Developing a Parking Monitoring System Based on the Analysis of Images from an Outdoor Surveillance Camera

I.V. Sukhinskiy, E.A.Nepovinnykh and G.I. Radchenko
*South Ural State University, Chelyabinsk, Russia
si1en7ium@gmail.com, kwadraterry@gmail.com, gleb.radchenko@susu.ru

Abstract – In this paper, we describe a solution for monitoring of parking availability based on computer vision. It allows us to detect and track cars in a parking lot, while collected historical data helps us to predict availability status of parking during the day based on data mining techniques. Parking Monitoring System tracks availability based on an analysis of images from an outdoor surveillance camera and analyzes in real time the state of the parking complex. The system is developed to determine the number and location of available parking places and to inform the drivers. We provide an algorithm for image capture and analysis to recognize a car in a parking lot and to define parking spaces as either reserved or free.

Keywords - computer vision, object recognition, parking, OpenCV

I. INTRODUCTION

The growth of the population of major cities exacerbates the problem of traffic and, as a consequence, the provision of parking spaces. The transport system of the metropolis is enormous, requires constant monitoring, timely expansion and modernization. Organization of monitoring of such a complex system is difficult without the use of intelligent information systems.

The technical system, which carries out the collection, pre-processing of data at a low level, highlights the characteristics and laws, prepares a consolidated report and renders it for user is called intelligent monitoring system [1].

Intelligent monitoring of parking availability involves parking places detection and tracking of parked vehicles. Different types of sensors exist to gather information on the movement [2, 3]: video detectors, radars, infrared sensors, ultrasonic sensors, passive acoustic sensors, inductive loop sensors. Each type of sensor has its advantages and disadvantages. Nowadays CCTV systems are very common and there is a very wide range of tasks in this area. One of the fundamental problems is the problem of motion detection and tracking. Video detector can replace multiple inductive sensors which detect vehicles in several places and is also quite economical to maintain. However, computer vision algorithms that are used in these sensors have significant limitations in accuracy of detection of the vehicle. They are sensitive to the image distortion caused by weather events, shadows, overlapping objects, the day and night cycle and a surface stains on the camera lens [4].

This topic is relevant due to the need to monitor the status of a parking lot in real time and providing timely information to users. Timely information about the available parking spaces will enable drivers to save time searching for a parking space, as well as significantly increase the usability of parking. Such system would save time of drivers and indirectly reduce the carbon dioxide emissions produced by vehicles, due to the reduction of local traffic and by reducing the time required to find a parking space [1, 4].

We have proposed the development of a system for tracking the availability of parking spaces based on the analysis of digital images obtained with a camera in real time.

The work consists of the following parts. Part I provides an overview of principles of work of currently available systems for tracking moving objects and provides a brief overview of algorithms for movement detection. Part II contains a description and analysis of the system requirements. Part III shows the architecture of the system of monitoring of availability of parking spaces based on the analysis of images from the outdoor surveillance camera. Part IV is devoted to the implementation of the system using technologies such as OpenCV, C ++ and Angular.js. Part V presents the results of system testing. The conclusion provides overview of complete work.

II. FORMULATION OF THE PROBLEM

The task of tracking an object is defined as, given a position of its areas of interest at a given time, to find a position of its area of interest at subsequent times. The rectangular area of interest is a set of pixels of a digital image, outlining the desired object [2, 5].

The aim of this work is to develop a system for tracking the availability of parking based on the analysis of digital images obtained with digital camera in real time.

To achieve this goal it is necessary to solve the following problems:

- to study the features of the existing systems for tracking of moving objects;
- to study the features of the OpenCV - a library of image analysis [6];

- development of a system architecture for monitoring of parking availability based on the analysis of images from the camera of outdoor surveillance;
- implementation of the algorithm of transformation of an image obtained from a digital device;
- development, implementation and optimization of the algorithm for parking image analysis and parking space availability detection;
- implementation of a web application that shows the state of the parking according to a surveillance camera; testing of the application.

