

FEATURE POSITIONING ON GOOGLE STREET VIEW PANORAMAS

Victor J. D. Tsai and Chun-Ting Chang

Dept. of Civil Engineering, National Chung Hsing University, Taichung 40227, Taiwan - jdtasai@nchu.edu.tw

Commissions IV, II, VI, WG II/6, IV/5,4,1,8, II/6, VI/1,2

KEY WORDS: Google Maps API, Google Street View, 3D Positioning, Intersection, Points of Interest (POI)

ABSTRACT:

Location-based services (LBS) on web-based maps and images have come into real-time since Google launched its Street View imaging services in 2007. This research employs Google Maps API and Web Service, GAE for JAVA, AJAX, Proj4js, CSS and HTML in developing an internet platform for accessing the orientation parameters of Google Street View (GSV) panoramas in order to determine the three dimensional position of interest features that appear on two overlapping panoramas by geometric intersection. A pair of GSV panoramas was examined using known points located on the Library Building of National Chung Hsing University (NCHU) with the root-mean-squared errors of $\pm 0.522\text{m}$, $\pm 1.230\text{m}$, and $\pm 5.779\text{m}$ for intersection and $\pm 0.142\text{m}$, $\pm 1.558\text{m}$, and $\pm 5.733\text{m}$ for resection in X, Y, and h (elevation), respectively. Potential error sources in GSV positioning were analyzed and illustrated that the errors in Google provided GSV positional parameters dominate the errors in geometric intersection. The developed system is suitable for data collection in establishing LBS applications integrated with Google Maps and Google Earth in traffic sign and infrastructure inventory by adding automatic extraction and matching techniques for points of interest (POI) from GSV panoramas.

1. INTRODUCTION

The emerging development in microelectronic industry has raised the productivity and performance in mobile mapping devices and the demands in consequent value-added services. Among them the location-based services (LBS) on web-based maps and images have come into real-time since Google launched its Street View imaging services in 2007 (Vincent, 2007). GSV images were taken from a ring of eight cameras plus a fish-eye lens on top for producing the most popular 360-degree panoramic views (Anguelov *et al.*, 2010). As shown in Figure 1, the 4th generation Street View car was equipped with a R5 system at a height of about 2.5 meters, Global Positioning Systems (GPS) units for vehicle positioning, three laser range scanners for the measuring distances of objects up to 50 meters 180° around the vehicle, and network antennas for scanning 3G/GSM and Wi-Fi hotspots. Each available GSV panorama can be requested in an HTTP URL form using Google Maps Application Programming Interface (Google Maps API), along with the projection type, the geodetic position of the Street View car and its moving direction with respect to the North at the time of image capture. Hence, it provides the user opportunities to determine the spatial position of features that appear on the GSV panoramas.

This research employs Google Maps JavaScript API, Google App Engine (GAE) for JAVA, Asynchronous JavaScript And XML (AJAX), and Proj4js (Evenden, 2003) in developing an internet platform for accessing the orientation parameters of GSV panoramas in order to determine the geographic position of any interest features that appear on two overlapping images by intersection, and to determine the position of the exposure center of a panorama from at least three known control points by resection. Extracted 3D positional information of the features from intersection can be packed in Keyhole Markup Language (KML) format and stored in GAE Servlet for future LBS applications integrated with Google Maps and Google Earth. Key elements of the developed platform were introduced in the next section, followed by the results and error analyses from examining known points appeared on the GSV panoramas, and conclusions.



Figure 1. Devices adapted on the 4th generation GSV car (Center image edited from http://en.wikipedia.org/wiki/File:Google_Street_View_car_Switzerland.jpg)

2. GOOGLE STREET VIEW AND MAPS API

2.1 Basics for Google Street View Panorama

Though with composing techniques in veiled, a GSV panorama is a 360° surrounding image generated from mosaicking the eight original images captured by the eight horizontal cameras in the sequences as shown in Figure 2. The percentage of horizontal overlap between adjacent original images is about 28.8% for removing the most distorted outside portion of an original image caused by camera lens. Meanwhile, the panorama was aligned with the heading direction of the vehicle at the center. Each GSV panorama was tile-indexed in a 6 zoom-level image pyramid with the size of 416*208 pixels in level 0, with a scale factor of 2, to 13312*6656 in level 5 for covering the field of view of 360° in horizontal and 180° in vertical. Thus, it is convenient to convert the rectangular image coordinates (row, column) in the developed panorama plane into spherical coordinates (latitude, longitude), and vice versa, in the following equivalent as in the level 5 zooming image:

$$360^\circ / 13312 \text{ pixels} = 0^\circ 1' 37.35577'' / \text{pixel} \quad (1).$$

