Context Based Pseudo Coloring

Khulna University of Engineering & Technology
Department of Computer Science and Engineering

CSE 4128: Image Processing and Computer Vision Laboratory

Submitted To:

Dr. Sk. Mohammad Masudul Ahsan
Professor
Department of Computer Science and Engineering

Dipannita Biswas
Lecturer
Department of Computer Science and Engineering

Submitted By:

Farhatun Shama
Roll: 1907033
Year: Fourth
Semester: First

Date: 30 June 2024

Objectives

The primary objectives of this project are as follows:

- To color an input grey-scale image based on a reference color image
- To obtain the histogram of the input image and match it to the histogram of reference image
- To detect and match objects from input and reference image
- To perform local histogram matching
- To learn about different image operations for pseudo coloring

Introduction

In the field of digital image processing, pseudo-coloring is a powerful technique that transforms grayscale images into color images by mapping each intensity level to a color based on a predefined colormap. This method not only enhances the visual aesthetics but also significantly improves the interpretation of image data by highlighting features that are not readily apparent in the original single-channel images. The process of pseudo-coloring is particularly valuable in applications where distinct features and elements within an image need to be distinguished or analyzed separately, such as in medical imaging, remote sensing, and materials science.

Object matching forms a crucial part of advanced image analysis, enabling the identification and alignment of corresponding features across different images. This process typically utilizes geometric and textural descriptors such as moments, area, and Haralick features. Moments, which are quantitative measures derived from the shape of an object, facilitate the capturing of essential shape characteristics invariant to translation, scale, and rotation. The area of an object provides a simple yet powerful descriptor for size comparison, while Haralick features, extracted from the Gray-Level Co-occurrence Matrix (GLCM), offer rich texture information, encompassing properties such as energy, contrast, and homogeneity. Together, these metrics allow for a robust comparison between objects in different images, ensuring that similar features are accurately matched and aligned irrespective of their spatial transformations or variations in lighting and perspective. Like object matching, histogram matching is a sophisticated technique that adjusts the intensity values of an image to match those of a target image. Both of these techniques can play a crucial role in the field of reference image based pseudo coloring.

Tools Used

- **Python**: A high-level, interpreted programming language known for its ease of use and readability. Python's vast ecosystem of libraries and frameworks significantly accelerates the development of applications, especially in complex data analysis and GUI development.
- **OpenCV (Open Source Computer Vision Library)**: A highly optimized library focused on real-time image processing and computer vision applications. OpenCV is used extensively for its powerful image processing capabilities, such as image transformations, filtering, and object detection.
- **Tkinter**: This is the standard GUI toolkit for Python. Tkinter allows the easy and effective management of graphical user interface elements, making applications accessible to users.

Theory

Feature descriptors are pivotal in image processing for tasks such as object recognition and matching. They enable the extraction and representation of significant attributes from images, which can then be used to perform various analyses and enhancements. In this project, features like- area,perimeter, Haralick features, and histogram matching techniques has been used to enhance image analysis and object matching capabilities.

General Features: These descriptors provide insights into the shape and structure of objects within an image.

- **Area** (A): Represents the total number of pixels that make up the object.

$$A = \sum_{\text{all pixels}} 1$$

- **Perimeter** (P): The total length of the object's boundary.

$$P = \text{Number of boundary pixels}$$

- **Roundness**: A measure of how closely the shape of an object approaches that of a perfect circle.

$$\text{Roundness} = \frac{4\pi A}{P^2}$$

- **Form Factor**: Indicates the compactness of an object.

$$\text{Form Factor} = \frac{4\pi A}{P^2}$$

- **Axis Ratio**: The ratio of the major axis to the minor axis of the object's bounding ellipse.

$$\text{Axis Ratio} = \frac{\text{Major Axis}}{\text{Minor Axis}}$$

- **Orientation**: The angle between the x-axis and the major axis of the ellipse that has the same second moments as the region.

$$\text{Orientation} = \tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$$

- **Extent**: The ratio of the pixels in the region to the pixels in the total bounding box.

$$\text{Extent} = \frac{\text{Area}}{\text{Bounding Box Area}}$$

- **Solidity**: The ratio of the area of the object to the area of its convex hull.

$$\text{Solidity} = \frac{\text{Area}}{\text{Convex Hull Area}}$$

- **Eccentricity**: The ratio of the distance between the foci of the ellipse and its major axis length.

$$\text{Eccentricity} = \sqrt{1 - \frac{\text{Minor Axis}^2}{\text{Major Axis}^2}}$$

- **Feret Diameter**: The maximum distance between any two parallel tangents on opposite sides of the object's projection.

$$\text{Feret Diameter} = \max(\text{distance between parallel tangents})$$

Haralick Features: Derived from the Gray-Level Co-occurrence Matrix (GLCM), Haralick features describe the texture of an object.

