CONVOLUTIONAL NEURAL NETWORK FOR CAMERA POSE ESTIMATION FROM OBJECT DETECTIONS

E. V. Shalnov[a], A. S. Konushin[a,b]

[a] MSU, Faculty of Computational Mathematics and Cybernetics, Russia, 119991, Moscow, GSP-1, 1-52, Leninskiye Gory, -
(eshalnov,anton.konushin)@graphics.cs.msu.ru
[b] HSE, Faculty of Computer Science, Russia, 125319, Moscow, 3, Kochnovsky Proezd

Commission II, WG II/5

KEY WORDS: Camera Pose, CNN, Head Detection, Computer Graphics

ABSTRACT:

Known scene geometry and camera calibration parameters give important information to video content analysis systems. In this paper, we propose a novel method for camera pose estimation based on people observation in the input video captured by static camera. As opposed to previous techniques, our method can deal with false positive detections and inaccurate localization results. Specifically, the proposed method does not make any assumption about the utilized object detector and takes it as a parameter. Moreover, we do not require a huge labeled dataset of real data and train on the synthetic data only. We apply the proposed technique for camera pose estimation based on head observations. Our experiments show that the algorithm trained on the synthetic dataset generalizes to real data and is robust to false positive detections.

1. INTRODUCTION

Automatic extraction of useful information from a video is the key computer vision task. Location and attributes of presented objects as well as action they perform are the most interesting parts of such information.

If scene geometry and camera pose are known then these tasks become easier. Indeed, such information restricts available object locations. For example, cars usually can be found on the roads and their size on an image are restricted by the camera location and orientation, i.e. it's pose. On the other hand, modern object detection algorithms assume unknown scene geometry and camera pose. Therefore, they have to search objects of any size at all locations. It leads to false detections and high computation time. If only camera calibration is known we can reduce influence of these factors.

Unfortunately, the most robust methods require interaction with a some template for calibration parameters estimation. It makes calibration of hundreds of thousands of surveillance cameras intractable. In this paper we propose an algorithm that automatically solves this task for surveillance cameras with known focal length. Our algorithm does not require any special calibration templates and directly infer parameters from surveillance video. The constructed algorithm takes object detector results as the input.

We apply supervised machine learning to solve calibration task. Requirement of huge labeled dataset restricts implementation of machine learning techniques in practice. We show how to construct synthetic dataset to solve the calibration task. It allows estimation camera parameters even if there is no real labeled dataset at all.

We show that camera pose can be estimated from people observation in surveillance video. We construct a calibration algorithm that uses head detector results and known camera focal length as the input. As opposed to previous works in the area, we don't assume the "perfect" people detector and implicitly take into account its object localization error and false detections. Our algorithm makes the following assumptions about the observed scene:

1. The camera is static, i.e. it does not change location, view direction and focal length;
2. Camera observes "flat" scene, i.e. ground is a plane;
3. All people stand on the ground plane;
4. All people presented in the scene has the same height (1.75 meters).

The assumption of a single flat ground plane is a standard for the works on surveillance calibration (Liu et al., 2011, Chen et al., 2007, Dubská et al., 2014, den Hollander et al., 2015, Micusik and Pajdla, 2010, Hoiem et al., 2008).

In real surveillance scenario camera produces continuous video stream and detector localizes thousands of objects per minute. In practice the set of detections contains false positives. Therefore, the calibration algorithm should a) work with input of various length and b) be robust to false positive. We show that the proposed calibration algorithm meets these requirements.

Our main contributions are:

1. We propose a technique for construction a synthetic dataset for scene geometry understanding;
2. We construct an algorithm for the camera pose estimation from people observation;
3. The introduced algorithm is shown to be robust to noise in the input data and allows input of any length.

2. RELATED WORK

Automatic surveillance camera pose estimation task has a long history of research (Caprile and Torre, 1990, Li et al., 2010, Liu et al., 2011, Chen et al., 2007, Pflugfelder and Bischof, 2007, den Hollander et al., 2015, Puwein et al., 2012, Dubská et al., 2014, Hoiem et al., 2008). The authors (Puwein et al., 2012) utilizes key points tracking to estimate camera focal length and relative location during rotation and zooming of a PTZ camera. Unfortunately, most of surveillance cameras are static, i.e. they do not change location, view direction and field of view. This limitation significantly reduces information that can be extracted from video.

