# Automatic calibration of surveillance video camera

I.A. Valuiskaia[1], E.V. Shalnov[1], A.S. Konushin[1,2]
yana.valuyskaya@graphics.cs.msu.ru|eshalnov@graphics.cs.msu.ru|anton.konushin@graphics.cs.msu.ru
[1]MSU, Faculty of Computational Mathematics and Cybernetics, Moscow, Russia
[2]HSE, Faculty of Computer Science, Moscow, Russia

*This paper is devoted to the automatic calibration of surveillance video camera using objects in the scene. In this paper the problem of estimation of three extrinsic parameters (camera height, tilt angle, roll angle) is considered. The idea of baseline method, proposed in [14], is based on convolutional neural network. As input data, head bounding boxes and the camera focal length are used. In this paper, the modification of the baseline method was proposed, and also methods based on random forest and gradient boosting were studied in order to understand the necessity of using neural networks. An experimental evaluation of the proposed methods on synthetic and TownCenter datasets demonstrated their high efficiency. The best results was shown by the proposed method based on neural networks.*

***Keywords:*** *automatic calibration, surveillance video camera, extrinsic camera parameters, neural networks, random forest, gradient boosting.*

## 1. Introduction

The purpose of camera calibration is to determine the intrinsic and extrinsic calibration parameters of surveillance video camera. The classical formulation of this problem assumes the presence of calibration object with a known geometry in the scene. An example of such a calibration object can be a chessboard, Fig. 1.

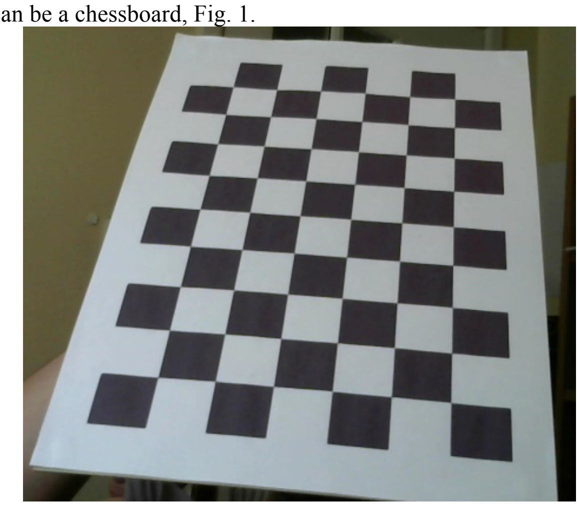

**Figure 1.** Camera calibration using special calibration object.

However, it is difficult to calibrate large amounts of cameras in video surveillance systems using the classical approach because it requires artificially putting a calibration object in the scene. Therefore, for video surveillance systems, automatic calibration, that is, analyzing objects in the scene (people, cars, buildings) is the most promising (Fig. 2).

**Figure 2.** Video surveillance footage.

Camera calibration is necessary to calculate the correspondences between 3D scene points and 2D points on the image plane, which allows you to calculate the distances between objects, track changes in the size of objects, estimate the 3D position of a person on the ground [13], etc.

In addition, information about camera calibration parameters can be used to improve object detection and object tracking algorithms [11], for example, by filtering false detections, which are geometrically incorrect [12]. For example, person height depends on the camera position and orientation, so the information about camera calibration will allow finding regions of interest, where a person of a specific size can be located, which can increase the accuracy of object detection and reduce the computation time. Such filtering can be used alongside with other methods that use some other criteria, for example, estimation temporal consistency of detections [10].

In this paper, the problem of determining the following parameters is considered: camera height above the ground, tilt angle and roll angle (Fig. 3). The baseline method is based on convolutional neural network [14]. In this paper, the modification of the architecture of the baseline neural network will be proposed. In addition, two methods based on other machine learning algorithms will be studied in order to understand the necessity of using neural network to solve the problem of camera calibration. As input data, head bounding boxes and camera focal length are used.