- **Energy** or Angular Second Moment:

$$\text{Energy} = \sum_i \sum_j P(i,j)^2$$

- **Contrast**:

$$\text{Contrast} = \sum_{n=0}^{N-1} n^2 \left[\sum_{i,j:|i-j|=n} P(i,j) \right]$$

- **Homogeneity**:

$$\text{Homogeneity} = \sum_i \sum_j \frac{P(i,j)}{1+(i-j)^2}$$

Histogram Matching: This technique adjusts the pixel intensity distribution of an image to match that of a reference image. The process involves mapping the cumulative distribution function (CDF) of the pixel values from the input image to match the CDF of the target image:

Methodology

A basic interface has been designed with the help of tkinter to make the project more user friendly. In this project, input image can be colored locally or globally. For global coloring multiple ways are used. They are- global histogram matching, coloring image based on matched objects directly and by using object based histogram matching. The object matching is done in 2 ways- by using regular features and by using haralick features.

Global Pseudocoloring

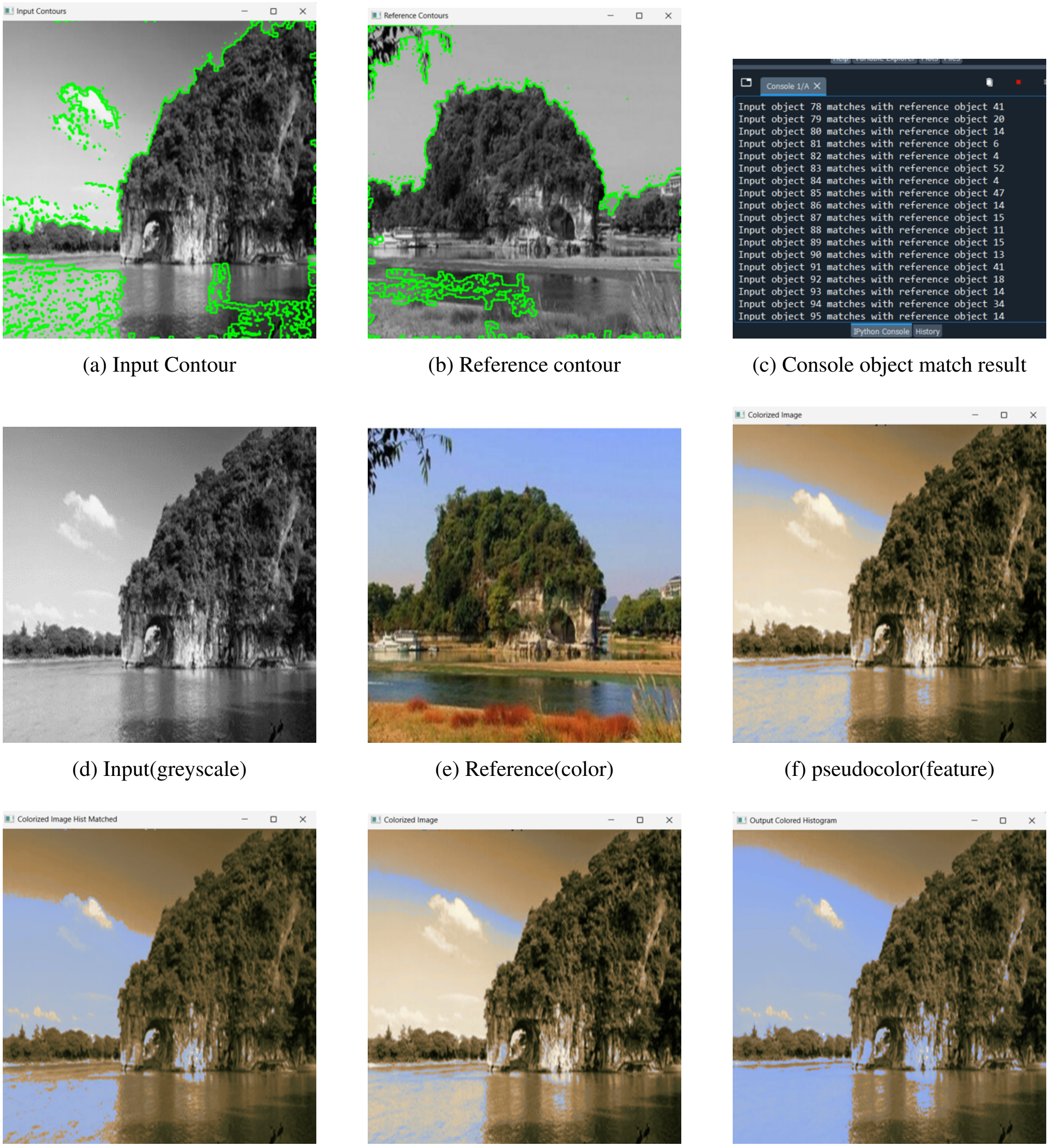
The object-based global pseudocoloring approach employed in this project incorporates several sophisticated image processing techniques to achieve accurate color mapping between the input and reference images. The methodology is systematically outlined as follows:

1. **Preprocessing of Images**:
    - **Closing Operation**: To ensure that the objects within the images are well-defined and free of internal gaps, a morphological closing operation is first applied. This step is crucial for filling any holes within the objects in both the input and reference grayscale images, leading to more robust contour detection.
2. **Contour Detection**:
    - Using edge detection algorithms, contours are extracted from both the input and reference images. This process identifies distinct objects by outlining their boundaries, which facilitates subsequent feature extraction and matching.

3. **Feature Vector Calculation**:
   - For each detected object, a feature vector is computed. These vectors encapsulate critical attributes of the objects, serving as the basis for comparing and matching objects across the input and reference images.
4. **Object Matching**:
   - Each feature vector in the input image is compared against all feature vectors from the reference image using a suitable distance metric. The objective is to find the most similar object, and this is determined by identifying the pair of objects (one from the input and one from the reference image) that exhibits the minimum distance between their feature vectors.
5. **Bounding Rectangle Calculation**:
   - Once matching objects are identified, the bounding rectangles for these objects are calculated. These rectangles are critical for defining the regions within the images where color mapping will be applied.
6. **Local Color Mapping**:
   - A local colormap is derived by mapping the grayscale values of the reference image object (reference grey patch) to the corresponding color values (reference color patch). This colormap is then used to pseudocolor the input image object.
7. **Application of Color Maps**:
   - The input image object is colored in two distinct ways using the derived local colormap:
     (a) **Direct Coloring**: The color values from the colormap are directly applied to the corresponding regions of the input image object.
     (b) **Histogram Matched Coloring**: Before applying the colormap, the histogram of the input image object's patch is matched to that of the reference image object's patch to align their intensity distributions. The colormap is then applied, ensuring that the colors reflect the matched histogram characteristics.

In addition to these object-based techniques, global histogram matching was also employed:

- **Global Histogram Matching**: This method involves adjusting the input image's histogram to directly match the histogram of the reference grayscale image. By doing so, the overall intensity distribution of the input image is altered to more closely resemble that of the reference, setting the stage for a coherent color mapping.
- **Global Color Mapping**:
  - A global colormap is generated by analyzing the reference grayscale image along with its colored counterpart. For each grayscale intensity, the mean RGB values are calculated to create a comprehensive map.
  - This global colormap is then applied to the input grayscale image, transforming it into a color image where each grayscale level is replaced by its corresponding color in the colormap, effectively pseudocoloring the entire image based on global characteristics.



