Master in Computer Vision *Barcelona*

Module 4: 3D Vision

Project: 3D recovery of urban scenes

Group 7: Josep Brugués i Pujolràs / Sergi García Sarroca
Òscar Lorente Corominas / Ian Riera Smolinska

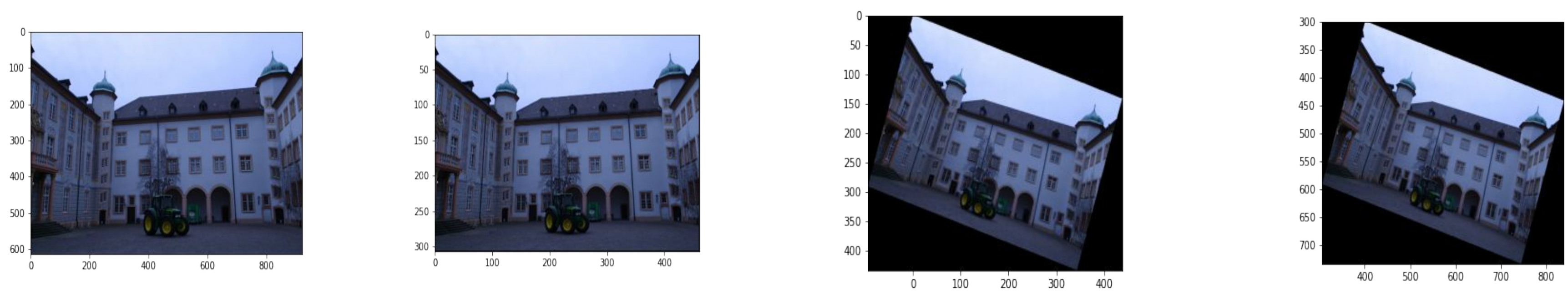
Contents:

Lab 1: Core Function

Inputs: *Image, Homography, Corners.*
Outputs: *I_rectified, rectified _image_axis, rectified_image_corners.*

1. Compute the corners of the input image.
2. Create mesh of coordinates.
3. Inverse of the H multiplied by the mesh coordinates ---> Positions on the image.
4. Map pixels with interpolation.

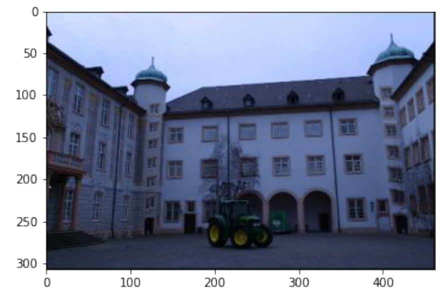
I.e:



Lab 1: Similarity Transformation

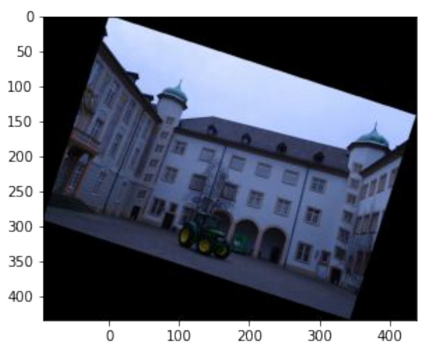
$$H_s = \begin{bmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} = \begin{bmatrix} scos(\theta) & -ssin(\theta) & tx \\ ssin(\theta) & scos(\theta) & ty \\ 0 & 0 & 1 \end{bmatrix}$$

Original



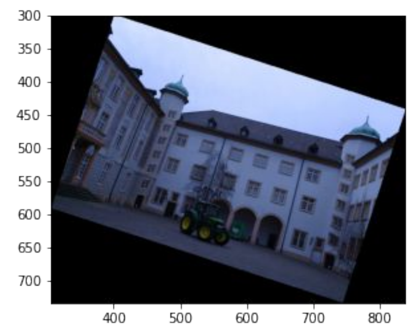
s = 0.5

Scaled



s = 0.5
θ = 0.1∗π

Scaled and Rotated



s = 0.5
θ = 0.1∗π
tx = 400
ty = 300

Scaled, Translated and Rotated

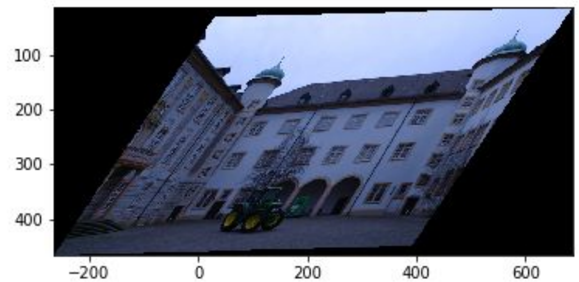
Lab 1: Affinities

$$H_a = \begin{bmatrix} A & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix}$$

$$A = R(\theta)R(-\phi)DR(\phi)$$

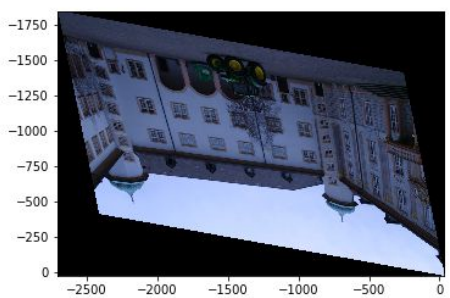
$$A = \mathrm{UDV^T} = \left(\mathrm{UV^T}\right)\left(\mathrm{VDV^T}\right)$$

Original



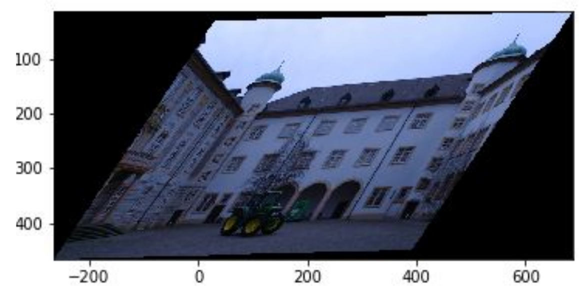
Affine Transformation 1

$$s = [1, 0.5]$$
$$\theta = 0.1 * \pi$$
$$\phi = 0.3 * \pi$$
$$(tx, ty) = (30, 30)$$



Affine Transformation 2

$$s = [3, 2]$$
$$\theta = \pi$$
$$\phi = -1.2 * \pi$$
$$(tx, ty) = (30, 30)$$

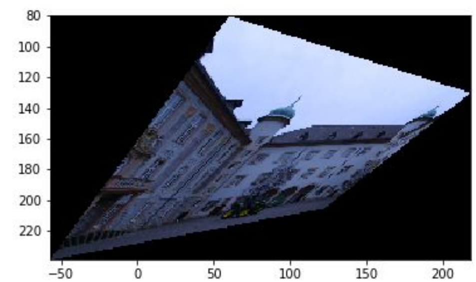


Affine Transformation 1 - SVD

Difference of 3.05×10^{-32}

Lab 1: Projective Transformations

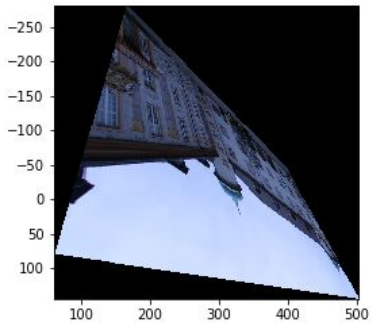
$$H_p = \begin{bmatrix} A & \vec{t} \\ \vec{v}^T & v \end{bmatrix}$$



$$A = \begin{bmatrix} 0.5 & -0.25 \\ -0.25 & 0.5 \end{bmatrix}$$

$$\vec{v} = [0.0015, 0.001]$$

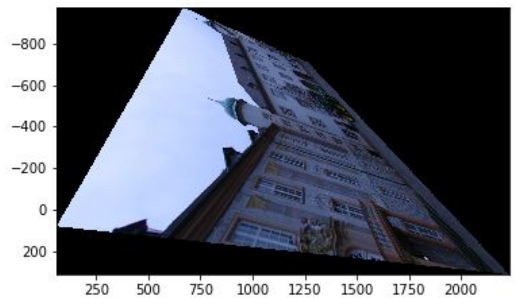
$$(tx, ty) = (60, 80)$$



$$A = \begin{bmatrix} 3 & 1 \\ 0.8 & -2 \end{bmatrix}$$

$$\vec{v} = [0.005, 0.005]$$

$$(tx, ty) = (60, 80)$$



$$A = \begin{bmatrix} 4 & 8 \\ -6 & 1 \end{bmatrix}$$

$$\vec{v} = [0.005, 0.002]$$

$$(tx, ty) = (60, 80)$$

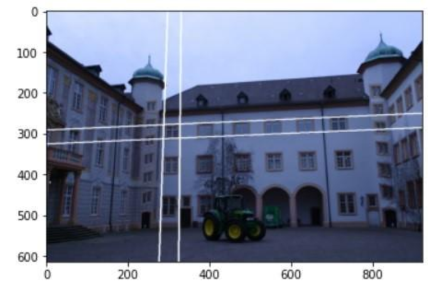
Lab 1: Affine Rectification

$$H_{a\leftarrow p} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l1 & l2 & l3 \end{bmatrix}$$

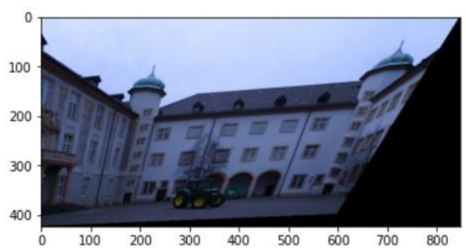
$$v_1 = l1 \times l2 \qquad v_2 = l3 \times l4$$

$$L_\infty = v1 \times v2$$

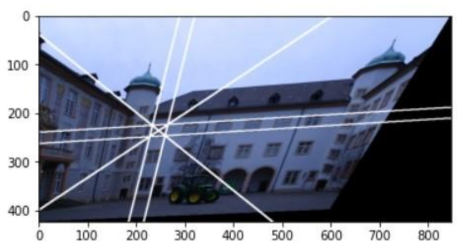
Original



Real World Parallel Lines



Affine Transformation



Recovered Parallelism

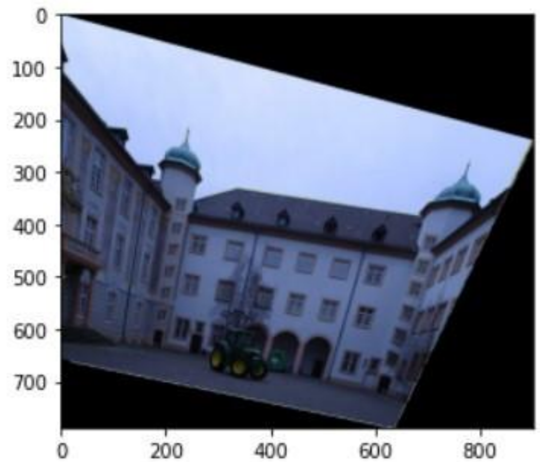
Lab 1: Affine Rectification Results

Image	Set of Parallel Lines	
	L1/L2	L3/L4
Original	0.10	1.34
Rectified	0.0	0.0

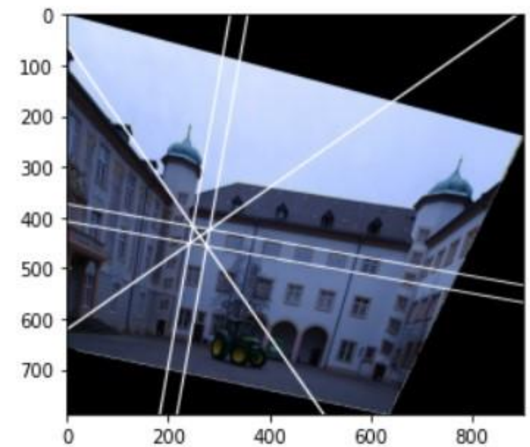
Lab 1: Metric rectification

$$H_{s\leftarrow a} = \begin{bmatrix} K^{-1} & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix}$$

$$(l_1 m_1, l_1 m_2 + l_2 m_1, l_2 m_2)\vec{s} = 0$$



Metric rectified



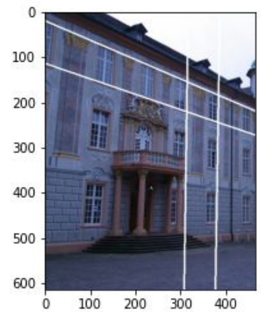
Perpendicularity Recovered

Lab 1: Metric Rectification Results

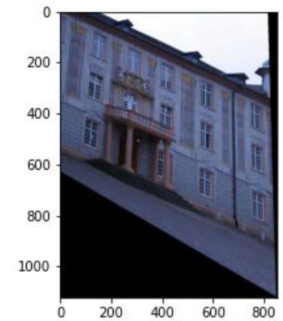
Images	Set of Orthogonal Lines		
	L1/L3	**L2/L4**	**L5/L6**
Affine	72.64	72.64	72.01
Metric	90	90	90

