

FACTA UNIVERSITATIS

Series: **Automatic Control and Robotics** Vol. 16, N° 1, 2017, pp. 1 - 13

DOI: 10.22190/FUACR1701001M

## TRANSFORMING SMARTPHONE INTO GEOSPATIAL VIDEO PROVIDER

UDC (621.395.721.5:004.38):004.722.45

**Aleksandar Milosavljević, Bratislav Predić, Dejan Rančić**

University of Niš, Faculty of Electronic Engineering, Republic of Serbia

**Abstract.** *Modern handheld devices, i.e. smartphones, have gone beyond their basic purpose for some time now. Beside powerful screens, which usually surpass those we use on our desktops, powerful cameras, and internet connectivity, these devices usually integrate many sensors. In this paper, we discuss how we can combine smart phone's camera, connectivity, GPS receiver, accelerometer, and magnetometer in order to transform this kind of a device into geospatial video source. To validate and test suggested solution, we implemented a simple mobile app that fetches frames and sensors' data, calculates georeference and streams geospatial video. In the paper, we also discussed limitations that are caused by the limited precision of used sensors.*

**Key words:** *GIS, video, augmented reality, mobile devices, sensors*

### 1. INTRODUCTION

Development of smartphones in the past few years is marked with rising trend of integration on an increasing number of sensors. These sensors are used to observe various characteristics of a user's environment which are used in various applications. Initially, handheld devices started detecting geographic location as the most important information about devices' environment. Development and miniaturizing of Global Navigation Satellite System (GNSS) receivers allowed integration of these sensors into practically all available smartphone models. The trend has continued, so today's smartphones also have sensors for measuring temperature, atmospheric pressure, humidity, one or more cameras, accelerometer, gyroscope, magnetometer, etc.

It is interesting to overview evolution of the positioning system since GNSS receiver is adopted for outdoor positioning. It usually allows reaching a good level of precision, but somewhere the signal is too noisy or not available (i.e. indoor or urban canyons), and GNSS positioning is not allowed. The future trend of positioning is to have a seamless solution,

---

Received August 9, 2016

**Corresponding author:** Aleksandar Lj. Milosavljević

University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Republic of Serbia

E-mail: alexm@elfak.ni.ac.rs

which means to have a continuous and stable localization everywhere, from outdoor to indoor scenario. In the last years, many research groups are working to study different solutions to bridge this gap, as an alternative to GNSS positioning, using different kinds of sensors. Summarizing the main available technologies, it is possible to group them considering several different characteristics:

- Wi-Fi is dedicated for indoor environments in a transmission range between 30-200 m. This procedure can have good performance, but it suffers from outliers, signal coverage and depends on access points and their geometric distribution [1];
- Inertial sensor navigation is another alternative approach that relies on the accelerometer, gyroscope and magnetometer in the pedestrian navigation in order to correct the drifts [2];
- Bluetooth can be used in the range-free localization system, where each smartphone periodically scans nearby Bluetooth enabled devices and sends the results to the localization server. This server collects the scanning results into a short period and finds their locations using range-free algorithms [3].
- Images or video from the integrated camera can be used through object recognition process to improve the precision of the positioning [4].

Positioning and navigation are usually estimated adopting an integration between the techniques described above with the range of precision between 50 m to 60 cm. In all fields of application, and especially in the area of video georeferencing, the accuracy of determining the position and the orientation of the phone is an essential prerequisite for providing a consistent quality of service. Detailed analyses of the stability of accelerometer and magnetometer measurements implemented in smartphones Samsung Galaxy S4, S5, and iPhone 4 is presented in [4]. The conclusion was that estimation of the orientation using these two sensors could provide solid grounds for video georeferencing task.

To integrate a video into GIS, the video must be geographically referenced. The potential of spatial video, which represents geographically referenced videographic data, as an additional data type within GIS was considered by Lewis et al. [5], whose work focused on representing video frames using Viewpoint data structures to enable video geospatial analysis. Georeferencing video is, in many ways, equivalent to camera pose estimation for outdoor augmented reality. Min et al. [6] presented an overview of different approaches to this problem. In their opinion, any outdoor augmented reality system had to use spatial information as the basic information to provide a digital platform for the system. Using geospatial data for camera pose estimation was examined by Sourimant et al. [7] in their research, which considered 3D building reconstruction. They presented a method based on the registration of GNSS measurements, video sequences, and rough 3D models of buildings. An excellent idea on how to apply augmented reality to the visualisation of a subsurface GIS infrastructure was presented by Schall et al. [8]. They relied on GPS and an inertial measurement unit for camera pose estimation.

