Published in IET Image Processing
Received on 11th June 2012
Revised on 9th November 2012
Accepted on 4th December 2012
doi: 10.1049/iet-ipr.2012.0323

ISSN 1751-9659

Three-dimensional positioning from Google street view panoramas

Victor J.D. Tsai, Chun-Ting Chang

Department of Civil Engineering, National Chung Hsing University, Taichung 40227, Taiwan
E-mail: jdtsai@nchu.edu.tw

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, cascading style sheets and HyperText markup language in developing a platform for accessing the orientation parameters of Google street view (GSV) panoramas in order to determine the three-dimensional (3D) position of points of interest (POIs) by intersection between two nearby GSV images and to validate the 3D position of a GSV panorama by resection from three known POIs. Extracted 3D positional information of the features are then packed in keyhole markup language format and stored in GAE Servlet for future LBS applications integrated with Google Maps and Google Earth. Experimental results from two tests on intersection and one test on resection of GSV panoramas in National Chung Hsing University campus were examined with error reports and source analyses. With automatic conjugate image-matching capability, the developed system is suitable for volumetric data collection in establishing web-based LBS applications integrated with GSV panoramas and Google Maps/Earth in which positional accuracy is not primarily concerned.

1 Introduction

The emerging development in microelectronic industry has raised the productivity and performance in mobile mapping devices and 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 May 2007 [1]. With Google street view (GSV) opened to internet communities and user contributions [2], many researches have been reported on various applications over GSV images, including automatic detection and blurring of faces and license plates for privacy protection [3], removal of pedestrians [4], secret detail recovery using visual cryptography scheme [5], visual three-dimensional (3D) city modelling based on structure-from-motion pipeline [6, 7], feature distinguish in a multi-view semantic segmentation framework [8], identification of commercial entities [9], generation of guiding video for interactive route recognition to car drivers [10] and tourism support through geographic Web 2.0-based community [11]. Consequently, Google Earth/Maps with street view images is becoming a powerful and popular service for chasing domination in the Internet mapping and searching against Microsoft Virtual Earth [12] with Google's continuing efforts in exploring new interfaces, finding better ways to integrate more user-contributed photos and developing platforms to extend coverage [2].

In custom panoramic camera system dubbed R5, GSV images were taken from a ring of eight cameras plus a fish-eye lens on top for producing the most popular 360° panoramic views [2]. For the fourth generation GSV car, as shown in Fig. 1, the R5 system was mounted at a height of about 2.5 m on a custom-hinged mast along with global positioning systems (GPS) units for vehicle positioning, three laser range scanners for the capture of coarse 3D data alongside the imagery in the front and two sides of the vehicle and network antennas for scanning nearby 3G/GSM and Wi-Fi hotspots. Each available GSV panorama can be requested in an HTTP URL form using Google Maps API, along with the projection type, the geodetic position of the GSV car and its moving direction with respect to the North at the time of image capture. Hence, it provides the user opportunities to find local business, lever 3D data for smart navigation, compute 3D models from laser data, place markers and overlay in the scene, or determine the spatial position of features that appear on the GSV panoramas.

Since GSV panoramas are associated with geodetic information and orientation parameters, this research attempts to apply geometry in determining 3D positions of the points of interest (POIs) that appear on two nearby panoramas and in verifying the provided GSV parameters. The goal of this research is to employ Google Maps JavaScript API [13], Google App Engine (GAE) for JAVA, Asynchronous JavaScript and XML (AJAX) and Proj4js [14] in developing an internet platform for two tasks: (i) accessing the orientation parameters of GSV panoramas in order to determine the 3D positions of POIs that appear on two panoramas by intersection and (ii) verifying the accuracy of the provided position of a GSV panorama by resection from three features with known 3D coordinates. As results, extracted 3D positional information of the POIs

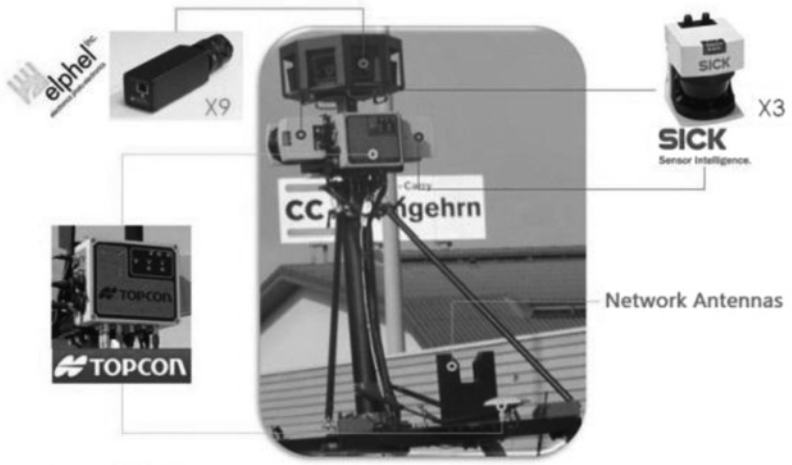

Fig. 1 *Devices adapted on the fourth generation GSV car*
(central image edited from http://en.wikipedia.org/wiki/ File: Google_Street_View_car_Switzerland.jpg)

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 system were introduced in the next two sections, followed by the experimental results and error analyses from examining known control points appeared on the panoramas.

2 GSV and maps API

2.1 Basics for GSV panorama

Although with composing techniques been veiled in a pipeline of complex algorithms [1], a GSV panorama is a 360° surrounding image generated from the eight original images captured by the eight horizontal cameras by stitching together in the sequences as shown in Fig. 2. The percentage of horizontal overlap between adjacent original images is about 28.8% for removing the most distorted outside portion of an image caused by camera lens. Meanwhile, the GSV panorama was always aligned with the heading direction of the vehicle at the centre and associated with its deviation from the North after a trajectory pose optimisation process based on sensor fusion [2].

