

Date: Tue, 20 May 2014 17:59:38 +0900
From: Tsukasa NAKANO
To: Naru Hirata
Cc: "TSUCHIYAMA, Akira", Kentaro Uesugi, MATSUNO Junya, Tooru Matsumoto,
Masayuki Uesugi, "NAKASHIMA, Yoshito", 道上達広, Takashi Matsushima,
Satoshi Okumura, Michihiko Nakamura
Subject: 球面上の点群の三角形分割

ひらたさま、

GSJ/AIST のなかのです。先週の E-mail (添付しました) で紹介した「凸包抽出用計算機コードの翻訳」の過程で以下のことに気づきました。

- [a] 球面上に分布している任意の点はその球に内接している凸多面体(== 凸包)の頂点と見なせる。
- [b] ぼくが翻訳した杉原厚吉さんが考案・開発された計算機コードを使うと、与えられた任意の点群に対して表面を三角形分割した凸包を得ることができる。

それゆえ、この凸包抽出用計算機コードを使えば「球面上に分布している点群の三角形分割」を高速に実行できます。そこで、翻訳したコード(CH3D)の応用例として、惑星・衛星探査機がレーザー高度計などで観測した天体の表面の「経度、緯度、高度」の「生」データからその天体の(第1近似の)形状モデルを簡便に作成するための計算機プログラムを書いてみました。それが実行する処理内容の概要は以下の通りです。

- [0] 様々な形式で「経度、緯度、高度」の生データが書き込まれている天体の観測データファイルをWEBから取得し、以下の処理に使いやすいかたちに整形する。
- [1] 各観測点の経度と緯度のデータから単位球の表面の x 、 y 、 z 座標値を計算し、それらの重複を調べる。こうして検出した、同一点で観測した複数個の高度データの取捨選択(例えば、平均値を使う)は重要だが、今回のプログラムではその調査結果をログに残すだけで重複している観測点の2番目以降に与えられたデータを単純に切り捨てている。
- [2] 凸包抽出用計算機コードを使って単位球の表面に投影した観測点を三角形分割する。その後、単位球に内接している凸多面体の頂点(== 単位球の表面に投影した観測点)すべての「経度、緯度、高度」のデータと凸多面体表面の三角形それぞれの頂点の「番号」のデータを出力する。
- [3] これらの頂点の単位球の中心からの距離をそこでの高度の観測データに応じた値に置き換えることにより単位球に内接する凸多面体を変形する。こうして得た(凸とは限らない)多面体をその天体の形状モデルとする。

今回は処理 [1] と [2] を実行するプログラム ll2ch (ll = Longitude and Latitude) を C 言語で新たに書きました。そして、観測データごとの C-shell scripts で ll2ch などを用いて [0]~[3] の一連の処理を行うことにしました。

プログラム ll2ch の使い方は後で詳しく説明します。~~この E-mail に添付した~~下記の gzip 圧縮されている書庫ファイル ss_ch.taz (ss = Surface of Sphere) の中に ll2ch のソースファイルや、それを用いて太陽系の 13 個の天体の合計 14 個の PLY/ZCP 形式の形状モデルのファイル群を作るための ~~以下の~~ C-shell scripts などが入っています。

gzip 圧縮されている書庫ファイル

http://www-bl20.spring8.or.jp/~sp8ct/tmp/ss_ch.taz

C-shell scripts

stooke.csh

P.J. Stooke の非球状・小天体のコレクションのページ

<http://publish.uwo.ca/~pjstooke/plancart.htm>

に置いてある以下の 11 個の天体のデータから形状モデルを作成する。

火星の衛星：Phobos、Deimos

木星の衛星：Amalthea

土星の衛星：Prometheus、Janus、Epimetheus、Hyperion

海王星の衛星：Proteus

小惑星：Vesta、Ida、Gaspia

なお、観測点の個数はいずれも 2701 で各々 216 点の重複があった。

gltm2b.csh

Clementine 探査機のレーザー高度計 (LIDAR) の観測データなどから M.T. Zuber 達が構築した月のデジタル地形モデル GLTM2B

<http://pds-geosciences.wustl.edu/missions/clementine/gravtopo.html>

を使って以下の 2 種類の形状モデルを作成する。

gltm2b_s：半径 1738 km の球を基準とした月の形状モデル

gltm2b_e：扁平率 1/3234.93 の回転楕円体を基準とした形状モデル

ただし、これらのモデルでは標高値を実際の値の 12.5 倍に強調している。また、GMTM2B の観測点の個数は 72548 でそれらのうちの 5 点が重複していた。

orad.csh

Pioneer Venus Orbiter (PVO) のレーザー高度計 (ORAD)

