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

| **Experiment 7** | |
| --- | --- |
| **AIM :** | Image Enhancement using Spatial Filtering |
| **OBJECTIVE:** | * Develop an adaptive enhancement technique capable of effectively regulating illumination and reflectance components in challenging lighting conditions. * Implement an "Iterative Homomorphic Filter" to iteratively attenuate the illumination component and enhance the reflectance component based on image types. * Evaluate the proposed filter's performance compared to conventional enhancement schemes, particularly in scenarios with optical sources where conventional methods fail due to excessive amplification and saturation of high-intensity regions. * Assess the enhancement efficiency of the proposed technique through both subjective and objective measures, aiming to demonstrate its superiority over existing algorithms available in the literature. |
| **INTRODUCTION:** | Conventional image enhancement methods often excel in improving dark images but struggle with those containing optical sources, leading to loss of vital original information due to over-amplification and saturation. To address this, an adaptive "Iterative Homomorphic Filter" is proposed. This filter, tailored to different image types, iteratively adjusts illumination and reflectance components based on image intensity. Experimental results show its superiority over existing algorithms in enhancing images under challenging illumination, preserving more original information. |
| **BLOCK**  **DIAGRAM:** |  |
| **IMPLEMENTATION:** | import numpy as np  import cv2  import matplotlib.pyplot as plt  from scipy.fftpack import fft2, ifft2, fftshift, ifftshift  def global\_avg\_intensity(img):  """Calculates the global average intensity of the input image"""  return np.mean(img)  def allocate\_filter\_params(avg\_intensity):  """Allocates the IHF filter parameters based on the global average intensity"""  if avg\_intensity < 70:  h\_gain = 2.53  l\_gain = 0.9  num\_iter = 2  cutoff = 120  elif 70 <= avg\_intensity < 145:  h\_gain = 1.58  l\_gain = 0.9  num\_iter = 3  cutoff = 1500  else:  h\_gain = 1.38  l\_gain = 0.9  num\_iter = 3  cutoff = 1500  return h\_gain, l\_gain, num\_iter, cutoff  def homomorphic\_filter(img, h\_gain, l\_gain, cutoff):  """Applies the Homomorphic Filter to the input image"""  img\_log = np.log1p(img)  img\_fft = fftshift(fft2(img\_log))  rows, cols = img.shape  crow, ccol = rows // 2, cols // 2  mask = np.zeros((rows, cols), np.float32)  y, x = np.ogrid[:rows, :cols]  d = np.sqrt((x - ccol)\*\*2 + (y - crow)\*\*2)  mask = 1 - np.exp(-(d \*\* 2) / (2 \* (cutoff \*\* 2)))  img\_filtered = mask \* img\_fft  img\_ifft = np.real(ifft2(ifftshift(img\_filtered)))  img\_enhanced = np.expm1(img\_ifft)  return img\_enhanced  def iterative\_homomorphic\_filtering(img, num\_iter):  """Applies the Iterative Homomorphic Filtering"""  avg\_intensity = global\_avg\_intensity(img)  h\_gain, l\_gain, num\_iter, cutoff = allocate\_filter\_params(avg\_intensity)  out\_1st\_iter = homomorphic\_filter(img, h\_gain, l\_gain, cutoff)  out\_final\_iter = out\_1st\_iter.copy()  for i in range(1, num\_iter):  h\_gain = h\_gain \* np.exp(-0.1 \* i)  l\_gain = l\_gain \* np.exp(-0.1 \* i)  out\_final\_iter = homomorphic\_filter(out\_final\_iter, h\_gain, l\_gain, cutoff)  fused\_img = 0.33 \* img + 0.33 \* out\_1st\_iter + 0.33 \* out\_final\_iter  return fused\_img  # Load the input image  input\_image = cv2.imread('B:\\07\_Dark\_Enhancement\\small\_baby.jpeg', cv2.IMREAD\_GRAYSCALE)  # Function to display image and its histogram  def display\_with\_histogram(title, img):  plt.figure(figsize=(10, 5))  plt.subplot(1, 2, 1)  plt.title(title)  plt.imshow(img, cmap='gray')  plt.axis('off')    plt.subplot(1, 2, 2)  plt.title('Histogram')  plt.hist(img.ravel(), bins=256, range=(0, 256), color='black', alpha=0.6)  plt.xlabel('Pixel Value')  plt.ylabel('Frequency')    plt.tight\_layout()  plt.show()  # Display the input image with its histogram  display\_with\_histogram('Input Image', input\_image)  # Apply Histogram Equalization  he\_image = cv2.equalizeHist(input\_image)  display\_with\_histogram('Histogram Equalization', he\_image)  # Apply Gamma Correction  gc\_image = np.uint8(np.power(input\_image / 255.0, 0.5) \* 255)  display\_with\_histogram('Gamma Correction', gc\_image)  # Apply Local Adaptive Gamma Correction with epsilon to prevent divide by zero  epsilon = 1e-8 # Small value to avoid division by zero  lagc\_image = input\_image.copy()  for i in range(input\_image.shape[0]):  for j in range(input\_image.shape[1]):  local\_mean = np.mean(input\_image[max(0, i-10):min(input\_image.shape[0], i+10),  max(0, j-10):min(input\_image.shape[1], j+10)]) + epsilon  gamma = 1 / (local\_mean / 255.0)  lagc\_image[i, j] = np.uint8(np.power(input\_image[i, j] / 255.0, gamma) \* 255)  display\_with\_histogram('Local Adaptive Gamma Correction', lagc\_image)  # Apply Piecewise Linear Transformation  plt\_image = cv2.normalize(input\_image, None, 0, 255, cv2.NORM\_MINMAX, cv2.CV\_8U)  display\_with\_histogram('Piecewise Linear Transformation', plt\_image)  # Apply Iterative Homomorphic Filtering  ihf\_1st\_iter = homomorphic\_filter(input\_image, 1.58, 0.9, 1500)  ihf\_2nd\_iter = homomorphic\_filter(ihf\_1st\_iter, 1.58 \* np.exp(-0.1), 0.9 \* np.exp(-0.1), 1500)  ihf\_fused = 0.33 \* input\_image + 0.33 \* ihf\_1st\_iter + 0.33 \* ihf\_2nd\_iter  display\_with\_histogram('IHF Fused', ihf\_fused) |
| **OUTPUT:** | **Terminal:**    **Input Image:**    **Histogram Equalization:**  **Local Adaptive Gamma Correction :**    **Gamma Correction:**    **Piecewise Linear Transformation:**    **IHF Fused:** |
| **REFERENCE:** | A. Nayak and A. Acharya, "Enhancement of Dark Images in Presence of Optical Sources Using Iterative Homomorphic Filter (IHF)," 2023 1st International Conference on Circuits, Power and Intelligent Systems (CCPIS), Bhubaneswar, India, 2023, pp. 1-7, doi: 10.1109/CCPIS59145.2023.10291651. keywords: {Optical filters;Reflectivity;Optical attenuators;Lighting;Optical saturation;Filtering algorithms;Traffic control;dark image enhancement;homomorphic filtering;contrast improvement;image fusion},  <https://ieeexplore.ieee.org/document/10291651> |
| **CONCLUSION:**  In conclusion, the proposed "Iterative Homomorphic Filter" presents a promising solution to the limitations of conventional image enhancement techniques, particularly in handling images with optical sources. By effectively regulating illumination and reflectance components based on image types, this adaptive filter demonstrates superior performance in preserving original information while enhancing image quality. The experimental results underscore its efficacy, surpassing existing algorithms in achieving enhanced images under challenging illumination conditions. This signifies its potential for practical applications in various fields requiring image enhancement, promising better preservation of crucial details in visually demanding scenarios. | |