

Assignement 4

This is a Python code for a class called ImageTool which provides a simple GUI for measuring the length of a line segment and rectifying an artwork image. It uses the OpenCV library for image processing and the tkinter library for creating a window.

The `__init__` method initializes the class with an image path and creates some instance variables for storing the image and its copy, the points clicked by the user and a window name. The `run` method calls the `create_window` method to create a GUI window for displaying the image and capturing user input.

The `create_window` method creates a named window using the `cv2.namedWindow` method and sets a mouse callback using the `cv2.setMouseCallback` method to capture mouse events. It then enters a loop that continuously displays the image in the window and waits for user input. If the user presses 'c', the image is restored to its original copy and any points clicked by the user are cleared. If the user presses 'q', the window is destroyed and the program exits.

The `on_mouse_click` method is the callback function used to capture mouse events. If the user clicks the left mouse button, the (x, y) coordinates of the click are appended to the points list. If there is only one point in the list, a red circle is drawn around it. If there are two points in the list, a red circle is drawn around the second point, and the distance between the two points is calculated and displayed in red text above the first point. If there are three points in the list, the `extract_artwork` method is called to rectify the artwork image.

The `matrix_rectification` method takes an affine rectified artwork image as input and performs a metric rectification on it using a homography matrix. It first displays the affine rectified image and captures the coordinates of four corner points using a mouse callback. It then converts the corner points to a numpy array and computes the homography matrix using the `cv2.findHomography` method. Finally, it uses the `cv2.warpPerspective` method to perform the metric rectification and displays the result.

The `affine_rectified` method takes an artwork image as input and displays it in a window, prompting the user to select the four corner points of the artwork by clicking on them in counter-clockwise order. Once the user has selected the four points, it computes the affine transformation matrix using the `cv2.getAffineTransform` method and applies it to the artwork image using the `cv2.warpAffine` method to produce an affine rectified image. The affine rectified image is then passed to the `matrix_rectification` method to perform metric rectification.