<http://nssdc.gsfc.nasa.gov/nmc/experimentDisplay.do?id=1978-051A-02>

の 144129 点の観測データ (2 点が重複) を用いて金星の半径 6052 km の球を基準とし、標高値を実際の値の 50 倍に強調した形状モデルを作る。

注

gltm2b.csh と orad.csh が作る形状モデルには以前に紹介した地球儀作成の場合と同様な標高値に応じた「色」を付けてあります。

http://www-bl20.spring8.or.jp/~sp8ct/tmp/stl_globe.pdf

また、オマケとして、上記の `scripts` で作った 14 個の PLY/ZCP 形式ファイルを使ってそれぞれの天体の形状モデルの像を自転軸 (?) の回りに 360 度回転させた鳥瞰図 (Bird Eye's View map) の GIF 形式動画ファイルを作成する C-shell script "`bev.csh`" も書庫ファイル `ss_ch.taz` に入れておきました。

注

`stooke.csh` が作る 11 個の形状モデルは「色無し」で少し寂しいので、`bev.csh` ではそれらの像の表面の陰影に加えて、形状モデルを構成している三角形を表す「ワイヤーフレーム」を描くようにしてあります。

書庫ファイル `ss_ch.taz` 中のプログラムの実行の手順は以下の通りです。

(0) 実行に必要な環境

`ss_ch.taz` 中のプログラムは UNIX 系の OS (Linux や MacOS X) の端末環境で実行することを想定しています。具体的には、以下の Linux に標準的なコマンド群がインストールされている必要があります。

```
tar、
make、gcc、wget、
csh (ファイル名 csh で起動する設定になっている tcsh でも良い)、
awk (ファイル名 awk で起動する設定になっている gawk でも良い)、
tail、tr、wc、cut、sort、cksum、gzip、grep、basename、sed
```

(1) 書庫ファイルのコピーとその解凍・展開

```
# この E-mail に添付した ss_ch.taz を適当な場所にコピーする。
tar xzf ss_ch.taz
→ ディレクトリ ss_ch/ が展開されるはず。
```

(2) ぼくが書いた ll2ch などのプログラム群のコンパイル

```
cd ss_ch
make
→ この結果、以下の 6 個の実行プログラムがコンパイルできるはず。
ll2ch、vf2zcp、zcp_dmp、zcp_stl、gif_trim、stl_bev_1
```

(3) 形状モデル "*.zcp.gz" (gzip 圧縮した PLY/ZCP ファイル) の作成

```
csh stooke.csh
csh gltm2b.csh
csh orad.csh
→ これらの実行後、ss_ch/ の下に以下のファイルができるはず。
stooke/*.txt、stooke/*.log、stooke/*.zcp.gz (各々 11 個ずつ)
gltm2bpr.*.gz (2 個)、gltm2b.log、gltm2b_?.zcp.gz (2 個)
pven001s.* (2 個)、orad.log、orad.zcp.gz
```

(4) 鳥瞰アニメーション "*.gif" の作成

```
csh bev.csh
```

→

14 個の GIF ファイルが入ったディレクトリ `ss_ch/bev/` ができるはずなので、それぞれの GIF を `firefox` など眺めてみて下さい。

疲れてきたのでプログラム `ll2ch` の説明などは次便にまわします。とりあえずは書庫ファイル `ss_ch.taz` を解凍・展開して、上記の手順 (1)~(4) を実行してみてください。WEB からのデータファイルのダウンロードの所要時間にもよりますが、すべての処理が 10 分以内に終わるはずですが、とり急ぎ、

添付した E-mail :

Date: Fri, 16 May 2014 15:35:23 +0900
 From: Tsukasa NAKANO
 To: "TSUCHIYAMA, Akira", Kentaro Uesugi, Takashi Matsushima, Satoshi Okumura
 Cc: MATSUNO Junya, Tooru Matsumoto, "NAKASHIMA, Yoshito", Naru Hirata, Masayuki Uesugi, 道上達広, Michihiko Nakamura
 Subject: 凸包抽出コードの翻訳

