14 October 2014

TO: DEEPWAVE Project File
FROM: Al Cooper
SUBJECT: A new terrain-elevation variable for DEEPWAVE

The Source For Data

During the Shuttle Radar Topography Mission (SRTM) of 2000,[1][2] the altitude of the Earth's surface was mapped from 56S to 60N latitude with resolution of 3 arc-sec or about 90 m at the equator. For the US and territories, the resolution was 1 arc-sec or about 30 m. The data from this mission is archived at this web site: http://www.webgis.com/srtm3.html. The files can be download in individual files that span 1 degree by 1 degree. The format of these files leads to the need for some processing that is documented here.

The R code that downloaded these files is in the 'chunk' of this document called 'download-zip-files'. Initially, the range downloaded covered 40S to 48S and 165E to 175E. After unzipping, the data set was about 115 MB. The heights in the 3-arc-sec files are presented in 1201×1201 arrays where the edges duplicate the values in the adjacent arrays. The missing-value flag is -32768. The format is row-major, i.e., the 1201 values for the first west-to-east row are presented first, then the next row to the north, etc. Because R is inherently column-major, there are some aspects of indexing in the code provided here that have indices reversed from what might have been expected. The unpacked files have 2,884,802 bytes.

The reference location for each 1-degree by 1-degree array is the name of the individual file (e.g., "S43E173.hgt" has a reference position of 43°S and 173°E at the center of the lower left element of the array). The values give the height in meters above the WGS84/EGM96 geoid. The measurement uncertainty was about 9 m at 90% confidence[3] (Farr et al.~2007), but there are some biases. The SAR-radar technique did not penetrate fully through vegetation and so might reflect the top of the vegetation canopy or some level intermediate between the canopy and the surface, and the radar penetrated a few meters into snow and so measured a height between the snow cover and the terrain (as measured in Feb. 2000). Also, there are some gaps, especially in mountainous areas.

Below, R code that downloads, unzips, and reads the data files is listed. The entire-Earth dataset would require about 25 GB to store, so the download should be limited in area to the region of the project. Here the latitude range from 40S to 48S and the longitude range from 165E to 175E was chosen, an area that required about 115 MB to store the unzipped files (with some missing because they were all over-ocean). Here the data files are saved in Rdata-format gzipped files suitable for loading via commands like "load(file='XS40E170.gz')", which will retrieve the 'height' matrix for that lat/lon square.

The first problem was that the values are binary two-byte or 16-bit signed integers and are in big-endian format (most significant byte first) while our processing machines are mostly little-

[1]Farr, T.G., M. Kobrick, 2000, Shuttle Radar Topography Mission produces a wealth of data, Amer. Geophys. Union Eos, v. 81, p. 583-585.

[2]Farr, T. G., et al. (2007), The Shuttle Radar Topography Mission, Rev. Geophys., 45, RG2004, doi:10.1029/2005RG000183

[3]The standard uncertainty would be about 5 m.

endian, so a byte-swapping conversion is necessary. This was readily performed by the R reading function 'readBin', as illustrated in the preceding code chunk. It was useful to construct a function that would return the terrain altitude for a given latitude and longitude, so that is shown in the next chunk:

```
HeightOfTerrain <- function (.lat, .long) {
  lt <- as.integer (.lat)
  lg <- as.integer (.long)
  if (is.na(lt)) {return (NA)}
  if (is.na(lg)) {return (NA)}
  if (lt < 0) {
    lNS <- "S"
    lt <- -lt
  } else {
    lNS <- "N"
  }
  if (lg < 0) {
    lEW <- "W"
    lg <- -lg + 1
  } else {
    lEW <- "E"
  }
  vname <- sprintf("Z%s%d%s%d", lNS, lt+1, lEW, lg)
  if (!exists(vname, .GlobalEnv)) {
    zfile <- sprintf("%s.gz", vname)
    if (file.exists(zfile)) {
      load(file=sprintf("%s.gz", vname))
      assign (vname, height, envir=.GlobalEnv)
    } else {
      return (NA)
    }
  }
  ix <- as.integer ((.long - floor (.long) + 1/2400) * 1200) + 1
  iy <- as.integer ( (ceiling (.lat) - .lat + 1/2400) * 1200) + 1
  hgt <- get(vname, envir=.GlobalEnv)[ix, iy]
  return (hgt)
}
```

