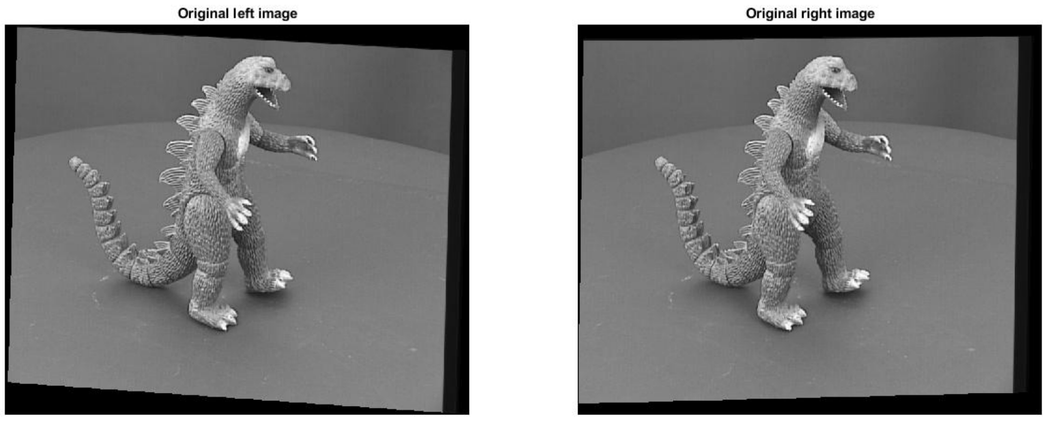
Computer Vision: Optional Lab
Disparity computation

Francesca Canale 4113133
Filippo Gandolfi 4112879
Marco Giordano 4034043

24/06/2019

1 Introduction

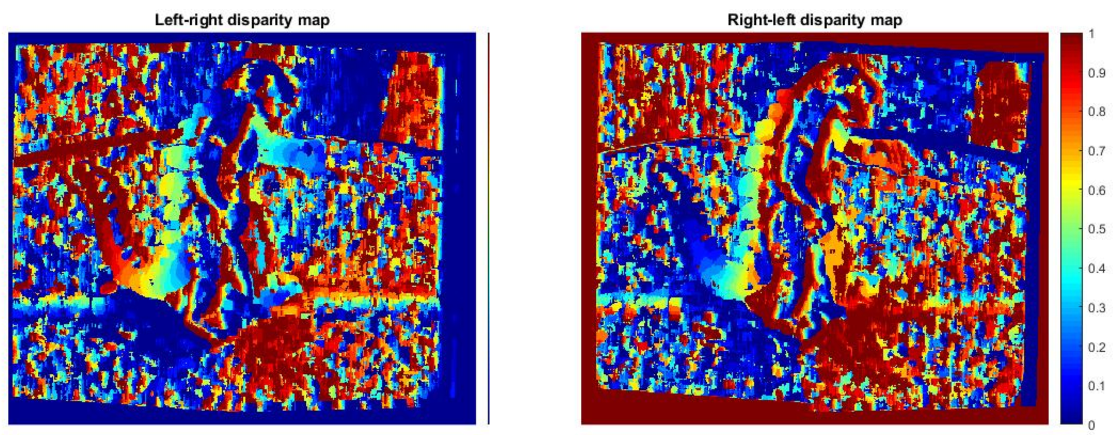
The goal of this optional lab session is to implement a function similar to MATLAB function *disparity*(). This function computes the disparity between two stereo images (couple of images of the same subject but taken by two near cameras, so the angle is different).



2 SSD

We implemented a function *my_ssd*() that takes in input two image patches of the same size and computes the sum of squared differences. This is useful in order to understand the similarity between the images.

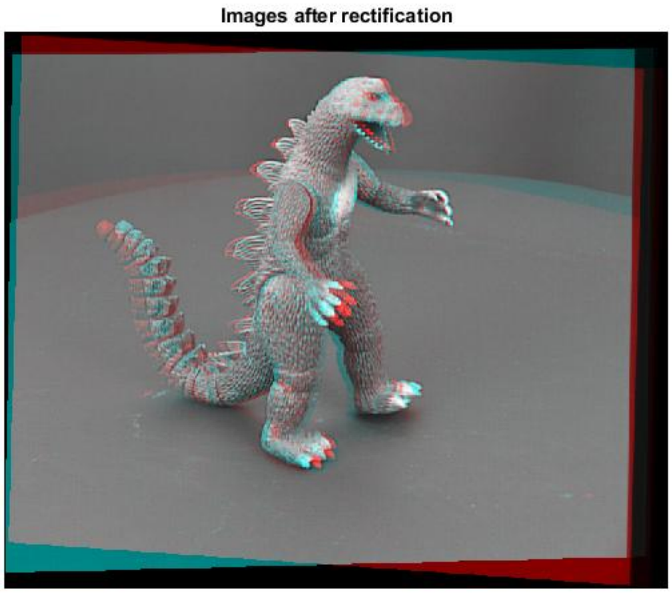
3 Disparity map



To compute the disparity map we implemented a function *my_disparity*().
The disparity map is a matrix in which for every pixel of the first image we put the column (respect to that point) of the most similar point in the second image. Note that we check the disparity only on the rows: this because the two cameras are almost at the same height (they are left and right cameras, not up and down cameras) and so the shifting is only horizontal.

