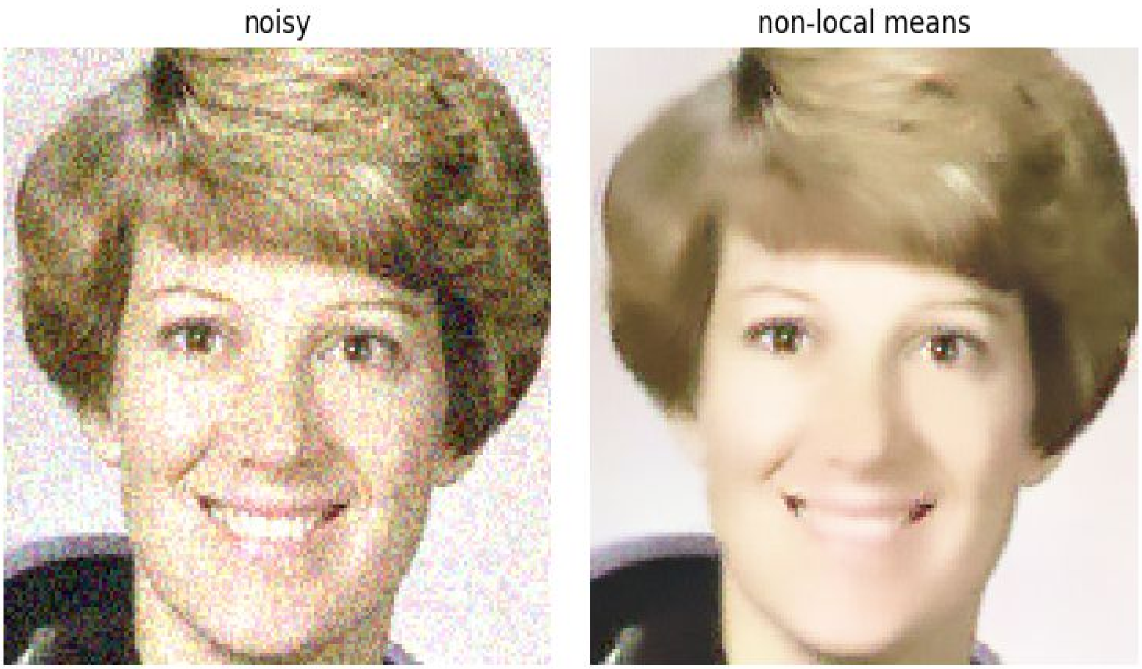
Digital Image Processing

Different Modifications of Non-Local Means Algorithm for Image Denoising.

Prateek Sahu-17IC0130



Introduction

Many natural or texture images contain structures that appear several times in the image. One of the denoising filters that successfully take advantage of such repetitive regions is Non-Local (NL) means. Unfortunately, the block matching of NL means using Euclidean distance cannot handle rotation or mirroring. In this report, I analyse moment invariant similarity measure using Hu moments approach for a rotationally invariant similarity measure that will be used as an alternative to, respectively a modification of the well-known block matching algorithm in nonlocal means denoising.Also I have analysed a fast implementation of NL means. I have implemented 3 NL mean algorithms :

1. NL means with euclidean distance with Gaussian kernel as a similarity measure.
2. NL means with Hu moment invariant as a similarity measure.
3. Fast Non-Local means using separable neighbourhood filtering.

I have taken PSNR as a metric for image comparison. In later sections I have compared all 3 algorithms in different noises etc. and gave a final conclusion on these algorithms.

Code:

All code and image used are given in following link:

https://github.com/Prateek1337/DIP_assignment_NLM

1. NL-means with euclidean distance with Gaussian kernel:

This is a standard implementation of NL means using Euclidean distance as similarity measure with Gaussian kernel. The Gaussian weighting in the distance measuring excludes the high influence of noisy pixels that are farther away from the centre pixel of the patch. Also, it helps to preserve edge by assigning a higher weight to the centre pixel.

