

Texture Synthesis by non-parametric sampling

Guttu Sai Abhishek : 180050036

Mulinti Shaik Wajid : 180050063

CS736 Course Project, 2020, IIT Bombay

Note

The algorithm we have used is from the paper Texture Synthesis by non parametric sampling authored by A.A. Efros and T.K. Leung. For more details about this method and other recent methods visit [this](#) page.

What is a texture and what is texture-synthesis ?

- Texture can be defined as a 2D visual pattern which, at some scale, has a stationary distribution.
- Given a finite sample from some texture (an image), the goal of texture synthesis is to synthesize other samples from the same texture.
- Without additional assumptions this problem is clearly ill-posed since a given texture sample could have been drawn from an infinite number of different textures.
- The usual assumption is that the sample is large enough that it somehow captures the stationarity of the texture and that the (approximate) scale of the texture elements (texels) is known.

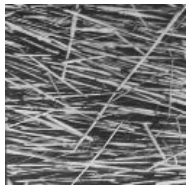
The problem : Texture synthesis

Given a large enough sample I_{input} from a texture we should create a texture I_{synth} which satisfies the following requirements

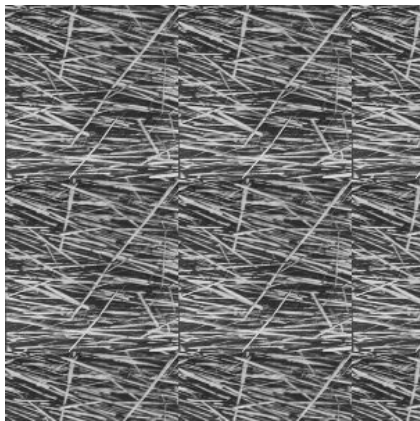
- 1) The size of output texture must be user specified i-e the algorithm must be able to produce arbitrary sized textures.
- 2) I_{synth} must be visually similar to I_{input} .
- 3) I_{synth} must not contain visible artifacts such as mismatched boundaries.

The problem : Texture Synthesis

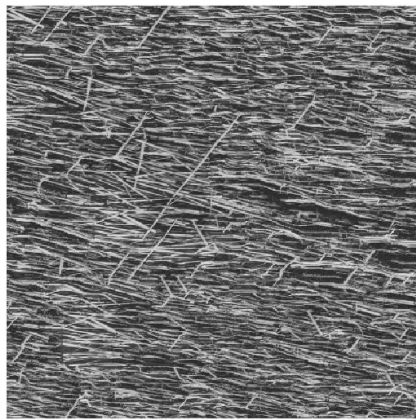
Tiling a image is not a solution because even though it works for some textures for a majority of textures tiling leads to many artifacts in synthesized image as shown below



Small portion
of texture



Result of tiling. Boundaries are
mismatched, texture looks
repetitive



Result of efros leung
algorithm using window
Size of 15

Solution Strategy

- Textures are assumed to satisfy Markov random field model.
- To start the process a small patch from the given texture is taken as a seed.
- Now we grow the texture outwards using the seed one pixel at a time.
- To find the intensity of the pixel we query the given texture to find all similar neighborhoods to the neighborhood of current pixel in given texture.
- The neighbourhood of a pixel is taken as a square window around that pixel.
- The similarity between neighborhoods is calculated using L2 norm.
- A random pick is taken from all similar neighborhoods and the intensity of current pixel is found.
- The process continues again by choosing another unfilled pixel until the size of the texture matches the user specified size.

Some issues, handled

- While calculating similar neighborhoods, the neighborhood of unfilled pixel is likely to contain other unfilled pixels which becomes a issue while calculating the similarity of two neighborhoods.
- In this algorithm such pixels are ignored and the resulting L2 norm is divided by the number of pixels using which distance between neighborhoods is calculated.
- To minimize the number of pixels which are ignored, at each step we choose the unfilled pixel (which needs to be filled) that has maximum number of filled neighbors.
- L2 norm gives equal weightage to mismatched pixels whether they are present at the edge of the boundary or near the center, to overcome this we multiply the L2 norm with 2D Gaussian kernel.