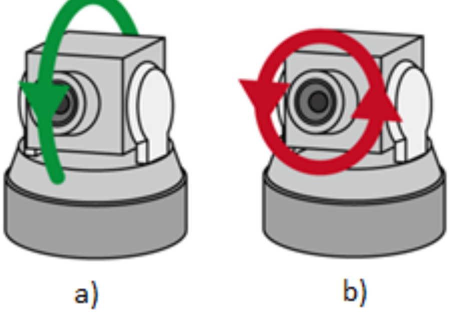

**Figure 3.** a) Tilt angle; b) Roll angle.

## 2. Related work

The existing approaches to the automatic calibration of surveillance video camera can be divided into two groups:

1. Analysis of the lines in the image.
2. Analysis of the objects' size.

The main idea of the first approach is to calculate and then analyze vanishing points and vanishing lines. Vanishing point is a point in the image plane where the projections of a set of parallel lines in space intersect. Each set of parallel lines in space defines its vanishing point. The set of such vanishing points is called a vanishing line, for example, horizon line.

This approach is used in articles [1, 2, 4, 6, 7, 9]. In articles [2, 7, 9] people are used as calibration objects. In the article [2] the method is based on calculating vertical vanishing point, which is specified by the position of feet and head of a person, as well as calculating vanishing point obtained by tracking person in several frames. Authors of the paper [9] consider the problem of calibrating video system, which consists of two cameras. Vanishing points and vanishing lines are computed by RANSAC/EM algorithm [8]. In the article [7], authors use some additional information on the distribution of human heights in the real world. Camera calibration parameters are found by maximizing the log likelihood function under the assumption that 90% of people in the image have height that differs from the average by less than 7.6%.

In [4], cars are used as calibration objects. However, there are some limitations on the scene conditions, which is necessary for calculating vanishing lines: essential part of car trajectory should be a straight line.

In papers [1, 6], buildings are used as calibration objects. In [6], authors proceed from the assumption that there are three orthogonal planes in the scene, so it is possible to calculate three vanishing points. In [1] authors consider the sequence of images of urban environment. Images should be overlapping so that for each pair of adjacent frames it is possible to find some feature point seen in both images.

The second approach is used in [3, 5, 14]. In [3] authors proposed the method for estimating camera calibration parameters for video surveillance system, which consists of several cameras with not overlapping viewing zones. The camera focal length and the tilt angle are obtained by analyzing the distribution of human heights depending on the feet location. However, additional information about person height in real world is necessary to determine the camera height above the ground.

In [14] convolutional neural network is used to estimate calibration parameters using head bounding boxes and camera focal length. For training synthetic data was used.

The main drawback of most methods is instability to changes of the scene or calibration object. The neural network method from [14] is chosen as baseline method, as the most promising and stable to changes of the scene. In this paper, its modification and some other methods based on random forest and gradient boosting will be proposed.

## 3. Proposed method

The input of the algorithm is a set of head bounding boxes $\{R_i\}_{i=1}^{N_j}$, and focal length $f_j$, where $R_i$ is a head bounding box, which is given in image coordinates by three numbers: $R_i = < x_i, y_i, scale_i >$, where $(x_i, y_i)$ is the coordinates of top left corner of bounding box, $scale_i$ is the size of bounding box (assuming that the human head can be enclosed in square); $N_j$ is the number of head bounding boxes for camera position in space $K_j$; for each unique camera position there is a set of bounding boxes.

The output of the algorithm is a triple of parameters characterizing the position of camera in space $K_j$:

$$K_j = < \theta_j, \varphi_j, h_j >,$$

where $\theta_j$ is a tilt angle in radians, $\theta_j \in \left[0, \frac{5\pi}{12}\right]$,

$\varphi_j$ is a roll angle in radians, $\varphi_j \in [-\frac{\pi}{12}, \frac{\pi}{12}]$,

$h_j$ is a camera height above the ground in meters, $h_j \in [0, 20]$.