The paper (Caprile and Torre, 1990) presents relationships between a camera focal length and location of three orthogonal vanishing points (TOVPs). Thus, the most of later works reduces the internal camera calibration task to localization of TOVPs. The authors (Li et al., 2010) presents a method to exploit vanishing points from static scene structures, such as buildings. However, they cannot be applied in scenes without structures. The paper (Dubská et al., 2014) uses motion direction of vehicles to extract horizontal vanishing point. The papers (Chen et al., 2007, Liu et al., 2011, den Hollander et al., 2015) use people observations to estimate vertical vanishing point and horizon line. In addition, this information provides camera orientation in world coordinates. They use known people height to estimate camera location. The authors (den Hollander et al., 2015) assume constant people height (1.8 meters) in the scene and the authors (Liu et al., 2011) use height distribution measured for European populations (Visscher, 2008). The methods (Chen et al., 2007, Liu et al., 2011) use orientation of people foreground blob to estimate vertical vanishing point. The quality of these methods significantly depends on accuracy of extracted masks. Moreover, these works approximates people with a vertical sticks. The proposed approximation is inaccurate especially when tilt angle close to $\frac{\pi}{2}$ (the camera view is an opposite top direction). Thus, we use a detector to estimate head size at different locations of the input image. The authors (Hoiem et al., 2008) also use people detection results for camera pose estimation, but they assume zero roll angle and tilt angle to be close to zero.

3. PROPOSED MODEL

We divide this section into several parts. In subsection 3.1 we propose a technique for synthetic dataset construction. Subsection 3.2 introduces our LogNormLoss layer for CNN that allows learning probabilistic prediction for regression tasks. In subsection 3.3 we propose an algorithm for camera calibration.

3.1 Dataset

The proposed algorithm requires a labeled dataset for training. We found that it is hard to use real surveillance videos for this task. Most of such data does not contain calibration parameters and groundtruth people location. On the other hand, computer graphics allows construction synthetic dataset of an arbitrary size with specified parameters.

We construct synthetic dataset with 100373 scenes. Each scene is a ground plane with people standing on it and camera placed above. Scenes are differ in the intrinsic and extrinsic parameters of the camera and location of captured people. The proposed algorithm uses only head locations in form of bounding boxes and does not need original images. Thus our dataset contains 1) camera calibration parameters; 2) location of people on the ground plane and 3) head detector results. We describe used camera, human models and applied head detector below.

Parameter	Caption	Minimum value	Maximum value
h	height (m)	0	20
t	tilt (rad)	0	$\frac{\pi}{2}$
r	roll (rad)	$-\frac{\pi}{12}$	$\frac{\pi}{12}$
f	focal length (pixels)	0	5000

Table 1. Limits of the camera parameters.

3.1.1 Camera Model Camera calibration contains intrinsic and extrinsic parameters. World coordinate system specifies extrinsic parameters. Thus we choose an unified world coordinate system for all scenes. It specifies ground as a plane $z = 0$. We assume that a camera is placed on the Z axis. Therefore, the height h is the only parameter of a camera location. The view direction of the camera is specified by two angles: tilt t and roll r of the camera.

In the constructed synthetic dataset cameras record FullHD frames (1920×1080). A principal point is assumed to be in the center of captured images and a camera has square pixels with aspect ratio equal to 1. In such assumptions the focal length f (measured in pixels) is the only intrinsic parameter of the camera.

Each scene is parametrized by camera calibration parameters. We sample these parameters from uniform distributions with the boundaries specified in the Table 1.

3.1.2 Human Model The only objects in our dataset are people and we apply human shape model (Pishchulin et al., 2015) to visualize them. Our dataset contains people in standard pose and constant shape. To make the dataset easier all people have the same height (1.75 meters). Thus only location on the ground plane specifies human shape.

