Computer Vision
CS 6476 , Spring 2018

PS 2

Professor :
Cedric Pradalier

Author :
Melisande Zonta

February 16, 2018

Contents

1 Implementation of the SSD match algorithm 2

2 Test on original images 3

2.1 Application of the SSD algorithm 3

2.2 Comparison of the results with the ground truth 4

3 Robustness of the SSD algorithm to certain perturbations 4

3.1 Add of Gaussian noise of noise 4

3.2 Increasing of the contrast 5

4 Implementation of the normalized correlation 5

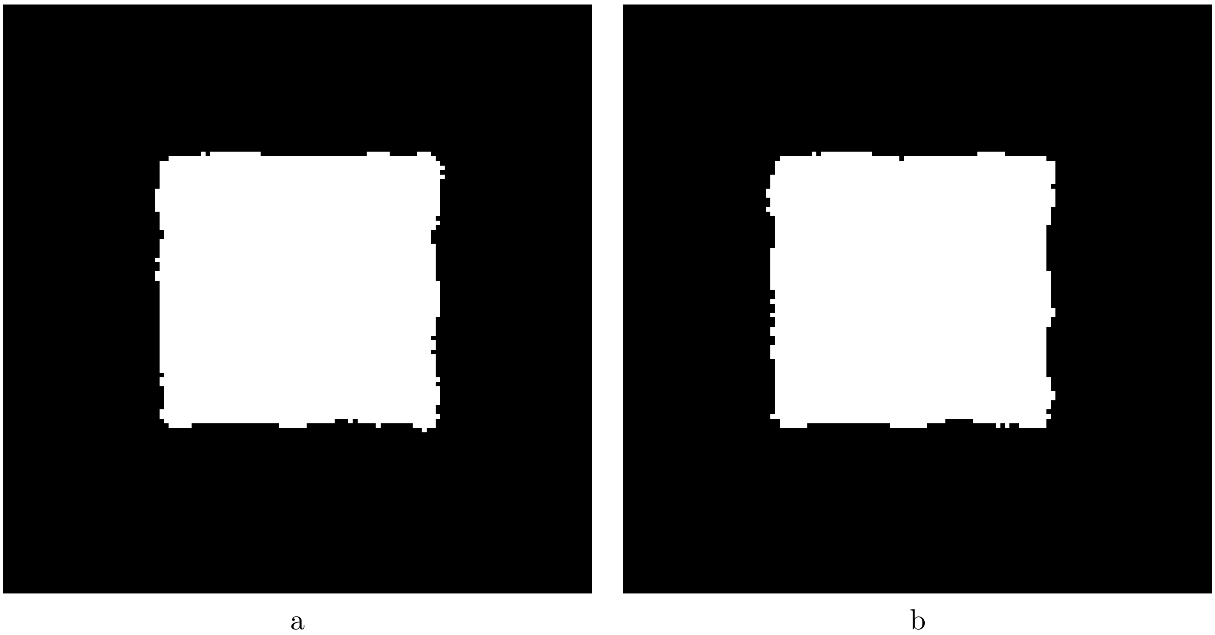
4.1 Application of the algorithm 5

4.2 Robustness Tests 6

5 Final Test on other images 8

1 Implementation of the SSD match algorithm

The Figure 1 a. and b. show respectively the left to right and right to left disparity results using the SSD scoring method.



a

b

Figure 1: *a.* ps2-1-a : Left to right Image. *b.* ps2-1-a : Right to left Image.