### 3.1 Baseline method

Baseline method based on convolutional neural network was proposed in [14]. The problem of the lack of training data was solved by constructing a synthetic dataset, which will be described in section 4.1. The general scheme of this algorithm:

1. Preparation of input data.
   At this step, bounding boxes $\{R_i\}_{i=1}^{N_j}$ are grouped by 64 observations for each unique camera position $K_j$ and focal length $f_j$ in matrices $M_j$ of dimension 3x8x8. Within each group, the bounding boxes are sorted by their sizes $scale_i, i = 1, .., 64$ in ascending order. Thus, convolutional layers can analyze both boxes that are close in sizes and far apart, that is, track the changes in the object size depending on its position relative to the camera.
2. Training neural network with the architecture shown in Fig. 4.

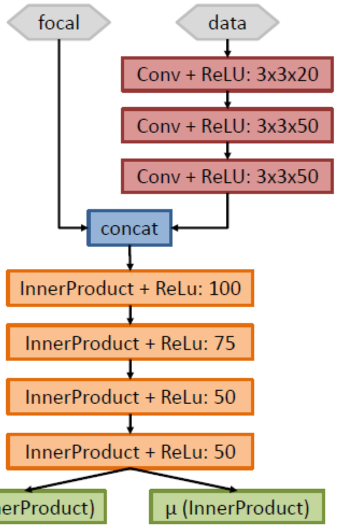

**Figure 4.** Neural network architecture of baseline method.

2.1. Two data input layers: focal length $f_j$ and matrices $M_j$ from step 1.
2.2. Convolutional layer with ReLU activation function and parameters: number of filters is 20; kernel size is 3x3.
2.3. Convolutional layer with ReLU activation function and parameters: number of filters is 50; kernel size is 3x3.
2.4. Convolutional layer with ReLU activation function and parameters: number of filters is 50; kernel size is 3x3.
2.5. Concatenation layer, which concatenates the output of the last convolutional layer and input focal length layer.
2.6. Fully connected inner product layer with ReLU activation function and 100 neurons.
2.7. Fully connected inner product layer with ReLU activation function and 75 neurons.
2.8. Fully connected inner product layer with ReLU activation function and 50 neurons.
2.9. Fully connected inner product layer with ReLU activation function and 50 neurons.
2.10. Two output layers: mathematical expectation of parameters $\mu$ and logarithm of variance of normal distribution $s$.

Neural network was trained by minimizing the following loss function:

$L(y|\mu, s) = -logN(y|\mu, diag(e^s) + \varepsilon)$, where

$y = < t_{gt}, r_{gt}, h_{gt} >$ are ground truth values of parameters (t is for tilt, r is for roll, h is for height);

$\mu = < t_{predicted}, r_{predicted}, h_{predicted} >$ are predicted values of parameters;

$s = < s_t, s_r, s_h >$ are logarithms of variance of normal distribution of parameters, $diag(e^s)$ – diagonal matrix of dimension 3x3; $N()$ – normal distribution density function; $\varepsilon = 10^{-6}$ is used to prevent overfitting to a single train sample.

### 3.2 Modifications of baseline method

Since the input data in the baseline method of automatic calibration of surveillance video camera is a set of head bounding boxes and camera focal length and not the frames of the video sequence itself, the question about using convolutional layers in neural network architecture arises. Despite the sorting of bounding boxes by their sizes during the construction of matrices $M_j$, due to the wide variety of locations (and sizes) of human head in the image, it is not guaranteed that constructed matrices contain local features that could be effectively recognized by the convolutional filters. Thus, a modification with the following architecture of neural network was proposed (Fig. 5):

1. Convolutional layers were removed.
2. Fully connected inner product layer with ReLU activation function and 150 neurons was added.
3. Three output layers: one for each of the estimated parameters.
4. As a loss function, Euclidean distance is used.