This paper is organized as follows. In Section 2, we introduced and described the concept of the geospatial video and emphasized its use in a geographic information system (GIS). In Section 3, we described a method for georeferencing and transmission of geospatial video using a mobile device. Design and implementation details regarding prototype of the mobile application GeoVideoCamera are presented in Section 4. In Section 5, we discussed the results gathered by the prototype and identified problems that arise from limited precision of the positioning and orientation sensors. Finally, the concluding remarks are given in Section 6.

## 2. GEOSPATIAL VIDEO

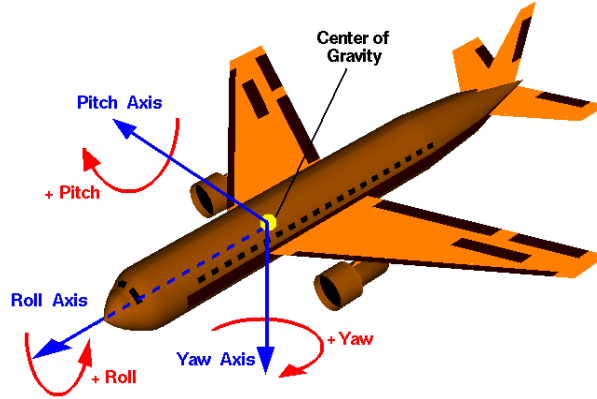
The term ‘video’ is used to denote moving images presented as a sequence of static images (called ‘frames’) representing snapshots of a scene captured at regularly spaced time intervals [9]. To integrate a video into GIS, the video must be geographically referenced. The georeferenced video will be determined as **geospatial video**. Because video consists of individual images (frames), georeferencing involves adding georeference information to each frame using information about the camera’s viewpoint at the moment that frame was captured. This information can be determined by seven parameters that we call the **observer viewpoint model**. An observer (or camera) view into 3D space is fully determined by the camera:

- Position
- Orientation
- Field-of-view (FOV)

Unlike field-of-view that represents a simple parameter, position, and orientation of the camera are complex characteristics and can be represented in various ways.

To represent the position of the camera in 3D space, we need three parameters. If we deal with geographic space, their interpretation depends on the coordinate system that we use. A geographic coordinate system can be geocentric, geodetic, or projected. Geocentric and Geodetic coordinate systems can be applied globally, while projected coordinate systems are usually tied to specific areas or countries. The geocentric coordinate system is an essentially Cartesian coordinate system where the coordinate centre is aligned with the Earth’s centre. To represent a certain point in space, the geocentric system requires specifying  $x$ ,  $y$ , and  $z$  coordinates. Geodetic coordinate systems use angular degrees to specify some point on the Earth’s surface. Latitude specifies the north–south, while longitude specifies the east–west position of a point. These two parameters are complemented with altitude parameter that represents the height of a point in relation to the sea level. The most popular geodetic coordinate system of today is called by underlying ellipsoid WGS84. WGS84 is the reference coordinate system used by the Global Positioning System (GPS), so it will be used for specifying position in our observer viewpoint model.

An orientation of the observer (i.e. camera) is represented by a vector positioned in the observer centre and directed toward the observer’s viewing direction. The orientation can be represented using rotations around each of the axes of a 3D coordinate system. To represent orientation we will need three parameters. Each of them represents a rotation around the certain axis of some reference coordinate system. The most commonly used coordinate system for such purpose is defined in aviation and is illustrated in Fig. 1 [10]. The rotations are defined using right-handed Cartesian coordinate system placed in aircraft centre of gravity, with  $x$ -axis aligned with an aircraft direction,  $y$ -axis pointing to the right side of the aircraft, and  $z$ -axis pointing from top to bottom. Rotation around  $z$ -axis is called **yaw**, around  $y$ -axis is called **pitch**, and around  $x$ -axis is called **roll**. All rotations are clockwise.



**Fig. 1** Aircraft rotations expressed using pitch, roll, and yaw angles

The described model can be used for specifying orientation in 3D geographic space if we tie this coordinate system to the observer's position, direct  $z$ -axis toward Earth's centre of gravity, and  $x$ -axis toward the geographic North Pole. In this case, clockwise rotation around  $z$ -axis represents an **azimuth** of the observer's viewpoint and will be called accordingly.

