A Distributed Tracking Algorithm for Counting People in Video by Head Detection*

Denis Kuplyakov[1,2][0000−0002−2957−3297], Yaroslav Geraskin[1][0000−0002−2764−272X], Timur Mamedov[1,2][0000−0001−6554−7988], and Anton Konushin[1,3][0000−0002−6152−0021]

[1] Faculty of Computational Mathematics and Cybernetics, Moscow State University, Moscow, Russia
[2] Video Analysis Technologies, Moscow, Russia
[3] NRU Higher School of Economics, Moscow, Russia
{denis.kuplyakov,yaroslav.geraskin, timur.mamedov,anton.konushin}@graphics.cs.msu.ru

Abstract. We consider the problem of people counting in video surveillance. This is one of the most popular tasks in video analysis, because this data can be used for predictive analytics and improvement of customer services, traffic control, etc. Our method is based on the object tracking in video with low framerate. We use the algorithm from [1] as a baseline and propose several modifications that improve the quality of people counting. One of the main modifications is to use a head detector instead of a body detector in the tracking pipeline. Head tracking is proved to be more robust and accurate as the heads are less susceptible to occlusions. To find the intersection of a person with a signal line, we either raise the signal lines to the level of the heads or perform a regression of bodies based on the available head detections. Our experimental evaluation has demonstrated that the modified algorithm surpasses original in both accuracy and computational efficiency, showing a lower counting error on a lower detection frequency.

Keywords: Computer Vision, Video Analytics, Tracking, People Counting.

1 Introduction

Counting people passing through certain zones of a public infrastructure, such as pedestrian crossings, sidewalks, squares, etc., is a practically important task. There are many solutions to this problem, one of them is object tracking. The task of object tracking is to create tracks for each person. A track unambiguously corresponds to a person. It marks this particular person locations in all frames in which he or she is visible. In order to count people a signal line is usually specified in the frame (see fig. 1). If the track crosses the signal line, we can say with confidence that the person also crossed it.

* Publication is supported by RFBR grant 18-08-01484

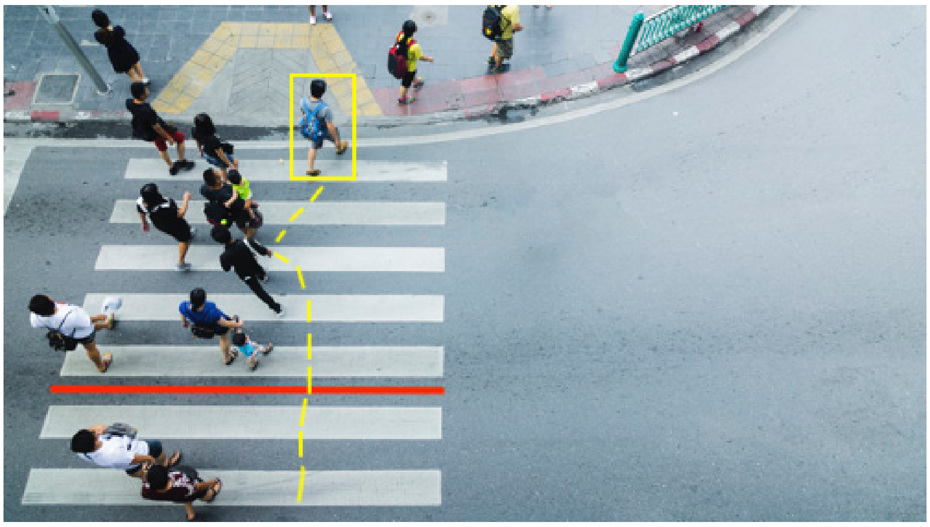

Fig. 1. An example of signal line for estimating the load of a crossing.

We propose a fully automatic people counting algorithm. The algorithm takes as input a video stream $\{F_i\}_{i=1}$ of frames captured by stationary camera and signal line that specified by an ordered pair of points $(L_a,\, L_b)$ on the frame. The output of the algorithm is a set of events $\{E_i\}_{i=1}$ that represented by triples of values $E_i = (k_i,\, r_i,\, d_i)$. The first value indicates the number of the frame where the signal line was crossed, the second value specifies the coordinates of the bounding box and the last value indicates the direction of the signal line intersection.