2 Test on original images

2.1 Application of the SSD algorithm

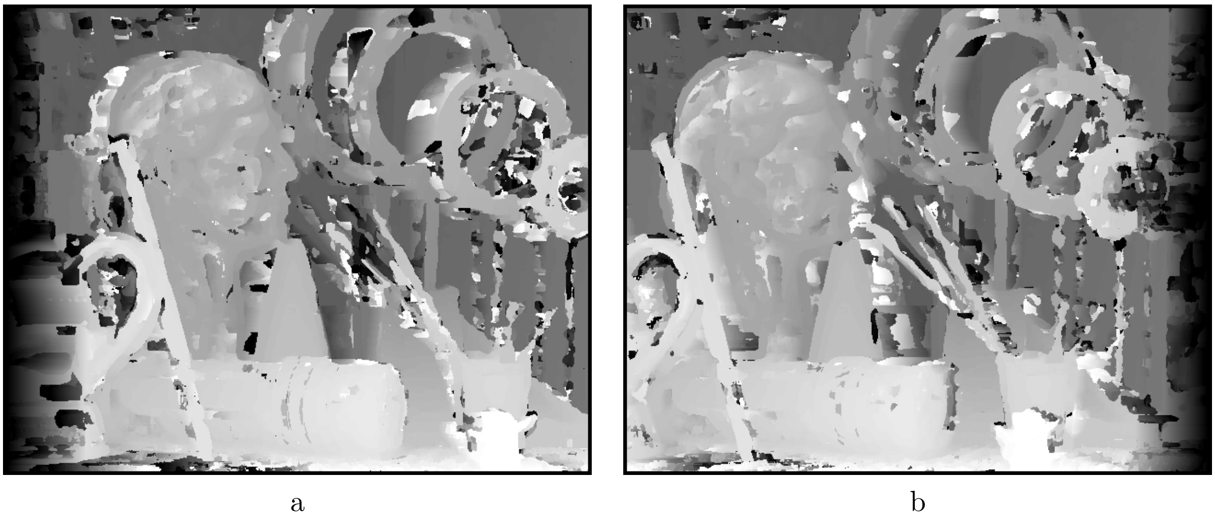


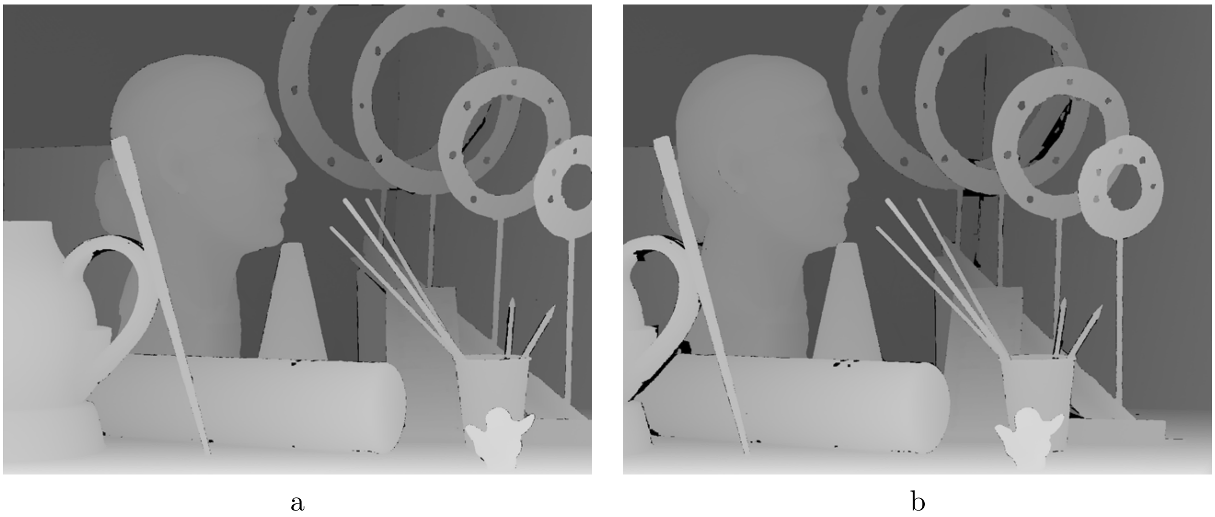
Figure 2: *a.* ps2-2-a : Left to right Image. *b.* ps2-2-a : Right to left Image.

The Figure 2 a. and b. are the result of the same algorithm applied to the real picture of the artist?s workshop.

The images were obtained for a window size of 9 pixels and a maximum measured disparity of 100 pixels.

Our results are quite reassuring since the left to right and right to left disparity images are quite similar. We more or less get the same output, with a darker region on one of the side, which is due to the fact that we cannot compute the disparity on one edge, because the corresponding pixels actually lie outside our target image.

2.2 Comparison of the results with the ground truth



a

b

Figure 3: *a.* ps2-2-b : Ground Truth Left Image. *b.* ps2-2-b : Ground Truth Right Image.

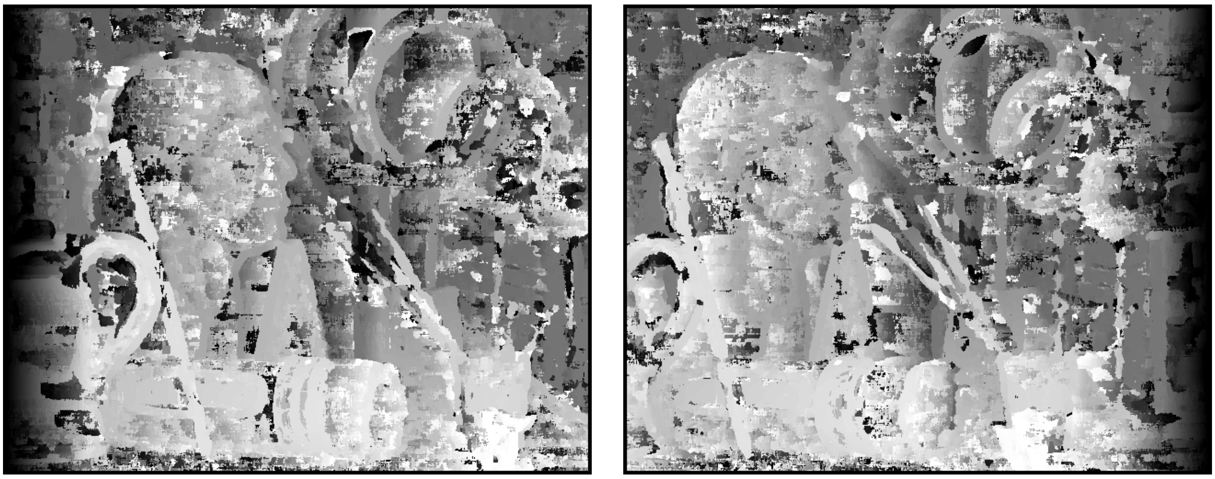
Most portions of our disparity maps match the ground truth, especially for large objects. The pens and paintbrushes' handles do not match. The main reason is that the window in this case covers areas located at varying depth, which makes it difficult to find a match. The larger brush, on the left, is better represented.

