CSCI 4000 Final Project

Spatial Accuracy Calculator

Learning by doing is an important strategy to retain knowledge. This course helps the students to acquire the ability to develop a real-world web application. In the development process, the students are required to apply the theories and knowledge (Apache Web Server, MySQL Database, PHPMyAdmin, PHP, JavaScript, AJAX) learnt in class. Students can choose between PHP, JavaScript as the main programming language.

Depending on your program design, you may not have all the skills needed to develop this application. Like in any real-world web application development scenario, you are required to find out and acquire these skills on your own (textbook, google etc). Most importantly, you need to understand and break down the problem which is in the GeoSciences domain, and convert it to a computer science problem.

This is an **individual** project.

You are required to build a **Spatial Accuracy Calculator** web application, for the following website:
http://www.leeleong.com/OtherWork/Spatial_Accuracy/index.php

The designers of your project team have completed the web user interface design using tables (not css). You are the only web application developer of the team. It is your job to design and build the main web application for the web site at this page:
http://www.leeleong.com/OtherWork/Spatial_Accuracy/calculator.php

The source code (html & css files) of the website should be downloaded from the D2L course website.

Problem Description:
An important aspect of using a GPS-enabled mobile device is to have accurate and precise determinations of location. **Accuracy** is a quantification of how close measurements are to the "target" or "true" value (the value accepted as being correct).

Input:
These coordinates can be entered as **decimal degrees** (DD) or in **degrees, minutes, and seconds** (DMS).

1. ***n* Control points/coordinates** (latitude x and longitude y) or "true" values taken from the USGS base map. In other words, there are n control points where i = 1, 2, 3, … n
   a. e.g. when i=1, Longitude x: $x_D = 28$, $x_M = 30$, $x_S = 59$,
      $x_{dd} = x_D + x_M/60 + x_S/3600 = 28 + 30/60 + 59/3600 = 28.51638889$
   b. e.g. when i=1, Latitude y: $y_D = 86$, $y_M = 27$, $y_S = 45$,
      $y_{dd} = y_D + y_M/60 + y_S/3600 = -86 + 27/60 + 45/3600 = 86.4625$
2. Collected field data of (many, depending on user input) **spatial coordinates** (latitude and longitude) from GPS-enabled mobile devices (for each control point).

Process:
Quantify Accuracy:

1. **Average Euclidean Error (*AEE*)** is the mean of the Pythagorean distances between *n* control points and the average of their associated field measurements.

$$AEE = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} \sqrt{(\Delta\,\mathrm{lat})^2 + (\Delta\,\mathrm{long})^2}$$

where Δ *lat* and Δ *long* are the differences between coordinates of the control points and the field measurement average.

2. **Root Mean Square Error** (***RMSE***), recommended by the Federal Geographic Data Committee. The horizontal (i.e., within the plane of the earth's surface) $RMSE_r$ for a region is given by

$$RMSE_r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\left(\bar{x}_{f,i} - x_{c,i}\right)^2 + \left(\bar{y}_{f,i} - y_{c,i}\right)^2\right)} = \sqrt{(RMSE_x)^2 + \left(RMSE_y\right)^2}$$

where

$$RMSE_x = \sqrt{\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\ (\bar{x}_{f,i} - x_{c,i})^2} \text{ and } RMSE_y = \sqrt{\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\ (\bar{y}_{f,i} - y_{c,i})^2}$$

are the root mean square errors in the *x*- and *y*- directions, respectively.

Here, $\bar{x}_{f,i}$ denotes the average *x*-coordinates of the *i*th set of field measurements; $x_{c,i}$ denotes the *x*-coordinate of the *i*th control point; $\bar{y}_{f,i}$ denotes the average *y*-coordinates of the *i*th set of field measurements; $y_{c,i}$ denotes the *y*-coordinate of the *i*th control point.