Our solution is an extension of the algorithm described in [1]. In this paper we propose the following improvements:

– use of the head detector instead of full-body detector;
– detection matching procedure is modified in order to work with small head bounding boxes;
– the use of body regression on the heads at different stages of tracking;
– an algorithm to automatically determine a region of interest (ROI) by signal line position is proposed to speed up detection.

2 Related Work

The modern methods of tracking are based on tracking-by-detection. There are a lot of ways to detect the desired object on the frame. Three most popular methods are: detection of body [2,3,4], detection of head [5,6] and using of key points. The first solution is popular because there are many datasets and ready-made solutions.

The head-tracking approach is well suited to track people in a crowd: usually video surveillance cameras are installed above the height of the person, where heads in a crowd can be seen better than full bodies. Heads are more resistant to overlapping than bodies. The number of ready-made solutions and data for training is less than for bodies. There are methods that use body parts detectors for tracking [7], key points of human pose [8], combined solutions (body and head) [9], and detector ensembles [10].

After detection we need to bind all the detections to tracks. As in the detection task, there are a lot of methods. The first group of algorithms for creating tracks is greedy algorithms. In most online algorithms tracks are constructed frame by frame, each frame creates a matrix of the cost of matching new detections and existing tracks, then the problem of matching is solved by a greedy algorithm (searching for the maximum in each row/column) [11,12] or by a Hungarian algorithm [13] [2,3,4]. Sometimes MCMC is used to bind all the detections to tracks [5,14].

Recently, neural networks have been used more often in tracking. For example, in paper [15] authors suggest using detector to obtain new detection by regression of detection on previous frame. However, this method has disadvantages. For example, it is able to work well only at a high frame rate and it also increases the load on the detector.

3 The Proposed Algorithm

3.1 Baseline

We use the solution from [1] as the baseline, which is an extension of SORT tracking algorithm [2]. We choose this algorithm as it is capable to work by detection on a sparse set of frames, which is performed on remote servers. This significantly reduces the amount of computational resources required for large-scale video surveillance systems (see fig. 2). Our proposed method inherits distributed nature of baseline.

Baseline works in online mode and use the Hungarian algorithm [13] to match detections. To improve results at a low detection rate ASMS visual tracking [16] is used to evaluate a speed of people between frames. The same approach with visual tracking speed estimation is used in [11].

The baseline algorithm consists of the following steps: (1) detection; (2) evaluation of the speed of detections using visual tracking; (3) prediction of the position of tracks by the Kalman filter; (4) matching; (5) extrapolation of the tracks; (6) detection of signal line crossing events.

Proposed improvements to some of the steps above are described below.

3.2 Detection

Since heads are seen better in the video and are less prone to occlusions we decided to use the head detector based on SSD [17] approach instead of the body detector. Another advantage of the head detector is that neural network body detectors can combine nearby people into one bounding box, which is less frequent for heads. The detector were trained on CrowdHuman [18] public dataset and on the dataset collected by Video Analysis Technologies. Experimental evaluation showed AUC of 0.66 on the test part of CrowdHuman. Usage of the head detector leads to a sub-task of restoring a bounding box of an entire body to find intersections of the signal line, which is located on the ground. We describe it in section 3.4.

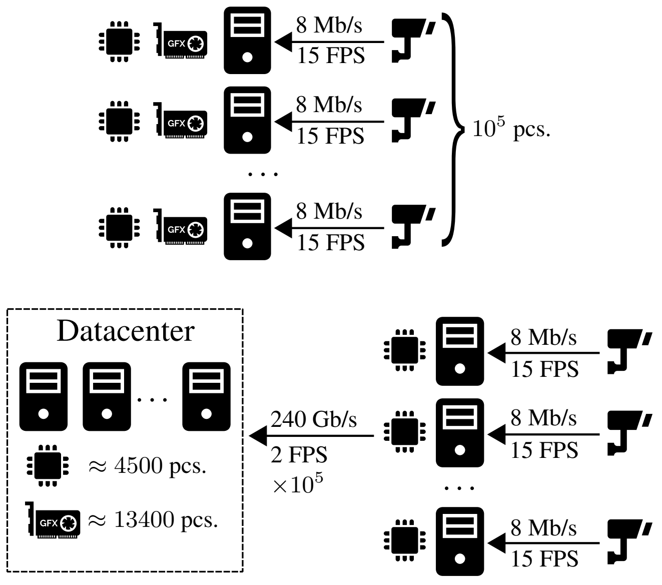

Fig. 2. Resources (CPUs, GPUs) required for large-scale video surveillance system (105 cameras) for (top) traditional tracking algorithms; (bottom) baseline algorithm with 2 Hz detection frequency.

