CS 663 Course Project

Image Quilting for Texture Synthesis and Transfer

170050077, 170050079, 170050081

Overview:
Our project is based on the paper given here.

We perform texture synthesis where we create a larger texture using patches of an existing texture and stitching them together using the algorithm described below.

We also perform texture transfer i.e transfer texture from one object to another. We use a slight modification of the texture synthesis algorithm for this.

Overview of the algorithm:

Texture Synthesis:

- Go through the image to be synthesized in raster scan order in steps of one block(minus the overlap)
- For every location, search the input texture for a set of blocks that satisfy the overlap constraints(above and left) within some error tolerance. Randomly pick one such block.
- Compute the error surface between the newly chosen block and the old blocks at the overlap region. Find the minimum cost path along this surface and make that the boundary of the new block. Paste the block onto the texture. Repeat.

Minimum Boundary Cut:
We use the minimum boundary cut between two overlapping blocks on the pixels where two textures match the best.

We compute cumulative minimum error for all paths:

$$E_{i,j} = e_{i,j} + min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$$

Then use backpropagation to then compute the cut.

Texture Transfer:
Texture transfer does two additional things than texture synthesis

- Error term of the image quilting algorithm is now the weighted sum of, α times the block overlap matching error plus (1 - α) times the squared error between the correspondence map pixels within the source texture block and those at the current target image position.
- Here we iteratively apply the transfer approach until we get a visually pleasing result. We reduce block size for each iteration, we use the output of the previous iteration for the target image. Most of the time, we set α in ith iteration to be 0.8*(i-1)/(N-1) + 0.1

Our Implementation of the algorithm:

We applied the simple and straightforward implementation of the algorithm, without making many optimizations to it. Block sizes for synthesis were selected depending on the texture.

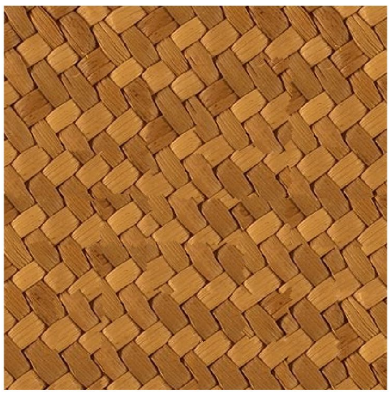
Comments on the results obtained:
(obtained results are given below)
This algorithm works well for stochastic structures and semi-structured textures.
With a stochastic texture and small block size, we can reconstruct most of the images
Problems faced by this algorithm:

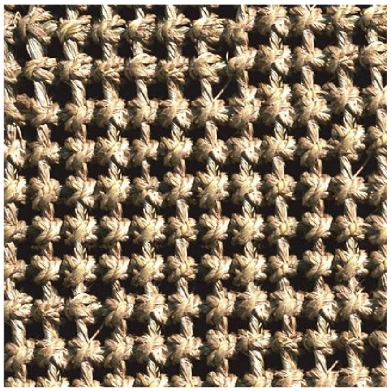
1. Excessive repetition: This happens due to the lack of much randomness(variability) in some images(eg. Berries image).
2. Mismatched or distorted boundaries: This is due to the image having more ordering and less randomness(eg. Cans image).

Some images don't give a visually appealing result for regular block size. In order to get a visually appealing result, we may have to reduce the block size which might be costly(eg. Girl image)

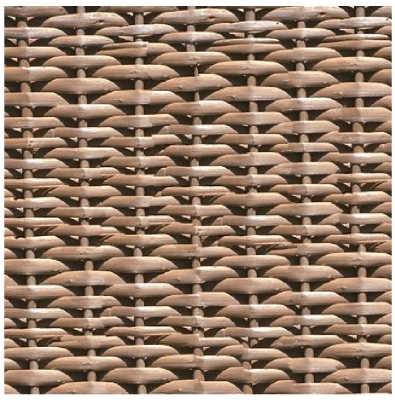
Dataset used:
The textures used below are taken from here.

Results obtained for texture synthesis:
Produced a texture that is roughly double the dimensions of the existing texture.