We construct at least 200 people in different locations for each scene. We place each person in such a way that the applied detector could find him. In addition, we reject scenes where the detector cannot find people.

3.1.3 Detector We treat detector results as features extracted from a scene. Modern person detectors are sensitive to camera angle and occlusions, therefore it cannot find people in some scenes. But heads are visible in most surveillance scenarios. Thus the proposed features consist of head bounding boxes.

Indeed, human model (Pishchulin et al., 2015) provides the true head location in synthetic data. However, we apply the head detector even to synthetic images. It allows us to avoid modeling of the detector noise and bias in head localization. We assume that these factors are equal on real data and the proposed synthetic datasets. Thus the distributions of features extracted from the real and synthetic data becomes closer.

We use fast implementation (Prisacariu and Reid, 2009) of head detector. It has two significant advantages over the modern detectors: 1) low computation time of the detector allows construction the huge dataset in reasonable amount of time and 2) it finds heads even if we do not model texture of the person skin.

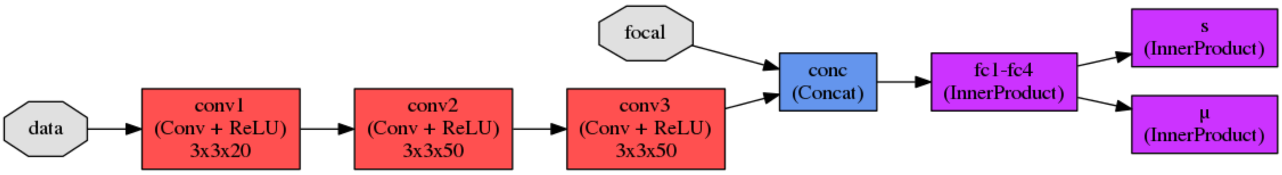

Figure 1. Visualization of the camera calibration network.

3.2 LogNormLoss Layer

We use Convolutional Neural Network (CNN) to estimate the calibration parameters. The Euclidean loss is a traditional loss layer for such regression tasks. It assumes equal ”penalty rules” for all predictions. In some cases it leads to inaccurate results. Imagine, the input data have outliers or huge noise level. A regression with the Euclidean loss tends to bias true predictions to compensate shift from groundtruth on such data. On the contrary, if the used model can indicate how ”good” the input data are, it can overcome this drawback. The proposed LogNormLoss layer is a solution for this task. It estimates the predicted value and confidence of the prediction by minimizing negative logarithm of normal distribution density function.

Formally, LogNormLoss layer assumes that the true value y is normally distributed with unknown mean μ and variance Σ:

$$p(y|x,\Theta) = N(y|\mu,\Sigma) \tag{1}$$

It estimates the parameters μ and Σ using maximization of the likelihood. If we assume that these parameters do not depend on the input data x, the layer describes all targets y with a single Gaussian. On the other hand, CNN with LogNormLoss layer trains μ and Σ as functions of the input data x and model parameters Θ.

The proposed loss layer has 3 inputs. The layer interprets first two inputs μ and s as a mean and logarithm of a variance of normal distribution. Thus, the loss is a negative logarithm of a normal distribution density function:

$$L(y|\mu,s) = -\log N(y|\mu, e^s + \epsilon) \tag{2}$$

We use $\epsilon = 10^{-6}$ to prevent overfitting to a single train sample.

If y is a vector we assume independence of the different components of the prediction:

$$L(y|\mu,s) = -\log N(y|\mu, diag\left(e^s\right) + \epsilon I) \tag{3}$$

The s parameter can be interpreted as a predicted error. The higher value it takes the fewer model confidence is. If the s values are equal for all input data x, this loss equal to Euclidean loss. But if they depend on the observed data, CNN trains to estimate accuracy of the predicted mean value μ for the current input x.

