Image Quiliting

Ola Pietka

March 2023

Introduction

The goal of this project was to implement the image quilting algorithm for texture synthesis and transfer, described by Efros and Freeman (2001).

Image quilting is a technique used in computer graphics and image processing to generate new images by stitching together small patches of existing images. Image quilting works by dividing the input image into overlapping patches, and then selecting a set of patches that closely match the target region to be filled in. The selected patches are then stitched together in a way that minimizes visible seams and produces a coherent output.

Algorithm Comparison

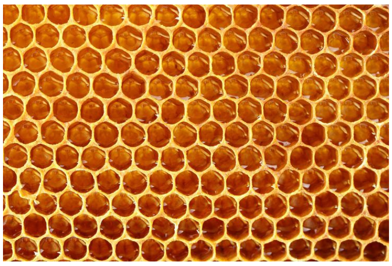

Figure 1: Original image (1) (Pinterest).

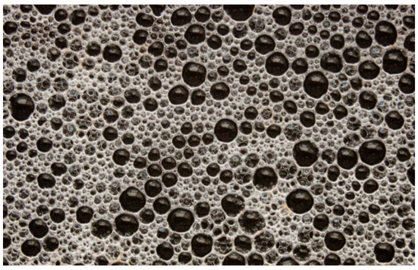

Figure 2: Original image (2) (DesignBeeb).

Figure 3: Original image (3) (iStock).

Randomly Sampled Texture

Image quilting with randomly sampled texture is a variation of the image quilting technique that involves generating a new texture by randomly sampling patches from an input texture.

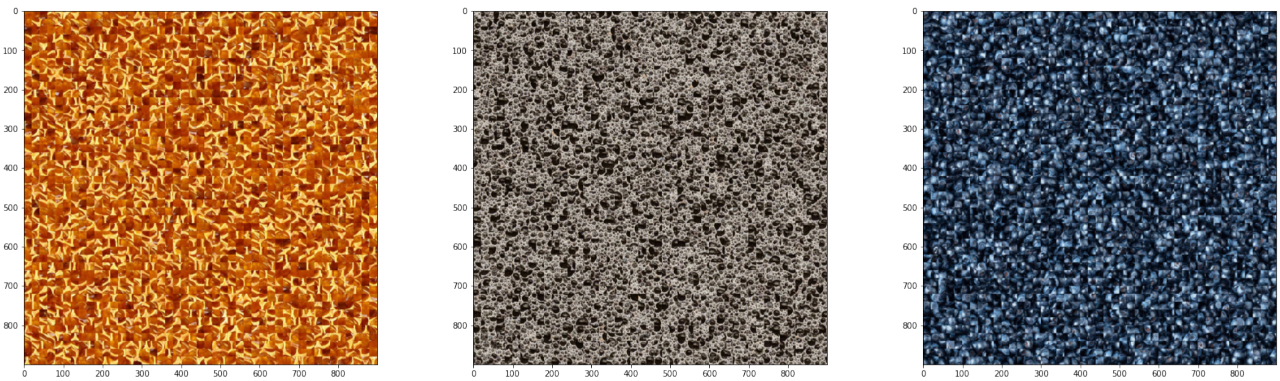

Figure 4: Random sampled textures for image 1, 2, and 3 (*output size=900, patch size=60*).

Overlapping Patches

In this technique, the input image is divided into overlapping patches, and a set of patches is selected to fill in the target region. The selected patches are then blended together along their overlapping edges to create a seamless output.

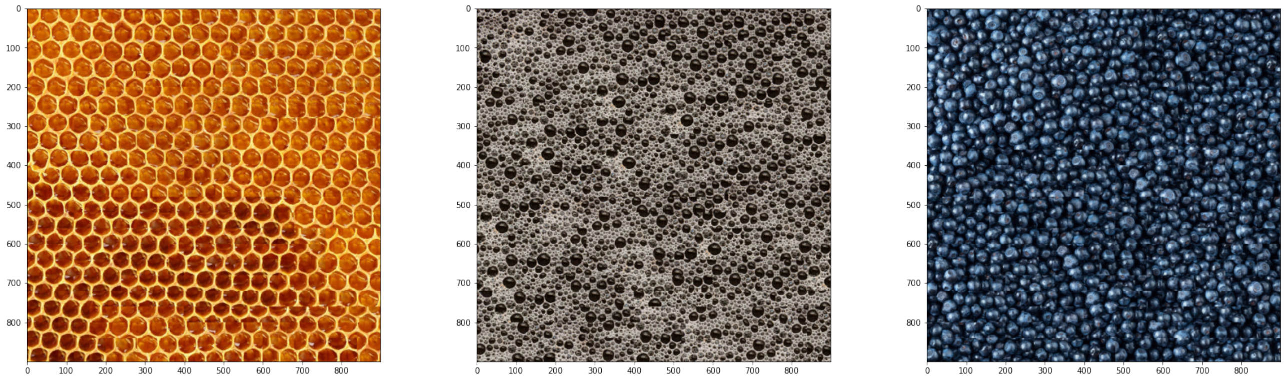

Figure 5: Overlapping texture for image 1, 2, and 3 (*output size=900, patch size=60, overlap=20, tol=10*).