The same issue arises regarding the vertical bars supporting the red rings in the back, but with an additional issue due to the repeating background which makes it difficult to distinguish the individual bars.

In the end since the objects get a reasonable disparity value, this window size will be used for the next tests.

3 Robustness of the SSD algorithm to certain perturbations

3.1 Add of Gaussian noise of noise



a

b

Figure 4: *a.* ps2-3-a : Left to right Image. *b.* ps2-3-a : Right to left Image.

Those results (Figure 3 a. and b.) were obtained for a window size of 9 pixels and a noise sigma of 25.

We use a quite high sigma value in order to get a very clear idea of the effect of a Gaussian noise on the disparity computation. Using the same parameters as in the previous question, we notice that we can still clearly see the similarities between the two disparities, but the erroneous pixels created a lot of disparity noise.

Regions in large objects that appeared correctly with a rather smooth disparity now have a more granular aspect. The erroneous patches already visible in the previous test are still visible, with additional smaller errors spread across the disparity map.

However, we notice that although this as a big impact in regions with smooth disparity (based on the ground truth), the other regions of the image, which were not properly matched in the previous test, are not as much affected by the noise (example of the pens and vertical bars).

3.2 Increasing of the contrast

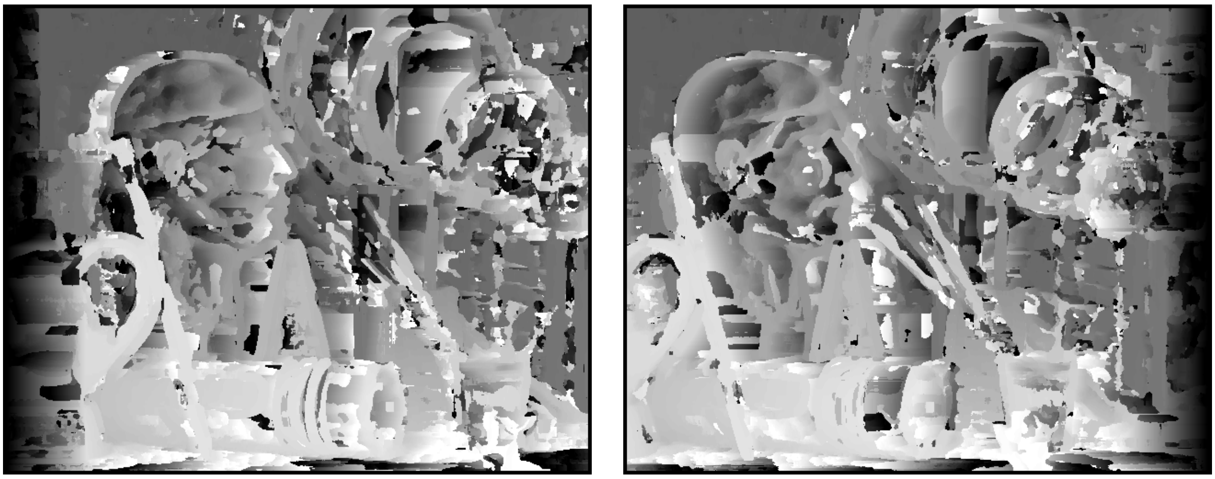


Figure 5: *a.* ps2-3-b : Left to right Image. *b.* ps2-3-b : Right to left Image.

In comparison, increasing the contrast simply by multiplying one of the image by a scaling factor has a very different impact on the output of the disparity computation. The result is visible in Figure 4 a. and b.

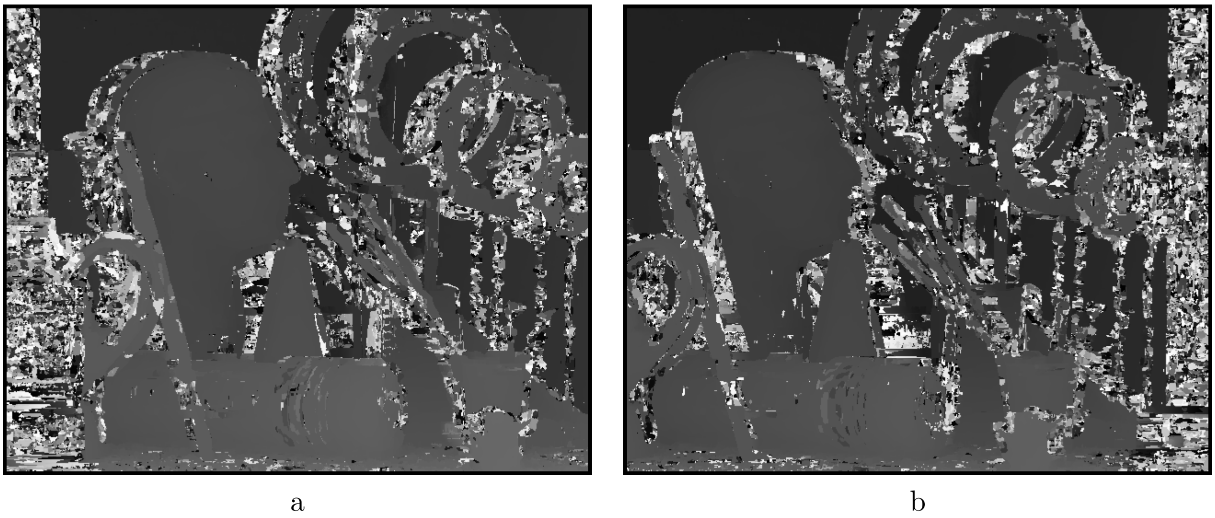
Most regions of smooth disparity that we had in the first test now get ?patchier?, they get broken up in smaller smoothed patches of disparity.

