3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

ABSTRACT

This project exposes how to get triangulation with the homogeneous algebraic method (DLT) to recover 3D points. Also, depth map computation by some local methods (SSD, NCC and Bilateral weights) with stereo rectified images are introduced.

A base code and functions are provided. All the code is attached parallel to this document.

Code split in 3 files: **Lab4.m , triangulate.m, stereo_computation.m**

Triangulation

When provided the Euclidean coordinates of two matching points x and x' in two different images, the homogeneous algebraic method (DLT) can be used to perform a triangulation for a 3D point X. Assume that two camera matrices P, P' and the size of images are known, the following formula is the way to get the X:

$$\mathbf{x} \equiv P\mathbf{X} \text{ and } \mathbf{x}' \equiv P'\mathbf{X}.$$

The problem also can be written as:

$$\mathbf{AX} = 0,$$

$$\mathbf{A} = \begin{pmatrix} x\mathbf{p}^{3T} - \mathbf{p}^{1T} \\ y\mathbf{p}^{3T} - \mathbf{p}^{2T} \\ x'\mathbf{p}'^{3T} - \mathbf{p}'^{1T} \\ y'\mathbf{p}'^{3T} - \mathbf{p}'^{2T} \end{pmatrix}$$

By using the DLT method, the singular vector associated to the minimum singular value of A.

- function 'triangulate'
  - ✓ **Input**: matched-pair points x, x' from two images, the two camera matrices P, P' and the size of image.
  - ✓ **Output**: the 3D point X.

In this function, the coordinates of points x and x' need to be transformed to the homogeneous coordinates. The matrix of A is created and applied with SVD. The solution of X is the last column of V ([U D V] = svd(A)).

Camera matrix

As mentioned previously, two camera matrices P and P' are need to perform a triangulation. These two camera matrices are related to an intrinsic parameter K and two extrinsic parameters R (rotation) and t (translation).

Assume that F and K are known, the essential matrix E is easy to be obtained by this formula:

$$\mathrm{E} = \mathrm{K}'^{\mathrm{T}}\mathrm{FK}.$$

3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

After extracting keypoints and applying Ransac, the matching points are got as below. Also, the fundamental matrix F can be computed.

Matched keypoints before Ransac

Matched keypoints after Ransac

For a given essential matrix E, known that

$$E = UDV^T = U\begin{bmatrix}1&0&0\\0&1&0\\0&0&0\end{bmatrix}V^T$$

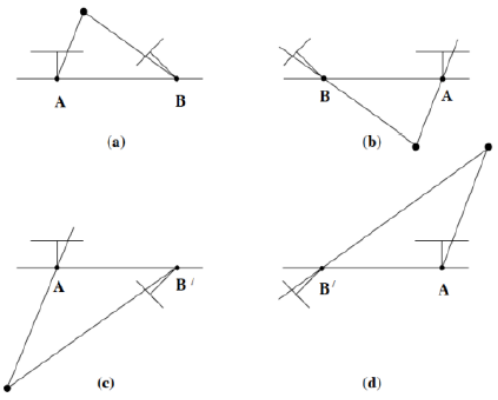
and

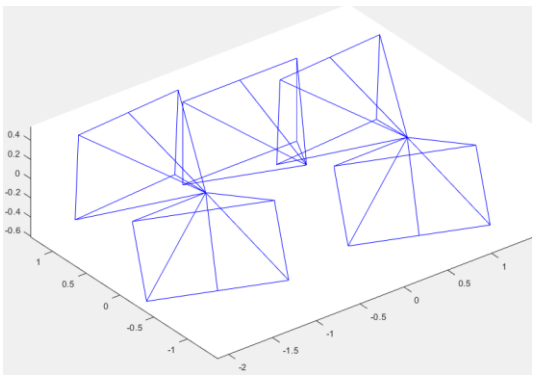
$$E = [T'_\times]R = U\begin{bmatrix}1&0&0\\0&1&0\\0&0&0\end{bmatrix}V^T \quad W = \begin{bmatrix}0&-1&0\\1&0&0\\0&0&1\end{bmatrix}$$

where T' is the last column of U. There are four possible choices for the second camera matrix P', namely:

$$P'_1 = [UWV^T \mid +\mathbf{u}_3] \quad P'_2 = [UWV^T \mid -\mathbf{u}_3]$$
$$P'_3 = [UW^TV^T \mid +\mathbf{u}_3] \quad P'_4 = [UW^TV^T \mid -\mathbf{u}_3]$$

corresponding to these four possible situations:





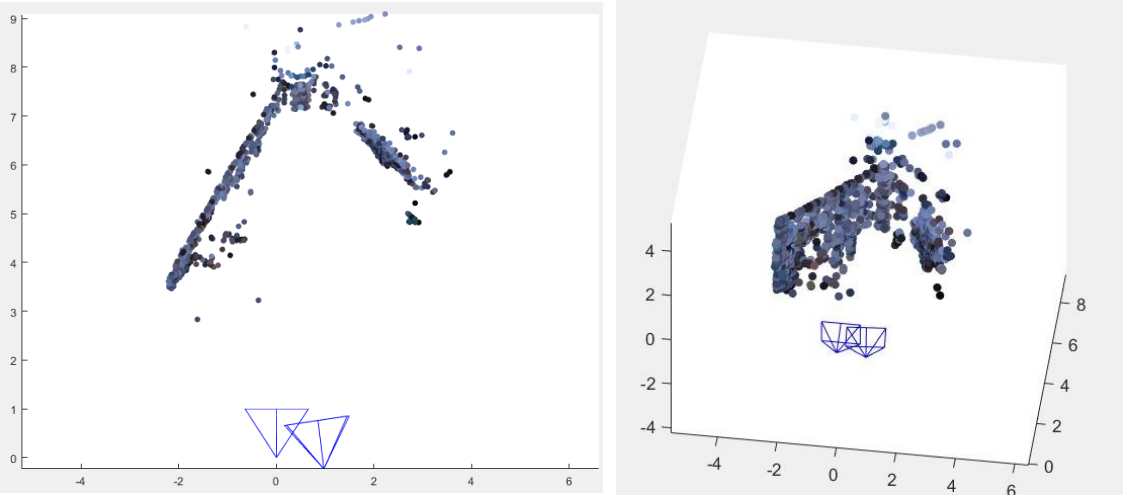
3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

A reconstructed point X will be in front of both cameras in one of these four solutions only. Thus, testing with a single point to determine if it is in front of both cameras is sufficient to decide between the four different solutions for the camera matrix P'.

Recover 3D points by triangulation

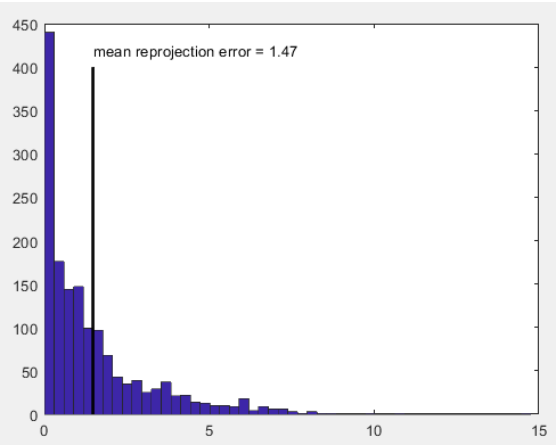
Since a set of matched points in images and the camera matrix P, P' are known now, to recover all the points in 3D space, the triangulation can be used with the formula: x≡PX. Below are the results of applying the triangulate function to recover 3D points:



To estimate the consequence of the triangulation, the reprojection error is considered. The process of reprojection error are:

- Project all the generated 3D points through these left and right cameras whose camera matrices are P and P' respectively. Two images which contained all the projected points x1' ,x2' are obtained.
- Saying that x1 and x2 are two sets of the reference matched points in two images.
- Compute the distances between x -> x1' and x -> x2', and then sum them up as the reprojection error for each 3D point.

The histogram of the reprojection errors and the mean reprojection error are shown below. In this case, the mean of the reprojection errors is 1.47.



