Mobile Solution for Estimating the Movement of Device

O. Kainz, F. Chovanec, M. Michalko, and F. Jakab

***Abstract*—The main aim of this paper is to propose the approach for visual estimation of selected parameters of the skeletal model. The outputs of this estimation form nodes of a human skeleton from a predefined skeletal model. The paper also analyzes mobile position sensors and also existing position sensing solutions based on a visual approach. This analysis introduces the physical principle of mobile device sensors and their use in practice. It also introduces algorithms and existing position sensing solutions. Experimental implementation is focused on the functionality of the designed software solution, tested for Android operating system. Testing was carried out in the laboratory environment and focused on the accuracy of height measurement and angular measurement. The accuracy of the measurement using different devices is evaluated and the output of the solution utilizes standardized novel human body description format.**

***Index Terms*—Capturing position, skeletal model, ARToolkit, android.**

I. Introduction

Computer vision is the scientific area that improves every year. New techniques are developed to improve the accuracy of the measurement of the human body dimensions. The skeletal model represents the simplified model of the person in the picture and allows the differentiation of parts of the human skeleton. This model is suitable for describing a human skeleton and has multiple uses. This paper deals with the solution of measurement of the skeletal model using a mobile device.

There are currently several approaches to measuring the height and determination of a human skeletal model, however there is no mobile software solution for such purpose. Therefore, the aim of this paper is to develop a software solution for height measurement, which is used to measure the human skeletal model. This skeletal model measurement is based on a visual approach using the square marker and the ARToolkit library. The result of the work is a functional software solution that is ready for use.

The first step in the solution was to carry out analysis of the position sensors available on smartphone mobile devices. This part of the analysis deals with the basic position and rotation sensors in mobile devices, their use and basic principles. Another part is the analysis of existing solutions for position capture based on visual approach. In this analysis positioning algorithms and various research work in position determination are described. The design of the individual components of the application both visually and functionally is presented next. The essential part is the proposal of the estimation of the position of the device. The main part of the paper is the implementation of the elaborated proposal.

II. Analysis of Sensors

This analysis is devoted to position sensors of smartphone mobile devices and available existing positioning solutions using cameras.

A. Position Sensors on Android Devices

This section describes some mobile device sensors that are used to orientate the device in space. These sensors include an accelerometer, a gyroscope and a magnetometer. They enable sensing the the rotation, inclination, acceleration or with magnetometer sense position relative to the Earth's magnetic field.

1) Accelerometer

Research [1] indicates that an accelerometer is a device for measuring the acceleration, either static or dynamic. It is used to measure the angle of rotation, tilt, shake or gravity. In mobile devices is used a 3-axis accelerometer and measures acceleration in x (lateral), y (longitudinal), and z (vertical) axes. The accelerometer measures the acceleration vector in units of m/s^2. If the device is in the stationary position, the vector size is approximately $g = 9.81\ m/s^2$. Even when the mobile device is at rest the accelerometer gives a non-zero output. If the device deviates from the stationary position the vector size of the x, y and z axes vary according to the velocity.

2) Gyroscope

According to the study of mobile sensors [2], the gyroscope is a rotation measurement device based on the principles of angular momentum. The measurement of device rotation runs in 3 axes and in *rad/s* units. When the device is at rest, gyroscope data shows zero. Mechanically, the gyroscope is a metal ring or a metal disc whose axis is free and determines the direction of orientation. The gyroscope maintains the rotary axis in a steady direction unless the external force acts on it. If it acts on it, the gyroscope tries to balance its own rotation axis with the rotation of the earth.

3) Magnetometer

Based on the study [3], it can be said that the magnetometer measure changes in the Earth's magnetic field. Provides magnetic field data reported in μT (*microTesla*) units. The magnetometer sensor on the mobile device is based on a Hall effect sensor that detects a 3-axis Earth's magnetic field. The Hall effect sensor measures the

Manuscript received February 9, 2018; revised March 15, 2018.

The authors are with the Department of Computers and Informatics, Technical University of Košice, Kosice, Slovakia (e-mail: ondrej.kainz@cnl.sk, filip.chovanec@student.tuke.sk, miroslav.michalko@cnl.sk, frantisek.jakab@cnl.sk).

doi: 10.18178/ijfcc.2018.7.1.510

magnetic field by generating a voltage that is proportional to the force and polarity of the Earth's magnetic field. This voltage is then transferred through electronic circuits to the values in which the magnetic field is measured. In a book [4] it is written that the magnetometer is most commonly used in conjunction with an accelerometer to produce an orientation sensor. The orientation sensor is a software sensor and measures the position of the device relative to the Earth's magnetic field.