Although the "edges" of the objects seem to be conserved, this looks like a worse result than in the Gaussian noise test. Indeed, whereas some smoothing on the disparity map could probably solve some problems caused by the noise, this is trickier with the contrast effect, as the mean values of larger regions has been lost.

4 Implementation of the normalized correlation

4.1 Application of the algorithm

The tests below were done using a disparity computation based on the normalized cross-correlation available in through the cv2.matchTemplate function.



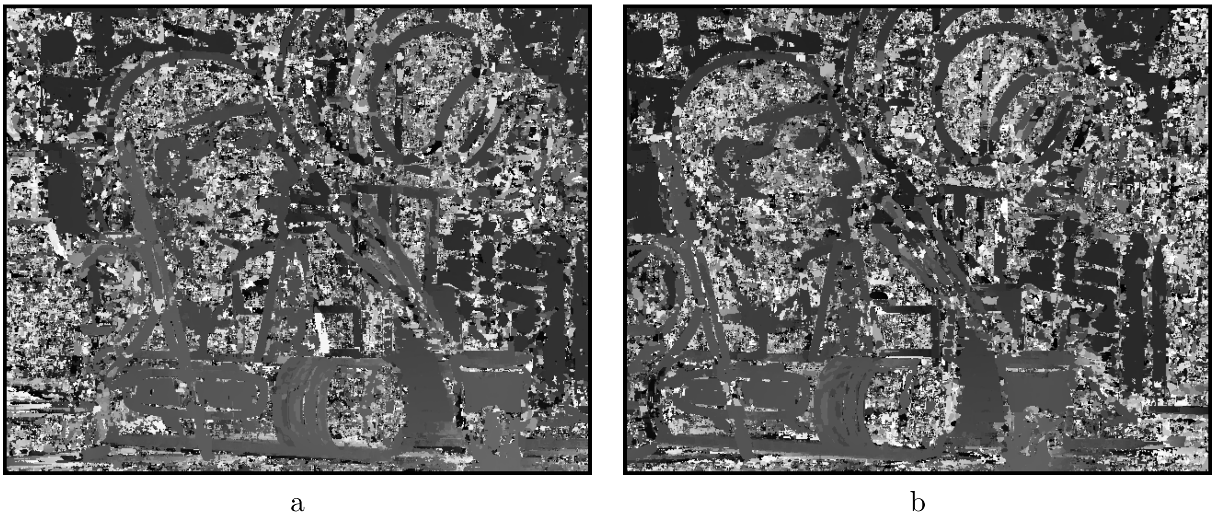
a

b

Figure 6: *a.* ps2-4-a : Left to right Image. *b.* ps2-4-a : Right to left Image.

Without noise (Figure 5 a. and b.), it seems that large areas of similar disparity get correctly matched but we have weird effects appearing as ?shadows? casted in the opposite direction of the matching.