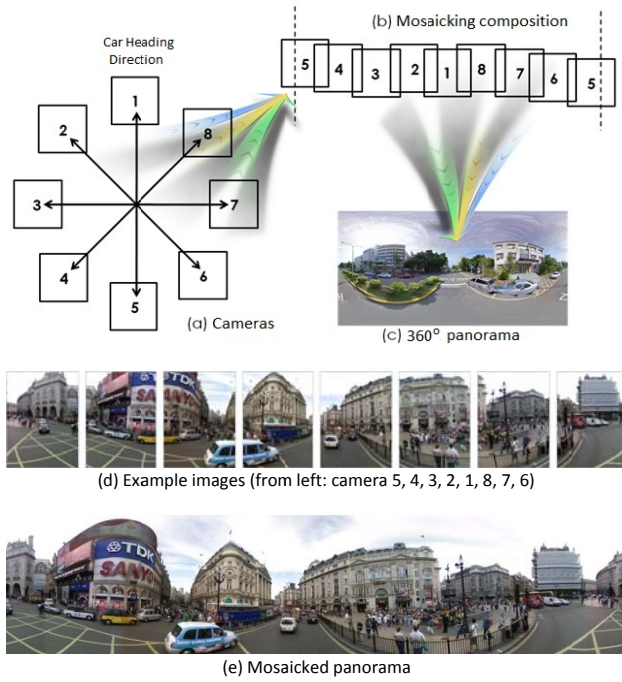


Figure 2. Composition of a GSV Panorama (Example images accessed from <http://maps.google.com/help/maps/streetview/technology/photos-into-street-view.html>)

2.2 Parameters Accessed from Google Maps API

The coordinates associated with the GSV panorama which can be accessed from Google Maps JavaScript API are latitude and longitude in WGS84 reference ellipsoid for its center using the *getLatLng()* method in *StreetviewLocation* class. The camera yaw in degrees to true north, i.e., the car heading direction with respect to true north, can also be requested by using the *getYaw()* method for camera 1 in the *Pov* class, which can be accessed by the *getPov()* method in *StreetviewLocation* class. Meanwhile, the elevation (height from ellipsoid) of the initial point of view can be accessed by Google Elevation API under Google Maps API Web Services.

When being pointed with the mouse cursor in a panorama, a feature or point of interest (POI) can be located from the heading and pitch attributes of the *StreetviewPov* class for its azimuthal angle and vertical angle from the horizon of the point of view, respectively. The POI may be a commercial entity, a traffic sign, or a specific target whose global location to be determined. It may be a labeled point already posted in the GAE Servlet with its coordinates accessed by *StreetviewLocation* class for computing the position of the initial point of view of a GSV panorama.

3. JavaScript IMPLEMENTATION

3.1 Coordinate Systems and Transformations

The coordinates used in the implementation include geodetic coordinates (latitude, longitude, elevation or height from ellipsoid) in WGS84 ellipsoid (EPSG: 4326), TWD97 Transverse Mercator 2° zone 121 X-Y coordinates based on GRS80 ellipsoid and International Terrestrial Reference Frame (ITRF) 1997 (EPSG: 3826), and TWD67 Transverse Mercator 2° zone 121 X-Y coordinates based on GRS67 ellipsoid (EPSG: 3828). The transformations among these coordinate and datum systems are made convenient from using Proj4js (Evenden, 2003;

Proj4js Track and Wiki, 2012) along with EPSG geodetic parameters (OGP Geomatics, 2012) in a JavaScript environment.

3.2 Intersection

As shown in Figure 3, computations of the POI's coordinates in a 3D rectangular coordinate system from two known exposure stations with angular observations are feasible in overlapped GSV panoramas, whose parameters are accessed via Google Maps API as described in Section 2.2. Let $(X, Y, Z)_P$ represent the coordinates of a POI, and $(X, Y, Z)_L$ and $(X, Y, Z)_R$ for the exposure centers for the left and right GSV panoramas, respectively. The coordinates of the POI can be computed by applying intersection of two lines with known directions (Ghilani & Wolf, 2006, 2008) using coordinate geometry as following:

$$X_P = \frac{(Y_L - Y_R) + X_L \cot \beta + X_R \cot \alpha}{\cot \alpha + \cot \beta} \quad (2)$$

$$Y_P = \frac{(X_R - X_L) + Y_L \cot \beta + Y_R \cot \alpha}{\cot \alpha + \cot \beta} \quad (3)$$

$$Z_P = \frac{(Z_L + \overline{LP} \tan \gamma_{LP}) + (Z_R + \overline{RP} \tan \gamma_{RP})}{2} \quad (4)$$

