CSC320H1, Winter 2019
Assignment 4

Name: JingXi, Hao
Student Number: 1000654188

Due Date: 12:00pm, Thursday, April 3, 2019

Part 4.1 Report And Experimental Evaluation For NLM Algorithm

In this section, we put the NLM algorithm into test in order to analyze its performance based on the distinct assignments of the parameters: the maximum search radius, w, the number of the nearest neighbours, k, and the patch size, p. Note that the algorithm is executed on several test images, including the given $Jaguar2$ image and the image taken on my own.

1. **Experiment 1 – Jaguar2**

   - Note that the default values for the parameters are:
     - w = 630 (maximum image dimension)
     - k = 4
     - p = 7
     - iters = 3 (fixed value here, not gonna test on this)

   - First, we show the source and target image on which the NLM algorithm will execute. In this case, we have the source and target image to be the same.

Figure 1: Source & Target

- **Test on the maximum search radius, w**

  In order to test the maximum search radius, w, we set other parameters with default values and run the NLM algorithm with several different values of w. In this case, we try with $w = 315,\ 630$, and 1260, where 315 is half of the default w and 1260 doubles the default w and 630 is the default value for w.

  - We show the NLM output images based on different w values, including the default one.

Figure 2: NLM with $w = 315$ – Half of the default w

Figure 3: NLM with $w = 630$ – Default w

Figure 4: NLM with $w = 1260$ – Double the default w

Based on the NLM output images shown above, there are no obvious enhancements or changes for the outputs when the value of w increases. However, if we take a closer look into the zoomed in output images, we still can see that the outputs get less noisy as w increases. This makes much sense since the number of the searched patches in the random search process would increase as w increases, which implies that the performance would get better when w increases. However, the large amount of increase in w only cause a small amount of increase in the number of the examined patches in the random search process. Thus, this is the reason why there are no obvious improvements in the outputs but the performance actually gets better slightly.

- **Test on the number of the nearest neighbours, k**

  In order to test the number of the nearest neighbours, k, we set other parameters with default values and run the NLM algorithm with several different values of k. In this case, we try with $k = 1,\ 4,\ 7$, and 15, where 4 is the default value for k.

  - We show the NLM output images based on different k values, including the default one.

Figure 5: NLM with $k = 1$

Figure 6: NLM with $k = 4$ – Default k

Figure 7: NLM with $k = 7$

Figure 8: NLM with $w = 15$

This case is designed to test the effect of the number of the nearest neighbours, k, on the NLM algorithm. It is expected to generate better results as k increases, which means that the less noisy outputs would be generated by the NLM algorithm as k increases. According to the

NLM output images shown above, we can clearly see that there are significant improvements in the NLM output images with the increasing value of k. This is because that each patch gets more accurate match due to comparing with more neighbour patches in the PatchMatch process. Hence, we can conclude that the NLM would enhance its performance as k increases.

- **Test on the patch size, p**

  In order to test the patch size, we set other parameters with default values and run the NLM algorithm with several different values of p. In this case, we try with $p = 3,\ 7$, and 11.

  – We show the NLM output images based on different p values, including the default one.

Figure 9: NLM with $p = 3$

Figure 10: NLM with $p = 7$ – Default p

Figure 11: NLM with $p = 11$

Based on the outputs, we can observe that the NLM output image with patch size equals to 3 and 11 are nosier than the NLM output image using $p = 7$. Then, this case indicates that we need to choose a proper patch size in order to attain better results. This makes sense since too large or too small of the patch size would introduce more inaccuracies in the PatchMatch process, which leads to bad effects on the performance of the NLM process. Therefore, select a proper patch size p is essential to enhance the performance of the NLM algorithm.

2. **Experiment 2 – Building**

- The purpose of this experiment is to disagree or confirm the conclusions we obtained from the last experiment to see whether those conclusions are random or solid.

- Note that the default values for the parameters are:
  - w = 564 (maximum image dimension)
  - k = 4
  - p = 7
  - iters = 3 (fixed value here, not gonna test on this)

- First, we show the source and target image on which the NLM algorithm will execute. In this case, we have the source and target image to be the same.

Figure 12: Source & Target

- **Test on the maximum search radius, w**

  In order to test the maximum search radius, w, we set other parameters with default values and run the NLM algorithm with several different values of w. In this case, we try with $w = 282,\ 564$, and 1128, where 282 is half of the default w and 1128 doubles the default w and 564 is the default value of w.

  – We show the NLM output images based on different w values, including the default one.

