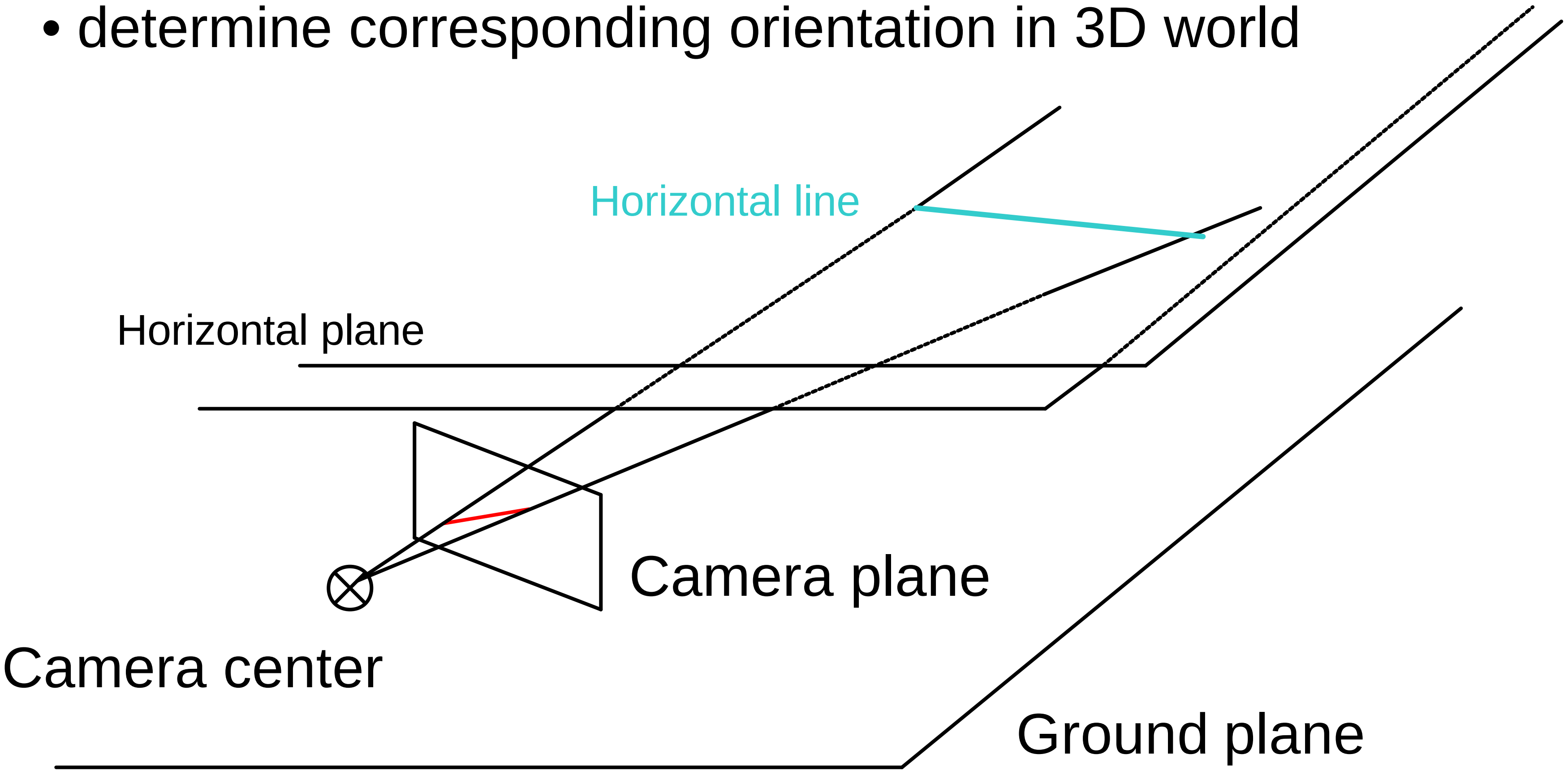
Camera center

Ground plane

- assume edges are horizontal
- determine corresponding orientation in 3D world

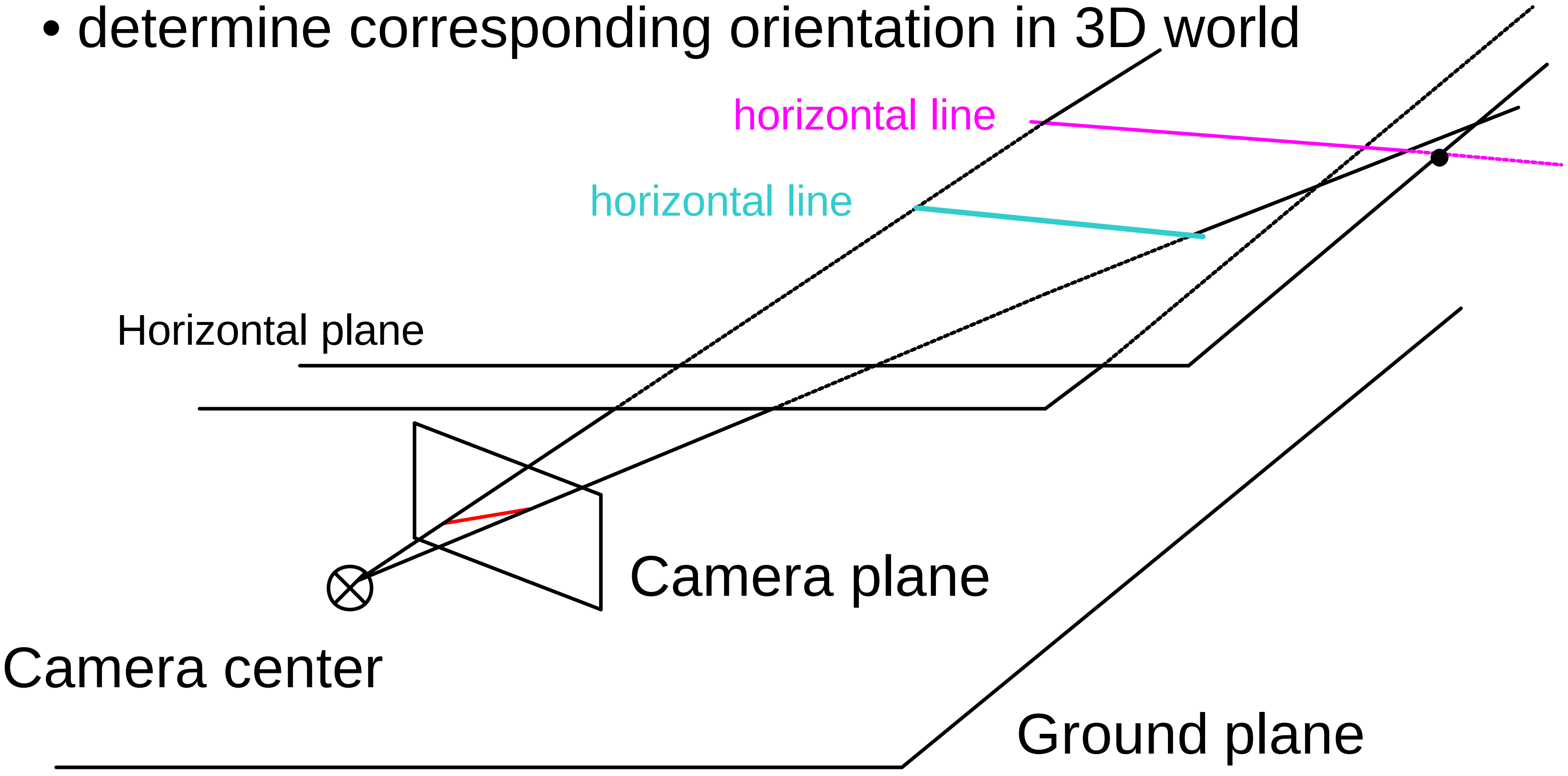


- assume edges are horizontal
- determine corresponding orientation in 3D world

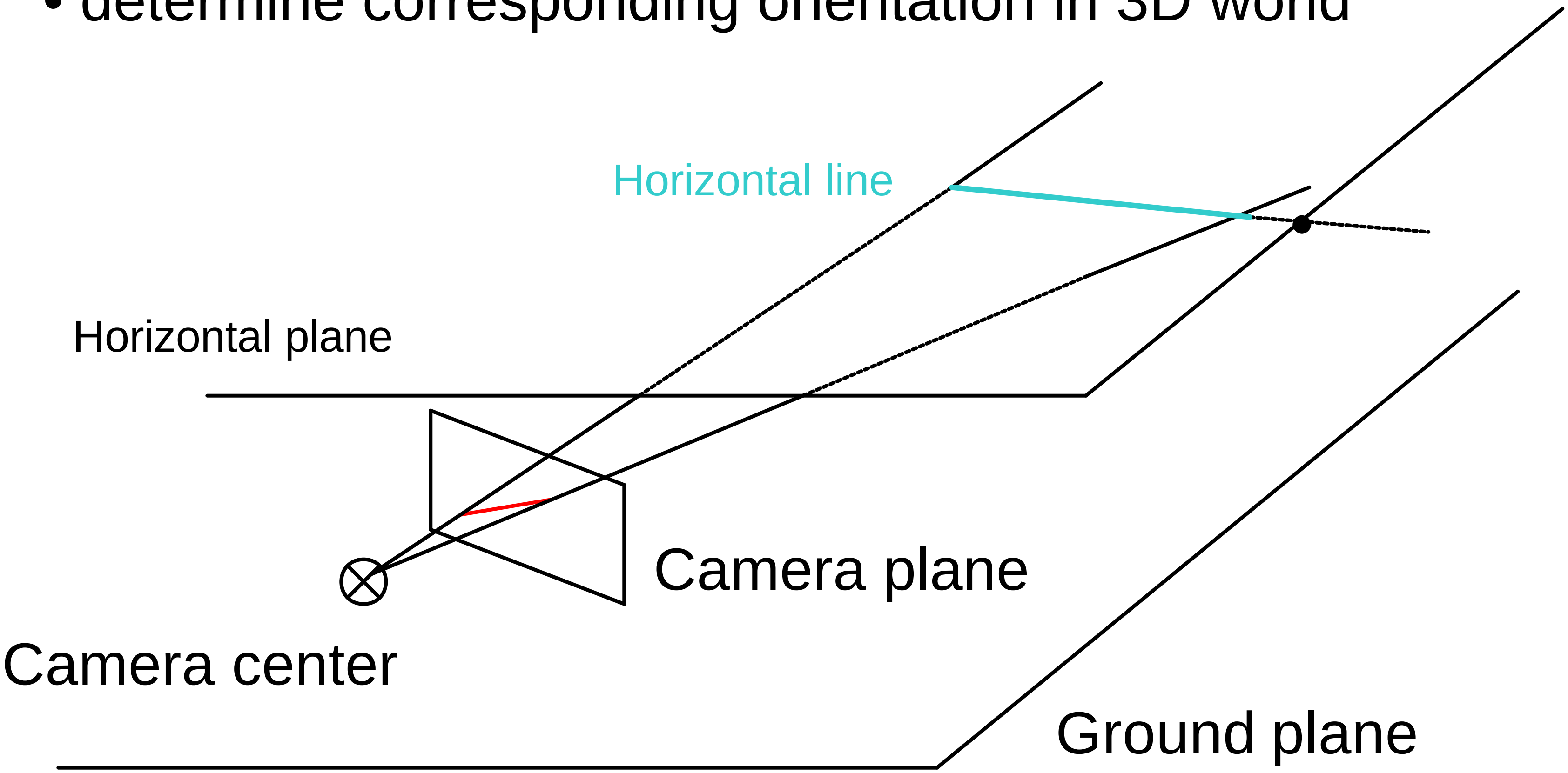


Project overview

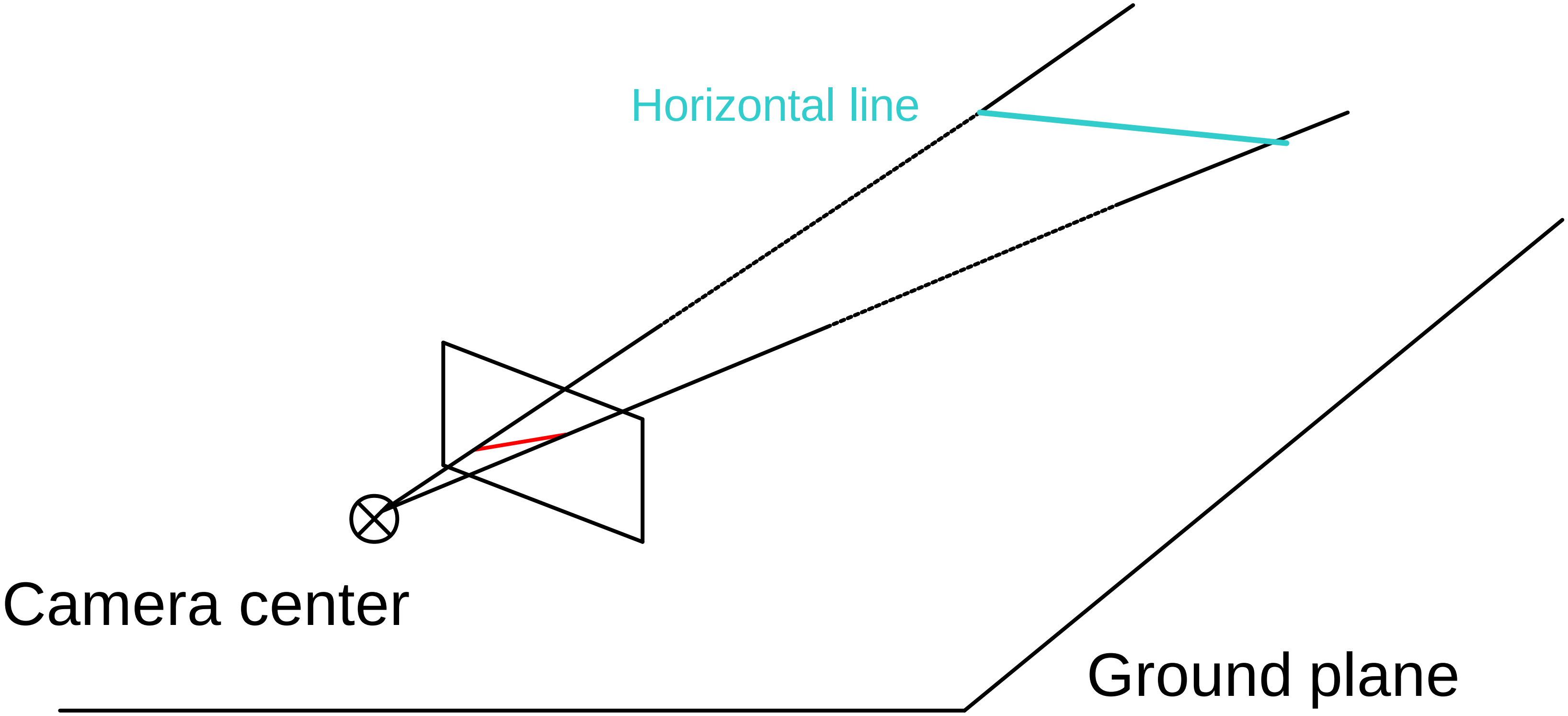
- assume edges are horizontal
- determine corresponding orientation in 3D world



- assume edges are horizontal
- determine corresponding orientation in 3D world

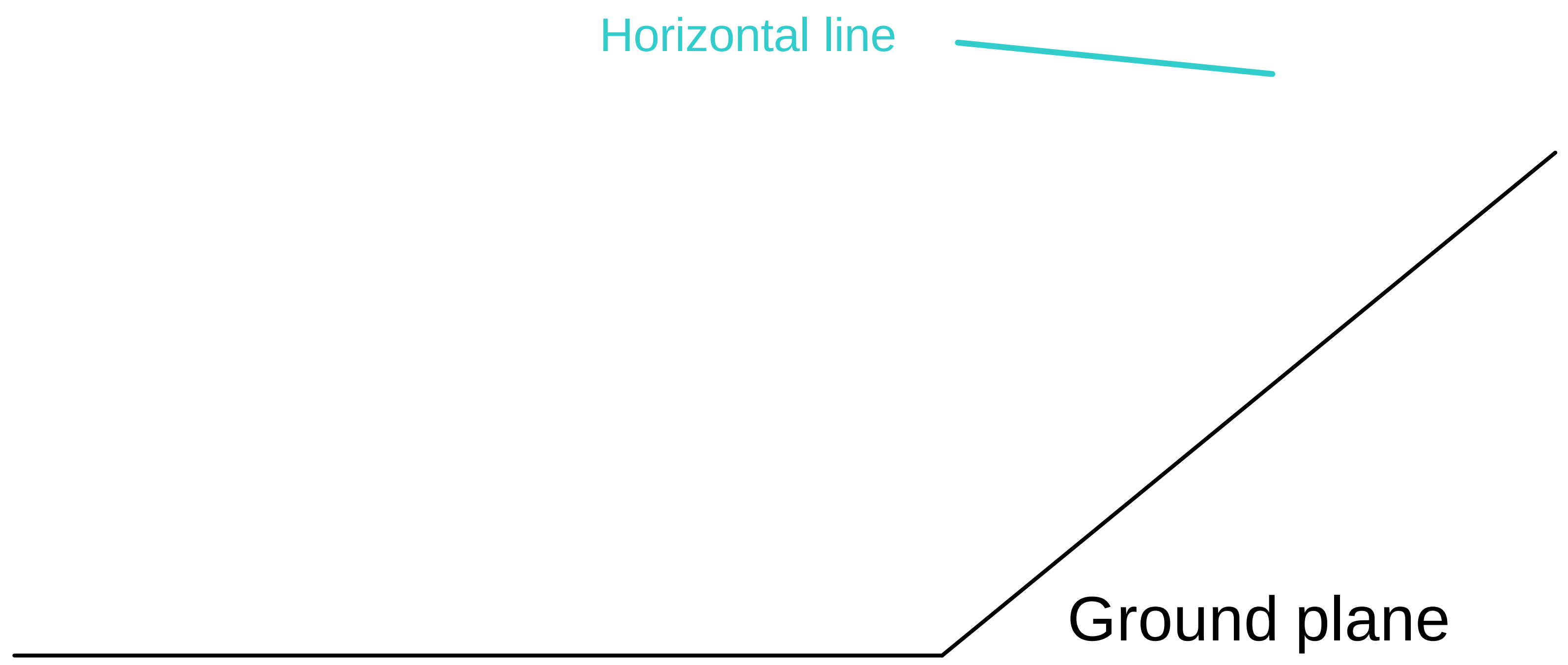


- assume edges are horizontal
- determine corresponding orientation in 3D world



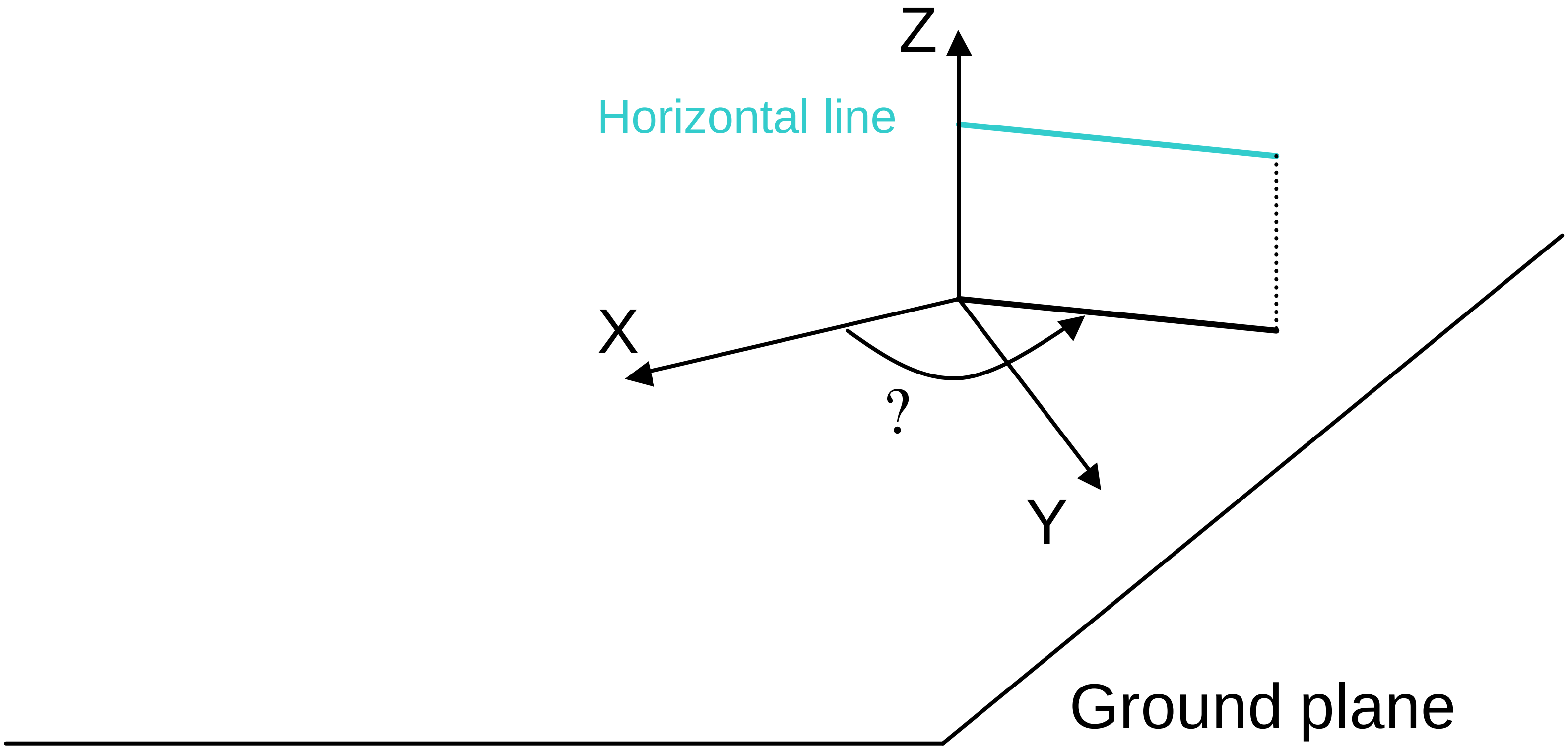
Project overview

- assume edges are horizontal
- determine corresponding orientation in 3D world



Project overview

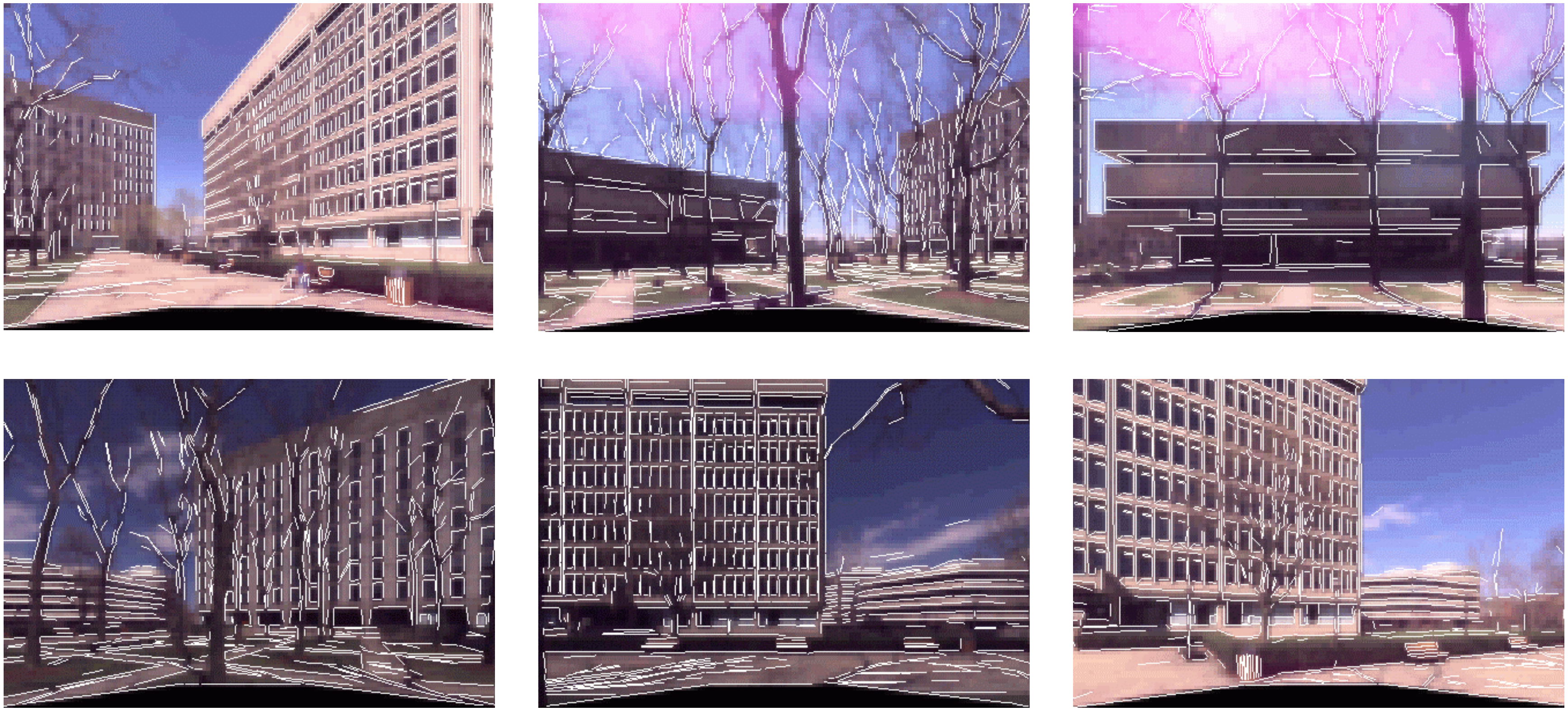
- assume edges are horizontal
- determine corresponding orientation in 3D world



