computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

WEB-SERVER APPLICATION FOR FAST AND ROBUST RECOGNITION OF EXHIBITS IN THE MUSEUM

Hanna Balakhontseva [1], Jurgen Sieck [2]

[1] Institute of Computer Systems
Odessa National Polytechnic University
av. Shevchenko 1, 65044 Odessa, Ukraine
Email: a.c.balackhontseva@gmail.com
[2] University of Applied Sciences Berlin
Wilhelminenhofstr. 75A, 12459 Berlin, Germany
Email: j.sieck@htw-berlin.de
URL: inka.htw-berlin.de

Abstract: This document represents a web-server based application designed for the purpose of exhibits' recognition in a museum, based on photos made with the help of phone. Image recognition was conducted with the help of SURF[1] algorithm. Server was implemented with the usage of Netty library. Because of the parallel approach used both in the classification part and during the exchange of data between client and server the system is fast and able to decrease time of working by usage of all available cores of hardware.

Keywords: Netty server, web application, SURF, scalable server, parallel performance.

1. INTRODUCTION

Nowadays, a variety of multimedia technologies surrounds us in our everyday live. Almost everything is performed with its help, from our daily work to leisure activities and entertainment.

Such technologies allow us to interact with any digital information, especially images. It is a known fact that most of the users' photographs are taken by mobile devices, and these photographs are full of contextual information. Therefore it is only natural to use them in order to obtain required information about the content that is depicted in the pictures.

One possible application, where photographs can provide some useful information for users about the objects that are displayed in them, is, for example, the recognition of exhibits in museums.

Today, different methods of explanations of the stories of exhibition pieces can be found, which depend on the wealth and popularity of the museum. Some museums can supply visitors with illustrated paper booklets, others will provide guests with a variety of audio guides where it is required to type in the number of the specific object, about which a person wants to hear information. Also, in new exhibitions videos films can be seen near exhibits. However, such technologies are available to only a limited number of visitors because of a limited number of headphones.

In the most innovative museums exist various recognition systems utilising the help of mobile phones and different codes: QR (quick response) code or HCCB (High Capacity Color Barcode) [2]. Unfortunately, they are not popular in the traditional museums, because firstly they can distract from the overall appearance and secondly they are not applicable for small objects, where there is not much space around, or for huge objects, where their location will not be obvious or visible to everybody.

Some prototype programs exist, where recognition of the exhibits is made with the help of markers (for example RFID [3]). However, this can lead to some problems, due to the fact that not all mobile phones can read these markers, thereby it will be not suitable for all visitors. Another disadvantage of this approach is that it is not possible for objects located in close proximity to each other in showcases.

The purpose of this work was to create a system, which can quickly process images, incoming from a mobile device in real-time, and work out an accelerated search of matches in a known database of images to return information about the processed image back to the client. At the same time, the system should be able to deal with varying illumination and user behaviour, so it should be able to recognise an object from various points of view, rotations and zooming. Furthermore, it should be able to support a large number of visitors with any type of mobile phones or tablet PC. The only limitation for our system is that devices should be equipped with a camera and the ability to use the Internet.

2. PREVIOUS WORKS

Nowadays, the problem of object recognition in images is not completely resolved. Due to different issues, such as lighting, extra objects in the frame and others, it is still an ongoing topic of research. Currently, there exist many approaches of recognition of images.

Most of the recognition systems in museums work only for paintings. An example of this is a system [4] developed by Boris Ruf and his colleagues. The advantage of their system is that they use only mobile phones with camera and Internet connection, but no additional infrastructure on site or extra hardware. The researchers use a client-system architecture and the client only acquires and sends sample data and eventually receives the incoming results. They decided to create a server system in order to avoid additional computation on the mobile phones, because they may have very slow CPUs and running the feature extraction on the mobile client might result in unbearably long waiting times for the user.

There are also systems, where information about exhibits is given not only based on the image processing, but also based on information about location position. Herbert Bay and his colleagues proposed an Interactive Museum Guide [5] that functions by recognising exhibits in the Swiss National Museum in Zurich. In order to reduce the search space, Bluetooth emitters were installed on site, and the recognition algorithm has an approximated SIFT[1] as a background.

Another type of system was created by Erich Bruns and Oliver Bimber [6]. They proposed a phone guide based on 3-layer neural networks and a bluetooth beacon. In their approach they used pervasive tracking, P2P communication, pervasive storage and pathway awareness. They also used a client-server system. However, in their case the mobile phone connects to the server only twice: at the begin and at the end of the visit. Thus it receives a neural network that was already trained on the server, where the training takes place.