After studying on a series of GSV images, we discovered that each GSV panorama was tile indexed in a 6 zoom-level image pyramid with the size of $(416{\cdot}2^{\text{zoom}} \times 208{\cdot}2^{\text{zoom}})$ pixels, that is, from 416×208 pixels in zoom-level 0, with a scale factor of 2 to $13\,312 \times 6656$ pixels in zoom-level 5, for covering the field of view of 360° horizontal and 180° vertical. As shown in Fig. 3, it is convenient to convert the rectangular image coordinates (row, column) as (r_P, c_P) of a pixel P in the developed panoramic plane into clockwise spherical coordinates (pitch, heading) as (γ_P, θ_P) in the following equivalents

$$\begin{cases} \gamma_P = (208 \cdot 2^{\text{zoom}-1} - r_P) \cdot 360^\circ / (208 \times 2^{\text{zoom}}) \\ \theta_P = (c_P - 416 \cdot 2^{\text{zoom}-1}) \cdot 360^\circ / (416 \times 2^{\text{zoom}}) \end{cases} \quad (1)$$

in which γ_P(pitch) represents the vertical angle of the pixel P from the horizontal plane of the camera lens centre and θ_P(heading) represents the horizontal viewing angle from the car heading direction which deviates ϕ (pano_yaw)

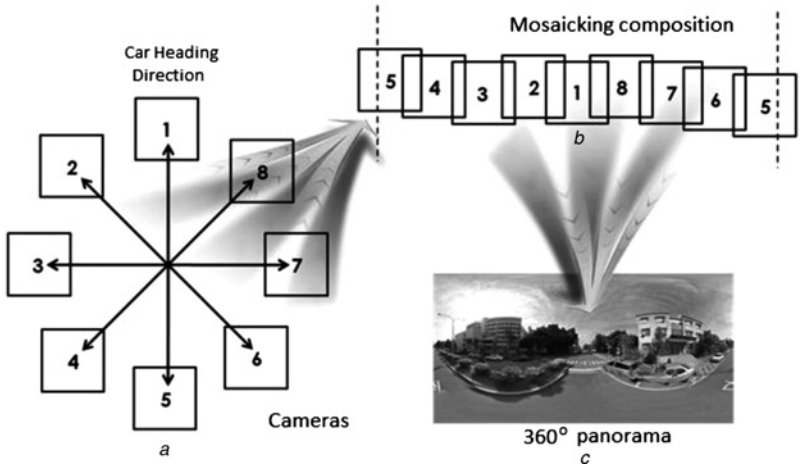

Fig. 2 *Image composition of a GSV panorama*
a Cameras
b Mosaicking composition
c 360° panorama

doi: 10.1049/iet-ipr.2012.0323

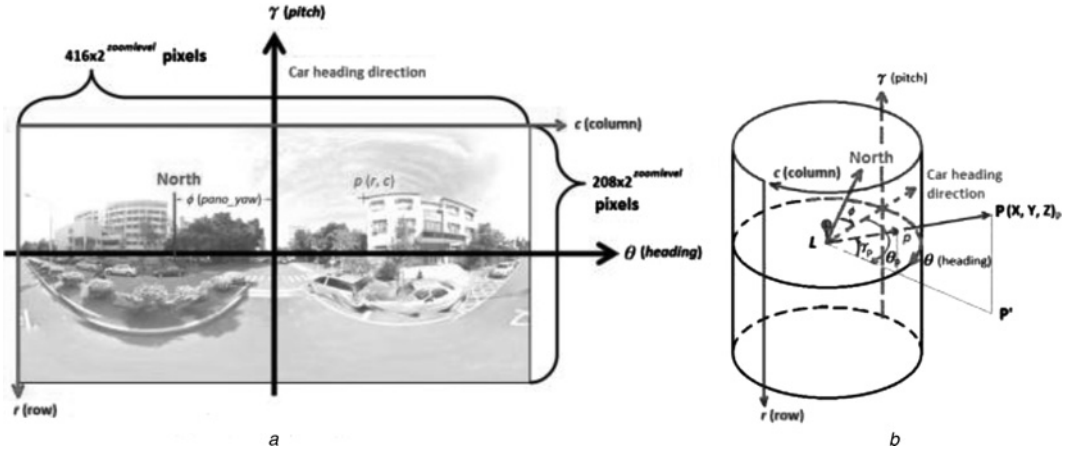

Fig. 3 *Coordinates applied in GSV images*

a Developed GSV panorama
b Spherical coordinates

degrees from the North. Thus, the azimuth of the pixel P from the exposure centre of the GSV panorama is

$$\phi_P = \phi + \theta_P \tag{2}$$

2.2 Parameters accessed from Google maps API

The coordinates associated with the panorama are latitude and longitude in WGS84 reference ellipsoid for its centre and can be accessed from Google Maps JavaScript API [13] using the getLatLng() method in StreetviewLocation class. The car heading direction with respect to true north, that is, pano_yaw (ϕ) in Fig. 3, can also be requested by using the getYaw() method for camera 1 in Pov class, which can be accessed by the getPov() method in StreetviewLocation class for the initial point of view. 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, for example, google.maps.ElevationService().

When being pointed with the cursor in a zoomed panorama, a feature or POI can be located from the heading (θ_P) (or azimuth in (2)) and pitch (γ_P) attributes of the StreetviewPov class as in 1 and Fig. 3, respectively. The POI may be a commercial entity, a traffic sign, or a specific target to be determined with its location in 3D space by intersection from two overlapped GSV panoramas. It may be a labelled point already posted in the GAE Servlet with its coordinates accessed by StreetviewLocation class for computing the position of the exposure centre of a panorama by resection from at least three known points.

3 JavaScript Implementation

3.1 Object coordinate systems and transformations

The coordinates used in the implementation include WGS84 geodetic coordinates (latitude, longitude, elevation or height from the ellipsoid) in WGS84 ellipsoid (EPSG: 4326), spherical Mercator projection for interactive web map publishing (EPSG: 900913), TWD97 Transverse Mercator 2° zone 121 X–Y coordinates based on GRS80 ellipsoid and International Terrestrial Reference Frame 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 with EPSG geodetic parameters in a JavaScript environment [14–16]. The following codes in Fig. 4 show an example of the implementation of transformation from WGS84 geodetic coordinates into TWD97 X–Y coordinates using Proj4js:

```
// Define parameters for the source and destination datums
Proj4js.defs["EPSG:4326"] ="+title=WGS 84 +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs ";
Proj4js.defs["EPSG:3826"]="+title=TWD97 TM2 zone 121 +proj=tmerc +lat_0=0 +lon_0=121 +k=0.9999
+x_0=250000 +y_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs";
var source = new Proj4js.Proj('EPSG:4326');      // source datum with (long, lat) in WGS 84
var dest = new Proj4js.Proj('EPSG:3826');        // destination datum with (x, y) in TWD97/TM2 Zone 121
// Set variables for {Object} point with x, y properties, which can be (long, lat) in wgs84 or (x, y) in twd97
var  point = new Proj4js.Point(120.4506, 24.0050)   // {Object} point with (long, lat) in wgs84
Proj4js.transform(source, dest, point);          // {Object} point with (long, lat) transformed into (x, y)
```

