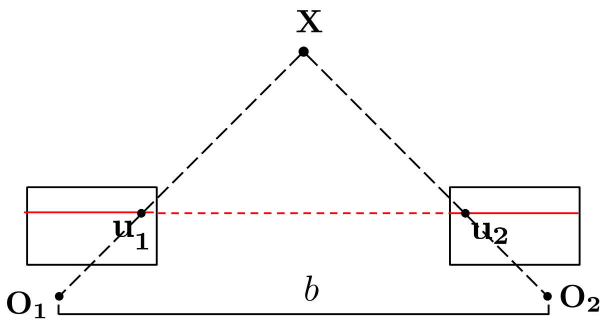
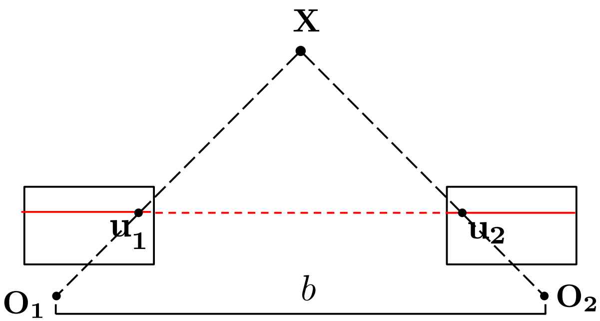
Computer Vision



Class 10

Raquel Frizera Vassallo

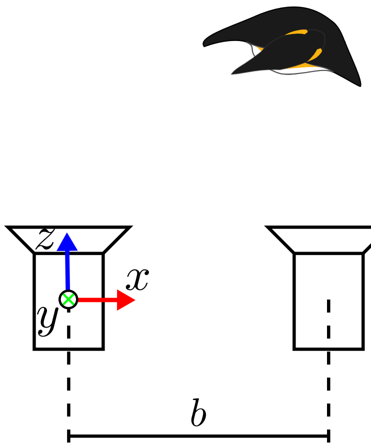
3D Reconstruction from Rectified Systems



Summary

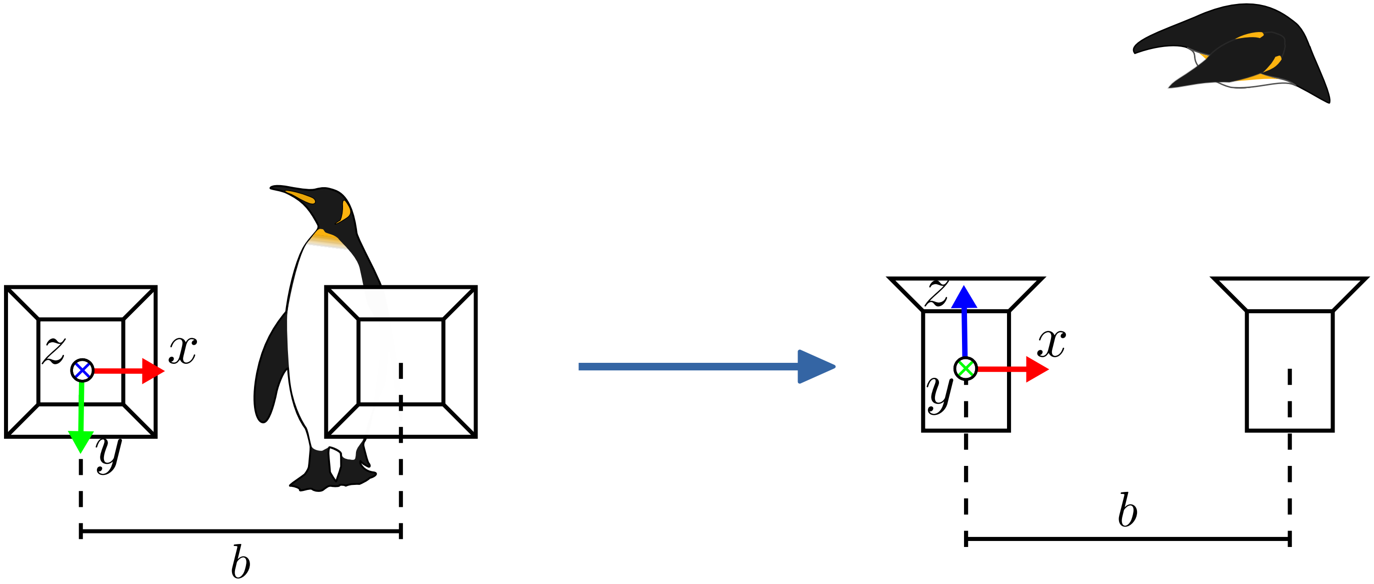
- Rectified System
- 3D reconstruction with rectified images
- Finding correspondences in rectified images
- Disparity Maps
- Preliminary Dense Reconstruction

Reconstruction in a Rectified System



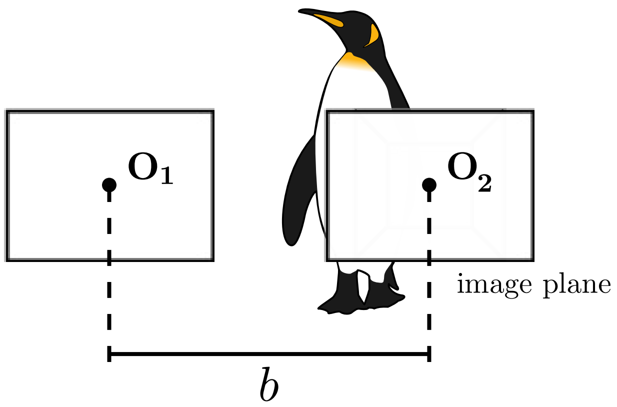
Rectified System

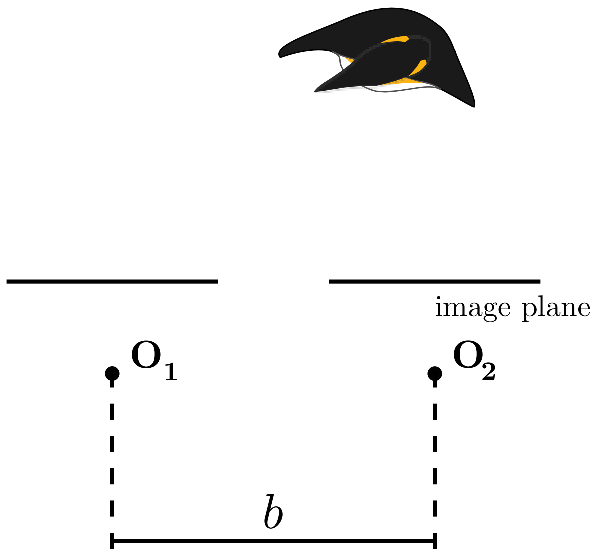
Let's consider 2 cameras with the same orientation but translated by b on the X axis.



Rectified System

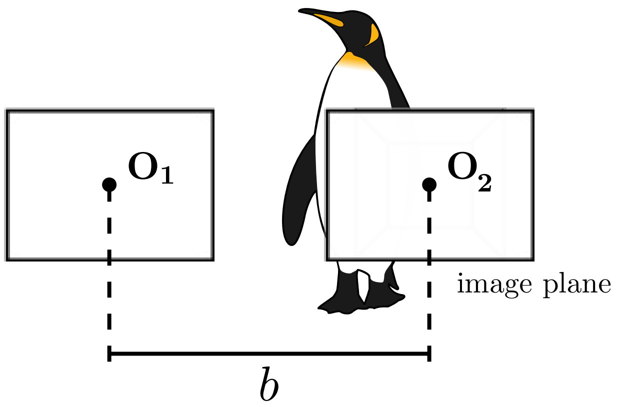
Representation of camera → Pinhole model.

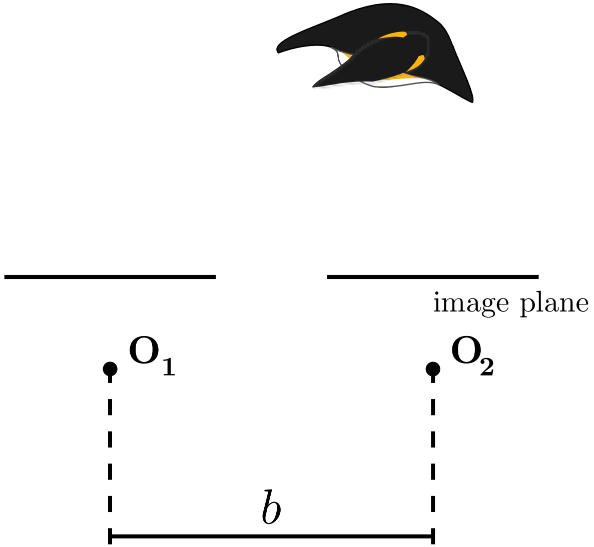




Rectified System

Let's consider the system calibrated. It means the matrices K, R and t were estimated by the processes already shown in previous classes.



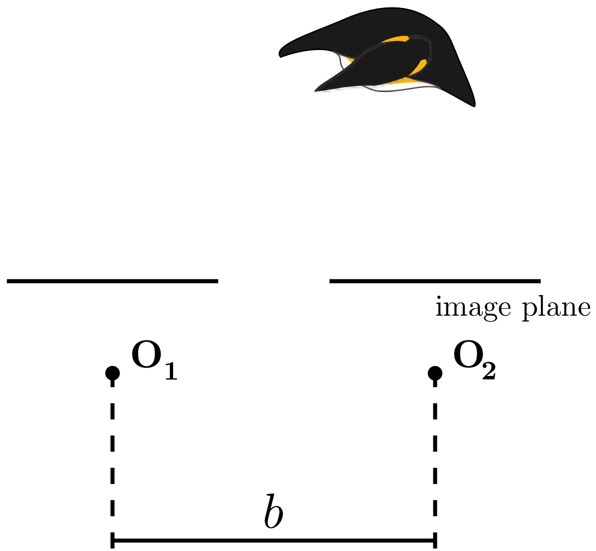


Rectified System

Let's consider the system calibrated. It means the matrices K, R e t were estimated by the processes already shown in previous classes.

$$K = \begin{bmatrix} fs_x & 0 & c_x \\ 0 & fs_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{12} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{t_{12}} = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$



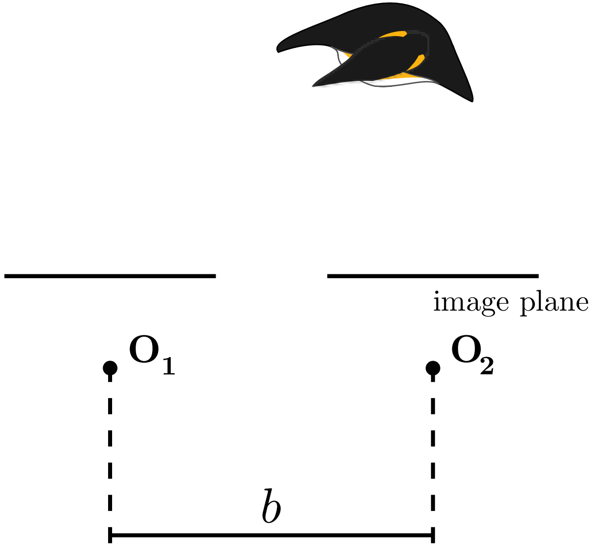
Rectified System

Let's consider the system calibrated. It means the matrices K, R e t were estimated by the processes already shown in previous classes.

$$K = \begin{bmatrix} f s_x & 0 & c_x \\ 0 & f s_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

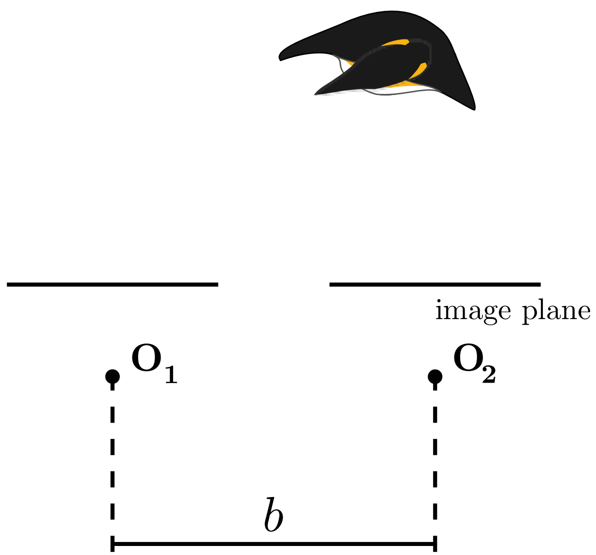
$$\boxed{R_{12} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{t}_{12} = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}}$$

Remember: Transform coordinates from the right camera reference frame (2) to the left camera frame (1).