? = azimuth of 3D edge

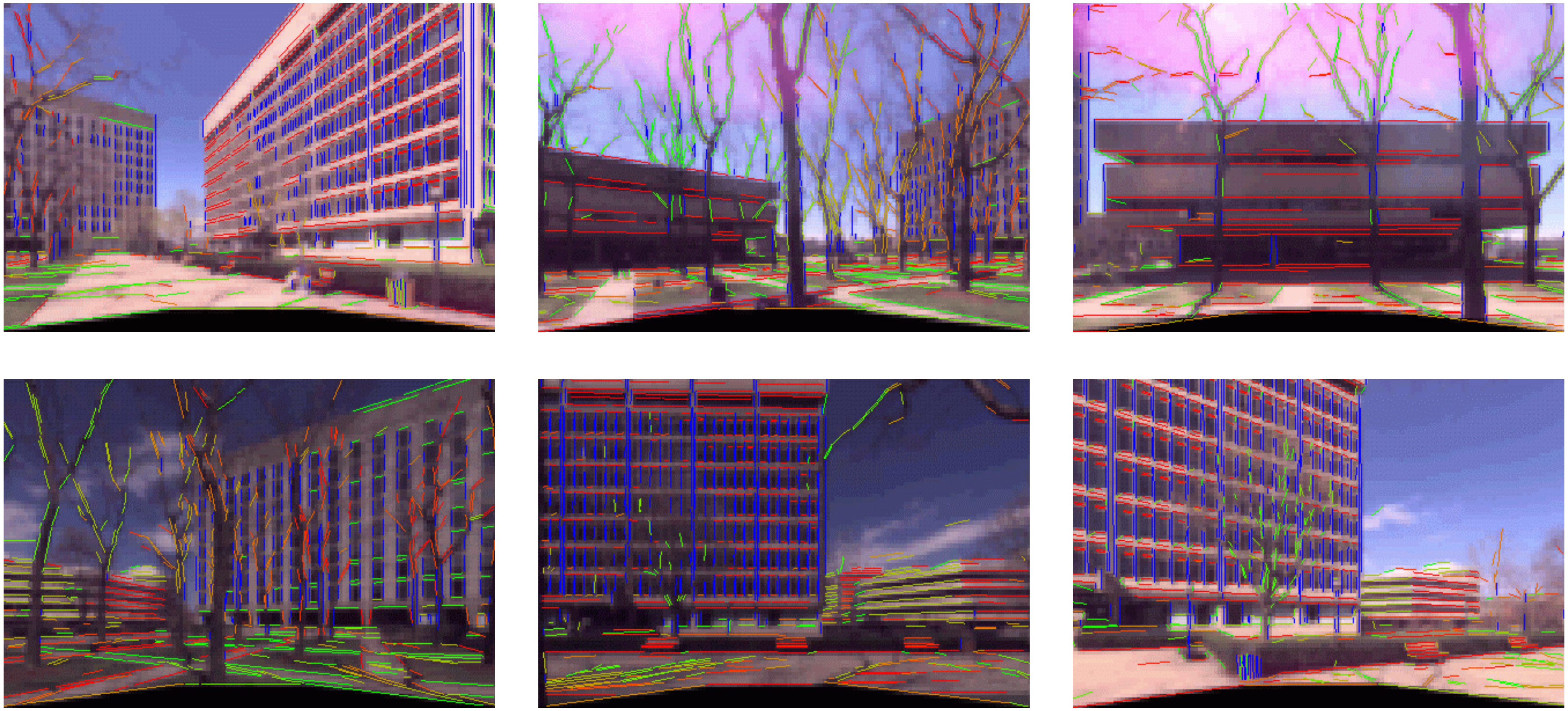
Project overview

- Edge detected…



Project overview

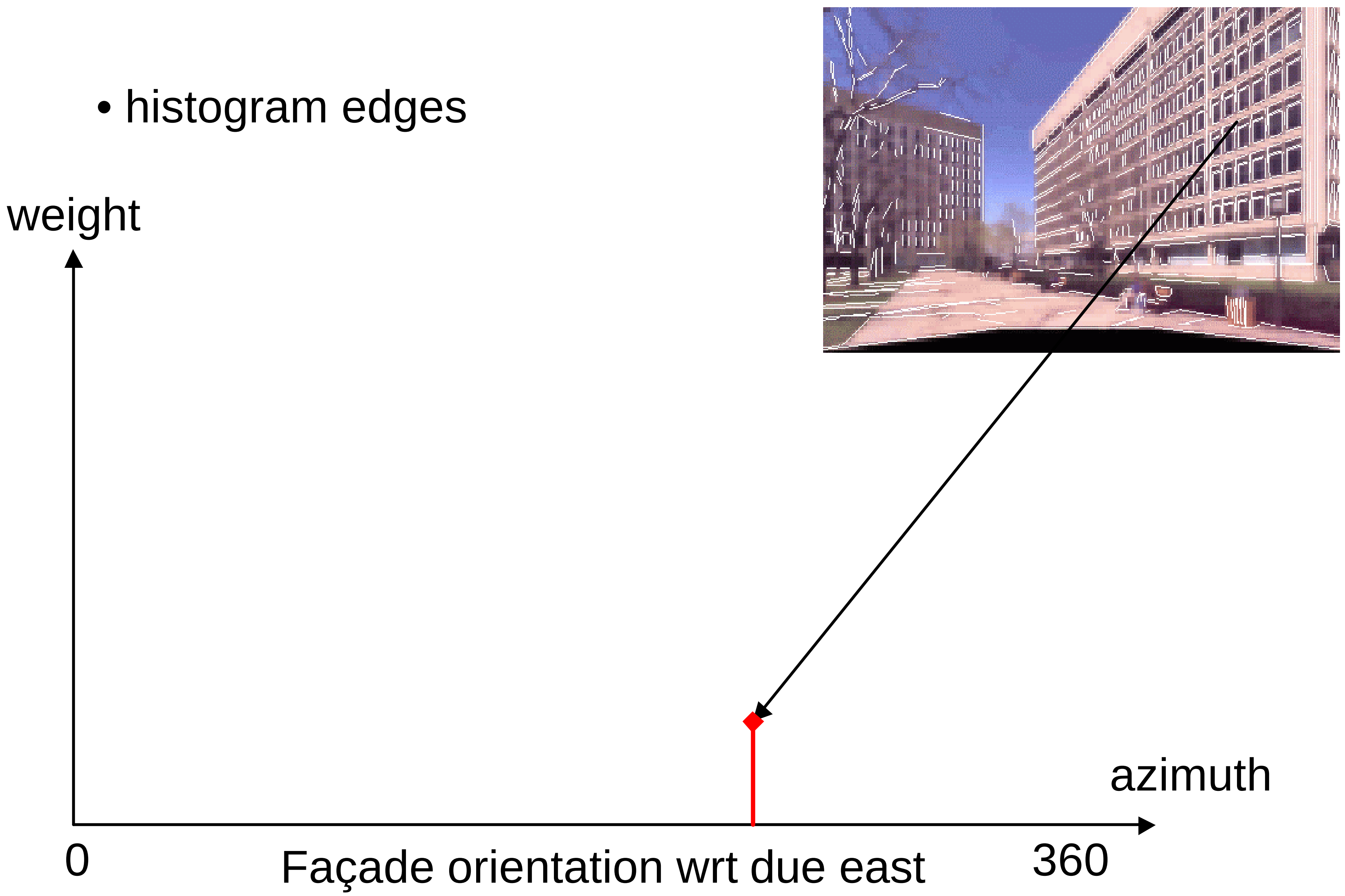
- Edge detected… now have an azimuth !



- each azimuth is color-coded.

