CSC320 Assignment 3 Report

Terminology Used in This Report

Term	Meaning
NN	Nearest Neighbour
NNF	Nearest-Neighbour Field
NNV	Nearest-Neighbour Vector
POV	Point of View (perspective)
SSD	Sum of Squared Differences (how I'm measuring D)
D-score	The value produced by D for some NNV

Notes

1. I was unable to run my program on the Canyon image set as it took up too much RAM on both my personal computer and CDF
2. I didn't test the algorithm on significantly different images (e.g. a rock vs. a cat) as I didn't believe it would be a **fair** test – the algorithm is designed under the assumption that the source and target images are **similar**. Thus, we can't really say much about the performance of the algorithm if we give it significantly different images.
3. We note that zones of similar colors in the color-visualization of the NNF are the direct manifestation of groups of highly similar NN vectors.
4. If we reconstruct an image using pixels from an image, the reconstruction will also be about as blurry as the pixel-source image. This is because high blurriness represents a low variance between pixel values, and drawing many values from an image of low variance will likely result in another image of low variance and thus high blurriness.
5. I am only analyzing the images produced by the **third** iteration of both **propagation and random search**. My implementation was able to converge to good NN matches relatively quickly with propagation (i.e. the level of change between the second and third iterations was small compared to the change between the first and second), regardless of the image set.
6. I don't have any test results for propagation or random-search independent of each other
    a. Without random search, propagation quickly runs out of good vectors to "spread around", and therefore the best possible matches don't get propagated around.
    b. Without propagation, random search takes much longer to find good vectors – the assumption that adjacent vectors probably have the same NNVs allows for huge speedups!

Result Analysis

The inputs and results obtained from the image sets used in this report will be organized in a table at the end of this report

Deer

- **Test Case Type:** Target is a zoomed-in version of the source image
- **Aspect of Algorithm Tested:** Ability to find NN patches regardless of image size ratios
- **NNF Colorization**
    - The deer is colored with relatively similar colors, especially in the upper body. This means that most of the NNVs in upper part of the deer have similar directions and magnitudes. This makes sense as most of the target image is composed of the deer's upper body, which has "moved" to fill the bottom right-hand corner of the target image. Thus, the similar NNVs in the upper part of the deer must represent this down-right shift in pixels.
    - On the other hand, the NNV coloration outside of the deer is essentially a random collection of many similar color groups. This is because the patches of the grass behind the deer in both the target and source are very similar to each other. Thus, it doesn't really matter where an NNV of a patch in the grass field points - as long as it ends up inside the grass field in the target it will be fine. This is due to the small SSD score between any grassy-patch – it's likely to be low as the color is almost completely green and is of relatively uniform intensity. Thus, low D-scores between any of the grassy patches result in random pairs of the grassy patches being matched together
- **NNV Arrow Visualization**
    - As stated above, the NNVs originating from the grassy regions of the source image are pointing to relatively random patches in the grassy regions of the target image.
    - As stated above, the NNVs originating from the upper body of the deer in the source are pointing to the upper body of the deer in the target.

- The NNVs originating from the legs of the deer are pointing to the deer's black antlers in the target image – this is because the SSD between black patches is 0 and is therefore the best match for a black patch is another black patch
- **Source Reconstruction**
  - The foliage is a bit blurrier than in the actual source image – this is probably due to the random matching between foliage patches described above
  - The upper part of the deer is a bit blurrier – this is probably due to the best-matching patches originating from a lower-resolution (and thus blurrier!) version of the deer in the target
  - The lower part of the deer has lost some information: the shadows on this part of the deer seem to have turned into splotches, and the black legs are also blurrier. This is because there isn't any real information about the lower part of deer in the target image, thus we are forced to make guesses as to what resides in the source image, given a limited selection of possible target image patches (e.g. the target's antlers are probably being used to provide pixel values for the legs, shadows and antlers in the source image).
- Final Verdict: PatchMatch seems to tolerate images of different sizes and constructs a reasonable NNF

