

Color based image segmentation – Skin color detection.

Report:

The program begins by generating a composite image from a list of individual images using the `concat_images()` function. This composite image is then saved as a PNG file named 'final.png'.

Next, the program loads the composite image and converts it to HSV color space. The Hue and Saturation channels are then extracted from the HSV image. The program defines the number of bins for Hue and Saturation channels and creates a 3D histogram for these channels. The program loops through each pixel in the image and updates the histogram. Finally, the histogram is normalized.

After creating the histogram, the program plots the 3D histogram using the matplotlib library. The plot displays the frequency of the Hue and Saturation channels in the image.

The program then loads four additional images ('joy1.bmp', 'gun1.bmp', 'pointer1.bmp', 'Handshake_0.jpg') and resizes them to a width of 400 pixels while maintaining the aspect ratio.

For each pixel in the image, the program checks if the Hue and Saturation values are present in the normalized histogram. If the frequency of the Hue and Saturation value is greater than 0, the pixel is set to white. Otherwise, it is set to black. The resulting image is saved as a new file with the prefix 'result_'.

Overall, this program performs color-based image segmentation using HS-pair histogram thresholding. It creates a 3D histogram of the Hue and Saturation values in an image and uses this information to segment images into regions of interest. The program can be used for a variety of applications, including object recognition and tracking, image processing, and computer vision.

Improvements:

1. The program currently assumes that all image files will be loaded and processed successfully. It would be helpful to add error handling to catch cases where an image file cannot be loaded or processed correctly.
2. The program could be made more modular by breaking it down into smaller functions. This would make the code more readable and easier to maintain.
3. The program could be made more efficient by optimizing the loops used to calculate the histogram. One way to do this is to use numpy's vectorized operations instead of loops.
4. HS-pair histogram thresholding is a relatively simple algorithm for image segmentation. There are more sophisticated and accurate algorithms available that could be used instead.
5. The program could be made more user-friendly by adding a simple user interface that allows the user to select the input and output files and adjust the segmentation parameters.

Original	Segmented
	
	
	
	

