VRMapper Requirements Design

Workflow, GUI Design and Use Cases

VRMapper is intended to be a fast and easy to use yet robust photographic surveying, mapping and visualization system for the hobbyist, engineer, surveyor, law enforcement officer, resource or GIS specialist and a myriad of other government and industry users. The basic vision is to enable a user to photograph and immersively document an indoor or outdoor scene, make critical measurements onsite (or latently), visually map or measure objects of interest from photographs and then easily create immersive presentations that can be simply shared with colleagues or clients.

VRMapper Core Functions

Hardware Devices

- devices to easily capture birds-eye aerial panorama's outdoors and normal panorama's indoors
  - VRPhotopod (25 foot with leveling and direction device – optional tripod)
  - Universal Camera Platform for VRPhotopod which also works on a normal camera tripod
  - Stand alone laser ranging platform which also integrates with the camera platform (measures distances to millimetres and slope angles to 6 minutes)
  - 4 foot tall sectional pole target and a cone target

VRMapper Software

- VRMapper viewer to enable immersive viewing (pan, zoom, FOV, direction, etc) of normal, partial panorama's and full 360 degree panorama's
- tools to visually **Measure** and **Map** point, line and polyline objects directly from the panorama's
- dialoque tools to input photo survey measurements (coordinates, distances, elevations and directions (manual & from panorama's))
- map display of photo survey network with hotspot pop-up window for each panorama. The direction of the panorama view is dynamically shown on the map display
- view photo EXIF information if available within image file
- tool to remap full-frame or circular fisheye image to equirectangular image for direct visual object mapping and measurement
- 2D (horizontal) and 1D (vertical) least square adjustment of photo survey observations with output reports and data edit functions (licencse existing libraries and wrap it within uDig – create and run input file and display adjustment result report
- full functionality of the uDig framework including GUI, GIS tools, WMS/WFS discovery portal, create print to PDF
- text and drawing tools to mark-up photo images and to make scene drawings
- export map, immersive panorama viewer and text box template to standalone presentation or to Google Earth and Virtual Earth/Worldwind account

This document outlines one option for developing the VRMapper software functions using the uDig framework and GUI.

The design proposes to develop a modified uDig GUI to immersively view and mark-up photos, partial and full 360 degree panorama's and to leverage a unique license to use certain libraries of a least square adjustment program to perform an adjustment of survey observations. uDig will be retained in its entirety for true spatial mapping, editing and viewing functions.

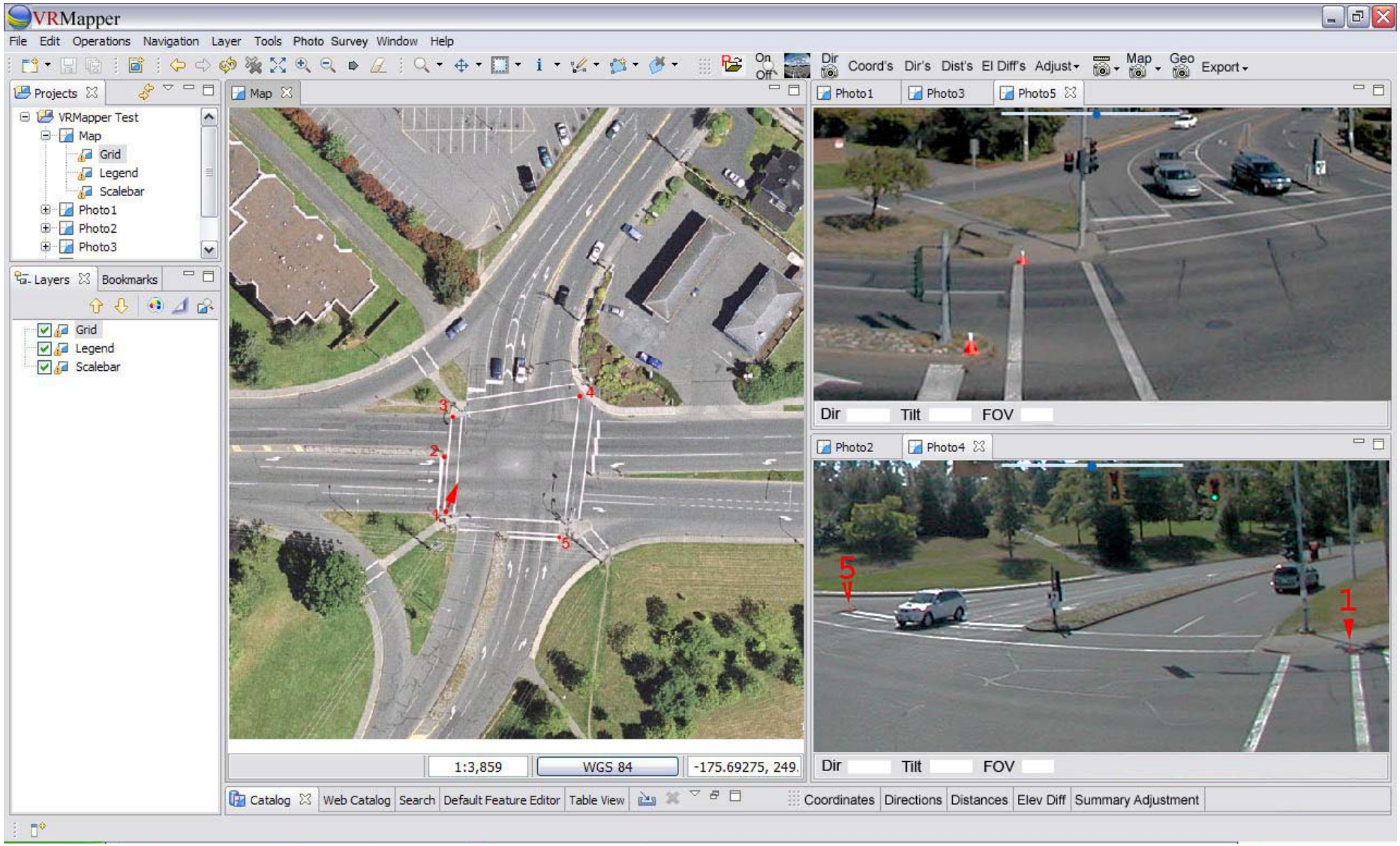
The photo survey functions enables the user to input coordinate, distance, elevation difference and direction observations and perform a 2D horizontal and 1D vertical network adjustment. The adjustment provides geocoding for the photographs and true directions for the image. A user will also be able to simply open photo's and input a bearing for the centre view of the image, input coordinates of the photo or manually place a location tag on the map views in order to geocode the photo. The photo survey also enables a user to make point and line measurements (including elevation differences) of visible objects from photos that have been adjusted in a network or from two photos that have the distance measured between the two photo locations. The interface will be enhanced to enable a user to import Worldwind map tiles, raster/orthoimages, templates or to create drawings/sketches and to visually sketch points, lines and polylines at true scale from adjusted panorama photo networks and integrate iTest code (seek mozilla or LGPL license) in uDig to enable export as a PDF file.

A unique feature of VRMapper is the ability to display photos and panoramas in an immersive viewer. The viewer will be able to pan, zoom in and out, show view direction, tilt, field of view, brightness control and can be marked up. The geocoded photos can be activated to pop-up in an immersive viewer so that the photo view direction is shown on the map. This creates a very powerful visual site document system when using the unique birds-eye aerial photo panorama views outdoors or for room layouts with immersive indoor panorama views. Another part of the visualization is to be able to export to a Google Earth and/or Worldwind account.

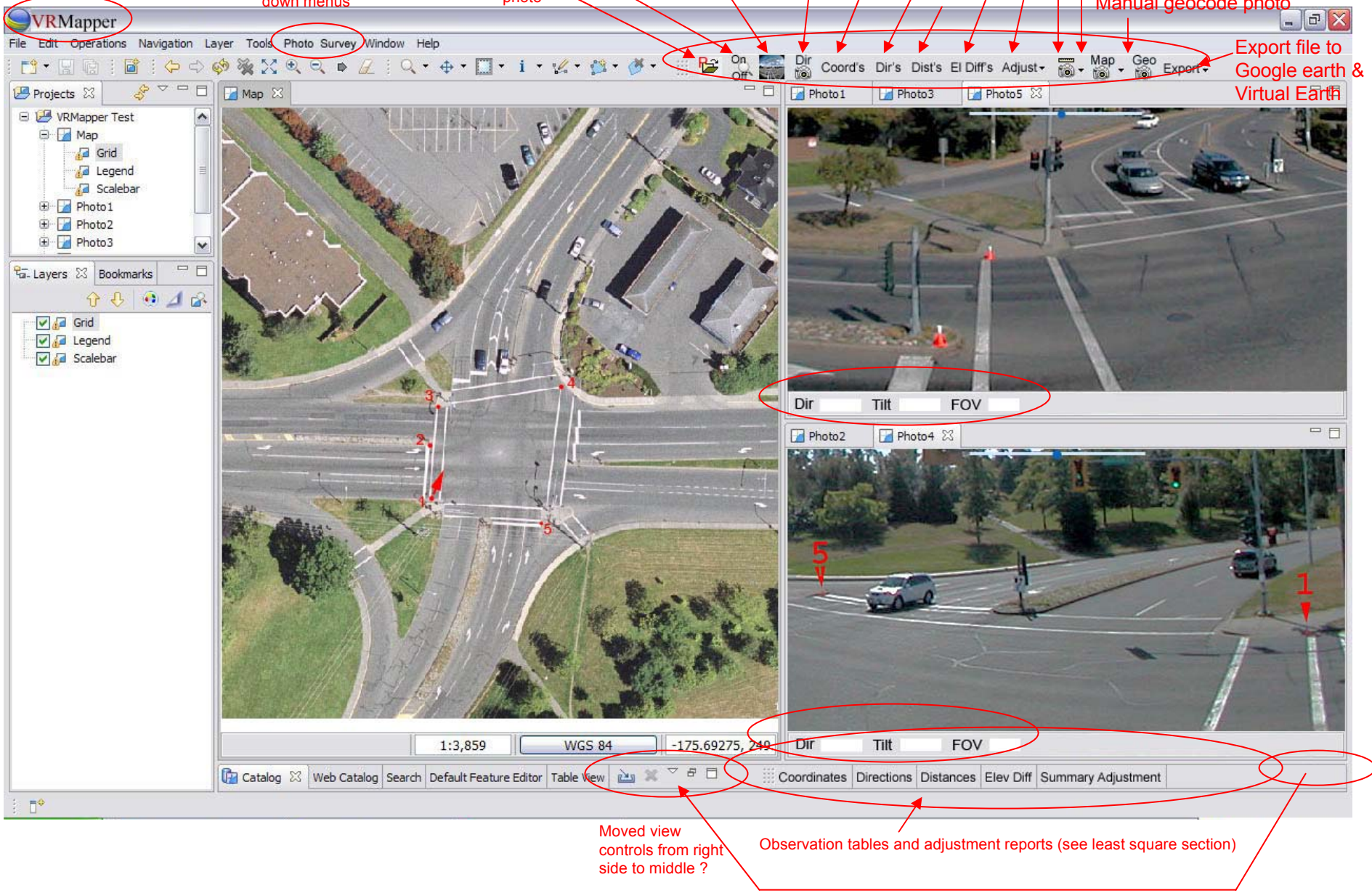
To do's:

- uDig is starting to build an active community and styling/editing and grid coverage is slated for enhancement in the next version. Investigate specific planned enhancements and bridge critical gaps as necessary. Investigate whether symbol brewer will be integrated into uDig, and if not, look at obtaining a license to implement this codein uDig. Ensure drawing/editing palette will meet user needs.
- Research common templates required for forensic and accident scene mapping
- Finalize critical design elements and determine strategy for implementing VRMapper Photo viewer – inside VRMapper only, or plus standalone, or plus java web version. Complete research on viewer modeling and reprojection from equirectangular to rectilinear (gnomonic) on the fly in the viewer
- Finalize research and modeling for lens correction and remapping full-frame or circular fisheye image to equirectangular

