Estimating the height of a person from a video sequence

O. Kainz, M. Dopiriak, M. Michalko, F. Jakab and P. Feciľak
Department of Computers and Informatics, Technical University of Kosice, Kosice, Slovakia
ondrej.kainz@tuke.sk, matus.dopiriak@student.tuke.sk, miroslav.michalko@tuke.sk, frantisek.jakab@tuke.sk, peter.fecilak@tuke.sk

***Abstract*— The research described in this paper focuses on a development of experimental solution, which estimates human height from a video sequence. The main requirements include portability, low hardware requirements, and accuracy. The paper includes analysis of the algorithms employed for image processing, human detection and methods for estimating height of human. The experimental software solution is implemented in Python programming language. The SSD algorithm was selected to detect the person in the image and the output was later on utilized to calculate the real human height. A calibration object in a form of ArUco marker and its known dimensions became the foundation for calculating the real human height. The experimental testing confirmed the accuracy of the developed solution to be sufficient for the real-world application. The most accurate height, with just a deviation of 0.001 meters, was estimated at a distance of 5 and 8 meters. The approach presented in this paper is also to be used as a complement by newly proposed unmanned aerial vehicle security system.**

Keywords*— *calibration, deep learning, human detection, human height estimation, image processing, motion detection.

I. Introduction

People are naturally equipped with the ability to detect and recognize various re-al-world objects. Even very small children are able to distinguish objects of different colors, sizes or shapes. The computer vision aims to solve this task using computers machines. The object detection and recognition algorithms are at the very core of the ability to under-stand what is in the image or video. However, prior to their use such input has to be processed, which may include edge detection, conversion of the image to gray color, segmentation of the image, etc. There are also various conditions that impact the accuracy of the detection, such as lightning or the distance of the objects from the camera. Significant improvements of hardware components over the last few decades have contributed to a considerable development of computer vision. Period we live in is a milestone for neural net-works which have rapidly increased the demand for this field.

In this era, the applications of computer vision are very noticeable. The development of self-driving cars or unmanned aerial vehicle (UAV) is directly connected to computer vision. Robotics is another field, which depends on computer vision. Robots have to be able to move and interact with the real world. Pedestrian detection may enable the police to catch criminals and to prevent physical attacks. Virtual reality enables to bring virtual objects to the real world. Knowing the dimensions of real-world objects is the next step for adaptation of the virtual objects to real scene. However, extracting the dimensions from the image or video may replace the general way of measuring people. It is useful for statis-tical measurements due to its massive automation, which does not require physical presence of person. Estimating the height of person can be also used as additional factor to the user authentication.

This paper proposes and implements an experimental solution which enables detection of the human from the video sequence and subsequently estimates the height of a person using the algorithm proposed by one of the authors. As a part of the research, we also compare specific human detection algorithms and evaluate their advantages and disadvantages. The proposed solution presumes calibration process to occur prior to the estimation itself. Outputs from the calibration are then inputs to the estimation. Once the scene will be calibrated the estimation of the height may take place using just single camera device. As such device we may also consider UAV, which will extend the research and is expected to enhance the proposed method for the height estimation.

II. Related Work

In this part we present several image processing, motion and human detection algorithms, included are also the methods for estimating height of the person.

A. Image processing

Object detection algorithms identify various features in the image. One of the fundamental features is an edge, which detects the structure of objects. As the publication by Jain et al. [1] describes, edges are those regions in an image where changes in intensity are considerable. Formally, those regions have discontinuities. Same source [1], introduced a Sobel edge detector, which utilizes a gradient-based technique. The gradient is a derivative of a function. In dimensions, the gradient is represented as a vector which has two significant attributes: magnitude and direction.

As discussed in [1], the Gaussian smoothing filter belongs to the category of linear smoothing filters. Such filter is utilized to remove noise and blur the image. The filter is usually used prior to edge detection.

B. Motion detection

Motion detection algorithms play significant role in the process since they are the input for human detection process. First, we cover the temporal difference, described in a research by Sehairi et al. [2]. It is a simple and performance-friendly method utilized for detecting changes in video frames. This algorithm begins with a selection of two consecutive frames, which are subsequently converted to gray color. Then, the difference between coordinates of each pixel of the frames is computed. The resulting number is in absolute value.

Another background subtraction technique was described in research by Piccardi [3], in this case the pair of images is required. The described method Running Gaussian Average creates a background model using Gaussian distribution of each pixel. It is used as a probability density function, which is updated by a running average. Another method, proposed

978-1-6654-2102-7/21/$31.00 ©2021 IEEE

by Stauffer and Grimson [4], is called Mixture of Gaussians. It differs from Running Gaussian Average in pixels modeling and updates to a model.