The set of parameters that fully describes camera's viewpoint in 3D geographic space is defined as the observer viewport model [11, 12]. The corresponding seven parameters are depicted in Table 1.

**Table 1** The observer view model parameters

Parameter	Unit	Range	Description
<i>lat</i>	angular degree	[-90, 90]	WGS84 latitude of a camera position.
<i>lon</i>	angular degree	[-180, 180)	WGS84 longitude of a camera position.
<i>alt</i>	meter	(-?, +∞)	The height of a camera position in relation to sea level.
<i>azimuth</i>	angular degree	[0, 360)	The angle between the north vector and the perpendicular projection of a camera view vector down onto the horizon.
<i>pitch</i>	angular degree	[-180, 180)	The angle between the horizon plane and a camera view vector.
<i>roll</i>	angular degree	[-180, 180)	The rotation around camera view vector.
<i>fov</i>	angular degree	(0, 180)	The horizontal field of view of a camera.

The term 'geospatial video' is defined based on the observer viewpoint model. It denotes a sequence of video frames where each of them is additionally described (i.e. georeferenced) using seven parameters from Table 1.

### 3. GEOREFERENCING VIDEO USING DEVICE SENSORS

In order to georeference video frames on a mobile device, we need to determine values of all seven parameters of the observer viewpoint model. First, we will discuss how to determine camera's field-of-view (the **fov** parameter).

The **fov** parameter exclusively depends on the type of camera that is built in the mobile device. Assuming that camera does not have optical zoom, this parameter remains constant all the time, and we should only measure it once for certain device model. A detailed description of the procedure for measuring camera field-of-view is given in [13]. The procedure is based on measuring distance ( $D$ ) between the camera and the wall perpendicular to its view direction, so as the width ( $W$ ) of the wall that is projected into the image. Since the viewpoint is located somewhere inside the camera, to measure **fov** we need to make the second measure with modified distance from the wall ( $D'$  and  $W'$ ). Based on these two measures, it is possible to calculate distance correction ( $k$ ), assuming that the **fov** is the same for both measures. Formulas that are used in the described procedure are as follows:

$$\arctan\left(\frac{W}{2 \cdot (D+k)}\right) = \arctan\left(\frac{W'}{2 \cdot (D'+k)}\right) \Rightarrow k = \frac{W' \cdot D - W \cdot D'}{W - W'} ,$$

$$fov = 2 \cdot \arctan\left(\frac{W}{2 \cdot (D+k)}\right) . \quad (1)$$

To determine device position, i.e. **lat**, **lon**, and **alt** parameters, we relied on measures from previously discussed Global Navigation Satellite System (GNSS) receiver. The readings that GNSS receiver provides are fully compatible with required parameters, so there's no need for additional processing. We should emphasise that, although we deal with the oldest, and the most widely used type of sensor in today's mobile devices, the precision, and availability of its readings is the most problematic from the georeferencing point of view. In this context, the particular problem is the value of the altitude, because, due to the physical issues, this parameter comes with the biggest error and often is not available at all.

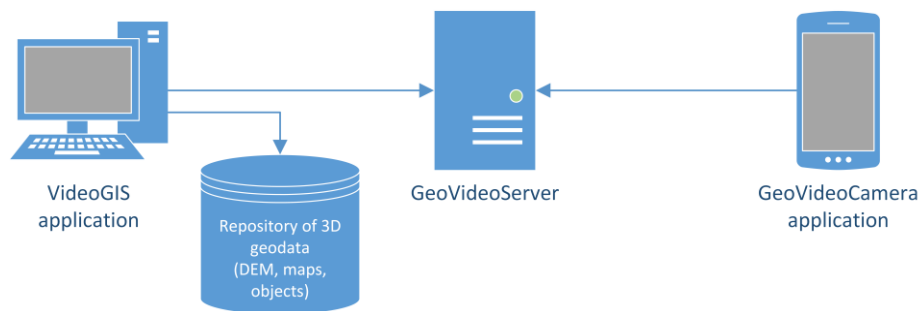
Finally, to determine absolute orientation of the device, i.e. azimuth, pitch, and roll parameters, is the most complex task because it involves fusion of measures from several different sensors. The basis for determining device's orientation is a measurement provided by the acceleration sensor (i.e. accelerometer). An accelerometer measures accelerations by all three axes. Since this sensor also measures the acceleration of the Earth's gravity, when the device is static the reading from an accelerometer represents a vector that points out to the Earth's centre of gravity. In other words, this reading can be used to determine **pitch** and **roll** parameters. To determine the third parameter of the orientation (i.e. **azimuth**) we need a reading from device's geomagnetic field sensor (i.e. magnetometer). A method for sensors' data fusion in order to get device's orientation is described in [14].