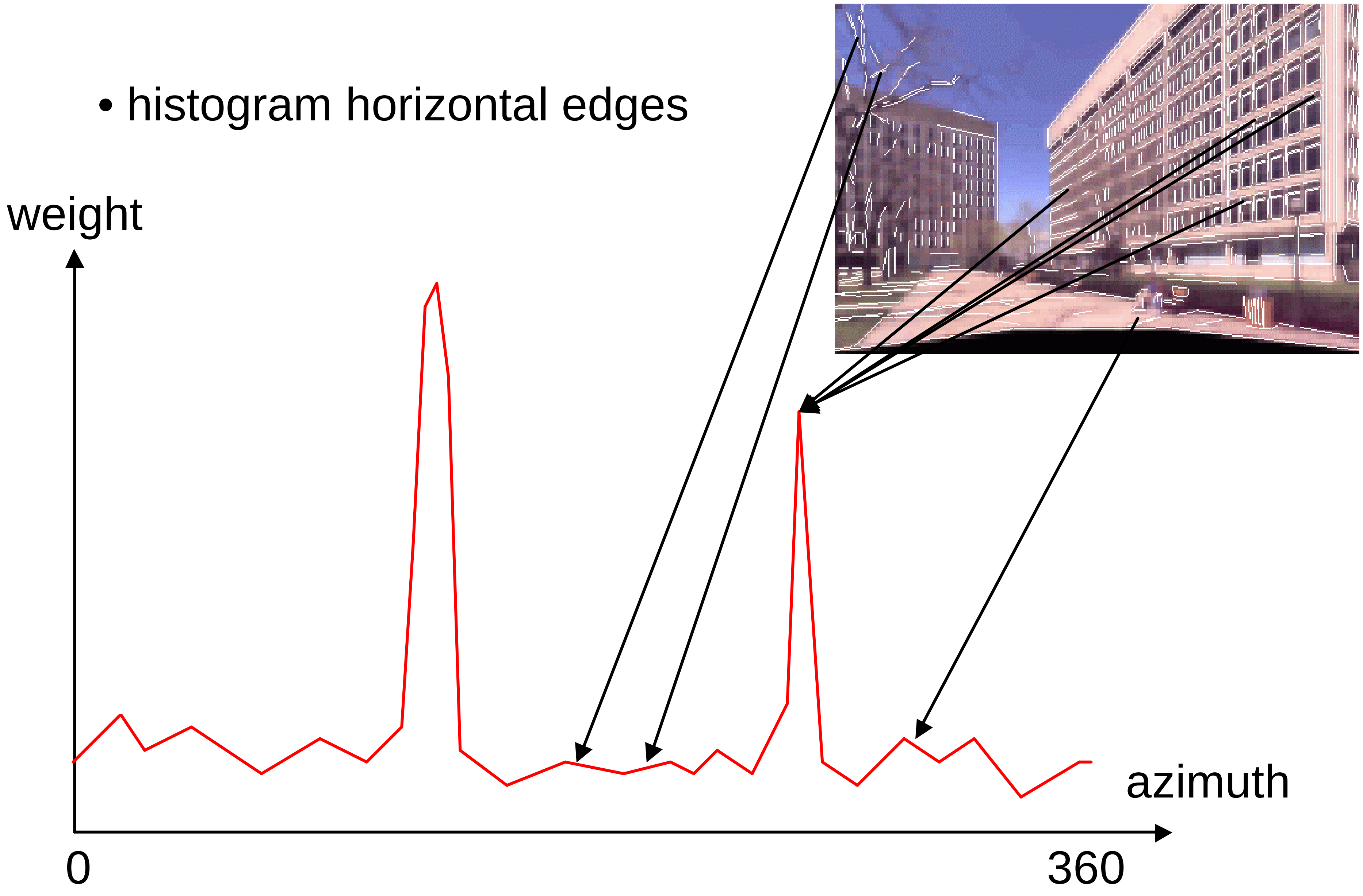
Project overview

- histogram edges



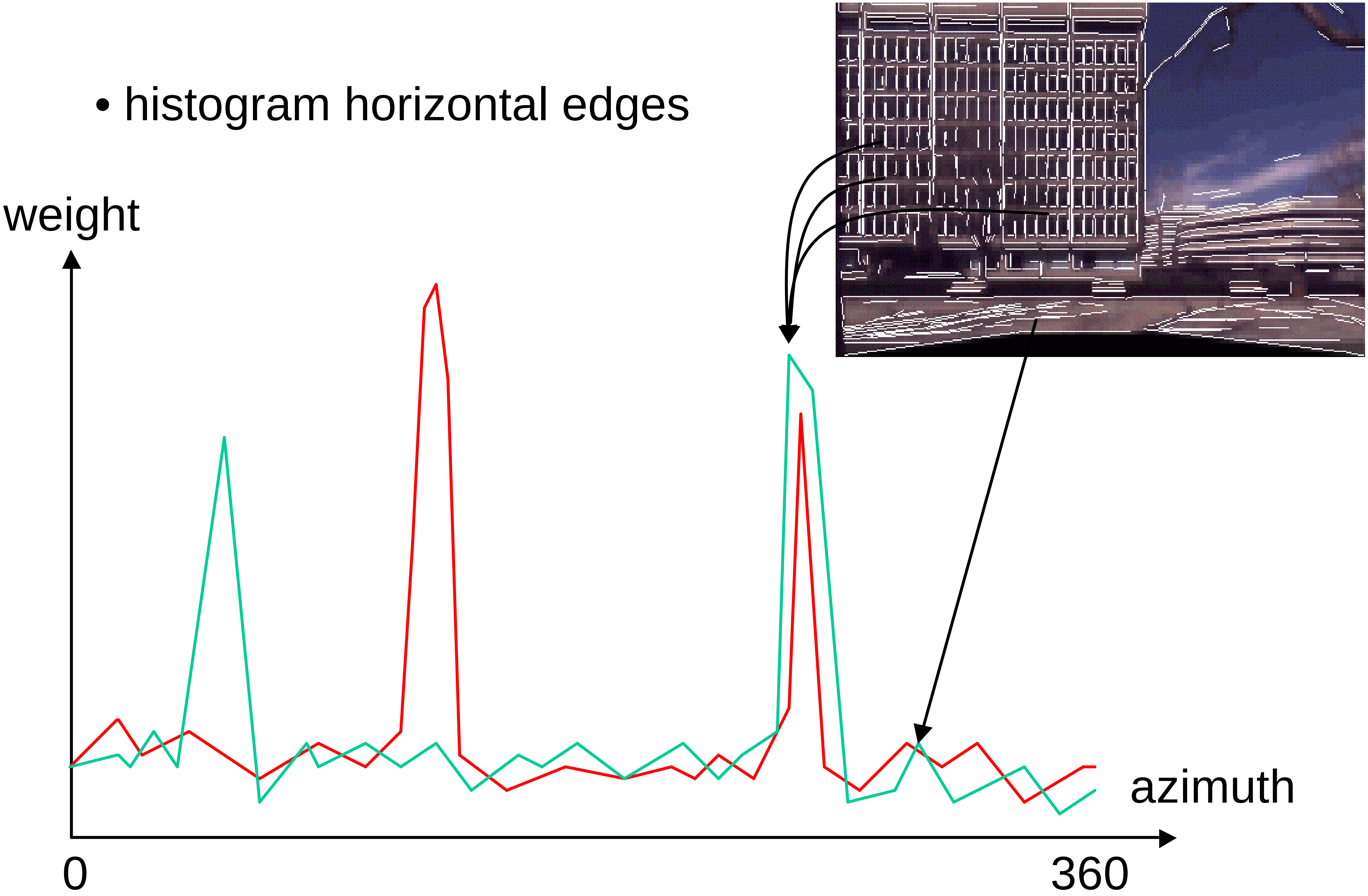
Project overview

- histogram horizontal edges



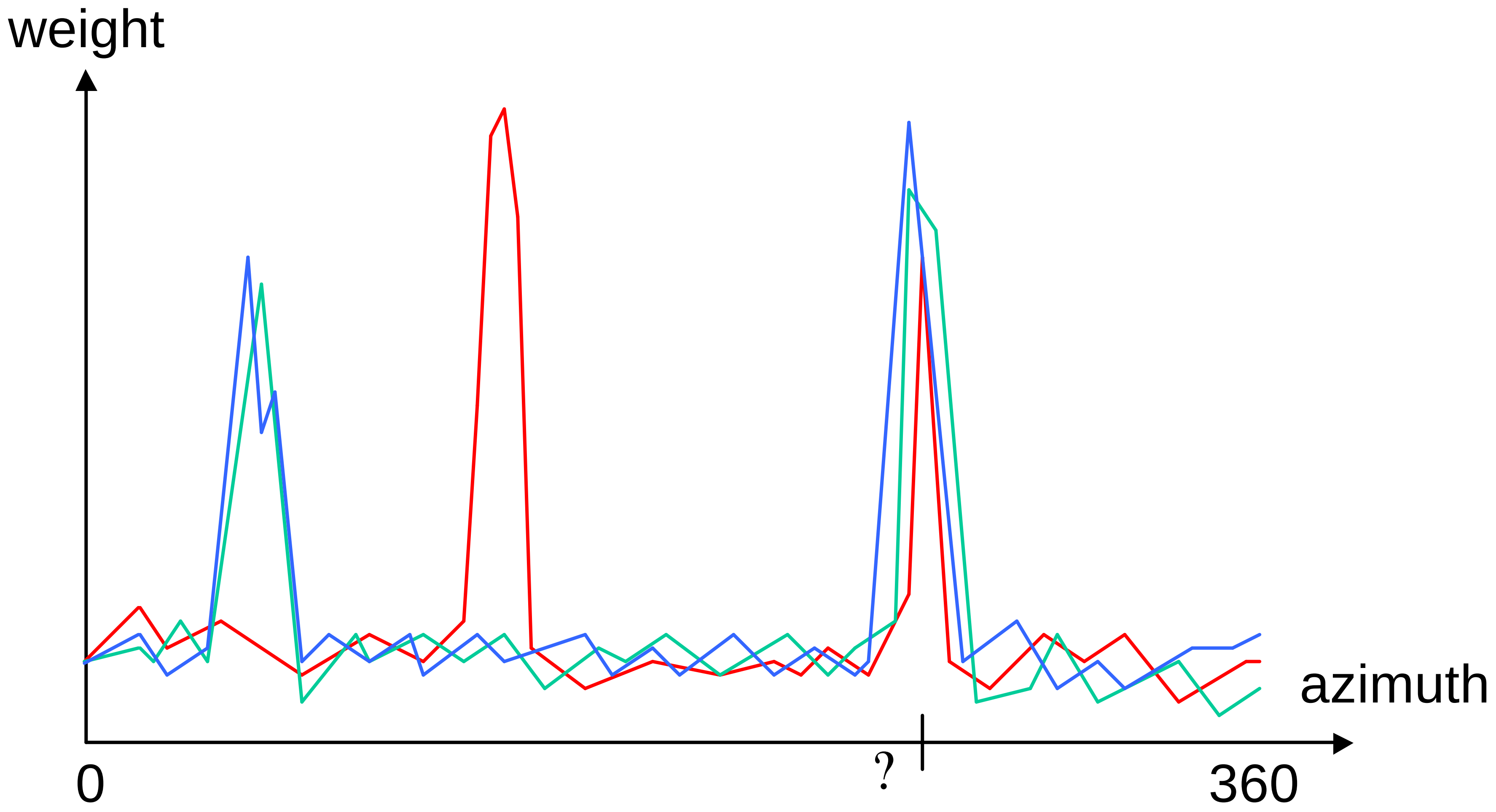
Project overview

- histogram horizontal edges



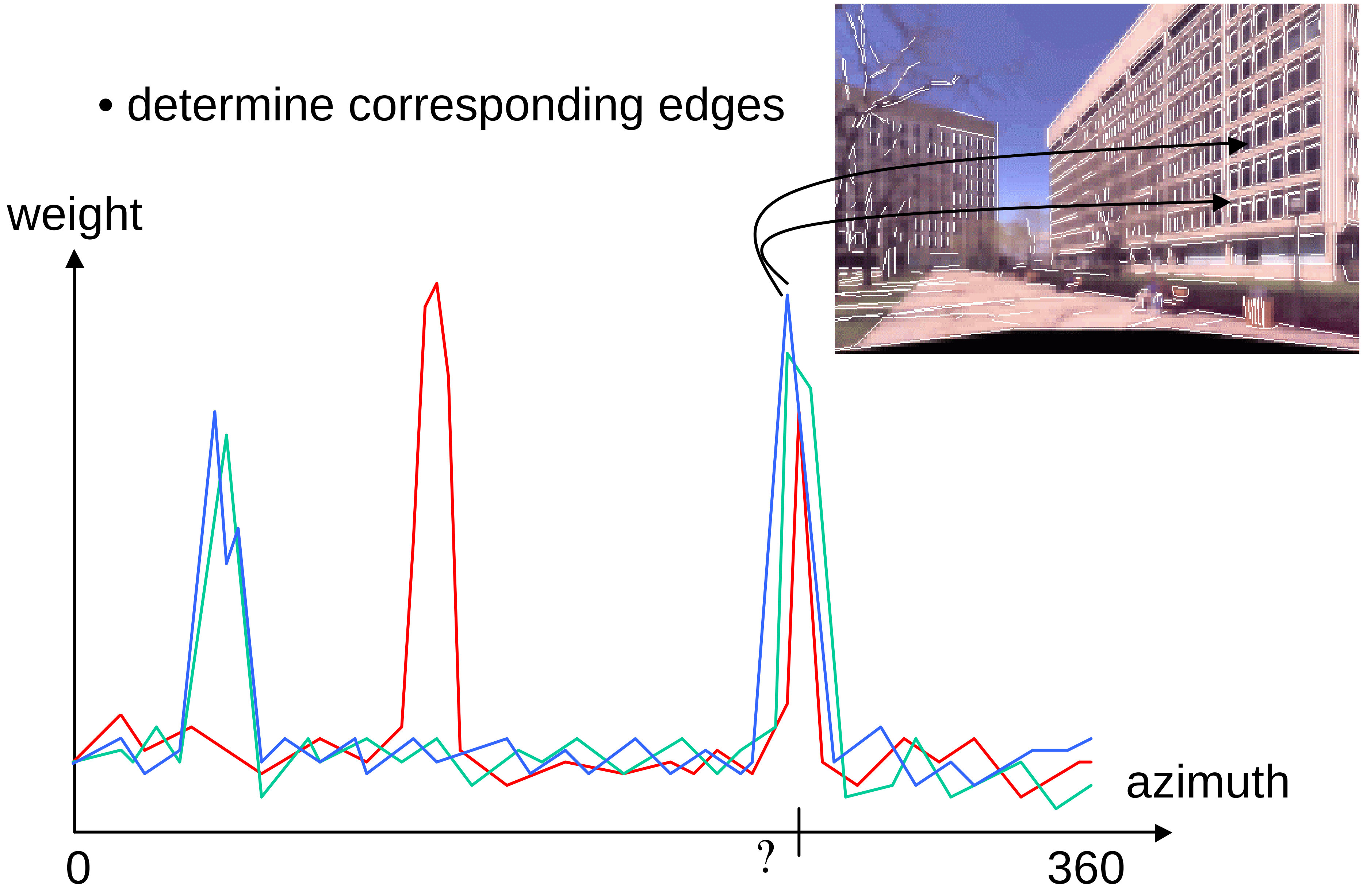
Project overview

- keep orientations that are supported by at least 3 nodes

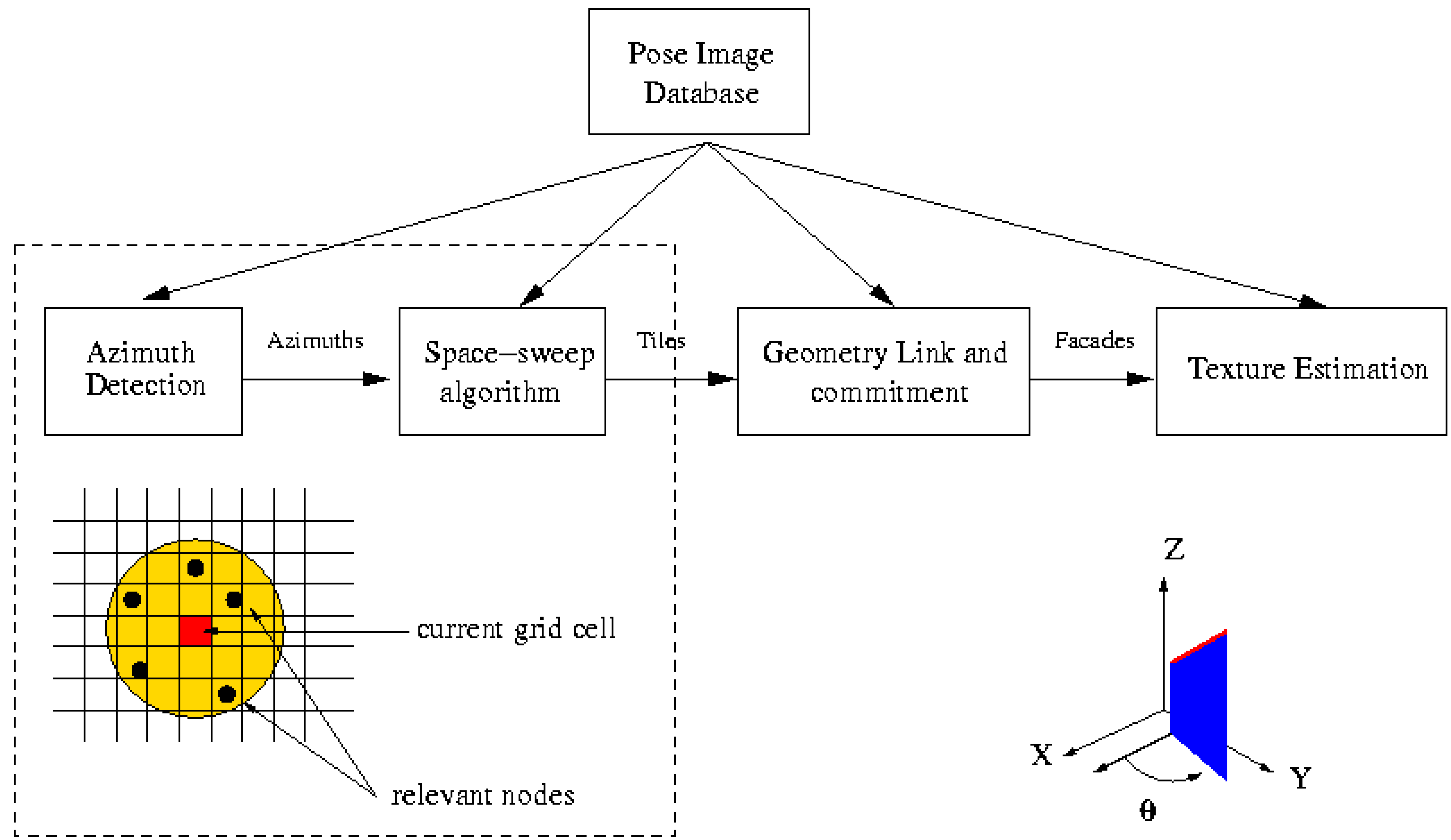


Project overview

- determine corresponding edges

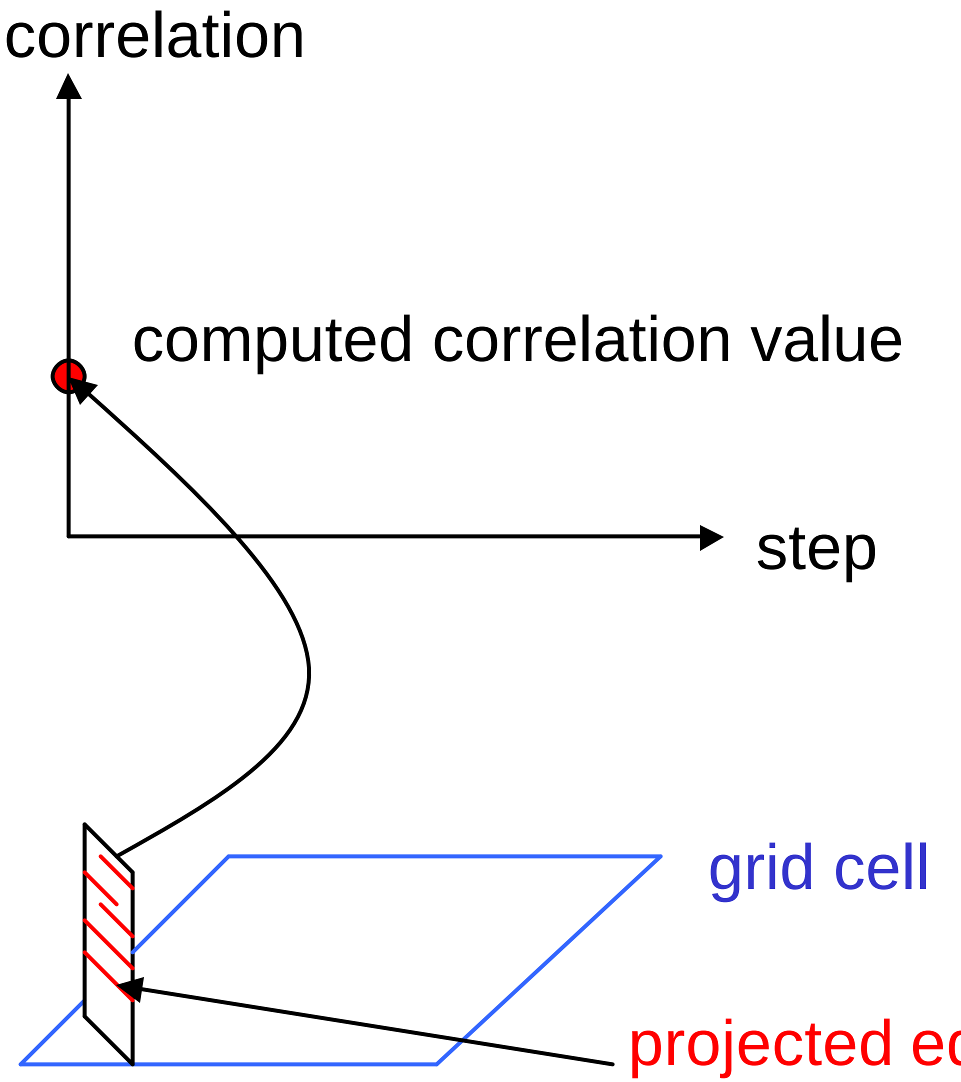


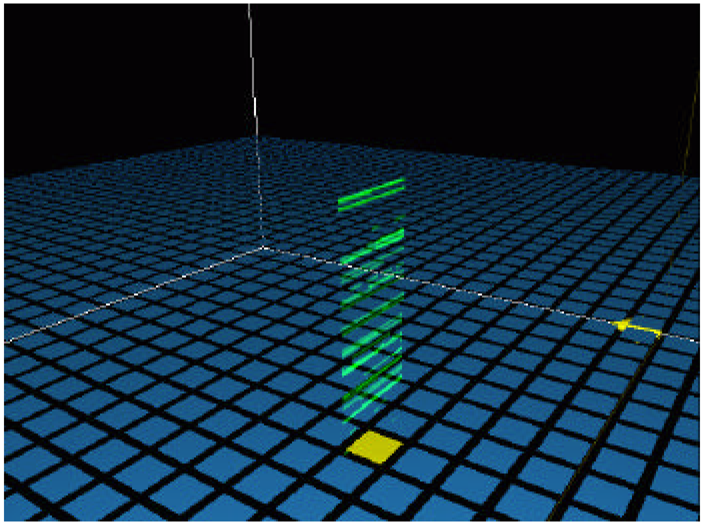
Project overview



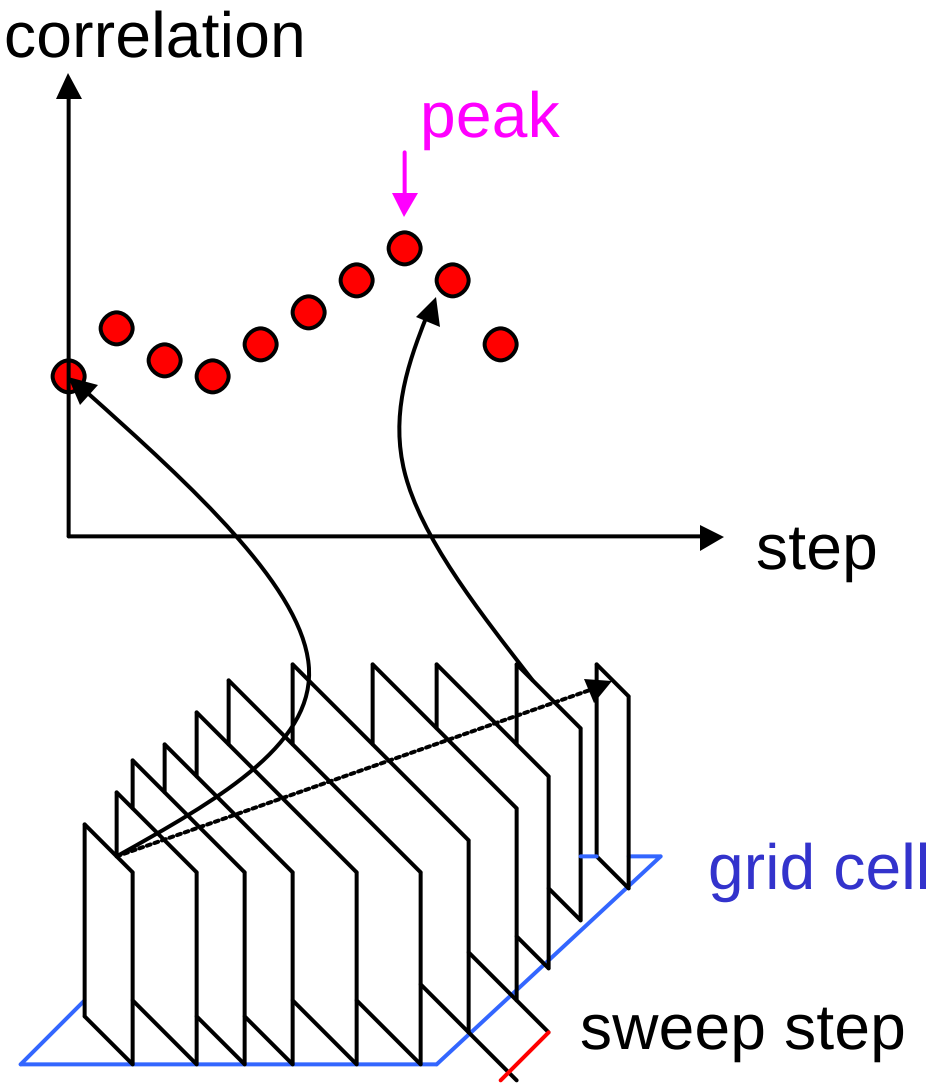
From : *Automatic Extraction of Textured Vertical Facades from Pose Imagery*, S. Coorg, S. Teller, 1998
MIT LCS TR-729