3. Accuracy per NSSDA ($Accuracy_{r1}$ and $Accuracy_{r2}$)
$Accuracy_{r1} = 1.7308(RMSE_r)$
$Accuracy_{r2} = 1.22385(RMSE_x + RMSE_y)$

4. Central Error (*CE*)
$CE = ((\text{Average } \Delta x)^2 + (\text{Average } \Delta y)^2)^{½}$

Output:

1. Average Euclidean Error (*AEE*)
2. Root Mean Square Error ($RMSE_r$)
3. Accuracy per NSSDA ($Accuracy_{r1}$ and $Accuracy_{r2}$)
4. Central Error (*CE*)

Example:

1. **User enters** the number of control points, *n*. Say users enter 3, *n*=3. Set an upper limit for *n*.
2. For each control point, **user can enter** the number of spatial coordinates, say user enters **5, 4 and 6** for *i*=1,2,3. Set an upper limit.
3. User **can choose** to enter field measurement coordinates in DD format or DMS format.
   a. If user chooses to enter in DMS format, the coordinates must be converted to DD format
   b. If user chooses to enter in DD format, no conversion is needed.
4. User enters the following DMS coordinates based on input mentioned in the previous 3 steps, x_d, y_d, x_c, y_c and subsequent data should be generated by the program.

Enter control point ID (20 characters max), i=1			C-RC-A				
Enter control point coordinates in Decimal Degrees (DD) or Degrees-Minutes-Seconds (DMS):							
Latitude, x	Longitude, y	Latitude, x			Longitude, y		
x_c	y_c	D	M	S	D	M	S
35.51166667	86.32388889	35	30	42	86	19	26
Enter field measurement coordinates in Decimal Degrees (DD) or Degrees-Minutes-Seconds (DMS):							
x_d	y_d	D	M	S	D	M	S
35.50888889	86.32333333	35	30	32	86	19	24
35.51444444	86.32555556	35	30	52	86	19	32
35.51027778	86.32305556	35	30	37	86	19	23
35.51305556	86.32666667	35	30	47	86	19	36
35.51138889	86.32361111	35	30	41	86	19	25

Average x_d = 35.51161111; Average y_d = 86.32444444

Enter control point ID (20 characters max), i=2			C-RC-B				
Enter control point coordinates in Decimal Degrees (DD) or Degrees-Minutes-Seconds (DMS):							
Latitude, x	Longitude, y	Latitude, x			Longitude, y		
x_c	y_c	D	M	S	D	M	S
35.52722222	86.34027778	35	31	38	86	20	25
Enter field measurement coordinates in Decimal Degrees (DD) or Degrees-Minutes-Seconds (DMS):							
x_d	y_d	D	M	S	D	M	S
35.52555556	86.33972222	35	31	32	86	20	23
35.52888889	86.34166667	35	31	44	86	20	30
35.52583333	86.34	35	31	33	86	20	24
35.52944444	86.3425	35	31	46	86	20	33

Average x_d = 35.52743056; Average y_d = 86.34097222

Enter control point ID (20 characters max), i=3			C-RC-C				
Enter control point coordinates in Decimal Degrees (DD) or Degrees-Minutes-Seconds (DMS):							
Latitude, x	Longitude, y	Latitude, x			Longitude, y		
x_c	y_c	D	M	S	D	M	S
35.48277778	86.4625	35	28	58	86	27	45
Enter field measurement coordinates in Decimal Degrees (DD) or Degrees-Minutes-Seconds (DMS):							
x_d	y_d	D	M	S	D	M	S
35.48055556	86.46222222	35	28	50	86	27	44
35.48083333	86.46388889	35	28	51	86	27	50
35.48527778	86.46194444	35	28	67	86	27	43
35.485	86.46222222	35	28	66	86	27	44
35.48111111	86.46	35	28	52	86	27	36
35.48444444	86.465	35	28	64	86	27	54

Average x_d = 35.48287037; Average y_d = 86.4625463

For the table **below**:
x_d: average x value of ith set of data points (field measurements)
y_d: average y value of ith set of data points (field measurements)
x_c: x value of ith control point
y_c: y value of ith control point