Fig. 4 *Example of the implementation of transformation from WGS84 geodetic coordinates into TWD97 X-Y coordinates using Proj4js*

3.2 Intersection

As shown in Fig. 5, computations of 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. Let $(X, Y, Z)_P$ represent the object coordinates of the POI, $(X, Y, Z)_L$ and $(X, Y, Z)_R$ the exposure centres for the left and right GSV panoramas, respectively, ϕ_{LP} and ϕ_{RP} for the azimuths and γ_{LP} and γ_{RP} for the pitches of the POI viewing from the left centre L and right centre R, respectively. The planimetric coordinates of the POI can then be computed by applying intersection of two lines with known directions [17] using coordinate geometry as following

$$\begin{cases} X_P = X_L + \overline{LP} \cdot \sin \varphi_{LP} \\ Y_P = Y_L + \overline{LP} \cdot \cos \varphi_{LP} \end{cases} \tag{3}$$

Note that

$\overline{LP} = \overline{LR}\big(\sin\beta/(\sin(\alpha+\beta))\big)$, according to the law of Sine in plane geometry

$$\phi_{LP} = \varphi_{LR} - \alpha, \quad \text{that is,} \quad \alpha = \phi_{LR} - \phi_{LP}$$

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

$$\varphi_{LR} = \tan^{-1}\left(\frac{X_R - X_L}{Y_R - Y_L}\right)$$

$$\sin\big(\phi_{LR} - \alpha\big) = \sin\phi_{LR}\cos\alpha - \cos\phi_{LR}\sin\alpha$$

$$\cos\big(\phi_{LR} - \alpha\big) = \cos\phi_{LR}\cos\alpha + \sin\phi_{LR}\sin\alpha$$

$$\sin\phi_{LR} = \frac{X_R - X_L}{\overline{LR}}$$

and

$$\cos\varphi_{LR} = \frac{Y_R - Y_L}{\overline{LR}}$$

Equation (3) can then be rewritten as

$$\begin{cases} X_P = \dfrac{(Y_L - Y_R) + X_L\cot\beta + X_R\cot\alpha}{\cot\alpha + \cot\beta} \\ Y_P = \dfrac{(X_R - X_L) + Y_L\cot\beta + Y_R\cot\alpha}{\cot\alpha + \cot\beta} \end{cases} \tag{4}$$

With the planimetric (X_P, Y_P) coordinates of the POI been solved, the average elevation derived from the two stations is taken for the elevation of the POI as following

$$Z_P = \frac{\left(Z_L + \overline{LP}^* \tan\gamma_{LP}\right) + \left(Z_R + \overline{RP}^* \tan\gamma_{RP}\right)}{2} \tag{5}$$

in which $\overline{LP} = \sqrt{(X_P - X_L)^2 + (Y_P - Y_L)^2}$ and $\overline{RP} = \sqrt{(X_P - X_R)^2 + (Y_P - Y_R)^2}$.

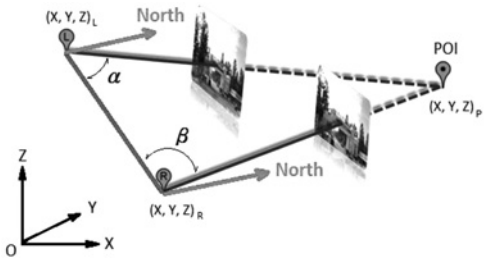

Fig. 5 *Intersection of two known directions in two GSV panoramas*

3.3 Resection

In the GSV panorama, the three-point resection [17] was applied to locate the exposure centre of the panorama in retrieving azimuths and pitches of three labelled points with known coordinates accessed via Google Maps API from the GAE Servlet as described in Section 2. As illustrated in Fig. 6, let A, B, C represent the three labelled points with known 3D coordinates picked by the user in a GSV panorama. We can compute the 3D coordinates of the exposure centre using the following procedures:

1. From the known coordinates of A, B, C calculate lengths $\overline{A'C'}$ and $\overline{B'C'}$ and the angle of $\angle A'C'B'$ at station C'

$$\overline{A'C'} = \sqrt{(X_C - X_A)^2 + (Y_C - Y_A)^2}$$

$$\overline{B'C'} = \sqrt{(X_C - X_B)^2 + (Y_C - Y_B)^2}$$

$$\overline{A'B'} = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}$$

$$\angle A'C'B' = \beta + \delta = \cos^{-1}\left(\frac{\overline{A'C'}^2 + \overline{B'C'}^2 - \overline{A'B'}^2}{2\overline{A'C'}\overline{B'C'}}\right)$$

2. Compute $\Phi = \alpha + \kappa = 360^\circ - (\beta + \delta + \omega + \varphi)$

where $\omega = \phi_A - \phi_C$, $\varphi = \phi_C - \phi_B$ and ϕ_A, ϕ_B, ϕ_C are the accessed azimuths of the object points A, B, C, respectively.

3. Calculate angles α, κ and β, δ using the following

$$\alpha = \tan^{-1}\left(\frac{\overline{B'C'}\ \sin\omega\ \sin\Phi}{\overline{A'C'}\ \sin\varphi + \overline{B'C'}\ \sin\omega\ \cos\Phi}\right)$$

$$\kappa = \tan^{-1}\left(\frac{\overline{A'C'}\ \sin\varphi\ \sin\Phi}{\overline{B'C'}\ \sin\omega + \overline{A'C'}\ \sin\varphi\ \cos\Phi}\right)$$

$$\beta = 180^\circ - \alpha - \omega$$

$$\delta = 180^\circ - \varphi - \kappa$$

4. Solve for lengths $\overline{A'L}$ using the law of sines

$$\overline{A'L} = \overline{A'C'}\frac{\sin\beta}{\sin\omega}$$

5. From the azimuth of $\overrightarrow{A'C}$ calculate the azimuth of $\overrightarrow{A'L}$

$$\phi_{A'L} = \phi_{A'C'} - \alpha$$

in which $\phi_{A'C'} = \tan^{-1}\big((X_C - X_A)/(Y_C - Y_A)\big)$.