Below are equations for the algorithms

$$u(x) = \frac{1}{C(x)} \int_{\Omega} e^{-\frac{(G_a * |f(x+\cdot)-f(y+\cdot)|^2)(0)}{\lambda^2}} f(y)dy, \tag{1}$$

where $G_a(x) = \exp\left(\frac{-|x|^2}{2a^2}\right)$ is a Gaussian with standard deviation a, λ a smoothing parameter and

$$C(x) = \int_{\Omega} e^{-\frac{(G_a * |f(x+\cdot)-f(y+\cdot)|^2)(0)}{\lambda^2}} dy$$

$$\begin{aligned} dist_f(x,y) &= \left(G_a * |f(x+\cdot) - f(y+\cdot)|^2\right)(0) \\ &= \int_{\mathbb{R}^2} G_a(t) \cdot |f(x+t) - f(y+t)|^2 dt, \end{aligned}$$

$$w(x,y) := \exp\left(-\frac{dist_f(x,y)}{\lambda^2}\right).$$

Complexity Analysis

The complexity of the algorithm is **O(|P|* N*|SW|)** where is: |P| is a number of pixels in similarity window, N is a number of pixels in the image and |SW| is a number of pixels in the search window. If size if the image is mxn , search widow is txt and similarity window is fxf the complexity will be O(m*n*t*t*f*f);

2. NL-means with Hu moment invariant.

Moment invariants are a classic statistical tool for object or pattern recognition. They have been introduced by Hu using the theory of algebraic invariants. Hu has derived his famous seven-moment invariants that are **invariant under translation, rotation and scaling**. Six of these moments are also invariant under mirroring, while the seventh moment switches its sign under mirroring,i.e. its absolute value is also invariant.

Following are the formulas for 7 hu moments:

$$h_0 = \eta_{20} + \eta_{02}$$
$$h_1 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$
$$h_2 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$
$$h_3 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$h_4 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$h_5 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})]$$
$$h_6 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

Where is $\eta_{p,q}$ is a central moment given by :

$$v_{p,q} = \int_{\mathbb{R}} \int_{\mathbb{R}} (x - x_c)^p (y - y_c)^q f(x,y) dx\, dy,$$

Xc and Yc are centroids of patch given by

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \frac{1}{m_{0,0}} \begin{pmatrix} m_{1,0} \\ m_{0,1} \end{pmatrix},$$

With the moments

$$m_{k,l} := \int_{\mathbb{R}} \int_{\mathbb{R}} x^k y^l f(x,y) dx\, dy.$$

Here f(x,y) is our image function at x,y. Let M ={M1,M2… Mk} be a set of moments that are invariant under mirroring as well as rotation. We denote the patches inside the image f with the centres x and y by P(x) and P(y), respectively. The moment distance measure is then defined via

$$dist_{\mathcal{M}f}(x,y) := K^{-1} \cdot \sum_{i=1}^{K} [M_i(P(x)) - M_i(P(y))]^2,$$

The corresponding filter is then given by

$$u(x) = \frac{1}{D(x)} \int_{\Omega} e^{-\frac{dist_{\mathcal{M}f}(x,y)}{\lambda^2}} f(y)dy,$$

with

$$D(x) = \int_{\Omega} e^{-\frac{dist_{\mathcal{M}f}(x,y)}{\lambda^2}} dy.$$

Note that we use the whole image domain Ω and do not replace it by a smaller search window.

The normalisation of the moment invariants:

The first problem that arises with this definition is the different magnitudes of the invariants. Because of that, it is very difficult to find a good filter parameter h, and the moments with larger magnitudes will dominate within the similarity measure. If we consider each moment of the set M is a random variable, we can use a normalisation such that the mean of the corresponding normalised random variable is 0 and the standard deviation is equal to 1 i.e we do a z-score normalization.

$$\widetilde{M}_i(P(x)) := \frac{M_i(P(x)) - \delta_i}{\sigma_i}$$

Here δi is mean and σi is the standard deviation of Mi.

