The New Horizons Geometry Visualizer: Planning the Encounter with Pluto

Dr. Henry Throop
Sr. Research Scientist
Southwest Research Institute
Boulder, CO
throop@boulder.swri.edu

IDL User Group
October 16, 2008
LASP, Boulder, CO
NHGV: New Horizons Geometry Visualizer

- The New Horizons Mission to Pluto
- NHGV: Brief demo and overview
- Developing NHGV in IDL

http://soc.boulder.swri.edu/gv
New Horizons Spacecraft

- **LORRI (Visible Imager)**
- **PEPSSI**
- **SWAP (Solar Wind)**
- **REX (Radio Experiment)**
- **ALICE (UV Spectrometer)**
- **Ralph (Visible/IR Imager/Spectrometer)**

$700M NASA-funded mission
Lead institution: SwRI
2006 launch; 2015 Pluto
<table>
<thead>
<tr>
<th></th>
<th>New Horizons</th>
<th>iPhone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>1 MP</td>
<td>2 MP</td>
</tr>
<tr>
<td>Storage</td>
<td>16 GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>Network</td>
<td>NASA Deep Space Network</td>
<td>3G</td>
</tr>
<tr>
<td>Battery</td>
<td>88 years, Plutonium-238</td>
<td>4 hour</td>
</tr>
</tbody>
</table>
What does GV do?

GV is a virtual planetarium of the Solar System, as seen from a moving spacecraft.

Answers questions like:

- When can we fit Pluto and its satellites into a single frame?
- What stars are visible when the Sun passes behind Charon?
- What is the sub-Solar longitude on Pluto at closest approach?
- On what day should we approach Pluto so as to see the whole planet?

- Designed as a ‘sandbox’ for planning observations.
- Not used to program commands into spacecraft.
GV Design Goals

Priority 1:
• Accuracy

Priority 2:
• Features
• Ease of use
• Speed
• Ease of development
• Ease of distribution
<table>
<thead>
<tr>
<th>Existing Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STK</strong></td>
</tr>
<tr>
<td>Proprietary; Expensive ($70K/seat); Overkill for most users</td>
</tr>
<tr>
<td><strong>JPL tools</strong></td>
</tr>
<tr>
<td>Minimal support; Some Solaris only</td>
</tr>
<tr>
<td><strong>Other tools</strong></td>
</tr>
<tr>
<td>Not SPICE-based</td>
</tr>
<tr>
<td>Not customizable</td>
</tr>
</tbody>
</table>
GV is Web-Based

- Works on all hardware platforms
- Works from anywhere in the world
- All users are guaranteed to be using up-to-date version.
- All support files (kernel files, documentation, etc) are kept in sync.
  - There are hundreds of frequently updated kernel files, and user-maintained local repositories are prone to be out-of-sync!
Tour of GV
Realistic planet surface models

2015 JUL 14 11:46:57

- $M_v = 7.6 \pm 0.9$ - Solar System Body
- $M_v = 8.3 \pm 7.6$ - FOV boresight
- $M_v = 5 \pm 6.3$
- $M_v < 5$
Star catalogs
New Horizons

Motion across the sky

Saturn RA=127.6559, Dec=18.210738, r=18 urad, d=22 AU, 2012 JAN 01 00:00:00
Motion across the sky
Mosaics and uncertainty ellipses
Web Interface

- ION (IDL-on-Net) is a discontinued product, so we can’t use it!
- Instead, we use regular IDL, on a Linux Apache server.
  - PHP code draws web page, waits for user input. HTML, not widgets!
  - IDL starts up, runs, creates output files, and exits.
  - PHP reads these files and displays to screen.
- Requires starting up IDL at every web call, but more robust than other options.
<!--
// Start of Yellow input table
// Adjust border/cellpadding/cellspsacing here for the entire GV input table.
-->