where

$$\overline{LP} = \sqrt{(X_P - X_L)^2 + (Y_P - Y_L)^2},$$

$$\overline{RP} = \sqrt{(X_P - X_R)^2 + (Y_P - Y_R)^2},$$

$$\alpha = \phi_{LR} - \theta_{LP},$$

$$\beta = \phi_{LR} - \theta_{RP} + 180^\circ,$$

$$\phi_{LR} = \tan^{-1} \left(\frac{X_R - X_L}{Y_R - Y_L} \right) + C,$$

C: a constant that depends on the quadrant in which R lies with respect to L,

θ_{LP}, θ_{RP} : POI's headings (azimuthal angles), and

γ_{LP}, γ_{RP} : POI's pitches (vertical angles) viewing from the left center L and right center R, respectively.

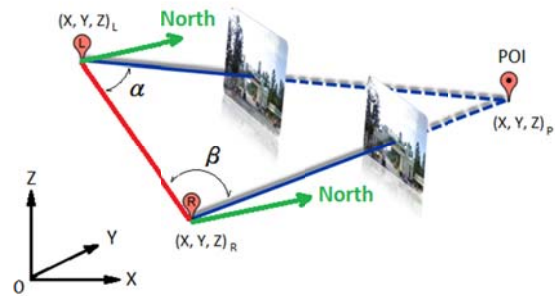




Figure 3. Concept in intersection from two known images

3.3 Resection

In a GSV panorama, the resection approach can also be applied to locate the exposure center of the panorama in retrieving headings (azimuthal angles) and pitches (vertical angles) of at least three labeled points with known coordinates accessed via Google Maps API from the GAE Servlet as described in Section 2.2. Computational procedures for the X, Y coordinates of the resection can be found in Ghilani & Wolf (2006, 2008). As with intersection, the average elevation derived from the known points is taken for the elevation of the GSV exposure center.

3.4 System Implementation

The platform was developed using AJAX techniques with CSS (Cascading Style Sheets) and HTML (HyperText Markup Language) in web user interface design and JavaScript in data communication among program sessions and GAE Servlet through internet. The platform can be run in Microsoft Internet Explorer (IE), Mozilla Firefox, and Google Chrome browsers. Figure 4 demonstrates the implemented software platform in which the window content is divided into five parts: left view panorama, right view panorama, Google Maps, Main menu, and Log for computed POI coordinates. The two GSV viewpoints and Maps viewport allow the user to zoom in/out the panoramas and map/image for detail, respectively. Once clicked on a cursor position in the Maps viewport, the nearest positions of found pair of GSV panoramas via Google Maps API will be marked as  and  with a line of sight for POI intersection.

The Main menu includes three processes: *Intersection*, *Resection*, and *Output* as shown in Figure 5. Within *Intersection* process, the information of the two GSV panoramas and the headings and pitches of the cursor center from the center of respective panorama are dynamically updated during each movement of respective pointing cursor and zooming of panoramas. Once the “Intersection” button was clicked, the POI information, including user-input POI name and description and coordinates in WGS84, TWD97 and TWD67, were output to the Log viewport and can be saved in the *Output* process to GAE Servlet in KML document for future LBS applications by clicking the “Save POI Info” button. While three known labeled points were caught individually by dragging cursor and zooming on the left view panorama in the *Resection* process, the “Resection” button can be clicked to compute and deliver the position of the center of the left GSV panorama to the Log viewport.



Figure 4. GUI of the implementation

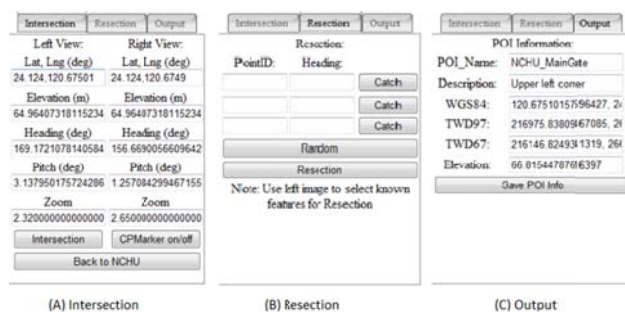


Figure 5. Main functions of the implementation

4. RESULTS AND ERROR ANALYSES

4.1 Positioning Results

A pair of GSV panoramas, as shown in Figure 4, was examined for the positioning precision using 14 known points located on the Library Building in NCHU campus, as shown in Figure 6. The root-mean-squared (RMS) errors of the 14 known points from GSV intersection are $\pm 0.522\text{m}$, $\pm 1.230\text{m}$, and $\pm 5.779\text{m}$ in X, Y, and Z directions, respectively as shown in the upper part of Table 1.

