An Android Application for Computing Selected Areas on Images in Restauration of Mosaics and Frescos

Miloš Stošić, Radomir Stanković

Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, Niš, Serbia
milosstosic88@gmail.com, radomir.stankovic@gmail.com

Abstract. This paper describes an Android application which allows computation of the physical area of a selected region of an image. The System is intended for helping archaeologists on fields and archaeological sites. The basic principle of the operations is that the user of the application takes a picture of some shape, object or landscape. First, the user has to select the unit of measurement that is always present on the screen, by using already implemented tools. Then you select an appropriate mosaic that you want to be calculated, and the calculated area is presented on the screen. The major advantage of the system is to resolve the problem of computating the area of the damaged part of the mosaics on field.

Keywords: augmented reality; android; mathematics; archaeology; calculating; mosaics.

1 Introduction

Repairing and conservation of mosaics, frescoes, or similar archaeological artefacts, is a task highly positioned in the list of works in the professional daily routine of an archaeologist. As the first step, this task involves a detailed analysis and classification of damages in order to plan and then perform the necessary repairing and conservation. Depending on the type of damages, positon of the artefact, and other relevant issues, a complex planning of necessary activities, including material and human resources and timing, has to be performed. In practice, the following scenario usually applied. The archaeologist usually takes a photo of a mosaic on the site, and then performs the analysis on the photo in his laboratory by using sophisticated computer and other facilities. As a part of this work, it is necessary to determine the size of various damaged areas in order to plan the material resources and estimate the time required to perform the task. Therefore, analysis and labeling of damaged areas on the photo, should be used to determine the size of the corresponding area on the site. The efficiency and correctness of this procedure is important for planning further steps in the corresponding conservation and reparation procedures. This task has to be frequently performed, and therefore the idea to automatize the task appears. After some discussion with experts in the institutions for protection of historic and cultural heritage, we came up with the idea to create a software that will allow determining the size of area on the site based on the size of the selected area of the screen of the device used to analyze the corresponding photos.

The increasing usage of mobile phones and tablet computers was a guideline for selecting the platform to implement the application. In the current version, the mobile devices under the Android operating system are selected, due to their wide usage in Serbia.

It should be noticed that in our best knowledge, there are no Android applications that can be used in solving this task.

For example, the Android application called "GPS field area measure" is widely known. However, it cannot be used for the task discussed in this paper, due to low preciseness. The precision of the GPS signal varies from 5 to 10 meters, while the size of areas of interest in analysis of the photos of the mosaics and frescoes could be in the range of few centimeters.

Therefore, development of a new application is required which is the main subject discussed in this paper.

The main functionality of the application can be shortly presented as follows. The user takes a photo of an object on the archeological site at the first step of the application. Then, selects the area of interest on the image by a correspondingly implemented tool, after which the user calls from the menu the option that computes the actual size on the field corresponding to the selected area at the image.

2 The principle of work

The main idea enabling to solve the task is to use a marker whose dimensions and size are known. The marker is put at the mosaic before the photo is taken. In this way, we know the ratio between the actual size of the marker and the area it occupies on the photo. The dimension of the photo we easily determine from the mobile device used to take the photo. In this way, when in the analysis we label a particular damaged area at the photo, we can compute its actual size on the site.

For a comparison of technologies and expanding the area of usage of the application, as well as raising the quality of the sys-tem, the required Android application is implemented in two ways:

- By using the Metaio SDK library.
- By creating our own library with all necessary functionalities that we call the ArhiCalculus library.

2.1 *Metaio SDK* library

In the first implementation, the application uses the *MetaioSDK* library. This library is used to determine the distance from the camera of the mobile device to the marker that was placed on the site. The application recognizes the marker. The *MetaioSDK* library already has implemented functions that will be called for calculating the distance.

Also, we read from the device the angle that determines the position of the device relative to the plane. Another very important information that we use is the width of the camera that is different in different phones. These data are used to calculate the area of selected field.

An experimental evaluation of this application based on *Metaio SDK* shows that the produced results are within the range of 90 percents of accuracy. In order to improve the accuracy of computing, another application is developed.

2.2 *ArhiCalculus* library

In the second approach, described in this paper, each part of the application is implemented individually without the usage of additional libraries. Instead of using *Metaio SDK* library we made our system for calculating the size of the area selected on the photo. Also, we had to set id marker or some other marker that will be the unit for measurement. At the beginning, we take a photo of the field. Then, using the drawing tool that is also implemented we select that unit for measurement and calculate its area.