Derivatives of the LogNormLoss layer has simple analytical form:

$$\frac{\partial L(y|\mu,s)}{\partial \mu_j} = \frac{\mu_j - y^j}{e^{s_j} + \epsilon} \tag{4}$$

$$\frac{\partial L(y|\mu,s)}{\partial s_j} = \frac{1}{2}\frac{e^{s_j}}{e^{s_j} + \epsilon}\left(1 - \frac{(\mu_j - y^j)^2}{e^{s_j} + \epsilon}\right) \tag{5}$$

Equations (3), (4) and (5) allows efficient implementation of the layer for modern GPUs. We implement the LogNormLoss layer in the Caffe framework (Jia et al., 2014).

3.3 Calibration Model

Our main goal is construction of calibration algorithm that predicts camera pose from people observations in the scene. It takes the bounding boxes of detected human heads and focal length (in pixels) as the input and predicts camera extrinsic parameters and confidence of the prediction.

We make several assumptions of the observed data. All people in the scene have the same height and stand on the ground plane. Thus all heads lie on a plane in a world coordinates. Therefore, if 3 found heads do not lie in a single line in the image, we can analytically estimate camera extrinsic parameters. Nevertheless, noise and quantization makes this solution inaccurate. Therefore, we construct each input sample from 64 head locations and solve the regression task using CNN.

The first problem we solve is how to present head detection to CNN. Initially the input head locations in a sample do not have any ordering. Hence, there are $64!$ permutations of the same head locations in a sample. If we use data without any ordering the constructed model should adapt to all of these permutations. To solve this problem we sort heads by size and arrange them in a grid. Consequently, the head locations forms a 3D array of size $3 \times 8 \times 8$, where each head bounding box is presented by location of its top left corner and size.

The introduced structure of the sample allows us to use convolutional layers (see Fig. 1). We apply 3 convolutional layers with ReLu non-linearity. Each convolution has size of 3×3. These layers allow 1) extract information from distant objects (correspond to convolution of columns) and 2) be robust to noise in data (convolution adjacent objects in a column).

After the third convolution and ReLu non-linearity we concatenate the constructed features with the camera focal length. The model applies five fully connected layers with non-linearity to this features. The model uses LogNormLoss layer to evaluate quality of the predicted camera location μ and its error s.

It is important to notice, as we use the detected bounding boxes, the proposed algorithm becomes sensitive to the applied detector, i.e. it fits to the detector. Thus, we should update synthetic dataset and repeat training of the CNN, if we want to use another detector. On the other hand, this solution allows us to skip modeling of the detector noise. Moreover, if results of another detector is similar to the ours, it is not necessary to train the model from scratch, the proposed CNN gives a good initialization.

4. TRAINING AND EVALUATION

4.1 Training

We train the calibration model on the constructed synthetic dataset. This dataset contains only groundtruth head detections without

	Tilt angle	Roll angle	Height
groundtruth	0.3497	-0.0251	—
”clear” data	0.3634 ± 0.0149	0.0130 ± 0.0182	8.3290 ± 0.3453
”cluttered” data	0.3237 ± 0.0176	-0.0345 ± 0.0219	7.0696 ± 0.4294
single observation	0.4694 ± 0.0649	-0.0513 ± 0.0402	11.0312 ± 2.1441

Table 2. Predicted camera parameters for TownCentre dataset for ”clear” and ”cluttered” data and sample with a single head. The table presents camera parameters and its predicted standard deviation.

video sequence		Tilt angle	Roll angle	Height
PETS 1	groundtruth	0.105	-0.0172	1.8786
	predicted	0.4458 ± 0.0612	-0.0012 ± 0.0356	4.5958 ± 0.8783
PETS 2	groundtruth	-0.0362	-0.0959	4.6097
	predicted	0.1607 ± 0.0715	-0.0374 ± 0.0428	9.377 ± 1.0811
PETS 3	groundtruth	0.2892	-0.0304	5.5016
	predicted	0.3579 ± 0.0241	-0.0246 ± 0.0225	5.8238 ± 0.2988
PETS 4	groundtruth	0.4582	-0.1095	6.5672
	predicted	0.4539 ± 0.0249	0.059 ± 0.0236	5.3678 ± 0.3262

Table 3. Predicted camera parameters for video sequence of PETS 2006 dataset. The table presents camera parameters and its predicted standard deviation.

