| **Name** | Adwait Purao |
| --- | --- |
| **UID no.** | 2021300101 |

| **Experiment 9** | |
| --- | --- |
| **AIM :** | Image Processing using Morphological Operation |
| **OBJECTIVE:** | 1. Develop an enhanced infrared (I-R) and visible (V-I) image fusion framework using morphological operations based unsharp masking for image enhancement.  2. Implement curvelet transform to decompose the source images into detailed and approximation coefficients.  3. Integrate approximation coefficients using the PCA fusion rule and combine detailed coefficients using the max fusion rule.  4. Evaluate the proposed fusion framework against existing methods, assessing both visual quality and metrics values to demonstrate its superiority in image fusion performance. |
| **INTRODUCTION:** | The fusion of infrared (I-R) and visible (V-I) images is pivotal in surveillance and remote sensing, enhancing visibility and situational awareness. This paper introduces an advanced fusion framework incorporating a novel enhancement method based on morphological operations and unsharp masking. By employing curvelet transform for coefficient generation and fusion rules like PCA and max fusion, the proposed framework achieves superior image quality and outperforms existing methods. This experiment aims to validate the efficacy of the proposed framework in enhancing image fusion outcomes. |
| **BLOCK**  **DIAGRAM:** |  |
| **IMPLEMENTATION:** | import numpy as np  import cv2  from scipy.ndimage import gaussian\_filter  from scipy.ndimage import grey\_opening, grey\_closing  from pywt import dwt2, idwt2  from sklearn.decomposition import PCA  def morph\_unsharp\_mask(img):  """  Applies morphological operations based unsharp masking to enhance the image.  """  # Gaussian blurring  blurred = gaussian\_filter(img, sigma=1)    # Morphological opening  opened = grey\_opening(img, size=(3, 3))    # Morphological closing  closed = grey\_closing(opened, size=(3, 3))    # Unsharp masking  sharpened = img - closed  enhanced = img + sharpened    return enhanced  def curvelet\_transform(img):  """  Applies curvelet transform to decompose the image into approximation and detailed coefficients.  """  cA, (cH, cV, cD) = dwt2(img, 'db1')  return cA, (cH, cV, cD)  def fusion\_approx(cA\_ir, cA\_vi):  """  Fuses the approximation coefficients using PCA.  """  X = np.stack([cA\_ir.ravel(), cA\_vi.ravel()], axis=1)  pca = PCA(n\_components=1)  cA\_fused = pca.fit\_transform(X).reshape(cA\_ir.shape)  return cA\_fused  def fusion\_detail(cH\_ir, cV\_ir, cD\_ir, cH\_vi, cV\_vi, cD\_vi):  """  Fuses the detailed coefficients using the max rule.  """  cH\_fused = np.maximum(cH\_ir, cH\_vi)  cV\_fused = np.maximum(cV\_ir, cV\_vi)  cD\_fused = np.maximum(cD\_ir, cD\_vi)  return cH\_fused, cV\_fused, cD\_fused  def fuse\_images(ir\_img, vi\_img):  """  Fuses the infrared and visible images using the proposed method.  """  # Resize the images to the same shape  ir\_img = cv2.resize(ir\_img, vi\_img.shape[:2][::-1])  # Enhance the source images  ir\_enhanced = morph\_unsharp\_mask(ir\_img)  vi\_enhanced = morph\_unsharp\_mask(vi\_img)    # Apply curvelet transform  cA\_ir, (cH\_ir, cV\_ir, cD\_ir) = curvelet\_transform(ir\_enhanced)  cA\_vi, (cH\_vi, cV\_vi, cD\_vi) = curvelet\_transform(vi\_enhanced)    # Fuse the approximation and detailed coefficients  cA\_fused = fusion\_approx(cA\_ir, cA\_vi)  cH\_fused, cV\_fused, cD\_fused = fusion\_detail(cH\_ir, cV\_ir, cD\_ir, cH\_vi, cV\_vi, cD\_vi)    # Reconstruct the fused image  fused\_img = idwt2((cA\_fused, (cH\_fused, cV\_fused, cD\_fused)), 'db1')    return fused\_img  def dwt\_sharpen\_fusion(ir\_img, vi\_img):  """  DWT and Sharpen filter based fusion.  """  # Resize the images to the same shape  ir\_img = cv2.resize(ir\_img, vi\_img.shape[:2][::-1])    # Apply DWT and sharpen filter  cA\_ir, (cH\_ir, cV\_ir, cD\_ir) = dwt2(ir\_img, 'db1')  cA\_vi, (cH\_vi, cV\_vi, cD\_vi) = dwt2(vi\_img, 'db1')    cA\_fused = (cA\_ir + cA\_vi) / 2  cH\_fused = np.maximum(cH\_ir, cH\_vi)  cV\_fused = np.maximum(cV\_ir, cV\_vi)  cD\_fused = np.maximum(cD\_ir, cD\_vi)    dwt\_sharpen\_result = idwt2((cA\_fused, (cH\_fused, cV\_fused, cD\_fused)), 'db1')  return dwt\_sharpen\_result  def pca\_multimodal\_fusion(ir\_img, vi\_img):  """  PCA based multimodal fusion.  """  # Resize the images to the same shape  ir\_img = cv2.resize(ir\_img, vi\_img.shape[:2][::-1])    # Apply PCA  X = np.stack([ir\_img.ravel(), vi\_img.ravel()], axis=1)  pca = PCA(n\_components=1)  pca\_multimodal\_result = pca.fit\_transform(X).reshape(ir\_img.shape)  return pca\_multimodal\_result  def curvelet\_fusion(ir\_img, vi\_img):  """  Curvelet multi-scale transform based fusion.  """  # Resize the images to the same shape  ir\_img = cv2.resize(ir\_img, vi\_img.shape[:2][::-1])    # Apply curvelet transform  cA\_ir, (cH\_ir, cV\_ir, cD\_ir) = curvelet\_transform(ir\_img)  cA\_vi, (cH\_vi, cV\_vi, cD\_vi) = curvelet\_transform(vi\_img)    cA\_fused = fusion\_approx(cA\_ir, cA\_vi)  cH\_fused, cV\_fused, cD\_fused = fusion\_detail(cH\_ir, cV\_ir, cD\_ir, cH\_vi, cV\_vi, cD\_vi)    curvelet\_result = idwt2((cA\_fused, (cH\_fused, cV\_fused, cD\_fused)), 'db1')  return curvelet\_result  def dwt\_unsharp\_fusion(ir\_img, vi\_img):  """  DWT and unsharp masking based fusion.  """  # Resize the images to the same shape  ir\_img = cv2.resize(ir\_img, vi\_img.shape[:2][::-1])    # Apply DWT and unsharp masking  ir\_enhanced = morph\_unsharp\_mask(ir\_img)  vi\_enhanced = morph\_unsharp\_mask(vi\_img)    cA\_ir, (cH\_ir, cV\_ir, cD\_ir) = dwt2(ir\_enhanced, 'db1')  cA\_vi, (cH\_vi, cV\_vi, cD\_vi) = dwt2(vi\_enhanced, 'db1')    cA\_fused = fusion\_approx(cA\_ir, cA\_vi)  cH\_fused, cV\_fused, cD\_fused = fusion\_detail(cH\_ir, cV\_ir, cD\_ir, cH\_vi, cV\_vi, cD\_vi)    dwt\_unsharp\_result = idwt2((cA\_fused, (cH\_fused, cV\_fused, cD\_fused)), 'db1')  return dwt\_unsharp\_result  # Load the input images  ir\_img = cv2.imread('B:\\morph\_processing\\Input\_Images\\stairs\\stairs\_ir.png', cv2.IMREAD\_GRAYSCALE)  vi\_img = cv2.imread('B:\\morph\_processing\\Input\_Images\\stairs\\stairs\_vi.png', cv2.IMREAD\_GRAYSCALE)  # Fuse the images using the different methods  dwt\_sharpen\_result = dwt\_sharpen\_fusion(ir\_img, vi\_img)  pca\_multimodal\_result = pca\_multimodal\_fusion(ir\_img, vi\_img)  curvelet\_result = curvelet\_fusion(ir\_img, vi\_img)  dwt\_unsharp\_result = dwt\_unsharp\_fusion(ir\_img, vi\_img)  proposed\_result = fuse\_images(ir\_img, vi\_img)  # Save the fused images  cv2.imwrite('dwt\_sharpen\_result.png', dwt\_sharpen\_result)  cv2.imwrite('pca\_multimodal\_result.png', pca\_multimodal\_result)  cv2.imwrite('curvelet\_result.png', curvelet\_result)  cv2.imwrite('dwt\_unsharp\_result.png', dwt\_unsharp\_result)  cv2.imwrite('proposed\_result.png', proposed\_result) |
| **OUTPUT:** | **Terminal:**    **Input Images:**  **VI Image: IR Image:**    **Curvelet: DWT Sharpen**    **DWT Unsharp: PCA Multimodal**    **Proposed Result:** |
| **REFERENCE:** | S. K. Panguluri and L. Mohan, "An Enhanced Image Fusion Framework Using Morphological Operations Based Unsharp Masking," 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2021, pp. 1-6, doi: 10.1109/ICCCI50826.2021.9402531. keywords: {Measurement;Visualization;Surveillance;Transforms;Image fusion;Image reconstruction;Morphological operations;infrared image;visible image;morphological operations based unsharp masking;curve-let transform},  <https://ieeexplore.ieee.org/document/9402531> |
| **CONCLUSION:**  We concluded that the introduced enhanced infrared (I-R) and visible (V-I) image fusion framework represents a significant advancement for surveillance and remote sensing applications. Through the integration of morphological operations and unsharp masking, along with curvelet transform and fusion rules such as PCA and max fusion, the framework produces superior fusion results. Comparative analysis demonstrates its enhanced visual quality and metric values over existing methods, highlighting its potential for enhancing visibility and situational awareness in surveillance scenarios and beyond. | |