2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

CamLoc: Pedestrian Location Estimation through Body Pose Estimation on Smart Cameras

Adrian Cosma
University Politehnica of Bucharest
ioan_adrian.cosma@stud.acs.upb.ro

Ion Emilian Radoi
University Politehnica of Bucharest
emilian.radoi@cs.pub.ro

Valentin Radu
University of Edinburgh
valentin.radu@ed.ac.uk

***Abstract*—Advances in hardware and algorithms are driving the exponential growth of Internet of Things (IoT), with increasingly more pervasive computations being performed near the data generation sources. With this wave of technology, a range of intelligent devices can perform local inferences (activity recognition, fitness monitoring, etc.), which have obvious advantages: reduced inference latency for interactive (real-time) applications and better data privacy by processing user data locally. Video processing can benefit many applications and data labelling systems, although performing this efficiently at the edge of the Internet is not trivial. In this paper, we show that accurate pedestrian location estimation is achievable using deep neural networks on fixed cameras with limited computing resources. Our approach, CamLoc, uses pose estimation from key body points detection to extend pedestrian skeleton when the entire body is not in view (occluded by obstacles or partially outside the frame). Our evaluation dataset contains over 2100 frames from surveillance cameras (including two cameras simultaneously pointing at the same scene from different angles), in 42 different scenarios of activity and occlusion. We make this dataset available together with annotations indicating the exact 2D position of person in frame as ground-truth information. CamLoc achieves good location estimation accuracy in these complex scenarios with high levels of occlusion, matching the performance of state-of-the-art solutions, but using less computing resources and attaining a higher inference throughput.**

***Index Terms*—vision-based localization, smart camera, embedded devices, location estimation, pose estimation**

I. Introduction

The current expansion of Internet of Things (IoT) devices and their advancing capabilities offer a perspective of the trend in computing for years to come. More of the computations previously reserved for server side are being migrated to the edge of the Internet on resource-constrained devices. While this is now possible due to technical advances, other social factors contribute to accelerating this trend such as shifting perception about data privacy [1]. This growing awareness and concern about how personal data can be used is also reflected in new legislation [2], which will put pressure on service providers to move more data processing in user proximity to avoid accidental misuse.

Intelligent systems rely on sensors to perceive their environment for context-aware services. The richest in information sensing modality is vision. A wide range of applications rely on vision for their environment perception (surveillance, robotics, building automation and control, etc.). Knowing the exact location of a person is also very relevant to applications that provide location-based services [4]. However, current vision based methods for pedestrian location estimation require heavy computations, which are commonly performed on server side [3]. Bringing this inference task locally to an edge camera or to its peripheral computation unit is not simple.

Fig. 1: Position estimation in the 2D-space in front of a camera indicated by the lower white dot using bi-box [3] (left) and our system, CamLoc, based on pose estimation (right).

This paper proposes CamLoc, a pose estimation-based localization system building on key body points detection [5], specifically designed to operate efficiently on devices with constrained resources. To improve location estimation in scenarios with occlusion, we extract the pedestrian visual skeleton determined from visible key body points and approximate the location of the feet based on body pose estimation on visible points and standard body proportions. We show that our approach requires less computing resources than the current state-of-the-art in pedestrian detection, bi-box regression [3], for which we calculate the location as provided by the central point on the lower side of the estimated bounding box, as shown in Figure 1. This is due to our careful selection of network architecture, using MobileNet in pose estimation.

We collect a large dataset comprising images from surveillance cameras (over 2100 frames) annotated with the exact location of a pedestrian in the 2D space in front of the camera. This dataset was designed to be particularly challenging for location detection due to occlusions from objects between pedestrian and camera masking portions of the human body (Figure 7), from 42 scenarios. Our evaluation shows that CamLoc performs better in scenarios with higher occlusion compared to bi-box regression [3] and to background subtraction [6]. This dataset is extended with images from a second camera placed at a different angle, showing that our system can be adapted to work with images from multiple concomitant cameras. We make this dataset available to accelerate the

development of location estimation in videos[1].

This paper makes the following contributions:

- We design CamLoc, a vision-based system for pedestrian location estimation in the 2D-space in front of a camera. This brings together multiple computer vision methods, key body points detection, pose estimation on these key points and geometry by extending the human skeleton when some key points are occluded or outside of view.
- To evaluate the performance of our system we collect a substantial dataset annotated with pedestrian positions in front of surveillance cameras. This includes both single-camera and multi-camera scenarios (two cameras aimed simultaneously at the same scene with different view angles).
- We evaluate the performance of CamLoc and its computing requirements in comparison to two other systems (bi-box regression and background subtraction). We show that CamLoc achieves comparable performance to bi-box, while requiring substantially less computing resources (one order of magnitude less memory footprint) and $10\times$ speedup in throughput on embedded devices (Nvidia Jetson TX2 which could be embedded for use in smart cameras) and on a desktop CPU (Intel i7).