Rectified System

With the system calibrated, we can project the scene into the image planes.

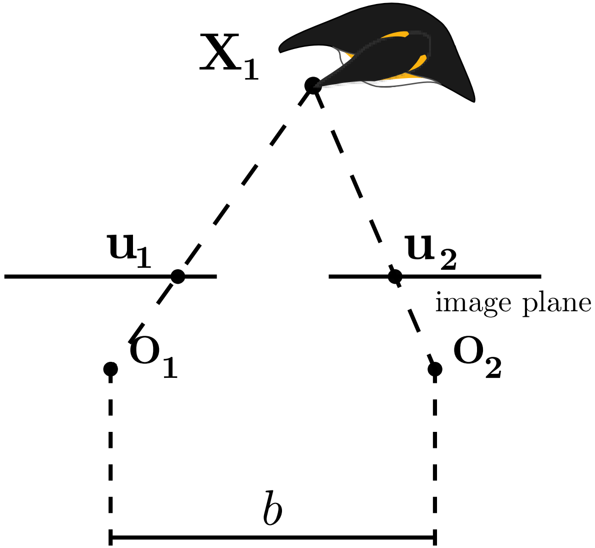


Rectified System

With the system calibrated, we can project the scene into the image planes.

$$\mathbf{X}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \qquad \mathbf{X}_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

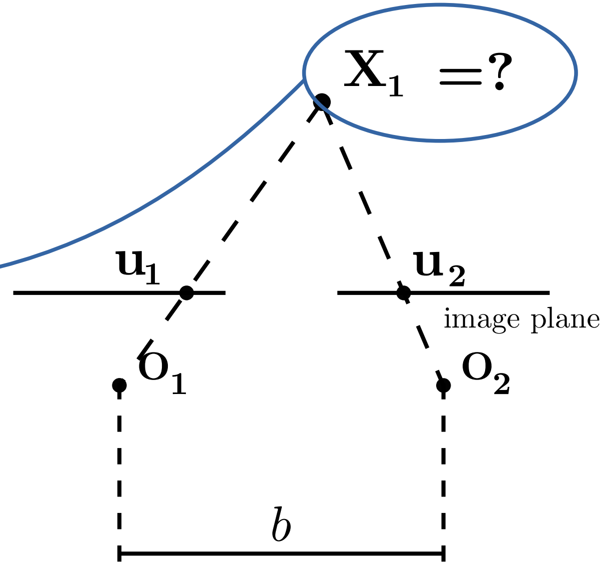
$$\mathbf{u}_1 = \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \qquad \mathbf{u}_2 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}$$



Rectified System

With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

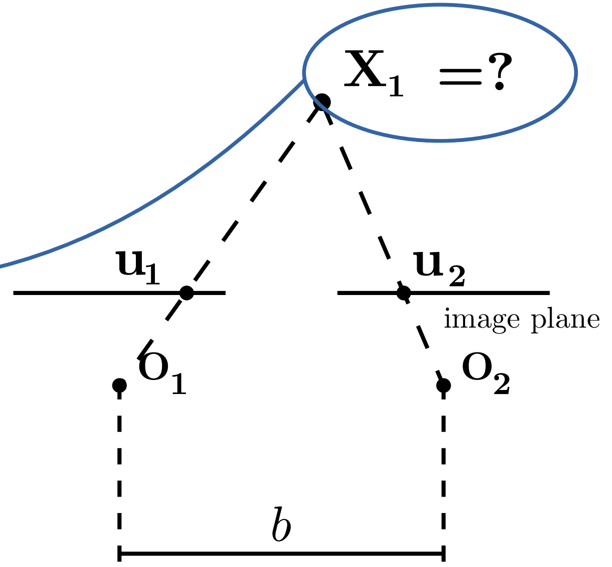
$$\mathbf{X}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$



Rectified System

With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

$$\mathbf{X}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$



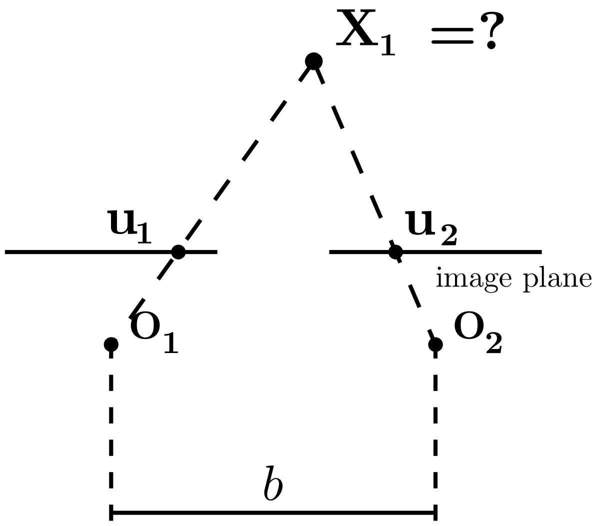
Rectified System

With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

$$\mathbf{X}_1 = R_{12}\mathbf{X}_2 + \mathbf{t}_{12}$$

$$R_{12} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{t}_{12} = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{X}_1 = \mathbf{X}_2 + \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$



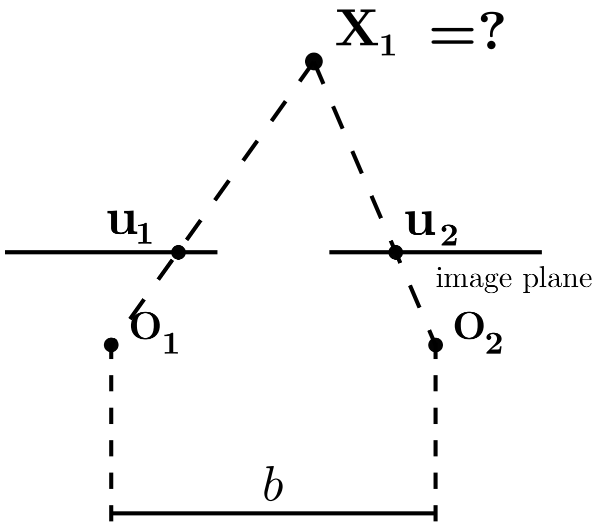
Rectified System

With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

$$\mathbf{X}_1 = \mathbf{X}_2 + \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

$$z_1\,\mathbf{u}_1 = K\mathbf{X}_1$$

$$\mathbf{X}_2 = z_2\,K^{-1}\mathbf{u}_2$$



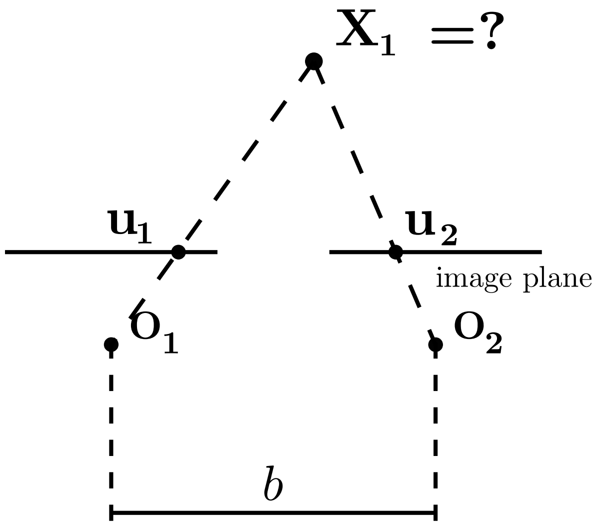
Rectified System

With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

$$\mathbf{X}_1 = \mathbf{X}_2 + \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

$$z_1\, \mathbf{u}_1 \; = K\mathbf{X}_1$$

$$\mathbf{X}_2 \; = z_2\, K^{-1}\mathbf{u}_2$$



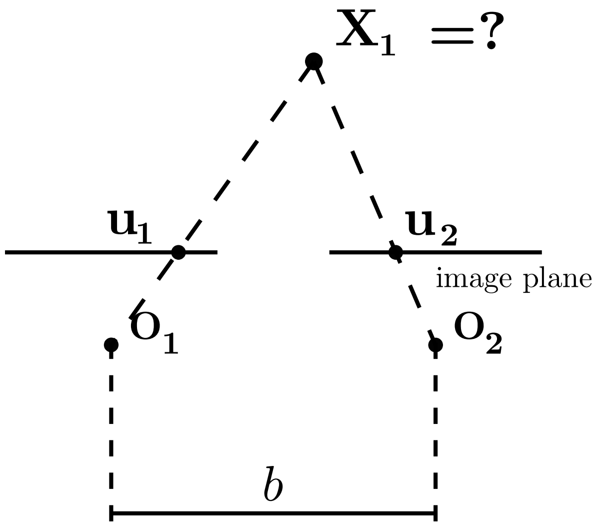
Rectified System

With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

$$\mathbf{X}_1 = \mathbf{X}_2 + \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

$$z_1\,\mathbf{u}_1 = K\mathbf{X}_1$$

$$\mathbf{X}_2 = z_2\,K^{-1}\mathbf{u}_2$$



Rectified System

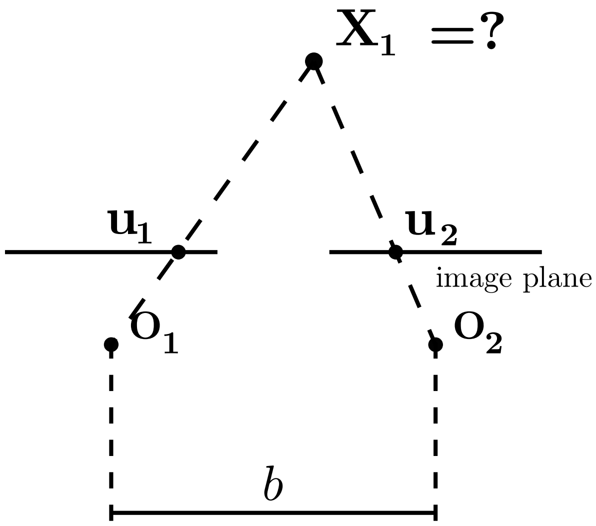
With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

$$\mathbf{X}_1 = \mathbf{X}_2 + \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

$$z_1 \mathbf{u}_1 = K\mathbf{X}_1$$

$$\mathbf{X}_2 = z_2 K^{-1}\mathbf{u}_2$$

$$\boxed{z_1 \mathbf{u}_1 = K\left(z_2 K^{-1}\mathbf{u}_2 + \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}\right)}$$



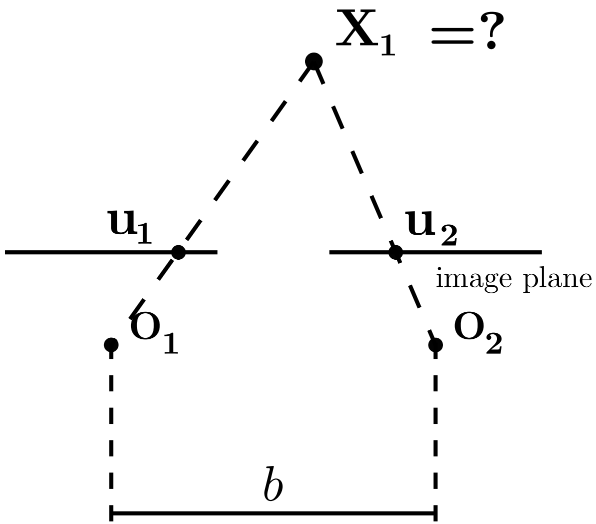
Rectified System

With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

$$z_1\,\mathbf{u}_1 = K\left(z_2\,K^{-1}\mathbf{u}_2 + \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}\right)$$

$$z_1\,\mathbf{u}_1 = z_2\,K K^{-1}\mathbf{u}_2 + \begin{bmatrix} fs_x & 0 & c_x \\ 0 & fs_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

$$\boxed{\begin{bmatrix} z_1\,u_1 \\ z_1\,v_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} z_2\,u_2 \\ z_2\,v_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} fs_x b \\ 0 \\ 0 \end{bmatrix}}$$



