A Method for Estimation Camera Georeference in GIS-based Video Surveillance

Aleksandar Lj. Milosavljević[1], Dejan D. Rančić[2] and Aleksandar M. Dimitrijević[3]

Abstract – **In this paper we present a method for estimation of parameters used in georeferencing surveillance camera video in order to integrate it with a three dimensional geographic information system (3D GIS). Camera's frame is fully georeferenced knowing 7 parameters that describe: geographic position, geospatial orientation, and field-of-view of the camera in the moment of capturing that frame. Since the precise measuring of these parameters is extremely difficult, in this paper we proposed a solution for their estimation based on Levenberg-Marquardt's numerical method. As the input, our method use 3D coordinates of identifiable points from camera's video frame. In order to determine these coordinates we rely on use of conventional GIS data such as aerial imagery and digital terrain model (DTM).**

Keywords – **Video Surveillance, Geographic Information Systems, Augmented Reality, Augmented Virtual Environments.**

I. Introduction

Video monitoring plays an important role in the surveillance system for different security and military applications [1]. A typical surveillance system consists of multiple fixed and/or PTZ cameras that monitor various areas. In the conventional scenario, every single surveillance camera connects directly to the appropriate display that is monitored by the operator. However, applying this approach to a larger number of cameras has serious drawbacks [2]. The problem with such systems arises when the number of cameras/displays starts to exceed the cognitive capacity of the operator. The operator must constantly perform mapping between camera image with the corresponding location in the real world, and this complicated task requires a lot of experience and practice [2]. To ensure the coordination and monitoring in a system with multiple cameras, it is necessary to introduce a single reference frame in which all these images can be mapped. GIS in this regard arises as a natural solution, but not only because it is used for referencing any geospatial data, but also because it provides semantic information (e.g., locations of roads, buildings, etc.) which can be extremely important in certain video surveillance applications [3].

To integrate some information into a GIS it must be somehow georeferenced. Georeferenced video, obtained from surveillance cameras or otherwise, is terminologically designated as geospatial video. Video represents display of moving scenes using still images (frames) that change in a given time interval [4]. As the video consists of individual frames, its georeferencing requires that each individual frame contains georeference. Georeferencing a frame requires information on camera viewpoint in the moment the frame was captured. Data set which determines the camera viewpoint is defined by the viewpoint model [5].

An observer (or camera) viewpoint into three-dimensional space is completely determined knowing: the position of the observer, the orientation of the observer, and the field-of-view (abbr. **fov**). In contrast to the field-of-view, which represents a simple parameter, the position and the orientation of the observer are more complex viewpoint features that can be represented in different ways.

For the presentation of the observer position we used the WGS84 coordinate system, so the position is determined using three parameters: latitude (abbr. **lat**), longitude (abbr. **lon**) and altitude (abbr. **alt**) .

The observer takes proper orientation in 3D space by applying rotation around all three coordinate axes. Consequently, for the representation of the orientation we also use three parameters, each of which represents the angle of rotation around respective axis. In order to give meaning to these three angles, it is necessary to define an appropriate reference coordinate system that defines axis around which rotations will be performed. The most commonly used reference system of this type is defined for the aviation industry where rotations are designated as *pitch*, *roll* and *yaw* [6]. This model can be applied to represent the orientation in the geographic space if the center of the coordinate system is placed to the position the observer, *z* axis is directed towards the gravitational center of the Earth, and *x* axis in the direction of geographic north pole. In this case the rotation in a clockwise direction around the *z* axis is the view **azimuth**, while the rotations around the *y* and *x* axes are **pitch** and **roll**, respectively.

Finally, in order to specify an absolute viewpoint into 3D geographic space, i.e. video frame georeference, it is necessary to determine 7 parameters: **lat**, **lon**, **alt**, **azimuth**, **pitch**, **roll** and **fov**. Although it is possible to measure them, it is a complicated technique that do not always guarantee a satisfactory precision of the results. An alternative to the measurement is the estimation of parameters using indirect techniques. One possibility, that is described in this paper, is based on use of Levenberg-Marquardt's [7,8] numerical methods for their estimation based on the 3D coordinates of the characteristic (identifiable) points with a video camera frame. To obtain coordinates of these points we relied on aerial imagery and digital terrain model (DTM). As the numerical method is applied, using more points results in a better overall estimation of the georeference parameters.