3.3 Matching

During the experiments we realized that IOU metric used in the basic algorithm is not suitable for head matching. As head bounding boxes are small enough due to the errors in position and speed estimation bounding box pairs required to match are closely positioned, but have no intersection. IOU equal to zero acquired in the case which leads to track partitioning. Therefore we suggest to increase by s times the size of the bounding boxes (while saving the center of the bounding box) before matching them using IOU metric. This approach solves the problem described above.

3.4 Signal Line Crossing

We got the problem of missing signal line intersections as head tracks are located on a human height and the signal line is located on the ground. Problem reveals itself most on the scenes where the signal line is placed orthogonal to the camera. Therefore we offer two solutions: rising of signal lines and body bounding box regression by head.

Rising of Signal Lines We propose to raise the signal line to the level of the human head (see fig. 3). This solution reduces the computational cost and time of the algorithm as it doesn't require any extra steps to project head track to the ground plane.

The signal line can be raised either manually or automatically. Automatic approach we propose is using the following anatomical fact: the height of the head is $\frac{1}{8}$ of the height of the entire human body. We can use a body detector to calculate average height of human body for the scene. And raise the signal line to a height equal to $\frac{7}{8}$ of the average height of a person on the scene.

This approach has a drawback: people's growth is different, which means that people's heads are on different planes, while people's legs are always on the same plane. Therefore, the raised signal line doesn't look so clearly since it's not clear where the plane is located.

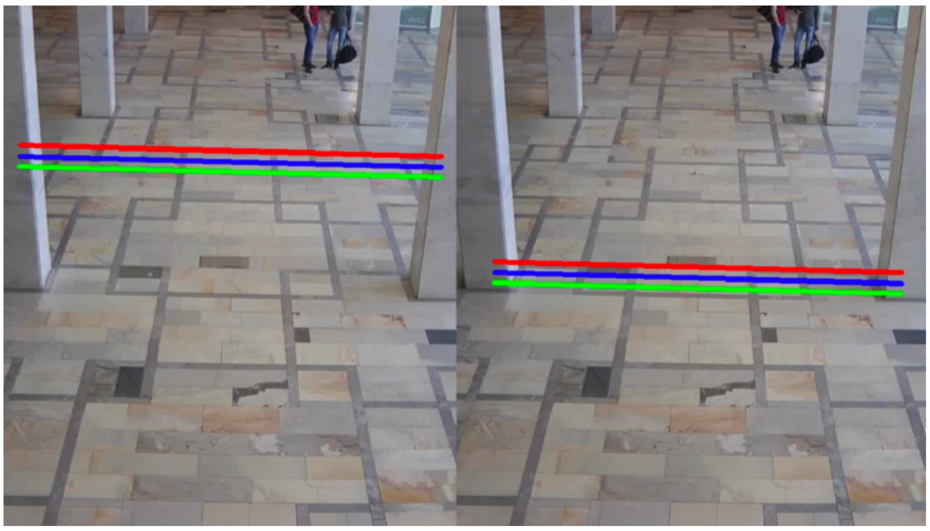
Fig. 3. Raised signal line (left picture) and signal line on a ground plane (right picture).

Bounding Box Regression by Head To solve the drawbacks that arose in the previous paragraph we propose to use regression of a body by a head, which is trained for the specific scene. Our idea consists of two stages: combining the detections of heads and bodies for the scene and training linear regression on the received data.