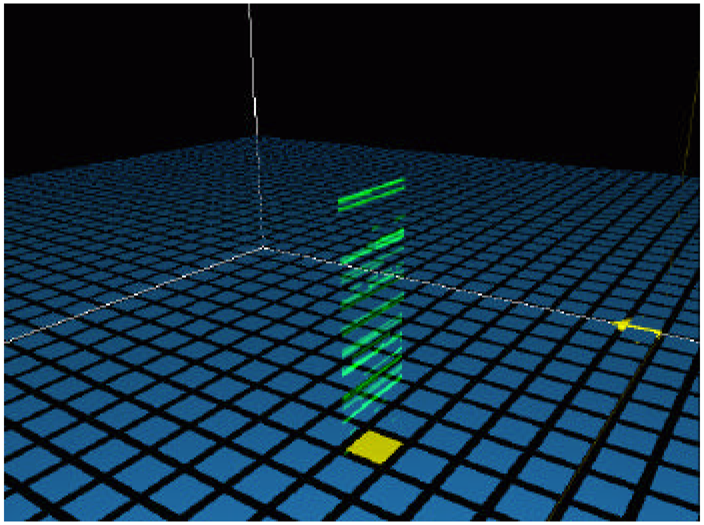
Project overview



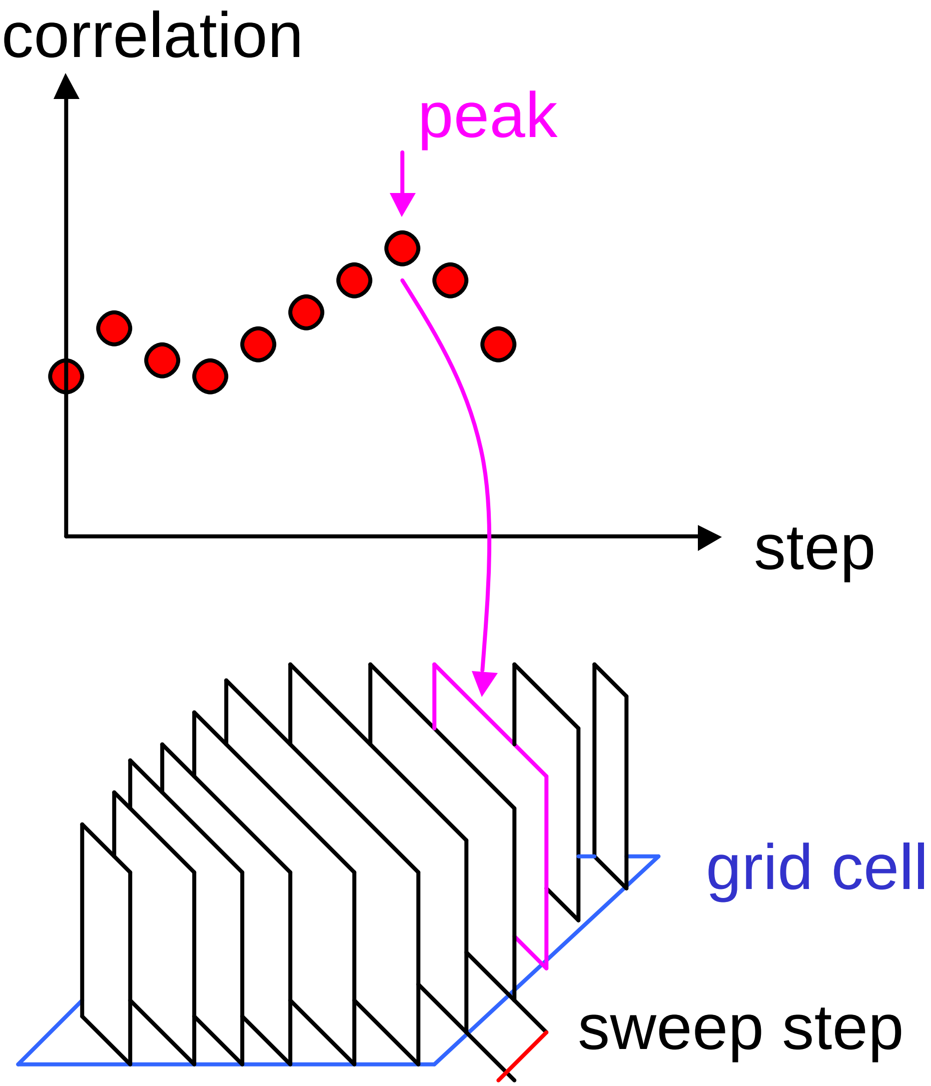


Project overview





Project overview



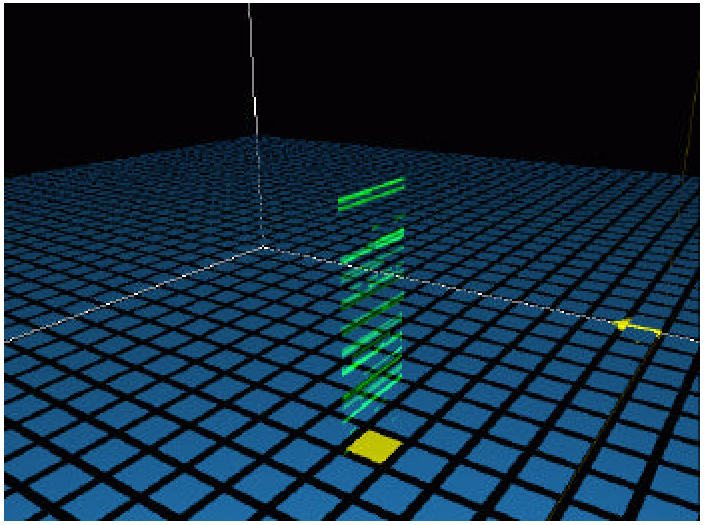
correlation

peak

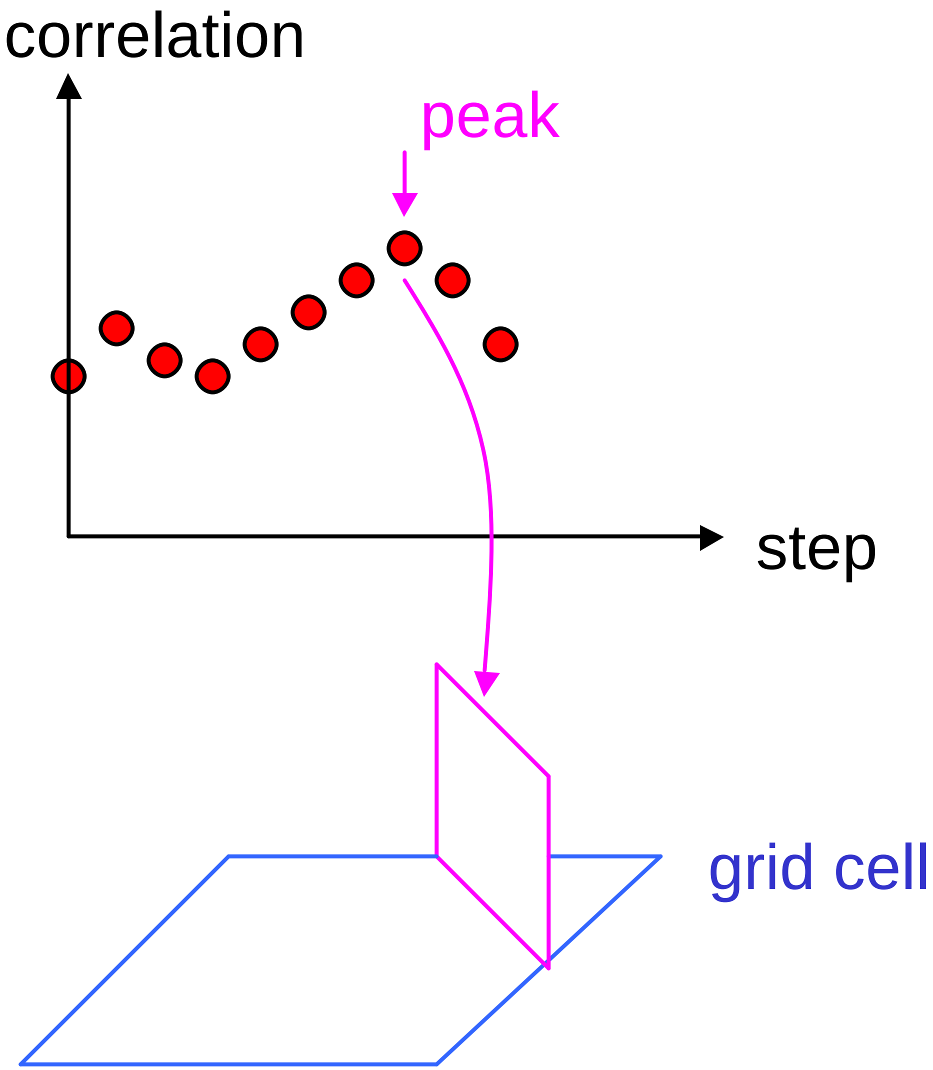
step

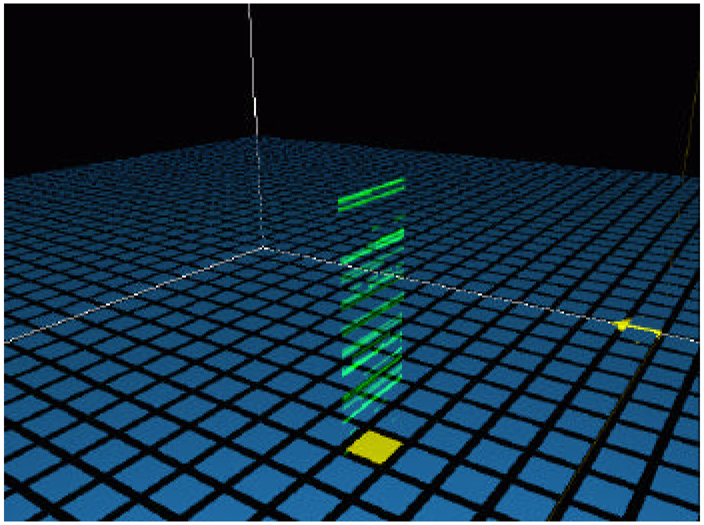
grid cell

sweep step

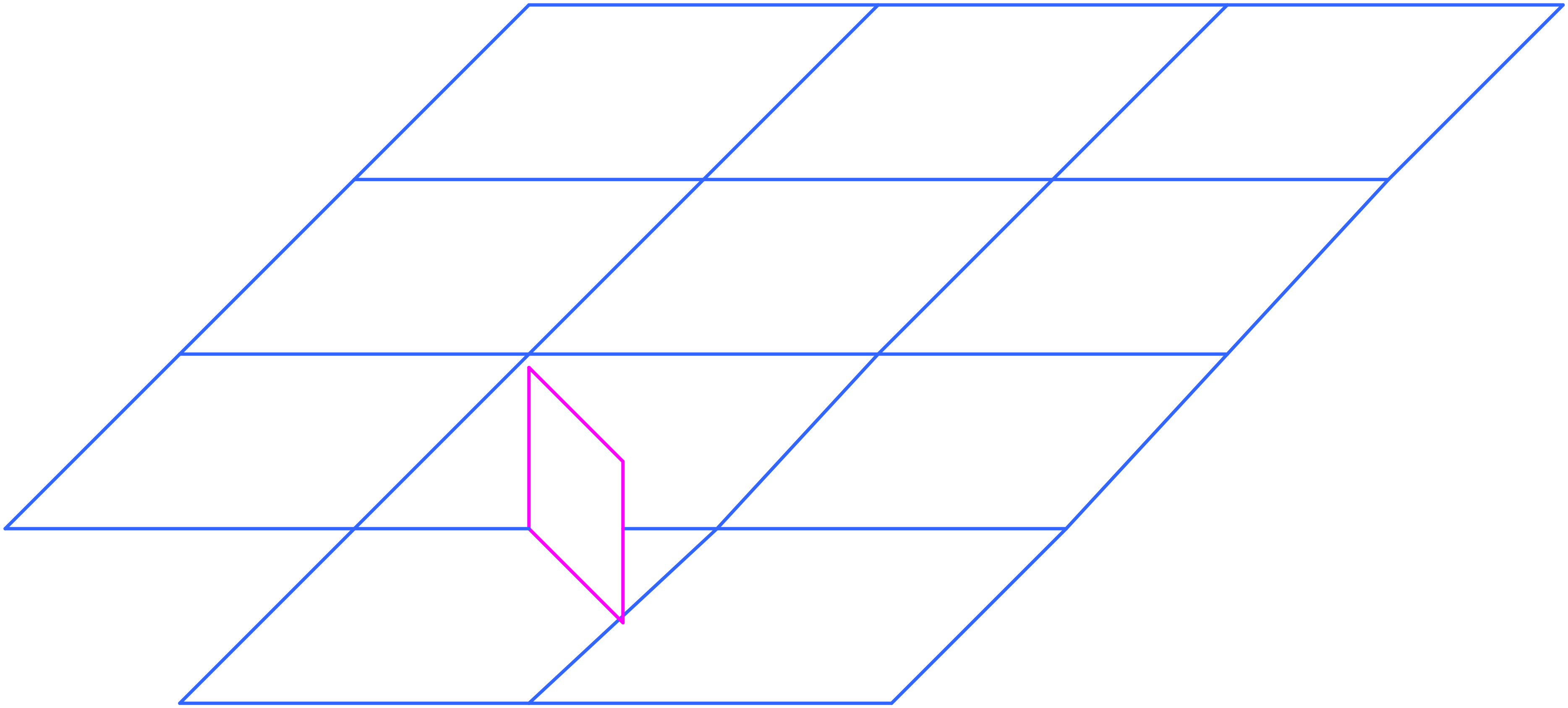


Project overview

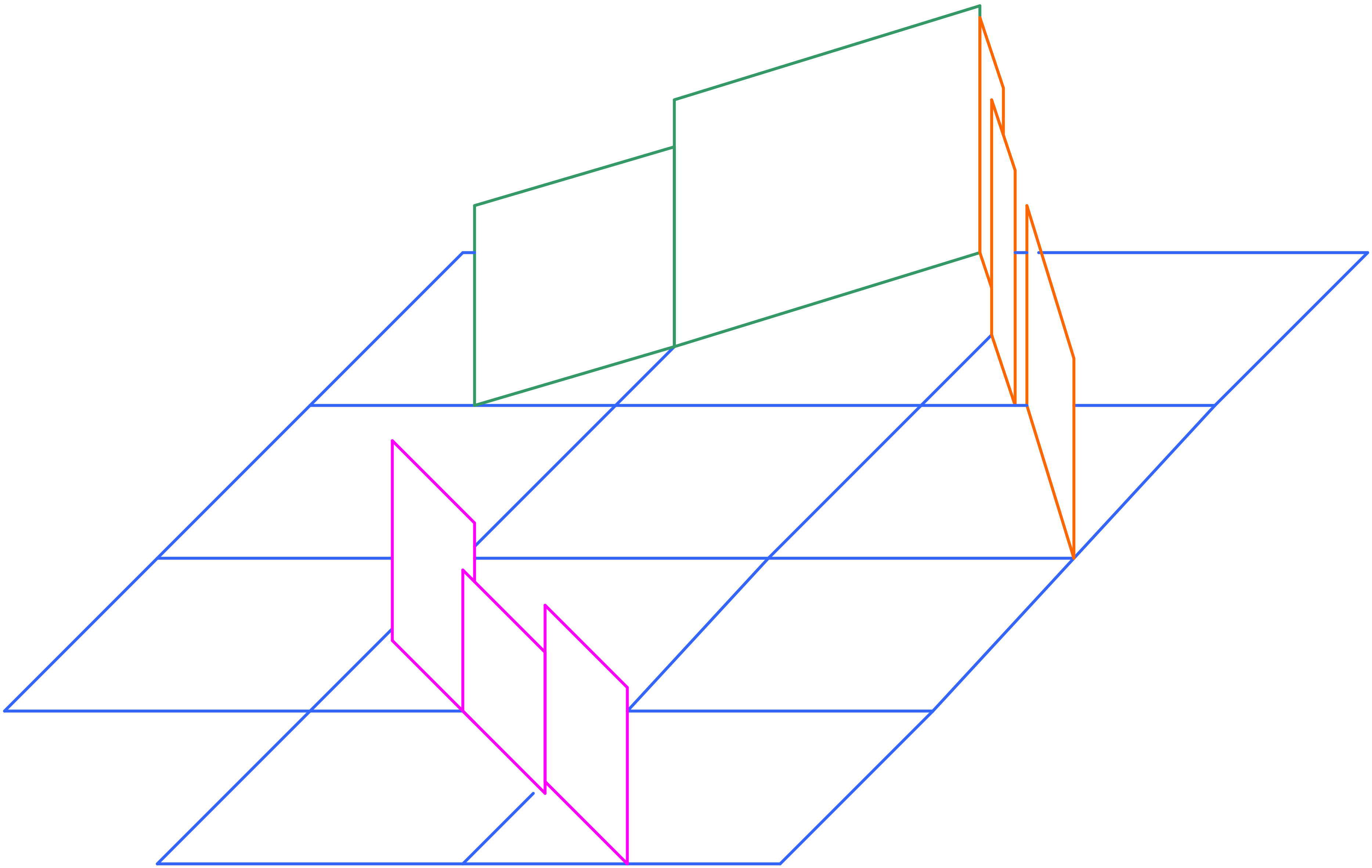




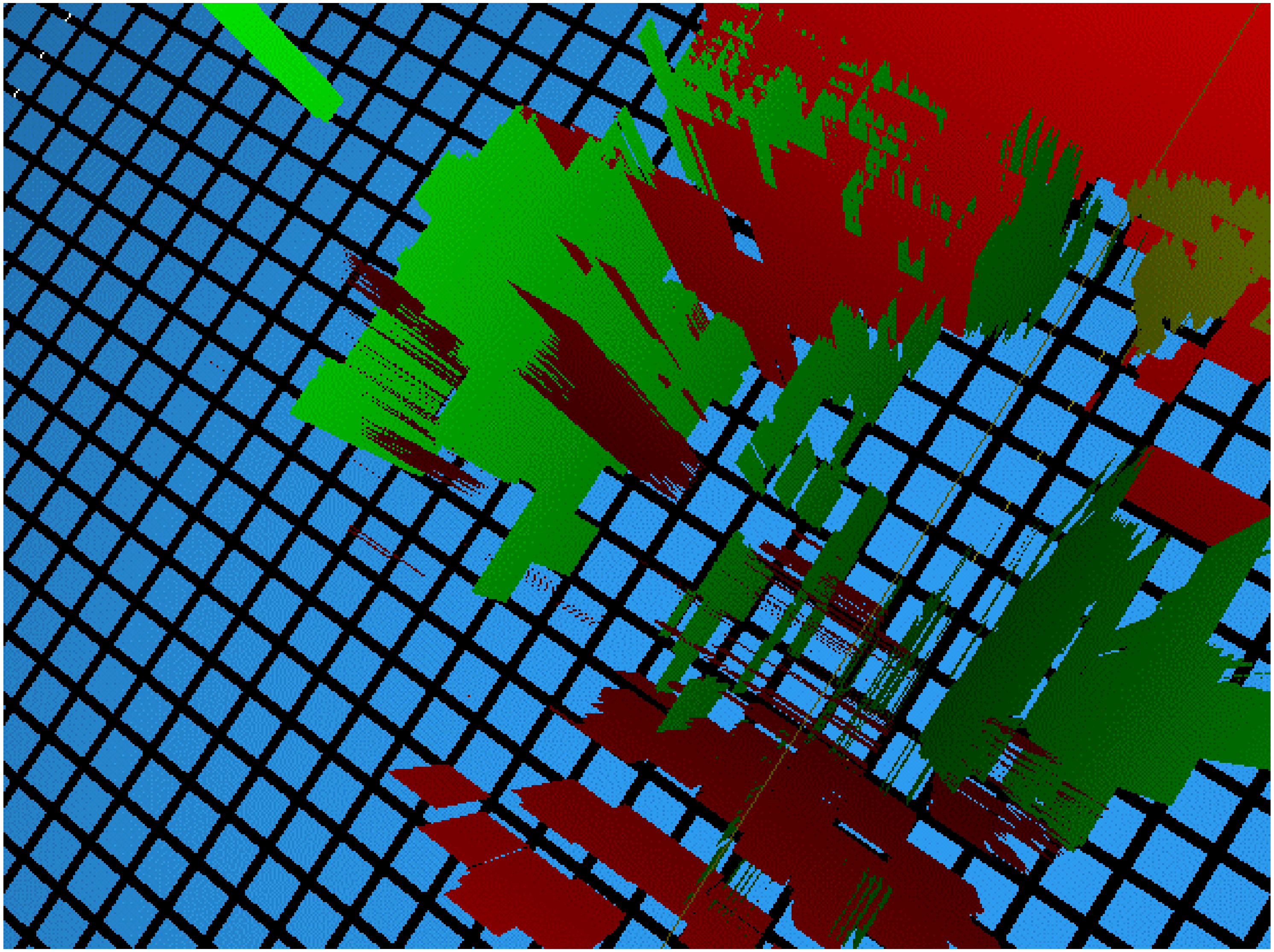
Project overview



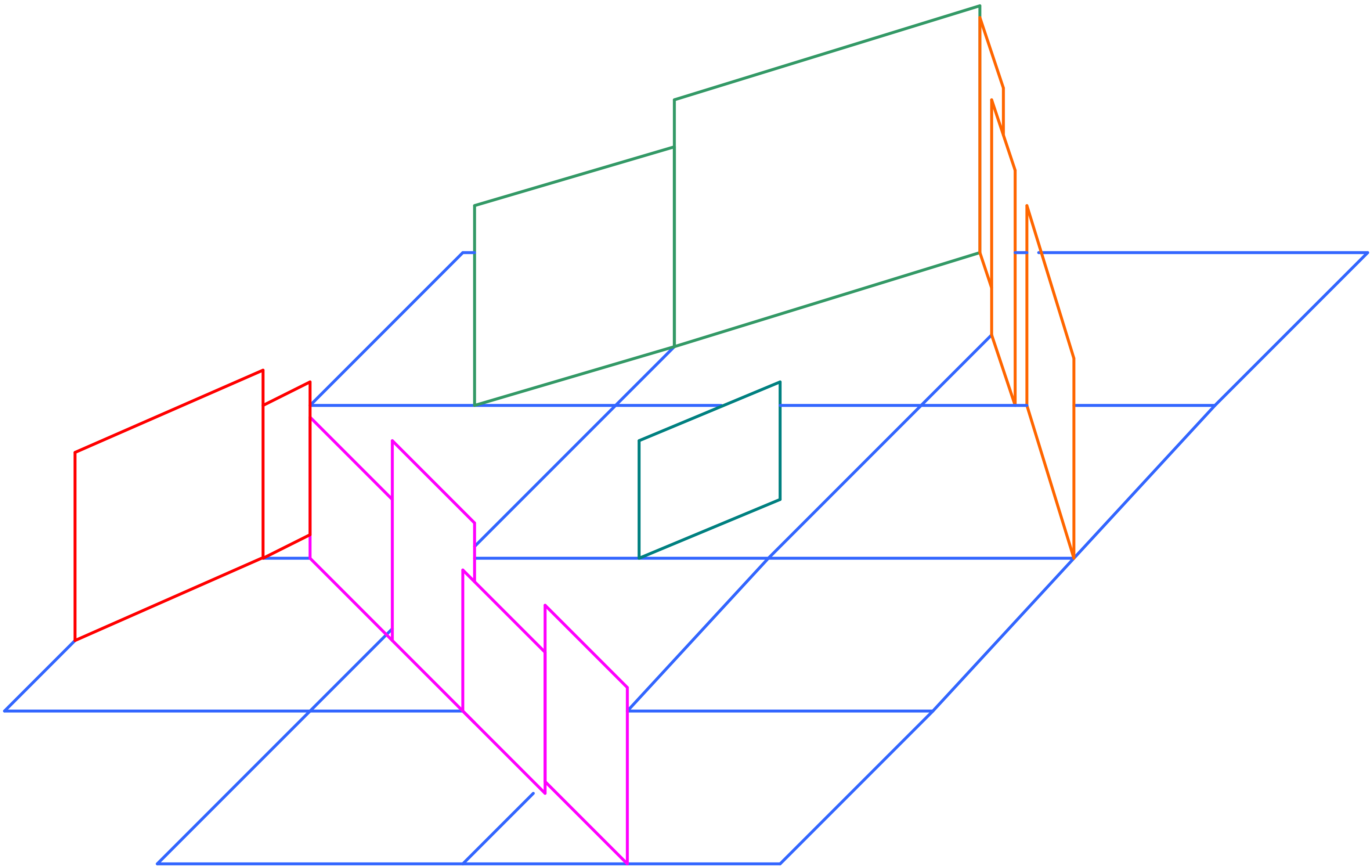
Project overview



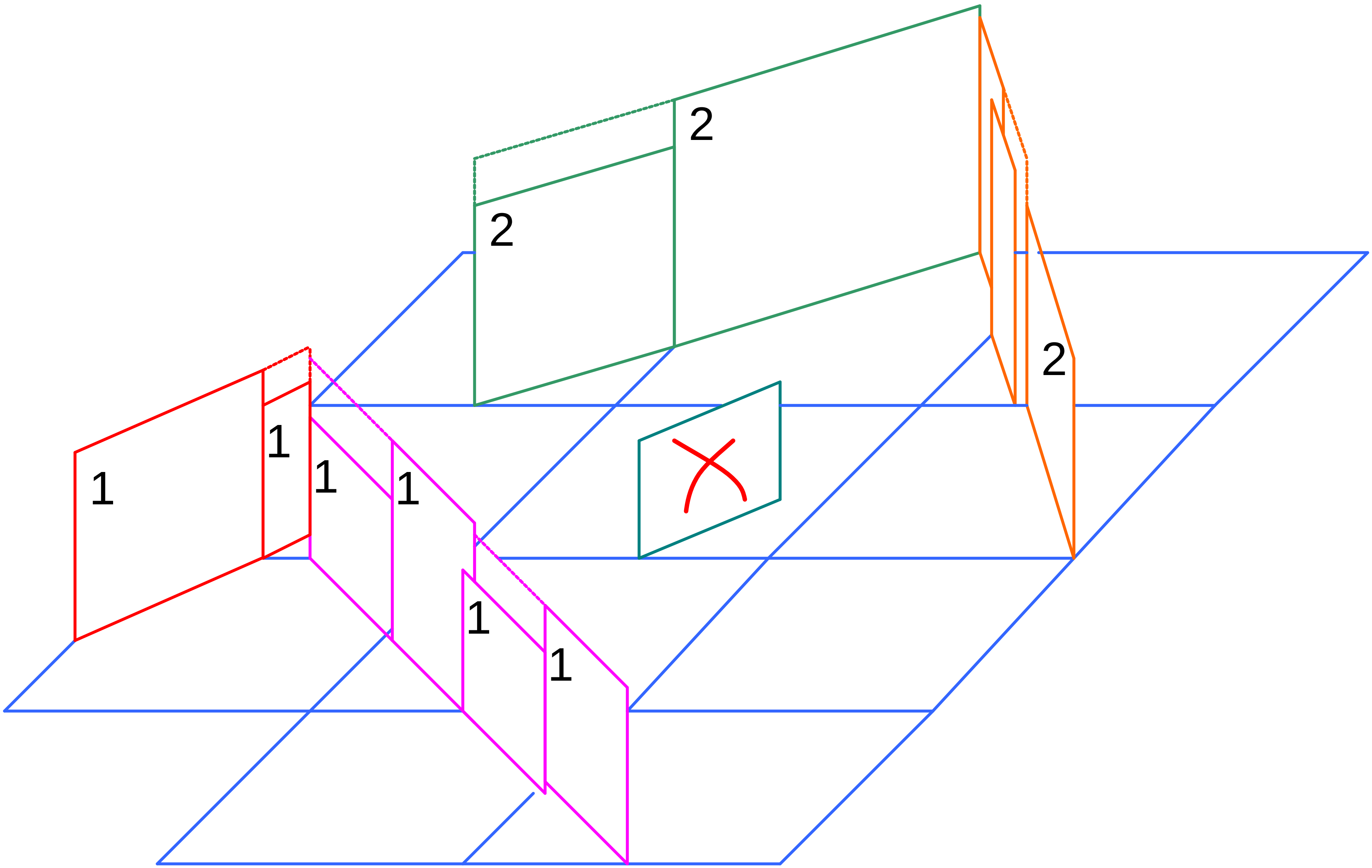
Project overview



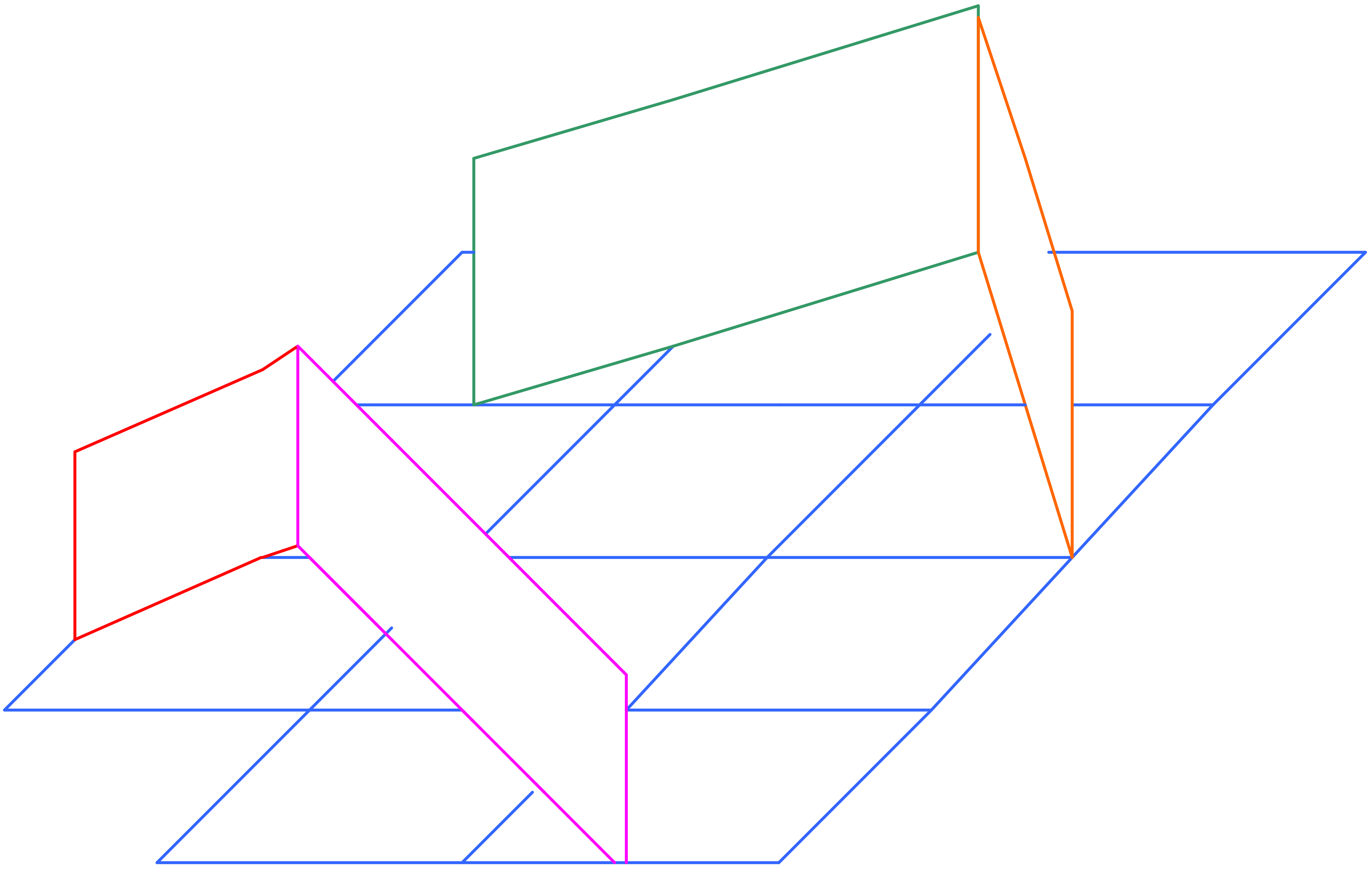
Project overview



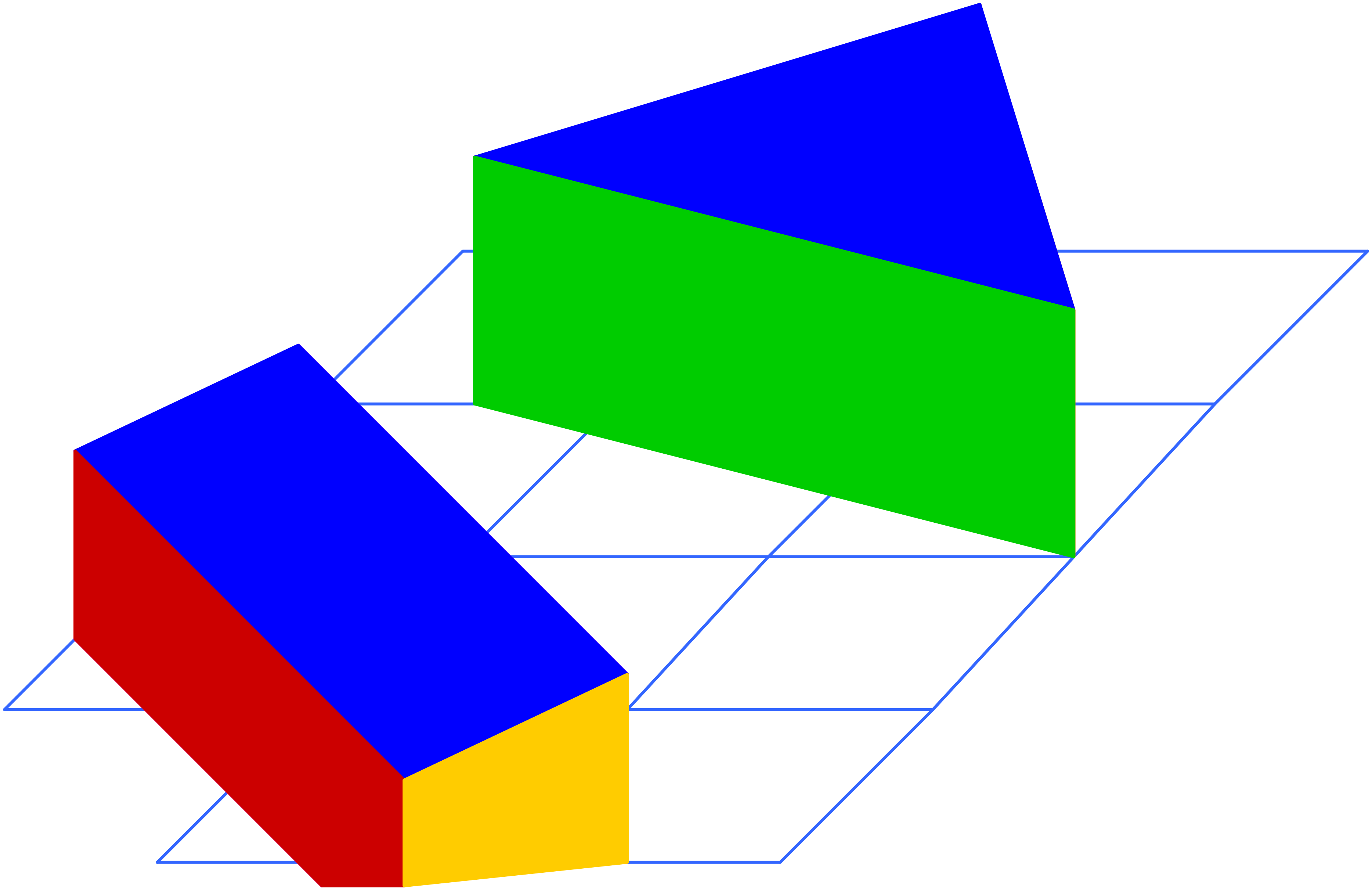
Project overview



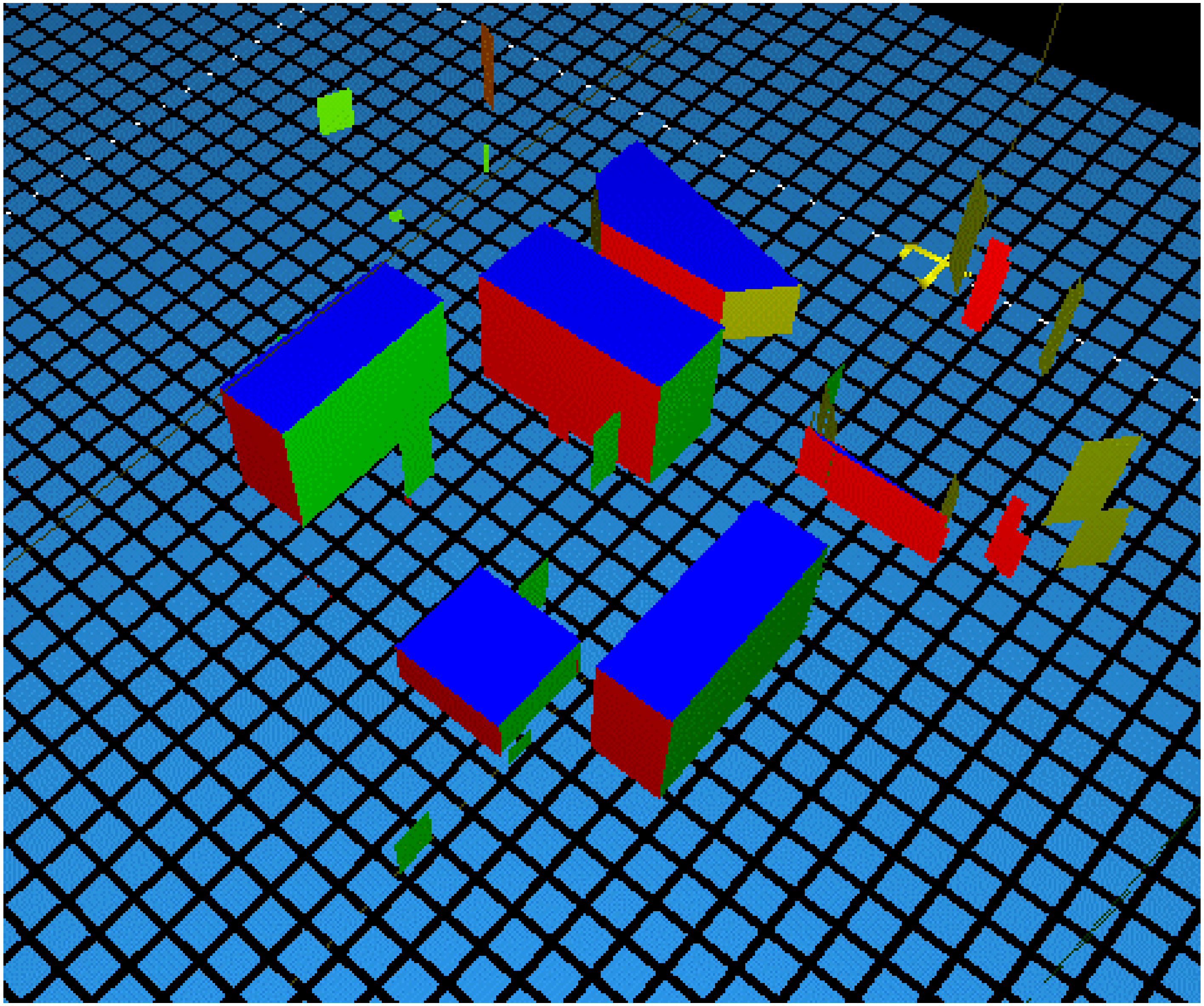
Project overview



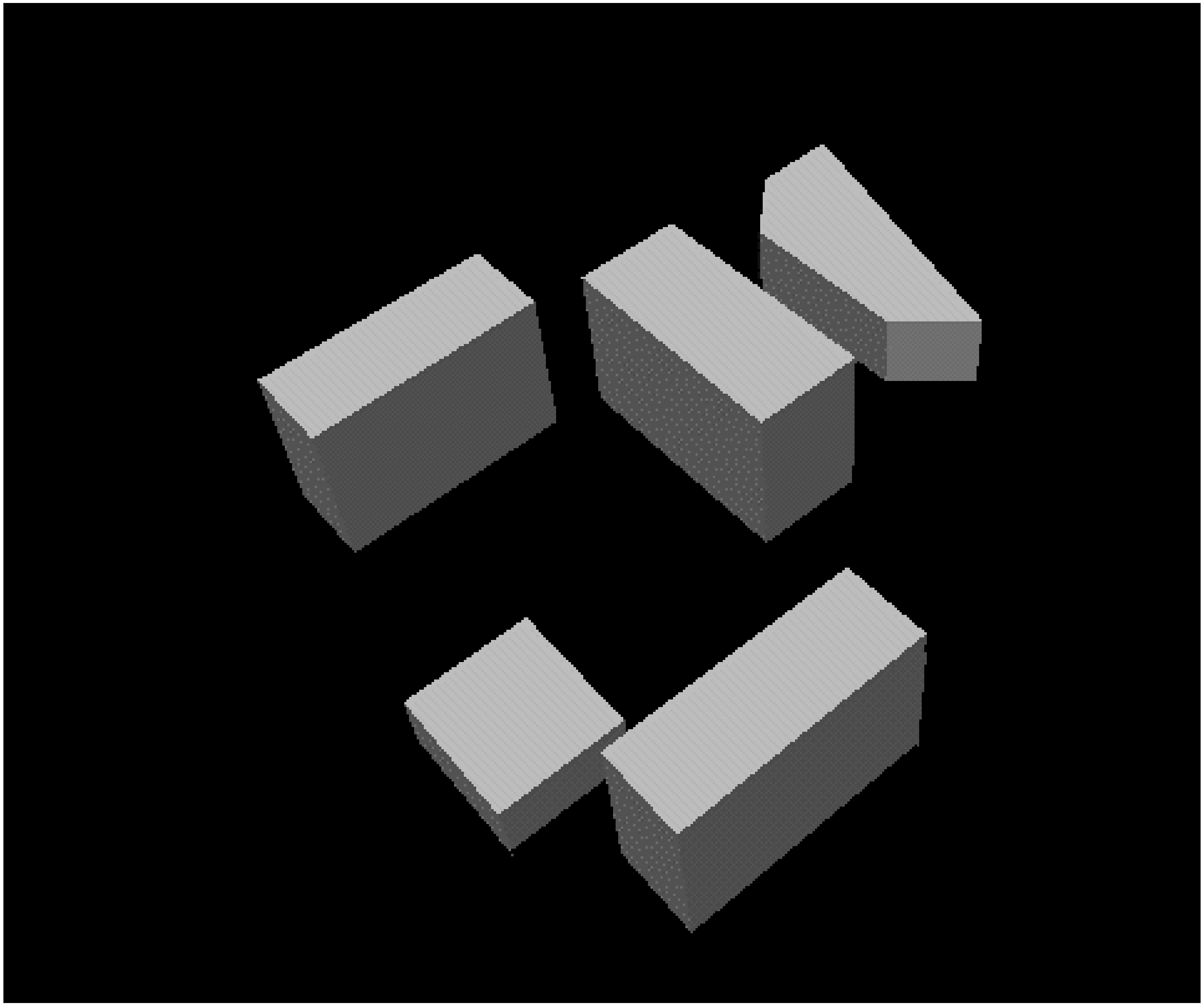
Project overview



Project overview



Project overview