みなさま、

GSJ/AIST のなかのです。下記の文書などで紹介した画像などの上の 3 次元物体像の「凸包 (Convex Hull)」を抽出する計算機コードを大幅に書き換えました。

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/ch.pdf>

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/bb.pdf>

具体的には、上記 1 個目の文書の 1 ページ目に記されている杉原 厚吉さんが考案・開発された凸包抽出用の FORTRAN のコード「CHULL3」(ファイル名は "chull3.f") を C の言語コード「CH3D」(ファイルとしては "ch3d.h" と "ch3d.c" の 2 個) に翻訳しました。ただし、かなりの部分が「意訳」です。また、その過程で見つけたバグや「無駄」もできる限り除去しました。さらに、CH3D には抽出した凸包の面・辺・頂点のデータの利用を容易にするコードを付加しました。

これに伴い、CHULL3 を流用していたアプリケーション・プログラムのコードも CH3D を使うように書き改めました。具体的には、前記の 2 個の文書ファイルで紹介している以下の 2 × 2 個の書庫ファイル (ただし、gzip 圧縮した TAR 形式書庫ファイル "*.taz" と zip 形式のファイル "*.zip" の内容は同一) のそれぞれに入っている 7 個の計算機プログラムのコードを書き換えました。

http://www-bl20.spring8.or.jp/~sp8ct/tmp/stl_ch.taz

http://www-bl20.spring8.or.jp/~sp8ct/tmp/stl_ch.zip

zcp_ch : PLY/ZCP 形式ファイル上の物体像の凸包を抽出する。

stl_ch : STL 形式ファイル上の物体像の凸包を抽出する。

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/ch.taz>

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/ch.zip>

si_ch_zcp : 画像上の物体像の凸包の PLY/ZCP データを出力する。

si_ch_stl : 画像上の物体像の凸包の STL データを出力する。

si_ch : 物体像の凸包を抽出し、その内外の画素の「色分け」を行う。

si_chs : 画像上の複数の物体像それぞれの凸包を抽出する。

si_bbs : 画像上の複数の物体像それぞれの bounding box を計算する。

注

以前に紹介した CHULL3 を使っていたプログラムもソースコードや実行ファイルの名前に "_f" を付けたものとして残してあります。

これら 2 × 2 個の書庫ファイルには 32 ビット Windows 用実行ファイルも入れてあります。そして、Windows 以外のシステムでも GNU-C Compiler (GCC) をインストール済みなら上記の新しいプログラムのコンパイルは容易です :

プログラム zcp_ch と stl_ch のコンパイル

[1] stl_ch.taz もしくは stl_ch.zip をダウンロード・解凍・展開する。

[2] 端末環境でディレクトリ stl_ch/src/ に移動して以下を入力する。

```
make -f Makefile.mingw
```

si_ch_zcp、si_ch_stl、si_ch、si_chs および si_bbs のコンパイル

[1'] ch.taz もしくは ch.zip をダウンロード・解凍・展開する。

[2'] 端末環境でディレクトリ ch/ に移動して前記の [2] と同じ入力を行う。

新しいプログラムは旧来のものと完全互換です。そして、処理速度はかなり向上しています (当社比 10 倍弱)。これらをどうぞ御利用ください。とり急ぎ、

Date: Sat, 31 May 2014 17:28:17 +0900
From: Tsukasa NAKANO
To: Naru Hirata
Cc: "TSUCHIYAMA, Akira", Kentaro Uesugi, MATSUNO Junya, Tooru Matsumoto,
Masayuki Uesugi, "NAKASHIMA, Yoshito", 道上達広, Takashi Matsushima,
Satoshi Okumura, Michihiko Nakamura
Subject: ss_ch

ひらたさま、

GSJ/AIST のなかのです。5/20 の E-mail (Subject : 球面上の点群の三角形分割) の続きです。前回の話の補足を交えた以下のことをつらつらと書きます。

(0) Knoppix 上での C-shell scripts の実行

- (1) 新しい書庫ファイル
- (2) Magellan ADF を用いた金星の形状モデル作成
- (3) stooke.csh、gltm2b.csh と orad.csh が作成するファイル
- (4) プログラム ll2ch の使用法
- (5) vf2zcp、zcp_dmp、zcp_stl、gif_trim と stl_bev_1 の使用法