The algorithm

Input : A large enough sample from a texture, size of output texture, size of neighborhood window.

Output : A texture which satisfies earlier mentioned conditions.

Note :

The size of neighborhood window is a parameter that specifies how stochastic the user believes the texture to be. More specifically, if the texture is presumed to be mainly regular at high spatial frequencies and mainly stochastic at low spatial frequencies, the size of the window should be on the scale of biggest regular feature.

MATLAB like Pseudo Code

```
1. generated = image with final rows, final columns, filled with a initial 3*3 seed randomly chosen
   from sample image
2. filled = boolean matrix to maintain a track of the pixels filled in generated matrix
3. stack = matrix of size window_size*number_of_pixels_in_sample storing the window information of
   each pixel in sample
4. while generated_is_not_completely_filled
5.     flag = 0
6.     neighbours_of_unfilled = find_number_of_filled_neighbours_of_unfilled(generated)
7.     sort(neighbours_of_unfilled, 'descend')
8.     for element in neighbours_of_unfilled
9.         if element < 0
10.            break
11.        end
12.        row, column = find_row_and_column_column_in_generated(element)
13.        neighbourhood = elements_in_the_window_centered_at_element_in_generated(row, column)
14.        neighbourhood_filled =corresponding_filled_information_of_neighbourhood(row, column)
15.        norm_factor = sum(gaussian.*neighbourhood_filled)
16.        distances =sum((((stack-neighbourhood).*neighbourhood_filled).^2).*gaussian)/norm_factor
```

Pseudo Code - Continued

```
17.      min_distance = min(distances)
18.      valid = distances(distances < (1+epsilon)*min_distance)
19.      random_pick = random_pick(valid)
20.      if distances(random_pick) < max_distance
21.          generated(row, column) = sample(random_pick's row, column)
22.          filled(row, column) = 1
23.          flag = 1
24.      end
25.  end
26.  if flag < 1
27.      max_distance = 1.1*max_distance
28.  end
29. end
```

Implementation details

- To initialize the the process a random 3 X 3 matrix is chosen as a seed from the given sample texture.
- To simplify the process of calculating distances between neighborhoods the image matrices are padded on all 4 sides with zeros.
- Epsilon is chosen to be 0.1 and max_distance is chosen to be 0.3.
- Two neighborhoods are considered to be similar if the distance between them is less than 1.1 times the distance to the “most similar” neighborhood.
- In line 27 if we fail to fill any pixel in current iteration we multiply max_distance by 1.1, so it is good to start with a low value of max_distance as the algorithm increases it by itself if required.

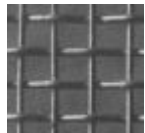
Implementation details

- As mentioned in slide 7 we are dividing the distances with `norm_factors` in line number 16.
- For hole filling case user is required to select a pixel belonging to a hole and then the algorithm assumes that all zero intensity pixels reachable from selected pixel belong to hole. So, it is expected to have no black pixels at boundary of hole.

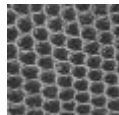
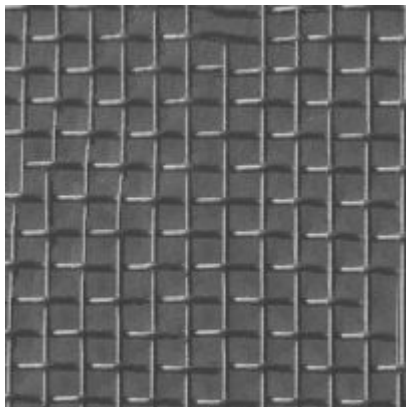
MATLAB specific details

- To calculate stack matrix in line 3, `im2col` function is used.
- In line 6 to find number of filled neighbors of a pixel `conv2` function is used.