3. Fast Non-Local means using separable neighbourhood filtering.

.It builds on the separable property of neighbourhood filtering to offer a fast parallel and vectorized implementation in contemporary shared memory computer architectures while reducing the theoretical computational complexity of the original filter. In practice, this approach is much faster than a serial, non–vectorized implementation and it scales linearly with image size.

The weight w(s, t) measures the similarity between two square patches centred, respectively, at sites s and t, and it is defined as follows:

$$w(s,t) = g_h\left(\sum_{\delta\in\Delta} G_\sigma(\delta)\left(v(s+\delta)-v(t+\delta)\right)^2\right) \quad , \qquad (2)$$

Our fundamental contribution consists in a method for computing very efficiently the weights.

$$w(s,t) = g_h\left(S_{d_x}(s+P) - S_{d_x}(s-P)\right) \qquad (4)$$

Where Sd_x is given by

$$S_{d_x}(p) = \sum_{k=0}^{p}\left(v(k)-v(k+d_x)\right)^2, \;\; p\in\Omega. \qquad (3)$$

The final algorithm is given by :

Algorithm 1 – Fast Nonlocal Mean – 1D

Input: v, K, P, h
Output: u
Temporary variables: images S_{d_x}, Z, M
Initialize u, M and Z to 0.
for all $d_x \in [\![-K, K]\!]$ **do**
  Compute S_{d_x} using Eq. (3)
  for all $s \in [\![0, n-1]\!]$ **do**
    compute weights w using Eq.(2) and Eq.(4)
    $u(s) \leftarrow u(s) + w \cdot u(s + d_x)$
    $M(s) = \max(M(s), w)$
    $Z(s) \leftarrow Z(s) + w$
  end for
end for
for all $s \in [\![0, n-1]\!]$ **do**
  $u(s) \leftarrow u(s) + M(s) \cdot v(s)$
  $u(s) \leftarrow \frac{u(s)}{Z(s)+M(s)}$
end for
return u

Complexity Analysis:

Compared to the original NL-means approach our algorithm requires three additional images to store partial results but it has a much better time complexity of $O(|\Omega|K^d 2^d)$ when compared to $O(|\Omega|K^d P^d)$ -- an improvement of order $O(P^d)$.

Experimental Results

Cameraman image:

Img 1.