True Look Mock-up of VRMapper GUI



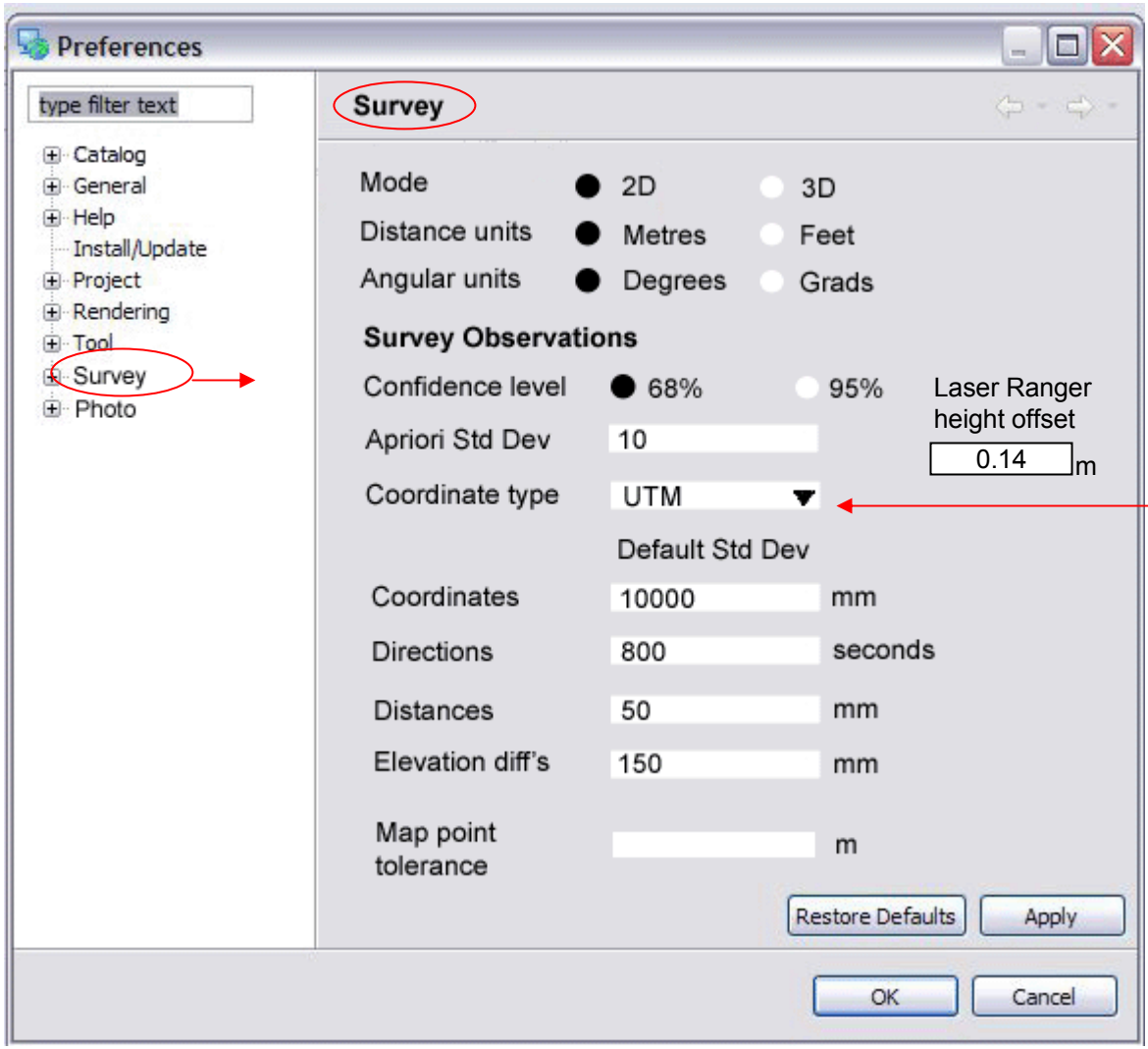
Summary of Changes to uDig GUI



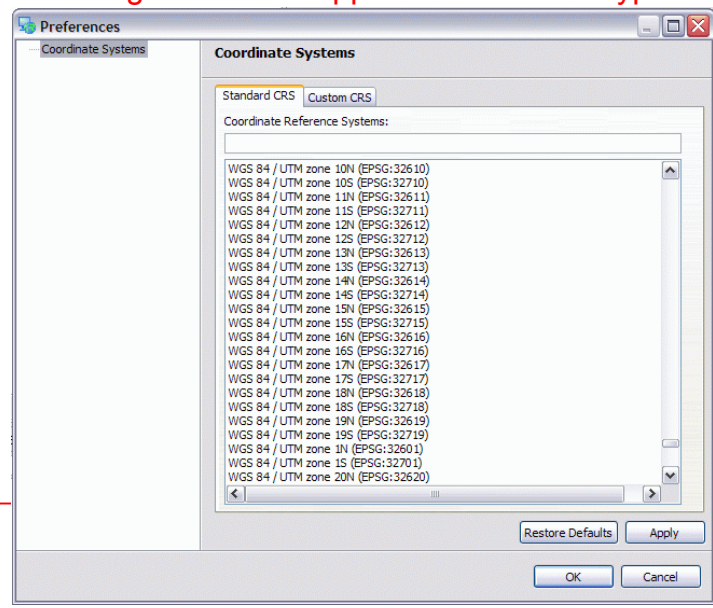
Preferences

Two new preference tabs are to be added – Survey and Photo

Survey preferences



uDig/ Geotools supported coordinate types



Note: In order for users to make sketches not in a true map projection but in a local coordinate system, we may want to implement two new coordinate systems:

-Local imperial (feet)
-Local metric

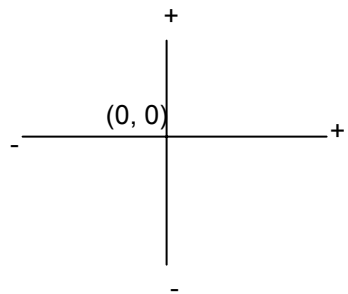
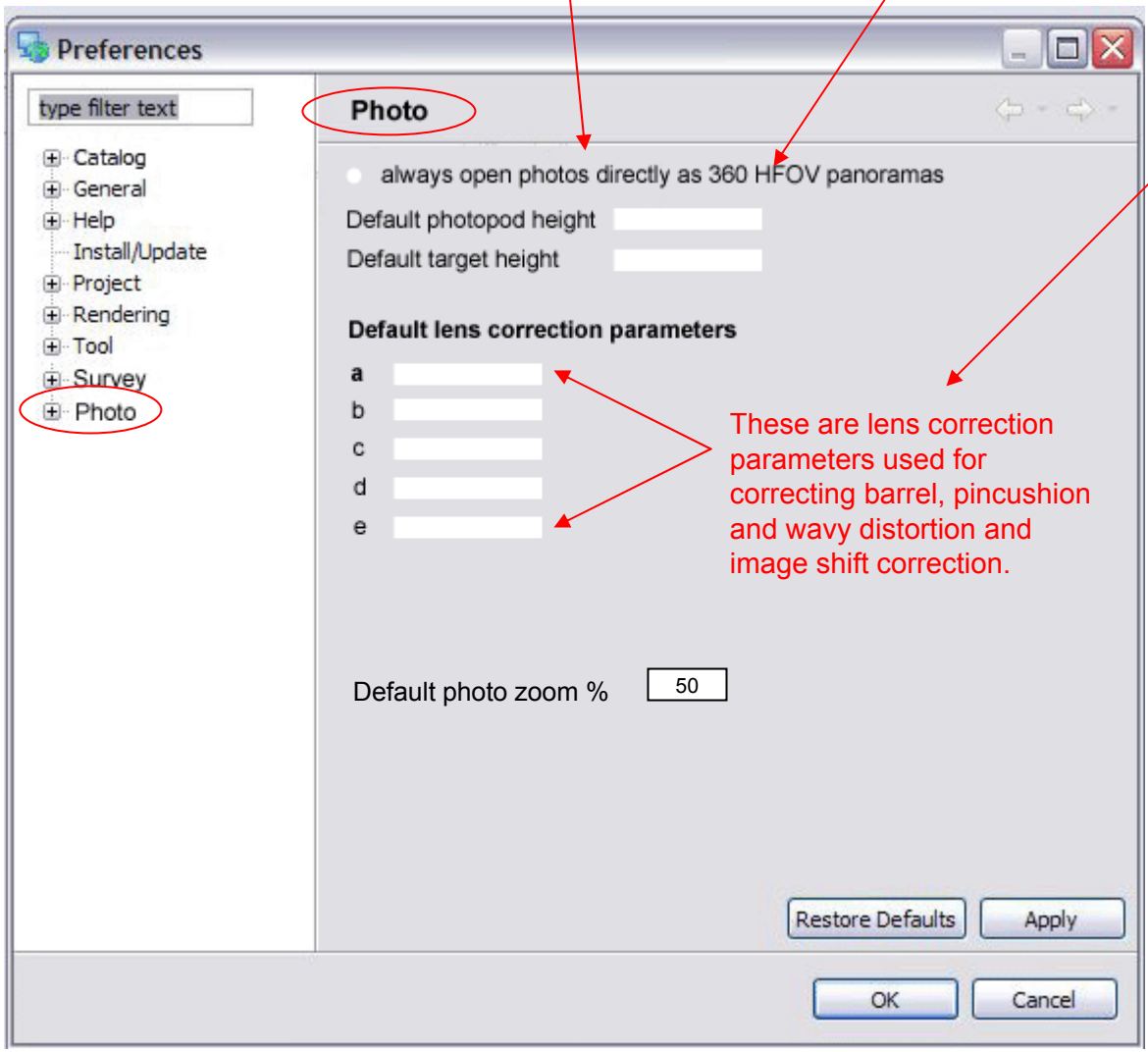


Photo preferences

May also want to have a preference of "always open photo as rectilinear photo"

Assigns the filename to be the photo point number.



These are lens correction parameters used for correcting barrel, pincushion and wavy distortion and image shift correction.

Remap function uses the lens correction parameters to correct for lens distortion and to then remap the image to equirectangular projection to be used for mapping visible objects. These lens parameters are derived from Helmut Dersch's.PanoTools code. Here is the link to the lens correction model: http://www.panotools.info/mediawiki/index.php?title=Lens_correction_model

We may just want to leave this function out of VRMapper and let PTLens or any of the stitching programs do it when the image is processed. We can look at adding this function in a later release if it is not too much work. Would likely have to look at PanoTools code to see how to create this tool.

If wizard is not envoked in preferences then VRMapper will launch by opening a blank map in a view as shown and user opens photos as desired.