Another test on resection for the two GSV panoramas was done using 8 (i.e., points 2, 5, 7, 8, 9, 18, 23, and 25) off the 14 known points on the NCHU Library Building to determine the precision of GSV positional parameters with the headings (azimuths) and pitches (vertical angles) in a least squares solution. As listed in the lower part of Table 1, the RMS errors of Google provided GSV positions are $\pm 0.142\text{m}$, $\pm 1.558\text{m}$, and $\pm 5.733\text{m}$ in X, Y, and Z, respectively for the two GSV panoramas.



Figure 6. Known points at the NCHU Library Building

Point ID	Left Panorama		Right Panorama		Difference		
	Heading	Pitch	Heading	Pitch	ΔX (m)	ΔY (m)	Δh (m)
2	150.4158	23.9554	-192.8971	26.2612	0.194	-0.425	6.753
3	150.3445	14.7592	-192.9707	16.1406	0.227	-0.398	6.324
5	189.0832	27.1509	-152.7719	24.1234	-0.717	0.992	6.139
6	188.9925	16.8263	-152.8106	14.5866	-0.672	1.105	5.876
7	151.8516	10.8074	-191.7048	11.5834	0.196	-0.516	6.176
8	151.8966	-2.7543	-191.7819	-3.6075	0.275	-0.570	5.576
9	187.8587	12.1810	-154.3627	10.4477	-0.659	1.084	5.794
16	63.1182	6.8131	-146.9189	5.3284	-0.744	1.246	5.648
18	204.2626	6.6288	-141.2059	4.9211	-0.547	1.772	5.525
20	211.0985	5.9215	-136.2743	4.5469	-0.640	1.839	5.508
23	118.0136	-6.0637	-213.1467	-12.3201	0.603	0.959	5.333
24	118.1016	-7.7699	-213.4219	-15.9637	0.712	0.854	5.363
25	193.8918	-14.6782	-121.6950	-9.0410	0.205	1.929	5.323
26	193.9693	-19.5286	-121.8892	-11.9340	0.181	1.852	5.347
RMS					0.522	1.230	5.779
Left Panorama Id: onMqoFLCjTFsReilyMkPQA							
WGS84 (λ, ϕ, h)			24.120254		120.674381		63.118
TWD97 (X,Y)			216901.740		2668379.451		(from Proj4js)
Adjusted (X,Y, h)			216901.914		2668378.353		57.460
Std. Dev. (X,Y, h)			0.135		0.279		0.796
$(\Delta X, \Delta Y, \Delta h)$			-0.174		1.098		5.658
Right Panorama Id: TraJlqq6CmkjThOLjkd7WA							
WGS84 (λ, ϕ, h)			24.120255		120.674477		63.563
TWD97 (X,Y)			216911.498		2668379.539		(from Proj4js)
Adjusted (X,Y, h)			216911.599		2668377.629		57.756
Std. Dev. (X,Y, h)			0.068		0.095		0.296
$(\Delta X, \Delta Y, \Delta h)$			-0.101		1.910		5.807
RMS $(\Delta X, \Delta Y, \Delta h)$			0.142		1.558		5.733

Table 1. Observations and errors in GSV positioning on the NCHU Library Building

4.2 Error Sources in GSV Positioning

Errors in POI's coordinates from geometric intersection among GSV panoramas may be caused by the following sources:

1. GSV car positioning errors: According to the specification, the worst position errors of the TOPCON IP-S2 GPS/IMU unit for 30 second outage duration (1 epoch/30s) are 0.055m and 0.030m in 2D plane and height direction, respectively and attitude errors of 0.030, 0.030, and 0.075 degrees in roll, pitch, and heading, respectively. The latitude, longitude, and elevation parameters of the GSV panorama may include systematic errors from interpolation. As shown in Table 1, the errors in Google provided GSV positional parameters dominated the errors in intersection, especially in elevation.
2. Angular resolution in panoramas: As already shown in Eq. (1), the conversion from rectangular image coordinates in level 5 zooming panorama into heading and pitch angles is about 1.63 minutes/pixel. The angular resolution will be decreased in the power of 2 when the panorama is zoomed out to lower levels, i.e., 3.25 minutes/pixel for level 4 and 6.5 minutes/pixel for level 3, etc. As a result, when the POI is not carefully located with the highest zoom level of the panorama, the heading and pitch angles of conjugate POI images from the respective panorama center may not be precise enough for deriving reliable results from intersection and resection.
3. Distortions of the panoramas: A sphere model was used for merging GSV panoramas from the planar images obtained the horizontal cameras in earlier products. On the other hand, a rectangular plane model is used in the latest Google Maps API version 3. However, the previous GSV products were not regenerated into the rectangle plane model. The panorama may include severe distortions in the upper portions (image from the 9th camera shooting the sky) and lower portions (rendered from other panoramas) as shown in Figure 7. It would be good to locate POIs on the central portions of ± 30 degrees in pitch (vertical angle) according to the vertical field of view of the camera lens. The distortion of image may also happen while the GSV panorama is viewed in different web browsers as shown in Figure 8, in which Google Chrome seems well in reducing the GSV panoramic distortion.
4. Small intersection angles between the known points: The intersection angle of a POI from the two adjacent GSV panoramas may be too sharp to cause significant error in intersection (Ghilani & Wolf, 2008).