doi: 10.1049/iet-ipr.2012.0323

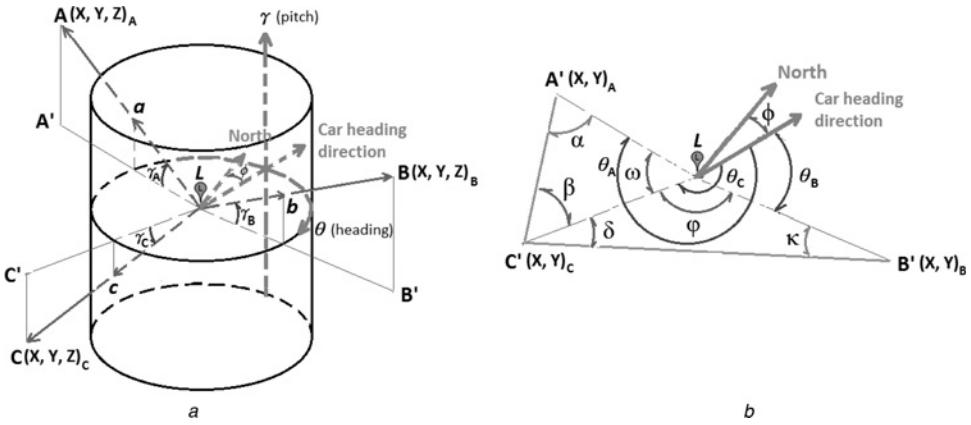

Fig. 6 *Three-point resection in a GSV panorama*

a Spherical coordinates in a GSV panorama
b Three point resection in 2D horizontal plane at L

6. Calculate the two-dimensional (2D) coordinates of the exposure centre L

$$\begin{cases} X_L = X_A + \overline{A'L} \cdot \sin \phi_{A'L} \\ Y_L = Y_A + \overline{A'L} \cdot \cos \phi_{A'L} \end{cases} \tag{6}$$

7. As with intersection, the average elevation derived from the three known points is taken for the elevation of the exposure centre as

$$Z_L = \frac{\left(Z_A - \overline{A'L} \tan \gamma_A\right) + \left(Z_B - \overline{B'L} \tan \gamma_B\right) + \left(Z_C - \overline{C'L} \tan \gamma_C\right)}{3} \tag{7}$$

where γ_A, γ_B, γ_C are the pitches of the object points A, B, C, respectively.

8. Compute the pano_yaw, that is, ϕ in Fig. 6*b* as the deviation of the car heading direction from the North, of the panorama

$$\phi = \phi_{A'L} + 180^\circ - \theta_A \tag{8}$$

3.4 System implementation

The internet platform was developed using AJAX technique with cascading style sheets (CSS) and HyperText markup language (HTML) in web user interface and JavaScript in data communication between program sessions and GAE Servlet through Internet. The developed platform can be run with Microsoft Internet Explorer (IE), Mozilla Firefox and Google Chrome web browsers. In Microsoft IE8, Fig. 7

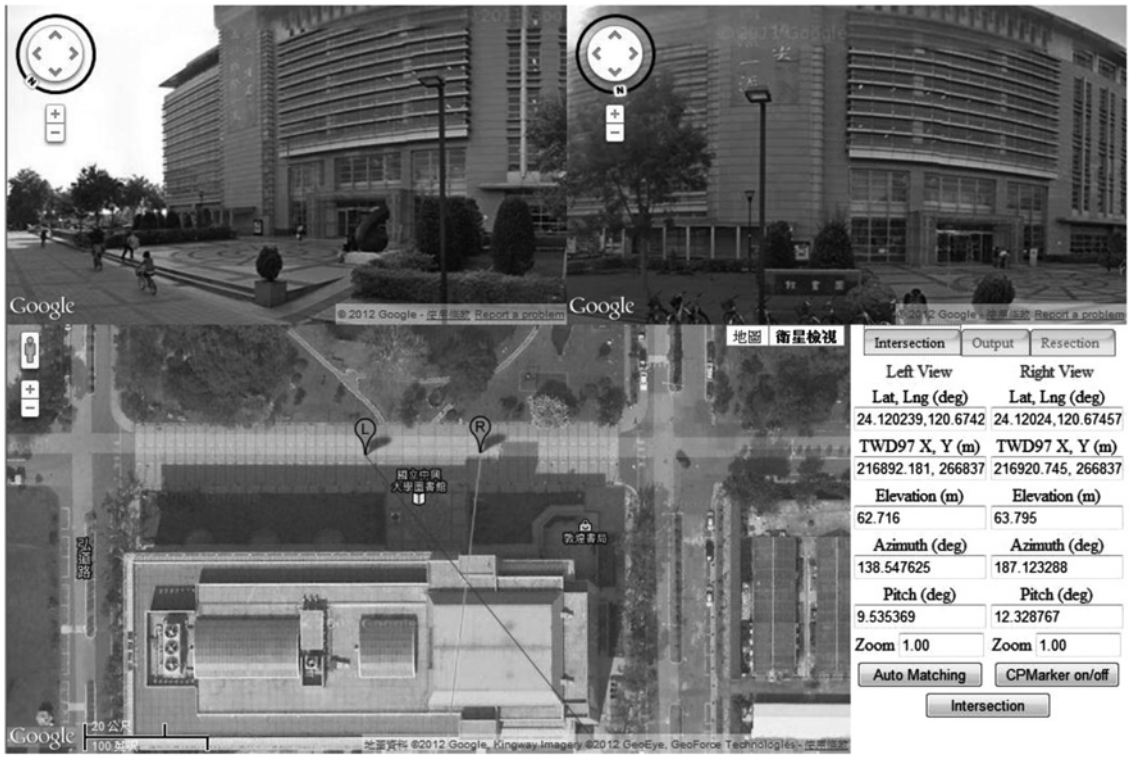

Fig. 7 *GUI screenshot of the implementation*

demonstrates the GUI illustration of the implemented platform in which the window content is divided into five parts: left view panorama, right view panorama, Google Maps, Main menu and Log for calculated POI coordinates. The two GSV panorama canvases and Maps canvas allow the user to zoom in/out the panoramas and map/image for detail, respectively. Once clicked on a cursor position in the Maps canvas, the nearest positions of found pair of GSV panoramas via Google Maps API will be marked as *L* and *R*, each with a line of sight for POI intersection and the panoramas will be shown in the two upper GSV image canvases, each centred with a cross cursor pointing to the respective pixel for the POI.