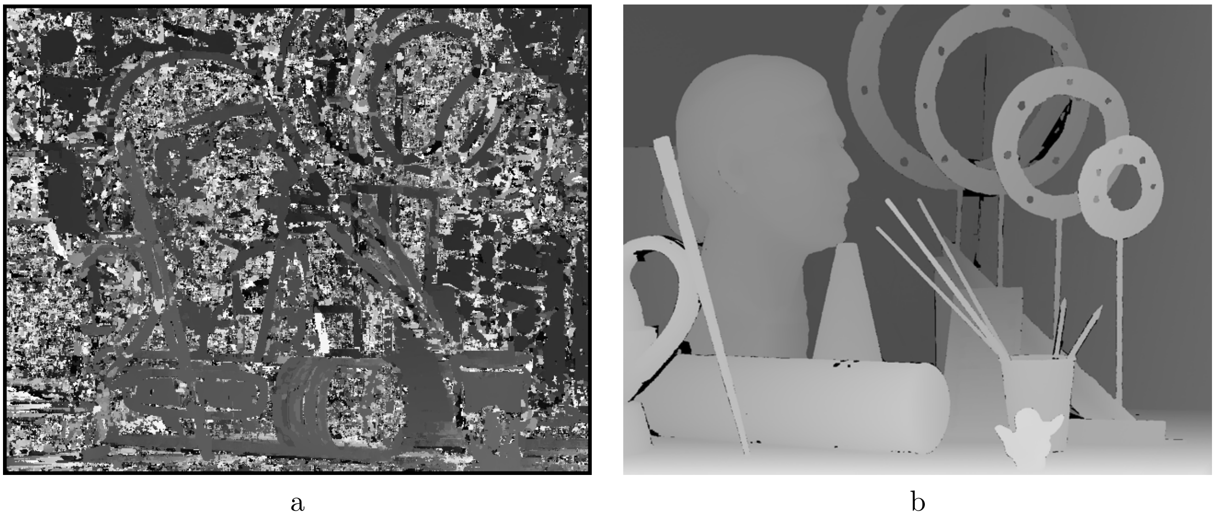
4.2 Robustness Tests



a

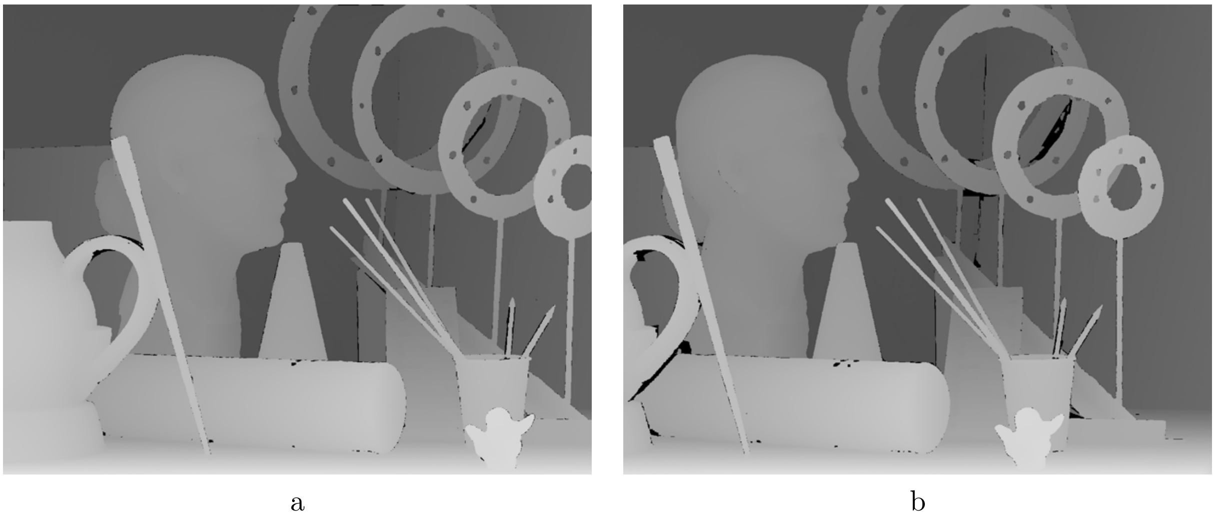
b

Figure 7: *a.* ps2-4-b : Left to right Image. *b.* ps2-4-b : Right to left Image.


a

b

Figure 8: *a.* ps2-4-b : Left to right Image. *b.* ps2-4-b : Right to left Image.


a

b

Figure 9: *a.* ps2-4-b : Ground Truth Left Image. *b.* ps2-4-b : Ground Truth Right Image.

The results from images subjected to Gaussian noise (Figure 6 a. and b.) are suspicious.

The matching does not really work, and an ?edge effect? is visible on the disparity map, with only the boundaries of large objects distinguishable.

Images resulting from increased contrast versions of the original one (Figure 7 a and b.) look very similar to the one obtained from the original images.

It seems that the cross-correlation method is not affected by contrast changes as the SSD one.

As a conclusion for this part, we could say that the cross-correlation method is more efficient than the SSD method.

Indeed, taking a proper implementation of this method, the fact that it does not suffer from contrast changes seems quite positive for real case stereo matching scenarios, where left and right images could slightly differ in contrast.

Furthermore, the apparent sensitivity of this method to Gaussian noise is not a big issue in modern applications, where this type of noise appears mainly in situations where the lights are low, typically at night.

If the setup is more favorable, this should not be a problem, especially considering that our tests were done with rather high level of noise, which is quite improbable in real case scenarios.