Contents

1/ Project overview

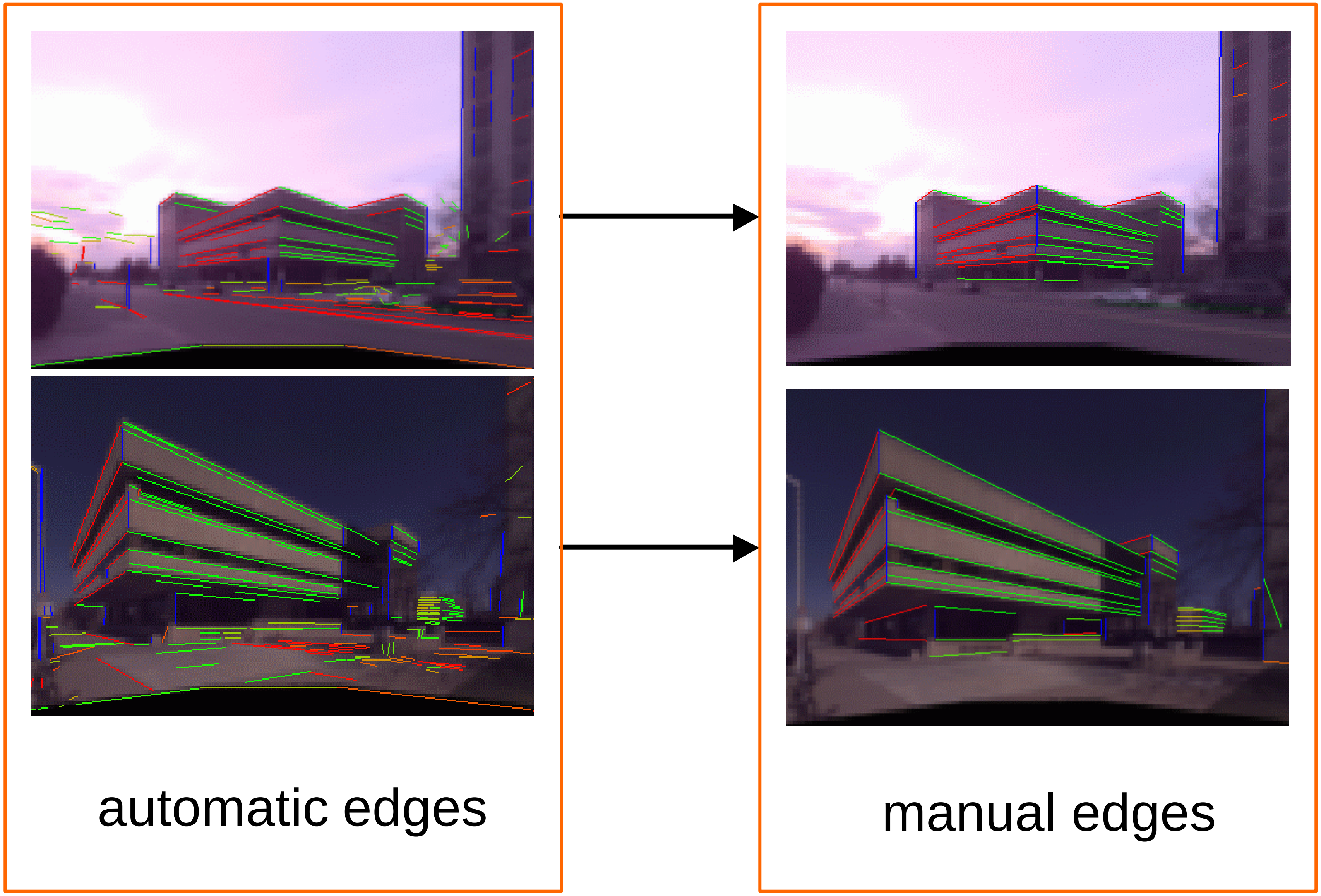
2/ code validation

3/ parallel algorithms

4/ a few ideas

Code validation

- edge detection improvement

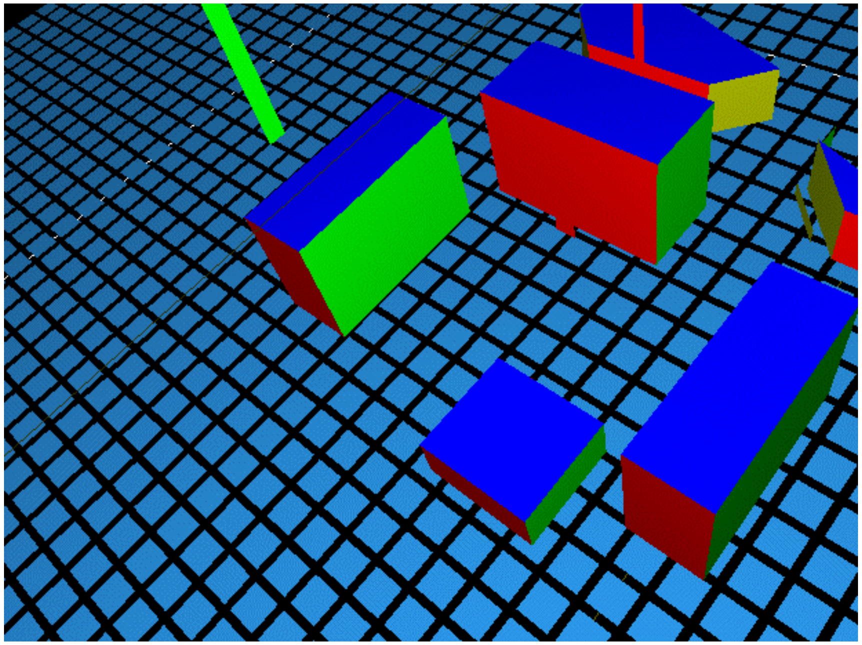


automatic edges

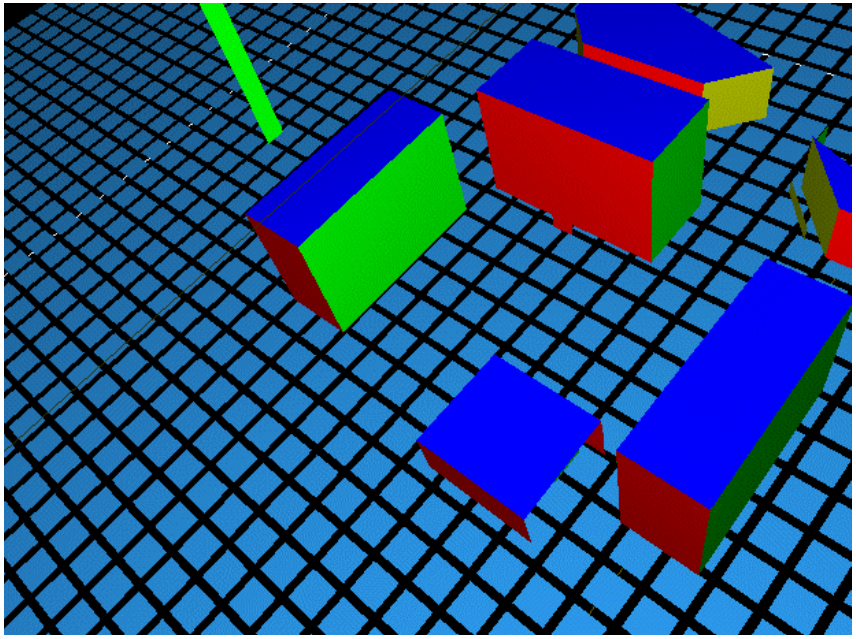
manual edges

Code validation

- edge detection improvement



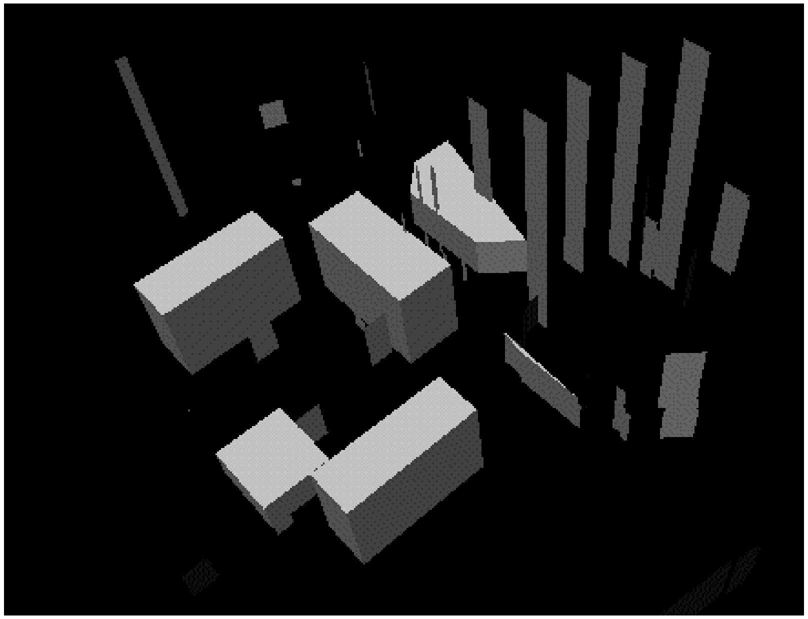
manual edges



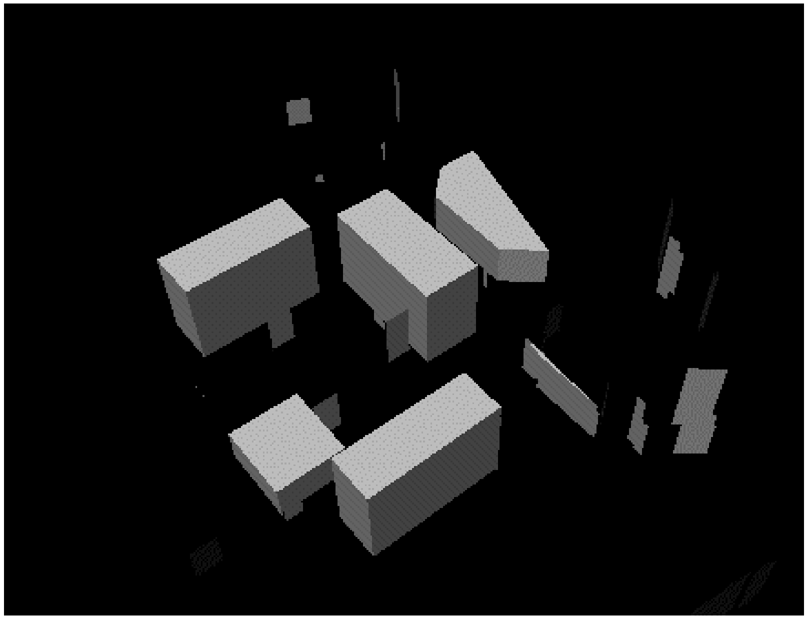
automatic edges

Code validation

- polygon filtering : remove polygons without support.



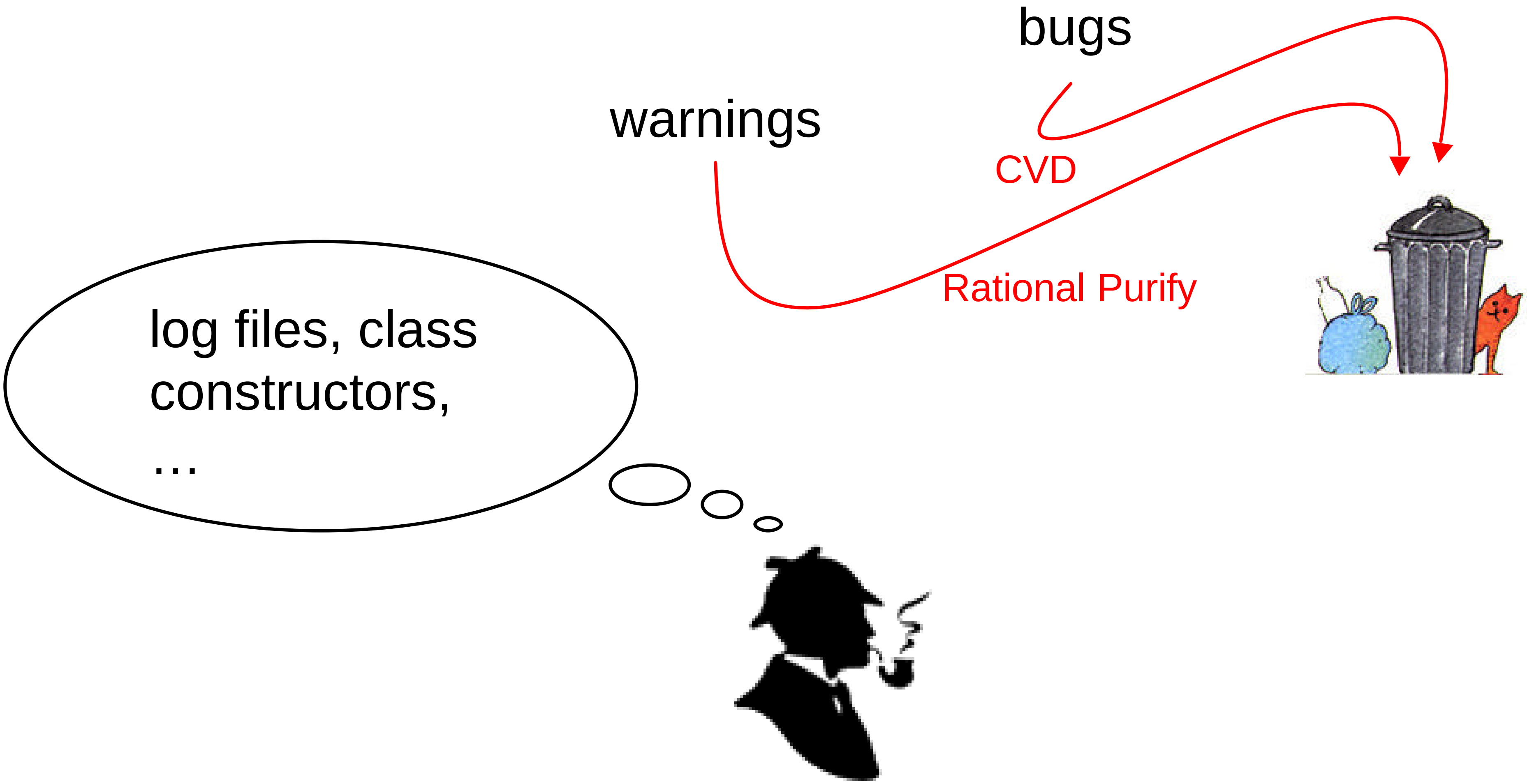
without filter



with filter

Code validation

- debugging



bugs

warnings

CVD

Rational Purify

log files, class constructors, …

Contents

1/ Project overview

2/ code validation

3/ parallel algorithms

4/ a few ideas

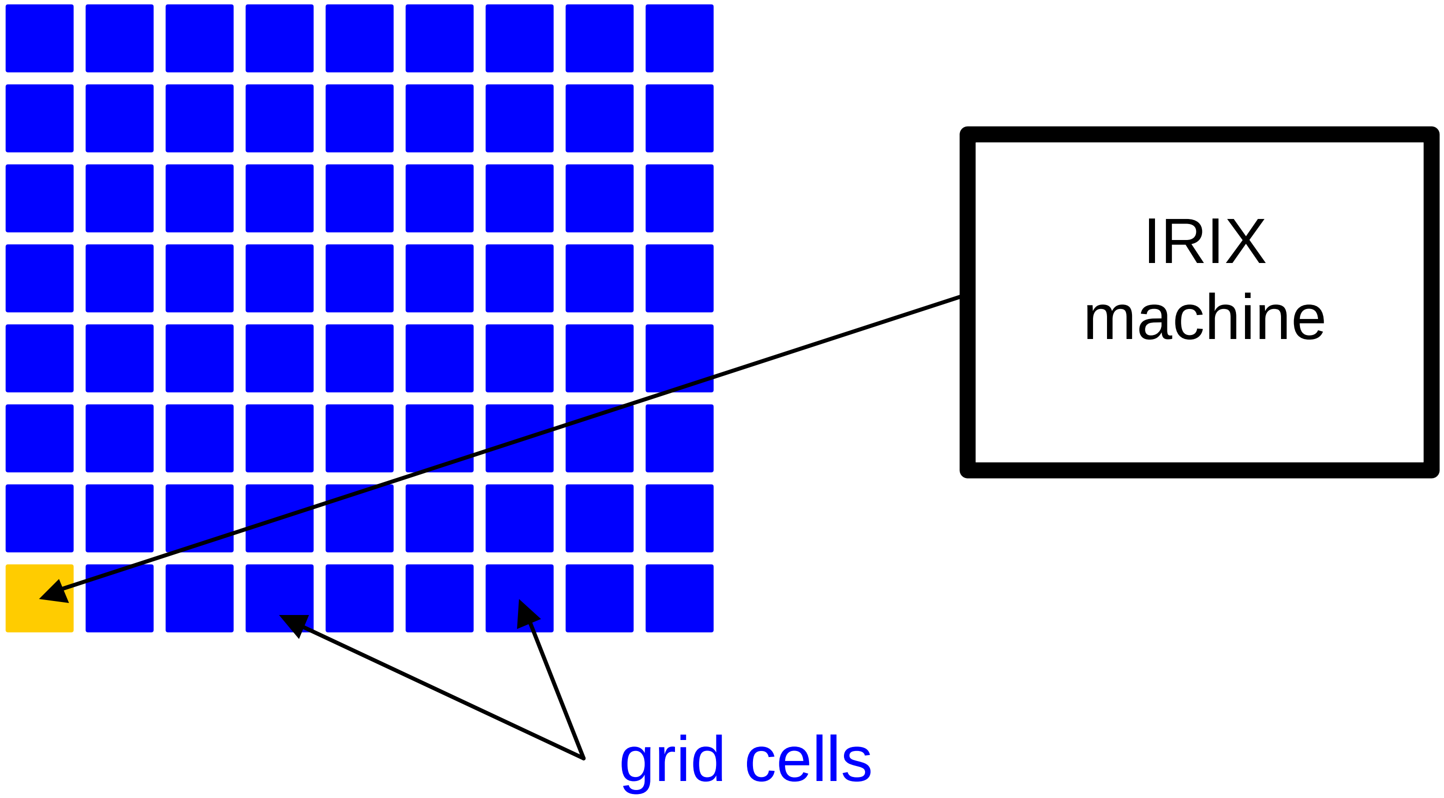
Parallel algorithms

- why parallelize ?