3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

Depth map computation

In this part, two kinds of local method, sum of squared Differences (SDD) and normalized cross correlation (NCC), for computing and generated the depth map will be introduced. The main idea of the depth map computation is to find out the corresponding areas between the left image and the right image, then compute the shift distances between each pair of patch, which are the values of the specified point in a disparity map. The brighter shades represent more shift and lesser distance from the point of camera, while the darker shades represent lesser shift and therefore greater distance from the camera.

A function named stereo_computation is created to obtain the depth map

- function 'stereo_computation'
  - ✓ **Input**: the left and the right image, the minimum and maximum of the disparity, the size of the sliding window, the method of the matching cost(can be 'SSD' or 'NCC')
  - ✓ **Output**: the disparity map (depth map)

For a pair of rectified images, sliding a window along the same line in the right image and compare its content to that of the reference window in the left image. Patch with minimum matching cost is picked as the matching patch. The matching cost is calculated by these two ways:

- Sum of squared Differences (SDD):

$$C(\mathbf{p},\mathbf{d}) = \sum_{\mathbf{q}\in N_\mathbf{p}} w(\mathbf{p},\mathbf{q})\,|I_1(\mathbf{q}) - I_2(\mathbf{q}+\mathbf{d})|^m, \quad \sum_{\mathbf{q}\in N_\mathbf{p}} w(\mathbf{p},\mathbf{q}) = 1$$

$$N_\mathbf{p} = \{\mathbf{q} = (q_1,q_2)^T \mid p_1 - \tfrac{n}{2} \le q_1 \le p_1 + \tfrac{n}{2},\ p_2 - \tfrac{n}{2} \le q_2 \le p_2 + \tfrac{n}{2}\} \quad \text{(m=2)}$$

  The patch which gets the minimax of the C(p,d) should be the matching patch.
- Normalized cross correlation (NCC):
  - NCC: Normalized Cross Correlation

$$NCC(\mathbf{p},\mathbf{d}) = \frac{\sum_{\mathbf{q}\in N_\mathbf{p}} w(\mathbf{p},\mathbf{q})(I_1(\mathbf{q}) - \bar{I}_1)(I_2(\mathbf{q}+\mathbf{d}) - \bar{I}_2)}{\sigma_{I_1}\sigma_{I_2}}$$

$$\bar{I}_1 = \sum_{\mathbf{q}\in N_\mathbf{p}} w(\mathbf{p},\mathbf{q}) I_1(\mathbf{q}); \quad \bar{I}_2 = \sum_{\mathbf{q}\in N_\mathbf{p}} w(\mathbf{p},\mathbf{q}) I_2(\mathbf{q}+\mathbf{d});$$
$$\sigma_{I_1} = \sqrt{\sum_{\mathbf{q}\in N_\mathbf{p}} w(\mathbf{p},\mathbf{q})(I_1(\mathbf{q}) - \bar{I}_1)^2};$$
$$\sigma_{I_2} = \sqrt{\sum_{\mathbf{q}\in N_\mathbf{p}} w(\mathbf{p},\mathbf{q})(I_2(\mathbf{q}+\mathbf{d}) - \bar{I}_2)^2}$$

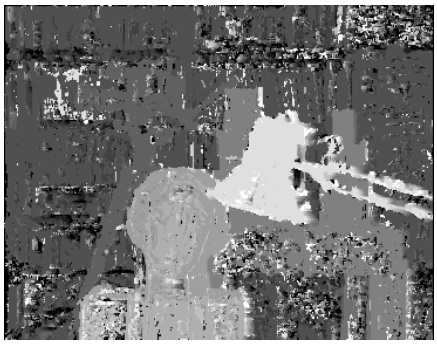
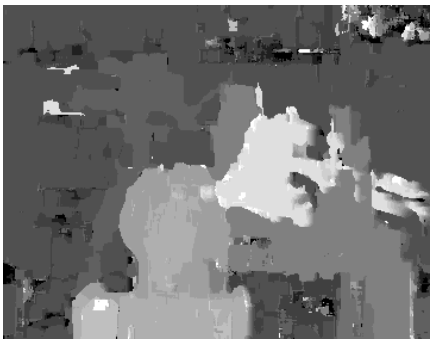
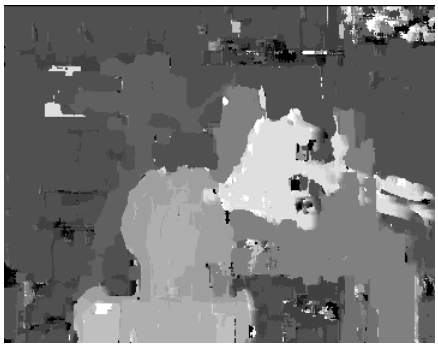
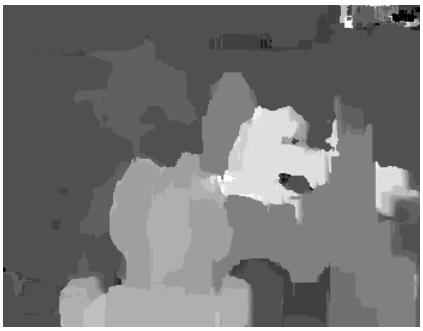
  The patch which gets the maximum of the NCC(p,d) should be the matching patch

