Patch Matching

Kamarsu Sriya Deepika-20161186
D. Shritishma Reddy-20161165

DIP PROJECT
EVALUATION-1

Approximate Nearest Neighbour Algorithm

- Patch-based sampling methods have become a popular tool for image and video synthesis and analysis.
- Applications include texture synthesis, image and video completion, summarization and retargeting, image recomposition and editing, image stitching and collages, new view synthesis, noise removal and more.
- Approximate Nearest Neighbour Algorithm algorithm offers substantial performance improvements over the previous state of the art (20-100x), enabling its use in interactive editing tools.

```
Function [NNF]=PatchMatch(target , source, window)
	//Defined in the paper
	radius=source_image_size/4 ; alpha=0.5 ; max_no_iters=-log(w)/log(alpha)
	//Random Initialization
	NNF->random_initialization_of_indices_from_source(leaving_padded_indices)
	Calculate L2 dists for each pixel window of source and its corresponding pixel window in target linked by
random NNF with and store in offset[ ]
	For max_iters:
		For each pixel in target image:(i,j)
			//Propagation->P
			Calculate idx=argmin(index)[offset(i,j),offset(i-1,j),offset(i,j-1)]
			if(idx==2)
				NNF(i,j,1)=NNF(i,j,1)+1;NNF(i,j,2)=NNF(i,j,2);Update offset of (i,j)
			if(idx==3)
				NNF(i,j,1)=NNF(i,j,1);NNF(i,j,2)=NNF(i,j,2)+1;Update offset of (i,j)
			//Random Search->S
			Calculate range of search according to radius and max_no_iters
			Define the boundaries of search for each pixel according to the above range
			Update NNF and offsets if we find a better L2 distance for a window corresponding to (i,j)
		end
	end
End
```

Reconstruction of image:
We move with a stride of window_size and reconstruct the image using the patch_matched from source image of the first pixel in each stride

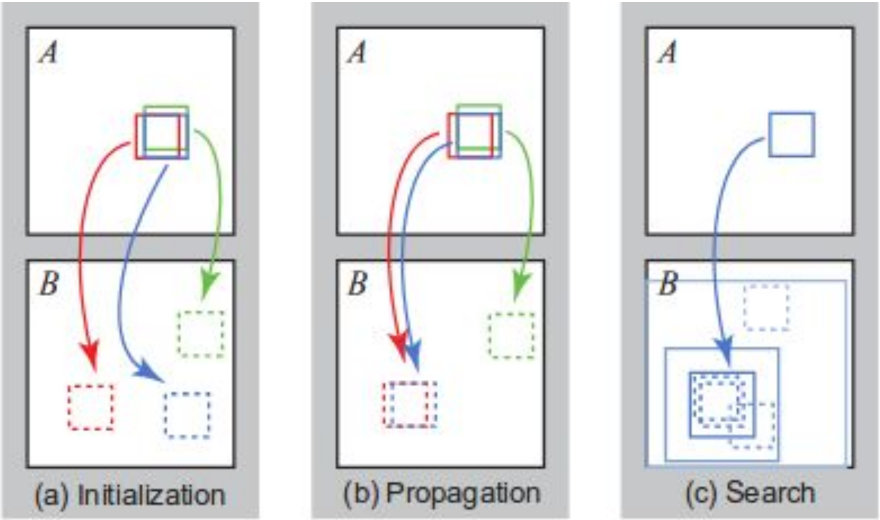


Figure 2: *Phases of the randomized nearest neighbor algorithm: (a) patches initially have random assignments; (b) the blue patch checks above/green and left/red neighbors to see if they will improve the blue mapping, propagating good matches; (c) the patch searches randomly for improvements in concentric neighborhoods.*

Halting criteria.

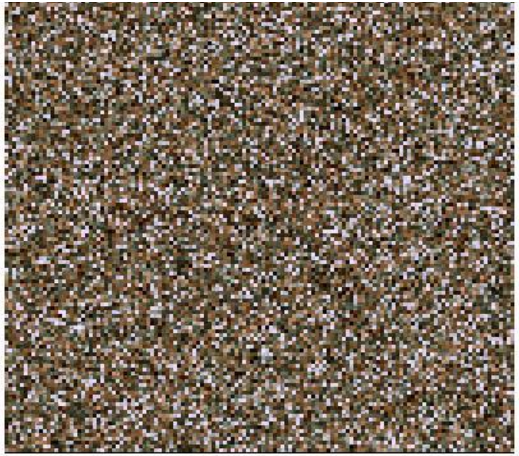
- Although different criteria for halting may be used depending on the application, in practice we have found it works well to iterate a fixed number of times. All the results shown here were computed with 4-5 iterations total, after which the NNF has almost always converged.
- The practical and theoretical proofs for convergence are provided below:

Practical Proof
Source Image

Image that has to be reconstructed from Source Image

After every iteration (window_size=3)

Random Initialised NNF



¼ iteration

After every iteration

¾ th iteration

1st iteration

After every iteration

2nd iteration

3rd iteration

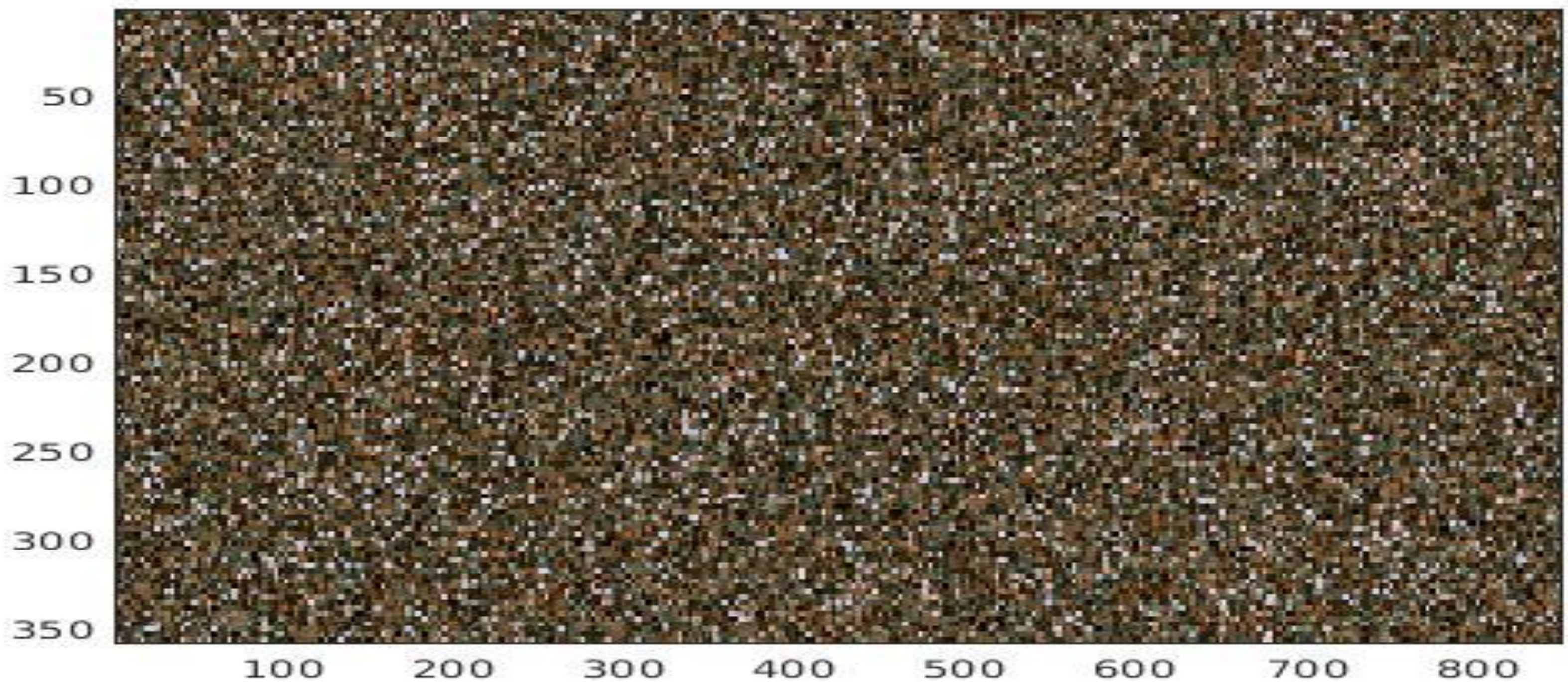
After every iteration

4th iteration

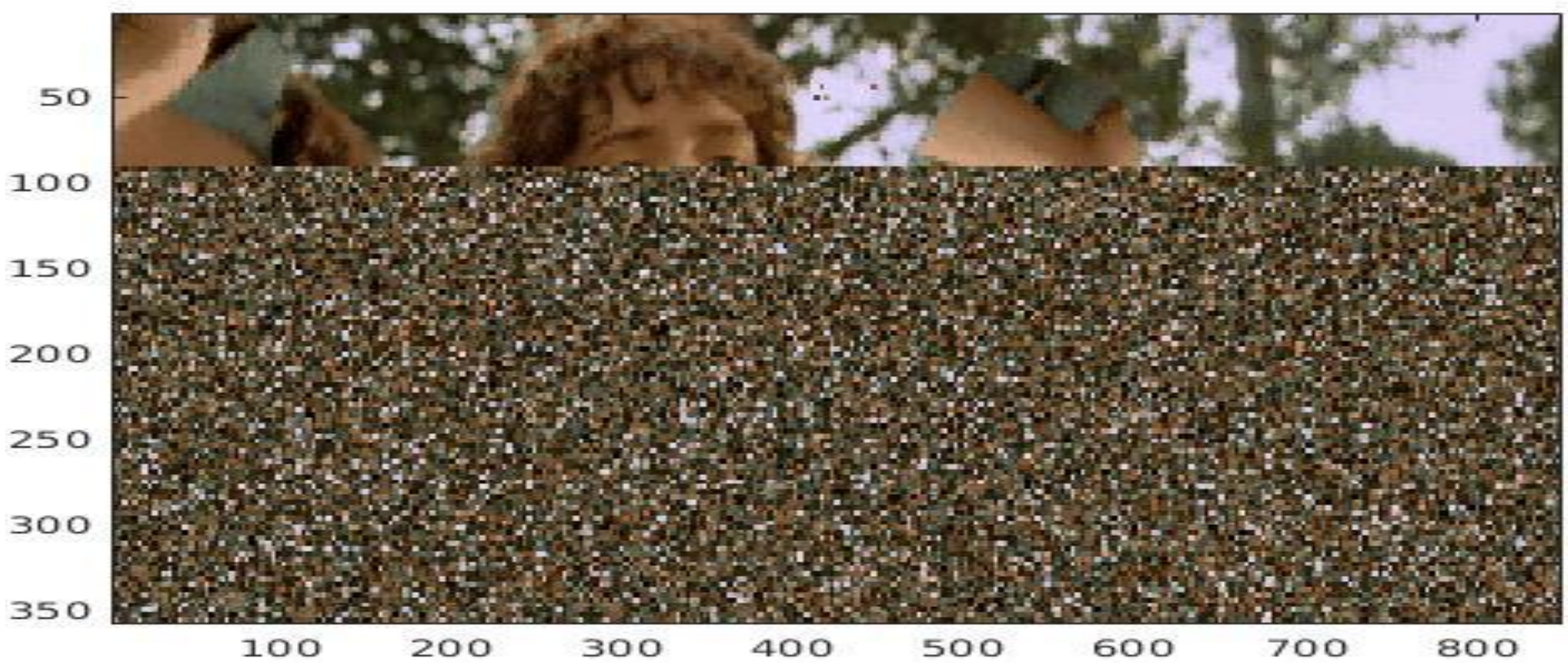
Source Image

Image that has to be reconstructed from Source Image

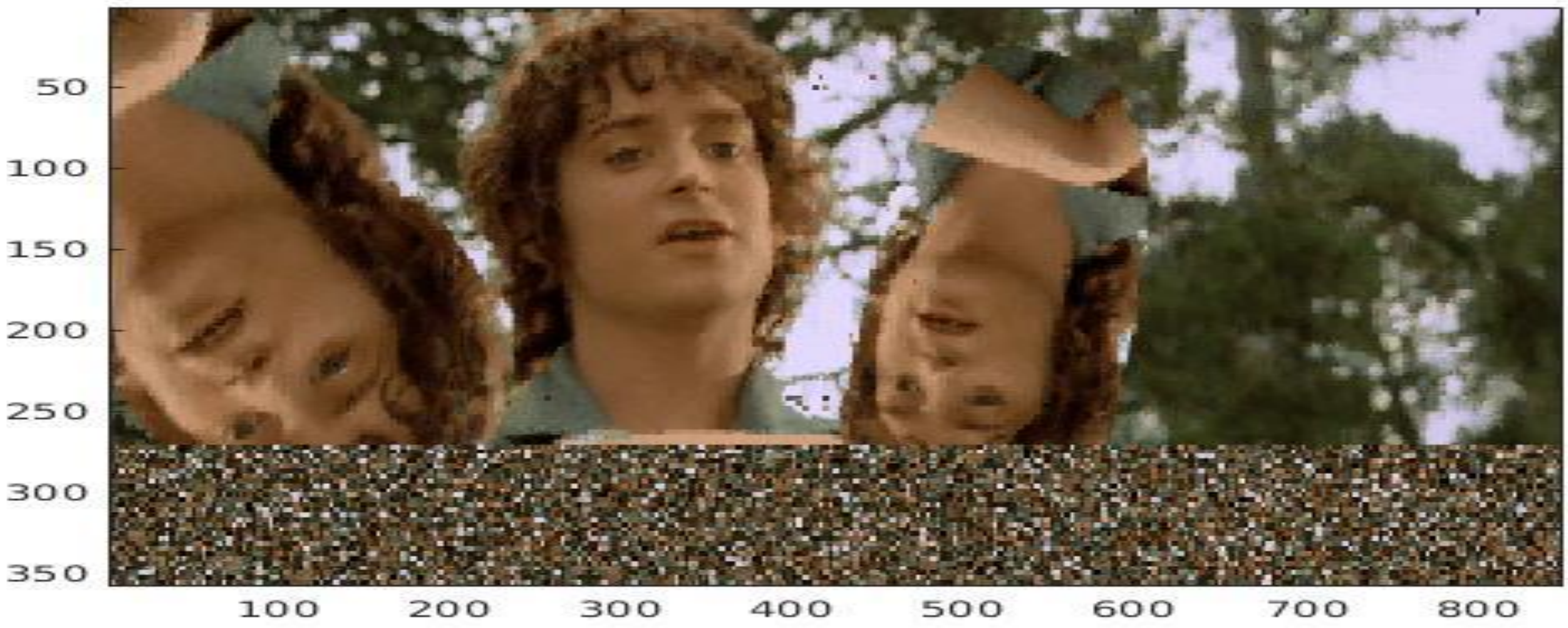
Random initialization



After ¼ iteration



After ¾ iteration



After 1st iteration

After 2nd iteration

After 3rd iteration

After 4th iteration

Theoretical proof

Step1:

- We start by analyzing the convergence to the exact nearest neighbourhood field and then extend the solution to an approximate solution
- Assume A and B have equal number of pixels M (same size)
- Probability that the random initialization for a pixel is the best offset=1/M