III. Existing solutions analysis

The task of optimizing traffic flow helps to solve the service "Yandex.Traffic", developed by Yandex. "Yandex. Traffic" shows user an image of congestion. In order to achieve this a service collects data from various sources on the busiest streets, analyzes them and displays it on "Yandex.Maps" [7]. In most large cities where traffic jams are serious problem, the service expects to score jams - the average level of congestion. Technology service is set up so that the traffic information is also collected from users themselves. To participate in data collection, drivers are required: Internet-connected phone or tablet with a GPS-receiver and installed application "Yandex. Navigator" or “Yandex. Maps” with enabled traffic reporting. Every few seconds, the device transmits its geographical position, direction and speed of the computer system to Yandex.Traffic. All data is anonymous, ie does not contain any information about the user or his car.

In addition to its coordinates drivers can report more information about service failures, repairs or other road troubles.

To find a parking space, the Yandex has released the "Yandex. Parking" service which is designed to quickly find, select and pay for a parking space [8].

The program is integrated into the application infrastructure of "Yandex". Thus, on the map you can see all the nearest parking lots, and use the navigator to construct the optimal route to the selected parking space. With the help of Yandex, a user can quickly pay the parking fee from card, mobile phone account or electronic wallet. Also, the program issues a warning when the paid parking time ends. The program remembers the place where the user left the car in case it is lost. Every 10-15 minutes, the service updates data on the utilization of municipal and commercial parking lots. There is information about cost, hours of operation and other important factors [9].

TIBA company developed the line Parking Guidance System. The TSGS-30 is an electronic system, designed to assist drivers in finding quickly and easily a vacant parking space inside a parking lot. The TSGS-30 utilizes sophisticated technology, based on using Ultra-Sonic wave detector with Red-Green lights on top of each parking space. The TSGS-30 uses also electronic signage of directional arrows, guiding the driver to the nearest low-occupancy zone. Ultra-Sonic sensors placed above each parking space indicate if the space is occupied. The information is sent to the management server, which updates the electronic traffic signage accordingly. The driver will be directed to a low occupancy zone. At the appropriate zone the driver will see a green light that will indicate the available parking space [10]. The system is too complicated and can not be used in small parkings or parkings near residential buildings.

IV. Architecture

The system of tracking of parking availability consists of two independent components: the system of capture and image analysis, and web-based interface.

The system of capture and image analysis receives a stream of frames from web cameras. Using image analysis algorithm the selection of cars in the parking lot and the definition employed / free parking spaces is made. Periodically, data on parking availability is recorded into the database.

The web interface allows users to monitor the status of parking. This web-based application is built using the architectural pattern MVC: customer request is processed by the controller, then the model extracts the required data from the database and generates a representation of the response (see. Figure 1).

