

PROCESSING GLOBAL SELF-  
CONSISTENT HIERARCHICAL HIGH-  
RESOLUTION SHORELINE (GSHHS)  
DATA VERSION 1.3 INTO ESRI ARCGIS  
VECTOR AND RASTER DATA

CCG/SOG Working Paper

Andy Turner and Andy Nelson

December, 2004

## Abstract

This paper details processing of publicly available Global Self-consistent Hierarchical High-resolution Shoreline (GSHHS) Data Version 1.3 (GSHHS\_1.3) into vector and raster GIS data layers which dichotomise Earth's surface into those areas which are primarily land and those which are primarily water. In the most disaggregate form five different classes are distinguished: ocean (water0); land bounded by ocean (land0); water bodies contained and bounded by land0 (water1); land contained and bounded by water1 (land1); water bodies contained and bounded by land1 (water2).

GSHHS\_1.3 was made available on the 1<sup>st</sup> of October 2004 in a binary format via the following URL:

<http://www.soest.hawaii.edu/wessel/gshhs/gshhs.html>

GSHHS\_1.3 offers a minor improvement from the previous release GSHHS Version 1.2 (GSHHS\_1.2) in that: Lingering crossovers, duplicate points and unclosed polygon problems have been resolved for about 50 polygons. Major errors in the Puget Sound coastline have also been corrected.

In a preceding paper on processing GSHHS\_1.2 the uncertainties and nature of the source data are described; Turner and Nelson (2004). It is recommended that readers have this other paper available whilst reading this one, in order to make sense of all the details. This previous paper also details the motivation for this work.

GSHHS\_1.3 processed into a compressed ESRI ArcGIS interchange vector format has been made available at the following URL:

[http://www.geog.leeds.ac.uk/people/a.turner/data/gshhs1.3\\_dd.zip](http://www.geog.leeds.ac.uk/people/a.turner/data/gshhs1.3_dd.zip)

We encourage users of these data to consider their uncertainty and report errors and offer enhancements to the developers and maintainers of the GSHHS data.

## Contents

Abstract.....	2
Contents.....	3
Acronyms.....	3
1. Introduction .....	4
2. Converting ASCII data into ArcGIS ungenerate format.....	5
Figure 2.1 A map for describing the projection task .....	6
Figure 2.2 A map of the processed projected data .....	8
3. ArcGIS processing details .....	8
4. Further work.....	9
References.....	9

## Acronyms

ArcGIS	ESRI GIS
DD	Decimal Degrees
DM	Decimal Minutes
ESRI	Environmental Systems Research Institute
GIS	Geographic Information System(s)
GMT	Generic Mapping Tool
GSHHS	Global Self-consistent Hierarchical High-resolution Shoreline
GSHHS_1.2	GSHHS Data Release Version 1.2
GSHHS_1.3	GSHHS Data Release Version 1.3

## 1. Introduction

The Global Self-consistent Hierarchical High-resolution Shoreline (GSHHS) Data Version 1.3 (GSHHS\_1.3) was made available on the 1<sup>st</sup> of October 2004 in a binary format via the following URL:

<http://www.soest.hawaii.edu/wessel/gshhs/gshhs.html>

The source data are lines in a standard geographic projection with units of Decimal Degrees (DD) with the Western most shorelines crossing at 0° West (0° W) and Eastern most values crossing 360° East (360° E). The most southerly line represented the coastline of Antarctica. As might be expected, no coastlines in the data cross 90° North (90° N) or 90° South (90° S), but a number do cross 0° W and 360° E.

The source data were converted into an ASCII format file by a program called GSHHS Version 1.5 distributed with the data together with the Generic Mapping Tools (GMT) package Version 4.0 available via the following URL:

<http://gmt.soest.hawaii.edu/>

It is likely that the maintainers of these data will release updates and higher resolution data in the same format in the future.

A Java program was compiled to convert the ASCII data into an ESRI ARC / INFO (ArcGIS) ungenerate format file which was to be directly read into the proprietary Geographical Information System (GIS) software. Section 2 describes details of the Java program. Section 3 describes enhancements in the subsequent ArcGIS processing.