(0) Knoppix 上での C-shell scripts の実行

いつもは Windows Vista 機として使っているノート PC に DVD (の内容をコピーした USB メモリ) を装着して Knoppix を起動し、その上で天体の形状モデルを作成してみました。ただし、ぼくは以下の手順で Knoppix の準備をしました。

[0] Knoppix のインストール

Knoppix は CD-R、DVD や USB メモリなどから起動可能な Linux です。以下のページに置いてあるイメージファイル "*.iso" を CD-R や DVD に書き込めば Knoppix のインストールは (一応) 終わりです。

<http://www.risec.aist.go.jp/project/knoppix/>

[1] Knoppix を USB メモリにコピー

CD-R や DVD から起動すると初期設定を毎回行わないといけないので、Knoppix を USB メモリにコピーした方が良いです。これは Knoppix の GUI メニューにある「Install KNOPPIX to flash disk」で行えます。

[2] USB メモリから Knoppix を起動

計算機を再起動し、F12 キー押し下げで USB メモリからの起動を選択。

[3] 今回の処理に必要なパッケージを Knoppix にインストール

Knoppix で端末を開いて以下を入力すれば OK のはずです。

```
sudo apt-get update
sudo apt-get install tcsh
```

ただし、ぼくは DVD 用の Knoppix を使っているのですが、CD-R 用の Knoppix では他にも必要なパッケージがあるかもしれません。

この後、前回の E-mail で説明した手順で書庫ファイルのダウンロード・展開、C 言語プログラムのコンパイルと C-shell scripts すべての起動を順次行くと、天体の形状モデルやそれらの鳥瞰アニメーションを問題なく作成できました。

(1) 新しい書庫ファイル

5/27 の上杉くんあての E-mail にも書いたように、球面上の点群の三角形分割プログラム ll2ch が使っている「翻訳した凸包抽出用コード (CH3D)」を改造しました。また、前回のものに加えて、Magellan 探査機のレーダー高度計の観測データを使って金星の形状モデルを作成する C-shell script "adf.csh" も紹介することにしました。ただし、Magellan の多量 (約 270 万点) の観測データを WEB からダウンロード・整形する C-shell script は書くのも実行するのも大変なので、そのファイルは書庫ファイルに入れてお渡しします。それらのファイルを修正・追加した、以前に紹介したものと同名の書庫ファイルを SPring-8 の FTP サイトにアップロードしておきましたので、今後はこちらをお使い下さい。

http://www-bl20.spring8.or.jp/~sp8ct/tmp/ss_ch.taz

この新しい書庫ファイルに入っているファイルの内訳は以下の通りです。

C 言語のソースコードなど

Makefile

以下のプログラム群を gcc でコンパイルするための「Makefile」

error.c、ch3d.h、ch3d.c、ll2ch.c

プログラム ll2ch のソース。ただし、ch3d.c は以前のもの改良版。

vf2zcp.c

多面体の頂点 (vertex) と面 (face; ただし、三角形) を記述したテキストデータを PLY/ZCP 形式のファイルに変換する vf2zcp のソース

zcp_dmp.c

PLY/ZCP 形式ファイルの上のデータのダンプ (vf2zcp の真逆の処理) を行うプログラム zcp_dmp のソース

zcp_stl.c

PLY/ZCP 形式ファイルを color STL 形式のものに変換する zcp_stl のソース

sgif.h、sgif.c、gif_trim.c

GIF 形式ファイルの上の複数の画像の領域のトリミング用プログラム gif_trim のソース。ただし、"sgif.[h,c]" は「STL データ処理用のプログラム」のものを改造した GIF 形式ファイルの高速書き込み用コード。

line.h、line.c、facet.h、facet.c、hvd.h、hvd.c、stl_bev_1.c

鳥瞰図描画用の「HVD シリーズ」のプログラム stl_bev_1 のソース。ただし、そのコンパイルには前記の "sgif.[h,c]" も必要。

stooke.txt、gltm2b.txt、orad.txt、adf.txt

天体表面の高度の観測データに関するメモ。adf.txt を今回追加した。

stooke.csh、gltm2b.csh、orad.csh、adf.csh

天体の形状モデルを作成する C-shell scripts。adf.csh を今回追加した。

adf_llr.bin.gz、b2a_adf_llr.c

今回追加した Magellan ADF に関連したファイル（後述）

bev.csh

天体の形状モデルの鳥瞰図の GIF 形式アニメーションを作成する C-shell script。ただし、以前のものとは異なり、作成済みの GIF を再作成しない。