🟦 not processed

🟨 being processed

🟥 processed



IRIX machine

grid cells

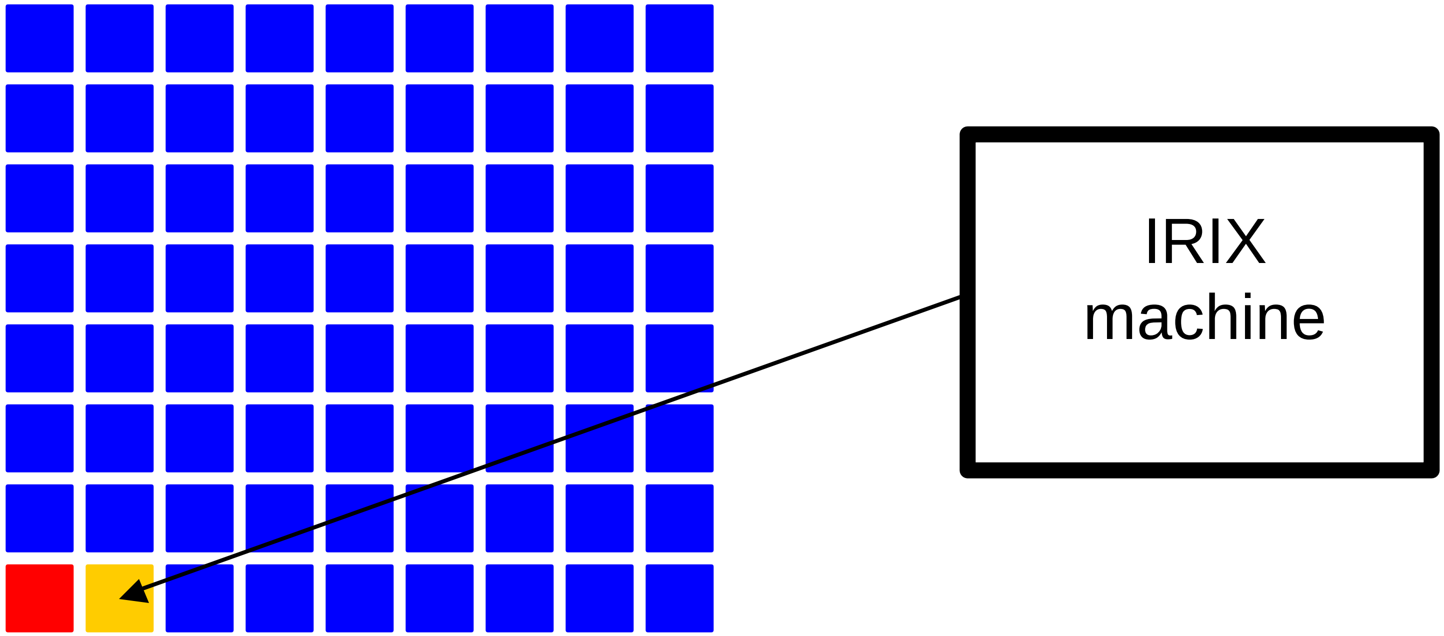
Parallel algorithms

- why parallelize ?

🟦 not processed

🟨 being processed

🟥 processed



IRIX
machine

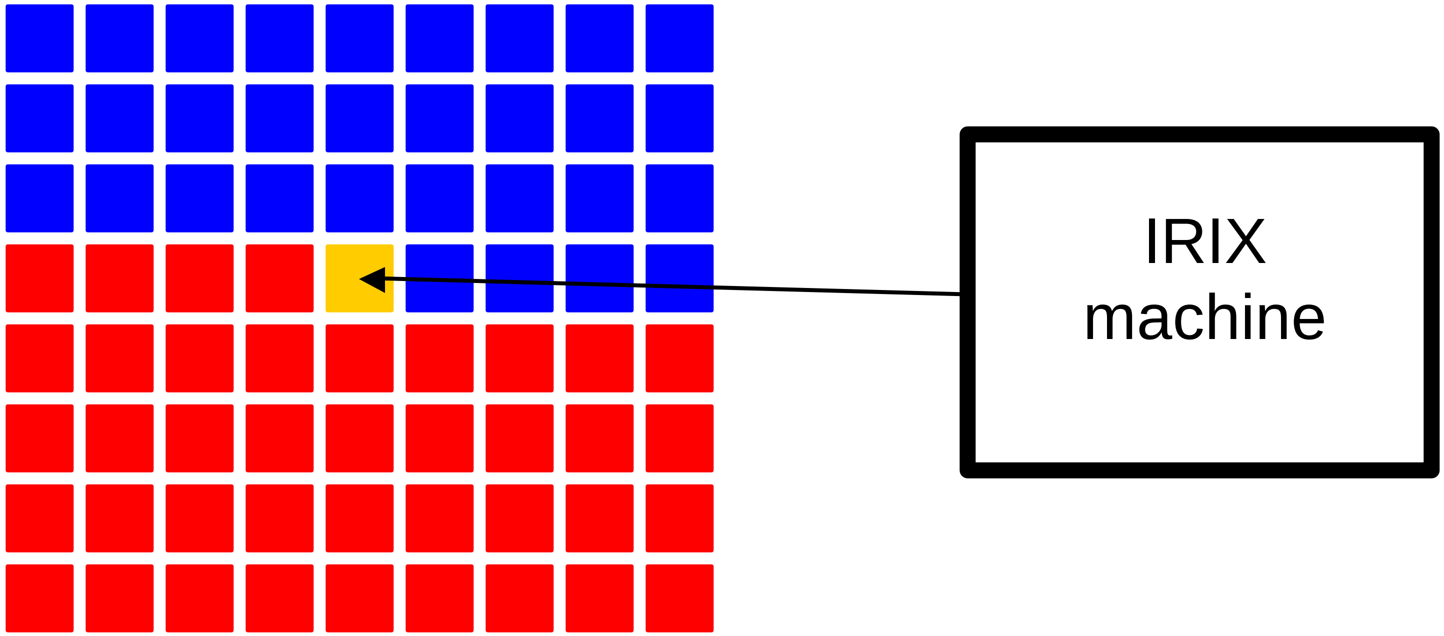
Parallel algorithms

- why parallelize ?

🟦 not processed

🟨 being processed

🟥 processed



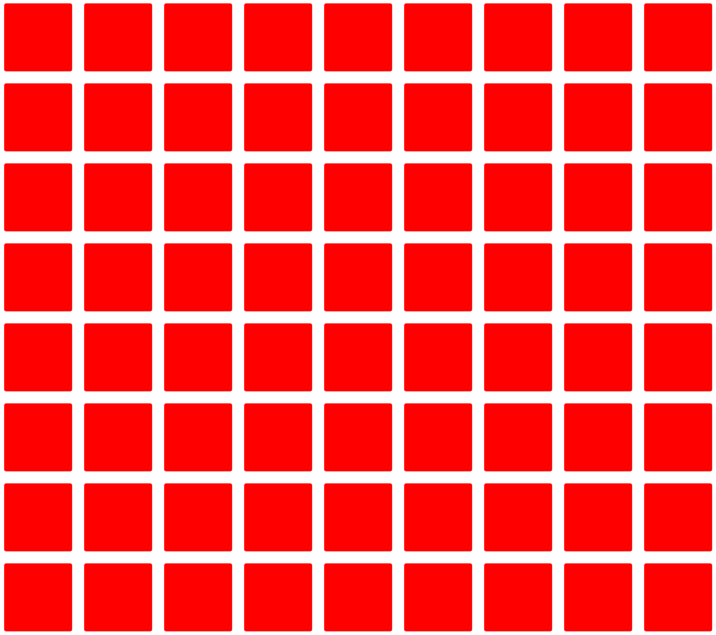
Parallel algorithms

- why parallelize ?

🟦 not processed

🟨 being processed

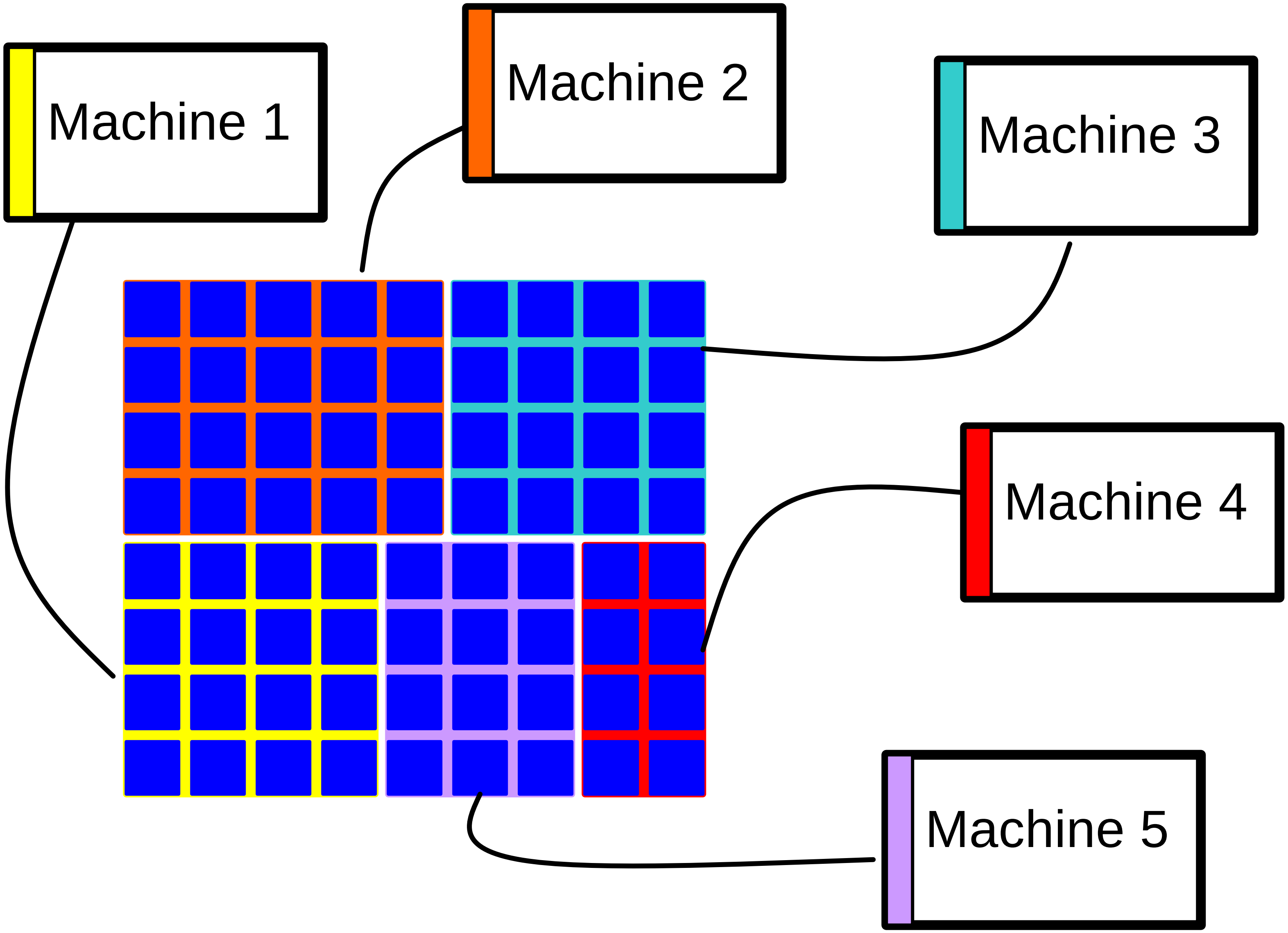
🟥 processed



= > 7 hours on IRIX single processor

Parallel algorithms

- first method



- first method

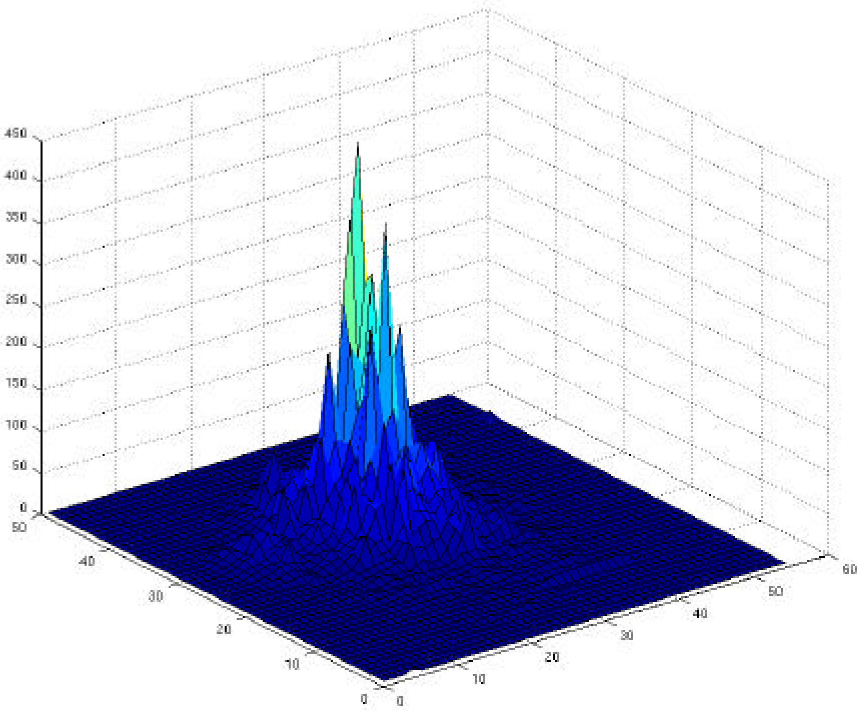
- CPU speed varies across machines
- time to process a cell is not constant

difficult optimization problem…

Static allocation

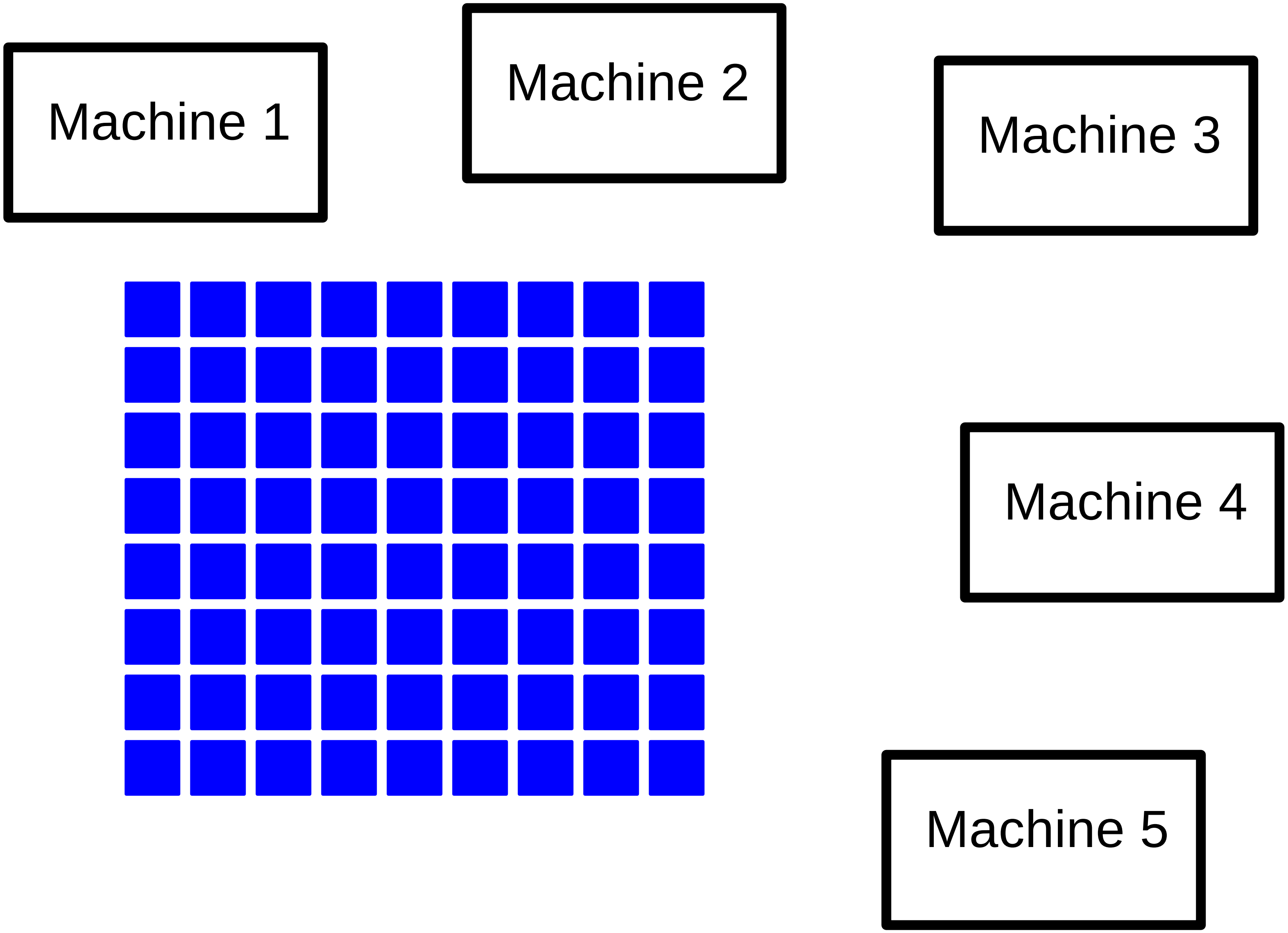
turpentine	2285 sec.
ray	2678
trace	2443
mosaic	2131
acetone	2711
hue	2342
panorama	1755
orange	1939