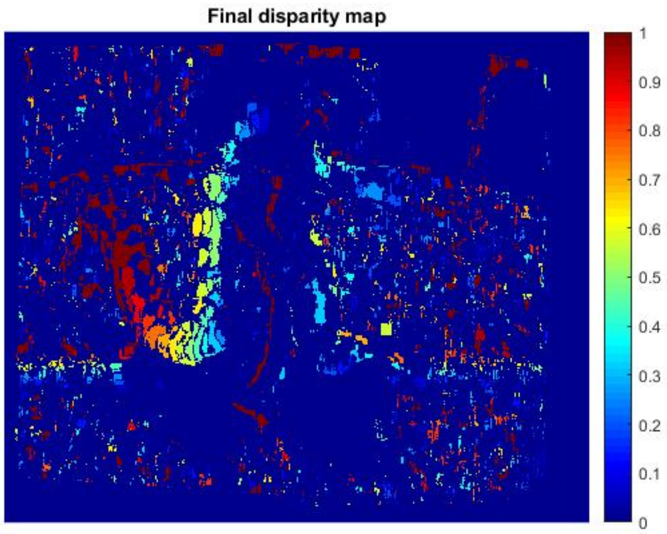
Our function need as inputs the two images, W that is the size of the correlation window and a search range $[d_{min}, d_{max}]$. W is needed to say how many near pixel are taken into consideration when computing the SSD and the range tells us where to search for the similarity point in the second image. When changing from the left-right disparity to the right-left one we can use the same function, but with d_{min} and d_{max} exchanged and with opposite sign to be consistent with the previous disparity map.

The choice of the range values is crucial: if we choose too small values we may not find the right correspondence point in the second image, if we choose too big values we put too much noise (correspondence point could be a wrong distant point). To find nice values for the range is useful to use a rectified image, which plots the two image one above the other with different colors. This could help to understand how much the two images are shifted.



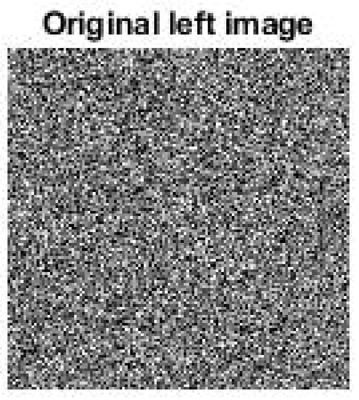
4 Disparity and Consistency computation

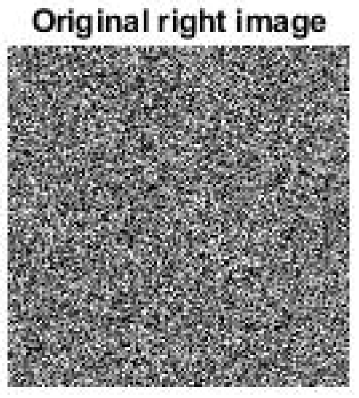
To compute the disparities and the Left-Right consistency we implemented a function $my\_disparity\_final()$. To compute the consistency we need to combine the disparity from left to right and the one from right to left. Having both disparity maps, we checked if $D_{lr}(i,j) == D_{rl}(i, j + D_{rl}(i,j))$. If this is true then the consistency is confirmed and we put $D_{final}(i,j) = D_{rl}(i,j)$. If this is false we put in $D_{final}(i,j)$ the minimum floating point ($-realmax(`single')$).

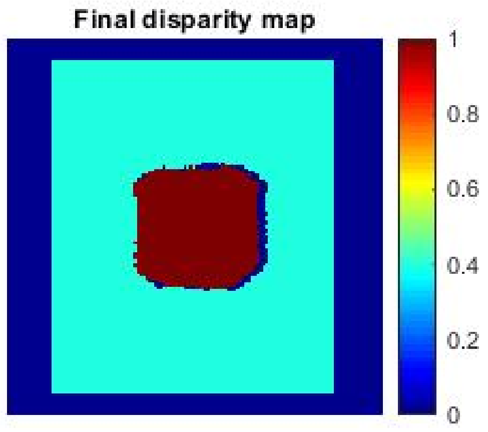


5 Conclusion

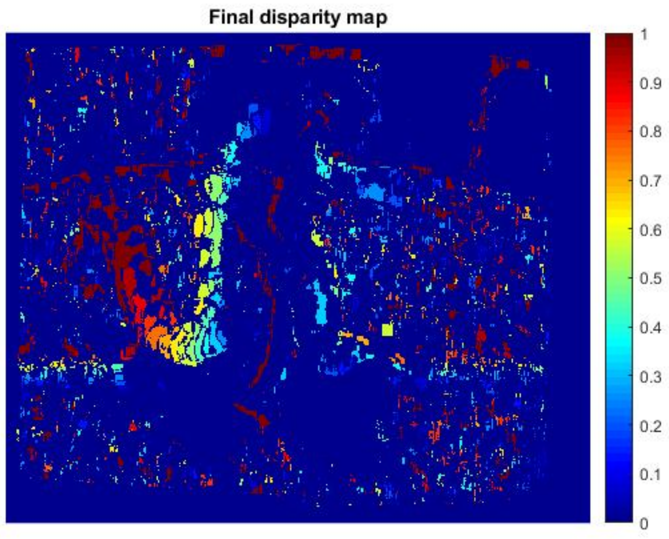
Comparing our results with those of Matlab, our function had a good with only the first couple (as we can see below) of image. It is the most simple because it is a synthetic image create for the scope.







In real case images (like dino image), matlab function is a lot better: for the result but also for the computation time: our takes like 2 minutes, matlab a few seconds.


Final disparity map