Lab 1: Stratified Rectification on Left Facade

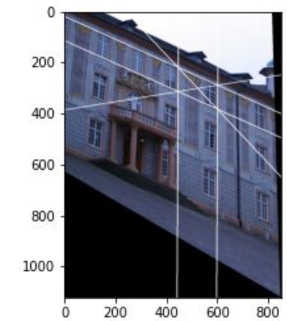
Original



Real Parallel Lines



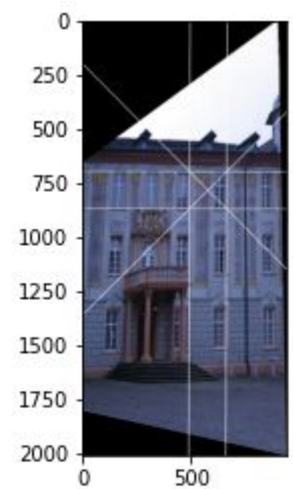
Affine Transformation



Parallelism Recovered

Metric Rectification

Perpendicularity Recovered



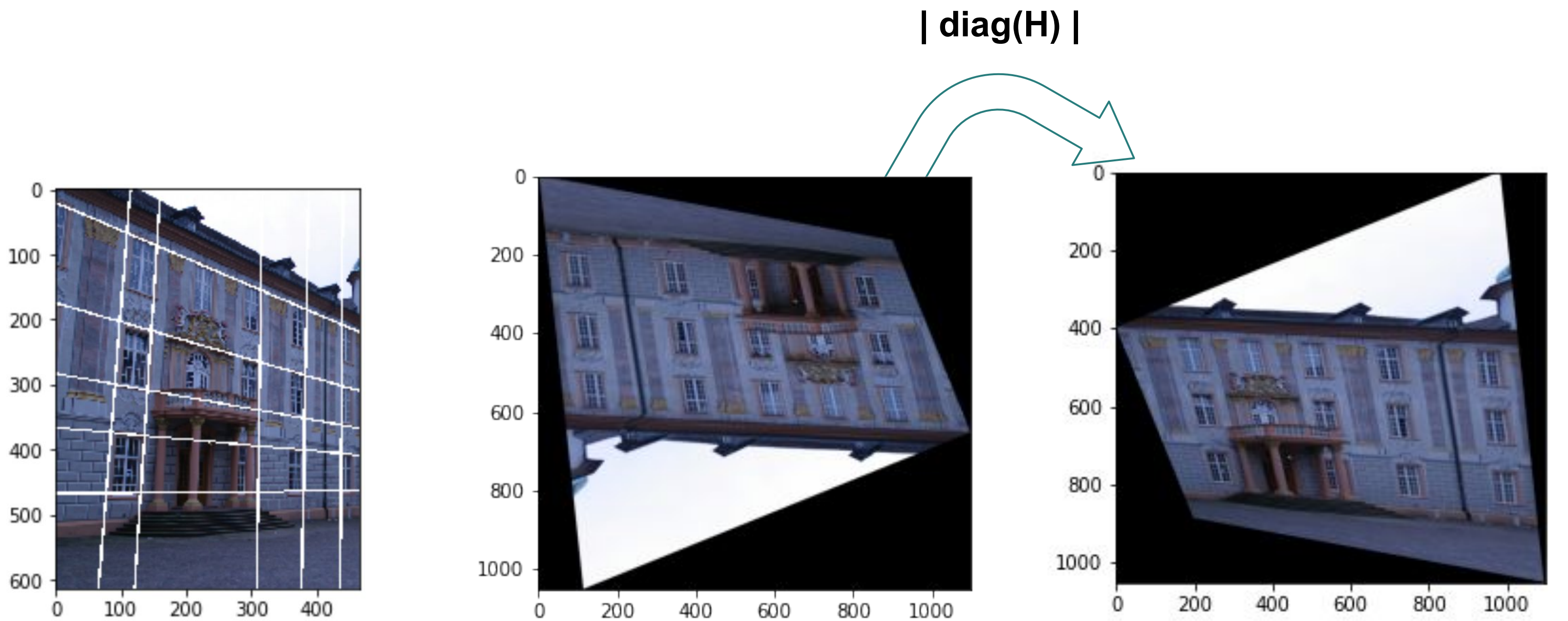
Proper Representation

Image	Set of Parallel Lines	
	L1/L2	L3/L4
Original	4.4	0.13
Rectified	0.0	0.0

Image	Set of Orthogonal Lines		
	L1/L3	L2/L4	L5/L6
Affine	113.8	113.8	124.14
Metric	90	90	90

Lab 1 - OPT: Single step metric rectification

Direct method



Set of Lines

Metric Recovered (Flipped)

Metric Recovered (Corrected)

Lab 2: Homography estimation

Problem Statement

Given a set of 2D-point correspondences between 2 images, calculate the homography that relates two images:

- From the same scene but taken from different viewpoints
- We need a minimum of 4 2D point correspondences -> SIFT and ORB

Algorithms

- Normalized Direct Linear Transformation (N-DLT) algorithm with RANSAC.
- Gold Standard Algorithm

Applications

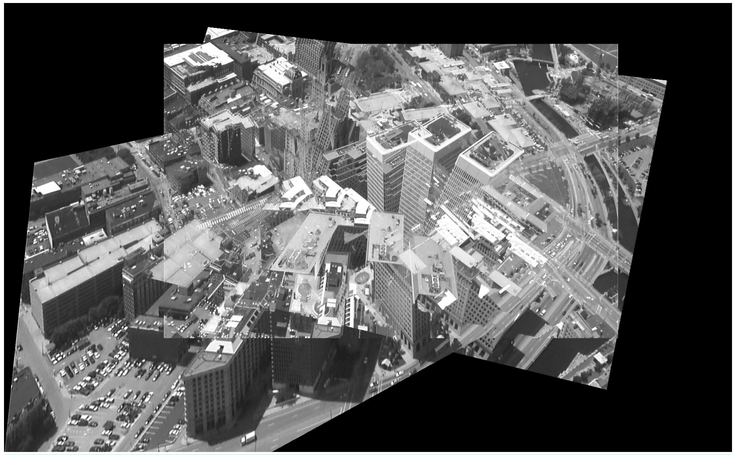
- Calibration with a planar pattern
- Logo detection and replacement
- Image mosaicking

Lab 2: Image mosaicking

Llanes panorama

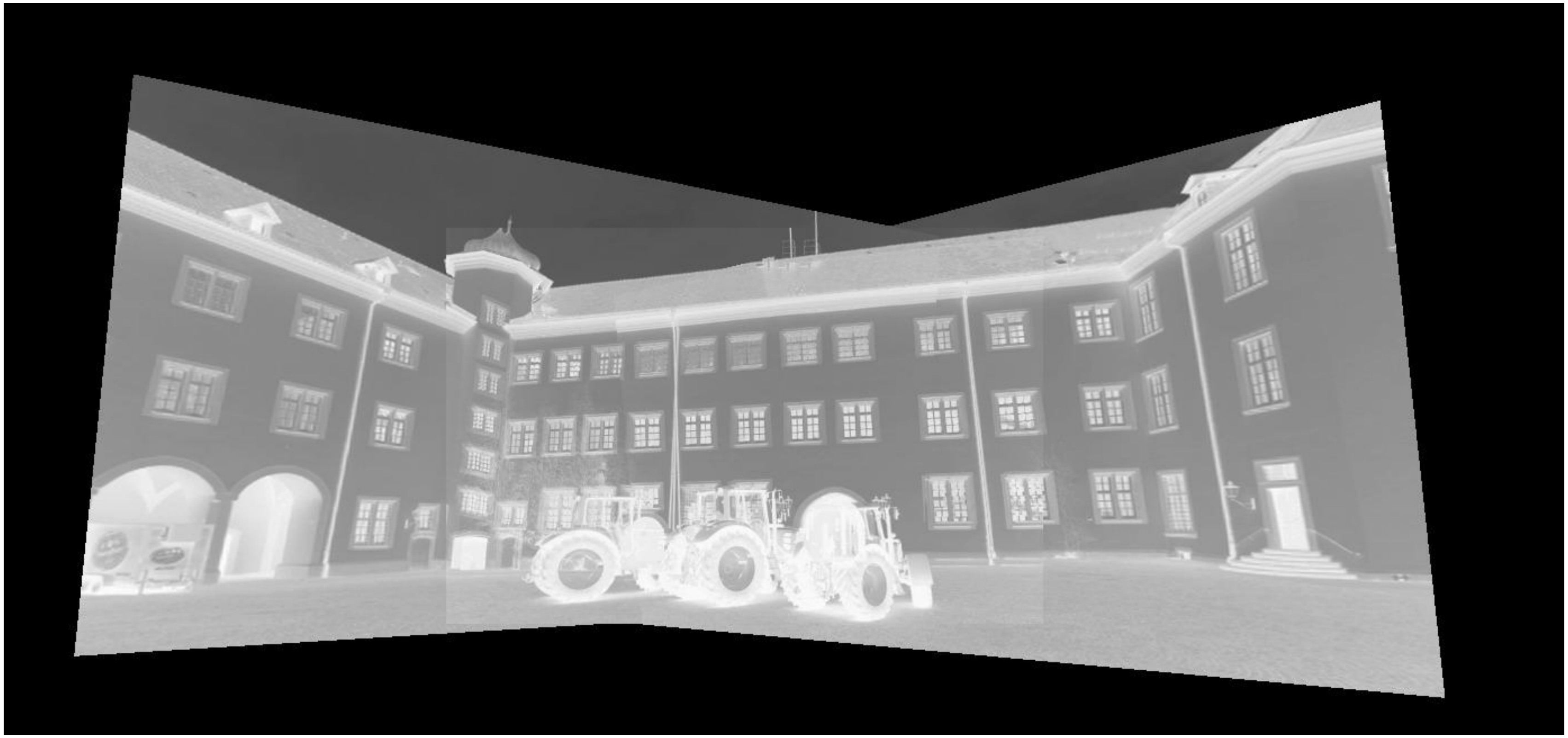
Castle mosaic

Aerial Site 13 mosaic



Aerial Site 22 mosaic

Lab 2: Image mosaicking



Castle mosaic

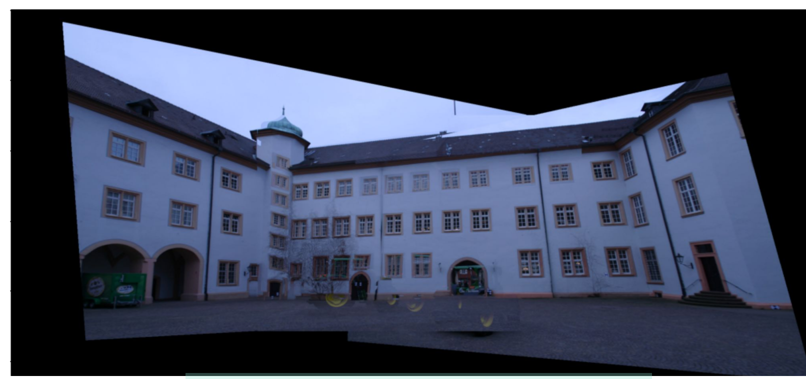
Lab 2: Gold Standard algorithm

Problem Statement

- The Gold Standard Algorithm is used to get a robust estimation of the homography H.
- It uses the Levenberg-Marquadt iterative algorithm to minimize the reprojection error.

H	Reprojection Error	
	Original	Refined
H_{12}	8963534	16
H_{23}	43022747	27

Refined moisaic



Non-refined moisaic

Lab 2: Calibration with a planar pattern

Problem Statement

Homographies can be used for camera calibration (Zhang's algorithm) by modeling a camera as the relationship between a set of 3D points X and their image projection x (x = PX) .

Approach

- The camera matrix P can be decomposed as

$$P = K[R|t]$$

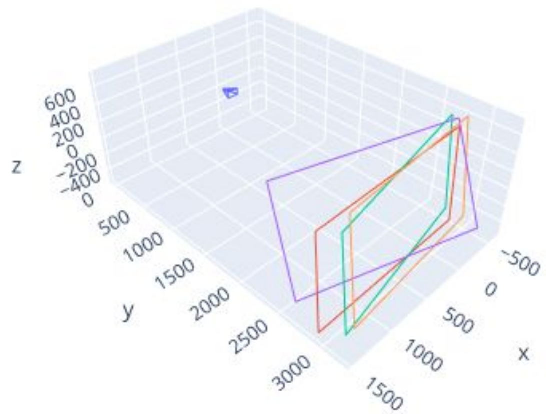
- To find K we use the image w of the absolute conic as:

$$\omega = K^{-T}K^{-1}$$

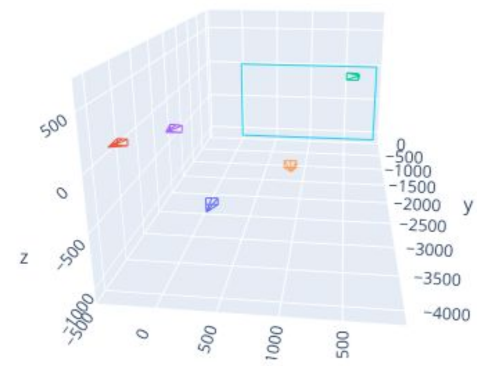
  with 6 unknowns.