The structure of this paper is as follows: the next section presents our view on the necessity of pedestrian location estimation; Section III mentions the more relevant related work; Sections IV and V present the three estimation approaches and our dataset; Section VI presents the evaluation of our proposed solution; and the final section, Section VII, presents our conclusions.

II. Motivation

There is a wide range of scenarios that require accurate location estimation, some of which are highlighted by Mautz in [7]: location-based services in indoor environments, private homes e.g. ambient assisted living (systems providing assistance to elderly people with daily activities in their home), context detection, in museums (visitors interest tracking and study of visitor behavior, location-based user guiding and triggered context-aware information services), logistics and optimizations (it is important to have information about the location of assets and staff members), virtual reality applications (to allow safe entertainment in restrained indoor spaces), and for many other applications.

Most recent solutions for indoor localization operate without purposefully deployed infrastructure, only by using the sensors and WiFi card available on smartphones to estimate user location [8]. The solution we propose here can be viewed as a replacement of mobile localization systems by tracking the movement of people only through the pervasive video surveillance infrastructure. It brings the advantage of not requiring a device to be carried or attached to the person being tracked. However, we believe it can have more impact in conjunction with mobile localization systems. This can improve mobile system performance by providing vision based labels to train a mobile model [4] or as an independent reference contributor to a mobile system for opportunistic calibration [9]. For instance, tracking can be performed on the phone using inertial sensors, wireless and other landmarks in the building [8], [9], tracing continuous estimation from entrance detection [10] and calibrating estimation with higher confidence when user passes in the view of a mounted camera in venues with sparse video coverage. This can be adopted in museums for digital guide, in shopping malls to find offers and at conference venues to find rooms.

On resource-constrained devices, such as smart cameras with local processing, optimisation of resource consumption is desirable. This motivates us to design a system that can operate with low latency and on independent frames to allow control of frame rate as desirable for energy consumption and as needed for interactive applications.

[1]CamLoc Dataset: https://bit.ly/2LzI8JE

III. Related Work

A. Object Detection

One of the first methods to use deep convolutional networks in the context of object detection was R–CNN [11]. This extracts region proposals using semantic segmentation and classify them using a SVM. As such, bounding boxes are generated with an estimated class. The main drawback is latency due to each region being processed independently. Fast R–CNN [12] tries to reduce the execution time and memory usage by implementing region of interest pooling, more specifically Spatial Pyramid Pooling [13] to share computations. Another iteration of this method is Faster R–CNN [14], which uses an ”attention” model through a Region Proposal Netwok. However, even with the optimizations brought by Faster R–CNN, detection is not in real time: Faster R–CNN achieving 5 FPS with VGG net [15]. A faster but slightly less accurate approach is offered by YOLO [16], and its significantly more accurate successor, YOLOv2 [17]. The YOLOv2 network operates in real time, at 67 FPS using a modified GoogLeNet architecture. This defines detection as a regression problem, and predicts bounding boxes and class probabilities in a single evaluation. Part of the larger class of single-stage detectors is RetinaNet [18] and its focal loss function, which significantly increases accuracy. It was designed to lower the loss for well classified cases, while emphasizing hard, misclassified examples. Most of the time, two-stage detectors like Fast–RCNN tend to perform better accuracy–wise than single–stage detectors. This is due to single-stage detectors using a fixed grid of boxes, rather than generated box proposals. A specialized method for pedestrian detection in situations with occlusions is bi-box regression [3], performing a regression to estimate the coordinates of a bounding box over the full human body.

B. Pose Estimation

Early approaches in estimating the pose of people [19], [20], used direct mappings, HOG or SIFT, to build the pose from silhouettes. Nowadays deep learning approaches have

been adopted being trained on large datasets [21]–[23]. One of the most successful proposed approaches is DeepCut [24], which initially detects people in the scene, and using a convolutional neural network, hypothesizes body parts to be reduced with non-maximum suppression in subsequent steps. An improvement to this is DeeperCut [25], which improves body part detection. Another approach [26] uses a processing pipeline to first detect people in images and then estimate their pose. If the detector confidence is slim, pose estimation is skipped. Key points are predicted using heatmap regression with a fully convolutional ResNet. The system is trained only on COCO data, achieving state of the art results at the time. Tome et al., [5] propose an approach to detect 3D human pose. This method uses a 6-stage processing pipeline to "lift" 2D poses, using a combination of belief maps provided by convolutional 2D joint predictors and projected pose belief maps. We use this method for the pose estimation component of CamLoc due to its performance.