Since the **azimuth** parameter is obtained using a magnetometer, the value that we get is relative to the Earth's magnetic north. GIS data are typically referenced according to the geographic, i.e. true north. The difference between these two is known as magnetic declination. Magnetic declination varies from the geographic location and in time, but fortunately current magnetic declination for the current location is available through a GNSS receiver readings.

#### 4. DESIGN AND IMPLEMENTATION OF GEOVIDEOCAMERA MOBILE APPLICATION

To verify proposed method for georeferencing video frames, so as to examine the precision of measured parameters, we implemented prototype mobile application GeoVideoCamera. The purpose of this rather simple application is to read video frames and sensor data and to georeference and dispatch geospatial video.

In order to receive and display geospatial video, we used our previously implemented system for GIS-based video that consists of GeoVideoServer and VideoGIS client application [12]. GeoVideoServer had to be slightly modified to enable receiving of geospatial video from this new data source. VideoGIS application has been used in its original form. In Fig. 2 is shown overall architecture of the system that includes GeoVideoCamera application.

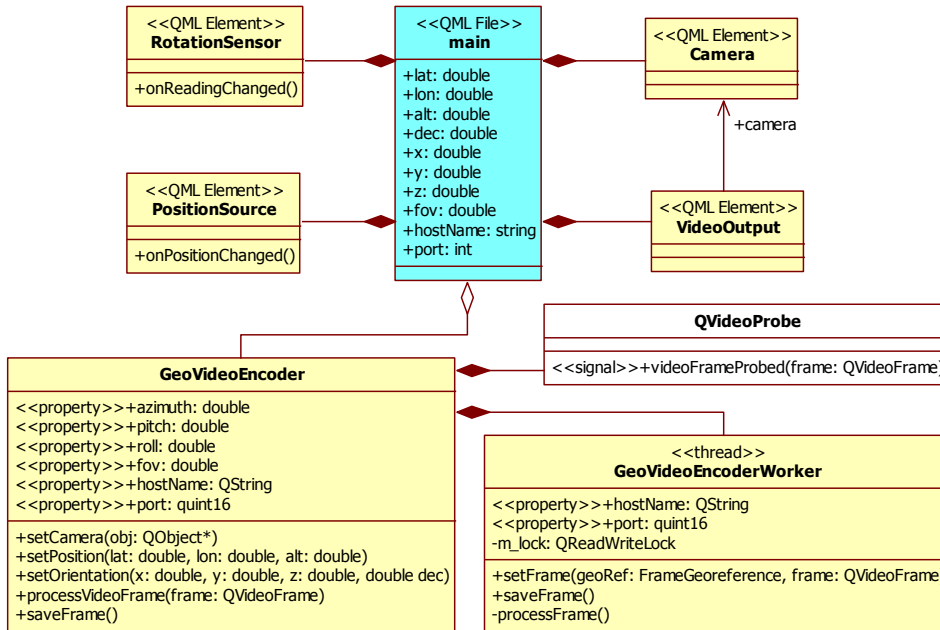


**Fig. 2** Overall architecture of GIS-based video system that includes GeoVideoCamera mobile application

The mobile application was implemented using Qt framework [15] in the version 5.6.0 and programming languages C++ and QML. Qt framework allows cross-platform development for all major desktop and mobile platforms. GeoVideoCamera application is implemented for Android platform and tested on a Samsung Galaxy Note 5 smartphone. A UML class diagram that presents the architecture of the application is shown in Fig. 3.

The part of the application that implements video display, so as reading and display of sensors' parameters was implemented using declarative techniques in the QML language. Fetching of video frames, calculation of their georeference, and sending them to the GeoVideoServer is implemented in C++ part of the application.