The same drawing tool is used to label the damaged area of the mosaic on the photo. The ratio between the known actual size of the marker and the area the marker occupies on the photo, as well as the size of the labeled area on the photo are used to compute the actual size of the labeled area on the site.

An experimental verification of the application shows that we get results within 95 to 98 percent of accuracy.

3 Implementation of the application

As shown in Fig. 1, the application consists of the following subsystems purposely designed for:

- Taking a photo of the area.
- Determination the real size of the photo.
- Drawing on the screen of the mobile device (drawing shapes, selecting unit of measurement, selecting area for calculation).
- Determination of the representation of the units of measurement on the screen.
- Selecting a part of the image with the help of different forms from the palette to draw.
- Calculating the surface of the selected portion of the photo.

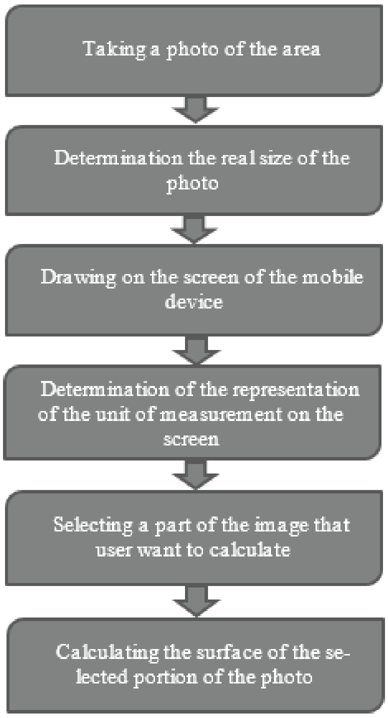

Taking a photo of the area
Determination the real size of the photo
Drawing on the screen of the mobile device
Determination of the representation of the unit of measurement on the screen
Selecting a part of the image that user want to calculate
Calculating the surface of the selected portion of the photo

Fig. 1. Application architecture

3.1 Taking a photo of areas whose surface is calculated

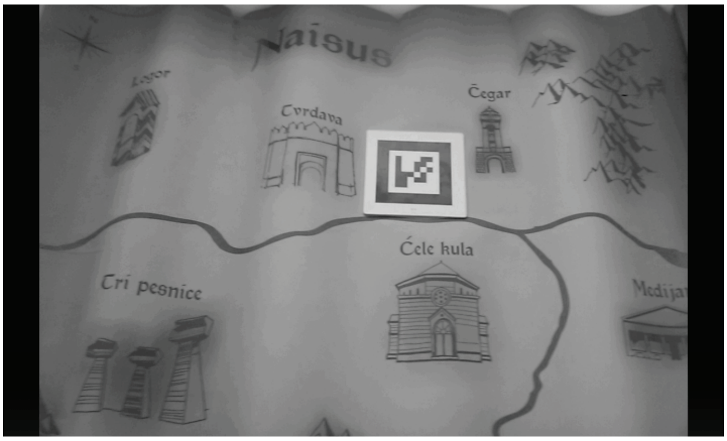


Fig. 2. Home page of the system, the user takes a photo of the region.

The Android Operating System has the support for several cameras that are available on Android devices which allows us to take photos by mobile devices based on this operating system [2]. These photos are used as the input in the application discussed in this paper. The photos are saved in the memory of the device.

In the present version of the application, the angle between the mobile device and the mosaic to be photographed should be around 90 degrees, with a deviation of -5 and 5 degrees. Overcoming this restriction will be a task in further work towards improving the application and its performance.

3.2 Image processing and graphical interface of elements

The uploaded image must have on it the marker which is used as a unit of measurement, as shown in Figure 3.

At the next step we implemented the algorithms for drawing geometrical shapes that will be multifunctional in this application, from the tools for selection a unit of measurement to the pallet of shapes for drawing on display and marking a regions whose areas are calculated.

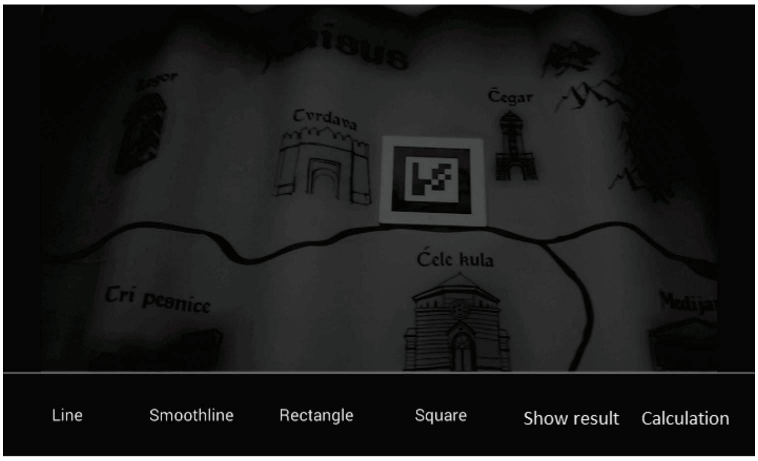


Fig. 3. Pallet of the shapes for drawing and unit of measurement on area.