(a) Input Contour

(b) Reference contour

(c) Console object match result

(d) Input(greyscale)

(e) Reference(color)

(f) pseudocolor(feature)

(g) pseudocolor (object Histogram)

(h) Pseudocolor Haralik

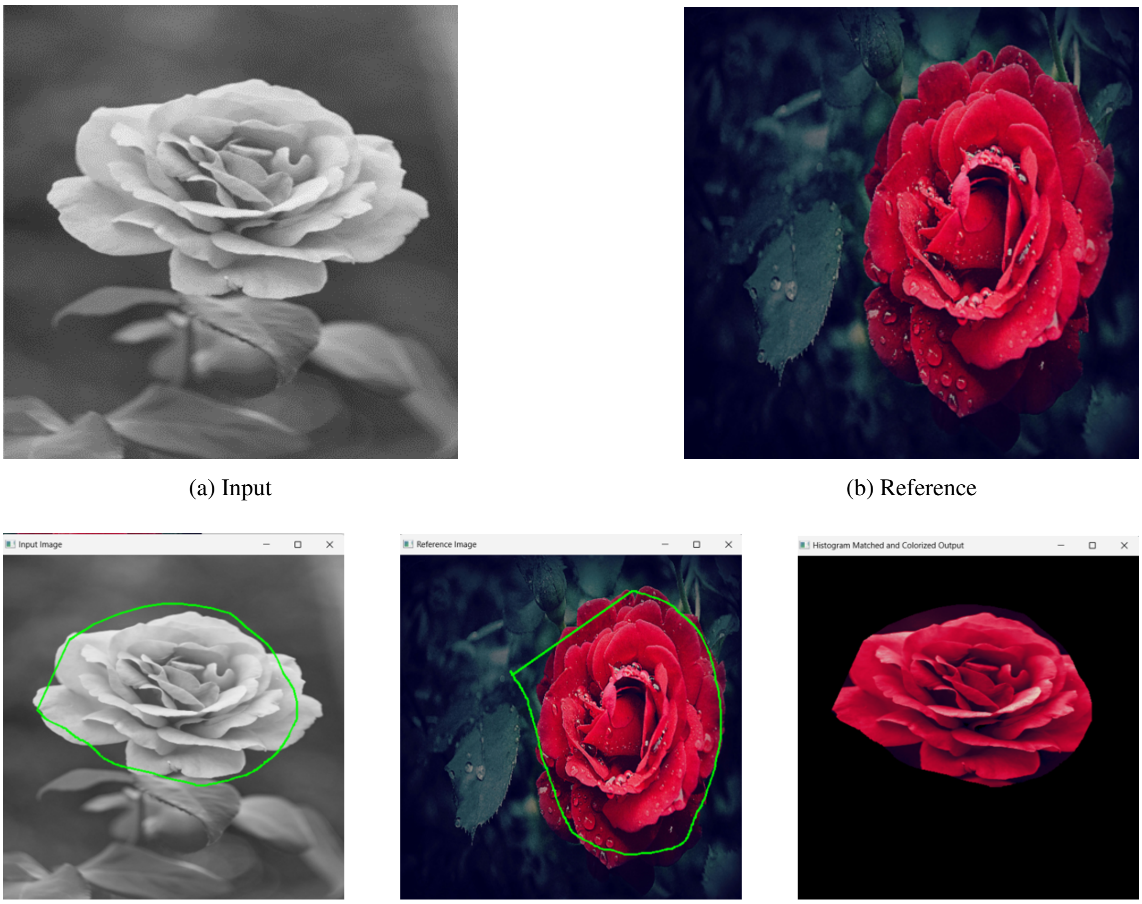
(i) pseudocolor (global Histogram)

Figure 1: Global Pseudocoloring

Local Pseudocoloring

Local pseudocoloring is a focused approach where specific regions within an image are enhanced to highlight important features or areas of interest. This method is particularly useful for detailed analysis and visualization of selected areas. The process involves several key steps:

1. **Selection of Region of Interest (ROI)**:
    - Users manually select regions of interest on both the reference and the input images. This selection is typically driven by the specific features or areas that are under study.
    - The selected ROIs dictate the scope within which all subsequent coloring and analysis occur, ensuring that enhancements are localized to regions deemed important by the user or application needs.
2. **Generation of Colormap**:
    - A colormap is generated using the indices from the grayscale ROI of the reference image and the corresponding colored ROI. This colormap maps specific grayscale intensities to colors.
    - The colormap serves as a translation mechanism between grayscale values and color values, based on the color distribution within the reference image's selected ROI.
3. **Application of Colormap to Input ROI**:
    - The input ROI is traversed pixel by pixel, applying the derived colormap to each pixel. This step transforms the grayscale ROI in the input image into a colorized version that reflects the color characteristics of the reference image.
    - Only the pixels within the selected ROI are colorized, while the rest of the image remains unchanged. This selective coloring emphasizes the ROI without altering the context provided by the surrounding areas.
4. **Histogram Matching**:
    - To enhance the visual consistency between the input and reference images, histogram matching is performed between the colorized input ROI and the original colored ROI of the reference image.
    - This process adjusts the color distribution in the input ROI to more closely resemble that in the reference ROI, improving the naturalness and fidelity of the color transformation.
5. **Marking and Coloring the ROI**:
    - In the final step, a mask is created where pixels inside the ROI are marked as '1' (indicating areas to be colorized) and all other pixels are marked as '0' (to be left unchanged).
    - During the coloring process, only those pixels marked as '1' are colored, ensuring that the rest of the image retains its original grayscale appearance.



(a) Input

(b) Reference

(c) Input ROI

(d) Reference ROI

(e) Pseudocolored Image

Figure 2: Local Pseudocoloring

Discussion

In the endeavor to enhance the visualization of grayscale images through pseudocoloring and object matching, the project confronted several notable challenges that impacted its efficiency and accuracy. The use of feature-based object matching occasionally resulted in mismatches, particularly when processing objects of disparate sizes and orientations, due to the scale and rotation invariance of the selected features. This issue was compounded by the computational intensity required to compute Haralick features, which significantly prolonged the output generation time. Furthermore, the orientation discrepancies between the reference and input images often hindered accurate matching, prompting the adoption of an average local colormap approach as a simplification strategy. Although this method eased the computational burden, it sometimes sacrificed match fidelity. Additionally, the morphological closing operations, while effective at eliminating irrelevant small objects and filling minor gaps, were not always sufficient to fully prepare the images for effective feature extraction and matching. Despite these hurdles, the project successfully demonstrated that pseudocoloring can substantially improve the interpretability and visual appeal of grayscale images, underscoring its potential utility in various applications of image processing.

Limitations

The implementation of pseudocoloring and object matching in this project revealed several limitations that could affect the overall performance and accuracy of the system. These limitations are detailed below to provide a comprehensive understanding of the challenges faced during the project's execution:

- **Inaccuracy in Object Matching:** The feature sets utilized for object matching occasionally failed to correctly identify and match objects across the reference and input images. This was particularly evident in scenarios where the objects varied significantly in their presentation or when the features were not robust enough to handle variations effectively.
- **Computational Efficiency of Haralick Features:** The calculation of Haralick features, while powerful for texture analysis, proved to be time-consuming. This significantly slowed down the output generation process, impacting the efficiency of the workflow and potentially limiting the method's applicability in real-time or high-volume scenarios.
- **Orientation Sensitivity:** The system demonstrated limitations in handling objects with differing orientations between the reference and input images. The lack of an adaptive mechanism to align orientations meant that the matching process could not always correctly pair objects, leading to inaccuracies in the pseudocoloring output.
- **Compromise on Color Accuracy:** To address some of the challenges with direct color mapping, an average local colormap was used instead of copying exact color values from the reference image. This approach, while reducing computational demands and simplifying the coloring process, resulted in a loss of color accuracy and did not faithfully reproduce the reference image's color nuances in the output.

These limitations highlight the need for further refinement of the feature extraction and matching algorithms, as well as enhancements in the computational strategies employed, to improve the fidelity and efficiency of the pseudocoloring process.

Conclusion

The exploration into the application of pseudocoloring techniques and object matching algorithms, as presented in this project, has demonstrated considerable potential in enhancing the interpretability of grayscale images. Despite the successes, the project faced inherent limitations related to the computational efficiency and accuracy of the feature-based matching process. Particularly, the reliance on Haralick features, while beneficial for detailed texture analysis, imposed significant computational demands that slowed the processing times. Moreover, discrepancies in object orientation and the use of an average local colormap to mitigate matching errors introduced challenges in maintaining color accuracy and fidelity.