19ECE457-WAVELETS AND APPLICATIONS

Batch ID: **01**

SPECKLE REMOVAL

TEAM MEMBERS:

CB.EN.U4ECE21127 - MEKA ANVITHA

CB.EN.U4ECE21139 - PUDU NIRUPAMA

Objective:

To remove speckle noise from images using wavelet transform by separating noise from image details through multi-resolution analysis. This method preserves edges and textures while enhancing image quality for applications like medical imaging and remote sensing.

What is denoising?

- The process of removing noise from an image or signal.
- The goal of denoising is to restore the original image or signal as closely as possible.

What is Speckle?

Speckle is a type of noise that appears as granular interference or a random pattern of bright and dark spots in an image. It is primarily associated with imaging systems that use coherent waves

Wavelet Shrinkage:

› Wavelet shrinkage is a method of noise reduction that uses wavelet transforms to separate the signal and noise in the frequency domain.

i) Statistical Modeling:

› Statistical modeling in wavelet shrinkage involves understanding and modeling the nature of the coefficients (signal and noise) in the wavelet domain.

ii) Modelling Components:

- Noise Estimation
- Threshold selection

- **Shrinkage functions:**
  i) Hard Shrinkage Function
  ii) Soft Shrinkage Function
  iii) Garrote Shrinkage Function
  iv) Firm Shrinkage Function

- **Shrinkage Rules**:
  i) Universal Threshold Rule
  ii) Minimax Thresholding
  iii) BayesShrink

Process:

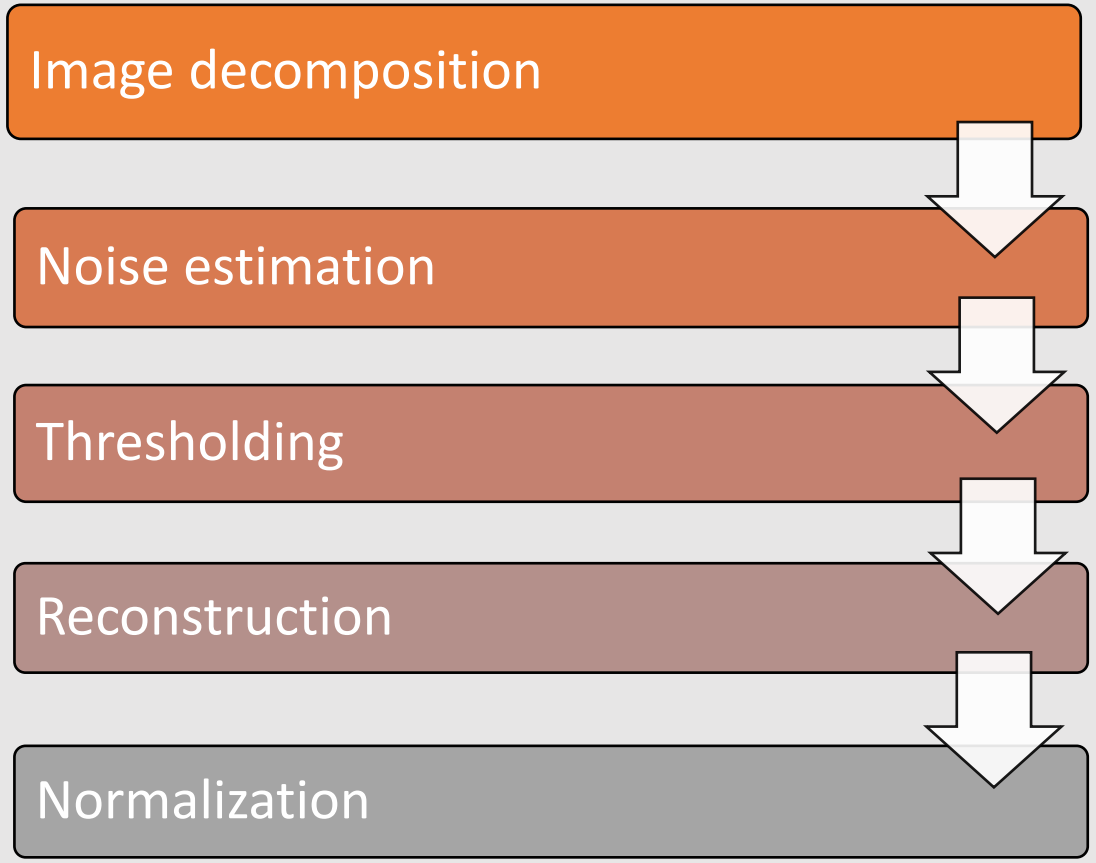
Image decomposition
Noise estimation
Thresholding
Reconstruction
Normalization

• Techniques Used:

- **Spatial Domain**:
  1. **Median Filtering**-non linear filter that replaces each pixel with median value of its surrounding pixels
  2. **Mean Filtering**-averages the pixel values
  3. **Adaptive filters**-adjust filter weights
     (a)Lee filter
     (b)Frost filter
- **Frequency Domain** :
  1. **Fourier Transform Filtering**-Transforms the image into the frequency domain using the Fourier Transform.
  2. **Wavlet Transform(Multiresolution Analysis)-**Uses wavelet transforms to decompose the image into multiple resolution levels.
- **Non-Local Means(NLM)Filtering**
- **Total Variation Denoising(TV**)
- **Deep learning-Based**

Code:

```
import pywt
import numpy as np
import cv2
import matplotlib.pyplot as plt

# Wavelet-based denoising function
def wavelet_denoise(image, wavelet='db1', level=2):
    # Perform multi-level wavelet decomposition
    coeffs = pywt.wavedec2(image, wavelet, level=level)

    # Thresholding
    threshold = np.sqrt(2 * np.log(image.size)) * np.median(np.abs(coeffs[-1][-1])) / 0.6745
    def soft_threshold(coeff):
        return pywt.threshold(coeff, threshold, mode='soft')

    # Apply thresholding to each coefficient
    denoised_coeffs = [coeffs[0]] + [
        tuple(soft_threshold(c) for c in details) for details in coeffs[1:]
    ]

    # Reconstruct the denoised image
    denoised_image = pywt.waverec2(denoised_coeffs, wavelet)
    return np.clip(denoised_image, 0, 1)

# Load the speckled image as input (normalized to range [0, 1])
image_path = r"C:\Users\NIRUPAMA\OneDrive - Amrita Vishwa Vidyapeetham\Documents\8th sem\wavlets and applications\speckled c.png"
speckled_image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
```

```
if speckled_image is None:
    raise FileNotFoundError(f"Image not found at the path: {image_path}")
speckled_image = speckled_image / 255.0  # Normalize the image

# Perform wavelet denoising
denoised_image = wavelet_denoise(speckled_image, wavelet='db1', level=2)

# Display results
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title("Speckled Image (Input)")
plt.imshow(speckled_image, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title("Denoised Image (Wavelets)")
plt.imshow(denoised_image, cmap='gray')
plt.axis('off')

plt.tight_layout()
plt.show()
```

Output:

THANK YOU