Remap details shown in appendix. Image is corrected for a,b & c lens distortion as is done in Panotools

Magnifier on/off similar to the one in PTGui

Details on survey observation input dialogue's shown in least square section

Measure point from
2 photos
3 photos

Measure line from
2 photos
3 photos

Map point from
2 photos
3 photos

Map line from
2 photos
3 photos

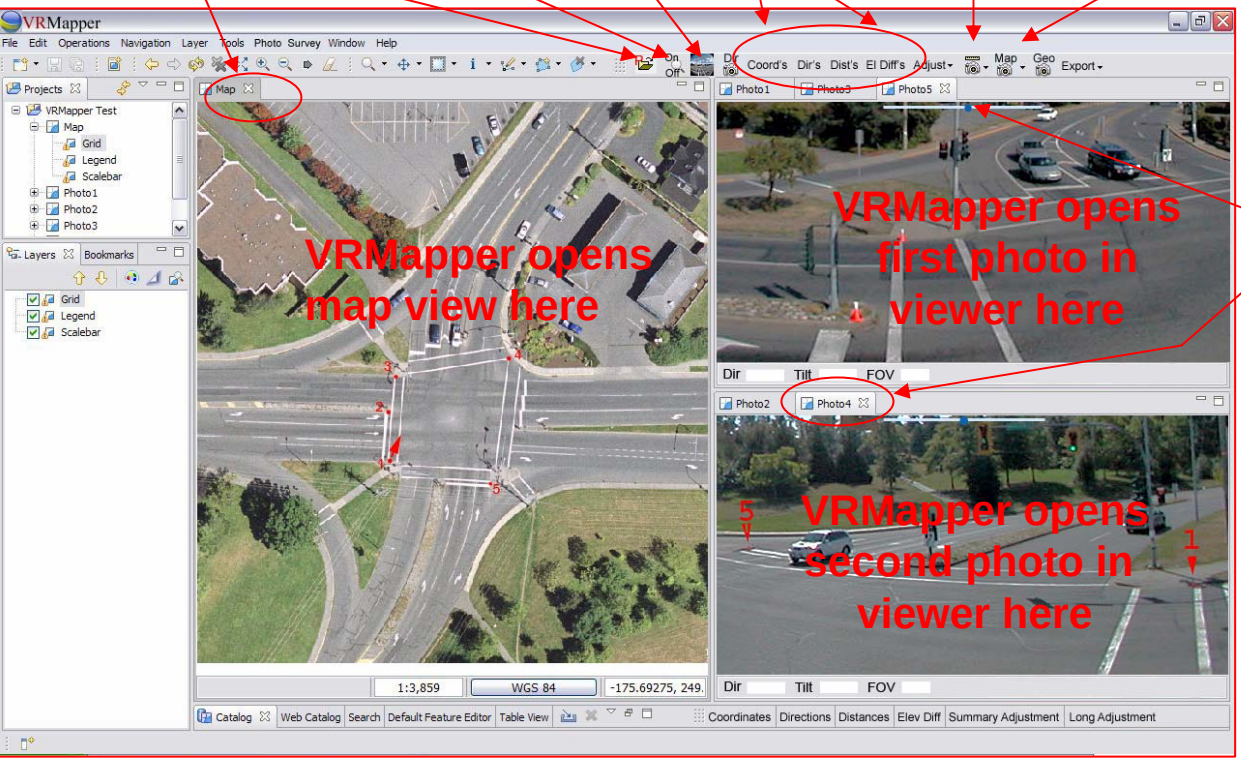
Export

Cretae PDF of selected
Photo
Map
Map & Photos

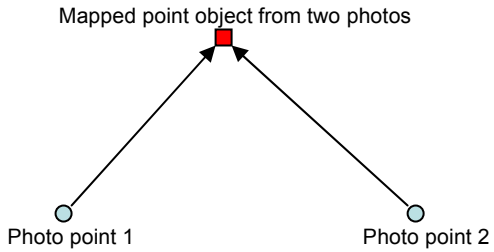
Create Google Earth KMZ file of selected
Photo
Map
Map & photos

Create Virtual Earth file of selected
Photo
Map
Map & photos

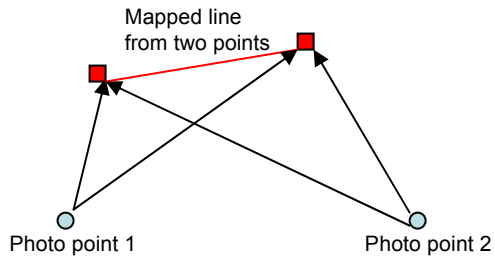
Standalone map & photos



Photo's 3 and up are opened in the view as shown



Calculates point or line by intersecting from two or more of the photo traverse points. Using two photos there is no redundancy (one solution). Using three supplies multiple position solutions. Suggest using least square engine to do measure function by seeding input and running adjustment and then displaying result to user. See diagrams.



Map

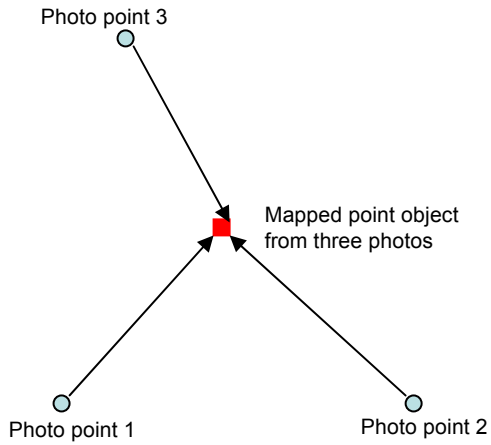
Same function as measure, except in this mode the point or line is mapped to the map window.

Measure & Map Tool Processing Model

The user selects measure or map point or line from two or three points. The user then mouse clicks the points.

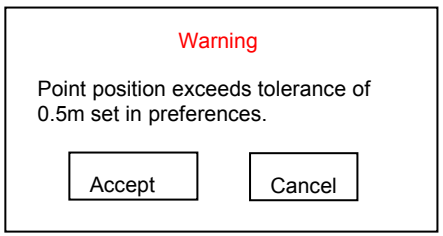
Example measure point from two photos:

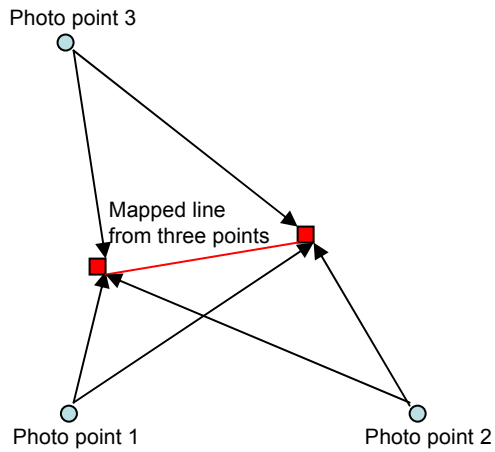
User has two or more equirectangular photos open with known coordinates. User mouse clicks point in photo 1 and then mouse clicks the point in Photo 2. VRMapper then determines the direction to the point from each photo and then runs the least square engine and outputs the position solution. Any adjusted or known photo points visually show in the map window on a map layer. Could have VRMapper temporarily show the direction lines to the mapped point in the map view.



Example map line from two photos:

User has two or more equirectangular photos open with known coordinates. User mouse clicks point 1 and 2 (each end of the line) in photo 1 and then mouse clicks the point 1 and 2 in Photo 2. VRMapper then determines the direction to the points from each photo and then runs the least square engine and outputs the position solution. Any adjusted or known photo points visually show in the map window on a map layer. VRMapper draws the line in the map view on the active map layer. Should prompt user if the desired layer is currently open. VRMapper could also show the direction lines to the mapped line in the map view temporarily until position solution accepted. When three photos are used, the threshold for the mean positional error for the position solution of a mapped point will determine whether a warning message will appear - otherwise the point or line is simply mapped to the map view layer. For example, if a point is mapped from three photos and the position error set in preferences is exceeded then a warning message is displayed.





Same process for mapping or measuring from three photos, except there are more observations processed in the least square engine

VRMapper Measure Tool Output

Output for Point case (from two photos 2D mode)
Simply display adjustment coordinates for point

Point position	
Northing	Easting
(value)	(value)

Output for Point case (from three photos 2D mode)

Simply display adjustment coordinates for point and average positional error

Point position		Mean Position error
Northing	Easting	
(value)	(value)	(value)

Output for Line case (from two photos 2D mode)
Simply display adjustment coordinates for point

Northing 1	Easting 1			
(value)	(value)	Bearing 1 to 2	Distance 1 to 2	Elevation Diff 1 to 2
(value)	(value)	(value)	(value)	(value)
Northing 2	Easting 2			

Elevation difference can be determined in 2D mode and absolute elevations can be determined in 3D mode

Output for Line case (from three photos 2D mode)
Simply display adjustment coordinates for point

Mean Position Error	Northing 1	Easting 1				
(value)	(value)	(value)	Bearing 1 to 2	Distance 1 to 2	Mean Elev Diff 1 to 2	Std Dev
(value)	(value)	(value)	(value)	(value)	(value)	(value)
	Northing 2	Easting 2				

Survey

Input coord's
Input distance
Input directions
Input directions & distances
Input Elevation
Input elevation diff
Photo Adjustment
Manual 1D adjustment
Manual 2D adjustment
Adjustment Summary

Coordinate transform…
Traverse
Plan check

Brg-Brg intersect
Brg-Dist intersect
Dist-Dist intersect

These are the standard survey input observations. See least square adjustment section for input dialogues and conditions

Runs horizontal and vertical adjustment (if in 3D mode)

Runs vertical adjustment

Runs horizontal adjustment

Displays short or long adjustment output listing (see least square adjustment section for details)

These survey functions can be added in a future release

Photo

Open Photo
Remap photo
Close all photos
Manual geocode photo on map
Show EXIF info

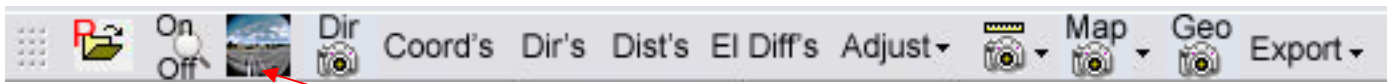
Measure point
from 2 photos
from 3 photos
Measure line
from 2 photos
from 3 photos

This is an important function to add fairly early on as it allows a user to remap a fisheye image to an equirectangular image which can be used directly for mapping and measuring. This would empower a user to remap a fisheye image with the click of a button to be usable for mapping/measuring objects without using any stitching software

Reads EXIF (if one exists) file of selected photo and presents a summary to the user. Wikipedia info: http://en.wikipedia.org/wiki/Exif#Applications_for_Displaying_Exif
Here is a free java library we will likely want to use:
http://www.sno.phy.queensu.ca/~phil/exiftool/

This function uses the least square engine to calculate coord's and elevation of a single point from two or more photo's

This function uses the least square engine to calculate coord's and elevation of two points (a line), bearing and distance and elevation difference



Open photo
Details shown in
Folllowing slide

Magnifier glass (loupe tool) can be toggled on or off and is modeled similar to PTGui's

Remap tool: the easiest way to implement this tool will be to review the Panotools code and determine the math functions/process to convert from a fullframe and circular fisheye format to equirectangular format. There are two steps to this process, first correct for lens distortion and then remap from fisheye to equirectangular. I have separate documents for code to do these functions by professor Dersch and Ken Turkowski (defish).

PanoramaTools by Helmut Dersch **Correct function** consists of several tools: - Radial shifts all pixel positions radially as specified by a third order polynomial. The coefficients are entered in the options submenu for each individual color. This enables correction of barrel and pincussion distortions as well as many chromatic errors. d=1 and c=b=a=0 leaves all positions as they are. It may also be used to correct the radial mapping of fisheye-lenses, which often don't follow the theoretical angle dependence (see below). Once you have determined the optimum coefficients for your lens (either by trial and error, or by some fitting process) you can save them. The radial distance r_src for each point in the source image is computed by the formula: r_src = (a * r_dest 3 + b * r_dest 2 + c * r_dest + d) * r_dest Units for r_src and r_dest is image width / 2, ie the edge of the image corresponds to r_src = 1. If you check the option 'vertical', only the vertical axis is affected by the shift, and the unit is relative to image height/2. This can be used to correct lens distortions in Panoramic cameras. Finally, the option 'horizontal' shifts each pixel horizontally by an amount determined by its vertical distance from the center line (d and e parameters). This can be used to correct deregistration errors in scanning cameras, and will be described in a separate documentation at my website. The five lens correction parameters can be easily derived by running an optimization in Panotools. Once the user has done this initial calibration, then they can use the parameters to directly correct for the lens distortion. The object of VRMapper is to supply a "Remap Tool" that will correct the lens distortion and then remap the image so that the user can directly use it for mapping. This is just for independent convenience for our users. A user could also correct or stitch the equirectangular panoramas in PTGui, PanoTools, etc. There is a phonemenal tool called AutopanPro that will stitch in seconds automatically – fisheye stitching is coming out in their next release – it is beta now.

Survey Observation Tables and Adjustment Reports

Direction observations input table. It would be nice to be directly editable in table.

Directions

From	To	Direction	StDev	Height Pod	Height Target

Elevation difference input table. It would be nice to be directly editable in table.

Elevation differences

From	To	Elevation diff	StDev

Coordinate observations input table. It would be nice to be directly editable in table.

Coordinates

Point	X	Y	StDev	Z	StDev

Distance observations input table. It would be nice to be directly editable in table.

Distances

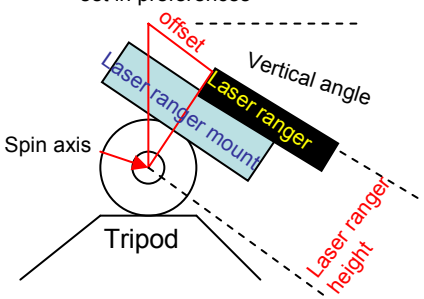
From	To	Hor Distance	StDev	Slope dist	Vertical angle	Offset corr

See least square output section for details. This is a summary of important adjustment results.

