**Abstract**

Boundary detection plays a critical role in various fields such as image processing, computer vision, and geographic information systems. In this paper, we present an efficient algorithm for finding convex-concave boundaries and establish a methodology for generating such a boundary. The algorithm can be divided into two sections: grid-based preprocessing of data, and linking of points using a binary tree. For the implementation, we looked through research papers and prior algorithms, and found the nexus of the idea which was a hull. Thus for a hull generation algorithm on a two dimensional geometric plane, we needed a simulation to illustrate our working, and in hindsight further improve the quality of our algorithm. Therefore, we opted to use a web-based simulation and created most of our simulator on standard JavaScript. The resulting algorithm boasted a space complexity of O (N) and a time complexity of O (2n + N). For the verification of the algorithm, we took some sample data provided in papers and ran our simulator and found that our data points should not be too far apart for the grid-based preprocessing to work efficiently. However, this can easily be solved by introducing a dynamic scale factor that performs a geometric transformation, typically an enlargement with a factor less than 1. This combined approach using convex-concave hulls and a binary tree offers a promising avenue for efficient boundary detection.