Jaguar

- **Test Case Type:** Target image is of source image's subject at a different stage of movement
- **Aspect of Algorithm Tested:** Ability to find NN patches regardless of subject's positioning
- **NNF Colorization**
  - The jaguar, the branches and a large part of the ground are colored with the almost same color (pale blue/pink), especially in the upper body. This makes sense as all these areas have essentially moved to the right in the target image (relative to the source image). Thus, these similar NNVs are representative of this right-shift in the source's subject(s).
  - The NNV coloration outside of the area described above is relatively random (ala the NNV coloration outside the deer in the last set). This is because these areas are essentially all brown/beige pixels in both the source and target images, thus it doesn't really matter where the NNVs point from the source to the target, as long as they point to a "dirt" pixel.
- **NNV Arrow Visualization**
  - There is a high correspondence between patches that belong to the jaguar/branches in both the target and source images. This is a manifestation of the NNF accurately tracking the rightwards movement of these subjects.
  - The NN correspondences between the "dirt' patches seems to be relatively random due to the reasoning described above.
- **Source Reconstruction**
  - The blurriness levels of the original and reconstructed source images seems to be relatively similar. This is due to the fact that the algorithm was comparing two images with roughly the same resolution/blurriness. Thus, we can expect the resolution of the pixels grabbed from the target image to be similar to the resolution of the original source image's pixels.
  - The jaguar seems to be slightly darker in the reconstructed source image. This is due to the fact that the lighting level decreased slightly between the source and target image, which means that the reconstructed source image had to be selecting jaguar pixels from a collection of slightly darker pixels. Thus, the jaguar was reconstructed to be darker than it was originally. This isn't that bad of a mistake however, as it shows that the algorithm was still able to correlate patches as long as they belonged to the same subject.
- **Final Verdict**
  - PatchMatch seems to tolerate different images of the same subject moving and constructs a reasonable NNF

Jaguar 2

- **Test Case Type:** Target image is a "scrambled" version of the source image
- **Aspect of Algorithm Tested:** Ability to find NN patches regardless of target positioning
- **NNF Colorization**
  - The image is organized into four color quadrants that take on the same shape as the "scrambling" done between the source and target images. This makes sense as this shows that there are four main NNV types: purple (points rightwards), pink (points left and down), white (points to same location), yellow (points right and up). These are essentially the directions that each quadrant got rearranged (e.g. the left upper quadrant (purple) was moved over directly to the right)
  - There is some noise on the borders of each quadrant – this is probably due to the borders being devoid of any sort of meaningful features (a lot of them are brown dirt/dark shadow pixels), which means they can be matched to almost any other bland patch, resulting in this random color noise

- **NNV Arrow Visualization**
  - The arrows in this image all seem to point to the correct quadrants as well as the correct positions. This makes sense as the NNVs were demonstrated above to be highly uniform in their distribution
- **Source Reconstruction**
  - The blurriness levels of the original and reconstructed source images seems to be relatively similar. This is due to the fact that the algorithm was comparing two images with roughly the same resolution/blurriness. Thus, we can expect the resolution of the pixels grabbed from the target image to be similar to the resolution of the original source image's pixels.
- **Final Verdict**
  - PatchMatch creates an almost perfect NNF for "scrambled" images