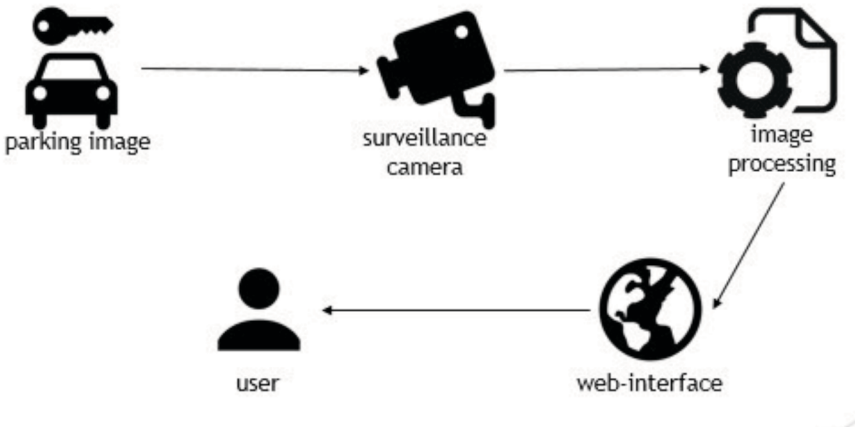

Figure 1: System components interaction diagram

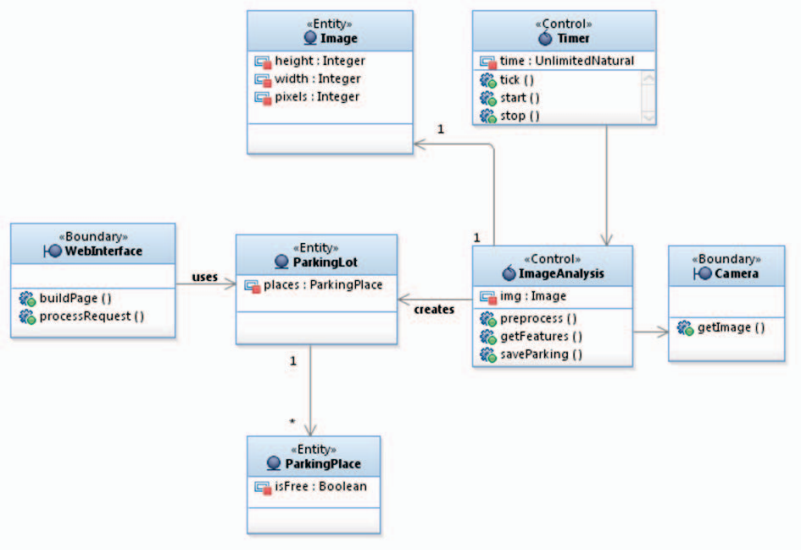

Figure 2: System components interaction diagram

During the analysis of the system requirements it was decided to allocate the following entity classes (see. Fig. 2):

- Image - the entity responsible for storing the image of parking in the computer's memory;
- ParkingPlace - essentially a parking space;
- ParkingLot - essentially a set of parking spaces.

The following are behaviour classes:

- ImageAnalisys - a class that is responsible for image processing and analysis;
- Timer - class which generates an event according to regular time intervals.

To ensure the interaction with the external environment of the system we identified the following interfaces classes:

- WebInterface - a class responsible for the operation of the web interface;
- Camera - the entity that provides access to an image from a surveillance camera

V. DEVELOPMENT

A. *Components*

The implemented system consists of two weakly coupled to each other components: a web server and an image analyzer.

The web server provides web GUI and API. Both are implemented as a web application, which provides direct access to the database. Image Analyzer with some frequency receives an image from the outdoor surveillance cameras, produces the transformation of perspective and compares it with the previous one, then locates the changed image area and updates the status of parking. New state of parking is then sent to the server.

The image analyzer is a separate application that interacts with the web server exclusively through the API.

B. *Image analyser*

Image Analyzer is an application that uses an OpenCV library [11]. After a predetermined interval

Figure 3. Image analysis result

image analyzer receives images from an outdoor surveillance camera. The transformation of the image changes the perspective of picture of parking so that cars are positioned perpendicular to the surveillance camera. This is done because calculations are simplified on a rectangular coordinate system, and because it is easier for an administrator to designate parking places on an image that went through a perspective transformation. The parameters of the transformation are initially set by an administrator. It is done by selecting 4 corners of a rectangular area on the parking lot (not on the image). The transformation is then calculated using those coordinates and applied to each image obtained from the camera [12, 13].