Seam Finding

The goal of seam finding is to identify and remove visible seams between adjacent patches or images, and produce a seamless output that looks like it was created from a single image.

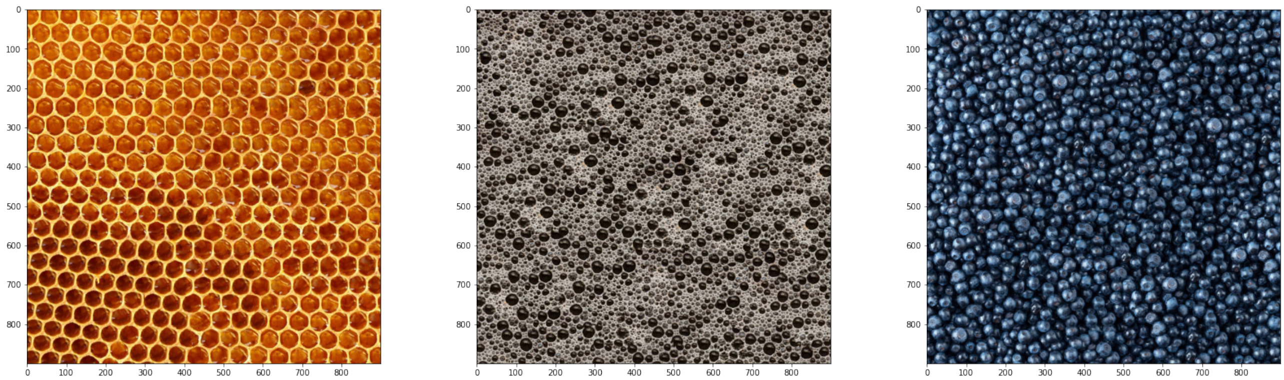

Figure 6: Seam finding texture for image 1, 2, and 3 (*output size=900, patch size=60, overlap=20, tol=3*).

Dynamic programming is a commonly used technique for seam finding in image quilting. We first define source patch and a patch (target) that closely match the source region to be filled in (Figure 7). Next, we compute cost matrix for overlapping regions, and find the minimum-energy seam that separates the the regions (Figure 8). The seams are then stitched together and applied into target and source patch to create coherent output (Figure 9).

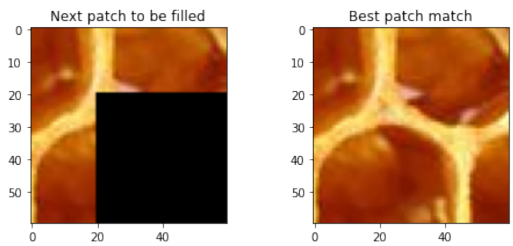

Figure 7: Left: patch to be filed. Right: best patch mach based on SSD.

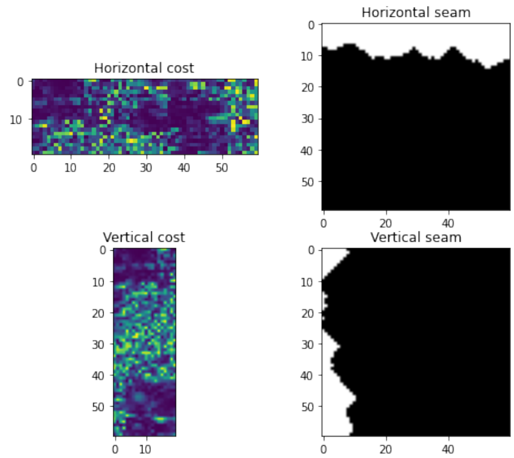

Figure 8: Left: cost matrix. Right: seam mask.

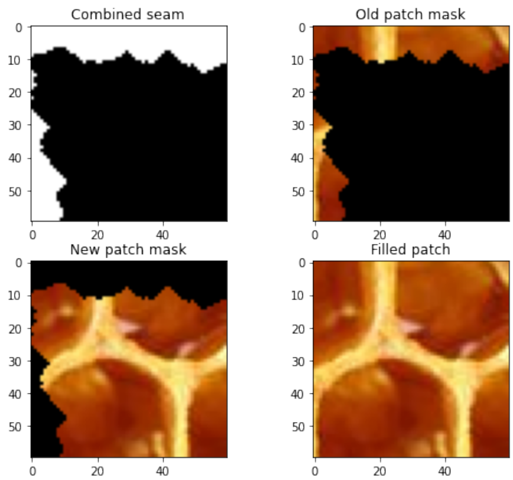

Figure 9: Top-left: combined seam. Top-right: source patch with mask. Bottom-left: target patch with mask. Bottom-right: filled patch.

Texture Transfer

Texture transfer is a technique that involves transferring the texture of one image to another image, while preserving the structure and content of the target image. The goal of texture transfer is to create a new image that has the same visual appearance and texture as the source image, but with a different content.

