COMP90086: Assignment 3 Report

Tanzid Sultan (ID# 1430660)

October 17, 2023

1. Existing Code: Reconstruction Error Analysis

For this section, we have chosen 5 different mask locations (given in pixel coordinates) for our two test images (*brick.png* and *basket.png*). Since both of these images are of comparable size, we have used the same 5 mask locations for both. The size of the masked region is 50x50 pixels. The table below shows the rms error of the reconstruction for each of the mask locations for the two images, computed using the original parameter settings in the provided code.

Mask Location	**Brick rms error**	**Basket rms error**
(200, 100)	35.72	46.59
(97, 230)	33.89	49.36
(315, 200)	32.21	51.30
(400, 80)	33.12	35.75
(560, 255)	32.80	53.57
Average	**33.55**	**47.31**

Table 1: Reconstruction rms error for each different mask locations and the average over all 5 masks. The locations are given as the coordinates (x, y) of the top-left corner pixel of the square mask region, in pixel units.

We note that there are no significant variations in the rms error at different mask locations from the average error over all locations, which is not surprising because our test images contain periodic texture and so the same pattern is repeated at different locations within the image. However, we also note that the average rms error of the basket image is a bit higher than that for the brick image. This seems to be a consequence of the fact that both the brick and basket images contain a periodic texture, but only the basket image contains an additional small stochastic component, due to the slight imperfections/irregularity in the basket weaves. The randomness of the weave imperfections are difficult to reconstruct because our infilling algorithm is *non-parametric* and designed specifically to generate periodic textures. This may explain why our algorithm performs better on the brick image.

2. Context Size and Shape

In this section, we analyze the effects of varying the size and shape of the context window used for finding the most most suitable pattern to use for reconstructing each masked pixel. In particular, we allow the context window to have a rectangular shape (instead of just using a square context window) and try a range of different sizes, however we still maintain the bottom right corner pixel of the window as the target pixel. We perform this experiment individually on each of our test images.

Window Size	**Rms Error**	**Run Time (sec)**
(2, 2)	62.77	10
(5, 5)	35.72	50
(15, 8)	32.55	215
(26, 12)	27.96	530
(42, 22)	24.49	1412

Table 2: Brick image reconstruction rms error and computation time for different context window sizes, given as *(width, height)* in pixel units.

Table 2. shows the results obtained for the brick image. We observe that when the context window is too small (2x2), the recontruction rms error is very high. Conversely, the rms error sharply decreases as the context window becomes larger. To understand why this is the case, we first note that our image contains a periodic texture pattern of vertical and horizontal lines. When the window is too small, it is unable to capture the scale of the full periodic structure and the global details in the texture, leading to poor reconstruction quality. As we increase the size of the context window, we start to capture some of the regularity of the texture in our reconstruction, however we see some artifacts, for instance we obtain bricks of incorrect width and we sometimes see that the reconstructed vertical lines don't align with the bottom part of the image. These artifacts are due to the context window not being large enough and also the fact that we reconstructed pixels in a left-to-right top-down fashion. When we make the context window large enough (which for the case of the bricks image would be roughly the size of a brick which is around 40 pixels wide and 20 pixels high), it is able to capture the full periodic structure and we obtain very good reconstruction with no visible artifacts.

Table 3. shows the results obtained for the basket image. Like the brick image, the basket image also contains a strong periodic structure, however it also contains a small stochastic component, due to the imperfections in the basket weaves. For this reason, even when the context window is not large enough to capture the full scale of the periodic structure, we find that the artifacts generated by the reconstruction are visually very similar to the random imperfections in the original texture of the basket weaves and so the quality of the reconstruction seems to be fairly good and further increasing the window size does not cause any significant change in our perceived quality. Furthermore, this observation is directly confirmed by the fact

Window Size	**Rms Error**	**Run Time (sec)**
(2, 2)	50.07	14
(5, 5)	46.59	57
(15, 8)	45.10	238
(26, 12)	47.56	579
(50, 25)	43.82	2075

Table 3: Basket image reconstruction rms error and computation time for different context window sizes, given as *(width, height)* in pixel units.

that the rms error for the basket image does not decrease appreciably as we make the context window larger (unlike for the brick image).

In summary, a larger context window will lead to better reconstructions of periodic textures (but may not improve recontruction of stochastic textures). We also note that a larger context window requires more computational overhead and we see from the tables that the compute time roughly scales linearly with the area of the context window (i.e. $time \propto width \times height$).

