

Computer vision - Homework 3

Nicolas Six

February 15, 2018

Question 1 SSD on test image

As you can see on Figure 1, the shift of 2 pixels of the central square is correctly detected. But it's border are noisy when the window's center is on the edges as there is no perfect match and the noisy background has bigger variation than the square. Thus, even when most of the window is on the square, the background has a large influence on the SSD.

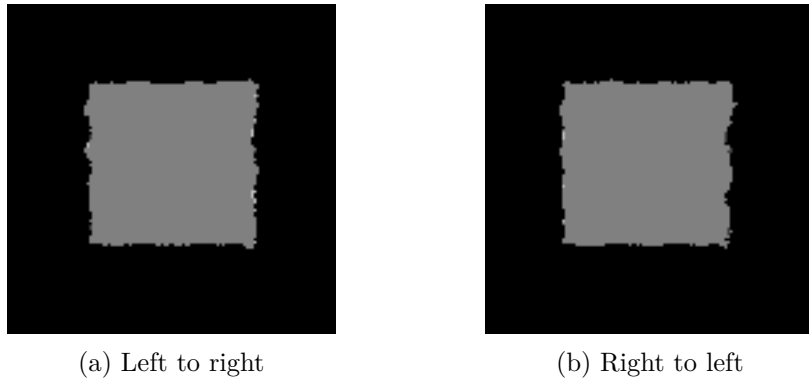


Figure 1: Disparity computed using SSD, with a windows size of 5 and d going from 0 to 3

To speed up the next step of this problem set, we implemented different version of the disparity algorithm with various level of optimization. The first is a basic three nested for loop with the SSD computed using numpy, each loop iterating on x , y and d respectively. The second is parallelized version of this algorithm, where the two first for loops are parallelized using python's multiprocessing package and the last for loop turned into a map to reduce the impact of python interpreter. The third is an algorithmic optimization, where the loop order is reversed so that we first iterate on d , allowing us to compute the first step of the SSD in a window agnostic

Algorithm	time (s)	time relative
Basic nested loops	2.275	100%
Basic, parallelized	1.296	57%
Reverse loop	1.306	57%
Reverse loop, parallelized	0.634	28%

Table 1: Timing of the different implementation for the basic square image, with a windows size of 5 and d going from 0 to 9

way, computing already the squared difference on each pixels so that only the sum on the window is needed locally. This is done using basic python nested for loop for this implementation. The last implementation is a parallelized version of the previous, using the multiprocessing package.

The timing of the different methods on my machine (Intel i5-3230M cpu, 2 cores / 4 threads) are displayed on Table 1. The time are get on only one run on the basic square image used in this question, but still show an idea of the gap between the different solutions.

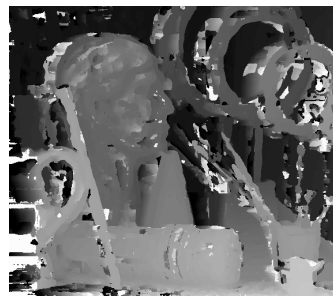
From now, only the last one will by used. On a full image, it give the result in about 2 minutes, which we consider as correct as entirely written in python without compilation, but is order of magnitude bigger than in C/C++ or cuda.

Question 2 SSD on real images

2.a Computed disparity



(a) Left to right



(b) Right to left

Figure 2: Disparity computed using SSD, with a windows size of 9 and d going from 30 to 120

2.b Comparison with ground truth

In comparison with ground truth, our results are very noisy. But our version still display the main features of the image. Most of the noise appear on the part present on only one of the images, but noise also appear on large uniform area. It also worth mentioning that the precision on the edges of the image is far from good, even if the image is cropped to part where the SSD can be computed large band are present on the image when the foreground is too close to the camera.

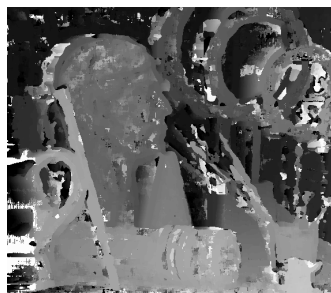
Question 3 SSD and perturbations

3.a Gaussian noise

As expected, SSD is not resilient to Gaussian noise. Adding a Gaussian noise of scale 10, on one image, only gave us the poor results displayed on Figure 3. Because all the distinctive part that the SSD is looking for are completely hidden into the noise, making it hard for SSD to match the right spot. Resulting in an even more noisy disparity image.