Given two rectified images as below (views from left and right):

3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

Changing the methods of the matching cost and the size of the sliding window with 3x3, 9x9, 20x20, 30x30 to evaluate the results:

a) SDD cost	**b) NCC cost**
	
window size: 3x3	window size: 3x3
	
window size:9x9	window size:9x9
	
window size: 20x20	window size: 20x20
	
window size: 35x35	window size: 35x35

3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

Comments on the results:

- The brighter shades represent more shift and lesser distance from the point of camera, while the darker shades represent lesser shift and therefore greater distance from the camera.
- Both for SSD and NCC cost methods, the smaller sliding window be set to slide, the more details are shown, while with more noise coming out. The larger sliding window is, the less details are presented on the map, simultaneously the disparity map is smoother.
- The difference between SDD and NCC cost is not obvious. From the results, it can be said that with a small window size, the NCC one appears more noise but more details than the SDD results. And the shape of the objects with NCC cost are sharper.
- The minimum disparity and the maximum disparity should be chosen properly, otherwise the searching window might shift along to a wrong direction or the distance is too small and limited to find out the matching patch, which would lead to a bad result.

To test the functions, two new facade images as below are implemented.

Considering the left image as reference, the car has shifted a quite long distance to the left side. So the minimum and maximum disparities are set as: [-50, 20]

Changing the method of the matching cost and the size of the sliding window with 3x3, 9x9, 20x20, 30x30 to evaluate the results:

SDD cost

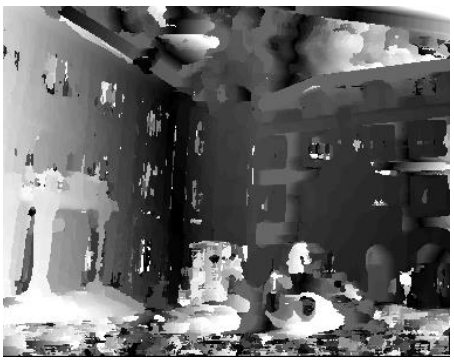
window size: 3x3

NCC cost

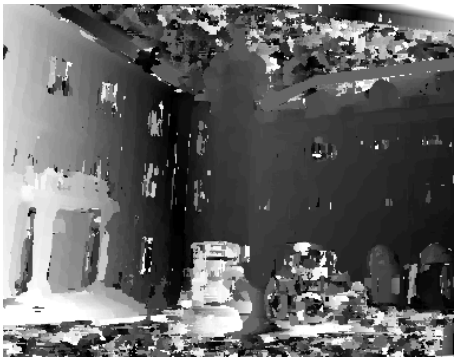
window size: 3x3

3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer



window size: 9x9



window size: 9x9

window size: 20x20

window size: 20x20

window size: 35x35

window size: 35x35

Comments on the results:

- Both for SSD and NCC cost methods, the smaller sliding window be set to slide, the more details are shown, while with more noise coming out. The larger sliding window is, the less details are presented on the map, simultaneously the disparity map is smoother.
- From the results, the difference between SDD and NCC methods is obvious. With a small window size, the NCC appears much more noise, but the shape of the objects with NCC is sharper.