3. Reconstruction Design

In this section, we augment the design of our reconstruction algorithm with some new ideas in order to obtain better reconstruction quality. We have specifically explored a combination of the following 3 ideas:

1. **Inward-Spiral Inpainting Order:**In the original algorithm, we start at the top-left corner of the the masked region and begin infilling by reconstructing the pixels left-to-right in a top down raster-scanning fashion. This order of filling the pixels can lead to boundary artifacts in the reconstruction. In particular, for periodic textures such as the brick wall image, this order of infilling can cause the vertical and horizontal lines at the boundaries of the masked region to be misaligned with the rest of the image, specially at the right and bottom sides of the boundary (this order of infilling only does a good job at aligning the top and left sides of the boundary, which is not surprising). To alleviate this problem, we have instead used a spiral infilling order, where we first infill the row along the top edge of the masked region going left to right, then we infill the column along the right edge going from top to bottom, followed by the row along the bottom edge going right to left and finally the column along the left-edge going bottom to top. We repeat this procedure such that we are infilling along the boundary of the masked region, spiralling inwards (in clockwise order). This way, we use context patterns from all four directions (e.g. the context windows for the left to right infilling scan contains the target pixels in the bottom right corner, the context windows for the top to bottom infilling scan contains the target pixels in the bottom-right corner, etc.) and this leads to

a more accurate and better quality reconstruction.

2. **Normalized Cross-Correlation and Sum of Absolute Diffferences:** Instead of using sum of squared differences (SSD) to measure similarity between the masked region context pattern and candidate context patterns, we tried using the *normalized cross-correlation* (NCC) and *sum-of-absolute differences* (SAD) metrics. For NCC, each pattern is first normalized by subtracting off the mean pixel intensity within the pattern and dividing by the standard deviation of the pixel intensity. Then the similarity score is computed by taking the cross-correlation between two different patterns. A higher cross-correlation value corresponds to a higher similarity score. Then during reconstruction, the candidate pattern is re-normalized to match the brightness level of the masked region context. The main advantage of using NCC is that this similarity measure is robust against brightness differences when comparing patterns, unlike SSD. Since SSD is sensitive to the pattern brightness, it may miss the true best candidate pattern if that pattern happens to have a drastically different brightness, while NCC will correctly identify that best candidate. The SAD metric also has advantages over SSD, mainly SAD is less sensitive to outlier pixels in the pattern making it more robust than SSD in most circumstances. For our two test images, we have found both NCC and SAD metrics yield better quality reconstructions, however SAD tends to perform slightly better, which may be due to the fact that our test images have roughly uniform brightness.

3. **Inpainting by Blocks:** Instead of reconstructing a single pixel at a time, we tried reconstructing blocks of pixels. This is achieved by assigning a block of target pixels within a context window pattern instead of a single target pixel. The main advantage of infilling by blocks rather than by single pixels is that it greatly speeds up the computations, which in-turn makes it more feasible for us to use even larger context windows.

By incorporating these improvements into our algorithm, we were able to achieve better reconstruction quality using smaller context windows. Figure 1. shows a comparison of reconstruc-

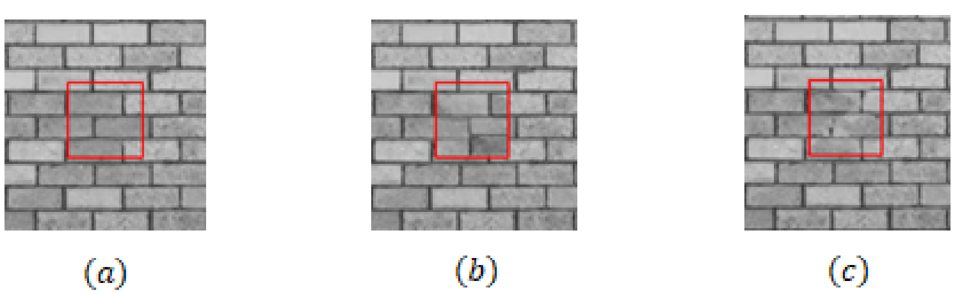

Figure 1: (a) Original image, (b) reconstruction with original algorithm, (c) reconstruction with improved algorithm. The red box denotes the masked region.

tions obtained for the brick test image using the original algorithm and the improved algorithm, keeping the context window size the same. The improved algorithm uses sum of absolute differences to select the best candidate pattern and clockwise spiral infilling order. Figure 1(b) shows the reconstruction obtained using the original algorithm with rms error of 32.55. We can clearly see the misalignment of the vertical lines with the rest of the image under the bottom edge of the masked region and misalignment of the horizontal lines across the right edge. Note that the reconstruction using the inward spiral infilling, shown in Figure 1(c), has eliminated these misalignment artifacts, while achieving a much lower rms error of 27.28. Moreover, the improved reconstruction is also able to accurately capture the periodicity of the texture even though the context window is not large enough. The use of the SAD metric to select the best candidate also results in a more consistent brightness/contrast across the reconstructed region.