Rectified System

With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

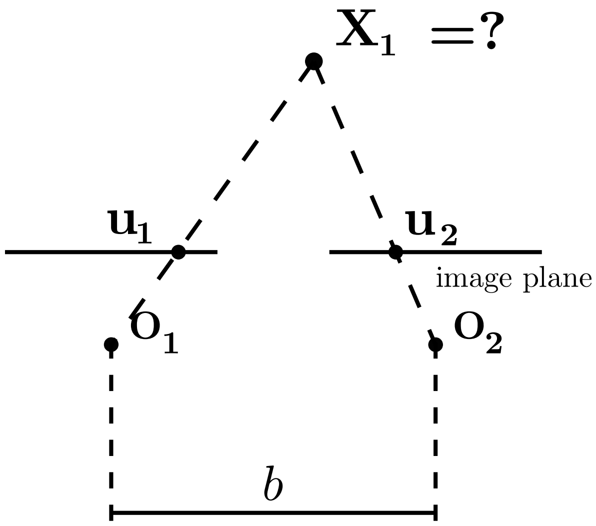
$$\begin{bmatrix} z_1\, u_1 \\ z_1\, v_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} z_2\, u_2 \\ z_2\, v_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} f s_x b \\ 0 \\ 0 \end{bmatrix}$$

$$z_1 = z_2 = z$$

$$v_1 = v_2$$

$$z u_1 = z u_2 + b f s_x$$

$$\boxed{\begin{aligned} u_1 - u_2 &= \frac{b f s_x}{z} = disparity \\ z &= \frac{b f s_x}{u_1 - u_2} \end{aligned}}$$



Rectified System

With this rectified configuration, it's possible to estimate the depth (z coordinate) of the point X_1, which means we can estimate the point X_1.

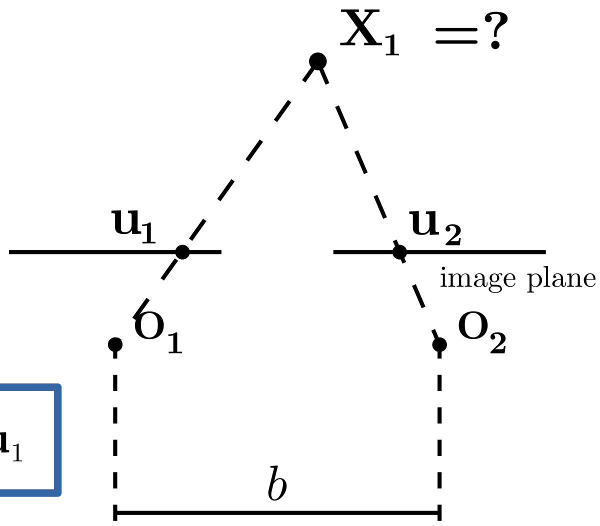
$$\begin{bmatrix} z_1 u_1 \\ z_1 v_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} z_2 u_2 \\ z_2 v_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} f s_x b \\ 0 \\ 0 \end{bmatrix}$$

$$z_1 = z_2 = z$$

$$v_1 = v_2$$

$$z u_1 = z u_2 + b f s_x$$

$$\boxed{u_1 - u_2 = \frac{b f s_x}{z} = disparity \qquad z = \frac{b f s_x}{u_1 - u_2}} \rightarrow \boxed{\mathbf{X}_1 = z_1 K^{-1} \mathbf{u}_1}$$



Important!!!

- There is almost no disparity for points far away from the camera. Therefore, it is very hard to retrieve the depth of such points.

$$z = \frac{bfs_x}{u_1 - u_2}$$

Important!!!

- There is almost no disparity for points far away from the camera. Therefore, it is very hard to retrieve the depth of such points.

$$z = \frac{b f s_x}{u_1 - u_2}$$

Very small

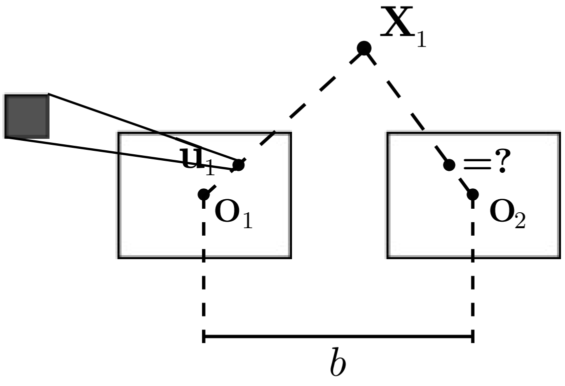
Example

- Example 1:
  - Near and far points, their disparities and depths

$$\boxed{z = \frac{b f s_x}{u_1 - u_2}}$$

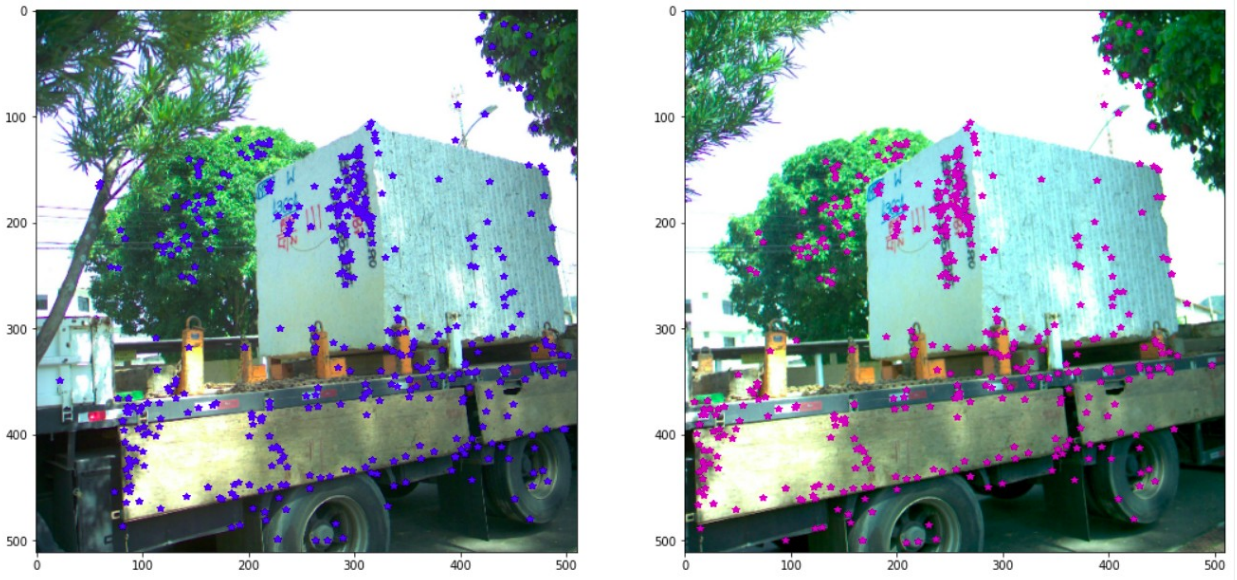
Ok! But, usually in computer vision we have the images, but we don’t have the correspondences! What should we do?

Finding Correspondences



Searching for Correspondences

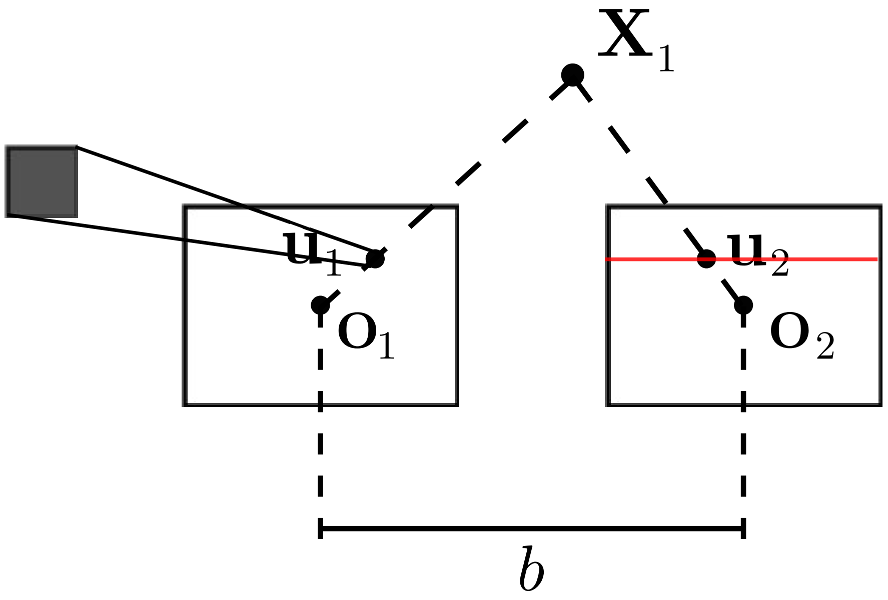
- We can apply feature detectors and matching methods to obtain the correspondences.



Searching for Correspondences

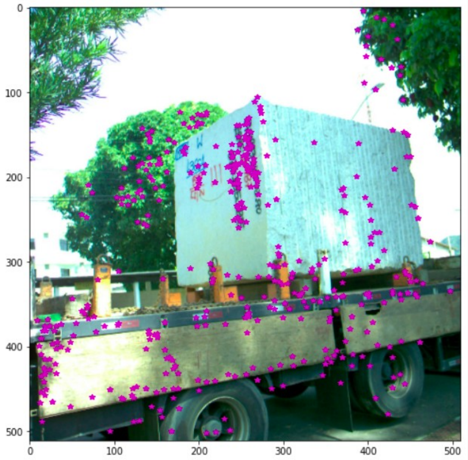
- But for a Rectified System, it is easier!
- We can search for correspondences by looking only on the epipolar line.
- And the epipolar lines are parallel!

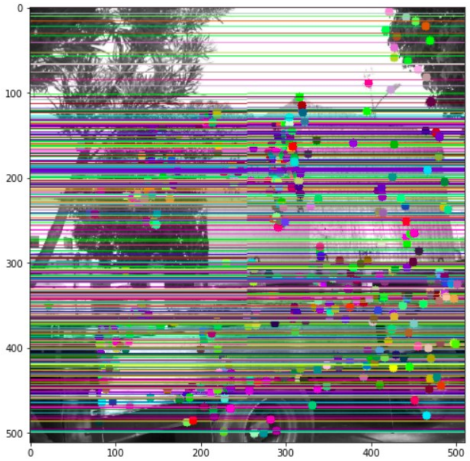
$$v_1 = v_2$$

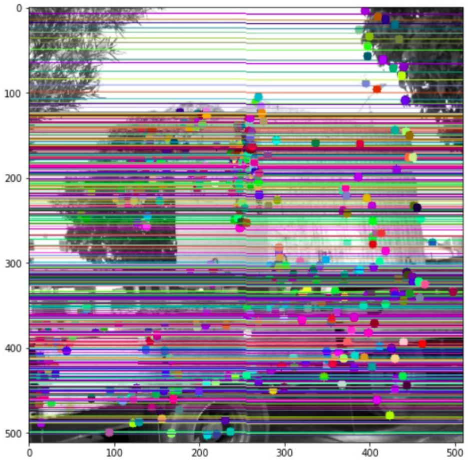


Searching for Correspondences

- In this case, we can obtain many correspondences from the parallel lines in both images!!!