(2) Magellan ADF を用いた金星の形状モデル作成

大昔の記事 https://www.gsj.jp/data/chishitsunews/94_01_07.pdf で解説した、Magellan CD-ROMs の 1～10 巻に分割して格納されていた Magellan 探査機のレーダー高度計の観測データのファイル (ADF ; Altimetry Data File) をもとにして金星の形状モデルを作る C-shell script "adf.csh" を書きました。ただし、オリジナルの ADF にはデータが VAX 形式浮動小数点数として記録されており、また、今回の処理に不要なものも多量に含まれるので、それをそのまま取り扱うのは困難です。そこで、ADF から抽出した金星表面の経度・緯度・半径の浮動小数点数データだけを IEEE 形式に変換して記録した（大昔に作成した）ファイル adf_llr.bin.gz (llr == Longitude, Latitude and Radius) を再利用することにしました。そして、このバイナリデータ（大昔のものなのでバイトの並びの順は当時主流だった big endian です）をテキストデータに変換するプログラム b2a_adf_llr だけを新たに書きました。

adf.csh の起動法は前回紹介した stooke.csh などの場合と同じです。

```
csh adf.csh
```

注

adf.csh は処理に用いる b2a_adf_llr を処理の前に自分でコンパイルします。

この処理の所要時間は数分程度（Core2Duo SU9300 搭載のノート PC で約 5 分）、使用するメモリ量は 1（32 ビット計算機の場合）～ 2（64 ビット機上）GB 強です。最終的には、ディレクトリ ss_ch/ にテキストファイル adf.log と gzip 圧縮されている PLY/ZCP 形式ファイル adf.zcp.gz ができるはずですが、

adf.csh は adf.log に

- [1] Magellan が観測した金星の経度・緯度・半径のデータ点の総数
- [2] ll2ch で検出した重複していた観測点の点データ（経度・緯度・半径）
- [3] 金星の形状モデルの多面体を構成する頂点の総数

の3種類の行を書き込む設定にしてありますが、実際には [2] に相当する行の出力はありませんでした（つまり、Magellan のレーザー高度計のデータには重複している観測点はない）。しかし、[1] は 2724487、[3] は 2724486 だったので、今回の処理の過程で1個の観測点のデータが消失してしまったようです。これは凸包抽出の処理で32ビット長整数に変換した座標値を使っているために生じたエラーだと思われます（消失した観測点は凸包抽出の際に単位球の内部にあると見なされた?）。

C-shell script "bev.csh" を起動すれば `adf.zcp.gz` に入っている金星の形状モデルの（自転している像の）鳥瞰アニメーション（GIF形式画像ファイル `ssh_ch/bev/adf.gif`）を作成できます。

```

csh bev.csh

```

ただし、`adf.zcp.gz` はデータ量が多いので（`gzip` 圧縮されている状態で約 81 MB）、その鳥瞰アニメーションの作成には他の形状モデルの鳥瞰図の描画に比べて長い処理時間を要します（Core2Duo SU9300 搭載のノート PC で約 2分）。

(3) `stooke.csh`、`gltm2b.csh` と `orad.csh` が作成するファイル

前回の E-mail の以下の部分を補足します。

On Tue, 20 May 2014 17:59:38 +0900 Tsukasa NAKANO wrote:

- > (3) 形状モデル "*.zcp.gz"（`gzip` 圧縮した PLY/ZCP ファイル）の作成
- > `csh stooke.csh`
- > `csh gltm2b.csh`
- > `csh orad.csh`
- > → これらの実行後、`ss_ch/` の下に以下のファイルができるはず。
- > `stooke/*.txt`、`stooke/*.log`、`stooke/*.zcp.gz`（各々11個ずつ）
- > `gltm2bpr/*.gz`（2個）、`gltm2b.log`、`gltm2b_?.zcp.gz`（2個）
- > `pven001s.*`（2個）、`orad.log`、`orad.zcp.gz`

上記の3個の C-shell scripts は以下の4種類のファイルを作ります。

- [1] WEB から得た天体（3種類・合計 13 個）の表面の経度・緯度・半径の観測データが書き込まれたテキストファイルを `gzip` 圧縮したもの。
- [2] 配布元（NASA）が作成した上記のデータの説明書のテキストファイルを `gzip` 圧縮したもの。ただし、P.J.Stooke のデータにはこれはない。
- [3] プログラム `ll2ch` が出力した重複した観測点に関するログ（*.log）
- [4] `gzip` 圧縮した天体の形状モデルの PLY/ZCP 形式ファイル（*.zcp.gz）