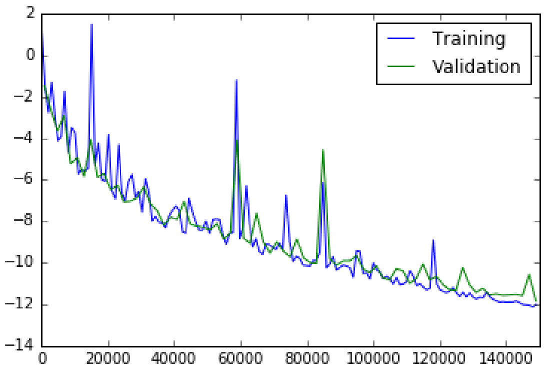

Figure 2. Training and test error on the synthetic dataset.

outliers (false detections). We found that the model trained on this ”clear” detections does not generalize to real data.

Therefore, we add noise to the data. We model two types of noise: 1) duplicated observations of a single object in the same place and 2) false positive detections. We choose at random a subset of found heads to construct a single sample. It may contain less than 64 heads. At the next stage we randomly replace up to 10% of chosen bounding boxes with random noise. Finally, we randomly peek 64 heads from the set and construct training sample. Thus the constructed sample may contain noise and duplicate observations. We peek 3 samples from each generated scene.

Our CNN has small number of parameters and the input is relatively small. Thus each batch contains 32768 samples. We use 80% of scenes for training and 20% for validation. We set gamma to 0.3 and step size to 2000. Training was performed in 150000 iterations.

4.2 Evaluation

We perform several tests to evaluate quality of the constructed model. First of all we test our model on the synthetic validation set. Training process (Fig. 2) shows that error rate on training and validation sets are similar, i.e. the model does not overfits to training data.

We test constructed model on the real data without noise. We choose the TownCentre dataset (Benfold and Reid, 2011) as it contains groundtruth head location for all presented people and the known calibration parameters. Unfortunately, we cannot use height of the camera as scale of the presented world coordinates differs from ours.

We apply the fastHOG detector (Prisacariu and Reid, 2009) to each frame. Detections that overlap with groundtruth is higher than 0.5 for IoU metric are marked as true positives. The detector precision is found to be 48% for this criterion. There are 19061 true positive heads in 4501 frames. To estimate quality on such ”clear” data we choose at random 40000 samples with 64 bounding boxes. The first row of Fig. 3 shows histogram of the model predictions.

To make the final solution we choose a distribution with the smallest differential entropy. For Gaussian distributions it also has the smallest determinant of the predicted covariance matrix Σ. The mean value of the chosen distribution is the predicted location of the camera. We show the chosen distribution in the second row of Fig. 3. Fig. 4 presents synthesized people on the real image from the video sequence. We see that the presented and synthesized people have similar sizes. Thus the proposed model predicts plausible camera location. In further experiments we use the predicted camera height as the groundtruth.

In the next experiment we use all head detections on the TownCentre dataset. We repeat the proposed calibration technique used for clear detections. Note, that in average a number of false positives in the constructed samples are much higher than in train samples (52% vs 10%). The constructed results are shown in the third row of 3. It shows that the predicted camera location is close to its true value even when there are a huge number of false positive detections.

In addition we experiment with duplicate detections in a sample. We choose a single true positive head found by the detector and construct a sample that contains 64 copies of this head. Such an extreme case of duplication corresponds to a scene with a single