<form method="get" action="gv.php">
<table style="text-align: left;">
    <tbody>
        <tr align="center">
            <td colspan="2" align="center">
                <input name="submit" value=" Plot " type="submit">
            </td>
        </tr>
        <tr>
            <td>
                <a href="gv_info.php#observer" target="_blank">
                    Observer
                </a>
            </td>
            <td>
                <p><?php
                    // Choose the observer
                    //
                    $file_observerlist = 'gv_observerlist.php';
                    if ($gv_version == 'LRO'){
                        $file_observerlist = 'gv_observerlist_lro.php';
                    } if ($gv_version == 'TGE'){
                        $file_observerlist = 'gv_observerlist_tge.php';
                    } if ($gv_version == 'Juno'){
                        $file_observerlist = 'gv_observerlist_juno.php';
                    }
                    include($file_observerlist);
                    echo " <select name=name_observer> ";
                    // Now loop over each observer, and create an entry
                    foreach ( $observerlist as $index => $observer ){
                        echo " <option ";
                        if ($name_observer == $observer){echo " selected ";}
                        "gv.php" 1935L, 60529C written
                ?></p>
            </td>
        </tr>
    </tbody>
</table>
</form>
<?--
// Start of yellow input table
// Adjust border/cellpadding/cellsapcing here for the entire GV input table.
-->

<table style="text-align: left;"
    border="1" cellpadding="1" cellspacing="0"
    class="input_table">

    <tbody>
    <tr>
      <td colspam="2 align=center">
        <input name="submit" value="Plot" type="submit">
      </td>
    </tr>
    <tr>
      <td>
        <a href="gv_info.php#observer" target="_blank">
          Observer
        </a>
      </td>
    </tr>
    </tbody>

<?php
//
// Choose the observer
//

$file_observerlist = 'gv_observerlist.php';
if ($gv_version == 'LRO'){
  $file_observerlist = 'gv_observerlist_lro.php';
}
if ($gv_version == 'TGE'){
  $file_observerlist = 'gv_observerlist_tge.php';
}
if ($gv_version == 'Juno'){
  $file Observerlist = 'gv Observerlist_juno.php';
}
include($file_observerlist);

echo "<select name=observer> ";

// Now loop over each observer, and create an entry

foreach( $observerlist as $index => $observer){
  echo " _option ";
  if ($name_observer == $observer){echo " selected ";}
  "gv.php" 1935L, 60529C written
  }";
PHP is interface between user and IDL

```php
// Call IDL to create all the frames
$command = '"gv_make_movie_frames.csh" . $session_id;
$command = 'csh -c "' . $command . '"';
exec($command);

// Create the tarball
$command = '"cd /home/throop/nh/gv/tmp/movies/; tar -cvf "' . $session_id . '"gv_frames.tar "' . $session_id . '/frame_*'.png";"
// echo $command . '"<brautor><brautor>
exec($command);

// Encode the AVI file
// Plays under VLC, MPlayer, but not Quicktime.
$command = '"cd /home/throop/nh/gv/tmp/movies/" . $session_id . ';" .
'/home/throop/bin/mencoder "mf://frame_*'.png" -mf fps=1 . $fps .'
-o gv_movie.avi -ovc lavc -lavcopts vbitrate=10000';"
exec($command);
```
csh scripts: Invoke IDL and log output

```csh
# Set variables for 'scat'. 'scat' starts a new shell when it is spawned, so
# these variables work properly
setenv HDCCUL /home/throop/occult/hdoccu1
setenv WCS_CATDIR $HDCCUL/catalogs
setenv TY2_PATH /home/throop/occult/catalogs/TYCH02
setenv UB1_PATH http://tdc-www.harvard.edu/cgi-bin/scat

setenv PATH /usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/throop/bin

set hdoccu1 = /home/throop/occult/hdoccu1
set wcs_catdir = /home/throop/occult/hdoccu1/catalogs

# Set path. Make sure 'scat' is on this.
set path = (/usr/kerberos/bin /usr/local/bin /bin /usr/bin /usr/X11R6/bin /home/throop/bin)

# Change to the proper directory
cd $GV

# Run IDL, and use the 'script' command to log all of the NHGV output printed to terminal.
# This allows us to check for the IDL error state. There is a small
# possibility we'll conflict with another user here, but since errors are pretty
# quick to happen, it's unlikely.

script -c "/usr/local/bin/idl $GV/gv.bat" $GV/idl_gv.log

# Add a line to the logfile.
# Logfile is usually updated by PHP script, and not in batch mode -- therefore,
# if we are in batch mode, we update the logfile right here.

if (`whoami` != 'apache') echo `date +%a, %d \ %b \ %Y \ %H:\%M:\%S \ %z` Batch[]whoami << $GV/logs/log_nhgv.txt
```
Flowchart of GV System