Next technique is the optical flow, which provides additional information, such as velocity and direction of moving objects in the form of a vector. Optical flow can be computed using various methods that differ one from another based on the number of tracking points. Lucas-Kanade method [5] estimates the motion of interesting features, e.g., corners. Another optical flow algorithm, the Gunner Farneback's algorithm [6] [7], estimates the motion of all the points in the frame.

As a part of this research, all of the abovementioned motion detection algorithms were tested and the results are stated later in the text.

C. *Human detection*

The object detection has become very popular in the last decades due to the development of deep learning techniques and a significant improvement of hardware components. Below, the most relevant approaches we considered the in the proposal phase of the solution are presented.

The Histogram of Oriented Gradients (HOG) detection algorithm has been released by Dalal and Triggs [8] in 2005. It aims to find the features of the objects through a gradient evaluation. The authors of the study [9] utilize HOG-SVM along with the Gaussian mixture model, which is used for foreground detection. The background contours detection process creates the silhouette of a person. The background subtraction is resistant to various light conditions, yet it detects the person with the shadow. The combination of these approaches depicts the silhouette more accurately.

The original version of a Region-based Convolutional Neural Network (R-CNN) [10] applies a selective search algorithm to extract the regions of interest. These regions of interest are utilized as an input for the pre-trained CNNs. The regions of interest are modified to be compatible with the networks. The Support Vector Machine (SVM) carries out the classification of specific features to identify the objects. A position of classified object is visualized by a bounding-box. The regression method is used to prevent erroneous localization. However, such algorithm is considered to have a slow detection speed. An im-proved version, known as Fast R-CNN, introduced in the research by Girshick [11], is able to fix substantial problems of R-CNN. The Softmax classifier is used instead of SVM, since it yields slightly better results in experiments. Experiments on PASCAL VOC 2012 dataset have reached the highest accuracy with 66% mean average precision (mAP). Another, im-proved version of Fast R-CNN is called Faster R-CNN [12]. The major difference from Fast R-CNN is the use of Region Proposal Network (RPN) instead of Selective Search. The pro-posed system has reached almost 79% mAP on the PASCAL VOC 2007 along with COCO dataset. Detection time is in the range from 5 to 17 frames per seconds (FPS).

YOLO (You Only Look Once) detection algorithm introduced in 2016 [13], presents a new way of speeding up the detection time. The main principle is the use of one CNN during the whole detection process. Multiple bounding boxes displaying the same object are correctly removed using the Non-Max Suppression. YOLO is able to run at speed 45FPS on a Titan X GPU. A faster version of YOLO can even run 150FPS. Thus, such algorithm is appropriate for the real-time object detection.

Single Shot MultiBox Detector (SSD), introduced in the research by Liu et al. [14], at-tempts to find a compromise between the speed and accuracy. Its structure is similar both to YOLO algorithm and Faster R-CNN. This method utilizes a single deep CNN. Initial layers use VGG-16 network, which can be interchangeably substituted by other networks. In the following research by Zhou et al. [15], the speed of SSD algorithm was improved. VGG 16 network was substituted by Wide Residual Network. SSD using WRN with 192X192 input size reaches 89FPS. It is 22FPS more compared to the tradictional approach of SSD algorithm.

CNN have revolutionized the field of human detection and became a basis for sub-sequent studies. Faster R-CNN produced some of the most accurate results but at lower speed. YOLO and SSD aimed to address this problem in order to reach real-time object detection. Their results are quite similar and depend on various conditions, such as the in-put size of the image.

D. *Human height estimation*

Estimating the height of the person from the image or video is not as easy as it may seem. The issue itself has not been fully resolved and it is still a subject of research. Below are presented several studies that deal with this topic and try to estimate the height using various techniques and approaches.

Approach introduced in the research by Othman et al. [16] presents a real-time object detection and the measurement of its size. This approach utilizes Raspberry Pi 3 and Raspberry Pi camera to capture a video sequence. The measurement is based on a reference object, which is always the one on the far left. The accuracy of the first experiment, in which the objects are in front of the camera, yields the accuracy from 95.45% to 98.23%. With the camera placed above the objects, the accuracy of the second experiment is greater, ranging is from 96.20% to 98.26%.

The use of two cameras as an input criterion was used by Mohd et al. [17]. Authors used two calibrated cameras in a parallel position to capture the same scene. Estimation of the size reaches precision, which is roughly 0.03 m.

In the research by Dokthurian et al. [18] a human height estimation using only one calibrated camera was proposed. In this case, two positions of the camera were utilized. The results of the proposed approach indicated 0.95% error, along with the standard deviation of 0.00158.