Combining head and body detections First of all we launch head and body detectors. To combine heads and bodies a matrix of head and body correspondence is built on each frame.

$$CM_{k,\,i,\,j} = cost_{match}(B_{k,\,i},\, H_{k,\,j}),\; i \in [1,\, nb_k],\; j \in [1, nh_k] \tag{1}$$

In eq. (1) nb_k, nh_k – the number of body and head detections on the k-th frame, $B_{k,\,i}$, $H_{k,\,j}$ – the bounding box of the i-th body and j-th head on the k-th frame. We calculate matching cost as:

$$cost_{match}(B_{k,i},\, H_{k,j}) = \begin{cases} ioh_{k,i,j}, \textbf{if } ioh_{k,i,j} \geq I_1 \,\wedge\, \tau_1 \leq dist(B_{k,i},\, H_{k,j}) \leq \tau_2 \\ 0, \textbf{ else} \end{cases} \tag{2}$$

$$ioh_{k,i,j} = \frac{area\,(intersection(B_{k,i},\, H_{k,j}))}{area(H_{k,j})} \tag{3}$$

$$dist(B_{k,i},\, H_{k,j}) = \frac{center_y(H_{k,j}) - top(B_{k,i})}{height(H_{k,j})} \tag{4}$$

In eq. (2) I_1 is threshold for $ioh_{k,i,j}$ (we are using $I_1 = 0.5$), $\tau_1,\, \tau_2$ – minimum and maximum normalized distances between the center of the head and the upper point of the body (we are using $\tau_1 = -1$ and $\tau_2 = 1$). So at least half of head bounding box area should be inside body bounding box area and vertically head center shouldn't be far from body top by at least one head height.

Next the assignment problem is solved using the Hungarian algorithm to maximize cost. Combined head and body detections with non-zero match cost form training dataset for a regression (see fig. 4).

Linear regression After the previous step we have the data to train a linear regression. The training dataset consists of head and body bounding box pairs: $(B_i,\, H_i)$. The regressor predicts the following values: $(B_h,\, B_w,\, shift_x)$, where $B_h,\, B_w$ – height and width of the body bounding box, $shift_x$ – a shift from the center of the head to the center of the predicted body normalized by head width:

$$shift_x = \frac{center_x(B) - center_x(H)}{H_w} \tag{5}$$

Prediction is done by linear regression with quadratic members:

$$\widehat{B_h} = k_0 + k_1 H_x + k_2 H_y + k_3 H_w + k_4 H_h + k_5 H_x^2 + k_6 H_x H_y + \\ \ldots + k_{13} H_h H_w + k_{14} H_h^2 \tag{6}$$

The same way $\widehat{B_w},\, \widehat{shift_x}$ are predicted, but with separate sets of coefficients. After learning coefficients of linear regression on training dataset, we use predicted value to restore body bounding box:

$$\widehat{B_x} = center_x(H) - \frac{\widehat{B_w}}{2} + shift_x H_w \tag{7}$$

$$\widehat{B_y} = H_y \tag{8}$$

This approach helps us keep the signal line on the ground plane, which solves drawbacks of the previous approach.

Now we have a choice to do speed estimation by visual tracking with head or regressed body detections. As heads are less prone to occlusions visual tracking of them may be more reliable. But body bounding boxes are several times larger and have more visual information to track. We check both choices on experimental evaluation.

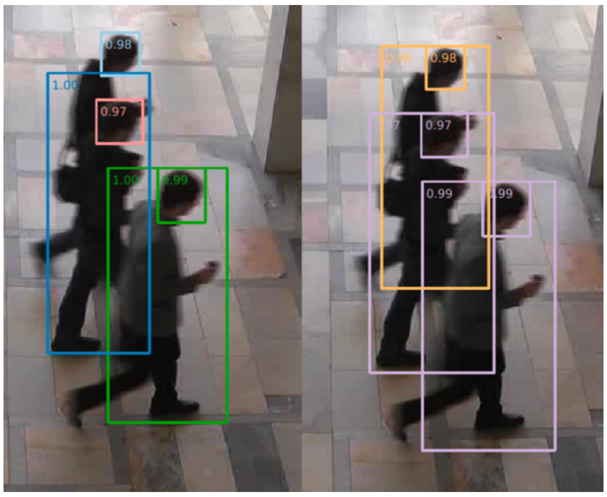

Fig. 4. Combined detections (left picture): head without pair (pink, blue), body without pair (dark blue), combined head and body (green). Results of body regression (right picture).