As shown in Fig. 8, the Main menu currently includes three processes: intersection, output and resection. Under the intersection process, the information of the two nearby panoramas and the azimuths and pitches of the cursor centre from the centre of respective panorama are dynamically updated during each movement of manually pointing cursor and zooming of panoramas, respectively. The 'Auto Matching' button provides automatic image matching capability via normalised cross-correlation (NCC) approach [18] in finding conjugate right-panorama POI image (azimuth and pitch) with respect to a 51×51 reference image centred at the cursor on the left panorama. The HTML5 canvas getImageData() method was applied to access the image data at specified rectangular blocks on both GSV panoramas, within which the colour images were converted into grey images for use in NCC matching. The user may interactively zoom in and drag the GSV panoramas to precisely locate the conjugate images. Concurrently, the 'CPMarker on/off' button offers an option to turn on/off the markers for control points on National Chung Hsing University (NCHU) campus roads for error analysis. Once the 'intersection' button was clicked, the solved POI coordinates in WGS84, TWD97 and TWD67 were output to the Log viewport and the output process.

In the output process, user-input descriptive information about the POI along with the 3D positional information from the intersection process can be saved to GAE Servlet in KML document for future LBS applications by clicking the 'Save POI' button. Finally, in the resection process, while three known feature points were caught individually by dragging cursor and zooming on the left view panorama and clicking the 'Catch' buttons, the 'Resection' button should then be clicked to compute and deliver the 3D position of the exposure centre of the left view panorama to the Log viewport.

4 Experimental results and error analyses

4.1 Positioning results from intersection

The developed platform was firstly examined using 17 control points located on the campus roads of NCHU, as shown in Fig. 9 with a clicked pop-up window showing coordinates of a control point, by comparing their coordinates derived from intersection between a pair of nearby GSV panoramas with their known coordinates obtained by GPS static surveying. The root-mean-squared errors (RMSEs) of the 17 control points are ± 2.358 m, ± 2.981 m and ± 6.724 m in *X*, *Y* and *Z* directions, respectively as shown in Table 1. In 95% confidence level, the RMSEs of the remained 14 control points are ± 1.652 m, ± 3.263 m and ± 6.931 m in *X*, *Y* and *Z* directions, respectively.

The second experiment was conducted on positioning POIs on one façade of a building and on verifying provided GSV parameters. Two nearby GSV panoramas were examined using 12 known feature points located on the NCHU Library Building, as shown in Fig. 10, by comparing their coordinates derived from intersection in a pair of GSV panoramas with their known coordinates surveyed by total station instrument from the control points. The RMSEs of the 12 feature points are ± 0.526 m,

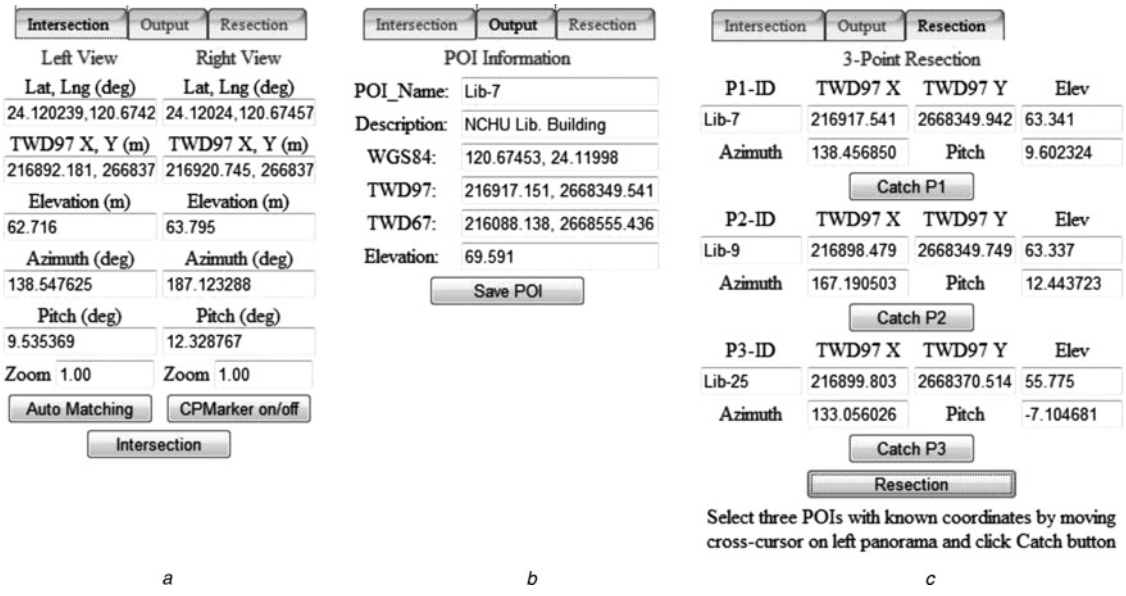

Fig. 8 *Main processes of the implementation*

a Intersection
b Output
c Resection

doi: 10.1049/iet-ipr.2012.0323

Fig. 9 *Distribution of control points on NCHU campus roads*

Table 1 Errors of intersection from two GSV panoramas for the control points on NCHU campus roads

CPID	Coordinates from GSV intersection			Coordinates from GPS static surveying			Differences (*m*)		
	TWD97 *X*	TWD97 *Y*	*Z*	TWD97 *X*	TWD97 *Y*	*Z*	Δ*X*	Δ*Y*	Δ*Z*
A02	217262.523	2668687.589	62.829	217263.168	2668685.827	57.868	– 0.645	1.762	4.961
A03	217263.172	2668511.075	63.538	217262.939	2668507.517	56.164	0.233	3.558	7.374
A04	217359.446	2668508.671	62.250	217359.079	2668507.474	56.830	0.367	1.197	5.420
A05	217262.302	2668419.176	63.387	217262.805	2668421.307	55.409	– 0.503	– 2.131	7.978
A06	217262.974	2668373.351	61.244	217261.903	2668375.276	55.167	1.071	– 1.925	6.077
A07	217139.835	2668382.161	62.354	217140.950	2668376.173	55.019	– 1.115	5.988	7.335
A08	217138.370	2668505.576	64.646	217139.625	2668508.199	56.355	– 1.255	– 2.623	8.291
A09	217139.111	2668577.894	62.218	217139.667	2668573.157	56.855	– 0.556	4.737	5.363
A10	217138.591	2668661.483	64.164	217139.935	2668656.648	57.708	– 1.344	4.835	6.456
A11	216965.700	2668656.321	65.584	216968.162	2668651.432	57.222	– 2.462	4.889	8.362
A12	216963.025	2668510.946	67.032	216967.911	2668508.350	56.189	– 4.886	2.596	10.843
A13[a]	216961.462	2668376.438	64.684	216967.450	2668376.406	55.320	– 5.988	0.032	9.364
A14[a]	216970.604	2668230.218	56.676	216966.441	2668231.638	53.791	4.163	– 1.420	2.885
A16[a]	216820.551	2668234.976	53.047	216822.306	2668234.989	53.386	– 1.755	– 0.013	– 0.339
A17	216822.569	2668374.942	58.410	216822.583	2668376.469	54.433	– 0.014	– 1.527	3.977
A18	216824.202	2668509.125	60.996	216823.064	2668508.741	55.624	1.138	0.384	5.372
A19	216823.336	2668649.755	62.264	216823.092	2668651.148	56.062	0.244	– 1.393	6.202
mean							– 0.783	1.114	6.231
standard deviation							2.292	2.850	2.607
RMSE							2.358	2.981	6.724
95% confidence level RMSE							1.652	3.263	6.931