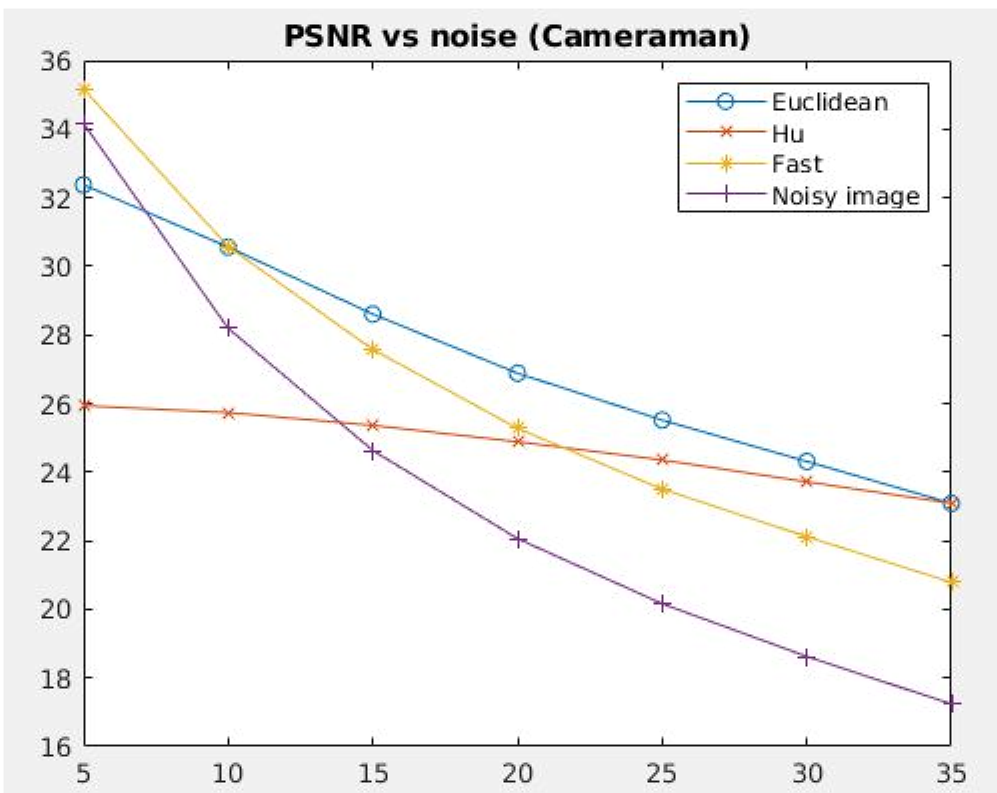


Fig-1

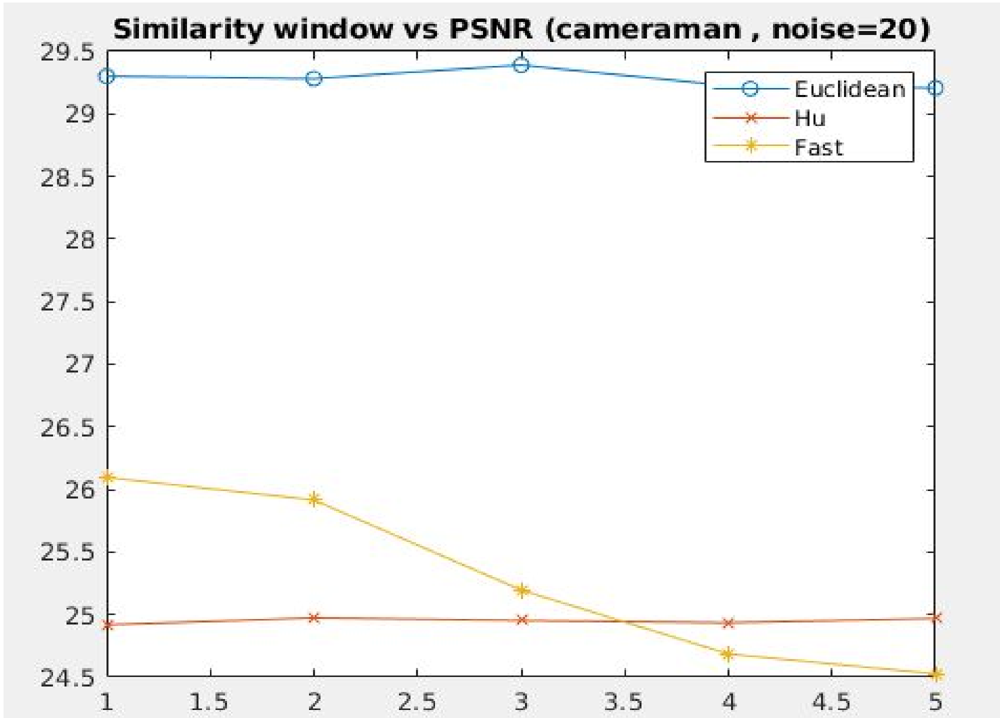

Similarity window vs PSNR (cameraman , noise=20)
Euclidean
Hu
Fast
29.5
29
28.5
28
27.5
27
26.5
26
25.5
25
24.5
1
1.5
2
2.5
3
3.5
4
4.5
5

Fig-2.