Jaguar 3

- **Test Case Type:** Target image is a rotated/resized version of the source image
- **Aspect of Algorithm Tested:** Ability to find NN patches regardless of image rotation
- **NNF Colorization**
  - There is a large turquoise diamond at the center of the NNF colorization. This represents the part of the source image that was cut out and rotated to form the target image. There is some gradual color transitions in this diamond, which represents the NNVs rotating to transform the diamond-like source extraction to the rectangular target.
  - Outside of this diamond, the NNF seems to be relatively random – this is because all the underlying pixels are similar brown "dirt" pixels and are therefore subject to the effects mentioned previously (where it doesn't matter where the NNVs in this area point, as long as it's within the corresponding area in the target image)
- **NNV Arrow Visualization**
  - The arrows that originate from inside the diamond seem to be pointing to correct locations
  - Arrows that originate from outside the diamond seem to be pointing to relatively random "dirt" locations
- **Source Reconstruction**
  - The reconstruction is blurrier than the original – this is because the target image is blurrier than the source image. Thus, the best-matching patches originate from an image blurrier than the source image. Therefore, any image created by drawing pixels from the target must be about as blurry as the target.
  - There was some information loss between the vines on the right third of the image – in the source the leaves are green, but they seem to have faded out to dirt pixels in the target. This is due to the fact that there are very few green pixels in the target image to match patches to. Since the algorithm decides NNVs randomly, we may not be able to match the green leaves in the source image with the small patch of green in the target. If we do not create a proper "green" matchings, it is likely we will create a reconstruction without much green in it
- **Final Verdict**
  - PatchMatch creates an almost perfect NNF for rotated images

Raptor

- **Test Case Type:** Target image is a slightly different POV of source image
- **Aspect of Algorithm Tested:** Ability to find NN patches regardless of subject orientation
- **NNF Colorization**
  - We can see that the left side of the raptor's head and its claws have similar colors. This makes sense as this represents the leftward rotation of these components between the source and target images.
  - We can see that the majority of the image is almost all 0-vectors. This makes sense because backgrounds between the two images are relatively consistent, which means that the NN patch in these areas is likely at the same location between the source and target.
  - There is more random noise near the bottom of the image, where the background is almost (or is) black. This is because there are many pixels in the image (especially in this area) to choose from that also belong to dark patches, Thus, the algorithm is likely to match random pairs of dark patches together, without regard as to their locations.
- **NNV Arrow Visualization**
  - All the arrows seem to be pointing to the right locations in the target image, provided they originate on top of the raptor.
  - Otherwise, if they originate from the background, they seem to be pointing to somewhat random locations on the background if they are from a dark location. Arrows that originate from the white scratches on the back seem to have reasonable destinations in the target image
- **Source Reconstruction**

- The reconstruction looks almost identical to the source image, with the addition of some noticeable blurriness/information loss around the raptor's left lower jaw. This is because the raptor is covering this part of the jaw up in the target, which means that the algorithm needs to infer what should be here, resulting in an imperfect reconstruction.

- **Final Verdict**
  - PatchMatch creates an almost perfect NNF for images of a slightly rotated subject

Stormtrooper

- **Test Case Type:** Target image is a fairly different POV of source image
- **Aspect of Algorithm Tested:** Ability to find NN patches regardless of subject orientation
- **NNF Colorization**
  - The background seems to be relatively random. This is probably because the pixels belonging to the background are relatively similar, so random background patches can be paired up without much D-value gain.
  - Interestingly, the Stormtrooper seems to be colored according to the different colors of his suit – e.g. where there are lighting reflections the Stormtrooper is green, where the Stormtrooper has black, the colors are redder. However, his coloring distribution is not nearly as uniform as those described in previous image sets.
- **NNV Arrow Visualization**
  - This is probably the most interesting result out of any of my images thus far. This is a clear demonstration of PatchMatch ignoring the locations of the source and target patches that it pairs up as NNs, and only focusing on color values. The following are examples of "weird" patch pairings
    - The source Stormtrooper's left eye is matched to the gun of the target Stormtrooper
    - The source Stormtrooper's right eye is paired up with the hand of the target Stormtrooper
    - The left arm of the source storm trooper is paired up with the temple of the target Stormtrooper
- **Source Reconstruction**
  - The reconstruction looks like a blurrier version of the original. What is interesting about this reconstruction is that it was formed from patches that don't even have the same relative locations (e.g. an eye was reconstructed using gun pixels). There is some information loss around the areas that should be reflecting the lighting in the original. This is because the Stormtrooper in the target image is rotated in such a way that not much light reflects off of him. Therefore, there are few proper "reflective" pixels for the reconstruction to choose from, leading to information loss around these pixels.
- **Final Verdict**
  - PatchMatch creates a good NNF for images of fairly rotated subjects, but this NNF is not very reasonable – unlike the other images where movement of specific image elements could be determined by looking at the NNF colorization, this NNF does not seem to track any sort of movement.