Coordinates | Directions | Distances | Elev Diff | Summary Adjustment

Laser Ranger Offset Correction

Note: laser ranger height is set in preferences



Offset = laser ranger height * Tan vertical angle

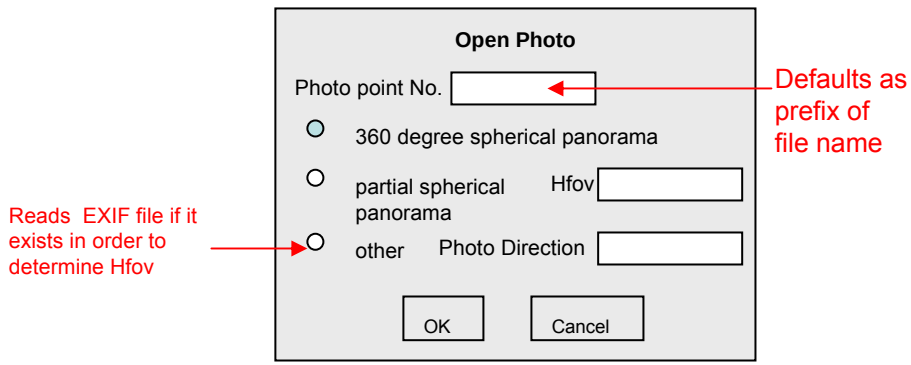
True horizontal distance = (slope distance * cos vertical angle) - offset

True horizontal distance = (slope distance * cos vertical angle) – (laser ranger height * Tan vertical angle)

Note: VRMapper populates a user's input of a slope distance and vertical angle and then calculates the horizontal distance and accounts for the laser ranger offset.

Opening Photos in Photo View

The open photo dialogue enables the user to open a spherical panorama, a partial panorama or a normal rectilinear photo. Preferences can be set to Open 360degree panoramas automatically without the dialogue.



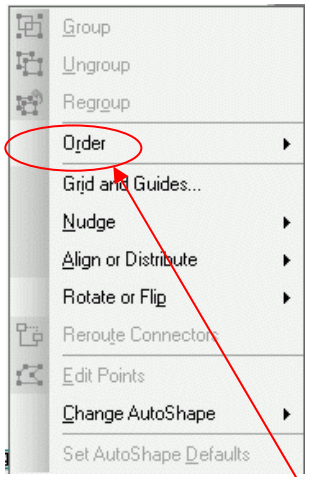
Note: user can set preferences to open 360 Hfov panorama's directly without the above prompt and the file number becomes the photo point number.

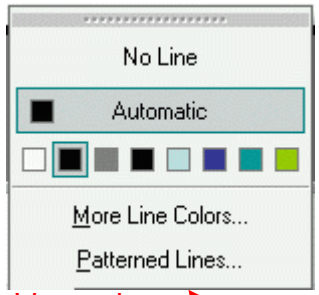
Sample EXIF display

File Name	113_1366.JPG
Camera Model Name	Canon EOS DIGITAL REBEL
Date/Time Original	2003:10:31 15:44:19
Shooting Mode	Program AE
Shutter Speed	1/60
Aperture	5.6
Metering Mode	Evaluative
Exposure Compensation	0
ISO	100
Lens	18.0 - 55.0mm
Focal Length	55.0mm
Image Size	2048x3072
Image Quality	Normal
Flash	On
Flash Type	Built-In Flash
Flash Exposure Compensation	0
Red Eye Reduction	Off
Shutter Curtain Sync	1st-curtain sync
White Balance	Auto
Focus Mode	AI Focus AF
Contrast	+1
Sharpness	+1
Saturation	+1
Color Tone	Normal
File Size	811KB
Image Number	113-1366
Drive Mode	Continuous shooting
Owner's Name	Phil Harvey
Camera Body No.	0560012345

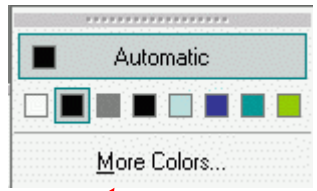
Drawing Palette

We need to develop a drawing/editing strategy that is in sync with uDig. I understand that the next version of uDig will undergo development in this area. We need to do some research and formulate our drawing/editing strategy. The initial drawing/edits can be rudimentary, however, we want to develop a strategy that will easily enable enhancement. I find the Powerpoint drawing palette to be very friendly (see screenshots). We may want to start developing a similar palette for VRMapper. It needs to be easy to use.





Line colour

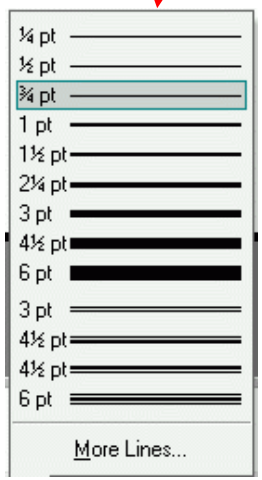


Text colour

We may want to have a pull down for templates. We should also look at integrating symbol brewer.

Shape fill colour

Line style



Line type

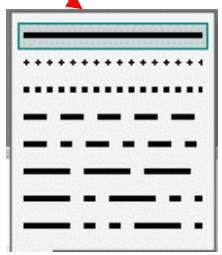
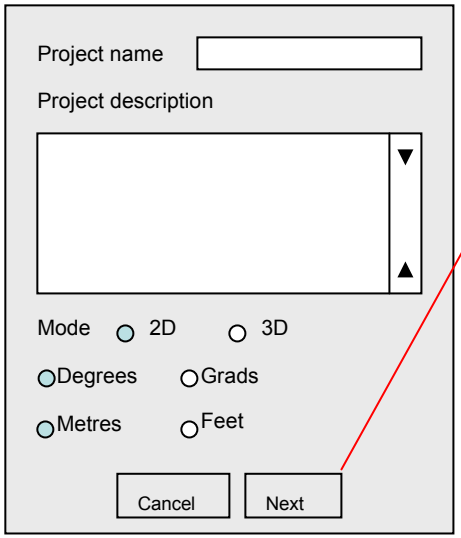
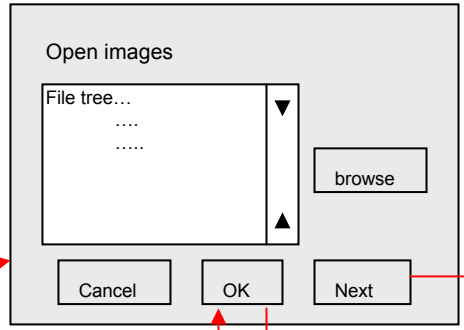


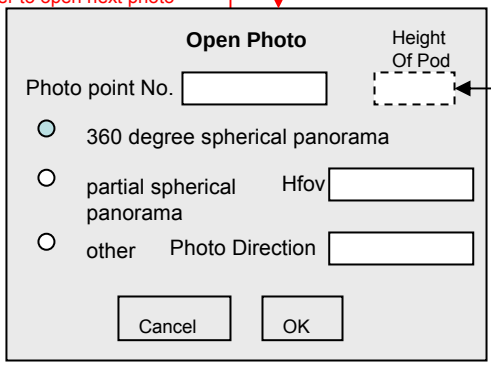
Photo Survey Adjustment Wizard



User cannot select feet and grads. It will always be gads and metres, degrees and metres or Degrees and feet

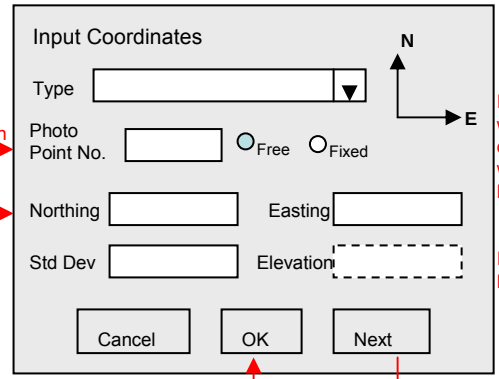


OK cycles back to allow user to open next photo



Height of pod prompt If in 3D mode

User can call a photo traverse point any alpha numeric they want. If photo point number does not Match opened photo then warn user

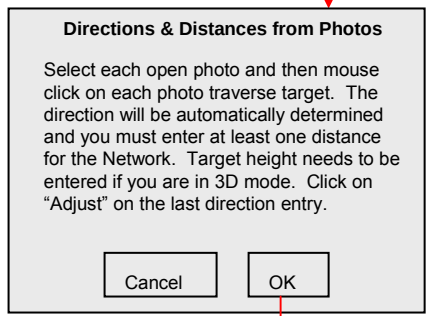


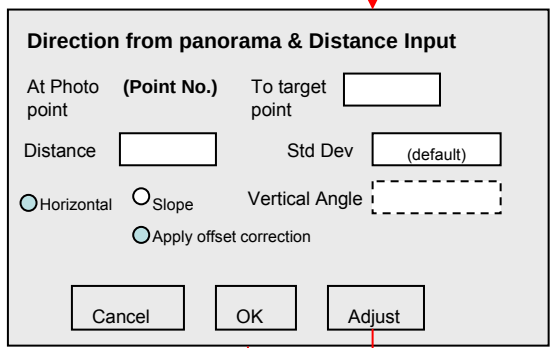
I labeled "Free" for what we call constrained so user will understand. See Notes below

If in3D mode, then Prompt for elevation

Populates table and cycles to allow user to input other point coordinates

User receives this prompt





Populates table and waits for next mouse click

When user finishes mouse picking targets on the open photos and hits adjust, then VRMapper does condition checks and then runs 2D (horizontal adjustment and displays results to user. If in 3D mode, then VRMapper calculates elevation differences and populates observation table and then runs 1D (vertical adjustment) and displays result to user. If results are good user accepts, if not, user edits observations and re-runs until acceptable. User Then proceeds to mapping, measuring or visualization tasks.

VRMapper 2D & 1D Observation Input Use Case and Condition Model

This document describes the user input use cases for 2D (horizontal) and 1D (vertical) mode, the validation checks prior to running the adjustment and the input/output after the adjustment.

Terms:

Panorama point: point where partial or full 360 degree panorama photo is taken.

Target point: cone or pole target placed at points where panorama's will be taken at a particular location.

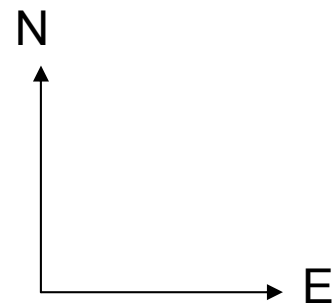
Panorama network: two or more panorama points taken at a site location that are normally intervisible with each other

Fixed coordinate: means the input coordinate is held fixed – not adjusted

Constrained coordinate: means the input coordinate is adjusted according to its weight. We will have a default weight (standard deviation) of ten metres. This will allow approximate coordinates derived by GPS to be entered and adjusted. All the adjustment goes into the coordinates because the standard deviation is high. This does not disturb the distance and direction observations as they are relatively more accurate.