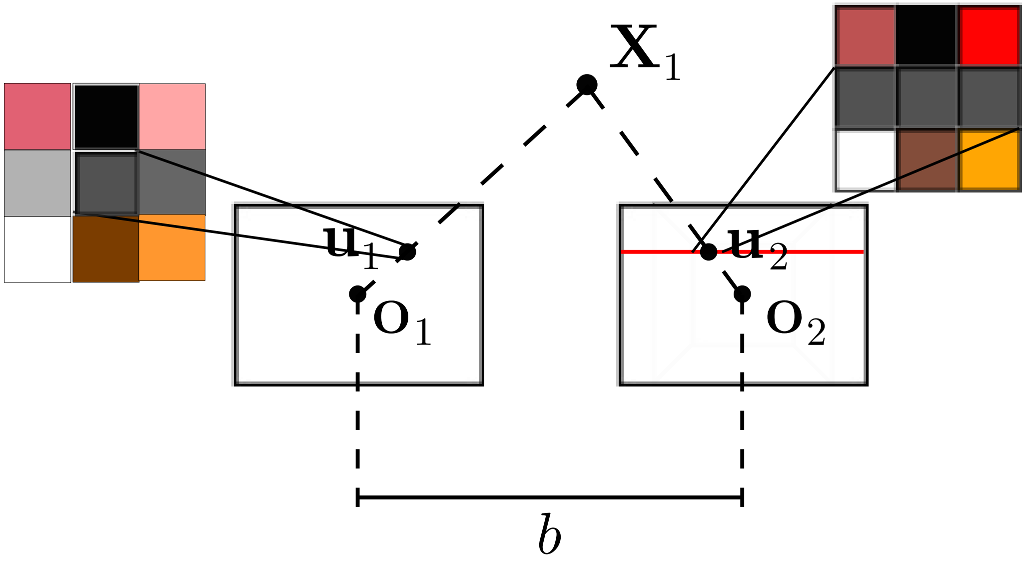




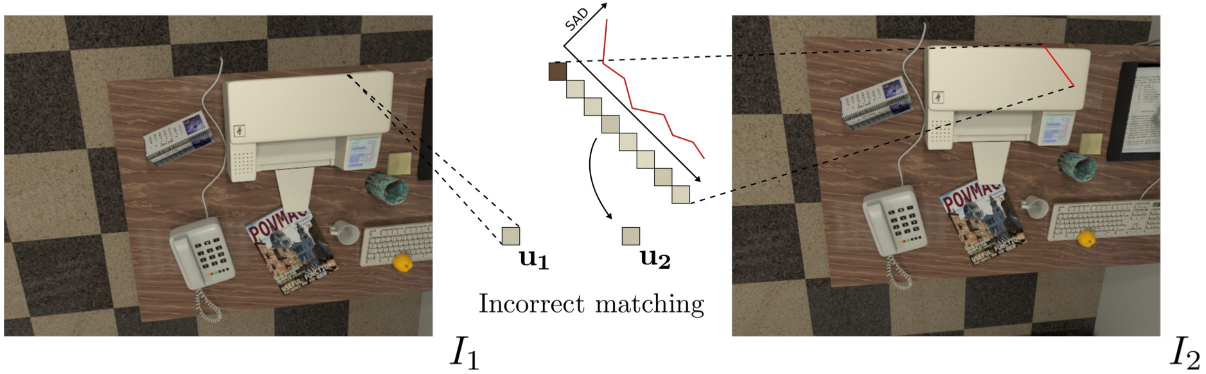


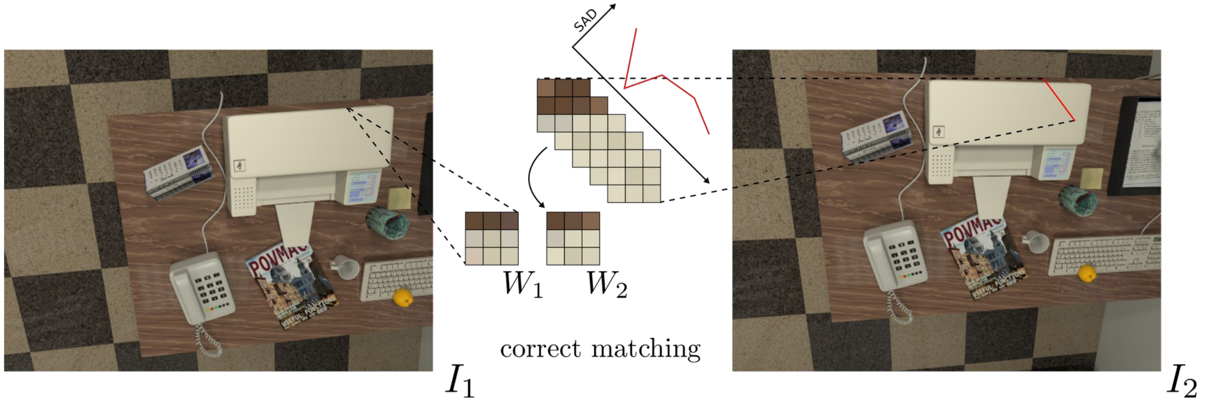
Searching for Correspondences

- For a dense reconstruction, we need dense matching.
- We can use a window around the pixel to search for a correspondence on the epipolar line of the other image!



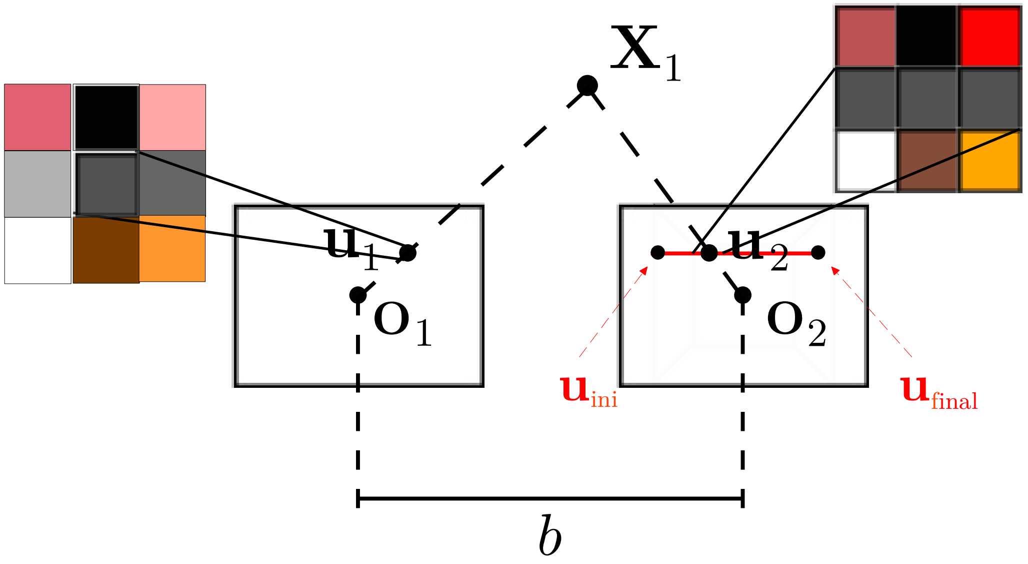
Similarity metrics (pixel vs window)





Searching for Correspondences

- We can also restrain the epipolar line to an initial and a final points.



$$z = \frac{bfs_x}{u_1 - u_2}$$

$$u_{ini} = u_1 - \frac{bfs_x}{z_{ini}}$$

$$u_{final} = u_1 - \frac{bfs_x}{z_{final}}$$

Examples

- Example 2
  - Epipolar line initial and final points in the right image, errors and discretization issue

$$\boxed{u_{ini} = u_1 - \frac{bfs_x}{z_{ini}}}$$

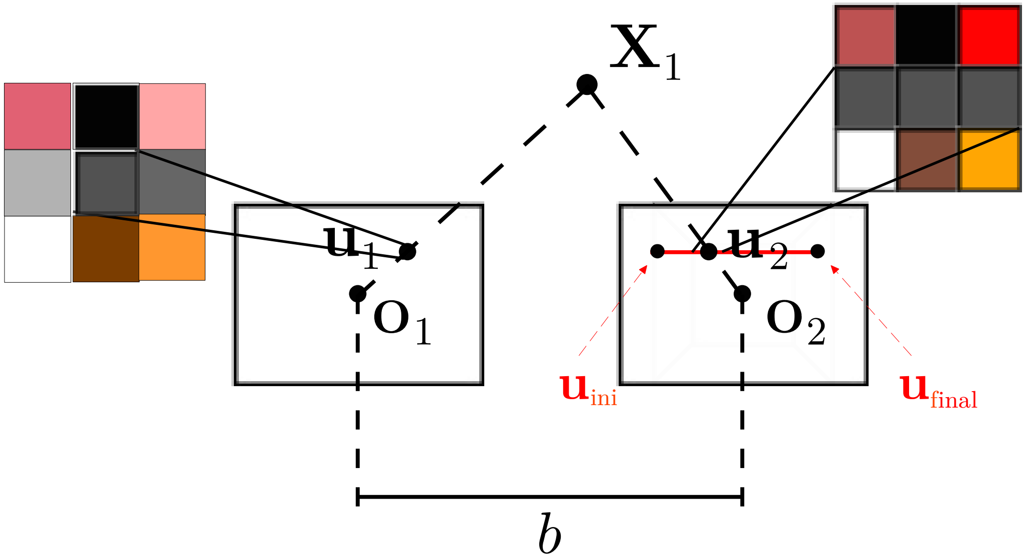
$$\boxed{u_{final} = u_1 - \frac{bfs_x}{z_{final}}}$$

- Example 3
  - Measuring distances

$$\boxed{z = \frac{bfs_x}{u_1 - u_2}}$$

Searching for Correspondences

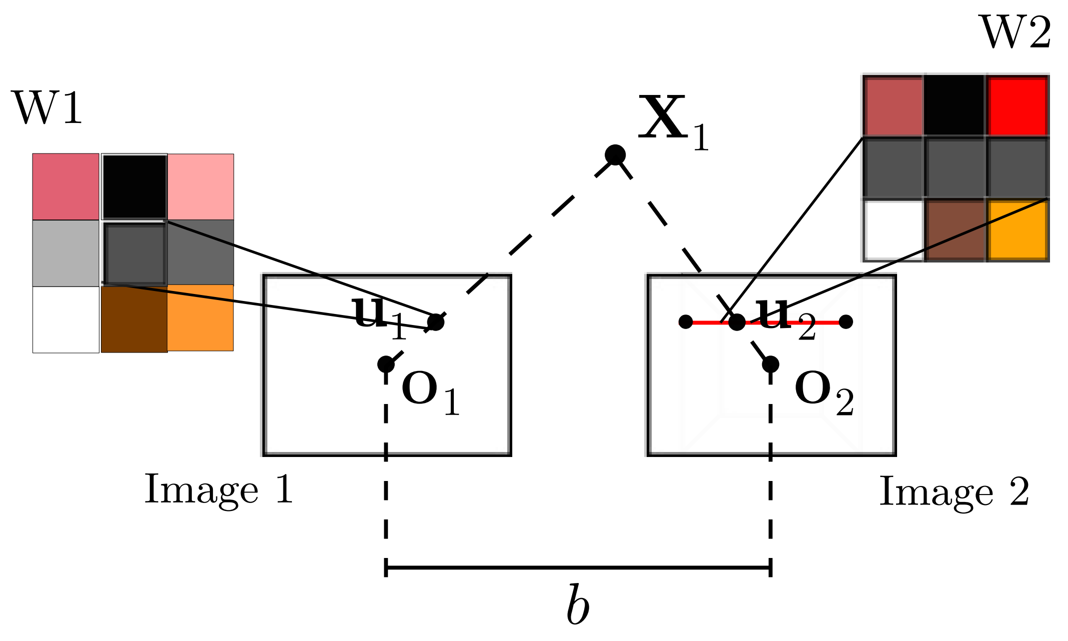
- Find a way to quantify the similarity between windows.



Use similarity metrics to compare windows.