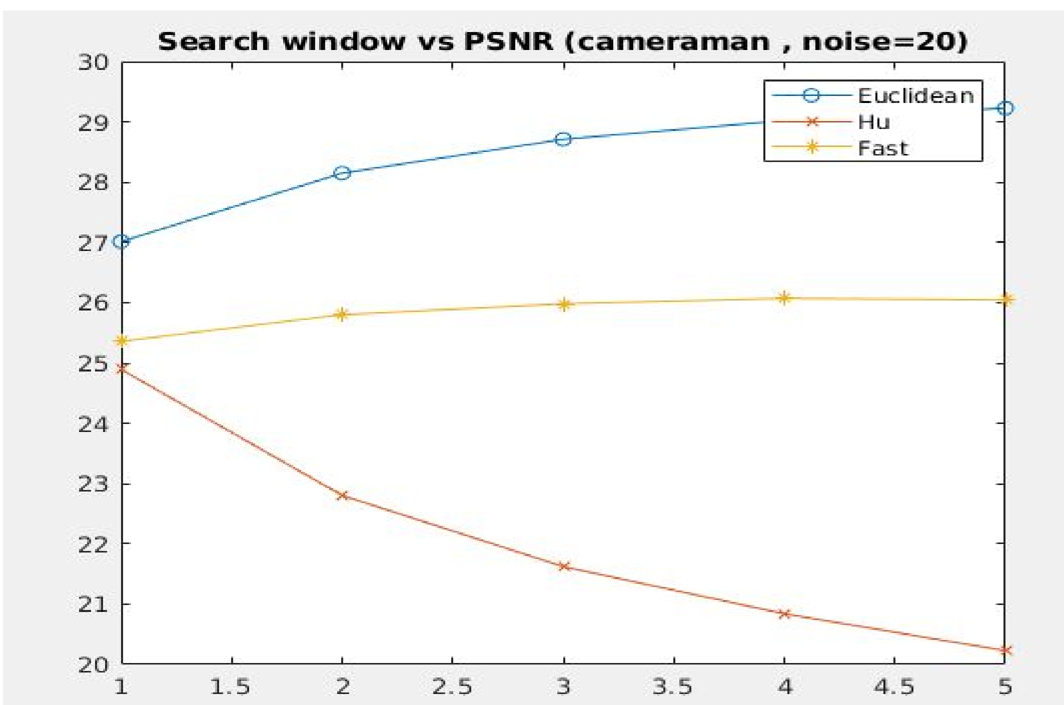

Search window vs PSNR (cameraman , noise=20)
Euclidean
Hu
Fast
30
29
28
27
26
25
24
23
22
21
20
1
1.5
2
2.5
3
3.5
4
4.5
5

Fig-3.