Next, using the resulting image comparison algorithm with the previous frame algorithm highlights potential changes in the state of parking. Each parking space that the algorithm for calculating the difference found to be changed then checked with the help of pre-trained neural network, to exclude the maximum number of false-positive results [14]. This information is converted into JSON format and sent via HTTP POST-request to the Web application. Presented here is the implementation of obtaining the difference between the two adjacent images:

```
std::vector<char> difference(const
std::vector<cv::Rect>& places, const
cv::Mat& transform, const cv::Mat& fst, const
cv::Mat& snd)
{
    std::vector<char> result;
        cv::Mat transformedFirst,
transformedSecond, greyFirst, greySecond, diff;
        cv::warpPerspective(fst,
transformedFirst, transform, fst.size());
        cv::warpPerspective(snd,
transformedSecond, transform, snd.size());
        cv::cvtColor(transformedFirst,
greyFirst, cv::COLOR_BGR2GRAY);
        cv::cvtColor(transformedSecond,
greySecond, cv::COLOR_BGR2GRAY);
        cv::absdiff(greySecond, greyFirst,
diff);
        for each (auto place in places)
        {
                auto cropped = diff(place);
                auto mean = cv::mean(cropped);
                bool changed = mean.val[0] >=
threshold;
                if (changed)
                {
                        changed =
*(classification_network.run(mat_to_array(croppe
d)));
                }
                result.push_back(changed);
        }
        return result;
}
```

The Image analysis result is shown in Fig. 3.

C. User Interface

The user interface is a web-application implemented in the Python programming language using Django web framework and Twitter Bootstrap Instruments [15, 16].

To identify ways by which you can access the Web application, there is a file ROUTES. It identifies the following ways:

- Home / parking displays all registered in the system;
- Parking / id / displays the status of parking numbered id;
- Parking / id / change / edit page displays the parking administration screen. The administrator can manually change the state of a parking space.

For each of these paths a controller and its presentation are defined.

When adding a new parking lot for the correct operation of the system administrator must perform the following steps:

1) configure video camera so that the angle was close to 90°;
2) for the correct execution of the transformation perspective, the administrator should note the angle of parking relative to the image using the mouse;
3) after performing the transformation of perspective administrator selects rectangular areas, designating parking spaces. After selecting each field must be set to be either free or busy.

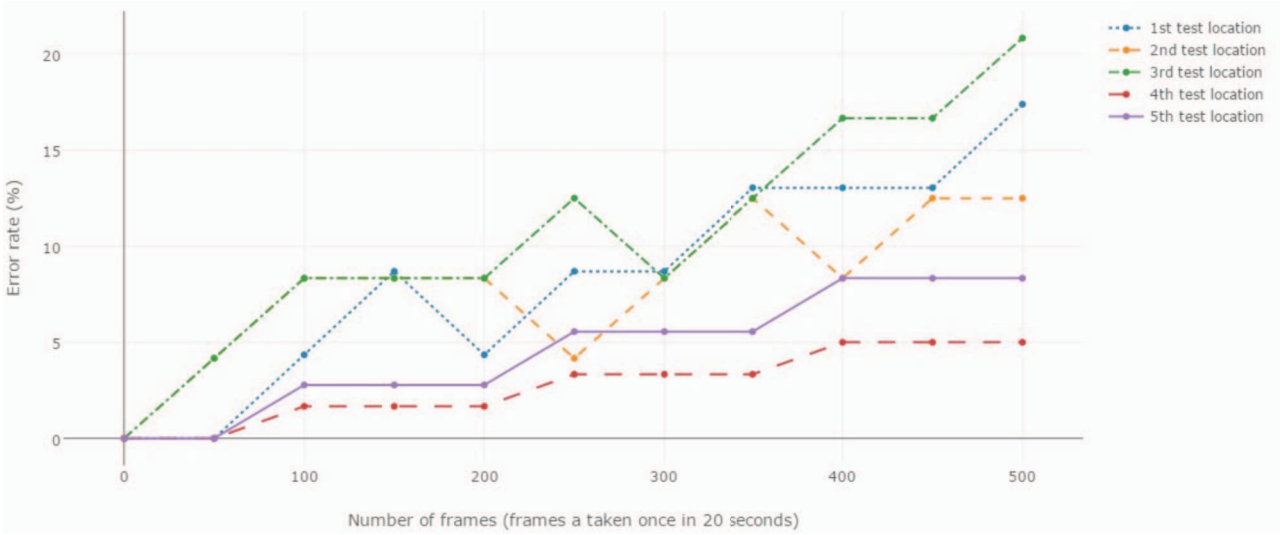

Figure 4: Image analysis testing

If the administrator has noticed an error in the calculation of free parking spaces, he can manually change the state of a parking space.