The study by Kainz [19] deals with several approaches that aim to extract the height from a single image captured by a digital device in a stationary position. The first method is based on a calibrated image, vertical field of view and distance between the camera and the measured object. The significant value, also relevant for this study, is the ratio between the pixel value of the object in the image and its real value in meters. This ratio is well known calibration factor (*Cf*). Alternative methods were also proposed, e.g., use of ArUco marker as a complementary aid for the calibration process. Thus, constant measurement of the distance is no longer required due to calibration process done by ArUco marker, which was previously positioned in various distances. Calculation of the calibration fac-tor is automatic and is based on the calibration, another extracted parameter is the distance of the marker from the lower part of the image. The real environment experiments showed error between 1% and 2% for both approaches.

More novel research by Tonini et al. [20] adapt the pinhole model for the estimation of the height from the images. Authors utilize UAV to capture the images, prior to estimation the calibration of camera is carried out and the lens distortion is compensated. However, several other factors had to be known as well, camera pitch angle and distance between the camera and the measured subject. Several tests were made in various distances, from 3 to 37 m having the root mean square error of ±0.075 m. Also, authors observed flat behavior in the height estimation in the distances between 15/25 m and 50 m.

Yin et al. [21] estimate the height of the human body in various poses from the single image that contains the information on depth. The height is predicted based on the com-bination of segmented human body parts in form of intermediate representation, their lengths and depth information. The Fully CNN is used to predict the lengths of the human body parts. The overall accuracy was estimated up to 99.1%.

Interesting approach to estimation of the height from the video using the gravity was proposed by Bieler et al. [22]. As a part of the research the ArticulatedFreeFall video dataset was created. Center of mass trajectory was estimated using the AlphaPose, which was used also for detection the location of the key body parts. Mean absolute error of the estimation for distances 4 to 7 m was 0.039 m.

Estimation of the height of person or any other object may be done in various forms, e.g. using time-on-flight camera. Leo et al. [23] used such approach to estimate the height, where the Kinect camera was utilized. The modified version of RANSAC is used in the re-search for planes extraction in a point cloud. The accuracy of the heigh estimation was 0.005 m, which is promising.

III. Methods

The proposed experimental solution is based on the research by Kainz [19]. Our goal was to design a performance-friendly, independent, and portable experimental software solution, which is able to detect moving objects, classify whether they are human, and, if so, estimate their height. As a part of the research, we carried out the design of the system and set the system requirements and restrictions.

First, let us define the solution on the global scale. In the Fig. 1 are depicted the two main processes – calibration and height extraction. Prior to height estimation the calibration process must occur. Calibration process utilizes ArUco markers with the known height. The height of marker in pixels and the distances from the lower part of the frame to the marker is acquired in several distances from the camera are extracted, both in pixels. For each distance, the calibration factor Cf is calculated and height in pixels from the low-er part of the frame to the marker is extracted. Both serve as the input to calculation of the function rule. The second process – the heigh extraction of some unknown person in specific distance from the camera utilizes the very function rule to estimate the height.

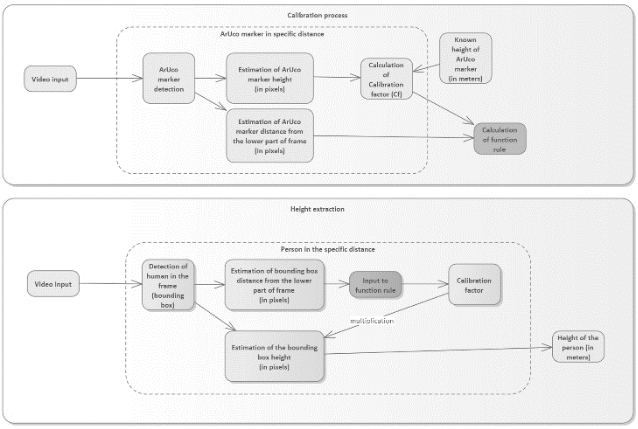

Calibration process
ArUco marker in specific distance
Video input
ArUco marker detection
Estimation of ArUco marker height (in pixels)
Calculation of Calibration factor (Cf)
Known height of ArUco marker (in meters)
Estimation of ArUco marker distance from the lower part of frame (in pixels)
Calculation of function rule
Height extraction
Person in the specific distance
Video Input
Detection of human in the frame (bounding box)
Estimation of bounding box distance from the lower part of frame (in pixels)
Input to function rule
Calibration factor
multiplication
Estimation of the bounding box height (in pixels)
Height of the person (in meters)

Fig. 1. An overall overview of the proposed approach.

A. Requirements and restrictions

The camera that observes the scene is presumed to be static. The solution requires at the least two input videos captured by one camera. The first input is used for calibration, a process in which a reference object in pixels is detected and two values are extracted. The first value is height of the object and second its distance from the lower part of the frame to the object, both in pixels. Note that the reference object is ArUco marker and its dimensions in meters are known. The second video input video represents the actual estimation of the unknown height. The main parameters in this case are height of a person and the distance from the bottom side of the frame, both in pixels. The very first frame of the second video is recognized as background. Thus, a person should not be visible at the beginning of the video. To provide good results, the objects should be captured from the front side and the posture of detected human being should be upright.