GV Web Entry Point

- Initialize settings
- Success?
  - Yes: Render HTML web page with current settings, Display image, tables, etc
  - No: Error message

'Plot' button pressed
- Create parameter file based on form settings
- Call gv.csh

Parameter File

PHP

Invoke IDL
- Log all output

CSH

Process input file
- Create output files

IDL

Output files: Image, Tables, Movie

IDL log file
SPICE *
(*) SPacecraft, Instrument, Camera kErnels

- SPICE is a spacecraft geometry toolset developed and maintained by NASA-JPL.
- Highly accurate astrometric / geometric computations. Hundreds of functions.
- Distributed for C, FORTRAN, IDL.
- SPICE is the standard for interplanetary spacecraft navigation.
- ‘Kernel’ files define trajectories, ephemerides, FOVs, leap-seconds, planet masses, etc.
- Using SPICE assures that everyone gets the same results!

http://naif.jpl.nasa.gov
ICY is the name for the IDL version of SPICE library

ICY APIs for accessing SPICE kernel data

- Loading and unloading SPICE kernels
- Converting between UTC and Ephemeris Time (LSK)
- Converting between Ephemeris Time and spacecraft clock (SCLK)
- Retrieving constants and orientation for natural bodies (PCK)
- Computing transformations between reference frames (FK)
- Computing positions of spacecraft and natural bodies (SPK)
- Computing orientations of spacecraft and instruments (CK)
- Retrieving instrument parameters (IK)
- Mapping between object names and NAIF IDs

ICY APIs for computing derived geometry

- Computing planetocentric, planetodetic, and planetographic coordinates
- Computing surface intercept point
- Computing sub-observer and sub-solar points
- Computing illumination angles
- Computing and propagating orbital elements

ICY APIs for coordinate conversions

- Converting from and to rectangular coordinates
- Converting from and to spherical coordinates
- Converting from and to cylindrical coordinates
- Converting from and to latitudinal coordinates
- Converting from and to R, RA, DEC
- Converting from and to geodetic coordinates

ICY APIs for operations with 3D vectors and matrices

- Performing simple operations on 3D vectors
- Performing simple operations on 3x3 matrices
- Projecting, combining and rotating 3D vectors
- Creating and converting transformation matrices
A few sample SPICE routines...

- **NCPOS C** - NOT Character position
- **NCPOSR C** - Character position, reverse
- **NEARPT C** - Nearest point on an ellipsoid
- **NPEDLN C** - Nearest point on ellipsoid to line
- **NPELP T C** - Nearest point on ellipse to point
- **NPLNPT C** - Nearest point on line to point
- **NVC2PL C** - Normal vector and constant to plane
- **MVP2PL C** - Normal vector and point to plane

O

- **ORDC C** - The ordinal position of an element in a set
- **ORDD C** - The ordinal position of an element in a set
- **ORDERC C** - Order of a character array
- **ORDERD C** - Order of a double precision array
- **ORDERI C** - Order of an integer array
- **ORDI C** - The ordinal position of an element in a set
- **OSCEL T C** - Determine conic elements from state

P

- **PCKCOV C** - PCK coverage
- **PCKFRM C** - PCK reference frame class ID set
- **PCKLDF C** - PCK Kernel, Load binary file
- **PCKUOF C** - PCK Kernel, Unload binary file
- **PCPOOL C** - Put character strings into the kernel pool
- **PDPOOL C** - Put d.p.'s into the kernel pool
- **PGRREC C** - Planetographic to rectangular
- **PI C** - Value of pi
- **PIPOOL C** - Put integers into the kernel pool
- **PJELPL C** - Project ellipse onto plane
- **PL2NVC C** - Plane to normal vector and constant
- **PL2NVP C** - Plane to normal vector and point
- **PL2PSV C** - Plane to point and spanning vectors
GV Parameter file