The resulting ArcGIS Interchange polygon vector data distinguishes 5 classes by a numerical code. Each polygon is assigned one (and only one) of these codes forming a continuous global surface where:

- 0 represents ocean;
- 1 represents land bounded by ocean (land0);
- 2 represents water bodies contained and bounded by land0 (water1);

- 3 represents land contained and bounded by water1 (land1); and,
- 4 represents water bodies contained and bounded by land1.

Please refer to Turner and Nelson (2004), our previous working paper for a more detailed description of the source data and a consideration of its uncertainties.<sup>1</sup>

## 2. Converting ASCII data into ArcGIS ungenerate format

This section describes the development of the Java program compiled to convert the ASCII data into ArcGIS ungenerate format. The Java source code is available on line at the following URL:

<http://www.geog.leeds.ac.uk/people.a.turner/src/andyt/java/misc/src/uk/ac/leeds/ccg/andyt/misc/GSHHS2ArcGIS.java>

The first step was to write code that would read in the source data file and write out a processed ArcGIS ungenerate format data file that had replaced all header lines for each coastline with a simple integer identifier and had added the key word “END” on a new line at the end of each set of coordinates and an “END” on a line at the end of the file. This was relatively straight forward and the resulting file could be used to generate a line or polygon coverage in ArcGIS using the GENERATE command.

A first look at the line or polygon coverage showed that Antarctica was problematic in that the coastline intersected itself due to a wrap-around effect. This had not been a problem in processing the Shapefile GSHHS\_1.2. The undocumented processing of the Shapefile must have resolved this issue. Nevertheless, some automatic way of identifying the Antarctic coastline so as to deal with it differently was wanted.

A key thing to notice and reason was that the Antarctica coastline is the only one which would completely ‘wrap-around’ from East to West. A logical consequence was that it

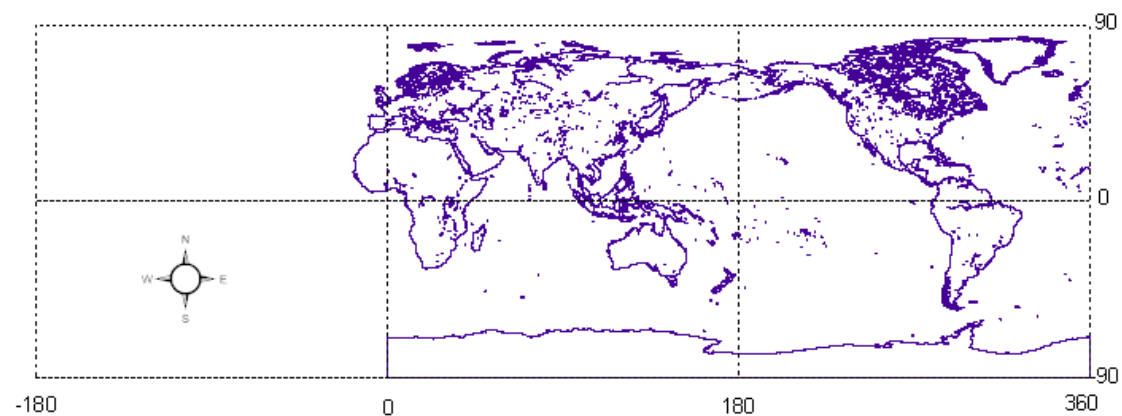
---

<sup>1</sup> Rather than continually referring to this or incorporating sections of it here, readers are kindly asked to either read this first or have it available whilst reading this in order to understand all the details herein.

could be identified and treated differently. The way this was done was by calculating the difference between the minimum Easting and maximum Easting of a coastline's coordinates. If the result was  $360^\circ$  then the coordinates were for the Antarctica coastline.

It was tested and found that all the coordinates of each coastline could be stored in memory. This greatly simplified matters as, the case of Antarctica could be caught and treated accordingly. Figure 2.1 is a map produced by inserting two additional coordinates in the Antarctic polygon between the wrap-around coordinates at  $(-90^\circ, 0^\circ)$  and  $(-90, 360^\circ)$  respectively. (The order in which these additional coordinates were inserted was determined from the general direction of the coordinates Eastings.)