[a]Clipped for 95% confidence level

Fig. 10 *Known POIs on the NCHU Library Building*

± 1.207 m and ± 5.817 m in *X*, *Y* and *Z*-directions, respectively, as shown in Table 2. Note that these known feature points appear on the GSV panoramas with their pitch angles within [– 30°, + 30°] for distinct pixel identification without image blurring.

Another test on the positioning precision of intersection for the same feature points on NCHU Library Building was done by applying the ADJUST program in a least squares approach [19] over the azimuth (horizontal angle) and pitch (vertical angle) observations on the five adjacent GSV panoramas as shown in Fig. 11 in zoom levels 3, 4 and 5, respectively. Table 3 shows the RMSEs from the least squares solutions of the 12 feature points identified in different zoom levels. It is shown that the magnitude of the RMSEs from all five GSV panoramas in different zoom level is almost the same as that from the intersection between a pair of GSV panoramas.

Table 2 Errors of intersection from two GSV panoramas for the known POIs on the NCHU Library Building

Point ID	Left panorama (*C*)		Right panorama (*D*)		Coordinate differences (*m*)		
	Heading	Pitch	Heading	Pitch	ΔX	ΔY	ΔZ
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
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
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
RMSE					0.526	1.207	5.817

Fig. 11 *Positions of the five GSV panoramas against NCHU Library*

4.2 Positioning errors from resection for the GSV panoramas

A test on resection for the two GSV panoramas was done using seven (i.e. points 2, 5, 7, 9, 18, 23 and 25) off the 12 known points on the NCHU Library Building to determine the GSV positional parameters independently using the ADJUST program [19]. As listed in Table 4, the RMS errors of Google provided GSV positions are ± 0.142 m, ± 1.558 m and ± 5.733 m in TWD97 *X*, *Y* and *Z*, respectively, for the two GSV panoramas.

Similarly, the observations (heading and pitch) of the seven control points on the five GSV panoramas in different zoom levels were used to solve for the positional parameters of the five GSV panoramas using the ADJUST program [19]. It is shown in Table 5 that the magnitude of the RMS errors from all five GSV panoramas in different zoom levels is nearly the same magnitude as that from the resection for an individual GSV panorama.

4.3 Error analyses in GSV panorama intersection

Errors in intersection among GSV panoramas may be caused by the following sources:

Table 3 RMSEs of intersection among the five GSV panoramas in different zoom levels

Zoom level	Number of known points	Number of observations[a]	RMSE (*m*)		
			ΔX	ΔY	ΔZ
3	12	102	0.459	1.185	5.874
4	12	102	0.491	1.210	5.831
5	12	102	0.472	1.344	5.881
3 + 4 + 5	12	306	0.456	1.191	5.851

[a] Each observation includes azimuth and pitch of a POI on a GSV panorama.

Table 4 Errors of resection for the exposure centres of a pair of GSV panoramas

Left panorama PanoID: onMqoFLCjTFsReilyMkPQA			
WGS84 (λ, φ, Z)	24.120254	120.674381	63.118
TWD97 (*X*, *Y*)	216901.740	2668379.451	(from Proj4js)
adjusted (*X*, *Y*, *Z*)	216901.914	2668378.353	57.460
std. Dev. (*X*, *Y*, *Z*)	0.135	0.279	0.796
(ΔX, ΔY, ΔZ)	– 0.174	1.098	5.658
Right panorama PanoID: TrAJlgq6CmkjThOLjkD7WA			
WGS84 (λ, φ, *Z*)	24.120255	120.674477	63.563
TWD97 (*X*, *Y*)	216911.498	2668379.539	(from Proj4js)
adjusted (*X*, *Y*, *Z*)	216911.599	2668377.629	57.756
std. Dev. (*X*, *Y*, *Z*)	0.068	0.095	0.296
(ΔX, ΔY, ΔZ)	– 0.101	1.910	5.807
RMS (ΔX, ΔY, ΔZ)	0.142	1.558	5.733

doi: 10.1049/iet-ipr.2012.0323

Table 5 Errors of resection from 7 known POIs for the five GSV panoramas in different zoom levels

GSV Pano	Coordinates by Proj4js from GSV API			Coordinates from resection			Differences (*m*)		
	TWD97 *X*	TWD97 *Y*	*Z*	TWD97 *X*	TWD97 *Y*	*Z*	Δ*X*	Δ*Y*	Δ*Z*
A	216921.053	2668379.517	63.914	216901.900	2668378.304	57.393	0.160	– 1.147	– 5.725
B	216911.397	2668379.539	63.353	216910.001	2668380.295	57.332	– 1.396	0.756	– 6.231
C	216901.704	2668379.451	63.118	216921.348	2668377.632	57.588	0.295	– 1.885	– 6.326
D	216891.880	2668379.474	62.731	216892.565	2668377.619	57.017	0.685	– 1.855	– 5.714
E	216881.410	2668379.498	62.343	216882.765	2668378.000	56.647	1.355	– 1.498	– 5.696
RMSE (Zoom levels: 3 + 4 + 5, 186 observations[a])							0.935	1.492	5.945
RMSE (Zoom level: 3, 62 observations[a])							0.925	1.486	5.948
RMSE (Zoom level: 4, 62 observations[a])							0.943	1.502	5.938
RMSE (Zoom level: 5, 62 observations[a])							0.915	1.497	5.959

[a]Each observation includes azimuth and pitch of a POI on a GSV panorama.

