



Image Quilting for Texture Synthesis and Transfer

(CS 663: Project Presentation)

Shivendra Kumar	(180100111)
Manan Garg	(180100063)
Devansh Chandak	(180110027)

Contributions:

Synthesis: Shivendra + Devansh

Texture Transfer: Manan + Devansh



Introduction

- Our project is based on the paper here (title above):
<http://graphics.cs.cmu.edu/people/efros/research/quilting/quilting.pdf>
- Image Quilting Algorithm:
 - Go through the image to be synthesized in raster (row by row) 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 Error Boundary Cut

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

- After DP above, in the end, the minimum value of the last row in E will indicate the end of the minimal vertical path through the surface and we can trace back and find the path of the best cut through the overlapped region.

Texture Transfer

- If we modify the synthesis algorithm by requiring that each patch satisfy a desired correspondence map, C , as well as satisfy the texture synthesis requirements, we can use it for texture transfer.
- The correspondence map is a spatial map of some corresponding quantity over both the texture source image and a controlling target image. (We have used intensity)
- For texture transfer, image being synthesized must respect two independent constraints:
 - (a) the output are legitimate, synthesized examples of the source texture
 - (b) that the correspondence image mapping is respected
- Hence, we modify the error term by the use of an 'alpha' parameter.

OUR RESULTS

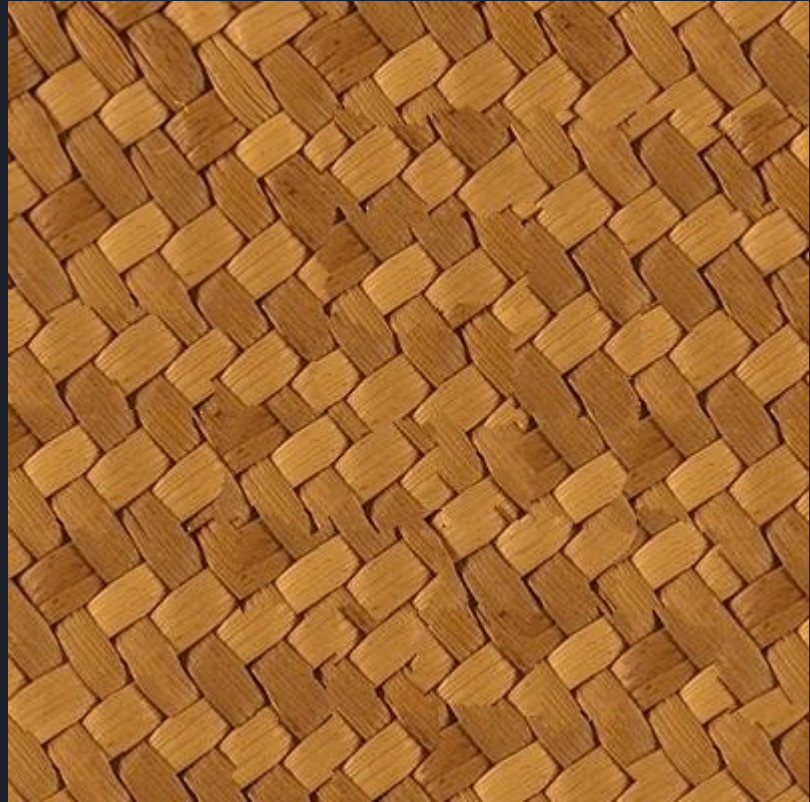
