

### **TMA1301 - COMPUTATIONAL METHODS**

Project Title:

**Estimating the Size of a Geographical Region**

GROUP PROJECT

REPORT

TUTORIAL SECTION: TT07

LECTURE SECTION:TC04

LECTURER: Pee Chih Yang

SUBMISSION DATE: 15/2/2019

PREPARED BY:

|  |  |  |
| --- | --- | --- |
| NAME | ID | EMAIL |
| Mohamad Amirul Ariff bin Mohamad Yusof | 1161200646 | amirulariff80@gmail.com |
| Noorsyahmi Syahiran bin Syazali | 1171302474 | syahszli@gmail.com |
| Mohab Mohamed | 1161302776 | Mohabmohamed10@gmail.com |
| Baobaid, Husam Omar Ali | 1161304070 | hussambaobeid@yahoo.com |

TABLE OF CONTENTS

[**1.**](https://docs.google.com/document/d/1tUzaHPYp2sKA4GGUm1dK0lkLcYZQENaSm9g-Iz4_hdE/edit#heading=h.msrkn1sj31bq)**Introduction** [**3**](https://docs.google.com/document/d/1tUzaHPYp2sKA4GGUm1dK0lkLcYZQENaSm9g-Iz4_hdE/edit#heading=h.msrkn1sj31bq)

**2**[**.**](https://docs.google.com/document/d/1tUzaHPYp2sKA4GGUm1dK0lkLcYZQENaSm9g-Iz4_hdE/edit#heading=h.n3mpmfctxk1d)**Codes and Algorithms: 3**

**i) Extract the region of interest in form of image 3**

**ii) Separating region of interest from background 4**

**iii) Monte Carlo Simulation 5**

**3.Discussion and Conclusion 7**

# 

# 1.Introduction

This project focuses on developing a system for estimating the area of a geographical region, firstly done by separating the region of interest from the background and extracting the region of an interest from GeoChart in a form of a image using JavaScript. The image is then loaded to Octave and by using the Monte Carlo Simulation we estimate the area of interest.