- For each image we have a set of two equations, so at least we need 3 images.
- K is found by Cholesky. Once K is found, the external parameters can be estimated

Lab 2: Calibration with a planar pattern

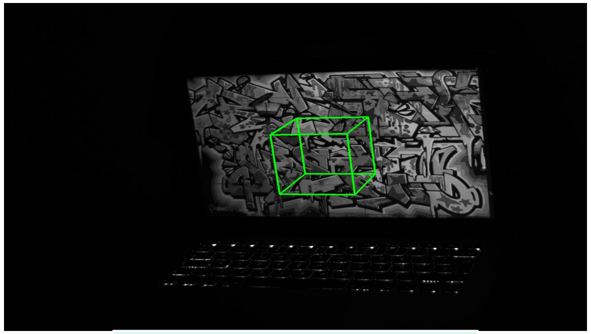


Planes positions

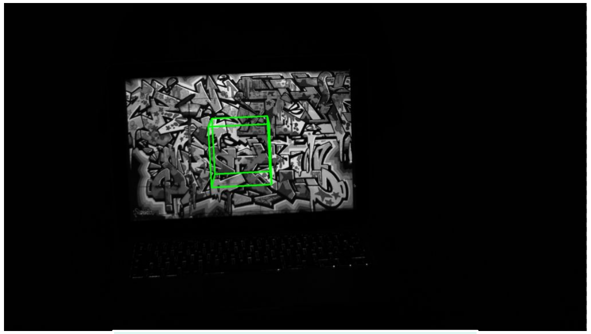


Camera positions

Once the images are calibrated and the relative pose between the camera and the planar patterns is recover, we can place virtual objects on the image



Cube on image 4

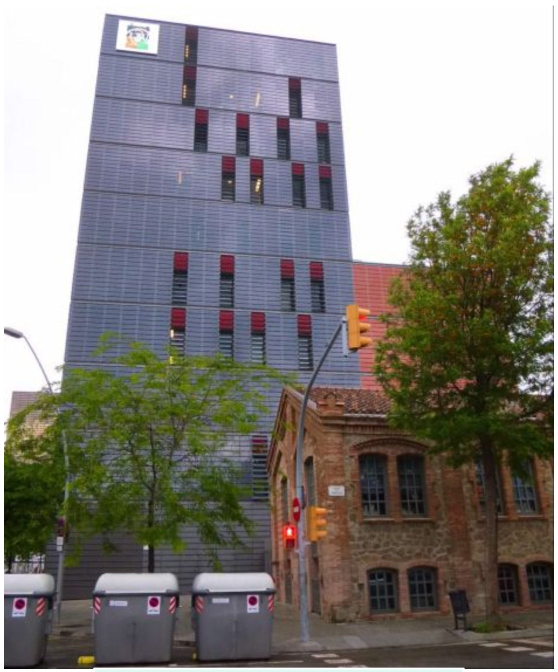


Cube on image 5

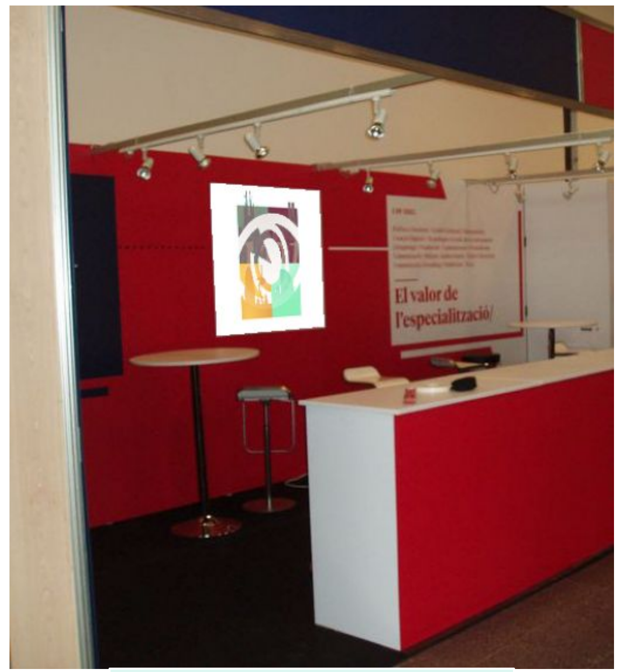
Lab 2: Logo detection and replacement

Implementation

- Find correspondences between the logo and the main image.
- Compute relating homography.
- Transform the logo with the homography.



Logo on UPF building



Logo on UPF stand

Lab 3: Fundamental matrix estimation

Problem Statement

Given a set of 2D-point correspondences between 2 images, calculate the fundamental matrix that relates two images:

- From the same scene but taken from different viewpoints
- We need a minimum of 8 point correspondences -> ORB or SIFT

Algorithms

- Normalized 8-point algorithm (algebraic method)
- Robust normalized 8-point algorithm (with RANSAC)

Applications

- Photo sequencing

Lab 3: Epipolar lines - Results

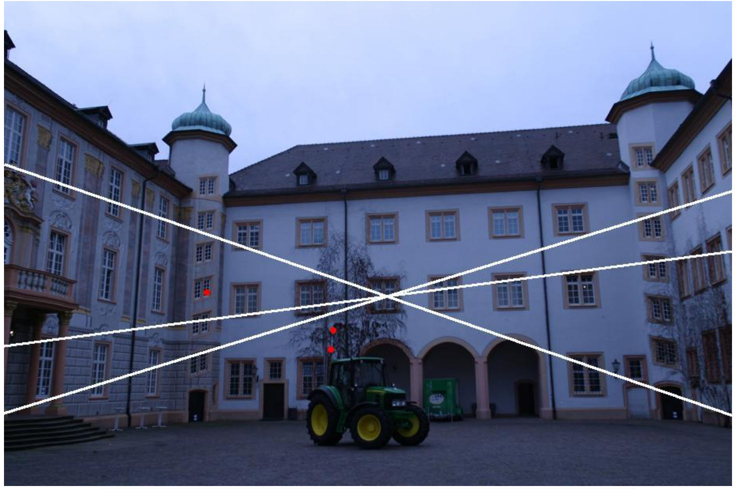
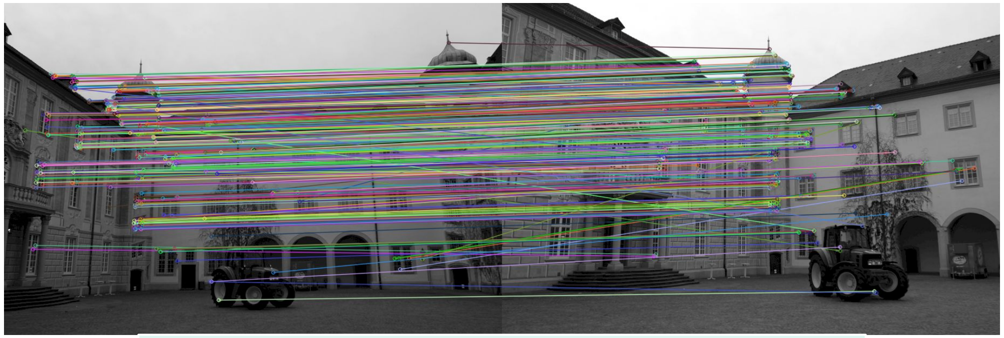
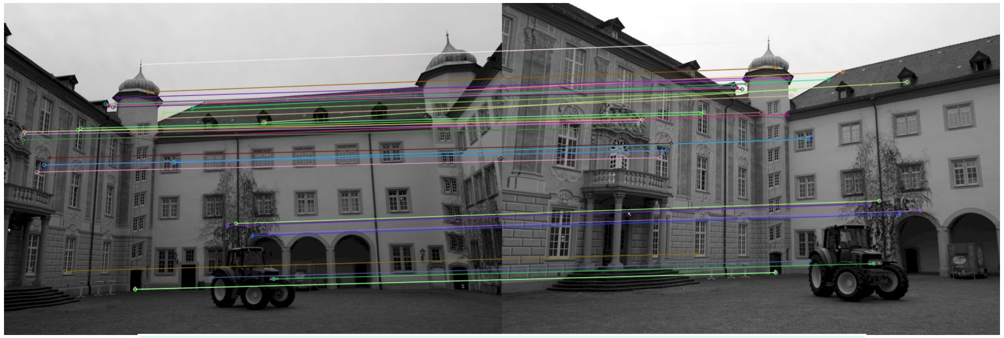


Image 1 - ORB



“Inliers” - ORB

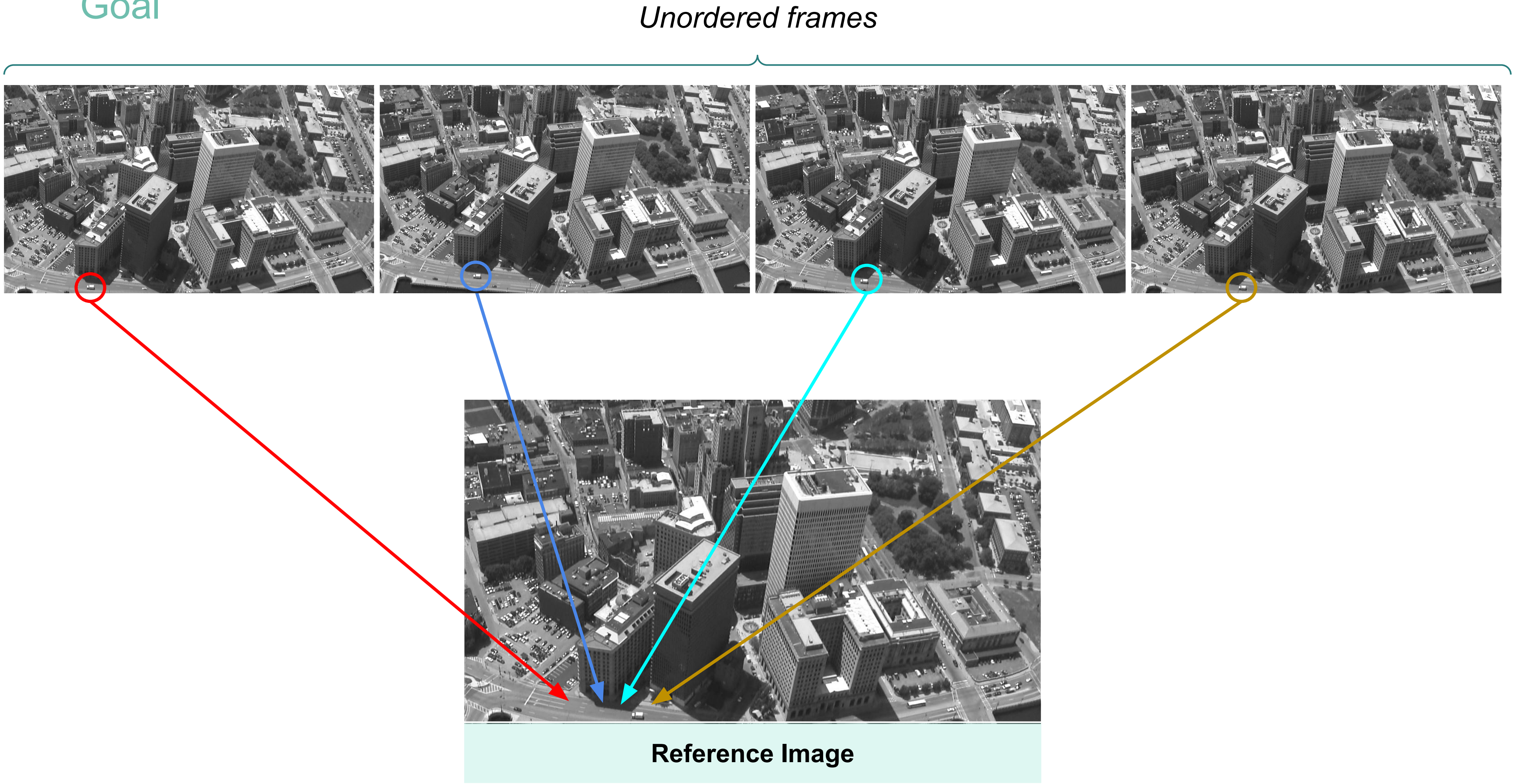
Image 1 - SIFT



Inliers - SIFT

Lab 3: Photo sequencing - Aerial

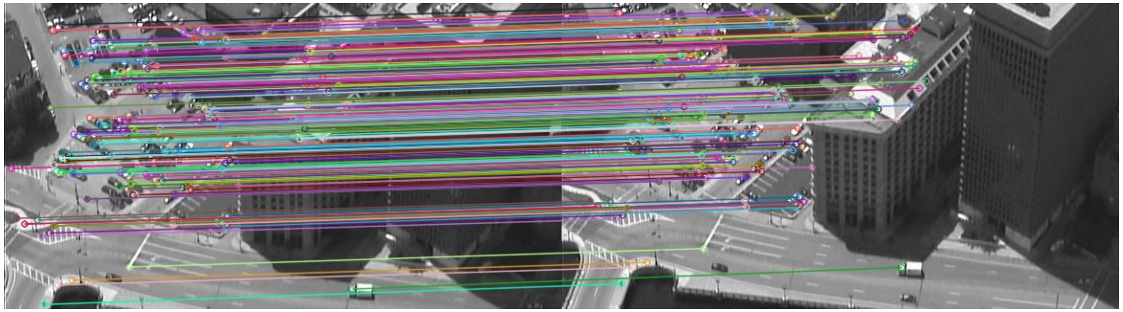
Goal



Tali Dekel, Yael Moses, and Shai Avidan, "Photo sequencing," International Journal of Computer Vision, vol. 110, no. 3, pp. 275–289, 2014.