Figure 13: NLM with $w = 282$ – Half of the default w

Figure 14: NLM with $w = 564$ – Default w

Figure 15: NLM with $w = 1128$ – Double the default w

According to the NLM output images, we are able to see that still there are no obvious changes or improvements in the condition of the outputs. However, it gets slightly better if we zoom in and compare each small area in detail. Therefore, this certainly consents the conclusion we obtained from **Experiment 1**, which is that increase the value of w can slightly enhance the NLM algorithm performance but not by much. This is due to that we have a really small amount of increase in the number of the examined patches during the random search process but the increase amount is too small to produce significant improvements.

- **Test on the number of the nearest neighbours, k**

  In order to test the number of the nearest neighbours, k, we set other parameters with default values and run the NLM algorithm with several different values of k. In this case, we try with $k = 1,\ 4,\ 7$, and 9. Note that $k = 9$ is tested instead of $k = 15$ here since it takes too long to run the NLM algorithm due to the larger image size in this case.

  - We show the NLM output images based on different k values, including the default one.

Figure 16: NLM with $k = 1$

Figure 17: NLM with $k = 4$ – Default k

Figure 18: NLM with $k = 7$

Figure 19: NLM with $k = 9$

As we can see, it is clearly that the performance of the NLM algorithm gets better since the output image using the larger k contains less noises. Thus, this case confirms our conclusion of how k would affect the performance of the NLM algorithm, which is that NLM algorithm produces better outputs when the value of k increases. This results in comparing with more neighbour patches in the PatchMatch process would find more accurate match for each patch in the source image. Therefore, this would increase the denoise performance.

- **Test on the patch size, p**

  In order to test the patch size, we set other parameters with default values and run the NLM algorithm with several different values of p. In this case, we try with $p = 3,\ 7$, and 11.

  - We show the NLM output images based on different p values, including the default one.

Figure 20: NLM with $p = 3$

Figure 21: NLM with $p = 7$ – Default p

Figure 22: NLM with $p = 11$

The NLM output images shown above reveals that the outputs get nosier when $p = 3$ and $p = 11$ in comparison with the output using $p = 7$. This indicates that the proper patch size would result in better performance for denoise process. Hence, this also agrees with the results gotten from the first experiment. Once again Too large patch size would introduce more unuseful information for each patch and too small patch size would lose some important characteristics of each patch. Therefore, those two cases would increase the chance of getting a wrong match. Thus, in order to enhance the performance of the NLM algorithm, a proper patch size need to be selected.

3. **Conclusion**

- The performance of the NLM algorithm gets better (the output becomes less noisy) when the maximum search radius, k, increases.
- The performance of the NLm algorithm gets better (the output becomes less noisy) when the number of the nearest neighbours, k, increases.
- The performance of the NLM would be affected by the selection of the patch size, p. Also, it is essential to pick a proper patch size, p, in order to enhance the NLM algorithm performance.

Part 4.2 Results For Generalized PatchMatch On Jaguar2

Note that we execute the generalized PatchMatch algorithm on Jaguar2 with $k = 3$, $w = 630$, $p = 7$, $alpha = 0.5$, and $iters = 3$. First, we show the source and target images.

Figure 23: Source

Figure 24: Target

Then, we show the results for generalized PatchMatch on Jaguar2.