3.5 Limiting the Detection Area

The entire frame is not required to find events of crossing the signal line. We can limit the detection area to the area around the signal line – region of interest (ROI). This solution speeds up the SSD detector as it is slower for images with higher resolution.

If the signal line is horizontal then the ROI is located on the top and bottom of the line. Otherwise the ROI is located on the left and right side of the line, as well as on the top of the line.

Let be $\alpha \leq \frac{\pi}{2}$ – the minimum angle between signal line and horizon, w_{mean}, h_{mean} – the average width and height of body detections and $(x_a,\, y_a)$, $(x_b,\, y_b)$ – the coordinates of the beginning and the end of the signal line.

$$x_1 = min(x_a,\, x_b) - s_w w_{mean}\left(1 + \frac{sin\,\alpha}{2}\right) \tag{9}$$

$$x_2 = max(x_a,\, x_b) + s_w w_{mean}\left(1 + \frac{sin\,\alpha}{2}\right) \tag{10}$$

$$y_1 = min(y_a,\, y_b) - s_h h_{mean} cos\,\alpha \tag{11}$$

$$y_2 = max(y_a,\, y_b) + s_h h_{mean} cos\,\alpha \tag{12}$$

Then $A = (x_1, y_1, x_2, y_2)$ is the region of interest. s_w, s_h in equations 9-12 are parameters. After visual testing we selected the following values for these variables: $s_w = 2$, $s_h = 1$.

4 Experimental Evaluation

4.1 Datasets

For an experimental evaluation of our algorithm we need datasets filmed by static camera with body tracks markup. Video sequences should be long enough to evaluate people counting quality. If dataset provides head tracks markup it allows us to check effect of body regression by comparing with head tracking and raised signal lines (section 3.4). Most of the public datasets including popular MOTChallenge dataset [19] have short videos, only body tracks markup or filmed by moving camera. So we used 19 videos from the collection of the Video Analysis Technologies company and the Towncentre dataset [5] to test our algorithm. For all videos signal lines were manually drawn at ground level as well as at head level. The table 1 provides detailed information about each test video.

Table 1. Videos that we used to test the algorithm.

Video name	Duration	Format	Number of events
SignalineBase/02	00:17:40	1280x720@25	254
SignalineBase/03	00:14:40	1280x720@25	134
SignalineBase/04	00:05:20	1280x720@25	69
SignalineBase/05	00:11:40	1280x720@25	172
SignalineBase/06	00:12:00	1280x720@25	219
SignalineBase/07	00:08:29	800x450@25	124
SignalineBase/10	00:38:59	704x576@25	120
SignalineBase/11	00:22:48	704x576@25	91
SignalineBase/13	00:05:01	720x576@12	9
SignalineBase/14	00:05:23	720x576@12	17
SignalineBase/17	00:16:42	1280x1280@20	48
SignalineBase/18	00:16:26	1280x1000@25	180
SignalineBase/20	00:15:00	1024x768@20	144
SignalineBase/21	00:19:40	800x600@25	149
SignalineBase/26	00:10:29	1280x720@15	90
SignalineBase/28	00:11:40	1280x960@25	72
SignalineBase/29	00:40:00	704x576@25	159
SignalineBase/31	00:29:10	1920x1080@25	381
Towncentre	00:03:00	1920x1080@25	162

4.2 Metrics

As a quality metric we use the average error of counting the number of intersections (events) [1]. The resulting events can include both true and false ones. The false events have no correspondences in the reference labeling. We say that an event E_i in the input set of data matches the event $\widehat{E_i}$ in the reference labeling if they correspond to the same person crossing the signal line at the same time. We match all events as described in the paper [1].

After the events have been matched we divide videos to segments with 10 reference events and calculate the following characteristics on them:

- GT_{seg} is the number of reference events on the segment;
- FP_{seg} is the number of unmatched events from the algorithm on the segment;
- FN_{seg} is the number of unmatched events from the reference events on the segment;
- $E_{seg} = \frac{FP_{seg} - FN_{seg}}{GT_{seg}}$ is an error on the segment.

Then final error is calculated as $E = \sum_{i=1}^{N} \frac{E_{seg}}{N}$, where N is the number of the segments.