C. Vision-based Indoor Localization

Although the majority of the existing indoor localization solutions are considered from a mobile system's perspective (using smartphone sensors and WiFi to estimate the location), there has also been research into positioning systems by means of computer vision. The advantage of these systems is that users are not required to carry special tags, as in many circumstances wearing a tag may not be viable (e.g. Ambient Assisted Living scenarios where the typical user is not well-versed with technology [27]).

Mautz et al., have a survey on optical indoor positioning systems [28]. Their work describes different systems and classifies them based on the reference used to determine the location of users in a scene such as images, projected patterns and coded markers. Tsai et al., propose a solution to extract foreground objects using a background model [29]. Several existing systems use RGB-D sensor for human positioning, such as the systems proposed by Munaro et. al., [30] and Saputra et. al., [31] that offer scalable multi-camera solutions for people tracking. Duque et al., [32] present a system for people localization in complex indoor environments by combining WiFi positioning systems with depth maps. Viola et al., [33] propose a system that detects and identifies people, even if occluded by others, using an algorithm for creating free-viewpoint video of interacting people using hand-held Kinect cameras. Nakano et al., [34] present potential applications for their proposed Kinect Positioning System, an indoor positioning system that uses Kinect without an infrared photophore.

Most of the positioning systems by means of computer vision use depth cameras, which cannot be considered part of the widely available infrastructure in most buildings. The localization solution that we propose in this work makes use of typical surveillance cameras, that many buildings are already equipped with.

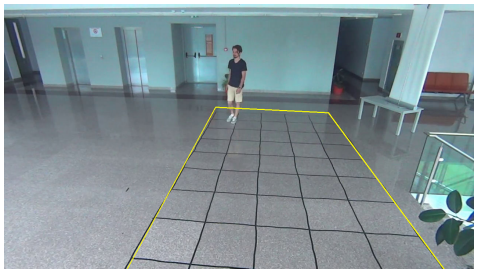

Fig. 2: Grid with homography points defined. Image has lens distortion corrected. Capture taken from S1_Wide.

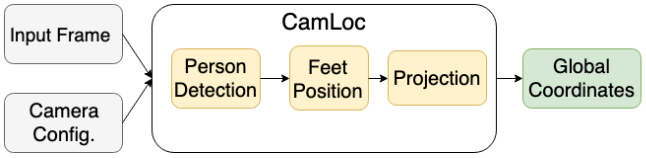

Fig. 3: Block diagram of the CamLoc system.

IV. Method

Given a camera with a view of the floor, localization can be performed by using a homography transformation from the position of the feet from the camera perspective to the floor plane (perspective-to-plane transformation). This method is based on the 2D direct linear transformation, developed by Abdel-Aziz and Karara [35]. It assumes that a set of 4 points must be defined a priori for a particular camera, as shown in Figure 2. The homography transformation is based on the following formulae, given the camera perspective coordinates x_c and y_c:

$$X_{floor} = \frac{ax_c + by_c + c}{gx_c + hy_c + 1} \quad Y_{floor} = \frac{dx_c + ey_c + f}{gx_c + hy_c + 1}$$

The parameters a, b, c, d, e, f, g, h can be calculated by transforming the equations in matrix format, given the set of camera points $\{(x_i, y_i) \mid i = \overline{0,4}\}$, and a predefined set of map points $\{(X_i, Y_i) \mid i = \overline{0,4}\} = \{(0,0), (0,1), (1,0), (1,1)\}$:

The next step is to estimate the position of the feet of a person in an occluded environment. For this, we consider three methods: background subtraction, bi-box regression and pose estimation. The general architecture of our system is presented in Figure 3. The pipeline takes a single frame and a camera configuration (homography transformation, lens information, camera position in the building) and using the feet positions detected by pose estimation with body extension, it outputs the global floor coordinates using the homography transformation. The pose estimation with body extension method was chosen over background subtraction and bi-box regression due to its superior performance.

A. Background Subtraction

Background subtraction is based on a Gaussian Mixture Model developed by Kaewtrakulpong et al. [36] for background modelling. As for all background subtraction methods,

the foreground mask is computed by subracting the current frame from the background model. While this masks static parts of the scene (and more generally, everything that can be considered background), its shortcomings are evident in the case of shadows or sudden change in illumination. As we are interested in detecting the moving person in the frames, the method fails when having to deal with other moving objects in the scene and it also cannot handle occlusions.