QML part of the application relies on the following modules: QtMultimedia 5.0, QtSensors 5.0, and QtPositioning 5.0. Reading device's orientation is done using `RotationSensor` QML element, reading of device's position using `PositionSource` QML element, a camera as the video source is represented by the `Camera` QML element, while video display is done using `VideoOutput` QML element. Fetching camera frames are done in the C++ part of the app using class `QVideoProbe` that is instantiated in `GeoVideoEncoder` class that serves as a data aggregator. In Listing 1 we included QML code snippet that shows how orientation and position reading are transferred to the `GeoVideoEncoder` instance.



**Fig. 3** An architecture of GeoVideoCamera mobile application

To enable robustness when sending georeferenced frames, we relied on a sequence of JPG images. The process of converting video frame into JPG image is done in a separate thread (class `GeoVideoEncoderWorker`) so it does not impact performance of sensor reading task. Since orientation, position and video frame data arrive asynchronously, current georeference is constantly updated and dispatched with a new frame when it becomes available.

It can be noticed that Qt framework provides aggregated method for reading orientation through `RotationSensor` QML element. Data about orientation are presented using `QRotationReading` class [16]. The rotation reading contains three angles ( $x$ ,  $y$  and  $z$ ), measured in degrees that define the orientation of the device in three-dimensional space. These angles are similar to yaw, pitch, and roll but are defined using only right-hand rotation with axes as defined by the right-hand Cartesian coordinate system. The three angles are applied to the device in the following order (see Fig. 4 [16]):

- Right-handed rotation  $z$   $[-180, 180]$ . Starting from the  $y$ -axis and incrementing in the counter-clockwise direction.
- Right-handed rotation  $x$   $[-90, 90]$ . Starting from the new (once-rotated)  $y$ -axis and incrementing towards the  $z$ -axis.
- Right-handed rotation  $y$   $[-180, 180]$ . Starting from the new (twice-rotated)  $z$ -axis and incrementing towards the  $x$ -axis.

**Listing 1** QML code snippet for reading device orientation and position

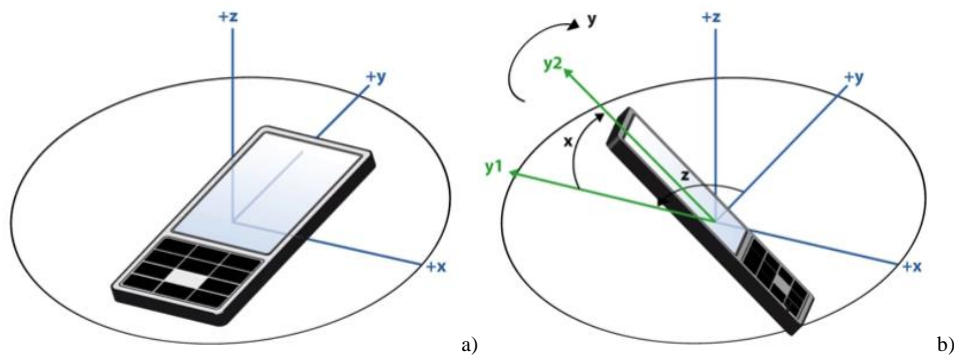
---

```

property double declination: 4.1
RotationSensor {
    active: true
    onReadingChanged: {
        qtGeoVideoEncoder.setOrientation(reading.x, reading.y, reading.z,
                                           parent.declination)
    }
}
PositionSource {
    active: true
    preferredPositioningMethods: PositionSource.AllPositioningMethods
    onPositionChanged: {
        if (position.latitudeValid && position.longitudeValid &&
            position.altitudeValid)
        {
            qtGeoVideoEncoder.setPosition(position.coordinate.latitude,
                                             position.coordinate.longitude,
                                             position.coordinate.altitude)
        }
        if (position.magneticVariationValid)
            parent.declination = position.magneticVariation
    }
}
Camera {
    id: camera
    captureMode: Camera.CaptureViewfinder
    focus.focusMode: Camera.FocusContinuous
    focus.focusPointMode: Camera.FocusPointAuto
    Component.onCompleted: {
        qtGeoVideoEncoder.setCamera(camera)
    }
}
VideoOutput {
    anchors.fill: parent
    source: camera
    fillMode: VideoOutput.PreserveAspectFit
}

```

---

**Fig. 4** An explanation of Qt's rotation sensor readings



To convert rotation readings to previously described *azimuth*, *pitch* and *roll* parameters required for georeferencing a frame, the following rotation matrix  $R$  is constructed:

$$R = R_z(z) \times R_x(x) \times R_y(y) \times R_z(-\pi/2) \times R_x(-\pi/2) \times R_z(-dec). \quad (2)$$