Similarity metrics

- Window notation: W1 (n x n) and W2 (n x n)



Similarity metrics

- SAD (Sum of Absolute Differences)

$$SAD(W_1, W_2) = \sum |W_1 - W_2|$$

- SSD (Sum of Square Differences)

$$SSD(W_1, W_2) = \sum (W_1 - W_2)^2$$

- NCC (Normalized Cross Correlation)

$$NCC(W_1, W_2) = \frac{\sum (W_1 - \overline{W}_1) \cdot (W_2 - \overline{W}_2)}{\sqrt{\sum (W_1 - \overline{W}_1)^2 \cdot \sum (W_2 - \overline{W}_2)^2}}$$

Similarity metrics

- ZSAD (Zero mean Sum of Absolute Differences)

$$ZSAD(W_1, W_2) = \sum |(W_1 - \overline{W}_1) - (W_2 - \overline{W}_2)|$$

- ZSSD (Zero mean Sum of Square Differences)

$$ZSSD(W_1, W_2) = \sum ((W_1 - \overline{W}_1) - (W_2 - \overline{W}_2))^2$$

- NCC (Normalized Cross Correlation)

$$NCC(W_1, W_2) = \frac{\sum (W_1 - \overline{W}_1) \cdot (W_2 - \overline{W}_2)}{\sqrt{\sum (W_1 - \overline{W}_1)^2 \cdot \sum (W_2 - \overline{W}_2)^2}}$$

Similarity metrics

- NCC (Normalized Cross Correlation)

$$NCC(W_1, W_2) = \frac{\sum(W_1 - \overline{W}_1)\cdot(W_2 - \overline{W}_2)}{\sqrt{\sum(W_1 - \overline{W}_1)^2 \cdot \sum(W_2 - \overline{W}_2)^2}}$$

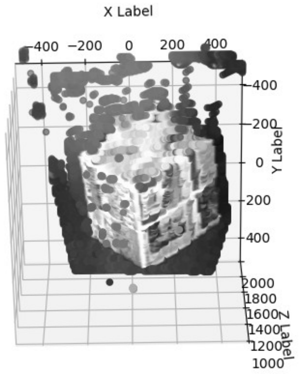
- FNCC (Fast Normalized Cross Correlation)

$$FNCC(W_1, W_2) = \frac{\sum W_1 \cdot \left(W_2 - \overline{W}_2\right)}{\sqrt{\sum(W_1 - \overline{W}_1)^2 \cdot \left(\sum W_2^2 - \frac{1}{n^2} \sum_{l=0}^{2} \left(\sum_{(j,k)\in W_2} W_2(k,j,l) \right)^2 \right)}}$$

Example

- Example 4:
  - Search the best match using the ZSAD, ZSSD and NCC metrics
  - Compute the depth

Preliminary Dense Reconstruction



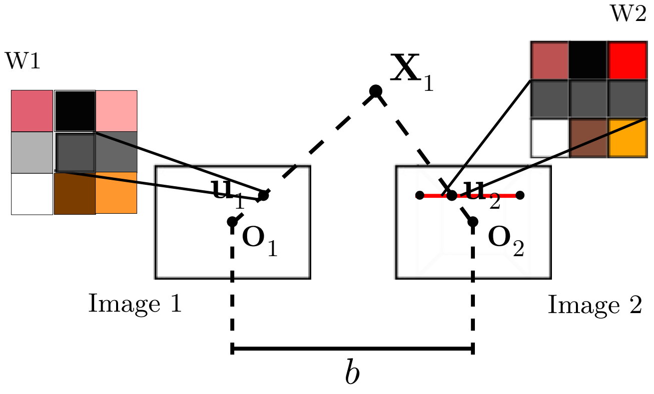
We have already seen that ...

- We can use a sliding window in the epipolar line of the other image and find the best match applying similarity metrics between windows;
- We can estimate the depth z and therefore the 3D point for each correspondent pair of points.

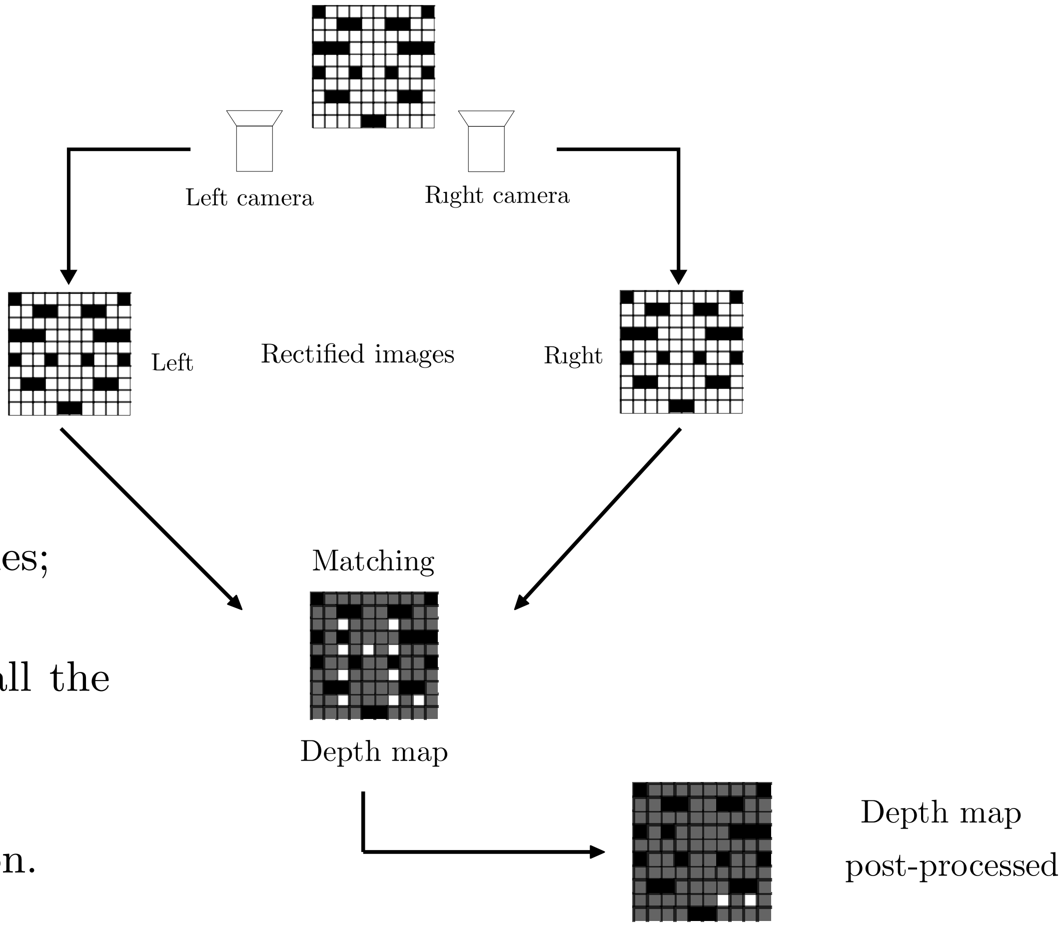
$$z = \frac{b f s_x}{u_1 - u_2}$$

$$z_1 = z_2 = z$$

$$\mathbf{X}_1 = z_1 K^{-1} \mathbf{u}_1$$



Now... To obtain a preliminary Reconstruction from a Rectified System

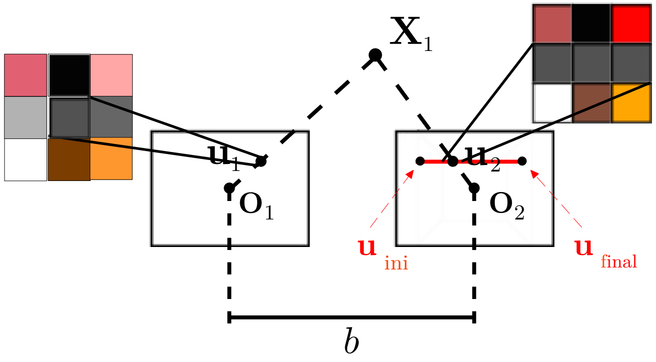


- Look for correspondences on the epipolar lines;
- Compute a depth map or disparity map;
- Use such map to estimate the depth z for all the obtained matches;
- Retrieve the 3D coordinates of the points;
- Post-processing to improve the reconstruction.

Reconstruction from a Rectified System

Goal: find $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, since $(\boldsymbol{u}_1, \boldsymbol{u}_2) \to z$.

- For each point in first image define the initial and final points in the epipolar lines of the second image to search for the correspondent points.

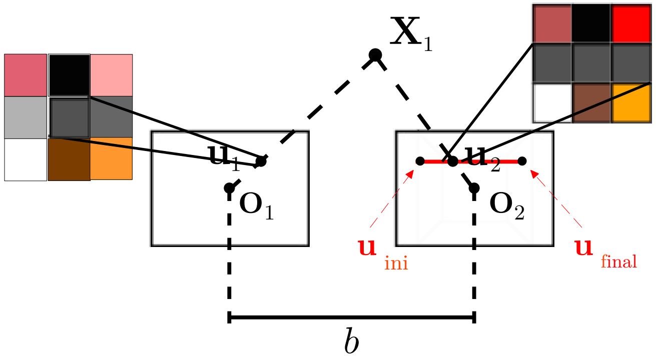


$$u_{ini} = u_1 - \frac{bfs_x}{z_{ini}}$$

$$u_{final} = u_1 - \frac{bfs_x}{z_{final}}$$

Reconstruction from a Rectified System

Goal: find $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, since $(\boldsymbol{u}_1, \boldsymbol{u}_2) \rightarrow z$.



- For each $\boldsymbol{u}_1$:

$$u_2 = \text{matching}(u_{\text{ini}}, u_{\text{final}}, v_1, I_1, I_2)$$

- Then:

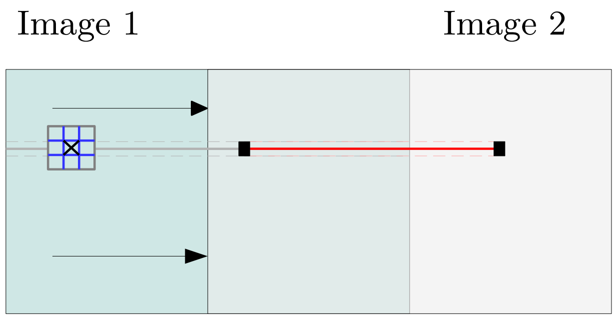
$$z = \frac{b f s_x}{u_1 - u_2}$$

Depth Map or Disparity Map

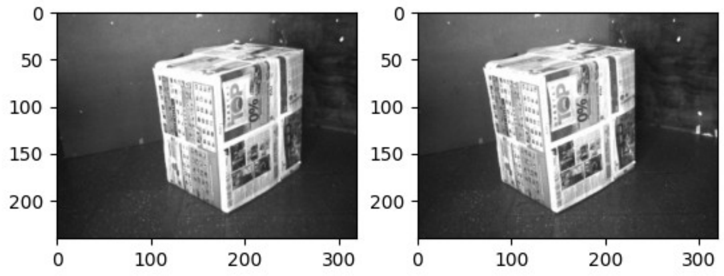
To build a Disparity Map:

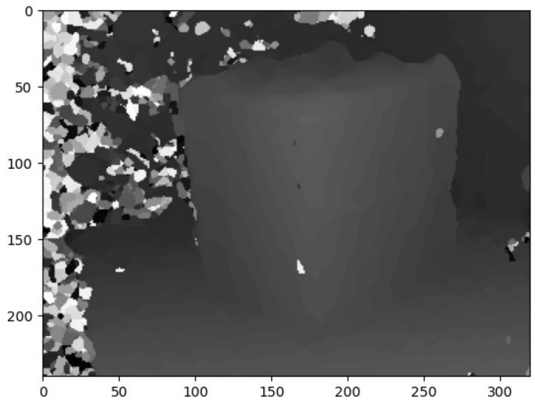
- Try a range of displacements between the rectified images and record the best displacement for each pixel by selecting the one with the best score according to NCC of the local image neighborhood.
- This is sometimes called plane sweeping, since each displacement step corresponds to a plane at some depth.
- It is not the best approach, but is a simple method that usually gives decent results.