Types of observations:

Types	Properties
coordinates	•Manual input from pull down dialogue box •Must ask user if coordinates are fixed or free to float •Prompt for N northing and E easting (from GeoTools list) coordinate and zone if and standard deviation (default from preferences) UTM = Universal Transverse Mercator Map Projection •Coordinate axes defined (see diagram to right) This would allow various coordinate type input/outputs



Point # ☐ ○Fixed ○Free

Northing ☐ Easting ☐ StdDev (default displays)

Coordinate UTM ▼

(Default Map projection coordinate system displays here)

distance

- Two methods - manual input from pull down dialogue box or input when user is picking directions from panorama's
- Units to be set in preferences (metres or feet) & standard deviation default from preferences
- Horizontal distance input or slope distance with vertical angle (VRMapper reduces to horizontal)

Manual Input dialogue

At point ☐ To point ☐

Distance ☐ StdDev (default displays)

◉ Horizontal ○ Slope Vertical Angle ☐

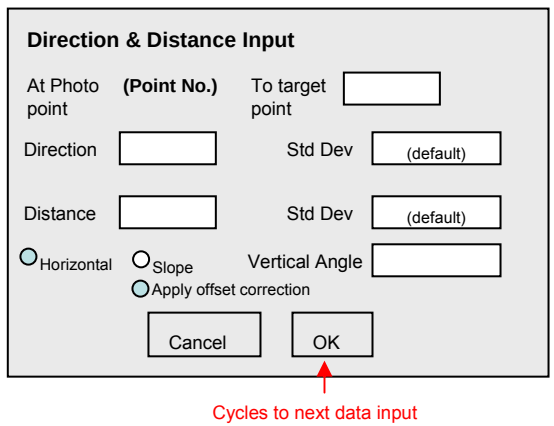
direction

- There are two methods for direction input – manual dialogue input and from panorama's
- Degree See wizard for input dialogue for directions from panorama
- Manual input from pull down menu (user prompts follow)
- or grad option is set in preferences

At point ☐ To point ☐

Direction ☐ StdDev (default displays)

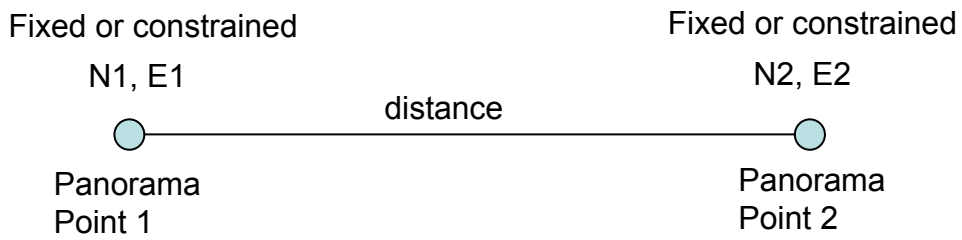
Direction & Distance Input



Use Case No. 1 – Two Panorama Points

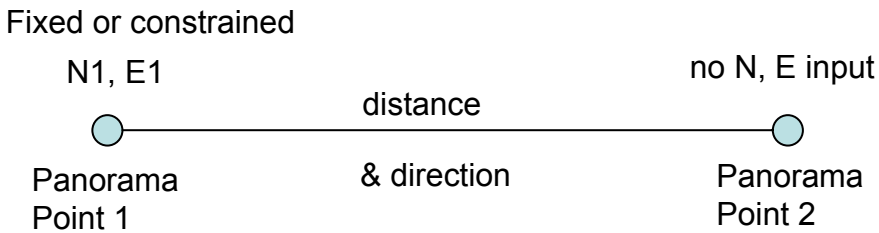
- the user must input one distance
- There are two possible minimum input condition cases – see diagrams below

(a)



Action: OK to run adjustment

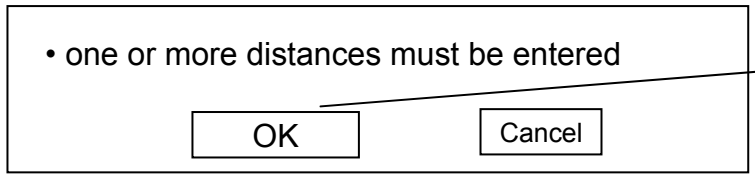
(b)



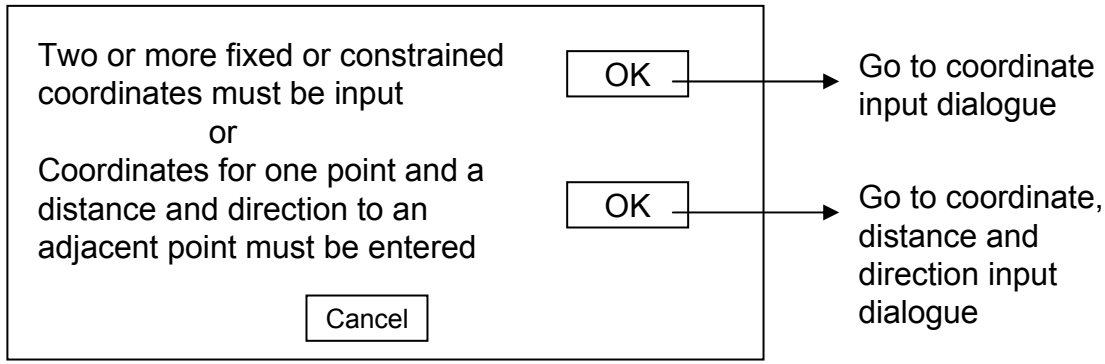
Note: if the user entered a constrained coordinate then it will be held fixed as there is only one coordinate input in this case. VRMapper calculates coordinates for second point and then the user can proceed with mapping, etc. Simple formula for calculating coordinates for second point is shown in last slide of this section.

Actions: Display appropriate message to user prior to running the adjustment

If no distance was entered always display:



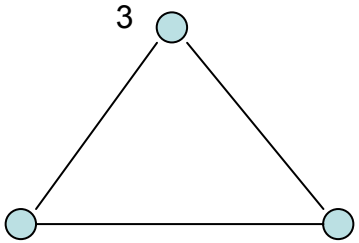
If one or no coordinates were input or only one coordinate was input then prompt user:



Use Case No. 2 – Three or more panorama points

•The goal is to ensure that there are sufficient coordinate, direction and distance observations to solve coordinates for all the points in the network. Here are some typical network examples for reference.

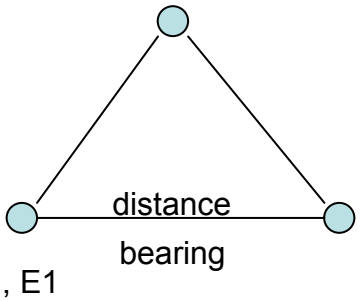
Simple three point case (a)



This case is easy as we have the one distance for scale and the two fixed or constrained coordinates to provide orientation for the network. The distance could be for any one of the legs of this network. If there are more observations than shown such as a coordinate for point 3 and other distances, then this simply provides a stronger positioning solution and would be good practice. The lines in the diagram indicate directions observed which would typically be derived from the panorama's.

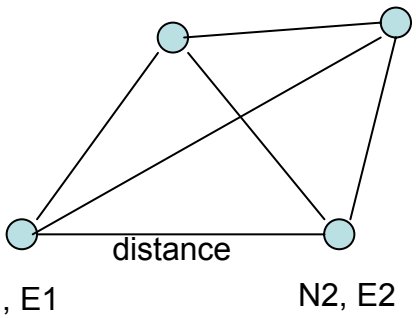
Conclusion: A 3 or more point network can be solved by the adjustment if the network has two or more fixed/constrained coordinates and one distance

Simple three point case (b)



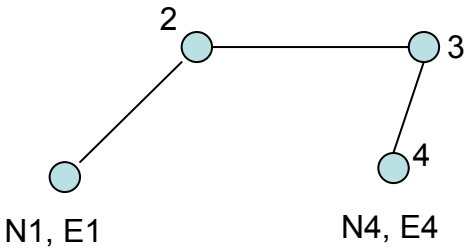
If we have coordinates for only one point, then we always need a direction/bearing from the known coordinate and a distance to one of the other points. This provides orientation and distance to solve coordinates for the second point.

Four or more point case



- this network adjustment will run fine. As we have two points with coordinates
- there must be one distance entered
- there must be two or more directions to any panorama point or a direction and a distance to the point

Open traverse case



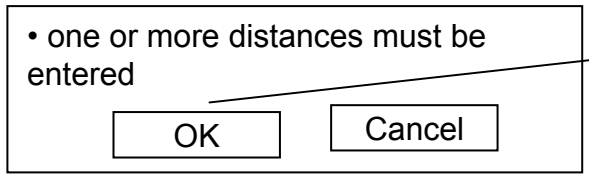
•There must be a distance and a direction for each leg

•If this network has fixed or constrained coordinates on each end, then the adjustment can be run directly

•If the network has only a fixed/constrained coordinate on the one point, then the user would need to be prompted to input a direction to the connected point for orientation So the second coordinate can be calculated

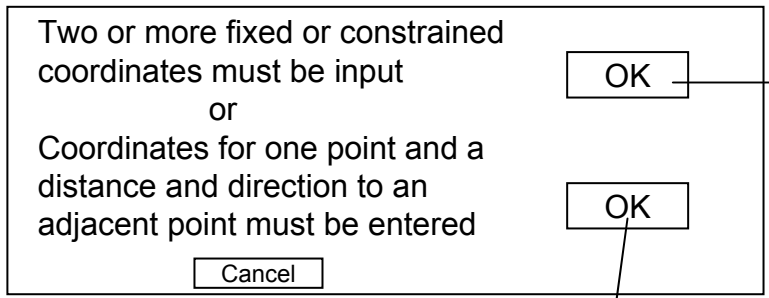
Actions: Display appropriate message to user prior to running the adjustment

(a) If no distance was entered always display:

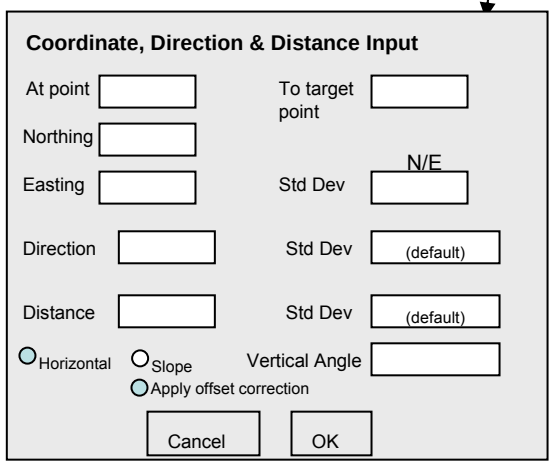


Go to distance dialogue input

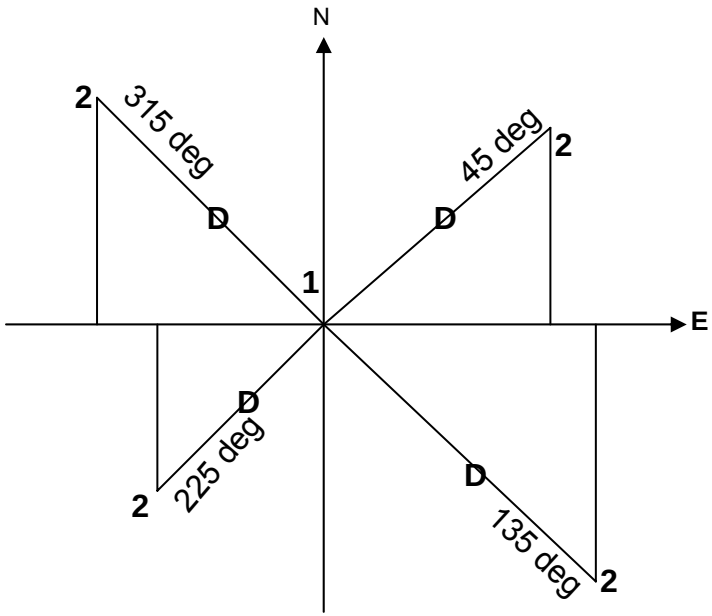
(b) If one or no coordinates were input then prompt user:



Go to coordinate input dialogue as user wants to input coordinates for points



D= distance



The following formula's show how to calculate the coordinates of point 2

* We know coordinates for point 1

* We know the bearing to point 2. See examples for each Quadrant (eg. Bearings of 45, 135, 225 and 225).

•Bearing is full circle. North = 0 degrees clockwise to 360 degrees.
For conversion: 360 degrees = 400 grads

* We know the distance from point 1 to point 2

Difference in northing = Distance * Cos (bearing)

Difference in easting = Distance * Sin (bearing)

N of point 2 = N of point 1 + difference in northing

E of point 2 = E of point 1 + difference in easting

If there are only two points we can start mapping

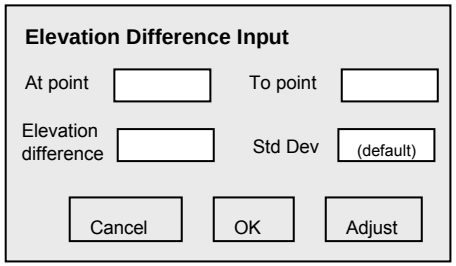
If there are more than two points we run the adjustment

Note: Prior to running the adjustment, check to ensure that each coordinate or panorama point has a direction and distanc or directions from two or more panorama points. We can do user tests with the adjustment to see what error snooping and messages/warnings the adjustment generates and prompt the user accordingly for any other situation.