B. *Analysis of Existing Solutions for Position Capture*

Several approaches have been created for position capture. This section describes some of these procedures and algorithms to determine the position of a particular device, such as robots or various camera sensing devices. The camera is used to capture the location, either by using the camera of the mobile phone or by using separate cameras.

1) *POSIT algorithm*

Based on a book publication [5], the POSIT algorithm (aka "Pose from Orthography and Scaling with Iterations") is used to calculate the position (position T and orientation R) of a 3D object of known dimensions. To calculate the position, it is necessary to find at least 4 non-coplanar points on the surface of the object. The first part of the algorithm determines the 3D position of an object based on scalability. A weak-perspective approximation approach is used. By supplying the known internal parameters of the camera to algorithm it is possible to determine the perspective of the object and thus to calculate the approximate position. This calculation is not very accurate, so it is necessary for this calculation to take place several times, typically 4 to 5 times.

2) *Position capture with LED visible light*

In paper [6] visible LED light and accelerometer are used to determine the position. The internal parameters of the light sensor and also the base position must be known. The positioning principle is based on capturing two different color LEDs lights that are at the same height and their position in the real world is the input for position calculation. The position calculation is based on three data:

- the real position of two color lights,
- the position of two color lights on the picture,
- accelerometer data.

This method has a positioning error less than 5 cm.

3) *Height determination using stereoscopy*

Koleda and Naščák in their paper [7] described a method for sensing body dimensions based on stereoscopy. Stereoscopy is the use of two cameras scanning the same scene from different angles. This shot can create a 3D model and get all 3 body dimensions. Stereoscopy gives two shots of the same scene from two different angles using a stereoscopic device. The captured images are then synchronized using the specified synchronization points. The height calculation is simplified by using trigonometry. This method allows fast image processing and thus fast measurement of body dimensions.

4) *Position capturing using a QR code*

Lee *et al.* [8] presented a proposal for robot movement based on QR code. QR codes are easily recognized by mobile device and these QR codes are strategically located in the operating environment. Each of them has its *x* and *y* coordinates in the real world as well as their unique identifier stored in the database. The mobile device serves as a QR code detector and is attached to the robot. Once a QR code is found, the mobile device via WiFi connects to an external database and obtains the coordinates of the QR code. The QR code is divided into several angular portions ranging from 0° to 359.9°. After reading the QR code, it gets 3 coordinates from the image. Together with coordinates from the database, it can detect the position of the robot and determine the next move step.

III. Proposal of Software Solution

This section describes how to capture a device's position using the camera on a mobile device running Android. OpenCV or ARToolkit library can be used to capture device’s location on smartphone devices. ARToolkit is used to design the application, since it offers enhanced possibilities to process markers and videos.

A. *ARToolkit Library*

According to the documentation [9] ARToolkit is open-source library primarily intended for application development in Augmented Reality.

ARToolkit uses square markers to track the user's position or 2D barcodes can be also used. Square markers have their prescribed shape. It forms a black pattern on a white background, which is bounded by a black border with a defined width. The inner pattern should be as simple as possible and centrally symmetrical for better orientation in the space. When multiple markers are used, they must be different so that they do not interfere with each other. It is necessary to choose the appropriate marker size that depends on its particular use in the application.

B. *Proposal of Graphical User Interface*

The mobile software solution should contain a simple user interface. This interface offers options such as measuring a human skeletal model, loading a measured model, and offering a marker download to a mobile device. At the same time, each option includes help statements for the user informing them of the success of the action or of an error.

C. *Proposal of Algorithm for Height Measurement*

For the needs of height measurements using the camera can be used the principle of similarity of triangles or OpenGL transformation matrix which ARToolkit library. Each of these options has its advantages and disadvantages as described below.

1) *Similarity of triangles*

In a project [10] the principle of similarity of triangles is used to calculate the height. If the size of one side of the square *W* is known, then it is necessary to give the camera to the calibration height *D*. After obtaining the marker width in the picture in pixels *P* and entering the size *D* of the height calibration it is possible to calculate the focal length *F* of the camera:

$$F = (P * D) / \mathrm{W} \qquad (1)$$

Once this focal length is obtained, it is possible to obtain any distance from the marker after fitting into a height calculation formula based on the principle of triangular similarity:

$$D´= (W *F) / P \tag{2}$$

Thus, the height is calculated by multiplying the actual marker size and focal length of the camera, which is divided by the width of the marker in the image. The disadvantage of this solution is the need to calibrate the program before measuring the height. This calibration is the measurement of the same height when the program is turned on.

