3D GIS APPLICATION BY IMPLEMENTING 3D CITY MODEL WITH GOOGLE EARTH AND GOOGLE MAP INTEGRATION

Fatih SARI[1], Hakan KARABORK[1],

[1] Department of Geomatic, Faculty of Engineering and Architecture,
Selcuk University, **Turkey**

fatihsari@selcuk.edu.tr, hkarabork@selcuk.edu.tr

Keywords: 3D City Models, 3D GIS, Google MAP

Abstract

Because of the rapid technological developments in web based geographical information systems, web based map applications and visual tours with 3d models, web applications are getting more indispensable part of presenting geographical information systems through web. In web based geographical information systems, the visualization is more important than desktop information systems. For the need of representing world as like its original view, spatial objects and real world furnitures have to be visualisate as real seeming. Also attribute data of spatial objects must be integrated with spatial objects for an effective web based information systems. In this part, 3d city models are providing effective solution for virtual earth concept.

In this study, a historical area in Konya city examined for integrating 3d models with attribute data. In this context, Google Earth and Google Maps API applications are integrated for the aim of getting together 3d city models, attribute data and other Google Maps API functions as like directions and street maps. The historical buildings are composed as textured 3d model by using various softwares. Then attribute data of the study area and buildings are associated with models by using Google Maps API and Google Earth API using XML databases. As a result, needings, problems and results are examined in integration section and detailed compare shared about software results which used for this study.

1. INTRODUCTION

As an affection of the age of technology, web based geographical information systems become an indispensable part of the projects that organizations and foundations have to realize. Because of it provides accessibility, easy usage functions and effective sharing options, it can address to the public people, so the web based geographical information systems are chosen for the purpose of solving techniques in presentation sections [1].

Since the development of the science of photogrammetry, new techniques and new algorithms come out for gathering information about earth objects. With the support of high performance computer hardwares, complex and huge datasets started to use. With these developments, photogrammetry applications have reflected to the archeological and restoration areas. Especially in 3D city models, software and photogrammetry techniques are using by organizations widely [2].

The needs for 3D city models are growing and expanding rapidly in a different kind of fields and areas. In a steady shift from traditional 2D-GIS toward 3D-GIS, a great amount of accurate 3D city models have become necessary to be produced in a short period of time and great amount of softwares become using for producing 3D city models. [3].

With an increasing number of people living in or moving to cities, cities are getting growing and carrying more people. This shows that vital functions of people and relating urban developments must be planned by using new techniques. In this situation, 3D city models provide effective planning, visualization tools and behave as a bridge between public and planners. But 3D city models has large amount of data. For the aim of public usage of the 3D city models on web browser, some algorithms and techniques are using nowadays [4].

Google Earth API is a commonly used code library all around the world, provided by Google which allow users to add Google Earth Map to their web pages. So users can display 3D models inside their web pages without installing Google Earth software by using code blocks. Similarly, Google Maps API provides users to add Google Map inside their web page. Google Maps API consists of code blocks that used to modify map according to user needings. API means Application Programming Interface and it provides function libraries to the users. All around the world, Google provides satellite images, road maps, terrestrial maps, 3D buildings, road directions and geocoding options. Because of this, in web based geographical information systems, the Google Earth and Google Map is being good source and increasing choice option by users day by day.

As important as 3D buildings, also serving attribute data of the 3D models are another important issue. With the aid of web applications, attribute data of the 3D buildings can be serve to the users. In this study Google Map API and Google Earth API integration has examined for the purpose of serving attribute data with 3D buildings at the same time. These two applications are applied to the same web page and integrated to each other with JavaScript codes. These maps are moving to the desired place at the same time. By adding combo boxes which has location data of buildings in the study area, two maps can move to the place easily. In this situation, Google Earth Map is displaying 3D buildings and Google Map is displaying attribute data of the building by using info windows. Attribute data are stored in XML databases so with the GeoXML function class, attribute data are shown on the map. The needings, the way of this study and integration steps are examined in this study.

2. MATERIAL METHOD

In this study, a web application has prepared in Konya city with cultural heritages which are carrying Ottoman and Seljuk Empire architectures. Konya city has an important role in Seljuk and Ottoman Empire age. Because of this in city centre there are too many museums, mosques, and madrasas and castle ruins. For the purpose of presentation and provide a simple view of the cultural heritages a web page has prepared with Google Earth and Google Map. Eight museums, ten mosques and two madrasas have added to the application. The mosques are Selimiye, Aziziye, Iplikci, Kapu, Alaaddin; the museums are Etnografya and Mevlana Museum and the madrasas are Ince Minare and Karatay.