Lab 3: Photo sequencing - Aerial

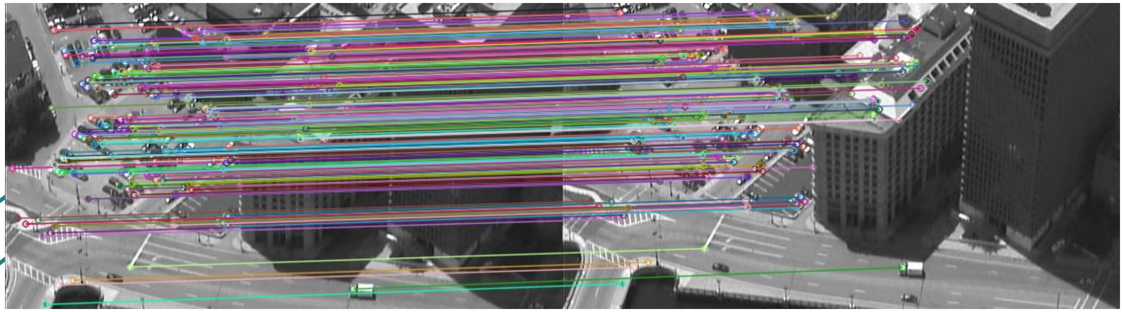
Keypoint matches



Lab 3: Photo sequencing - Aerial

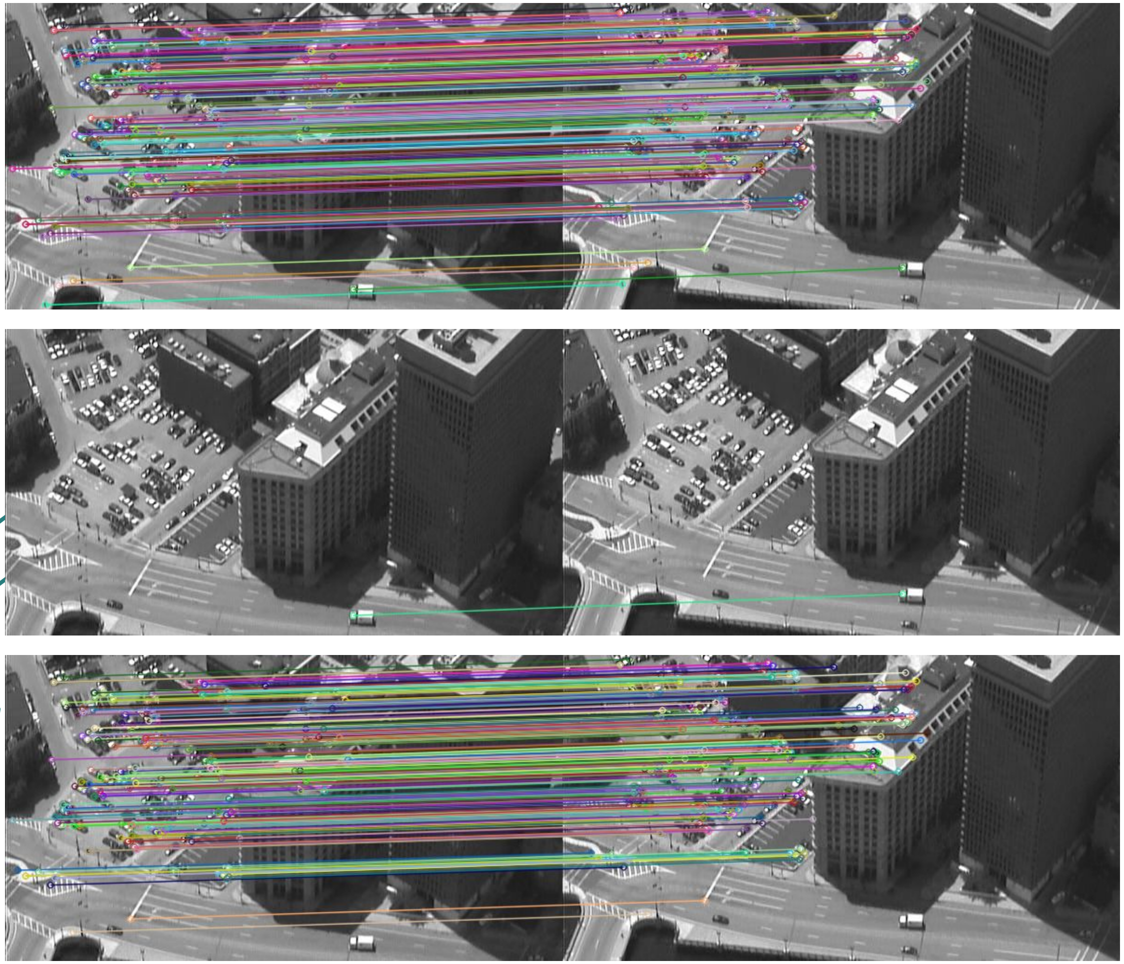
Match on the dynamic object

We need this!



Lab 3: Photo sequencing - Aerial

Inliers

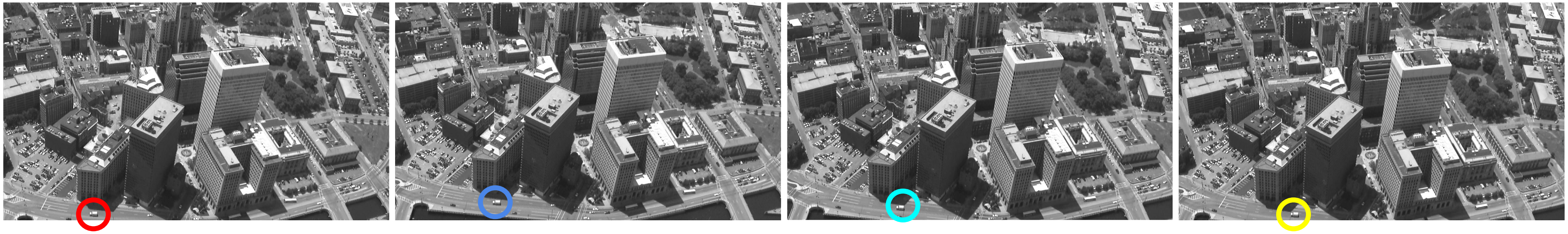


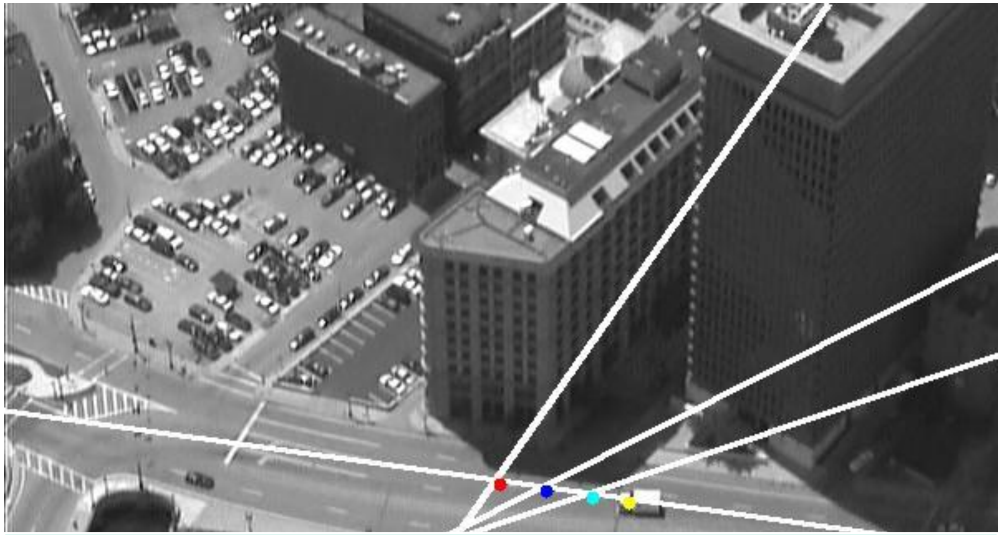
Only the static parts!

