Computer Graphics

Assignment 1 - Rasterization

Due date: Feb 4th 2025 (Tuesday) 11:59 PM

1 Tasks

With the provided main function, write 2 functions on your own in Python as follows:

i. **rasterization** that takes a blank (all pixel's values are zero) image frame with size 100 by 100 and a 3 by 2 matrix storing the location of 3 vertices of a triangle as input, apply the "contact" strategy, i.e., contact is regarded as coverage, to rasterize the triangle on the input image frame. Then return the rasterized image.

   E.g.,

   rasterization(image: numpy.ndarray, triangle: numpy.ndarray) → np.ndarray

ii. **supersample_rasterization** that takes a blank (all pixel's values are zero) image frame with size 100 by 100, a 3 by 2 matrix storing the location of 3 vertices of a triangle, and a supersample factor as input, then return the rasterized image with supersampling under the specific factor. Try different factors to see what has been changed.

   E.g.,

   supersample_rasterization(image: numpy.ndarray, triangle: numpy.ndarray, factor: int) → numpy.ndarray

Note that you may need to write another function for the Point-in-Triangle test using barycentric coordinates during the rasterization process.

2 Results

You can test your code with the following 4 sets of points:

s1: numpy.array([[22,61], [21,77], [96,66]])

s2: numpy.array([[21,20], [46,75], [88,68]])

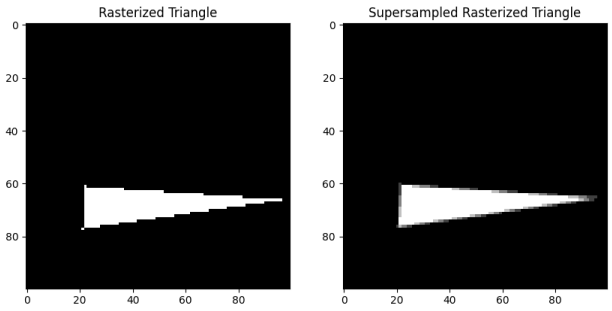
s3: numpy.array([[5,33], [31,75], [45,24]])

s4: numpy.array([[10,10], [10,90], [90,50]])

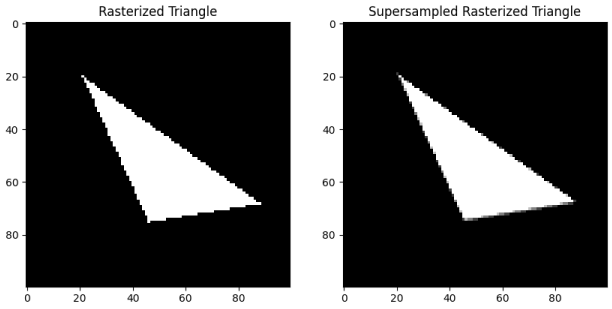
Your program should output results that look like in Figure 1 and Figure 2:

3 Items to submit

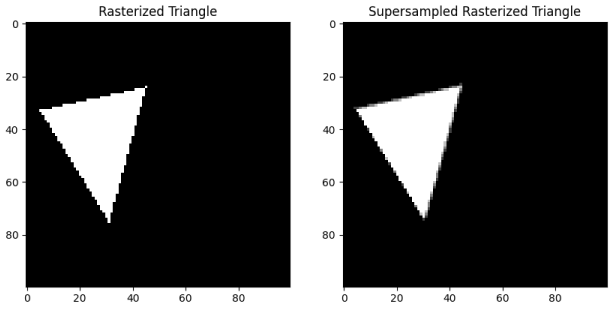
A zip file that contains your Python code and a PDF report showing the output figures from the main function.



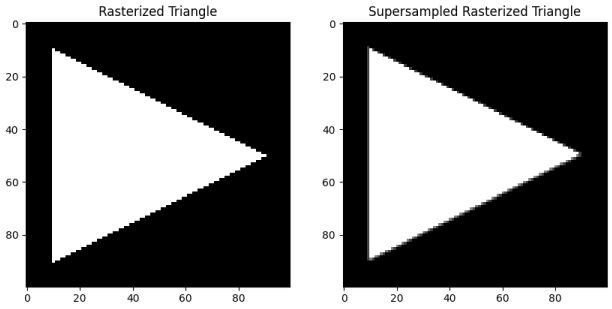
(a) Point set 1



(b) Point set 1

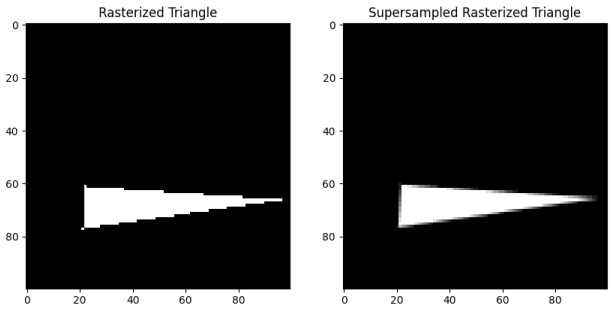


(c) Point set 3

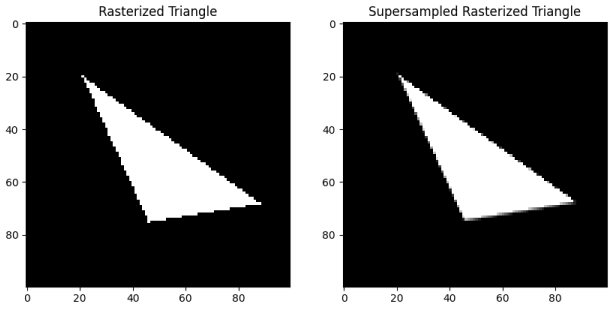


(d) Point set 4

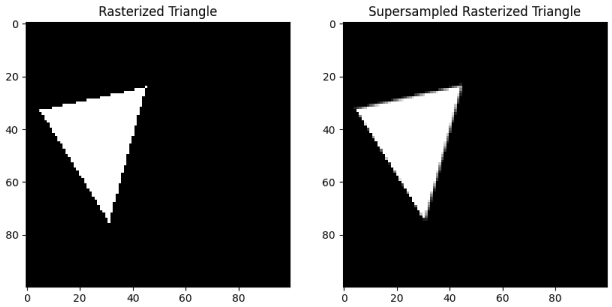
Figure 1: Supersampling factor is 4



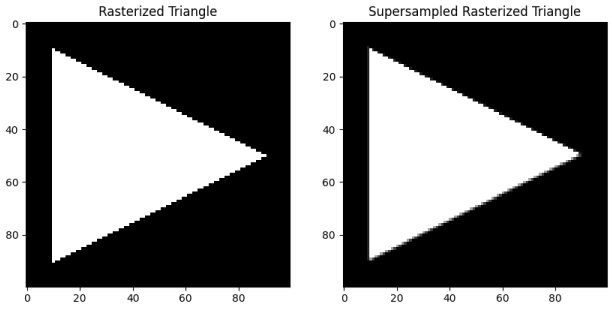
(a) Point set 1



(b) Point set 1



(c) Point set 3



(d) Point set 4

Figure 2: Supersampling factor is 8