Figure 1: Study area, Konya city in Turkey

These cultural heritages are commonly close to each other and easily available because of being in the middle of the city centre. Figure 2 shows the city centre of Konya city.

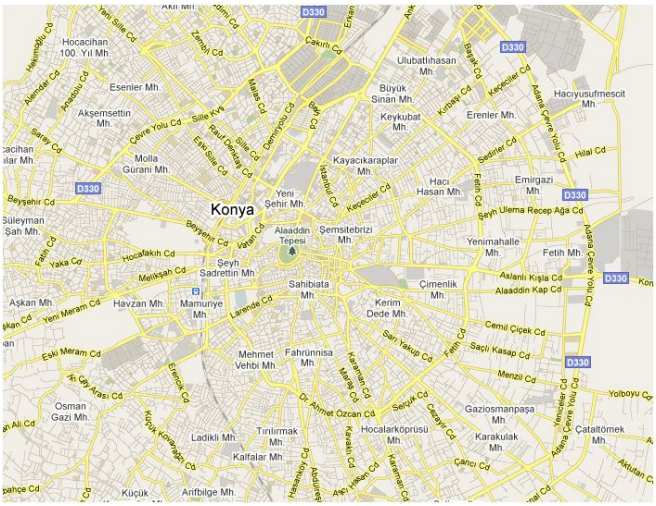

Figure 2: City centre of Konya

Google Earth and Google Map API are using JavaScript language and enable users to customize applications and add maps to their own web pages [5]. For this application all the customizing and managing the data are prepared in JavaScript code blocks [6]. These code blocks are integrated into *HTML (Hyper Text Markup Language)* codes by separating <*script*> tags.

Figure 3: Google Earth and Google Map

For displaying 3D models, Google Earth map and for displaying locations, directions, placemarks and different map types, Google Map has integrated to the web page. Addition to these features, Google Map used to display attribute data of the 3D buildings with Info window property. In this study attribute data of the 3D buildings are stored in XML databases.

3. APPLICATION

The first step of the application is defining Google Earth Map which 3D buildings will be shown on map and Google Map which will represent attribute data and location information of 3D buildings. The code blocks that used to define Maps and Map features are given below [5].

```
google.earth.createInstance
(
                    'earth',
                    function(ge) {
                    ge = ge;
```

Google Earth Map Define

```
map = new GMap2($('#map').get(0));
```

Google Map Define

After Map definitions, zoom options, map type, center coordinates and other map feature definitions are added to the application.

```
var la = ge.createLookAt('');
la.set(lat, lng, 10, ge.ALTITUDE_RELATIVE_TO_GROUND, 5, 70, 300);
ge.getView().setAbstractView(la);
ge.getNavigationControl().setVisibility(true);
```

Google Earth

```
map.setCenter(new GLatLng(38.02601078, 32.50697851), 14);
map.addControl(new GLargeMapControl());
map.enableContinuousZoom();
map.enableDoubleClickZoom();
map.enableScrollWheelZoom();
```

Google Map

With defining two maps coordinate and moving options together, two maps can be control at the same time. By appending map coordinates with click options to any map moves the other map to clicked location.
After this, with marker options, locations are defined with their coordinates. The codes below show the marker definitions.

```
window.placemark10 = DS_ge.createPlacemark('');
```

Google Earth

```
var point = new GLatLng(lng,lat);
var marker = new GMarker(point);
```

Google Map

Showing attribute data on markers are examined with GeoXML and Info window features. By clicking to the markers, an info window opens and shows the attribute data of the location which saved in XML database between <name> tags.

```
var link = ge.createLink('');
var href = 'http://alierdi.com/akademik_site/sunu/SU_KAMPUS.xml'
link.setHref(href);
```

map.openInfoWindow(map.getCenter(),
document.createTextNode("name"));