Deep Dream Doge x 2

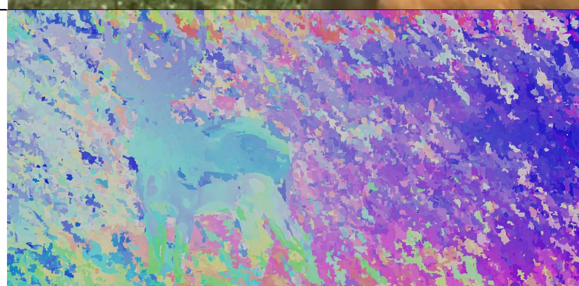
- **Test Case Type:** Target and source image have multiple similar patches.
- **Aspect of Algorithm Tested:** Ability to find NN patches in the event of multiple similar patches
- **NNF Colorization**
  - This seems to be fairly random, however some ring-shapes in the source image seem have similar vectors, which suggests they point to similarly shaped areas in the target
- **NNV Arrow Visualization**
  - Like the NNF colorization, this seems to be completely random, but it is not completely unreasonable. Many of the vectors that originate on a "Deep-Dream animal" or eye in the source image seem to finish at another "Deep-Dream animal" or eye in the target image, albeit at different positions. Assuming these animals/eyes are similar, their SSD would be fairly low, resulting in such a pairing being kept through many iterations of the algorithm
- **Source Reconstruction**
  - This is probably the blurriest reconstruction out of all my image sets thus far. A lot of fine detail has been lost in this reconstruction, probably because the NNVs pointed in the "right" direction, but weren't very specific as to where they should point. The reconstruction does reflect the "rings" present in the NNF
- **Final Verdict**
  - When fed two images that have multiple patches with highly similar features (in this case eyes and weird fish-bird-dog heads), PatchMatch essentially gets "stuck" on these features.
  - Section 3.3 of the paper mentions that PatchMatch gets "stuck" when confronted with this type situation, however it notes that the chances of such an instance occurring are very low.

Conclusions about PatchMatch:

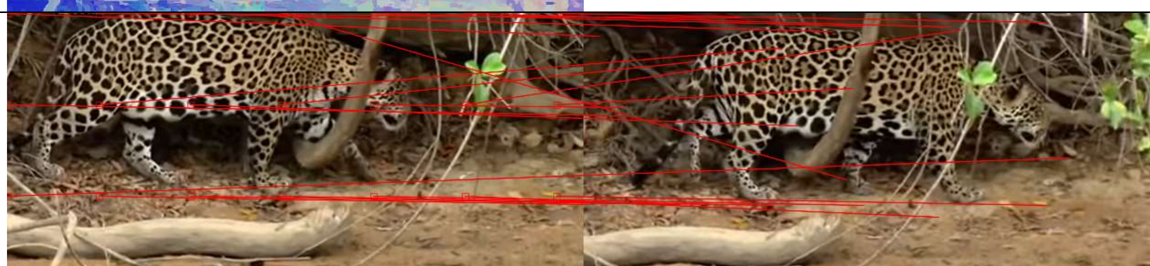
- Handles trivial image transformations (scaling, rotations, scrambling etc.) well
- Tolerant of slight subject orientation and position changes
- Doesn’t perform so well when confronted with a significantly rotated object
- Performs poorly when confronted with images that have highly similar features