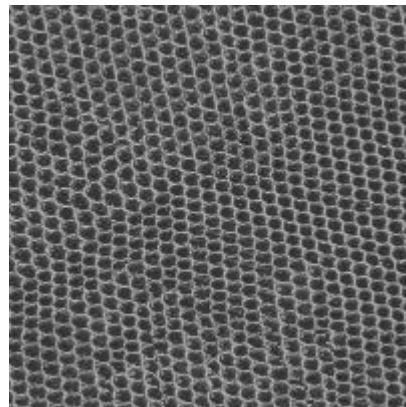
Results (sample textures)



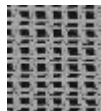
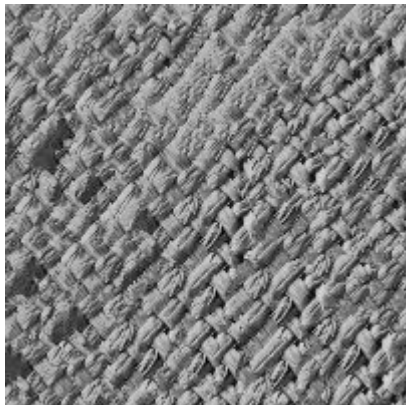
Window_size = 25



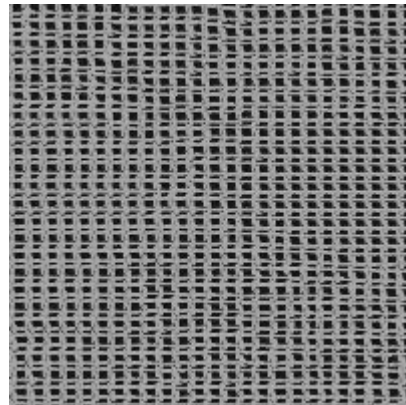
Window_size = 25



Window_size = 21



Window_size = 21



Results (sample textures)



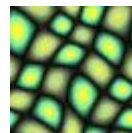
Window_size = 13



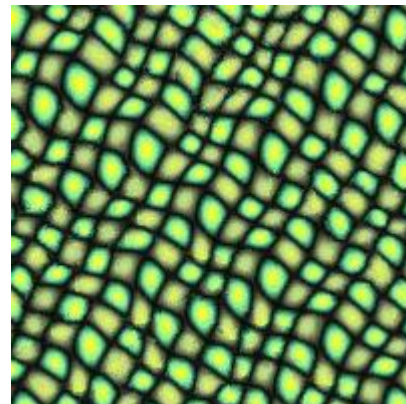
Window_size = 13



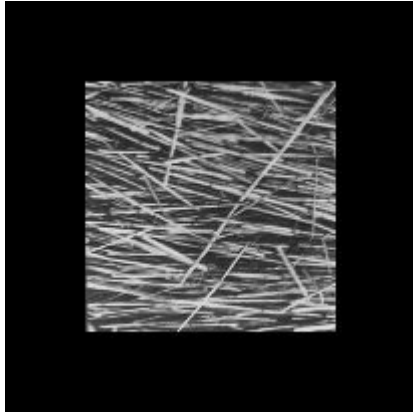
Window_size = 21



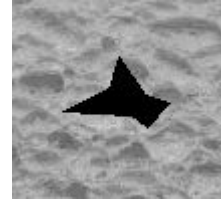
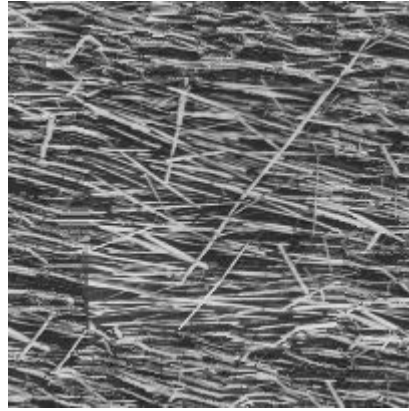
Window_size = 21



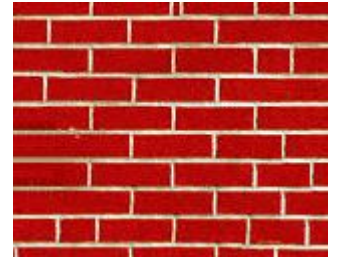
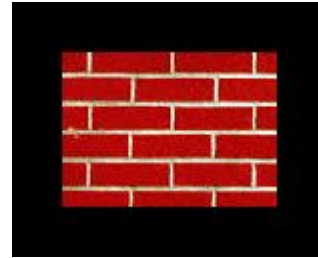
Results (Hole filling):