In the previous formula,  $R_x$ ,  $R_y$ , and  $R_z$  are used to represent rotation matrices (3). The subscript determines around which axis the rotation is taken, while the parameters in the parentheses are the angles of rotation measured in a counter-clockwise direction. Parameters  $x$ ,  $y$  and  $z$  are previously described readings gathered from the rotation sensor, while parameter  $dec$  represents magnetic declination in the current position.

$$R_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{bmatrix}, \quad R_y(\varphi) = \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix}, \quad R_z(\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Based on the calculated rotation matrix  $R$ , vectors that determine camera orientation in the absolute coordinates of the 3D GIS are calculated with the following formulae:

$$\begin{aligned} \overrightarrow{V_{look}} &= [x_{look} \quad y_{look} \quad z_{look}]^T = R \times [0 \quad 1 \quad 0]^T, \\ \overrightarrow{V_{side}} &= [x_{side} \quad y_{side} \quad z_{side}]^T = R \times [1 \quad 0 \quad 0]^T, \\ \overrightarrow{V_{up}} &= [x_{up} \quad y_{up} \quad z_{up}]^T = R \times [0 \quad 0 \quad 1]^T. \end{aligned} \quad (4)$$

As the subscripts suggest, the first vector  $V_{look}$  determines view direction, while the second  $V_{side}$  and the third  $V_{up}$  determine relative right side and up directions, respectively. Absolute orientation parameters can be fully determined by two of these three vectors. Formulae that calculate *azimuth*, *pitch*, and *roll* based on  $V_{look}$  and  $V_{side}$  vectors are the following:

$$azimuth = \arctan\left(\frac{x_{look}}{y_{look}}\right), \quad pitch = \arcsin\left(\frac{z_{look}}{\sqrt{V_{look}}}\right), \quad roll = -\arcsin\left(\frac{z_{side}}{\sqrt{V_{side}}}\right). \quad (5)$$

## 5. EXPERIMENTAL RESULTS

As the result of the implemented prototype of GeoVideoCamera mobile app, Fig. 5 (a), (c), and (e) depict three cases of geospatial video projected into virtual 3D GIS scene that is created within VideoGIS application. Although it is possible to recognize rough overlapping between video display and the aero-photo image, quality of registration, for the purpose of projecting video, is not satisfying. The reason for such deviation can be found in low precision of sensor readings.

In order to estimate the accuracy of georeference's parameters acquired using smartphone's sensors, we compared it with referent manually calculated georeference. To do so, we extended GeoVideoCamera with the ability to save georeferenced frames, and later used our tie-points method for georeferencing fixed surveillance cameras. The quality of established georeferences can be overviewed in Fig. 5 (b), (d), and (f).



a) Case 1 sensor estimated georeference



b) Case 1 referent georeference



c) Case 2 sensor estimated georeference



d) Case 2 referent georeference



e) Case 3 sensor estimated georeference



f) Case 3 referent georeference

**Fig. 5** Smartphone originated geospatial video frames (a, c, e) projected into virtual 3D GIS scene compared to the referent manually georeferenced frames (b, d, f)

Once we had referent values, we were able to compare it with corresponding values obtained using smartphone sensors. Since geographic latitude and longitude values are not appropriate for error representation, we converted those positions to Universal Transverse Mercator (UTM) coordinate system where coordinates are represented in meters. Based on differences calculated for each of the three cases, we calculated sample standard deviation for each of the parameters. Appropriate results are shown in Table 2.