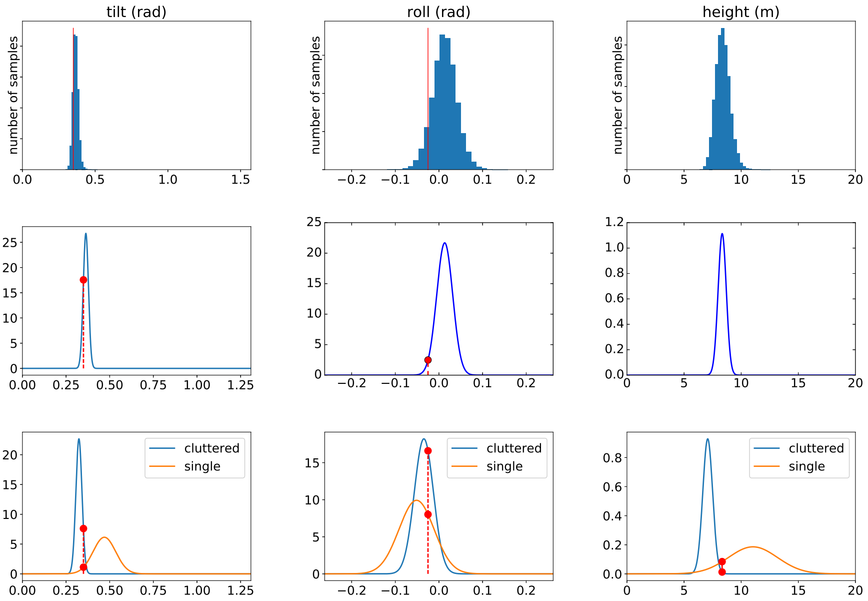

Figure 3. Camera calibration results on the TownCentre dataset. The first row presents histograms of the predicted camera locations from the true positive detections (blue). The second row presents chosen distribution of camera parameters. The third row presents predicted camera location from detections containing false positives (blue) and single true positive head (green). The true camera location is shown in red. Columns correspond to tilt, roll, height parameters.

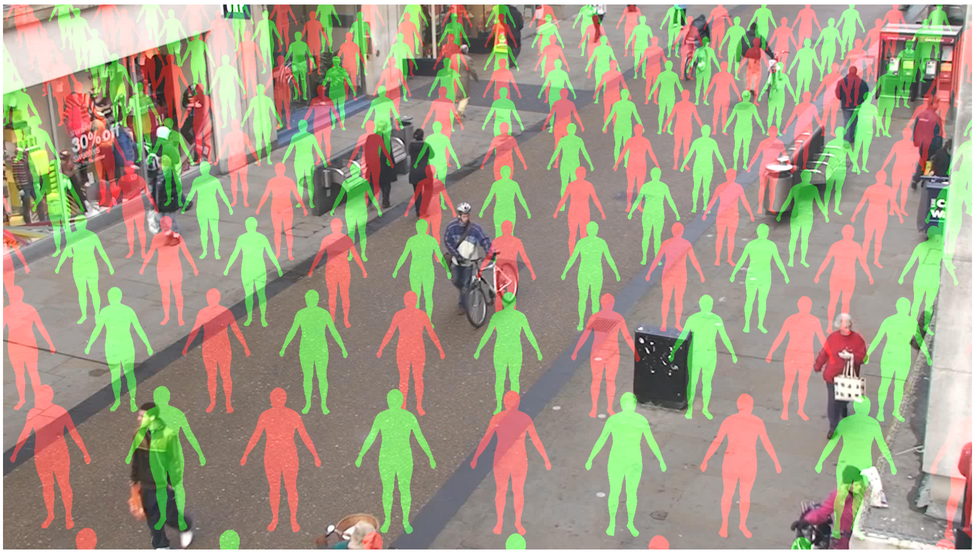

Figure 4. Visualization of people sampled with the predicted camera location.

person standing in the same place for a long time. Camera location cannot be predicted from this sample as is it specifies only distance to the single person in a scene. Fig. 3 shows that the model predicts a very significant error for camera locations that can produce such a detection. Thus a determinant of the predicted covariance matrix is a good measure of the model confidence.

Our training data assumes that people can be found in each location of the input images. Thus, each training sample contains people uniformly distributed in the image plane. Hence, the long input video sequence is preferable as it gives better statistics of people sizes across image plane.