The paper is organized as follows: Section 2 presents an overview of the model and techniques used in the integration

[1] Aleksandar Lj. Milosavljević is with the Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E–mail: aleksandar.milosavljevic@elfak.ni.ac.rs

[2] Dejan D. Rančić is with the Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E–mail: dejan.rancic@elfak.ni.ac.rs

[3] Aleksandar M. Dimitrijević is with the Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E–mail: aleksandar.dimitrijevic@elfak.ni.ac.rs

of geospatial video and 3D GIS. Section 3 describes the proposed method for estimating camera georeference, while Section 4 provides details of the actual software implementation of the proposed method. Finally, in the conclusions (Section 5) we gave a short summary of the work and emphasized the advantages of the proposed approach.

II. GIS-BASED VIDEO SURVEILLANCE

GIS-based video surveillance it the term used for systems that integrate geospatial video and 3D GIS using augmented reality techniques. Augmented reality technique are used to combine real scene viewed by the user with a virtual scene generated by the computer, so that they augment real scene with additional information [9]. Surveillance system, in this conjunction, offers real world view through videos from multiple cameras. On the other hand, 3D GIS establishes a virtual environment that augments, or is augmented with such video depicting reality.

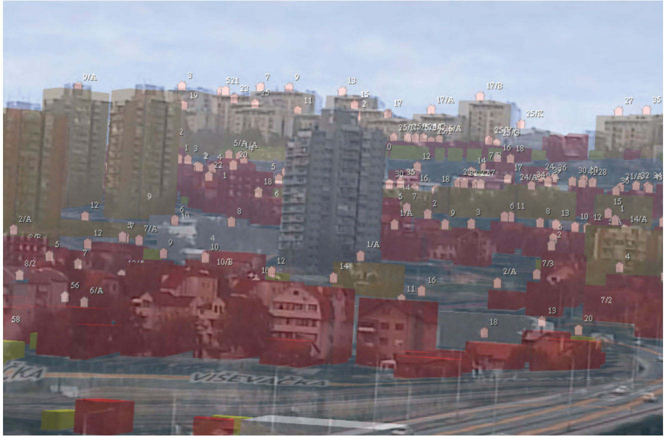


Fig. l. GIS-augmented video

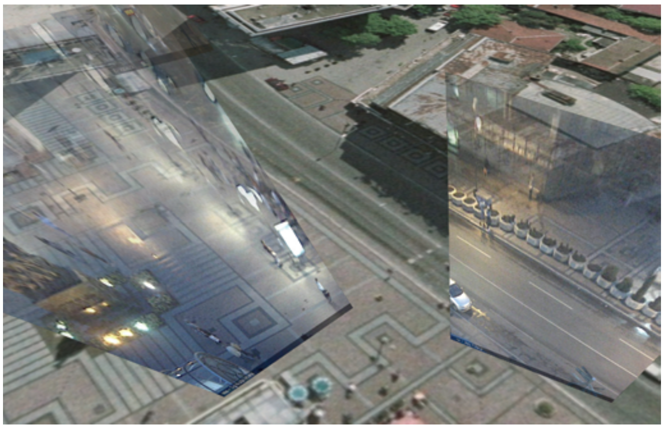


Fig. 2. Video-augmented GIS

Inspired by Milgram's continuum between real world and virtual reality [10], we defined models of integration of 3D GIS and geospatial video. Analogously to the augmented reality and the augmented virtuality, that are defined in this continuum, we defined two models: **GIS-augmented video** and **Video-augmented GIS**. The first model is analogous to the augmented reality and is closer to the real world, i.e. to the video as the way to see the real world. The second model is analogous to the augmented virtuality that is located on the virtual reality side, i.e. 3D GIS as a tool to create it. A key feature of GIS-augmented video is direct display of video, while the corresponding 3D GIS scene is created in the background providing geolocation for each pixel in the frame (illustration is given in Fig. 1). Video-augmented GIS is characterized by projection of video frames into 3D GIS scene enabling us with the ability of free movement through the scene and mixing multiple video streams (illustration is given in Fig. 2). Video demonstration of the implemented prototype of GIS-based video surveillance is available at: http://www.youtube.com/watch?v=VJGC2P3t8xg.

Details regarding the implementation of the identified models of integration through a prototype of GIS-based video surveillance are beyond the scope of this paper. However, in order to understand the proposed method, one need to be familiar with the transformations that are performed within graphic adapters (via OpenGL). It is also necessary, to establish correspondence between the seven parameters of the viewpoint model and the appropriate transformations.