Texture Synthesis



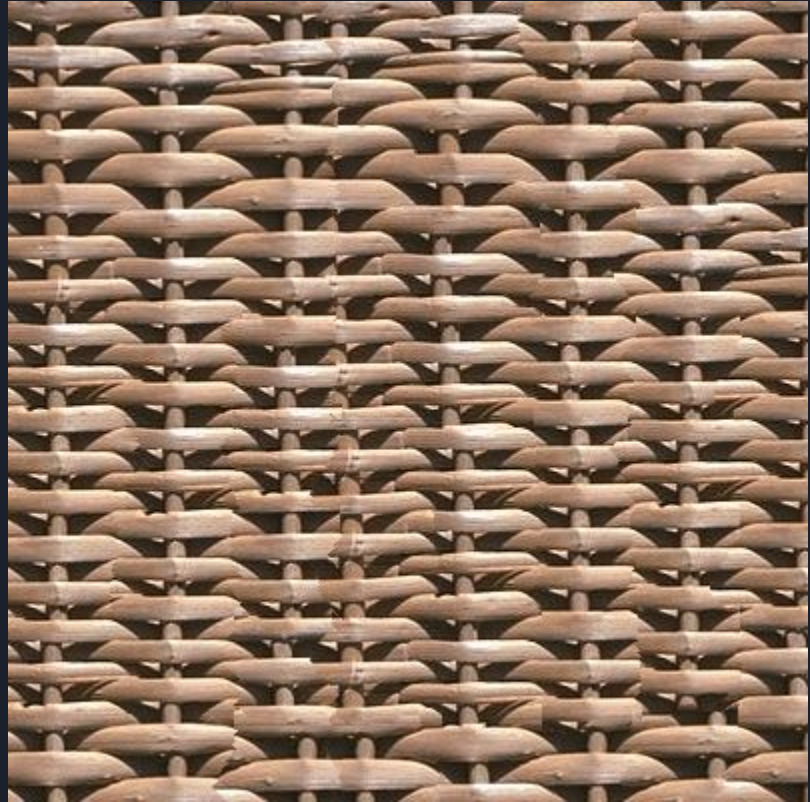
Texture Synthesis



Texture Synthesis



Texture Synthesis



Texture Synthesis



Texture Synthesis



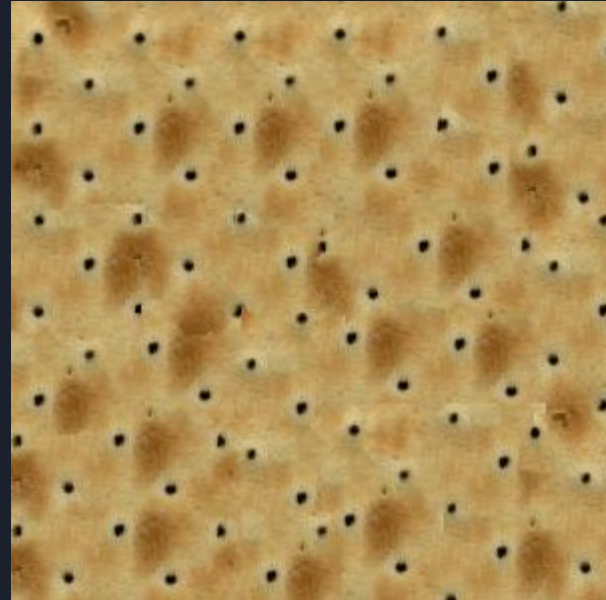
Texture Synthesis



Texture Synthesis



Texture Synthesis



Texture Synthesis



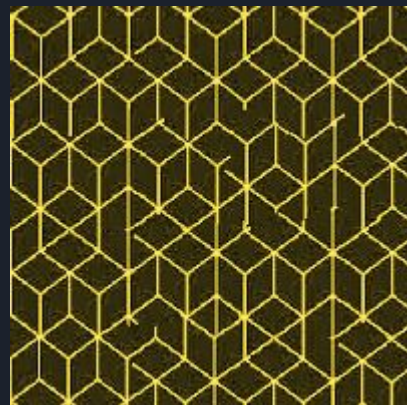
Texture Synthesis



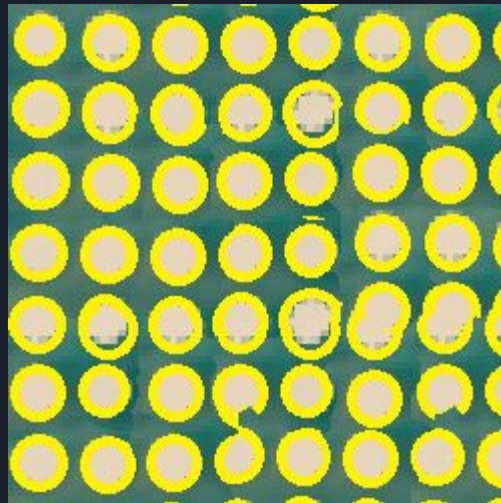
Texture Synthesis on some more images



Texture Synthesis on some more images



Texture Synthesis on some more images

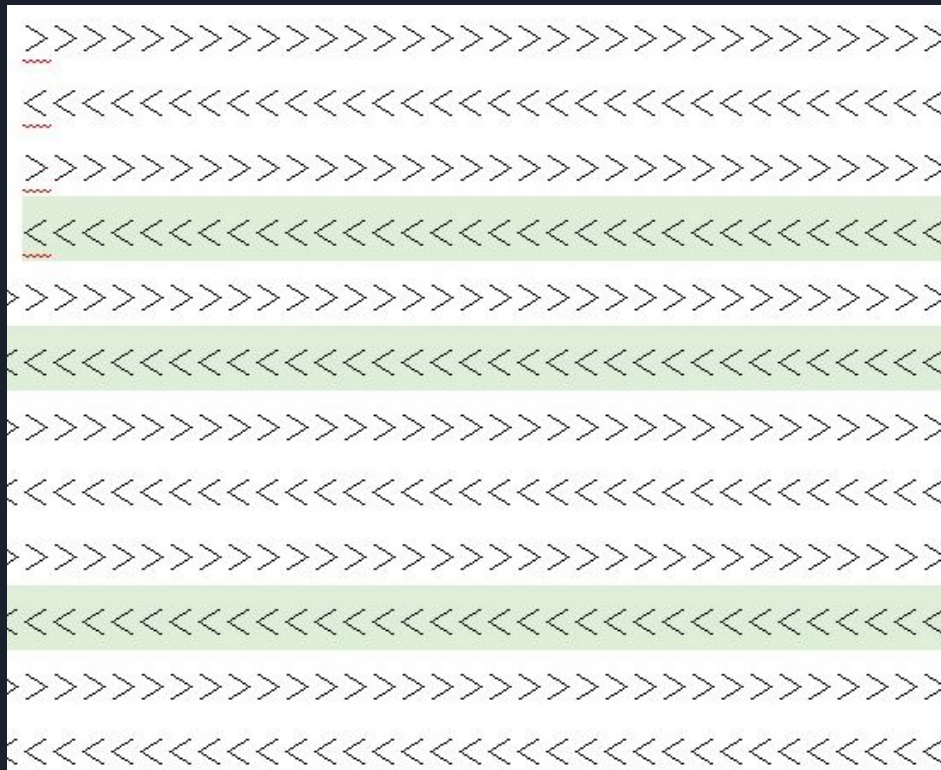
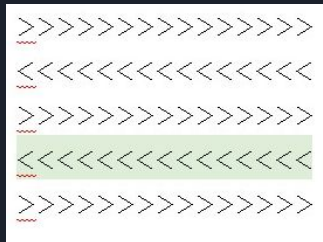


Texture Synthesis on some more images

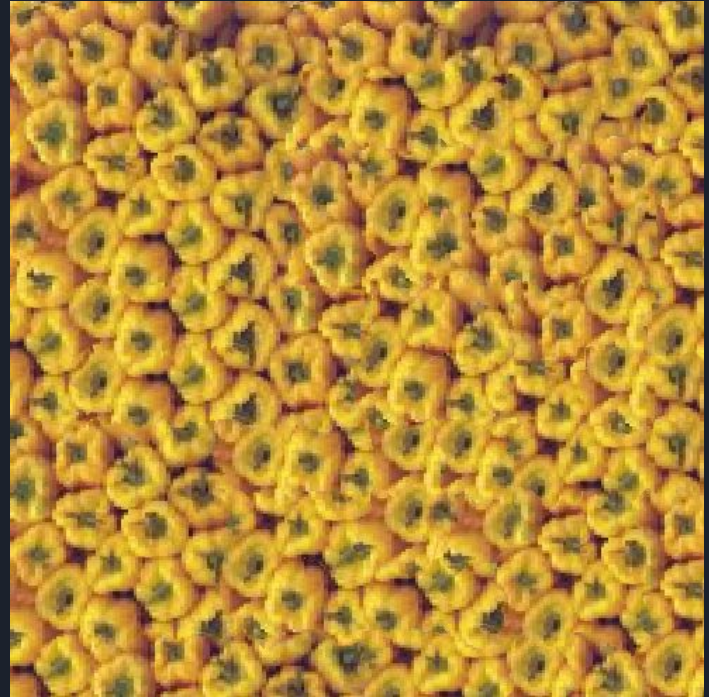
ype methods selec
tion with the varia
ting variables. The
ion model used to
larly effective in c