Figure 2.1 A map for describing the projection task



The Antarctica coastline polygon would be reasonably straightforward to project and construct if it could be guaranteed that the coastline only crosses  $180^\circ\text{E}$  once (and indeed  $0^\circ\text{E}$  or  $360^\circ\text{E}$  once). Although for GSHHS\_1.3 this was the case, for the greatest flexibility in processing future versions of GSHHS the processing algorithm was developed to cope when this was not guaranteed. Supposing  $180^\circ\text{E}$  was crossed multiple times, say three times; then in projecting the data, there would be a need to insert additional coordinates so as to close an additional polygon<sup>2</sup>. (With it being crossed five times, there would be a need to ensure not one, but two additional polygon closures.) Indeed any coastline that crossed  $180^\circ\text{E}$  required intersection calculations to be performed and something had to be done to ensure closure of all the polygons.

---

<sup>2</sup> For a polygon to be closed it has to have the same start and end coordinates.

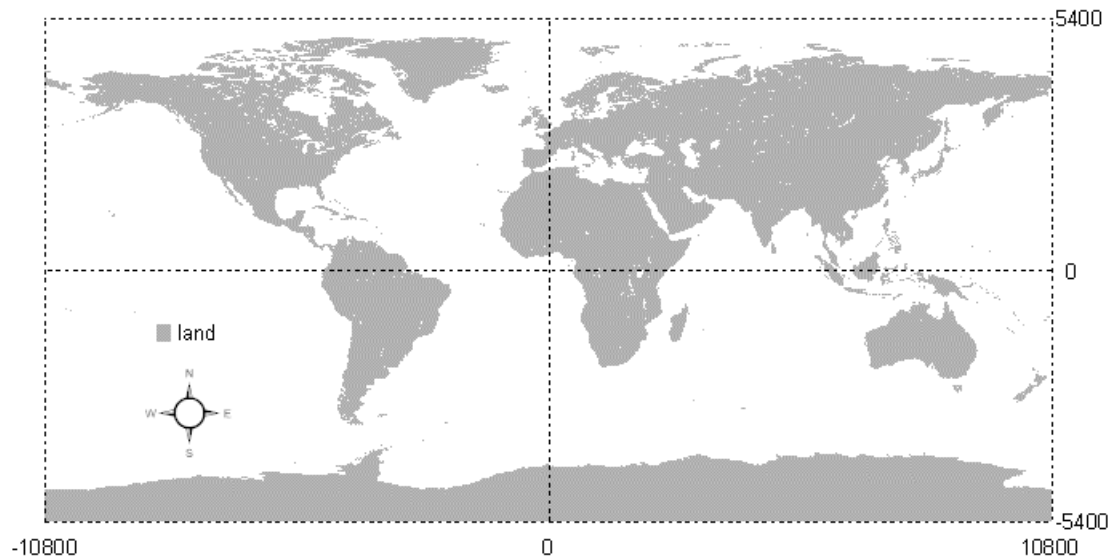
Now, coastline polygons which do not cross or touch  $180^{\circ}\text{E}$  could be simply dealt with. Either their coordinates were written out in order, or if their coordinate Eastings were greater than  $180^{\circ}$  then  $360^{\circ}$  was subtracted from them first. To cope with the other more complex cases where  $180^{\circ}\text{E}$  was crossed more work was needed. To do this work, the coordinates were loaded as an ordered collection of ordered collections of coordinates. This was done as follows: To begin with an ordered collection of coordinates was initialised and each coordinate read was added to it until  $180^{\circ}\text{E}$  was crossed. (If coordinates being added had Eastings greater than  $180^{\circ}$  then  $360^{\circ}$  was subtracted from them first.) When  $180^{\circ}\text{E}$  was crossed, the intersect Northing where it would cross  $180^{\circ}\text{E}$  ( $-180^{\circ}\text{E}$ ) was calculated to a given precision. (Basic trigonometry and the Java double primitive precision were used.) An appropriate intersect coordinate was then added to the collection of coordinates, then a new coordinate with the same Northing but with an Easting on the other side (i.e. Easting multiplied by a value of -1) was constructed and added to a newly initialised ordered collection of coordinates. The newly initialised collection was then to be added to until  $180^{\circ}\text{E}$  was crossed again. As and when this occurred, another new ordered collection was initialised and intersection performed as before. This process was to repeat as many times as necessary, until finally, the last coordinate of the particular coastline was read. Before the polygons can be written out with confidence, one further processing step is necessary.