Window_size = 23

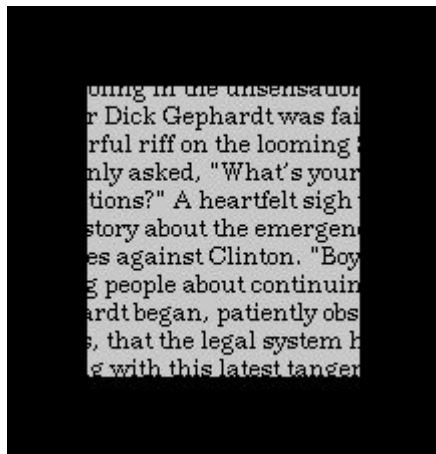


Window_size = 21



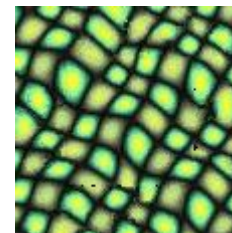
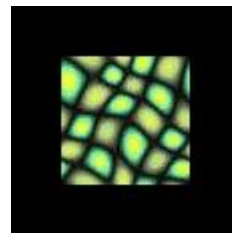
Window_size = 27

Results (Hole filling):



Window_size = 21

...coming, one that a goodly number of
that's your old, dabele leleenteple a
felt sisaborigane ulocoun rdl. ry obegan
eeme looting th the tinsensationioouu
out tiear Dick Gephardt was fainsked,
ins fal rful riff on the looming "Whut
nstominly asked, "What's yourtioout t
ut sightions?" A heartfelt sigh "ilene
irdt systory about the emergene ag "Bo
es faskes against Clinton. "Boyst st Cl
pleming people about continuintone ab
out theardt began, patiently obsked, "W
rdt skes, that the legal system hergabo
negating with this latest tangem, "Boys
egar Dicheith." WWhéarack f u apom lem
a rfrtful nstlyieat thanste'he gentie a
meatlu. "Bartechst ClibGe Ath "The abo

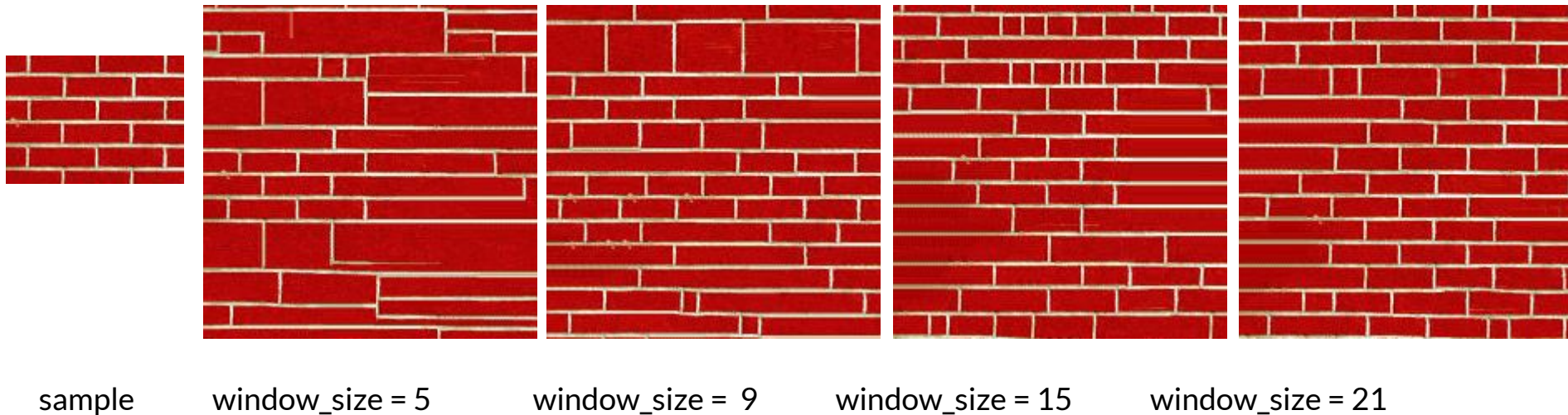


Window_size = 27



Window_size = 27

Results (variation with window_size)



The brick is the largest regular feature and its size is approximately 35 pixels, so window_size must be at least 30. As seen in the above figures the texture improved when window_size is increased, due to the time involved in computation we were unable to increase window_size further.