**Table 2** Overview of the georeference parameters estimation errors

Param.	Case 1			Case 2			Case 3			Standard Deviation
	Est.	Ref.	Diff.	Est.	Ref.	Diff.	Est.	Ref.	Diff.	
<i>lat</i>	43.3231680	43.3230651		43.3308309	43.3307556		43.3312637	43.3313397		
<i>lon</i>	21.9303367	21.9304257		21.8928929	21.8928816		21.8916177	21.8917139		
<i>UTM<sub>x</sub></i>	575431.79	575439.12	-7.331	572386.72	572385.90	0.825	572282.83	572290.54	-7.705	<b>7.543 m</b>
<i>UTM<sub>y</sub></i>	4797123.27	4797113.03	10.241	4797941.13	4797932.76	8.371	4797988.09	4797996.61	-8.523	<b>11.127 m</b>
<i>alt</i>	238.830	227.940	10.890	200.930	199.552	1.378	236.883	223.587	13.296	<b>12.192 m</b>
<i>azimuth</i>	180.929	181.221	-0.292	319.957	314.933	5.024	341.775	344.049	-2.274	<b>3.905 °</b>
<i>pitch</i>	-33.222	-32.571	-0.651	0.020	1.137	-1.117	-28.633	-27.160	-1.473	<b>1.386 °</b>
<i>roll</i>	-2.935	0.170	-3.105	-0.911	-0.844	-0.067	0.860	3.666	-2.806	<b>2.959 °</b>
<i>fov</i>	70.000	71.375	-1.375	70.000	78.669	-8.669	70.000	71.014	-1.014	<b>6.248 °</b>

Although this experiment was conducted on a small set of samples, some trends can be identified and explained. Firstly, there is a significant error in position estimation (*UTM<sub>x</sub>*, *UTM<sub>y</sub>*, and *alt*). The main reason for that is GPS precision and non-ideal satellite coverage in cases 1 and 3 where photos are taken from a window of a building. The *azimuth* parameter also has bigger deviation because it is derived from magnetometer's readings which are highly influenced by local fluctuations in a magnetic field. The *pitch* and the *roll* parameters tend to be more stable because they are derived from accelerometer's readings that are indifferent to the environment and mainly influenced by hand vibrations.

Finally, it is interesting to mention that there's significant deviation with *fov* parameter that should be constant. That should remind us, that referent values are the result of numerical method estimation. It means that they may contain a significant error as long as the result "looks good" (small error in tie-points matching).

## 5. CONCLUSIONS

In this paper, we presented a method for georeferencing mobile devices' video based on the use of built-in sensors for positioning and orientation. Development of mobile devices (i.e. smartphones) in the past few years is marked with a trend of integration of an increasing number of sensors. These sensors can be used by applications and services to observe different characteristics of the device's environment.

For the purpose of georeferencing and transmitting geospatial video we relied on the following components of a smartphone:

- Built-in camera for fetching video frames
- GNSS receiver for determining device's position
- Accelerometer and magnetometer for determining device's absolute orientation
- Wi-Fi transceiver for sending georeferenced video frames

In order to validate proposed solution, so as the precision of the georeference obtained in this manner, we implemented mobile application GeoVideoCamera. This simple application reads video frames and appropriate sensor data, calculates a georeference and sends geospatial video. The application connects with existing GeoVideoServer, while for the display of geospatial video we use previously implemented VideoGIS application [12].

The implemented prototype confirmed that smartphones could be used as the source of geospatial video. It also confirmed openness of the architecture of our GIS-based video surveillance for the inclusion of different types of geospatial video sources. On the other hand, precision and availability of information about position turn out to be a limiting factor in the potential use of such system. The most critical parameter in this matter is the altitude. The altitude is gathered from a GNSS receiver, and it is very unprecise. The problem becomes especially significant when the device is at a low altitude relative to the ground (e.g. pedestrian on the street).

An interesting application of the proposed system, and the one where the impact of identified problems would be minimal is to mount the mobile device to an unmanned aerial vehicle (UAV), commonly known as a drone. The future experiments, if conditions allow, could go in this direction.

Analyses of a possible application for this kind of geospatial video would also be interesting to investigate in the future having in mind significant presence of such data sources. Standardizing communication protocol for this data source according to the Open Geospatial Consortium (OGC) SensorThings API [17] specification would be another direction in the future work.

**Acknowledgement:** *This paper was realized as a part of the projects "Studying climate change and its influence on the environment: impacts, adaptation and mitigation" (III 43007), financed by the Ministry of Education and Science of the Republic of Serbia within the framework of integrated and interdisciplinary research for the period 2011-2014.*