5 Final Test on other images

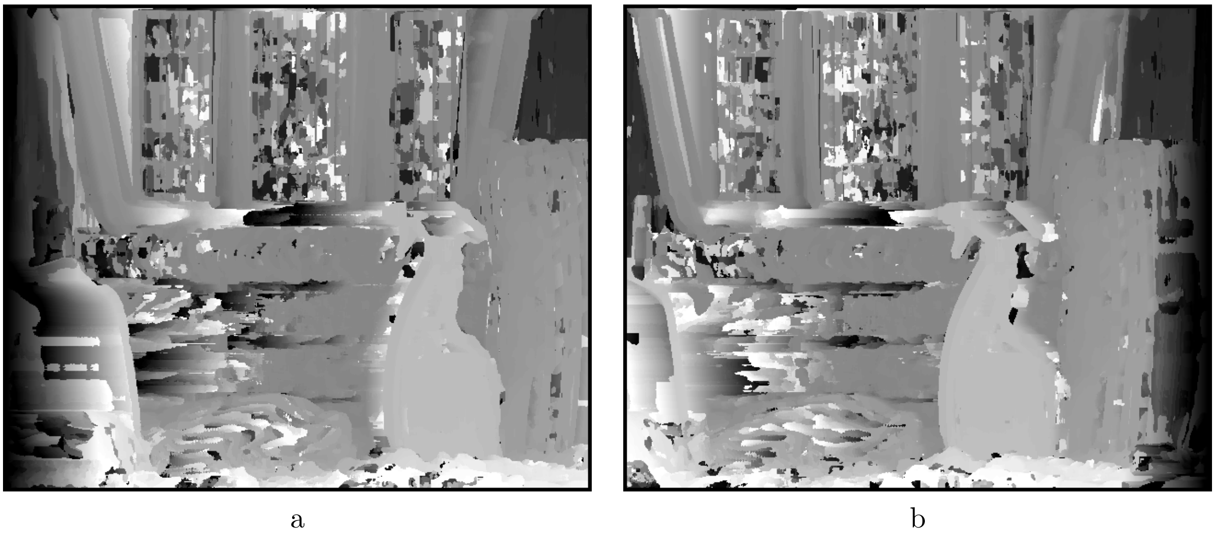


Figure 10: *a.* ps2-5-ssd : Original left to right image. *b.* ps2-5-ssd : Original right to left image.

The images presented in Figure 8.a. and b. correspond to the results obtained without noise and using the same parameters as previously.

We see that the resulting disparity maps are a lot less accurate than what was obtained on the previous set of images.

This is mostly due to the numerous horizontal patterns seen in the windows, cushions and wood strips in the background, which cause the algorithm to fail at finding the good matches in those areas.

However, we can still see that larger objects such as the red piece on the left and the bottle, are effectively detected with higher disparities, placing the in the forefront, as in the ground truth image.

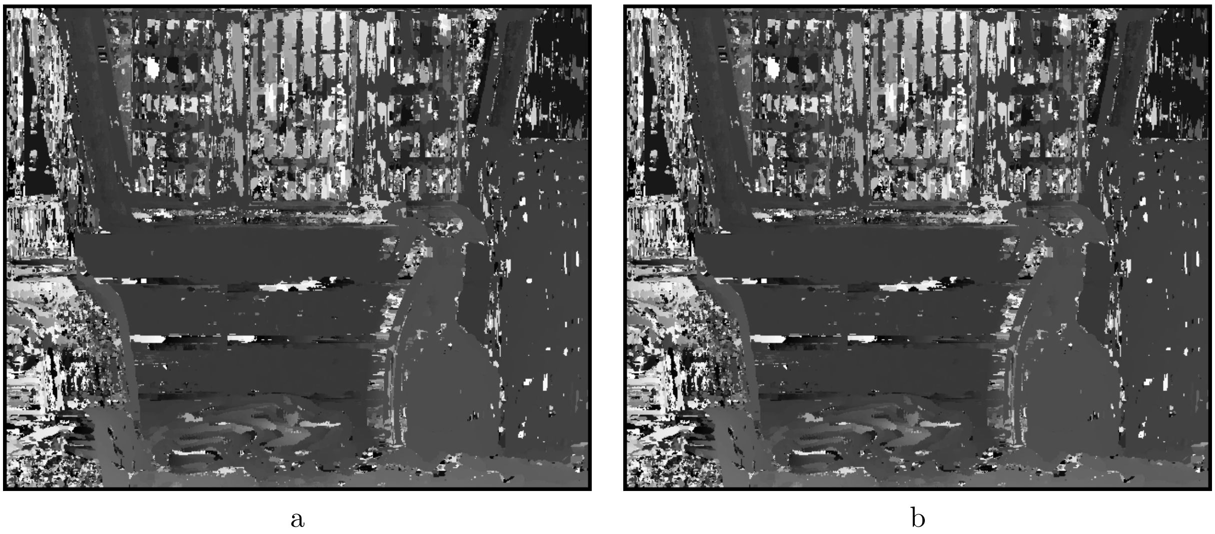


Figure 11: *a.* ps2-5-corr : Original left to right image. *b.* ps2-5-corr : Original right to left image.

The results 9.a. and b. are using cross-correlation.

As explained in the sections above, a pre-processing or a post-processing on the disparity maps may well be necessary in order to correct the different flaws that could affect a real image.

Smoothing could probably solve some of the effect of the Gaussian noise, although a pre-computing smoothing could also be applied beforehand, to further reduce those defects.