2) OpenGL transformation matrix

OpenGL uses for transformation a matrix of size 4x4. This matrix is shown in the equation (3) where the first column is for the *x* axis, the second for the *y* axis, the third column for the *z* axis and the last column contains the shift data. This last column can be used for height detection using marker and a mobile device's camera. The value of m_{12} is a shift in the *x* axis; m_{13} is a shift in *y* axis and m_{14} in *z* axis, which is the searched height between marker and mobile´s camera.

$$\begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix} \tag{3}$$

The ARToolkit library uses a transformation matrix to work with OpenGL video, from which the distances between the marker and the device can be extracted. Therefore, it is best suited for software design solution.

IV. Implementation of Software Solution

This section presents the implementation of the proposed application for device positioning, and also describes output format for data in XML.

Application called *Skeleton app* was developed in the environment Android Studio. For positioning purposes, the application uses a square marker, specifically the Hiro marker. Its size is 10 cm and the initialization code for the marker scan is as follows:

```
ARToolKit.getInstance().addMarker("single;Data/patt.
hiro;100");
```

Because the application uses only one marker, the first parameter is „*single*“, which specifies that only one square marker is initialized. The path to this marker follows in the app, and the last parameter is the number that determines the actual size of the printed marker. This value is in millimeters, which in this case is 100 mm.

A. Implementation of Skeletal Model Measurement

Implementation of the skeletal model measurement is based on the proposal in the Section III. The first step is to turn on the camera of the mobile device and find the marker from which data is extracted into the transformation matrix. This section is implemented in the following code fragment.

```
if (ARToolKit.getInstance().queryMarkerVisible(mark
erID)) {
float[] markerDistance = ARToolKit.getInstance().qu
eryMarkerTransformation(markerID);
((rendererCallback)mActivity).onDistanceCalculated(
 markerDistance[14], markerDistance[12],
markerDistance[13]);
}
}
```

Using the first method on line 3, the OpenGL transformation matrix is obtained from the Hiro marker from which the position data is extracted in the ***onDistanceCalculated*** method. This method has 3 parameters that indicate the distance in all 3 axes. The parameter ***markerDistance[14]*** indicates the distance in the z-axis direction, which is the searching height from the ground. *The* ***markerDistance[12]*** follows as a distance in the x-axis and the ***markerDistance[13]*** in the y-axis from the center of the Hiro marker. Since the measured height is in millimeters, it is necessary to make the conversion from millimeters to centimeters.

For easier deployment, the *HashMap <String, String>* method is used, where is used the ID + measured value. The measurement model contains 12 buttons, from each button has its unique ID and every ID is marked on skeletal model. After pressing the button, the auxiliary variable *isMeasuring* is set to *true* and the point is being measured. This measurement takes 3 seconds and values are stored in the list during this measurement. At the end of the measurement, the mobile device vibrates to indicate the end of the measurement and sets the value of the *isMeasuring* variable to *false*. It then goes through this list of values and calculates the average value which saves in the XML structure.

1) Once all of the skeletal model values have been measured, this model can be saved to the storage device in XML format. Implementation of the saving of this model is described in the following subsection.

B. XML Format

The human body model is in XML human body description format (HBDF) which was designed in the dissertation work [11] and its structure is described below.

This XML contains 3 structures:

- Information about measuring device - information about the measuring device is pre-defined and contains a unique device *ID* and device *name*. Application device ID is chosen number *4* and name of device is *Skeleton app*. This structure uniquely identifies the measuring device.
- Information about measured person - the structure contains information about the person such as ID, name, weight, height, age and date of measurement.
- Measured data - the structure contains data of human skeletal models and is divided into three parts according to the human skeleton.

Each point contains an indication of the point height from the ground in centimeters. If there is no point measured, its value is set to *-1*. This value was chosen on the basis that the measured height cannot reach negative values.

V. Software Solution Testing

This section describes application testing and experimental deployment in a real environment. Testing focuses on the accuracy of the specified height measurement using the camera and the overall functionality of the application. Application testing was performed on mobile devices Nexus 5X and Xiaomi Mi 4.

A. Testing the Accuracy of the Height Measurement

The following testing is focused on the accuracy of the height measurement on two different devices. The same marker was used on both mobile devices and the measurements were running simultaneously.

This test consists of 10 measurements from the initial 20 cm to 200 cm where the increment is 20 cm. For measurement on the Nexus 5X was selected a camera resolution of *1600 × 1200px* and on the Xiaomi Mi 4 resolution *1920 × 1080px*. During measurements it was most important to measure the heights from the center of the Hiro marker to the end of the camera lens of the mobile device. The measurement results for both devices are shown on Table I.