Computing time : 45 min.
Standard dev : 3.4 %



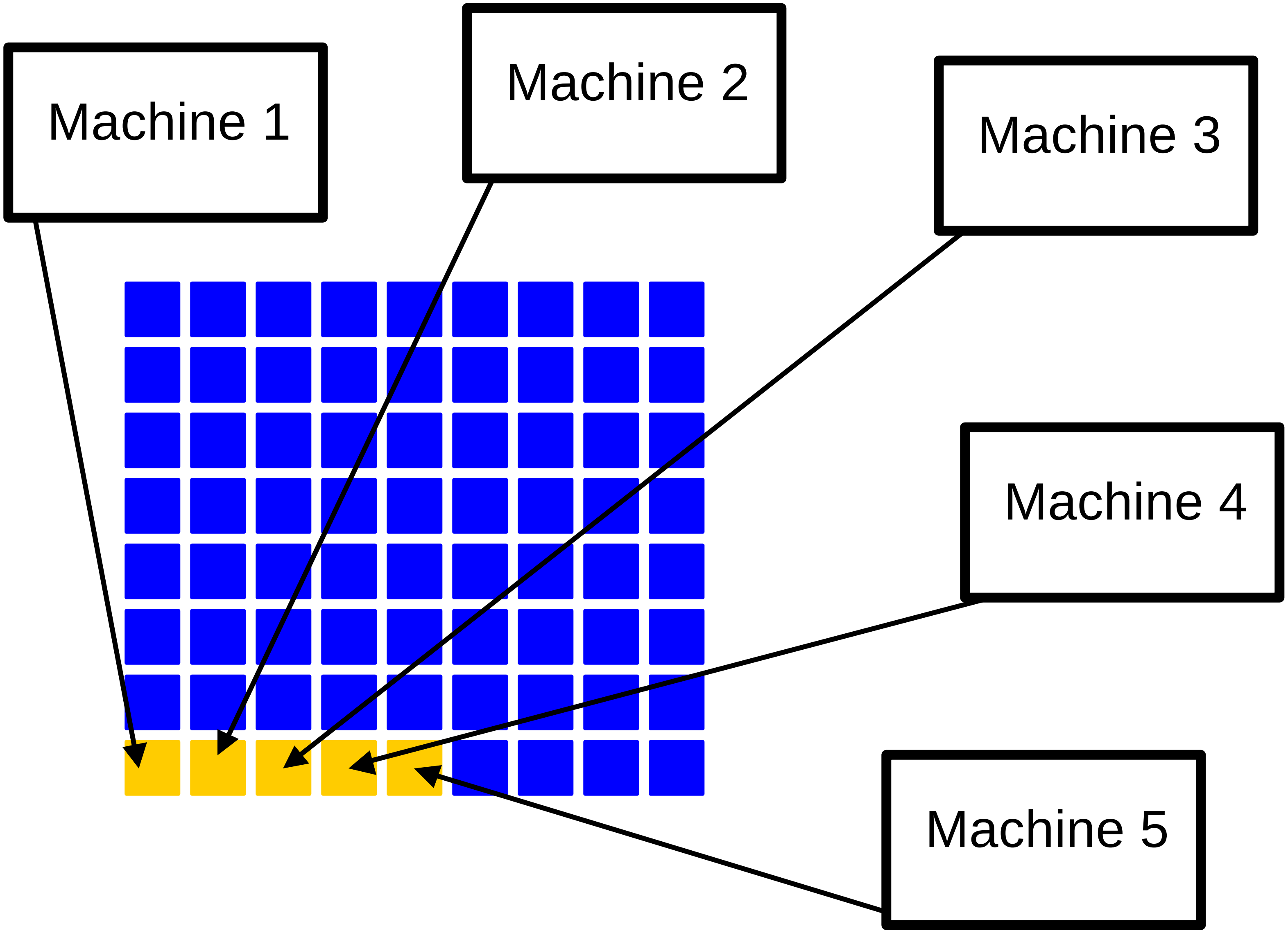
Parallel algorithms

- second method



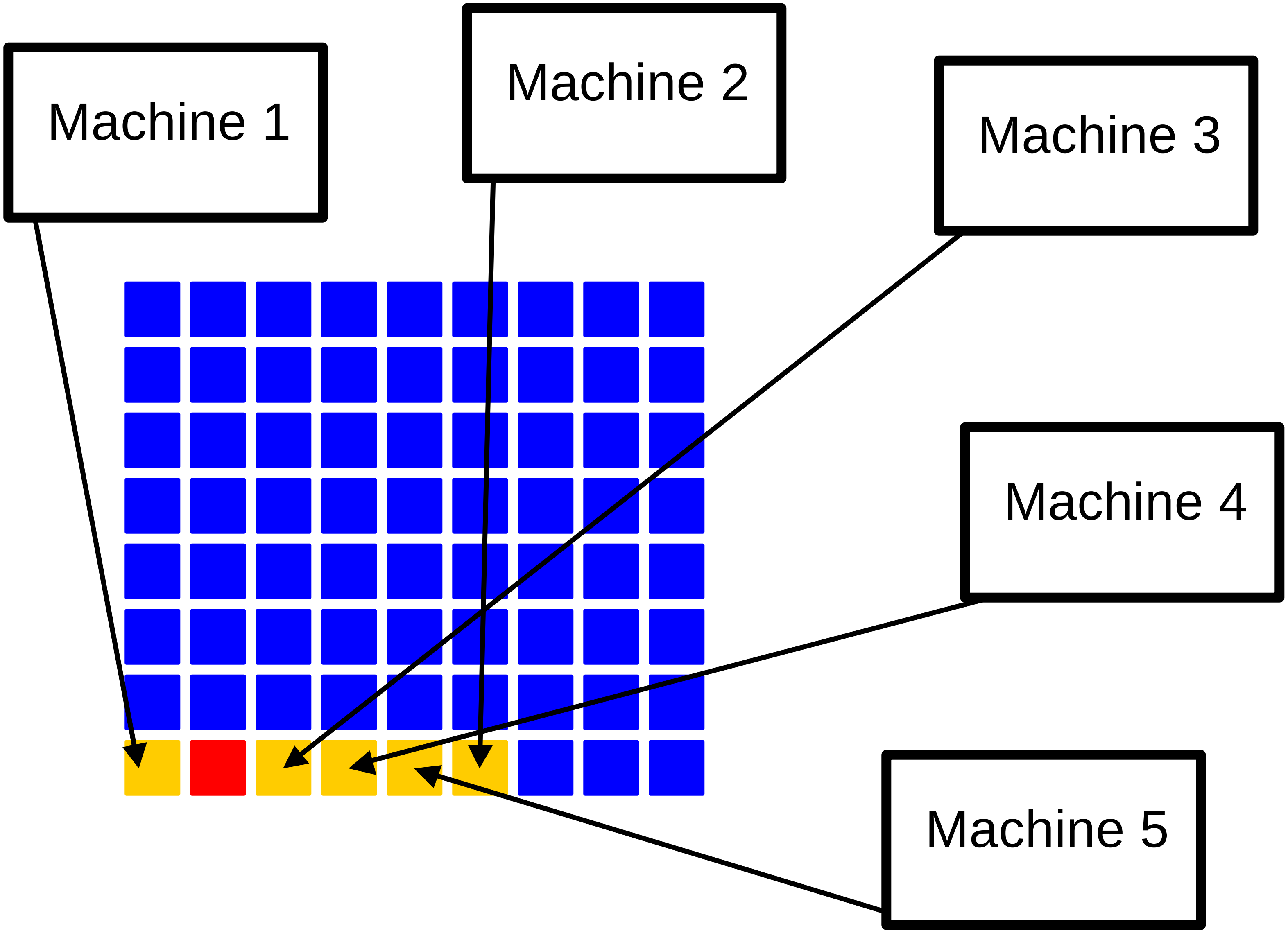
Parallel algorithms

- second method

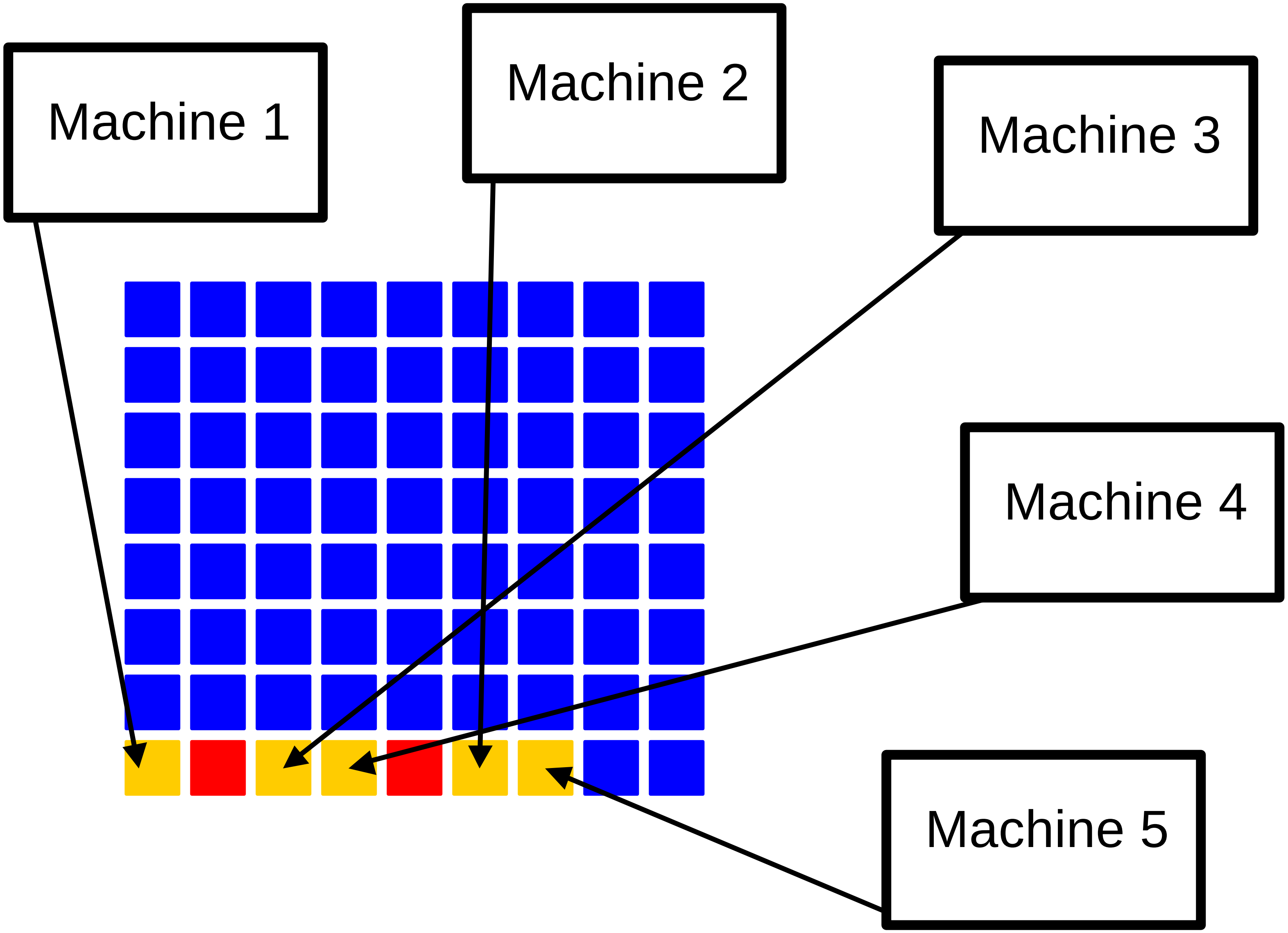


Parallel algorithms

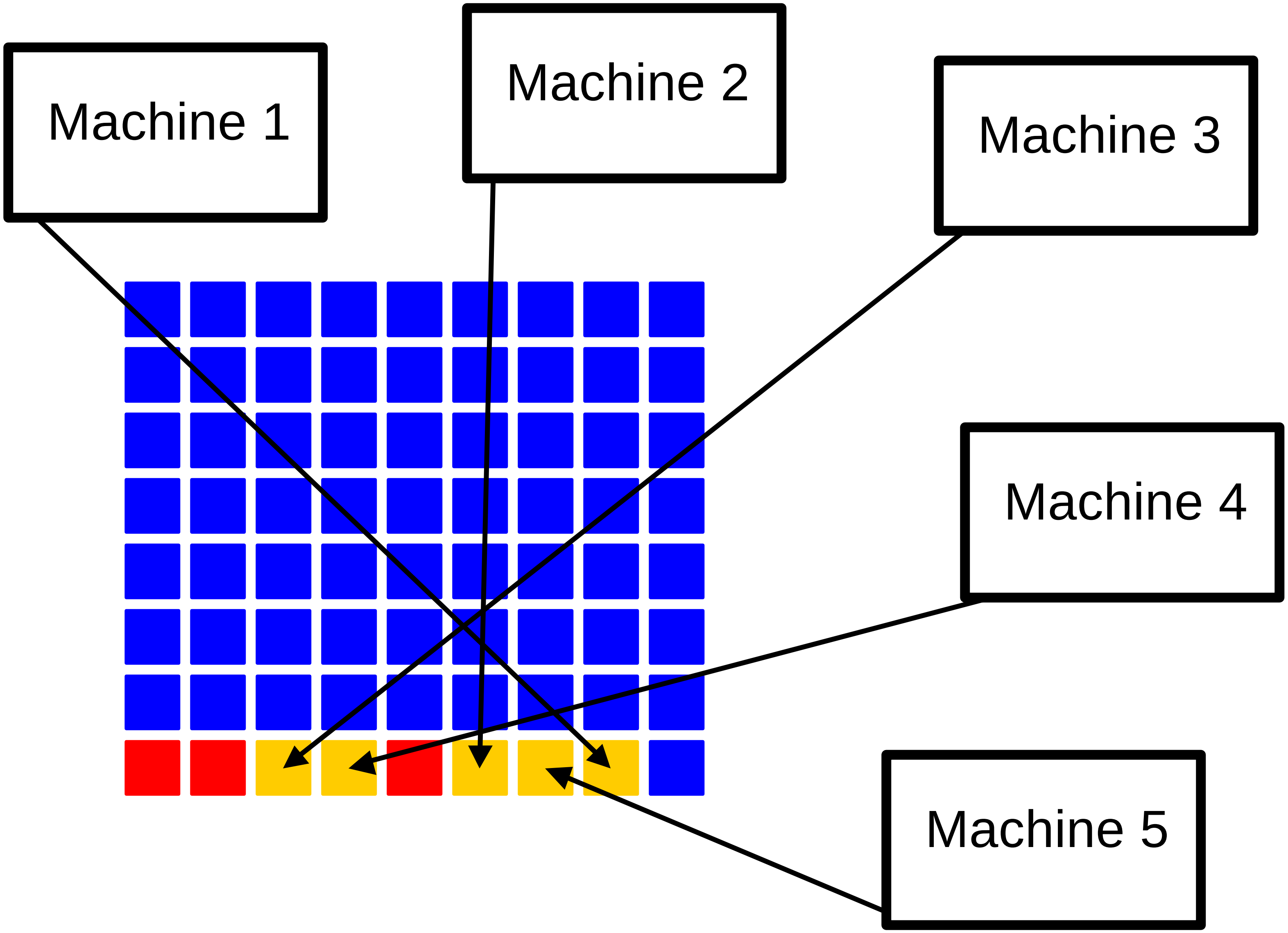
- second method



- second method

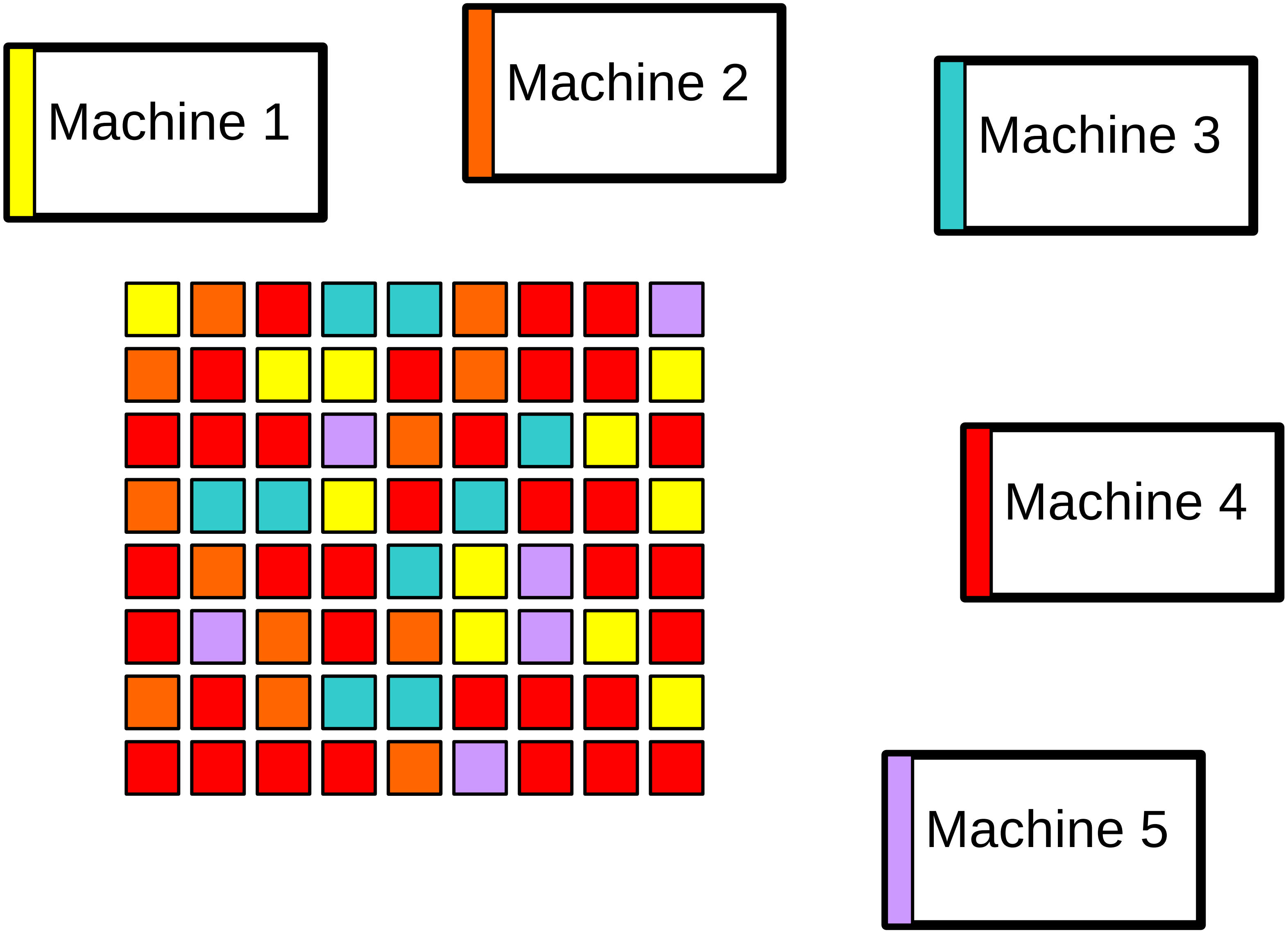


- second method



Parallel algorithms

- second method



Machine 1

Machine 2

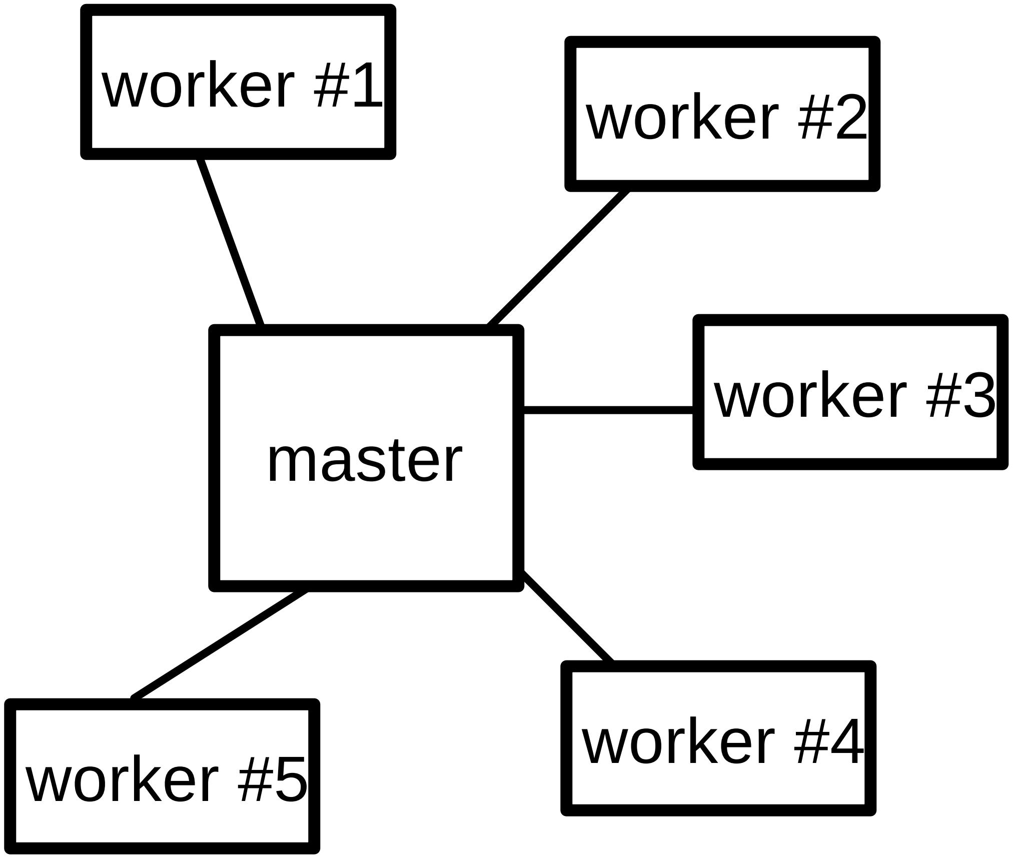
Machine 3

Machine 4

Machine 5

- second method

- master-worker model (MPI)



worker #1

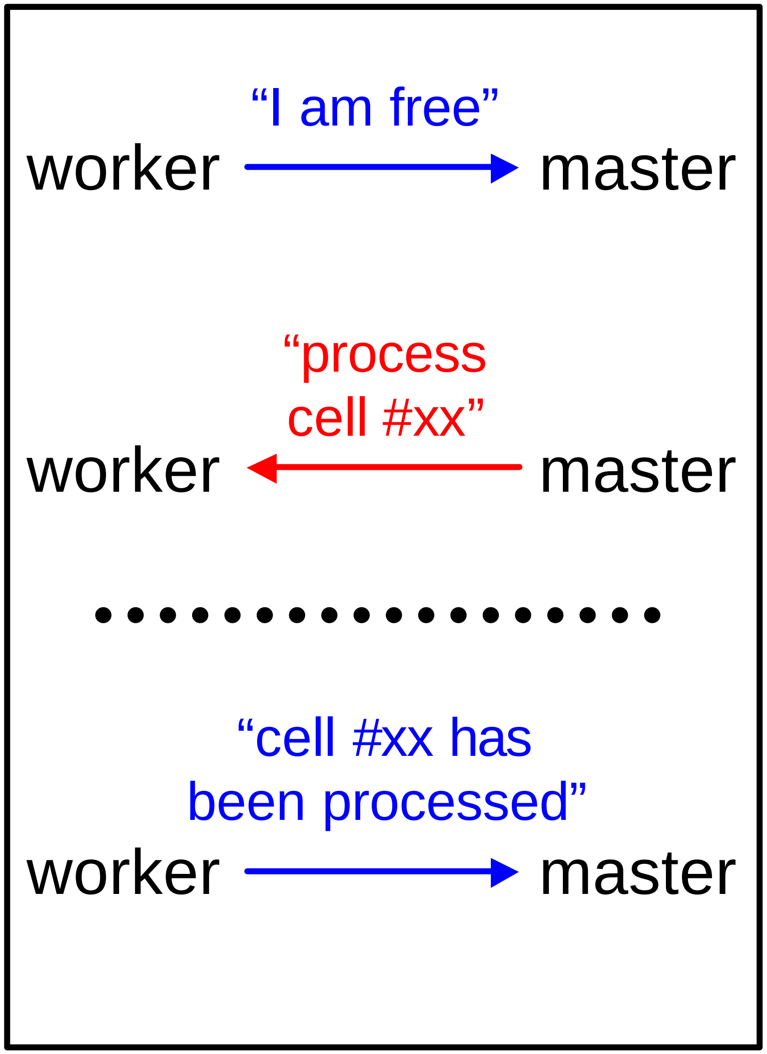
worker #2

master

worker #3

worker #4

worker #5



“I am free”

worker → master

“process cell #xx”

worker ← master

……………………

“cell #xx has been processed”

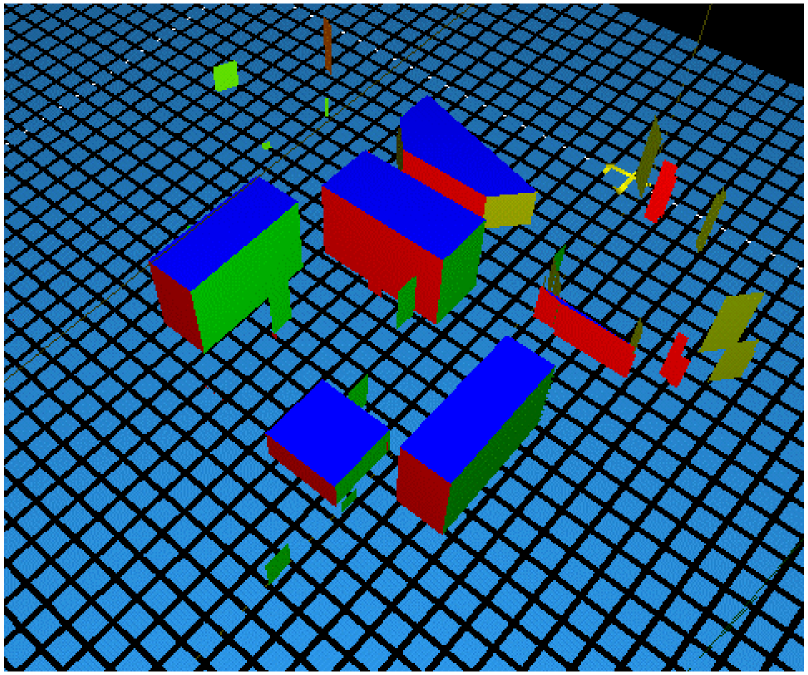
worker → master

Communication scheme

Parallel algorithms

- second method

panorama	1246.3	(sec.)
	1246.3	
	1248.5	
	1248.6	
	1250.0	
	1240.6	
turpentine	1250.3	
	1247.0	
	1243.0	
	1233.8	
acetone	1249.1	
	1239.1	
	1250.5	
	1238.3	
mosaic	1255.6	
	1278.9	
trace	1276.0	
	1275.9	
orange	1277.2	
hue	1268.8	
ray	1230.4	



• Computation time	21mn
• Communication time	3mn
• Idle time	6mn

Standard deviation : 2.6 %

Parallel algorithms

	Time	Speed-up
Method 1 (rectangles)	**45 min.**	10.0
Method 2 (queue)	**30 min.**	15.2

- method 2 is faster and more flexible.
- last step of reconstruction still needs to be parallelized

  (takes ~10 minutes on a single processor IRIX)