B. Calibration and height estimation

The solution, as can be seen in Fig. 1, is divided into two main processes:

1) *Calibration process*: The calibration utilizes ArUco marker as a reference object, the real height of this object in meters is known. The goal is to extract the height of the marker and distance from the lower part of the frame to lower point of the marker, both in pixels. Then, based on the height of marker, the *Cf* is calculated for the various distances from the camera. The *Cf* and distance from the lower point of the frame serves as the input for the calculation of the function rule.

2) *Height estimation process*: The height estimation is the actual estimation of the un-known height of any person in the video. This step is independent from the calibration process. To estimate the movement in the video, the background subtraction is utilized. The area where the movement occurred is separated from the background, i.e., becomes foreground, while the background remains black. Once, we retrieve the region of interest the automated human detection is performed. Providing the detection yields positive results the height of the person in pixels is stored and another height from the bottom of the frame is used as the input to the function rule. Function rule then outputs the calibration factor which is then multiplied with the height of the person in pixels, outputting the actual height in meters. As the person moves in the video, new frames are calculated, and all values of the estimated heights are averaged.

C. Comparison of motion detection techniques

One of the most challenging parts during the implementation of the motion detection algorithm is creating a binary image, while finding the proper value representing the variance between two consecutive frames. Based on our testing, the temporal difference algorithm reaches performance 19.98FPS for the frame size of 640x480 pixels (see Fig.2a). Also, the algorithm contains the black parts inside when detected moving object. Output of the traditional background subtraction algorithm is depicted in the Fig.2b, this also produces small holes but does not contain any noise. The Mixture of Gaussians algorithm is more advanced, also, silhouettes on binary frames depicted in Fig.2c are more accurate and al-most all crucial information is captured. The classic background subtraction algorithm and Mixture of Gaussians reach performance 18.31FPS and 16.0FPS, respectively, for the frame size of 640x480 pixels. Our experiments showed high computational requirements of the Gunner Farnebrack's algorithm, having performance only 5.85 FPS for the frame size of 640x480 pixels. The most suitable approach is the Mixture of Gaussians, due to its compromise between performance and accuracy.

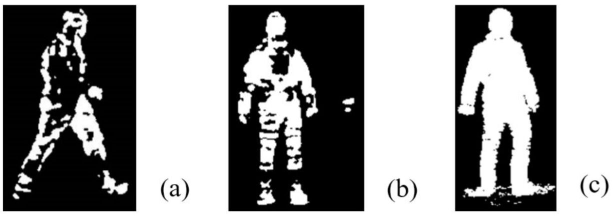


Fig. 2. Background subtraction: (a) Temporal difference, (b) Traditional, (c) Mixture of Gaussians algorithm.

D. Detection of human

As a part of the implementation, the testing of HOG algorithm was following the re-search [9], however we were not able to detect humans properly. The error might be due to our choice of different pre-trained SVM classifier or due to shadow non-removal. Note that experiments showed performance of 7.64 FPS for the frame size of 640x480 pixels.

Also, we tested the YOLOv3 pre-trained neural network. The YOLO algorithm, optimized for CPU computation, reached performance of 7.44 FPS for the frame size of 640x480. The drawback of this algorithm is double detection of one person and a larger deviation of the bounding-boxes height among the frames. The Fig.3a depicts one frame of YOLO algorithm optimized for GPU computation, we reached only 0.91 FPS again for the frame size of 640x480.

The Fig.3b depicts one frame from a video sequence captured and recognized by the SSD algorithm using a pretrained neural network called MobileNet. Experiments showed performance of 1.22 FPS and almost perfect bounding-box. The SSD algorithm provides a compromise between accuracy and performance. Therefore, this algorithm was selected for the implementation of the experimental software application.

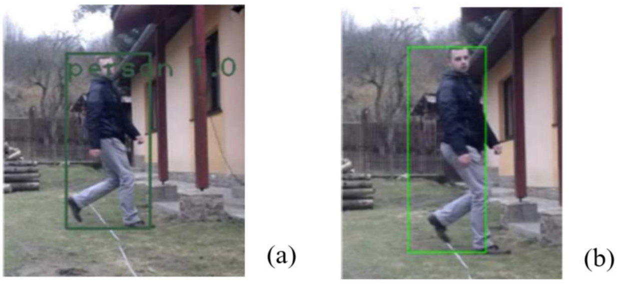


Fig. 3. Human detection (with bounding box): (a) YOLO algorithm, (b) SSD algorithm.