- Probability that the random initialization for all the pixels is not the best offset=$(1-1/M)^M$
- Probability that at least one pixel has the best offset in the random initialization=$1-(1-1/M)^M$
- Lim (p) $=1-e^{-1}$
  (M->infinity)

Convergence Theoretical Proof

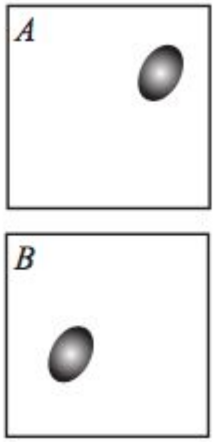
Step2:

- For an approximate solution, we can also consider a “correct” assignment to be any assignment within a small neighbourhood of size ‘C’ pixels around the offset.
- In that case $p=1-(1-C/M)^M$
- $\lim_{M\to\infty} (p) = 1-e^{-M}$
  (M->infinity)

Convergence Theoretical Proof

Step-3:

- Consider a synthetic example in which both A and B have M pixels each but we are interesed in matching a region of m pixels in A to a region of m pixels in B
- No further information is provided about M-m pixels in A and B



Convergence Theoretical Proof

- The probability that any of the m pixels lands within the neighborhood *C* of the correct offset is given by $p=1-(1-C/M)^m$
- The probability that we did not converge on iterations 0, 1, ..., *t* 1 and converge on iteration *t* is *p*(1-*p*)^*t* .
- The probabilities thus form a geometric distribution, and the expected time of convergence is E(t) = 1/*p* -1
- To simplify , let the relative feature size be gamma=m/M
- Take the limit as M->inifnity

$$\langle t \rangle = [1-(1-C/M)^{\gamma M}]^{-1} - 1$$
$$\lim_{M\to\infty} \langle t \rangle = [1-\exp(-C\gamma)]^{-1} - 1$$

- By taylor expansion for small gamma,

$$\langle t \rangle = (C\gamma)^{-1} - \frac{1}{2} = M/(Cm) - \frac{1}{2}$$

- <t> is our expected number of iterations to convergence remains a constant for large image resolutions and a small feature size m relative to image resolution

Milestones left

Illustrating the applications that use randomised NNF algorithm