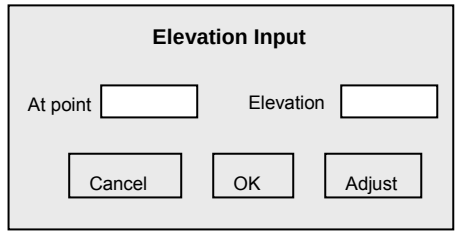
Vertical Adjustment

There are two ways to input elevation differences – manual dialogue or derived from 3D mode in VRMapper (user would know height of photopod and height of photo traverse point targets.

Manual dialogue input

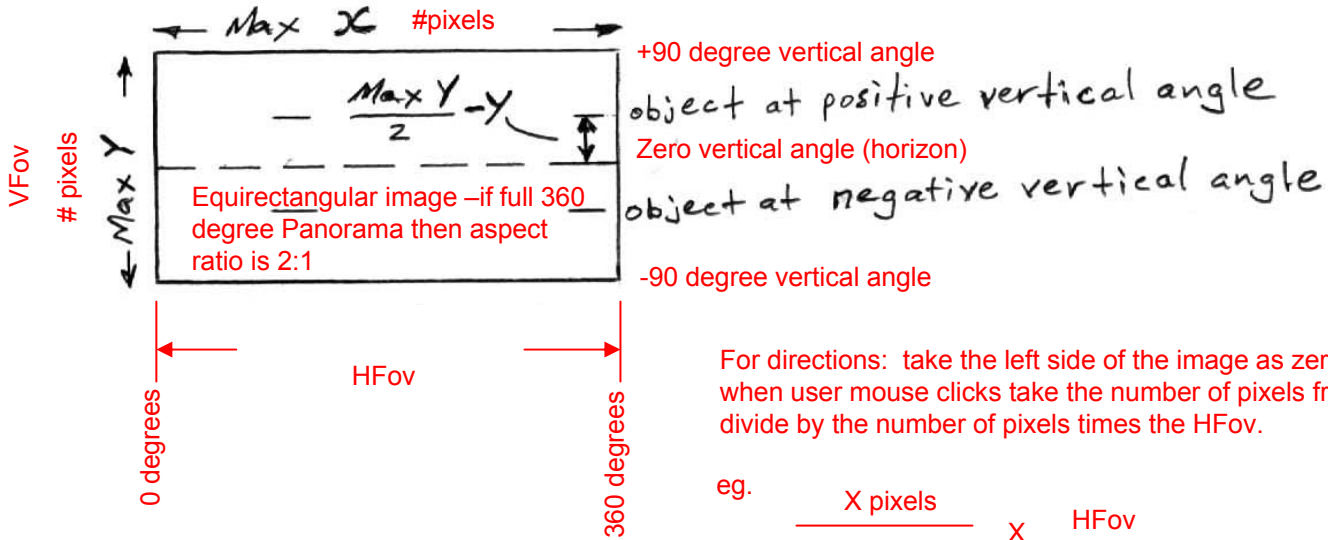


A vertical adjustment can be run manually from the main pull down menu. Manual dialogue entries for known elevation and elevation differences are made and the adjustment is then run.



If the user is in 3D mode and if the horizontal 2D adjustment is accepted then a vertical adjustment can be run. First VRMapper checks that the user has input at least one elevation. Otherwise, the user is prompted to input at least one elevation. The elevation difference formula follows:

Determining vertical angle and directions from equirectangular panorama



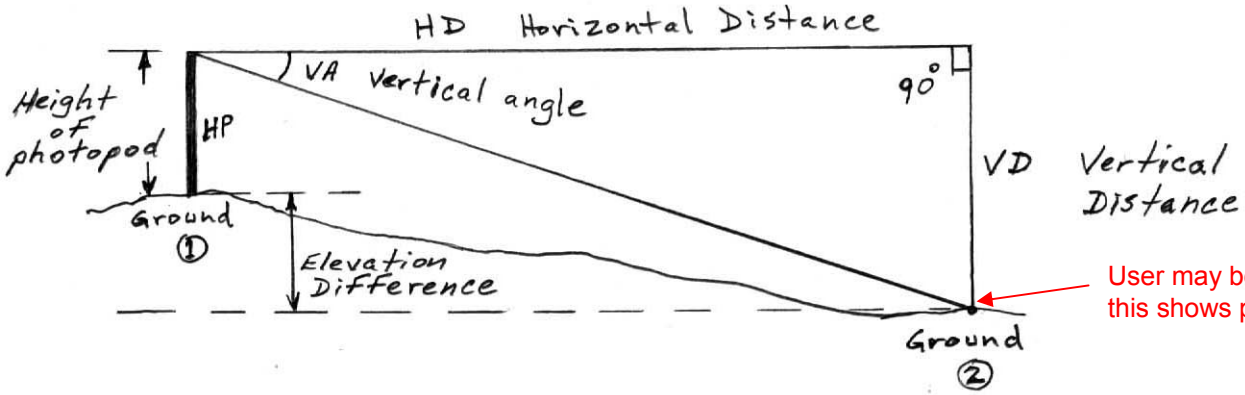
For directions: take the left side of the image as zero degrees. Then to calculate direction when user mouse clicks take the number of pixels from the left side to the point picked, divide by the number of pixels times the HFov.

eg. $\frac{\text{X pixels}}{\text{Max X pixels}} \times \text{HFov}$

Same principle for vertical angle. Simply a ratio of pixels above or below the horizon. The VFov is calculated by knowing the HFov and ratioing the number of pixels as we know the aspect ratio is 2:1.

eg. $\frac{\text{Max Y pixels}}{\text{Max X pixels}} \times \text{HFov} = \text{VFov}$

Here is a diagram (side view) illustrating the geometry to determine elevation differences



User may be using a cone or pole target this shows point to ground level

If the user is in 3D mode and the 2D adjustment was accepted then we have coordinates for all of the photo network points. VRMapper can calculate the distance between each photo traverse point. It is simply pythagorus theorem – square root of the squares of the differences between the northings and eastings of the coordinates (HD in diagram). We saw how the vertical angle was calculated from the equirectangular image and we know the height of the photopod and the height of the target, therefore we can simply calculate the elevation difference

The horizontal distance is calculated by the following formula commonly called an "inverse" or "join" by land surveyors.

$$HD = \sqrt{(N1 - N2)^2 + (E1 - E2)^2}$$

$$VA = \frac{\frac{Max\ Y}{2} - Y}{Max\ Y} * VFov$$

$$VD = HD \tan VA$$

$$El.\ Diff. = HP + VD - target\ height$$

This following section describes the desired features of an image viewer for VRMapper

Input Image: Equirectangular is a common output image for stitching programs (PTGui, etc.). Ben Kreunen's page http://nodename.com/lab/dispmapPano/ Shows a typical full 360degree panorama mapped to the equirectangular (sometimes called spherical) format. The page also shows the image reprojected in a rectilinear map projection in an immersive viewer. We are using equirectangular map projection in VRMapper for the main survey and mapping functions as the map projection preserves direction to points so that we can easily derive direction observations for the survey network and directions to visible objects to be mapped. The equirectangular projection looks fairly normal through most of the projection, however, the distortion increases at the top and bottom. One effect is that lines appear bent. I believe that using equirectangular is visually fine for the survey and mapping functions of VRMapper.

Critical Features: The critical features of the initial VRMapper viewer are as follows:

- display equirectangular image in a drag sizable window
- Enable user to pan by holding down the mouse and dragging image
- Zoom in or out with cntrl + and – keystrokes and zoom in and out using scroll wheel
- Show the field of view,
- direction (if orientation known)
- Brightness slider
- Show filename
- Be able to pan back and forth across end seams

For reference the panotools wiki gives viewer overview: http://www.panotools.info/mediawiki/index.php?title=Panorama_Viewers

Desired Viewer Features: As the equirectangular image displays inherent distortion, it would be nice to migrate to a viewer that reprojects the equirectangular image to a rectangular (gnomonic) projection that makes the image appear normal to the eye – see Ben's example above. Many panorama viewers display in this manner. I understand that the equirectangular image is remapped on the fly when the user pans/zooms to new areas of interest. We need to consider whether the viewer will just be used within VRMapper or whether it will be used in web or standalone presentations as well. Do we need a stand alone viewer (like FSP, etc.) and java browser viewer (like PTViewer, etc.) or both? We need to consider in the design how accommodate display of full 360 degree panorama's and partial panorama's and even normal photos. The original PTViewer written by Helmut Dersch has been enhanced by Fulvio Senore and the viewer and the source code is available here: http://www.fsoft.it/panorama/ptviewer.htm In order to create our viewer, it may be easiest to review Fulvio's code for PTViewer 2.8 or his standalone version and then create our own from those principles.

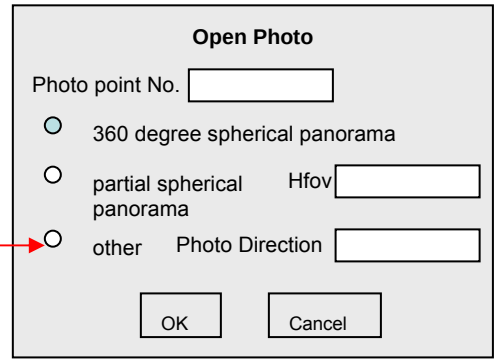
I understand that our design function is to be able to have the map view have a hotspot that can be activated to display the photo in a pop up viewer window and the map will show the direction of view within VRMapper program. Here is a link to a commercial java applet that shows a good example of displaying a map and the field of view. The image is not remapped on the fly to a rectangular projection.
http://www.idyll-on-the-rocks.com/tour/index.html

We also have had discussion about being able to export the panorama to a google earth site. We could also provide a publish feature for Worldwind/Virtual Earth as well. The code for displaying a panorama using PTViewer on Google Earth with a text description and the panorama is at the two main panotools wiki's:

http://wiki.panotools.org/Adding_Panoramas_to_Google_Earth

http://www.panotools.info/mediawiki/index.php?title=Geo-referencing_panoramas_with_Google_Map

Opening Types of Images in VRMapper



Reads EXIF file if it exists in order to determine Hfov

Note: user can set preferences to open 360 Hfov panorama's directly without the above prompt and the file number becomes the photo point number.

Type 1: 360 degree HFov panorama (stitched to equirectangular projection)
This is the test 360 degree panorama image from photo point number one. It has an aspect ratio of 2:1 (3555 pixels HFov and 1778 pixels VFov). The user checks the radio button in the open photo dialogue. VRMapper then cycles to open the next image. Preferences can be set to open images directly as 360 degree panoramas without the dialogue pop-up (many mapping users will be using all 360 degree panoramas.

Type 3: normal photo (other)
This image is used for visual presentation not for mapping. The user can enter the direction of the center of the image if known.

Approximately 50 degree HFov

Type 2: partial panorama
This is one of the test 187 degree panorama images from photo point number one. It has a HFov of 187 degrees (1880 pixels HFov). The user checks the radio button in the open photo dialogue and inputs the HFov. VRMapper then cycles to open the next image.

Note: horizontal angles/directions are calculated by using ratios of the pixel count in the image where a point is mouse clicked. I would use the left side of the image as "0" direction. In panorama example one, if the user clicks on a target VRMapper records the pixel number count from the left side. Eg. X pixel is 500 then VRMapper calculates the direction of the target by referencing to the left side of the image to the target and calculates the direction.

$$\frac{500}{3555} * 360 = 50.6329 \text{ degrees (direction)}$$

Similarly for vertical angles. The pixel count is ratioed above and below the horizon line to calculate the positive or negative vertical angles to a target or map object. See slide 22 for vertical details.