(a) Left to right

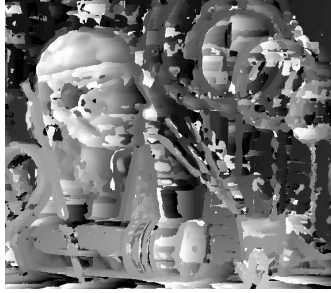


(b) Right to left

Figure 3: Disparity computed using SSD with Gaussian noise added to the left image, with a windows size of 9 and d going from 30 to 120

3.b Contrast

For contrast the results are the same as for Gaussian noise. As you can see on Figure 4, we even get more noise than before. This come from the fact that rising the values of one image completely change the result of the SSD subtraction. This illustrate the fact that to work properly the SSD need the pixels representing the same world point to have the exact same value.



(a) Left to right



(b) Right to left

Figure 4: Disparity computed using SSD with contrast artificially higher on the left image, with a windows size of 9 and d going from 30 to 120

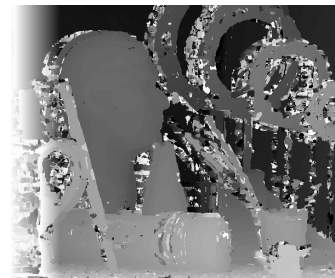
Question 4 Normalized correlation

4.a On original images

Normalized correlation perform here far better than SSD. If the noise on the pixels not present on both images is still here (and actually, their are no way to get correct value there using stereo), the objects are clearly visible and far more uniform than when using the SSD. Band of approximated value are still present on the side of the images, where the closest objects are not on the two images, but the furthest are still.



(a) Left to right



(b) Right to left

Figure 5: Disparity computed using Normalized correlation, with a windows size of 9 and d going from 30 to 120

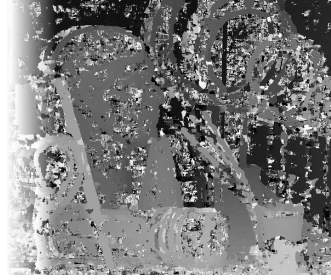
4.b with Gaussian noise and modified contrast

Normalized correlation is not really more resilient to Gaussian noise than SSD, for the same reasons. Although it's interesting to note that the noise

have more effect when applied to the compared image than to the source image. This is easy to explain as the second case, the noise compared is always the same so chance are that the best match will still be the good one, as the other image is noise free.



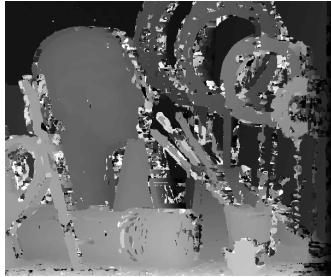
(a) Left to right



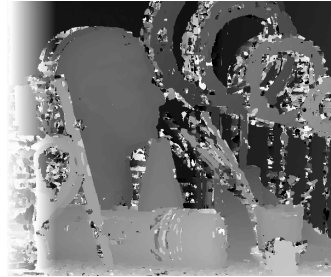
(b) Right to left

Figure 6: Disparity computed using Normalized correlation with added Gaussian noise, with a windows size of 9 and d going from 30 to 120

As it can be inferred from the normalized correlation formula, this method is resilient to contrast variation. There is little to no variation with the original disparity. The only small variation may have been introduced by the upper bound of 255 imposed by the picture format.



(a) Left to right



(b) Right to left

Figure 7: Disparity computed using Normalized correlation with modified contrast, with a windows size of 9 and d going from 30 to 120

Question 5 Second image pair

5.a Experiments on second pair

To answer this question, we applied basic transformation to the images before computing the disparity using the two different methods. To have a reference for the effects of this different transformations we also computed the disparity on the original images, which can be seen on Figure 8 and ??.