VI. TESTING

Testing of image analyzer was done using images of various difficulty levels, which reflect the status of the parking in the city of Chelyabinsk during daytime. Number of parking spaces are in the range of 12 to 70. For the testing we used the service "Online Streets" from Intersvyaz [17].

Service "Online Streets" provides access to view video from surveillance cameras on the busiest sections of the city of Chelyabinsk in real time. On some cameras a parking is clearly visible. Monitoring of these parking lots was conducted for 7 days in real time.

For the test we used five most appropriate for surveillance large car parkings in Chelyabinsk.

The results of the algorithm can be described to be good. The graph comparing the performance of the algorithm in various parking lots is shown in Fig. 4. This graph shows a correlation between the number of mistakes of the algorithm and the number of frames that the algorithm has analyzed.

VII. CONCLUSION

This article presents the design and implementation of a Web application that analyzes the state of parking lot according to a surveillance camera.

The developed system has been tested on real digital images taken with cameras of Chelyabinsk and “Online Streets" from Intersvyaz [10]. Testing showed that results of algorithm were generally good.

References

[1] Mimbela L.E.Y. A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems [Online]. Available from: http://www.fhwa.dot.gov/policyinformation/pubs/vdstits2007/ [Accessed 08 Feb 2016]

[2] T. Gao, P. Wang, C. Wang, and Z. Yao, “Feature Particles Tracking for Moving Objects,” *J. Multimed.*, vol. 7, no. 6, pp. 408–414, Dec. 2012.

[3] C. Goerick, D. Noll, and M. Werner, “Artificial neural networks in real-time car detection and tracking applications,” *Pattern Recognit. Lett.*, vol. 17, no. 4 SPEC. ISS., pp. 335–343, 1996.

[4] Sivaraman S., Trivedi M. M. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis //Intelligent Transportation Systems, IEEE Transactions on. – 2013. – Т. 14. – №. 4. – С. 1773-1795.

[5] Fischler M. A., Firschein O. (ed.). *Readings in Computer Vision: Issues, Problem, Principles, and Paradigms.* Morgan Kaufmann, 2014.

[6] Bradski G., Kaehler A. Learning OpenCV: Computer vision with the OpenCV library. – " O'Reilly Media, Inc.", 2008.

[7] Yandex maps, [Online]. Available from: https://maps.yandex.ru/ [Accessed 08 Feb 2016]

[8] Yandex parking, [Online]. Available from: https://yandex.ru/support/parking/ [Accessed 08 Feb 2016]

[9] Yaprobki service description [Online]. Available from: https://company.yandex.ru/technologies/yaprobki [Accessed 08 Feb 2016]

[10] TIBA Parking Systems [Online]. Available from: http://www.tibaparking.com/tsgs-30-single-spot-guidance-system-2/ [Accessed 08 Feb 2016]

[11] OpenCV library, [Online]. Available from: http://opencv.org/.

[12] R. O. Duda, P. E. Hart “Use of the Hough transformation to detect lines and curves in pictures”, *Communications of the ACM.* 15(1), pp. 11-15. 1972.

[13] X. S. Zhou and T. S. Huang, “Relevance feedback in image retrieval: A comprehensive review,” *Multimed. Syst.,* vol. 8, no. 6, pp. 536–544, 2003.

[14] W. K. Pratt, *Introduction to digital image processing*. CRC Press, 2013.

[15] Django web-framework, [Online]. Available from: https://www.djangoproject.com/ [Accessed 08 Feb 2016].

[16] Twitter Bootstrap Documentation, [Online]. Available from: http://getbootstrap.com/ [Accessed 08 Feb 2016]

[17] Street online sevice from Intersvyaz, [Online]. Available from: http://www.is74.ru/home/streets/ [Accessed 08 Feb 2016]