If for a coastline,  $180^{\circ}\text{E}$  were crossed more than twice, it would be necessary to determine which of the collections of coordinates represent a single polygon (trunk), and which represent multiple individual polygons (limbs). It is always the case that one or other ordered set of ordered sets of coordinates are those representing the trunk and the other the limbs. To distinguish which is which, the maximum difference in the Northings for the coordinates that intersect  $180^{\circ}\text{E}$  (or  $-180^{\circ}\text{E}$ ) can be calculated by going through each collection of coordinates. The collection for which the difference in the Northings for the coordinates that intersect  $180^{\circ}\text{E}$  ( $-180^{\circ}\text{E}$ ) is greater belongs to the set comprising the single polygon trunk. The trunk can thus be written out in coordinate order, and the limbs can be written out, but with an additional coordinate added at their ends to ensure polygon closure.

There is one other thing: At the same time as doing all this, a value was to be multiplied to all Eastings and Northings so the coordinates were in different units. Multiplying by

60 would give coordinates in Decimal Minute (DM) units. Figure 2.2 is a map of the projected data.

Figure 2.2 A map of the processed projected data



### 3. ArcGIS processing details

This section provides details of the processing that was done in ArcGIS to produce the ESRI ArcGIS interchange vector format data. Firstly the PRECISION command was issued to ensure all coordinate components were to be constructed using ESRI ArcGIS double precision. Due to lingering crossovers, the data could not be generated as a polygon coverage so it was first generated as a line coverage using the GENERATE command and then converted into a polygon coverage using the CLEAN command specifying a very small fuzzy tolerance. For the record, the CLEAN fuzzy tolerance was declared to be 0.0000000010825644.

An intersection was detected by ArcGIS when building polygon topology at (18.703°, 79.754°) and (18.697°, 79.759°). This is in addition to the intersections of the Antarctic coastline. It is an open question as to whether there are still more lingering crossovers in the data, but a visual inspection revealed this as such. One way to identify further lingering crossovers would be to use ArcGIS and iteratively remove the intersection where the attempted BUILD of polygon topology failed. A list of



intersection locations could be provided to the maintainers of the GSHHS data. The one intersection we discovered has been reported, but we have left it to others to clean the data.

With the polygon coverage, what was next needed was to identify which were land and which were water. The method for doing this is unchanged since the GSHHS\_1.2 processing except that the precision of coordinates was now higher and lower fuzzy tolerance could be used. For the three UNION command used in Step 8 the following fuzzy tolerances were declared: 0.0000000010048, 0.0000000010703, 0.0000000010826.

Having added identification for the five classes of the hierarchy, the polygon coverage was exported in the ArcGIS interchange format using the EXPORT command. Both the ungenerate file and the interchange file have been compressed using Windows XP and made available online at the following URLs:

[http://www.geog.leeds.ac.uk/people/a.turner/data/gshhs\\_f.zip](http://www.geog.leeds.ac.uk/people/a.turner/data/gshhs_f.zip)

[http://www.geog.leeds.ac.uk/people/a.turner/data/gshhs1.3\\_dd.zip](http://www.geog.leeds.ac.uk/people/a.turner/data/gshhs1.3_dd.zip)

#### 4. Further work

There is an ongoing challenge to define and map geographical features and the boundaries between them, coastlines are but one feature. Always improvements can be made in the methods and more detailed information about uncertainty and variability can be made available for users of the data. Comparisons between GSHHS coastline data and other coastline data should be encouraged and a full evaluation of the pre-processed (reformatted) data should be done. In the mean time we hope this work is useful for our further research.

#### References

Nelson A., and Turner A. (2004) Processing Global Self-consistent Hierarchical High Resolution Shoreline Data Version 1.2 Into ESRI ArcGIS Vector And Raster Data. CCG, School of Geography Working Paper, November 2004.