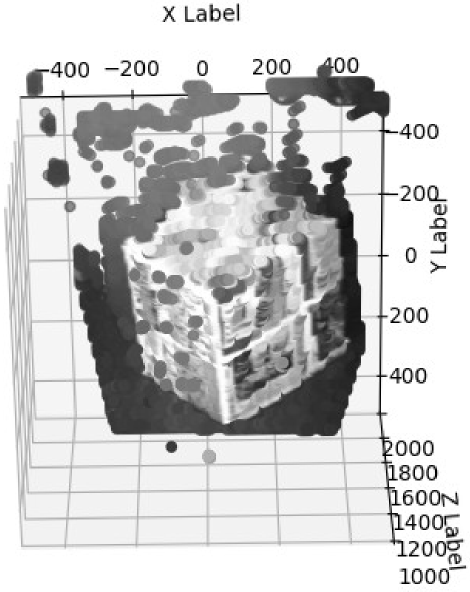
Plane Sweeping



Depth Map or Disparity Map

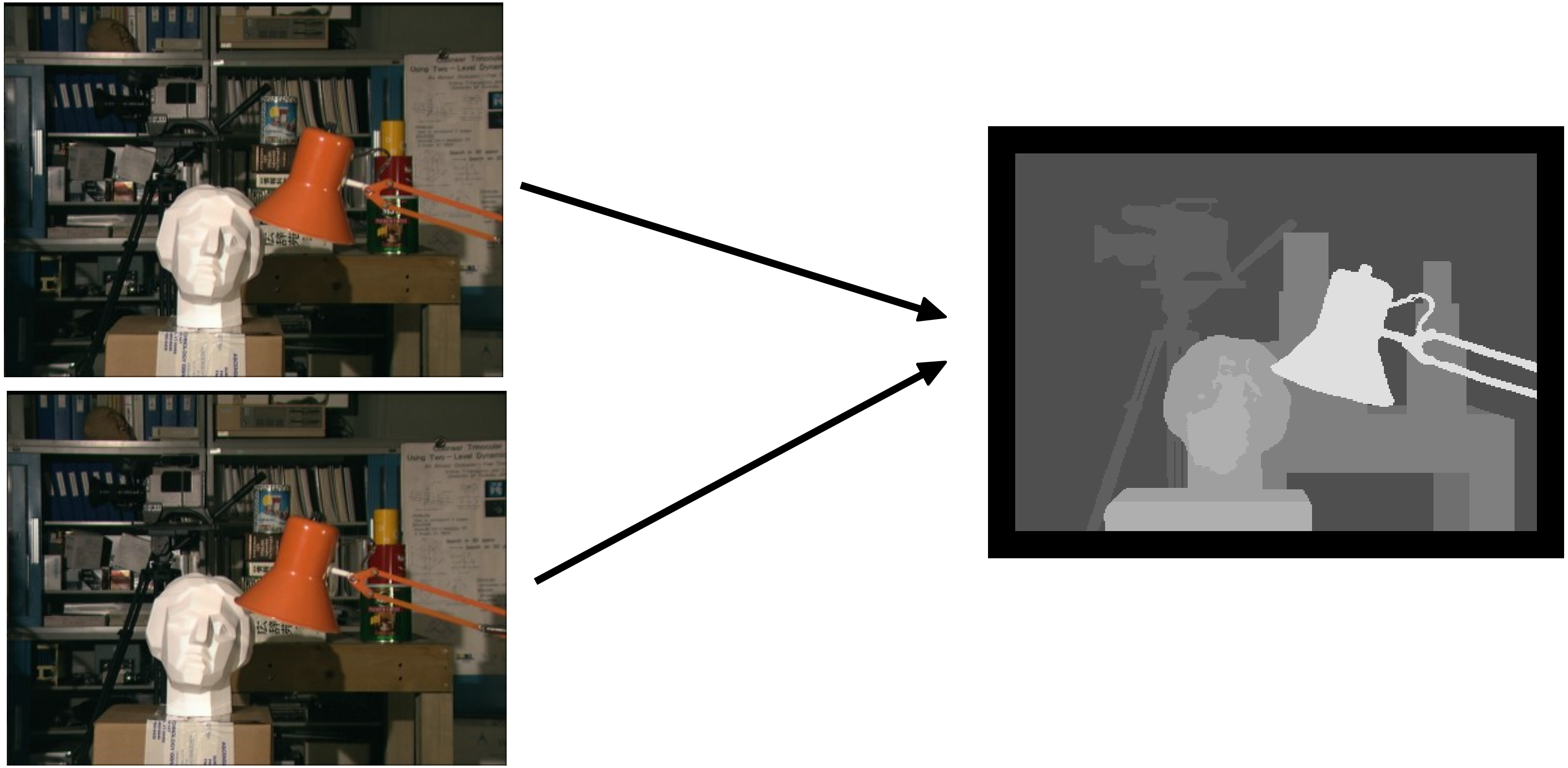






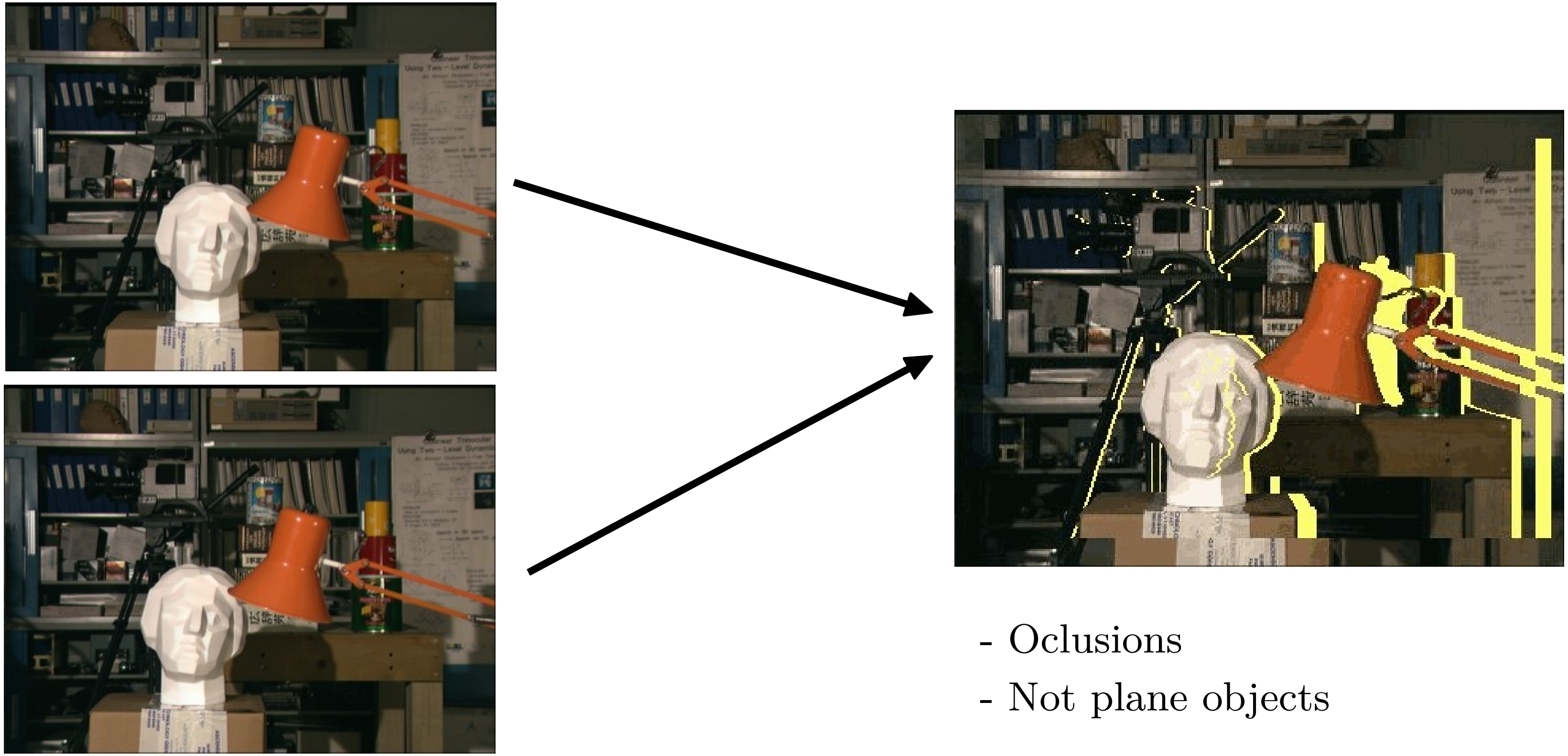
Depth Map or Disparity Map

Tsukuba



Some Problems

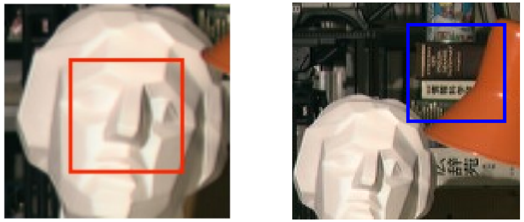
Tsukuba

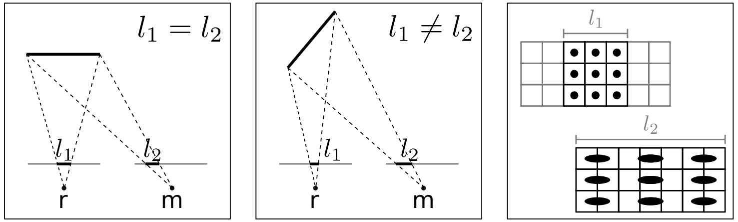


- Oclusions
- Not plane objects

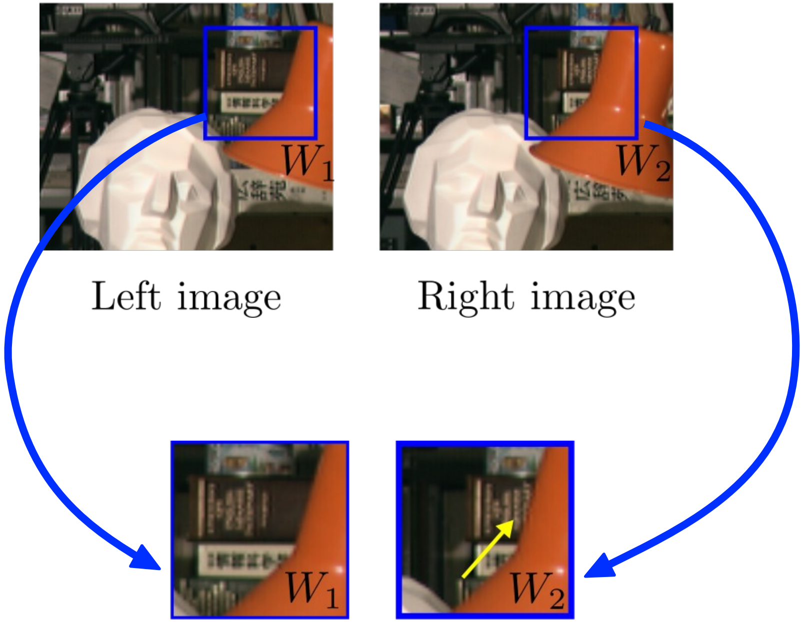
Some Problems

- We consider that the window contains a plane in the world and that this surface is parallel to the image plane.





- Oclusions.

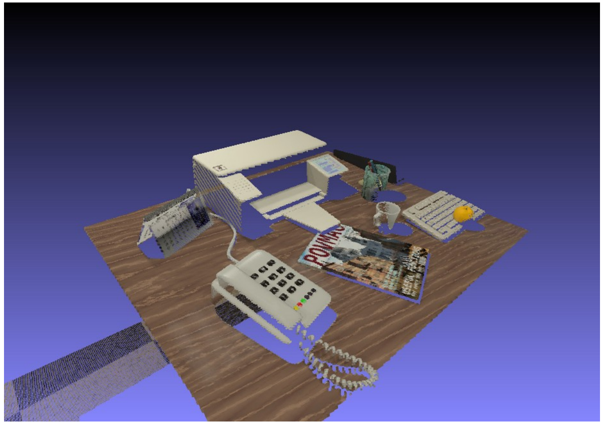


Background is misaligned

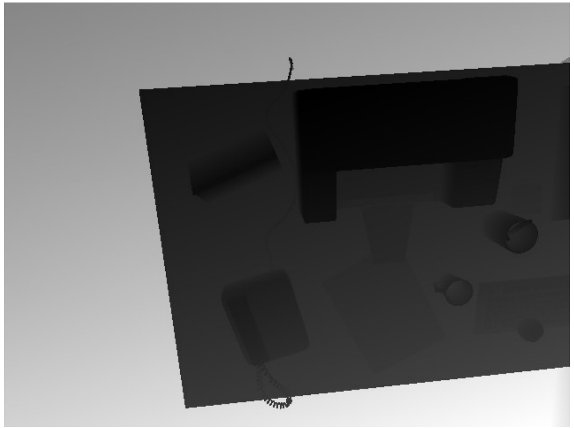
Post-processing

- We can improve the preliminary 3D reconstruction by applying some post-processing.