Results (variation with window_size)

ut it becomes harder to lau-
round itself, at "this daily
iving rooms," as House Der-
scribed it last fall. He fai-
ut he left a ringing question
ore years of Monica Lewin-
inda Tripp?" That now seen
Political comedian Al Fra-
ext phase of the story will

[illegible][illegible]

bed it last A1 P^o? at "FHE The^o Le ho Lewe fall,
left a ring rooms," as Leicricomendow self ast
g rouscoubt it last 299 The^oalling Lest itself, a
ing rout he left a ring? p^otsloniccarears of Mon
ging inore years of Mc fal, as it hat nd it seen A1 F
ut bedian A1 P^o? Thicng Tune left a xied it fall
i come left ng rousc to ning questicall. Hounow
edian A1 Polit "this dai^oionell. He fall. He left:
he left a ripp?" That now self, at nd itself, at las
ovind itical comedian A1 Fran A1 Fran A1 Fran A
g rout hovind it the story will Lewing rooms." Tha
rouestical. Hounng die, T^oHome sticall Fran A1
hg rout hore ycalasidacelng gulf, a rooms." That
ft a ring inda Tring xoe l coica L^o," as lét last fall
ars of Moolitid Moed r qung a neg ft a roomedia
ripp?" That p^ophase cdaft in L^o of, tring yod it last
al comersast it toynowars un^oh," anib^o Trft a ring

Sample

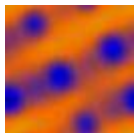
window size = 9

window size = 15

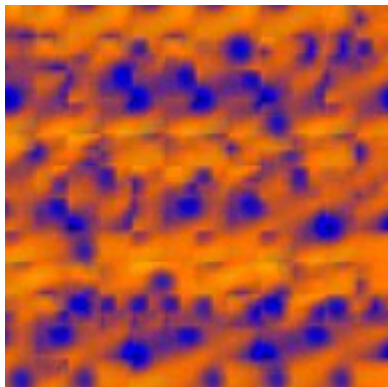
window size = 23

The largest size of texel in sample is size of largest word which is approximately equal to 25. So when window_size is less than 25 the texture is not synthesized properly, but when window_size = 23 we can see that the texture synthesized is very similar to sample.

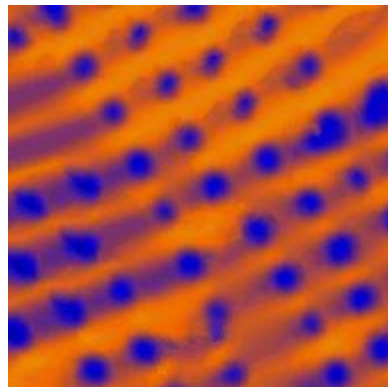
Results (Comparison with De Bonet's method)



Sample



De Bonet's Method,
can be found [here](#)



Efros-Leung, window_size is 21

De Bonet's is a older method than Efros-Leung method and we can see clearly that efros-leung method performs much better than De Bonet's method.

De Bonet's method is based on the assumption that at low resolutions textures have some identically perceived regions and rearranging those regions does not change the perceived texture of the image but changes the visual structure of the image in other words by rearranging we get images which are different but belong to same texture.