Figure 4 shows our application of this method. A bounding rectangle is created over the ”moving” parts of the image, assuming the presence of a person. The feet position is calculated as the middle point of the lower segment of the bounding rectangle.

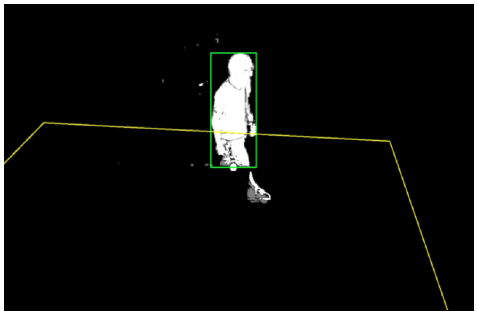

Fig. 4: Background subtraction method. The presence of occlusions in the bottom part of the person makes this approach unreliable in estimating the feet position.

B. Bi-box Regression

The bi-box [3] regression approach performs both pedestrian detection and occlusion estimation as it generates two bounding boxes: one for the full human body and the other for the visible part of the body. Even though the original authors used the VGG-16 [15] neural architecture for the backbone of the network, we used the ResNet-50 [37] architecture as it is faster and more accurate.

Detection of bounding-boxes of visible and occluded pedestrians is facilitated by the CrowdHuman Dataset [38] and Caltech Pedestrians Dataset [39], in which annotations account for occlusions. We used a pre-trained model on ImageNet [40], and fine-tuned on the CrowdHuman Dataset, using focal loss [18].

The feet position is calculated similarly to background subtraction, given the estimated bounding box.

C. Pose Estimation

The occurrence of occlusions in front of the pedestrian body motivated the use of pose information for inference. Since modern pose estimators ([5], [19]) are able to detect subsets of body parts, this information can be used to extend the person's body in the occluded area based on known body proportions [41]. This leads to an accurate estimated feet position.

Pose estimation neural architectures first detect person joints and through belief maps connect them to form body parts. Tome et al. [5] uses a multi-stage convolutional neural network to output the pose information of a person. The network takes in a rescaled 224x224 frame and outputs the human skeleton. As such, the method for extending the human body and estimating the feet position is the following:

- If the feet are found in the detections, take the point between them.
- Else, perform linear regression on the midpoint between complementary body parts (i.e. right/left shoulder, right/left hip) and extend onto the regressed line accordingly, considering the detected joints (e.g. extend the body starting from the lowest joint detected).

Body extension is done on the regressed line from the detected joints, to account for natural body position (e.g. leaning against an object) and for possible lens distortions. However, when insufficient joints are detected, regression cannot be performed and the estimation is cancelled. The percentages of cancelled estimations are shown in Table IV.

Scene Name	# scenarios	# frames
S1_Wide	33	1929
S2_Narrow	9	267
Total	42	2196

TABLE I: Number of frames and scenarios in each scene.

Fig. 5: Scenes and camera perspectives.

V. Pedestrian Location Estimation Dataset

Location detection is performed by first getting the video frames from cameras, running them through the deep learning model (such as bi-box regression or pose estimation), post-processing (e.g. extending the body, estimating the feet), and then computing global coordinates.

To address the problem of vision-based location estimation, we start by collecting a sizeable dataset of video images annotated with the exact location of a person moving in a 2D space in front of the camera. The collected dataset captures a single person in 2 different scenes from a total of 3 cameras. One of the scenes offers multiple points of view, from 2 cameras simultaneously. A total of 42 scenarios, with varying levels of occlusion, are investigated across the 2 scenes, as also shown in Table I. Each scene has an artificial grid drawn on the floor, which is used for validation. Global positioning is given relative to the origin of the grid.

A. Scenes Description

The two scenes can be seen in Figure 5. Scenarios include obstacles at different distances and varying clothing.

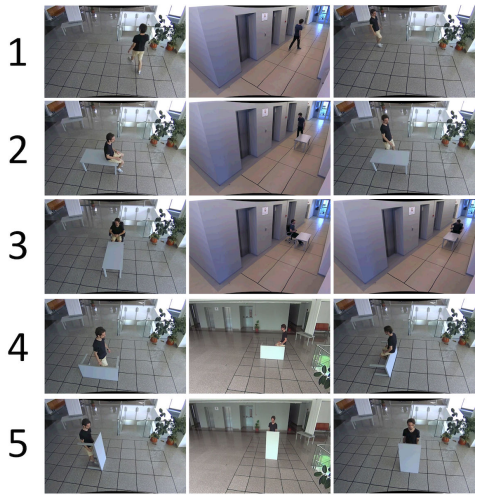

Fig. 6: Sample images from each scenario type (1 to 5). Not all scenes contain every type of scenario.