Fulvio's FSPViewer (standalone)

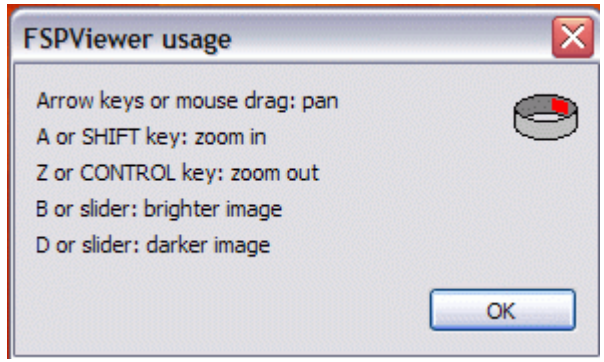
http://www.fsoft.it/panorama/fspviewer.htm

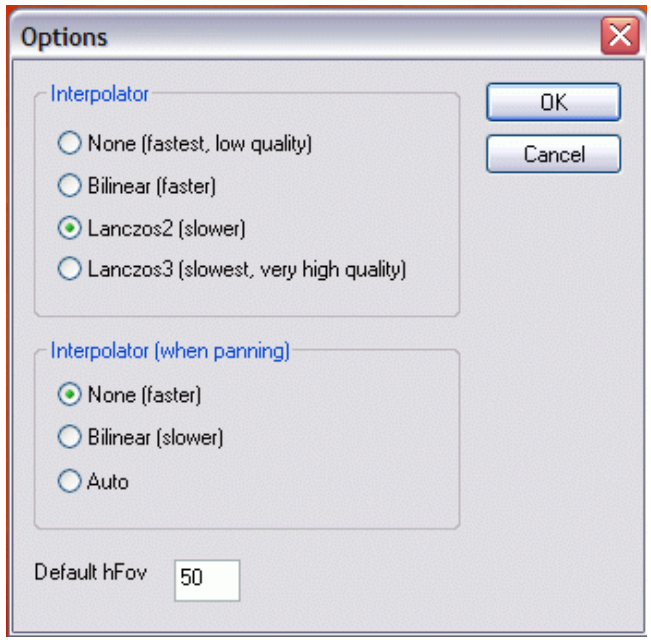
Jim's comment: good basic navigation controls

Zoom percent

HFov

VFov





Jim's note: we are adding direction and tilt. I like how Fulvio shows zoom % and The horizontal and vertical field of view and I think they should be shown in our basic viewer as well. The slide brightness control would be handy and the zoom extents as full image as well.

Other links by Fulvio Senore

This link shows all PTViewers active control parameters

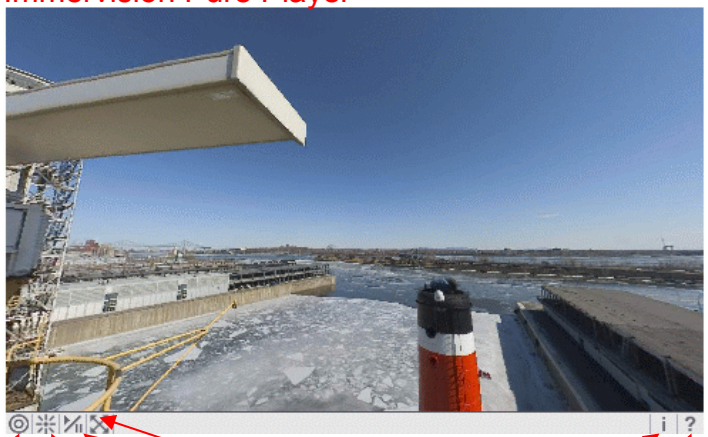
http://users2.ev1.net/~wufdog/PT/ptviewertester/ptviewertester.htm

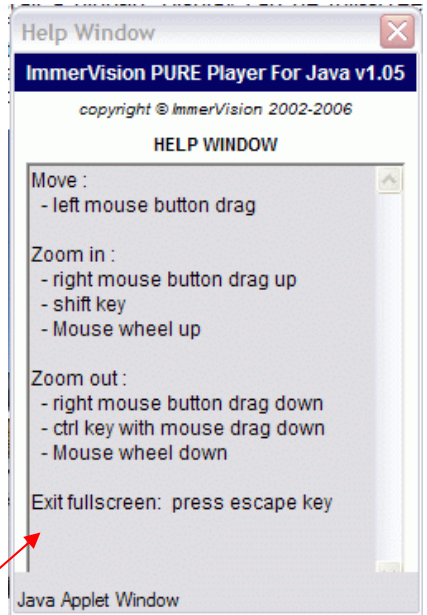
Shows PTViewer Applet - Parameter Tags

http://users2.ev1.net/~wufdog/PT/ptviewerscripting/_ptViewer_Start.htm

Samples of the Common Viewers

Immervision Pure Player





PTViewer

This is from Ben Kreunen's site. He shows the Pan angle, tilt angle and Field of view. These values change as the view is changed or zoomed. See the link below for the active viewer sample.

The Mouse is currently over Hotspot No:	
Pan Angle (Degrees)	165.7903842
Tilt Angle (Degrees)	-13.63932061
Field of View (Degrees)	91.33412286

http://www.path.unimelb.edu.au/~bernardk/tutorials/360/ptviewer/vr/ptv002.html

Deval VR viewer

Equirectangular projection http://en.wikipedia.org/wiki/Plate_carr%C3%A9e_projection

From Wikipedia, the free encyclopedia

The **equirectangular projection**, also called the **equidistant cylindrical projection** or **geographic projection**, is a very simple map projection attributed by Ptolemy to Marinus of Tyre, who Ptolemy claims invented the projection about 100 AD.[1] The projection maps meridians to equally spaced vertical straight lines, and parallels to equally spaced horizontal straight lines.

Definition

In modern notation:

$$(\lambda, \phi) \mapsto (\lambda \cos \phi_1, \phi)$$

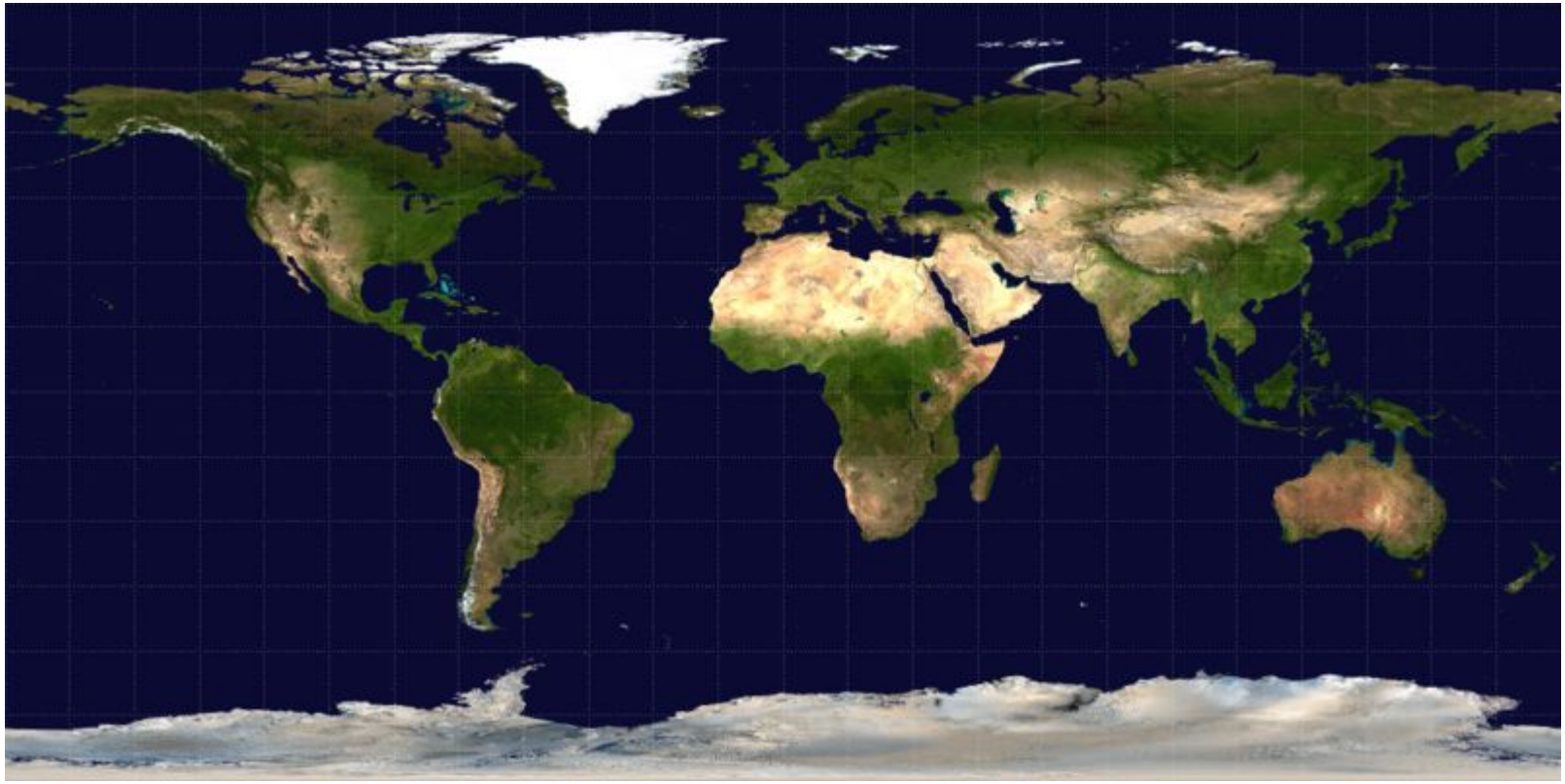
λ is the longitude from the central meridian of the projection,
ϕ is the latitude
ϕ_1 are the standard parallels (north and south of the equator) where the scale of the projection is true.

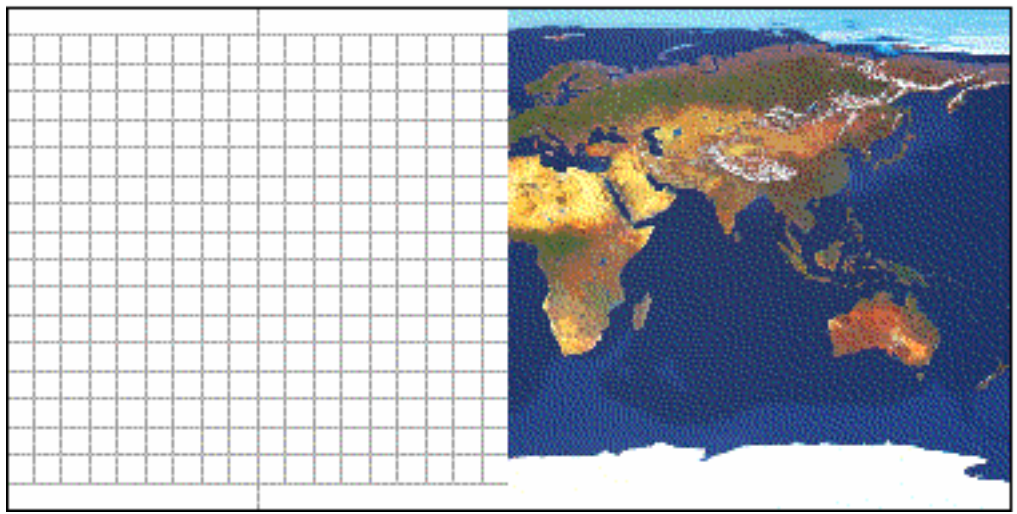
Note that on the right side of the equation, the coordinates λ and ϕ are linear, not angular, measurements. The point $(0,0)$ is at the center of the resulting projection (in particular, this requires the input range to be $[-180, 180]$ rather than $[0, 360]$).
The **plate carrée** (French, for "flat and square"), is the special case where ϕ_1 is zero.

Deducing the Equidistant Cylindrical Projection

Suppose an arbitrary projection in whose equatorial aspect:
All meridians are standard equally-spaced vertical lines
All parallels are horizontal, equally-spaced, equally long lines
The rectangular result is the very simple cylindrical equidistant projection (*equidistant* only along meridians and two parallels), having a multitude of alternate names.
It is neither conformal nor equal-area, and despite the resemblance to the stereographic cylindrical, it is not truly created by a perspective method.
Since scale is constant along meridians, *y* is simply *R*φ; two parallels are standard at ±φ0, with circumference *R* cos φ0. Constant scale along equal-length parallels means:
x = *R*λ cos φ0
y = *R*φ
Different standard parallels only change the map's width/height ratio. For the common special case of standard Equator (usually known as the *plate carrée*), cos φ0 = 1, and latitude and longitude are directly mapped into *y* and *x* respectively, therefore a world map is a 2:1 rectangle.