Contents

1/ Project overview

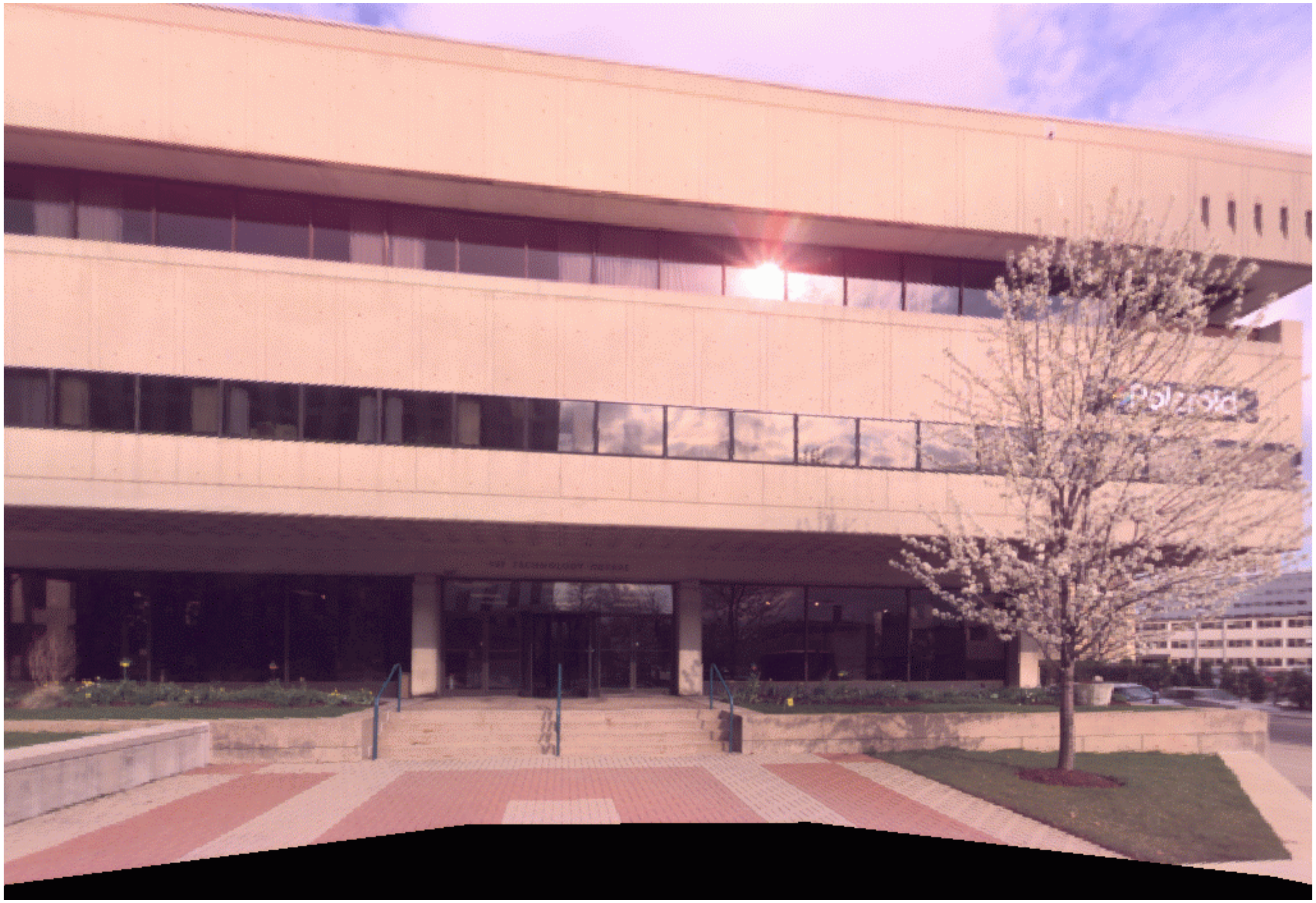
2/ code validation

3/ parallel algorithms

4/ a few ideas

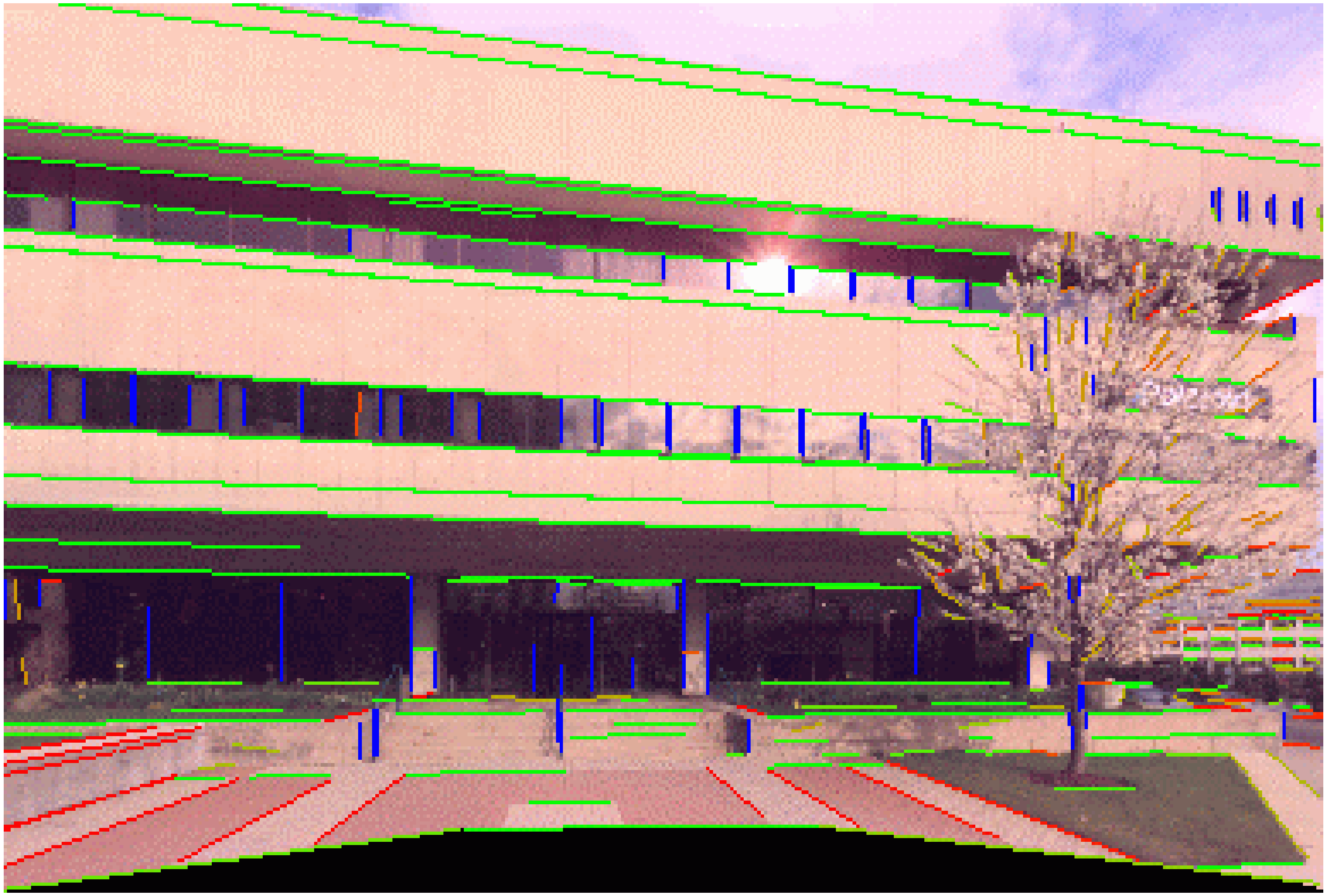
A few ideas

- high-frequency edge filtering



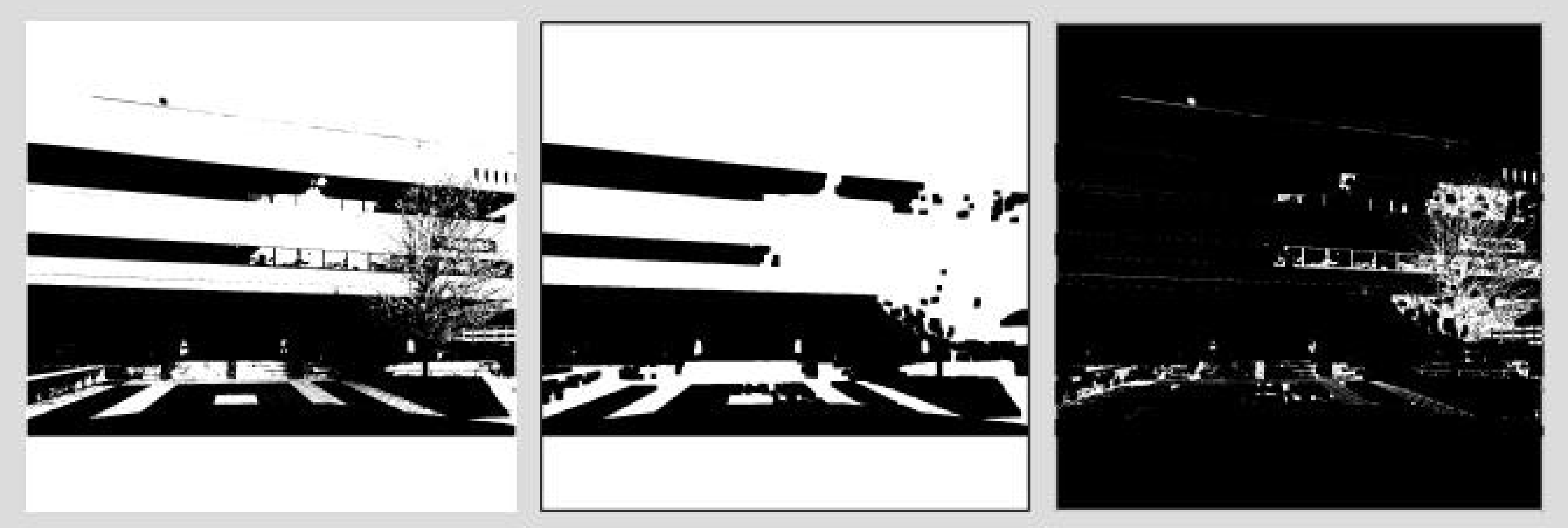
A few ideas

- high-frequency edge filtering



A few ideas

- high-frequency edge filtering



boolean image

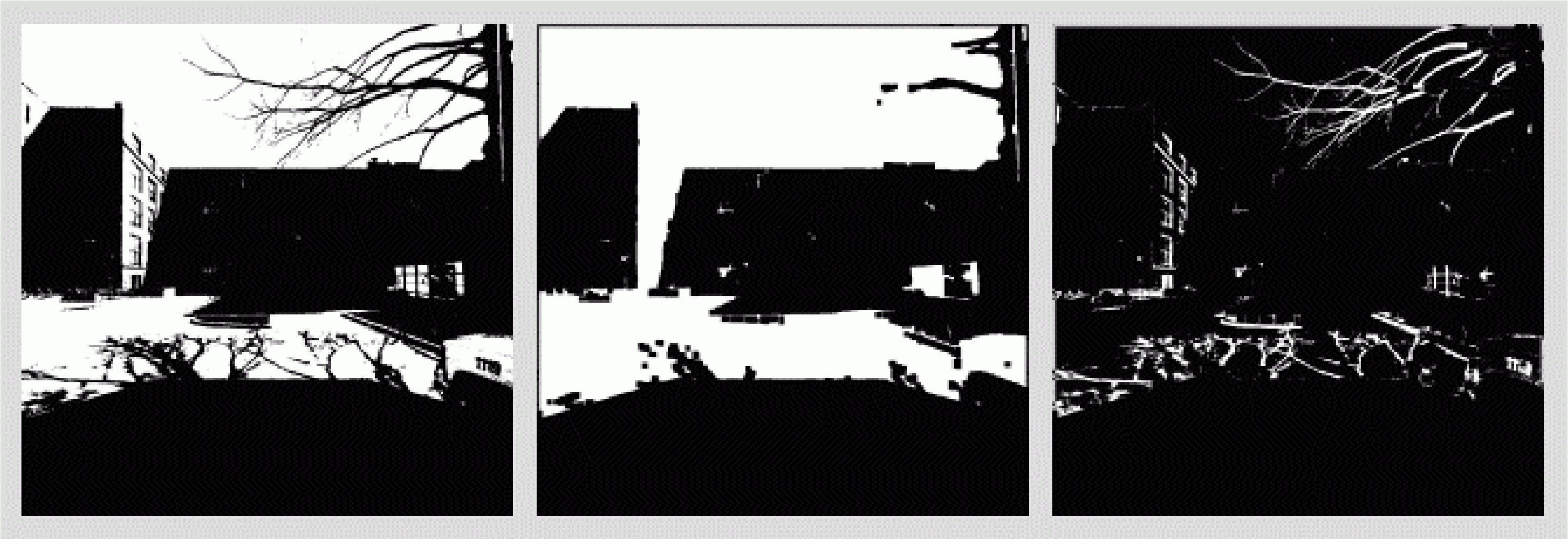
high-frequency detection

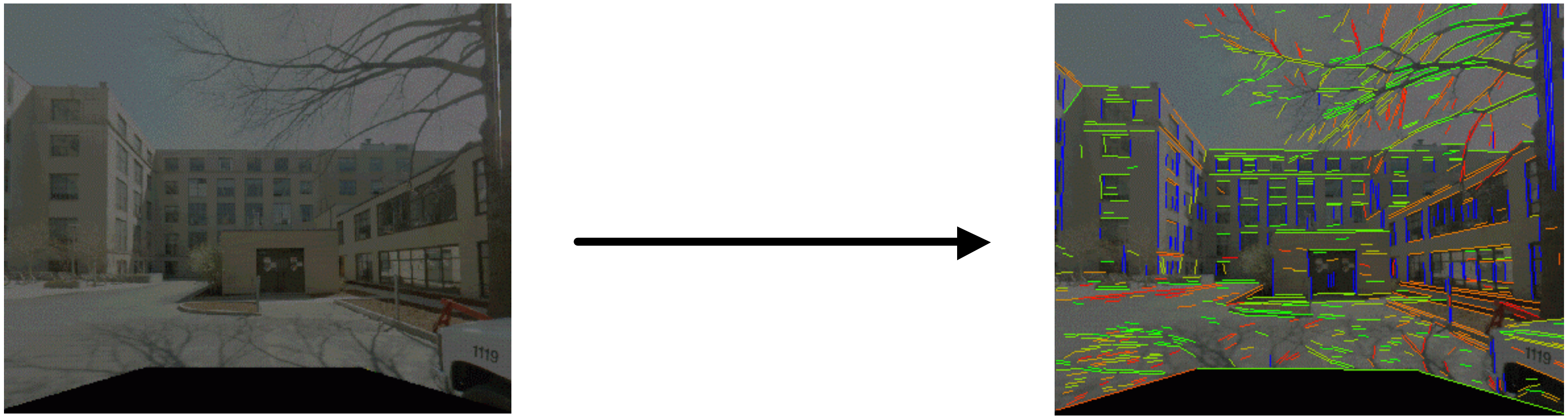
mask

- on courtesy of Alexis Charignon

A few ideas

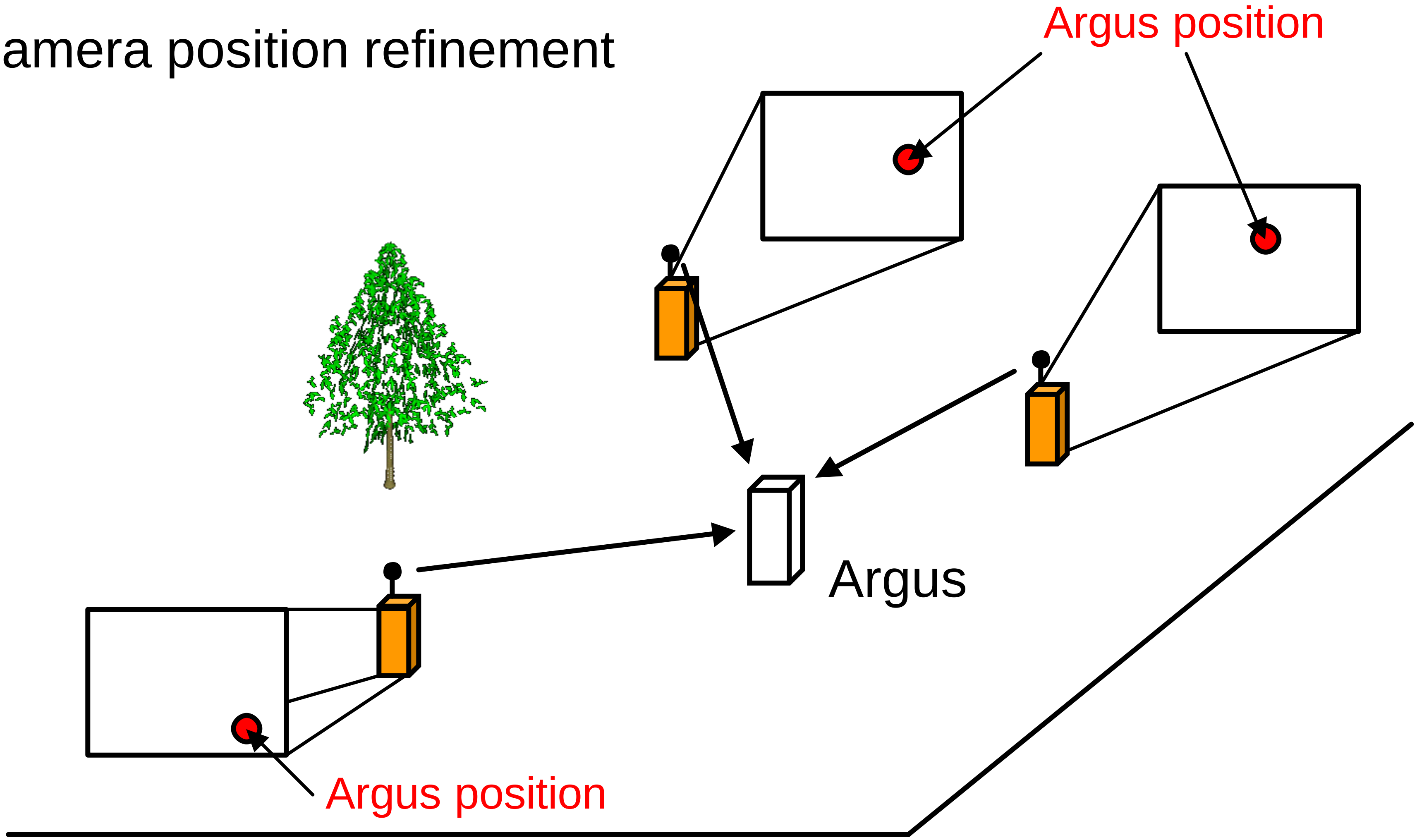
- high-frequency edge filtering





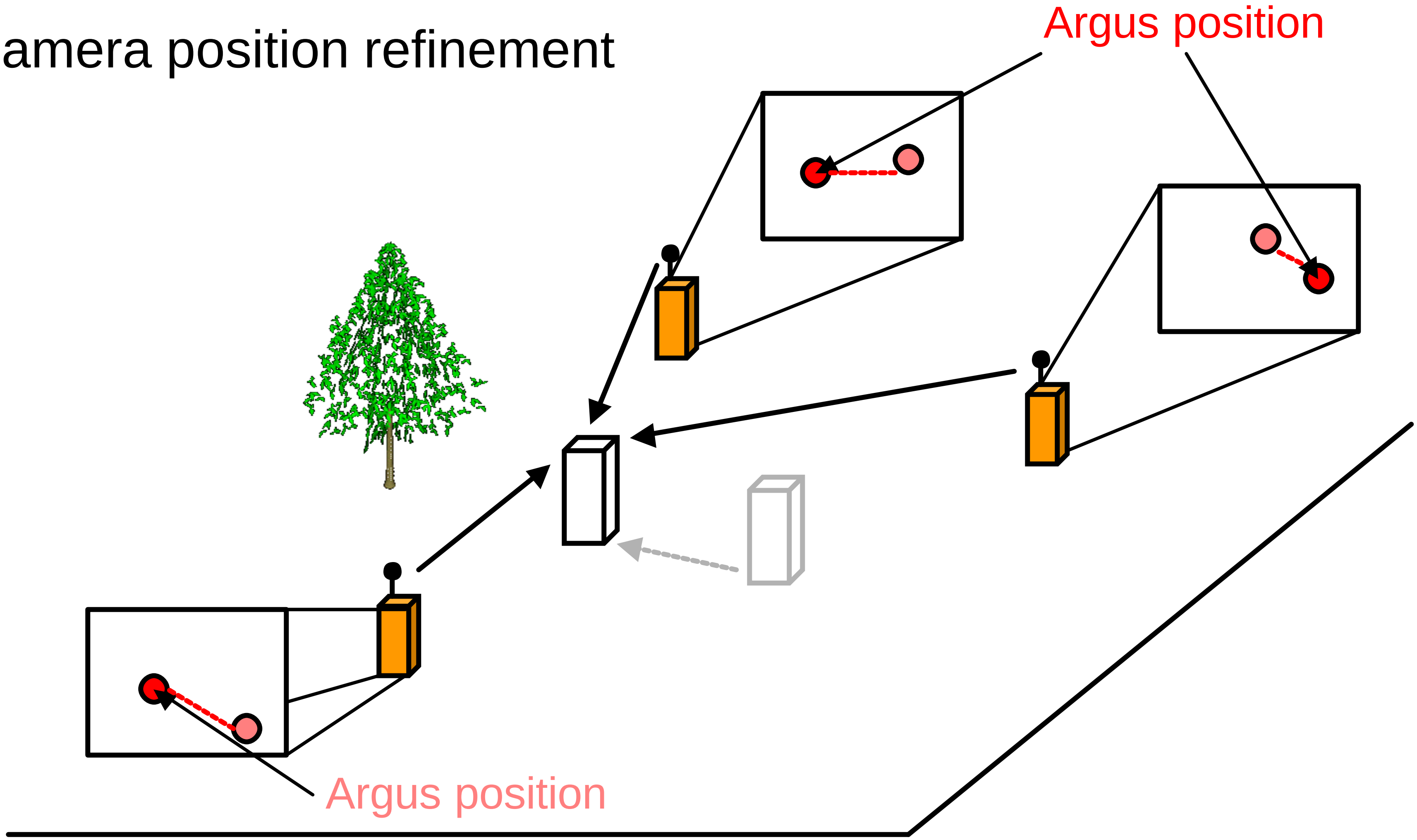
A few ideas

- camera position refinement



A few ideas

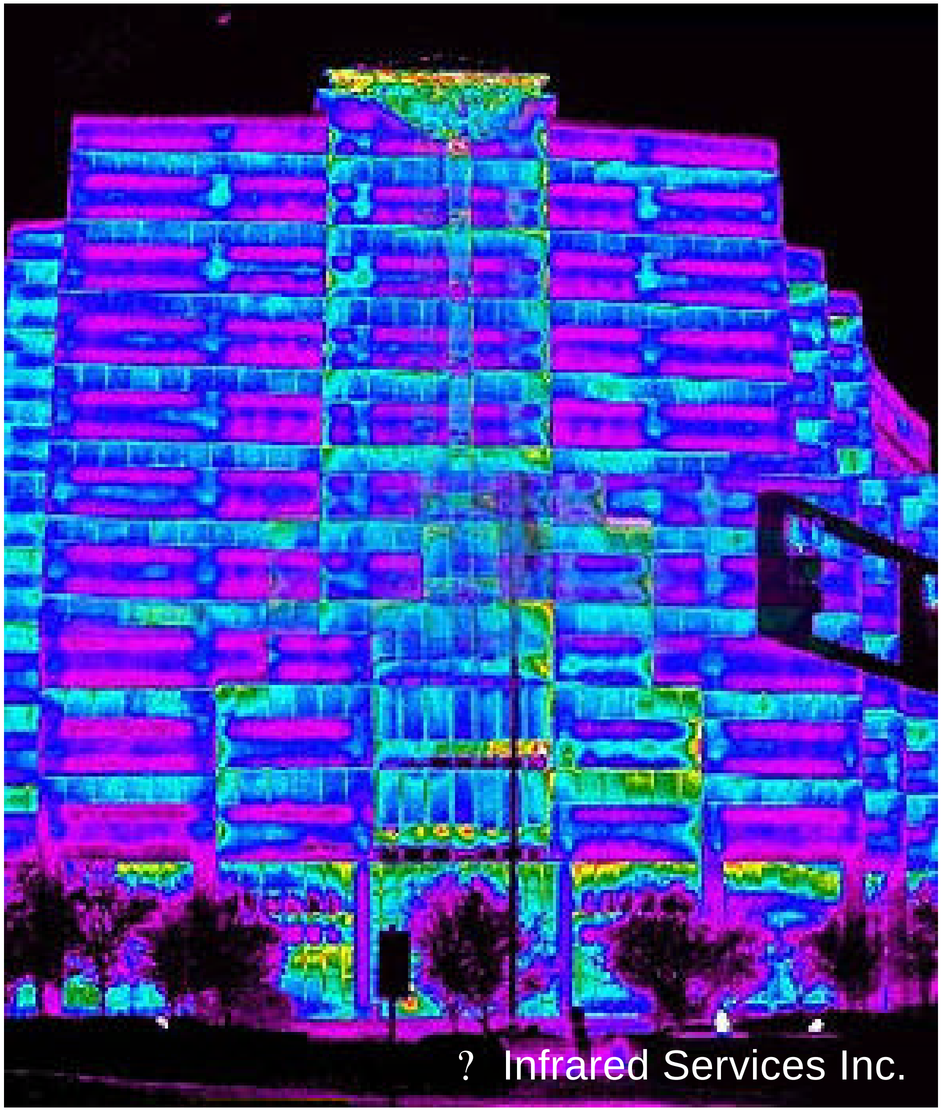
- camera position refinement



- orange robots keep track of relative Argus motion.

A few ideas

- using alternative sensors



? Infrared Services Inc.

IR imagery

Conclusion

Conclusion

- reconstruction code validated on Tech Square
- most CPU-intensive program parallelized
- speed-up = 15 on IRIX cluster

Next steps

- refining camera positions
- porting the code to Linux

Acknowledgements

- Seth Teller
- Manish Jethwa, Neel Master
- Alexis, Yann, Stefane, Adnan
- MIT Graphics Group

The End

Thanks…

Questions ?