Scenario Type	Description
1. Baseline	No occlusions. This is the best case scenario
2. Table	A simple table, used for testing localization when the person is sitting.
3. Table and Chair	A more complex variant of the previous scenario, where the feet are not always visible.
4. Table Sideways	Used for occluding the lower part of the body.
5. Table Standing	Occluding most of the person, except the upper part of the body.

TABLE II: Descriptions of scenario types across the scenes.

1) Scene 1: (Wide Space) S1_Wide represents a wide open-space such as a wide hallway, a lobby or a large room. Two camera perspectives are available at a perpendicular angle. These can be seen in Figure 5 in images *(a)* and *(b)*. The two cameras are positioned at 2.8 metres and 1.8 metres, respectively, from the ground. The grid is a 540 cm x 300 cm rectangle, evenly divided into squares of 60 cm in length.

2) Scene 2: (Narrow Space) S2_Narrow represents a narrow space, a typical hallway. The space reaches over 10 metres from the camera. The camera is at 2.5 metres from the ground, and the grid is a 225 cm x 1000 cm rectangle, divided into 75 cm x 90 cm rectangles.

B. Occlusions and Obstacles

The scenarios captured by the dataset can be grouped in 5 broad categories, described in Table II. Situations with various levels of occlusions were considered, which could arise in real life scenarios. These include a person standing upright, sitting and with various body parts occluded by obstacles. Sample images from each type of scenario are shown in Figure 6. In some extreme cases, the body is almost completely covered (see scenario type 5. Table Standing), raising problems for vision-based positioning algorithms.

C. Data Annotation

For the S1_Wide scene, the dataset offers the perspectives of two synchronized cameras. In this case, the ground truth annotation represents a combination of the annotations for the two camera perspectives: when the person is not visible on one of the cameras, the ground truth from the other camera represents the shared position. This approach is useful for situations when tracking the movement of people across video frames, including moving outside the coverage of one of the cameras. Otherwise, the position of the person is given by the midpoint between the annotations of the two perspectives. Figure 7 shows the annotation process.

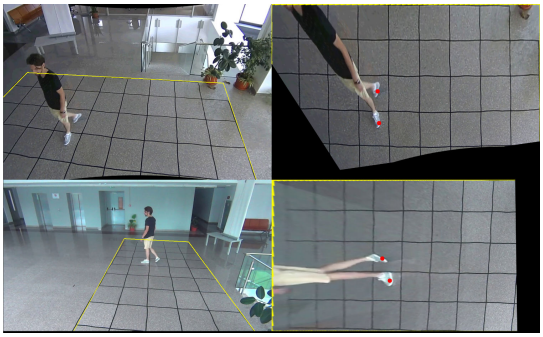

Fig. 7: Capture from the annotation tool, in the multi-camera scenario. The annotation in this scenario combines both camera views. The red circles are the annotations on the transformed grid; the locations of both feet are marked and then averaged to get the person's location in one frame.

Separately annotating two frames from different perspectives leads to different global coordinates. This is due to the differences in the set of points that define the homography and differences in synchronization. In the case of the S1_Wide scene, which benefits from two camera perspectives, the localization mismatch level is low, the average localization mismatch for each axis being less than 20 cm even when the distance between the person being tracked and the camera is over 6 metres.

D. Dataset processing

The videos from the surveillance cameras were preprocessed to remove the barrel lens distortion. This was achieved with the use of a Linux's *ffmpeg* command-line tool, *defish0r*, which automatically corrects distortion, at the price of losing some information at the edges of the frame.

The frames were not scaled to a predefined set of dimensions. Instead, a configuration file is present for each scene, with the following information:

- image height and width
- camera height and X,Y coordinates, with their respective units of measurement
- grid height and width, with units of measurement
- the set of points to define the homography transformation

Generally, the origin of the plane coordinate system is the lower left corner of the grid, as viewed by the camera. This is not the case for the multi-camera scenes, where the origin was chosen to be the same for both cameras.

The dataset is offered as a set of frames from the gathered videos, with absolute X,Y coordinates annotations for each frame organized in *.csv* files.

The dataset was collected in a realistic environment from surveillance cameras [42] in an office building. It contains 2196 frames, and their distribution on each scene is shown in Table I.

VI. Evaluation

Evaluation is performed by analyzing the errors in localization with respect to the ground truth annotations. Error is calculated as the squared error between the global ground truth coordinates $p_{gt} = (x_{gt}, y_{gt})$ and the predicted coordinates $p_p = (x_p, y_p)$.