Lab 3: Photo sequencing - Aerial

Result

Unordered frames





Van 3D trajectory

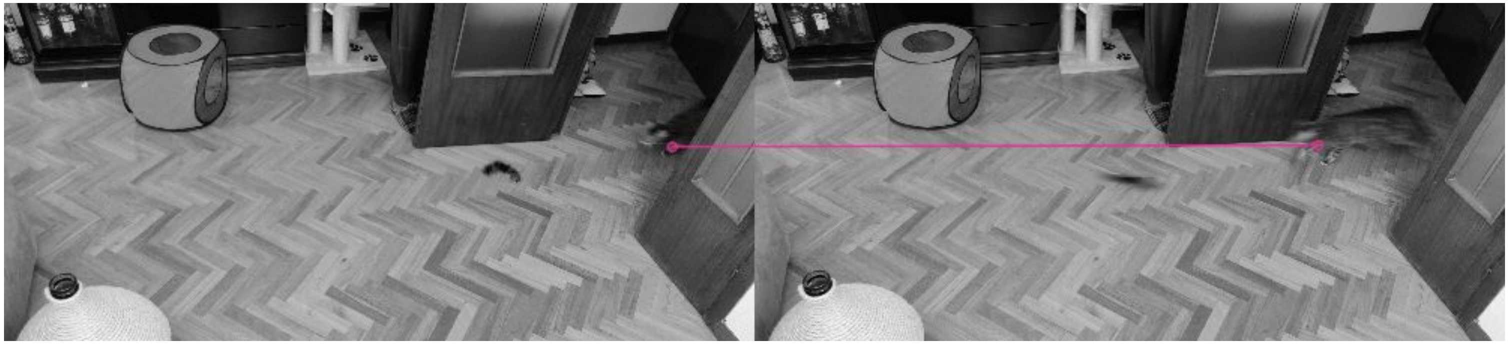
Lab 3: Photo sequencing - Nala

Initial frames

- **Much closer point of view**
- **Blurry dynamic object**

Lab 3: Photo sequencing - Nala

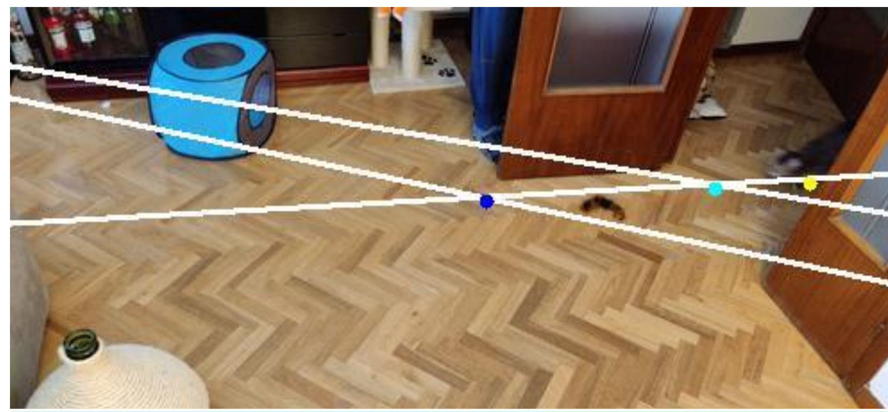
Matches on the dynamic object



- **Not so accurate**

Lab 3: Photo sequencing - Nala

Result



Nala 3D trajectory

Lab 3: Photo sequencing - BCN street

Initial frames

- **Moving in opposite directions**

Lab 3: Photo sequencing - BCN street

Result

Pedestrian 3D trajectory

Van 3D trajectory

Lab 4: Reconstruction from two images

Goal

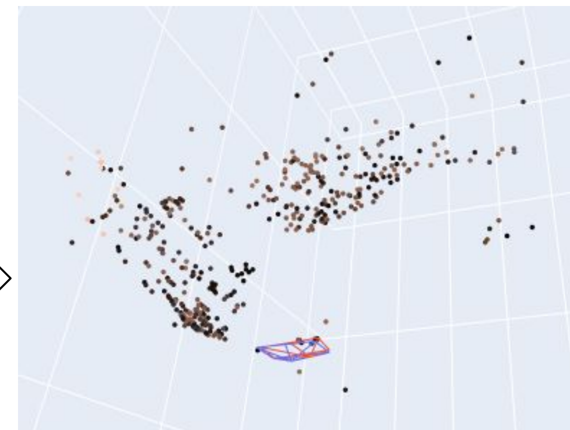
View 1

View 2

Triangulation

-2D correspondences

-Camera projection matrices

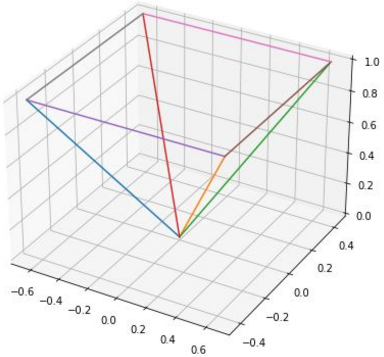


3D reconstruction

Lab 4: Reconstruction from two images

First camera matrix: *P*

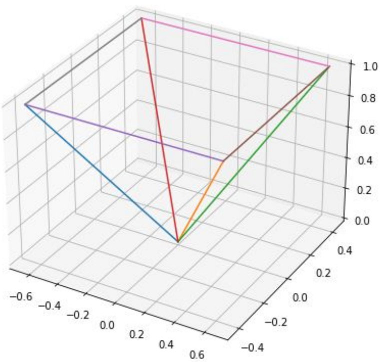
$$P = K[I \mid 0]$$



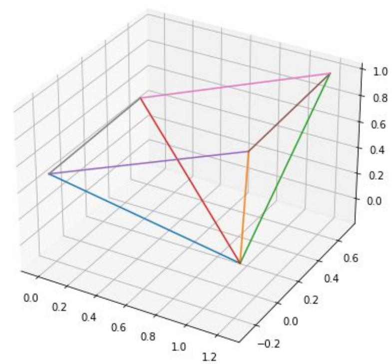
Lab 4: Reconstruction from two images

Second camera matrix: 4 candidates

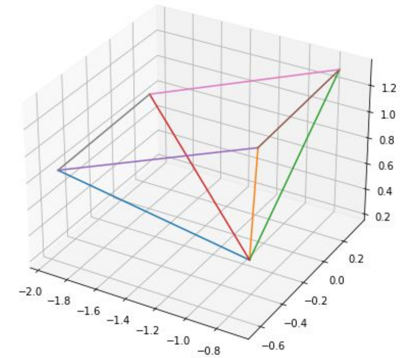
$$P = K[I \mid 0]$$



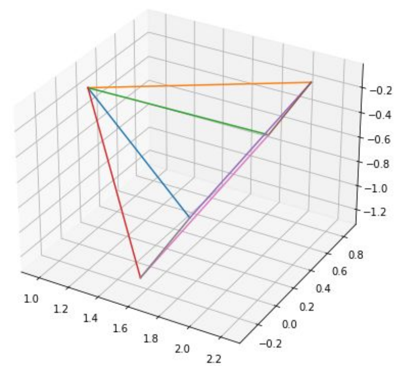
$$E = K'^T F K$$



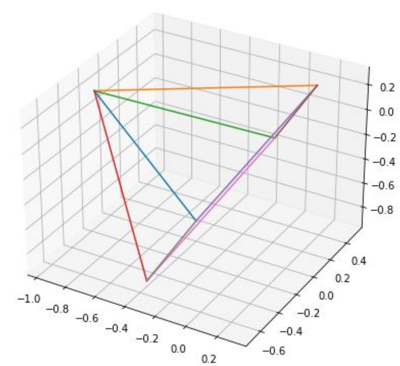
$$P'_1 = [UWV^T \mid +\mathbf{u}_3]$$



$$P'_2 = [UWV^T \mid -\mathbf{u}_3]$$



$$P'_3 = [UW^TV^T \mid +\mathbf{u}_3]$$

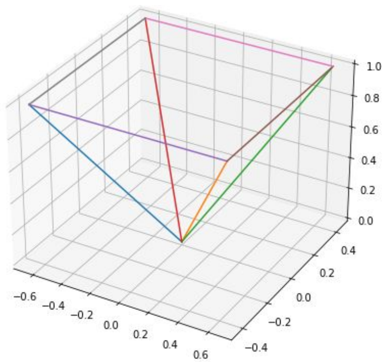


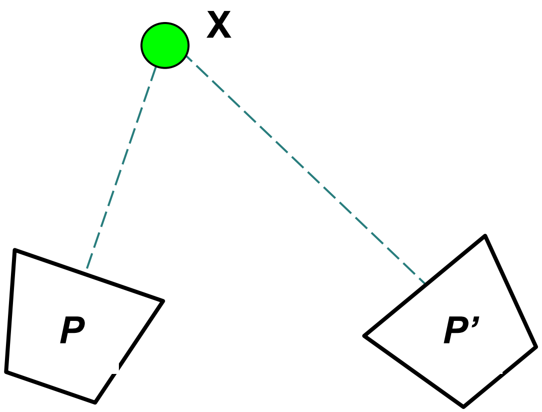
$$P'_4 = [UW^TV^T \mid -\mathbf{u}_3]$$

Lab 4: Reconstruction from two images

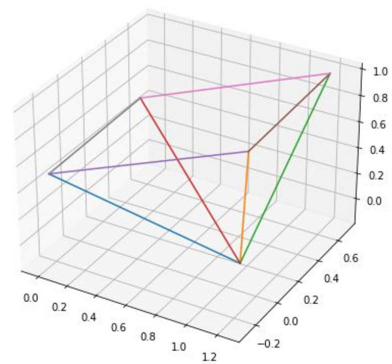