4.3 Experimental Results

Rising of Signal Lines At first we have tested baseline with modifications proposed in sections 3.2, 3.3 and manually raised signal lines as described in section 3.4. The algorithm is marked as **heads-no-regression** in the results table (see table 2) and parameter s shows how many times head bounding boxes have been increased.

Table 2. Results of experiments with rising the signal lines. The table shows counting error E in percents for the algorithms depending on the detection frequency.

Algorithm / Detection frequency	5 Hz	3 Hz	2 Hz	4/3 Hz
SORT [2]	8.3	10.8	17.3	-
baseline [1]	7.5	7.5	7.5	8.1
heads-no-regression $s = 1$	4.6	5.5	8.2	9.3
heads-no-regression $s = 2$	3.9	4.5	5.1	6.5

Experiments with rising the signal line clearly shows the advantage of the head detector over the body detector. The head detector gives a significant increase in an accuracy reducing the error by ≈ 2 times. The algorithm without matching modification ($s = 1$) has poor results at low FPS due to the previously voiced problems which shows importance of the modification proposed in the section 3.3.

Body Bounding Box Regression by Head Next we have tested body bounding box regression by head. As mentioned in section 3.4 there are two alternative ways to apply it. There are two algorithms we have tested (see table 3):

- **heads-regression-vistrk** — baseline with modifications proposed in sections 3.2, 3.3, visual tracking of regressed bodies to estimate speed and signal lines on the ground plane (section 3.4);

– **heads-vistrk-regression** — baseline with modifications proposed in sections 3.2, 3.3, visual tracking of heads to estimate speed and signal lines on the ground plane (section 3.4).

Table 3. Results of experiments with bounding box regression by head. The table shows counting error E in percents for the algorithms depending on the detection frequency.

Algorithm / Detection frequency	5 Hz	3 Hz	2 Hz	4/3 Hz
baseline [1]	7.5	7.5	7.5	8.1
heads-regression-vistrk	4.5	4.9	5.7	6.8
heads-vistrk-regression	4.1	4.1	4.3	5.9
heads-no-regression	3.9	4.5	5.1	6.5

As you can see the second configuration gives better results. Visualization showed us that visual tracking of heads is more reliable as they are less prone to occlusions. It worth noting that **heads-vistrk-regression** performed better than **heads-no-regression** ($s = 2$) on low detection frequency.

Limiting the Detection Area Next we tested limiting of the detection area (section 3.5). It allowed us to increase the speed of the algorithm almost without affecting the counting error (see table 4). Detection area reduced by almost 60% on some of the videos.

Table 4. Results of experiments with limiting the detection area. The table shows counting error E in percents for the algorithms depending on the detection frequency.

Algorithm / Detection frequency	5 Hz	3 Hz	2 Hz	4/3 Hz
baseline + detection ROI	7.6	7.6	7.6	8.3
heads-regression-vistrk + detection ROI	4.8	4.9	5.8	7.0
heads-vistrk-regression + detection ROI	4.0	4.0	4.3	5.9

5 Conclusion

We have proposed the algorithm of counting people in a video, which is an extension of the algorithm described in [1]. Usage of the head detector and body bounding box regression allowed to increase a counting accuracy. Detection area limiting saves computational resources. Comparing with the baseline the proposed method is able to work on a lower detection frequency showing lower counting error.

For the future work we plan to use automatic camera calibration algorithms (see fig. 5) [20,21]. This will allow to perform person tracking on the ground map and further improve accuracy of people counting.