Considering that the predictors (object detectors / pose estimators) might not offer confident enough predictions for every annotated frame, the percentage of missing predictions is also taken into account.

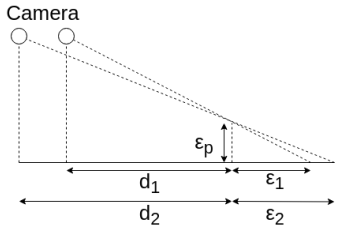

Fig. 8: Projection error with varying distance from camera

Due to the way localization is performed, by projecting a detection from the camera perspective onto the floor, the localization error should have a positive correlation to the distance between the person and the camera. Using the properties of similar triangles, as shown in Figure 8, maintaining the same camera height and varying the distance, leads to the following assertion:

$$\frac{\epsilon_2 - \epsilon_1}{\epsilon_1} = \frac{d_2 - d_1}{d_1}$$

This indicates that localization error is generally proportional to pedestrian distance to camera. The error is also dependent on the predictor feet accuracy ϵ_p from the camera perspective.

A. Detection in Single Images

Pose estimation was performed using the technique described by Tome et al. in [5]. The backbone architecture used is MobileNet [43], which was chosen for its good trade-off between speed and accuracy. It makes use of depthwise separable convolutions for faster inference times. Lightweight neural architectures such as this one are becoming prevalent in the space of mobile applications. The network was trained on the COCO dataset [44].

Compared to a bounding rectangle, pose information offers a more accurate estimation of the feet position, especially in cases with occlusions. The body position can be inferred from just a few detected body parts by using known body proportions. This is invariant to the camera position, since that information is contained in the estimated body proportion.

Scene	Mean error (cm) **background**	Mean error (cm) **bi-box**	Mean error (cm) **CamLoc**
Baseline	88.9	59.4	**37.7**
Table	83.1	**38.8**	49.1
Table and Chair	87.4	81.6	**58.8**
Table Sideways	48.9	40.1	**38.4**
Table Standing	74.5	52.8	**44.1**

TABLE III: Mean error value for background, bi-box and CamLoc, using three cameras in two scenes.

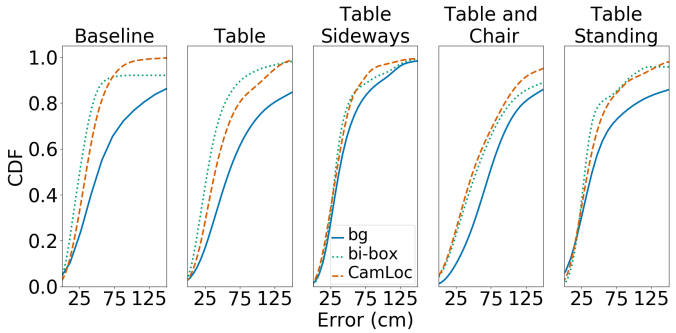

Fig. 9: Error cumulative distribution functions for each scenario type (as presented in Table II).

Table III shows descriptive statistics of each scene, analyzed from a single camera perspective. In most situations, CamLoc has lower errors compared to background and bi-box. Error cumulative distribution functions for each of the scenario types are presented in Figure 9.

In Figure 9 it can be observed that in the case of all methods, the position error is the lowest in the *Baseline* scenario (scenario in the first row of images of Figure 6), where no occlusions occur. In this scenario, bi-box shows a slightly better performance than CamLoc. This is also the case in the *Table* scenario (scenario in the second row of images of Figure 6) where the occlusions of the person are still minimal. However, when the occlusions are more significant (scenarios in the third, fourth and fifth rows of Figure 6), CamLoc is outperforming bi-box. In figure 9 it also be noticed that in most cases, the lowest localization errors are obtained by Cam1 in the S1_Wide scene, most probably due to the position of the camera closer to the monitored scene.

B. Performance in Multi-Camera

Considering the S1_Wide scene, where positioning can be inferred from two different cameras at the same time, localization could be improved by merging locations from both cameras using distance-weighted averaging:

$$P(p_1, p_2, d_1, d_2) = \begin{cases} p_1 & p_2 \in \emptyset \\ p_2 & p_1 \in \emptyset \\ \dfrac{d_2 p_1 + d_1 p_2}{d_1 + d_2} & \text{otherwise} \end{cases}$$

	Missing predictions (%) **background**	Missing predictions (%) **bi-box**	Missing predictions (%) **CamLoc**
Cam1+Cam4	**3.1**	**2.8**	**0.3**
Cam1	5.0	18.5	9.1
Cam4	9.4	16.7	4.4

TABLE IV: Performance results of multi-camera compared to individual cameras.