We evaluate the proposed method on four video sequences of the more challenging PETS 2006 dataset (Thirde et al., 2006). It's important to notice, that the first and second video sequences of this dataset violate our assumption of a single ground plane. These video sequences contain people on several floors. Nevertheless, we apply the proposed method to all video sequences in the dataset and use all detector results as the features. Our evaluation (Table 3) reveals that the proposed method correctly estimate camera location on the third and fourth sequence and cannot predict plausible camera pose on the first two sequences. However the predicted deviation is significantly larger for such failure cases, thus the model indicates low confidence in these predictions.

5. CONCLUSIONS

In this paper we present a novel approach to camera pose estimation. It utilizes 3 main concepts: the synthetic training set, intermediate scene representation and prediction of the result error. Our experiments show that in spite of training on synthetic dataset, the constructed algorithms generalize to real data. The proposed algorithm is shown to be robust to noise in the input data and allows input of any length.

In our experiments we use people observation and the head detector (Prisacariu and Reid, 2009) to estimate camera pose. However the proposed approach can be integrated with any kind of objects in the scene, that we can model in synthetic dataset and localize on both real and synthetic data.

The future works include several aspects: (1) integrate camera calibration with detectors to prevent false positives of unlikely sizes (2) speed up the applied detector by skipping image regions where people of the plausible sizes cannot be found. (3) Integrate camera calibration algorithm with detectors of other objects to predict extrinsic and intrinsic parameters.

ACKNOWLEDGEMENTS

This work was supported by Microsoft: Moscow State University Joint Research Center (RPD 1053945) and by Skolkovo Institute of Science and technology, Contract 081-r, appendix A2.

REFERENCES

Benfold, B. and Reid, I., 2011. Stable multi-target tracking in real-time surveillance video. In: *CVPR*, pp. 3457–3464.

Caprile, B. and Torre, V., 1990. Using vanishing points for camera calibration. *International journal of computer vision* 4(2), pp. 127–139.

Chen, T., Del Bimbo, A., Pernici, F. and Serra, G., 2007. Accurate self-calibration of two cameras by observations of a moving person on a ground plane. In: *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, IEEE, pp. 129–134.

den Hollander, R. J., Bouma, H., Baan, J., Eendebak, P. T. and van Rest, J. H., 2015. Automatic inference of geometric camera parameters and intercamera topology in uncalibrated disjoint surveillance cameras. In: *SPIE Security+ Defence*, International Society for Optics and Photonics, pp. 96520D–96520D.

Dubská, M., Herout, A. and Sochor, J., 2014. Automatic camera calibration for traffic understanding. In: *BMVC*.

Hoiem, D., Efros, A. A. and Hebert, M., 2008. Putting objects in perspective. *International Journal of Computer Vision* 80(1), pp. 3–15.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Li, B., Peng, K., Ying, X. and Zha, H., 2010. Simultaneous vanishing point detection and camera calibration from single images. In: *International Symposium on Visual Computing*, Springer, pp. 151–160.

Liu, J., Collins, R. T. and Liu, Y., 2011. Surveillance camera autocalibration based on pedestrian height distributions. In: *Proceedings of the British Machine Vision Conference*, p. 144.

Micusik, B. and Pajdla, T., 2010. Simultaneous surveillance camera calibration and foot-head homology estimation from human detections. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, pp. 1562–1569.

Pflugfelder, R. and Bischof, H., 2007. People tracking across two distant self-calibrated cameras. In: *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, IEEE, pp. 393–398.

Pishchulin, L., Wuhrer, S., Helten, T., Theobalt, C. and Schiele, B., 2015. Building statistical shape spaces for 3d human modeling. *arXiv preprint arXiv:1503.05860*.

Prisacariu, V. and Reid, I., 2009. fasthog-a real-time gpu implementation of hog. *Department of Engineering Science*.

Puwein, J., Ziegler, R., Ballan, L. and Pollefeys, M., 2012. Ptz camera network calibration from moving people in sports broadcasts. In: *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, IEEE, pp. 25–32.

Thirde, D., Li, L. and Ferryman, F., 2006. Overview of the pets2006 challenge. In: *Proc. 9th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006)*, pp. 47–50.

Visscher, P. M., 2008. Sizing up human height variation. *Nature genetics* 40(5), pp. 489–490.

Revised June 2015