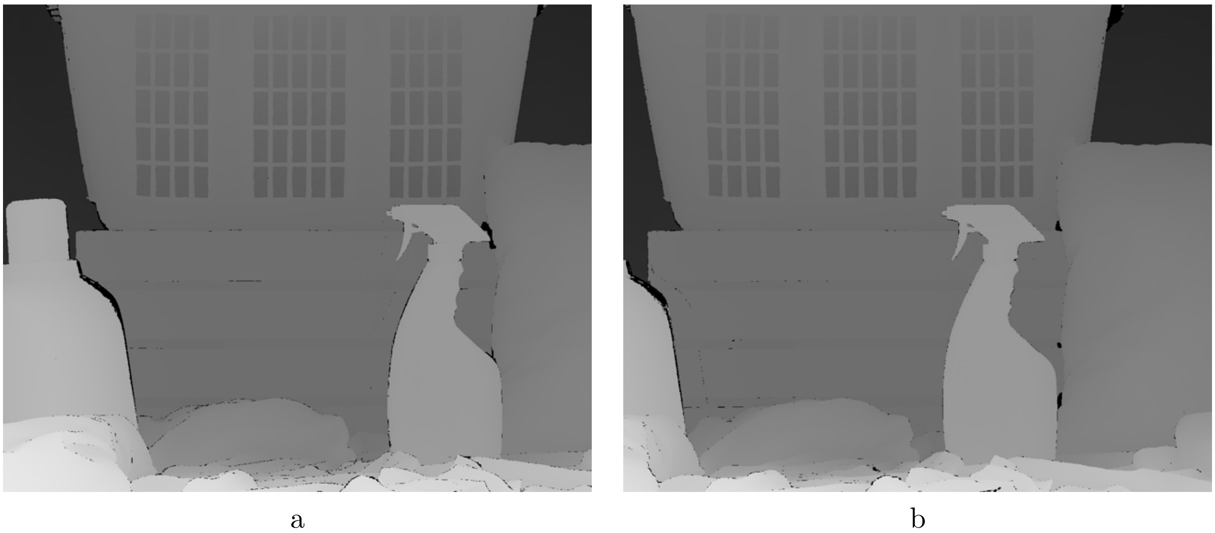


Figure 12: *a.* ps2-5-ground-truth : Left Image. *b.* ps2-5-ground-truth : Right Image.

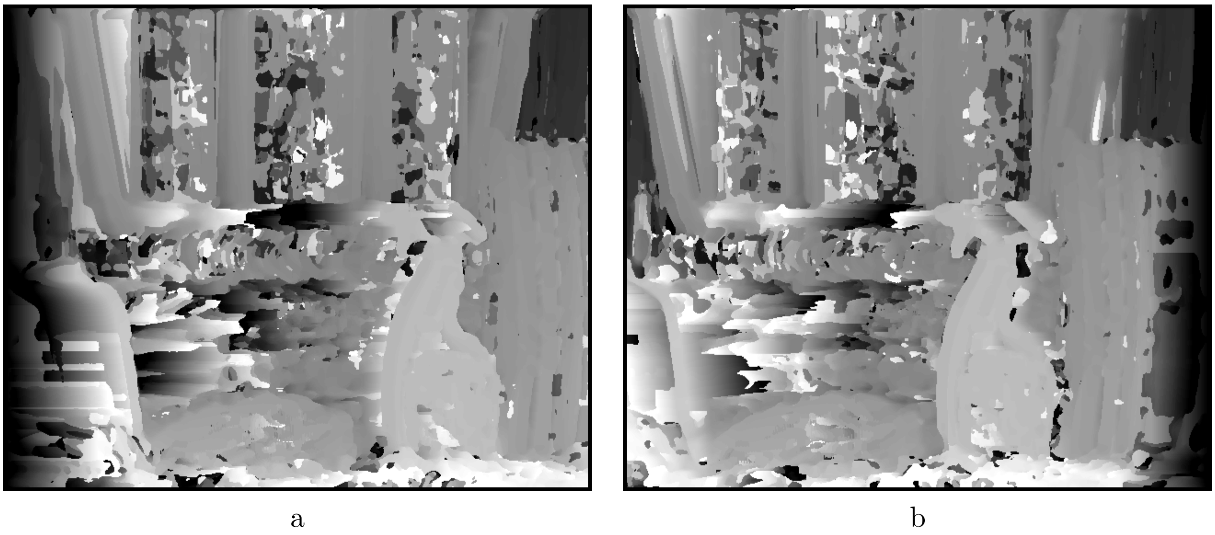


Figure 13: *a.* ps2-5-ssd-blur : Left to right image after blurring. *b.* ps2-5-ssd-blur: Right to left image after blurring.