Sample Equirectangular image of the earth – aspect ratio is 2:1 360 HFOV by 180 VFOV





These links may help as references for viewer design:

PTViewer web java applet	http://www.fsoft.it/panorama/ptviewer.htm
FSPviewer standalone viewer	http://www.fsoft.it/panorama/FSPViewer.htm
Math to map gnomonic projection	http://mathworld.wolfram.com/GnomonicProjection.html
PanoToolsNG viewer overview	http://wiki.panotools.org/Panorama_Viewers
Paper on mapping Equirectangular projection	http://nodename.com/blog/wp-content/themes/fullwidth2/print.php?p=36
Math and code to remap sphere to Equirectangular image	http://nodename.com/blog/2006/01/16/psyarks-displacementmapfilter-tutorial/
Panotools info site wiki On viewers	http://www.panotools.info/mediawiki/index.php?title=Panorama_Viewers
Ben Kreunen's sample of Equirectangular and gnomonic (rectangular) projection. Many Viewers remap this way. The gnomonic looks natural.	http://nodename.com/lab/dispmapPano/
Henry Bottomley's java code for Various map projections.	http://www.btinternet.com/~se16/js/MapProjections.java
Good visual overview of panoramic Image projections	http://www.cambridgeincolour.com/tutorials/image-projections.htm
Panotools wiki on projections	http://wiki.panotools.org/Projections

Least Square Input/Output Overview and VRMapper Output Presentation

This document outlines the least square input/output format and the adjustment output presentation for VRMapper. The VRMapper development will use a sample five point panorama network with the coordinate, distance photopod and target height information. The 2D adjustment will first be developed and tested with the vertical 1D adjustment to follow.

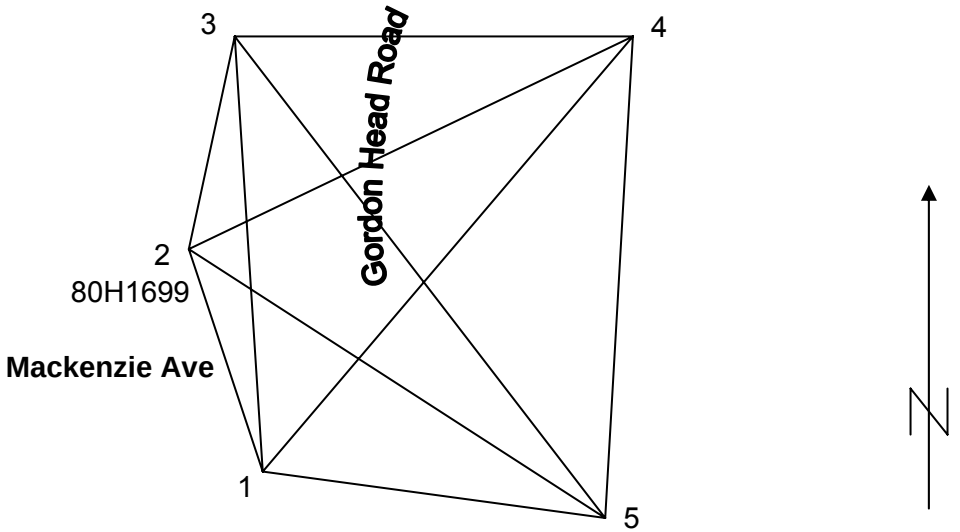
Description of Sample Road Intersection Test Network

Location: Corner of Mackenzie Ave and Gordon Head Road, Victoria

Network: Occupied five points with VRPhotopod

Apartments

Chevron Gas Station



Apartments

University Lands

Point numbers 1, 2, 3, 4, 5 were occupied with the VRPhotopod at a height of **5.62** metres and the photographs are under the “intersection/stitched” folder as “int1, int2, int3, int4, int5”. **Point number five** was also occupied with the VRPhotopod at **6.92** metres AGL. The targets were small orange traffic cones. Some of the cones had the 0.1 metre cone targets placed at the top of the cone.

Positioning Control

Points 1, 2, 3, 4 and 5 were occupied using autonomous GPS using approximately one minute of averaged values derived from a Garmin II GPS receiver. Point number 2 is integrated survey area monument 80H1699. The autonomous derived coordinate compared to the published MASCOT (geodetic) value for 80H1699 agreed to less than one metre in northing and easting.

Observed autonomous GPS derived Coordinates and Fixed point 80H1699 to seed in horizontal least square adjustment. 80H1699 is a geodetic control monument.

	Session	One	Session	Two	**Mean**	**Mean**
Point	Northing	Easting	Northing	Easting	**Northing**	**Easting**
80H1699	**5368433.76**	**476367.70**				
1	8421	6367	8421	6367	**8421**	**6367**
2	8435	6367	8434	6368	**8434.5**	**6367.5**
3	8442	6368	8441	6370	**8441.5**	**6369.0**
4	8445	6396	8447	6396	**8446.0**	**6396.0**
5	8419	6389	8419	6391	**8419**	**6390**

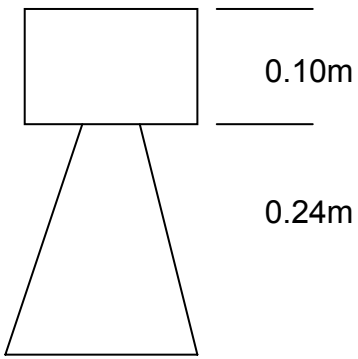
Distances measured by cloth tape

From	To	Forward	Reverse	Mean
1	2	12.02	12.06	12.04
1	5	25.06	25.14	25.10
2	3	8.55	8.58	8.57
3	4	27.83 weighted	27.70	27.80
4	5	30.88	30.84	30.86

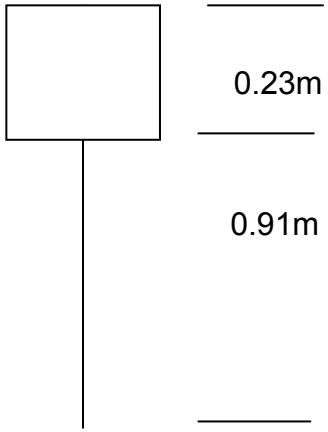
Distances measured by laser distance ranger

From	To	Forward	Reverse	Mean
1	2	12.09	12.06	12.07
1	3	20.52	20.52	20.52
1	5	25.10		

Target Heights



Cone target. Orange traffic cone with white translucent polyethelene target



Pole target. Aluminum monopod with polyethelene target

Input File for Sample Network

Uses an XML input file with the following tags:

<network>	network steup
<description> ... </description>	project details
<parameters ... />	network parameters
<points-observations> ... </points-observations>	observations
</network>	

The following is the test sample of the XML input file for the intersection. VRMapper 2D uses only certain network parameters and three types of observations (coordinates, distances and directions):
2) the <network> tag is static (always the same)
3) the <description> tag is from a freeform input prompt in VRMapper
4) <parameters…./> there are five parameters, four are static and one (confidence level - CL) is derived from the preferences set in VRMapper. The two options are standard (68%CL “0.68) and 95%CL (0.95). The confidence level parameter is shown highlighted in below in the sample XML input file below.
5) <points-observations> are a set input syntax. The observations are highlighted in blue below and are derived from the VRMapper dialogue input. Directions are calculated by VRMapper by clicking on the panorama target points or there is a manual direction input dialogue as well. The user input information is shown below highlighted in yellow.
6) The standard deviations are from the preferences or when the user overides the preference in an input dialogue box.

```
********** Start of XML Input file ******************************
<?xml version="1.0" ?>
<network axes-xy = "en"  angles="right-handed">
<description>
* five node photo survey network
* directions observed at each node with VRPhotopod - 5.62 metres above ground level
* distances are horizontal derived by cloth tape
* autonomous GPS seeded coordinates for points 1, 3, 4, 5 and point 2 is a known ISA monument
 </description>
<parameters
    sigma-apr = "10.000"
    conf-pr   = "0.950"
    tol-abs   = "10000.000"
    sigma-act = "apriori"
    update-constrained-coordinates = "yes"
/>


<points-observations>
<point id="1" x="6367.00000" y="8421.00000" adj="XY" />
<point id="2" x="6367.70000" y="8433.76000" fix="XY" />
<point id="3" x="6369.00000" y="8441.50000" adj="XY" />
<point id="4" x="6396.00000" y="8446.00000" adj="XY" />
<point id="5" x="6391.00000" y="8419.00000" adj="XY" />
```

<obs from="1">
<direction to="2" val="253-55-26" stdev="900.00" />
<direction to="3" val="258-47-17" stdev="900.00" />
<direction to="4" val="303-29-31" stdev="900.00" />
<direction to="5" val="356-59-46" stdev="900.00" />
<distance from="1" to="2" val="12.04" stdev="200.00" />
<distance from="1" to="5" val="25.10" stdev="200.00" />
</obs>

<obs from="2">
<direction to="1" val="53-17-06" stdev="900.00" />
<direction to="3" val="244-46-01" stdev="900.00" />
<direction to="4" val="299-42-07" stdev="900.00" />
<direction to="5" val="359-14-38" stdev="900.00" />
<distance from="2" to="3" val="8.57" stdev="200.00" />
</obs>

<obs from="3">
<direction to="4" val="129-23-16" stdev="900.00" />
<direction to="5" val="187-34-30" stdev="900.00" />
<direction to="1" val="233-38-31" stdev="900.00" />
<direction to="2" val="240-39-58" stdev="900.00" />
</obs>

<obs from="4">
<direction to="5" val="74-20-49" stdev="900.00" />
<direction to="1" val="114-55-08" stdev="900.00" />
<direction to="2" val="131-44-06" stdev="900.00" />
<direction to="3" val="146-11-24" stdev="900.00" />
<distance from="4" to="3" val="27.80" stdev="200.00" />
</obs>

<obs from="5">
<direction to="1" val="117-12-11" stdev="900.00" />
<direction to="2" val="140-09-18" stdev="900.00" />
<direction to="3" val="153-12-36" stdev="900.00" />
<direction to="4" val="203-12-14" stdev="900.00" />
<distance from="5" to="4" val="30.86" stdev="200.00" />
</obs>

</points-observations>
</network>
********** end of XML input file **********

(number)

VRMapper should run data condition checks prior to running the adjustment. The following notes outline the desired adjustment output. May want to have a short summary output window that shows critical information. The window would also have a detail button that will take the user to a full listing of the detailed output of the adjustment.

VRMapper Adjustment Output Format

********** VRMapper Horizontal Adjustment Output **********

Date: *(date)* Version: *(version No.)*

Network Description

(display free form text from VRMapper dialogue entry)

Observation Input Summary

(number)	Fixed points	Confidence level	(*95% or 68%)*
(number)	Constrained points	apriori	10
(total)		aposteriori	*(from adjustment)*

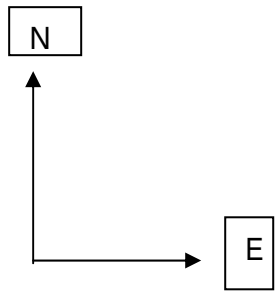
(number)	Number of directions
(number)	Number of distances
(number)	Total observations
(number)	Defects

Map projection coordinates: (display type of map projection system from preferences)

Coordinate Summary

Fixed

Point No.	North Coordinate	East Coordinate
(number)	*(number)*	
Etc.		



Adjusted Coordinates

Point No.	Adjusted Northing	Adjusted Easting	Correction	Mean position error
(number)	*(number)*	*(number)*	*(number)*	*(number)*
Etc. for each point				

Summary of Observations

At point	To point	Type	Input value	Adjusted value	Difference Dist. mm/ft Dir. Sec/grads	Std dev Dist. mm/ft Dir. Sec/grads
1	2	dir	137-44-02	137-47-21	199	605
	3	dir	142-35-00	142-36-49	109	524
	4	dir	187-22-58	187-17-58	-299	519
	5	dir	240-46-27	240-46-18	-9	631
	5	dist	25.10	25.11	11	53
2	Etc.					

Rejected Observations

At point	To point	Type	Difference Dist. mm/ft Dir. Sec/grads
3	2	dir	-1684
2	3	dir	1488