これらのうちの [4] と前記の `adf.zcp.gz` は Stanford 大学で開発された PLY 形式のサブセットの、Z 社（Z-corporation）の3次元プリンタへの出力用の形式の CAD データファイルです。ぼくは使い勝手が良いので、カラーの物体像の CAD データを格納する用途にこの形式を採用しています。後述するプログラム `zcp_dmp` を使えば、この形式のファイルの CAD データをテキストデータに変換できます。

[3] のファイルの内容については後でプログラム ll2ch の使用法と一緒に説明します。なお、すでに説明したように、前記の `adf.log` には ll2ch が出力したログの行の前後にそれとは別のログ情報の 1 行づつを追加しています。

`stooke.csh` は [2] のファイルを作成しません。しかし、それが作った 11 個の [1] のファイルのそれぞれの先頭付近に、そこに記録されている天体の経度・緯度・半径の観測データに関する簡単な説明が記されています。そして、それらのデータは CSV 形式で書き込まれているので、[1] のファイルの内容は MS-Excel などで閲覧できます。なお、これらのファイルの名前は P.J.Stooke のページにあるファイルのものからそれぞれの天体の名前そのものに付け替えました。

On Tue, 20 May 2014 17:59:38 +0900 Tsukasa NAKANO wrote:

- > `stooke.csh`
- > P.J. Stooke の非球状・小天体のコレクションのページ
- > <http://publish.uwo.ca/~pjstooke/plancart.htm>
- > に置いてある以下の 11 個の天体のデータから形状モデルを作成する。
- > 火星の衛星：Phobos、Deimos
- > 木星の衛星：Amalthea
- > 土星の衛星：Prometheus、Janus、Epimetheus、Hyperion
- > 海王星の衛星：Proteus
- > 小惑星：Vesta、Ida、Gaspia
- > なお、観測点の個数はいずれも 2701 で各々 216 点の重複があった。

注

話が前後しますが、`stooke/*.txt` の上の観測データは「加工済み」のもので、その経度と緯度はいずれも 5 度刻みの「格子点」の値でした。

経度：0～360 度（格子点数は 73）

緯度：-90～90 度（格子点数は 37）

つまり、それぞれの $73 \times 37 = 2701$ 個の観測点のうちの経度 0 と 360 度の子午線上のものや、緯度 -90 と 90 度の両極の観測データは多重に記録されています。

`gltm2b.csh` と `orad.csh` が作成する [1] と [2] のファイルは以下の通りです。

`gltm2b.csh` : [1] `gltm2bpr.tab.gz` と [2] `gltm2bpr.lbl.gz`

`orad.csh` : [1] `pven001s.dat.gz` と [2] `pven001s.lbl.gz`

これらはいずれも gzip 圧縮されたテキストファイルで、特に [1] は CSV 形式で月や金星の表面の観測データが書き込まれています（ただし、行数が多いので、古い MS-Excel では閲覧できないかもしれません）。また、NASA が用意した [2] のファイルのそれぞれに観測データやその書き込み形式の詳細が記されています。

(4) プログラム ll2ch の使用法

ll2ch は指定した観測データファイル（標準入力の指定も可）の上のテキスト行に並んだ経度と緯度の値だけを読み取り、単位球の球面上に観測点を投影します。その後、それらのうちの重複した点を調査し、起動法に応じてその結果を出力します。また、出力先のファイルが指定された場合は、重複した点を除外した後にそれらの球面上の点群の三角形分割を行ない、その結果を球面に内接している凸多面体の頂点や面を記述するテキストデータとして出力します。

ll2ch は以下の 2 種類のいずれかの端末入力で起動します。

[1] ll2ch ll.txt

[2] ll2ch ll.txt ch.txt

ただし、文字列 ll.txt は観測データファイルの名前で "-" を指定するとデータを標準入力から読み取ります。そして、このテキストデータファイルの行構成は以下のようにする必要があります。

1 行目：観測点の個数 (N)