Adding a netCDR terrain-height variable

For a netCDF file, it is then possible to define new variables that represent the elevation of the terrain below the aircraft and also the height of the aircraft above the terrain. For example, here is code that does this:

```
setwd ("./TerrainData")     # Save the data in a subdirectory
Project <- "DEEPWAVE"
Flight <- "rf12"
fname <- sprintf("%s%s/%s%s.nc", DataDirectory (), Project, Project, Flight)
fnew <- sprintf("%s%s/%s%sZ.nc", DataDirectory (), Project, Project, Flight)
# copy file to avoid changing original:
file.copy (fname, fnew, overwrite=TRUE)

## [1] TRUE

# load data needed to calculate the new variables:
Data <- getNetCDF (fnew, c("LATC", "LONC", "GGALTB"))
SFC <- vector ("numeric", length(Data$Time))
netCDFfile <- open.ncdf (fnew, write=TRUE)
# have to use a loop here because HeightOfTerrain looks
# up and loads needed files so is not suited to vector ops
for (i in 1:length (Data$Time)) {
  if (is.na(Data$LATC[i])) {next}
  if (is.na(Data$LONC[i])) {next}
  SFC[i] <- HeightOfTerrain (Data$LATC[i], Data$LONC[i])
}

# replace missing values with interpolated values for gaps up to 5 s in length:
#     (104 points in DW Flight 12, incl one at 83822)
#     ba,rn=FALSE means keep values as NA if na.approx can't replace them;
#     this is needed to keep SFC the same size as other variables.
SFC <- zoo::na.approx (SFC, maxgap=10, na.rm = FALSE)
SFC[is.na(SFC)] <- 0      # replace missing values with zero; mostly ocean pts
ALTG <- Data$GGALTB - SFC
Data["SFC"] <- SFC
SaveRData <- "DEEPWAVEterrain.Rdata.gz"
# comment one of these
save(Data, file=SaveRData, compress="gzip")
load(file=SaveRData)
```

This example was for project DEEPWAVE and flight rf12. Note in the code that there is a step for interpolating to fill in short periods (up to 10 s) that otherwise would be missing values; that is the 'zoo::' command above. When there is no terrain but only ocean, the dataset did not include lat-long squares for those regions, so there are also missing-value regions over ocean that are not filled in. I think the remaining missing-value regions after interpolation to fill small gaps are mostly over ocean and could be replaced by zero, so for now I have done that to have better

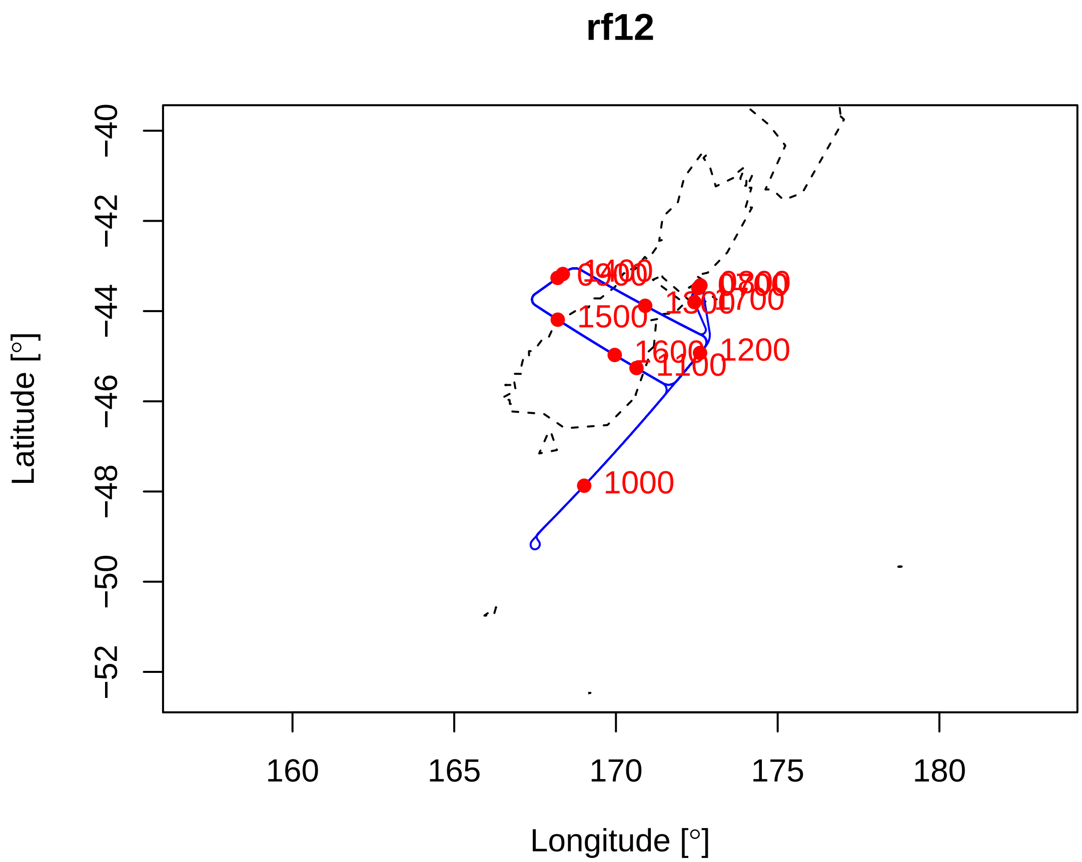

Figure 1: The flight track for the GV on flight 12 of the DEEPWAVE project.