ype methods seethods seles. Tmethods
tion with the vaith the variused with the
ting variables. Ti variæ. This. Thvariables
ion model used moused used model us
ion with the vare varihe ve variables. the
ing variables. Tes. Thles.es. Tlel usables
ion model used used modeluselusel us
arly effective inive inlv effentiveffective
ion mth the vardel us vanodel with the v
larly efiabiles. Thactive.moffactivariables
ting vadel used with th thodel model us

Texture Synthesis on some more images



Distorted boundaries



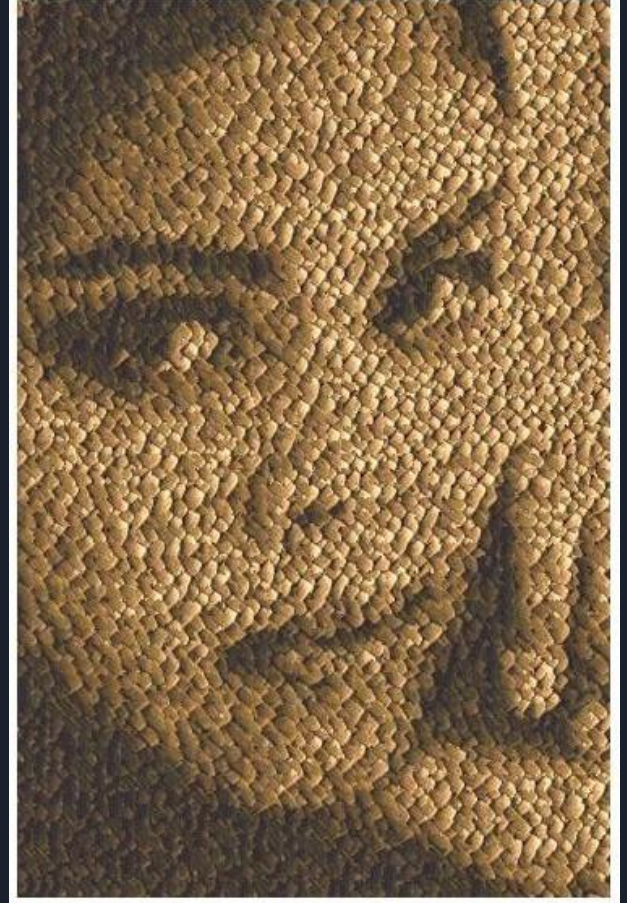
Excessive Repetition