<u>**Image Sets**</u>

Deer

Source	
Target	
NNF Colorization	
NNV Arrow Visualization	
Source Reconstruction	

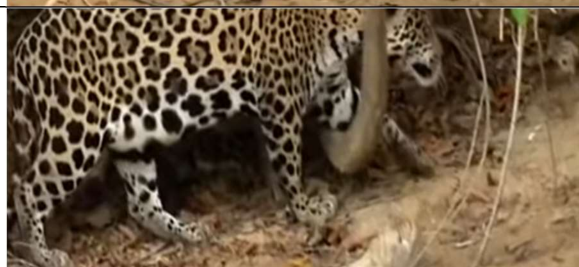
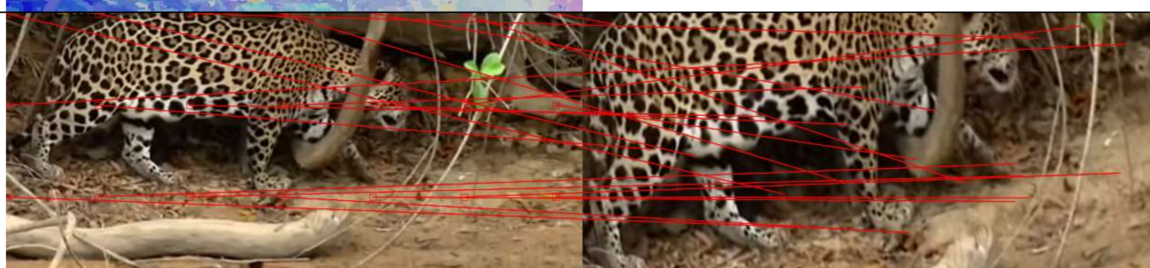
Jaguar

Source	
Target	
NNF Colorization	
NNV Arrow Visualization	
Source Reconstruction	

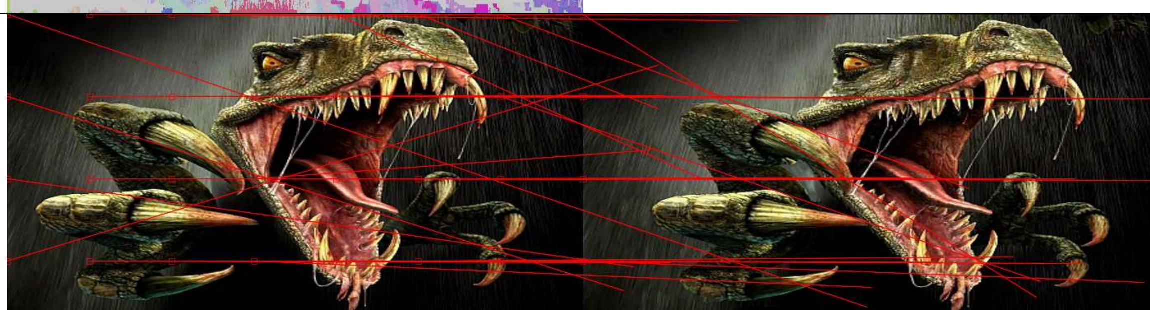
Jaguar 2

Source	
Target	
NNF Colorization	
NNV Arrow Visualization	
Source Reconstruction	

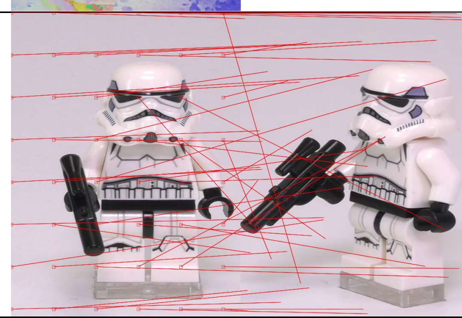
Jaguar 3

Source	
Target	
NNF Colorization	
NNV Arrow Visualization	
Source Reconstruction	

Raptor

Source	
Target	
NNF Colorization	
NNV Arrow Visualization	
Source Reconstruction	

Stormtrooper

Source	
Target	
NNF Colorization	
NNV Arrow Visualization	
Source Reconstruction	

Doge Deep Dream x 2

Source	
Target	
NNF Colorization	
NNV Arrow Visualization	
Source Reconstruction