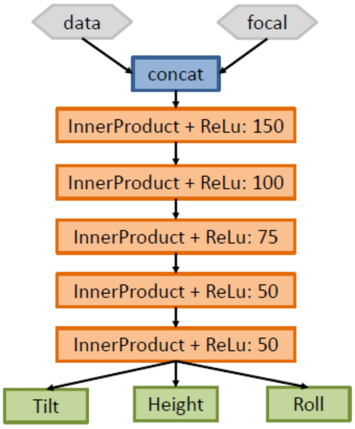

**Figure 5.** Neural network architecture of proposed method.

In addition, the neural network without focal length input layer was trained, because this information is not always available.

### 3.3 Other methods based on other machine learning algorithms

It was decided to develop and evaluate methods based on other machine learning algorithms in order to understand the necessity of using neural networks to solve the problem of automatic camera calibration.

In this paper, two methods were proposed: first method is based on random forest, second method is based on gradient decision tree boosting.

As input data, vector of 193 elements (64 observations of bounding boxes plus focal length) was used.

A method based on random forest: three models were trained, each has 60 tree with a depth of 18).

A method based on gradient decision tree boosting: three models were trained, each consists of 300 decision trees with a depth of 9.

## 4. Results

### 4.1 Datasets

To evaluate proposed methods two datasets were used:

1. Synthetic dataset.
2. TownCenter[1] dataset.

Synthetic dataset was generated using HumanShape[2] and neural network head detector FasterHog[3]:

1. About 300000 samples were generated, each containing 64 observations of human head.
2. Samples were generated for camera calibration parameters if following ranges: $tilt \in [0, 75°]$; $roll \in [-15°, 15°]$; $height \in [0, 20]$.
3. Outliers were artificially added to generated samples to emulate the incorrect detections of human head.
4. 80% of synthetic data were used for training purposes.

An example of the generated synthetic image is shown in Figure 6. A detail description of constructing synthetic dataset can be found in [14].

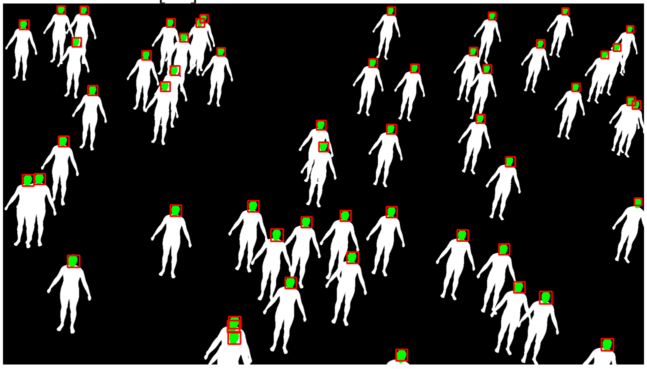

**Figure 6.** Image from the synthetic dataset.

Second dataset contains footage from real video surveillance camera. This TownCenter dataset contains 4500 frames, obtained from a 5-minutes video sequence. About 70000 head bounding boxes are marked on these frames. For the evaluation, 40000 samples were constructed (64 observations each). An example from a TownCenter dataset is shown in Fig. 7.

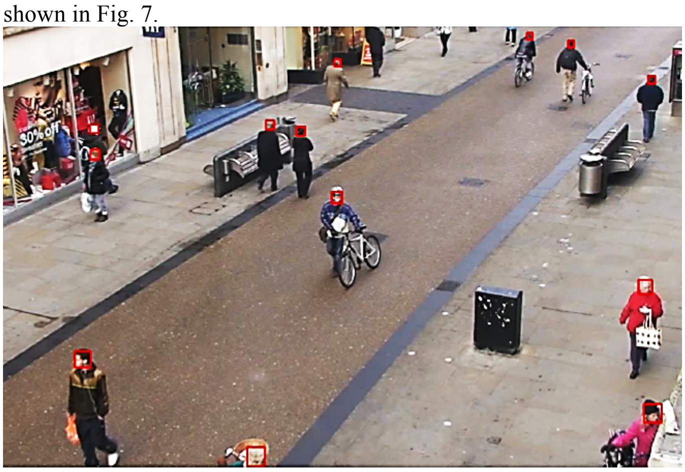

**Figure 7.** Image from TownCenter dataset.

### 4.2 Quality metrics

L1 metric was used to evaluate the quality of developed algorithms.

$$d(label, score) = \frac{1}{N}\sum_{j=1}^{N}|label_j - score_j|,$$

where $label_j$ is a true value of a parameter for the $j$ sample, $score_j$ is a predicted value of a parameter for the $j$ sample, $N$ is a number of samples.