- As an example we will considered the work done by Leonardo de Assis Silva.
- Consider a scene with the following point cloud and depth map:



Point cloud

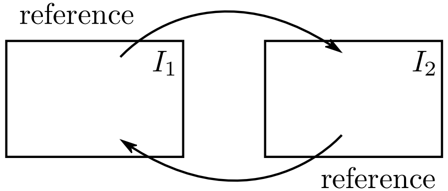


Depth map

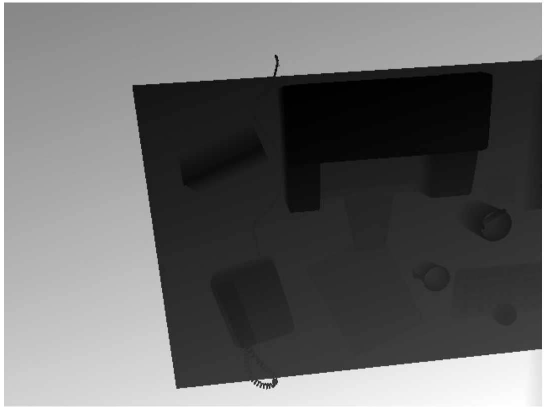
Ground truth

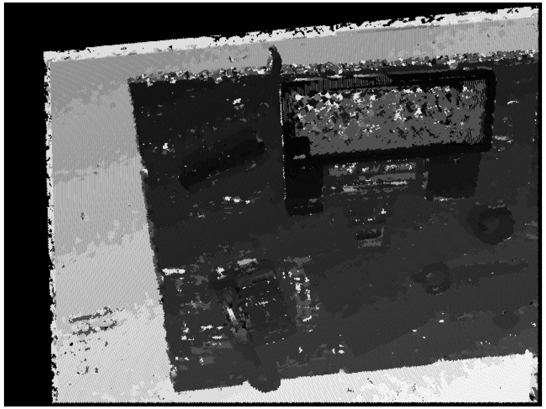
Post-processing

LRCheck (Left Right Check)

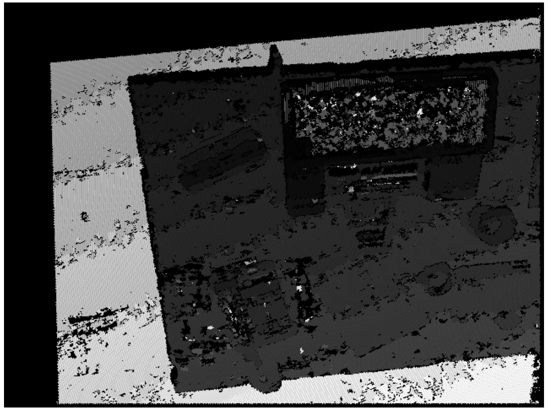


Ground truth





LRCheck

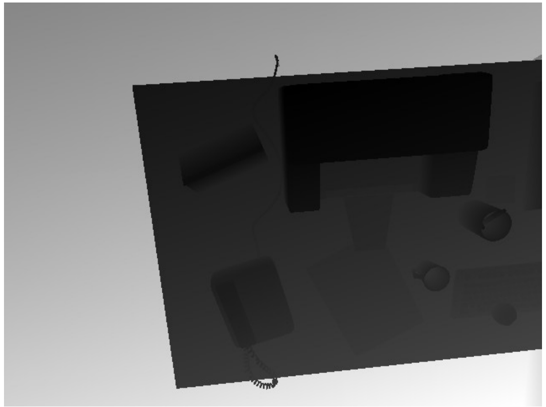


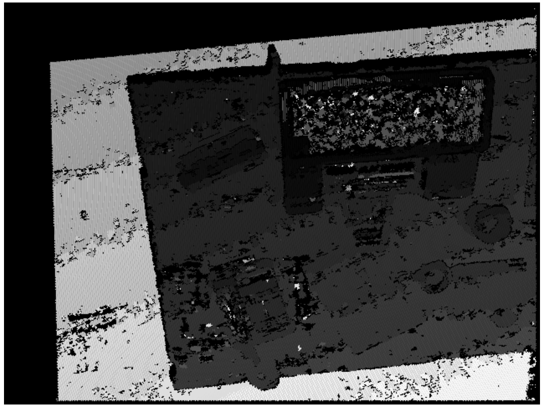
Post-processing

Suspixels

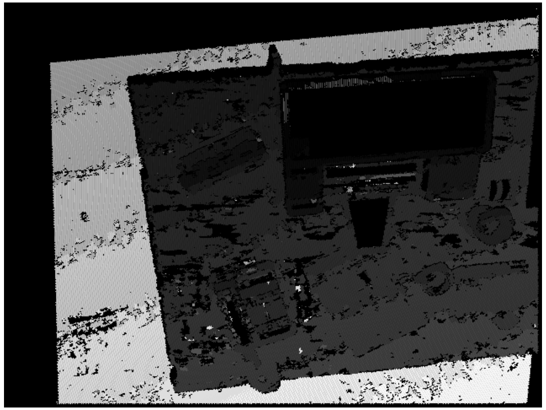
$$Var(W) = \frac{1}{n} \sum_{j=0}^{n^2-1} |w_j - \overline{W}|$$

Ground truth





Eliminating
Suspixels

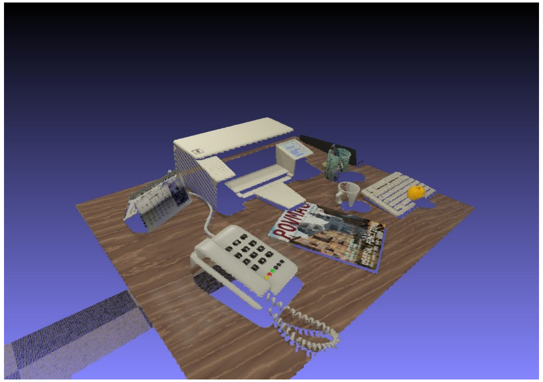


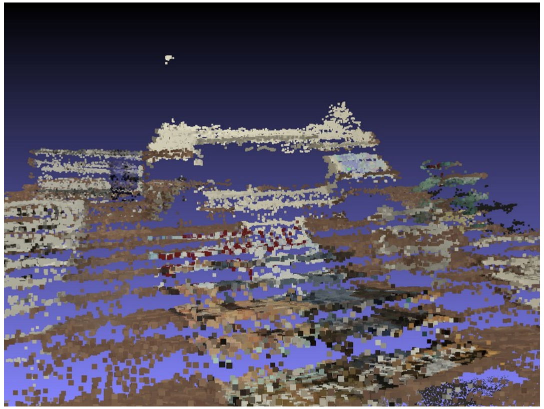
Post-processing

Suspixels

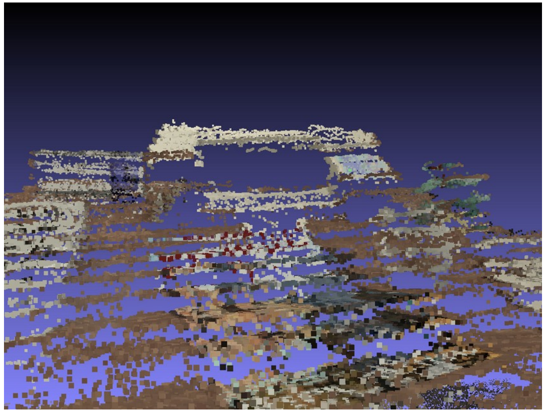
$$Var(W) = \frac{1}{n}\sum_{j=0}^{n^2-1} |w_j - \overline{W}|$$