E. Estimation of human height

Proposed experimental solution should enable estimation of the height of a person using a single static camera utilizing a reference object - ArUco marker. Bearing in mind the nature of the solution, we selected the approach proposed by Kainz [19]. As already mentioned, approach described by the author includes estimation of calibration factor *Cf* and distance from the lower point to the object, both extracted from the scene in various positions. The ArUco marker will be used as the reference object with the known height (both in meters and pixels). Once the calibration is done, the heights of unknown subjects, in our case human, can be estimated.

The calibration process can be divided into two parts:

1) *Extracting the height of marker and its distance from the lower part of the frame*: This part is based on the use of a reference object - the ArUco marker, which is uniquely identified with ID. Extracted information is in pixels (in the Fig. 4 depicted as marker's distance). Note that also dimensions of marker in meters are also known.

2) *Calculation of function rule*: A value in pixels for each extracted distance of marker must exist. For simplification, we substitute absolute distance between the camera and the object (in meters) with the distance from the lower edge of the frame to the lower part of the detected object, in pixels (in the Fig. 4 depicted as marker's distance). Another value in pixels is the height of the object (see marker's height in the Fig. 4), this value is the input for the formula (1) – the dimensionless number called calibration factor *Cf* is defined as:

$$Cf = h_{real} / h_{pixel} \tag{1}$$

where h_{real} is the height of the reference object in meters and h_{pixel} is the height of the reference object in pixels.

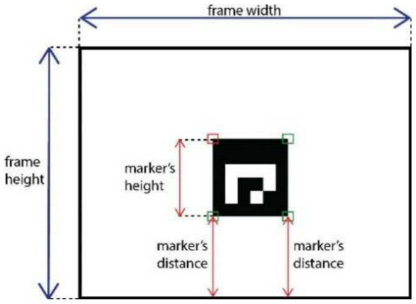


Fig. 4. Marker's distance and height extraction from the frame.

The detection range of marker is determined by the start and end point. Such approach allows selection of a specific location inside an image, allowing height to be extracted as accurately as possible. Having *Cf* only for some distances was not optimal solution, thus the function rule was to be found. The function rule will return a calibration factor for each value of the distance (extracted from the image). Convenient functions were selected based on the curve acquired from the testing procedures. Example of such curve, depicted in Fig.5, is similar to cubic, exponential and quadratic function. Data points are the distances from the bottom of the marker to the lower point of the frame, as the domain, and the *Cfs*, as the range of the function, are utilized to fit the function and compute its coefficients. Yet another way to calculate the *Cf* is to utilize linear function, which is based on relationship between the distance of the marker and its height. This is the end of the calibration process.

As for the height estimation process, based on the specific distance of the person from the camera the value in pixels is acquired. This value is extracted from the lower part of the frame to the lower part of the bounding box. The same value is then used as the input to the already calculated function rule. The output is the very *Cf*, then the only thing left to do is to multiply *Cf* with the height of the bounding box in pixels.

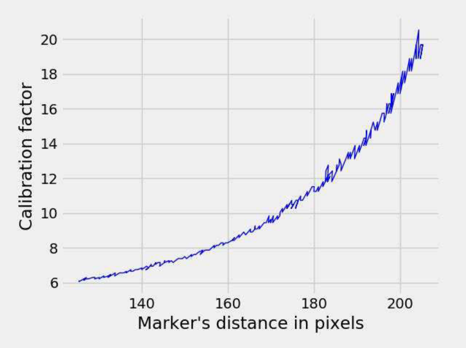
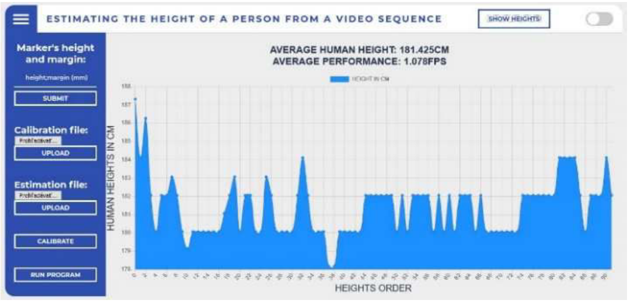


Fig. 5. Relation between the distance of reference object and calibration factor.

F. Experimental implementation

The proposed experimental software solution is implemented as a web application for desktop. The Python programming language was selected due to its useful libraries, frameworks, simplicity, and high efficiency in implementation.

The main functionalities of the solution include enabling or disabling of the motion detection, adding the height of the marker and its margin, adding of valid calibration and estimation file, starting the calibration and estimation process. The Fig.6 depicts user interface with visualized data from the height extraction. The Flask framework is used to send the data to the browser, this includes the information on the average height and all of the heights. Programming language Javascript (ECMAScript 2018) and its library chart.js are responsible for data visualization in the browser.