Texture Transfer



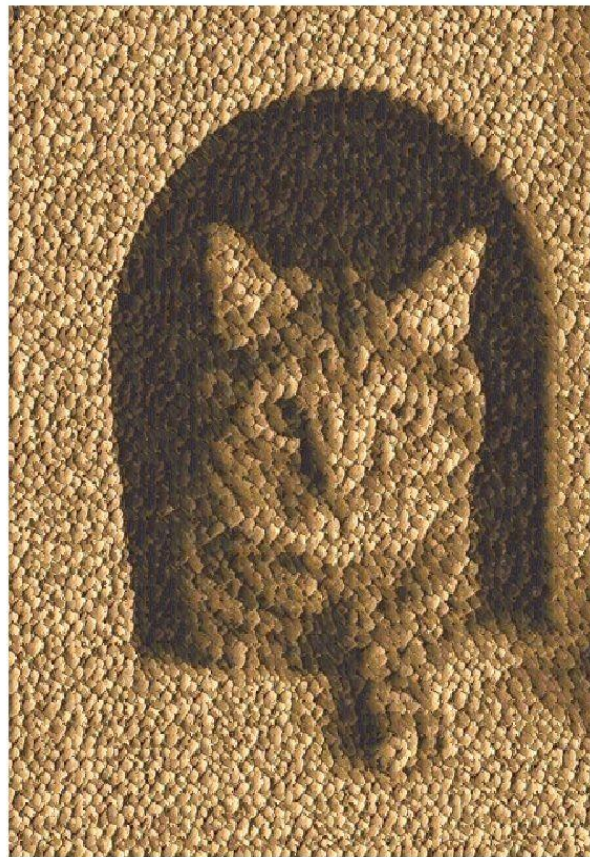
Texture Transfer



Texture Transfer



Texture Transfer



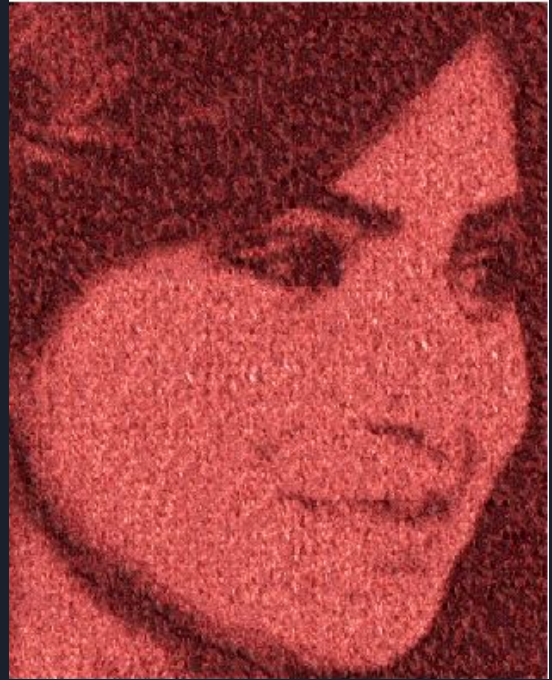
Texture Transfer



Not all look great!



Texture Transfer

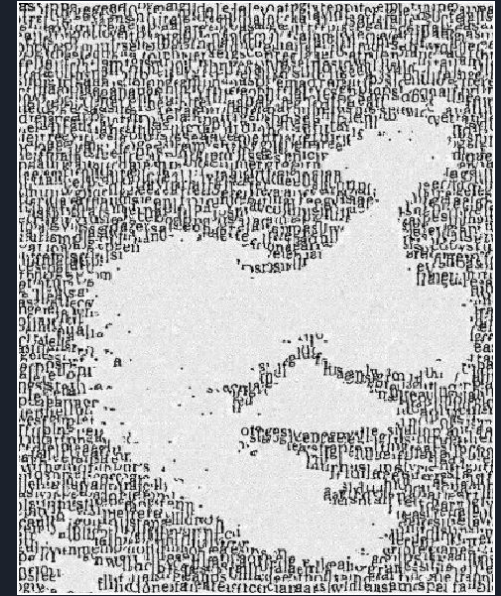


Texture Transfer

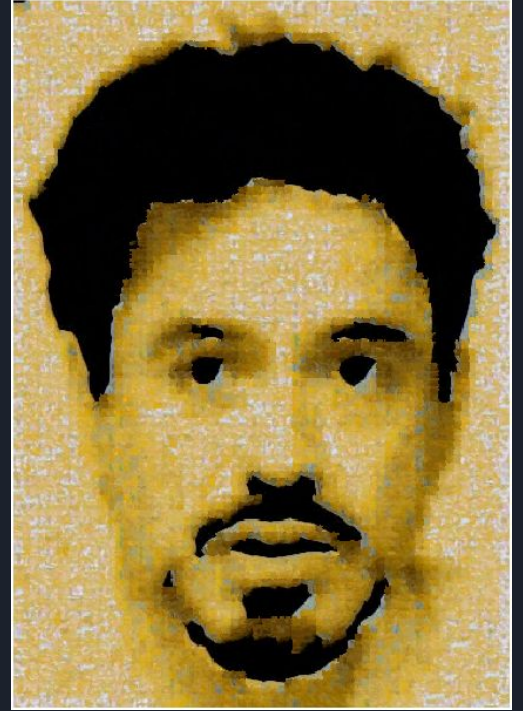
a single conceptual and mat
e the wealth of simple-cell
urophysiologically¹⁻³ and infe
cially if such a framework has
ps us to understand the fun
r way. Whereas no generic
ns (DOG), difference of off
th of simple-cell recreas no
icall¹⁻³ and inferriference