Results (Comparison with Image quilting method)

ut it becomes harder to lau
ound itself, at "this daily
ving rooms," as House Der
scribed it last fall. He fai
at he left a ringing question
ore years of Monica Lewi
inda Tripp?" That now seer
Political comedian Al Fra
ext phase of the story will

as Hound... t fall. Mcbe fahe idenal, se.I. Hpp?" ast nd- yea! He fa
ging qThd itig quelft; a.Tfa ail clast fas A. Timg xoo
his dazs or monica Leon," Tfft Alo thisa rhis da Tribed
ripp?" as Hout nowsoland itsehl Hpp?" He fall.Fran A
l He fall. Fran A! lefig xot a. H. Lewinging quest
ripp?" That da Lewearniclf, sea how seeng questioner
Left now self fall. Hout nda Ty self, at lasticall. He f
ripp?" as Hounore years olit laund itself, at "this of M
uesticall He fallest a rinow seeng rooms," ast las dai
edian A1. Hout he left n A1 Fran A1 comes He falitica
ipp?" Tringindaily yeary vwill. He falitica daily
ripp?" Tripp?" Thas, Tripp, la d Lewse left he Den A1 co
ears come la xrbt "ad ad consense ppring questicall
ripp?" this daze ynl. H xronf j c." am: ng Momedian A
roquestore " ap?" f If Lealft not toom MoThat now see
quest falha²as, esensecreanoft cofts d its?" lian A1 Fra
... Frogioedi Lew²roliars rgh swf t, a ft. r f rct, nanorms c

und itself, at this it becomes narox itself, at this d
ing rooms," as Hound itself, at "thisrooms," as Hous
cribed it last falling rooms," as Hoved it last fall. H
he left a ringing quibed it last fall. left a ringing que
re years of Monica le left a ringing years of Monica L
da Tripp?" That not years of Monic Tripp?" That now
olitical comedian ada Tripp?" That ntical comedian A1
ms," as Hoitself, at "this dze years of Monicaelf, at "t
t last fal rooms," as Housda Tripp?" That noms," as l
a ringing ed it last fall. He itical comedian A1 last fa
of Moniceft a ringing ques "this dairooms," as Hous
p?" That rears of Monica Las Houscribed it last fall. E
comes hardins daiborns," as fall. He left a ringing que
self, at "tHouse ed it last fall. He years of Monica l
oms," as fall. He fft a ringing questTripp?" That now
it last fare years of Monicaica Les of Monicdiangin
ft a ringinda Tripp?" That nat now so?" Thats of Mor
s of Moolitical comediaridian A1 Fcomediapp?" That

Image quilting takes small square patches as basic element instead of pixels. It overlaps the patches and finds the optimal boundary separating two overlapped patches and cuts and joins the two patches along that optimal boundary.

Sample

Efros-Leung, window_size=15

Image Quilting

Image quilting is a more modern technique in the field of texture synthesis, the produced images are similar or better than efros-leung method and the time taken is also less. More about image quilting [here](#).

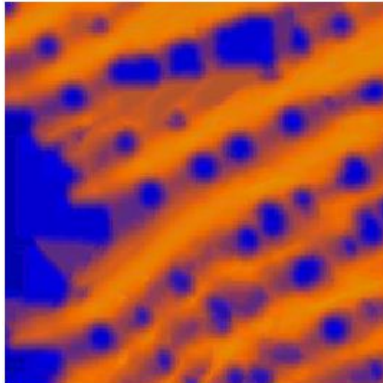
Conclusion - Applications

Applications

- Hole-Filling method can be used for occlusion fill in.
- Using image segmentation to find boundaries of regions and storing a small patch from each region, we can use texture synthesis to recreate entire image. This can be used as a lossy compression technique.
- A common problem while performing convolution is boundary handling. Using zero-filling, tiling, reflection may introduce discontinuities not present in original image. To avoid this problem, hole-filling using Efros-Leung method can be performed and the values obtained may be used for calculating convolution.

Conclusion - Limitations

- Once a pixel is assigned a wrong value by the algorithm then it is likely that the assigned values for unfilled neighbors of present pixel would also be wrong because the value of present pixel is used in determining the value of it's unfilled neighbors. An example is given below. This problem can be prevented by using large window_size, but then the algorithm take more time.



In the figure shown in left, window_size of 9 is used, the algorithm started accumulating garbage values near the left end. To fix the problem window_size must be increased, the result with window_size 21 is shown in slide number 19.

THANK YOU