ESTIMATING THE HEIGHT OF A PERSON FROM A VIDEO SEQUENCE

AVERAGE HUMAN HEIGHT: 181.425CM
AVERAGE PERFORMANCE: 1.078FPS

Fig. 6. User interface: Output from the height estimation from video.

IV. Results

The functionality of the experimental software solution was carried out in the real environment. Also, we include the requirements necessary to achieve the most accurate results. Experimental videos were made using Logitech C930E camera, which is able to capture an image in resolution of 1920x1080 pixels. However, we captured videos in resolution of 640x480 pixels.

All the experiments were carried out in the outdoor conditions due to Covid-19 re-strictions and also because of greater distances, which can be up to 9 m. The surface is the grass and therefore, the reference object had to be at least 0.02 m above the surface. Human detection using the SSD algorithm is the most accurate when the person does not blend with surroundings. Lightning conditions are also important since the direct sunlight makes the video brighter, mainly the white parts can be washed out. The white margin around the ArUco marker separates the reference object from its surroundings due to its contrast with black color. We used the ArUco marker identified as ID 1 (see Fig. 4) with the height of 0.236 m. However, the lower part of the reference object is located 0.034 m from the ground. We predicted that the distance of the lower part of the ArUco marker is at the level of the feet. All tests were carried out with one person with the height of 1.8 m. We consider that humans wear shoes, which increases the height, as does their hair. As a result, we decided to add 0.015 m to the real height, resulting in a height of 1.815 m.

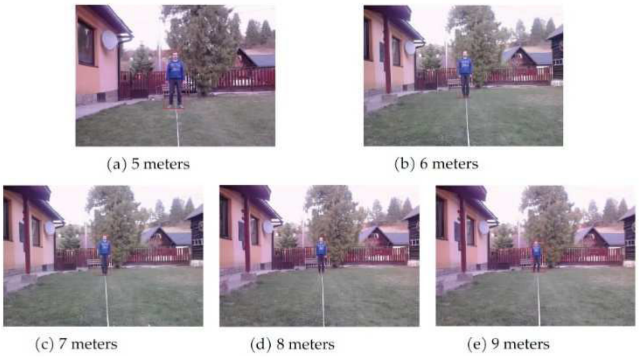
The input for the SSD algorithm is a colored binary frame, which can improve the performance and the accuracy of the algorithm. The experimental verification executed at a distance of 5 m showed that the use of a colored binary frame distorts the resulting height and therefore cannot be implemented. Deviation without colored binary frame was 0.028 m, however deviation with the colored binary frame was 0.116 m.

The marker is 0.034 m above the surface due to the detection accuracy, this height is referred to as the margin. The calibration process was done in the range from 4 to 6 m and the subsequent human detection at a distance of 5 m showed significant results. When the margin was added to the calculations, then the difference from the real height was 0.001 m. Yet, when the margin was not taken into consideration, then the difference from the real height was 0.1 m.

Due to Covid-19 restrictions, the testing of the implemented software solution was done by one person at a distance of 5 to 9 m from the camera. The recorded videos, in which a person either stands or walks, were trimmed to 1 to 3 seconds. The number of processed frames was approximately 30 to 90 frames.

The Table I. contains the human heights acquired from testing. The person was standing in front of the camera and was not moving at all. Note that the calibration pro-cess was carried out prior to testing. The results of detection from various distances are depicted in the Fig. 7. The most promising results give linear and cubic functions. They are characterized by the lowest average deviation, linear and cubic functions having a value of 0.0172 and 0.0162, respectively. The linear function is the most accurate at a distance of 5 and 8 m from the camera with deviation of 0.004 and 0.002, respectively. The cubic function is the most accurate at a distance of 5 and 8 m with deviation of 0.001 and 0.003, respectively. It is remarkable that a quadratic function has deviation only 0.001 m at mentioned distances, however its average deviation is 0.037. The exponential function acquired the highest average deviation 0.948.

The second test was done in the same external environment by the same person as in the first test. The biggest difference is that the person walked horizontally across the sur-face in two directions and took small steps. Table II. depicts the summary of the second testing procedure in the same format as the previous one. The cubic and the quadratic functions produced the lowest average deviations, 0.037 and 0.0372, respectively. The linear function and the exponential function produced a slightly higher average deviation, 0.0396 and 0.0498, respectively. The lowest deviations were measured at distances of 5 and 8 m from the camera.


(a) 5 meters

(b) 6 meters

(c) 7 meters

(d) 8 meters

(e) 9 meters

Fig. 7. Testing of detection in various distances.