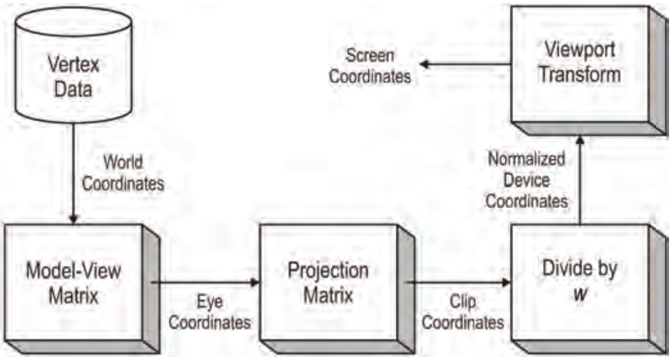


Fig. 3. Vertices transformation from world coordinates to screen coordinates in OpenGL

Fig. 3 shows the process of vertices (3D coordinates of points in space) transformation from world coordinates to screen coordinates of the window [11]. We see that the first transformation is done by multiplying a vertex vector with a model-view matrix. After this transformation we get the coordinates relative to the viewpoint, i.e. eye coordinates. The following transformation is perspective projection applied using projection matrix. The output of this transformation are clip coordinates. Their role is to discard vertices that are not in the view frustum defined with a given projection matrix. It should be noted that OpenGL works with homogeneous coordinates. Homogeneous coordinates allow geometric operations in the projected space. Because of this, all vectors in OpenGL have four components labeled *x*, *y*, *z* and *w*, and the corresponding transformation matrices are 4x4 size. In order to normalize homogeneous clip coordinates the *x*, *y* and *z* components are divided with *w*. This provides us with a normalized device coordinates (NDC). NDC coordinates correspond to the screen coordinates with the difference that they range from -1 to 1. For their transformation into real screen coordinates viewport transformation is used. Viewport is a rectangular area on the screen, i.e. window, where the 3D scene is displayed. It is specified using the coordinate of the lower-left corner, so as width and height in pixels.

In order to overlap video frames received from the camera with the corresponding 3D GIS scene, which is the feature of GIS-augmented video, the model-view matrix must be initialized according to the current video frame georeference:

```
double xC, yC, zC;
Geodetic2Geocentric(lat, lon, alt, xC, yC, zC);
Matrix4x4 MV;
MV.rotateX(-90.0);
MV.rotateY( roll);
MV.rotateX(-pitch);
MV.rotateZ( azimuth);
MV.rotateX( lat - 90.0);
MV.rotateZ(-lon - 90.0);
MV.translate(-xC, -yC, -zC);
```

It should be noted that the geodetic coordinates (*lat*, *lon*, *alt*) are not suitable to be used as the world coordinates of the vertices. Instead we use the geocentric coordinates (*x*, *y*, *z*) based on the Cartesian coordinate system centered at the gravitational center of the Earth. Geocentric coordinates are represented in meters.

Initialization of the projection matrix use information on frame size in pixels, and the field-of-view (parameter *fov*):

```
Matrix4x4 P;
double aspect = frameWidth / frameHeight;
P.perspective(fov / aspect, aspect, 1.0, 5000.0);
```

III. Description of the Proposed Method

The method for estimation of the viewpoint model parameters will be presented for the case of fixed surveillance cameras. Specificity, i.e. advantage of the fixed camera is that the georeference parameters do not change. They are fixed and depend on camera mounting and lens characteristics. With some modifications, proposed method can be applied to PTZ cameras. In the second case, we also have a fixed camera position (**lat**, **lon**, **alt**), while absolute orientation (**azimuth**, **pitch**, **roll**) and the field-of-view (**fov**) depend on the *pan*, *tilt* and *zoom* parameters of the camera.

To estimate camera georeference, as the input data we use measurements that connect points in camera frame (screen coordinates - *exM*, *eyM*) and points in geographic space (geodetic coordinates - *latM*, *lonM*, *altM*). Calculation of 7 parameters of camera viewpoint is performed using Levenberg-Marquardt's numerical method. In this case numerical method evaluates the transformation parameters that best (with the lowest mean square error) maps input 3D coordinates to output screen coordinates. Transform itself is explicitly set, in this case it is equivalent to OpenGL transformation shown in Fig. 3. Using previously defined model-view and projection matrix, the transformation has the following form:

```
double xM, yM, zM;
Geodetic2Geocentric(latM, lonM, altM, xM, yM, zM);
Vector4D v(xM, yM, zM, 1.0);
MV.transform(v);
P.transform(v);
ex = 0.5 * (v.X / v.W + 1.0) * (frameWidth  - 1);
ey = 0.5 * (1.0 - v.Y / v.W) * (frameHeight - 1);
```

Mean square error, that is minimized during the process, is calculated between screen coordinates obtained by transformation (*ex*, *ey*) and expected screen coordinates (*exM*, *eyM*). In each cycle of the iterative process the modification of the georeference 7 parameters is carried out in order to decrease the error. The process ends when the error falls below a certain value or when increment in all parameters falls below a certain value.