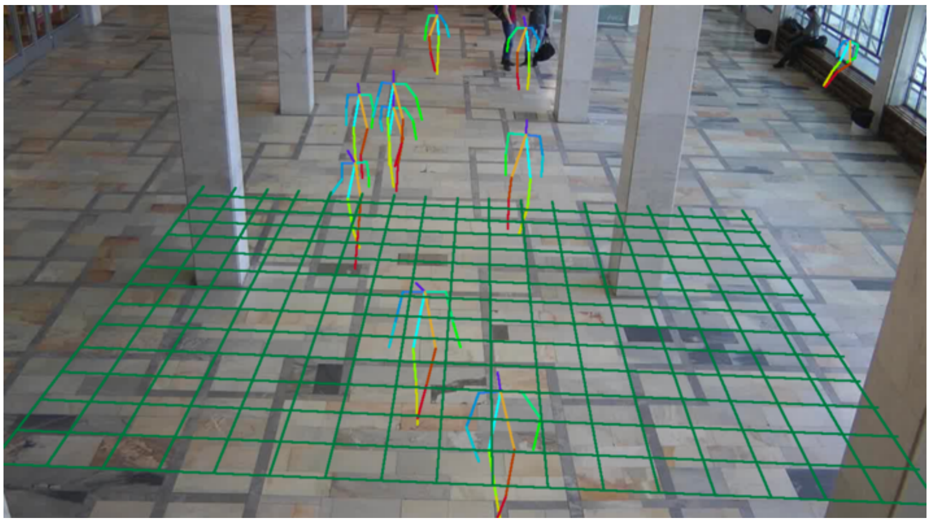

Fig. 5. Visualization of the ground plane restored from the calibration, estimated from the human poses from the video.

References

1. Kuplyakov, D.A., Shalnov, E.V., Konushin, V.S., Konushin, A.S.: A Distributed Tracking Algorithm for Counting People in Video. Programming and Computer Software 45(4), 163-170 (2019)
2. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple Online and Realtime Tracking. CoRR abs/1602.00763 (2016)
3. Wojke, N., Bewley, A., Paulus, D.: Simple Online and Realtime Tracking with a Deep Association Metric. ArXiv e-prints (2017)
4. Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., Yan, J.: Poi: Multiple object tracking with high performance detection and appearance feature. In: European Conference on Computer Vision, pp. 36-42 (2016)
5. Benfold, B., Reid, I.: Stable Multi-Target Tracking in Real-Time Surveillance Video. In: CVPR, pp. 3457-3464 (2011)
6. Shalnov, E., Konushin, V., Konushin, A.: An improvement on an MCMC-based video tracking algorithm. Pattern Recognition and Image Analysis 25, 532-540 (2015)
7. Shu, G., Dehghan, A., Oreifej, O., Hand, E., Shah, M.: Part-based multiple-person tracking with partial occlusion handling. In: CVPR, pp. 1815-1821. IEEE Computer Society (2012)
8. Xiu, Y., Li, J., Wang, H., Fang, Y., Lu, C.: Pose Flow: Efficient Online Pose Tracking. In: (2018)
9. Henschel, R., Leal-Taixe, L., Cremers, D., Rosenhahn, B.: Improvements to Frank-Wolfe optimization for multi-detector multi-object tracking. CoRR abs/1705.08314 (2017)
10. Cobos, R., Hernandez, J., Abad, A.: A fast multi-object tracking system using an object detector ensemble. (2019)
11. Bochinski, E., Senst, T., Sikora, T.: Extending IOU Based Multi-Object Tracking by Visual Information. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1-6 (2018)

12. Bochinski, E., Eiselein, V., Sikora, T.: High-Speed tracking-by-detection without using image information. In: Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on, pp. 1-6 (2017)
13. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics (NRL) 2(1-2), 83-97 (1955)
14. Kuplyakov, D., Shalnov, E., Konushin, A.: Markov chain Monte Carlo based video tracking algorithm. Programming and Computer Software 43(4), 224-229 (2017)
15. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking Without Bells and Whistles. In: The IEEE International Conference on Computer Vision (ICCV), (2019)
16. Vojir, T., Noskova, J., Matas, J.: Robust scale-adaptive mean-shift for tracking. In: Scandinavian Conference on Image Analysis, pp. 652-663 (2013)
17. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. Lecture Notes in Computer Science (2016)
18. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: CrowdHuman: A Benchmark for Detecting Human in a Crowd. arXiv preprint arXiv:1805.00123 (2018)
19. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixe, L.: MOT20: A benchmark for multi object tracking in crowded scenes. arXiv:2003.09003[cs] (2020)
20. Shalnov, E.V., Konushin, A.S., Konushin, V.S.: Convolutional neural network for camera pose estimation from object detections. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 42(2-W4), 1-6 (2017)
21. 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 42(6), 361-366 (2016)