The speed of the abovementioned tests was calculated for each distance and then averaged. The first test ran at an average speed of 1.151FPS and the second one at an average speed of 1.120FPS. The system was able to estimate human height using 99.7% of the frames. When walking, the results were significantly worse and the estimation was per-formed using only from 49.7% of the frames, includes those in which a person was in the range of the calibration. Naturally, the experimental results confirm that the most accurate height estimate is reached when the person is static. The results of the experiments corelated with the findings and testing described in [19].

The next testing to be carried out will include UAV device, as a source of the video. It is expected UAV to be in constant height while capturing the video. Change of height may be also possible, thus, expanding the concepts proposed research.

V. Conclusions

In this paper, we developed the experimental software solution based on comparison and selection of the most convenient algorithms. The key features of the solution were to implement a performance friendly, independent, and portable software able to detect moving objects, classify whether they are human, and if so, estimate the height.

The image processing techniques were analyzed. We also taken into account that person might be in motion. Thus, motion detection algorithms, including Mixture of Gaussians, were analyzed. Then, we performed a comparative analysis of human detection algorithms, including the SSD algorithm and YOLO algorithm. Specific approaches were also experimentally verified.

The benefits of the experimental software application include simple calibration, in which the ArUco marker has to be positioned across the surface. Then, the video used for the estimation of the height is recorded in the same manner without any further activity. The experimental solution does not require any special or high-performance hardware. The application of the system is relatively wide since the information extracted from the video can be used in the field of computer vision, for example robotics or user authentication.

Experimental verification evaluated the whole solution and its quality for use in the real environment. The analysis showed that the most accurate height, with deviation only 0.001 m, was estimated at a distance of 5 and 8 m.

The system can be extended by height extraction of other objects, since the deep learning algorithms are able to classify multiple objects even in one image. The high performance and accuracy may be reached using a special hardware, for example Raspberry Pi 4, Jetson Nano or be running on GPU,

TABLE I. TESTING: PERSON STANDS IN FRONT OF THE CAMERA IN VARIOUS DISTANCES.

Distance	5 m		6 m		7 m		8 m		9 m		
Function rule	Height (m)	Dev. (m)	Height (m)	Dev. (m)	Height (m)	Dev. (m)	Height (m)	Dev. (m)	Height (m)	Dev. (m)	Avg. accuracy (%)
exponential	1.843	0.028	2.005	0.190	1.957	0.142	1.871	0.056	1.873	0.058	94.35
quadratic	1.814	0.001	1.890	0.075	1.908	0.093	1.814	0.001	1.830	0.015	97.28
cubic	1.814	0.001	1.840	0.025	1.861	0.046	1.812	0.003	1.821	0.006	98.29
linear	1.819	0.004	1.850	0.035	1.851	0.036	1.813	0.002	1.824	0.009	98.29

TABLE II. TESTING: PERSON WALKS THE HORIZONTAL AXIS.

Distance	5 m		6 m		7 m		8 m		9 m		
Function rule	Height (m)	Dev. (m)	Height (m)	Dev. (m)	Height (m)	Dev. (m)	Height (m)	Dev. (m)	Height (m)	Dev. (m)	Avg. accuracy (%)
exponential	1.832	0.017	1.796	0.019	1.954	0.139	1.838	0.023	1.866	0.051	96.99
quadratic	1.819	0.004	1.784	0.031	1.888	0.073	1.794	0.021	1.758	0.057	99.58
cubic	1.819	0.004	1.783	0.032	1.895	0.080	1.810	0.005	1.751	0.064	99.43

which can reach a real-time height estimation. Considered is also estimation of the shape of human body from refined 3D shape (e.g., using Hierarchical Mesh Deformation) [24]. The outputs of this work will be further used for the estimation of heights of the persons in the newly proposed UAV-based security system. Height of the person [25][26] is to serve as addition indicator for the identification and recognition processes.

Acknowledgment

This publication was realized with support of the Operational Programme Integrated Infra-structure in frame of the project: Intelligent systems for UAV real-time operation and data processing, code ITMS2014+: 313011V422 and co-financed by the European Regional Development Fund.

References

[1] Jain, R.; Kasturi R.; Schunck, B. G. Machine vision. McGraw-Hill, 1995; pp. 140–185, 112–139.

[2] Sehairi K.; Chouireb F.; Meunier J. Comparative study of motion detection methods for video surveillance systems. Journal of Electronic Imaging 2017, 26, doi: 10.1117/1.jei.26.2.023025.

[3] Piccardi. M. Background subtraction techniques: a review. IEEE International Conference on Systems, Man and Cybernetics 2004, doi: 10.1109/icsmc.2004.1400815.

[4] Stauffer C.; Grimson W.E.L. Adaptive background mixture models for real-time tracking. Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1999; pp. 246– 252, doi: 10.1109/cvpr.1999.784637.

[5] Lucas B. D.; Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence, 1981.