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTC_START</td>
<td>'2015 Jul 14 11:46:57'</td>
</tr>
<tr>
<td>UTC_END</td>
<td>''</td>
</tr>
<tr>
<td>NUM_DT</td>
<td>'1'</td>
</tr>
<tr>
<td>UNITS_DT</td>
<td>'Timesteps'</td>
</tr>
<tr>
<td>NAME_TARGET</td>
<td>'Pluto'</td>
</tr>
<tr>
<td>UNITS_RADEC</td>
<td>'deg'</td>
</tr>
<tr>
<td>RA_LON_CENTER</td>
<td>'188.04019'</td>
</tr>
<tr>
<td>DEC_LAT_CENTER</td>
<td>'25.68787'</td>
</tr>
<tr>
<td>RA_LON_FOY</td>
<td>'0.0'</td>
</tr>
<tr>
<td>DEC_LAT_FOY</td>
<td>'0.0'</td>
</tr>
<tr>
<td>UNITS_POS_FOY</td>
<td>'Degrees from Target'</td>
</tr>
<tr>
<td>NUM_FOOTPRINTS_X</td>
<td>''</td>
</tr>
<tr>
<td>NUM_FOOTPRINTS_Y</td>
<td>''</td>
</tr>
<tr>
<td>DX_FOOTPRINT</td>
<td>''</td>
</tr>
<tr>
<td>DY_FOOTPRINT</td>
<td>''</td>
</tr>
<tr>
<td>DT_FOOTPRINT</td>
<td>''</td>
</tr>
<tr>
<td>FRAME</td>
<td>'J2000'</td>
</tr>
<tr>
<td>RADIUS</td>
<td>90</td>
</tr>
<tr>
<td>UNITS_RADIUS</td>
<td>'Degrees'</td>
</tr>
<tr>
<td>CENTER_FOV</td>
<td>'Star Trackers'</td>
</tr>
<tr>
<td>FOV_ROTATE</td>
<td>0.000</td>
</tr>
<tr>
<td>VEC_ROTATE_EXT</td>
<td>'Orbit Normal'</td>
</tr>
<tr>
<td>DT_DOWNTRACK</td>
<td>0</td>
</tr>
<tr>
<td>DO_PLOT_E1</td>
<td>0</td>
</tr>
<tr>
<td>DO_PLOT_E2</td>
<td>0</td>
</tr>
<tr>
<td>DO_PLOT_S1</td>
<td>1</td>
</tr>
<tr>
<td>DO_PLOT_EPOCAM1</td>
<td>1</td>
</tr>
<tr>
<td>DO_PLOT_PLANETES</td>
<td>1</td>
</tr>
<tr>
<td>SURFACE_STYLE</td>
<td>'Albedo'</td>
</tr>
<tr>
<td>PROJECTION</td>
<td>'Rectangular'</td>
</tr>
</tbody>
</table>

Values entered by user.
File generated by PHP.
Read and processed by IDL:
Compiled on-the-fly and executed as IDL code.
Map Projections

- Cartesian - for sky
- Stereographic - for sky
- Stereographic - for wrapping surfaces
- IAU vs IDL systems
  - Most go 0 .. 360, not 180 W .. 180 E
  - Some have N pole in different directions
- Spherical bodies only
Longitude systems

- IAU vs IDL longitude systems
- Most go 0..360, not 180 W, 180 E
- Some have N pole in different directions
- Spherical bodies only
Cartesian Projection
Stereographic projection
Stereographic projection
Multiple stereographic projections
Things that Worked

• SPICE
  • Robust and powerful; forms the basis of all of GV’s computations.

• Calling IDL from Web, using PHP interface

• Unix shell utilities and external functions
  • File management
  • Star catalog searches
  • Movie encoding

• IDL: Rapid development, etc.
Things that didn’t work as easily...

- Direct Graphics: should use Object Graphics!
- Higher quality
- Much faster 3D rendering than direct graphics.
- More flexible text labels.
- Direct graphics has bugs, quirks.
- Polyfill works differently on Mac, Unix
- \([1,1]\) pixel works differently on X, Zbuffer
- Clipping works differently on X, Zbuffer
- IDL’s map routines support Earth’s longitude system but not other planets’
- IDLDE (Workbench) is nice but no easy integration with VIM editor
Onward...

http://soc.boulder.swri.edu/gv