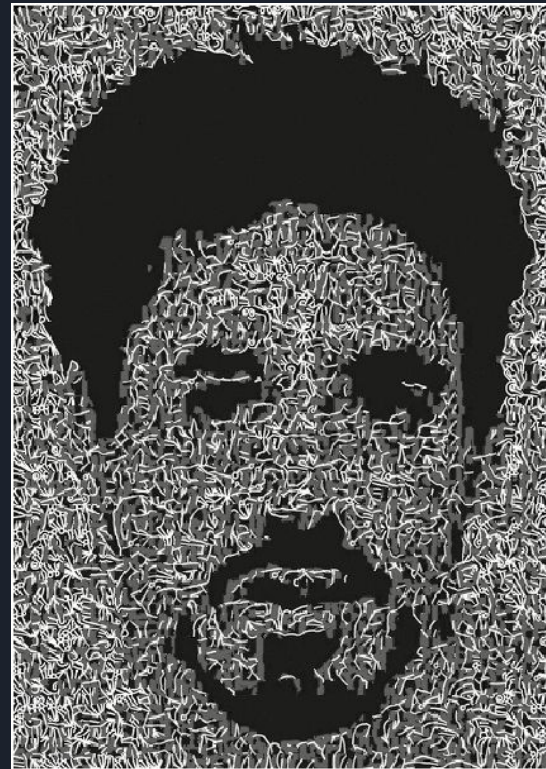
Fb/SraviCreations



A scary one



Texture Transfer





Conclusions

- 1) The Image Quilting algorithm extremely well in case of stochastic textures and fairly well in semi-structured textures
- 2) The time taken to run the algorithm obviously depends heavily on the patch size
- 3) The image quality depends on the texture and the patch size
- 4) The algorithm presented is a very simple yet elegant approach towards solving the problem
- 5) Just like other contemporary texture synthesis algorithm, this is not a perfect algorithm and has some minor drawbacks
- 6) Our algorithm has a complexity of $(MN)^2 / P^2$
- 7) Drawbacks include repeated patterns and minor discontinuity in a few blocks depending on the texture, stochasticity factor and the patch size. It is due to input texture not containing enough variability



Conclusions

- 8) Texture Transfer takes a a long time, as we transfer texture on real portrait images - Images are larger than the texture images and the extra loss factor requires more time to compute as well
- 9) We have used two passes for texture transfer. Multiple passes might yield better results with dynamic weighing factor and changing block sizes in each iteration
- 10) If we look at rdj and orange image, we can see that the texture does not look what we would want it to be. This happened because the orange image also has a block portion and as we have used intensity for our correspondence map, rdj's hair match perfectly with black patches in the image.
- 11) Also when we transfer a highly colourful texture to an image which does not have an eclectic intensity spectrum, we do not get the desired results as can be seen from deepika and carpet image. This is because the intensity variations in target images is less and thus only a few colours are able to map. This issue can be overcome by histogram equalisation and contrast stretching as it would lead to larger intensity variations in target image.



Challenges

```
imname = 'windows.png';
%=====
% use this for custom images (MS paints)
% a = im2double(imread(append('../inputs/custom/',imname)));

% else
[input_img,map ] = imread(append('../inputs/synthesis/',imname));
a = ind2rgb(input_img, map);
% =====algorithm starts
```

- In synthesis the image used by the paper gave greyscale results. With some modifications we could get the full RGB results.

So we had to use extra line for those images in the code: we let it be like this since anyways we are running the code once per image.

THANK YOU!