This metric is chosen because it allows easy interpretation of the results.

### 4.3 Experimental evaluation

The results of the experimental evaluation of synthetic dataset are given in Table 1.

All proposed algorithms (except the neural network based method without using focal length information) showed good quality on synthetic dataset in comparison with the baseline one. The best result for the tilt and roll angles was demonstrated by a modified neural network algorithm. The camera height

[1] http://www.robots.ox.ac.uk/~lav/Research/Projects/2009bbenfold_headpose/project.html
[2] https://github.com/e-sha/pyhumanshape
[3] https://github.com/e-sha/fasterhog

above the ground was best determined by method based on gradient decision tree boosting.

The results on TownCenter dataset are given in Tables 2 and 3. Table 2 shows the average error on all test samples. Table 3 shows the error for estimated camera calibration parameters, obtained by averaging over all samples.

On TownCenter dataset proposed neural network algorithm showed an advantage over the baseline one and all other proposed algorithms on all three estimated parameters. The fact that the results on TownCenter dataset are better than on synthetic dataset can be explained by the fact that TownCenter dataset does not contain incorrect head observations.

Experimental evaluation shows the applicability and effectiveness of methods based on machine learning algorithms in automatic calibration of surveillance video camera problem. However, the best result was shown by the modification of baseline neural network method. The proposed neural network method on synthetic dataset showed results that exceed results of the baseline algorithm by about 1.7 times, which confirms the hypothesis from section 3.2 that a wide variety of locations of human head in the image does not allow to use convolutional layers efficiently.

## 5. Conclusion

Compared with the existing methods, the machine learning based approach to solve automatic camera calibration problem has greater resistance to changes in the scene conditions and can be applied to different types of objects. The problem of the lack of training data can be solved using synthetic dataset.

In this article, the modification of the baseline [14] neural network method was proposed. As input data head bounding boxes as long as camera focal length were used. Some modifications of the architecture of the neural network were proposed. The experimental evaluation of the proposed method showed that it is more efficient than the baseline one.

In addition, other methods based on random forest and gradient boosting were proposed in order to evaluate the necessity of using neural network to solve the problem of automatic camera calibration. The experimental evaluation of these methods showed their applicability, however, the best results on both synthetic and TownCenter datasets were shown by the proposed neural network method.

**Table 1.** Average error on synthetic dataset.

| Estimated parameter | Baseline method | Proposed neural network method (with focal length) | Proposed neural network method (without focal length) | Proposed random forest based method | Proposed DT-boosting based method |
|---|---|---|---|---|---|
| Tilt angle, radians | 0.084 | 0.04984 | 0.1096 | 0.0859 | 0.0555 |
| Roll angle, radians | 0.0925 | 0.0553 | 0.0575 | 0.0764 | 0.0586 |
| Camera height, metres | 0.8177 | 0.5226 | 1.1008 | 0.6140 | 0.4904 |

**Table 2.** Average error on TownCenter dataset.

| Estimated parameter | Baseline method | Proposed neural network method (with focal length) | Proposed neural network method (without focal length) | Proposed random forest based method | Proposed DT-boosting based method |
|---|---|---|---|---|---|
| Tilt angle, radians | 0.0196 | 0.0186 | 0.2591 | 0.0668 | 0.0298 |
| Roll angle, radians | 0.0398 | 0.0344 | 0.0487 | 0.0527 | 0.0367 |
| Camera height, metres | 0.6690 | 0.4692 | 0.6597 | 1.1187 | 0.8134 |