Second camera matrix: *P’*

$$P = K[I \mid 0]$$

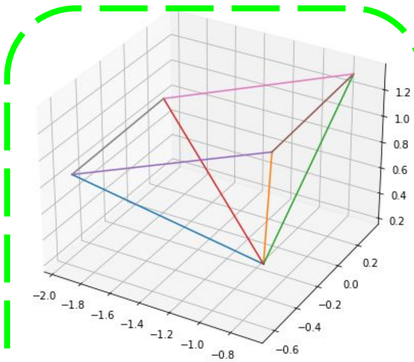




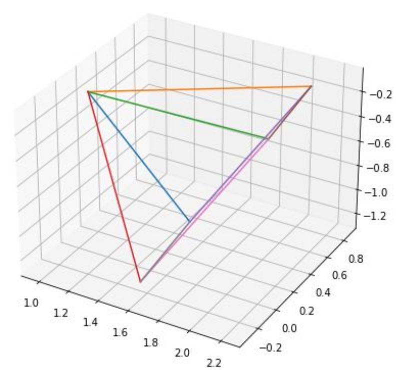
$$E = K'^T F K$$



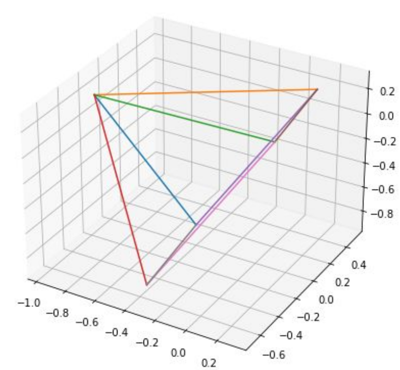
$$P'_1 = [UWV^T \mid +\mathbf{u}_3]$$



$$P'_2 = [UWV^T \mid -\mathbf{u}_3]$$



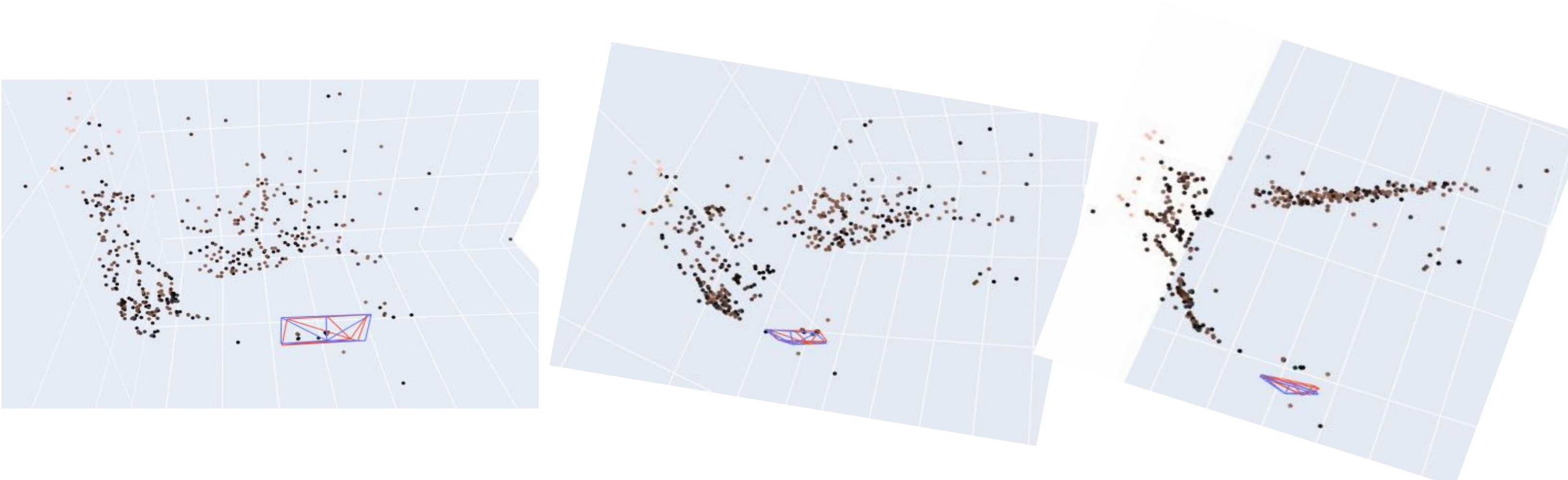
$$P'_3 = [UW^TV^T \mid +\mathbf{u}_3]$$



$$P'_4 = [UW^TV^T \mid -\mathbf{u}_3]$$

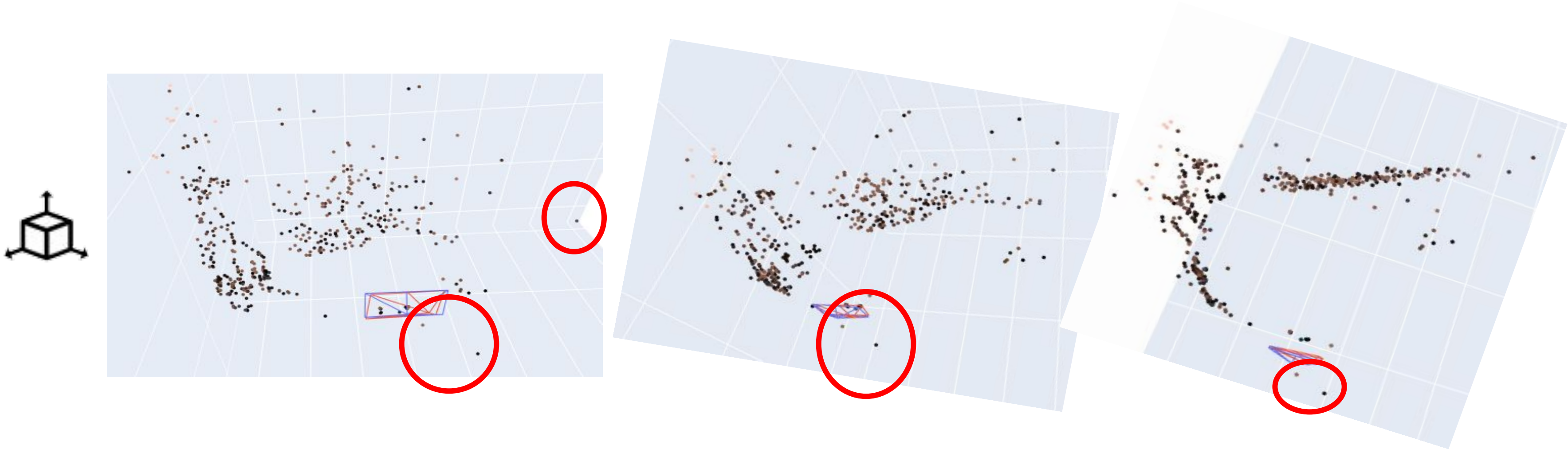
Lab 4: Reconstruction from two images

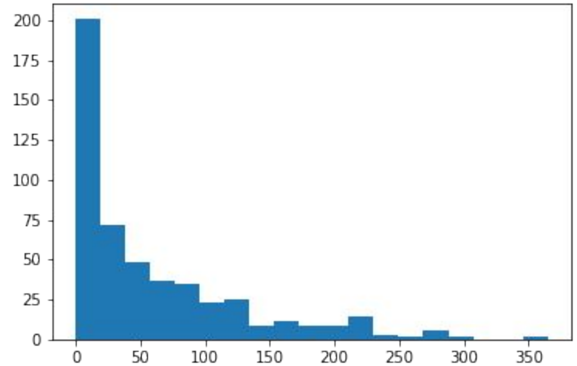
Final 3D reconstruction



Lab 4: Reconstruction from two images

Reprojection error

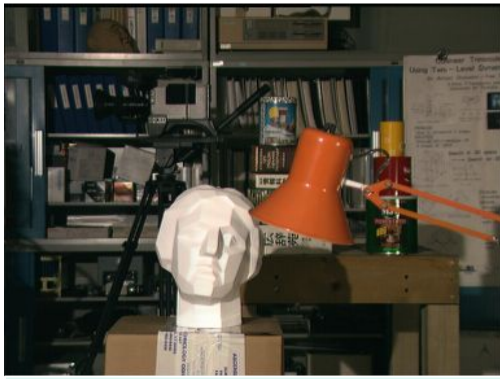




Histogram of errors

Lab 4: Disparity Map Computation

Goal



View 1

View 2

Disparity Estimation

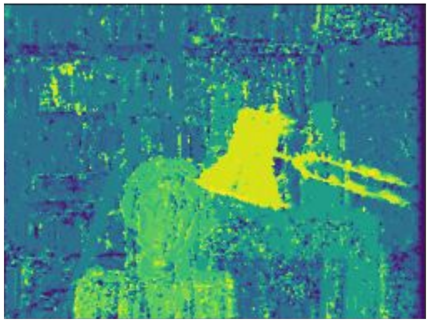
- Sliding window
- Cost function
  - · **SSD**
  - · NCC
- Minimum cost

Disparity Map

Lab 4: Disparity Map Computation

Results: Small window size

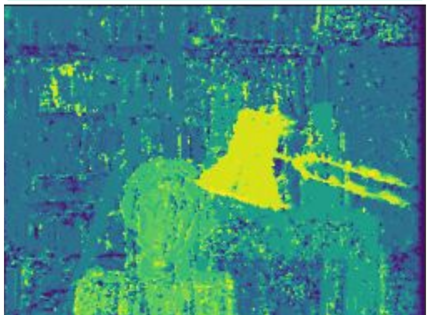
Window size: 3x3



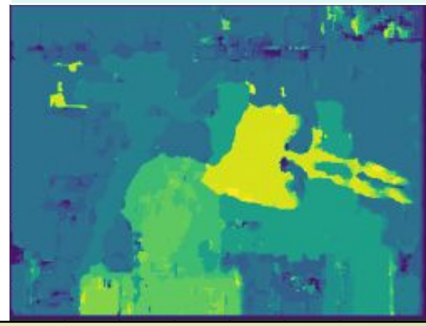
Lab 4: Disparity Map Computation

Results: Larger window sizes

Window size: 3x3

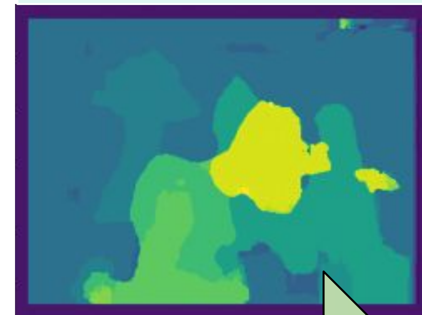


Window size: 9x9



Window size: 21x21

Window size: 31x31

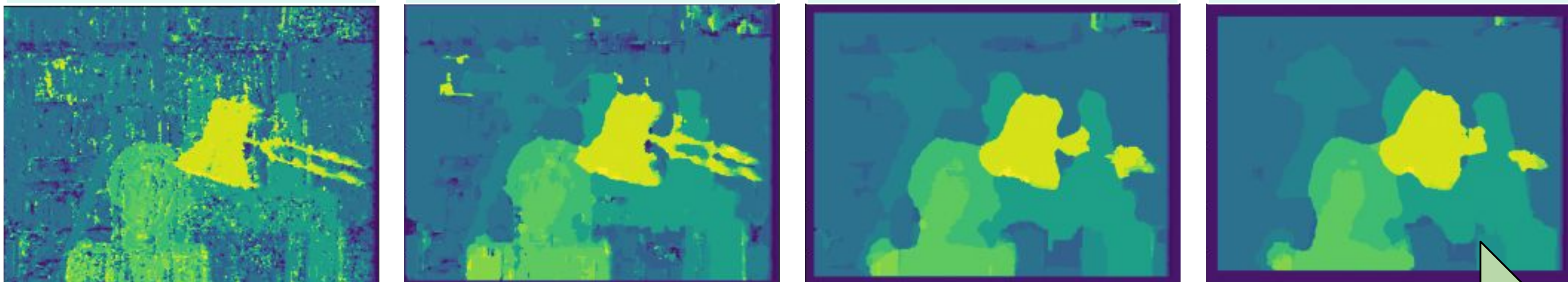


- **Smoother disparity maps**

Lab 4: Disparity Map Computation

Results: Larger window sizes

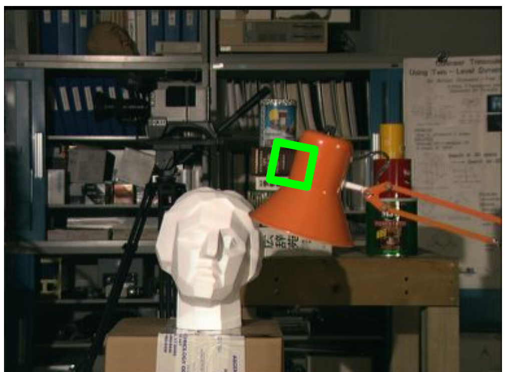
Window size: 3x3	Window size: 9x9	Window size: 21x21	Window size: 31x31



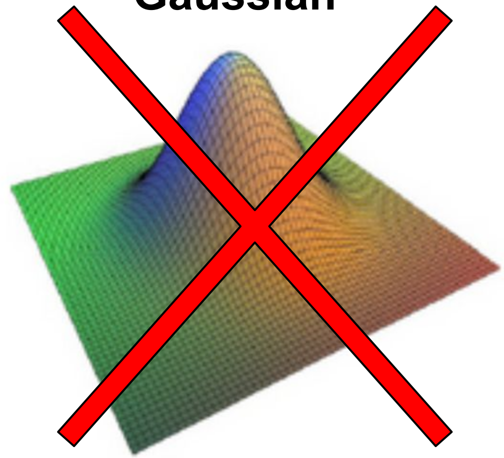
- **Smoother disparity maps**
- **Less details**