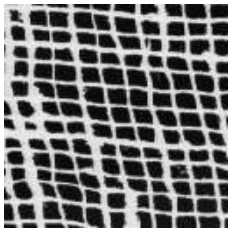
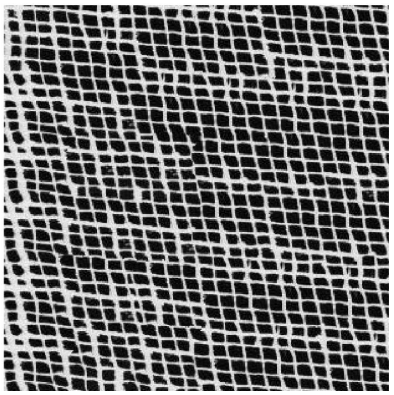


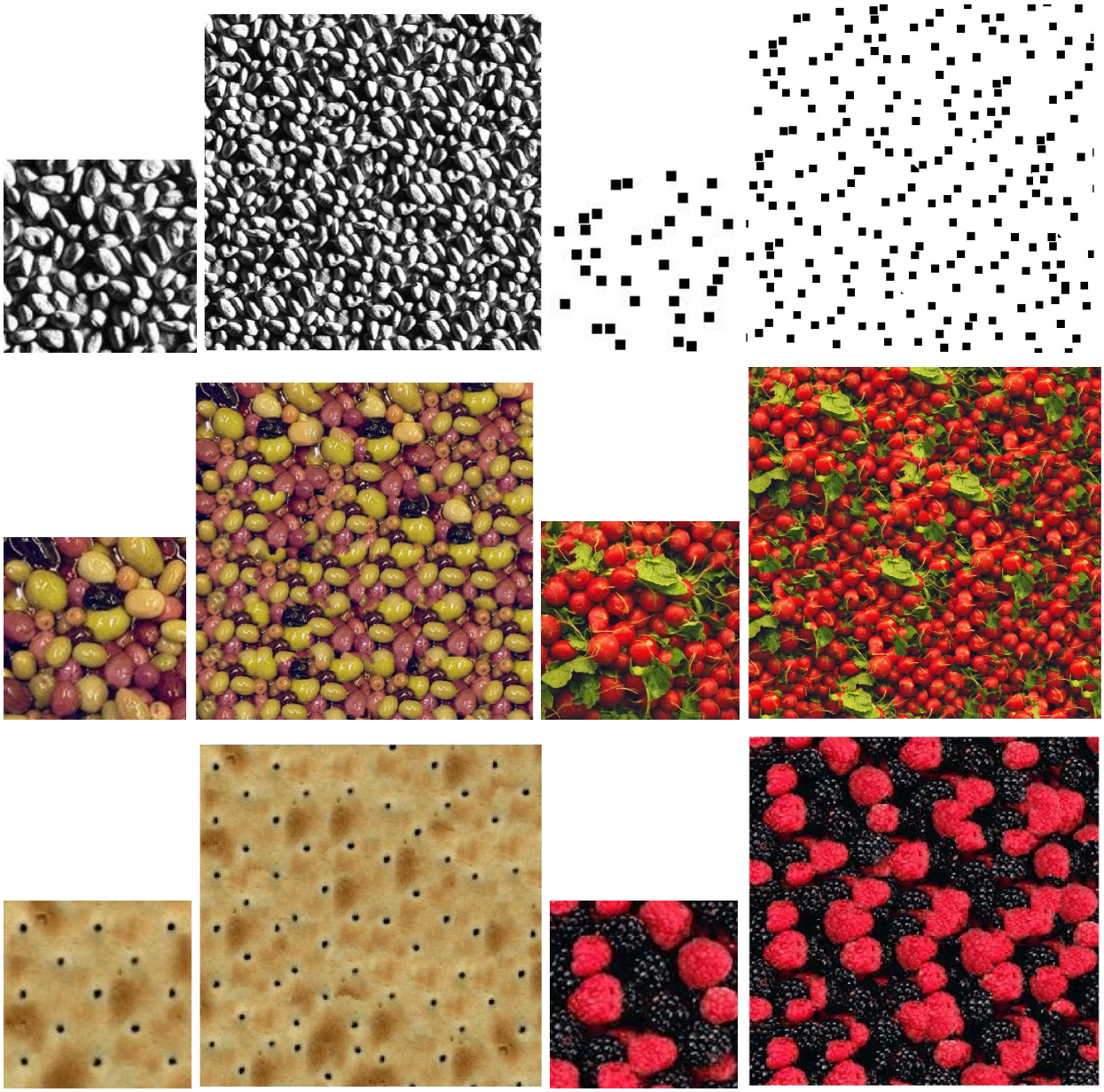








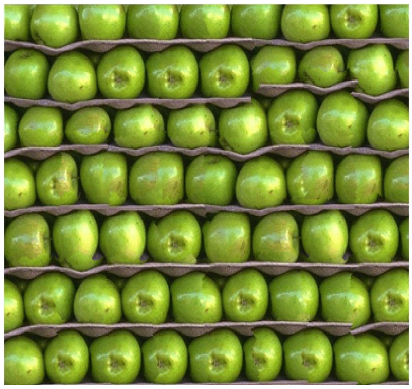












Results obtained for texture transfer: