

A Raspberry-Pi based Surveillance Camera with Dynamic Motion Tracking

Oussama Tahan

School of Engineering

International University of Beirut BIU, Beirut, Lebanon

Abstract—Surveillance is the monitoring of behavior, activities, or other information for the purpose of influencing, managing or protecting people and belongings. It is very useful to governments and law enforcement to maintain social control, recognize threats, and prevent criminal activity. With the emergence of video cameras and recorders, video surveillance has become easier and the task starts to be more accurate with proof. And since theft can be done by housekeepers, employees themselves in houses, companies and offices and because no available surveillance system can detect motion and follow it, a real time system for human detection and motion tracking will be proposed in this paper. This system will be designed as smart automated video surveillance for monitoring people, detecting and tracking their movements. Our system also reduces the number of cameras used and still accomplish the wanted task. This reduces the cost of employing a surveillance system. The monitoring system was implemented using raspberry pi 3model B+, web camera, two servo motors assembled by a pan-tilt and it was controlled using OpenCV and python.

Keywords—Camera; Surveillance, Raspi, Automation

I. INTRODUCTION

Closely monitoring people's actions in specific areas is defined by surveillance. Long time ago, surveillance was performed by employing specific people who always keep an eye on the zone which they are responsible to achieve its security. They control and observe entering and leaving people in order to identify intruders. On another hand, they monitor suspicious actions done by individuals in order to prevent or at least reduce loss, theft and damage. With the emergence of video cameras and recorders, video surveillance has become the most useful approach for monitoring people and events occurring in specific places. It began with the simple closed-circuit television- CCTV- monitoring and then it was suggested to the police to use the surveillance cameras in public places. After many decades, IP (internet protocol) cameras were released. They could send and receive information across computer networks which led then to the webcam. Webcams marked the beginning of the decline of closed-circuit television. Today, surveillance can be watched from anywhere in the world using internet and wireless communication.

By this evolution, surveillance is not a monopoly only for official places and businesses; it also can be expanded to cover home security. Average homeowners are now able to utilize inexpensive video surveillance by employing such systems [1].

Security-cameras are main components in surveillance systems. They provide businesspeople and homeowners with live or recorded video footage. These videos help people in the prevention of theft and provide an undeniable proof which

have led to the incarceration of criminals. Security cameras can be placed in a wired way or a wireless one. Wireless cameras are more advanced and flexible in installment, placing and could be hidden. Nowadays, due to the emergence of advanced network communication systems and advanced video processing, surveillance systems can be optimized in order to serve better the interests of people [2].

By developing enhanced surveillance systems, these people might be notified when something abnormal is happening. For example, at homes or businesses, the presence of unknown persons can be detected using face detection and recognition [3].

Surveillance systems' role can be diverged in a totally different sense that helps business growth. As an example, they can help managers know whether customers are being greeted in a good way and how was the behavior of employees with clients. These cameras can give a valuable feedback to improve the service provided. This leads to the increase of customers' confidence in the service or the product offered which will help them have a loyal customer.

Nowadays, the increase of percentage of theft in markets and small companies is considerable even with the presence of surveillance systems and their wide usage [4].

In order to reduce this problem of theft, a smart surveillance system that takes care of many challenging issues is needed. It is noticeable that cameras that are usually used cover only a specific region of interest and consequently, this leads to the need of deploying many cameras in order to cover an entire area. For this, we propose building a smart surveillance system that automatically detects and tracks a motion in order to keep the region of interest in the field of view of the camera. The camera can rotate horizontally and moves downward and upward in order to follow the motion in the whole area. On the other hand, while detecting motions, taking care of abrupt motions is needed. Sudden changes in the speed and direction of the object's motion or sudden camera motion are other challenges in object detection and tracking. This system can be of high benefit for homeowners and businessowners. Recently, because of the augmented rate of theft done by housekeepers, employees in houses and companies and because no available surveillance systems can detect motion and follow it, a real time system for human detection, motion tracking and analysis will be proposed in this paper. Covering motions in those indoors environment can reduce theft and vandalism in such places.

In this paper, we show in Section II related work while in Section III and Section IV we present our system design and system architecture respectively. In Section V we present the

implementation of the system and finally we conclude and present our future work in Section VI.

II. RELATED WORK

In the late 90's, tracking people and interpreting their behavior has become a common need for different applications such as smart rooms, wireless virtual reality interfaces and video databases. One of the most important visual problems is the ability to find and follow human people's hand, head and body. In order to reduce this problem and address this need, "Person finder" or "Pfinder" had been developed. It is a real-time system for tracking people and interpreting their behavior. Pfinder works in the presence of camera rotation and zoom by using of image to image registration techniques as a preprocessing step. It runs at 10 Hz on a standard SGI Indy computer, and has been tested on thousands of people in different locations of world and has performed quite reliably.

Considered as a real-time interface device, Pfinder has been used for information, video games, performance spaces. Also, it has been used for gesture recognition systems. For example, it recognizes 40-word subset of America Sign language with accuracy and precision.

Moreover, using simple 2D models, Pfinder uses MAP approach (Maximum a Posteriori Probability) to detect and track human body. The main idea of this approach is to incorporate a priori knowledge about people primarily to bootstrap it and recover from errors. So, in a wide range of viewing conditions, the system provides users by a multiclass statistical model of color and shape to obtain a 2D representation of head and hands [5].

W4 is a real time visual surveillance system composed of a set of techniques integrated into PC [6]. This system is used for simultaneously detecting people and their body parts and monitoring their activities. W4 answers to any manager, chief officer or homeowner questions about what people are doing, what they are holding, when they act and sometimes it can identify who is at that place based on face detection. W4 system can work in outdoor environment and during the night or other low light level situations.

The major features of W4 are:

- Modeling background scenes to detect foreground objects even when the background has many objects or for example in outdoor environment the background is full of tree branches, plants and vehicles.
- Recognizing people based on the shapes and periodic motion cues.
- Detecting six main body parts: head, hands, feet and torso.
- Determining whether the person detected is carrying an object.

W4 operates on gray-scale video imagery. It was implemented in C++ and runs under Windows NT operating system. It is also considered as low-cost surveillance system because it can run on Pentium PCs at 20-30 Hz.

SkyNet is a system developed in China that uses artificial intelligence combined with CCTV surveillance. It uses facial recognition and GPS tracking to overlay personal identification information. The system recognizes faces based on face++ linked cameras. SkyNet system knows which

person is acting inappropriately and can identify him/her because Chinese people are reportedly given a photo national ID by the age of 16 and the photos are stored in the government's database. Knowing the person's ID and having a photo will make the task of tagging and tracking him easily. At 2015, it is advanced in a way that the system can even track and identify vehicles and tag them with colors and types. The system is developed also for emotionally control the public.

In conclusion, we cannot neglect the need of existing such methods and benefit from their advantages to ensure human safety. However, the remaining of some problems, such as the disability to detect human motion in undefined postures, or automatically tracking the motion for a single person, when using a traditional web camera covering only a specific region make the situation worse. For these reasons, a real time system that detect human and track its motion with dynamic camera will be developed. This system provides users the ability to follow human motion in every single frame throughout the area, since security cameras will be viewed remotely from a laptop.

III. SYSTEM DESIGN

The main goal of this work is to put at the disposal of the beneficiary a real-time system for human and motion detection and tracking. This system, considered as an automated video surveillance for detecting and monitoring people, can be used by homeowners, security agencies in official places as well by the managers or employees in companies and economic institutions. It was built in order to let the users monitor and take the right decision when something abnormal is taking place.

Security cameras have several use cases that can be listed as follows:

- Real Time Monitoring: By using surveillance cameras, a user will be provided with a real time monitoring system of his place of business. This system neglects the need of the user to look through archival footage to find what he needs; instead he can simply watch things in the moment they happen. Best of all, through his mobile device, the user can view the video monitoring service wherever he is.

- Catch a Criminal: When a crime is committed and there is a surveillance camera, the user will be able to get a viable image of the criminal. For example, even in darkness, the user can use the camera footage and get the images in order to be able to recognize who the criminal is. Without this security system, it will be difficult to get a detailed description about the perpetrator.

- Sense of Security: Surveillance cameras help some users to create a sense of security in certain areas. In other words, if there are security cameras, people may believe that there may be less of a chance that a crime will be committed in the area watching over them.

- Theft Reduction: To prevent and reduce theft, surveillance cameras are an excellent way to be used in this case. With the presence of security camera, the user can track the motion of human instantaneously which is sometimes enough for a prospective thief to rethink about his plan of action. This will lead to a reduction in the incidents of theft,

saving business money of user and keeping his home and offices secure.

The user, who wants to benefit from this surveillance system, has just to put away the camera in some corner of the room and keep monitoring the scene via the monitor of the raspberry pi.

IV. SYSTEM ARCHITECTURE

Our system consists of a combination of hardware and software components. These components interact together in order to perform an autonomous movement tracking for efficient surveillance. In general, the system works like follows: The camera is first launched and starts capturing and recording the scene moment by moment. A python code has been developed to run and analyze the frames. The task emphasizes on background subtraction which is the key of detection of the motion using video processing. Background subtraction will be done continuously between the frames of the video recorded and their difference value will be computed and compared to a threshold value. This will help in avoiding negligible motions such as just moving the hand or tilting the head that do not require the movement of the camera in order to be covered.

The hardware components we used are the following:

- Raspberry pi 3 Model B+: the raspberry pi is a mini-computer which has a credit-card size, it can be plugged into a computer monitor or TV, it can be connected to many peripherals such as keyboard and mouse. It can perform simple to advanced functions like simply browsing the internet or playing high definition video. The main advantages of raspberry pi computer lie in its low power consumption and in its low mass, thus making it easily transportable [7] [8].

- Web camera: it is a mini camera that can be connected to the raspberry pi using USB cable. It has many interesting features such as its small size that allows it to be hidden and it can be placed in any wanted manner. Also, it provides a view of 120 degree and it supports night vision thus, it is possible to capture pictures during the night or through dark environment.

- Servomotors: A servo motor is an actuator that can rotate precisely in terms of angular position. It takes also into consideration both acceleration and velocity. Each of them can be controlled through coding. A servo motor also has capabilities that a regular motor does not have.

In addition to these hardware components, the used software components are the following:

- Raspbian OS: it is the operating system that we installed and used for the raspberry pi. It is optimized and free. It can be said to be as a set of basic programs and utilities that makes the raspberry pi running and it comes with thousands of packages and pre-compiled software [7].

- OpenCV: Open Source Computer Vision Library is an open source library used to perform computer vision and machine learning based operations and tasks. It includes 2500 libraries that can be used for detecting and recognizing faces, recognizing objects and detecting movement of humans and objects in a video and many more. It has a purpose also to accelerate the use of machine perception in commercial products. The library is cross-platform and free and has C++, Python and java interfaces [9].

- Thonny: it is a platform and python IDE and is supported by Raspbian Operating System [10].

- VNC: Virtual Network Computing is used to control computer devices remotely over a network connection. It shares the keystrokes and the mouse clicks from one device to another which allows the management of the remote computer [11].

Figure 1 and Figure 2 present the overall behavior of the different components.

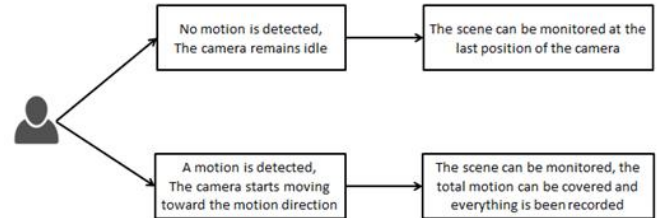


Figure 1 Overall behavior of the system

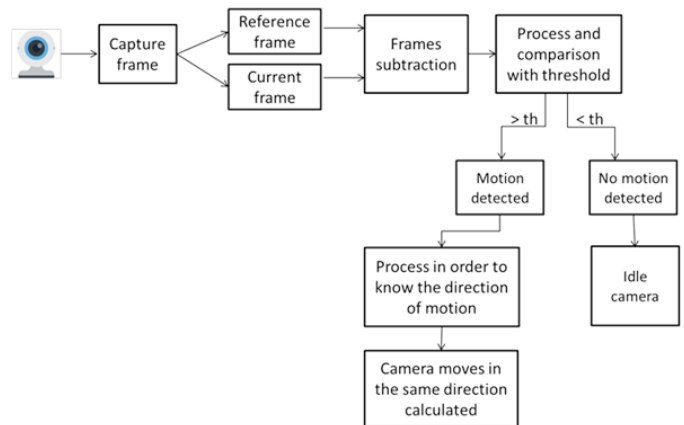


Figure 2 Detailed behavior of the system

The user of our system can surveil and cover the whole motion happening through the usage of a single camera. The process starts by launching the camera and capturing frames. These frames are processed in order to detect and localize motion. This is done by performing frames subtraction between a reference frames and current frame. If a motion is detected, its direction is calculated, and the camera is moved toward this direction in order to keep track of the moving human or object.

V. IMPLEMENTATION

In this section, we present the implementation details of the system.

A. Hardware implementation

- Servo Motors: Each servo motor has 3 wires. The power wire is connected to the 5V pin of the raspberry pi. The ground wire is connected to the ground pin of the raspberry. Finally, the signal wire is connected to a control pin of the raspberry.

- Pan-tilt: After assembling the pan-tilt as shown in Figure 3, we connect to it the pan servo that moves horizontally and the tilt servo that moves vertically. Finally, we attach the webcam onto the pan-tilt.



Figure 3 The used pan tilt

B. Software implementation

Virtual Environment: It is a tool that helps keep dependencies required for each project separate. So, the main purpose of a virtual environment is to create an isolated environment which means that each project can have its own dependencies, regardless what dependencies every other project has.

In this virtual environment, all the libraries needed to build our surveillance system will be added into it. In this context, we will mention all the libraries used for our python program:

- CV2: It makes it easier to find the package with search engine. CV2 is considered as an updated version of the old interface in old OpenCV that was named as “cv”. It is the name that OpenCV developers chose when they created the binding generators.
- Numpy: It is the most fundamental package for scientific computing with python. It contains: a powerful N-dimensional array object, sophisticated functions, tools for integrating C/C++, useful linear algebra, Fourier transform and random number capabilities. To download this library, we simply execute the following command: “sudo pip3 install numpy” [12].
- Imutils: A series of functions to make basic image processing functions such as rotation, resizing, translation with OpenCV and Python3. We use “pip3 install imutils” to download this library [13].
- Deque: It can be implemented in python using the module “collections”. Deque is preferred in the cases where we need quicker append and pop operations. So there are several operations on Deque such as ‘append()’ to insert the value in its argument to the right end of deque, ‘appendleft()’ to insert the value in its argument to the left end of deque and ‘pop()’ to delete an argument from the left end of deque [14].
- Argparse: It makes it easy to write user-friendly command –line interfaces. The program defines what arguments it requires, and Argparse will figure out how to parse those out of sys.argv. In addition, the Argparse generates help and usage messages and issues errors when users give the program invalid arguments.
- Datetime: It supplies classes for manipulating dates and times.

- Time: This module provides various time- related functions such as time.clock(), time.sleep(secs), time.get_clock_info(name).
- VideoStream: In short, we stream live video using Motion JPEG, which just sends JPEG frames successively.

C. Implementation Summary

In our system, the mission was to build a functional system that detects motion, determine its direction and follow it using the raspberry pi, USB web camera, two servomotors and a pan-tilt bracket.

The task starts by installing the necessary packages for the web camera and by checking if our USB web camera is detected by the raspberry pi by inspecting the dev directory. If a file named video0 exists, this implies that the web camera is successfully detected by the raspberry pi. As shown in Figure 4 this file exists.

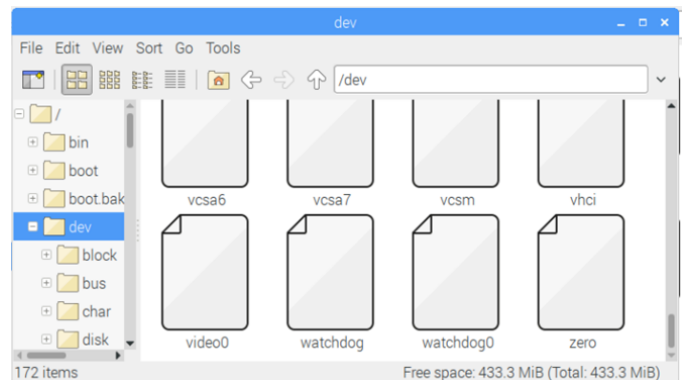


Figure 4 video0 file existing in the dev directory

After this, we install the fswebcam package using the following command:

```
$ sudo apt-get apt install fswebcam
```

After being installed, we test its functionality by taking a sample picture through the terminal using the command shown in Figure 5.

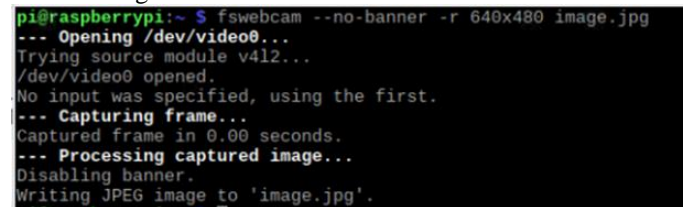


Figure 5 Taking a picture using the web camera through command line

We then wrote a functional python code that records a video in a real time manner using the Thonny platform, opencv and numpy libraries. The main functions applied were cv2.VideoCapture(0), where 0 is the parameter indicating that we are using an external web camera, cv2.VideoCapture(0).read() and cv2.VideoCapture(0).release() at the end when we need to end the live video recording.

In addition to recording a live video, the next step is to detect the motion of moving human. This process is done using background subtraction between consecutive frames. This is accomplished using the function cv2.absdifference followed

by a blurring function of cv2.GaussianBlur and a threshold calculation using cv2.threshold. Based on the result of this function, a decision is taken whether there is a noticeable motion or not.

Just detecting the motion is not enough for accomplishing our final goal. Therefore, a python code that detects the motion direction is needed. We wrote a code that detects the direction of motion by calculating the difference between frames as shown before and draws rectangular contours around the moving human or object using the functions cv2.findContours, imutils.grab_contours and cv2.boundingRect. After that, we calculate the centroid of the moving mass. The difference between the positions of the calculated centroids in both reference and current frames gives us two values “dx” and “dy”. Both “dx” and “dy” determine the translation along horizontal direction and vertical direction respectively. The values obtained are compared to a specific threshold of 20 px. This threshold is placed in order to avoid negligible movements. If the “dx” sign is positive, the direction is noted to be to the east. Otherwise, it will be noted to the west. Concerning the vertical axis, if the value “dy” is positive, the vertical direction is set to be to the north and to the south otherwise.

In terms of hardware controlling, after assembling the pan tilt, we connect it to GPIO21 and GPIO22. The first is dedicated for the movement across the horizontal axis while the second is dedicated for the vertical one. We can note that each servo motor can be controlled in order to move to certain angle as indicated in Figure 6. We can send an electrical signal called PWM, which stands for pulse width modulation. This is done using GPIO.PWM(servo, 50) function, where 50 tends for setting the frequency to 50 Hz. According to the below formula, we set a duty cycle for the pwm using this function pwm.ChangeDutyCycle(dutyCycle) where $DutyCycle = angle / 18.0 + 3.0$ (Equation 1).

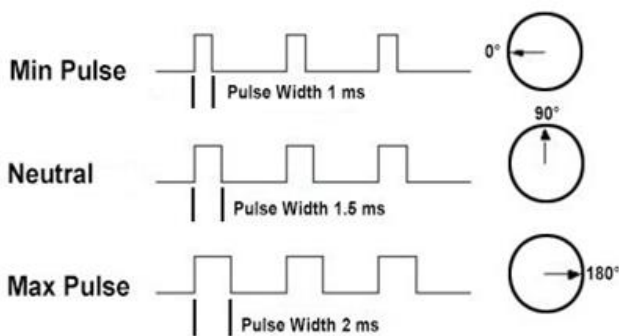


Figure 6 Controlling the position of a servo motor using pulse width modulation

Equation 1 has the known form of a linear equation $y = a x + b$. The value of the slope a and the value of the offset b can be obtained by considering two known points A and B that will be indicated later. The servo motor has a range of 180° . Theoretically it can be at the initial position of 0° when a pulse of 1 ms is applied and when 2% duty cycle obtained by calculating the ratio of pulse width over the period both in milliseconds which is $1 / 20$. The servo also can be at neutral

position of 90° which corresponds to the pulse of width 1.5 ms and a duty cycle 7.5%. For the maximum pulse of 2 ms for 180° the duty cycle is 10%. Therefore, the points A and B can be set respectively to (0, 2%) and (180, 12%) so the slope is obtained through the following calculation $(12-2) / (180-0)$ which corresponds to $1/18$ in the Equation 1 and the value of b can be found by setting the corresponding coordinates of the point A for example and the value of the slope to get a 3.0 value.

Finally, after interpreting and analyzing all the steps above, a final algorithm for our system is set and presented in the chart shown in Figure 7

D. Results

The results of our developed algorithm were satisfactory. At first, the servo motors are positioned at 90 degrees on the vertical and 90 degrees for the horizontal. It will be able to move from 90 degrees to 0 degrees if the motion is to the west and from 90 degrees to 180 degrees if the motion is to the east. Hence, this servo is working for the pan. The servo responsible for the tilt, is the one responsible for the vertical axis that moves from 90 degrees to 0 degrees progressively if the motion is upward considered. It moves from 90 degrees to 180 degrees if it is downward. It can be noticed also that if no motion is detected the camera will remain idle since the two servo motors will remain at the position of 90 degree, a simple message is shown for the user indicating that no motion is detected.

If a motion is detected, for example across the horizontal axis, the servo motor responsible for panning increases or decreases 10 degrees depending on the direction of the movement. But rotating a servo motor every time a motion is detected and without putting positional constraints is not an efficient approach. This is because the monitored moving object might leave the scene. Therefore, we created a virtual bounding box. Only when the monitored object or human gets near the boundaries of the scene, servos will be rotated in order to keep tracking the moving target. When a motion occurs at the center of the scene, the servos will retain their last positions.

VI. CONCLUSION & FUTURE WORK

The system we discussed in this paper is a real time system for detecting, tracking motion using a camera and servo motors. It is hoped that the outcome of this system would serve in future researches regarding this domain of work. In our future works, we aim to extend our system in a way that the user can control the directions of the pan-tilt camera, adjust the angles and speed of the camera to monitor the entire region through a web interface and android application. In addition, the user will receive alerts and notifications through these platforms when the system detects an abrupt motion to inform him that there is something abnormal is happening. We are looking forward for enhancing our system design also to be able to detect more than one person and track their motions to ensure more security, real time monitoring, reduce thefts and crimes. The ways of enhancement and improvement can also include face detection and recognition in order to alert the user about the identity of moving human which will help also in increasing the accuracy of the system.

REFERENCES

- [1] T. Juhana and V. G. Anggraini, "Design and implementation of Smart Home Surveillance system," in *10th International Conference on Telecommunication Systems Services and Applications (TSSA)*, Denpasar, Indonesia, 2016.
- [2] K. Kobayashi, K. Iwamura, K. Kaneda and I. Echizen, "Surveillance Camera System to Achieve Privacy Protection & Crime Prevention," in *10th International Conference on Intelligent Information Hiding & Multimedia Signal Processing*, Kitakyushu, Japan, 2014.
- [3] H. Yu, J. Shen and X. Du, "Camera surveillance system based on image recognition," in *International Conference on Electronics and Optoelectronics*, Dalian, 29-31 July 2011.
- [4] S. Fleck and W. Straßer, "Smart Camera Based Monitoring System and Its Application to Assisted Living," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1698 - 1714, 2008.
- [5] C. Wren, A. Azarbayejani, T. Darrell and A. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780 - 785, 1997.
- [6] I. Haritaoglu, D. Harwood and L. Davis, "W/sup 4/: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809 - 830, 2000 .
- [7] W. Gay, *Advanced Raspberry Pi: Raspbian Linux and GPIO Integration*, Berkely, CA, USA: Apress, 2018.
- [8] M. Richardson and S. Wallace, *Getting Started with Raspberry Pi*, USA: Maker Media, Inc, 2013.
- [9] K. Pulli, A. Baksheev, K. Korniyakov and V. Eruhimov, "Realtime Computer Vision with OpenCV," *Queue - Processors*, vol. 10, no. 4, p. 40, 2012.
- [10] A. Annamaa, "Thonny: a Python IDE for Learning Programming," in *ACM Conference on Innovation and Technology in Computer Science Education*, Vilnius, Lithuania, 2015.
- [11] B. Harvey, "Virtual Network Computing," *Linux Journal*, no. 5, 1999.
- [12] S. v. d. Walt, S. C. Colbert and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22 - 30, 2011.
- [13] "Github - imutils," [Online]. Available: <https://github.com/jrosebr1/imutils>. [Accessed 12 04 2019].
- [14] P. S. Foundation, "Python Documentation - Collections," Python Software Foundation, 19 10 2019. [Online]. Available: <https://docs.python.org/2/library/collections.html>. [Accessed 23 03 2019].