The parameters used in texture transfer include:

- Patch size: The size of the patches used in the texture transfer. Smaller patches can capture more fine-grained texture details, while larger patches can capture more global structure.
- Overlap size: The size of the overlapping regions.
- Alpha: Regularization factor employed to manage the balance between the replication of texture and the accuracy of the match to the intended image correspondence map.
- Tolerance: Determines from how many lowest cost patches choose the target patch.

Figure 10: Texture image (Darchuk).

Figure 11: Guidance image (Haberler).

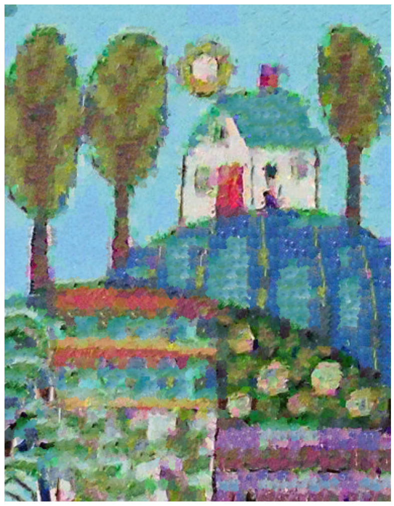

Figure 12: Output image.

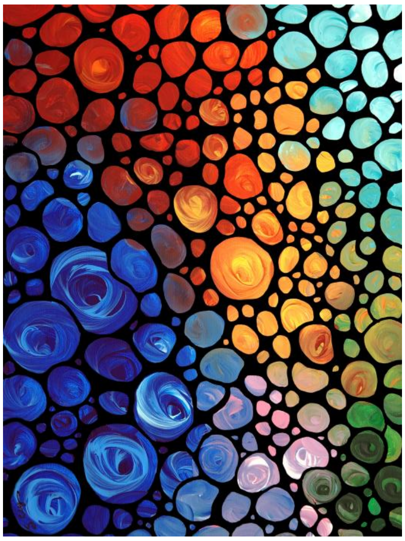

Figure 13: Texture image (Cummings).

Figure 14: Guidance image (Adamkova).

Figure 15: Output image.

Iterative Approach

Iterative texture transfer works by gradually refining the transfer of a texture from a source image to a target image through a series of iterations. The differences between the non-iterative approach are:

- The patch size and overlap size are decreased on each iteration by some fixed *reduction factor*.
- The alpha at each iteration i is set to: $\alpha_i = 0.8 * \frac{(i-1)}{(N-1)} + 0.1$.
- The error cost matrix is $(ssd_{overlap} + ssd_{prev}) * \alpha_i + ssd_{target} * (1 - \alpha_i)$.

Iterative approach is often better than non-iterative approach because it allows for a gradual refinement of the texture transfer, resulting in better quality transfers and more accurate preservation of the source texture's details and characteristics. By iteratively adjusting the transfer parameters, the algorithm can better capture the nuances and subtleties of the source texture, leading to a more natural and convincing transfer. Iterative approach is also more robust to variations in the source and target images. Since the transfer is optimized gradually, the algorithm can adapt to changes in the input images and adjust the transfer parameters accordingly. This can make the transfer more stable and less prone to artifacts and distortions.

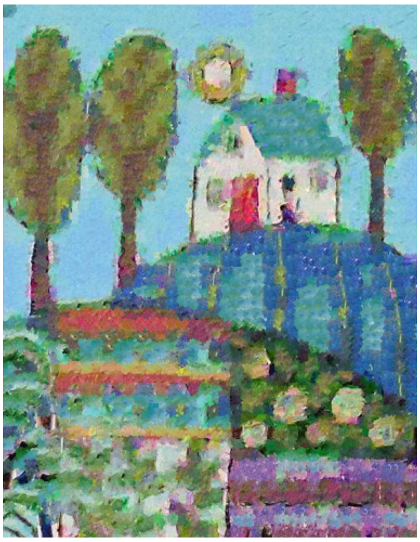

Figure 16: Non-iterative approach.

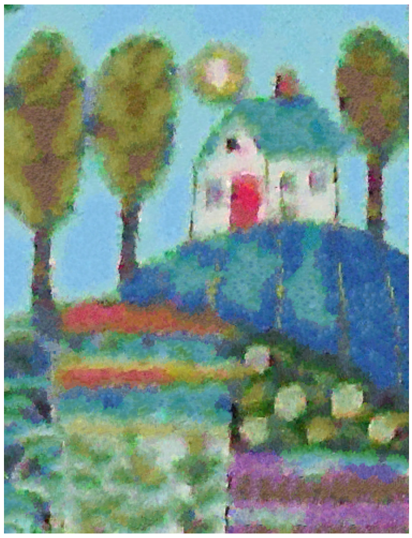

Figure 17: Iterative approach.

Figure 18: Non-iterative approach.

Figure 19: Iterative approach.

References

Marcela Adamkova. Folk art landscape image.

Sharon Cummings. Mosaic art image.

Olha Darchuk. Spring trees art image.

DesignBeeb. Foam texture image.

Alexei Efros and William Freeman. 2001. Image quilting for texture synthesis and transfer. page 527–532.

Yeni Haberler. Folk art house image.

iStock. Blueberry texture image.

Pinterest. Honey texture image.