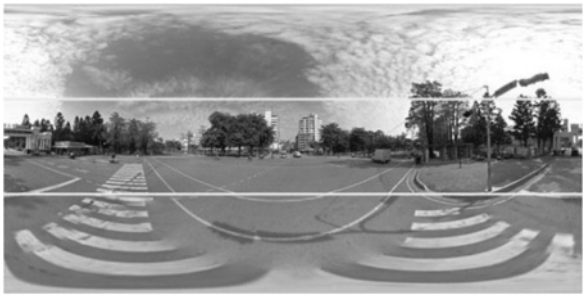

Fig. 12 *Distortions in the GSV Panorama*

1. GSV car positioning errors: According to the specification, the worst position errors of the TOPCON IP-S2 GPS/IMU unit for 30 s outage duration (1 epoch/30 s) are 0.055 m and 0.030 m in 2D plane and height direction, respectively, and attitude errors of 0.030, 0.030 and 0.075° in roll, pitch and heading, respectively. However, the latitude/longitude/yaw parameters accessed by StreeviewLocation class and elevation accessed by Google Elevation API of the panorama may include systematic errors from interpolation. As shown in Tables 4 and 5, the errors in Google provided GSV positional parameters dominated the errors in intersection, especially in elevation.
2. Angular resolution in GSV panoramas: as already shown in (1) and Fig. 3, the conversion from rectangular image coordinates in level 5 zooming panorama into heading and pitch angles is about 1.63 min/pixel. The angular resolution will be decreased in the power of 2 when the panorama is zoomed out to lower levels, that is, 3.25 min/pixel for level 4 and 6.5 min/pixel for level 3 etc. As a result, it is supposed that when the POI is not identified with the highest zoom level of the panorama would not deliver reliable results for intersection and resection. However, our experimental results in Table 3 show that identifying conjugate POI images in different zoom levels is irrelevant to the promotion of positioning precision.
3. Distortions of the panoramas: a spherical model was used for merging GSV panoramas from the planar images captured by 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 ninth camera shooting the sky) and lower portions (rendered from other panoramas) as shown in Fig. 12. It would be good to locate POIs on the central portions of ± 30° in pitch (vertical angle) according to the vertical field of view of the camera lens. The distortion of image may happen while the GSV panoramas are viewed in four different web browsers as shown in Fig. 13, in which Google Chrome and Microsoft IE9 seem well with less GSV panoramic distortions. (Note that Microsoft IE8 does not support HTML5 canvas object, the 'Auto Matching' in the intersection process does not apply in finding conjugate POI position in the right panorama.) However, an experiment on the observations of the same feature point 7 with the different browsers shows no significant difference in errors from intersection as illustrated in Table 6.
4. Small intersection angles of the control points: The intersection angle of a POI from the adjacent GSV panoramas may be too sharp to cause significant error in intersection [17] as shown in Table 1, in which the control points on the NCHU campus roads with the lines of sight from the GSV exposure centres almost parallel to the heading direction of GSV car.

5 Conclusions

This research employs Google Maps JavaScript API and Web Service, GAE for JAVA, AJAX, Proj4js, CSS and HTML5 in developing an internet platform for accessing the orientation parameters of GSV panoramas in order to determine the global position of POI features by intersection from two overlapping images and to validate the 3D position of a GSV panorama by resection from three known control points. Extracted positional information of the POI features from intersection are packed in KML format and stored in GAE Servlet for future LBS applications integrated with Google Maps and Google Earth. Experimental results from three tests on positioning known points in nearby GSV panoramas by intersection and a test from determining the position of a GSV panorama by resection were demonstrated. Potential error sources in GSV positioning were also analysed 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 traffic signs, commercial entities, utility poles and manholes, landmarks and infrastructures along the streets