The function P takes into account the positions from both cameras and the distances to the camera. As such, when one of the cameras misses the prediction for a frame, the other camera supplies the position. If both cameras have inferred a position for the current frame, a weighted average of the two is computed using the inverse of their respective distances. This way, the position provided by the camera that is further away is penalized. This is motivated by the fact that localization errors increase with the distance from camera.

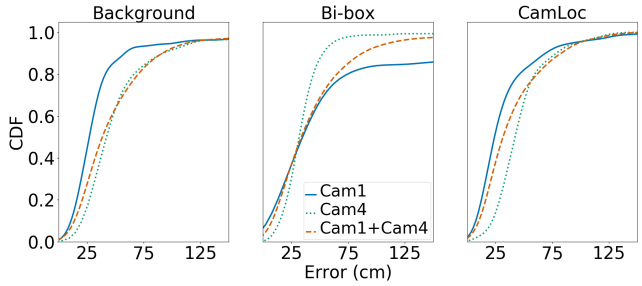

Fig. 10: CDF: multi-camera compared to individual cameras.

The performance of the multi-camera approach is presented in both the CDF of Figure 10 and in Table IV. Since the procedure takes into account distances from two cameras, it can be noticed that it reduces the errors of the worse performing camera. An important benefit of having multiple cameras is the significant improvement of the prediction ratios (less missed predictions) as shown in Table IV.

C. Computing Resources Footprint

We assess the performance of the three chosen vision-based location estimation methods on two devices: an Intel i7-4500U (assuming near to camera computations on a small computer) and the Jetson TX-2 (a common device for embedded computing). The Jetson TX2 is a development platform with one integrated 256-core NVIDIA Pascal GPU, 8GB of memory and a quad-core ARM Cortex-A57 CPU. While the Intel i7-4500U is designed for the mobile computing space, with 2 cores at 1.8 GHz and 4 MB cache memory.

Table V shows the inference time with a batch size of one (one image at a time) and memory footprint, achievable on the actual hardware. The difference in memory footprint between Intel and Jetson TX2 is due to the internal libraries used for convolutional computations by each of the two devices, Jetson relying on cuDNN, a highly optimized computation library for NVidia GPUs, maximizing speed in detriment to memory footprint, while the Intel suite uses the MKL-DNN library, also a highly optimized computation library. Both background and CamLoc are more favorable in terms of memory footprint and inference time (throughput) to bi-box.

Device	**Memory (MB)**	**Inference time (sec)**	**Performance (FPS)**
Intel i7-4500U (bg)	197	0.023	43.47
Intel i7-4500U (bi-box)	2825	7.20	0.13
Intel i7-4500U (CamLoc)	287	0.52	1.92
Jetson TX2 (bg)	178	0.05	20
Jetson TX2 (bi-box)	1970	1.86	0.53
Jetson TX2 (CamLoc)	620	0.16	6.25

TABLE V: Performance of background, bi-box and CamLoc on NVidia Jetson TX2 and Intel i7-4500U.

CamLoc on the Jetson TX2 achieves a frame rate of just above 6 frames per second. This is close to real-time performance, which is useful to many applications that require quick location estimation for interactive services.

In real-world scenarios, the majority of surveillance cameras within buildings have significant time periods without any movement or people in view. To skip over these irrelevant frames where no movement is registered, CamLoc is extended to run a preliminary event recognition step before performing the entire estimation process through the neural network (just to determine if anything has changed from previous frames). This is done at the pixel level by observing radical changes between frames. In the case of frames that do not include a person, this initial person detection step improves the energy efficiency of CamLoc up to a factor of 10, based on observed activity in front of the camera. Data from an office building shows that for a typical week, the surveillance camera pointed at the most populated area of the building (the ground floor hallway) has a person in sight approximately 55% of the time, while a surveillance camera on the hallway of an upper floor has a person in sight only approximately 1% of the time. Based on these observations, the initial event detection step brings significant energy savings to CamLoc, reducing energy consumption by 1.5× and 9× respectively.

VII. Conclusions

The trend of performing more computations on IoT devices for edge intelligence is likely to expand over the coming years, with computer vision being adopted more outside of the cloud. In this paper, we show that adopting computer vision techniques (action detection, human pose estimation and projections) our CamLoc system can perform efficient location estimation on single images from a fixed camera. Although achieving similar accuracy to current state-of-the-art for location detection, bi-box, CamLoc requires significantly less memory (one order of magnitude less) with a superior throughput (10× speedup). Our results show that computer vision systems such as CamLoc can operate efficiently on embedded devices, opening the opportunity for complex interactive applications driven by smart cameras in user proximity.