Figure 25: NNF-COL in order 0

Figure 26: Reconstructed Source in order 0

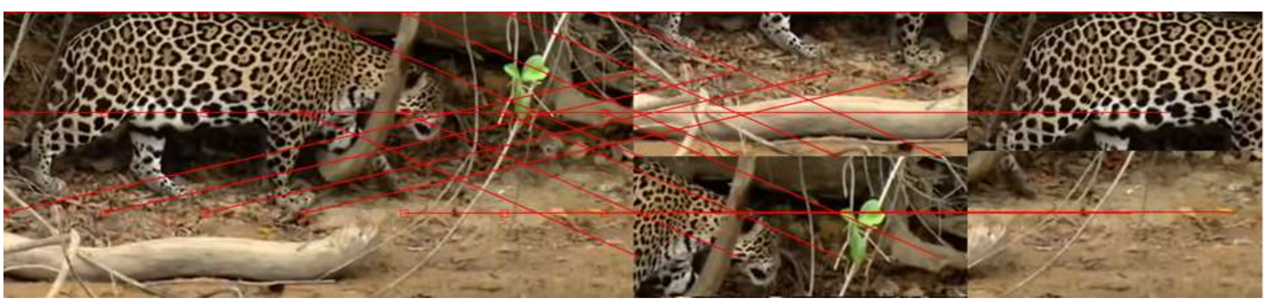

Figure 27: NNF-VEC in order 0

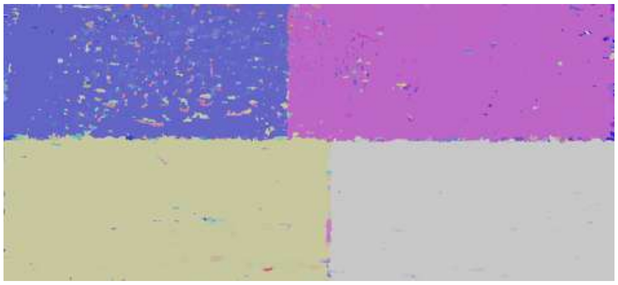

Figure 28: NNF-COL in order 1

Figure 29: Reconstructed Source in order 1

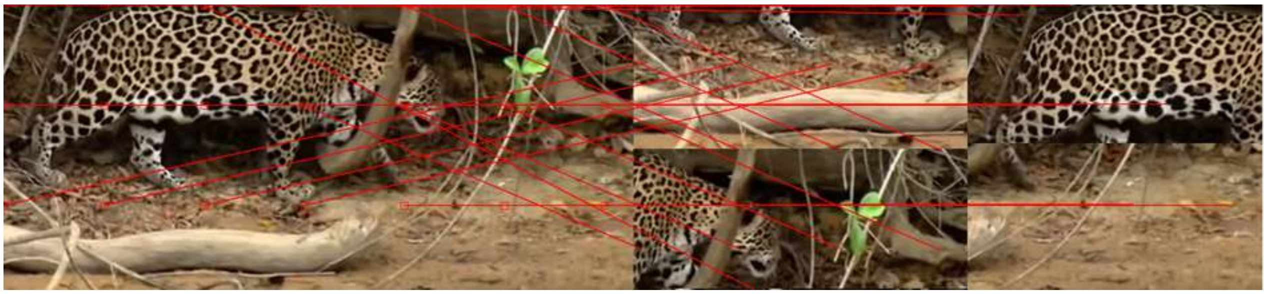

Figure 30: NNF-VEC in order 1

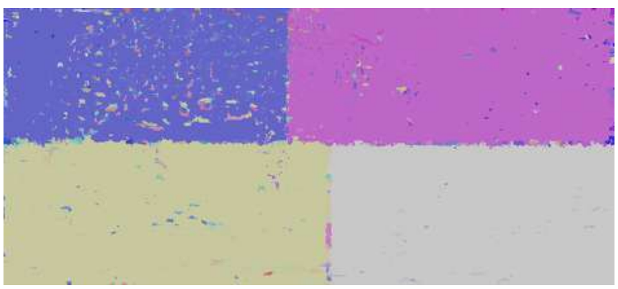

Figure 31: NNF-COL in order 2

Figure 32: Reconstructed Source in order 2

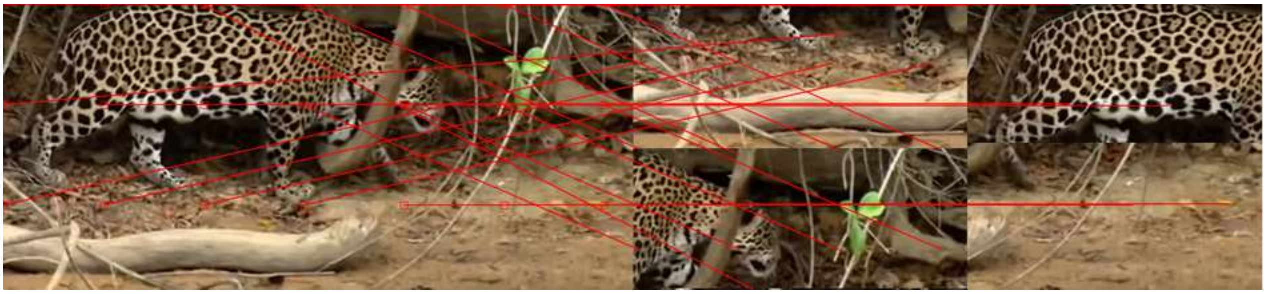

Figure 33: NNF-VEC in order 2

As we can see that $order0$ produces better results. This makes sense since $k = 0$ stores the best match found in the target image for each patch in the source image based on our implementation of the generalized PatchMatch algorithm.