2 ~ N+1 行

少なくとも経度と緯度の 2 個の値（度単位の値）が行頭に空白かタブコード区切りで書き込まれている必要がある。ll2ch はこれらの後の行末までのデータを三角形分割の処理に使わない。しかしながら、三角形分割の結果を表す多面体の頂点のデータとしてこれらそれぞれの行全体をそのまま出力に転写する。

ll2ch を [1] のような端末入力で起動すると、球面上の重複した点の調査結果を表す、下記の 4 個以上の値が並んでいる行を標準出力に書き出します。

- 1) 重複した観測点が記されている観測点の番号。ただし、観測データファイル ll.txt の上の 2 行目に記されている点の番号を 0、3 行目の点番号を 1、... としている。
- 2) 重複した点ごとに ll.txt の上の並びの順に割り振った 0 から始まる番号。なお、球面上の観測点の三角分割の処理に際してはこの番号が 1 以上の重複した観測点を除外してそれを実行する。
- 3)、4)、...

その観測点に対応する観測データファイル ll.txt の上の行全体のコピー。

また、ll2ch を [2] のように 2 個のパラメータを付けて起動すると、球面上での観測点の三角形分割の結果の多面体の頂点と面（三角形）を記述するテキストデータを以下のような行構成（値の区切りは空白かタブコード）でファイル ch.txt ("- " を指定すると標準出力）に書き出します。

1 行目：多面体を構成する頂点 (N_v) と三角形の個数 (N_f)

2 ~ N_v+1 行

頂点に相当する観測データファイル上の観測点の行全体のコピー

N_v+2 ~ N_v+1+N_f 行

三角形の 3 頂点それぞれの番号。ただし、これは上記の 2 行目に並んでいる頂点の番号を 0 とする 2 ~ N_v+1 行の順番に応じた番号である。

ll2ch で三角形分割を行っても形状モデルは完成しません。引き続きファイル `ch.txt` の各頂点の行（上記の 2 ~ $Nv+1$ 行）に並んでいる経度、緯度と天体の半径などの観測値を使って形状モデルの表面の座標値を計算する必要があります。この `x`、`y`、`z` 座標値を（空白もしくはタブコード区切りで）並べた行を `ch.txt` の各頂点の行と置き換えたテキストファイルを作成し（これは `awk` を使えば容易に処理できます）、それをプログラム `vf2zp` に食わせてやれば `PLY/ZCP` 形式ファイルに変換できます。

(5) プログラム `vf2zcp`、`zcp_dmp`、`zcp_stl`、`gif_trim` と `stl_bev_1` の使用法

`zcp_stl` と `gif_trim` については以下の説明書をご覧ください。

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/stl.pdf#page=5>

また、以下はこれらを含む「STL データ処理用のプログラム」の公式ページです。

<https://www.gsj.jp/researches/openfile/openfile2006/openfile0448.html>

注

このページにリンクされている説明書と書庫ファイルは旧版です。

それから、`vf2zcp` と `zcp_dmp` も「STL データ処理用のプログラム」なのですが、うっかりして前記の文書 (`stl.pdf`) に説明を追加し忘れていました。これらの使用法は以下の文書（の 8 ページ）に記載されています。

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/ch.pdf#page=8>

プログラム `stl_bev_1` についてはきちんとした説明書をまだ書いていません。とりあえずは、これを含む「HVD シリーズ」のプログラム群の概略を記した下記のスライドをご覧ください。

http://www-bl20.spring8.or.jp/~sp8ct/tmp/hvd_120301.pdf#page=11

P.S.

(1) MOLA (Mars Orbiter Laser Altimeter) と LOLA (Lunar Orbiter Laser Altimeter) の観測データをもとにした火星と月の形状モデルの作成は観測点が超多数なので断念しました（そもそも、このような球面上に稠密に分布する観測点から全球の形状モデルを作る必要性もありませんし）。また、「かぐや」の LIDAR による観測データについてはまったく無知なので、それを用いた月の形状モデル作成にもトライしませんでした。

(2) Clementine 探査機の観測データに関しては下記の研究ノートもご覧ください。

https://www.jstage.ist.go.jp/article/geoinformatics1990/6/3/6_3_147/_pdf

Date: Thu, 26 Jun 2014 19:39:37 +0900
 From: Tsukasa NAKANO
 To: Naru Hirata
 Cc: "TSUCHIYAMA, Akira", Kentaro Uesugi, MATSUNO Junya, Tooru Matsumoto,
 Masayuki Uesugi, "NAKASHIMA, Yoshito", 道上達広, Takashi Matsushima,
 Satoshi Okumura, Michihiko Nakamura
 Subject: CH3D の改造