The class that is implemented for drawing on the screen of the device has a lot of methods that define a way for presenting shapes. All functions in a similar way, the systems detects the touch on the screen and then, depending on the selected options, moving a finger across the screen draws the desired shape.

3.3 Determination of the representation of units of measurement on the screen

Selecting an appropriate geometric shape opens up the possibility of drawing on the screen. In this case, we chose the shape of a rectangle for rendering.

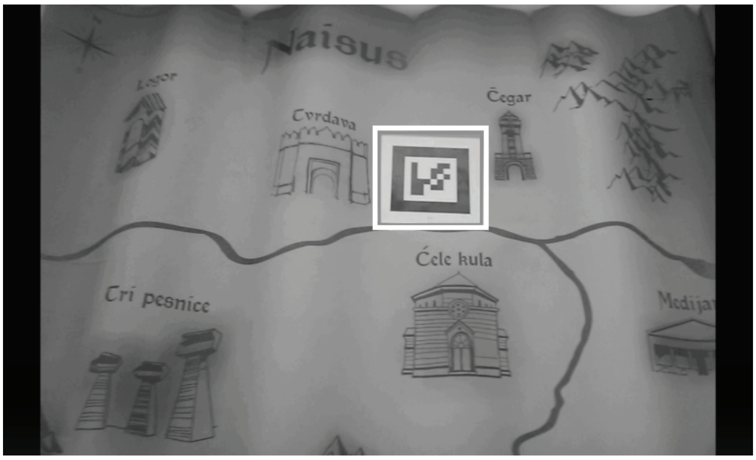
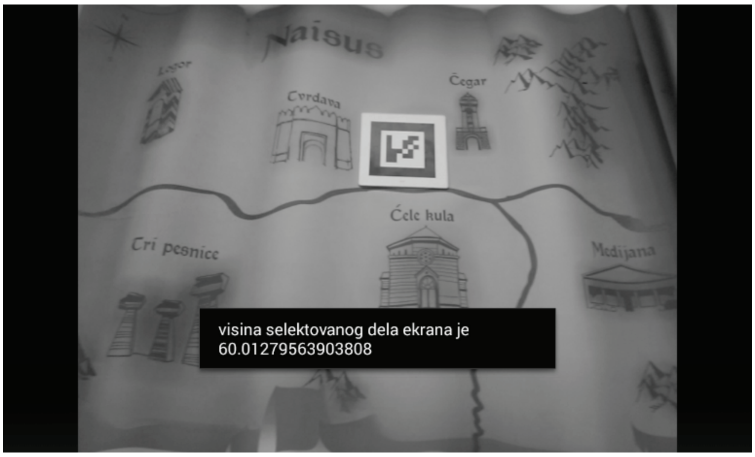


Fig. 4. Selection of a unit of measurement

Marker bordered by the selected geometric shape, as shown in Figure 4, is a unit of measurement that will be used when calculating the real size of the areas. For these calculations we need the features of cameras that we can get from Android devices that we use. We can not use a fixed value for the size of the sensor cameras because it varies for other companies of mobile devices. For example, the devices that we used during testing had two versions of the camera sensor size of 3.08 and 3.42 mm.

Also, the results largely depend on the resolution of the screen of the mobile device on which the calculation is made. Firstly, we need to define the dimensions of the screen, which can be read directly from the device. We can get this value by using real size of the unit and real screen size of a device that use the application. The values that we get from this algorithm are proceed to the next form which calculate real size of the selected shape.



visina selektovanog dela ekrana je
60.01279563903808

Fig. 5. Realistic height of the photographed part of the mosaic.

3.4 Determination of the actual area of the selected part on the screen

The archaeologists want, when on mosaic encounter specific damage, to take a photo of it and only on the basis of one image to obtain a realistic surface part of the mosaic that is damaged.