[6] Farnebäck G. Two-Frame Motion Estimation Based on Polynomial Expansion. Image Analysis LectureNotes in Computer Science 2003; pp. 363–370. doi: 10.1007/3-540-45103-x_50.

[7] Pan Z.; Jin Y.; Jiang X.; Wu J. An FPGA-Optimized Architecture of Real-time Farneback Optical Flow. IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines 2020; pp. 223-223, doi: 10.1109/FCCM48280.2020.00054.

[8] Dalal N.; Triggs B. Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005; doi: 10.1109/cvpr.2005.177.

[9] Ahmed A. H.; Kpalma K.; Guedi A. O. Human Detection Using HOG-SVM, Mixture of Gaussian and Background Contours Subtraction. 13th International Conference on Signal-Image Technology & Internet-Based Systems 2017; doi: 10.1109/sitis.2017.62.

[10] Girshick R.; Donahue J.; Darrell T.; Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. IEEE Conference on Computer Vision and Pattern Recognition 2014; doi: 10.1109/cvpr.2014.81.

[11] Girshick R. Fast R-CNN. IEEE International Conference on Computer Vision 2015; doi: 10.1109/iccv.2015.169.

[12] Ren S.; He K.; Girshick R.; Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017, 39, pp. 1137–1149, doi: 10.1109/tpami.2016.2577031.

[13] Redmon J.; Divvala S.; Girshick R.; Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. IEEE Conference on Computer Vision and Pattern Recognition 2016; doi: 10.1109/cvpr.2016.91.

[14] Liu W.; Anguelov D.; Erhan D.; Szegedy Ch.; Reed S.; Fu Ch-Y.; Berg A. C. SSD: Single Shot MultiBox Detector. European Conference on Computer Vision 2016, pp. 21– 37.

[15] Zhou T.; Pang X.; Chen W. Improvement of Real Time Detection Algorithm Based on SSD. International Conference on Intelligent Human-Machine Systems and Cybernetics 2018; doi: 10.1109/ihmsc.2018.10189.

[16] Othman N.; Salur M.; Karakose M.; Aydin I. An Embedded Real-Time Object Detection and Measurement of its Size International Conference on Artificial Intelligence and Data Processing 2018; doi: 10.1109/idap.2018.8620812.

[17] Mohd M. Y.; Noor R.; Hasbi H.; Azman A. Stereo vision images processing for real-time object distance and size measurements. International Conference on Computer and Communication Engineering 2012; doi: 10.1109/iccce.2012.6271270.

[18] Dokthurian S.; Pluempitiwiriyawej Ch. Wangsiripitak S. Real-Time Vision Based Human Height Measurement Using Sliding Window on Selected Candidates. IEEE Region 10 International Conference TENCON 2018; doi: 10.1109/tencon.2018.8650380.

[19] Kainz O. Advanced Visual andNon-visual Approaches in Estimating the Parameters of Multi-dimensional Objects of Real-world Scenes. PhD thesis. Technical University of Košice, 2017.

[20] Tonini, A.; Redweik, P.; Painho, M.; Castelli, M. Remote Estimation of Target Height from Unmanned Aerial Vehicle (UAV) Images. Remote Sensing 2020, 12(21); doi: https://doi.org/10.3390/rs12213602.

[21] Yin, F.; Zhou, S. Accurate Estimation of Body Height From a Single Depth Image via a Four-Stage Developing Network. IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020; doi: https://doi.org/10.1109/cvpr42600.2020.00829.

[22] Bieler, D.; Gunel, S. G.; Fua, P.; Rhodin, H. Gravity as a Reference for Estimating a Person's Height From Video. IEEE/CVF International Conference on Computer Vision 2019; doi: https://doi.org/10.1109/iccv.2019.00866.

[23] Leo, M.; Natale, A.; Del-Coco, M.; Carcagnì, P.; Distante, C. Robust Estimation of Object Dimensions and External Defect Detection with a Low-Cost Sensor. Journal of Nondestructive Evaluation 2017, 36(1). doi: https://doi.org/10.1007/s10921-017-0395-7.

[24] Zhu, H.; Zuo, X.; Wang, S.; Cao, X.; Yang, R. Detailed Human Shape Estimation From a Single Image by Hierarchical Mesh Deformation. IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019; doi: https://doi.org/10.1109/cvpr.2019.00462.

[25] Kainz, O.; Jakab, F.; Horecny, M. W.; Cymbalak, D. Estimating the object size from static 2D image. 2015 International Conference and Workshop on Computing and Communication (IEMCON) 2015. https://doi.org/10.1109/iemcon.2015.7344423.

[26] Kainz, O.; Cymbalak, D.; Jakab, F.; Michalko, M. Personal mobile meter for development of Human Body Skeletal Model. 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) 2016. https://doi.org/10.1109/iemcon.2016.7746282.