To begin the iterative process is necessary to define the initial values for the estimating parameters. In the case where a function has only one minimum initial values do not affect the final outcome. However, if there are multiple local minima the initial value should be close to the expected solution. When applied to our case, given the complexity of the transformation, this means that it is necessary at the beginning of the process to determine the approximate values of the parameters, i.e. roughly determine camera georeference.

List of steps through which the proposed method is carried out are:

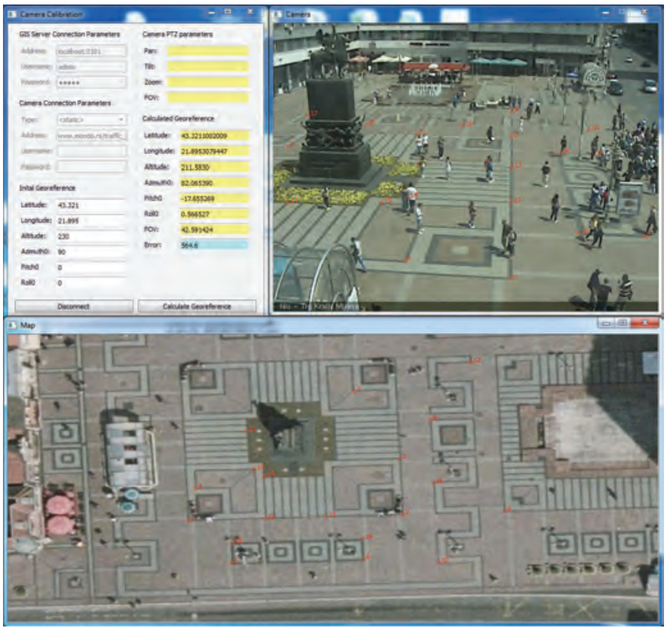
1. setting initial georeference,
2. identification of characteristic points in video (about 10 points),
3. finding 3D coordinates of the identified points,
4. estimation of optimal georeference using iterative process, and
5. verification of the results obtained using GIS-based video surveillance.

IV. Implementation of the Proposed Method

In order to test, so as to make practical usage of the proposed method we implemented an application for georeferencing surveillance cameras. The application is implemented in C++ using Qt framework. To implement Levenberg-Marquardt's numerical method we used an open source library *levmar* [12]. Obtaining video frames is done using the HTTP protocol, so present solution supports only network (IP) or web cameras. In order to display the map of the area of interest and obtain 3D coordinates of the characteristic points the application uses our GIS server that implements OGC WMS service and service for retrieving terrain elevation. Fig. 4 depicts the user interface of the implemented application.

The user interface of the applications is divided into three parts: the main application window for setting and reading parameters, the window that display video, and the window that for viewing and navigating through a geographic map. Beside display of video and map, these windows additionally provide parallel insertion of characteristic points. When a characteristic point is identified in the video window its screen coordinate is remembered to be paired with an adequate 3D coordinate. The same action in the map window results in a request to the GIS server to read terrain elevation for a given coordinate.

In the example shown in Fig. 4 georeferencing was performed with 19 points. The result of this particular georeferencing is shown in Fig. 5 where we have video projected into virtual 3D GIS scene. The accuracy of obtained georeference can be noticed on the edges of the frame.



GIS Server Connection Parameters

Address: localhost:2101

Username: admin

Password: *****

Camera Connection Parameters

Type: <static>

Address: www.mondo.rs/traffic

Username:

Password:

Inital Georeference

Latitude: 43.321

Longitude: 21.895

Altitude: 230

Azimuth0: 90

Pitch0 0

Roll0 0

Camera PTZ parameters

Pan:

Tilt:

Zoom:

FOV:

Calculated Georeference

Latitude: 43.3211002009

Longitude: 21.8953079447

Altitude: 211.5830

Azimuth0: 82.065390

Pitch0 -17.655269

Roll0 0.566527

FOV: 42.591424

Error: 564.6

Fig. 4. Application for georeferencing surveillance cameras

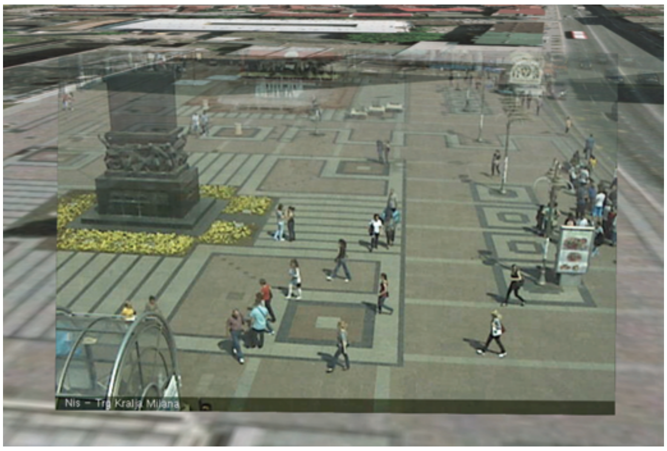


Fig. 5. The result of georeferencing shown using projected video mode

V. Conclusions

Integration of surveillance cameras' video into 3D GIS requires their prior georeferencing. As video consists of individual frames, its georeferencing comes down to georeference each individual frame. A frame georeference is defined by 7 parameters of the viewpoint model that absolutely determines the camera position, orientation and field-of-view in the moment the frame was captured. Advantage, i.e. simplification in terms of georeferencing surveillance cameras is that they are always mounted in a fixed location. Therefore the first 3 parameters of the georeference are constants. With fixed surveillance cameras the other 4 parameters are also constant, while for PTZ cameras they depend on the camera's local orientation (pan and tilt) and zoom.

Since GIS-based video surveillance provides overlapping of video frames with a virtual 3D GIS scene that is created in the background, even very small errors in georeference become apparent. Therefore, it is necessary to precisely determine camera georeference. Although there is a possibility of measuring these parameters, it is a set of complicated techniques which results does not always guarantee a satisfactory precision. As the alternative, in this paper we describe a method for estimating the georeference using 3D coordinates of the characteristic points identified in the video frame. The 3D coordinates are obtained using areal imagery and digital terrain model, while the estimation is done using Levenberg-Marquardt's numerical method.

One of the advantages of using numerical method to estimate the parameters is the possibility of adding an arbitrary number of characteristic points. This minimizes errors that originate from coordinate reading, so as the model imperfection (DTM and areal imagery). It also compensates parameter estimation errors that are due to the camera lens aberrations.

References

[1] I. Oner Sebe, J. Hu, S. You, U. Neumann, “3D Video Surveillance with Augmented Virtual Environments”, in First ACM SIGMM international workshop on Video surveillance, 2–8 November 2003, Berkeley, CA, pp. 107-112.

[2] N. Kawasaki, Y. Takai, “Video Monitoring System for Security Surveillance based on Augmented Reality”, in Proc. of the 12th International Conference on Artificial Reality and Telexistence, 4-6 December 2002, Tokyo, Japan, pp. 180-181.

[3] K. Sankaranarayanan, J.W. Davis, “A Fast Linear Registration Framework for Multi-Camera GIS Coordination”, in Proc. of the 5th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'08), 1-3 September 2008, Santa Fe, NM, pp. 245-251.

[4] Free On-Line Dictionary Of Computing (FOLDOC), “video”, http://foldoc.org/video, 2010.

[5] A. Milosavljević, A. Dimitrijević, D. Rančić, “GIS-augmented video surveillance”, International Journal of Geographic Information Science, vol. 24, no. 9, September 2010, pp. 1415-1433.

[6] National Aeronautics and Space Administration (NASA), “Aircraft Rotations (Body Axes)”, http://www.grc.nasa.gov/WWW/K-12/airplane/rotations.html, 2008.

[7] K. Levenberg, “A method for the solution of certain problems in least squares”, Quarterly of Applied Mathematics, vol. 2, pp. 164-168, 1944.

[8] D. Marquardt, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters”, SIAM Journal on Applied Mathematics, vol. 11, no. 2, pp. 431-441, 1963.

[9] R.T. Azuma, “A Survey of Augmented Reality”, Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355-385, 1997.

[10] P. Milgram, F. Kishino, “A Taxonomy of Mixed Reality Visual Displays”, IEICE Transactions on Information Systems, vol. E77–D, no. 12, pp. 1321-1329, 1994.

[11] S.H. Ahn, “OpenGL Transformation”, http://www.songho.ca/opengl/gl_transform.html, 2011.

[12] M. Lourakis, “levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++”, http://www.ics.forth.gr/~lourakis/levmar, 2011.