The below shows the XML database codes for a location.

```
<Placemark>

<name>Alaaddin Mosque is on the top of the so-called Alaaddin Mound (Alaattin Tepesi). The Palace of the Seljukid
sultans was once inside a citadel that was standing on this small hill. The Alaaddin Mosque was built adjacent to the
palace following the example of Early Islamic palace mosques. The Mosque comprises two different structural units
juxtaposed in different periods.

</name>
<styleUrl>#msnx_ylw-pushpin</styleUrl>
<Polygon>
<tessellate>1</tessellate>
<outerBoundaryIs>
<LinearRing>
<coordinates>32.43212390063241,37.87589832461899,0 32.43213134949626,37.87585974683354,0
        32.43210429744164,37.87585170864367,0 32.43210942002594,37.87580851629912,0
        32.43213531257779,37.87580842742926,0 32.43214509421942,37.87577225577592, </coordinates>
    </LinearRing>
    </outerBoundaryIs>
    </Polygon>
    </Placemark>
<Placemark>
```

After adding XML databases, the web page has ready for displaying attribute data. In Figure 4 the web page are shown. By choosing *" Choose Place"* combo box, on the left side 3D buildings are shown and on the right attribute data and location informations are shown.

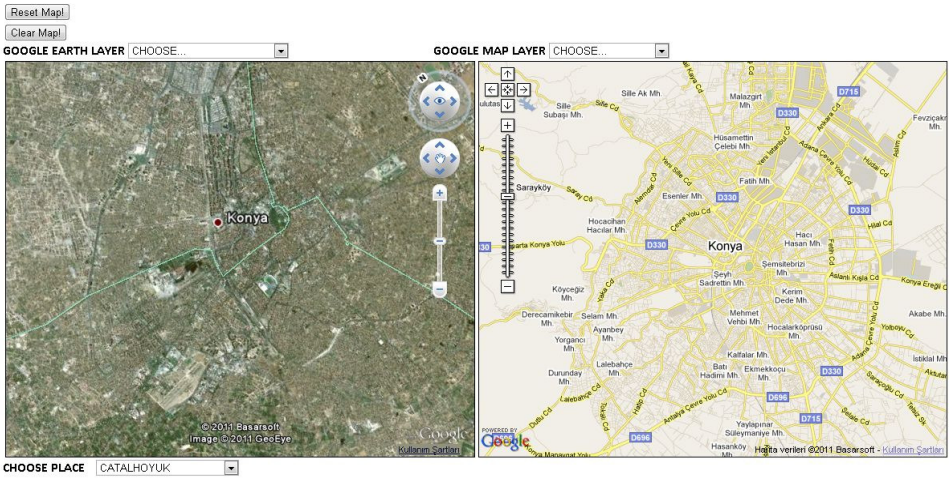

Figure 4: Web application page

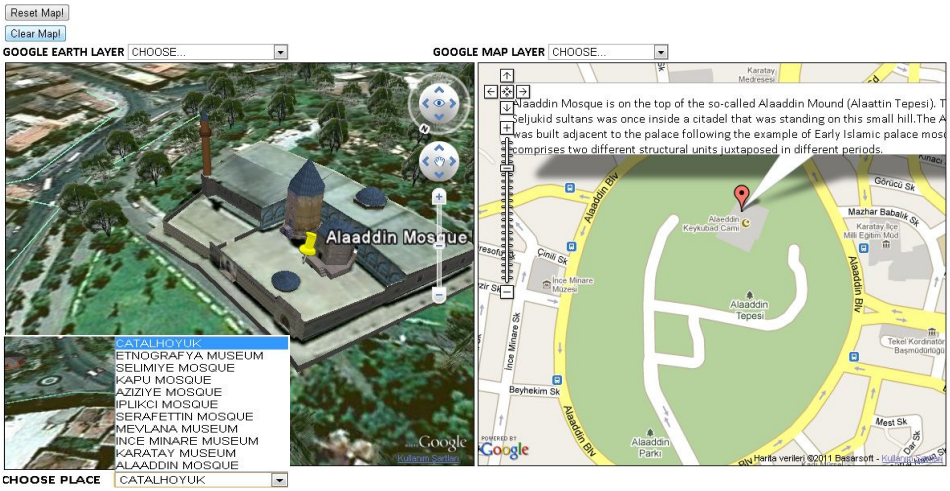

Figure 5: Web application info window and 3D building page

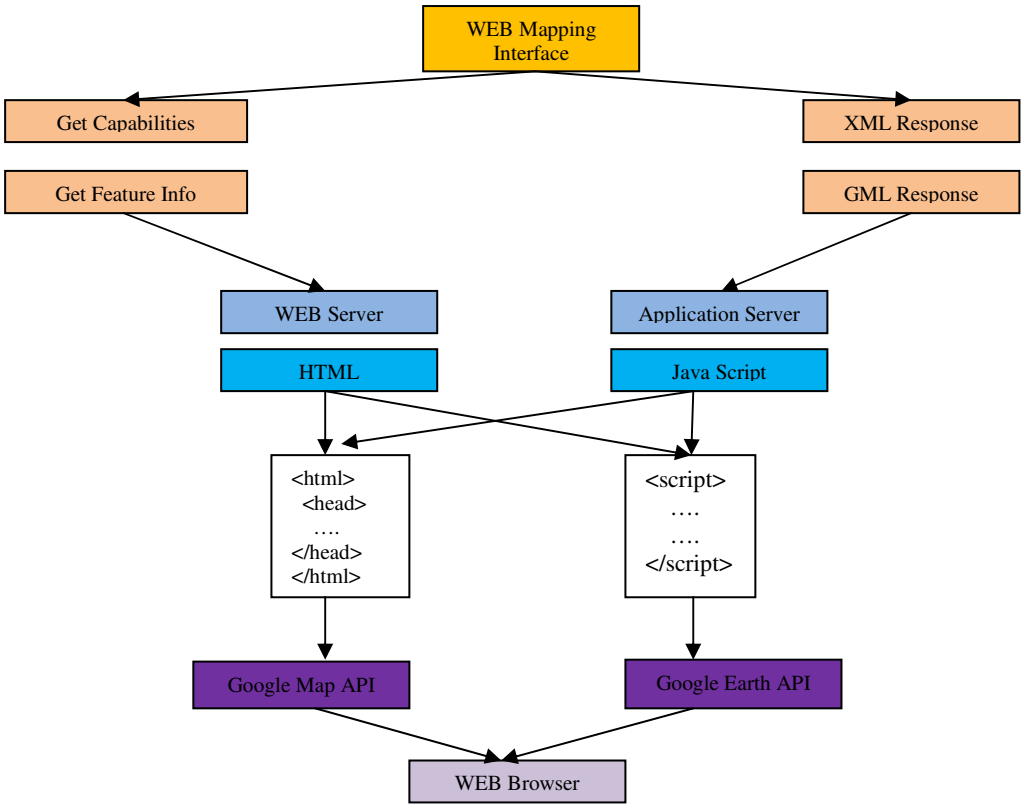

Figure 6: Flowchart of the application

4. CONCLUSIONS

With this application Google Earth and Google Map API applications are examined. The protocols and features are determined. Especially in web based geographical information systems how Google maps can be used is presented. With integrated Google Earth and Google Map, all the facilities are brought together.