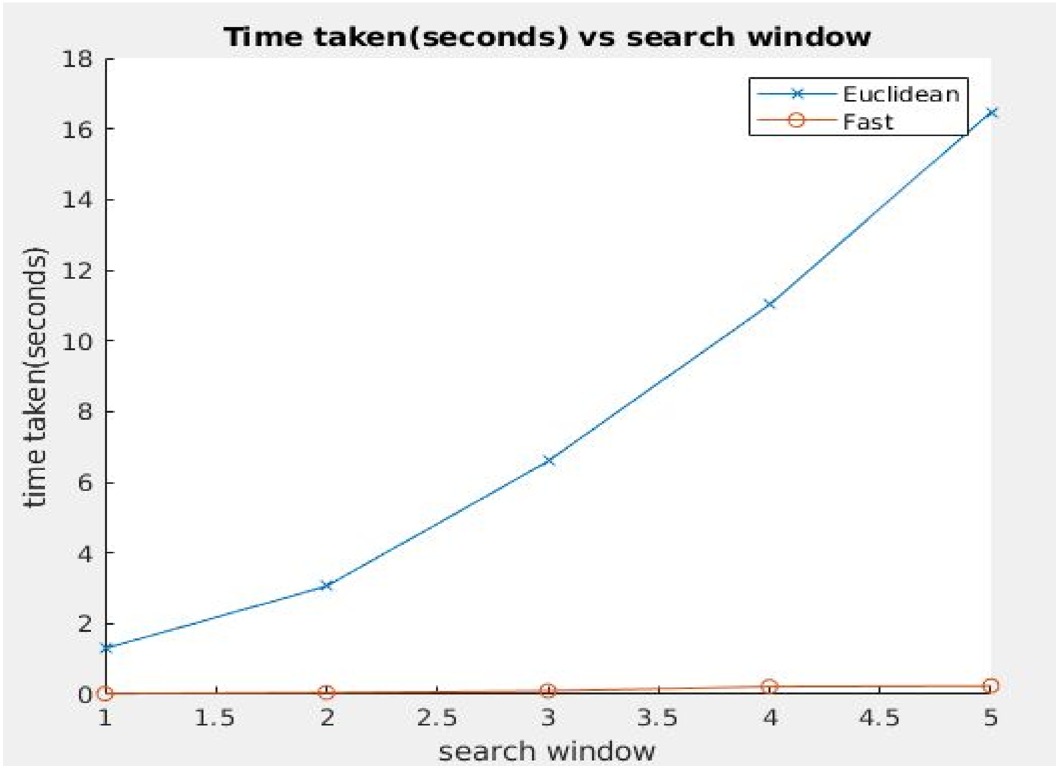
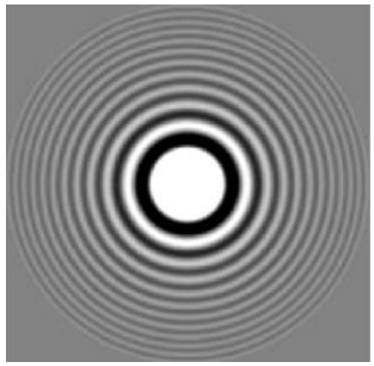

Time taken(seconds) vs search window
Euclidean
Fast
time taken(seconds)
search window
18
16
14
12
10
8
6
4
2
0
1
1.5
2
2.5
3
3.5
4
4.5
5

Fig-4

Ring image:



Img-2

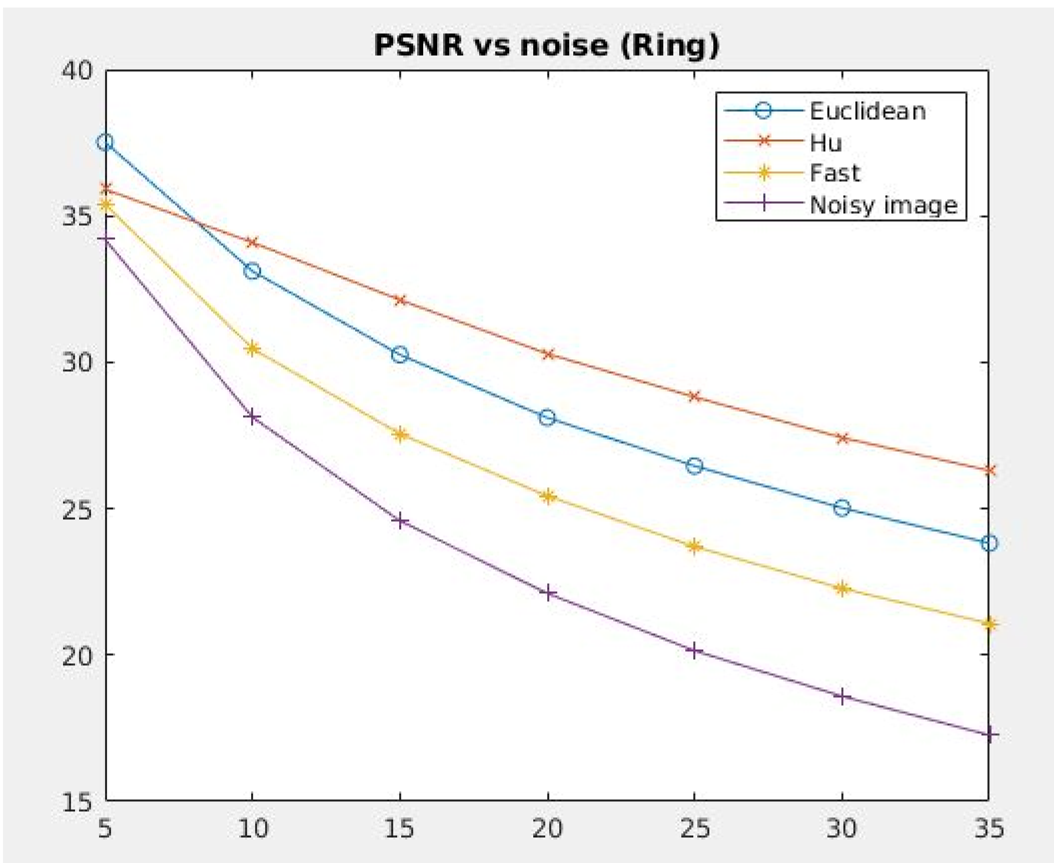


Fig-5

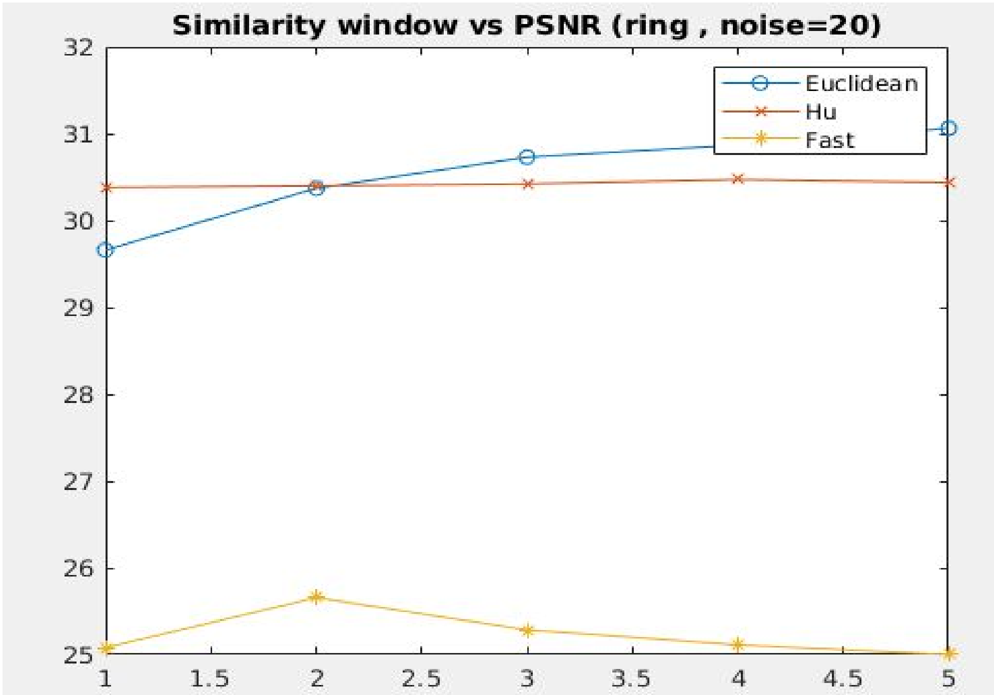

Similarity window vs PSNR (ring , noise=20)
Euclidean
Hu
Fast
32
31
30
29
28
27
26
25
1
1.5
2
2.5
3
3.5
4
4.5
5

Fig-6.