A similar approach was followed by Herbert Bay and his colleagues [7], when they presented an interactive museum guide, where the main device is a tablet PC prototype equipped with USB webcam. With this device a visitor can take a picture of an exhibit and compare it with the existing pictures in the database. In their paper the researchers compare the development of SIFT [1] and SURF [8] prototypes. The authors conclude that the latter one is better in performance and speed. We can see that their approach is not very practical in usage firstly because of the size of equipment (a mobile phone is more convenient – as it is suitable to be held single-handedly) and secondly it is more expensive, as the museum would need to provide a tablet PC for every visitor, who wants to use it and this can result in large numbers required. To use the visitors' mobile phones as devices for interaction between the visitor and the guide system is much more comfortable and desirable in both cases.

3. RECOGNITION PART

A recognition system based on SURF algorithm [9] was created as a part of the research project for the final semester. It proposed to use keypoint detection and feature extraction. FLANN[10] was used to find matches between keypoints in different images.

As soon as the picture is taken it will be sent to the analyser. The next steps of detecting the keypoints, extracting their descriptors and finding matches were done with the help of an OpenCV library. Then the distance of each matched pair is found, and the smaller the distance the more similar the points are. Therefore, distance filtering was proposed in order to leave only significant matches.

After these matches are found the more detailed processing of sub-images near each point takes place. It is made by finding correlation between the histograms of H and S channels from the HSV colour model of the unknown image and one from the database.

The final similarity between two sub-images is calculated as an arithmetic average of correlation of channels H and S: the higher this is, the more similar are the two images.

For classification a two-round stochastic approach was proposed, which in the first round takes N representatives from every class in the database and in the second round 2N representatives of the winner classes. More detail about it was presented in [9].

Exhibits from the Pergamon Museum in Berlin were chosen for testing. Some of these are illustrated in Fig. 1.

140 images were taken as a training set: around 20 objects per class. 63 unknown images were then used as the test set. The average execution time for each run was 1 second.

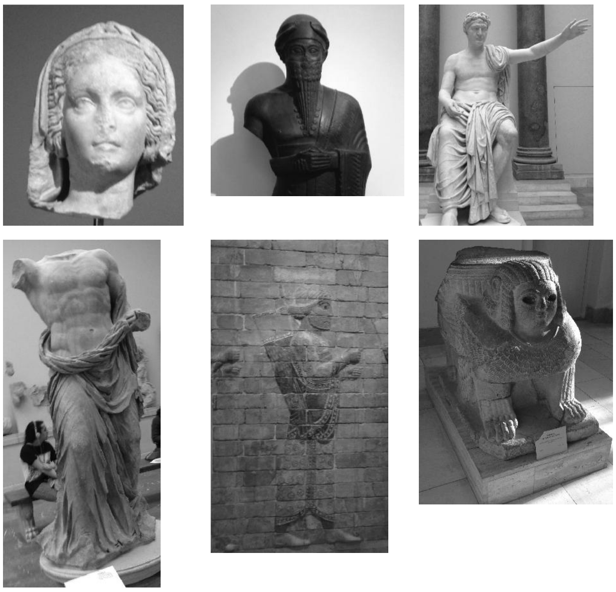


Fig. 1 – Ehibits for testing.

4. WEB-SERVER REALISATION

The server itself was implemented with the help of a Netty library[1], which can support a large amount of connections at the same time. Another of Netty's advantages is scalability, which means that the more parallel performers it has (cores, processes or threads), the more tasks it can perform at the same time. To use Netty the only requirement is to set the number of computational threads, that it can use at the same time. In this project the number of threads was equal to the number of cores on the computer.

As a part of the program it was necessary to integrate the classifier, described in the previous chapter, into the current system.

The system can be split into four main parts:

1) The database of the exhibits' photographs and information about them, which will be displayed on the website after the recognition process.

2) The classifier, which is able to recognise pictures corresponding to a class.

3) The server, which receives requests from the client in the form of a picture, then passes this picture to the classifier and obtains a class of the image in the return. With this class information the server asks the database about information for the object, displayed in the picture, and in finally sends the information to the client, who can read the results of their request (Fig. 2).

4) The client, who takes photographs of the exhibits, sends them to the server and gets information about these pictures (Fig. 3).

One of the subtasks, carried out to make transmission from the mobile phone to the server easier and quicker, is the rescaling of the images. There is no reason to send pictures with a resolution of 3000x2000 pixels (in case of the iPhone) through the router. Much less time is required to send the images, if they are first reduced in size on the phone.

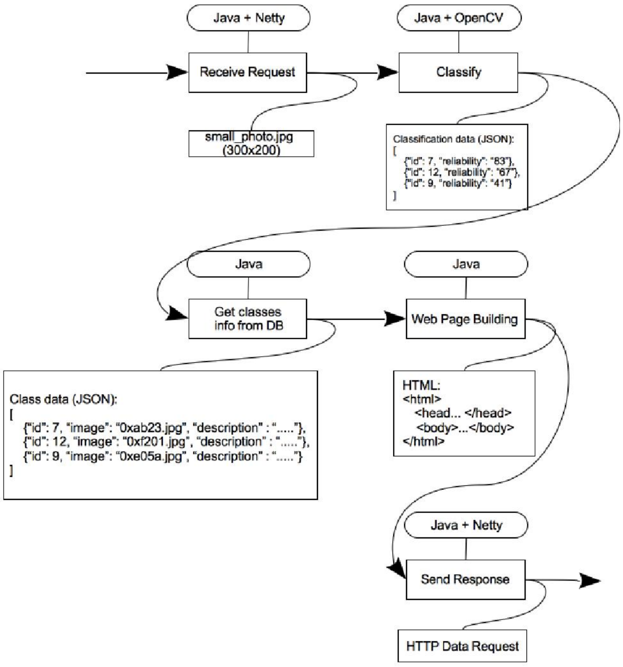

Java + Netty
Java + OpenCV
Receive Request
Classify
small_photo.jpg (300x200)
Classification data (JSON):
[
{"id": 7, "reliability": "83"},
{"id": 12, "reliability": "67"},
{"id": 9, "reliability": "41"}
]
Java
Java
Get classes info from DB
Web Page Building
Class data (JSON):
[
{"id": 7, "image": "0xab23.jpg", "description" : "....."},
{"id": 12, "image": "0xf201.jpg", "description" : "....."},
{"id": 9, "image": "0xe05a.jpg", "description" : "....."}
]
HTML:
<html>
<head... </head>
<body>...</body>
</html>
Java + Netty
Send Response
HTTP Data Request

Fig. 2 – Server data-flow diagram.

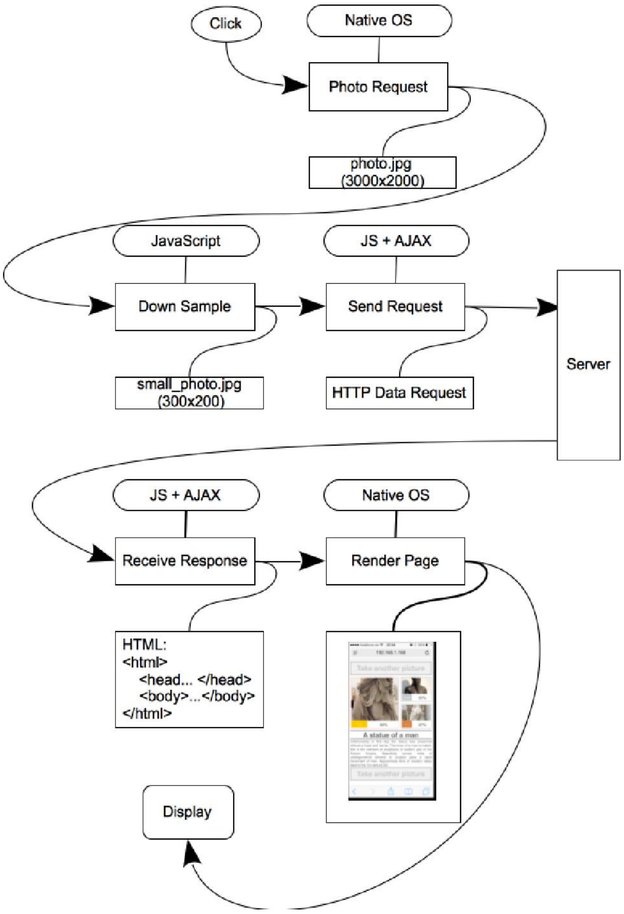


Fig. 3 – Client data-flow diagram.

[1] http://netty.io/

Therefore, in this approach JavaScript first scales the image to the appropriate size and only then sends the image through AJAX.

During tests the ideal area for a stable recognition with the smallest possible resolution was calculated to be 720x640 pixels. That means that a desired image should have an area of about 460800 pixels. It will be denoted as IdealArea. The algorithm for rescaling an image is described below:

1) IdealArea = idealWidth * idealHeight;
2) recieved Area = W * H;
3) DesiredArea = W' * h';
DesiredArea ≈ IdealArea;
4) From the (3) can be written that:

$$W^{'} = \frac{IdealArea}{H^{'}}, \tag{1}$$

5) A new variable "aspect" was introduced, a constant difference between the width and the height. That is done in order to retain the proportions of the original image (Fig. 4), so that even if the image is rescaled the aspect ratio should remain the same.

$$Aspect = \frac{W}{H} = \frac{W^{'}}{H^{'}} \Rightarrow W^{'} = Aspect * H^{'}, \tag{2}$$

6) From (4) and (5)

$$\frac{IdealArea}{H^{'}} = Aspect * H^{'}, \tag{3}$$

From this equation the size of desired height can be determined:

$$H^{'} = \sqrt{\frac{IdealArea * H}{W}}, \tag{4}$$

7) Now the size of desired width can be determined:

$$W' = aspect * H', \tag{5}$$

It was calculated, that the time spent to compress the image and to send the resulting smaller image is much less then the time, which is spent to only send an image in its original size. Although the network bandwidth of the museum is unknown, it is a known fact that generally compression is a much faster process than the transfer of large data. The compression, based on rescaling, is performed directly in the browser on java script.

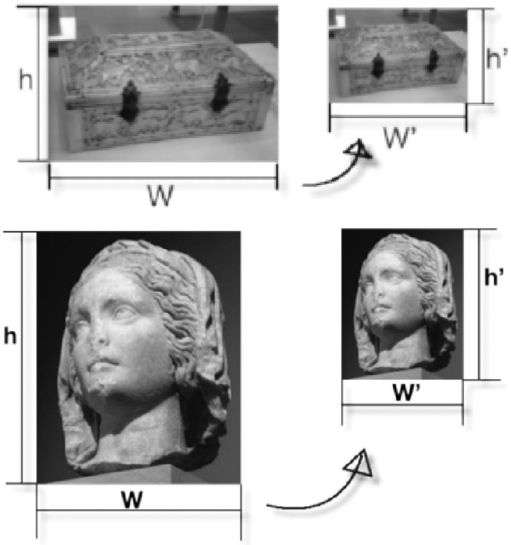
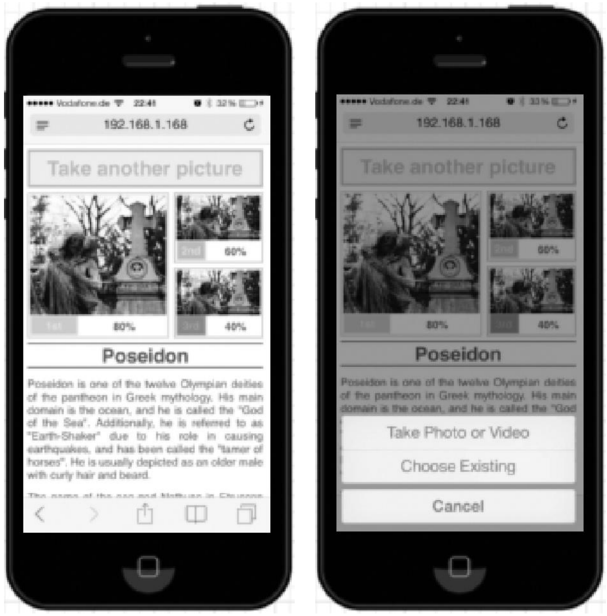


Fig. 4 – Process of rescaling the image while retaining its original aspect ratio.

5. TESTING

At the beginning, when the user first opens the website, the standard layout with information about random exhibits can be seen (Fig. 5a). At the very top of the displayed webpage, there is a button called "Take another picture". By pressing this button the user starts the process of the application. First of all, the application asks the user if they want to take a new photograph or choose from existing ones (Fig. 5b). As part of the application, a process was implemented asking the user for access to the phone's camera or image gallery for the purpose of taking or selecting photographs to obtain information about the depicted exhibit.



Poseidon

Poseidon is one of the twelve Olympian deities of the pantheon in Greek mythology. His main domain is the ocean, and he is called the "God of the Sea". Additionally, he is referred to as "Earth-Shaker" due to his role in causing earthquakes, and has been called the "tamer of horses". He is usually depicted as an older male with curly hair and beard.

a) b)

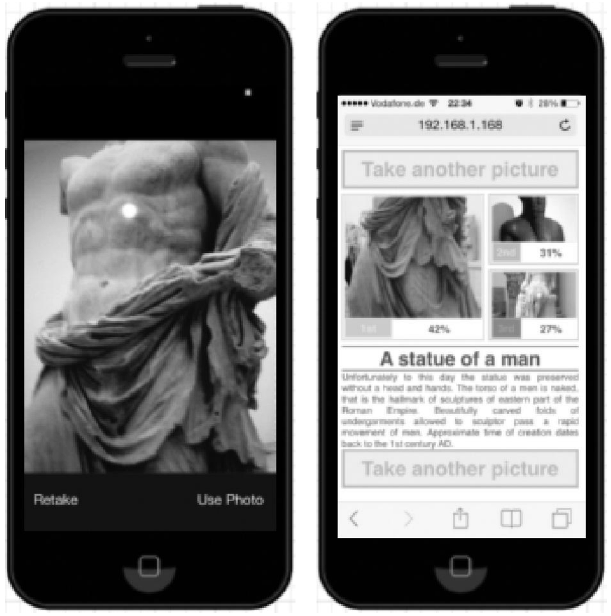
Fig. 5 – Interface of the application.

If the user selects a mode of taking a new photograph they will use a standard developed by the phone camera program, where they can take or retake a photograph. At the end of this step the user should press the button “Use photo” to send it to the server (Fig. 6a). After processing the photograph on the server side, the search result will be send back to the user (Fig. 6b).

The top layout is split into three pictures:

The largest one is the first result of the classifier. It is displayed with the yellow mark “1st” and a red frame around the picture. Underneath the picture, the name of the exhibit and information about it are shown.

The two remaining pictures, which are shown next to the main one, are respectively the second and the third place of the classifier’s recognition hits. It may occur that the system made a wrong decision and instead of, for example, a ball displayed on the photo, said that it is a dish. That is why there is not only the first result is displayed, and a user, who sees the difference exactly, can click on the second or third results and thus read the relevant information.



A statue of a man

Unfortunately to this day the statue was preserved without a head and hands. The torso of a man is naked, that is the hallmark of sculptures of eastern part of the Roman Empire. Beautifully carved folds of undergarments allowed to sculptor pass a rapid movement of man. Approximate time of creation dates back to the 1st century AD.

a) b)

Fig. 6 – Interface of the application.

As previously mentioned, the server, with the help of Netty, can perform this work very fast because of the parallel approach. The same parallel approach was used in the classifier code. To show this, a simulation test was carried out. N number of users simultaneously sent a request for classification every 5 seconds. This was carried out as a part of demonstrating the possibilities of the server performance and as an experiment. In total 13 tests were completed: each one for 4, 5, 6, …, 16 simultaneously started users. And as a result, the time of waiting for the response of answers was measured for every user (Fig. 7).

For the result graph, characteristics such as minimum and maximum time of waiting, average and median value of the time of waiting and a standard deviation were measured.

As can be seen in the graph in Fig. 7, with a load of 8 users or less all values of all parameters are the same. The time of waiting is between 2 and 4 seconds. We assume that the number 8 is connected with the fact that there are 4 cores on the MacBook, which was used during testing, and in hyper-threading each of them works almost as 2. This means that in total the MacBook can work simultaneously with 8 tasks. Therefore, the Netty server divides work between these 8 cores. With the number of users increasing to more then 8 the waiting time is growing. However, a positive result is that not all clients’ waiting time grows; we can see that the minimum time is the same – about 2 seconds. This means that some of the users, which sent their requests first, receive the response with the same quick speed. However those users, who connected later, will need to wait for a longer time. The average and maximum time of execution grows exponentially and doubles with the increase of the number of clients in one and a half the time.

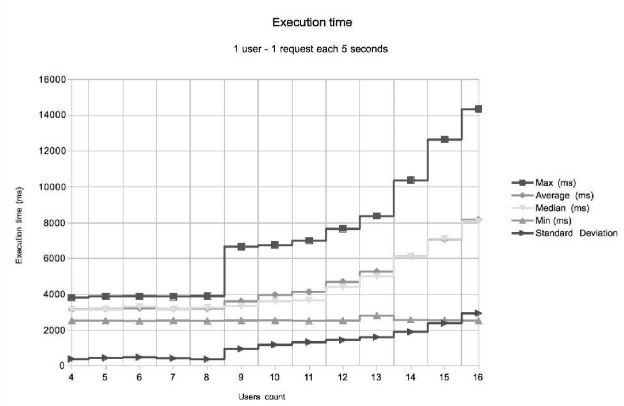
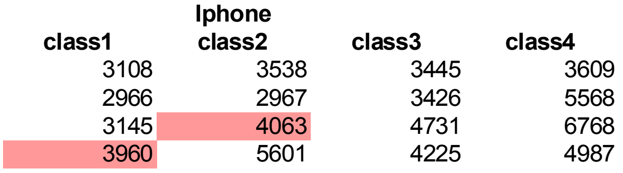


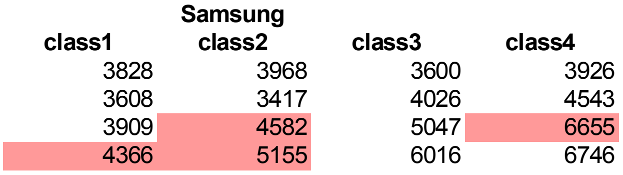
Fig. 7 – Execution time for loading test.

Tests were performed on two different devices: iPhone5 with iOS 7.0.3 and Samsung Galaxy S4 with Android 4.3. The database, which was used during this testing, is the same as was used for the FP1 (Fig. 1). Some of the results of both execution tests can be seen in the Fig. 8 and Fig. 9 respectively. Incorrectly recognised exhibits are marked in colour. The numbers in the cells are the waiting time values in milliseconds. As can be seen from the tables, for correct results the average time of waiting for response from the server is not more than 4 seconds.



	Iphone		
class1	class2	class3	class4
3108	3538	3445	3609
2966	2967	3426	5568
3145	4063	4731	6768
3960	5601	4225	4987

Fig. 8 – Recognition results for Iphone in ms.



	Samsung		
class1	class2	class3	class4
3828	3968	3600	3926
3608	3417	4026	4543
3909	4582	5047	6655
4366	5155	6016	6746

Fig. 9 – Recognition results for Samsung in ms.

As a test for robustness, an experiment was carried out with 50 test photoprahs of one single object. The photographs were taken at a distance from 1m to 2.5m from the exhibit, with an approximate angle range of 120 degrees in front of the exhibit and with a person's height of between 160cm–190cm. As a result 76% of the photographs were recognised correctly.

6. CONCLUSIONS

As a result of this research project, a system was developed, that is able to support multiple users at the same time and provide them with a written description about the objects of interest. As a GUI application, a web page was created, through which users can interact with the whole system. The user interface is a simple page, with a button located at the top. By pressing it, the user can take a photograph of an object of interest. The next step, with the help of JS and AJAX, is to send this photograph to the server. The server passes the photograph to the classifier, which returns the class number to the server. With this class information, the server is able to retrieve a description of the exhibit of interest and return it to the user. Of course the fact was taken into account, that the classifier may provide the wrong answer and to improve the result of its work, the system sends the user not only the first hit of its recognition process, but also the second and the third matches. When the result is provided to the user, they see a big image of the allegedly recognised object and a small description taken from the database. There are also images of two other objects, which were close matches to the possible result. If the user recognises, that system gave them a wrong object with the wrong description, they can select one of the other two images and see the relevant descriptions. The whole server side was implemented with the help of a Netty library. With such an implementation, the working process of the server can be made in the parallel way and thus can decrease the amount of waiting time for the users. We suggest that further decreasing the waiting time can be done in two ways: increasing the number of cores or decreasing the time of the image processing. However, if we look at previous results [9] we can see that image processing alone takes less then 1 second.

The GUI part of the application was written with the help of JS, CSS and HTML. Currently, the system is stable for Android and iOS and was tested on Safari, Google Chrome and FireFox browsers. The layout was displayed identically and without errors on all tested platforms.

As a possible plan for the future, indoor-location can be integrated into the system. Based on the indoor-location, it can be determined, for example, in which room the user is at a certain moment in time and based on this information a comparison can be made in the classifier with only the objects, that are located in this room. This will decrease the image processing time and improve the whole system. As precise indoor-location systems are very complicated to development it would be more suitable to find an existing one and integrate it into the system.

It can also be seen from the recognition results above, that the recognition accuracy is not currently very precise. Therefore, improvements can be made in the direction of the recognition quality; for example, one possibility is to select a neural network as a classifier. Then if we integrate a back-propagation as a method to train this network we can possibly take into account the results and their errors, which users can mark on their mobile phones during application use.

7. ACKNOWLEDGMENT

This paper describes the work undertaken in the context of the project SIGNAL hosted by the research group Information and Communication Systems INKA that is generously funded by the European Regional Development Fund (ERDF).

8. REFERENCES

[1] D. Lowe, Object recognition from local scale-invariant features, in *Proceeding of the seventh IEEE International Conference on Computer Vision*, Corfu, Greece, (September 1999), pp. 1150-1157.

[2] High Capacity Color Barcode Technology, 2009, Electronic resource: http://en.wikipedia.org/wiki/High_Capacity_Color_Barcode.

[3] C. A. Walton, Portable Radio Frequency Emitting Identifier, U.S. Patent 4,384,288, issue date May 17, 1983.

[4] B. Ruf, E. Kokiopoulou and M. Detyniecki, Mobile museum guide based on fast SIFT recognition, in *Proceedings of the 6th International Conference on Adaptive Multimedia Retrieval: Identifying, Summarizing and Recommending Image and Music AMR'08*, 2008, pp. 170-183.

[5] H. Bay, B. Fasel and L. V. Gool, Interactive museum guide: Fast and robust recognition of museum objects, in *Proceedings of the first International Workshop on Mobile Vision*, 2006.

[6] E. Bruns and O. Bimber, PhoneGuide. Adaptive image recognition on mobile phones, in *Proceedings of the EG Mobile Graphics Workshop,* 2009.

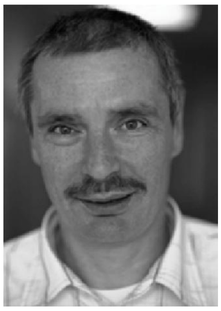
[7] H. Bay, B. Fasel, L. V. Gool, Interactive museum guide: Fast and robust recognition of museum objects, in *Proceedings of the First International Workshop on Mobile Vision*, May 2006.

[8] H. Bay, T. Tuytelaars and L. J. V. Gool, SURF: Speeded up robust features, in A. Leonardis, H. Bischof and A. Pinz, editors ECCV'2006, Vol. 3951 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 404-417.

[9] H. S. Balakhontseva, J. Sieck, Fast and robust recognition and classification of the museum exhibits, in Proceedings of the *IV International Academic Conference of Young Scientists and students "Modern Information Technology",* Odessa, Ukraine (April 22-26, 2014), pp. 55-56.

[10] M. Muja and D. G. Lowe, Fast matching of binary features, in *Proceedings of the Conference on Computer and Robot Vision* (CRV), Toronto, Canada, 2012, pp. 404-410.

***Hanna Balakhontseva**, received bachelor degree in 2012 from Odessa National Polytechnic University (ONPU) in speciality "Computer sciences" and currently studies as a master student at ONPU in Ukraine and at HTW in Berlin. She is currently working on her masters thesis investigating the development and research of recognition algorithms of text characters in images with non-uniform background. Her research interests are focused on different algorithms of object location and recognition with the help of image processing. Currently she is involved with the research project SIGNAL with the topic of projecting images onto moving T-shirts.*



***Prof. Dr. Jürgen Sieck** received his degree in mathematics in 1981 and his PhD in computer science in 1989 from the Humboldt University Berlin, Germany. Now he is the senior researcher at the research group "Informations- und Kommunikationsanwendungen" (INKA) and professor for computer sciences with a specialisation on algorithms, multimedia and mobile application for the degree programme Applied Computer Science at the University of Applied Sciences HTW Berlin. Previously, he was visiting lecturer at Monash University Melbourne (Australia), Yuanpei University Hsinchu (Taiwan) and at the University of Cape Town (South Africa) and was visiting researcher at the International Computer Science Institute in Berkley (USA), Polytechnic of Namibia and at Old Dominion University Norfolk in Virginia (USA). In February 2013 he was awarded an honorary doctorate from Odessa National Polytechnic University (Ukraine).*