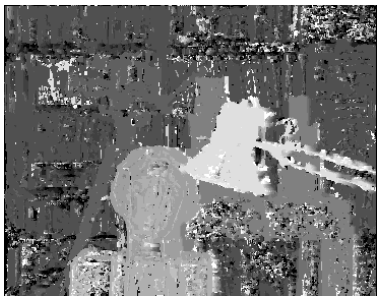
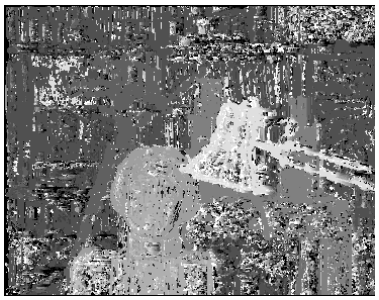
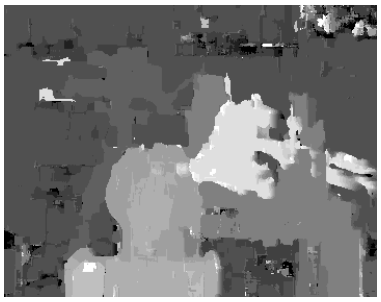
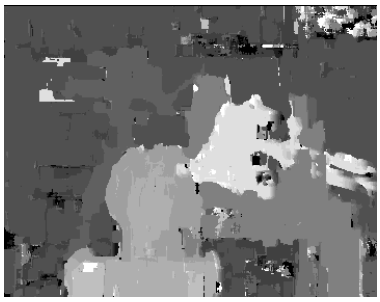
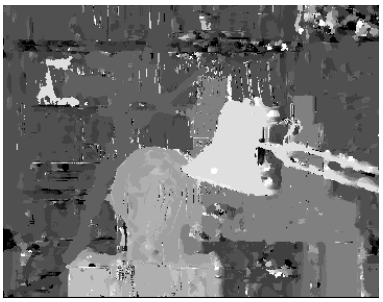
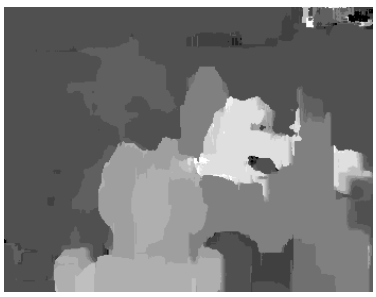
3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

Bilateral weights

As mentioned before, with the sliding window getting larger, the image will be filtered strongly. This is the main drawback of the depth map while using SSD and NCC matching cost. To deal with this problem, Bilateral weights method can be applied on it.

A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences (e.g., range differences, such as color intensity, depth distance, etc.). This preserves sharp edges.

a) SDD cost	**b) NCC cost**	**c) Bilateral weights**
		
window size: 3x3	window size: 3x3	window size: 3x3
		
window size: 9x9	window size: 9x9	window size: 9x9
		
window size: 20x20	window size: 20x20	window size: 20x20

3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer


window size: 35x35


window size: 35x35


window size: 35x35

Comments on the results:

- With a small sliding window (3x3 & 9x9), the bilateral weights results get more noise generated than SSD and NCC.
- As the window size increases, the advantage of bilateral weights method is more and more obvious. The depth map preserves a lot of sharp edges and more information of the image.

Two facade images are used to test with the bilateral weights filter (window size:35x35):


a) SSD cost (weight=1)


b) NCC cost (weight=1)


c) with bilateral weights filter

As we can see, the bilateral weights filter preserves more sharp edges and information of the image.

3D RECOVERING OF URBAN SCENES LAB 4

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

PRPJECT CONCLUSIONS

Once this project done, we can extract some points to consider:

- Once two images including many set of matching points and two camera intrinsic matrices K, K' are provided, the camera matrices P, P' of two cameras can be computed by the fundamental matrix F. Then the triangulation can be applied to reconstruct the 3D points in real world.
- Two main procedures of depth map computation by local method are: searching the matching patch and calculating the disparity.
- For searching the matching patch, SDD, NCC and bilateral weights filter are three methods to calculate the matching cost. The noise and the sharpness of the edge are related to the sliding window size.
- When a large size of the sliding window is taken, the bilateral weights filter is a good way to compute the depth map because it preserves much more sharp edges and information of the image.