# 2.Codes and Algorithms

**i) Extract the region of interest in form of image**

By using the below JavaScript code in a HTML file

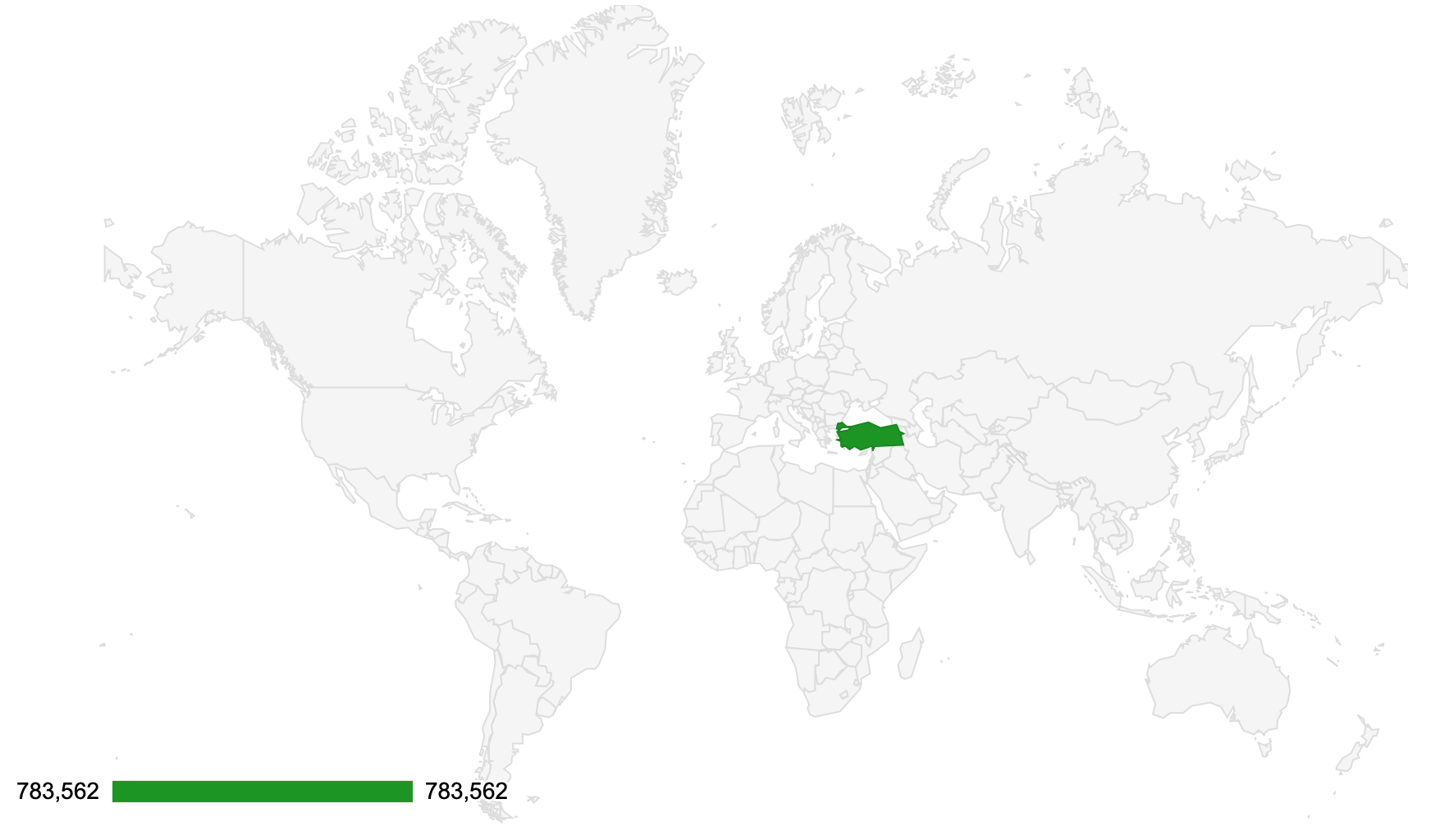
File name: RegionExtraction.html

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**<html>  
 <head>  
 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>  
 <script type="text/javascript">  
 google.charts.load('current', {  
 'packages':['geochart'],  
 'mapsApiKey': 'AIzaSyD-9tSrke72PouQMnMX-a7eZSW0jkFMBWY'  
 });  
 google.charts.setOnLoadCallback(drawRegionsMap);  
 function drawRegionsMap() {  
 var data = google.visualization.arrayToDataTable([  
 ['Country', 'Area'],  
 ['Turkey', 783562],  
 ]);  
 var options = {};  
 var chart = new google.visualization.GeoChart(document.getElementById('regions\_div'));  
 chart.draw(data, options);  
 }  
 </script>  
 </head>  
 <body>  
 <div id="regions\_div" style="width: 900px; height: 500px;"></div>  
 </body>  
</html>**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

After running the code, the output will be the worldwide map with the selected country (Turkey) shaded with green.

****

**ii) Separating region of interest from background**

We are able to get a closer and more refined picture by adding the code **region: 'TR'** and place it inside **var options = {};** this will zoom into the region and the output as below.

By cropping, we separate our region of interest.

****

The image then will be placed inside the Assignment folder as “Turkey.png”

**iii) Monte Carlo Simulation**

**INSTRUCTION TO USER BEFORE USING SIMULATION  
   
Prerequisite:** The user requires Octave or similar frontends that can run \*.m files, in order to run the simulation program. **a) estimate\_area (the main function) is called to start the program.  
   
b) To run the program:**1) User should set the directory of Octave to the Assignment folder.  
2) Key in : “estimate\_area” in the command window3) Press Enter

**SIMULATION CODE**

File name: estimate\_area.m

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**%To load the image and convert it to binary**