appearing plots. The flight track is shown in Fig. 1, and the altitude of the terrain below the aircraft is shown in Fig. 2 for one pass from that flight over the South Island.

```
#SFC[is.na(SFC)] <- 0
r <- setRange(Data$Time, 82400,85200)
Z <- plotWAC (Data$Time[r], SFC[r], ylab="Terrain Elevation [m]")
title (Flight)
```

Considerations for routine implementation

The result of this processing is a new netCDF file that duplicates the original except for the addition of two variables, SFC and ALTG (respectively surface elevation and altitude above the ground). The netCDF file has an identifying 'Z' at the end of the name, in this case DEEPWAVErf12Z.nc. If

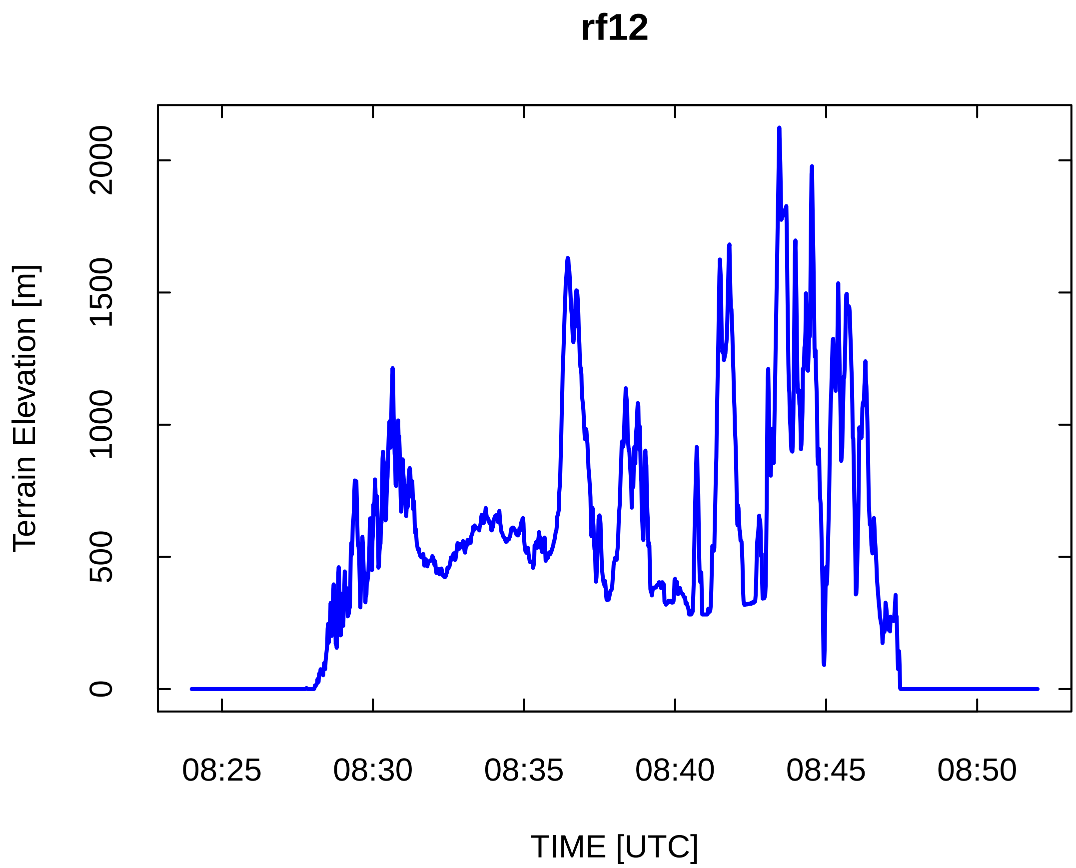

Figure 2: The elevation of the terrain below the position of the aircraft during a portion of DEEP-WAVE Flight 12.

this is a desirable variable to include in production files, that could be done in two ways. The function used here, 'HeightOfTerrain (lat, long)', with appropriate communication tools can be called from C/C++ programs, so that may be the best way to get this variable into nimbus. Alternately, this program could be run on production files to add the variable as I have done here, although some additional changes will be needed because this version doesn't add all the variable attributes that are RAF conventions. Finally, after starting this and downloading the data files I found another source, maintained in Scotland, that appears to be better because, although it is based on this same global data set, it has few gaps because they have been filled using reference topographic maps or other sources.

– End of Memo –

Reproducibility:

PROJECT:	HeightOfTerrain
ARCHIVE PACKAGE:	HeightOfTerrain.zip
CONTAINS:	attachment list below
PROGRAM:	HeightOfTerrain.Rnw
ORIGINAL DATA:	/scr/raf_data/DEEPWAVE/rf12.nc
GIT:	git@github.com:WilliamCooper/HeightOfTerrain.git

Attachments:
HeightOfTerrain.Rnw
HeightOfTerrain.pdf
DEEPWAVEterrain.Rdata.gz
SessionInfo