5. CONCLUSIONS

This research employs Google Maps JavaScript API and Web Service, GAE for JAVA, AJAX, Proj4js, CSS and HTML in developing an internet platform for accessing the position and orientation parameters of GSV panoramas in order to determine the global positions of POI features that appear on two overlapping images by intersection, and to determine the position of the exposure center of a panorama from at least three known points by resection. Extracted 3-D positional information of the features from intersection were packed in KML format and stored in GAE Servlet for future LBS applications integrated with Google Maps and Google Earth.

Experimental results from positioning known points in a pair of GSV panoramas are $\pm 0.522\text{m}$, $\pm 1.230\text{m}$, and $\pm 5.779\text{m}$ for intersection and $\pm 0.142\text{m}$, $\pm 1.558\text{m}$, and $\pm 5.381\text{m}$ for resection in X, Y, and Z, respectively. Potential error sources in GSV positioning were analyzed and illustrated that the errors in Google provided GSV positional parameters dominate the errors

in geometric intersection.

The developed system is suitable for volumetric data collection in establishing LBS applications, in which the positional accuracy is not primarily concerned. Possible applications can be settled in finding and positioning of commercial entities, traffic signs, utility poles and manholes, landmarks, and infrastructures along the streets where GSV panoramas are available. The current state of the GSV positioning system is not complete. Future work on extending its functionalities includes adding automatic feature detection and image matching techniques for object recognition and matching among GSV panoramas. Other features in GSV auxiliary data and Google Maps can also be included to provide the user opportunities to place markers and overlay in the scene, find local business from commercial entity recognition, and lever 3D data and sequence of GSV panoramas for smart navigation.

ACKNOWLEDGEMENTS

The support from the National Science Council, Taiwan, under Grant No. NSC100-2221-E005-075-MY2, was appreciated in improving the developed system in this research.

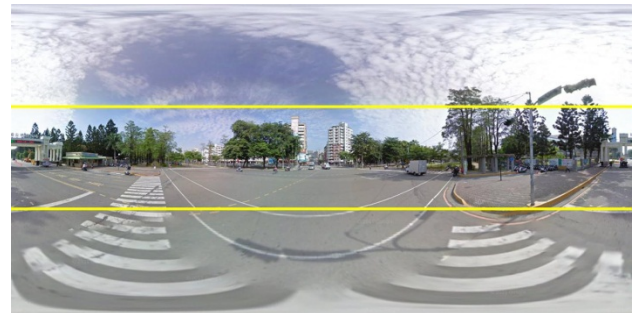


Figure 7. Distortions in the Google Street View Panorama



(A) Google Chrome



(B) Mozilla Firefox4

Figure 8. Distortion of GSV panorama in different web browsers

REFERENCES

- Anguelov, D., C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weave, 2010. "Google Street View: Capturing the World at Street Level," *Computer*, vol. 43, no. 6, pp. 32-38.
- Evenden, G. I., 2003. *Cartographic Projection Procedures for the UNIX Environment—A User's Manual*, USGS Open-File Report 90-284.
- Ghilani, C. D. and P. R. Wolf, 2006. *Adjustment Computations: Spatial Data Analysis*, 4th ed., John Wiley & Sons, Hoboken, NJ, pp. 255-282.
- Ghilani, C. D. and P. R. Wolf, 2008. *Elementary Surveying: An Introduction to Geomatics*, 12th ed., Pearson Education, Upper Saddle River, NJ, pp. 269-294.
- OGP Geomatics, 2012. EPSG Geodetic Parameter Dataset, <http://www.epsg.org/> (11 Jan. 2012).
- Proj4js Track and Wiki, 2012. <http://trac.osgeo.org/proj4js/> (08 Jan. 2012)
- Vincent, L., 2007. "Taking online maps down to street level," *Computer*, vol. 40, no. 12, pp. 118-120.