	ID	x_d	x_c	$\Delta x = x_c - x_d$	Δx^2	y_d	y_c	$\Delta y = y_c - y_d$	Δy^2	$SS = \Delta x^2 + \Delta y^2$	$(SS)^{½}$
1	C-RC-A	35.51161	35.51167	5.556E-05	3.086E-09	86.32444	86.32389	-5.556E-04	3.086E-07	3.117E-07	5.583E-04
2	C-RC-B	35.52743	35.52722	-2.083E-04	4.340E-08	86.34097	86.34028	-6.944E-04	4.823E-07	5.257E-07	7.250E-04
3	C-RC-C	35.48287	35.48278	-9.259E-05	8.573E-09	86.46255	86.46250	-4.630E-05	2.143E-09	1.072E-08	1.035E-04
					Average Δx^2 = 1.835E-08				Average Δy^2 = 2.643E-07	Average SS = 2.827E-07	Average $(SS)^{½}$ = 4.623E-04
					$RMSE_x$ = $(1.835E\text{-}08)^{½}$ = 1.355E-04				$RMSE_y$ = $(2.643E\text{-}07)^{½}$ = 5.141E-04	$RMSE_r$ = $(2.827E\text{-}07)^{½}$ = 5.317E-04	

1. Average Euclidean Error (*AEE*) = Average $(SS)^{½}$= **4.623E-04**
2. Root Mean Square Error ($RMSE_r$) = RMSEr = $(2.827E\text{-}07)^{½}$ = **5.317E-04**
    a. $RMSE_x = (1.835E\text{-}08)^{½}$ = **1.355E-04**
    b. $RMSE_y = (2.643E\text{-}07)^{½}$ = **5.141E-04**
3. Accuracy per NSSDA
    a. $Accuracy_{r1} = 1.7308(RMSE_r)$ = **9.203E-04**
    b. $Accuracy_{r2} = 1.22385(RMSE_x + RMSE_y)$ = **7.950E-04**
4. Central Error = $((\text{Average } \Delta x)^2 + (\text{Average } \Delta y)^2)^{½}$ = **4.398E-04**

Application Interface Design:

You can design your own user interface, but …
You must allow user to enter the number of control points, *n*.
You must allow user to enter the number of spatial coordinates for each control point.
You must allow user to enter coordinates based on the input mentioned above (*n* and each number of spatial coordinates).

The following output must be generated by the program:
1. Average Euclidean Error (*AEE*)
2. Root Mean Square Error ($RMSE_r$)
3. Accuracy per NSSDA ($Accuracy_{r1}$ and $Accuracy_{r2}$)
4. Central Error (*CE*)

Submission instructions:
You need to test the above programs, and provide **two test cases** for each program (question). Do a screen capture of the input and related output for each test case. Use any graphic editing software (e.g. Microsoft Paint, Adobe Fireworks, GIMP) to cut out the program input and output (from the screen capture), paste them into a word document under a related question number, save the document as a pdf file.

You need to submit the following:
1. A pdf file containing the screen captures of program input and output of all test cases, name the file **lastname_firstname_final_project.pdf**.
2. All html and php files. Zip your files into a single zip file (or rar file) **lastname_firstname_final_project.zip**.

Please submit electronic copy (the above mentioned **two files**) to D2L digital dropbox.

Grading guidelines (programming questions):
Your programs will be judged on several criteria, which are shown below.

- Correctness (50%): Does the program compile (run) correctly? Does the program do what it's supposed to do?
- Design (20%): Are operations broken down in a reasonable way (e.g. classes and methods)?
- Style (10%): Is the program **indented** properly? Do variables have **meaningful names**?
- Robustness (10%): Does the program handle erroneous or unexpected input gracefully?
- Documentation (10%): Do all program files begin with a **comment** that identifies the author, the course code, and the program date? Are all the classes, methods and data fields clearly **documented (comments)?** Are unclear parts of code **documented (comments)?** (Some items mentioned may not apply to some languages)

A program that does not compile (run) will get at most **50% of the possible points**.