ACKNOWLEDGEMENT

This work was supported by a grant of Romanian Ministry of Research and Innovation, CCCDI-UEFISCDI, project number PN-III-P1-1.2-PCCDI- 2017-0272/17PCCDI-2018, within PNCDI III.

REFERENCES

[1] Jennifer Golbeck and Matthew Louis Mauriello. User perception of facebook app data access: A comparison of methods and privacy concerns. *Future Internet*, 8, 2016.

[2] EU Commission. 2018 reform of eu data protection rules. https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en, 2018. [Online; accessed 20-December-2018].

[3] Chunluan Zhou and Junsong Yuan. Bi-box regression for pedestrian detection and occlusion estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[4] Valentin Radu and Maximilian Henne. Vision2sensor: Knowledge transfer across sensing modalities for human activity recognition. *ACM IMWUT*, 3(3), 2019.

[5] Denis Tome, Christopher Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *CVPR 2017 Proceedings*, pages 2500–2509, 2017.

[6] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *ECCV*. Springer, 2000.

[7] Rainer Mautz. Indoor positioning technologies. 2012.

[8] Valentin Radu and Mahesh K Marina. Himloc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced wifi fingerprinting. In *Proc. IPIN*. IEEE, 2013.

[9] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of MobiSys*. ACM, 2012.

[10] Valentin Radu, Panagiota Katsikouli, Rik Sarkar, and Mahesh K Marina. Poster: Am I indoor or outdoor? In *Proc. MobiCom*. ACM, 2014.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[12] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proc. ECCV*. Springer, 2014.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[17] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[18] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE TPAMI*, 2018.

[19] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. *IEEE TPAMI*, 28(1), 2006.

[20] Ahmed Elgammal and Chan-Su Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*. IEEE, 2004.

[21] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proc. CVPR*, 2014.

[22] U Iqbal, A Milan, M Andriluka, E Ensafutdinov, L Pishchulin, J Gall, and SB PoseTrack. A benchmark for human pose estimation and tracking. *arXiv preprint arXiv:1710.10000*, 2(3):4, 2017.

[23] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2014.

[24] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proc. CVPR*, 2016.

[25] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.

[26] Chenshu Wu, Zheng Yang, Yunhao Liu, and Wei Xi. Will: Wireless indoor localization without site survey. *IEEE Transactions on Parallel and Distributed Systems*, 24(4):839–848, 2013.

[27] Andreas Braun and Tim Dutz. Low-cost indoor localization using cameras–evaluating ambitrack and its applications in ambient assisted living. *Journal of Ambient Intelligence and Smart Environments*, 8(3):243–258, 2016.

[28] Rainer Mautz and Sebastian Tilch. Survey of optical indoor positioning systems. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–7. IEEE, 2011.

[29] Tsung-Han Tsai, Chih-Hao Chang, and Shih-Wei Chen. Vision based indoor positioning for intelligent buildings. In *Proc. Intelligent Green Building and Smart Grid (IGBSG)*. IEEE, 2016.

[30] Matteo Munaro, Filippo Basso, and Emanuele Menegatti. Openptrack: Open source multi-camera calibration and people tracking for rgb-d camera networks. *Robotics and Autonomous Systems*, 75:525–538, 2016.

[31] Muhamad Risqi Utama Saputra, Guntur Dharma Putra, Paulus Insap Santosa, et al. Indoor human tracking application using multiple depth-cameras. In *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on*, pages 307–312. IEEE, 2012.

[32] Jaime Duque Domingo, Carlos Cerrada, Enrique Valero, and Jose A Cerrada. An improved indoor positioning system using rgb-d cameras and wireless networks for use in complex environments. *Sensors*, 17(10):2391, 2017.

[33] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[34] Yoshiaki Nakano, Katsunobu Izutsu, Kiyoshi Tajitsu, Katsutoshi Kai, and Takeo Tatsumi. Kinect positioning system (kps) and its potential applications. In *International Conference on Indoor Positioning and Indoor Navigation*, volume 13, page 15th, 2012.

[35] Karara H. M. Abdel-Aziz, Y. I. Direct linear transformation into object space coordinates in close-range photogrammetry. 1971.

[36] Pakorn KaewTrakulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. 2002.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[38] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[39] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[41] G. Livshits, A. Roset, K. Yakovenko, S. Trofimov, and E. Kobyliansky. Genetics of human body size and shape: body proportions and indices. *Annals of Human Biology*, 29(3):271–289, 2002.

[42] VIVOTEK Inc. *VIVOTEK FD816B-HT Fixed Dome Camera*.

[43] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[44] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014.