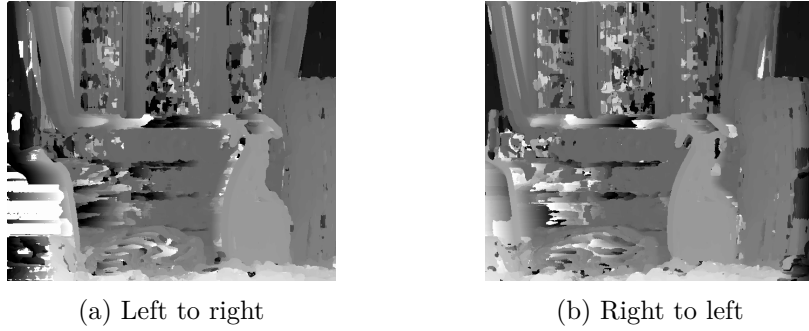


Figure 8: Disparity computed using SSD, with a windows size of 9 and d going from 10 to 115

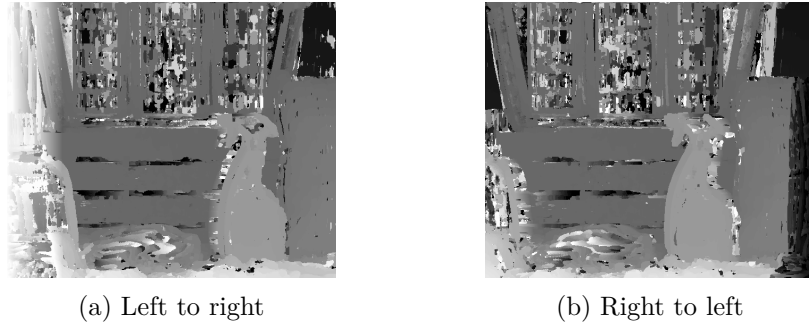


Figure 9: Disparity computed using Normalized correlation, with a windows size of 9 and d going from 10 to 115

Our first experiment was to add Gaussian blur in order to remove the noise of the camera. The best result achieved are displayed on Figure 10 and 11. We must note that there is no significant improvement on both case. The best improvement is on the bottom central clothes where the previous noise disappeared for both the SSD and normalized correlation. But at the same time noise start to appear on other place of the image, such as on the wooden box, meaning that smoothing more will not improve the result.



(a) Left to right

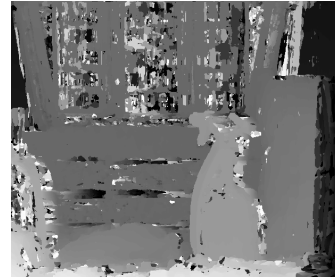


(b) Right to left

Figure 10: Disparity computed using SSD on smoothed images, with a windows size of 9 and d going from 10 to 115



(a) Left to right



(b) Right to left

Figure 11: Disparity computed using Normalized correlation on smoothed images, with a windows size of 9 and d going from 10 to 115

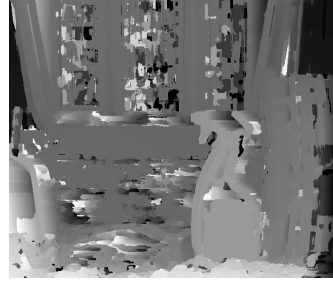
We also noted that the right image as an average and median intensity 2% below the left one. Thus, we lowered the contrast of the left image to equalize that. The Figure 12 show disparity we get using SSD, you can see that the results are quite similar to the one obtained on the original image (Figure 8). But with this new image, we can see that the intensity on the wooden box is a gradient, different from the left to the right images, explaining the difficulties of SSD on the wooden box.

We also tried to add some smoothing in addition to the contrast correction but this doesn't allow us to get good results. Which are thus not displayed here, but are available in the attached images or by running the code.

As a conclusion, normalized correlation is the best performer here. Considering the small amount of noise added by the smoothing we can say that our best result is Figure 11. But those are quite far from ground truth and most of the small feature are very noisy, such as the hole in the laundry



(a) Left to right



(b) Right to left

Figure 12: Disparity computed using SSD on contrast adjusted images, with a windows size of 9 and d going from 10 to 115

basket.