**img = im2bw(imread("Turkey.png"));**

**%Starting Monte Carlo Simulation**

**x = round(rand(1000,1) \* 1578);**

**y = round(rand(1000,1) \* 712);**

**x(x==0) = 1;**

**y(y==0) = 1;**

**AreaofRect = 1388292.63;**

**totalpoints = 1000;**

**SumOfBlack = 0;**

**for i=1:1000**

**if( img(y(i),x(i)) == 0 )**

**SumOfBlack = SumOfBlack+1;**

**endif**

**endfor**

**%Estimating the area of the region**

**AreaofObj = AreaofRect \* (SumOfBlack/1000);**

**AreaofObj**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**IMPLEMENTED AlGORITHM**

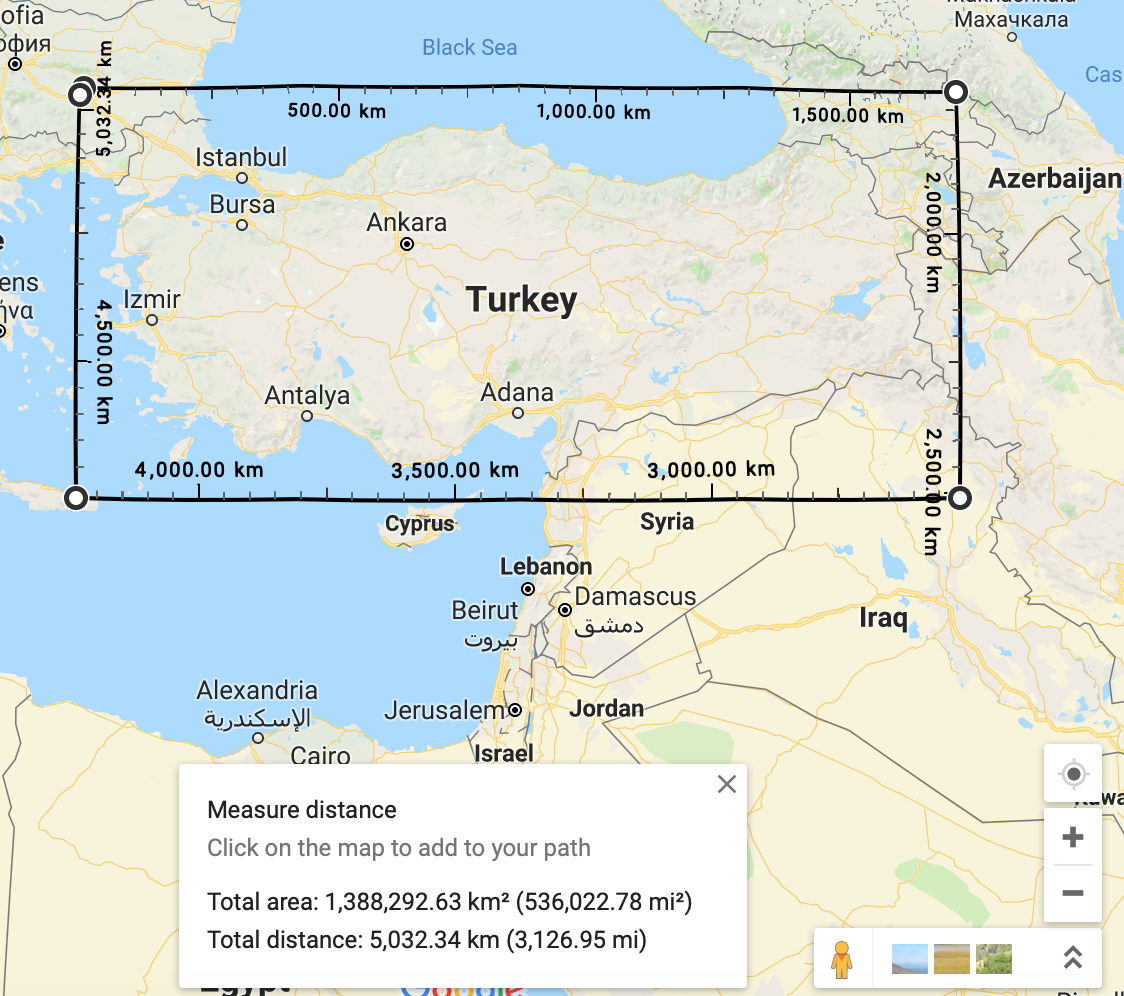
* Place the area of interest inside a rectangle that has a known area value
* Distribute random points all over the rectangle
* Determine the total number of the random points that lies inside the area of interest
* The area estimation of the region corresponds to the number of random points inside it and can be estimated by the given formula:

**Area of region = Area of rectangle \* (Number of random points inside region / Total number of random points)**

# 

# 3.Discussion and Conclusion

Simulation running requires getting the width and height in pixels of the extracted image, and also to find the estimated size of the image which can be done in Google Maps as below:

****

By having all parameters known that are required for the simulation, we can proceed with the estimation of the area. Here is an example of the program **estimate\_area.m** running on Octave:

**--> estimate\_area**

**AreaofObj: 772862.50712**

Running it second time,

**--> estimate\_area**

**AreaofObj: 766754.01955**

The actual region size is **783,562 km²** from Wikipedia.

From the simulation results, we can conclude that the area estimation simulation works because the estimation results are close from the actual region size.