ひらたさま、

GSJ/AIST のなかのです。どうしても良い話かもしれませんが、5/20 と 5/31 の E-mail で紹介した「天体表面の経緯度と地形高度の点データからその天体の形状モデルを簡便に作成するプログラム (== 球面上の点群を三角形に分割するプログラム、ss_ch)」の心臓部である「点群の 3 次元凸包を抽出する C 言語コード (CH3D)」を少し書き換えました。この CH3D は

杉原厚吉
 FORTRAN 計算幾何プログラミング、
 岩波コンピュータサイエンス、岩波書店、1998、402p

に載っていた FORTRAN コード (CHULL3) をぼくが C 言語コードに翻訳 (意識) したのですが、その過程で 3 次元の点の座標値を変換する整数の語長 (ビット数) を可変にしたのにも関わらず、これまではそれを 32 ビット長に固定していました (オリジナルの CHULL3 では 28 ビット長だったので)。今回はそれを CH3D のコンパイル時に下記のように設定できるよう改造しました。

CH3D を 32 ビット CPU 用にコンパイルした場合 : 32 ビット長
 64 ビット CPU : 64 ビット長

注

32 ビット CPU の上で 64 ビット長の整数値を使うことも可能です。ファイル `ss_ch/Makefile` の 15 行目を以下のように書き換えた後にプログラム `ll2ch` を "make" して下さい。

```
$(CC) -DCH3D_I=long -DCH3D_U=size_t error.c ch3d.c ll2ch.c -lm -o ll2ch
→
$(CC) -DCH3D_I="long long" (この後の "-DCH3D_U=..." の記述は変更不要)
```

そして、5/31 の E-mail (該当部分のみを以下に引用しました) で紹介した Magellan 探査機のレーダ高度計の観測データを用いた金星の形状モデル作成処理に 64 ビット長の整数値を使うようにすれば「観測点の消失」を防げることがわかりました。めでたしめでたし。

On Sat, 31 May 2014 17:28:17 +0900 Tsukasa NAKANO wrote:

```
> (2) Magellan ADF を用いた金星の形状モデル作成
> ..
```

- > `adf.csh` の起動法は前回紹介した `stooke.csh` などの場合と同じです。
- > ...
- > この処理の所要時間は数分程度 (Core2Duo SU9300 搭載のノート PC で約 5 分)、
- > 使用するメモリ量は 1 (32 ビット計算機の場合) ~ 2 (64 ビット機上) GB 強です。
- > 最終的には、ディレクトリ `ss_ch/` にテキストファイル `adf.log` と `gzip` 圧縮されて
- > いる `PLY/ZCP` 形式ファイル `adf.zcp.gz` ができるはずです。
- > `adf.csh` は `adf.log` に
- > [1] Magellan が観測した金星の経度・緯度・半径のデータ点の総数
- > [2] `ll2ch` で検出した重複していた観測点の点データ (経度・緯度・半径)
- > [3] 金星の形状モデルの多面体を構成する頂点の総数
- > の 3 種類の行を書き込む設定にしていますが、実際には [2] に相当する行は
- > ありませんでした (つまり、Magellan のレーザー高度計のデータには重複して
- > いる観測点はない)。しかし、[1] は 2724487、[3] は 2724486 だったので、
- > 今回の処理の過程で 1 個の観測点のデータが消失してしまったようです。これは
- > 凸包抽出の処理で 32 ビット長整数に変換した座標値を使っているために生じた
- > エラーだと思われます (消失した観測点は凸包抽出の際に単位球の内部にあると
- > 見なされた?)。
- > ...

なお、64 ビット CPU の上で点の座標値を 64 ビット長整数に変換して処理を行うと 32 ビット長整数にした場合の 1.5~2 倍の計算時間を要します (とは言え、オリジナルの FORTRAN コードの `CHULL3` を使った場合の数倍は高速ですが)。また、これは以前から設定済みですが、点の個数の数え上げなどに用いる整数値はすべて座標値の場合と同様に使用する CPU のビット数に応じた語長のものを使うように設定してあります。とり急ぎ、

P.S.

新しい `CH3D` のコードは以前と同じ書庫ファイル

http://www-bl20.spring8.or.jp/~sp8ct/tmp/ss_ch.taz

に入れてあります。