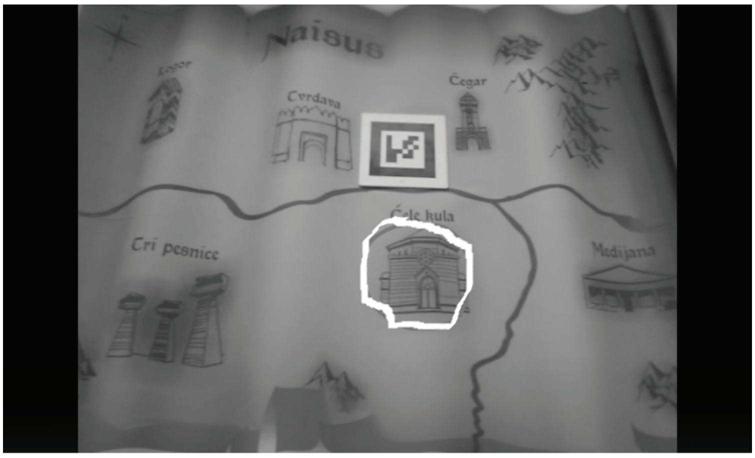


Fig. 6. Selection of the desired part of the mosaic with option "Smooth line".

Drawing “Smooth line”

We came up with the idea that the subsystem for drawing on the screen that is explained in the previous section, can be applied to the calculation of the surface. In the application menu, the user can see a pallet of different shapes for drawing on screen.

However, in a range of forms there is a "Smooth line", as in Figure 6, an option that allows the user to display draws of the free forms and thus gives archaeologist ability to mark a specific landscape mosaic of the surface of which wants to count. The system detects finger touch on the screen. Depends from a shape that have been selected different information are stored in a special system variables. With „Smooth line“ shape, when user touch the screen and move finger every point is stored to one Array variable, that can be used later for calculating area. One of the method for drawing shapes on the screen is presented:

```
private void onTouchEventSmoothLine (MotionEvent event){
switch (event.getAction()) {
                       case MotionEvent.ACTION_DOWN:
isDrawing = true;
                  mStartX = mx;
                  mStartY = my;
                  mPath.reset();
                  mPath.moveTo(mx, my);
                  invalidate();
                  break;
                      case MotionEvent.ACTION_MOVE:
float dx = Math.abs(mx - mStartX);
float dy = Math.abs(my - mStartY);
if (dx >= TOUCH_TOLERANCE || dy >= TOUCH_TOLERANCE) {
mPath.quadTo(mStartX, mStartY, (mx + mStartX) / 2, (my +
mStartY) / 2);
mStartX = mx;
mStartY = my;
XKoordinate[i]=mx;
YKoordinate[i]=my;
i++;}
mCanvas.drawPath(mPath, mPaint);
invalidate();
                      case MotionEvent.ACTION_UP:
isDrawing = false;
mPath.lineTo(mStartX, mStartY);
mCanvas.drawPath(mPath, mPaintFinal);
mPath.reset();
invalidate();
break;
}}
```

There have been created a method that register motion on the screen by finger touch, action down, action up, and action move.

“Smooth line” area calculation.

When calculating the area, as well as the basic idea that we upgraded later, was used Green's integral [3].

Green's Theorem is a higher dimensional analogue of the Fundamental Theorem of Calculus. It relates the double integral over a closed region to a line integral over its boundary.

Let C be a positively oriented, piecewise smooth, simple closed curve in a plane, and let D be the region bounded by C. If L and M are functions of (x, y) defined on an open region containing D and have continuous partial derivatives there.

$$\oint_C (L\,dx + M\,dy) = \iint_D \left(\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} \right) dx\,dy$$

Where the path of integration along C is counterclockwise.

In physics, Green's theorem is mostly used to solve two-dimensional flow integrals, stating that the sum of fluid outflows from a volume is equal to the total outflow summed about an enclosing area. In plane geometry, and in particular, area surveying, Green's theorem can be used to determine the area and centroid of plane figures solely by integrating over the perimeter.

Figure 6 shows results of measurement of an area after selection of an irregular shape on the image calculating the area of se-lected shape.

The accuracy in the calculation area in this manner is from 94% to 99%.

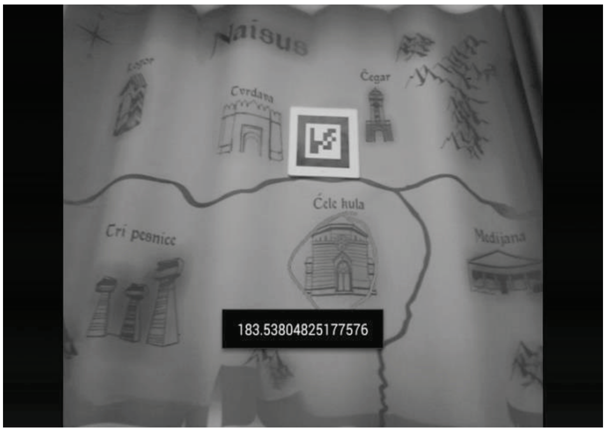


Fig. 7. Displaying value after calculating an area, in cm^2.

4 Testing of the system

We establish shapes of different sizes at different distances from user. It can be concluded that the increase of the distance be-tween the mobile devices increases the percentage of errors in the calculation of the surface. During testing of the system obtained results, real size of the object, calculated object size, percent of error and the distance between user and object were presented at Table 1.

Table 1. Results of testing software

Number	*Real area of the object (cm^2)*	*Result value (cm^2)*	*Error (%)*	*Distance between object and mobile phone (cm)*
1.	184,2153	183,5380	~1	90
2.	54,2235	55,2256	~2	135
3.	2023,9863	1922,4781	~6	450
4.	47,6857	44,8645	~6	450
5.	1950,6538	2015, 9963	~4	250

A problem is that small shapes whose area should be computed, when photographed from large distances will not be visible enough to be accurately marked on the screen. So the error rate, which increase with increasing the surface size depends a lot on the precision and drawing on the screen.

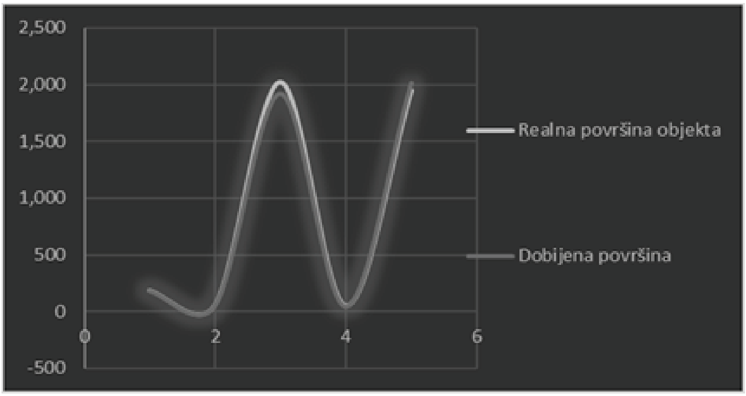


Fig. 8. Graphical display of calculations.

Table 1 and Figure 8, accurately show the percentage of deviations for the third measurement over a large area and a distance of more than 4 m. Figure 8, graphically presents the values obtained after testing the system. X-axis indicates the number of measurements, and the Y axis represents the resulting surface in cim^2.

5 Problems during the realization of the application

During the implementation of this system one of the first problem that we met was how to find a way of computing real size of a shape that have been photographed with camera of the mobile device, only based on a photo. After solving this problem, we get new possibilities and a number of new problems and challenges. Specifically, according to the algorithm that we have developed for calculating the surface several parameters that characterize the mobile device camera must be used, and these parameters are different from device to device.

When the system is implemented using *Metaio SDK* tool, we have to take care of the level of markers recognition, which is determined by the amount of light in the room and the distance from the device to the marker, which should not be beyond certain limits. In the other mode that was not a problem, of whichever amount of light in the room or the distance from device to measurement unit, error rate is still within the prescribed range, which has met all our criteria [4, 5].

6 Conclusion

One of the key problems, take a photo of facility at an angle, led us to a wide range of new research. How to, regardless of the angle of photo of the object, get the screen display as you normally look at the surface. Also, the use of this system is not limited to the archeological sites and archaeologists as the users of this system, but its implementation is far greater, thus the application gets to universality. As improvements for next step realization we will implement database for storing images and results. Also, user will have gallery of images that have been photographed and saved.

Acknowledgements

This work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia (project ON174026).

References

1. J. J. Park, I. Stojmenović, H. Z. Jeong, G. Yi, Computer Science and Its Application, Ubiquitous Information Technologies.
2. R. Vobbilisetty, Camera API – Android, Android developers.
3. G. Green, An Essay on the Application of Mathematical Analysis to the Theories of Electricity and Magnetism (Nottingham, England: T. Wheelhouse, 1828). abstract (Heading 5).
4. M. Stošić, Đ. Manoilov, D. Tatić, „Uporedna analiza karakteristika prepoznavanja markera korišćenjem Metaio SDK i Vuforia SDK", YuInfo 2014, Kopaonik, Srbija.
5. M. Stošić, Đ. Manoilov, „Unapređenje rada sistema koji koristi MetaioSDK, na osnovu količine osvetljenja i rastojanja uređaja od objekta koji se prepoznaje", IEEESTEC 2014, Niš, Srbija