TABLE I: Accuracy Measurement of Height Estimation

	Accuracy measurement			
	Nexus 5X		***Xiaomi Mi 4***	
Real height	**Measured height**	**Deviation**	**Measured height**	**Deviation**
20 cm	20,3 cm	1,96 %	20 cm	0 %
40 cm	40,6 cm	1,48 %	39,9 cm	0,25 %
60 cm	60,9 cm	1,50 %	59,8 cm	0,33 %
80 cm	81,4 cm	1,75 %	80,1 cm	0,50 %
100 cm	102,1 cm	2,10 %	100,8 cm	0,79 %
120 cm	122,9 cm	2,42 %	121,3 cm	1,07 %
140 cm	144 cm	2,86 %	141,5 cm	1,06 %
160 cm	165,2 cm	3,25 %	162 cm	1,23 %
180 cm	186 cm	3,33 %	183 cm	1,64 %
200 cm	207,1 cm	3,55 %	203,8 cm	1,86 %

The Xiaomi Mi 4 mobile device has a camera resolution of 13 Mpix and an average deviation of 0.87%. The Nexus 5X mobile device has a camera resolution of 12.3 Mpix and an average deviation of 2.37%. Result from the measurement is, that the accuracy of the height measurement depends on the resolution of the camera's mobile device. It is also important that the shadow of the person or mobile device does not overlap with the Hiro marker. If the shadow overlaps, data is distorted, and less accurate height calculations occur.

B. Testing the Accuracy (Camera Rotation)

The following testing focuses on the accuracy of the angle measurement between the mobile device and the Hiro marker. As a testing device was used Xiaomi Mi 4 and the constant test distance was 50 cm. The measurement consists of 6 measurements from the initial 90°, i.e. the perpendicular position to the final angle of 45°. The accuracy results of measuring the angular measurement are in Table II.

TABLE II: Accuracy of the Angular Measurement

Xiaomi Mi 4		
Rotation	**Measured height**	**Deviation**
90°	50,1 cm	0,2 %
80°	49,7 cm	0,6 %
70°	49,3 cm	1,42 %
60°	47,6 cm	5,04 %
50°	46 cm	8,69 %
45°	45,6 cm	9,65 %

The results show that the most accurate results of measuring the position are at a perpendicular (90°) measurement. This is due to the fact that the dimensions of the marker are not distorted and so the algorithm can more accurately calculate the height between the mobile device and the marker. Therefore, it is necessary to measure the height and measure the data into the skeletal model in the range between 90° and 80°. The deviation starts to be too large when measuring is provided at an angle of less than 70°.

C. Evaluation of the Software Solution

The mobile application has been tested on 2 different mobile devices. The proposed solution and implementation is the most appropriate solution for mobile devices with regard to accuracy of measurement compared to calculations by using accelerometer and gyroscope. From the measurement results it can be determined, that the accuracy of measuring the position of the device depends on the resolution of the camera of the device and on the angle of measurement. Therefore, it is necessary to scan the position with respect to the marker in 90° angle to achieve the most accurate results. The height calculation can also distort the overlap of the shade of the device or another shade with the marker part.

The advantage of the application is its easy use, export of data into XML format that can be used for further processing and its mobility. The application can be used wherever it is needed. Another advantage is that it runs on most used mobile versions of the Android operating system.

The disadvantage of the application is a need to use the mobile device with high resolution cameras for the most accurate results. This is due to the camera height calculation, where using weaker mobile cameras can cause distortion and poor calculation.

VI. Conclusion

The one of the tasks in this paper was to analyze mobile position sensors. These sensors, which were analyzed, include accelerometers, gyroscopes and magnetometers. In this analysis are explained the functions and physical principles of their operation as well as the possibilities of application in real deployment. The other analyzed part was the existing position capture solutions based on the visual approach. This section describes various camera positioning algorithms and approaches for detecting the position of the scanning device, such as robots.

The principal goal of this paper was to design a software solution, all its functional elements, a graphical interface and a way of measuring the height using a mobile device and lastly the way of inserting this data into a skeletal model. Consequently, according to the proposal solution was implemented in the form of a mobile application on the chosen mobile platform, specifically Android. The mobile application can measure the height at which the device is positioned relative to the square marker stored on the ground and record that height. With this application the user can measure the whole human skeleton and insert the data into a predefined skeletal model. The last step was to save the model to an output XML format that can be used for further processing.