By storing attribute data in XML databases, for 3D city models, with using Google Earth 3D buildings, desired attribute data can be shown on map. Integrating with Google Map, road directions, street names, 3 kinds of map types and location informations can be obtained together with 3D buildings.

5. REFERENCES

[1] SARI.F.,ERDI,A.,KIRTILOGLU.O.S., *İnternet Tabanlı Coğrafi Bilgi Sistemi Uygulamarında GeoServer-ArcGISServer ve Google MAP API Entegrasyonu,* 13.Harita Bilimsel ve Teknik Kurultayı, Ankara 2011

[2] F. Karsli, E. Ayhan, E. Tunc, *Building 3d Photo-Texture Model Integrated Wıth Gis For Architectural Heritage Conservation*, CIPA 2003 New Perspectives to save The Cultural Heritage, 29 September , Antalya - Turkey.

[3] Y. Takase, *, N. Sho, A. Sone, K. Shimiya, *AUTOMATIC GENERATION OF 3D CITY MODELS AND RELATED APPLICATIONS*, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV-5/W10

[4] Bo Maoa, Yifang Bana, Lars Harrie, *A multiple representation data structure for dynamic visualisation of generalised 3D city models*, ISPRS Journal of Photogrammetry and Remote Sensing

[5] SARI.F., ERDI.A.,KIRTILOGLU.O.S.,*Kampüs Bilgi Sistemi Oluşturma Çalışmaları ve Panoramik Görüntüler,Konya Selçuk Üniversitesi Örneği,* 13.Harita Bilimsel ve Teknik Kurultayı, Ankara 2011

[6] Google Maps API Developer Guide on http://code.google.com/intl/tr-TR/apis/maps/index.html

ACKNOWLEDGEMENTS

This authors wish to acknowledge the financial assistance given by the Coordinate of Science Research project of Selcuk University.