Ground truth



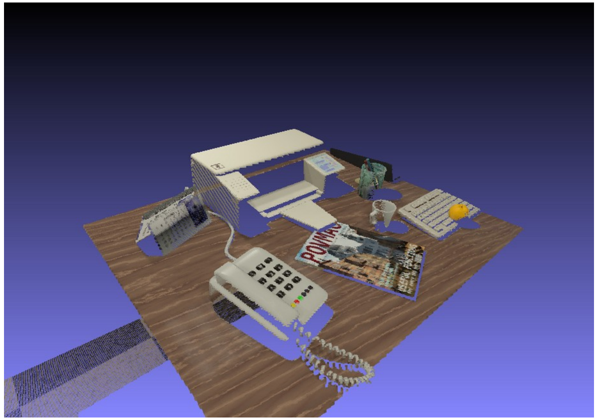


Eliminating Suspixels

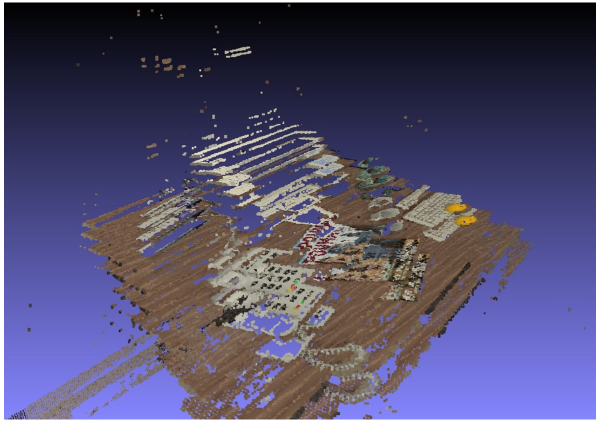


Post-processing

Discrete depth



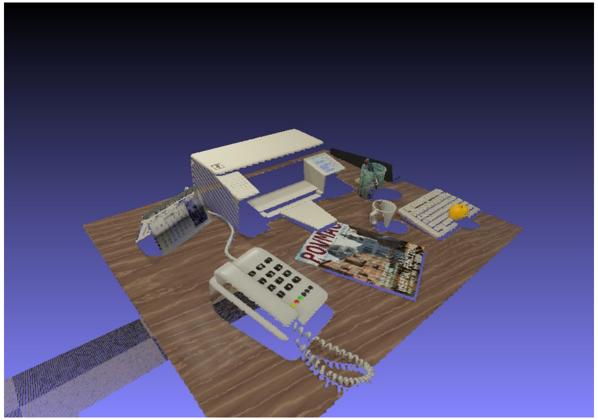
Ground truth



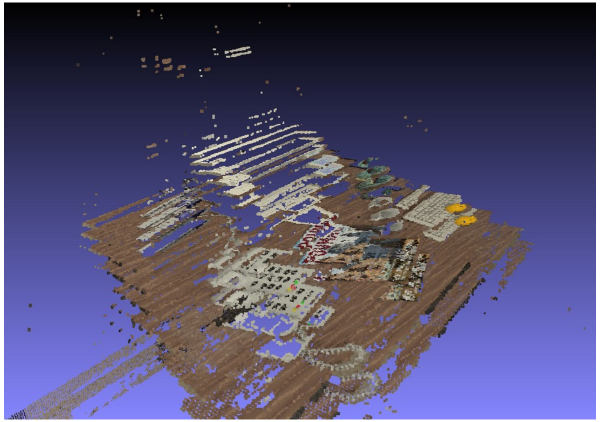
Steps discretization

Post-processing

Discrete depth

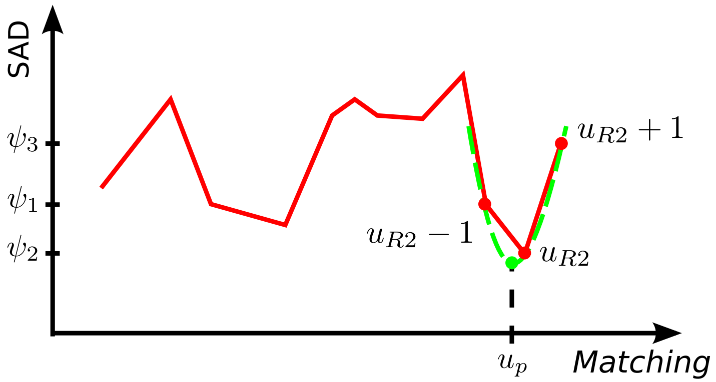


Ground truth



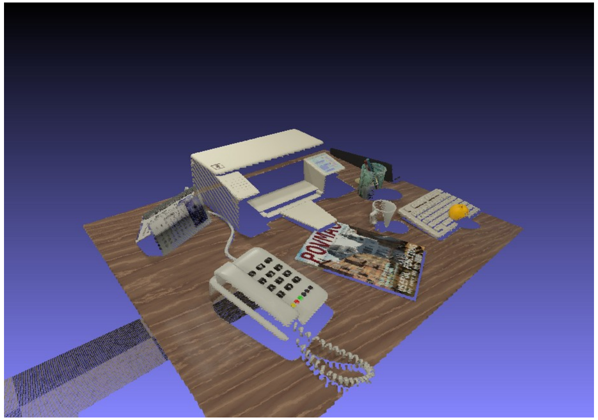
Steps discretization

Parabole approximation

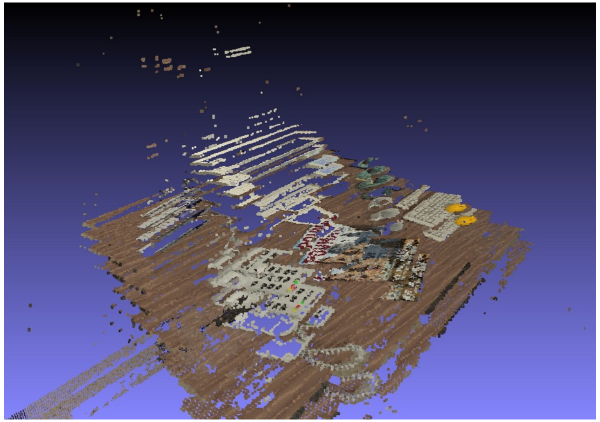


Post-processing

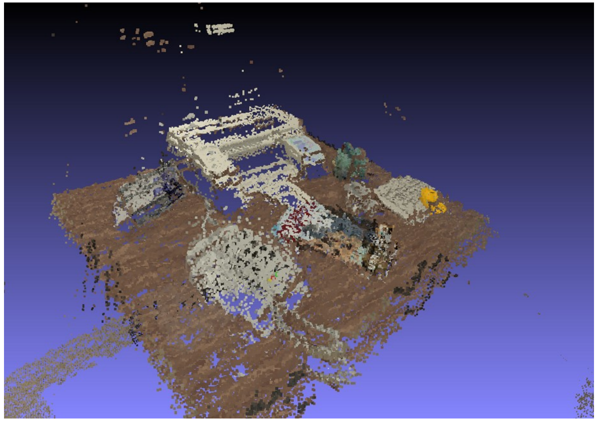
Discrete depth



Ground truth



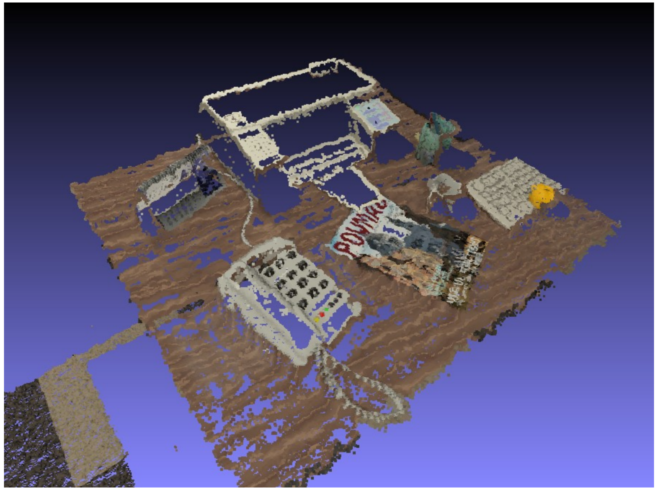
Steps discretization



After parabole approx.

Post-processing

Spatial Filtering

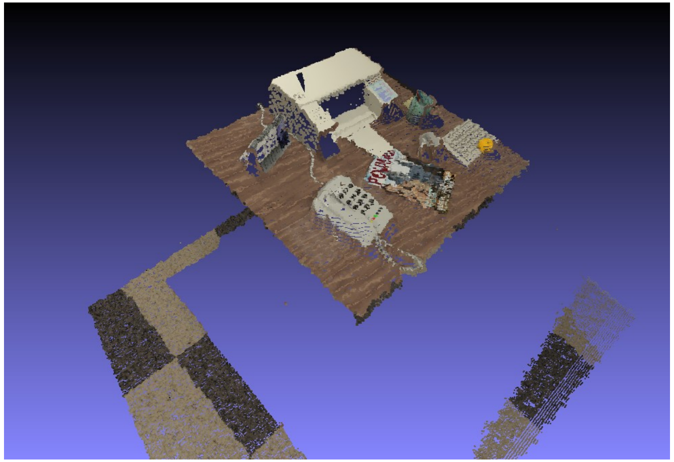


Point cloud

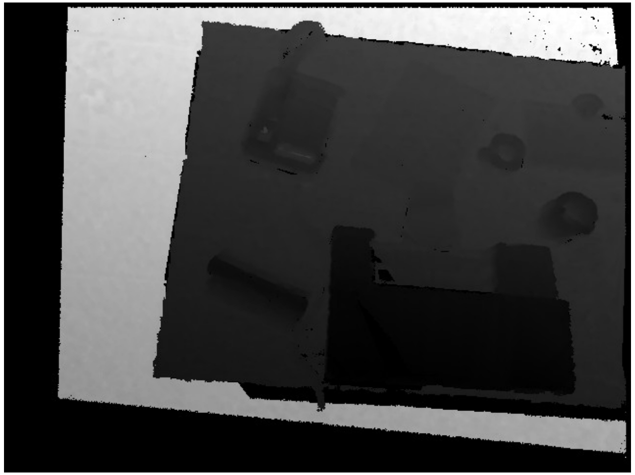
Depth map

Post-processing

Densifying a point cloud using homogeneous areas

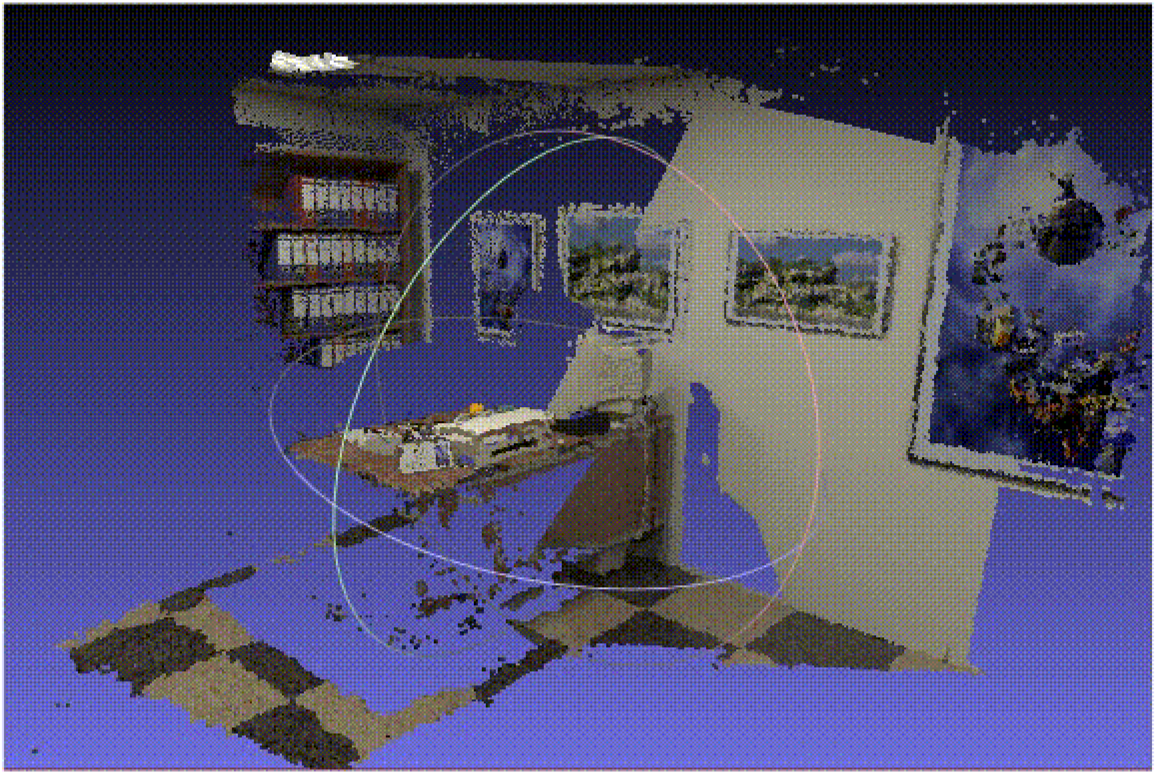


Point cloud

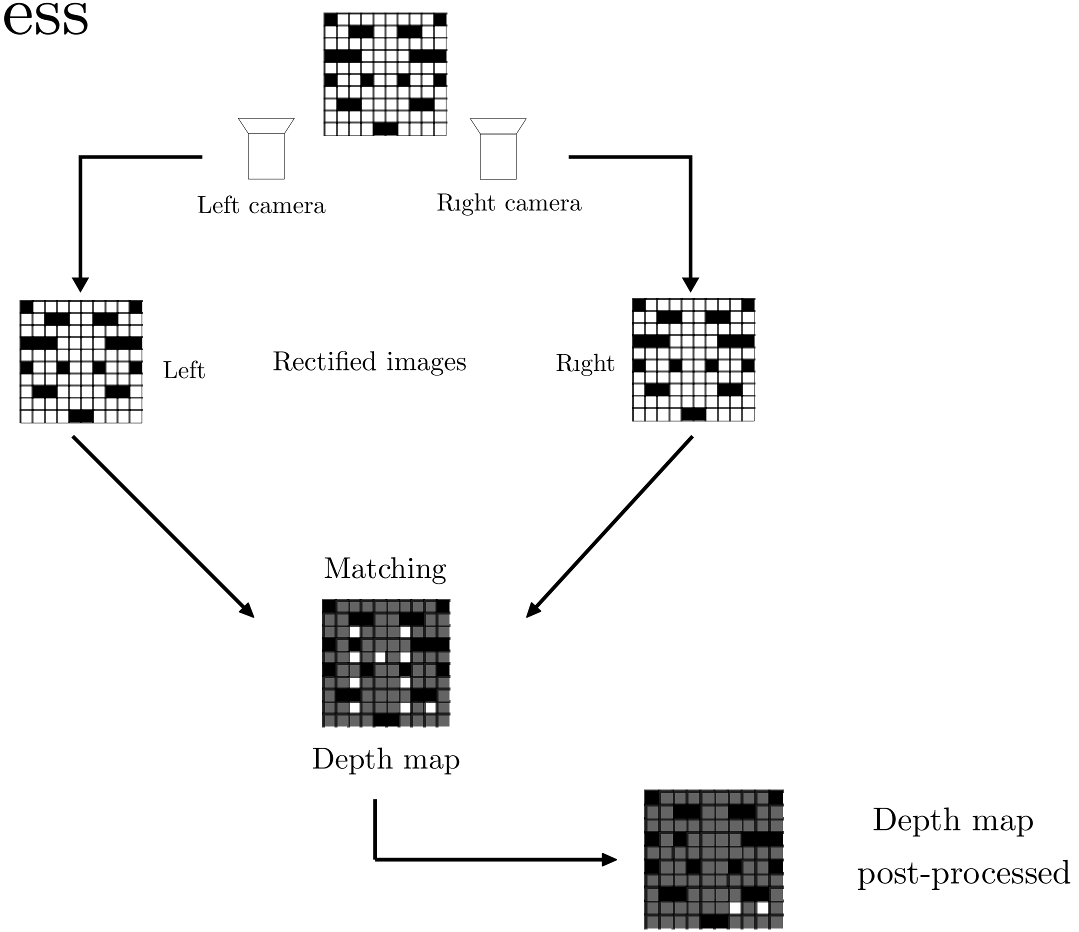


Depth map

Dense 3D Reconstruction



3D Reconstruction process



Credits

- Richard Hartley and Andrew Zisserman. **Multiple View Geometry in Computer Vision,** 2003**.**
- N. Navab. **3D Computer Vision II Winter Term 2010/2011**.
- R. Ait-Jellal, A. Zell. **A fast dense stereo matching algorithm with an application to 3d occupancy mapping using quadrocopters**. IEEE, 2015.
- Leonardo de Assis Silva. **Aproximação Planar por Partes Para Reconstrução 3D Densa.** Dissertação de Mestrado. Programa de Pós-Graduação em Enegnharia Elátrica, UFES, 2016.