Lab 4: Disparity Map Computation

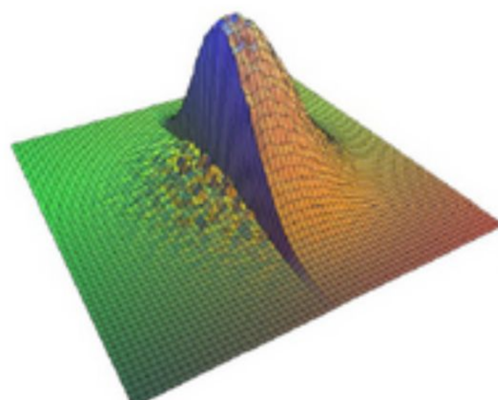
Bilateral weights



Gaussian



Bilateral

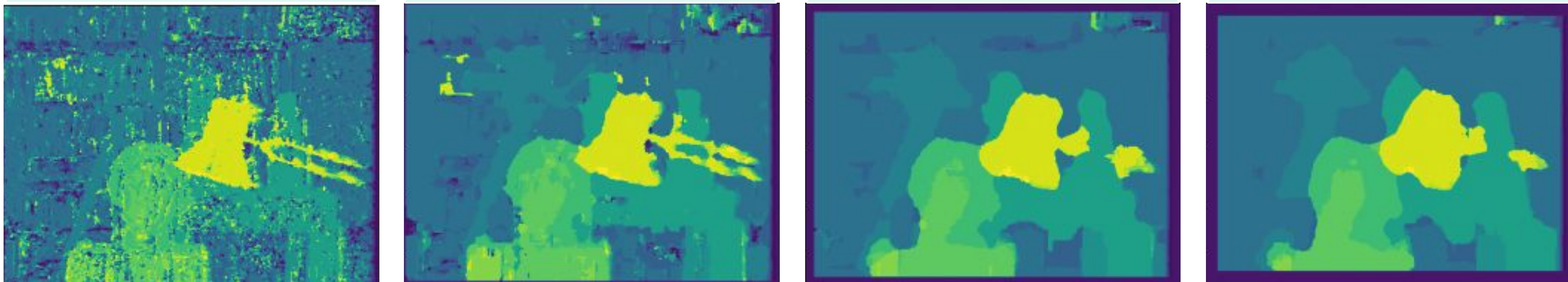


- **Color information**
- **Geometric information**

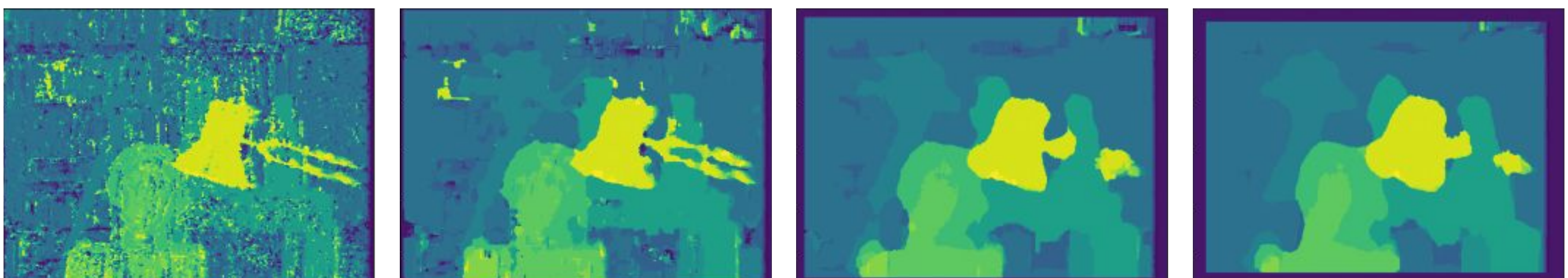
Lab 4: Disparity Map Computation

Comparing results

Window size: 3x3	Window size: 9x9	Window size: 21x21	Window size: 31x31



Previous results

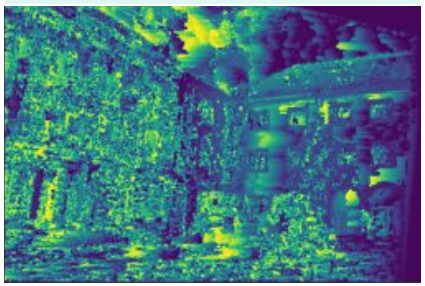


Bilateral weights

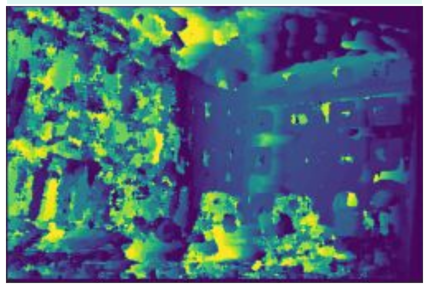
Lab 4: Disparity Map Computation

Facade images

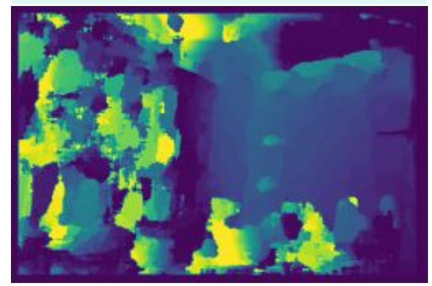
Window size: 3x3



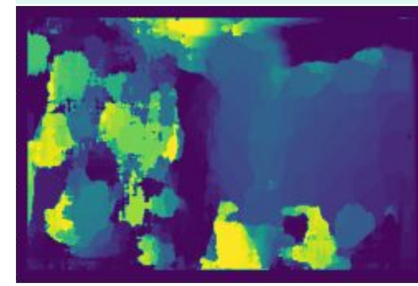
Window size: 9x9



Window size: 21x21



Window size: 31x31



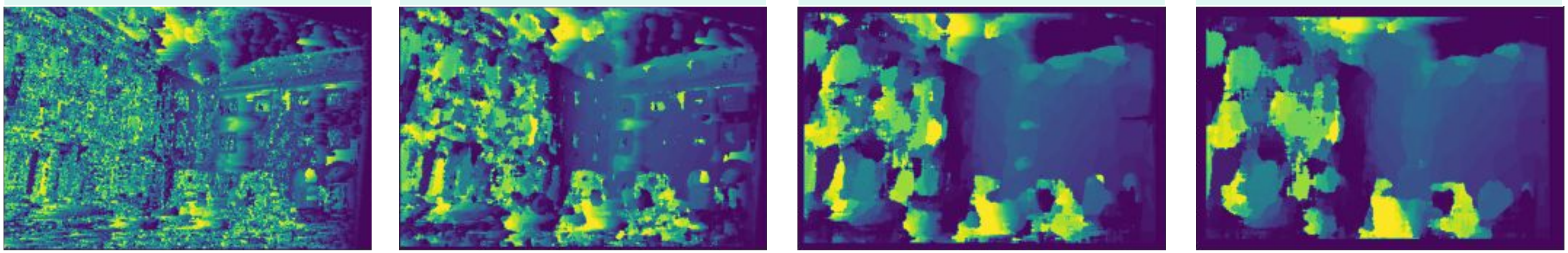
Worse results

- Objects further away
- Repetitive patterns

Lab 4: Disparity Map Computation

Facade images

Window size: 3x3	Window size: 9x9	Window size: 21x21	Window size: 31x31

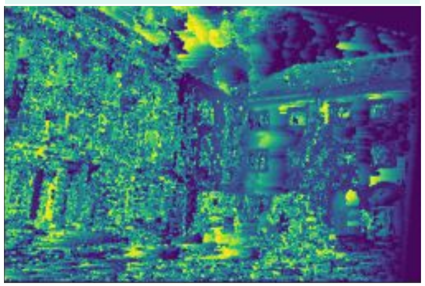


Worse results
- **Objects further away**
- Repetitive patterns

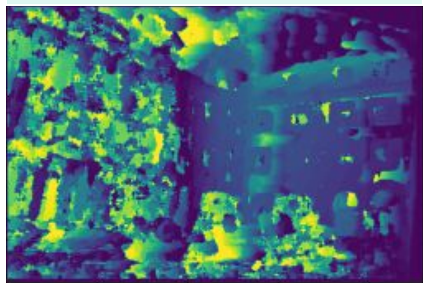
Lab 4: Disparity Map Computation

Facade images

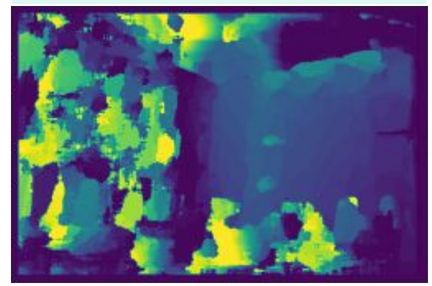
Window size: 3x3



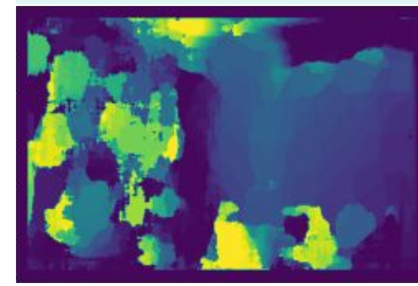
Window size: 9x9



Window size: 21x21



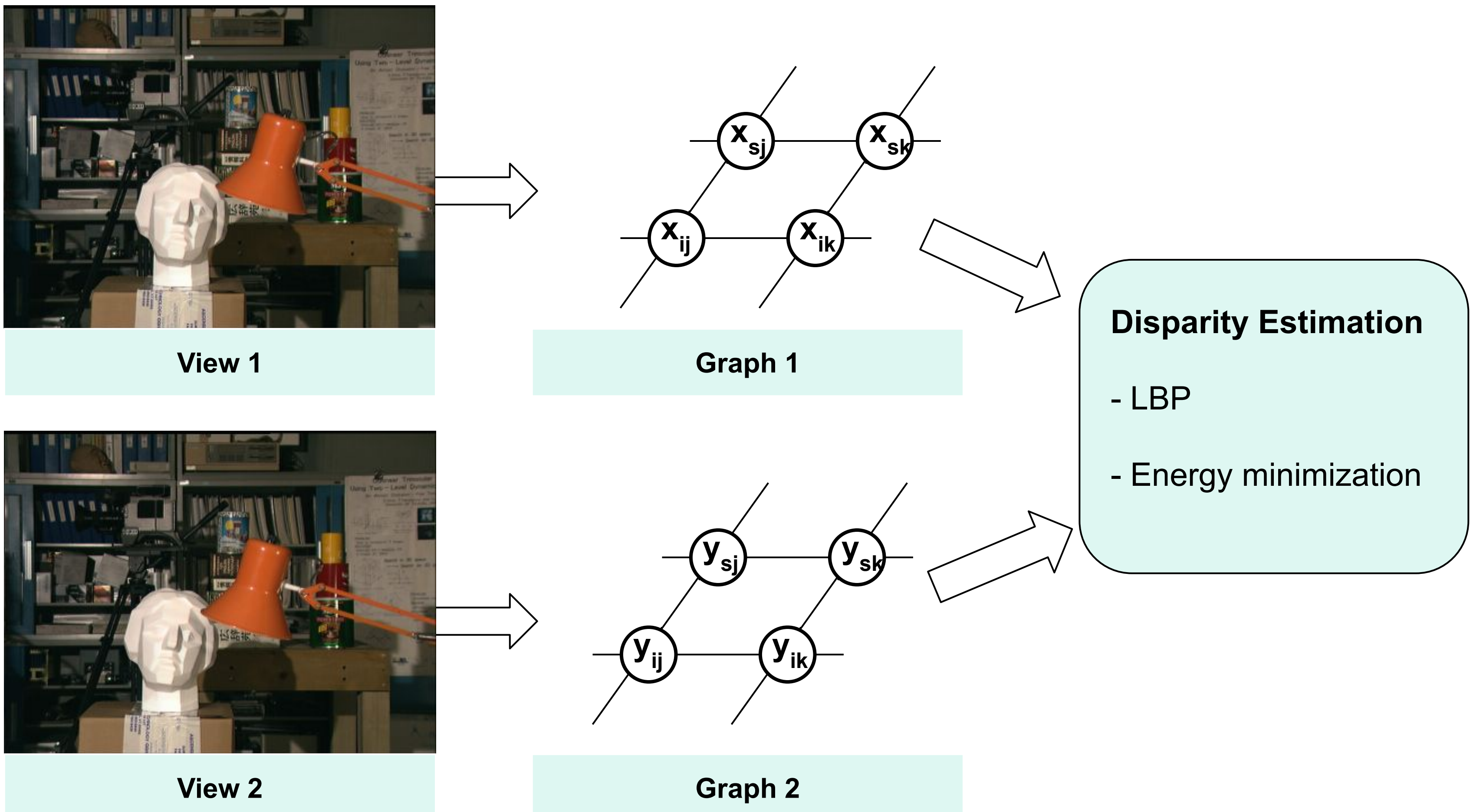
Window size: 31x31



Worse results
- **Objects further away**
- **Repetitive patterns**

Lab 4: Disparity Map Computation

Loopy Belief Propagation (LBP)



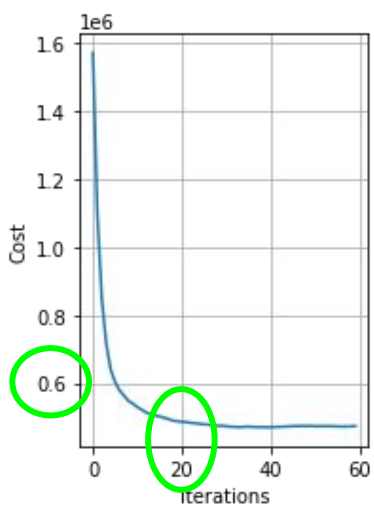
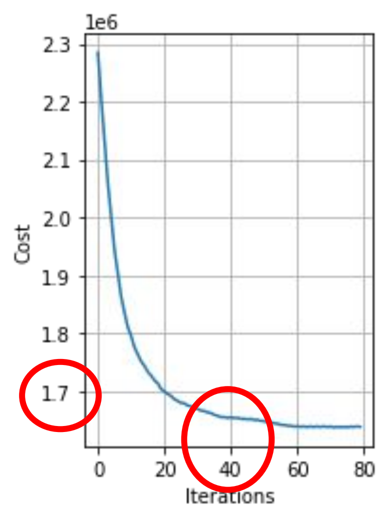
Lab 4: Disparity Map Computation

LBP: Results

Scene1	Facade
	

Lab 4: Disparity Map Computation

LBP: Cost functions

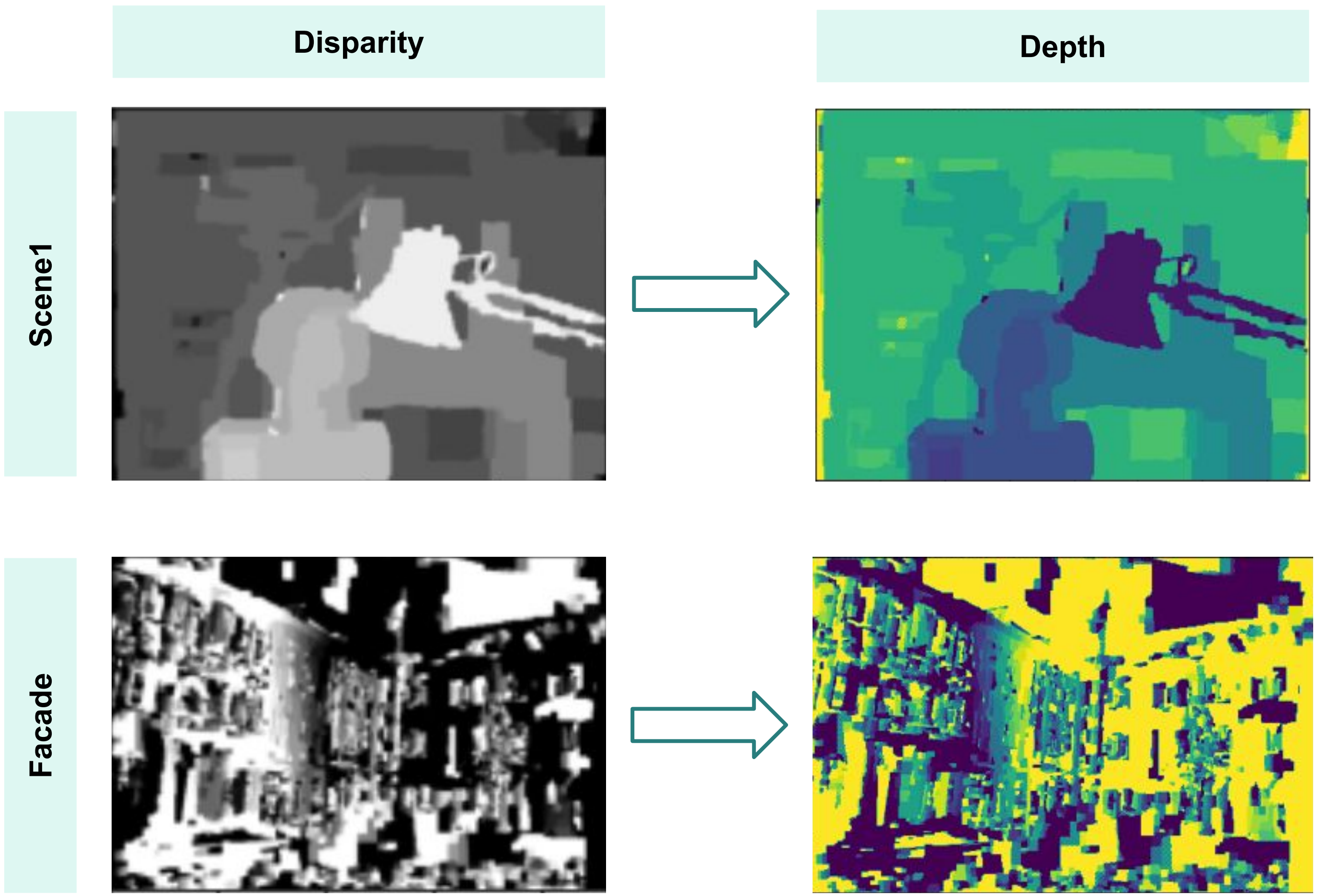
Scene1	Facade
	
	