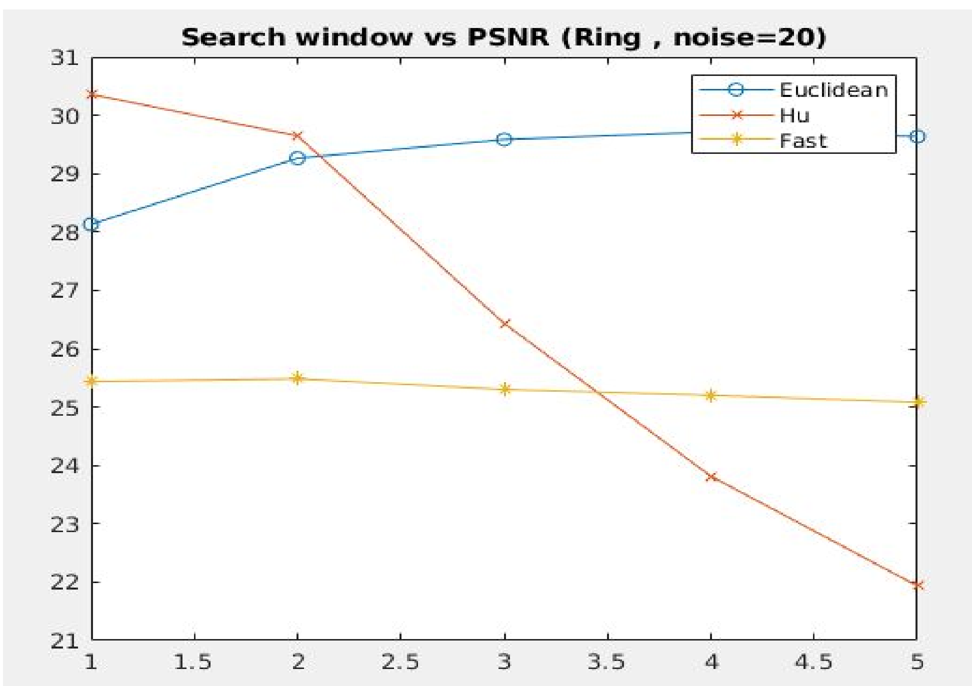

Search window vs PSNR (Ring , noise=20)
Euclidean
Hu
Fast
31
30
29
28
27
26
25
24
23
22
21
1
1.5
2
2.5
3
3.5
4
4.5
5

Fig-7

Analysis

1. Comparing fig-5 and fig-1 we can say that **Hu similarity metric performs far better in ring image** than cameraman image due to more rotational similarity in patches in ring image than cameraman image. We can say that rotational invariant metric will perform better in symetric images.
2. Also Comparing fig-5 and fig-1 we can say that **effect of more noise is more visible in other guassian and fast nlm than Hu** because Hu is averaged many rotationally invariant patches for same pair pixels.
3. fig-7 and fig-3 shows that **Hu method is more susceptible to increasing in searching window** because it does more smoothing/averaging than other methods. Whereas PSNR of euclidean and fast NLM will peak at much higher search window than Hu.
4. Also from fig-6 and fing-2 we can say that **euclidean and fast NLM are more sensitive towards change in similarity window than Hu**. This is because with increase in similarity window chances of finding similarity pixels decrase more for asymmetric patches than symmetric ones .
5. From fig-4 it is clear that **fast nlm is much faster than using euclidean measure** Also its running is less affected by increase in search window whereas running time of euclidean nlm increases with increase in search window size.
6. **Euclidean nlm perform much better than fast NLM in both the images** and Hu perform better than both euclidean and fast NLM in ring images.

Conclusion:

We can conclude that fast NLM is much faster than euclidean nlm but also is less efficient in denoising that it counter parts. Also rotational invariant NLM using Hu moment as similarity metric performs better for simiteric images and asymmetric. **If time is not big factor then we can select euclidean NLM for non symmetric images and rotaionally invariant NLM for symmetric ones. If time is very crucial then we should go with fast NLM.**

References:

1. ftp.math.ucla.edu/pub/camreport/cam08-01.pdf
2. https://www.ias.ac.in/article/fulltext/sadh/030/06/0765-0775
3. https://ieeexplore.ieee.org/abstract/document/7351074
4. https://www.ipol.im/pub/art/2011/bcm_nlm/article.pdf
5. https://in.mathworks.com/help/matlab/