JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 20 No. 1 (2012), pp. 103-118

Mobile Vision Based Augmented Reality Navigation System

Adam Wojciechowski

Institute of Information Technology
Lodz University of Technology
Wólczańska 215, 90-924 Łódź
adam.wojciechowski@p.lodz.pl

Abstract. *Augmented reality applications become more and more popular due to increasing computation speed of the mobile handsets and need of easy and intuitive navigation. Existing systems are implemented on the basis of GPS and compass where relative position and orientation of the handset is considered. Such hardware requirements decrease amount of compatible devices and limits system usage to outdoors where GPS signal is available. The paper presents a vision-based augmented reality navigation system implemented on the Android mobile phone. Image data of interest object and handset position are acquired by the users, however GPS position is not required for the system operation.*
Keywords: *augmented reality, navigation, objects recognition.*

1. Introduction

People coming to unknown city have always the same problem. How to reach the desired museum, restaurant or hotel? For those, who know the city, the answer is easy. For all the others, even the map can be insufficient help, due to current location determination problem.

First solution implementing technical knowledge is a GPS navigation system. For sure, it can guide user to desired destination in an easy way. However, some of the users will end up in a lake instead of café parking place.

Better solution utilizes not only GPS, but also compass and camera built in the device, displaying navigation data superimposed on the original image captured by the camera, which makes imformation more readable to the user. Such a solution posses several disadvantages. They become useless in building, as the GPS signal is damped. What is more, not all mobile devices contain compass and GPS receiver.

The solution presented in this paper implements computer vision based augmented reality navigation which seems nowadays to be the most interesting one. This technology can be a great improvement for human cognitive abilities, providing user with additional data about environment and its elements. In such a way, user gains knowledge that is no longer required to be memorized. Increasing speed of computers and miniaturization allows users for keeping all the required data and processing tools in the pocket. With the help of recognition algorithm and external data base containing description of the objects, user can gain huge recognition skills.

In the presented solution the main driving feature of the augmented reality navigation and localisation is image processing. There is no need of GPS data acquisition however it boosts processing time.

2. Related works

Augmented reality systems are used in more and more fields. One of the most interesting augmented reality applications is car repairing and maintenace [1, 2, 3]. Using special data goggles and wireless access to a powerful computer, mechanics can access all the required information while fixing the car, for example information on the sequence of steps as well as the tool needed to accomplish the step (fig. 1). Such an application can minimize costs of maintenance of the car and risk of failure or mistake.

Another domain focusing on application of augmented reality technology in diverse scenarios is industry [3, 4, 5, 6]. Similar solutions, displaying i.e. patient's imaging data or vital functions parameters, can be used by surgeons during complicated surgeries [3, 7, 8].

AR technology can be also very useful in city and building navigation and

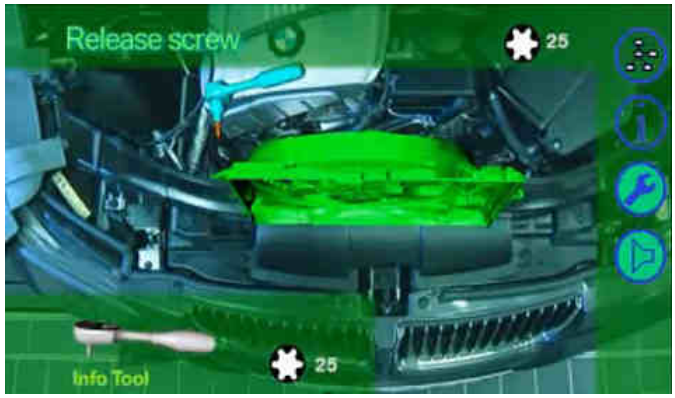

Figure 1. BMW augmented reality interface (http://www.bayforce.com/2011/02/10/business-in-two-worlds/)

selected aspects of target localization [3, 9, 10]. Examples of applications similar to presented one are Wikitude[1] (developed by Mobilizy) and Layar[2]. Both are using a build-in camera, compass and GPS module to determine users position and viewing direction. Having this information, all geo-tags that can be seen from obtained position are shown on the users screen (fig. 2, 3). The only difference are sources of data. These used by Wikitude come from Wikipedia and Qype geo-tagged entries, while Layar is based on data bases created and shared by the users, which increases the number of points of interest. What is more, Layar enables user to play augmented reality based games, follow sightseeing tours and find nearest cash machine or restaurant.

The number of existing augmented reality applications is still increasing. Some of them are based on simple image processing, allowing users to see virtual 3D objects on the mobile phone screens in the place of 2d tag seen by the camera (fig. 4). Such a solution is tested by the marketing companies as a new and catchy way of advertising the products. However camera image analysis based applications seem nowadays to be the most universal and promising.

[1] http://www.wikitude.com/
[2] http://www.layar.com/

Figure 2. Wikitude augmented reality interface (http://en.wikipedia.org/wiki/File: Wikitude.jpg)

Figure 3. Layar augmented reality interface (http://www.verious.com/component/ layar-ar)

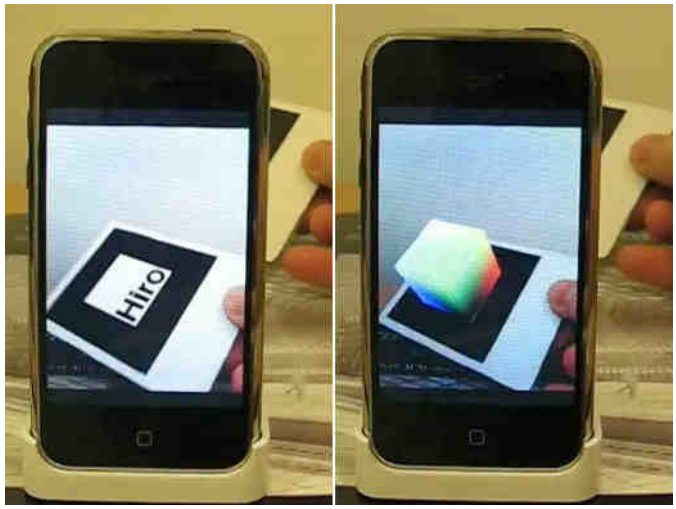

Figure 4. iPhone AR toolkit interface (http://arblog.inglobetechnologies.com/?p=280)

3. Vision based augmented reality navigation system

The designed system is based on the client-server architecture. The user application is thin client installed on the mobile device, processing only small amount of data, to decrease the battery consumption and improve speed. The mobile application is responsible for the acquisition of user's environment such as GPS data and pictures seen by the camera, as well as the direction of viewing of the camera using magnetometer.

The obtained data are sent to the server, responsible for the image content recognition. The recognition speed and precision is improved by additional data containing the location of the user. The descriptor vectors, calculated from the captured image are compared with the vectors that are stored in the data base, taking into account only objects that are visible from the user's position. The response sent by the server contains the exact location of the detected object, as well as position of the detected object in the picture, name of the object and short description that can be displayed on the user screen (fig. 5). For application interface design appropriate web usability and ergonomy rules were taken into account [11, 12].

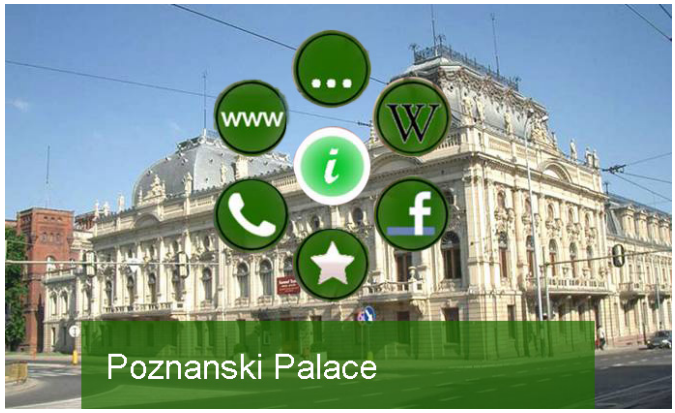

Figure 5. Smartphone application interface with possible options

Due to transmission speed and processing time limitations, the frame rate of the transmitted images does not provide real time preview and server retrieves only fraction of the frames visible for the user. Such a solution, as well as the simple communication protocol, minimizes redundancy in data transmission and maximizes the speed. To improve performance, user's preview is not synchronized with the data transmission. During the operation of the program, it is possible to change position of the mobile device, without resending image to the server. In such a situation pointer on the screen will be in the wrong position. To avoid such effects and improve the performance, the viewing direction of the camera is tracked, using data from the accelerometer and the magnetometer.

3.1. Keypoint features extraction

To make this paper more self-contained, the algorithm used for the features extraction will be briefly explained, although its detailed description can be found in [13, 14]. It is called Speeded-up Robust Features (SURF) and its performance can be divided into two parts: keypoints detection and descriptors calculation.

To detect the characteristic elements in the image, the SURF algorithm uses the approximations of Hessian matrices, which are the convolutions of input image with the second derivates of Gaussian functions. Approximation stands for replacement of Gaussians with box filters, the examples of which are presented in the figure 6.

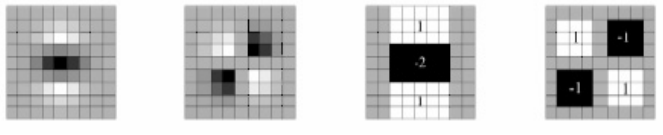

Figure 6. Left to right: the (discretised and cropped) Gaussian second order partial derivatives in y-direction and xy-direction, and approximations thereof using box filters. The grey regions are equal to zero [13].

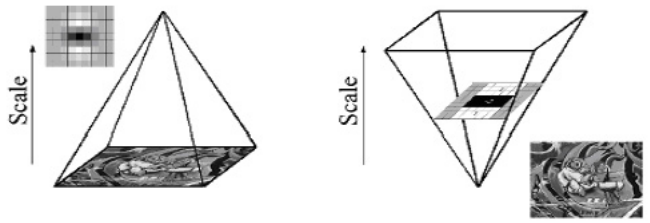

Figure 7. Scale-space representation of the images. Instead of image size reduction (left), the upscaling of the filter is used (right) [13].

Denoting the responses of them by D_{xx}, D_{xy} and D_{yy}, the definition of an approximated Hessian matrix looks as following equation 1.

$$H_{appr} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{1}$$

Its determinant is the detector of the SURF algorithm (eq. 2).

$$det(H_{appr}) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{2}$$

where w is a coefficient required to fulfill the energy conservation law and is approximately equal to 0.9. Because D_{xx}, D_{xy} and D_{yy} are matrices, the determinant of H_{appr} is also a matrix composed of different values. The greatest of them correspond to the characteristic points of the image.

To make the algorithm resistant to scale changes, the scale space representation was introduced. It can be portrayed as the pyramid consisting of layers (fig. 7). Each layer corresponds to its own scale and constitutes an original image convolved with the box filter increased to a proper size. The smallest box filter used in

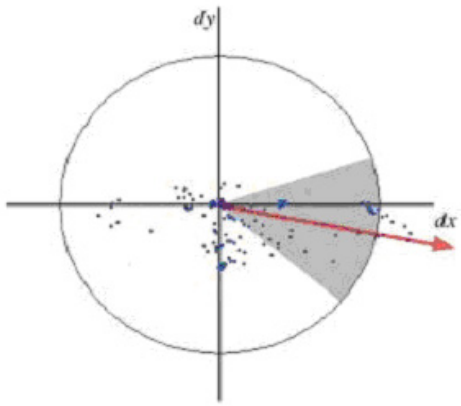

Figure 8. The determination of dominant orientation - direction vector [13].

the algorithm has size 9x9 pixels and corresponds to the Gaussian function with the mean $\sigma = 1.2$. The σ coefficient is also equivalent to the scale at which keypoint detection is being performed. To preserve the relative sizes of the lobes of the box filters, the other filters must have dimensions 15x15, 21x21 and 27x27 (which correspond to scales 2.0, 2.8 and 3.6 respectively). In order to detect the keypoints of even greater scale, additional octaves (sets of filter sizes corresponding to 2 times greater scale) can also be used.

Once the interest points are detected, they need to be described by certain parameters. This task is performed in two steps. Firstly, the reproducible orientation of the feature should be found, in order to make the algorithm independent of the rotation.

To decide what is the orientation of the interest point, the Haar wavelet functions are used. Each element detected by the algorithm is marked by a circular area of radius 6s, where s is the scale at which the point was detected. Then, the responses of wavelet filters within these areas are computed and the results are introduced to the coordinate system (fig. 8). The horizontal axis of this system represents the value of horizontal Haar wavelet response of a point. Similarly, the vertical axis is related to the vertical Haar filter. When all the responses within the circular area are inserted to the graph, they create a set of non-uniformly distributed points. The direction of the feature is estimated by calculating the number of such

points within a circular window of the angle $\pi/3$. When the optimum position of the window is found (the number of points within a window is maximum), all the responses included in this window are summed up in both x and y directions, and the resultant numbers constitute the size of the direction vector.

The final step in the process of descriptor extraction begins with surrounding each interest point with rectangular area oriented along the direction vectors. It is subdivided into 16 smaller regions (4x4), in each of which the Haar wavelet responses in horizontal (denoted as dx) and vertical (denoted as dy) directions are again calculated. Summation of these values in the 16 sub-regions gives 32 parameters. Another half of the descriptor constitute the sums of absolute values of the responses. The example of rectangular areas around the interest points is presented in the figure 9.

Finally, the set of 64 coefficients describing the feature are the following values (eq. 3),

$$v_1 = \sum dx, v_2 = \sum dy, v_3 = \sum |dx|, v_4 = \sum |dy| \tag{3}$$

calculated 16 times for each of the sub-regions of the rectangular area surrounding the interest point.

3.2. Data base training

The descriptors explained in the previous paragraph will constitute the crucial factor utilized for the purpose of object recognition. Therefore, the data stored in the data base will consist of set of records, each of which will contain 64 descriptor coefficients, identifier of the object and identifier of the side of the world, at which feature corresponding to this descriptor is visible in reality. The functionality of this last piece of data will be explained in the next part of this paper. A special dedicated piece of software[3], is capable of calculating the descriptors of the images placed in a relevant folder structure, and filling the data base with proper values.

The usage of images displaying whole or most of the building is not recommended for the purpose of data base creation. The best solution is to put to the data base only small pictures containing the crucial and most important details of the building. Such attitude is reasonable especially in cases, when a building's façade features a repetitive pattern, like the figure 10 shows. In such cases only a

[3]application was created as a part of Information Technology TEWI platform project supported by European Union funds, Innovative Economy Programme,

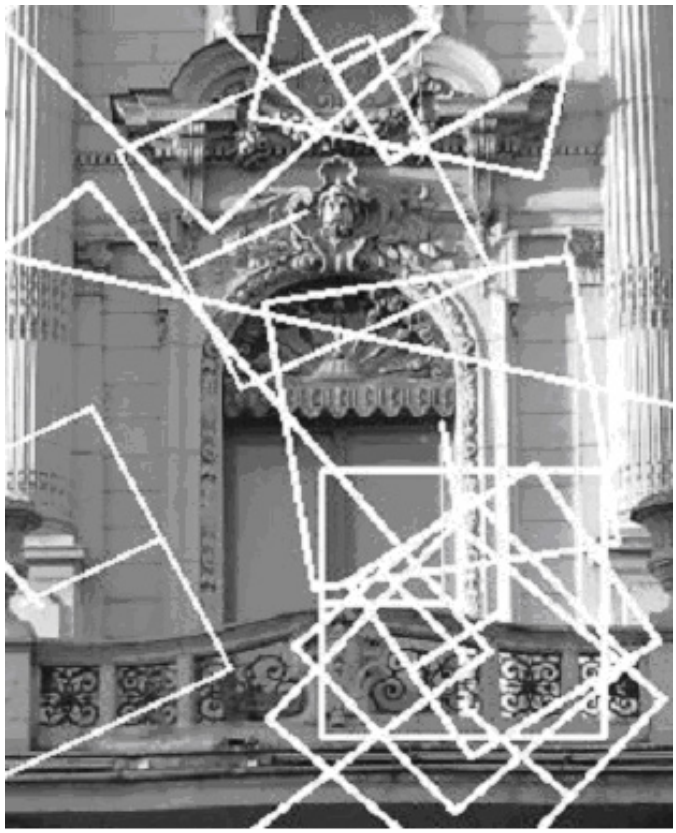

Figure 9. Example of the interest points and rectangular areas aligned with their direction vectors [15].

Figure 10. Comparison of the template training image (right) with the tested image of the White Factory in Lodz. The white points indicate elements recognized in both figures.

single element, which repeats in the building structure, should be stored in the data base. Because the algorithm does not have to make comparisons between object features separately, its calculations are much less time-consuming. Figure 10 also shows that the assumption, that only one instance of repeated component is sufficient, is reasonable, since the algorithm classifies other instances as being of the same type (the white points indicate the key features classified as the same as on the pattern image).

Because a descriptor can be treated as a set of coordinates of a point in a 64-dimensional space, the comparison of test images with the pattern images is performed by the calculation of Euclidean distance between each pair of such points. If the distance is smaller than certain threshold, the descriptors are interpreted as belonging to the same feature.

3.3. Taking advantage of geographical coordinates

Because the client application is going to be installed on mobile devices, which contemporarily very often include GPS navigation, it is possible to improve the performance of object recognition by means of analysis of the user's position. Thanks to GPS data sent by the client to image processing unit of the application, the algorithm can filter out the buildings which are not enclosed within certain radius. Moreover, it is possible to estimate with large probability, which sides of the

building are currently visible to the user. That is why the training images are divided into folders corresponding to different sides of the world. To take advantage of the GPS coordinates, the data base must obviously contain information about buildings latitude and longitude.

4. Results

In order to determine performance of the system, several tests were performed. The most significant factors that influence system effectiveness are recognition rate, GPS accuracy, sensors sensitivity and connection speed. In presented solution, GPS and additional sensors are used, however correct operation of the overall system requires the phone featuring only camera and internet connectivity. To evaluate recognition effectiveness of the system, a special piece of software was implemented, which allows for the selection of input image, simulation of geographical coordinates and outputs information whether any of the data base buildings was detected or not. The set of input images is presented in the figure 11. Each of them was tested with simulated location close to the Firestation at Księży Młyn in Łódź. Only the first 5 pictures present the mentioned object, the other are just photographs taken in the nearby. The output of the testing software indicated that the fire-station was recognized in the images named firestation1.jpg, firestation2.jpg and firestation4.jpg. This means that images 3 and 5 were not classified properly.

In order to test GPS accuracy, several measurements were performed for fixed position of the tester, that was outside the building with clear view of the sky. Each time the GPS receiver was reset to ensure that previous measurements have no influence on current one. After collection of the data, statistical analysis was performed to determine the mean value of the measurement and standard deviation. The mean value of the position is $51.74362°N$ and $19.48176°E$, while the standard deviation is 0.00003 and 0.00002 respectively. This gives the accuracy of measurements equal to 3.6 meter. To determine the transmission speed in ideal conditions, frame rate on the server simulator was tested. For this purpose the server simulation software, that does not provide image processing, was used. The transfer time was equal to 646ms with standard deviation 158ms. The last, but the most important test, was integration test of the mobile client with the real server recognizing the images. The main problem was adjustment of two parameters used by the recognition algorithm of the server: hessian threshold, which is the minimum value of the hessian detector (let us call it precision), for which an element is inter-

Figure 11. Set of testing images for the Firestation at Księży Młyn in Łódź.

preted as the keypoint, and match threshold (sensitivity) – the minimum number of tested descriptors matched with data base descriptors for which the building is detected. It turned out that for the test in the real city space server works better for about 50 % greater precision than for static images taken in similar conditions as the training set. What is more, according to our predictions, the server algorithm is capable of detecting objects in different lighting conditions, which - according to our preliminary comparisons – did not influence the time and effectiveness of the recognition. The selected test images are presented in figure 12.

5. Conclusions and future work

The designed system is very promising, taking into consideration obtained results. It can fill the market niche, while currently existing solutions are not available on all device types. The main advantage of the system is scalable architecture, that enables many implementations. The amount of work and costs of the hardware needed to drive the system strongly depends on the amount of objects that should be recognized and number of the customers.

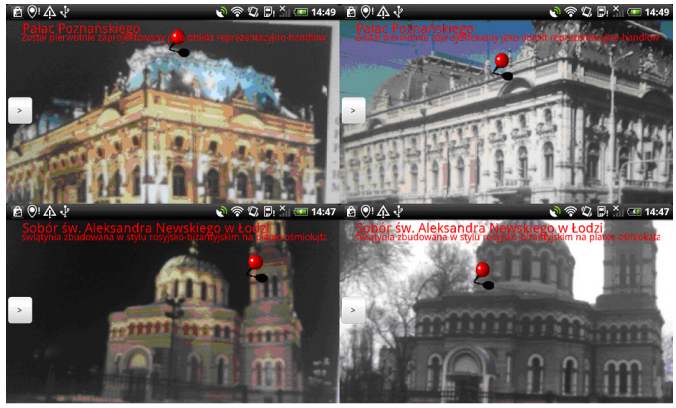

Figure 12. Results of object recognition

Presented client application can work indoors as a mobile guide in the museums, as well as outdoors, in the whole cities. It is not strongly dependent on GPS coordinates of the user, however collection of such data improves the recognition speed and efficiency. What is more, the position and rotation of the camera is not essential, however it increases performance experienced by the users.

The most important component influencing the performance quality of the application is the recognition module. The SURF algorithm used in this part detects the key features in the images, but its sensitivity can be adjusted by modifying the hessian threshold parameter, which is the minimum value in hessian matrix, for which the point is interpreted as a characteristic element. Increasing this coefficient causes that SURF calculates descriptors only for the strongest corners and blob structures. Such a situation can cause, that the algorithm misses some important building features. On the other hand, if the value of hessian threshold is low, even very unimportant elements are noticed, which leads to increase of the number of detected key points, incorrect comparison results and, obviously, long calculation time. The first task of the application development process will be to empirically find a compromise between these two situations. The testing application, which is going to be implemented, will allow for finding the best hessian threshold in both processes: data base training and image recognition.

What is more, the graphical user interface will be redesigned to satisfy users needs, as well as provide data in clear and readable way. User will gain possibility to access data not only from proposed database, but also from other sources.

References

[1] Platonov, J., Heibel, H., Meier, P., and Grollmann, B., *A mobile markerless AR system for maintenance and repair*, In: ISMAR '06 Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2006, pp. 105–108.

[2] Gausemeier, J., Fruend, J., Matysczok, C., Bruederlin, B., and Beier, D., *Development of a real time image based object recognition method for mobile AR-devices*, In: Proceedings of the 2nd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa (Afrigraph 2003), 2003, pp. 133–139.

[3] Azuma, R., *Recent Advances in Augmented Reality.* IEEE Computer Graphics and Applications, Vol. 21, No. 6, 2001, pp. 34–47.

[4] Tumler, J., Mecke, R., Schenk, M., Huckauf, A., Doil, F., Paul, G., Pfister, E. A., and Bockelmann, *Mobile Augmented Reality in Industrial Applications: Approaches for Solution of User-Related Issues*, In: IEEE International Symposium on Mixed and Augmented Reality 2008, 2008, pp. 87–90.

[5] Mizell, D., *Boeing's wire bundle assembley project*, Mahwah: Lawrence Earlbaum Associates, 2001, pp. 447–467.

[6] Pentenrieder, K., Bade, C., Doil, F., and Meier, P., *Augmented reality-based factory planning - an application tailored to industrial needs*, In: Proceedings of the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007), 2007.

[7] De Paolis, L. T., Pulimeno, M., and Aloisio, G., *An Augmented Reality System for Surgical Pre-Operative Planning*, In: Proceedings of the BEBI'08, 2008, pp. 70–73.

[8] De Paolis, L. T. and Aloisio, G., *Visualisation and Interaction System for Surgical Planning*, In: Proceedings of the EICS4Med'11, 2011, pp. 65–68.

[9] Allen, M., Regenbrecht, H., and Abbott, M., *Smart-Phone Augmented Reality for Public Participation in Urban Planning*, In: Proceedings of the ozCHI 2011, 2011, pp. 11–20.

[10] Andrzejczak, J. and Szrajber, R., *Augmented reality as a space for presenting and passing the information about works of art and architectural monuments*, Studies in Computational Intelligence, Vol. 401, 2012, pp. 49–60.

[11] Nielsen, J., *Designing Web Usability: The Practice of Simplicity*, New Riders Publishing, Thousand Oaks, 1999.

[12] Stobińska, M. and Pietruszka, M., *The Effect of Contrasting Selected Graphical Elements of a Web Page on Information Retrieval Time*, Journal of Applied Computer Science, Vol. 17, No. 2, 2009, pp. 113–121.

[13] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., *SURF: Speeded Up Robust Features*, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, 2008, pp. 346–359.

[14] Bay, H. and Tuytelaars, T. andVan Gool, L., *SURF: Speeded Up Robust Features*, In: ECCV '06 Lecture Notes in Computer Science, Vol. 3951, 2006, pp. 404–417.

[15] Krajewski, T., *Virtual object recognition for a virtual Lodz tour-guide*, MSc thesis at the Technical University of Lodz, 2011.