The contribution of this paper is the development of an informatics solution for the measurement of selected parameters of the skeletal model, which resulted in the creation of a mobile application. Measurement of a human skeletal model uses a combination of a mobile device and a square marker, namely Hiro marker. Using the mobile application is possible to measure the human skeletal model; it means different limbs and important points on the human body. These data (in novel HBDF format) can be saved and further processed.

From the results of the experimental test it was found that the accuracy of the height measurement depends on the resolution of the mobile camera and also on the angle of sensing the square marker. The results show that the average application deviation is less than 2% depending on the resolution of the mobile device's camera. It is also necessary to have a sufficiently illuminated marker for correct calculation and ensure that the marker is not shaded by the mobile device.

One of the possible extensions of the mobile application is to improve the application's graphical interface. In the future, it is also possible to add a function of automatic measurement of skeletal model points that would collect the height data at certain time intervals.

Acknowledgment

This publication is the result of the Project implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology, Phase II., ITMS: 313011D232, supported by the Research & Inovation Operational Programme funded by the ERDF. We support research activities in Slovakia/This project is being co-financed by the European Union.

References

[1] P. H. Lehne, A. A. Glazunov, K. Mahmood, and P. S. Kildal, "Analyzing smart phones' 3D accelerometer measurements to identify typical usage positions in voice mode," in *Proc. 2016 10th European Conference on Antennas and Propagation (EuCAP),* Davos, 2016, pp. 1-5.

[2] M. Liu, “A study of mobile sensing using smartphones,” *International Journal of Distributed Sensor Networks*, p. 11, 2013.

[3] Z. Ma, Y. Qiao, B. Lee, and E. Fallon, "Experimental evaluation of mobile phone sensors," in *Proc. 24th IET Irish Signals and Systems Conference (ISSC 2013),* Letterkenny, 2013, pp. 1-8.

[4] V. Nagpal, *Android Sensor Programming By Example*, Birmingham: Packt Publishing Ltd, 2016.

[5] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, Sebastopol: O'Reilly Media, Inc., 2008.

[6] T. Tanaka and S. Haruyama, "New position detection method using image sensor and visible light LEDs," in *Proc. 2009 Second International Conference on Machine Vision, Dubai*, 2009, pp. 150-153.

[7] P. Koleda and Ľ. Naščák. (2012). Meranie výšky telesa pomocou dvoch kamier. [Online]. Available: https://www.tuzvo.sk/files/FEVT/fakulta_fevt/akta_fevt-2-2012-koleda-nascak.pdf

[8] S. J. Lee, G. Tewolde, J. Lim, and J. Kwon, "QR-code based localization for indoor mobile robot with validation using a 3D optical tracking instrument," in *Proc. 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Busan, 2015, pp. 965-970.

[9] ARToolKit Documentation. [Online]. Available: https://artoolkit.org/documentation/

[10] A. Rosebrock. (January 19, 2015). Find distance from camera to object/marker using Python and OpenCV. [Online]. Available: http://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/

[11] O. Kainz, “Pokročilé vizuálne a nevizuálne prístupy k určeniu parametrov viacrozmerných objektov z reálnej scény,” Ph.D. dissertation, Košice, TUKE, 2017.

Ondrej Kainz was born in 1988. In 2017, he received a PhD degree in computer science from the Technical University of Kosice. Since the very same year, he is an assistant professor at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at the Technical university of Kosice. His scientific research interests include computer vision, image processing, teaching, and computer networks.

Miroslav Michalko received his M.Sc. and Ph.D. degree in informatics from Technical University in Kosice (Slovakia). For more than 10 years he is a member of well recognized research institution - Computer Networks Laboratory at the Department of Computers and Informatics at Technical University of Kosice (DCI TUKE). Now he is an assistant professor at DCI TUKE and his lectures are focused on Computer Networks, he is more than 10 years active instructor in Cisco Networking Academy. His research includes multimedia content delivery, video streaming services, web and cloud services, multimedia on mobile devices, innovative teaching&learning techniques and IoE/IoT solutions.

František Jakab was born in 1959. He graduated from St. Petersburg Electro technical University, PhD degree received from Technical university of Kosice & is associated professor since 2008. He has extensive experience in networking and utilization of ICT in education where he established well known research centre - Computer Networks Laboratory. He has been a coordinator of several large international projects, coordinator of national wide ICT projects and research grants. He involved into Cisco Networking academy program in Slovakia and since 2001 in position of coordinator of Program in Slovakia, 2008-2014. He was awarded as an “IT person of the year 2006” in Slovakia.