- **More iterations**
- **Higher cost**

Lab 4: Disparity Map Computation

Depth from disparity

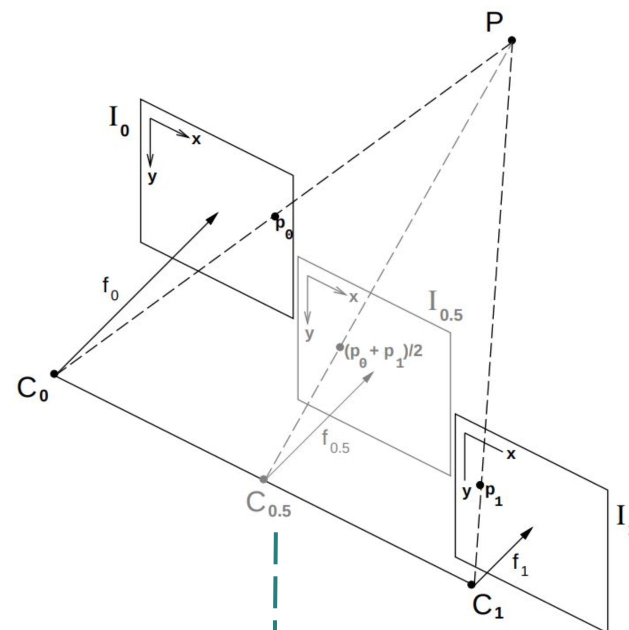
$$disparity = x - x' = \frac{baseline * f}{z}$$



Lab 4: New view synthesis

Method

	Image	Disparity map
Left		
Right		



Here, s = 0.5

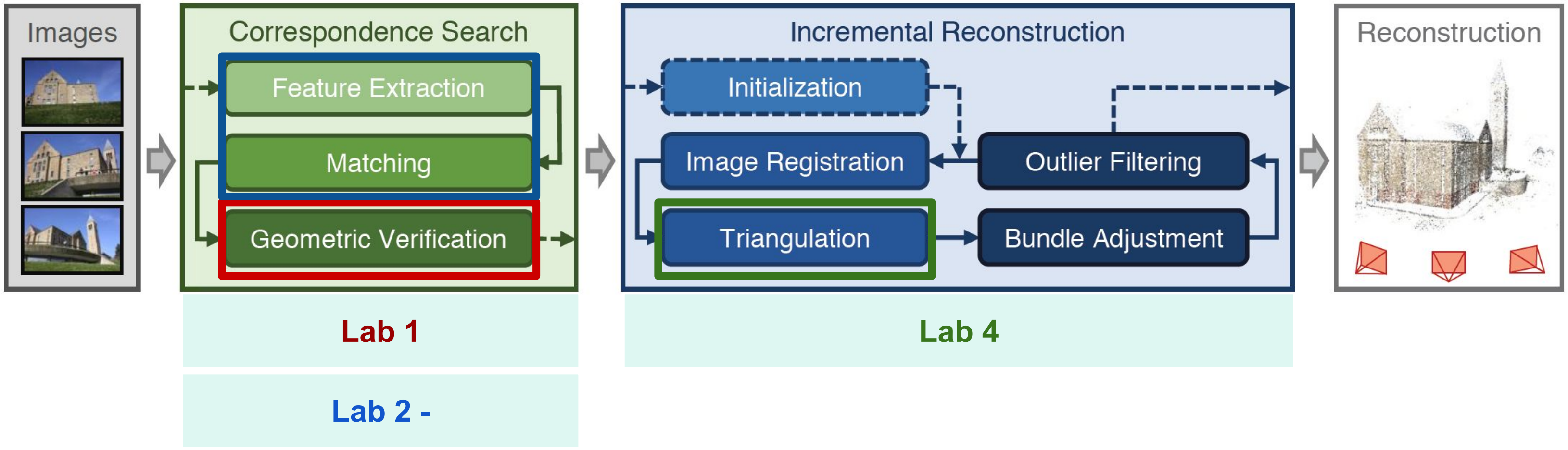
S. M. Seitz and C. R. Dyer, "View morphing," Conference on Computer Graphics and Interactive Techniques, ser. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 21–30.

Lab 4: New view synthesis

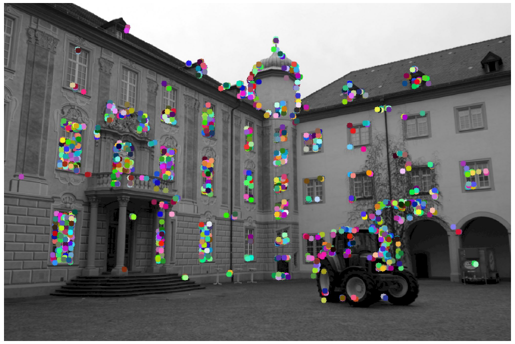
Resulting GIF

GIF generated with 9 new views

Lab 5 - Intro SFM



Lab 5 - Correspondence Search



find_features_orb

match_features_hamming

compute_fundamental_robust

refine_matches

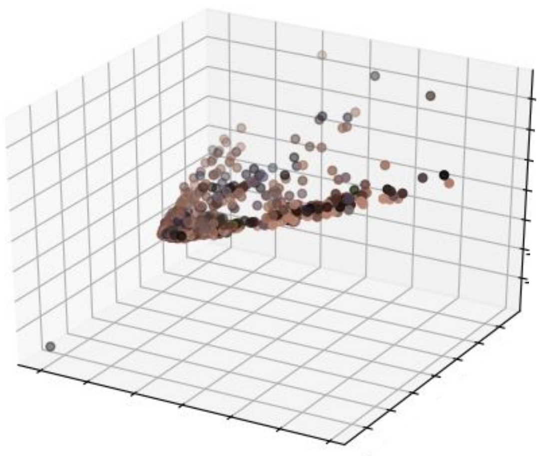
display_epilines

Lab 5 - Projective reconstruction

Projective camera matrices

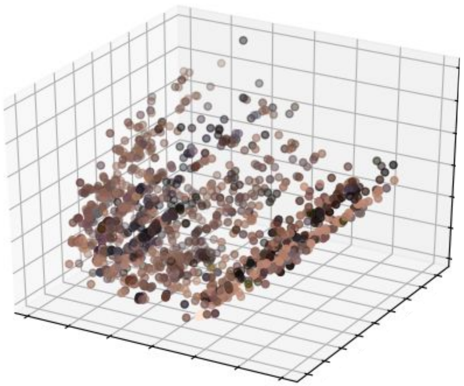
$$P_0 = [I \mid 0]$$

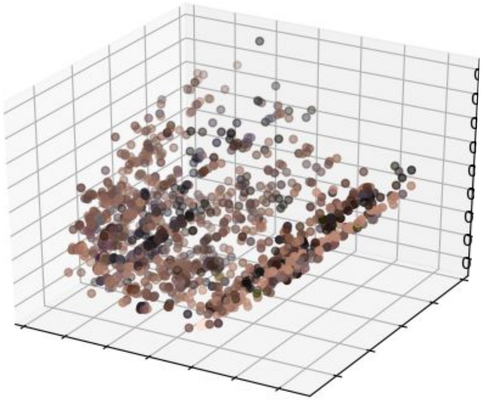
$$P' = \left[[\mathbf{e}']_\times F + \mathbf{e}'\mathbf{v}^\top \mid \lambda\mathbf{e}'\right]$$



Lab 5 - Geometric Verification: Rectification

Affine	Metric





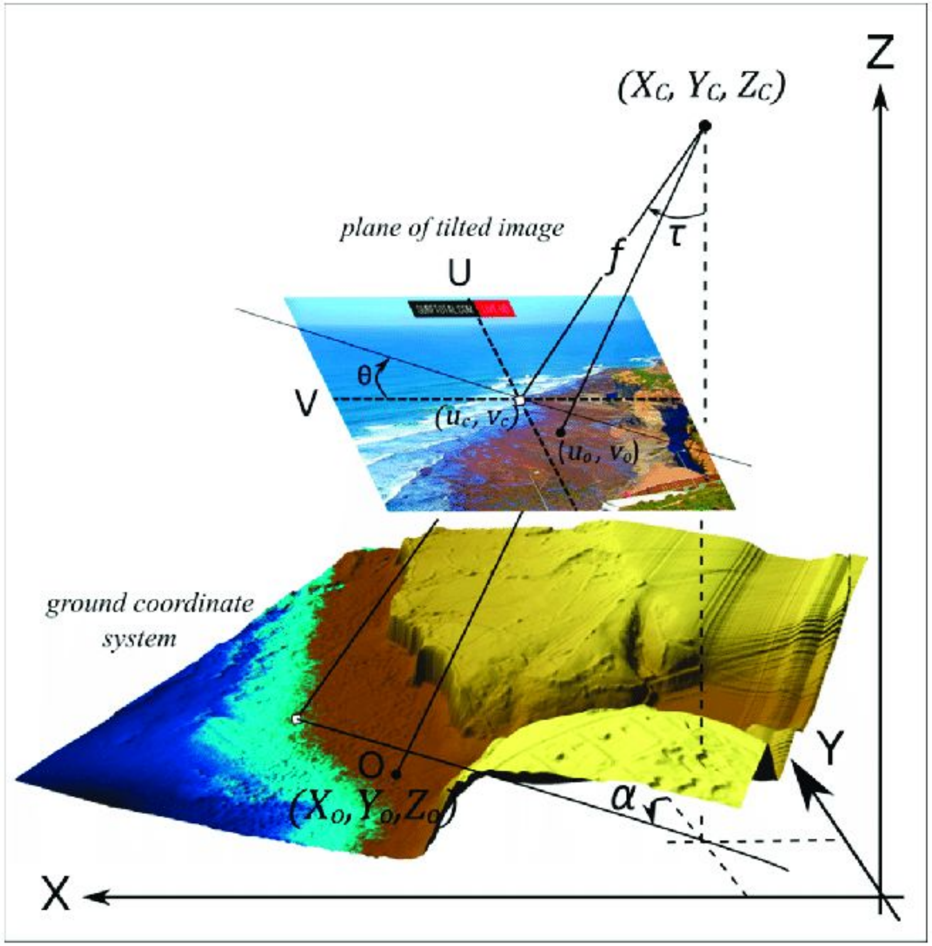
Lab 5 - Reprojection Error

$$\sum_i d\left(\mathbf{x}_i, \hat{\mathbf{x}}_i\right)^2 + d\left(\mathbf{x}'_i, \hat{\mathbf{x}}'_i\right)^2$$

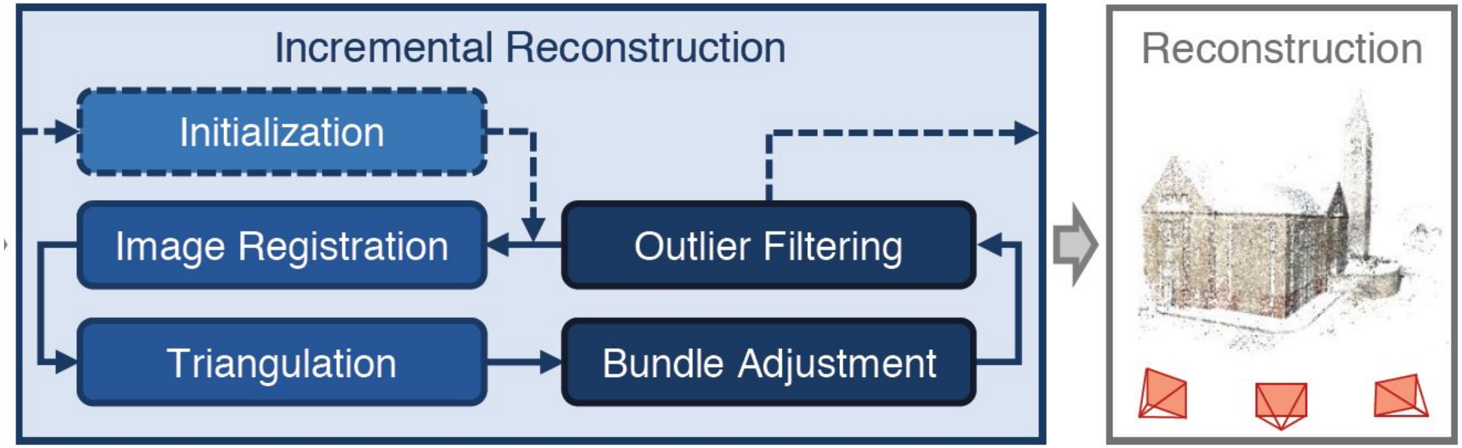
where $\hat{x} = PX$ and $\hat{x}' = P'X$

Reprojection error	Intrinsics	
	Yes	No
Reprojective		3.50695067e-07
Affine		3.50695140e-07
Euclidean	8.614e08	3.50695165e-07

Lab 5 - Resection method



Lab 5 - Incremental Reconstruction



Conclusions

- To obtain good results we rely completely on finding good correspondences
- RANSAC is more robust, but it is random and results are not consistent
- The relative position between images of a set is important
- The methods that we applied need to be supervised, it is not automatic
- From just a pair of close images, we can mosaic them, perform a 3d reconstruction, calculate the depth maps and even generate new synthetic views

Q&A

Group 7: Josep Brugués i Pujolràs / Sergi García Sarroca
Òscar Lorente Corominas / Ian Riera Smolinska