**Table 3.** Error for estimated camera calibration parameters on TownCenter dataset.

| Estimated parameter | Baseline method | Proposed neural network method (with focal length) | Proposed neural network method (without focal length) | Proposed random forest based method | Proposed DT-boosting based method |
|---|---|---|---|---|---|
| Tilt angle, radians | 0.0157 | 0.0037 | 0.2600 | 0.0667 | 0.0006 |
| Roll angle, radians | 0.0377 | 0.0014 | 0.0432 | 0.0511 | 0.0315 |
| Camera height, metres | 0.5562 | 0.1935 | 0.5191 | 1.1004 | 0.7091 |

## 6. References

[1] Bobkov V.A., Ronshin Y.I., Kudryashov A.P. Line identification using uncalibrated urban environments (In Russian) //Information Technology and Computer Systems. – 2007. - №.1. – p.63-71.

[2] Chen T. et al. Accurate self-calibration of two cameras by observations of a moving person on a ground plane //Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on. – IEEE, 2007. – p. 129-134.

[3] den Hollander R. J. M. et al. Automatic inference of geometric camera parameters and intercamera topology in uncalibrated disjoint surveillance cameras //SPIE Security+ Defence. – International Society for Optics and Photonics, 2015. – p. 96520D-96520D-15.

[4] Dubská M., Herout A., Sochor J. Automatic Camera Calibration for Traffic Understanding //BMVC. – 2014.

[5] Hoiem D., Efros A. A., Hebert M. Putting objects in perspective //International Journal of Computer Vision. – 2008. – T. 80. – №. 1. – p. 3-15.

[6] Li B. et al. Simultaneous vanishing point detection and camera calibration from single images //International

Symposium on Visual Computing. – Springer Berlin Heidelberg, 2010. – p. 151-160.

[7] Liu J., Collins R. T., Liu Y. Surveillance camera autocalibration based on pedestrian height distributions //Proceedings of the British Machine Vision Conference. – 2011. – p. 144.

[8] Pflugfelder R., Bischof H. Online auto-calibration in man-made worlds //Digital Image Computing: Techniques and Applications, 2005. DICTA'05. Proceedings 2005. – IEEE, 2005. – p. 75-75.

[9] Pflugfelder R., Bischof H. People tracking across two distant self-calibrated cameras //Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on. – IEEE, 2007. – p. 393-398.

[10] Sergeev A. E., Konushin A. S., Konushin V. S. Suppression of false positive detections in video streams of video surveillance systems (In Russian) // Computer Optics. — 2016. — T. 40, № 6. — p. 958–967

[11] Shalnov E. V., Konushin V. S., Konushin A. S. An improvement on an MCMC-based video tracking algorithm //Pattern Recognition and Image Analysis. – 2015. – T. 25. – №. 3. – p. 532-540.

[12] Shalnov E. V., Konushin A. S. Increasing accuracy of detectors using scene geometry (In Russian)// Software product and systems. — 2017. — T. 30, № 1. — p. 106–111.

[13] Shalnov E. V., Gringauz A. D., Konushin A. S. Estimation of the people position in the world coordinate system for video surveillance // Programming and Computer Software. — 2016. — Vol. 42, no. 6. — p. 361–366

[14] Shalnov E. V., Konushin A. S. Convolutional neural network for camera pose estimation from object detections // International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences. – 2017. – T. 42.

## About authors

Anton S. Konushin, PhD, associate professor at the Department of Computing Systems and Automation and head of the Graphics and Media Lab at the Faculty of Computational Mathematics and Cybernetics at Lomonosov Moscow State University, associate professor at the Big Data and Information Retrieval School at the Faculty of Computer Science at Higher School of Economics.
Email anton.konushin@graphics.cs.msu.ru.

Evgeny V. Shalnov, junior researcher at the Department of Computing Systems and Automation at the Faculty of Computational Mathematics and Cybernetics at Lomonosov Moscow State University.
Email eshalnov@graphics.cs.msu.ru.

Iana A. Valuiskaia, graduated from the Faculty of Computational Mathematics and Cybernetics at Lomonosov Moscow State University in 2017 (master degree).
Email yana.valuyskaya@graphics.cs.msu.ru.