#### REFERENCES

- [1] U. Schatzberg, L. Banin, Y. Amizur, "Enhanced WiFi ToF indoor positioning system with MEMS-based INS and pedometer information", in Proc. of Position, Location and Navigation Symposium – PLANS 2014, 5-8 May 2014, Monterey, CA, pp. 185-192, [Online]. Available: <http://doi.org/10.1109/PLANS.2014.6851374>
- [2] B. Shin, S. Lee, C. Kim, J. Kim, T. Lee, C. Kee, S. Heo, H. Rhee, "Implementation and Performance Analysis of Smartphone-based 3D PDR System with Hybrid Motion and Heading Classifier", in Proc. of Position, Location and Navigation Symposium – PLANS 2014, 5-8 May 2014, Monterey, CA, pp. 201-204, [Online]. Available: <http://doi.org/10.1109/PLANS.2014.6851376>
- [3] S. Lee, B. Koo, M. Jin, C. Park, M.J. Lee, S. Kim, "RangeFree Indoor Positioning System Using Smartphone with Bluetooth Capability", in Proc. of Position, Location and Navigation Symposium – PLANS 2014, 5-8 May 2014, Monterey, CA, pp.657-662, [Online]. Available: <http://doi.org/10.1109/PLANS.2014.6851428>

- [4] I. Aicardi, P. Dabove, A. M. Lingua, M. Piras, “Sensors integration for smartphone navigation: performances and future challenges”, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XL-3, 2014, pp. 9-16, [Online]. Available: <http://doi.org/10.5194/isprsarchives-XL-3-9-2014>
- [5] P. Lewis, S. Fotheringham, A. Winstanley, “Spatial video and GIS”, *International Journal of Geographical Information Science*, Vol. 25, No. 5, pp. 697-716, [Online]. Available: <http://doi.org/10.1080/13658816.2010.505196>
- [6] S. Min, L. Mei, Z. Feizhou, W. Zhipeng, W. Daozheng, “Hybrid Tracking for Augmented Reality GIS Registration”, in *Proc. of the 2007 Japan-China Joint Workshop on Frontier of Computer Science and Technology*, 1-3 November 2007, Wuhan, China, pp. 139-145, [Online]. Available: <http://doi.org/10.1109/FCST.2007.20>
- [7] G. Sourimant, L. Morin, K. Bouatouch, “GPS, GIS and video registration for building reconstruction”, in *Proc. of the 2007 IEEE international conference on image processing (Vol. 6)*, 16 September – 19 October 2007, San Antonio, TX, pp. 401–404, [Online]. Available: <http://doi.org/10.1109/ICIP.2007.4379606>
- [8] G. Schall, E. Mendez, E. Kruijff, E. Veas, S. Junghanns, B. Reitinger, D. Schmalstieg, “Handheld Augmented Reality for underground infrastructure visualization”, *Personal and Ubiquitous Computing*, Vol. 13, No. 4, pp. 281–291, [Online]. Available: <http://doi.org/10.1007/s00779-008-0204-5>
- [9] Free On-Line Dictionary Of Computing (FOLDOC), “video”, <http://foldoc.org/video>, 2010.
- [10] National Aeronautics and Space Administration, Aircraft Rotations (Body Axes), <http://www.grc.nasa.gov/WWW/K-12/airplane/rotations.html>, 2008.
- [11] A. Milosavljević, A. Dimitrijević, D. Rančić, “GIS-augmented video surveillance”, *International Journal of Geographic Information Science*, Vol. 24, No. 9, 2010, pp. 1415-1433, [Online]. Available: <http://doi.org/10.1080/13658811003792213>
- [12] A. Milosavljević, D. Rančić, A. Dimitrijević, B. Predić, V. Mihajlović, “Integration of GIS and video surveillance”, *International Journal of Geographic Information Science*, Vol. 30, No. 10, 2016, pp. 2089-2107, [Online]. Available: <http://doi.org/10.1080/13658816.2016.1161197>
- [13] J. Jongerius, “Measuring Lens Field of View (FOV)”, *Pano Help*, [Online]. Available: <http://www.panohelp.com/lensfov.html>, 2009.
- [14] B. Delporte, L. Perroton, T. Grandpierre, J. Trichet, “Accelerometer and magnetometer based gyroscope emulation on smart sensor for a virtual reality application”, *Sensor & Transducers Journal*, Vol. 14, 2012, pp. 32-47, [Online]. Available: <https://hal-upec-upem.archives-ouvertes.fr/hal-00826243>.
- [15] The Qt Company, *Qt Framework Homepage*, <http://www.qt.io>, 2016.
- [16] The Qt Company, *QRotationReading Class*, <http://doc.qt.io/qt-5/qrotationreading.html>, 2016.
- [17] Open Geospatial Consortium, *OGC SensorThings API, Part 1: Sensing*, Version 1.0, [Online]. Available: <http://docs.opengeospatial.org/is/15-078r6/15-078r6.html>, 2016.