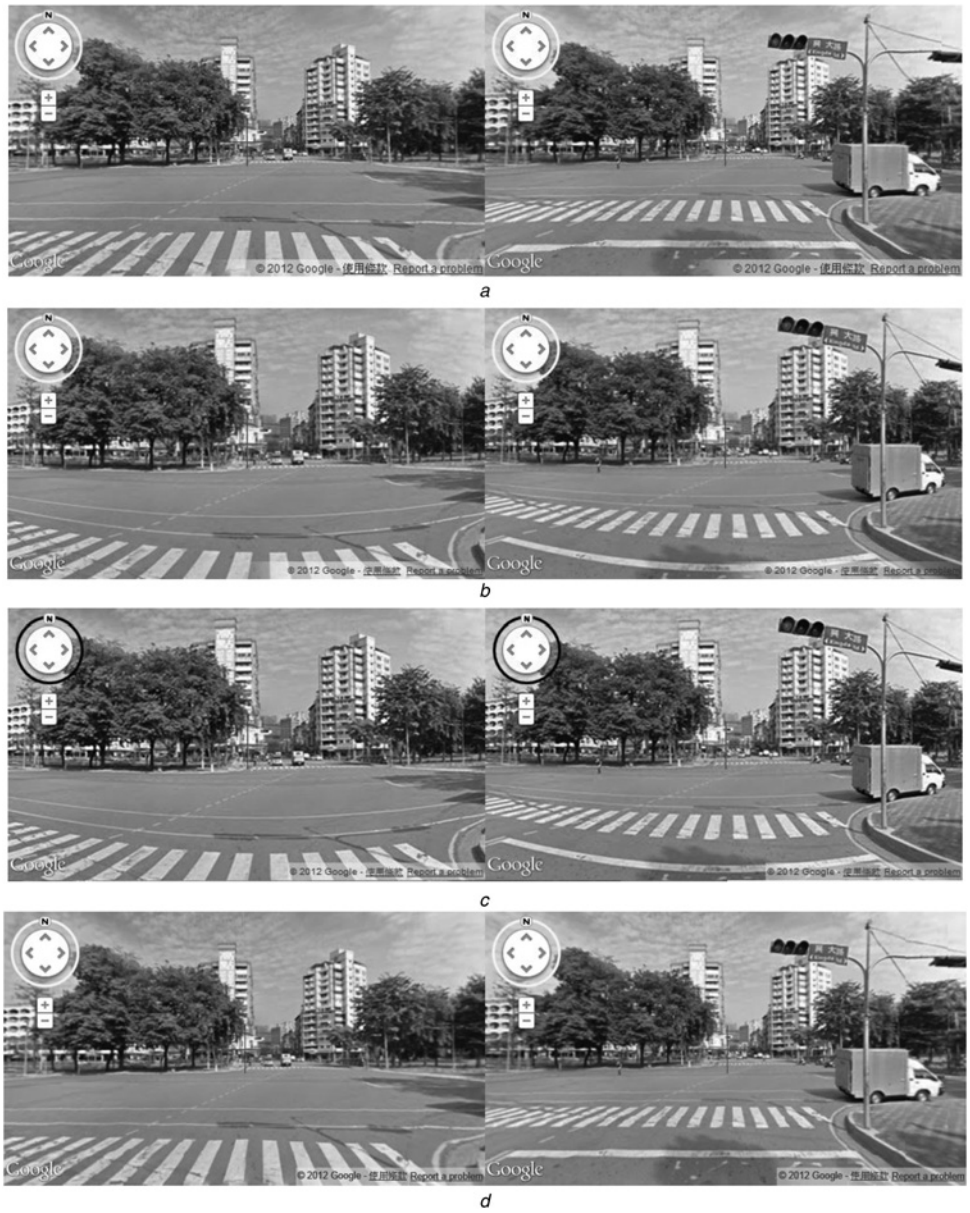

Fig. 13 *Distortions of the GSV panoramas in different web browsers*

a Google Chrome 23.0.1271.64
b Mozilla Firefox 16.0.2
c Microsoft Internet Explorer (IE) 8
d Microsoft Internet Explorer (IE) 9

Table 6 Observations and errors in different browsers for intersection of the same POI (7) in zoom level 4

Browser	Left panorama (*C*)		Right panorama (*D*)		Coordinate differences (*m*)		
	Heading	Pitch	Heading	Pitch	ΔX	ΔY	ΔZ
IE8	167.495319	11.610795	187.578512	11.993033	– 0.319	1.876	4.884
IE9	167.498525	11.630848	187.512440	12.003522	– 0.300	1.783	5.910
Chrome	167.510306	11.637743	187.490591	11.985256	– 0.296	1.738	5.917
Firefox	167.498859	11.576944	187.567438	11.983546	– 0.316	1.856	5.877

where GSV panoramas are available. Future work on extending the functionalities of the developed system includes adding feature detection techniques for specific object recognition among GSV panoramas. For example, a city-wide traffic sign inventory system with automatic traffic sign recognition is concerned for establishing a web-based geographic information system application over GSV images and Google Maps. 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

doi: 10.1049/iet-ipr.2012.0323

scene, find local business from commercial entity recognition and lever 3D data and sequence of GSV panoramas for smart navigation.

6 Acknowledgments

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

7 References

1 Vincent, L.: 'Taking online maps down to street level', *Computer*, 2007, **40**, (12), pp. 118–120
2 Anguelov, D., Dulong, C., Filip, D., *et al.*: 'Google Street View: Capturing the World at Street Level', *Computer*, 2010, **43**, (6), pp. 32–38
3 Frome, A., Cheung, G., Abdulkader, A., *et al.*: 'Large-scale privacy protection in Google Street View'. IEEE 12th Int. Conf. on Computer Vision, Kyoto, Japan, 2009, pp. 2373–2380
4 Flores, A., Belongie, S.: 'Removing pedestrians from Google street view images'. 2010 IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), San Francisco, CA, June, 2010, pp. 53–58
5 Weir, J., Yan, W.: 'Resolution variant visual cryptography for Street View of Google Maps'. IEEE Int. Symp. on Circuits and Systems, Paris, France, 2010, pp. 1695–1698
6 Torii, A., Havlena, M., Pajdla, T.: 'From Google Street View to 3D city models'. IEEE 12th Int. Conf. on Computer Vision Workshops, Kyoto, Japan, 2009, pp. 2188–2195
7 Sato, T., Pajdla, T., Yokoya, N.: 'Epipolar geometry estimation for wide-baseline omnidirectional Street View images'. IEEE Int. Conf. on Computer Vision Workshops, Barcelona, Spain, Nov., 2011, pp. 56–63
8 Xiao, H., Quan, L.: 'Multiple view semantic segmentation for Street View images'. IEEE 12th Int. Conf. on Computer Vision, Kyoto, Japan, 2009, pp. 686–693
9 Zamir, A.R., Darino, A., Patrick, R., Shah, M.: 'Street View challenge: Identification of commercial entities in Street View imagery'. Tenth Int. Conf. on Machine Learning and Applications, Honolulu, Hawaii, Dec. 2011, pp. 380–383
10 Peng, C., Chen, B.-Y., Tsai, C.-H.: 'Integrated Google Maps and smooth Street View videos for route planning'. Int. Computer Symp. (ICS), Tainan, Taiwan, Dec., 2010, pp. 319–324
11 Lin, Y.C., Tsai, P.H., Ha, N.H.: 'The impacts of information technology enabled services on service innovations: a case study in a publishing company'. IEEE Asia-Pacific Services Computing Conf., Jeju, Korea, Dec., 2011, pp. 41–47
12 Jones, W.D.: 'Microsoft and Google vie for virtual world domination', *IEEE Spectr.*, 2006, **43**, (7), pp. 16–18
13 Google, 'Google Maps JavaScript API V3', http://code.google.com/intl/en/apis/maps/documentation/javascript/reference.html, accessed Dec. 2011
14 Evenden, G.I.: 'Cartographic Projection Procedures for the UNIX Environment—A User's Manual'. USGS Open-File Report, 2003, pp. 90–284
15 Proj4js Track and Wiki, http://trac.osgeo.org/proj4js/, accessed Jan. 2012
16 Geomatics O.G.P.: 'EPSG Geodetic Parameter Dataset', http://www.epsg.org/, accessed Jan. 2012
17 Ghilani, C.D., Wolf, P.R.: 'Elementary surveying: an introduction to geomatics' (Pearson Education, Upper Saddle River, NJ, 2008, 12th edn.), pp. 269–294
18 Wolf, P.R., Dewitt, B.A.: 'Elements of photogrammetry with applications in GIS' (McGraw-Hill, Madison, WI, 2000, 3rd edn.), pp. 334–341
19 Ghilani, C.D., Wolf, P.R.: 'Adjustment computations: spatial data analysis' (John Wiley & Sons, Hoboken, NJ, 2006, 4th edn.), pp. 255–282