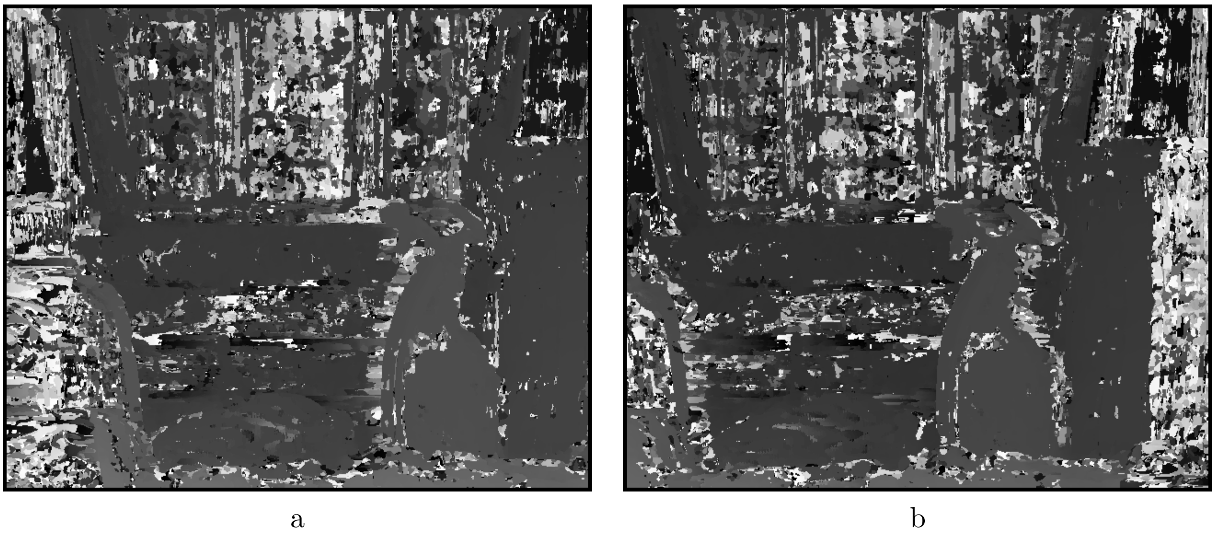
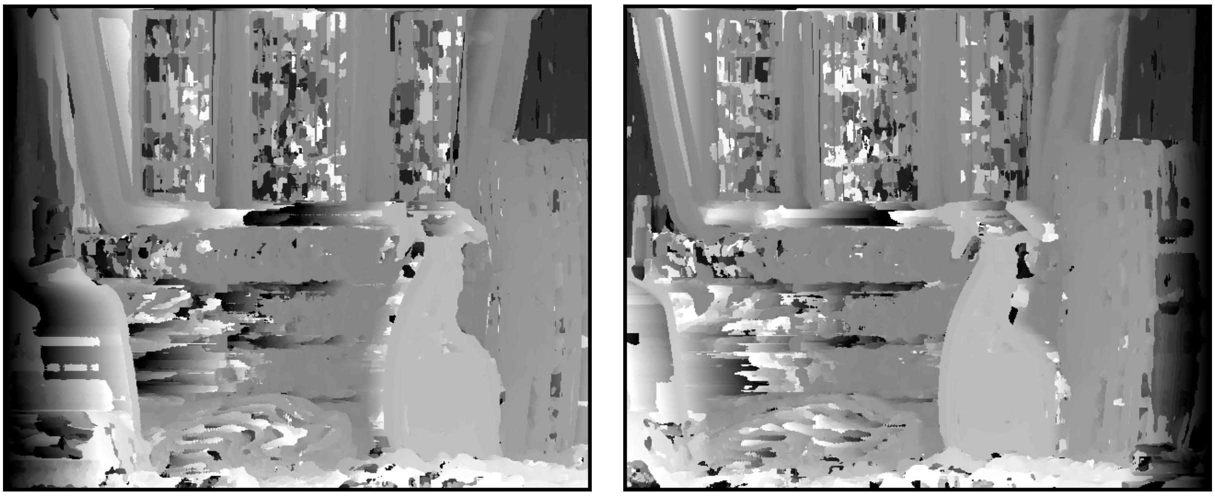


Figure 14: *a.* ps2-5-corr-blur : Left to right image after blurring. *b.* ps2-5-corr-blur: Right to left image after blurring.

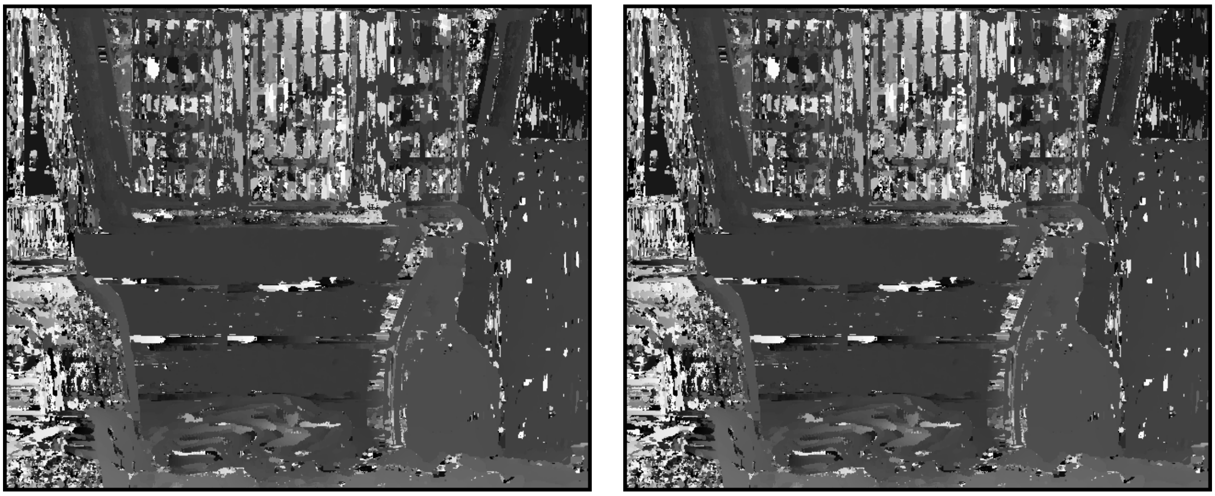
We applied a gaussian smoothing with window size 5. The results obtained after smoothing are quite similar with the original images for either the SSD processing or the cross correlation one.



a

b

Figure 15: *a.* ps2-5-ssd-contrast : Left to right image after contrast enhancement. *b.* ps2-5-ssd-contrast: Right to left image after contrast enhancement.



a

b

Figure 16: *a.* ps2-5-corr-contrast : Left to right image after contrast enhancement. *b.* ps2-5-corr-contrast: Right to left image after contrast enhancement.

The results available in Figure 16 a. and b. finally show the effect of multiplication (contrast enhancement), with a patchy effect, especially in zones with very smooth disparity as the background wooden part. This time we applied the contrast enhancement for both images but we don't see a real improvement compared to a single application.